Horizon 2020
European Union funding
for Research & Innovation

Big Data technologies and extreme-scale analytics

MARVEL

Multimodal Extreme Scale Data Analytics for Smart Cities Environments

D1.3: Architecture definition for MARVEL framework[†]

Abstract: The purpose of this deliverable is to provide a refined specification of the conceptual architecture for the MARVEL Edge-to-Fog-to-Cloud (E2F2C) ubiquitous computing framework. The document describes all components of the MARVEL framework, focusing on their roles and interactions within the framework. Specifications of the architecture to the MARVEL use cases have also been provided together with initial components' customisations. Finally, the document also includes mappings to the relevant Big Data, AI, and Fog computing reference architectures. The results of this deliverable will serve as a guideline towards the MARVEL Minimum Viable Product (MVP). This deliverable is a living document that will be revised over the course of the project to account for the latest project achievements and updates of the relevant reference architectures and guidelines in the relevant project areas.

Contractual Date of Delivery	31/08/2021
Actual Date of Delivery	31/08/2021
Deliverable Security Class	Public
Editors	*Dragana Bajovic, Nikola Simic (UNS)*
Contributors	All *MARVEL* partners
Quality Assurance	*Toni Heittola (TAU)* *Claudio Cicconetti (CNR)*

[†] *The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957337.*

The *MARVEL* Consortium

Part. No.	Participant organisation name	Participant Short Name	Role	Country
1	FOUNDATION FOR RESEARCH AND TECHNOLOGY HELLAS	FORTH	Coordinator	EL
2	INFINEON TECHNOLOGIES AG	IFAG	Principal Contractor	DE
3	AARHUS UNIVERSITET	AU	Principal Contractor	DK
4	ATOS SPAIN SA	ATOS	Principal Contractor	ES
5	CONSIGLIO NAZIONALE DELLE RICERCHE	CNR	Principal Contractor	IT
6	INTRASOFT INTERNATIONAL S.A.	INTRA	Principal Contractor	LU
7	FONDAZIONE BRUNO KESSLER	FBK	Principal Contractor	IT
8	AUDEERING GMBH	AUD	Principal Contractor	DE
9	TAMPERE UNIVERSITY	TAU	Principal Contractor	FI
10	PRIVANOVA SAS	PN	Principal Contractor	FR
11	SPHYNX TECHNOLOGY SOLUTIONS AG	STS	Principal Contractor	CH
12	COMUNE DI TRENTO	MT	Principal Contractor	IT
13	UNIVERZITET U NOVOM SADU FAKULTET TEHNICKIH NAUKA	UNS	Principal Contractor	RS
14	INFORMATION TECHNOLOGY FOR MARKET LEADERSHIP	ITML	Principal Contractor	EL
15	GREENROADS LIMITED	GRN	Principal Contractor	MT
16	ZELUS IKE	ZELUS	Principal Contractor	EL
17	INSTYTUT CHEMII BIOORGANICZNEJ POLSKIEJ AKADEMII NAUK	PSNC	Principal Contractor	PL

Document Revisions & Quality Assurance

Internal Reviewers

1. *Toni Heittola, (TAU)*
2. *Claudio Cicconetti, (CNR)*

Revisions

Version	Date	By	Overview
1.0.0	31/08/2021	Editors	Addressed comments from the PC.
0.8.0	28/08/2021	Editors	Addressed comments from IR1 and IR2.
0.7.1	27/08/2021	IR1 (TAU), IR2 (CNR)	Comments from IR1 TAU Comments from IR2 CNR
0.7.0	26/08/2021	Editors	Addressed comments from IR1 and IR2.
0.6.2	23/08/2021	IR2 (CNR)	Comments from IR2 CNR
0.6.1	20/08/2021	IR1 (TAU)	Comments from IR1 TAU
0.6.0	17/08/2021	Editors	Phase 3 consolidated version (for internal review).
0.5.2	04/08/2021	Editors	Phase 2 consolidated version.
0.5.1	16/07/2021	Editors	Phase 1 consolidated version.
0.5.0	18/06/2021	Editors	3rd and final version of the ToC.
0.4.2	23/06/2021	All partners	Comments on the 2nd version of the ToC.
0.4.0	17/06/2021	Editors	2nd version of the ToC.
0.3.1	16/06/2021	WPL (TAU)	Comments on the ToC.
0.3.0	08/06/2021	Editors	ToC.

Disclaimer

The work described in this document has been conducted within the MARVEL project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957337. This document does not reflect the opinion of the European Union, and the European Union is not responsible for any use that might be made of the information contained therein.

This document contains information that is proprietary to the MARVEL Consortium partners. Neither this document nor the information contained herein shall be used, duplicated or communicated by any means to any third party, in whole or in parts, except with prior written consent of the MARVEL Consortium.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AAC	Automated Audio Captioning
AI	Artificial Intelligence
AMP	Amplifier
API	Application Programming Interface
ASC	Acoustic Scene Classification
ASR	Automatic Speech Recognition
AudioAnony	GANs for audio anonymisation
AV	Audio-Visual
AVAD	Audio-Visual Anomaly Detection
AVCC	Audio-Visual Crowd Counting
AVDrone	Drone-based Audio-Visual data collection
BDVA	Big Data Value Association
BT	BlueTooth
CATFlow	Data acquisition Framework
CCTV	Closed-Circuit Television
CPU	Central Processing Unit
DatAna	Data acquisition Framework
devAIce	audio AI technology integrated in a device
D#.#	Deliverable
DFA	Deterministic Finite-state Automata
DFB	Data Fusion Bus
DL	Deep Learning
DMD	Data Management and Distribution
DMP	Data Management Platform
DNN	Deep Neural Network
DoA	Description of Action
DynHP	Compressed models
EC	European Commission
EdgeSec	Security services at the edge
ELK	Elasticsearch Logstash Kibana
ESR	Ethics Summary Report
ETL	Extract, Transform, Load

EU	European Union
E2F2C	Edge-to-Fog-to-Cloud
FedL	Framework and implementation of ML algorithms – Federated learning
FL	Federated Learning
FoV	Field of View
GA	Grant Agreement
GAN	Generative Adversarial Network
GPS	Global Positioning System
GPGPU	General-Purpose Graphics Processing Unit
GPU	Graphics Processing Unit
GPURegex	GPU Pattern Matching Framework
HDD	Hierarchical Data Distribution
HDFS	Hadoop Distributed File System
HEI	Higher Education Institution
HLS	HTTP Live Streaming
HPC	High-Performance Computing
HTTPS	Hypertext Transfer Protocol Secure
HW	Hardware
H2020	Horizon 2020
IAM	Identity and Access Management
ICT	Information Communications Technology
ID	Identifier
IdP	Identity Provider
IoT	Internet of Things
ISO	International Organisation for Standardisation
IT	Information Technology
JMS	Java Message Service
JSON	JavaScript Object Notation
Karvdash	Kubernetes CARV dashboard
KPI	Key Performance Indicator
LoRa	Long Range
M#	Month #
MCU	Microcontroller Unit
MEMS	Micro-Electro-Mechanical Systems
ML	Machine Learning

MQTT	Message Queuing Telemetry Transport
NFS	Network File System
NIST	National Institute of Standards and Technology
NVMe	Non-Volatile Memory Express
openSMILE	open-source Speech and Music Interpretation by Large-space Extraction
PCB	Printed Circuit Board
PDM	Pulse Density Modulation
PSoC	Programmable System-on-Chip
RAM	Random Access Memory
RF	Radio Frequency
RPi	Raspberry Pi
SDIO	Secure Digital Input Output
SDK	Software Development Kit
SELD	Sound Event Localisation and Detection
SendMiner	sensAI Voice-Activity-Detection
SED	Sound Event Detection
SED@Edge	Sound Event Detection at the Edge
SGX	Software Guard Extensions
SLA	Service Level Agreement
SLURM	Simple Linux Utility for Resource Management
SmartViz	Advanced visualisation toolkit
SNR	Signal-to-Noise Ratio
SotA	State-of-the-Art
SQL	Structured Query Language
SSD	Solid State Drive
SSH	Secure Shell
SSL	Secure Sockets Layer
SSO	Single Sign-on
SW	Software
S2S	Site-to-Site
S3	Simple Storage Service
TRL	Technology Readiness Level
TTS	Text-to-Speech
T#.#	Task #.#
UART	Universal Asynchronous Receiver-Transmitter

VAD	Voice Activity Detection
ViAD	Visual Anomaly Detection
VCC	Visual Crowd Counting
VVV	Volume, Velocity, and Variety
VideoAnony	GANs for video anonymisation
WiFi	Wireless Fidelity
WP	Work Package

Executive Summary

The purpose of this deliverable is to provide a revised version of the MARVEL conceptual architecture, with all MARVEL components and assets described and connected into a common MARVEL framework. The deliverable has been developed within the scope of WP1 – Project setup of the MARVEL project under Grant Agreement No. 957337.

Towards motivating the MARVEL framework and architecture, we first reviewed and analysed functional and non-functional end-user requirements collected in D1.2 – "MARVEL's Experimental protocol" [1]. This bottom-up approach allowed to clearly identify how each of the MARVEL components, and their mutual interactions, map to and address the requirements, explaining and motivating their roles within the overall framework.

The deliverable presents the current and the expected TRLs and outlines the key innovations across all MARVEL components, serving as an innovations roadmap for the MARVEL framework.

Based on the roles' functional similarity, MARVEL components are organised into seven subsystems: 1) Sensing and perception subsystem, 2) Security, privacy, and data protection subsystem, 3) Data management and distribution subsystem, 4) Audio, visual, and multimodal AI subsystem, 5) Optimised E2F2C processing and deployment subsystem, 6) E2F2C infrastructure, and 7) System outputs: User interactions and the decision-making toolkit. Within the identified subsystems, each of the components was then described in full detail, with explanations of their inner workings, inbound and outbound interfaces, and accompanying illustrative figures. Subsystems are described as a synergy of the participating components, focusing on high-level subsystem roles.

User interactions and user interface are addressed both for the overall framework and for each component separately. Component-wise user interactions include considerations of both IT and non-IT users and address component instantiation (configuration and initialization) and also access rights and procedures (authentication and authorisation). Configuration and initialization aspects are discussed with regards to IT users, while authentication and authorisation aspects concern both IT and non-IT users.

Towards future deployments, starting with the Minimum Viable Product (MVP) at M12, the deliverable provides architecture specifications for each of the MARVEL use cases. This is achieved through a detailed listing of participating MARVEL components in each use case, including use case-targeted customisation, wherever possible. Initial allocations of the HPC resources, including management and orchestration, have also been reported for each use case. Towards a more refined definition of the MARVEL use case architectures, the deliverable also provides the per-use case instantiations of the architectural blueprint developed within the DataBench project, preparing the ground for future benchmarking tasks within the project.

Finally, the deliverable provides mappings of the MARVEL conceptual architecture to the relevant Big Data, AI, and Fog computing reference architectures, establishing a bridge to efficiently account for further developments in the relevant project areas - Big Data, AI, Continuum computing, etc., and future EU strategic agendas in the respective domains.

1 Introduction

1.1 Purpose and scope

The purpose of this deliverable is to describe in detail the refined specification of the MARVEL conceptual Edge-to-Fog-to-Cloud (E2F2C) architecture. This revision is grounded on a thorough understanding of the underlying technologies, an updated State-of-the-Art (SotA) review in the relevant project areas, alignment with relevant reference architectures and models, as well as on end-user requirements. Specifically, the deliverable defines each of the MARVEL components in terms of its role, functionality, and inbound and outbound interfaces, breaking the technological silos and connecting the components together into a common computing MARVEL framework.

Variations in the architecture for each use case are specified by: (i) defining use case-components mappings, including details on the component application in each specific use case; and by (ii) defining data value chain specifications using the Architectural blueprint of the DataBench project [2]. HPC customisation for different use case experiments and framework executions are specified.

The deliverable presents the current and the expected TRLs and outlines the key innovations across all MARVEL components, serving as an innovations roadmap for the MARVEL framework. Mappings to the relevant reference architectures, including Big Data Reference Architecture proposed by BDVA [3], the European AI, Data and Robotics Framework and Enablers [4], and the NIST Fog Computing Conceptual Model (Edge-Fog computing) [5] are also provided, linking the MARVEL architecture to the SotA big data and E2F2C reference architectures.

1.2 Intended readership

Deliverable D1.3 – 'Architecture definition for the MARVEL framework' is a public document. The content found in this document aims to guide partners towards the realisation, integration, and deployment of the MARVEL framework and its application in the MARVEL use cases. Similarly, it can be used as an introductory document providing guidelines for future users of the MARVEL framework. Finally, it can also be used as a reference document by the external readers developing similar frameworks, or similar constituting subsystems/components.

1.3 Relation to other work packages, deliverables and activities

This document is the main output of **Task 1.4. – 'Technology convergence: specifications and E2F2C distributed architecture'** of WP1 of the Description of Action (DoA):

> **WP1 - Setting the scene: Project set up**. This WP materialises the Baseline Phase of the project and sets the basis for the realisation of all the Pillars of the project. Among the objectives of the WP are the description of the project's specifications and E2F2C distributed architecture and specification of MARVEL guidelines and protocols for achieving responsible AI at all project levels.

The document builds on the work and the outputs of the preceding tasks in WP1:

- State-of-the-art surveys of **Task 1.1. The critical role of multimodal analytics in addressing societal challenges** – towards understanding better the end-user needs in terms of multimodal analytics in smart cities;

- State-of-the-art surveys of **Task 1.2. Extreme-scale multimodal analytics: progress beyond the state-of-the-art** – for understanding the key challenges and the gaps in the relevant scientific and technological MARVEL areas that need to be addressed;
- End-user requirements from **Task 1.3. Experimental protocol - real-life societal trial cases in smart cities environments** – as the indicative targets for functionalities and KPIs that the platform design should achieve.

This document is also in direct relation to **Task 5.3. Continuous integration towards MARVEL's framework realisation** of WP5:

> **WP5 - Infrastructure management and integration.** The main objective of this WP is to ensure successful E2F2C framework delivery for distributed extreme-scale audio analytics, with provision and configuration of HPC infrastructure, orchestration of resource management, seamless integration of MARVEL services, and quantifiable progress against societal, academic and industry-validated benchmarks together with continuous alignment with the responsible AI guidelines.

In particular, the outputs of D1.3 – components and overall framework definition, and also use case architecture specification, serve as a baseline for developing the **Minimum Viable Product (MVP) (M12)** within Task 5.3.

Regarding relation to the project objectives, this document is one of the key pillars of Objective 3:

> **Objective 3:** Break technological silos, converge very diverse and novel engineering paradigms, and establish a distributed and secure Edge-to-Fog-to-Cloud (E2F2C) ubiquitous computing framework in the big data value chain.

By its nature, the document also contributes to all of the remaining objectives of the MARVEL project, by defining in detail the relevant MARVEL components, each contributing to specific project objectives, components' interactions, and their applications in MARVEL use cases.

1.3.1 MARVEL use cases and datasets

Table 1 presents MARVEL use cases and the associated datasets across the three MARVEL pilots/experiments: GRN (pilot), MT (pilot), and UNS (experiments)[1], for later reference.

Table 1: MARVEL use cases and datasets

No.	Use case	MARVEL dataset	Partner owner
GRN1	Safer roads	*GRN-AV-traffic-entity* *GRN-TXT-traffic-data*	GRN (Road Traffic in cities)
GRN2	Road User Behaviour	*GRN-AV-traffic-entity* *GRN-AV-traffic-state* *GRN-TXT-traffic-data*	GRN (Road Traffic in cities)
GRN3	Traffic Conditions and Anomalous Events	*GRN-AV-traffic-entity* *GRN-TXT-traffic-data*	GRN (Road Traffic in cities)

[1] UNS is a Higher Education Institution (HEI) providing experimental use cases for MARVEL.

		GRN-AV-traffic-state	
GRN4	Junction Traffic Trajectory Collection	*GRN-AV-traffic-entity* *GRN-TXT-traffic-data*	GRN (Road Traffic in cities)
MT1	Monitoring of Crowded Areas	*TrentoOutdoor – Real Dataset*	MT (City surveillance)
MT2	Detecting Criminal and Anti-Social Behaviours	*TrentoOutdoor – Real Dataset* *TrentoOutdoor – Staged Dataset*	MT (City surveillance) FBK (staged recording)
MT3	Monitoring of Parking Places	*TrentoOutdoor – Real Dataset* *TrentoOutdoor – Staged Dataset*	MT (City surveillance) FBK (staged recording)
MT4	Analysis of a Specific Area	*TrentoOutdoor – Real Dataset*	MT (City surveillance)
UNS1	Drone Experiment	*UNS Drone*	UNS (Crowd monitoring)
UNS2	Audio-Visual Emotion Recognition	*UNS Audio-Video Emotion*	UNS (Crowd monitoring/ Security)

1.4 Structure of the report

The structure of this report is as follows:

- Section 2 provides the rationale and motivation behind the MARVEL conceptual architecture by analysing requirements identified within the MARVEL project.
- Sections 3-9 provide descriptions of MARVEL subsystems and their individual components;
  - Section 3 describes the Sensing and perception subsystem;
  - Section 4 describes the Security, privacy, and data protection subsystem;
  - Section 5 describes the Data management and distribution subsystem;
  - Section 6 describes the Audio, visual, and multimodal AI subsystem;
  - Section 7 describes the Optimised E2F2C processing and deployment subsystem;
  - Section 8 describes the E2F2C infrastructure; and
  - Section 9 describes the System outputs: User interactions and the decision-making toolkit.
- Section 10 discusses user interfaces of the MARVEL framework.
- Section 11 presents mappings of the MARVEL conceptual architecture to different Big Data and E2F2C reference architectures.
- Section 12 outlines architecture specifications and specialisations across the MARVEL use cases.
- Section 13 concludes the deliverable.

2 MARVEL architecture overview

2.1 Rationale, motivation, and MARVEL use case requirements

MARVEL will evaluate the technological development provided by the partners in ten use cases, addressing a variety of real-life aspects in smart city scenarios. Among them, eight trial cases will be implemented in real settings, i.e., using daily capturing from existing city sensors, led by two pilot partners GRN and MT, while the remaining two use cases led by UNS will be implemented in controlled environments.

The MARVEL architecture is comprehensive and can be specialised to each of the ten use cases, wherein the specialisation involves a subset of the components. The specialisation is dependent on both the availability of the sensors and other relevant hardware as well as the actual usefulness of a given technology under the particular context (see Tables 7-16 and D1.2 for details [1]).

The ten use cases can be split into two types: **traffic-related** and **person-related**. The former includes all use cases led by GRN, i.e., GRN1-4, together with two use cases, i.e., MT3 and MT4 led by MT. All the other use cases address people, as a crowd or as individuals, without using any identifiable information.

MARVEL's use cases can be further categorised into two groups based on the time-span they cover and the corresponding reaction time: **real-time monitoring** and **long-term analytics**. The former requires a prompt reaction from the public authorities and includes use cases GRN1, GRN3, MT1, MT2, MT3, UNS1, and UNS2. The latter group of use cases, i.e., GRN2, GRN4, and MT4, focuses on collecting information in the long-term to help improve either mobility or security.

Following a detailed analysis of the MARVEL use cases conducted within Task 1.3 – 'Experimental protocol - real-life societal trial cases in smart cities environments' and reported in D1.2 – 'MARVEL's experimental protocol', Table 2 provides consolidated MARVEL system requirements. (See ahead also Table 4 for a detailed mapping between the requirements and the architecture.)

Table 2: Consolidated MARVEL system requirements

No	System requirements and the associated use case KPIs (FR) /use cases (NFR)	Relevant use cases
	Functional requirements	
FR1	*Increased accuracy of decision-making* • Detection of cyclists – GRN-KPI1 • Detection of driver actions – GRN-KPI3 • Detection of road obstruction – GRN-KPI4 • Detection of events in crowds – MT-KPI1 • Detection of antisocial/illegal/anomalous events – MT-KPI4 • Detection of targeted events at parking places – MT-KPI5 • Audio-visual crowd anomaly detection – UNS-KPI1 • Audio-visual emotion recognition – UNS-KPI5	GRN1-4, MT1-4
FR2	*Decreased detection/reaction time* • Low latency in detecting cyclists – GRN-KPI2 • Obstruction detection time – GRN-KPI5	GRN1. GRN3, MT1, MT2,

	• Detection time for events in crowds – MT-KPI2 • Reaction time for antisocial/illegal behaviour – MT-KPI3 • Detection time of targeted events at parking places – MT-KPI6 • Reduction of audio-visual crowd anomaly detection time – UNS-KPI3 • Reduction of reaction time in AV emotion recognition – UNS-KPI7	UNS1
FR3	*Increased system robustness* • Detecting cyclists at any time of day – GRN-KPI1 • Robustness to operating conditions (e.g., light, distance) – UNS-KPI2	GRN1, UNS1
FR4	*Hidden patterns revealed, insights* • Collection of trajectories of pedestrians and vehicles – GRN-KPI6 • Collection of trajectories and events in a specific area – MT-KPI7	GRN4, MT4
FR5	*Multimodality* • Different data modalities successfully accommodated in the platform (audio, video, GPS, etc.)– UNS-KPI4 • Audio and video data modalities accommodated in AV emotion recognition – UNS-KPI6	UNS1
FR6	*Efficiency* • Efficiency of the planning of roads – GRN-KPI7 • Efficiency in urban planning – MT-KPI8	GRN4, MT4
	Non-functional requirements	
NFR1	*Scalability* • In terms of cost to add new devices/junctions – GRN1 • Cost to add new audio/video feed or device – GRN3 • Costs to add new devices – MT1-4	GRN1, GRN3, MT1-4
NFR2	*Modularity* • Integration of new equipment – UNS1-2	UNS1-2
NFR3	*Learning over distributed datasets* • Distributed audio-visual datasets for emotion recognition – UNS2	UNS2
NFR4	*Efficacy* • Demonstrated awareness with car drivers – GRN1 • Number of citizens reports – MT2 • Integrate the system in at least one safety campaign –GRN2	GRN1, MT2
NFR5	*System adoption/System attractiveness* • Suggest the driver behaviour detection in a safety campaign – GRN2 • Potential end-users: transport authorities, road users – GRN2 • Police – MT1 • Armed forces, municipalities – MT1-4 • Private companies with public participation (managing parking areas, bike sharing, car sharing, etc.) – MT1, MT3-4	GRN2, MT1
NFR6	*End-user experience* • Safer cycling – GRN1 • System welcomed by cyclists – GRN1 • Traffic managers – GRN2 • Relevant authorities personnel – GRN3 • Road engineers – GRN4	GRN1-4, MT1-4

	- Armed forces – MT1-4 - City/municipality – MT1-4 - Private companies with public participation (managing parking areas, bike sharing, car sharing, etc.) – MT1, MT3-4 - Citizen satisfaction – MT4	
NFR7	*Privacy preservation* - Audio and video anonymisation – UNS2 - Data protection – UNS2	GRN1-4, MT1-4, UNS1-2
NFR8	*Cyber security* - Secure transmission – UNS1 - Data protection – UNS2	GRN1-4, MT1-4, UNS1-2

The requirements in Table 2, together with the accompanying use case KPIs from D1.2, define and set the end goals that the MARVEL framework needs to achieve in the specific domain areas (traffic and public areas monitoring). Their fulfilment requires achievements both at the level of individual MARVEL components, their different combinations, and also for the overall MARVEL framework. We first present the revised MARVEL conceptual architecture in Section 2.2, explaining the role and the nature of each subsystem, together with the subsystems and components overview in Table 3. In Table 4, we then provide the mapping of the components to the user requirements from Table 2. The idea of the mapping in Table 4 is twofold. First, it provides a rationale and motivation for the MARVEL framework by showing how each component and their combinations address the user requirements. Second, it serves as a bridge, providing initial endpoints, between the user KPIs and the performance targets of the technological assets to be achieved within the project in the technical work packages, as the underlying technical enablers of the user KPIs.

2.2 Revised MARVEL Conceptual Architecture

We now present a revised MARVEL conceptual architecture that has been elaborated with a significant degree of more details and precision with respect to its version at month zero of the project, based on the consolidated requirements, revised SotA review, and relevant reference architectures and models.

MARVEL will develop a disruptive E2F2C ubiquitous computing framework that enables multi-modal perception and intelligence for Audio-Visual (AV) scene recognition and event detection in a smart city environment. MARVEL will collect, analyse, and mine multi-modal AV data streams of the three MARVEL pilots - city areas monitoring (MT, FBK), traffic monitoring (GRN), and small-scale drone experiment (UNS), in order to develop joint AV data representations and models for improved AV analytics and classification with respect to the scenarios where the audio and visual modalities are processed separately. To achieve real-time alerts and fast time-to-insights, MARVEL develops a ubiquitous computing framework with computations and AI tasks being distributed and performed at all layers of the underlying infrastructure - edge, fog, and cloud, with their deployment optimised and tailored according to the needs of the specific pilot, and with security and privacy implemented at each architectural layer.

The MARVEL framework consists of 29 technological components of a wide range of functionalities and the associated framework roles. To achieve coherence in the framework presentation, the components have been grouped into seven subsystems. This mapping was based on logical and functional similarity between components and their relations. The inherent part of the framework is also the deployment infrastructure that for MARVEL consists of the edge, fog, and cloud tiers (layers) and is thus referred to as the E2F2C infrastructure. In the project, the generic E2F2C infrastructure is instantiated with three specific E2F2C infrastructure examples induced by the three MARVEL pilots: GRN, MT, and UNS (experiment). Depending on the context, we will refer to the three architectural tiers - edge, fog, and cloud also as components of the MARVEL framework, as appropriate.

Specifically, the MARVEL architecture consists of the following seven subsystems:

1. Sensing and perception subsystem
2. Security, privacy, and data protection subsystem
3. Data management and distribution subsystem
4. Audio, visual, and multimodal AI subsystem
5. Optimised E2F2C processing and deployment subsystem
6. E2F2C infrastructure
7. System outputs: User interactions and the decision-making toolkit.

Figure 1 illustrates the revised conceptual architecture for the MARVEL framework, with colour coding applied across MARVEL subsystems (as detailed below). We next explain each of the subsystems; further details will be provided in

Table 3 and also in Sections 3-9.

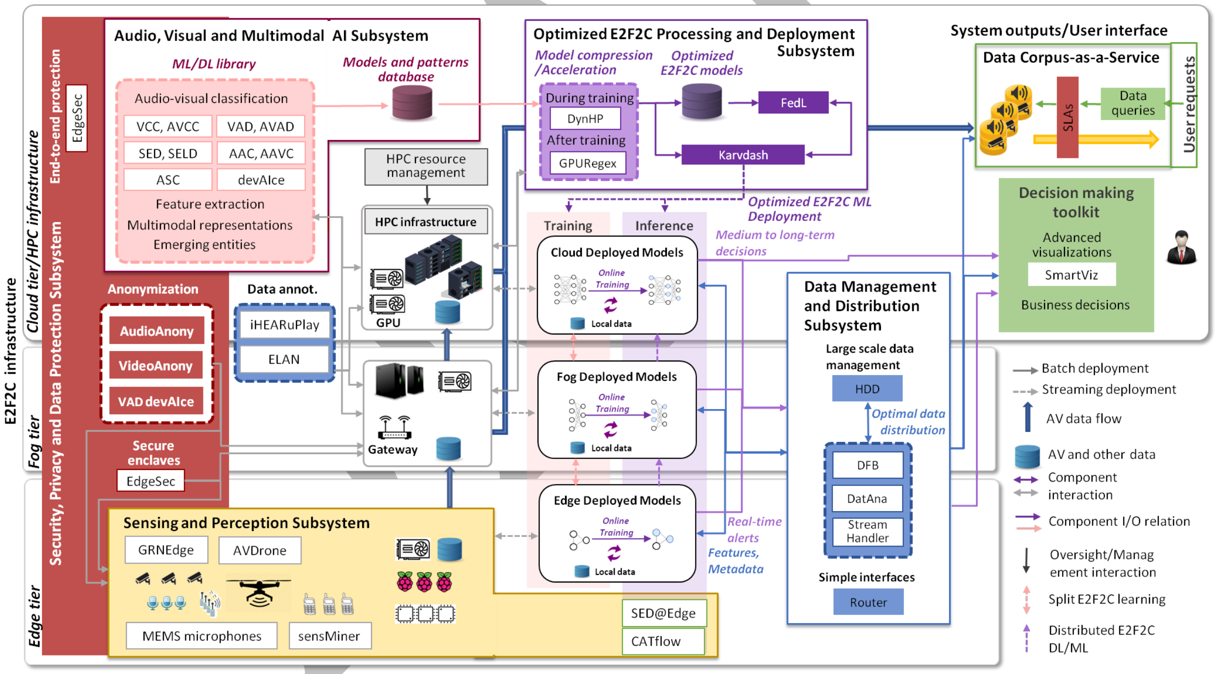


Figure 1. Revised MARVEL conceptual architecture

1. Sensing and perception subsystem consists of advanced MEMS microphones (IFAG), SED@Edge (FBK), for sound event detection at the edge, GRNEdge (GRN), a smart edge device collection, AVDrone (UNS) for drone-based AV data capturing, CATFlow (GRN) for object detection and classification in a road traffic video and SensMiner android app (AUD) for collecting audio data and audio tags, together with geolocation data, and a number of different devices, such as cameras, microphones, drones, etc.

The main role of this subsystem is to collect the sensing and perception elements of the edge layer of the E2F2C infrastructure at hand. This naturally includes *sensing and computing* devices, including the enabling software (e.g., routines for data collection, I/O devices/components interfaces, etc.), but also the embedded software elements for edge processing, e.g., for edge-based inference and training, embodying the *perception* aspect of the subsystem. This approach is well-aligned with the recent EU Strategic Research, Innovation and Deployment Agenda for the AI, Data and Robotics Partnership[2] and, as such, also accounts for the current surge of edge computing.

Regarding sensing, at the current version, sensing devices include mostly cameras and microphones, mounted either in stand-alone configurations or within other subsystem components – GRNEdge, AVDrone, and as an integral part of mobile phones (e.g., audio collection through sensMiner). As a generic case, the subsystem also subsumes devices producing other data modalities (e.g., text – through smartphone apps and similar, IoT data), including their potential edge-based processing.

2. Security, privacy, and data protection subsystem consists of EdgeSec (FORTH) for the security of edge devices, VideoAnony (FBK) for video anonymisation at the edge, AudioAnony (FBK) for audio anonymisation at the edge, together with the Voice Activity Detection (VAD) functionality of devAIce platform (AUD).

This subsystem spans all the other subsystems of the MARVEL architecture (sometimes such components are referred to as fabrics[3]). The role of this subsystem is two-fold: (i) to achieve *security* of the data and devices, against malicious attacks on data (e.g., stealing, eavesdropping, manipulation) and equipment (e.g., software manipulations), including end-to-end security; and (ii) proper *anonymisation to ensure privacy and protection of personal data*. For the first role, the relevant component is EdgeSec, while for the second, the relevant components are AudioAnony, VideoAnony, and the VAD module of devAIce.

3. Data management and distribution subsystem consists of four different data management platforms, i.e., Data Fusion Bus – DFB (ITML), StreamHandler (INTRA), Data acquisition framework – DatAna (ATOS), and Hierarchical Data Distribution – HDD (CNR).

The goal of this subsystem is to handle massive amounts of data coming from various sources and deal with their *management and proper distribution* at all architectural levels. Among others, the subsystem will consider the variety of formats and frequency of data. Besides the presented MARVEL platforms, which mostly target large-scale data management, the subsystem also includes *simple interfaces* for data transfer (routers) and systems for AV data transfer. Finally, the subsystem includes necessary software components for data preparation, including software packages for audio, visual, and audio-visual annotation (e.g., ELAN software [6], iHEARuPlay platform [7]).

4. Audio, visual and multimodal AI subsystem consists of devAIce SDK (AUD), Visual Crowd Counting – VCC (AU) and Audio-Visual Crowd Counting – AVCC (AU), Visual Anomaly Detection – ViAD (AU) and Audio-Visual Anomaly Detection – AVAD (AU), Automated Audio Captioning – AAC (TAU), Sound Event Detection – SED (TAU), Sound Event Localisation and Detection – SELD (TAU), and Acoustic Scene Classification – ASC (TAU).

[2] https://ai-data-robotics-partnership.eu/wp-content/uploads/2020/09/AI-Data-Robotics-Partnership-SRIDA-V3.0.pdf

[3] NIST Big Data Interoperability Framework: Volume 6, Reference Architecture, https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.1500-6r2.pdf

This subsystem encompasses all components, software libraries, etc. that build ML and DL models from available AV and other data. This subsystem is where the majority of the *AI-based functionalities* of the platform reside. Based on the type of data over which they operate, the components can be classified as audio-only (devAIce, SED, SELD, AAC, ASC), video only (ViAD, VCC), and audio-visual (AVAD, AVCC), with the potential addition of the categories accounting for future modalities (e.g., text: text-video, text-audio, text-audio-video, etc.). Further classifications can be based on components types in the sense whether they perform classification/detection, regression, detection of emerging entities, and feature extraction, with special emphasis on multimodal representations.

5. Optimised E2F2C processing and deployment subsystem consists of GPURegex (FORTH) for Graphics Processing Unit (GPU) acceleration for expression matching, DynHP (CNR) for Deep Learning (DL) model compression during training, FedL (UNS) for personalised Federated Learning (FL), and Karvdash (FORTH) for deployment and optimisation of AI tasks.

The role of this subsystem is multi-fold, geared overall towards optimised processing and deployment to be achieved by the framework. First, its baseline role is to achieve *deployment* of various tasks and services that are offered by the framework, in a user-controllable fashion using dashboards, across all levels and possible deployment points/execution sites in the E2F2C infrastructure at hand; this role in its large majority resides at Karvdash component. Second, the subsystem also deals with *optimisation* aspects of data and other processing: producing optimised ML/DL models that can fit resource-constrained devices (DynHP), GPU-based acceleration (GPURegex), and FedL (enabling training over distributed/private datasets).

6. E2F2C infrastructure consists of the HPC cluster (PSNC), HPC resource management and optimisation (PSNC), together with the three underlying infrastructural tiers – cloud, fog, and edge (GRN, MT, and UNS).

The E2F2C infrastructure subsystem represents the underlying infrastructure of the framework with the three tiers edge, fog, and cloud. Each of the three tiers depends on the actual infrastructure over which the framework is executed. The *cloud tier* includes the HPC infrastructure and HPC resource management and orchestration. In addition to these components, the cloud tier may involve an external cloud (elements). The *fog tier* typically consists of servers in local data centers, e.g., for various analytics purposes, but may also include gateways, GPUs, and other computing hardware; the fog tier may also involve distributed computing platforms of Raspberry Pis, or other elements specific to the infrastructure at hand. The *edge tier* represents the computing devices of the Sensing and perception subsystem, and in that sense is the overlapping point of the two subsystems. The typical devices forming the edge tier are microcontrollers, Raspberry Pis, Arduino platforms, but can also include edge GPUs, Intel NUC, NVIDIA Jetson Nano, and other computing platforms the design of which is targeted for edge operation.

7. System outputs: User interactions and MARVEL Data Corpus-as-a-Service and decision-making toolkit. Finally, the system outputs for MARVEL will be the Decision-making toolkit that consists of the SmartViz component (ZELUS) for advanced visualisations, and short and long-term decision-making, and also MARVEL Data Corpus-as-a-Service (STS), a large-scale corpus of processed multimodal AV public data.

Regarding the classification/encapsulation of MARVEL components within the above subsystems, we would like to note that, given the multifaceted nature of many of the components, different classifications/inclusions are certainly possible. For example, SED@Edge – as a distillation-based methodology for DL compression can alternatively be encapsulated by the Optimised E2F2C processing and deployment subsystem, and similarly for HDD, given the strong emphasis of this component on optimisation. However, the presented subsystems definition is carefully selected to best align with subsequent integration and development activities.

Each of the defined subsystems and their constituting components is briefly explained in

Table 3, and with their roles in the MARVEL platform delineated. The table also provides the components' TRL levels: initial (at M0 of the project), current (at M8), and expected (at M30, for the final release of MARVEL); for the MARVEL subsystems, we include the average TRLs computed as a (rounded) arithmetic mean of the TRLs of the participating components, together with the range of the components' TRL values (in brackets). Details of each component explaining their inner workings, including inbound and outbound interfaces and role(s) within the MARVEL platform are given in the following sections – Sections 3-9. High-level overviews of each of the seven subsystems are also provided in Sections 3-9.

Table 3: MARVEL components across MARVEL subsystems

Component information		**Description**	**TRL**		
Owner	**Subsystem /Component**	**Functionalities/roles in the platform**	**Initial**	**Current**	**Expected**
Sensing and perception subsystem		This subsystem describes the functionalities and roles of devices or components located at the edge. The main role of these devices or components is to collect data and optionally process the data at the edge. The devices include sensors (e.g., microphones and cameras) and optionally, processing components, e.g., light-weight processing units (e.g., Raspberry Pis, NVIDIA Jetson GPUs). The data type at the output of these devices is either raw audio, video, or both, processed audio-visual data or structured non-binary data.	4 (0-7)	4 (3-7)	7 (4-9)
IFAG	Advanced MEMS microphones	Audio data acquisition via MEMS microphones (IM69D130) and Edge processing of the acquired data (Cypress PSoC64 Standard Secure – AWS Wi-Fi BT Pioneer kit).	6	6	8
FBK	SED@Edge	The tool allows performing SotA sound event detection on very constrained devices. The tool reads as input monaural audio and provides probabilities for a predefined set of sound events.	4	4	8
GRN	GRNEdge	The role of this component is to collect audio-visual data and optionally process the audio-visual data at the edge. The component either stores the data on local storage at the edge for offline use or streams the data over wireless channels for consumption in real-time.	1	3	6

UNS	AVDrone	AVDrone is drone-mounted equipment for audio and video data acquisition and streaming, complemented with ground-based audio capturing and annotation using sensMiner. The component will optionally involve edge processing.	0[4]	3	4-5[5]
AUD	sensMiner	SensMiner is an Android app used for recording audio, user tags, and associated metadata (timestamps, GPS). The app will not be made available to the public, but only to the partners that need it.	7	7	9
GRN	CATFlow	The main algorithm in CATFlow is a video object detection and tracking module trained specifically for traffic objects, e.g., cars, trucks, and bicycles. It can execute anywhere a GPU is available (edge/fog/cloud) and can be fed with any video stream originating from any arbitrary location and camera.	5	5	7
Security, privacy, and data protection subsystem		This subsystem concerns ensuring that the MARVEL infrastructure preserves the privacy and security of the processed data, and security of the overall system. This involves two different aspects - Making the processing secure on edge, fog, or cloud against possible attacks by securing the data, the infrastructure, and the transmissions. - Ensuring the privacy of citizens is preserved by removing from the data all information that may allow identifying a person, as near as possible to the sensors.	3 (2-8)	3 (2-8)	6 (4-9)
FORTH	EdgeSec	EdgeSec is a security framework that aims to provide a platform of trust. It enables secure computing and focuses on the preservation of the confidentiality and integrity of the applications by leveraging Intel SGX security features. This framework allows developers to (almost) transparently execute their applications inside of an SGX enclave either inside the secure Docker containers that we provide or on native Linux host machines.	2	2	5
FBK	VideoAnony	The software anonymises video content, i.e., removing identifiable information from the input video, by detecting the person/face and obfuscating the faces with basic techniques e.g., blurring and more	2	2	6

[4] TRL 0 in the table typically indicates that the respective component did not exist/was not conceptually defined at M0 of the project.

[5] UNS is an experimental use case provider and hence the AVDrone component (as the underlying edge infrastructural tier for the Drone experiment) will be validated in a lab environment (TRL 4), with the possibility to extend the validation to relevant outdoor environment (TRL 5).

		advanced GAN-based face conversion techniques that are yet to be researched and developed under MARVEL. GPU is needed for real-time performance on the edge.			
FBK	AudioAnony	The tool removes the identity of the speaker from the audio. There are different strategies, from signal processing approaches to any-to-any voice conversion. Some approaches can run on edge devices already, others at the moment would require a device with GPU or enough computational power.	2	2	4
AUD	VAD (devAIce)	AUD's Voice Activity Detection (VAD) is part of the devAIce SDK, and detects voiced segments either in batched or online fashion.	8	8	9
Data management and distribution subsystem		The subsystem will be handling massive amounts of data (with various sources, formats, and frequencies) dealing with their proper management and distribution.	4 (4-6)	4 (4-6)	6 (6-8)
ITML	Data Fusion Bus (DFB)	Enables trustworthy transfer of data between connected components and permanent storage. It comprises a set of dockerized, open-source components, including Apache Kafka and ELK.	4	4	6
INTRA	StreamHandler	Allows the data flow within the interconnectivity of components via Kafka topics allowing the dockerized deployments of the framework.	6	6	8
ATOS	DatAna	Apache NiFi-based data processing framework. Able to create complex data flows and move data from fog MiNiFi agents to the cloud NiFi nodes, or multiple modern storage or messaging systems.	4	4	6
CNR	Hierarchical Data Distribution (HDD)	Distributed adaptive data delivery and access algorithmic schemes for guaranteeing real-time delay requirements while effectively prolonging network lifetime in wireless industrial edge networks.	4	4	6
Audio, visual, and multimodal AI subsystem		This subsystem provides the components implementing the AI capabilities of the MARVEL architecture. These components are operating on single modalities (audio or visual) or multimodal data (audio-visual), depending on the availability of both audio and visual synchronised sensors.	3 (2-4)	4 (3-6)	5 (3-6)
AUD	devAIce	AUD's devAIce is a C++ SDK used to wrap all of AUD's intelligent audio analysis modules, including sound event detection and acoustic scene classification,	3	6	6

		which are relevant for MARVEL.			
AU	Visual anomaly detection (ViAD)	Detecting deviations from normality within given images or video frames.	4	4	5
AU	Audio-Visual anomaly detection (AVAD)	Detecting deviations from normality within given images or video frames as well as from the corresponding audio from the scene.	4	4	5
AU	Visual crowd counting (VCC)	Counting the total number of people present in given images or video frames.	4	4	5
AU	Audio-Visual crowd counting (AVCC)	Counting the total number of people present in given images or video frames as well as from the corresponding ambient audio.	4	4	5
TAU	Automated audio captioning (AAC)	Creating textual descriptions of the contents of general audio signals.	2	3	3
TAU	Sound event detection (SED)	Detecting sounds in general audio signal that have a textual label assigned to them (sound event) along with the start and end timestamps of the sound activity.	2	3	5
TAU	Sound event localisation and detection (SELD)	Jointly detecting sound events and estimating the direction of arrival for them. The direction of arrival is with respect to the position of the microphone.	2	3	5
TAU	Acoustic scene classification (ASC)	Recognising the acoustic scene class of recordings.	2	3	5
Optimised E2F2C processing and deployment subsystem		The E2F2C subsystem integrates dispersed execution sites into a unified, distributed execution environment, enabling the deployment of services at all layers spanning from Edge to Fog to Cloud/HPC-Centre.	3 (2-3)	3 (2-3)	5 (4-5)
FORTH	GPURegex	GPURegex is a real-time high-speed pattern matching engine that leverages the parallelism properties of GPGPUs to accelerate the process of string and/or regular expression matching. It is offered as a C API and allows developers to build applications that require text-based pattern matching capabilities while simplifying the offloading and acceleration of the workload.	3	3	5
CNR	DynHP	Training and compressing a deep neural network model under a fixed memory budget for model's deployment at edge/fog layer.	2	2	4
UNS	FedL	FedL is an algorithmic and protocol framework for federated learning. Within MARVEL, the framework will be advanced to personalised algorithms and robust protocols for complex operating environments.	3	3	5

FORTH	Karvdash	Karvdash will provide a dashboard for instantiating services as orchestrated containers, and deployed via appropriate automation to execution sites selected by a dynamic online optimisation strategy.	3	3	5
E2F2C infrastructure		This subsystem represents the underlying infrastructure for the MARVEL framework consisting of the edge, fog, and cloud tiers. The infrastructure provided by PSNC to MARVEL will consist of access to HPC supercomputer and access to virtualised private cloud.	3 (0-8)	3 (1-8)	7 (4-9)
PSNC	HPC infrastructure Eagle cluster & LabITaaS	The MARVEL HPC infrastructure will be hosted at PSNC's facilities. The name of the Eagle cluster describes HPC resources (1087 nodes) and LabITaaS for cloud resources (120 servers).	8	8	9
PSNC	HPC management and orchestration PSNC SLURM & PSNC OpenStack	Ensuring that large number of resources and a high-speed network and storage offered by HPC will be effectively utilised to guarantee high performance of data processing. Management and monitoring is provided via SLURM software and OpenStack for cloud resources.	7	7	9
MT	Cloud tier	MT will make exclusive use of the cloud services provided by the MARVEL consortium.	0[6]	1	6
GRN	Cloud tier	GRN will make exclusive use of the cloud services provided by the MARVEL consortium.	0	1	6
UNS	Cloud tier	The cloud tier will potentially be explored by other components to be used within UNS use cases (e.g., Karvdash, SmartViz). UNS will make exclusive use of the cloud services provided by the MARVEL consortium.	0	1	4-5
MT	Fog tier	The fog tier will consist of a DELL workstation located at FBK premises, that, via secure connection captures the raw data from the sensors. The server may be equipped with an NVIDIA GeForce RTX 3080.	0	1	6
GRN	Fog tier	The GRN fog tier consists of an HPE ProLiant DL385 GEN10 Plus Server with one NVIDIA Tesla T4 16GB installed.	4	4	6
UNS	Fog tier	The fog tier consists of a PC, distributed Raspberry Pi 3.0 and 4.0 computing	3	3	4-5

[6] We are confident that the infrastructure of the MT pilot will reach the desired level of technology readiness because each infrastructural tier (edge, fog, cloud) is made of components that are either commercial or are to be developed within MARVEL but are building on existing technology assets of sufficiently high TRLs. The same remark applies to the GRN and the UNS pilots.

		environment, and a server machine. UNS data server supports distributed network storage with RAID 1+1 protection.			
MT	Edge tier	The MT edge consists of a series of Raspberry Pis installed in the nearest cabinet where selected cameras are installed. It is expected that at the end of the experimentation some modules for the analysis of the video and audio can be installed and executed on the Raspberry Pi.	0	1	6
GRN	Edge tier	The GRN edge tier consists of a number (8-10) of audio-video sensors (GRNEdge component) that stream AV data to the fog layer. It is expected that during pilot execution the AV data is anonymised at the edge and in addition, a selection of AI tasks are processed on some of the edge devices.	0	2	6
UNS	Edge tier	The edge tier for the UNS drone experiment consists of the AVDrone component. For the Emotion recognition use case, the data collection edge tier consists of a smartphone with an audio-visual data capturing Android app.	0	3	4-5
System outputs: User interactions and the decision-making subsystem		The interface between the end-user and the MARVEL framework. The decision-making subsystem consists of a variety of visualised information and insights that can support the users take long- and short-term decisions.	3 (1-4)	3 (1-4)	6 (5-6)
ZELUS	SmartViz	An extensible visualisation toolkit including but not limited to monitoring dashboards, configurable visual representations, and collaboration features that support easy exploration and insight gaining from big volumes of multidimensional data for the needs of IT and non-IT experts.	4	4	6
STS	MARVEL Data Corpus-as-a-Service	A data pool of processed multimodal audio-visual data that will be developed throughout the project’s duration. MARVEL’s Data Corpus will be released as a service and will enable smart cities to build and deploy innovative applications that are based on multimodal perception and intelligence.	1	1	5

Table 4 provides a mapping of system requirements from Table 2, gathered through extensive analysis of MARVEL use cases, to MARVEL subsystems and components. Several of the defined user goals are complex and synthetic in nature and can be achieved only by joint improvements at several fronts. The details are provided in Table 4.

Table 4: Mapping of system requirements in Table 2 to MARVEL subsystems and their enabling components/functionalities

System requirements	MARVEL subsystem	Enabling component/functionality
FR1, FR2, FR3, FR4, FR5; NFR1, NFR2	Sensing and perception subsystem	As the main source of data in MARVEL, this subsystem is a cross-cutting enabler that affects almost all the end-user requirements. With regards to FR1-FR4, it provides the necessary data for model-building (FR1-FR2-FR3) and patterns and insights extraction (FR4). In addition, this subsystem supports FR3 by using hardware that enables perception and inference at the edge (SED@Edge, also CATFlow), thus achieving fast and close-to-the-source inference. Given the variety of sensing devices (cameras, microphones, GPS, etc.) and data modalities, the subsystem ensures fulfilment of FR5. The subsystem contributes also to non-functional requirements: the variety of devices provided makes it possible to add new types with reasonable effort (NFR2) and the use of flexible technologies enables scalability, in terms of reduced marginal cost for adding new sensors (NFR1).
FR1, FR3; NFR7, NFR8	Security, privacy, and data protection subsystem	This subsystem is the key-enabling subsystem for NFR7 and NFR8. NFR7 is addressed by the anonymisation components of MARVEL VideoAnony, AudioAnony, and VAD (devAIce). NFR8 is addressed by EdgeSec that enables security of the whole E2F2C infrastructure, including end-to-end/network security. For FR1 and FR3, the subsystem contributes by ensuring trustworthy data and reliable system operation, and also by enabling model building even when handling personal data.
FR1,FR4, FR5, FR6; NFR1, NFR6	Data management and distribution subsystem	The data management and distribution subsystem, by handling large volumes of audio-visual and other data, metadata, features extracted on-the-fly, etc. ensures that the appropriate data is available where needed by the AI tasks (FR1 and FR4). FR4 in particular is addressed by managing a large-scale collection of trajectories (traffic, urban mobility) with the accompanying GUIs (NFR6). FR5 is supported by the DFB, by its capability to fuse large-scale multimodal data. System efficiency (FR6) is contributed to by the HDD component that optimises the distribution of data across the E2F2C infrastructure. Finally, all data management platforms, i.e., DFB, StreamHandler, and DatAna, are designed for scalability, thus they fit NFR1.
FR1, FR2, FR3, FR5; NFR6	Audio, visual, and multimodal AI subsystem	This is the key subsystem to address FR1, FR2, and FR3. FR1 is addressed by implementing a deep learning model that exploits audio-visual input data to learn high-performing representations leading to increased performance. FR2 is addressed by implementing an efficient deep learning model that is able to decrease the processing time needed to provide the audio-visual inference results (e.g., crowd counting, audio-visual anomaly detection). Robustness to operating conditions (e.g., low light) is enabled by multimodal, audio-visual models (FR3), which also underpins FR5. The subsystem indirectly contributes also to NFR6, by achieving high-performance that leads to a better user experience.
FR1, FR2, FR3, FR4, FR5;	Optimised E2F2C processing and deployment	This subsystem enables the deployment of models for real-time or batch inference through Karvdash (FR1-FR2-FR3-FR4). FR2 is especially relevant for this subsystem, through components that

NFR1, NFR2, NFR3	subsystem	achieve acceleration (GPURegex) and enable edge deployment through model compression (DynHP). FedL contributes to FR3 by designing robust and adaptive communication protocols, and to NFR3 by enabling learning over distributed datasets. NFR1 is achieved by both Karvdash and FedL, by being able to scale out to many deployment points, as dictated by the infrastructure. NFR2 is enabled by virtualisation of resources through Kubernetes (Karvdash).
FR1, FR4, FR3; NFR1, NFR2	E2F2C infrastructure	The E2F2C infrastructure is a cross-cutting enabler for all the requirements. For FR1 and FR4, the HPC infrastructure, including HPC resource management and orchestration, is a key enabler for model building from large scale-datasets, to be collected and uploaded to the PSNC premises after appropriate preparation. The fog and edge tiers are particularly important for FR3, specifically for enabling fast reaction to the detected events. The subsystem contributes also to NFR1 and NFR2, by creating the possibility to add and/or replace the existing equipment in a seamless manner.
FR1, FR2, FR4, FR6; NFR4, NFR5, NFR6	System outputs: User interactions and the decision-making toolkit	The decision-making toolkit provides analytical reasoning for medium to long-term business analytics, addressing FR1 and FR4. FR2 is underpinned by effective, real-time visualisations that enable fast reaction from the end-users. FR6 is achieved through long-term analytics in road planning and urban mobility. NFR4, NFR5, and NFR6 significantly rely on system attractiveness, usability, and user-friendliness that will be enabled by the decision-making toolkit, empowered by the SmartViz functionalities.

2.3 Architecture modeling and deployment

Figure 1 focuses on the **logical interactions** between MARVEL subsystems and their constituting components; colour-coding is applied to differentiate between different MARVEL subsystems. For example, yellow is used for the Sensing and perception subsystem, brick red for the Security, privacy, and data protection subsystem, etc. Interactions between components are illustrated using arrows, including interaction types, as indicated in the legend on the bottom right part of the figure, or with the text next to the arrows.

Besides the logical viewpoint, the figure also embeds information about the **physical placement** of components in terms of edge, fog, and cloud tier. For the majority of components the typical deployment point is shown by the subsystem location – i.e., within the edge, fog, and cloud fields/levels of the figure. Alternatively, deployment location is also indicated by the grey deployment arrows (e.g., EdgeSec). We detail the deployment in the following paragraphs. See also Table 5 below.

The Sensing and perception subsystem is located at the edge, as also shown in the figure. The Security, privacy, and data protection subsystem concerns the whole MARVEL framework, and is therefore shown in the figure across all the three architectural layers; any of its components (EdgeSec, Audio/VideoAnony, VAD (devAIce)) can run at any of the three architectural layers, which is indicated by the grey deployment arrows pointing to the respective infrastructure boxes.

Data management and distribution subsystem spans all the three layers, with the orchestrating components located at the cloud, while different agents/connectors for data sources, etc., can run at the edge or fog. This is shown in the figure by the subsystem field spanning edge, fog, and cloud levels.

The Audio, visual and multimodal AI subsystem, which typically requires extensive computing power for DL model building, is located at the cloud level. There are of course other deployment possibilities for this subsystem, depending on the available resources, the point of available data and the model to be trained. This is accounted for by the pink field, marked as *"Training"*, spanning each of the three architectural levels. The components of the Optimised E2F2C processing and deployment subsystem are also typically located at the cloud level (e.g., the core functionalities of Karvdash, FedL servers, GPURegex, DynHP), but they can also have (i) alternative deployment points; e.g., in addition to the cloud, GPURegex can be deployed both at the edge and at the fog, depending on the availability of GPU machines, and similarly with DynHP); (ii) or by their nature, they have multiple points of presence (e.g., Karvdash, FedL clients); the latter is indicated by the violet arrows originating from the subsystem and pointing to the pink and green fields of the figure (*"Training"* and *"Inference"*, respectively).

E2F2C infrastructure naturally spans the three layers, with the HPC infrastructure component including the HPC management and orchestration component being located at the cloud level. The components of System outputs: User interactions and MARVEL Data Corpus-as-a-Service – SmartViz and MARVEL Data Corpus, are both located at the cloud level, as shown in the figure.

Detailed deployment information about physical placement and components' deployment requirements is provided in Table 5. Regarding components deployment (i.e., in the edge, fog, and cloud tier), the table indicates both the nominal deployment, where little or no requirement is needed and the targeted deployment (e.g., edge), together with the respective requirements.

Table 5: Edge, fog and cloud deployment of MARVEL components and components' requirements

Component owner	Subsystem /Component	Component requirements	Edge	Fog	Cloud
Sensing and perception subsystem		This section gives an overview of the requirements for deploying the components in the sensing and perception subsystem. In more specific terms, it indicates any specific hardware and operating system platforms required, output data type and format, and method of transmission or storage.	X		
IFAG	Advanced MEMS microphones	Connected to a Linux/Windows platform for example RPi.	X		
FBK	SED@Edge	The tool receives as input monaural audio signal and provides as output an estimated sound event (or probabilities for a set of events) with a given temporal resolution. It can operate with different HW constraints. The tool needs annotated data representative of the application scenario	X		

		for supervised training or model adaptation.			
GRN	GRNEdge	GRNEdge devices will be deployed in the selected locations for executing the use-cases/pilots. An audio-video sensor that collects data, encodes data into MKV format, and either stores the data on local storage at the edge or streams the data over wireless channels (modem). The onboard processor (Raspberry Pi) board is used to synchronise and encode the audio and video streams, but can also be used for lightweight processing at the edge. GRNEdge can be upgraded with a GPU to compute AI tasks or implement AI-based components at the edge. The component requires a power supply.	X		
UNS	AVDrone	AVDrone is an edge-based data acquisition hardware setup with data streaming and storage functionality. The component requires a wireless connection with the Fog tier. To achieve multimodal processing, in-situ synchronisation is advised.	X		
AUD	sensMiner	SensMiner will not be deployed in the MARVEL architecture. It works locally on Android phones, and will only be used during the data collection phase.	X		
GRN	CATFlow	CATFlow will be deployed mainly at the Fog layer, but can also be deployed at the edge layer if sufficient computational power is available and edge deployment enables system response targets (KPIs). The main algorithm in CATFlow is a video object detection and tracking system trained specifically for traffic objects, e.g., cars, trucks, and bicycles. It can execute anywhere a GPU is available (edge/fog/cloud) and can be fed with any video stream originating from any arbitrary location and camera.	X	X	
Security, privacy, and data protection subsystem		For this subsystem hardware requirements mostly apply, which are critical as these components are required to operate as close as possible to the data sources. For video anonymisation, the limited computational power of edge devices could be an issue, where a fog host is most likely. For audio anonymisation different solutions (with different performance) can be employed. One very **critical** requirement is the availability of an Intel CPU to deploy EdgeSec. This would prevent the use of	X	X	X

		Raspberry Pi or micro-controllers as edge devices.			
FORTH	EdgeSec	The only hard requirement is the presence of an Intel CPU with SGX support. Ideally, an up-to-date LTS Ubuntu distribution and root access would be appreciated to be able to install the necessary drivers and to create the necessary secure communication channel between the devices of each layer. In case there is no SGX support available (most likely to happen in the edge), the minimum viable solution could be deployed where best cybersecurity practices will be applied considering the device and its role in the whole infrastructure, including also securing the communication channels.	X	X	X
FBK	VideoAnony	The software can be potentially run on the edge or fog with real-time performance if the (edge) device is powered by GPU and with face blurring applied for anonymisation. For the GAN-based technique, it would be challenging to run on the edge. Research on the GAN-based model compression is needed.	X	X	
FBK	AudioAnony	There are different possible solutions. At the moment the neural voice conversion would probably not fit on an edge device, but other signal processing solutions would fit.	X	X	
AUD	VAD (devAIce)	The VAD module of devAIce can be used on fog and cloud nodes. It can also be used in Raspberry PIs, which are considered as “high-end” edge nodes.	X	X	X
Data management and distribution subsystem		Application requirements and use case specification from the underlying networking environment. Definition of the data/networking problem that the deployment is bound to address. Desired networking protocols information. Details on the edge-fog-cloud setting are under consideration. Types of data in the network (VVV).	X	X	X
ITML	Data Fusion Bus (DFB)	Can run on a cluster, using Docker and Kubernetes. Minimum requirements: CPU: 8 cores, RAM: 32GB, Storage: 256GB. Recommended requirements: CPU: 8-16 cores, RAM: 64GB, Storage: >512GB.			X

INTRA	StreamHandler	Can be executed within a cluster, utilising Docker and Kubernetes. Minimum requirements: CPU: 8 cores, RAM: 32GB, Storage: 256GB. Recommended requirements: CPU: 8-16 cores, RAM: 64GB, Storage: >512GB.			X
ATOS	DatAna	Edge/Fog: MiNiFi. Small footprint 50Mb binary distribution. Linux (CentOS, Debian, Ubuntu relatively modern version). Java version requires JDK8+. Decent Java heap and RAM depend on the data to process. It can be installed in devices such as a Raspberry PI or similar. Cloud: NiFi. Minimal: Java 8 or 11, Linux, Unix, Windows, MacOS. Can be run on a laptop, but recommended to be run in a cluster of servers. Memory and size depend on the data flows to be defined. Sufficient storage is needed, as data is stored in a disk while processing. Recommended: Minimum of 3 dedicated nodes, 8+ cores per node (more is better), 6+ disks per node (SSD or spinning), at least 8 GB RAM. Sufficient disk space.	X	X	X
CNR	Hierarchical Data Distribution (HDD)	An underlying network of IoT devices that can support either distributed or centralised processing and data functions. Can run in small networks, but in order to effectively improve performance on a large scale, larger and less intuitive network deployments are preferred. Available device HW/FW/SW information on communication and computation abilities.	X	X	X
Audio, visual, and multimodal AI subsystem		Analysis of multi-modal data (audio and visual) for providing the AI capabilities of MARVEL architecture. Functionalities include joint analysis of audio-visual data and single-modality data, depending on the task to be addressed in relation to specific use cases.	X	X	X
AUD	devAIce	devAIce consists of different modules, each with its own requirements. Most modules require a fog node to run. Some modules might run on high-end edge nodes.	X	X	X
AU	Visual anomaly detection (VAD)	Minimum requirements (training): 64GBs of RAM, 1TB NVMe SSD, 8 x 2.0GHz CPU cores, 4 x 2080Ti GPUs, Ubuntu 20.04 OS. Minimum requirements (inference at the edge): NVIDIA Jetson TX2. Minimum requirements (inference at the	X	X	X

		fog): 16GBs of RAM, 1 x 2.0GHz CPU cores, 1 x 2080Ti GPUs, Ubuntu 20.04 OS.			
AU	Audio-Visual anomaly detection (AVAD)	Minimum requirements (training): 64GBs of RAM, 1TB NVMe SSD, 8 x 2.0GHz CPU cores, 4 x 2080Ti GPUs, Ubuntu 20.04 OS. Minimum requirements (inference at the edge): NVIDIA Jetson TX2. Minimum requirements (inference at the fog): 16GBs of RAM, 1 x 2.0GHz CPU cores, 1 x 2080Ti GPUs, Ubuntu 20.04 OS.	X	X	X
AU	Visual crowd counting (VCC)	Minimum requirements (training): 64GBs of RAM, 1TB NVMe SSD, 8 x 2.0GHz CPU cores, 4 x 2080Ti GPUs, Ubuntu 20.04 OS. Minimum requirements (inference at the edge): NVIDIA Jetson TX2. Minimum requirements (inference at the fog): 16GBs of RAM, 1 x 2.0GHz CPU cores, 1 x 2080Ti GPUs, Ubuntu 20.04 OS.	X	X	X
AU	Audio-Visual crowd counting (AVCC)	Minimum requirements (training): 64GBs of RAM, 1TB NVMe SSD, 8 x 2.0GHz CPU cores, 4 x 2080Ti GPUs, Ubuntu 20.04 OS. Minimum requirements (inference at the edge): NVIDIA Jetson TX2. Minimum requirements (inference at the fog): 16GBs of RAM, 1 x 2.0GHz CPU cores, 1 x 2080Ti GPUs, Ubuntu 20.04 OS.	X	X	X
TAU	Automated audio captioning (AAC)	Minimum training requirements at the cloud: • Total of 64 GB GPU memory • 2.1 GHz CPUs of number equal to 10x the number of GPUs used • 64 GB RAM • 1TB SSD Minimum inference requirements at the cloud: • Total of 32 GB GPU memory • 4 CPUs @ 2.1 GHz • 64 GB RAM • HDD/SSD to hold the incoming and outgoing data if needed.			X
TAU	Sound event detection (SED)	Minimum training requirements at the cloud: • Total of 32 GB GPU memory • 2.1 GHz CPUs of number equal to 10x the number of GPUs used • 64 GB RAM • 1TB SSD			X

		Minimum inference requirements at the cloud: • Total of 16 GB GPU memory • 4 CPUs @ 2.1 GHz • 64 GB RAM • HDD/SSD to hold the incoming and outgoing data if needed.			
TAU	Sound event localisation and detection (SELD)	Minimum training requirements at the cloud: • Total of 32 GB GPU memory • 2.1 GHz CPUs of number equal to 10x the number of GPUs used • 64 GB RAM • 1TB SSD Minimum inference requirements at the cloud: • Total of 16 GB GPU memory • 4 CPUs @ 2.1 GHz • 64 GB RAM • HDD/SSD to hold the incoming and outgoing data if needed.			X
TAU	Acoustic scene classification (ASC)	Minimum training requirements at the cloud: • Total of 32 GB GPU memory • 2.1 GHz CPUs of number equal to 10x the number of GPUs used • 64 GB RAM • 1TB SSD Minimum inference requirements at the cloud: • Total of 16 GB GPU memory • 4 CPUs @ 2.1 GHz • 64 GB RAM • HDD/SSD to hold the incoming and outgoing data if needed.			X
Optimised E2F2C processing and deployment subsystem		The E2F2C processing and deployment subsystem requires a container-based management substrate spanning the whole computing continuum from Edge to Fog to Cloud/HPC-Centre, upon which it distributes services.	X	X	X
FORTH	GPURegex	The only hard requirement is the presence of a general-purpose GPU that supports the OpenCL standard. It could be either a dedicated card or an integrated chip within the CPU die. An OpenCL enabled processor would also be sufficient, however, the benefits of the acceleration will be diminished.	X	X	X
CNR	DynHP	DynHp requires input data, a model to train and compress, and a memory budget available.	X	X	X
UNS	FedL	FedL should run simultaneously at the edge, fog, and cloud tiers. The possibility for partial edge model deployment will also be explored (e.g., using split learning). Requirements for FedL are inherited from	X	X	X

		the adopted ML/DL model, thus dictating the deployment tier (nominally, FL clients will run at the fog tier; in case of lightweight ML models and/or more powerful edge hardware, FL clients can also run at the edge).			
FORTH	Karvdash	Karvdash requires a Kubernetes environment running on one or more nodes (each node should have at least 8 GB RAM). Storage can be available as a local device shared across nodes (via NFS or similar).	X	X	X
E2F2C infrastructure		This subsystem represents the underlying infrastructure for the MARVEL framework, spanning the edge, fog, and cloud tiers.	X	X	X
PSNC	HPC infrastructure	The infrastructure provided by PSNC to MARVEL will consist of access to HPC supercomputer and access to virtualised private cloud.			X
PSNC	HPC management and orchestration	The infrastructure provided by PSNC to MARVEL will consist of access to HPC supercomputer and access to virtualised private cloud.			X
MT	Cloud tier	MT will make exclusive use of the cloud services provided by the MARVEL consortium.			X
GRN	Cloud tier	GRN will make use of the cloud services provided by the MARVEL consortium.			X
UNS	Cloud tier	Given the small scale of the UNS drone experiment, no particular requirements for the cloud tier are needed.			X
MT	Fog tier	The fog tier will consist of a DELL workstation located at FBK premises, that, via secure connection captures the raw data from the sensors. The server may be equipped with an NVIDIA GeForce RTX 3080.		X	
GRN	Fog tier	The GRN fog tier consists of an HPE ProLiant DL385 GEN10 Plus Server with one NVIDIA Tesla T4 16GB installed at premises authorised by GRN.		X	
UNS	Fog tier	Drone experiment requires wireless connection at the Fog tier and resources for model training and inference.		X	
MT	Edge tier	The edge consists of a series of Raspberry Pis installed in the nearest cabinet where selected cameras are installed. It is expected that at the end of the experimentation some modules for the analysis of the video and audio can be installed and executed on the Raspberry Pi.	X		

GRN	Edge tier	Deployed at the junction where use cases are executed. The GRN edge tier consists of a number (8-10) of audio-video sensors (GRNEdge) that stream AV data to the fog layer. It is expected that the AV data is anonymised at the edge and in addition, a selection of AI tasks are processed on some of the edge devices.	X		
UNS	Edge tier	The drone experiment requires a wireless connection at the edge tier (drone and ground-based deployment).	X		
System outputs: User interactions and the decision-making toolkit		The decision-making toolkit requires JSON formatted data as input and a minimum of: • CPUs: >=2 • RAM: >=4GB (depending on the data volume and utilised visualisations) • Disk space: >=30GB • web browser from the user's side. APIs/other methods (e.g., direct access to data) are required to get the indicator values and the generated events.			X
ZELUS	SmartViz	Labelled streaming or historical data. Inference results of processed data by other components.			X
STS	MARVEL Data Corpus-as-a-service	Minimum requirements at the cloud for installing the Apache Ambari, HBase and Hadoop file system: • Total of 32 GB RAM memory • 8 CPUs @ 2.1 GHz • At least 3 PB (in total) of disk space for holding the corpus data			X

2.4 Summary of innovations of the MARVEL architecture

This section provides the identified and planned innovations for each of the MARVEL components, presented across MARVEL subsystems.

2.4.1 Sensing and perception subsystem

Advanced MEMS microphones, IFAG

IFAG is developing two boards, one working with an array of four microphones and the other one with eight that will provide data to the edge node. The board with four microphones will be connected via USB and the one with eight via WiFi.

SED@EDGE, FBK

FBK plans to further develop this technology from TRL 4 to TRL 8, making it robust enough to be deployed in real urban scenarios. Performance improvements will be achieved by applying the distillation process to more recent and performing architectures and trying to reduce the performance drop due to quantization. FBK will explore self-supervised or cross-modal supervised adaptation strategies to be performed on the edge devices.

GRNEdge, GRN

The innovations with respect to the GRNEdge component include: (i) the synchronised transmission of audio-video data; (ii) deployment of lightweight AI models within the edge component; and (iii) deployment of anonymisation algorithms within the edge component.

AVDrone, UNS

AVDrone will enable target users to classify the crowd behaviour, providing a faster response in the case of an anomaly, hence preventing dangerous or undesirable situations. The innovative setup includes video capturing from the drone and audio capturing from the drone and ground-based MEMS microphones that could help crowd scene classification and localisation of the recorded event. Data will be stored locally, but also streamed in real-time to an external IP address using Wi-Fi and/or LTE-M module. During the course of the project, the component will be enriched with innovative MARVEL technologies, such as MARVEL AI technologies (e.g., AVAD, AVCC, devAIce, FedL) and also technologies that operate or enable operation close to the source e.g., EdgeSec, Audio/VideoAnony, DynHP, etc.

SensMiner, AUD

SensMiner is an Android app that enables crowdsourced audio data collection. The user interface of the component will be customised and enriched by crowd behaviour ontologies induced by the Drone experiment use case within which sensMiner will be applied.

CATFlow, GRN

The innovations with respect to the CATFlow Component include fusing the output of CATFlow with SED models and deploying CATFlow at the edge.

2.4.2 Security, privacy, and data protection subsystem

EdgeSec, FORTH

EdgeSec is part of and enhances the functionality of MARVEL's Security, privacy, and data protection subsystem. EdgeSec addresses some of the security issues and risks that may arise in the host machines of all three layers of the MARVEL envisioned architecture (i.e., edge, fog, and cloud) and also helps with the preservation of the confidentiality and the integrity of data in transit by establishing secure communication channels between the communicating parties. Application of EdgeSec within MARVEL use cases will enable novel, security embedded edge and fog deployments thus preventing malicious data hijacking, data theft, and manipulation, as well as software integrity verification and edge-to-edge communication protection.

VideoAnony, FBK

VideoAnony performs video anonymisation through GAN-based techniques which have the advantages of maintaining most of the context information intact while removing only the most identifiable information of each data subject, i.e., the faces. On top of existing SotA GAN-based solutions, the component will further advance the applications towards high facial naturalness, temporal smoothness, and reducing the risk of re-identification in real-world CCTV scenarios, where extreme facial poses and varying environmental conditions are the main challenges to address.

AudioAnony, FBK

One approach to voice anonymisation while preserving the audio content is to use voice conversion. Established solutions exist, either using a pipeline of ASR and TTS or using neural conditioned auto-encoders. However, they are not suitable to run on resource-constrained devices. Moreover, they are currently applied to close-talking noise-free recording and have never been tested in outdoor real scenarios. These are the main directions along which innovations for this component are expected.

VAD (devAIce), AUD

devAIce for intelligent audio analytics, including AUD's award-winning openSMILE audio feature extraction toolkit, and modules for sound event detection, acoustic scene classification, and speech analysis. A core module is voice activity detection (VAD), which will be used inside the MARVEL architecture to accurately and reliably detect voice segments so they can be appropriately anonymised before further processing. The current VAD module is optimised for near-field microphone recordings. However, the MARVEL use cases require the VAD module to work in far-field, noisy, outdoor conditions. This will result in performance degradation at the current level of technology. Future innovations will be focused on improving this performance and making VAD work robustly for all MARVEL use cases.

2.4.3 Data management and distribution subsystem

Data Fusion Bus (DFB), ITML

Being a data management platform that implements fusion of non-binary data originating from different components, DFB serves as a facilitator across various use case scenarios. In that regard, DFB's contribution to MARVEL's innovations lies in offering a high-quality, scalable, secure, and performant solution to heterogeneous data aggregation and processing, while being versatile and adjustable to different use case setups.

StreamHandler, INTRA

StreamHandler is a platform that can process non-binary data. It is a high-performance (low latency and high throughput) distributed streaming platform for handling real-time data based on Apache Kafka. It can efficiently ingest and handle massive amounts of data into processing pipelines, for both real-time and batch processing. The platform and its underlying technologies can support any type of data-intensive ICT services (Artificial Intelligence, Business Intelligence, etc.) from cloud to edge.

DatAna, ATOS

DatAna is based on the Apache NiFi ecosystem and existing data processors. DatAna can be used also in edge/fog devices (i.e., in a Raspberry Pi). Apache MiNiFi agents can be deployed on the device and then connect to other NiFi instances to generate topologies of connected elements, paving the way to gather and process data from devices and move it to a NiFi instance in another server or in a cluster.

In MARVEL, it is expected that specific templates of NiFi data flows for the scenarios of the project will be created. These can serve as best practices for future scenarios where data management using NiFi could be of interest, hence facilitating future usage and smart composition of data flows. If required by the scenarios, it is planned to deliver specific NiFi processors to improve performance or features not provided by the off-the-shelf NiFi processors, while also improving the features of managing the topologies of MiNiFi agents connected to NiFi nodes or clusters.

Hierarchical Data Distribution (HDD), CNR

HDD was developed for achieving smart data distribution in wireless environments with heterogeneous nodes. HDD innovatively extends current data distribution technologies to make them "AI-ready", i.e., to couple them with MARVEL distributed AI algorithms such that data can be available where and when needed by the tasks (including the edge), optimising the involved use of resources.

2.4.4 Audio, visual, and multimodal AI subsystem

devAIce, AUD

devAIce for intelligent audio analytics, including AUD's award-winning openSMILE audio feature extraction toolkit, and modules for sound event detection, acoustic scene classification, and speech analysis. devAIce's VAD module will be improved to work more robustly in the environments where the MARVEL architecture will be deployed, as discussed in 2.4.2. Moreover, new modules will be added that match the MARVEL use cases (e.g., siren detection).

Visual anomaly detection (ViAD), AU

The innovation in the visual anomaly detection component is the design of an efficient underlying model for allowing operation in real-time while maintaining high-performance. Through MARVEL data training, the VAD component is expected to increase performance.

Audio-Visual anomaly detection (AVAD), AU

The innovation in the audio-visual anomaly detection component is the exploitation of both audio and visual data to learn anomaly descriptions for increasing anomaly detection videos. The complementary information appearing in the available audio stream, as sound events can correspond to anomalies that are otherwise ignored by visual anomaly detection methods, is expected to lead to increased performance.

Visual crowd counting (VCC), AU

The innovation in the visual crowd counting component is the design of an efficient underlying model, providing a just-in-time response, for allowing operation in real-time while maintaining high-performance.

Audio-Visual crowd counting (AVCC), AU

The innovation in the audio-visual crowd counting component is the design of an efficient underlying model, providing a just-in-time response, exploitation of both visual and audio data for providing more robust responses in cases where the visual input is of low quality, e.g., low resolution or low scene visibility. The complementary information appearing in the available audio stream, as ambient sound can provide an indication of the appearance of large crowds or not, can lead to increased performance in cases where the visual stream is not of high quality.

Automated audio captioning (AAC). Sound event detection (SED), Sound event localisation and detection (SELD), Acoustic scene classification (ASC), TAU

The main innovation of the AAC is the ability to provide high-level and abstract knowledge through general audio. Through MARVEL, the AAC component is expected to increase its performance using the data provided by MARVEL. The main innovation of ASC, SED, and SELD is the increase in robustness, using the data gathered by MARVEL. As components, they can offer audio-based perception to the final system of MARVEL.

2.4.5 Optimised E2F2C processing and deployment subsystem

GPURegex, FORTH

The innovative feature of the GPURegex component is the ability to accelerate text- and regular expression-based pattern matching, a process that could be part of an event recognition and decision-making engine. By accelerating this heavily computational process, it is more feasible to achieve (near) real-time recognition of events.

DynHP, CNR

The main innovation of DynHP is the ability to train and compress at the same time a DNN model under a fixed memory budget without significant accuracy drop. In MARVEL, this feature is expected to enable the deployment of DNN models in devices with limited resources. DynHP will be further improved by (i) expanding the categories of DNN models that it can handle and (ii) novel training and compression methods the go in the direction of making the process more self-adaptive.

FedL, UNS

FedL implements MARVEL personalised federated learning framework that distributes training of ML/DL models across different MARVEL clients holding their local dataset. Innovations will be reflected in novel federated learning protocols and algorithms. Personalisation will be achieved by tailoring algorithms to different data distributions across different FL clients (e.g., different crowd behaviours in different cities). Performance improvement in terms of response time, throughput, and reliability will be explored by novel hybrid FL-SL (split learning) approaches and also by novel robust adaptive protocols that account for complex operating environments.

Karvdash, FORTH

Karvdash enables user-friendly, consolidated management of distributed services deployed in the entire execution-site continuum, spanning from the Edge to the Cloud. It provides a web-based graphical frontend to coordinate accesses to the E2F2C execution platform, orchestrate service execution in containers from pre-defined templates, and interact with collections of data, which are made available automatically to application containers when launched. The current version of Karvdash already hides the locations and capacities of the actual physical resources engaged in the E2F2C execution environment, as well as the details of access to them. The enhancements to be developed within the MARVEL project will further improve the efficient use of execution resources by applying an optimisation framework that considers trade-offs arising in the selection of alternative execution sites, as well as constraints related to system-level resource availability and application-level requirements.

2.4.6 E2F2C infrastructure

HPC infrastructure - Eagle cluster & LabITaaS, PSNC

Eagle HPC system: Access to world-class processing capabilities allows for training models on large data sets which is the key for achieving good models used on edge devices for inference. Large, easily controlled computing power allows for quick development cycles for both supervised and unsupervised training meaning less time wasted waiting for a new generation of the model.

Open Stack allows for very flexible allocation of computing resources and various class storage services connected with HPC system complementing somehow rigid software environment of the HPC system with flexible-on-demand services that can act as a gateway for the data or stage in the processing chain.

HPC management and orchestration - PSNC SLURM & PSNC OpenStack, PSNC

PSNC SLURM resource management system: contrary to the traditional method of accessing HPC systems via SSH and command-line tools, this implementation has a REST API available making integration with external control, monitoring and complex workflow mechanisms way easier. Additionally, web-token-based authentication allowed for seamless integration with modern web-service based components allowing single identity processing in all MARVEL components thus complying with GDPR and making security issues less of a burden.

PSNC OpenStack: all functionality of Open Stack governed cloud is available via REST API allowing for building flexible flows that adjust used resources exactly to current needs allowing for scripted deployment and utilisation of auto-scaling and reliability mechanisms.

Edge, Fog, and Cloud tier, GRN

Edge: Ease of installing and setting up the data collecting device (e.g., GRNEdge) for data streaming, especially for ad-hoc studies in transport.

Fog: Ease in discovering a device (preferably automatically) and ingesting the data (obtaining and importing data for immediate use or storage in a database). Once a device is added to the system, it automatically starts being processed by the fog-layer, without any further user intervention.

Cloud: GRN will use the cloud services of other MARVEL consortium partners. GRN sees the seamless real-time processing of large amounts of data coming from the fog layer, the auto-scaling - both up and down - depending on the resource requirements, and the communication of decisions taken at cloud level back to the fog/edge layers as the target innovation.

Edge, Fog, and Cloud tier, MT

The innovation consists of monitoring some public spaces and inform the control room of local police if something anomalous happens. This includes **detecting possible dangerous situations** such as gatherings, robberies, aggressions/fights, drug dealing, car accidents. Events are notified:

- to the central station via an alarm
- by creating a custom view on a smart interface to highlight the relevant cameras.

Events are then saved for further analysis.

Edge

Currently, the edge layer is not implemented in the Municipality of Trento infrastructure due to the ageing infrastructure. By adding this layer it is possible to have some real-time analysis on the possibly dangerous situation that will happen in the city. Moreover, with low budget and effort, the edge can be installed in other cabinets located around the city for expanding in the future the usage of the MARVEL framework.

The edge consists of a series of Raspberry Pis installed in the nearest cabinet where selected cameras are installed. It is expected that at the end of the experimentation some modules for the analysis of the video and audio can be installed and executed on the Raspberry Pi. Microphones will be mounted on dedicated devices (probably similar to Raspberry Pi) for audio recording. These devices will also be available for some edge processing although with limited performance.

Fog

The connection of new edges can be easier and also data ingestion and elaboration can be more immediate with respect to the current situation.

Cloud

MT will make use of the cloud tier provided by the MARVEL consortium. Data will be transferred from the fog layer located at FBK premises. Real-time data transfer during inference for some use cases may not be necessary.

The use of cloud services will help in the elaboration and processing of video and audio coming from the fog in terms of resources and data processing capacity.

Edge, Fog, and Cloud tier, UNS

The complete E2F2C infrastructure for the Drone experiment use case was designed specifically for this use case. The main innovations are in the edge tier, in the combination of ground and drone-based synchronised AV data capturing and real-time streaming (AVDrone component of the MARVEL architecture). Future innovations will include installing modules for edge/fog-based data inference (e.g., AVAD, AVCC, devAIce), security (EdgeSec), and anonymisation (AudioAnony, VideoAnony, and VAD devAIce).

2.4.7 System outputs: User interactions and the decision-making toolkit

SmartViz, ZELUS

SmartViz is a versatile data visualisation solution that empowers domain experts to discover patterns, behaviours, and correlations of data items. It consists of a set of visualisation tools developed to allow a more straightforward exploratory analysis of data by using interactive presentations, intuitive monitoring dashboards, configurable visual representations, and collaboration features. Moreover, temporal inspection and predefined views (adaptation of display based on similar previously encountered situations) allow end-users to quickly gain a solid understanding of examined data and benefit from existing stored knowledge.

Data Corpus-as-a-Service, STS

Data Corpus-as-a-Service will be the user's endpoint for accessing, through respective queries, the vast processed audio-visual data that will be stored in MARVEL infrastructure. Enrichment and sharing the Data Corpus will drive research and innovation in multimodal processing and analytics while the latter activities will contribute to the process of benchmarking of edge, fog, cloud, and AI technologies. Furthermore, besides its contribution to the areas of open science and open data, MARVEL Data Corpus-as-a-Service will empower smart cities authorities to better support their societies, deriving new knowledge and advance the existing one, while it will enable them to build and deploy innovative applications that are based on multimodal perception and intelligence.

2.5 Integration guidelines

The purpose of the integration process is to implement and deploy a cohesive and performant, functional E2F2C platform that delivers the envisioned extreme-scale AV analytics on smart cities settings. The challenge of this task is to guarantee the seamless convergence of the total of the innovative technologies and orchestrated infrastructure offered within MARVEL.

To tackle this challenge, we follow an agile approach to platform integration and especially apply the well-established practice of continuous integration. More specifically, the delivered framework is implemented in iterations, with the addition of small increments of services and functionalities at each iteration. This approach respects the natural, incremental way of developing complex systems while enabling stakeholders to monitor the implementation progress, give early feedback, and react promptly to potential technical or other obstacles that may arise. Finally, with continuous integration, qualitative, non-functional aspects of the developed platform are considered early on, including interoperability, scalability, accountability, transparency, responsibility, and performance, thus achieving quality assurance in system development iterations and releases.

Guidelines for the agile approach to integration should encompass both agile processes and corresponding tools to support technical activities. A typical agile, incremental process to software development is depicted in Figure 2.

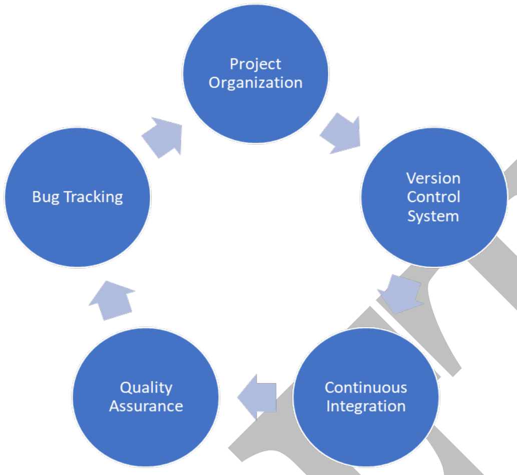


Figure 2. Steps of an agile, iterative development process

In the remainder of this section, an outline of integration steps and guidelines is presented.

Technical project organisation: A detailed roadmap should be in place before the integration efforts are initiated. At integration initiation, the roadmap should contain scenarios and use cases to be implemented, a list of components and their specifications/requirements, required infrastructure needed, and planning of upcoming releases. Initial execution plans for MARVEL use cases/pilots are provided in D1.2.

According to the roadmap, during each iteration, technical tasks should be added to a backlog, tracked while they are in progress, and marked accordingly when completed. This way, at any given point a clear view of the project's current state and upcoming steps is available to participants. A backlog, project organisation software tool should be used to facilitate the above process (e.g., Redmine[7]).

Source version control system: During the execution of integration, all open-source components under development should be stored in a version control system (VCS, e.g., git[8]/GitLab[9]). Furthermore, participating components should ideally be developed and delivered as containerised microservices, in all cases that it is feasible, to further facilitate automation. In case a component cannot meet this requirement, integration and deployment will be realised in a manual or semi-automatic way. Each component should:

- Expose an interface to the other components, preferably REST, or other well-documented methods, if applicable. Otherwise, custom or closed interfaces should be implemented manually and documented separately.
- Be packaged in a Docker container, if applicable.

[7] https://www.redmine.org/

[8] https://git-scm.com/

[9] https://gitlab.com/

- Be self-sufficient and have all needed external libraries and other dependencies already installed in the container.
- Provide detailed documentation of at least its exposed interface, input/output data format, user manual (if applicable), as well as build, deployment, and execution instructions.

Continuous Integration/Continuous Delivery: At each iteration, a functional subset of the platform is going to be delivered for testing and demonstration purposes. As integration advances, the delivered platform will contain an increasing number of components and services, gradually reaching the full version of MARVEL. In this stage, automated tools (e.g., Jenkins[10]) will be used to facilitate the deployment processes, from retrieving the docker image of a component from the VCS to orchestrating the execution of multiple components on the specified infrastructure.

Quality Assurance: It is important to guarantee that each delivered increment meets high standards of quality both in terms of design and code implementation, as well as in terms of execution reliability, performance, and interoperability with other components. To that end, we can use automated tools (e.g., Sonarqube[11]) for code quality, test coverage, etc., in conjunction with the realisation of functional, integration, and acceptance testing efforts.

Bug tracking: During the development and testing of the MARVEL platform, any bug or other system instability should be promptly recorded and made available to developers for fixing. This can be achieved using a backlog tool that is part of the project organisation step.

[10] https://www.jenkins.io/

[11] https://www.sonarqube.org/

3 Sensing and perception subsystem

The sensing and perception subsystem originates from six devices provided by the MARVEL partners. The devices are MEMS microphones and respective interface boards (IFAG), sound event detection at the edge (SED@Edge – FBK), Audio-Visual sensing at the edge (GRNEdge – GRN), Audio Visual Sensing on board drones (AVDrone – UNS), audio recording and annotation (sensMiner – AUD), and traffic objects detection (CATFlow – GRN).

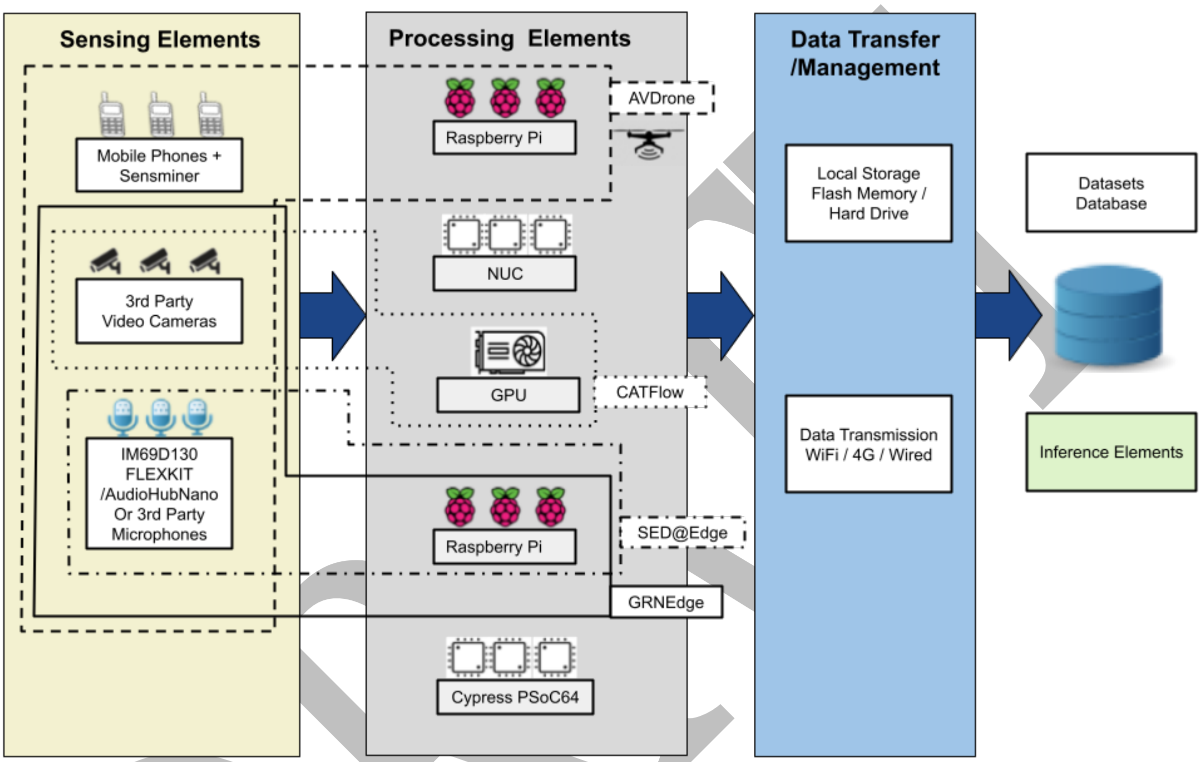

Sensing Elements
Processing Elements
Data Transfer /Management
Mobile Phones + Sensminer
3rd Party Video Cameras
IM69D130 FLEXKIT /AudioHubNano Or 3rd Party Microphones
Raspberry Pi
AVDrone
NUC
GPU
CATFlow
Raspberry Pi
SED@Edge
GRNEdge
Cypress PSoC64
Local Storage Flash Memory / Hard Drive
Data Transmission WiFi / 4G / Wired
Datasets Database
Inference Elements

Figure 3. Summary of the Sensing and perception subsystem in MARVEL

The flow diagram in Figure 3 depicts a set of components that interact and integrate to provide the sensing and/or perception function desired at the edge. On the left of the diagram are the sensors (in yellow), namely microphones and cameras, whose output is passed on to the processor boards on which the selected software is executed (in grey). The output from the processor boards is then either stored locally on storage media or transmitted over an IP channel to the fog layer. The data on the local storage is transferred offline in batch mode to the fog. The data on the local storage is manually transferred to the fog. The currently available edge devices (listed in the above paragraph) are composed of a selection of the components in Figure 3, which expresses the similarities and differences between the subsystems.

The output of MEMS microphones, which is DPM encoded, is converted to USB format via IFAG's Audio Hub Nano board. The combination of IFAG microphones and the 3rd party Cypress board allows for the development of SELD component. SED@Edge takes the output of MEMS microphones and processes the sound signals to detect events of interest at the edge, thus reducing the transmission rate required. GRNEdge takes input from microphones and cameras, synchronises and combines the inputs into an AV format on a Raspberry Pi board, and either stores or streams the AV data. Similarly, the AVDrone mounts the camera and microphone on the drone and transfers to the fog layer. In addition, the output from ground installed microphones is processed on cell phones using the sensMiner app, which is also used for events' tagging during data collection. The subsystem allows for the sensing and perception system to evolve throughout the MARVEL project. For example, the CATFlow software can be installed on the GRNEdge (upgraded with GPU) to provide the object detection function at the edge, and similarly with AVDrone for edge-based security (EdgeSec), anonymisation (Audio/VideoAnony), AI (e.g., AVAD), etc. The following sections provide more details on the devices.

3.1 Advanced MEMS microphones

For the acquisition of audio data, IFAG provides the MEMS microphone IM69D130 and for data pre-processing the AudioHubNano4D. The block diagram of the IM69D130 is depicted in Figure 4 and the one of the AudioHubNano4D in Figure 5.

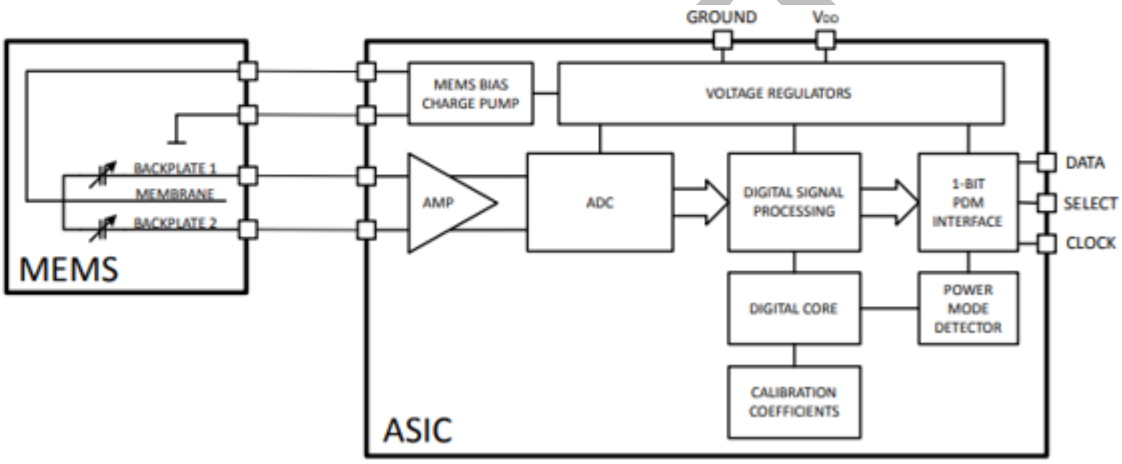


Figure 4. Block diagram of advanced MEMS microphones - IM69D130

The IM69D130 MEMS microphone comes with the following features:

- High SNR (69 dB)
- Wide dynamic range
- Low distortion (below 1 % distortions at 128 dBSPL)
- High acoustic overload point.

and its main benefits are:

- High fidelity and far-field audio recording
- Noise-free and ultra-low latency audio signals for advanced audio signal processing
- Ultra-low group delay for latency-critical applications
- No analog components required.

These microphones will be connected to the AudioHubNano4D.

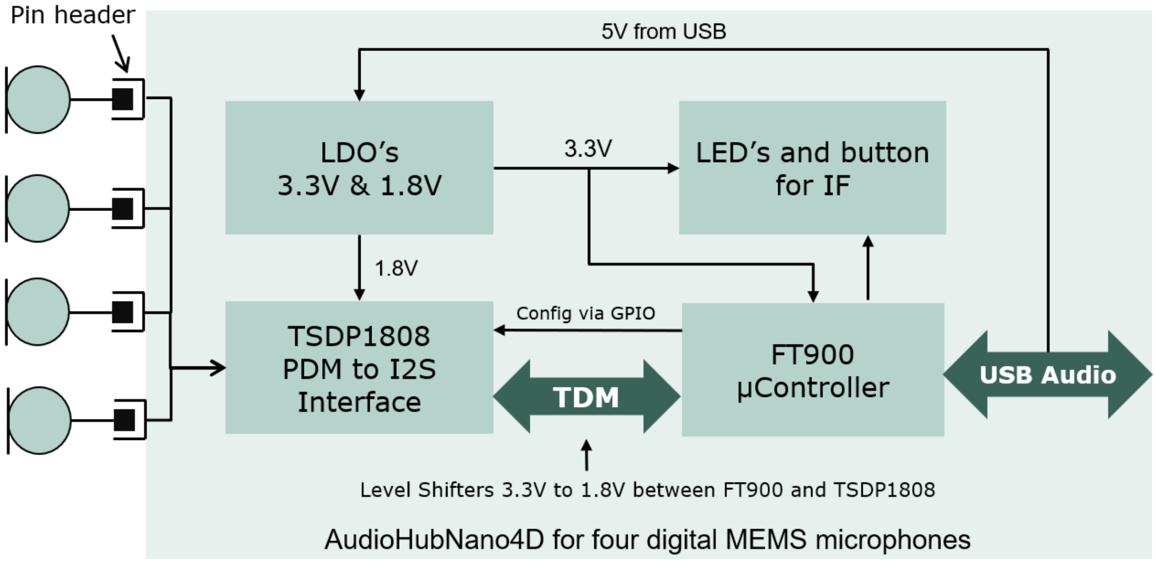


Figure 5. Block diagram of AudioHubNano4D

Which provides the following features:

- Audio streaming over the USB interface
- 48 kHz sampling rate
- 24-bit audio data (stereo)
- Mode switch for toggling between normal mode and low power mode with four pre-defined gain configurations
- LED indication for the configured gain level in normal mode and low power mode
- Volume unit meter display with onboard LEDs
- Powered through Micro USB

Two to eight MEMS microphones IM69D130 as shown in Figure 6 will either be connected to the edge node via the Flex evaluation kit (EVAL_IM69D130_FLEXKIT) depicted in Figure 7 or the AudioHubNano, see Figure 8.

The Flex evaluation kit provides an output in Digital PDM format, with the four output pins VDD, Data, Clock, Select, GND, and has 6 µs group delay at 1 kHz.

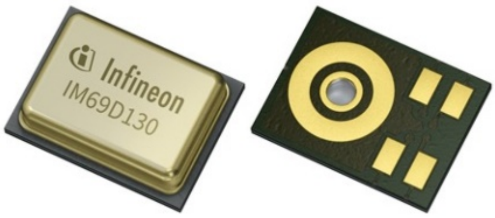


Figure 6. Image of the MEMS microphones IM69D130

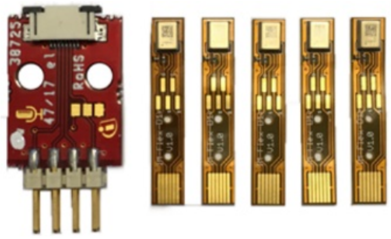


Figure 7. Image of the EVAL_IM69D130_FLEXKIT

The AudioHubNano is connected and powered via USB. Two MEMS microphones can simultaneously connect to it. In the future, a new version of the AudioHubNano, the AudioHubNano4D (find block diagram in Figure 5) will be provided that will allow the connection of four MEMS microphones simultaneously.

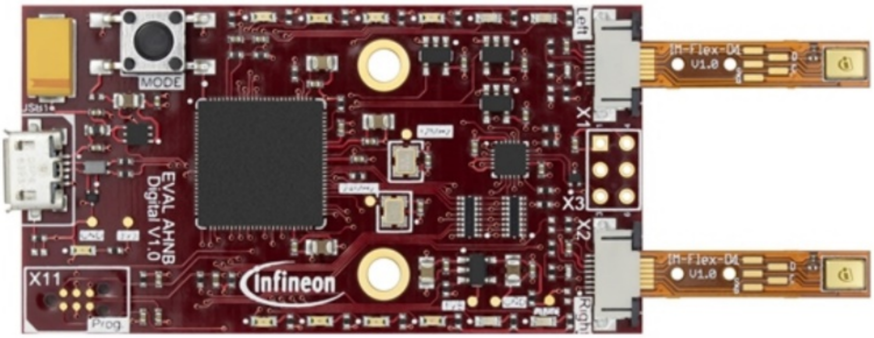


Figure 8. Image of AudioHubNano

The provided data will be processed in the edge. For this task, either up to eight microphones will be connected to a *Cypress PSoC64 Standard Secure – AWS Wi-Fi BT Pioneer kit* (PSoC 64 + CYW434W Wi-Fi + BT Combo Chip), find the block diagram in Figure 9, or via an RPi.

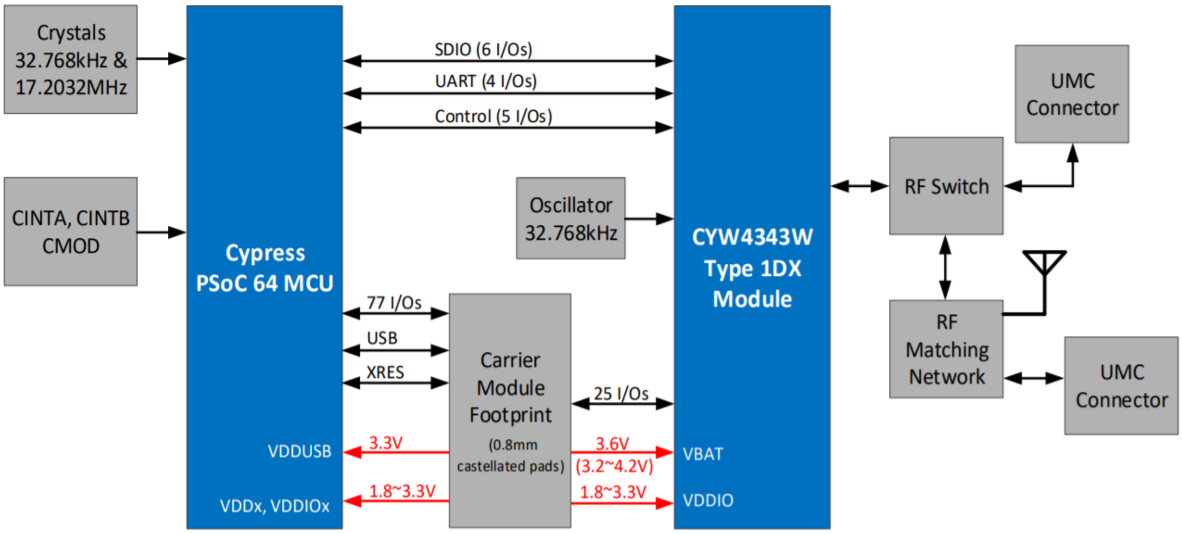


Figure 9. Block diagram of envisioned Edge processing

3.2 SED@Edge

The SED@Edge kit consists of a PCB board with embedded MEMS microphones acquiring at 16kHz. The processing unit is a low-power microcontroller (MCU), in which a neural network-based multiclass classification algorithm runs in real-time. The kit is able to detect classes ranging from a dog barking to a car passing for a total of ten urban environment-related classes. Nevertheless, the classifier can be retrained to change the type of classes relevant to MARVEL's application domains.

The SED@Edge kit is a standalone kit that directly acquires audio from MEMS microphones and outputs the classes of the events present in the acoustic scene. Thus, it is an edge component that can communicate with the rest of the MARVEL's infrastructure through common protocols (ranging from serial to LoRa/Wi-Fi).

Thus, as illustrated in Figure 10 below, the input of the SED@Edge kit is raw audio signals while the output is the detection's probability for every event in the predefined set related to the application domain. The event's probability is inferred from the raw audio waveform by computing the frequency representation of the signal, which is then fed into a distilled recurrent convolutional neural network.

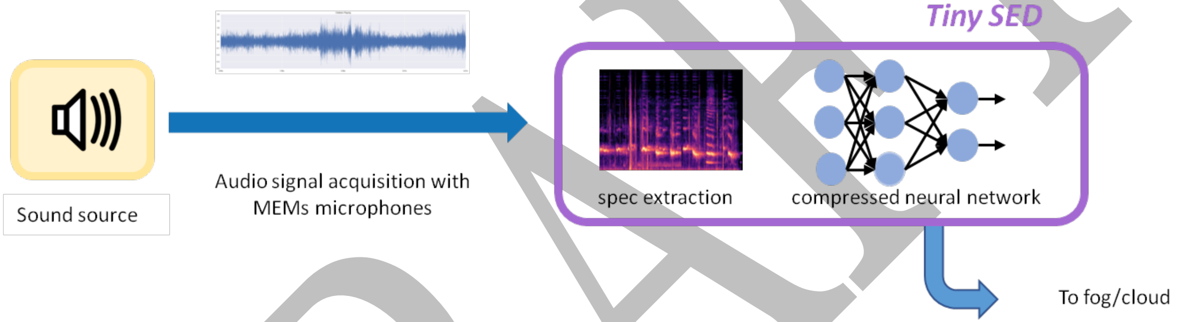

Sound source
Audio signal acquisition with MEMs microphones
Tiny SED
spec extraction
compressed neural network
To fog/cloud

Figure 10. Schematic representation of data processing of the SED@EDGE component

3.3 GRNEdge

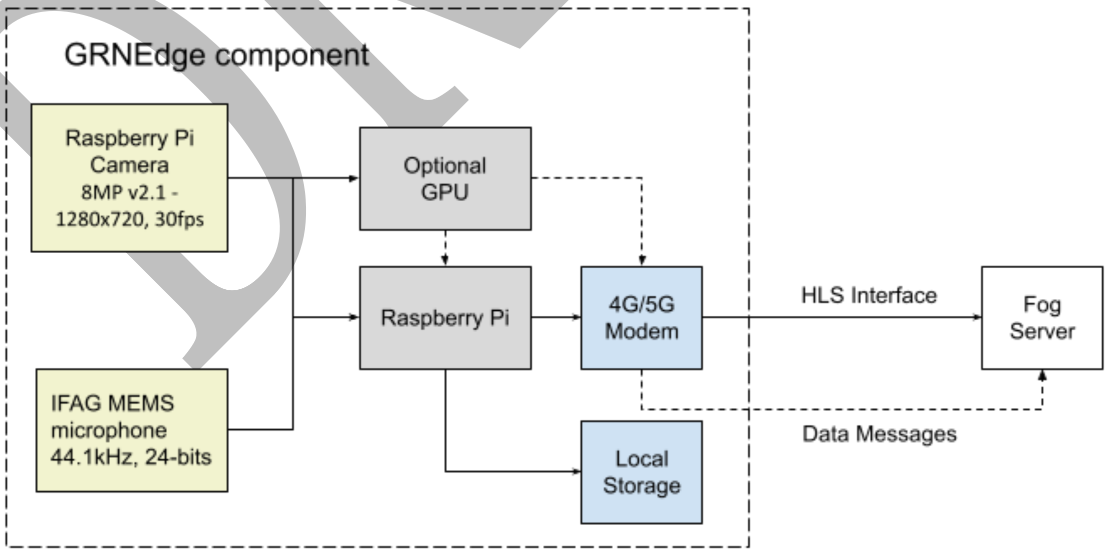


Figure 11. High-level schematic of the GRNEdge sensing device, with an optional GPU

The GRNEdge, illustrated in Figure 11, is an audio-video sensor that collects data, encodes data into MKV format, and either stores the data on local storage at the edge or streams the data over wireless channels, WiFi, and LTE. The on-board processor (typically Raspberry Pi) board is used to synchronise the audio and video streams but can also be used for lightweight processing at the edge. The HLS protocol is used to stream the audio-video data via the radio modem. Optionally, the GRNEdge device can be upgraded with a GPU such that AI models are executed at the edge. In the latter case, it may no longer be necessary to stream the AV data and instead data messages are used to transfer the structured non-binary data.

3.4 AVDrone

AVDrone, illustrated in Figure 12, is a data capturing and streaming component that will be used for audio and video monitoring of public events. The component consists of data capturing devices organised into two groups. Firstly, the DJI M600 drone will be equipped with Raspberry Pi v.2.1 camera for aerial video recording as well as with IFAG MEMS microphones attached at AudioHub Nano IFAG boards. Secondly, several IFAG MEMS microphones will be deployed on the ground in order to perform additional data collection that can support event localisation. Besides that, ground-based additional RPi cameras will be deployed on the ground as an additional hardware for more precise detection. The AVDrone component will have an interface to the sensMiner app, as it will be exploited for acoustics recordings, annotation, and recording GPS tags.

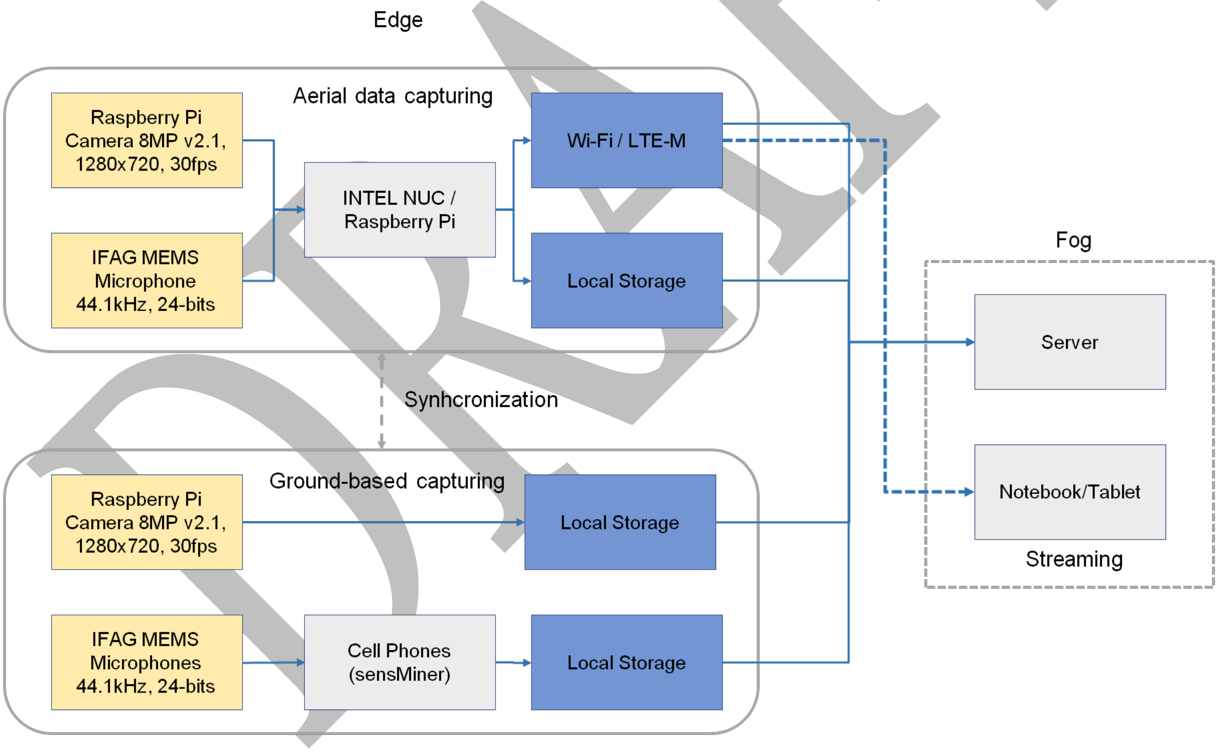


Figure 12. Block diagram of AVDrone

AVDrone component will support video streaming to the fog layer using Wi-Fi and we will also explore the possibility to complement the WiFi connection with LTE-M. As it is equipped with INTEL NUC, the component will support edge processing and possibly AI analytics offered by other MARVEL components (e.g., AVAD, AVAC). Figure 13 shows a snapshot of the video stream obtained using AVDrone.

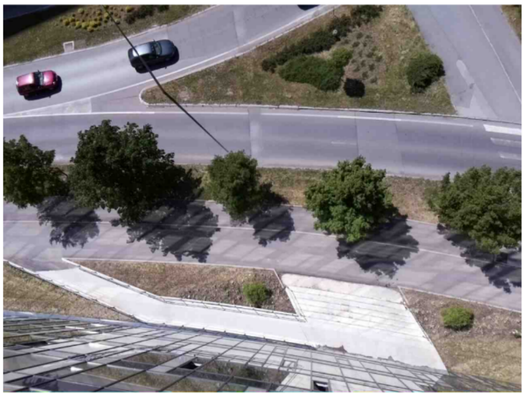


Figure 13. AVDrone video stream snapshot

3.5 sensMiner

SensMiner is an Android app developed by AUD to record environmental acoustics as well as user annotations. Audio is being recorded and the user can in parallel annotate audio and store the corresponding segment in the phone memory. SensMiner is a standalone app used *exclusively* for data collection. In MARVEL, it will be used as part of AVDrone-based data collection within the UNS Drone experiment. During framework runtime, it will not interact with other MARVEL components. Furthermore, SensMiner does not perform any processing or analytics.

SensMiner collects raw audio, GPS information, and user tags. Audio is recorded at 16bit PCM format at 44.1kHz. All data is stored as JSON files on the user's smartphone and need to be manually transferred.

3.6 CATFlow

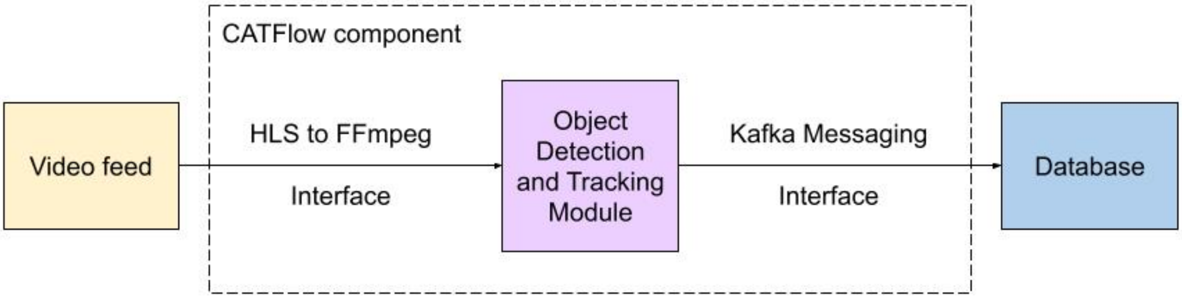


Figure 14. Dataflow in CATFlow and the interfaces to other components

CATFlow is a system designed by GRN to analyse video footage of roads in real-time, to determine how road users use the given infrastructure. Figure 14 depicts the dataflow in CATFlow and the interface with other components. The input is a camera stream. At the time of writing, the HLS streaming protocol is used, and then the frames are fetched using FFMPEG. Each frame is sequentially fed first into an object detector and then into a multi-object tracker in order to track entities across the scene. Internally, it uses entry and exit lines or boundaries. Having both an entry and an exit line, and knowing the distance between these, the vehicle speed can be calculated. Pedestrians on the other hand are tracked across the screen since pedestrians follow a random path. No personal data is computed. Since there can be many instances of CATFlow, Kafka is used as a message broker. The information is then stored into an OLAP database, ready to be consumed by an API to create reports or visualisations.

4 Security, privacy, and data protection subsystem

This subsystem includes all the MARVEL's components which address issues about security, protection of the recorded data and systems in place, and preservation of the privacy of the citizens.

In practice, the subsystem addresses two different problems.

- **Cybersecurity**: ensuring that the MARVEL framework is safe against cyber-attacks attempting to get access to the system to compromise its functionalities or to access the data. This is mainly achieved by the EdgeSec component applied across the whole E2F2C architecture, achieving end-to-end system security.
- **Privacy**: preventing the misuse of the data processed by the system as well as limiting critical effects in case of data leaks. Privacy preservation is addressed by the **audio and video anonymisation** components as well as by the voice activity detection (VAD) module.

In MARVEL we are addressing the risk of identifying citizens from the recorded data. For video data, this will be achieved by locating face areas and applying either face blurring or face conversion as soon as possible in the MARVEL infrastructure, depending on the available computing resource of the deployed devices. In the case of plates, blurring will be applied. For the audio data, **voice activity detection** (VAD) will first be applied to select relevant segments of the audio stream. This component can safely run on edge devices. One approach to audio anonymisation is to remove completely the speech segments. This would also remove privacy-related spoken content, but would, on the other hand, also remove all useful information that might exist in the audio signals. Alternatively, voice conversion or signal-based anonymisation techniques will be applied to the speech segments. This strategy is expected to preserve the acoustic environment but will not remove possible privacy-compromising content. For voice conversion, similar issues as in video anonymisation apply in terms of processing power.

The block diagram of this subsystem is reported in Figure 15. Note that EdgeSec ensures cybersecurity from the sensor up to the cloud processing services. Anonymisation can be deployed either on edge or fog (or distributed across the two) depending on the computational power availability. Details about the single components are reported in the following sections.

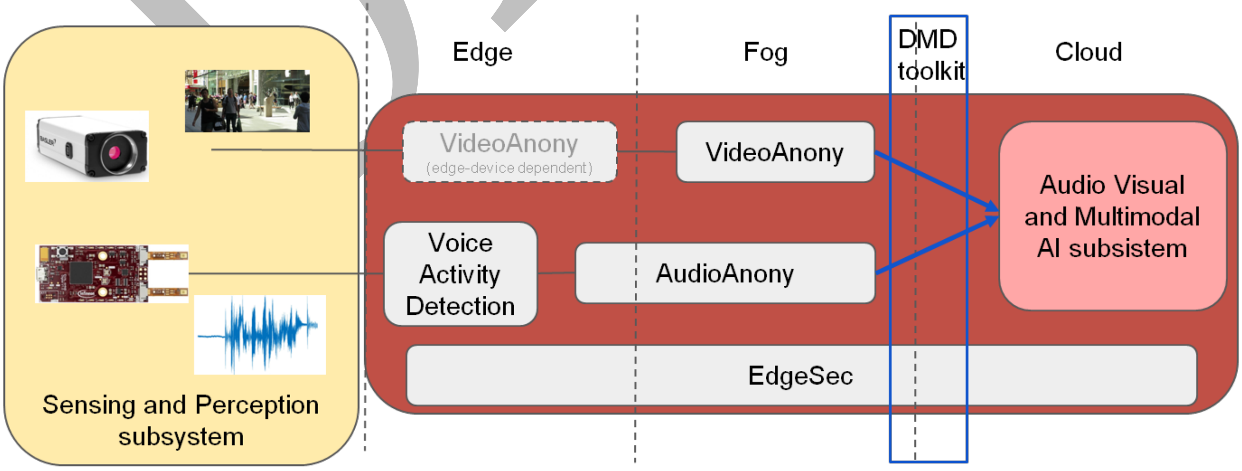

Edge
Fog
DMD toolkit
Cloud
VideoAnony (edge-device dependent)
VideoAnony
Voice Activity Detection
AudioAnony
Audio Visual and Multimodal AI subsistem
EdgeSec
Sensing and Perception subsystem

Figure 15. Security, privacy, and data protection subsystem

4.1 EdgeSec

EdgeSec is a security framework that aims to provide a platform of trust. It is based on the open-source SCONE framework[12] which enables secure computing and focuses on the preservation of the confidentiality and integrity of the applications by leveraging Intel Software Guard eXtensions[13] (SGX) security features. Furthermore, each physical or virtual host running EdgeSec will be part of a peer-to-peer VPN network in order to fully encrypt the network communication in the whole E2F2C architecture. Apart from the edge layer devices, EdgeSec can be deployed on any other device regardless of its place in the processing layer. Its two main goals are: (i) to encrypt data and network traffic in order to protect them from unauthorised access, retain confidentiality and integrity of data in transit and verify that an unknown and potentially malicious entity cannot remotely access any host; and (ii) to attest programs in order to ensure that only the correct, unmodified programs are being executed in a genuine SGX enclave. A high-level overview of the EdgeSec component is illustrated in Figure 16.

There are some minimum hardware and software requirements to be considered when deploying the EdgeSec component. First and foremost, EdgeSec leverages the security features of Intel SGX so the bare minimum requirement is an SGX-enabled processor. It is recommended to also enable Intel SGX through the BIOS of each physical machine, as this will facilitate the testing, development, and deployment procedures. Moreover, some primary software dependencies that should be met are a long-term support version of Ubuntu, ideally v18 or newer, or any other Linux-based operating system (e.g. Debian), the latest Docker engine package (docker-compose package would also be beneficial but not required), latest Intel SGX drivers and access to the physical (or virtual) machine through a privileged account.

EdgeSec does not perform any kind of data processing apart from the encryption and decryption of the data when host-to-host communication is necessary. These operations are totally transparent to the users of this component, so there is no need for any interface or connection with any other MARVEL component. When deployed, each host will be given a unique private IP address and every communication between this and any other host in the MARVEL E2F2C architecture will be tunnelled directly from/to this private IP address.

[12] https://sconedocs.github.io/

[13] https://software.intel.com/content/www/us/en/develop/topics/software-guard-extensions.html

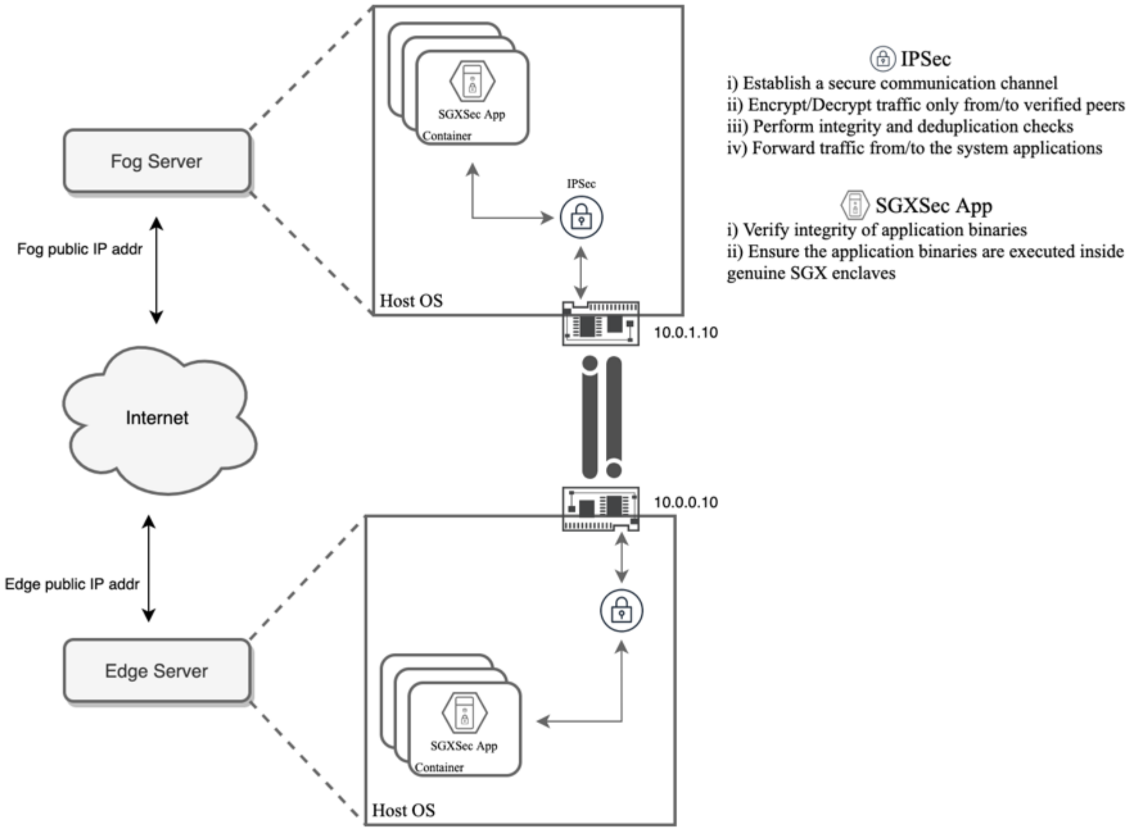

IPSec
i) Establish a secure communication channel
ii) Encrypt/Decrypt traffic only from/to verified peers
iii) Perform integrity and deduplication checks
iv) Forward traffic from/to the system applications
SGXSec App
i) Verify integrity of application binaries
ii) Ensure the application binaries are executed inside genuine SGX enclaves
Fog Server
Fog public IP addr
Internet
Edge public IP addr
Edge Server
SGXSec App
Container
IPSec
Host OS
10.0.1.10
10.0.0.10
SGXSec App
Container
Host OS

Figure 16. High-level overview of EdgeSec architecture

4.2 VideoAnony

VideoAnony is a software for anonymising video content, i.e., removing personal data from any input video, by either blurring the detected faces or converting the detected faces to a different identity.

The software takes as input the raw video feed and then processes the video in a frame per frame manner following two main steps: (i) person/face detection, i.e., to localise persons and their faces on the image frame, (ii) face obfuscation by blurring the detected faces (the ready-to-use technique) and advanced GAN-based face conversion technique (the technique that will be researched and developed under MARVEL). The software will output anonymised videos that can be released and distributed among the MARVEL consortium, where further vision-based AI tasks can make analysis on top.

The software can be potentially deployed on the edge if the edge device is equipped with a GPU to allow for real-time performance. For example, an NVIDIA Jetson Nano with 4GB memory can serve the processing for object detection in real-time, while a GAN-based face obfuscation technique may require model compression before the deployment on the edge. The software can also be located at fog level for offline dataset processing to anonymise recorded videos. The component is illustrated in Figure 17.

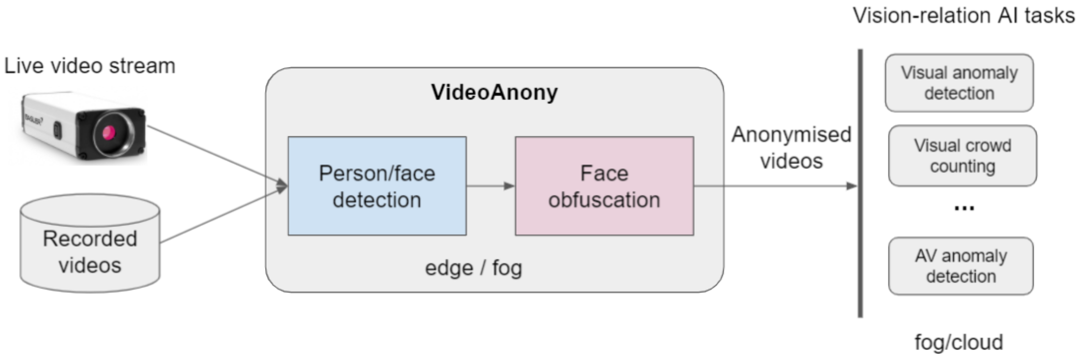


Figure 17. Video anonymisation applied at the edge or fog

4.3 AudioAnony

The audio anonymisation tool receives as input the digitalized audio samples captured by the MEMS microphones and provides an audio stream as output. The stream is made available for further processing to other components.

Solutions based on signal processing suitable for any edge processing exist, with limited anonymisation capabilities. We plan to investigate the use of any-to-any voice conversion strategies to achieve anonymisation. At the moment, it is not possible to define the computational requirements as established algorithms and models with sufficient reconstruction accuracy do not exist. Probably these approaches would not fit on a microcontroller like the STM-32.

Keeping in mind that the closer to the source the better, anonymisation can be applied in different layers of the MARVEL infrastructure, eventually distributing the processing, depending on the requirements and the available resources. Figure 18 illustrates audio anonymisation applied at the edge, while Figure 19 illustrates audio anonymisation applied at the fog.

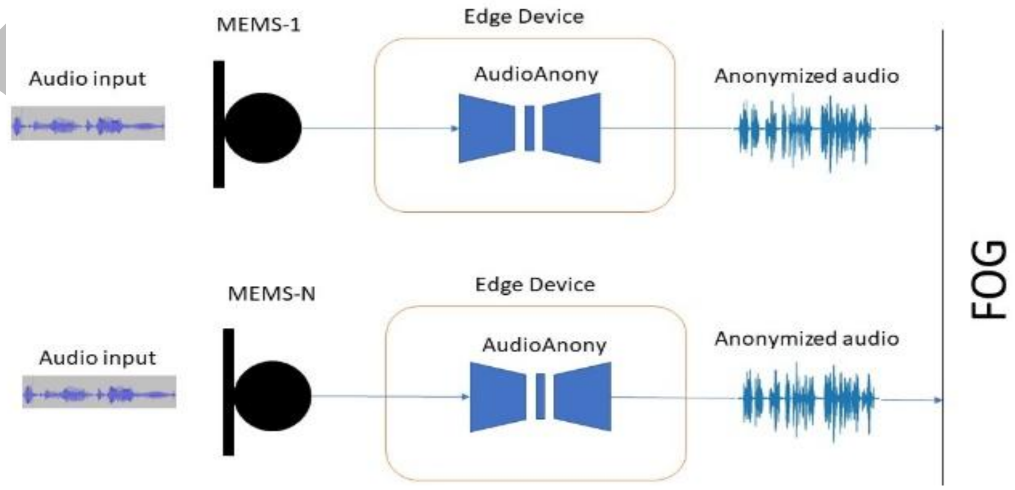


Figure 18. Audio anonymisation applied at the edge

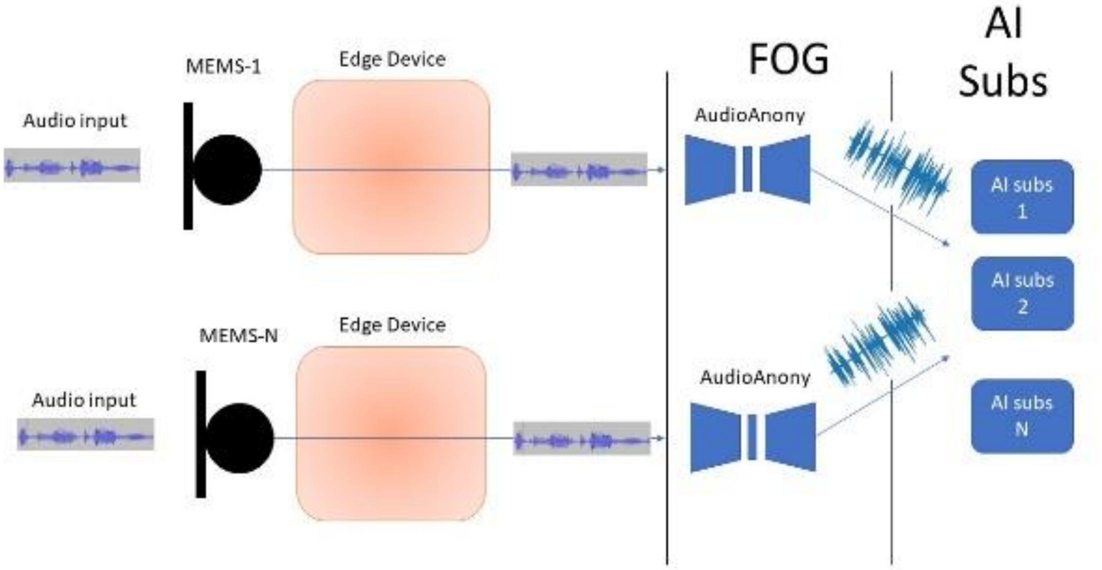


Figure 19. Audio anonymisation applied at the fog

Other solutions exist that apply anonymisation both at edge and fog, possibly with different approaches. Based on the specific operational conditions, ad-hoc solutions will be implemented to maximise the data protection given the computational constraints. For example, a less effective but computationally light approach could be applied to the edge to secure the data during an edge to fog transmission, and then a more powerful anonymisation could be applied before moving the data to the processing units or to the storage. Furthermore, generative models are based on encoder-decoder architectures. One possible strategy to deal with computational power and bandwidth limitations would be to run the encoder at the edge and the decoder at the fog tier. These concepts are partially valid also for the video anonymisation component.

4.4 Voice activity detection (devAIce)

AUD’s voice activity detection (VAD) technology, made available through the devAIce SDK, will be used to detect voice in the audio recordings, both in a batched (offline) and streaming (online) fashion. These voice segments will then be correspondingly removed or otherwise properly anonymised.

This VAD module works in real-time on cloud and fog nodes and appropriately powerful edge nodes (e.g., Raspberry Pis). It accepts as input an audio stream and outputs start/end times of detected speech segments. These segment times are then propagated to a voice anonymisation module to appropriately anonymise them.

An illustration of AUD’s VAD place inside the MARVEL architecture is shown in Figure 20. Assuming a raw audio stream captured by on-premise MEMS (or other) microphones, VAD is the first step in the audio(-visual) processing chain. It computes start/end times of voiced segments and forwards this information to the voice anonymisation module, which appropriately removes private information. The anonymised audio is afterward streamed to other modules that are responsible for audio analytics (e.g., sound event detection and acoustic scene classification).

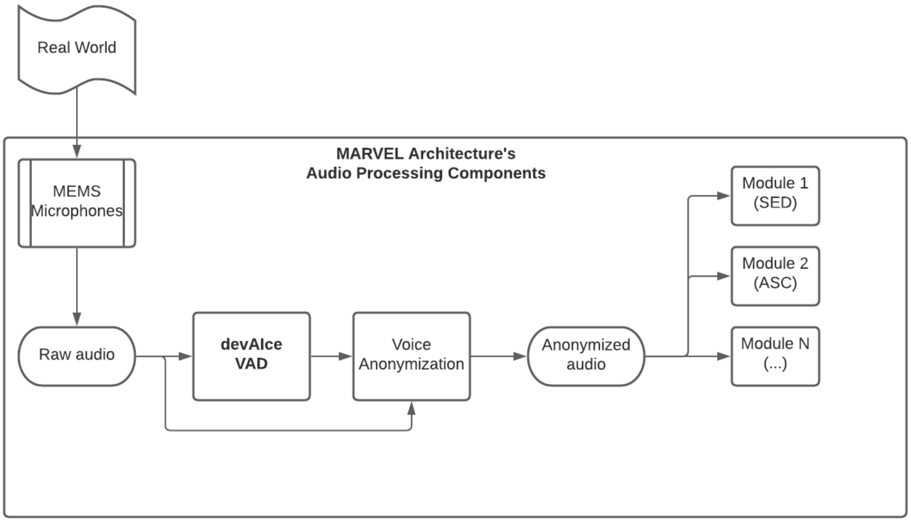


Figure 20. devAIce VAD inside the MARVEL architecture

5 Data management and distribution subsystem

The data management and distribution subsystem includes the DFB, StreamHandler, DatAna, and HDD components. The goal of this subsystem is to handle massive amounts of data coming from various sources and deal with their management and proper distribution. Among others, the subsystem will consider the variety of formats and frequency of data.

DFB is a customisable component that implements a trustworthy way of transferring large volumes of heterogeneous data between several connected components and the permanent storage. StreamHandler provides the hooks for interconnecting, storing, transforming, and processing data as well as training, validating, and executing ML and DL algorithms, resulting in a full Big Data solution with AI capabilities. DatAna is a framework based on the usage of the Apache NiFi ecosystem to allow the processing of data flows between the edge/fog and the cloud. HDD is a set of distributed adaptive data delivery and access algorithmic schemes for guaranteeing real-time delay requirements while effectively prolonging network lifetime in wireless industrial edge networks.

As explained in more detail below, HDD interconnects with the rest of the Data Management and Distribution subsystem components through the common infrastructure of a network plane. An array of functionalities provided by the Apache Kafka module for handling large volumes of data streams in real-time facilitates the merging of DFB and StreamHandler components and strengthens the emphasis on real-time applications. DatAna's NiFi and MiNiFi enablers, the data flows of which might be directed to other systems using dedicated processors for data egressing to Kafka, could further facilitate the tight co-existence of the subsystem components.

An overview of the data management and distribution subsystem, also showing the main interactions and data flows, is illustrated in Figure 21.

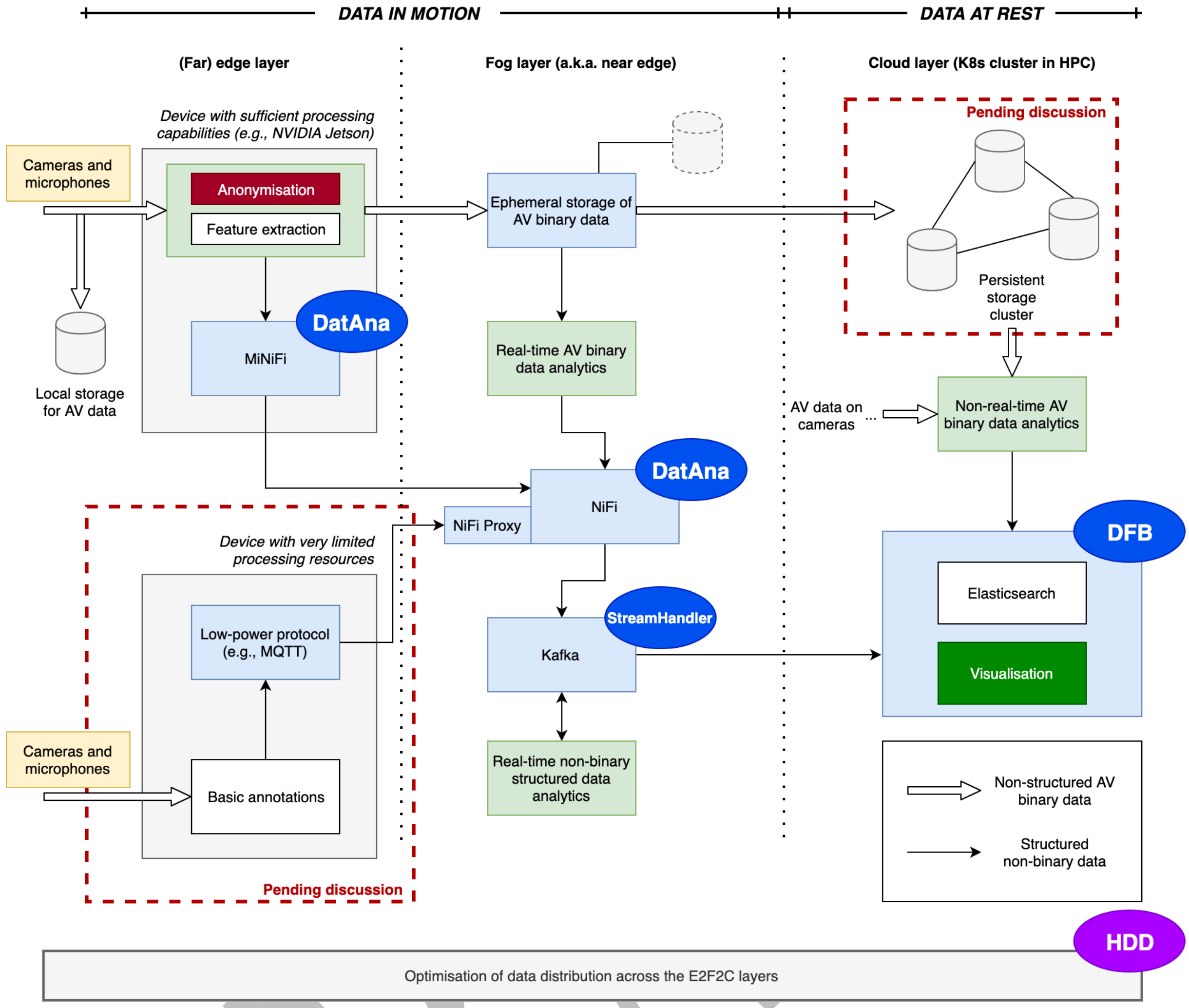

DATA IN MOTION
DATA AT REST
(Far) edge layer
Fog layer (a.k.a. near edge)
Cloud layer (K8s cluster in HPC)
Device with sufficient processing capabilities (e.g., NVIDIA Jetson)
Cameras and microphones
Anonymisation
Feature extraction
DatAna
MiNiFi
Local storage for AV data
Ephemeral storage of AV binary data
Real-time AV binary data analytics
Pending discussion
Persistent storage cluster
AV data on cameras
Non-real-time AV binary data analytics
DatAna
NiFi
NiFi Proxy
DFB
Elasticsearch
Visualisation
Device with very limited processing resources
Low-power protocol (e.g., MQTT)
StreamHandler
Kafka
Cameras and microphones
Basic annotations
Real-time non-binary structured data analytics
Non-structured AV binary data
Structured non-binary data
Pending discussion
HDD
Optimisation of data distribution across the E2F2C layers

Figure 21. An overview of the Data management and distribution subsystem

We distinguish between data in motion and data at rest. *Data in motion* refers to the analytics that is processed in real-time, e.g., for detecting anomalies or automated feature extraction. *Data at rest* refers to the analytics that is triggered manually by an operator, possibly in response to an automatic trigger received. Furthermore, we distinguish between *non-structured AV binary data*, which are the raw audio and video streams produced by the microphones and cameras in the field, possibly anonymised at the edge, and the *structured non-binary data*, which are the features extracted by the raw data. The former, i.e., non-structured AV binary data, are stored only for the minimum amount of time required by the respective use case, which can vary from do-not-store-at-all to days and weeks. The reason for keeping at a minimum the time window during which AV binary data can be accessed after collection is that they are continuous high-bandwidth streams, which occupy a significant amount of storage. On the other hand, long-term storage solutions can be considered for structured non-binary data, which have smaller rate requirements. The tools employed by DatAna, DFB, and StreamHandler all manage only structured non-binary data natively: a technical discussion is pending in the related WPs on how to handle at best the non-structured AV binary data, in accordance with their volume, velocity, and variety (VVV) characteristics as identified by the data providers. As can be seen, the HDD contribution is cross-cutting with respect to the other components since it deals with the optimisation of data placement across all the layers.

The figure above is only for illustrative purposes, and it does not refer to any pilot specifically. The actual deployment of each pilot will depend on the specific characteristics, technical constraints, and resources available.

5.1 Data Fusion Bus

The Data Fusion Bus (DFB) is a customisable component that implements a trustworthy way of transferring large volumes of heterogeneous data between several connected components and the permanent storage. It comprises a collection of dockerized, open-source components which allow easy deployment and configuration as needed.

DFB's architectural design addresses several challenges that are raised by both the large volume and the heterogeneous nature of data from different sources, taking into consideration the needs and restrictions of the employed components. The main addressed challenges include:

- seamless aggregation of data with different structures or formats
- a cluttering threat to the components due to the quantity of the input data
- access of data through a common, safe, accessible interface.

Inherent to DFBs design is the efficient handling of the enormous volume of the data that need storage and manipulation, as well as mechanisms to remediate potential bottlenecks, lag, or high demand on network traffic. These design decisions enable horizontal scalability while providing a solution that is cloud-native with stateless components capable of being deployed with flexibility. DFB follows the middleware approach by aligning data streams for time and granularity and creating a user interface that serves as the interface of the platform, customised to aggregate multiple streams, thereby allowing seamless service of data to the network analysis and visualisation.

The key capabilities of DFB are:

- Data aggregation from heterogeneous data sources and data stores
- Real-time analytics, offering ready-to-use ML algorithms for classification, clustering, regression, and anomaly detection
- An extendable and highly customisable User Interface for Data Analytics, manipulation, and filtering, as well as functionality for managing the platform
- Web Services for exploiting the platform outputs for Decision Support
- Applications for Smart Production, Digitisation, and IoT, among others.

The key modules of DFB are:

a. *Apache Kafka*, an open-source framework for stream processing
b. *Elasticsearch*, a distributed, multitenant-capable, full-text search engine
c. *DFB Core & UI*, implementation of a REST API and a client GUI, respectively, for management and monitoring of the DFB components
d. *Keycloack*, an open-source software product that provides single sign-on to applications and services.

Figure 22 depicts DFB's overall inner architecture and its relation to other components of the MARVEL platform. This figure shows DFB's main modules mentioned above, as well as its interfacing with edge devices that may produce non-binary data and other fog/cloud components that provide their processed data into DFB.

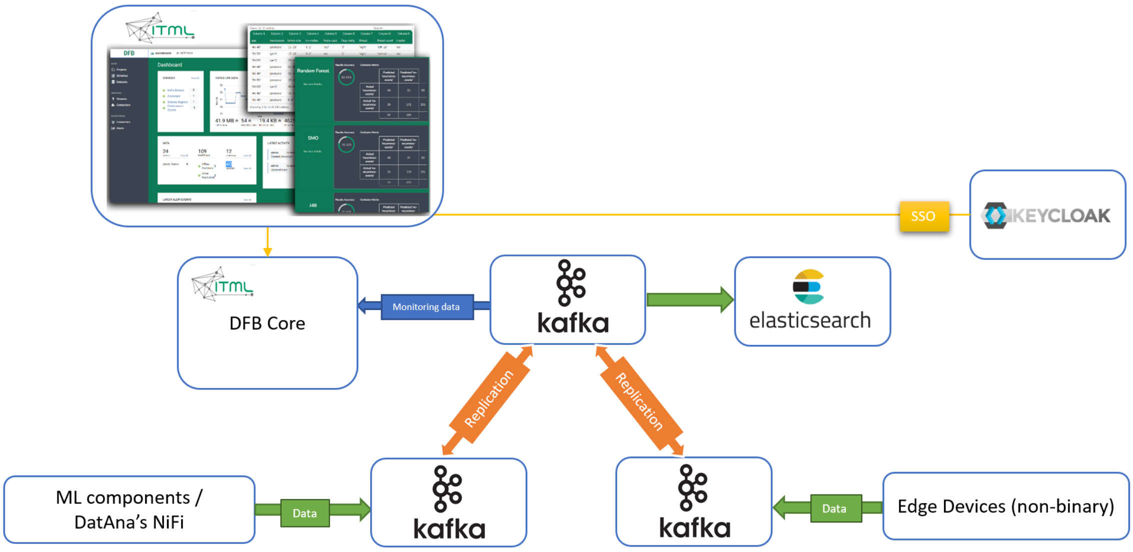


Figure 22. DFB's overall architecture

Inbound and outbound interfaces and relation with other MARVEL components.

The main inbound interface for DFB is Kafka's messaging system that is based on the publish-subscribe pattern. More specifically, any MARVEL data-producing component, including real-time event detection and data analytics components, connects to DFB and publishes any relevant data to a specific, predefined topic.

Regarding outbound interfaces, in the general case, any data-consuming component can subscribe to a topic and receive instant updates on published data. Within the context of MARVEL, data collected from Kafka brokers is subsequently passed onto an Elasticsearch Logstash Kibana (ELK) stack for storage and further processing and visualisation. Although DFB offers its own graphical UI for visualisation of aggregated data, collected streamed and stored data can be made available to MARVEL's visualisation components (e.g., SmartViz).

Finally, DFB offers a complete, standalone Single Sign-on module based on Keycloack open-source product to ensure authenticated and authorised access to fused data.

Processing, handling, fusion, and synchronisation of AV data

As DFB is typically deployed on the cloud, it does not offer any edge processing options. It is designed and optimised for handling non-binary data that is streamed to the Kafka interfaces in real-time. For the MARVEL use cases, DFB will not directly process or handle AV data; instead, any results of AV processing from ML components can be made immediately available for real-time analytics or stored for later filtering, processing, searching, and visualisation.

Compatibilities, complementarities, etc. with other data management platforms

Significant compatibility between DFB and INTRA's StreamHandler has been identified in the early stages of the specification of MARVEL architecture. As both components address the issue of scalable collection, fusion, and management of large volumes of non-binary data, a hybrid solution for MARVEL is followed. Data collection and aggregation will be undertaken by DFB, reinforced by a set of StreamHandler's security and performance features that has been gathered over this component's application in large-scale domains.

There is also an important complementarity with ATOS's DatAna component, as they can naturally fit to the publish-subscribe pattern for data management. More specifically, DatAna's NiFi module can be connected to DFB's Kafka inbound interface and publish its data collected at the edge/fog layer to any relevant topic defined in DFB at the cloud layer.

5.2 StreamHandler

INTRA's StreamHandler Platform provides the hooks for interconnecting, storing, transforming, and processing data as well as training, validating, and executing machine learning and deep learning algorithms, resulting in a full Big Data solution with Artificial Intelligence Capabilities. StreamHandler is a high-performance (low latency and high throughput) distributed streaming platform for handling real-time data based on Apache Kafka. It can efficiently ingest and handle massive amounts of data into processing pipelines, for both real-time and batch processing. The platform and its underlying technologies can support any type of data-intensive ICT services (Artificial Intelligence, Business Intelligence, etc.) from cloud to edge. In the following paragraphs, an overview of INTRA's StreamHandler Platform is presented, together with a description of the high-level architecture and the individual platform components, their interactions, as well as the key technologies involved.

The key capabilities and features offered by the platform are:

- Real-time monitoring and event-processing
- Interoperability with all modern data storage technologies and popular data sources
- Distributed messaging system
- High fault-tolerance - Resiliency to node failures and support of automatic recovery
- Elasticity - High scalability
- Security (encryption, authentication, authorisation)

In particular, the platform, whose architecture is illustrated in Figure 23, is a fully featured industrial grade solution: (i) which is capable to scale out and accommodate various and from different domains big data, interoperating with all modern data storage technologies as well as other persistence approaches and (ii) can support all important Big Data languages including Python, Java, R, and Scala as well as other traditional programming approaches.

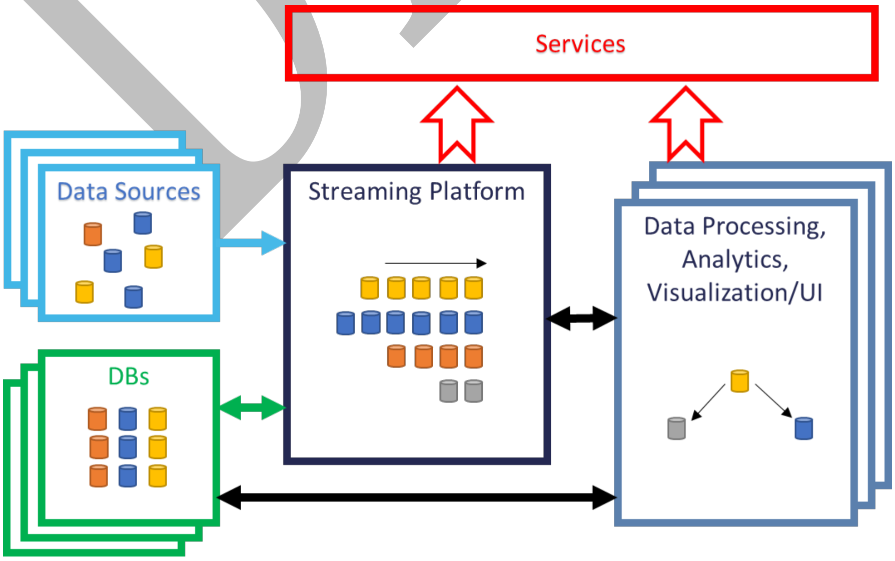


Figure 23. StreamHandler's overall architecture

Data sources, Data stores, Data Analytics, and Visualisation applications as well as the supporting Processing Infrastructure and Machine learning and Deep learning Infrastructure are components that complement the offered solution and their choice and implementation are dependent on the targeted use cases and scenarios.

INTRA's StreamHandler platform can interoperate with IoT architectures building an end-to-end IoT integration with the platform with the use of MQTT protocol. MQTT is a widely used ISO standard (ISO/IEC PRF 20922) publish-subscribe-based messaging protocol. MQTT has many implementations such as Mosquitto or HiveMQ. MQTT and Apache Kafka are a perfect combination for **end-to-end IoT integration from edge to data center** (and back, of course, i.e., bi-directional).

Inbound and outbound interfaces and relation with other MARVEL components.

Utilising the Kafka interface, StreamHandler can communicate by sending and receiving Kafka topics. Data sources and Data stores, which represent data streams and data sources, both in a structured or unstructured format that can be made available and potentially be connected to the Big data platform, generated by any IoT device and/or gateway on the edge. Similarly, and according to the requirements, appropriate persistent storage can be used, as depicted in the input/output data components (Figure 23). The described data sources will be seamlessly integrated with processing components by the means of integration connectors (Connectors). The Big data platform can efficiently interoperate with all the modern data storage technologies of a Big data ecosystem such as RDBMS, NoSQL, HDFS Hadoop, Apache HBASE, etc. as well as other persistence approaches such as Mongo, MySQL, JDBC, etc.

Processing, handling, fusion, and synchronisation of AV data

StreamHandler does not offer any edge processing options. It is designed and optimised for handling non-binary data that is streamed in real-time to the Kafka interfaces. For the MARVEL use cases, StreamHandler will not directly process or handle AV data; instead, any results of AV processing from ML components can be made immediately available for real-time analytics or stored for later filtering, processing, searching, and visualisation.

Compatibilities, complementarities, etc. with other data management platforms

Significant compatibility between StreamHandler and ITML's DFB has been already established within the MARVEL context. Since both frameworks address the issue of scalable collection, fusion, and management of large volumes of non-binary data, a hybrid solution for MARVEL is followed. Data collection and aggregation will be undertaken by DFB, reinforced by a set of StreamHandler's security and performance features that has been gathered over this component's application in large-scale domains.

5.3 DatAna

DatAna is a framework based on the usage of the Apache NiFi ecosystem to allow the processing of data flows between the edge/fog and the cloud. NiFi provides functionality for data ingestion, transformation, processing, and movement of heterogeneous data sources following a data-flow paradigm. At the edge/fog layer MiNiFi agents can be installed (i.e., at a Raspberry Pi) to ingest and process data, and then move the resulting data flows to the cloud where a single NiFi or a cluster of NiFi resides. This offers the possibility of creating MiNiFi to NiFi topologies.

NiFi's base unit of work is called a FlowFile, which provides an encapsulation of the data to be processed with some extra attributes that define that data (like filename, last update time, etc.). The FlowFiles are updated and routed using what in Nifi is called "Processors". NiFi moves the data in the FlowFile from processor to processor through the connections that link them. It can also be used to route the FlowFiles through different paths depending on the specific routing attributes that can be defined in the data flow. NiFi provides many processors[14] for data ingestion, processing, and output to many other systems.

Developers can use the NiFi graphical user interface to easily draw data flows (data pipelines), select adequate processors to ingest, process, and output data to many other systems using the available processors, as well as developing new custom processors if required.

NiFi is flexible both in the inputs and outputs. NiFi does not impose any specific data format. Different data flows can be set in NiFi/MiNiFi getting data from different sources (i.e., CSV, or even some specific video or audio files to be moved to the cloud). NiFi and MiNiFi can run independently or different remote deployments can be connected using the NiFi Site-to-Site (S2S) protocol[15], thus allowing the creation of a topology of MiNiFi/NiFi instances. For instance, several MiNiFi agents installed at the edge/fog (i.e., in several Raspberry Pi) can communicate with a remote NiFi instance or a NiFi cluster, passing the results of the data flows from several IoT gateways to the remote NiFi (i.e., in the cloud).

Inbound interfaces

NiFi and MiNiFi provide a plethora of off-the-shelf data ingestion processors for different kinds of data sources. Some of the available processors allow ingesting data from files, Kafka, MQTT, JMS, HTTP, FTP, UDP, HDFS, S3, or databases such as MongoDB, Twitter, RSS, etc. There is no native processor available to ingest natively AV data, although the processors to get files might serve that purpose (but the AV processing in NiFi is probably not the best solution).

Outbound interfaces

The output of NiFi or MiNiFi data flows might be directed to other systems using dedicated processors for data egress. The list is similar to the ingestion processors. Of especial interest are the processors that connect to Kafka.

Processing

NiFi does not provide any off-the-shelf processors to handle and process AV data, except for a processor to extract metadata from media files (including audio and video)[16] that relies on Apache Tika[17] to extract metadata.

In fact, the way NiFi handles data flows is not very suitable to process video or audio binary data. There are some attempts to do so, for instance ingesting a video file as a single data flow and then dividing it into images per frame and creating a single data flow for each of the images for further processing. This approach is feasible but expensive if applied extensively

[14] https://nifi.apache.org/docs.html

[15] https://nifi.apache.org/docs/nifi-docs/html/administration-guide.html#site_to_site_properties

[16]https://nifi.apache.org/docs/nifi-docs/components/org.apache.nifi/nifi-media-nar/1.5.0/org.apache.nifi.processors.media.ExtractMediaMetadata/

[17] http://tika.apache.org/

for many cameras, and it is probably not in line with MARVEL's current view on how to process and analyse video.

On the other hand, NiFi/MiNiFi allows the possibility of running ML models, over a data flow of an entire AV file, as services, Python scripts, or custom processors written in Java. But again, this is not a native usage of ML over video and the performance might not be the best compared to other external tools (i.e., ML/DL algorithms designed to run on the NVIDIA hardware directly using the NVIDIA SDK), which might prove to be more performant solutions.

NiFi is, on the other hand, very useful to perform data ingestion and transformation of many other types of data. NiFi offers processors to perform data transformation tasks over the data flows, including for instance splitting, aggregation, and enrichment. For instance, CSV files resulting from the pre-processing and analysis of AV files (i.e., as a result of CATFlow) can be ingested and further processed and integrated both at the fog or cloud layers. The possibility of using Kafka processors either as a producer (providing data to NiFi) or consumer (receiving data from NiFi) is very common to abstract the interfaces with other data platforms. Data from other city sensors, datasets, or services can be also processed using MiNiFi/NiFi, which opens new possibilities for further applications of MARVEL, opening the framework to unforeseen use cases involving other types of data.

Complementarities with other data management platforms

As stated above, NiFi provides processors to consume and produce data for other systems, and in especial to Kafka. Therefore, there is a clear way to interface with other systems based in Kafka, such as ITML's Data Fusion Bus or INTRA's StreamHandler. These systems might be interfaced using the off-the-shelf Kafka processors without ruling out the possibility of developing dedicated custom processors to ensure seamless integration with the other Data Platforms, if needed.

For instance, an easy use case is NiFi as a Kafka producer. A MiNiFi agent at the fog layer can provide data to a "central" NiFi and then deliver the data to Kafka. This is done in the GUI of NiFi directly by using the above-mentioned processors in the user interface without the need of writing code. The opposite case is also very common, where Kafka is used as a data producer for NiFi. Therefore, Kafka might be used to abstract the interfaces for NiFi, among other possibilities.

Figure 24 presents an overview of the main elements of DatAna and an attempt of drafting the relations with other data management platforms. This figure is an initial capture, as the final design of the relation of the different data management platforms is under discussion and their interactions can be tailored in different ways for the needs of the different scenarios.

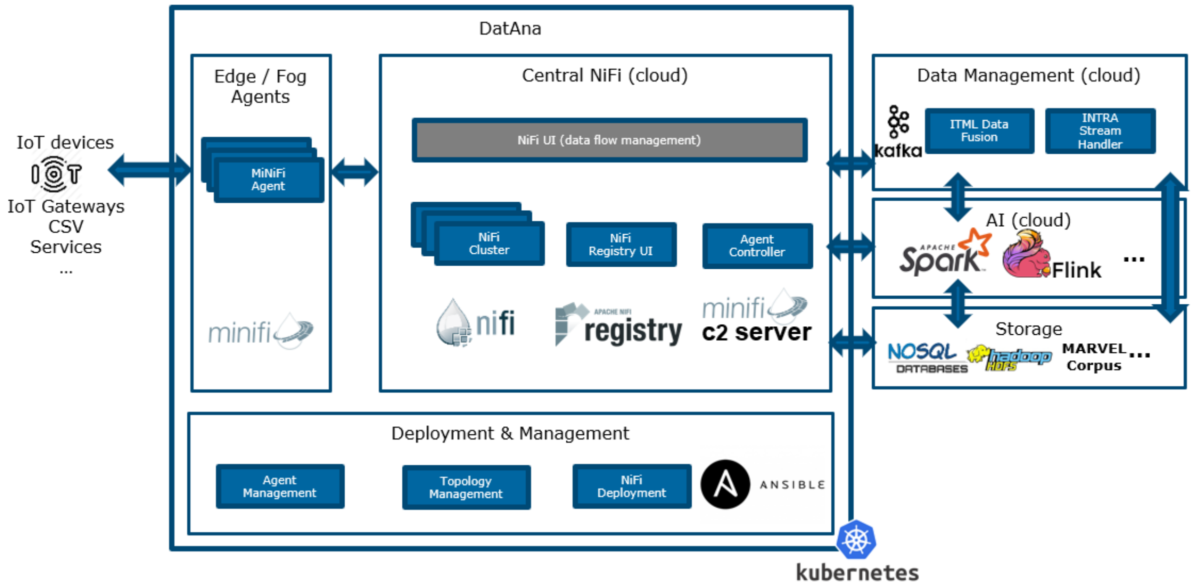


Figure 24. DatAna conceptual building blocks and relations

As a sample data pipeline, DatAna is able to ingest and process data at the edge/fog layers, passing the resulting data flows to the central or cloud layer and then connecting with other data management platforms for further processing or analysis, or alternatively storing data in different repositories. This scenario is just one possibility, as other data management platforms in the cloud might, for instance, be connected directly from the fog layer if no further data processing is needed at the edge/fog.

The AV data is not represented in the figure, as it is not the main data type managed in NiFi (binary). The deployment and management layer at the bottom of the figure represents building blocks to be developed in MARVEL to facilitate handling multiple MiNiFi agents. Kubernetes is shown as one possibility for deployment of DatAna, due to the native features for deployment, configuration, and distribution of software provided by Kubernetes that can help in the development of the agent management building block. In MARVEL, this functionality is enabled through the Karvdash component of the MARVEL architecture, which provides managed access to a container execution environment (spanning from the edge to the fog and cloud layers).

5.4 Hierarchical Data Distribution (HDD)

HDD is a set of distributed adaptive data delivery and access algorithmic schemes for guaranteeing real-time delay requirements while effectively prolonging network lifetime in wireless edge networks. The current design of HDD considers best effort, low-latency (ms) applications (that might require a certain maximum delivery delay from data producers to data consumers), running on low-power resource-constrained devices. A central controller computes a near-optimal set of multi-hop paths from producers to consumers, which guarantee a maximum delivery delay while maximising the energy lifetime of the networks. HDD uses both centralised methods which employ local area wireless communication to renew the data distribution schedules according to the current network energy map and distributed ones which periodically rotate the available data distribution paths in proportionally fair manners, as well as hybrid ones, which carefully assign a proxy role to some of the network nodes, in order to further boost the network hierarchy. HDD targets large-scale network deployments which are characterised by an inherent computational intractability when it comes to the algorithmic management of the available data. A high-level view of HDD is depicted in Figure 25 in blue.

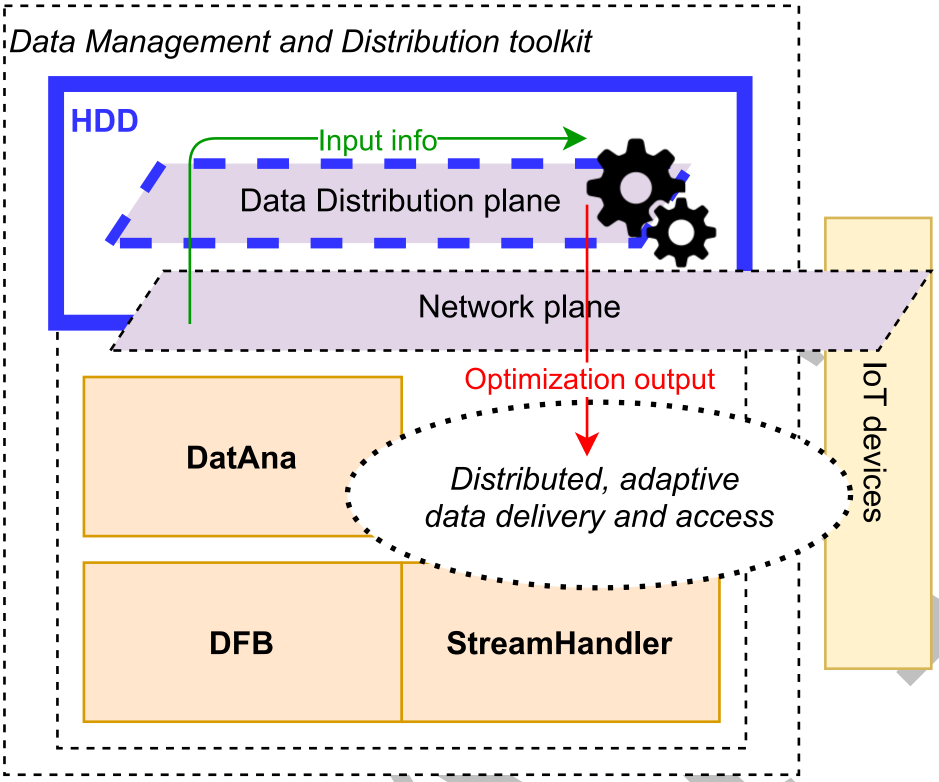


Figure 25. High-level view of the role of HDD within MARVEL's Data management and distribution subsystem

Inbound and outbound interfaces

As seen in the figure, HDD interconnects with the rest of the Data Management and Distribution subsystem components through the common infrastructure of a network plane. Being a Proof of Concept (POC) and not a plug&play component for every application scenario, it needs as input the following information about requirements/constraints/targets (green arrow): Definition of the data problem to be addressed, available device HW/FW/SW information on communication and computation abilities, desired networking protocol information, details on the edge-fog-cloud setting under consideration, as well as types of data in the network (VVV). After the algorithmic processes take place in the data distribution plane, the optimisation output is returned (red arrow) to the network plane for rendering the data delivery and access process more distributed, adaptive, and efficient.

Edge processing

Depending on the underlying network and devices, HDD supports either distributed or centralised processing. The component's role is to decide the extent of distributiveness depending on the application requirements and constraints. In the distributed case, the edge nodes locally share some burden of the computations with limited knowledge of the available global data. Given an application, in order to formalise the computation version of the processing problem, we need to define the target objective function, which, indicatively can refer to latency, data loss rate, energy consumption, network lifetime, and others. In order to construct the objective function, we build on top of existing typical edge processing (networking) formulations. Processing of AV data is not currently supported, and it will be worked out collaboratively with the other components (DatAna, DFB, StreamHandler) in the frame of the Data Management and Distribution Subsystem.

Complementarities with other data management components

Currently, the implementation of sending/receiving data with other components of the Data Management and Distribution Subsystem is under discussion. A straightforward case would be to produce and feed data directly to Kafka. Then, appropriate elastic indices could be defined for various topics/categories of data. The offered UIs (e.g., from DFB/StreamHandler) already provide many functions for visualising and monitoring collected data. On top of that, the DFB/StreamHandler core modules expose APIs that offer more elaborate services over data. The kind of actions that need to be applied to these data, so as to verify if this is covered by the UI, or if we need to develop new services for that matter is under discussion.

6 Audio, visual, and multimodal AI subsystem

The audio, visual, and multimodal AI subsystem contains data analysis components that implement the AI functionalities of the MARVEL framework. Depending on the targeted task, single-modality data (audio or visual) is analysed, or multi-modal (audio-visual) data is jointly processed to exploit the complementary information appearing in both audio and visual streams for better describing the contents of the scene. In the context of a smart city environment, such an approach is very important, as several aspects of our daily life, especially in the case of anomalous events, cannot be completely described unless both audio and visual information is provided. As an example, when a park is monitored by a surveillance camera and an explosion occurs outside the camera field of view, visual data analysis will not be able to capture the event as the scene can appear to be normal, e.g., people running for their everyday exercise routine may be a normal activity observed in that park. This is a good reason for implementing a variety of complementary data analysis functionalities. In our previous example, a component implementing sound event detection functionality can detect the event, while a component implementing audio-visual anomaly detection functionality can detect the anomalous event that can be then associated with the label provided by the sound event detection module.

The components of the subsystem analysing single-modality data provide functionalities for audio feature extraction, automated audio captioning, sound event detection, sound event localisation and detection, acoustic scene classification, visual anomaly detection, and visual crowd counting. Components that jointly analyse audio-visual data for increasing performance in the case where an event is better described exploiting both visual and audio information, and increasing robustness in the case of low visual data quality provide functionalities for audio-visual anomaly detection and audio-visual crowd counting. Within the MARVEL framework, the functionalities included in this subsystem provide data analytics addressing the corresponding MARVEL requirements listed in Table 2. A detailed description of the components is provided in the following subsections. A schematic illustration of the subsystem is given in Figure 26.

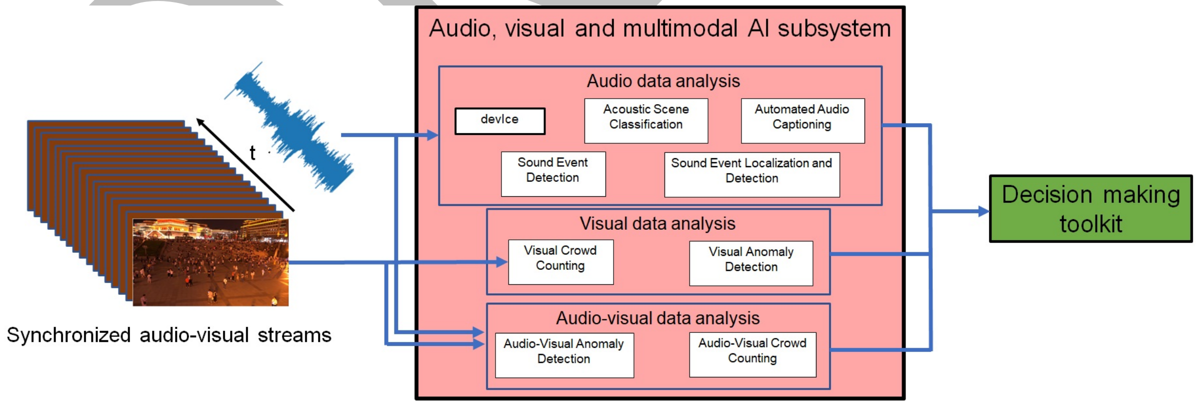

Audio, visual and multimodal AI subsystem
Audio data analysis
devIce
Acoustic Scene Classification
Automated Audio Captioning
Sound Event Detection
Sound Event Localization and Detection
Visual data analysis
Visual Crowd Counting
Visual Anomaly Detection
Audio-visual data analysis
Audio-Visual Anomaly Detection
Audio-Visual Crowd Counting
Decision making toolkit
t
Synchronized audio-visual streams

Figure 26. Audio, visual, and multimodal AI subsystem

6.1 devAIce

devAIce is a C++ SDK that encapsulates the majority of AUD's intelligent audio analytics technology. Inside MARVEL, it will primarily be used for three tasks:

1. Voice activity detection for purposes of (voice) anonymisation (covered in Section 4.4 of the current report),

2. Feature extraction to support MARVEL's other audio(-visual) analytic modules,
3. Audio analytics to support MARVEL's decision-making toolkit.

As devAIce contains several modules, its runtime depends on the particular module. In general, all modules should easily run on cloud and fog devices, and most should run on high-end edge nodes (Raspberry Pis).

All devAIce modules will accept as input an (anonymised) audio stream. The feature extraction module (which includes AUD's award-winning openSMILE toolkit) will be used to extract relevant audio features which will be propagated to upstream audio(-visual) processing modules (including devAIce's own audio processing modules). The output features come in the form of 32-bit float arrays which will be consumed by the other components.

In addition, devAIce includes intelligent audio analytics modules itself. Those relevant to MARVEL's goals are acoustic scene classification (ASC) and sound event detection (SED). Those modules, which operate either on raw (anonymised) audio streams or feature streams, output either a single decision on the segment level (for the case of ASC), or frame-level decisions (for the case of SED). Those decisions will be propagated to the upper levels of MARVEL's decision-making toolkit. An illustration of devAIce inside MARVEL's architecture is provided in Figure 27.

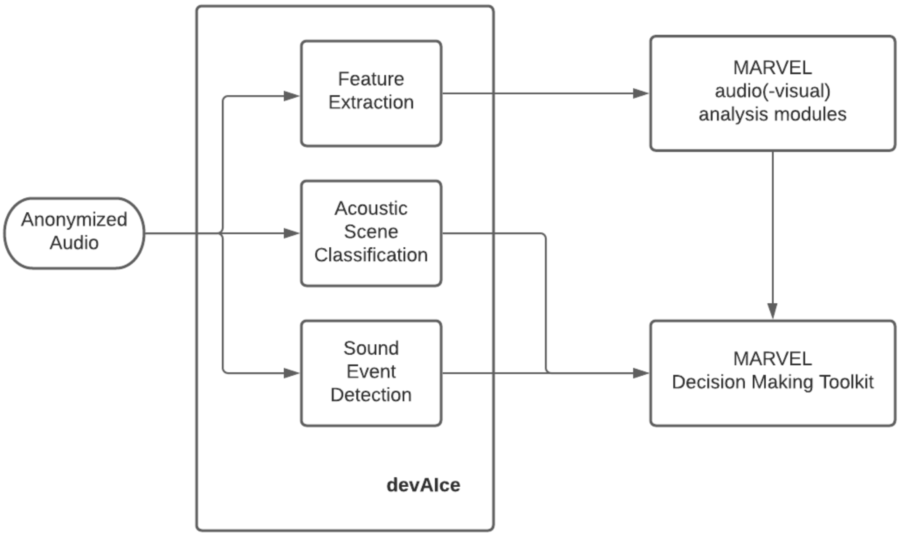

Anonymized Audio
Feature Extraction
Acoustic Scene Classification
Sound Event Detection
devAIce
MARVEL audio(-visual) analysis modules
MARVEL Decision Making Toolkit

Figure 27. devAIce inside MARVEL architecture

6.2 Visual anomaly detection

The goal of visual anomaly detection is to establish a representation of "normal situations" based on the available training data, and detect whenever an event occurs that sufficiently deviates from normal situations. For instance, for a camera overlooking a pedestrian walkway, any non-pedestrian objects would be considered anomalous, and for a camera overlooking a street, any illegal activity such as jaywalking is anomalous. It is important to note that the anomalies flagged in the dataset are excluded during the training process and only used to evaluate the performance of trained models.

In the training phase, the input to the component is a set of training examples, each containing an image/video frame, a flag that indicates whether anomalies are present in the given image/video frame, and optionally spatial annotations such as bounding boxes or pixel masks specifying the location of anomalies within the given image/video frame, and the output would be a trained model capable of detecting anomalies. During the inference phase when the model is deployed, the input to the component would be a series of video frames, and the output would be a flag indicating whether anomalies are present in each frame. If spatial annotations were provided in the training phase, the output also specifies the location of anomalies within the video frame.

The component can be deployed on edge, fog, and cloud. Raw video frames are processed on edge and fog, and anonymised video frames are processed on the cloud. An illustration of visual anomaly detection is in Figure 28.

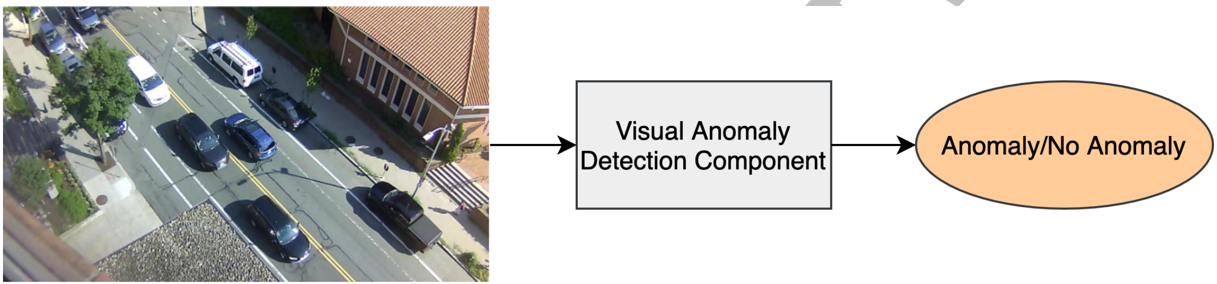


Figure 28. Illustration of visual anomaly detection

6.3 Audio-Visual anomaly detection

Similar to video anomaly detection, the aim of audio-visual anomaly detection is to establish a representation of "normal situations" based on both audio and video available in the training data, and detect whenever an event occurs that sufficiently deviates from normal situations. For instance, in a scene recorded in a train station, the anomalous video clip could depict people running away from something and the corresponding audio could contain gunshot sounds.

In the training phase, the input to the component is a set of training examples, each containing an image/video frame, the ambient audio waveform corresponding to the image/video frame, a flag that indicates whether anomalies are present in the given image/video frame and optionally spatial annotations such as bounding boxes or pixel masks specifying the location of anomalies within the given image/video frame, and the output would be a trained model capable of detecting anomalies. During the inference phase when the model is deployed, the input to the component would be a series of video frames and the ambient audio waveform corresponding to the image/video frame and the output would be a flag indicating whether anomalies are present in each frame. If spatial annotations were provided in the training phase, the output also specifies the location of anomalies within the video frame.

The component can be deployed on edge, fog, and cloud. Raw video frames are processed on edge and fog, and anonymised video frames are processed on the cloud. An illustration of audio-visual anomaly detection is in Figure 29.

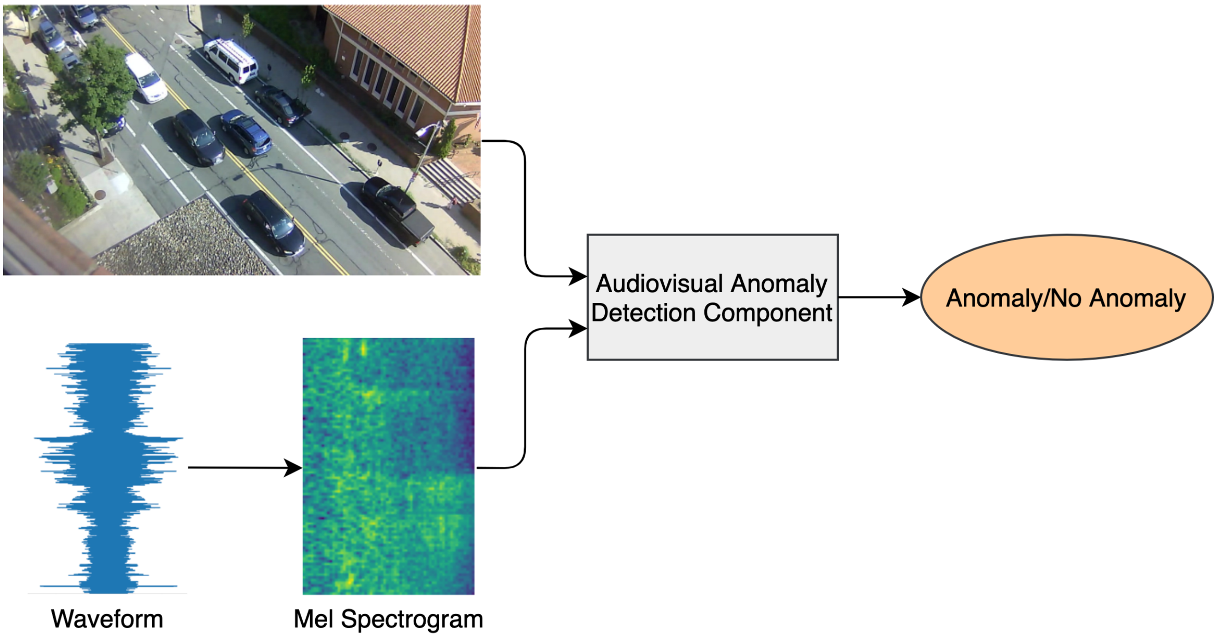


Figure 29. Illustration of audio-visual anomaly detection

6.4 Visual crowd counting

Crowd counting is the task of counting the total number of people present in an image. The images could contain very few people, tens of thousands of people (for instance, in stadiums), or even no people at all (background-only images) and may be taken from various perspectives and different times of day or at night. In most methods available in the literature, the output of the model is not just a single number representing the total count, but a density map that shows the number of people at different locations in the image, where the total count would be the sum of all locations in the density map.

In the training phase, the input to the component is an image and the total count of people present in that image. Optionally, head annotations specifying the location of the heads of people in the image can be provided as well. The output is a trained model capable of counting people in given images. During the inference phase, the input to the component is a series of images/video frames and the output is the total count of people in each image/video frame. If head annotations were provided in the training phase, the output can also include a density map specifying the number of people at different locations in the image/video frame.

The component can be deployed on edge, fog, and cloud. Raw video frames are processed on edge and fog, and anonymised video frames are processed on the cloud. An illustration of visual crowd counting is in Figure 30.

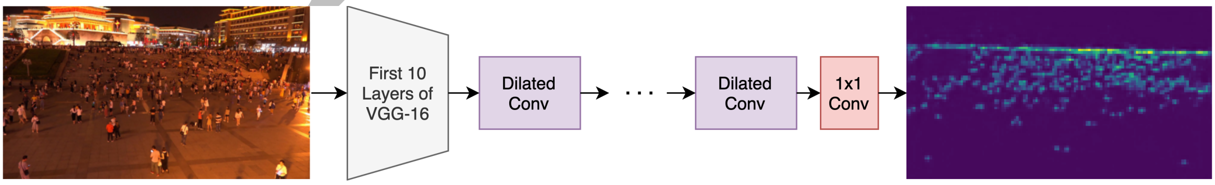


Figure 30. Illustration of visual crowd counting

6.5 Audio-Visual crowd counting

Similar to visual crowd counting, the goal of audio-visual crowd counting is to count the total number of people present in an image, however, for each image, an audio clip corresponding to the ambient noise in the scene where the image was taken is available as well. The ambient audio has been shown to improve the accuracy of crowd counting in situations where the quality of the image is low, for instance, low resolution, low illumination, severe occlusion, or presence of equipment noise. Similarly, the output of audio-visual crowd counting models is density maps representing the number of people at different locations of the input image.

In the training phase, the input to the component is an image, the ambient audio waveform corresponding to the image/video frame, and the total count of people present in that image. Optionally, head annotations specifying the location of the heads of people in the image can be provided as well. The output is a trained model capable of counting people in given images. During the inference phase, the input to the component is a series of images/video frames and the ambient audio waveform corresponding to the image/video frame, and the output is the total count of people in each image/video frame. If head annotations were provided in the training phase, the output can also include a density map specifying the number of people at different locations in the image/video frame.

The component can be deployed on edge, fog, and cloud. Raw video frames are processed on edge and fog, and anonymised video frames are processed on the cloud. An illustration of audio-visual crowd counting is in Figure 31.

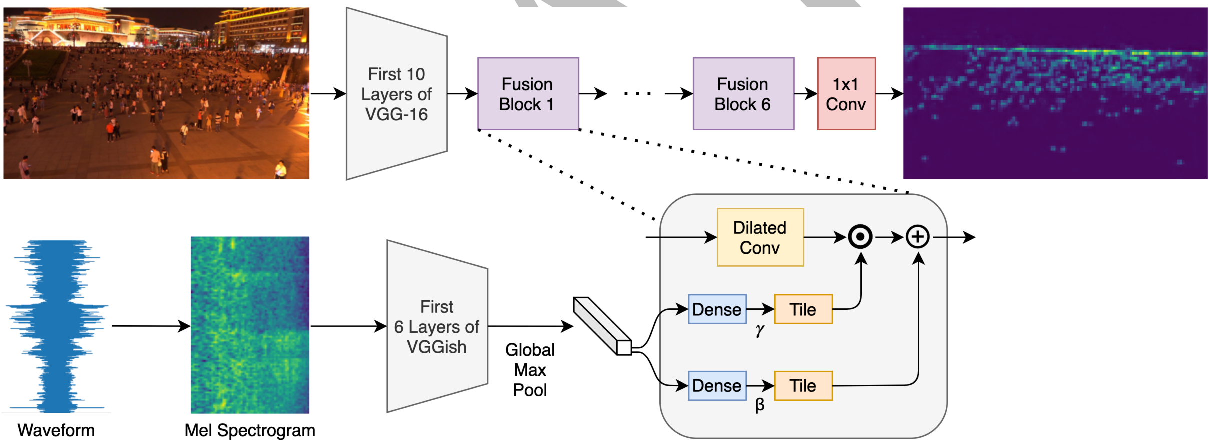


Figure 31. Illustration of audio-visual crowd counting

6.6 Automated audio captioning

Automated audio captioning (AAC) will provide textual descriptions that can be used as extra and indicative information, to assist the decision-making process by the corresponding MARVEL components. For each unit of time, e.g., 30 seconds, 1 minute, or any other fit unit of time according to the use case, the AAC system will provide a sentence that will describe what is happening in the audio, for example, "People talking while music playing in the background and cars passing by". The knowledge that the AAC system will develop is dependent on the data and their annotations that will be used during training. Apart from working individually, the AAC system can collaborate and enhance the predictive behaviour of other systems by providing high-level information given the audio signals. The AAC takes as an input an audio signal and can provide as an output either the textual description of the input audio signal or an intermediate representation that has encoded all the information needed for the creation of the textual description.

The development of the AAC system requires an audio captioning dataset, which consists of audio signals that have captions as annotations. During its training, the AAC system learns to map the input audio signals to the corresponding captions. During prediction, the AAC system takes as an input the audio signals and produces the textual descriptions that describe the contents of the input audio signals. The current state-of-the-art automated audio captioning systems require extremely large neural networks (over 100M parameters) in order to produce a good performance. These neural networks require a large amount of computational resources, thus the AAC systems are feasible to run only on the cloud. An illustration of an AAC system is in Figure 32.

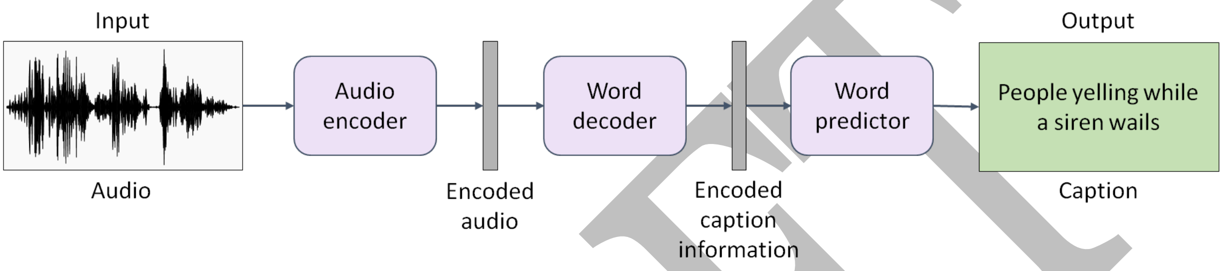


Figure 32. Illustration of an AAC system

6.7 Sound event detection

The sound event detection (SED) system can provide the detection of characteristic sounds in short time units, e.g., one second. A characteristic sound is a sound that can be described by a specific label, i.e., a sound event. For example, “car passing by”, “engine revving”, “people talking”, and “glass breaking”. Such functionality can be used in the various use cases of MARVEL, offering the ability to detect actions and events through sound. The specific sound events will be dependent on the use cases and the detection of the sound events can be used as standalone information or as complementary information to other systems. As standalone information, detection of sound events can offer the detection of activities, core functionality of the audio perception of MARVEL. As complementary information, SED can be used in audio-visual detection or even in the decision-making toolkit, providing an extra channel of information.

A SED system takes as an input an audio signal and provides detection of sound events in pre-specified units of time, e.g., one second, one minute, etc. During training, the SED systems need a strongly labelled audio dataset, where the sound events are indicated with their start and end times. During prediction, the SED system takes as an input an audio signal and predicts the activity of the sound events that have been trained on. An illustration of a SED system is in Figure 33.

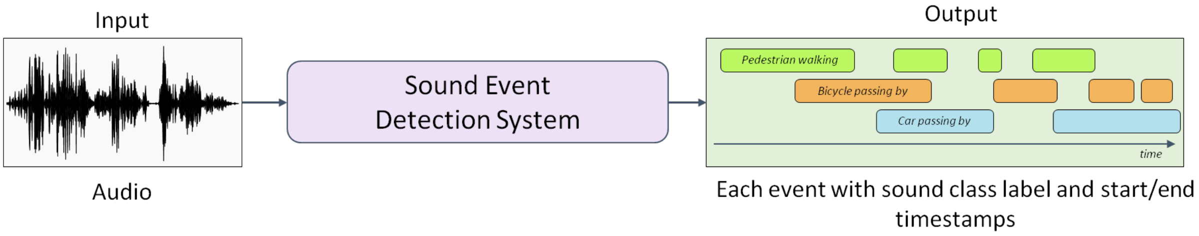

Input
Audio
Sound Event Detection System
Output
Pedestrian walking
Bicycle passing by
Car passing by
time
Each event with sound class label and start/end timestamps

Figure 33. Illustration of a SED system

6.8 Sound event localisation and detection

Sound event localisation and detection (SELD) is very similar to SED. SELD is a joint task, where a system jointly performs sound event localisation and SED. The localisation consists of the detection of directional characteristics of the sound events. These characteristics can be simple, e.g., left to right or right to left direction or more complex like the azimuth and elevation of the direction of arrival of the sound that is classified as belonging to a sound event. The detection of the directional characteristics is performed with respect to the microphone, which is considered the position of reference. A SELD system can be used in many of the use cases of MARVEL, offering jointly the detection of sound events and the indication of the direction of arrival of the sound that is recognised as a sound event.

A SELD system has the same requirements as a SED system regarding the detection of sound events. On top of that, a SELD system needs the annotation of the directional characteristics for each sound event and in the same temporal resolution as the sound events. A SELD system is very similar to a SED system, as is illustrated in Figure 33, plus the indication of the directional characteristics for each sound event.

6.9 Acoustic scene classification

Acoustic scene classification (ASC) provides the classification of the acoustic scene of the audio signal. For example, an ASC system can detect labels like “crowded square”, “busy street”, “office”, “beach”, etc. An ASC system can be used to detect complementary information to various systems in MARVEL (e.g., AV perception systems), offering the indication of, for example, the degree of crowdedness in market squares or streets. Additionally, ASC can be used as a standalone system in use cases where detecting the acoustic scene can be needed.

An ASC system needs for training an audio dataset where the acoustic scene is annotated. The time resolution can be coarser than that of the SED/SELD cases, for example, 30 seconds or even minutes. During training, an ASC system takes as an input the audio signals and learns to predict the correct acoustic scene class. During prediction, the ASC system takes as an input audio signal and outputs the predicted acoustic scene class for the input audio. An illustration of an ASC system is in Figure 34.

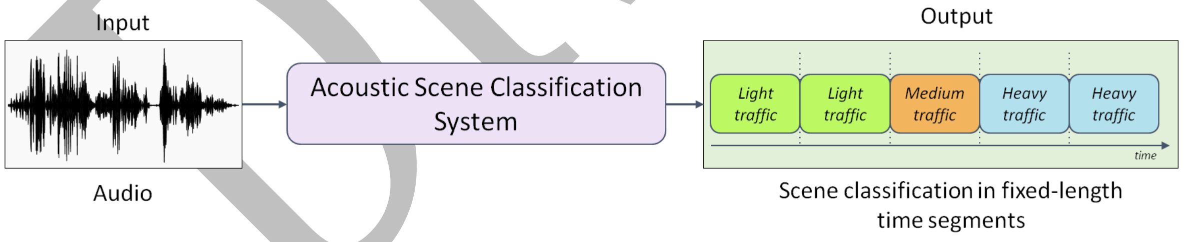

Input
Audio
Acoustic Scene Classification System
Output
Light traffic
Light traffic
Medium traffic
Heavy traffic
Heavy traffic
time
Scene classification in fixed-length time segments

Figure 34. Illustration of an ASC system

7 Optimised E2F2C processing and deployment subsystem

The E2F2C processing and deployment subsystem realises the deployment logic that will exploit the full potential of the ML/DL approach implemented in MARVEL. Its deployment optimisation and management modules will decide where the processing should be made, optimising the deployment based on resource availability at the edge and fog layers, both for training and inference. The goal is to establish efficient distributed E2F2C DL layouts that enable real-time decision-making in all layers of the architecture.

For training, the E2F2C processing and deployment subsystem will enable distributing complex AI tasks in an optimal (or nearly-optimal) way on a set of distributed edge devices, taking into account data availability, network and device resource consumption, and accuracy of the target AI task. It will also extend state-of-the-art architectures for efficient edge computing (e.g., supporting stateless programming paradigms) to efficiently support distributed AI tasks, e.g., distribution of several layers of a NN to all the layers of the framework from the edge devices to the Cloud. For inference, performed at several exit points of the E2F2C architecture, decisions will be made locally, or if this is not possible, processing tasks will be offloaded and the data will be transmitted to the next device up in the architecture.

In the E2F2C architecture, DL/ML model training at the HPC/Cloud layer will include processing of large AV data-at-rest (high-throughput computing, parallel execution using MPI, and multiple high-end GPUs). The fog layer will encompass multiple execution sites for model execution (inference) and stream (data-in-motion) processing, while the edge layer will be mainly for data collection through cameras, microphones, misc. sensors, and mobile apps, as well as processing when the hardware resources allow it.

The E2F2C processing and deployment subsystem implements the appropriate control and data plane for deployment and orchestration of tasks for both the training and inference phases of the DL/ML model lifecycle. The main components of this subsystem are as follows:

- DynHP is a methodology to train and compress at the same time a DNN model under a fixed memory budget without significant accuracy drop, removing as many parameters as possible that are not strictly useful for achieving the target accuracy. DynHP generates optimised ML/DL models that can fit resource-constrained devices.
- FedL is a personalised federated learning framework that distributes training of ML/DL models across different MARVEL clients holding their local dataset. Personalisation is achieved by tailoring algorithms to different data distributions across different FL clients.
- GPURegex is a real-time pattern matching engine that leverages the parallelism properties of GPGPUs to accelerate string and/or regular expression matching. In the context of MARVEL, the pattern matching features of the accelerated engine can be integrated into a larger processing pipeline that transforms raw AV data into text-based representations. These are digested by our pattern matching engine and compared against a set of predefined and preloaded signatures of potentially dangerous events. This can lead to more accurate and/or (near) real-time recognition of events.
- Karvdash/PLATFORMA is a managed execution platform. The main features are: (i) identity and access management (IAM, with Single Sign-On), (ii) encapsulation of tasks in containers for their orchestrated execution, (iii) data storage and access to data stores,

and (iv) support for "elastic" serverless processing, capable of interacting with data processing platforms.

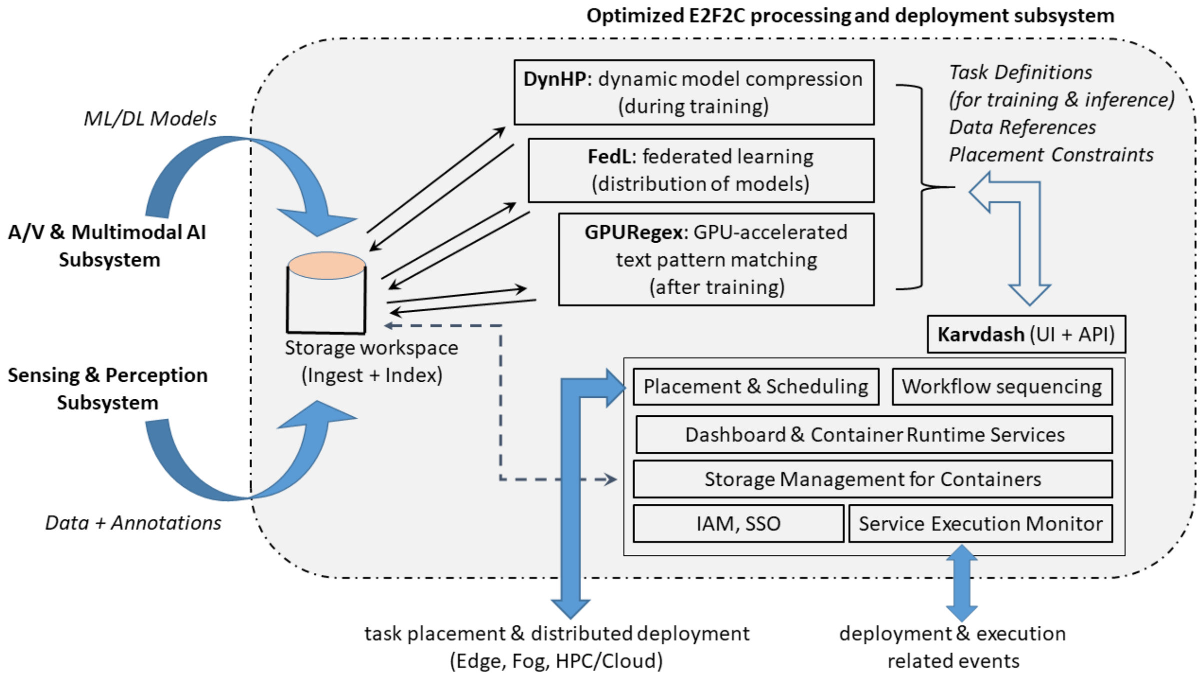

Optimized E2F2C processing and deployment subsystem
ML/DL Models
A/V & Multimodal AI Subsystem
Sensing & Perception Subsystem
Data + Annotations
Storage workspace (Ingest + Index)
DynHP: dynamic model compression (during training)
FedL: federated learning (distribution of models)
GPURegex: GPU-accelerated text pattern matching (after training)
Task Definitions (for training & inference)
Data References
Placement Constraints
Karvdash (UI + API)
Placement & Scheduling
Workflow sequencing
Dashboard & Container Runtime Services
Storage Management for Containers
IAM, SSO
Service Execution Monitor
task placement & distributed deployment (Edge, Fog, HPC/Cloud)
deployment & execution related events

Figure 35. Outline of the Optimised E2F2C Processing and Deployment Subsystem that supports deployment and execution in a distributed execution environment spanning from the Centre (HPC/Cloud) to the Edge, including the intermediate layer of Fog

7.1 GPURegex

GPURegex is a real-time high-speed pattern matching engine that leverages the parallelism properties of general-purpose GPUs (GPGPUs) to accelerate the process of string and/or regular expression matching. This component has been specifically designed and implemented in order to accelerate the most common process (i.e., pattern matching) of security applications such as network intrusion detection and/or prevention systems, load balancers, and firewalls. However, in the context of the MARVEL project, GPURegex can also be slightly modified and integrated into the processing pipeline of an event detection component. GPURegex is offered as a C API and allows developers to build applications that require pattern matching capabilities while simplifying the offloading and acceleration of the workload. Note that the ability to efficiently perform text- and regular expression-based comparisons is provided. In order to be able to operate on audio and/or visual data and extract meaningful results, some data transformations during the preprocessing phase are required. These transformations should aim to extract text-based representations of the corresponding raw AV data and feed our engine with the extracted signatures. Figure 36 presents a high-level overview of the GPURegex architecture.

In this paragraph, we list some prerequisites in order to be able to deploy GPURegex. First of all, the presence of a GPU that supports the OpenCL standard is required. It could be either a dedicated card or an integrated chip within the CPU die. However, the parallelism properties of any OpenCL enabled device (even a CPU) could be leveraged, in case of GPU absence. The performance benefits in the case where only an OpenCL processor is available will be diminished but still notable when compared to baseline solutions executed on ARM/x86 processors without exploiting the parallelism features if they exist. OpenCL version 1.2 or newer is also required, as well as a new version of CUDA software development kit if the accelerator is provided by NVIDIA, which is expected to be the case regarding the cloud layer infrastructure.

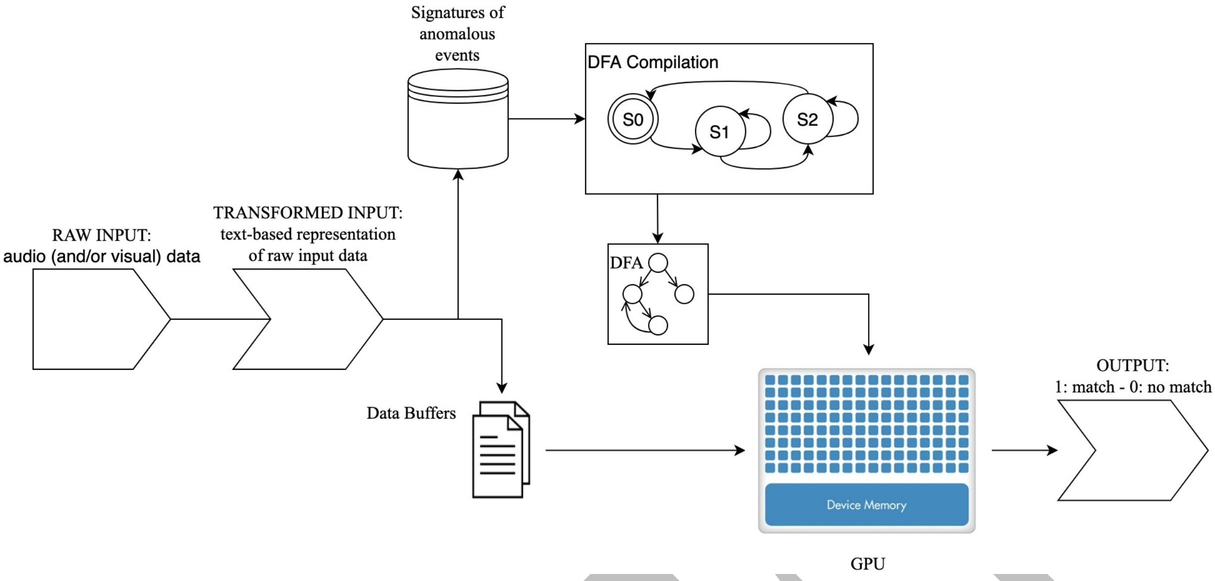


Figure 36. High-level overview of GPURegex architecture

7.2 DynHP

DynHP is a **methodology** for training a deep neural network model and compressing them at the same time. The type of compression operated by DynHP is *pruning*, i.e., the parameters of a DNN are zero-ed at training time. Pruning can be structured or unstructured; in structured pruning, the idea is to "remove" the neurons (groups of parameters) of a DNN, while in unstructured pruning the parameters are removed in a scattered way. As regards compression, structured pruning is more effective since, mathematically, it allows removing entire rows of the weights matrices that constitute the parameters of a DNN. Moreover, pruning can be either "soft" or "hard". In soft pruning, the parameters are turned on and off during the whole training process and only at the end they can be finally "removed". Conversely, in "hard pruning", the parameters that are "switched off" cannot be recovered during the rest of the training. DynHP combines structured pruning with hard pruning. Since the structured hard pruning process might degrade the performance of the training, DynHP adopts a strategy to contrast it. Precisely, it tunes adaptively the size of the minibatches depending on gradient-related information and the amount of available memory. Figure 37 exemplifies the train and compression process with DynHP.

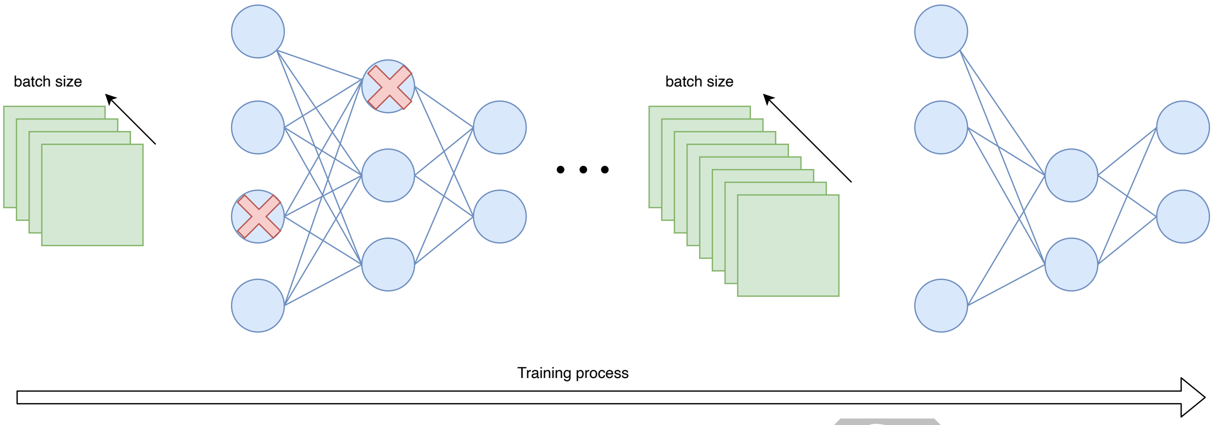


Figure 37. DynHP Compression mechanism during training: structured pruning with incrementally adaptive batch size

DynHP is designed to work on a fixed memory budget defined at the beginning of the training as the sum of the initial size of the DNN and the initial size of the minibatch. During the training, the DNN reduces its size because of the structured hard pruning, and the freed memory is used as an available budget to grow the batch size as needed by the learning process. At the end of the training, DynHP outputs a compressed model, as illustrated in Figure 38.

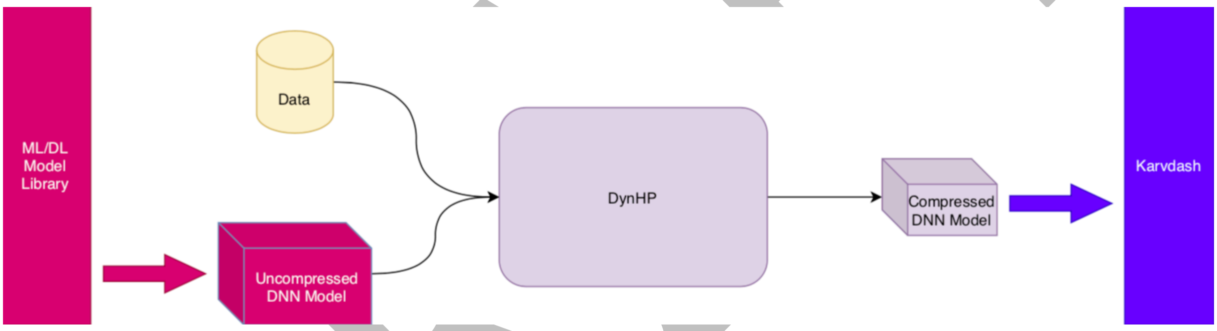


Figure 38. DynHP workflow

Currently, the component has been tested on image recognition tasks, and it must be extended to include models suitable for processing audio and video signals.

While the DynHP methodology has been devised specifically for the edge and fog layer, which typically include devices with limited computing capabilities, which would otherwise be unable to execute the inference or training phases with the original DNN model, it can be used also in the cloud for different purposes, e.g., to run a higher number of concurrent instances on a fixed amount of HW resources or to reduce the energy consumption due to processing. In relation to the MARVEL architecture, the DynHP takes as input the ML/DL models from the model library, and it passes the compressed model to Karvdash for deployment. The input and output format details are still to be defined as part of the WP3 activities.

7.3 FedL

This component implements MARVEL Personalised federated learning framework that distributes training of ML/DL models across different MARVEL clients holding their local dataset, with both datasets and clients located at the edge or fog layer of the MARVEL architecture. The clients are orchestrated by the central server, located at the cloud layer of the

MARVEL architecture. An instance of the component is initialised by an ML/DL trained model from the MARVEL model library, and the models are then distributed to the clients. Clients revise the model using their data and send model updates or model gradients to the central server which fuses them and returns the updated global model to all the clients. The process continues iteratively until a stopping criterion is met (e.g., the desired accuracy is reached). To protect the privacy of local datasets, the framework will explore the use of differential privacy by adding noise to the gradients during local model updates.

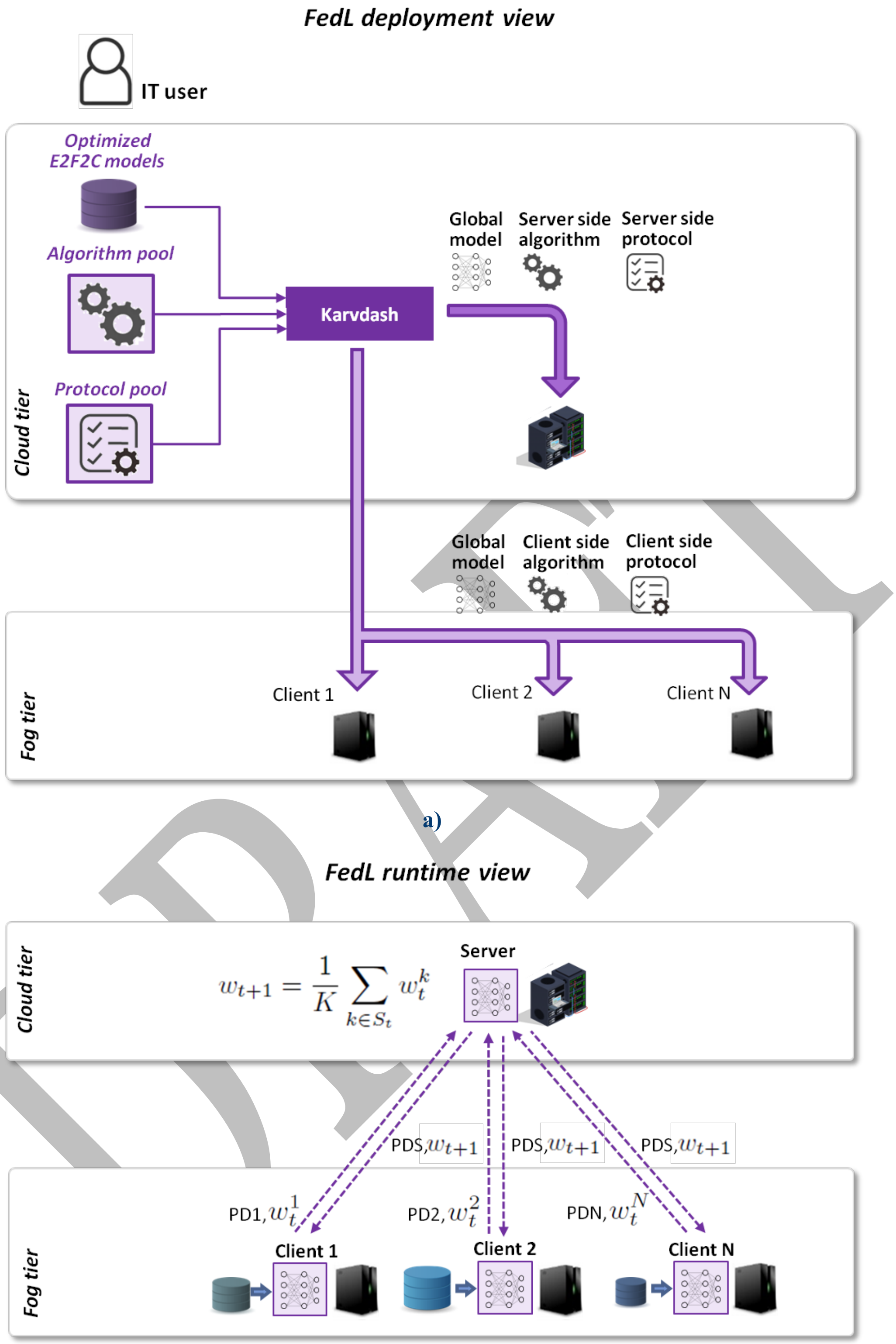
FedL will have access to the model database of the Audio, visual and multimodal AI subsystem and also to the compressed models produced by the DynHP component. The exact models to be used by FedL are use case-specific and are to be defined/selected by IT users of the MARVEL platform together with the appropriate FL protocol and other parameters (e.g., learning/protocol parameters).

Deployment of FL clients and FL servers before and during runtime will be achieved by Karvdash. The FL clients and the FL server will run and communicate in containerised environments, enabled by Karvdash.

Data necessary for model updates will be enabled by the Data management and distribution subsystem (exact components to be used – DFB, DatAna, or StreamHandler will be custom-defined for each specific use case that will deploy MARVEL FedL module); optimal data distribution will be designed by HDD. Real-time inference results will be sent to the MARVEL decision-making toolkit for raising real-time alerts and to SmartViz for the accompanying use case tailored visualisations.

FedL should run simultaneously at the edge, fog, and cloud tiers. The possibility for partial edge model deployment will also be explored (e.g., using split learning). Requirements for FedL are inherited from the adopted ML/DL model, thus dictating the deployment tier (nominally, FL clients will run at the fog tier; in case of lightweight ML models and/or more powerful edge hardware, FL clients can also run at the edge); possibility for partial edge model deployment will also be explored (e.g., using split learning (SL)). A common, minimal SL setup consists of several network layers at the client-side, whereas the rest is located at the server or fog. This way, SL can hide information from the server about full network architecture.

Figure 39 illustrates graphically the deployment (top) and runtime view (bottom) of the FedL component.



a)

*PDi /S= Protocol Data at client *i*/Server
** S_t – subset of participating clients at round *t*, $|S_t|=K$

b)

Figure 39. FedL component: a) deployment view; b) runtime view

7.4 Karvdash

Karvdash is a service for facilitating interaction with Kubernetes-based environments, by supplying the landing page for users working on the platform, allowing them to launch services, design workflows, request resources, and specify other parameters related to execution through a user-friendly interface. Karvdash aims to make it straightforward for domain experts to interact with resources in the underlying infrastructure without having to understand lower-level tools and interfaces.

Karvdash is deployed upon a Kubernetes installation as a service and provides a web-based, graphical interface to:

- Manage services or applications that are launched from customisable templates;
- Securely provision multiple services under one externally-accessible HTTPS endpoint;
- Organise container images stored in a private Docker registry;
- Perform high-level user management and isolate respective services in per-user namespaces;
- Interact with datasets that are automatically attached to service and application containers when launched.

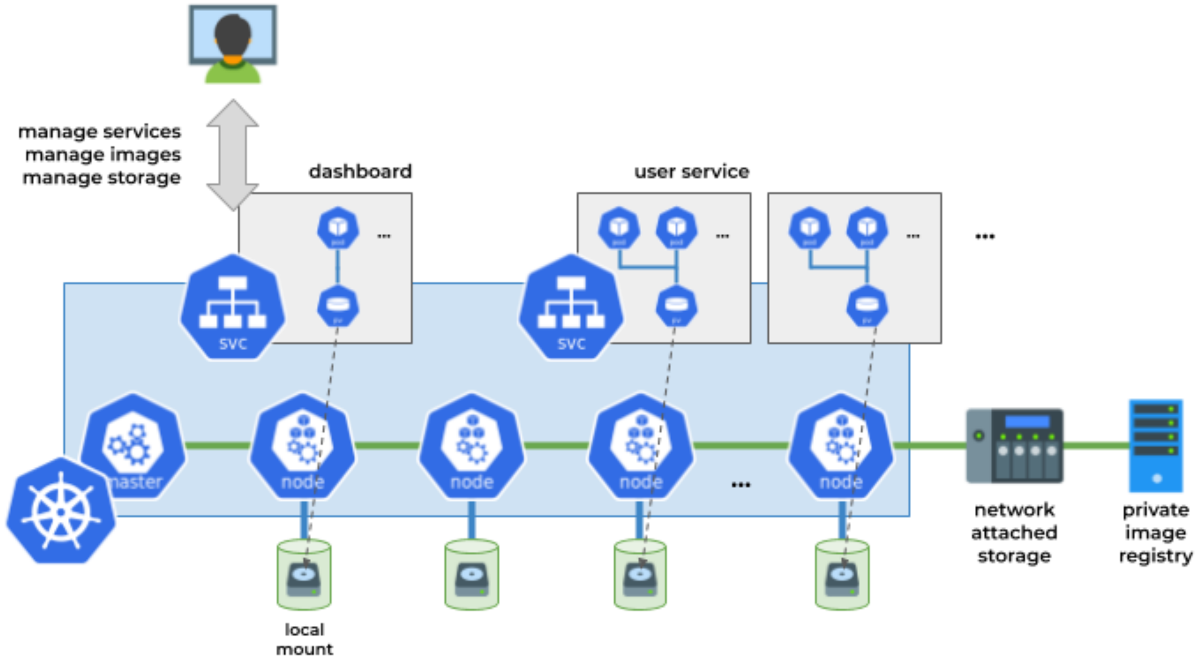


Figure 40. Karvdash provides a dashboard service to facilitate user interaction with the distributed environment realised by Kubernetes

Karvdash does not do any processing but provides the mechanisms to efficiently deploy the AI models and related support services included in the MARVEL data management and distribution toolkit in all computing continuum layers that support container-based execution. To configure and start services, Karvdash uses a service templating mechanism - each service is defined with a series of variables. The user can specify values to the variables as execution parameters through the dashboard before deployment, and Karvdash will set other "internal" platform configuration values, such as private Docker registry location, external DNS name, and others.

Internally, at the Kubernetes level, each Karvdash user is matched to a unique namespace, which also hosts all of the user’s services. Containers launched within the namespace are given Kubernetes service accounts which are only allowed to operate within their own namespace. This practice organises resources per user and isolates users from each other.
Internally, at the Kubernetes level, each Karvdash user is matched to a unique namespace, which also hosts all of the user’s services. Containers launched within the namespace are given Kubernetes service accounts which are only allowed to operate within their own namespace. This practice organises resources per user and isolates users from each other.

8 E2F2C infrastructure

The E2F2C infrastructure represents the underlying infrastructure of the framework with the three tiers edge, fog, and cloud. Each of the three levels depends on the actual infrastructure over which the framework is executed.

MARVEL considers High Performance Computing (HPC) as one element of a complex workflow (“HPC in the loop" paradigm) starting with data generated at smart sensors in an IoT environment. Data is being locally pre-processed at the edge and relevant parts are forwarded to decentralised “fog nodes” close to the edge. A subset of data is then transferred for centralised Data Analytics in clouds or for simulation and modelling in centralised HPC centres.

In the MARVEL project, the E2F2C infrastructure consists of the HPC cluster (PSNC), HPC resource management and optimisation (PSNC), together with the three underlying infrastructural tiers – cloud, fog, and edge (GRN, MT, and UNS).

The main goal of the edge tier is collecting the data via the Sensing and perception subsystem. The typical devices forming the edge layer are microcontrollers, Raspberry Pis, Arduino platforms but can also include edge GPUs, Intel NUC, NVIDIA Jetson Nano, and other computing platforms the design of which is targeted for edge operation.

The fog tier is responsible for data analytics that does not require extensive computing power. This layer covers a wide range of devices, from small computing platforms like Raspberry Pi, through GPU, all the way to local data centers.

Tasks that require the most computing power are performed in the cloud tier. The use of HPC infrastructure allows for efficient DL/ML models training.

The overview of E2F2C infrastructure is presented in Figure 1. This section provides a set of detailed figures presenting available HPC infrastructure and use cases’ infrastructure across different tiers. In the project, the generic E2F2C infrastructure is instantiated with three specific E2F2C infrastructure examples induced by the three MARVEL pilots: GRN, MT, and UNS (experiment), described and illustrated in Sections 8.3-8.5.

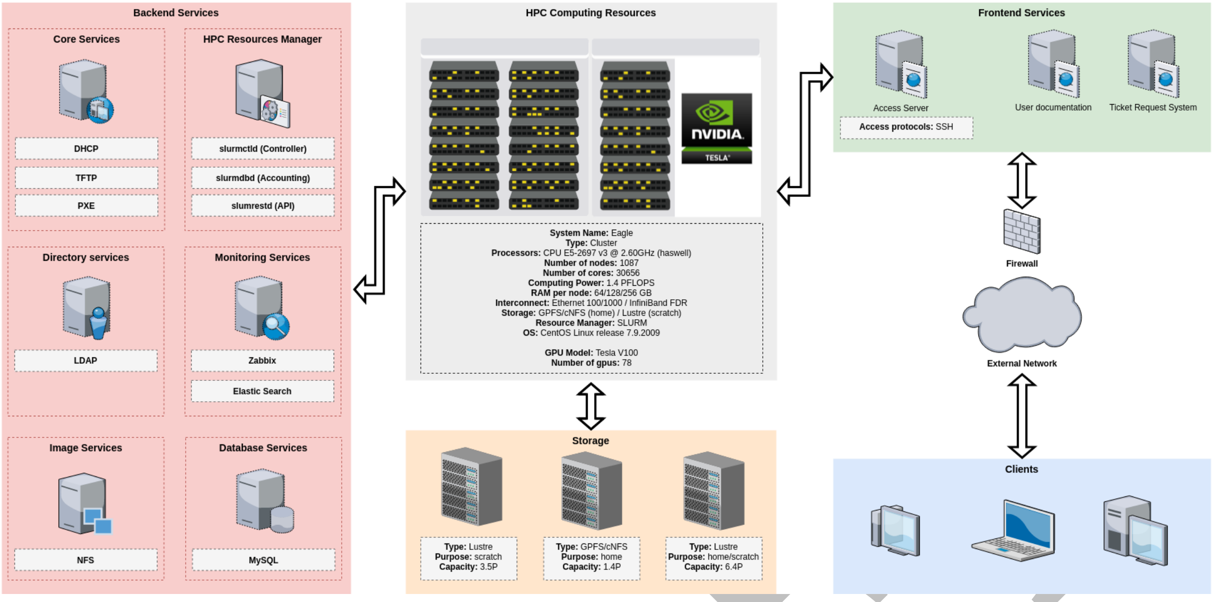
8.1 HPC infrastructure

The HPC and cloud infrastructure available at PSNC will be provided for MARVEL purposes and workflows defined by the scientific communities. PSNC provides two geographically distributed data centers connected with 40 fibers and redundant networking. This redundancy allows to provide a reliable, productive functionality for critical services under clouds ecosystems, i.e., Infrastructure as a Service (IaaS), Software as a Service (SaaS), Platform as a Service (PaaS). Additionally, the major data center provides High Performance Computing (HPC) infrastructure for grand challenges: big applications running in parallel environments.

The major HPC services are provided by two systems: Eagle (since 2016) and Altair (since 2021) delivering conventional power and specialised accelerating environment GP-GPU for AI and big data analysis. The remaining cloud services are provided for specialised requirements at universities (dedicated resources, higher availability and critical services), industry and administration entities.

PSNC resources are used by the scientific community from all over Poland, but also Europe at PRACE and EuroHPC activities (calls under Tier-0 and Tier-1 grant proposals). PSNC also serves abroad users on the basis of agreements on the exchange of computing power under

Partnership for Advanced Computing in Europe (PRACE), nuclear physics – Worldwide LHC Grid Computing (WLCG), Low-Frequency Array for radio astronomy (LOFAR), as well as a number of international ESFRI flagship projects.



System Name: Eagle
Type: Cluster
Processors: CPU E5-2697 v3 @ 2.60GHz (haswell)
Number of nodes: 1087
Number of cores: 30656
Computing Power: 1.4 PFLOPS
RAM per node: 64/128/256 GB
Interconnect: Ethernet 100/1000 / InfiniBand FDR
Storage: GPFS/cNFS (home) / Lustre (scratch)
Resource Manager: SLURM
OS: CentOS Linux release 7.9.2009

GPU Model: Tesla V100
Number of gpus: 78

Type: Lustre
Purpose: scratch
Capacity: 3.5P

Type: GPFS/cNFS
Purpose: home
Capacity: 1.4P

Type: Lustre
Purpose: home/scratch
Capacity: 6.4P

Access protocols: SSH

Figure 41. PSNC HPC infrastructure

The infrastructure provided to MARVEL, illustrated in Figure 41, will consist of access to the HPC supercomputer and access to virtualised private cloud. The supercomputer Eagle, implemented at the beginning of 2016, is one of the most powerful supercomputer systems in the world. The machine offers 1.4 PFLOPS of processing power. It consists of 1087 nodes with a total of more than 30 thousand cores and 120 TB of operational memory (RAM). Additionally, to take full advantage of the AI-based algorithms for audio-visual analytics, the computing platform is equipped with 78 GPUs NVIDIA V100 with 32GB of memory each. These are available as part of the HPC cluster and, at least in the near future, it is the only way to access this resource. PSNC will also provide a sufficient amount of storage memory to handle MARVEL tasks. One of the HPC Infrastructure responsibilities is maintaining the MARVEL Data corpus, which requires at least 3.3 PB of storage. Access to the virtualised private cloud is available via LabITaas. The LabITaas is the processing and data collection node, offering cloud-based services.

The total infrastructure capacity can be summed up under the following values:

- 1,4 PFLOPS of computing power,
- 30 thousand cores,
- 120 TB of operational memory.

PSNC provides HPC/Cloud resources for various MARVEL subsystems operating in the cloud layer. One of them is the Data management and distribution toolkit with orchestrating components located in the cloud tier. Another part of the MARVEL framework that requires HPC resources is the Audio, visual, and multimodal AI subsystem, which typically requires extensive computing power for DL model building. The components of the Optimised E2F2C processing and deployment subsystem are also typically located at the cloud level (e.g., the core functionalities of Karvdash, FedL servers, GPURegex, DynHP). Furthermore, HPC infrastructure will host MARVEL Data Corpus that is the final result of the MARVEL project.

8.2 Management and orchestration of HPC resources

Managing large and complex data sets is a critical process in many areas of scientific, academic, cultural, social, and economic activities. Rapidly changing demands on data access require creating common data management services based on an adequately scalable, capacious and efficient and ubiquitous infrastructure and its services. Such infrastructure should enable migration or replication of data across-domain and timeless importance as well as inclusion of newly acquired data sets of scientific or economic applications.

The infrastructure should also ensure reliable security of this data and provide effective access to data through an extendable set of access services and data presentation mechanisms, supporting various access protocols and data formats. In addition, the service should integrate the functionality and resources for data processing, including large-scale calculations (HPC, HTC), data analytics in the BigData model, or machine learning (ML, AI), without the need to move data to another location.

The spectrum of services in the field of storage, access, and processing of data required by science and economy is very wide, multifaceted, and dynamically changing. The new scientific ways to analyse big amounts of data, the knowledge-based market, ability to provide access to open data is a necessary condition for development, decisive for the competitiveness of researchers, scientific institutions, and companies. Supporting science and economy by providing services, infrastructure, and platforms for integrated data management is one of the basic challenges that have been undertaken by PSNC in the last decade.

In order to harness HPC power to provide fast complex analytics on multimodal data, integration of high-level services and ML modelling with HPC must take into account scheduling and resource management aspects of accessing resources. The goal of this component is to ensure that a large number of resources and a high-speed network and storage offered by HPC will be effectively utilised to guarantee high performance of data processing. It will provide the means for accessing HPC in terms of authentication, resource discovery, and job monitoring. This component will also manage an efficient way to run hybrid, cloud-based applications within an HPC environment accessing GPU as well as offer provisioning and orchestration of resources among multiple infrastructures.

The pilot and pre-production testbeds will be composed of existing computing and storage resources. Access to the HPC cluster Eagle is managed by Simple Linux Utility for Resource Management (SLURM) while virtualised private cloud is available with OpenStack which is deployed as Infrastructure-as-a-Service (IaaS).

8.3 Cloud tier

This subsection reviews the cloud tier of each of the MARVEL pilots/experiments (GRN, MT, and UNS), by discussing the underlying SW frameworks/HW infrastructure, and also the main role(s) of the cloud tier within the pilots' execution.

GRN pilot

GRN will make use of the cloud tier provided by the MARVEL consortium. The intention is to transfer the data from the fog tier to the cloud tier in real-time via Kafka messages or equivalent.

MT pilot

MT will make use of the cloud tier provided by the MARVEL consortium. Data will be transferred to the cloud from the fog server installed at FBK premises. Data will be transferred during the dataset collections for sharing across the partners and for training purposes. For the real-time pilot execution, the use of cloud resources for inference may not be necessary.

UNS experiments

The cloud tier for the drone-based experiment will be hosted at the HPC infrastructure of PSNC. Within the drone experiment, potential usage of the cloud tier will be for model training that cannot be done at the UNS fog tier and for hosting certain MARVEL components needed for the experiment execution – "nodes" of the DMD subsystem (e.g., NiFi), in case the DMD subsystem will be applied, and similarly for Karvdash.

8.4 Fog tier

This subsection reviews the fog tier of each of the MARVEL pilots/experiments (GRN, MT, and UNS), by discussing the underlying devices, communication frameworks, and also the main role(s) of the fog tier within the pilots' execution.

GRN pilot

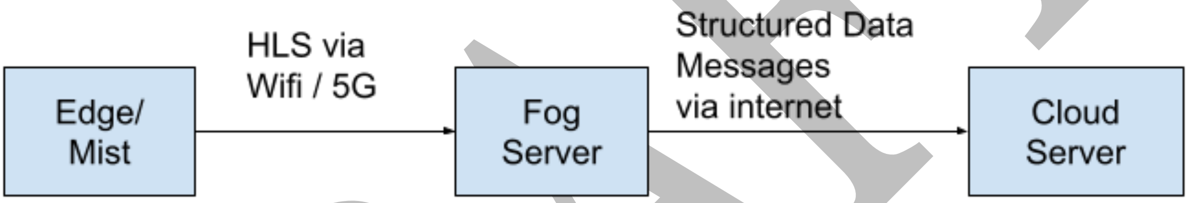


Figure 42. The fog tier and interfaces in the GRN pilot

The HP fog server receives audio-video data (HLS protocol) over wireless channels and outputs in real-time the processed data as Kafka messages, which are received by the cloud tier, Figure 42.

MT pilot

The fog tier will consist of a DELL workstation located at FBK premises, that, via secure connection captures the raw data from the sensors. The server may be equipped with an NVIDIA GeForce RTX 3080.

UNS experiments

The fog tier for the drone-based experiment will consist of a tablet or laptop/desktop for real-time views and a server for fog-based inference. Training of ML/DL models will be either performed at the local servers (fog) or in the cloud. Edge and anonymisation will nominally be performed at the fog tier, with the possibility of shifting these functionalities to the edge, e.g., by attaching Intel NUC to the drone.

8.5 Edge tier

This subsection reviews the edge tier of each of the MARVEL pilots/experiments (GRN, MT, and UNS) by discussing the devices to be used, their capabilities and roles, communication options, as well as different deployment options within the pilots' execution.

GRN pilot

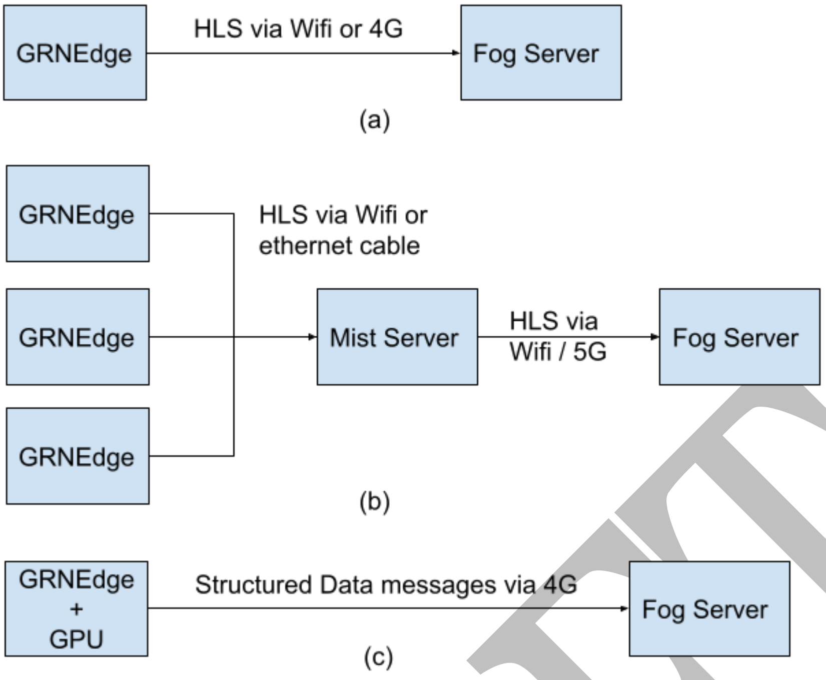


Figure 43. The three edge tier types and interfaces in the GRN pilot

The edge tier in the GRN pilots will take the form of various configurations. Figure 43 shows three examples of such configurations. Example (a) is the case where an isolated GRNEdge device transfers AV data directly to the fog server. In case (b), three co-located GRNEdge devices transfer AV data to a mist server (positioned in very close physical proximity to the edge devices) and then these are forwarded to the fog server. Alternatively, the mist server carries out processing at the edge. In case (c) the GRNEdge device is upgraded with an onboard GPU and only non-binary data is transferred over the wireless network to the fog server.

MT pilot

The edge consists of a series of Raspberry Pis devices installed in the nearest cabinet where selected cameras are installed. In addition, microphones will be mounted on dedicated devices where some limited computational resources are available. It is expected that at the end of the experimentation some modules for the analysis of the video and audio can be installed and executed on the Raspberry Pis.

Cameras. The tables below review the information of cameras to be used within the MT pilot.

Piazza Fiera

Camera	Type	Optic	Brand	Model	Height	Microphone
Fixed	Digital	10-40	Basler	BIP2-1600c-dn	8 m	No
Fixed	Digital	10-40	Basler	BIP2-1600c-dn	8 m	No
Fixed	Digital	10-40	Basler	BIP2-1600c-dn	8 m	No

Piazza Duomo

Camera	Type	Optic	Brand	Model	Height	Microphone
Fixed	Digital		Basler	BIP2-1600c-dn	40 m	No
Fixed	Digital		Basler	BIP2-1600c-dn	40 m	No
Fixed	Digital		Basler	BIP2-1600c-dn	40 m	No

Piazza Santa Maria Maggiore

Camera	Type	Optic	Brand	Model	Height	Microphone
Fixed	Digital	4-12	Basler	BIP2-1600c-dn	4m	No
Fixed	Digital	7,5-75	Basler	BIP2-1600c-dn	4m	Yes
Fixed	Digital	4-12	Basler	BIP2-1600c-dn	7m	Yes

Piazzale Ex Zuffo

Camera	Type	Optic	Brand	Model	Height	Microphone
Fixed	Digital		Basler	BIP-1600c	8 m	Possible
Fixed	Digital		Basler	BIP-1600c	8 m	Possible
Fixed	Digital	3-10,5	Axis	P1435-LE	10 m	Not possible

Train Station - Piazza Dante

Camera	Type	Optic	Brand	Model	Height	Microphone
Fixed	Digital		Basler	BIP-1600c	4 m	No
Fixed	Digital	4-12	Basler	BIP2-1600c-dn	10 m	Possible
Fixed	Digital	4-12	Basler	BIP2-1600c-dn	10 m	Possible
Fixed	Digital	4-12	Basler	BIP2-1600c-dn	3,5 m	Possible

Microphones:

- The Infineon Audiohub Nano (EVAL AHNB IM69D130V01) with four IM69D130 digital microphones on flex board and the following features:
  - Plug&Play
  - Audio streaming over a USB interface
  - 48 kHz sampling rate
  - 24-bit audio data (stereo)
  - Powered through Micro-USB
- Some large aperture ST MEMS arrays will be explored in the staged recordings (FBK).

Figure 44 presents the overall E2F2C infrastructure for the MT pilot.

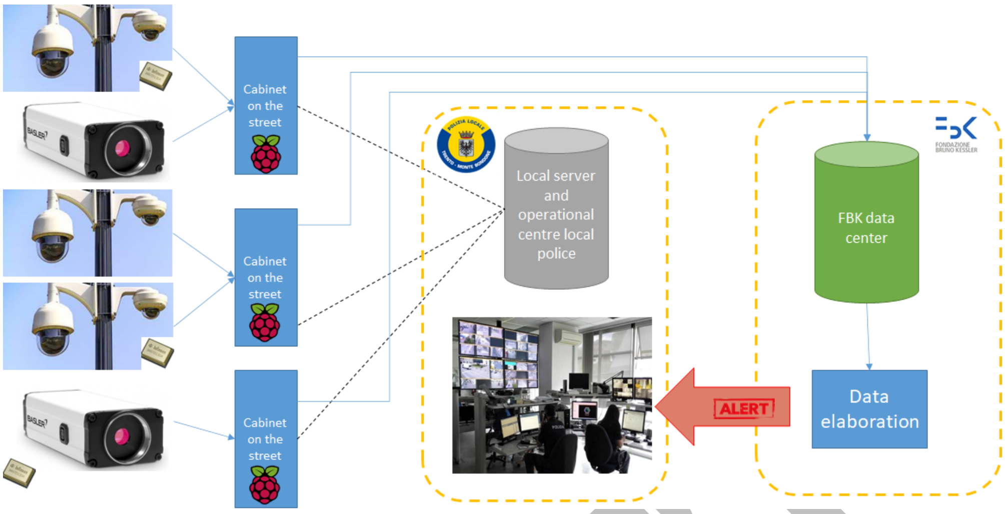


Figure 44. E2F2C infrastructure for the MT pilot

UNS experiments

Hardware

The following elements are integrated on the custom-designed carriers attached to the DJI Matrice 600 Pro drone:

- Raspberry Pi 4 - Model B (4GB RAM) with external camera module V2.1. The Camera Module has a Sony IMX219 8-megapixel sensor and can take high-definition video.
- MEMS microphone(s):
  - The Infineon Audiohub Nano (EVAL AHNB IM69D130V01) with 2 IM69D130 digital microphones on flex board.
- External battery pack to power up all components.
- The hardware will potentially involve also custom-designed LTE-M module developed at the University of Novi Sad (LTE-M is not yet commercially released in Serbia). The LTE-M node integrates the BG96 cellular module from Quectel.

Another RPi with the same configuration (camera, microphones) will be placed near the ground (maybe attached to the lamppost) to capture audio and video.

Video stream captured from RPi camera module is stored locally on the device, but also streamed to an external IP address. Streaming link can be accomplished by direct Wi-Fi communication with RPi module in the local network (temporary solution). Another way to stream captured video is via the LTE-M link that is still in the process of testing.

MEMS microphones will be connected to the same RPi as the camera, and the audio signal will be stored locally on the RPi module (this solution is in the process of testing).

Software

As one of the software components, we will use SensMiner which is an Android app developed by AUD to record environmental acoustics as well as user annotations. Audio is being recorded and the user can in parallel annotate audio and store the corresponding segment in the phone memory.

Figure 45 presents the overall E2F2C infrastructure for the UNS drone experiment.

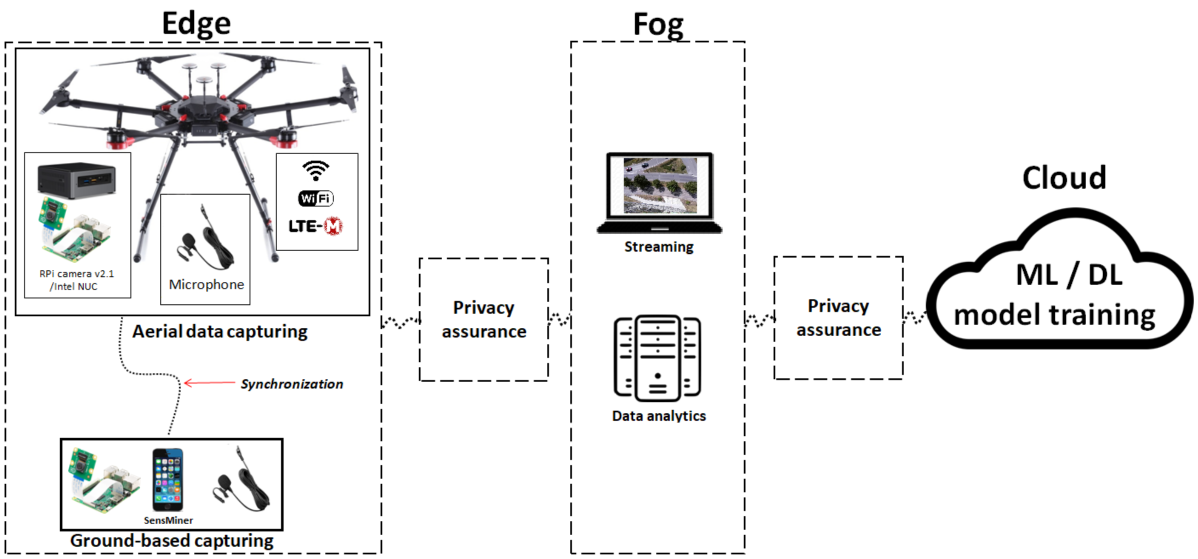


Figure 45. E2F2C infrastructure for the UNS drone experiment

9 System outputs: User interactions and the decision-making toolkit

The decision-making toolkit will be the key means of interaction with MARVEL end-users. It will serve as the key interface of all processes performed in MARVEL in order for the insights produced by them to be able to be consumed by the users. Receiving input from the Data management and distribution subsystem as well as insights from edge-fog-cloud deployed models which will allow the visualisation of recommendations for medium to long-term decisions, the decision-making toolkit can be envisioned as a dashboard with pre-configured widgets according to the preferences of users that will be designed jointly with the pilot providers and other partners before the first MVP of the project. An indicative figure/wireframe of the system outputs can be seen in Figure 46.

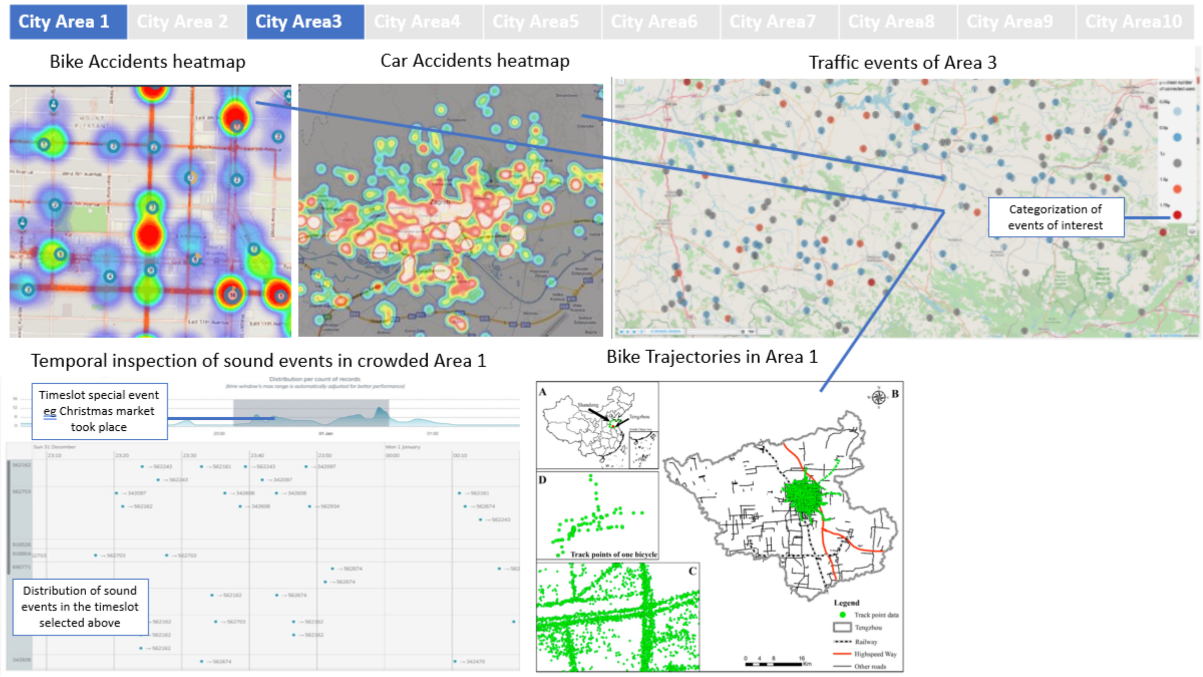

City Area 1
City Area 2
City Area3
City Area4
City Area5
City Area6
City Area7
City Area8
City Area9
City Area10
Bike Accidents heatmap
Car Accidents heatmap
Traffic events of Area 3
Categorization of events of interest
Temporal inspection of sound events in crowded Area 1
Timeslot special event eg Christmas market took place
Distribution of sound events in the timeslot selected above
Bike Trajectories in Area 1
Track points of one bicycle
Legend
Track point data
Railway
Highspeed Way
Other roads

Figure 46. An example of system outputs (Decision-making Toolkit)

The decision-making toolkit will be user-friendly and intuitive in order to allow its use by non-IT users with no need for extensive training. Short self-paced tutorials will be provided to navigate the user through its widgets and functionalities on their first use. More information (at a high level) is provided in Section 10.2 of this document and will be further described after the necessary workshops with end-users for the envisioning, prototyping, and testing of the preconfigured views as well as the rest of the elements to be included.

MARVEL Data Corpus-as-a-Service will serve as a key source of interaction with developers and researchers. It will serve as a pool of data available to the public, enabling the scientific and industrial community to enhance audio-visual analytics and enforcing the foundations for smart city services that will improve the security and well-being of their citizens.

9.1 SmartViz

SmartViz can visualise data coming in real-time or in batch mode from the Data Corpus and the Data management and distribution subsystem.

Based on its modular internal architecture, presented in Figure 47, it can be used to provide visualisation configurations, covering the needs of a variety of user paths that might be required in order to address all MARVEL stakeholders. Using its Data Intake adapters, SmartViz is capable of connecting with multiple data sources and then use its internal data API and configuration options to produce predefined as well as user-defined visualisation dashboards. Existing adapters can already support RESTful APIs and real-time data feeds (i.e., Kafka topics). However, more data adapters can be plugged in to support different data sources if needed. The output of the adapters is handled by a middleware, that transforms information into internal data representations, which can therefore feed the visualisations. This middleware also hosts configuration options, managing parameters like user authentication and authorisation, dashboard sharing options, interface layout options, etc.

The Frontend part of the tool is served as a web application directly accessible by end-users. Taking advantage of the described setup, data can be displayed using a pool of visualisations. Such visualisations can include elements varying from simple charts and graphs to more complex timeline representations, geospatial depictions, and real-time rendering of data streams. Moreover, advanced filtering mechanisms, interconnected visualisations, and data comparison features are available to allow multiple ways of data presentation and foster exploratory data analysis.

Using its aforementioned capabilities, SmartViz aims to help users greatly decrease the time needed to gain a solid situational awareness. Therefore, decision-makers will be empowered to discover patterns, behaviours, and correlations of data items via a visual data exploration process that will be supported and enhanced by knowledge gained about the available information by the rest of the MARVEL subsystems.

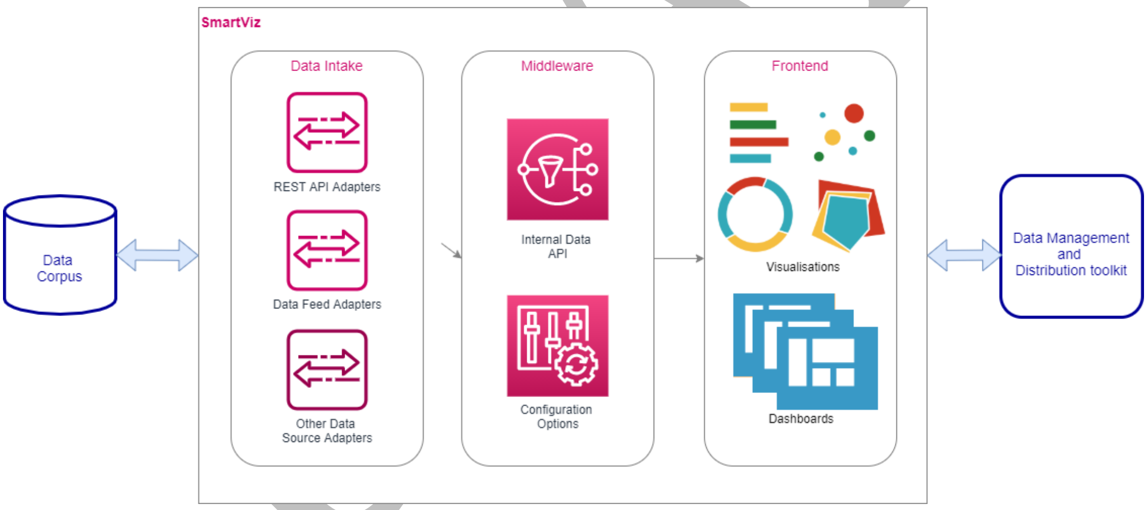


Figure 47. High-level internal architecture of SmartViz

9.2 MARVEL Data Corpus-as-a-Service

MARVEL Data Corpus-as-a-Service will be one of the major outcomes of the MARVEL project trying to address the lack of extremely large public sets of annotated audio-visual recordings that have been an obstacle for the scientific and industrial community to enhance audio-visual analytics. MARVEL Data Corpus is an extreme-scale public corpus of processed multimodal audio-visual open data, obtained free of charge and released as a service, under the aegis of specific service level agreement (SLA). It will address all aspects that are related to efficient sharing of heterogeneous data pools such as accessibility, operability, managing streaming and network, legal considerations, security, privacy and technical concerns.

MARVEL's Data Corpus will enable smart cities to build and deploy innovative applications that are based on multimodal perception and intelligence and will be freely available for all societal EU data marketplaces. Furthermore, MARVEL Data Corpus-as-a-Service will enable the possibility for SMEs and start-ups to build on top of these data assets and create new businesses by exploring extreme-scale multimodal analytics.

From a technical point of view, the MARVEL Corpus component consists of four subcomponents:

i. an extreme-scale audio-visual recording repository where processed data will be stored;

ii. a database for storing the respective metadata where folder structure might be different from a physical file directory structure;

iii. a management application; and

iv. a series of application programming interfaces (APIs) that will allow accessing and querying recordings stored in the repository.

Figure 48 represents a high-level architecture of the MARVEL Data Corpus-as-a-Service including the previous subcomponents. The extreme-scale audio-visual recording repository where processed data will be stored will adapt the HBase[18] distributed database, an open-source non-relational distributed database modelled after Google's Bigtable[19] that will be built and run on top of Hadoop Distributed File System[20] (HDFS).

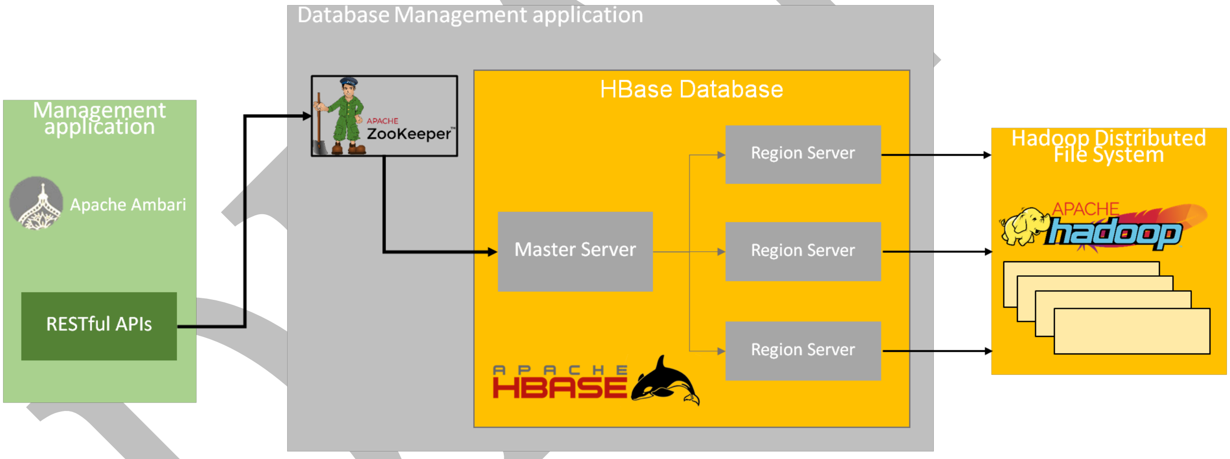


Figure 48. A high-level architecture of the MARVEL Data Corpus-as-a-Service

Based on the previous figure, the management application of the MARVEL database will be based on Zookeeper[21], an open-source project that provides services like maintaining configuration information, naming and providing distributed synchronisation. The database that will store the respective metadata will be based on an HBase distributed database in which a master server assigns regions to the region servers and will handle load balancing of the regions across region servers by unloading the busy servers and shifting the regions to less occupied servers. In HBase, a table is both spread across a number of region servers as well as

[18] https://hbase.apache.org/

[19] https://cloud.google.com/bigtable

[20] https://hadoop.apache.org/

[21] https://zookeeper.apache.org/

being made up of individual regions. As tables are split, the splits become regions. Regions store a range of key-value pairs, and each region server manages a configurable number of regions. The management application of the MARVEL Corpus will be based on the Ambari[22] project that is aimed at making Hadoop management simpler by developing software for provisioning, managing, and monitoring Apache Hadoop clusters. Ambari provides an intuitive, easy-to-use Hadoop management web user interface backed by its RESTful APIs.

[22] https://ambari.apache.org/

10 User interface

10.1 IT users of MARVEL platform

This subsection considers the user interface of MARVEL components. In particular, the section presents in Table 6:

- access to and interactions with the component;
- configuration/initialization of the component; and
- authentication and authorisation aspects.

Table 6: Summary of UI requirements per component

Component	Component access and platform interactions	Authentication/ Authorisation	Configuration/ Initialization
Sensing and perception subsystem			
Advanced MEMS microphones	No user interaction; processing of the data by the partners (pilots) GRN, MT, UNS.	No user interaction; processing of the data by the partners (pilots) GRN, MT, UNS.	Connection via PDM or USB.
SED@Edge	Operates on the device where microphones are mounted. Access will be restricted to FBK staff, no intervention is expected from end-users managing the pilot infrastructure.	Access will be restricted to FBK staff.	The software will be developed, configured, installed, and managed by FBK staff. Access will be restricted to FBK staff, no intervention is expected from end-users managing the pilot infrastructure.
GRNEdge	AV data collected by audio-video sensors are streamed to the fog layer.	Authorisation and authentication will be managed by GRN.	The device is installed, configured, initialised, accessed, and managed by personnel authorised by GRN.
AVDrone	Operated by licensed users.	Authorisation and authentication will be inherited from the MARVEL platform.	Configuration, initialization, and management will follow appropriate instructions from MARVEL partners.
sensMiner	Being an Android app, it is accessed by the respective phone owner.	Inherited from the phone settings on which the app is installed.	Configuration will be done by AUD, with tags available to all users during data collection.
CATFlow	Access will be provided by GRN.	Authorisation and authentication will be managed by GRN.	Configuration and initialisation will be handled by GRN.
Security, privacy, and data protection subsystem			
EdgeSec	Accessed by IT users as a base for secure computing (several provided security services and instantiation of secure Docker containers).	Local authentication on the host machine.	Prior configuration includes BIOS setup, drivers/software installation, management of encrypted keys.

VideoAnony	Connects either with live video streams from cameras or receives an already recorded video.	Authentication is needed (no further details are given).	Uses standard configuration files for initiation and execution.
AudioAnony	Connects either with live audio streams from the microphones or receives an already recorded audio.	Authentication on the host machine.	Initiation & configuration not available to IT users.
VAD (devAIce)	Accessed by the consuming components of the MARVEL architecture.	AUD will provide licenses to activate the VAD/devAIce SDK. It can also be tested offline with a Python package.	Initialized by the consuming components of the MARVEL architecture, based on instructions detailed in the devAIce documentation.
Data management and distribution subsystem			
Data Fusion Bus	Receives data from different data-producing components through a Kafka interface. Its UI can be accessed by IT users to monitor and process collected data.	Has its own Keycloak module that manages user identities and privileges.	Initiated as a dockerized service on Kubernetes cluster. UI Configuration can be realised by an admin user, to grant/revoke rights.
StreamHandler	Receives data from different data-producing components through a Kafka interface.	Authentication and authorisation procedures for StreamHandler will follow the standard procedure within the MARVEL framework.	Initiated as a dockerized service on Kubernetes cluster.
DatAna	DatAna relies on the Apache NiFi graphical user interface to define graphically data flows and its API. Potential extensions of this GUI and API might be needed in the scope of MARVEL.	It can be used with no authentication or relying on the authentication features provided by NiFi (via client certificates, username/password, Apache Knox, or OpenId Connect).	As it relies on Apache NiFi and MiNiFi, the configuration should follow the recommendations from NiFi. If multiple instances of DatAna are to be installed, the configuration will include the necessary steps to connect them (i.e., multiple MiNiFi agents connected to one or several NiFi instances in the Fog/Cloud layers) and to set them up (i.e., setting up a NiFi cluster with several machines for scaling it up).
HDD	Access will follow the standard procedure within the DMD subsystem. Interactions with DMD components only.	Authentication and authorisation procedures for the HDD component will follow the standard procedure within the MARVEL framework.	Comes with default configurations that can be modified by the user. Initialization is done by the user.
Audio, visual, and multimodal AI subsystem			
devAIce	Access to the devAIce SDK is controlled by a license, provided	Access to the devAIce SDK is controlled by a	Comes with a default configuration. AUD can adapt this based on the

	by AUD.	license, provided by AUD.	MARVEL corpus data.
ViAD, AVAD, VCC, AVCC	Follow MARVEL's standard procedure for access and interactions.	Access will be secured by MARVEL's user management (IdP) system.	Default configurations are provided. Optionally, configurations can be modified by IT users.
AAC, SED, SELD, ASC	Follow MARVEL's standard procedure for access and interactions.	Access will be secured by MARVEL's user management (IdP) system.	Default configurations are provided. Optionally, configurations can be modified by IT users.
Optimised E2F2C processing and deployment subsystem			
GPURegex	No UI available. GPURegex interacts with other components in a processing pipeline, operating as an accelerator for a specific workload.	Authentication and authorisation for GPURegex will follow the unified single sign-on (SSO) architecture selected for the MARVEL framework.	Initialization and configuration occur via a C API, as a standalone component.
DynHP	DynHP will access the Model Library for the required uncompressed models and to the Data Management Module. The output compressed model will be passed to the Karvdash Module.	Authorisation and authentication will follow the standard procedure within the MARVEL framework.	Initialization includes an input model to compress, a configuration file, and a dataset to use for training and compression.
FedL	Can be accessed by authorised IT users.	Authorisation and authentication will follow the standard procedure within the MARVEL framework.	Can be configured by authorised IT users (model, learning parameters, FL protocol, and possibly protocol-defining parameters).
Karvdash	Can be accessed by IT users via HTTPS over secure. Offers IT users a dashboard to running services on the E2F2C platform.	All accesses use HTTPS, for authenticated users, through its own authentication module.	Installed on a Kubernetes deployment with Helm.
E2F2C infrastructure			
HPC infrastructure and HPC management and orchestration	*Eagle:* Access to the system is usually done via SSH tools and internally one can use command line interface to submit and define jobs. Some functionality, especially related to monitoring job state or resource state can be done using REST API *Cloud:* Definition, configuration, and other interaction with the service may be done using either user portal (horizon), using REST API, or CLI.	Authorisation and authentication are done via the PSNC project management portal where one defines resources, users, etc. As for the authentication, it is possible to integrate external IdP as identity provider both for Open Stack and Eagle cluster.	Configuration and initialization of the projects on both platforms (HPC and cloud) is done using project management portal.
Edge, Fog, and Cloud tier MT	MT manages authorisation and authentication to its network and provides access to devices of its edge tier (e.g., cameras) to FBK.	FBK manages authorisation and authentication to its server (fog layer). MT will make use of the PSNC cloud, hence for this tier authorisation and	Configurations of the relevant infrastructure elements at the fog tier will be handled by FBK. Configurations at the cloud tier will be handled

		authentication will follow those of the MARVEL platform.	by PSNC.
Edge, Fog, and Cloud tier GRN	GRN considers providing access to a VM on its fog layer.	Authorisation and authentication to this VM will be managed by GRN For use on the PSNC cloud, authorisation and authentication will follow those of the MARVEL platform.	Configuration and the initialisation of this VM will be handled by GRN.
Edge, Fog, and Cloud tier UNS	All interactions are the same as with the AVDrone component.	For use on the PSNC cloud, authorisation and authentication will follow those of the MARVEL platform.	Configuration is inherited from the AVDrone component.
System outputs: User interactions and decision-making toolkit			
SmartViz	Mainly used by MARVEL's end-users. Connects with components that produce data that can be visualised, especially through the Data Management component.	Access will be secured by MARVEL's user management (IdP) system.	Configured at a user level as a standalone UI application.
MARVEL Data Corpus-as-a-service	The decision-making toolkit must have access to the stored processed audio-video data of the MARVEL Data Corpus. Such access will be provided by respective APIs with respect to the user authentication and authorisation.	Authentication and authorisation will be handled by the respective APIs of the toolkit.	MARVEL Data Corpus will be continuously augmented during the duration of the project with the data produced by the three MARVEL pilots (GRN, MT, UNS).

10.2 Non-IT users of MARVEL platform

As mentioned in earlier paragraphs, the decision-making toolkit will serve as the key means of interaction with the MARVEL end-users. The toolkit will:

- Support the easy exploration of queries regarding all MARVEL use cases providing visualised evidence of processed audio-visual data in order to make medium to long-term decisions. The processed data will be also visualised in juxtaposition with other context-enriching data (e.g., weather data, information from incident reporting systems, parking sensors, etc.) and will allow the user to interact with them in order to investigate potential correlation or causations.
- Visualise data across time and/or interaction points or moments of interest in order to facilitate the revelation of hidden insights. This feature can be also leveraged in all use cases but should be particularly interesting in the ones related to the drivers' behaviour in Malta and the analysis of a specific area in the municipality of Trento.
- Provide attention maps based on real-time data to enable the real-time monitoring of the streets of Malta and highlight any traffic anomalous events as well as visualise the bicycles' movement trajectories in the city. The same concept will be tried out for the analysis of a specific area and the monitoring of a parking place in the municipality of

Trento. The possibility of enhancing those maps with textual annotations and indications of associated audio events will also be explored.

- The maps described earlier will enable the triggering of alerts based on detected events for short-term decisions and monitoring, supported by a rule-based engine. A monitoring dashboard will provide an overview of the ongoing alerts and the events detected.

End-users will only need a browser to access SmartViz. Regarding authentication and authorisation this will not be performed at the SmartViz level but at the framework level where the MARVEL solution will be running.

10.3 User interface for MARVEL Data corpus

Since the management application of the MARVEL corpus will be based on the Apache Ambari project, a user interface for MARVEL Data Corpus will be deployed on top of the latter aiming to provide friendly and simple access to third-party users and stakeholders in order to perform simple and advanced queries to the metadata and gain access to the stored processed multimodal audio-visual data. Although MARVEL Data Corpus will be free of charge, user registration will be mandatory in order to gain access to the corpus.

Furthermore, since Data Corpus will be provided also as a service, all aspects that are related to efficient sharing of heterogeneous data pools, namely accessibility, operability, managing streaming and network, legal considerations, security, and privacy concerns will be considered. Data Corpus will adopt a Service Level Agreement-enabled Big Data analytics framework under the auspices of security SLAs in order to make third parties confident for using the dataset. Based on the adopted security SLA's that will be tailored for the needs of MARVEL Data Corpus access, it is expected to lead to a maximisation of the impact that the MARVEL corpus will have on the international scientific and research community.

11 Mappings to the relevant reference architectures

11.1 Mapping to BDV – Big Data and Analytics/Machine Learning Reference Model

Figure 49 provides a mapping of the MARVEL's conceptual architecture to BDV – Big Data and Analytics/Machine Learning Reference Model (BDV RA) [3]. The Sensing and perception subsystem of the MARVEL architecture maps to the ”Things/Assets, Sensors and Actuators (Edge, IoT, CPS)” block of the BDV RA. This subsystem is also the main source of (AV) data in the MARVEL architecture, and hence it maps also to the ”Media Image Audio” vertical data types block of the BDV RA. The Security, privacy, and data protection subsystem maps to the horizontal block ”Data protection” in the related aspects of privacy protection, anonymisation, and GDPR compliance, but also with regards to Responsible AI; the detailed analysis of MARVEL's approach and guidance to responsible and trustworthy AI can be found in deliverables D1.2 [1] and D9.4 [8]. of MARVEL. Besides data protection, this subsystem also concerns framework security at all architectural levels including secure transmissions, and as such it maps to the cross-cutting vertical concern ”Cybersecurity and Trust” of the BDV RA. The Data management and distribution subsystem maps to the ”Data management” horizontal block of the BDV RA.

The Audio, video, and multimodal AI subsystem maps to the ”Data Analysis” horizontal block of the BDV RA, in the aspects of ML/DL model building from large-scale datasets located at the cloud tier, and similarly for model building over data-at-rest or data-in-motion (through FedL), indicated by the pink field *Training* in the middle of the figure spanning edge, fog, and cloud. The subsystem Optimised E2F2C processing and deployment concerns deployment of AI tasks (ML/DL models), but also model optimisation (compression, acceleration), and hence it maps also to the ”Data Analysis” block of the BDV RA, and similarly for the violet E2F2C spanning field *Inference*, which indicates ML/DL models at runtime, making inference. The E2F2C infrastructure of the MARVEL architecture maps to the ”Data Processing Architecture” horizontal field of BDV RA, and, specifically, the Cloud tier of MARVEL maps to the ”Cloud and High Performance Computing (HPC) horizontal field of the BDV RA.” The last subsystem of MARVEL: System outputs: user interactions and the decision-making toolkit maps to the ”Data Visualisation and User Interaction” horizontal field of BDV RA, where the mapping of the MARVEL Data Corpus-as-a-Service is in the sense of user interactions through various queries over data. Finally, the MARVEL Data Corpus-as-a-Service maps also to the ”Media Image Audio” vertical data types block. Moreover, it maps to and provides a specific instance of the vertical field ”Data sharing platforms, Industrial/Personal” of the BDV RA.

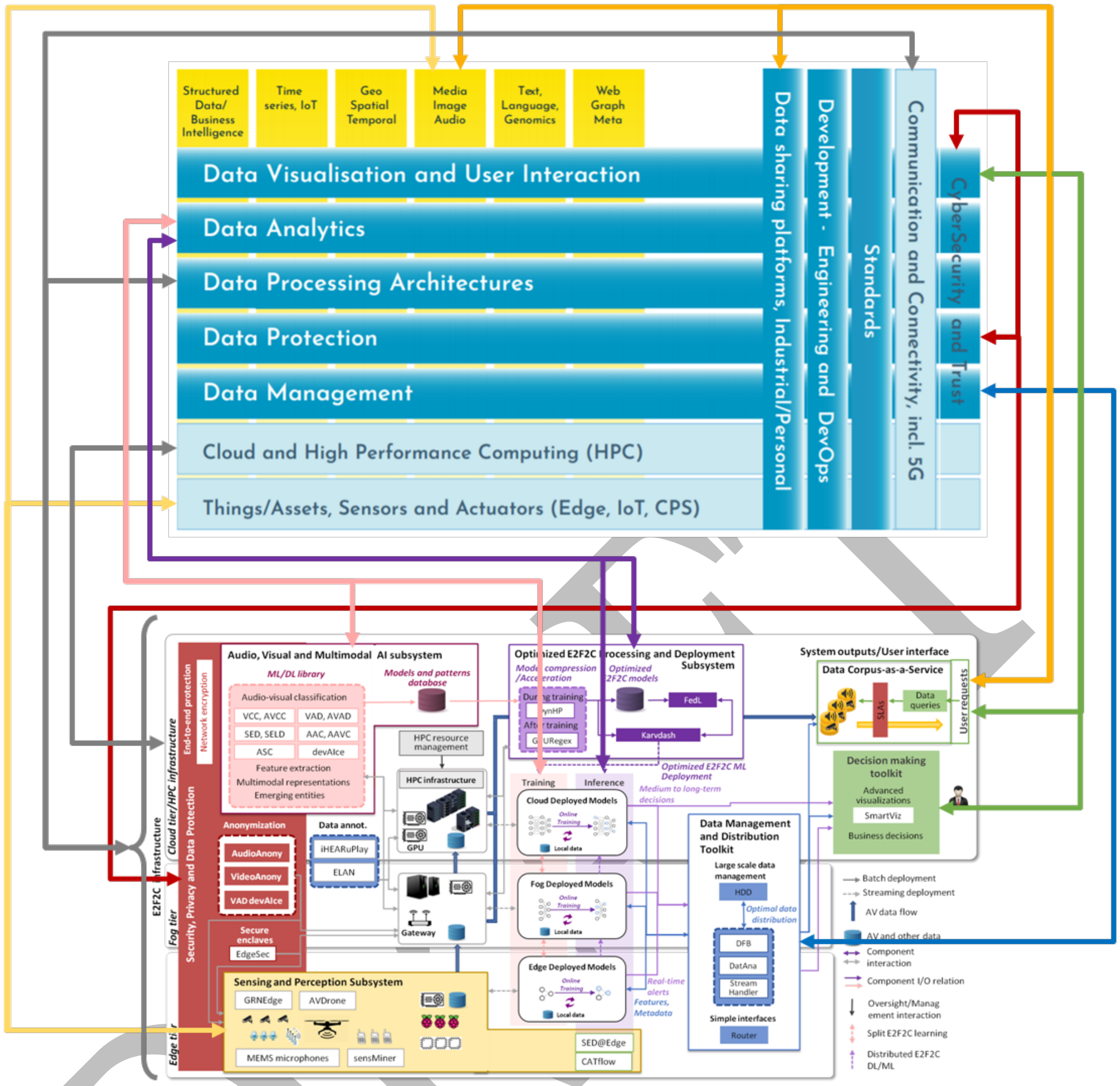


Figure 49. Mapping of MARVEL's conceptual architecture to BDV – Big Data and Analytics/Machine Learning Reference Model [3]

11.2 Mapping to the NIST Fog Computing Conceptual Model (Edge-Fog computing)

Figure 50 provides a mapping of MARVEL's conceptual architecture to the NIST Fog Computing Conceptual Model (Edge-Fog computing) [5]. With the NIST Fog Computing Conceptual model, the identified computing tiers of the underlying infrastructure are: Centralised (cloud) services tier, located at the top of the figure, Fog computing tier, in the middle of the figure, and End-devices tier, at the bottom of the figure. As a sub-tier, the model also identifies the Mist Computing tier right above the End-devices tier, to indicate computing close to the data sources. MARVEL conceptual architecture in the E2F2C computing sense exhibits a close similarity with the NIST Computing Conceptual model, as shown by the grey arrows in Figure 50. Specifically, with the clod and the fog tiers of the two models, the mapping is one-to-one. The main difference arises with the bottom tier, which in the case of the NIST model consists only of edge sensors and actuators, whereas computations are performed at the close-by Mist computing tier. With MARVEL, the Mist computing is subsumed by the MARVEL edge tier. This is mainly motivated by the physical deployment configurations that are foreseen in the project, obtained from MARVEL pilots (MT, GRN, UNS), where computing devices from the Mist computing tier are in most (if not all) cases located *in-situ,* i.e., they are physically collocated with the sensing devices, and hence, from a communications perspective, can be modelled as one tier – edge. On the other hand, computing devices in the fog tier typically require wireless connection with the edge tier, motivating the separate tier – fog.

Besides the E2F2C computing, there are also other important similarities/consistencies to note. The *Data mining and analytics* indicated by red arrows in the fog computing tier of the NIST model correlates with the *Training* and *Inference* fields in the MARVEL architecture, indicated respectively by pink and violet E2F2C spanning fields in the middle of the figure; in addition to the fog, with MARVEL, Data mining and analytics-related computing services are also to be deployed at the edge. The *Collaboration/Federation links* in the NIST model, indicated by dashed red lines, correlate with MARVEL's Distributed E2F2C Processing and Deployment subsystem, shown in violet at the top-middle part of the figure; this subsystem deals with the *optimal deployment of AI and other services over distributed E2F2C resources* (MARVEL's Karvdash component), with *federated learning* as a specific instance of collaborative distributed computing (MARVEL's FedL component).

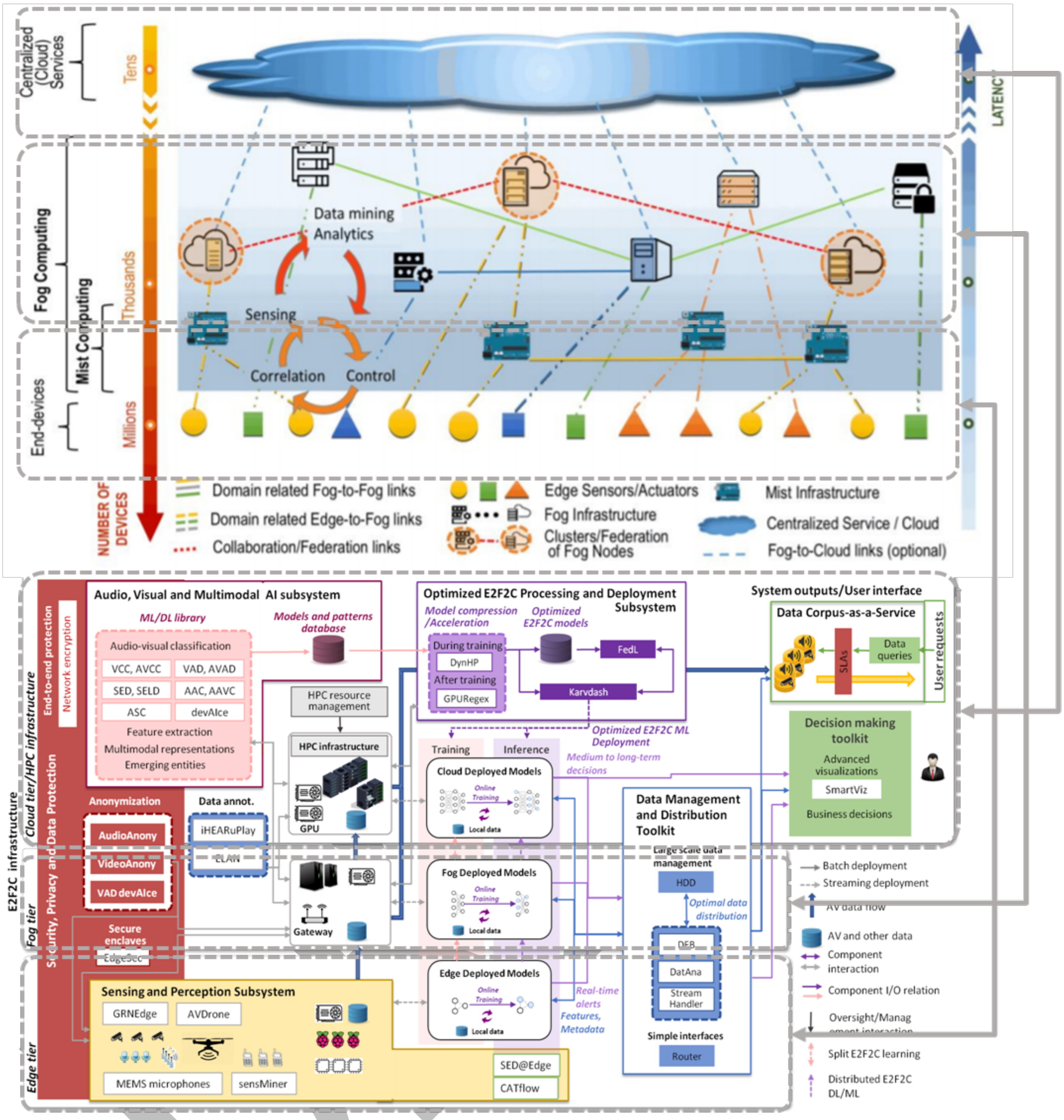


Figure 50. Mapping of MARVEL's conceptual architecture to the NIST Fog Computing Conceptual Model (Edge-Fog computing) [5]

11.3 Mapping to the European AI, Data and Robotics Framework and Enablers

The European AI, Data and Robotics Framework and Enablers proposed within the Strategic Research, Innovation and Deployment Agenda: AI, Data and Robotics Partnership [4] presents the key societal, innovation and technology enablers for development and adoption of AI, Data, and Robotics in Europe. We focus here on the technology enablers of the framework, identifying which MARVEL subsystems and components contribute the most to which technological enablers of the framework, while also touching upon some of the societal and innovation ecosystem enablers. Figure 51 provides a mapping of MARVEL's conceptual architecture to European AI, Data and Robotics Framework, and Enablers. We next detail each of the mappings shown in the figure.

As described in [4], Sensing and perception technologies are the crossover point of the physical world and its digital representation, and encompass technologies to deliver data and create information over which learning and decision-making take place. Combining MARVEL's technologies for acquiring, aggregating, fusing and streaming signals from cameras, microphones, and other devices, MARVEL's Sensing and perception subsystem maps to this block of the framework. In addition, the Sensing and perception subsystem of MARVEL encompasses also technologies for embedding perception at the edge, such as DL models that can fit in microcontrollers and similar resource-constrained devices.

Knowledge and Learning technologies of the framework create knowledge from data obtained by the Sensing perception technologies. In MARVEL, these technologies are represented by the Audio, visual, and multimodal AI subsystem and by the Optimised E2F2C processing and deployment subsystem.

Reasoning and Decision making technologies build on the knowledge from the Knowledge and Learning technologies to deliver edge and cloud-based decision-making. MARVEL's decision-making toolkit, with real-time alerts and medium to long-term business analytics, contributes to this block of the framework. Action and Interaction technologies in the context of MARVEL map to aspects of human interaction, and are in MARVEL achieved by the System outputs: User interactions and the decision-making toolkit subsystem.

Finally, Systems, Methodologies, Hardware, and Tools provide the technologies that enable the construction and configuring of systems, integrating AI, Data, and Robotics technologies into systems, to ensure the core system properties such as safety, robustness, dependability, and trustworthiness. In MARVEL, these technologies are represented by the subsystems: Data management and distribution subsystem, Optimised E2F2C processing and deployment, E2F2C infrastructure, and also Security, privacy, and data protection subsystem.

The security, privacy, and data protection subsystem by addressing GDPR, and data protection in general, and the Responsible AI within the project contributes to the European Fundamental Rights, Principles, and Values aspects/enablers of the framework. MARVEL's Data Corpus-as-a-Service, envisioned as a large-scale database of AV data to be obtained free of charge, and with SLA-embedded queries, contributes both to Data for AI innovation ecosystem enabler, but also to Capturing Value for Business, Society, and People aspects of the framework.

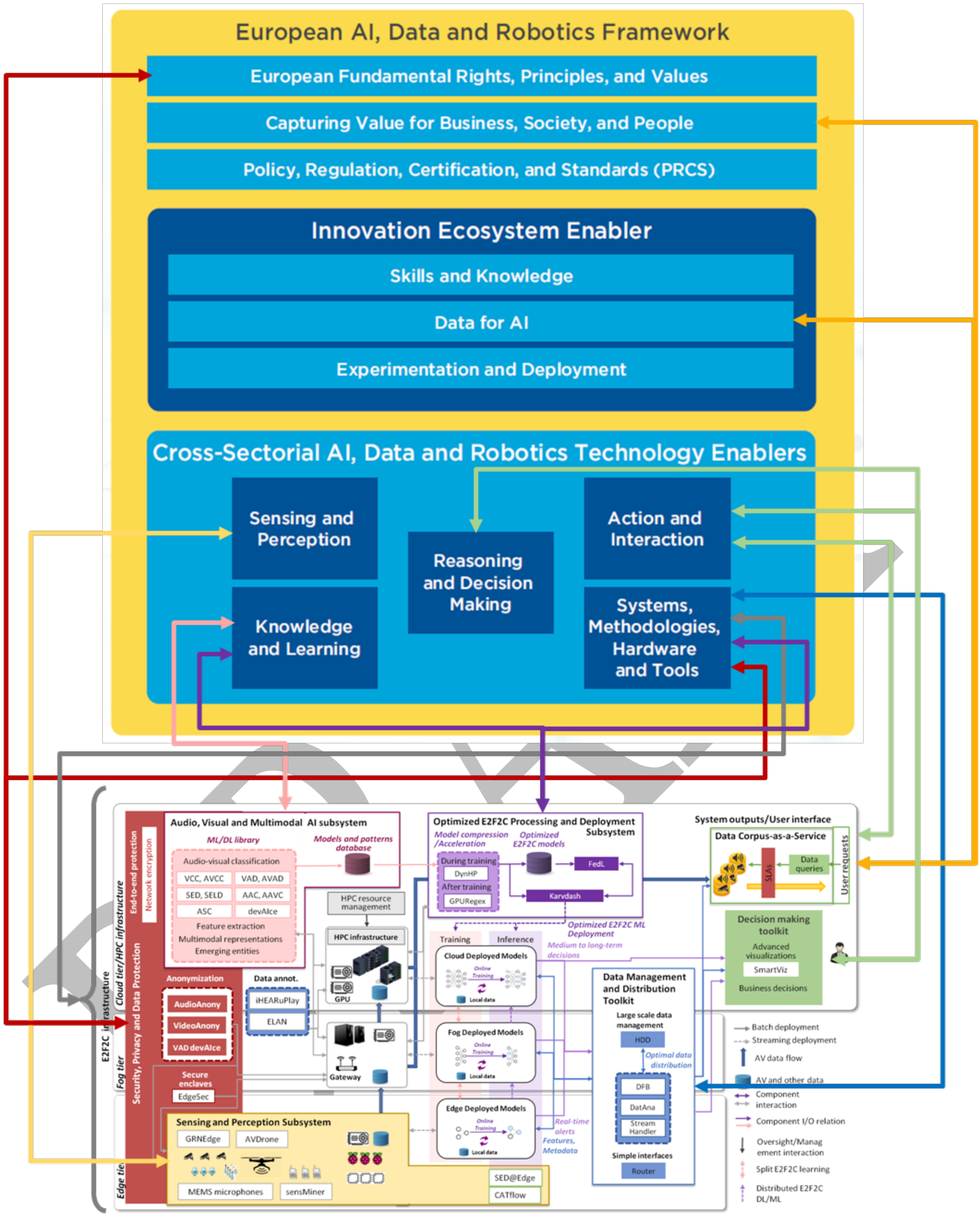


Figure 51. Mapping of MARVEL's conceptual architecture to European AI, Data and Robotics Framework and Enablers [4]

11.4 Mapping to Data value pipeline, *DataBench*

Figure 52 provides mapping to the Data value pipeline created within the DataBench project (for details, see also Section 12.3 below). The first pipeline stage – "Data Acquisition /Collection" is in MARVEL represented mainly by the Sensing and Perception subsystem, for Data acquisition, Streaming, Data Extraction and in some cases also Storage, depending on the type of processing and storage at the edge devices. For aspects related to data acquisition, the first pipeline stage is also represented by the MARVEL Data management and distribution subsystem, however, the subsystem is mainly mapped to the second pipeline stage – “Data Storage/Preparation”. With regards to data protection, this pipeline stage is represented by the MARVEL’s Security, privacy, and data protection subsystem, while data curation, integration, and publication are represented by the database subcomponent of MARVEL Data Corpus-as-a-Service. The third stage of the pipeline – Analytics/AI/Machine Learning is represented by two MARVEL subsystems: (i) Audio, visual and multimodal AI subsystem – for ML model training; and (ii) Optimised E2F2C processing and deployment – for operation, model verification. Finally, the third pipeline stage – Action/Interaction/Visualisation/Access is represented by the final MARVEL subsystem: Systems outputs: User interactions and the decision-making toolkit, which includes data presentation environment (both for MARVEL Data Corpus and the decision-making toolkit), and user action and interaction.

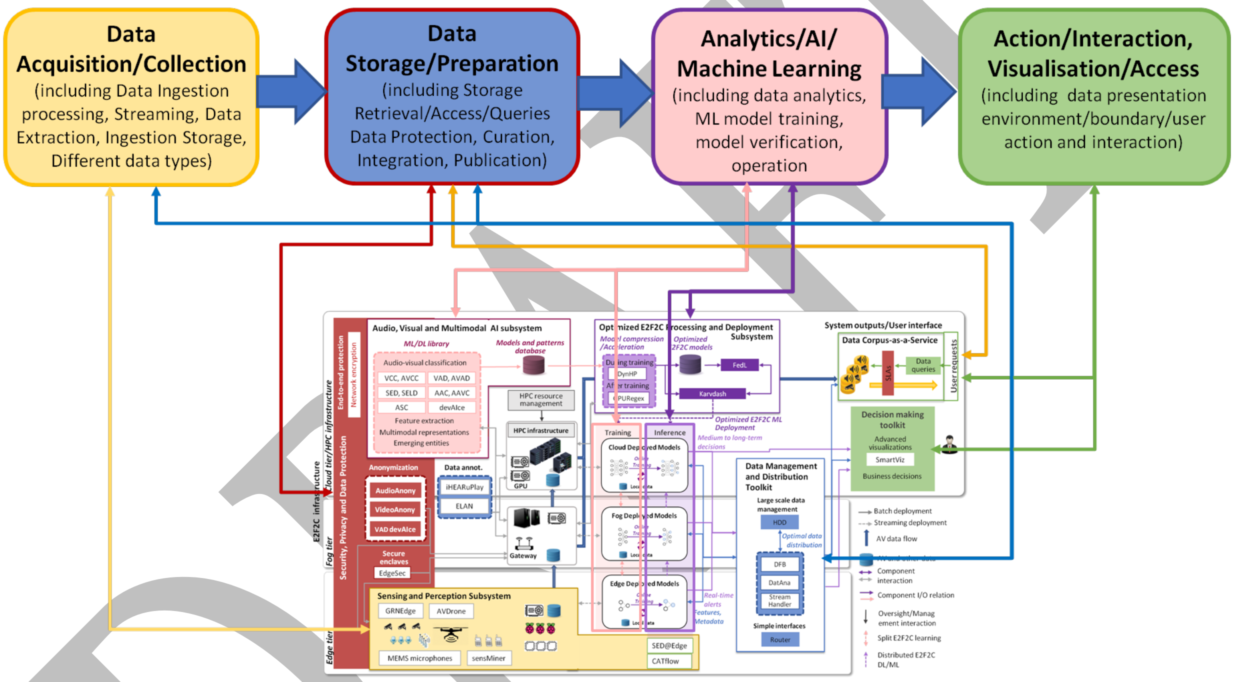

Data Acquisition/Collection
(including Data Ingestion processing, Streaming, Data Extraction, Ingestion Storage, Different data types)
Data Storage/Preparation
(including Storage Retrieval/Access/Queries Data Protection, Curation, Integration, Publication)
Analytics/AI/ Machine Learning
(including data analytics, ML model training, model verification, operation
Action/Interaction, Visualisation/Access
(including data presentation environment/boundary/user action and interaction)
E2F2C infrastructure
Cloud tier/HPC infrastructure
Fog tier
Edge tier
Security, Privacy and Data Protection
End-to-end protection
Network encryption
Audio, Visual and Multimodal AI subsystem
ML/DL library
Models and patterns database
Audio-visual classification
VCC, AVCC
VAD, AVAD
SED, SELD
AAC, AAVC
ASC
devAIce
Feature extraction
Multimodal representations
Emerging entities
Anonymization
AudioAnony
VideoAnony
VAD devAIce
Secure enclaves
EdgeSec
Data annot.
iHEARuPlay
ELAN
HPC resource management
HPC infrastructure
GPU
Gateway
Sensing and Perception Subsystem
GRNEdge
AVDrone
MEMS microphones
sensMiner
SED@Edge
CATflow
Optimized E2F2C Processing and Deployment Subsystem
Model compression /Acceleration
Optimized E2F2C models
During training
DynHP
After training
GPURegex
FedL
Karvdash
Optimized E2F2C ML Deployment
Training
Inference
Cloud Deployed Models
Fog Deployed Models
Edge Deployed Models
Online Training
Local data
Medium to long-term decisions
Real-time alerts
Features, Metadata
Data Management and Distribution Toolkit
Large scale data management
HDD
Optimal data distribution
DFB
DatAna
Stream Handler
Simple interfaces
Router
System outputs/User interface
Data Corpus-as-a-Service
SLAs
Data queries
User requests
Decision making toolkit
Advanced visualizations
SmartViz
Business decisions
Batch deployment
Streaming deployment
AV data flow
AV and other data
Component interaction
Component I/O relation
Oversight/Management interaction
Split E2F2C learning
Distributed E2F2C DL/ML

Figure 52. Mapping of MARVEL’s conceptual architecture to the Data value pipeline, *DataBench*

12 Variations of the architecture across different use cases

12.1 Use case-components mappings

Tables 7-16 provide use case-components mappings, with detailed information on the application of the components for each use case. We use the following notation: “Y”=Yes indicates that the component will be used in the respective use case; “N”=No indicates that the component will not be used; “M”=Maybe indicates that the component will likely be used, provided that the corresponding conditions are met (e.g., availability of the necessary hardware). Table 17 then provides the overview of the use case-components mappings in a single table.

Table 7: MARVEL components in the GRN1 use case

GRN use case 1: Safer Roads			
Component owner	**Subsystem /Component**		**Comments on how the component will be used**
Sensing and perception subsystem			Sensing and perception components will be used mainly to collect unimodal and multimodal data (mainly audio and video). In addition, some of the components provide resources that allow processing at the edge, e.g., sound event detection.
IFAG	Advanced MEMS microphones	Y	Multi-channel acoustic data acquisition and processing in the edge: • data acquisition through two to eight IM69D130 MEMS microphones, • connected to either Flex evaluation kit or AudioHubNano4D; • data transmission to a processing node via PDM or USB respectively; • possible processing nodes are either Cypress PSoC64 Standard Secure or a RPi.
FBK	SED@Edge	Y	In this use case, the tool is probably not particularly important given that cyclists and pedestrians are targeted. Anyway, detecting traffic-related sound events could be useful to understand the traffic conditions.
GRN	GRNEdge	Y	Component collects synchronised single channel audio and data and streams the data to the fog layer.
UNS	AVDrone	N	AVDrone is not available as a hardware component in the GRN pilot.
AUD	sensMiner	N	SensMiner will not be used in this pilot.
GRN	CATFlow	Y	Component takes road traffic video as input and detects and tracks traffic objects, such as cars, bicycles, trucks, buses, etc.
Security, privacy, and data protection subsystem		Y	The subsystem will ensure the secure transmission of the data from the sensors to the cloud services. In addition, the anonymisation components will minimise the risk of subject re-identification and plate recognition.
FORTH	EdgeSec	M	EdgeSec will encrypt 100% of the traffic from/to the hosts in the whole E2F2C infrastructure and will attest to the integrity of binaries using Intel SGX security features when available.
FBK	VideoAnony	Y	VideoAnony will detect the cyclists and pedestrians and blur their

			faces.
FBK	AudioAnony	Y	Audio anonymisation can be applied but the amount of speech content is probably negligible in this use case.
AUD	VAD (devAIce)	Y	VAD will be applied to detect any segments that contain speech.
Data management and distribution subsystem			The subsystem will be handling massive amounts of data (with various sources, formats and frequencies) dealing with their proper management and distribution.
ITML	Data Fusion Bus (DFB)	Y	DFB will be deployed on the cloud and connect to data-producing components on the edge/fog/cloud layers. These components feed data streams of non-binary data, either collected at the edge or produced by data processing components. All data streams are stored, aggregated, and are available for further filtering, searching, and forwarded to visualisation components.
INTRA	StreamHandler	Y	StreamHandler will be executed on the cloud context and will be interconnected with the Edge/Fog/Cloud layers. The non-binary data will be fed directly to the framework and will be utilised to trigger the visualisation aspects.
ATOS	DatAna	Y	DatAna, as well as other DMD subsystem components, can be used to move/process/transform data in the edge/fog and the cloud.
CNR	Hierarchical Data Distribution (HDD)	Y	HDD, in synergy with the rest of the DMD subsystem components, will be used in order to move data where needed while satisfying the related edge/fog application constraints and requirements (real-time or offline).
Audio, visual, and multimodal AI subsystem		Y	The audio, visual and multimodal AI subsystem integrates data analysis tasks needed for the AI functionalities of the MARVEL framework, enabling AI-based decision-making.
AUD	devAIce	Y	devAIce will be used for audio feature extraction and for detecting relevant audio events.
AU	Visual anomaly detection (ViAD)	Y	Visual anomaly detection will receive as input video frames and provide a label indicating whether they correspond to normal or anomalous situations for the scene under consideration.
AU	Audio-Visual anomaly detection (AVAD)	Y	Audio-Visual anomaly detection will receive as input video frames and synchronised audio snippets and provide a label indicating whether they correspond to normal or anomalous situations for the scene under consideration.
AU	Visual crowd counting (VCC)	Y	Visual crowd counting will receive as input a video frame and provide a number indicating the number of people appearing in them.
AU	Audio-Visual crowd counting (AVCC)	Y	Audio-Visual crowd counting will receive as input video frames and synchronised audio snippet and provide a number indicating the number of people in the scene.
TAU	Automated audio captioning (AAC)	N	The component is not applicable for this use case.
TAU	Sound event detection (SED)	Y	Sound event detection will receive as an input single-channel audio segment and provides a list of detected sound events (label and timestamp) as output. In the GRN1 use case, sound classes are related to motorised micro-mobility modes (bicycles, e-bikes, etc.).
TAU	Sound event localisation	Y	Sound event detection and localisation will receive as an input multi-channel audio segment captured with a microphone array,

	and detection (SELD)		and as output, it provides a list of detected sound events (label and timestamp) and their azimuth and elevation. In the GRN1 use case, sound classes used in the detection are related to motorised micro-mobility modes (bicycles, e-bikes, etc.). SELD is used, if multi-channel audio capturing is available.
TAU	Acoustic scene classification (ASC)	Y	Acoustic scene classification will receive as an input single-channel audio segment and provides a scene class label to describe the overall characteristics of the acoustic scene. In the GRN1 use case, classes are related to traffic modes (e.g. type/level of traffic).
Optimised E2F2C processing and deployment subsystem		Y	The E2F2C subsystem integrates dispersed execution sites into a unified, distributed execution environment, enabling the deployment of services at all layers spanning from Edge to Fog to Cloud/HPC-Centre.
FORTH	GPURegex	Y	GPURegex will provide accelerated pattern matching capabilities in order to contribute to event recognition. In order to be able to provide meaningful results, a preprocessing phase is needed in order to extract event signatures from raw AV data and feed them to our engine.
CNR	DynHP	Y	DynHP will be used to train and compress the DNN model identified for this task such that it can be deployed to fog/edge devices.
UNS	FedL	N	The component cannot be applied due to the lack of matching datasets.
FORTH	Karvdash	Y	Karvdash will provide a dashboard for instantiating services as orchestrated containers, and deployed via appropriate automation to execution sites selected by a dynamic online optimisation strategy.
E2F2C infrastructure		Y	This pilot will build and operate on E2F2C infrastructure; it may take advantage of HPC infrastructure to perform various tasks within the MARVEL project.
PSNC	HPC infrastructure	Y	HPC and cloud infrastructure in this use case can be used for tasks that require high computing power like model training or cloud-based inference (e.g., long-term decision-making). This component will also host the MARVEL Data corpus.
PSNC	HPC management and orchestration	Y	This component will allow for efficient use of the HPC and cloud infrastructure provided by PSNC.
GRN	Cloud tier	Y	GRN will make exclusive use of the cloud services provided by the MARVEL consortium (namely PSNC).
GRN	Fog tier	Y	The GRN fog tier consists of an HPE ProLiant DL385 GEN10 Plus Server with one NVIDIA Tesla T4 16GB installed.
GRN	Edge tier	Y	The GRN edge tier consists of a number (8-10) of audio-video sensors (GRNEdge) that stream AV data to the fog layer.
System outputs: User interactions and the decision-making toolkit		Y	Based on analytical reasoning for medium to long-term business decision-making based on queries execution over the processed audio-visual data, the toolkit can activate workflows for pushing alerts or messages to the systems necessary (boards/apps/etc) depending on thresholds set regarding the traffic/weather conditions/etc. It can also support real-time visualisations of alerts and detected events for short-term decisions and monitoring, supported by a rule-based engine.
ZELUS	SmartViz	Y	In this case, SmartViz could provide a dashboard with an update of the locations for which alerts and communications need to be sent or in general an overview of the areas where appropriate

			action is needed.
STS	Data Corpus-as-a-Service	Y	Data Corpus-as-a-Service can provide a query only to the relative data of the respective use case.

Table 8: MARVEL components in the GRN2 use case

GRN use case 2: Road User Behaviour			
Component owner	**Subsystem /Component**		**Comments on how the component will be used**
Sensing and perception subsystem		Y	Sensing and perception components will be used mainly to collect unimodal and multimodal data (mainly audio and video). In addition, some of the components provide resources that allow processing at the edge, e.g., sound event detection.
IFAG	Advanced MEMS microphones	Y	Multi-channel acoustic data acquisition and processing in the edge: • data acquisition through two to eight IM69D130 MEMS microphones, • connected to either Flex evaluation kit or AudioHubNano4D; • data transmission to a processing node via PDM or USB respectively • possible processing nodes are either Cypress PSoC64 Standard Secure or a RPi.
FBK	SED@Edge	Y	The tool can be used to detect noise events related to driving styles/behaviour of interest (honking, tyre screeching, engine revving,...).
GRN	GRNEdge	Y	Component collects synchronised single-channel audio and data and streams the data to the fog layer.
UNS	AVDrone	N	AVDrone is not available as a hardware component in the GRN pilot.
AUD	sensMiner	N	SensMiner will not be used in this pilot.
GRN	CATFlow	Y	Component takes road traffic video as input and detects and tracks traffic objects, such as cars, bicycles, trucks, buses, etc.
Security, privacy, and data protection subsystem		Y	The subsystem will ensure the secure transmission of the data from the sensors to the cloud services. In addition, the anonymisation components will minimise the misuse of the data to identify subjects. Plates are particularly relevant.
FORTH	EdgeSec	M	EdgeSec will encrypt 100% of the traffic from/to the hosts in the whole E2F2C infrastructure and will attest to the integrity of binaries using Intel SGX security features when available.
FBK	VideoAnony	Y	In case the cameras are installed for covering the roads, VideoAnony will detect the cyclists and pedestrians that appear in the scene and blur their faces.
FBK	AudioAnony	Y	Audio anonymisation can be applied but the amount of speech content is probably negligible in this use case
AUD	VAD (devAIce)	Y	VAD will be used to detect signals that contain speech.
Data management and distribution subsystem		Y	The subsystem will be handling massive amounts of data (with various sources, formats, and frequencies) dealing with their proper management and distribution.

ITML	Data Fusion Bus (DFB)	Y	DFB will be deployed on the cloud and connect to data-producing components on the edge/fog/cloud layers. These components feed data streams of non-binary data, either collected at the edge or produced by data processing components. All data streams are stored, aggregated and are available for further filtering, searching and forwarded to visualisation components.
INTRA	StreamHandler	Y	StreamHandler will be executed on the cloud context and will be interconnected with the Edge/Fog/Cloud layers. The non-binary data will be fed directly to the framework and will be utilised to trigger the visualisation aspects.
ATOS	DatAna	Y	DatAna can be deployed both at the edge/fog or on the cloud, perform some processing over the non-binary data, and connect with the other DMD subsystem components for further aggregation and storage.
CNR	Hierarchical Data Distribution (HDD)	Y	HDD, in synergy with the rest of the DMD subsystem components, will be used in order to move data where needed while satisfying the related edge/fog application constraints and requirements (real-time or offline).
Audio, visual, and multimodal AI subsystem		Y	The audio, visual and multimodal AI subsystem integrates data analysis tasks needed for the AI functionalities of the MARVEL framework, enabling AI-based decision-making.
AUD	devAIce	Y	devAIce will be used for audio feature extraction and for detecting relevant audio events.
AU	Visual anomaly detection (ViAD)	Y	Visual anomaly detection will receive as input video frames and provide a label indicating whether they correspond to normal or anomalous situations for the scene under consideration.
AU	Audio-Visual anomaly detection (AVAD)	Y	Audio-Visual anomaly detection will receive as input video frames and synchronised audio snippets and provide a label indicating whether they correspond to normal or anomalous situations for the scene under consideration.
AU	Visual crowd counting (VCC)	N	As this use case focuses on drive behaviour in locations of low appearance of people, crowd counting is not of interest.
AU	Audio-Visual crowd counting (AVCC)	N	As this use case focuses on drive behaviour in locations of low appearance of people, crowd counting is not of interest.
TAU	Automated audio captioning (AAC)	Y	The audio captioning will receive as an input single-channel audio segment and provides a textual description of the content. AAC can be used if audio learning examples are provided with a full textual description.
TAU	Sound event detection (SED)	Y	Sound event detection will receive as an input single-channel audio segment and provides a list of detected sound events (label and timestamp) as output. In the GRN2 use case, sound classes are related to traffic sounds indicating anomalous behaviour (e.g., vehicle horns, bicycle bells, excessive speed, etc.).
TAU	Sound event localisation and detection (SELD)	Y	Sound event detection and localisation will receive as an input multi-channel audio segment captured with a microphone array, and as output it provides a list of detected sound events (label and timestamp) and their azimuth and elevation. In the GRN2 use case, sound classes are related to traffic sounds indicating anomalous behaviour (e.g., vehicle horns, bicycle bells, excessive speed, etc.). SELD is used, if multi-channel audio capturing is available.
TAU	Acoustic scene	Y	Acoustic scene classification will receive as an input single-channel audio segment and provides a scene class label to describe

	classification (ASC)		the overall characteristics of the acoustic scene. In the GRN2 use case, classes are related to traffic modes (e.g., general type/level of traffic).
Optimised E2F2C processing and deployment subsystem			The E2F2C subsystem integrates dispersed execution sites into a unified, distributed execution environment, enabling the deployment of services at all layers spanning from Edge to Fog to Cloud/HPC-Centre.
FORTH	GPURegex	Y	GPURegex will provide accelerated pattern matching capabilities in order to contribute to event recognition. In order to be able to provide meaningful results, a pre-processing phase is needed in order to extract event signatures from raw AV data and feed them to our engine.
CNR	DynHP	Y	DynHP will be used to train and compress the DNN model identified for this task such that it can be deployed to fog/edge devices.
UNS	FedL	N	The component cannot be applied due to the lack of matching datasets.
FORTH	Karvdash	Y	Karvdash will provide a dashboard for instantiating services as orchestrated containers, and deployed via appropriate automation to execution sites selected by a dynamic online optimisation strategy.
E2F2C infrastructure		Y	This pilot will build and operate on E2F2C infrastructure; it may take advantage of HPC infrastructure to perform various tasks within the MARVEL project.
PSNC	HPC infrastructure	Y	HPC and cloud infrastructure in this use case can be used for tasks that require high computing power like model training or cloud-based inference (e.g., long-term decision-making). This component will also host the MARVEL Data corpus.
PSNC	HPC management and orchestration	Y	This component will allow for efficient use of the HPC and cloud infrastructure provided by PSNC.
GRN	Cloud tier	Y	GRN will make exclusive use of the cloud services provided by the MARVEL consortium.
GRN	Fog tier	Y	The GRN fog tier consists of an HPE ProLiant DL385 GEN10 Plus Server with one NVIDIA Tesla T4 16GB installed.
GRN	Edge tier	Y	The GRN edge tier consists of a number (8-10) of audio-video sensors (GRNEdge) that stream AV data to the fog layer.
System outputs: User interactions and the decision-making toolkit		Y	Data presentation and advanced visualisations that reveal hidden insights of valuable knowledge and multisource, multimodal summaries, that allow users to explore and understand audio-visual, sensor, and other context-enriching data (e.g., weather data, information from incident reporting systems, parking sensors, etc.) and interact with them.
ZELUS	SmartViz	Y	SmartViz can visualise historical data across time or related to a particular spot of interest or juxtaposition with other information like weather data, past education campaigns, bad examples of planning, accidents, etc. in order to identify patterns, hidden relationships, and areas for further investigation.
STS	Data Corpus-as-a-Service	Y	Data Corpus-as-a-Service can provide a query only to the relative data of the respective use case.

Table 9: MARVEL components in the GRN3 use case

GRN use case 3: Traffic Conditions and Anomalous Events			
Component owner	**Subsystem /Component**		**Comments on how the component will be used**
Sensing and perception subsystem		Y	Sensing and perception components will be used mainly to collect unimodal and multimodal data (mainly audio and video). In addition, some of the components provide resources that allow processing at the edge, e.g., sound event detection.
IFAG	Advanced MEMS microphones	Y	Multi-channel acoustic data acquisition and processing in the edge: • data acquisition through two to eight IM69D130 MEMS microphones, • connected to either Flex evaluation kit or AudioHubNano4D; • data transmission to a processing node via PDM or USB respectively • possible processing nodes are either Cypress PSoC64 Standard Secure or a RPi.
FBK	SED@Edge	Y	The tool can be used to detect noise events related to anomalous traffic events (e.g., repeated honking and long-lasting engine idling). This tool does not detect anomalies with respect to the training data but it detects a list of hand-crafted sound events.
GRN	GRNEdge	Y	Component collects synchronised single-channel audio and data and streams the data to the fog layer.
UNS	AVDrone	N	AVDrone is not available as a hardware component in the GRN pilot.
AUD	sensMiner	N	sensMiner will not be used in this pilot.
GRN	CATFlow	Y	Component takes road traffic video as input and detects and tracks traffic objects, such as cars, bicycles, trucks, busses, etc.
Security, privacy, and data protection subsystem		Y	The subsystem will ensure the secure transmission of the data from the sensors to the cloud services. In addition, the anonymisation components will minimise the risk of person and plate recognition.
FORTH	EdgeSec	M	EdgeSec will encrypt 100% of the traffic from/to the hosts in the whole E2F2C infrastructure and will attest to the integrity of binaries using Intel SGX security features when available.
FBK	VideoAnony	Y	VideoAnony will detect the cyclists and pedestrians that appear in the scene and blur their faces.
FBK	AudioAnony	Y	Audio anonymisation can be applied but the amount of speech content is probably negligible in this use case
AUD	VAD (devAIce)	Y	VAD will be used to detect speech segments.
Data management and distribution subsystem		Y	The subsystem will be handling massive amounts of data (with various sources, formats and frequencies) dealing with their proper management and distribution.
ITML	Data Fusion Bus (DFB)	Y	DFB will be deployed on the cloud and connect to data-producing components on the edge/fog/cloud layers. These components feed data streams of non-binary data, either collected at the edge or produced by data processing components. All data streams are stored, aggregated and are available for further filtering, searching

			and forwarded to visualisation components.
INTRA	StreamHandler	Y	StreamHandler will be executed on the cloud context and will be interconnected with the Edge/Fog/Cloud layers. The non-binary data will be fed directly to the framework and will be utilised to trigger the visualisation aspects.
ATOS	DatAna	Y	DatAna can be deployed both at the edge/fog or on the cloud, perform some processing over the non-binary data, and connect with the other DMD subsystem components for further aggregation and storage.
CNR	Hierarchical Data Distribution (HDD)	Y	HDD, in synergy with the rest of the DMD subsystem components, will be used in order to move data where needed while satisfying the related edge/fog application constraints and requirements (real-time or offline).
Audio, visual, and multimodal AI subsystem		Y	The audio, visual and multimodal AI subsystem integrates data analysis tasks needed for the AI functionalities of the MARVEL framework, enabling AI-based decision-making.
AUD	devAIce	Y	devAIce will be used for audio feature extraction and for detecting relevant audio events.
AU	Visual anomaly detection (ViAD)	Y	Visual anomaly detection will receive as input video frames and provide a label indicating whether they correspond to normal or anomalous situations for the scene under consideration.
AU	Audio-Visual anomaly detection (AVAD)	Y	Audio-Visual anomaly detection will receive as input video frames and synchronised audio snippets and provide a label indicating whether they correspond to normal or anomalous situations for the scene under consideration.
AU	Visual crowd counting (VCC)	Y	Visual crowd counting will receive as input a video frame and provide a number indicating the number of people appearing in them.
AU	Audio-Visual crowd counting (AVCC)	Y	Audio-Visual crowd counting will receive as input video frames and synchronised audio snippet and provide a number indicating the number of people in the scene.
TAU	Automated audio captioning (AAC)	N	The component is not applicable for this use case.
TAU	Sound event detection (SED)	Y	Sound event detection will receive as an input single-channel audio segment and provides a list of detected sound events (label and timestamp) as output. In the GRN3 use case, sound classes are related to specific sounds that can be regarded as anomalous for the scene (e.g., anomalous traffic queues and low traffic speeds).
TAU	Sound event localisation and detection (SELD)	Y	Sound event detection and localisation will receive as an input multi-channel audio segment captured with a microphone array, and as output it provides a list of detected sound events (label and timestamp) and their azimuth and elevation. In the GRN3 use case, sound classes are related to specific sounds that can be regarded as anomalous for the scene (e.g., anomalous traffic queues and low traffic speeds). SELD is used, if multi-channel audio capturing is available.
TAU	Acoustic scene classification (ASC)	Y	Acoustic scene classification will receive as an input single-channel audio segment and provides a scene class label to describe the overall characteristics of the acoustic scene. In the GRN3 use case, classes are related to traffic modes (e.g., type/level of traffic).
Optimised E2F2C processing and		Y	The E2F2C subsystem integrates dispersed execution sites into a

deployment subsystem			unified, distributed execution environment, enabling the deployment of services at all layers spanning from Edge to Fog to Cloud/HPC-Centre.
FORTH	GPURegex	Y	GPURegex will provide accelerated pattern matching capabilities in order to contribute to event recognition. In order to be able to provide meaningful results, a pre-processing phase is needed in order to extract event signatures from raw AV data and feed them to our engine.
CNR	DynHP	Y	DynHP will be used to train and compress the DNN model identified for this task such that it can be deployed to fog/edge devices.
UNS	FedL	M	FedL can be used across any two or more datasets (use cases) that share the same feature space, or to a single dataset by splitting it into partitions.
FORTH	Karvdash	Y	Karvdash will provide a dashboard for instantiating services as orchestrated containers, and deployed via appropriate automation to execution sites selected by a dynamic online optimisation strategy.
E2F2C infrastructure		Y	This pilot will build and operate on E2F2C infrastructure; it may take advantage of HPC infrastructure to perform various tasks within the MARVEL project.
PSNC	HPC infrastructure	Y	HPC and cloud infrastructure in this use case can be used for tasks that require high computing power like model training or cloud-based inference (e.g., long-term decision-making). This component will also host the MARVEL Data corpus.
PSNC	HPC management and orchestration	Y	This component will allow for efficient use of the HPC and cloud infrastructure provided by PSNC.
GRN	Cloud tier	Y	GRN will make exclusive use of the cloud services provided by the MARVEL consortium.
GRN	Fog tier	Y	The GRN fog tier consists of an HPE ProLiant DL385 GEN10 Plus Server with one NVIDIA Tesla T4 16GB installed.
GRN	Edge tier	Y	The GRN edge tier consists of a number (8-10) of audio-video sensors (GRNEdge) that stream AV data to the fog layer.
System outputs: User interactions and the decision-making toolkit		Y	Text-annotated attention maps which will enhance video streams with textual information and indications of associated audio events;
ZELUS	SmartViz	Y	SmartViz can visualise traffic in heat maps and indicate any accidents.
STS	Data Corpus-as-a-Service	Y	Data Corpus-as-a-Service can provide a query only to the relative data of the respective use case.

Table 10: MARVEL components in the GRN4 use case

GRN use case 4: Junction Traffic Trajectory Collection			
Component owner	**Subsystem/Component**		**Comments on how the component will be used**
Sensing and perception subsystem		Y	Sensing and perception components will be used mainly to collect unimodal and multimodal data (mainly audio and video). In addition, some of the components provide resources that allow

			processing at the edge, e.g., sound event detection.
IFAG	Advanced MEMS microphones	Y	Multi-channel acoustic data acquisition and processing in the edge: • data acquisition through two to eight IM69D130 MEMS microphones, • connected to either Flex evaluation kit or AudioHubNano4D; • data transmission to a processing node via PDM or USB respectively • possible processing nodes are either Cypress PSoC64 Standard Secure or a RPi.
FBK	SED@Edge	Y	In this use case, the tool is probably not very relevant. It can be used to detect hand-crafted sound events related to traffic conditions for long-term analytics.
GRN	GRNEdge	Y	Component collects synchronised single channel audio and data and streams the data to the fog layer.
UNS	AVDrone	N	AVDrone is not available as a hardware component in the GRN pilot.
AUD	sensMiner	N	sensMiner will not be used in this pilot.
GRN	CATFlow	Y	Component takes road traffic video as input and detects and tracks traffic objects, such as cars, bicycles, trucks, buses, etc.
Security, privacy, and data protection subsystem		Y	The subsystem will ensure the secure transmission of the data from the sensors to the cloud services. In addition, the anonymisation components will minimise the risk of person and plate recognition.
FORTH	EdgeSec	M	EdgeSec will encrypt 100% of the traffic from/to the hosts in the whole E2F2C infrastructure and will attest to the integrity of binaries using Intel SGX security features when available.
FBK	VideoAnony	Y	VideoAnony will detect the cyclists and pedestrians that appear in the scene and blur their faces.
FBK	AudioAnony	Y	Audio anonymisation can be applied but the amount of speech content is probably negligible in this use case
AUD	VAD (devAIce)	Y	VAD will be used to detect speech segments.
Data management and distribution subsystem			The subsystem will be handling massive amounts of data (with various sources, formats, and frequencies) dealing with their proper management and distribution.
ITML	Data Fusion Bus (DFB)	Y	DFB will be deployed on the cloud and connect to data-producing components on the edge/fog/cloud layers. These components feed data streams of non-binary data, either collected at the edge or produced by data processing components. All data streams are stored, aggregated and are available for further filtering, searching and forwarded to visualisation components.
INTRA	StreamHandler	Y	StreamHandler will be executed on the cloud context and will be interconnected with the Edge/Fog/Cloud layers. The non-binary data will be fed directly to the framework and will be utilised to trigger the visualisation aspects.
ATOS	DatAna	Y	DatAna can be deployed both at the edge/fog or on the cloud, perform some processing over the non-binary data, and connect with the other DMD subsystem components for further aggregation and storage.

CNR	Hierarchical Data Distribution (HDD)	Y	HDD, in synergy with the rest of the DMD subsystem components, will be used in order to move data where needed while satisfying the related edge/fog application constraints and requirements (real-time or offline).
Audio, visual, and multimodal AI subsystem		Y	The audio, visual and multimodal AI subsystem integrates data analysis tasks needed for the AI functionalities of the MARVEL framework, enabling AI-based decision-making.
AUD	devAIce	Y	devAIce will be used for audio feature extraction and for detecting relevant audio events.
AU	Visual anomaly detection (ViAD)	N	As this use case focuses on accumulating results of driving trajectories, visual anomaly detection is not of interest.
AU	Audio-Visual anomaly detection (AVAD)	N	As this use case focuses on accumulating results of driving trajectories, audio-visual anomaly detection is not of interest.
AU	Visual crowd counting (VCC)	N	As this use case focuses on accumulating results of driving trajectories, audio-visual anomaly detection is not of interest.
AU	Audio-Visual crowd counting (AVCC)	N	As this use case focuses on accumulating results of driving trajectories, audio-visual anomaly detection is not of interest.
TAU	Automated audio captioning (AAC)	N	The component is not applicable for this use case.
TAU	Sound event detection (SED)	Y	Sound event detection will receive as an input single-channel audio segment and provides a list of detected sound events (label and timestamp) as output. In the GRN4 use case, sound classes are related to traffic entities (e.g., vehicles, pedestrians).
TAU	Sound event localisation and detection (SELD)	N	The component is not applicable for this use case.
TAU	Acoustic scene classification	Y	Acoustic scene classification will receive as an input single-channel audio segment and provides a scene class label to describe the overall characteristics of the acoustic scene. In the GRN4 use case, classes are related to traffic modes (e.g., type/level of traffic).
Optimised E2F2C processing and deployment subsystem		Y	The E2F2C subsystem integrates dispersed execution sites into a unified, distributed execution environment, enabling the deployment of services at all layers spanning from Edge to Fog to Cloud/HPC-Centre.
FORTH	GPURegex	N	The component is not applicable to this use case.
CNR	DynHP	Y	DynHP will be used to train and compress the DNN model identified for this task such that it can be deployed to fog/edge devices
UNS	FedL	M	FedL can be used across any two or more datasets (use cases) that share the same feature space, or to a single dataset by splitting it into partitions.
FORTH	Karvdash	Y	Karvdash will provide a dashboard for instantiating services as orchestrated containers, and deployed via appropriate automation to execution sites selected by a dynamic online optimisation strategy.
E2F2C infrastructure		Y	This pilot will build and operate on E2F2C infrastructure; it may take advantage of HPC infrastructure to perform various tasks within the MARVEL project.
PSNC	HPC infrastructure	Y	HPC and cloud infrastructure in this use case can be used for tasks that require high computing power like model training or cloud-

			based inference (e.g., long-term decision-making). This component will also host the MARVEL Data corpus.
PSNC	HPC management and orchestration	Y	This component will allow for efficient use of the HPC and cloud infrastructure provided by PSNC.
GRN	Cloud tier	Y	GRN will make exclusive use of the cloud services provided by the MARVEL consortium.
GRN	Fog tier	Y	The GRN fog tier consists of an HPE ProLiant DL385 GEN10 Plus Server with one NVIDIA Tesla T4 16GB installed.
GRN	Edge tier	Y	The GRN edge tier consists of a number (8-10) of audio-video sensors (GRNEdge) that stream AV data to the fog layer.
System outputs: User interactions and the decision-making toolkit		Y	Text-annotated attention maps will enhance video streams with textual information and indications of associated audio events.
ZELUS	SmartViz	Y	SmartViz can provide predefined pieces of maps where the bicycle or car movement can be depicted.
STS	Data Corpus-as-a-Service	Y	Data Corpus-as-a-Service can provide a query only to the relative data of the respective use case.

Table 11: MARVEL components in the MT1 use case

MT use case 1: Monitoring of Crowded Areas			
Component owner	**Subsystem /Component**		**Comments on how the component will be used**
Sensing and perception subsystem		Y	Sensing and perception components will be used mainly to collect unimodal and multimodal data (mainly audio and video). In addition, some of the components provide resources that allow processing at the edge, e.g., sound event detection.
IFAG	Advanced MEMS microphones	M	In this use case, no microphones are foreseen for real-life data collection. The use case will consider small-scale experimental installation of microphones. Multi-channel acoustic data acquisition and processing in the edge: • data acquisition through two to eight IM69D130 MEMS microphones, • connected to either Flex evaluation kit or AudioHubNano4D; • data transmission to a processing node via PDM or USB respectively; • possible processing nodes are either Cypress PSoC64 Standard Secure or a RPi.
FBK	SED@Edge	M	The tool can be used to detect relevant sound events as long as the audio is available. In this use case, no microphones are foreseen for real-life data collection. In case audio signals are recorded for some data related to this use case, the application of SED@Edge will be explored for the respective dataset.
GRN	GRNEdge	N	GRNEdge is not available as a hardware component in the MT pilot.
UNS	AVDrone	N	AVDrone is not available as a hardware component in the MT pilot.
AUD	sensMiner	N	sensMiner will not be used in this pilot.

GRN	CATFlow	N	CATFlow is not required for this MT use case.
Security, privacy, and data protection subsystem		Y	The subsystem will ensure the secure transmission of the data from the sensors to the cloud services. In addition, the anonymisation components will minimise the misuse of the data to identify subjects. Audio may be not always available. For some video recordings, anonymisation may not be necessary.
FORTH	EdgeSec	Y	EdgeSec will encrypt 100% of the traffic from/to the hosts in the whole E2F2C infrastructure and will attest to the integrity of binaries using Intel SGX security features when available.
FBK	VideoAnony	Y	VideoAnony will detect the cyclists and pedestrians that appear in the scene and blur their faces. The necessity of anonymisation depends on the height of the cameras.
FBK	AudioAnony	M	Audio anonymisation can be applied (in different forms) as long as the audio is available and the sound is intelligible. In this use case, no microphones are foreseen for real-life data collection. In case audio signals are recorded for some data related to this use case, the application of AudioAnony will be explored for the respective dataset.
AUD	VAD (devAIce)	M	In this use case, no microphones are foreseen for real-life data collection. In case audio signals are recorded for some data related to this use case, the application of VAD (devAIce) will be explored for the respective dataset.
Data management and distribution subsystem		Y	The subsystem will be handling massive amounts of data (with various sources, formats, and frequencies) dealing with their proper management and distribution.
ITML	Data Fusion Bus (DFB)	Y	DFB will be deployed on the cloud and connect to data-producing components on the edge/fog/cloud layers. These components feed data streams of non-binary data, either collected at the edge or produced by data processing components. All data streams are stored, aggregated and are available for further filtering, searching and forwarded to visualisation components.
INTRA	StreamHandler	Y	StreamHandler will be executed on the cloud context and will be interconnected with the Edge/Fog/Cloud layers. The non-binary data will be fed directly to the framework and will be utilised to trigger the visualisation aspects.
ATOS	DatAna	Y	DatAna can be deployed both at the edge/fog or on the cloud, perform some processing over the non-binary data, and connect with the other DMD subsystem components for further aggregation and storage.
CNR	Hierarchical Data Distribution (HDD)	Y	HDD, in synergy with the rest of the DMD subsystem components, will be used in order to move data where needed while satisfying the related edge/fog application constraints and requirements (real-time or offline).
Audio, visual, and multimodal AI subsystem		Y	The audio, visual and multimodal AI subsystem integrates data analysis tasks needed for the AI functionalities of the MARVEL framework, enabling AI-based decision-making.
AUD	devAIce	M	No microphones are foreseen in this use case.
AU	Visual anomaly detection (ViAD)	Y	Visual anomaly detection will receive as input video frames and provide a label indicating whether they correspond to normal or anomalous situations for the scene under consideration.
AU	Audio-Visual anomaly detection (AVAD)	M	In this use case, no microphones are foreseen for real-life data collection. In case audio signals are recorded for some data related

			to this use case, the application of AVAD will be explored for the respective dataset.
AU	Visual crowd counting (VCC)	Y	Visual crowd counting will receive as input a video frame and provide a number indicating the number of people appearing in them.
AU	Audio-Visual crowd counting (AVCC)	M	In this use case, no microphones are foreseen for real-life data collection. In case audio signals are recorded for some data related to this use case, the application of AVAD will be explored for the respective dataset.
TAU	Automated audio captioning (AAC)	M	The audio captioning will receive as an input single-channel audio segment and provides a textual description of the content. AAC can be used, if audio learning examples are provided with a full textual description and audio signals are available.
TAU	Sound event detection (SED)	N	The component is not applicable for this use case.
TAU	Sound event localisation and detection (SELD)	N	The component is not applicable for this use case.
TAU	Acoustic scene classification (ASC)	M	Acoustic scene classification will receive as an input single-channel audio segment and provides a scene class label to describe the overall characteristics of the acoustic scene. In the MT1 use case, classes are related to the overall ambience created by different crowd sizes and types. ASC can be used if audio signals are available in the use case.
Optimised E2F2C processing and deployment subsystem		Y	The E2F2C subsystem integrates dispersed execution sites into a unified, distributed execution environment, enabling the deployment of services at all layers spanning from Edge to Fog to Cloud/HPC-Centre.
FORTH	GPURegex	M	GPURegex will provide accelerated pattern matching capabilities in order to contribute to event recognition. In order to be able to provide meaningful results, a pre-processing phase is needed in order to extract event signatures from raw AV data and feed them to our engine.
CNR	DynHP	Y	DynHP will be used to train and compress the DNN model identified for this task such that it can be deployed to fog/edge devices.
UNS	FedL	Y	FedL will be implemented across the datasets from the UNS1 use case Drone experiment and the MT1 use case Monitoring of Crowded Areas.
FORTH	Karvdash	Y	Karvdash will provide a dashboard for instantiating services as orchestrated containers, and deployed via appropriate automation to execution sites selected by a dynamic online optimisation strategy.
E2F2C infrastructure		Y	This pilot will build and operate on E2F2C infrastructure; it may take advantage of HPC infrastructure to perform various tasks within the MARVEL project.
PSNC	HPC infrastructure	Y	HPC and cloud infrastructure in this use case can be used for tasks that require high computing power like model training or cloud-based inference (e.g., long-term decision-making). This component will also host the MARVEL Data corpus.
PSNC	HPC management and orchestration	Y	This component will allow for efficient use of the HPC and cloud infrastructure provided by PSNC.

MT	Cloud tier	Y	MT will make use of the cloud tier provided by the MARVEL consortium.
MT	Fog tier	Y	The fog tier will consist of a DELL workstation located at FBK premises, that, via secure connection captures the raw data from the sensors. The server may be equipped with an NVIDIA GeForce RTX 3080.
MT	Edge tier	Y	The edge consists of a series of Raspberry Pis installed in the nearest cabinet where selected cameras are installed. It is expected that at the end of the experimentation that some of some submodules can be installed and executed near the cameras and MEMS microphones.
System outputs: User interactions and the decision-making toolkit		Y	Text-annotated attention maps which will enhance video streams with textual information and indications of associated audio events and multisource, multimodal summaries, that allow users to explore and understand audio-visual, sensor, and other context-enriching data (e.g., weather data, information from incident reporting systems, parking sensors, etc.) and interact with them.
ZELUS	SmartViz	Y	SmartViz can visualise data across time or in juxtaposition with other information like weather data, activities happening in the location, etc. in order to identify patterns and hidden relationships. It can also try to depict trajectories of crowd movement depending on classifications.
STS	Data Corpus-as-a-Service	Y	Data Corpus-as-a-Service can provide a query only to the relative data of the respective use case.

Table 12: MARVEL components in the MT2 use case

MT use case 2: Detecting Criminal and Anti-Social Behaviours			
Component owner	**Subsystem/Component**		**Comments on how the component will be used**
Sensing and perception subsystem		Y	Sensing and perception components will be used mainly to collect unimodal and multimodal data (mainly audio and video). In addition, some of the components provide resources that allow processing at the edge, e.g., sound event detection.
IFAG	Advanced MEMS microphones	Y	Multi-channel acoustic data acquisition and processing in the edge: • data acquisition through two to eight IM69D130 MEMS microphones, • connected to either Flex evaluation kit or AudioHubNano4D; • data transmission to a processing node via PDM or USB respectively; • possible processing nodes are either Cypress PSoC64 Standard Secure or a RPi.
FBK	SED@Edge	Y	The tool can be used to detect the presence of a list of predefined events which may be used to trigger alarms in the control room (screams, glass breaking, engine noises, gunshots, ...).
GRN	GRNEdge	N	GRNEdge is not available as a hardware component in the MT pilot.
UNS	AVDrone	N	AVDrone is not available as a hardware component in the MT pilot.

AUD	sensMiner	N	sensMiner will not be used in this pilot.
GRN	CATFlow	N	CATFlow is not required for this MT use case.
Security, privacy, and data protection subsystem		Y	The subsystem will ensure the secure transmission of the data from the sensors to the cloud services. In addition, the anonymisation components will minimise the risk of person recognition. Audio may be not always available.
FORTH	EdgeSec	Y	EdgeSec will encrypt 100% of the traffic from/to the hosts in the whole E2F2C infrastructure and will attest to the integrity of binaries using Intel SGX security features when available.
FBK	VideoAnony	Y	VideoAnony will detect the cyclists and pedestrians that appear in the scene and blur their faces. The necessity of anonymisation depends on the height of the cameras.
FBK	AudioAnony	Y	Audio anonymisation can be applied to remove the identity of the speakers from speech signals.
AUD	VAD (devAIce)	Y	VAD will be used to detect speech segments.
Data management and distribution subsystem		Y	The subsystem will be handling massive amounts of data (with various sources, formats, and frequencies) dealing with their proper management and distribution.
ITML	Data Fusion Bus (DFB)	Y	DFB will be deployed on the cloud and connect to data-producing components on the edge/fog/cloud layers. These components feed data streams of non-binary data, either collected at the edge or produced by data processing components. All data streams are stored, aggregated and are available for further filtering, searching and forwarded to visualisation components.
INTRA	StreamHandler	Y	StreamHandler will be executed on the cloud context and will be interconnected with the Edge/Fog/Cloud layers. The non-binary data will be fed directly to the framework and will be utilised to trigger the visualisation aspects.
ATOS	DatAna	Y	DatAna can be deployed both at the edge/fog or on the cloud, perform some processing over the non-binary data, and connect with the other DMD subsystem components for further aggregation and storage.
CNR	Hierarchical Data Distribution (HDD)	Y	HDD, in synergy with the rest of the DMD subsystem components, will be used in order to move data where needed while satisfying the related edge/fog application constraints and requirements (real-time or offline).
Audio, visual, and multimodal AI subsystem		Y	The audio, visual and multimodal AI subsystem integrates data analysis tasks needed for the AI functionalities of the MARVEL framework, enabling AI-based decision-making.
AUD	devAIce	Y	devAIce will be used for feature extraction.
AU	Visual anomaly detection (ViAD)	Y	Visual anomaly detection will receive as input video frames and provide a label indicating whether they correspond to normal or anomalous situations for the scene under consideration.
AU	Audio-Visual anomaly detection (AVAD)	Y	Audio-Visual anomaly detection will receive as input video frames and synchronised audio snippets and provide a label indicating whether they correspond to normal or anomalous situations for the scene under consideration.
AU	Visual crowd counting (VCC)	Y	Visual crowd counting will receive as input a video frame and provide a number indicating the number of people appearing in them.

AU	Audio-Visual crowd counting (AVCC)	Y	Audio-Visual crowd counting will receive as input video frames and synchronised audio snippet and provide a number indicating the number of people in the scene.
TAU	Automated audio captioning (AAC)	Y	The audio captioning will receive as an input single-channel audio segment and provides a textual description of the content. AAC can be used, if learning examples are provided with a full textual description and audio signals are available.
TAU	Sound event detection (SED)	Y	Sound event detection will receive as an input single-channel audio segment and provides a list of detected sound events (label and timestamp) as output. In the MT2 use case, sound classes are related to aggression and robberies. SED can be used if audio signals are available in the use case.
TAU	Sound event localisation and detection (SELD)	Y	Sound event detection and localisation will receive as an input multi-channel audio segment captured with a microphone array, and as output it provides a list of detected sound events (label and timestamp) and their azimuth and elevation. In the MT2 use case, sound classes are related to aggression and robberies. SELD is used if multi-channel audio capturing is available.
TAU	Acoustic scene classification (ASC)	Y	Acoustic scene classification will receive as an input single-channel audio segment and provides a scene class label to describe the overall characteristics of the acoustic scene. In the MT2 use case, classes are related to the overall ambience created by different crowd sizes, crowd actions, and types of crowds. ASC can be used if audio signals are available in the use case.
Optimised E2F2C processing and deployment subsystem		Y	The E2F2C subsystem integrates dispersed execution sites into a unified, distributed execution environment, enabling the deployment of services at all layers spanning from Edge to Fog to Cloud/HPC-Centre.
FORTH	GPURegex	Y	GPURegex will provide accelerated pattern matching capabilities in order to contribute to event recognition. In order to be able to provide meaningful results, a pre-processing phase is needed in order to extract event signatures from raw AV data and feed them to our engine.
CNR	DynHP	Y	DynHP will be used to train and compress the DNN model identified for this task such that it can be deployed to fog/edge devices.
UNS	FedL	M	FedL can be used across any two or more datasets (use cases) that share the same feature space, or to a single dataset by splitting it into partitions.
FORTH	Karvdash	Y	Karvdash will provide a dashboard for instantiating services as orchestrated containers, and deployed via appropriate automation to execution sites selected by a dynamic online optimisation strategy.
E2F2C infrastructure		Y	This pilot will build and operate on E2F2C infrastructure; it may take advantage of HPC infrastructure to perform various tasks within the MARVEL project.
PSNC	HPC infrastructure	Y	HPC and cloud infrastructure in this use case can be used for tasks that require high computing power like model training or cloud-based inference (e.g., long-term decision-making). This component will also host the MARVEL Data corpus.
PSNC	HPC management and orchestration	Y	This component will allow for efficient use of the HPC and cloud infrastructure provided by PSNC.

MT	Cloud tier	Y	MT will make use of the cloud tier provided by the MARVEL consortium.
MT	Fog tier	Y	The fog tier will consist of a DELL workstation located at FBK premises, that, via secure connection captures the raw data from the sensors. The server may be equipped with an NVIDIA GeForce RTX 3080.
MT	Edge tier	Y	The edge consists of a series of Raspberry Pi installed in the nearest cabinet where selected cameras are installed. It is expected that at the end of the experimentation that some of some submodules can be installed and executed near the cameras and MEMS microphones.
System outputs: User interactions and the decision-making toolkit		Y	Based on analytical reasoning for medium to long-term business decision-making based on queries execution over the processed audio-visual data, the toolkit can activate workflows for pushing alerts or messages to the intervention teams, depending on thresholds set regarding what is considered to be classified as illegal or dangerous behaviour. It can also support real-time visualisations of alerts and detected events for short-term decisions and monitoring, supported by a rule-based engine.
ZELUS	SmartViz	Y	In this case, SmartViz can provide a dashboard with an update of the spots for which alerts and communications need to be sent or in general an overview of the areas where appropriate action is needed. A heatmap of emotions will also be considered.
STS	Data Corpus-as-a-Service	Y	Data Corpus-as-a-Service can provide a query only to the relative data of the respective use case.

Table 13: MARVEL components in the MT3 use case

MT use case 3: Monitoring of Parking Places			
Component owner	**Subsystem /Component**		**Comments on how the component will be used**
Sensing and perception subsystem		Y	Sensing and perception components will be used mainly to collect unimodal and multimodal data (mainly audio and video). In addition, some of the components provide resources that allow processing at the edge, e.g., sound event detection.
IFAG	Advanced MEMS microphones	Y	Multi-channel acoustic data acquisition and processing in the edge: • data acquisition through two to eight IM69D130 MEMS microphones, • connected to either Flex evaluation kit or AudioHubNano4D; • data transmission to a processing node via PDM or USB respectively; • possible processing nodes are either Cypress PSoC64 Standard Secure or a RPi.
FBK	SED@Edge	Y	The tool can be used to detect sound events of interest in the parking place. The list of events has to be defined in advance.
GRN	GRNEdge	N	GRNEdge is not available as a hardware component in the MT pilot.
UNS	AVDrone	N	AVDrone is not available as a hardware component in the MT pilot.

AUD	sensMiner	N	sensMiner will not be used in this pilot.
GRN	CATFlow	N	CATFlow is not required for this MT use case.
Security, privacy, and data protection subsystem		Y	The subsystem will ensure the secure transmission of the data from the sensors to the cloud services. In addition, the anonymisation components will minimise the misuse of the data to identify subjects, audio may not be always available. For some video recordings, anonymisation may not be necessary.
FORTH	EdgeSec	Y	EdgeSec will encrypt 100% of the traffic from/to the hosts in the whole E2F2C infrastructure and will attest to the integrity of binaries using Intel SGX security features when available.
FBK	VideoAnony	Y	VideoAnony will detect the cyclists and pedestrians that appear in the scene and blur their faces.
FBK	AudioAnony	Y	Audio anonymisation can be applied, although speech is not really likely to occur
AUD	VAD (devAIce)	Y	VAD will be used to detect speech segments.
Data management and distribution subsystem		Y	The subsystem will be handling massive amounts of data (with various sources, formats, and frequencies) dealing with their proper management and distribution.
ITML	Data Fusion Bus (DFB)	Y	DFB will be deployed on the cloud and connect to data-producing components on the edge/fog/cloud layers. These components feed data streams of non-binary data, either collected at the edge or produced by data processing components. All data streams are stored, aggregated and are available for further filtering, searching and forwarded to visualisation components.
INTRA	StreamHandler	Y	StreamHandler will be executed on the cloud context and will be interconnected with the Edge/Fog/Cloud layers. The non-binary data will be fed directly to the framework and will be utilised to trigger the visualisation aspects.
ATOS	DatAna	Y	DatAna can be deployed both at the edge/fog or on the cloud, perform some processing over the non-binary data, and connect with the other DMD subsystem components for further aggregation and storage.
CNR	Hierarchical Data Distribution (HDD)	Y	HDD, in synergy with the rest of the DMD subsystem components, will be used in order to move data where needed while satisfying the related edge/fog application constraints and requirements (real-time or offline).
Audio, visual, and multimodal AI subsystem		Y	The audio, visual and multimodal AI subsystem integrates data analysis tasks needed for the AI functionalities of the MARVEL framework, enabling AI-based decision-making.
AUD	devAIce	Y	devAIce will be used to extract audio features.
AU	Visual anomaly detection (ViAD)	Y	Visual anomaly detection will receive as input video frames and provide a label indicating whether they correspond to normal or anomalous situations for the scene under consideration.
AU	Audio-Visual anomaly detection (AVAD)	Y	Audio-Visual anomaly detection will receive as input video frames and synchronised audio snippets and provide a label indicating whether they correspond to normal or anomalous situations for the scene under consideration.
AU	Visual crowd counting (VCC)	Y	Visual crowd counting will receive as input a video frame and provide a number indicating the number of people appearing in them.

AU	Audio-Visual crowd counting (AVCC)	Y	Audio-Visual crowd counting will receive as input video frames and synchronised audio snippet and provide a number indicating the number of people in the scene.
TAU	Automated audio captioning (AAC)	Y	The audio captioning will receive as an input single-channel audio segment and provides a textual description of the content. AAC can be used, if audio learning examples are provided with a full textual description and audio signals are available.
TAU	Sound event detection (SED)	Y	Sound event detection will receive as an input single-channel audio segment and provides a list of detected sound events (label and timestamp) as output. In the MT3 use case, sound classes are related to robbery prevention and car damages. SED can be used if audio signals are available in the use case.
TAU	Sound event localisation and detection (SELD)	Y	Sound event detection and localisation will receive as an input multi-channel audio segment captured with a microphone array, and as output it provides a list of detected sound events (label and timestamp) and their azimuth and elevation. In the MT3 use case, sound classes are related to robbery prevention and car damages. SELD is used if multi-channel audio capturing is available.
TAU	Acoustic scene classification (ASC)	Y	Acoustic scene classification will receive as an input single-channel audio segment and provides a scene class label to describe the overall characteristics of the acoustic scene. In the MT3 use case, classes are related to the overall ambience created by different parking lot usage types. ASC can be used if audio signals are available in the use case.
Optimised E2F2C processing and deployment subsystem		Y	The E2F2C subsystem integrates dispersed execution sites into a unified, distributed execution environment, enabling the deployment of services at all layers spanning from Edge to Fog to Cloud/HPC-Centre.
FORTH	GPURegex	N	The component is not applicable to this use case.
CNR	DynHP	Y	DynHP will be used to train and compress the DNN model identified for this task such that it can be deployed to fog/edge devices.
UNS	FedL	N	No available data set matching.
FORTH	Karvdash	Y	Karvdash will provide a dashboard for instantiating services as orchestrated containers, and deployed via appropriate automation to execution sites selected by a dynamic online optimisation strategy.
E2F2C infrastructure		Y	This pilot will build and operate on E2F2C infrastructure; it may take advantage of HPC infrastructure to perform various tasks within the MARVEL project.
PSNC	HPC infrastructure	Y	HPC and cloud infrastructure in this use case can be used for tasks that require high computing power like model training or cloud-based inference (e.g., long-term decision-making). This component will also host the MARVEL Data corpus.
PSNC	HPC management and orchestration	Y	This component will allow for efficient use of the HPC and cloud infrastructure provided by PSNC.
MT	Cloud tier	Y	MT will make use of the cloud tier provided by the MARVEL consortium.
MT	Fog tier	Y	The fog tier will consist of a DELL workstation located at FBK premises, that, via secure connection captures the raw data from the sensors. The server may be equipped with an NVIDIA GeForce RTX 3080.

MT	Edge tier	Y	The edge consists of a series of Raspberry Pis installed in the nearest cabinet where selected cameras are installed. It is expected that at the end of the experimentation that some of some submodules can be installed and executed near the cameras and MEMS microphones.
System outputs: User interactions and the decision-making toolkit		Y	Text-annotated attention maps which will enhance video streams with textual information and indications of associated audio events and multisource, multimodal summaries, that allow users to explore and understand audio-visual, sensor, and other context-enriching data (e.g., weather data, information from incident reporting systems, parking sensors, etc.) and interact with them.
ZELUS	SmartViz	Y	SmartViz can visualise data across time or in juxtaposition with other information like weather data, activities happening in the location, etc. in order to identify patterns and hidden relationships. It can also depict maps of car allocation or movement in order to facilitate decisions for space management.
STS	Data Corpus-as-a-Service	Y	Data Corpus-as-a-Service can provide a query only to the relative data of the respective use case.

Table 14: MARVEL components in the MT4 use case

MT use case 4: Analysis of a Specific Area			
Component owner	**Subsystem /Component**		**Comments on how the component will be used**
Sensing and perception subsystem		Y	Sensing and perception components will be used mainly to collect unimodal and multimodal data (mainly audio and video). In addition, some of the components provide resources that allow processing at the edge, e.g., sound event detection.
IFAG	Advanced MEMS microphones	M	Multi-channel acoustic data acquisition and processing in the edge: • data acquisition through two to eight IM69D130 MEMS microphones, • connected to either Flex evaluation kit or AudioHubNano4D; • data transmission to a processing node via PDM or USB respectively; • possible processing nodes are either Cypress PSoC64 Standard Secure or a RPi.
FBK	SED@Edge	M	As long as audio is available the tool can be used to detect traffic-related events, although this seems more a computer vision task.
GRN	GRNEdge	N	GRNEdge is not available as a hardware component in the MT pilot.
UNS	AVDrone	N	AVDrone is not available as a hardware component in the MT pilot.
AUD	sensMiner	N	sensMiner will not be used in this pilot.
GRN	CATFlow	N	CATFlow is not required for this MT use case.
Security, privacy, and data protection subsystem		Y	The subsystem will ensure the secure transmission of the data from the sensors to the cloud services. In addition, the anonymisation components will minimise the misuse of the data to identify subjects, audio may not be always available. For some video recording, anonymisation may not be necessary.

FORTH	EdgeSec	Y	EdgeSec will encrypt 100% of the traffic from/to the hosts in the whole E2F2C infrastructure and will attest to the integrity of binaries using Intel SGX security features when available.
FBK	VideoAnony	M	VideoAnony will detect the cyclists and pedestrians that appear in the scene and blur their faces.
FBK	AudioAnony	M	As long as the audio is available, audio anonymisation can be applied. However, speech content is not likely to be recorded.
AUD	VAD (devAIce)	M	VAD (devAIce) will be used to detect speech signals, in case audio is available.
Data management and distribution subsystem		Y	The subsystem will be handling massive amounts of data (with various sources, formats, and frequencies) dealing with their proper management and distribution.
ITML	Data Fusion Bus (DFB)	Y	DFB will be deployed on the cloud and connect to data-producing components on the edge/fog/cloud layers. These components feed data streams of non-binary data, either collected at the edge or produced by data processing components. All data streams are stored, aggregated and are available for further filtering, searching and forwarded to visualisation components.
INTRA	StreamHandler	Y	StreamHandler will be executed on the cloud context and will be interconnected with the Edge/Fog/Cloud layers. The non-binary data will be fed directly to the framework and will be utilised to trigger the visualisation aspects.
ATOS	DatAna	Y	DatAna can be deployed both at the edge/fog or on the cloud, perform some processing over the non-binary data, and connect with the other DMD subsystem components for further aggregation and storage.
CNR	Hierarchical Data Distribution (HDD)	Y	HDD, in synergy with the rest of the DMD subsystem components, will be used in order to move data where needed while satisfying the related edge/fog application constraints and requirements (real-time or offline).
Audio, visual, and multimodal AI subsystem		Y	The audio, visual and multimodal AI subsystem integrates data analysis tasks needed for the AI functionalities of the MARVEL framework, enabling AI-based decision-making.
AUD	devAIce	M	devAIce can be used to extract audio features (provided microphones are installed to the scene).
AU	Visual anomaly detection (ViAD)	Y	Visual anomaly detection will receive as input video frames and provide a label indicating whether they correspond to normal or anomalous situations for the scene under consideration.
AU	Audio-Visual anomaly detection (AVAD)	M	In the case where microphones are installed to the scene and synchronised to the visual data stream, audio-visual anomaly detection will receive as input video frames and synchronised audio snippets and provide a label indicating whether they correspond to normal or anomalous situations for the scene under consideration.
AU	Visual crowd counting (VCC)	Y	Visual crowd counting will receive as input a video frame and provide a number indicating the number of people appearing in them.
AU	Audio-Visual crowd counting (AVCC)	M	In the case where microphones are installed to the scene and synchronised to the visual data stream, audio-visual crowd counting will receive as input video frames and synchronised audio snippet and provide a number indicating the number of people in the scene.

TAU	Automated audio captioning (AAC)	M	The audio captioning will receive as an input single-channel audio segment and provides a textual description of the content. AAC can be used, if audio learning examples are provided with a full textual description and audio signals are available.
TAU	Sound event detection (SED)	M	Sound event detection will receive as an input single-channel audio segment and provides a list of detected sound events (label and timestamp) as output. In the MT4 use case, sound classes are related to specific events in the area to support long-term decision-making by public authorities. SED can be used if audio signals are available in the use case.
TAU	Sound event localiSation and detection (SELD)	M	Sound event detection and localisation will receive as an input multi-channel audio segment captured with a microphone array, and as output it provides a list of detected sound events (label and timestamp) and their azimuth and elevation. In the MT4 use case, sound classes are related to specific events in the area to support long-term decision-making by public authorities. SELD is used, if multi-channel audio capturing is available.
TAU	Acoustic scene classification (ASC)	M	Acoustic scene classification will receive as an input single-channel audio segment and provides a scene class label to describe the overall characteristics of the acoustic scene. In the MT4 use case, classes are related to the overall ambience created by area usage types/levels. ASC can be used if audio signals are available in the use case.
Optimised E2F2C processing and deployment subsystem		Y	The E2F2C subsystem integrates dispersed execution sites into a unified, distributed execution environment, enabling the deployment of services at all layers spanning from Edge to Fog to Cloud/HPC-Centre.
FORTH	GPURegex	M	GPURegex will provide accelerated pattern matching capabilities in order to contribute to event recognition. In order to be able to provide meaningful results, a pre-processing phase is needed in order to extract event signatures from raw AV data and feed them to our engine.
CNR	DynHP	Y	DynHP will be used to train and compress the DNN model identified for this task such that it can be deployed to fog/edge devices.
UNS	FedL	M	FedL can be used across any two or more datasets (use cases) that share the same feature space, or to a single dataset by splitting it into partitions.
FORTH	Karvdash	Y	Karvdash will provide a dashboard for instantiating services as orchestrated containers, and deployed via appropriate automation to execution sites selected by a dynamic online optimisation strategy.
E2F2C infrastructure		Y	This pilot will build and operate on E2F2C infrastructure; it may take advantage of HPC infrastructure to perform various tasks within the MARVEL project.
PSNC	HPC infrastructure	Y	HPC and cloud infrastructure in this use case can be used for tasks that require high computing power like model training or cloud-based inference (e.g., long-term decision-making). This component will also host the MARVEL Data corpus.
PSNC	HPC management and orchestration	Y	This component will allow for efficient use of the HPC and cloud infrastructure provided by PSNC.
MT	Cloud tier	Y	MT will make use of the cloud tier provided by the MARVEL consortium.

MT	Fog tier	Y	The fog tier will consist of a DELL workstation located at FBK premises, that, via secure connection captures the raw data from the sensors. The server may be equipped with an NVIDIA GeForce RTX 3080.
MT	Edge tier	Y	The edge consists of a series of Raspberry Pis installed in the nearest cabinet where selected cameras are installed. It is expected that at the end of the experimentation that some of some submodules can be installed and executed near the cameras and MEMS microphones.
System outputs: User interactions and the decision-making toolkit		Y	Data presentation and advanced visualisations that reveal hidden insights of valuable knowledge and multisource, multimodal summaries, that allow users to explore and understand audio-visual, sensor, and other context-enriching data (e.g., weather data, information from incident reporting systems, parking sensors, etc.) and interact with them.
ZELUS	SmartViz	Y	SmartViz can visualise historical data across time or in juxtaposition with other information like weather data, incidents, etc. in order to identify patterns, hidden relationships, and areas for further investigation.
STS	Data Corpus-as-a-Service	Y	Data Corpus-as-a-Service can provide a query only to the relative data of the respective use case.

Table 15: MARVEL components in the UNS1 use case

UNS use case 1: Drone experiment			
Component owner	**Subsystem/Component**		**Comments on how the component will be used**
Sensing and perception subsystem		Y	Sensing and perception components will be used mainly to collect unimodal and multimodal data (mainly audio and video). In addition, some of the components provide resources that allow processing at the edge, e.g., sound event detection.
IFAG	Advanced MEMS microphones	Y	Multi-channel acoustic data acquisition and processing in the edge: • data acquisition through two to eight IM69D130 MEMS microphones, • connected to either Flex evaluation kit or AudioHubNano4D; • data transmission to a processing node via PDM or USB respectively; • possible processing nodes are either Cypress PSoC64 Standard Secure or a RPi.
FBK	SED@Edge	Y	In this use case, the tool can be used on edge devices (either on the drone or on the ground if any) to detect the presence of a list of predefined sound events.
GRN	GRNEdge	N	GRNEdge is not available as a hardware component for this UNS use case.
UNS	AVDrone	Y	AVDrone is a configuration of drone and ground-based equipment for data capturing and streaming serving as the edge tier for the UNS Drone experiment. The setup includes sensMiner app from AUD.
AUD	sensMiner	Y	sensMiner is used in the data collection phase of the Drone experiment for additional audio data, (coarse) event annotations

			(tagging), and GPS tagging.
GRN	CATFlow	N	CATFlow is not relevant for this UNS use case.
Security, privacy, and data protection subsystem		Y	The subsystem will ensure the secure transmission of the data from the sensors to the cloud services. In addition, the anonymisation components will minimise the misuse of the data to identify subjects. Audio may be not audible from the drone sensors. For some video recordings, anonymisation may not be necessary.
FORTH	EdgeSec	Y	EdgeSec will encrypt 100% of the traffic from/to the hosts in the whole E2F2C infrastructure and will attest to the integrity of binaries using Intel SGX security features when available.
FBK	VideoAnony	Y	VideoAnony will detect the people in the scene and blur their faces. The necessity depends on the height of the cameras.
FBK	AudioAnony	Y	Audio anonymisation can be applied on the drone or on the first processing device
AUD	VAD (devAIce)	Y	VAD will be used on the first processing device to detect speech signals. It will not be deployed in the drone because speech is unlikely to occur there.
Data management and distribution subsystem		M	The subsystem will be handling massive amounts of data (with various sources, formats, and frequencies) dealing with their proper management and distribution.
ITML	Data Fusion Bus (DFB)	M	DFB will be deployed on the cloud and connect to data-producing components on the edge/fog/cloud layers. These components feed data streams of non-binary data, either collected at the edge or produced by data processing components. All data streams are stored, aggregated and are available for further filtering, searching and forwarded to visualisation components.
INTRA	StreamHandler	M	StreamHandler will be executed on the cloud context and will be interconnected with the Edge/Fog/Cloud layers. The non-binary data will be fed directly to the framework and will be utilised to trigger the visualisation aspects.
ATOS	DatAna	M	DatAna can be deployed both at the edge/fog or on the cloud, perform some processing over the non-binary data, and connect with the other DMD subsystem components for further aggregation and storage. In this use case, DatAna might be deployed directly in the Raspberry PI in the drone (MiNiFi agent) and connected to a NiFi instance in the cloud/server potentially offline (to be tested: ideally, if no connection is available the data should be moved automatically to the NiFi when the connection is restored).
CNR	Hierarchical Data Distribution (HDD)	M	HDD, in synergy with the rest of the DMD subsystem components, will be used in order to move data where needed while satisfying the related edge/fog application constraints and requirements (real-time or offline).
Audio, visual, and multimodal AI subsystem		Y	The audio, visual and multimodal AI subsystem integrates data analysis tasks needed for the AI functionalities of the MARVEL framework, enabling AI-based decision-making.
AUD	devAIce	Y	devAIce will be used for audio feature extraction.
AU	Visual anomaly detection (ViAD)	Y	Visual anomaly detection will receive as input video frames and provide a label indicating whether they correspond to normal or anomalous situations for the scene under consideration.

AU	Audio-Visual anomaly detection (AVAD)	Y	Audio-Visual anomaly detection will receive as input video frames and synchronised audio snippets and provide a label indicating whether they correspond to normal or anomalous situations for the scene under consideration.
AU	Visual crowd counting (VCC)	Y	Visual crowd counting will receive as input a video frame and provide a number indicating the number of people appearing in them.
AU	Audio-Visual crowd counting (AVCC)	Y	Audio-Visual crowd counting will receive as input video frames and synchronised audio snippet and provide a number indicating the number of people in the scene.
TAU	Automated audio captioning (AAC)	Y	The audio captioning will receive as an input single-channel audio segment and provides a textual description of the content. AAC can be used, if audio learning examples are provided with a full textual description are available.
TAU	Sound event detection (SED)	Y	Sound event detection will receive as an input single-channel audio segment and provides a list of detected sound events (label and timestamp) as output. In the UNS1 use case, sound classes are related to crowd activity (e.g., signing, shouting) and normal environmental events (e.g., birds, bicycles).
TAU	Sound event localisation and detection (SELD)	Y	Sound event detection and localisation will receive as an input multi-channel audio segment captured with a microphone array, and as output it provides a list of detected sound events (label and timestamp) and their azimuth and elevation. In the UNS1 use case, sound classes are related to crowd activity (e.g., signing, shouting) and normal environmental events (e.g., birds, bicycles). SELD is used, if multi-channel audio capturing is available.
TAU	Acoustic scene classification (ASC)	Y	Acoustic scene classification will receive as an input single-channel audio segment and provides a scene class label to describe the overall characteristics of the acoustic scene. In the UNS1 use case, classes are related to the overall ambience created by crowd (e.g., neutral, party).
Optimised E2F2C processing and deployment subsystem		Y	The E2F2C subsystem integrates dispersed execution sites into a unified, distributed execution environment, enabling the deployment of services at all layers spanning from Edge to Fog to Cloud/HPC-Centre.
FORTH	GPURegex	N	The component is not applicable to this use case.
CNR	DynHP	Y	DynHP will be used to train and compress the DNN model identified for this task such that it can be deployed to fog/edge devices.
UNS	FedL	Y	FedL will be implemented between the datasets from the UNS1 use case Drone experiment and the MT1 use case Monitoring of Crowded Areas.
FORTH	Karvdash	M	Karvdash will provide a dashboard for instantiating services as orchestrated containers, and deployed via appropriate automation to execution sites selected by a dynamic online optimisation strategy.
E2F2C infrastructure		Y	This pilot will build and operate on E2F2C infrastructure; it may take advantage of HPC infrastructure to perform various tasks within the MARVEL project.
PSNC	HPC infrastructure	Y	HPC and cloud infrastructure in this use case can be used for tasks that require high computing power like model training or cloud-based inference (e.g., long-term decision-making). This

			component will also host the MARVEL Data corpus.
PSNC	HPC management and orchestration	Y	This component will allow for efficient use of the HPC and cloud infrastructure provided by PSNC.
UNS	Cloud tier	Y	UNS will make exclusive use of the cloud services provided by the MARVEL consortium.
UNS	Fog tier	Y	The fog tier consists of a PC, distributed Raspberry Pi 3.0 and 4.0 computing environment, and a server machine. UNS data server supports distributed network storage with RAID 1+1 protection.
UNS	Edge tier	Y	The edge tier for the Drone experiment consists of the AVDrone component.
System outputs: User interactions and the decision-making toolkit		Y	Based on analytical reasoning for medium to long-term business decision-making based on queries execution over the processed audio-visual data, the toolkit can activate workflows for pushing alerts or messages to the intervention teams, depending on thresholds set regarding what is considered to be classified as illegal or dangerous behaviour. It can also support real-time visualisations of alerts and detected events for short-term decisions and monitoring, supported by a rule-based engine.
ZELUS	SmartViz	Y	In this case, SmartViz can provide a dashboard with an update of the spots for which alerts and communications need to be sent or in general an overview of the areas where appropriate action is needed. It can also visualise data across time or in juxtaposition with other information like weather data, activities happening in the location, etc. in order to identify patterns and hidden relationships. It can also try to depict trajectories of crowd movement depending on classifications.
STS	Data Corpus-as-a-Service	Y	Data Corpus-as-a-Service can provide a query only to the relative data of the respective use case.

Table 16: MARVEL components in the UNS2 use case

UNS use case 2: Audio-Visual Emotion Recognition			
Component owner	**Subsystem/Component**		**Comments on how the component will be used**
Sensing and perception subsystem		Y	Sensing and perception components will be used mainly to collect unimodal and multimodal data (mainly audio and video). In addition, some of the components provide resources that allow processing at the edge, e.g., sound event detection.
IFAG	Advanced MEMS microphones	Y	Multi-channel acoustic data acquisition and processing in the edge: • data acquisition through two to eight IM69D130 MEMS microphones, • connected to either Flex evaluation kit or AudioHubNano4D; • data transmission to a processing node via PDM or USB respectively; • possible processing nodes are either Cypress PSoC64 Standard Secure or a RPi.
FBK	SED@Edge	N	Detecting sound events is not relevant for this use case.

GRN	GRNEdge	N	GRNEdge is not available as a hardware component for this UNS use case.
UNS	AVDrone	N	The component is not relevant for the UNS2 use case as different edge tier will be custom-made for this use case.
AUD	sensMiner	N	sensMiner is not relevant for this use case.
GRN	CATFlow	N	CATFlow is not relevant for this UNS use case.
Security, privacy, and data protection subsystem		Y	The subsystem will ensure the secure transmission of the data from the sensors to the cloud services. In addition, the anonymisation components will minimise the misuse of the data to identify subjects. Anonymisation could be detrimental for the emotion recognition system, therefore it may be necessary to use specific, possibly less effective, solutions.
FORTH	EdgeSec	Y	EdgeSec will encrypt 100% of the traffic from/to the hosts in the whole E2F2C infrastructure and will attest to the integrity of binaries using Intel SGX security features when available.
FBK	VideoAnony	Y	VideoAnony will detect the people in the scene and blur their faces. The necessity depends on the height of the cameras.
FBK	AudioAnony	Y	Audio anonymisation can be applied. However, it will have a critical impact on the emotion recognition tool (which is based on voice). Anonymised features are probably more suitable here.
AUD	VAD (devAIce)	Y	VAD will be used to detect speech segments.
Data management and distribution subsystem		M	The subsystem will be handling massive amounts of data (with various sources, formats, and frequencies) dealing with their proper management and distribution.
ITML	Data Fusion Bus (DFB)	M	DFB will be deployed on the cloud and connect to data-producing components on the edge/fog/cloud layers. These components feed data streams of non-binary data, either collected at the edge or produced by data processing components. All data streams are stored, aggregated and are available for further filtering, searching and forwarded to visualisation components.
INTRA	StreamHandler	M	StreamHandler will be executed on the cloud context and will be interconnected with the Edge/Fog/Cloud layers. The non-binary data will be fed directly to the framework and will be utilised to trigger the visualisation aspects.
ATOS	DatAna	M	DatAna can be deployed both at the edge/fog or on the cloud, perform some processing over the non-binary data, and connect with the other DMD subsystem components for further aggregation and storage.
CNR	Hierarchical Data Distribution (HDD)	M	HDD, in synergy with the rest of the DMD subsystem components, will be used in order to move data where needed while satisfying the related edge/fog application constraints and requirements (real-time or offline).
Audio, visual, and multimodal AI subsystem		Y	The audio, visual and multimodal AI subsystem integrates data analysis tasks needed for the AI functionalities of the MARVEL framework, enabling AI-based decision-making.
AUD	devAIce	Y	devAIce will be used to extract audio features. Moreover, the emotion recognition modules of devAIce will be used for speech emotion recognition
AU	Visual anomaly detection (ViAD)	N	As this use case focuses on emotion recognition, visual anomaly detection is not of interest.

AU	Audio-Visual anomaly detection (AVAD)	N	As this use case focuses on emotion recognition, visual anomaly detection is not of interest.
AU	Visual crowd counting (VCC)	N	As this use case focuses on emotion recognition, visual anomaly detection is not of interest.
AU	Audio-Visual crowd counting (AVCC)	N	As this use case focuses on emotion recognition, visual anomaly detection is not of interest.
TAU	Automated audio captioning (AAC)	N	The component is not applicable for this use case.
TAU	Sound event detection (SED)	N	The component is not applicable for this use case.
TAU	Sound event localisation and detection (SELD)	N	The component is not applicable for this use case.
TAU	Acoustic scene classification (ASC)	N	The component is not applicable for this use case.
Optimised E2F2C processing and deployment subsystem		Y	The E2F2C subsystem integrates dispersed execution sites into a unified, distributed execution environment, enabling the deployment of services at all layers spanning from Edge to Fog to Cloud/HPC-Centre.
FORTH	GPURegex	N	The component is not applicable to this use case.
CNR	DynHP	Y	DynHP will be used to train and compress the DNN model identified for this task such that it can be deployed to fog/edge devices
UNS	FedL	Y	FedL will be implemented by splitting the database to be generated within the use case to smaller datasets; exact splitting is yet to be defined (e.g., one option is datasets with different emotions' frequency, thus mimicking practical settings).
FORTH	Karvdash	M	Karvdash will provide a dashboard for instantiating services as orchestrated containers, and deployed via appropriate automation to execution sites selected by a dynamic online optimisation strategy.
E2F2C infrastructure		Y	This pilot will build and operate on E2F2C infrastructure; it may take advantage of HPC infrastructure to perform various tasks within the MARVEL project.
PSNC	HPC infrastructure	Y	HPC and cloud infrastructure in this use case can be used for tasks that require high computing power like model training or cloud-based inference (e.g., long-term decision-making). This component will also host the MARVEL Data corpus.
PSNC	HPC management and orchestration	Y	This component will allow for efficient use of the HPC and cloud infrastructure provided by PSNC.
UNS	Cloud tier	Y	UNS will make exclusive use of the cloud services provided by the MARVEL consortium.
UNS	Fog tier	Y	The fog tier consists of a PC, distributed Raspberry Pi 3.0 and 4.0 computing environment, and a server machine. UNS data server supports distributed network storage with RAID 1+1 protection.
UNS	Edge tier	Y	Data collection: a smartphone with an audio-visual data capturing Android app. Experiment execution: a camera and a microphone with data logging (e.g., Raspberry Pi); edge processing HW will also be considered (e.g., Intel NUC).

System outputs: User interactions and the decision-making toolkit		Y	Text-annotated attention maps will be used to enhance and augment video streams with textual information and indications of associated audio events. It can also support real-time visualisations of alerts and detected events for short-term decisions and monitoring, supported by a rule-based engine.
ZELUS	SmartViz	Y	In this case, SmartViz can provide a dashboard with an update of the spots for which alerts and communications need to be sent or in general an overview of the areas where appropriate action is needed. A heatmap of emotions will also be considered.
STS	Data Corpus-as-a-Service	Y	Data Corpus-as-a-Service can provide a query only to the relative data of the respective use case.

Table 17: Overview of the use case –components mappings

Subsystems	Components	Use cases									
		GRN1	GRN2	GRN3	GRN4	MT1	MT2	MT3	MT4	UNS1	UNS2
Sensing and perception	IFAG-MEMS	Y	Y	Y	Y	M	Y	Y	M	Y	Y
	SED@edge	Y	Y	Y	Y	M	Y	Y	M	Y	N
	CATFlow	Y	Y	Y	Y	N	N	N	N	N	N
	GRNEdge	Y	Y	Y	Y	N	N	N	N	N	N
	AVDrone	N	N	N	N	N	N	N	N	Y	N
	sensMiner	N	N	N	N	N	N	N	N	Y	N
Security, privacy, and data protection	EdgeSec	M	M	M	M	Y	Y	Y	Y	Y	N
	VideoAnony	Y	Y	Y	Y	Y	Y	Y	M	Y	Y
	AudioAnony	Y	Y	Y	Y	M	Y	Y	M	Y	Y
	VAD (devAIce)	Y	Y	Y	Y	M	Y	Y	M	Y	Y
Data Management and distribution	DFB	Y	Y	Y	Y	Y	Y	Y	Y	M	M
	Stream Handler	Y	Y	Y	Y	Y	Y	Y	Y	M	M
	DatAna	Y	Y	Y	Y	Y	Y	Y	Y	M	M
	HDD	Y	Y	Y	Y	Y	Y	Y	Y	M	M
Audio, visual, and multimodal AI	ViAD	Y	Y	Y	N	Y	Y	Y	Y	Y	N
	AVAD	Y	Y	Y	N	M	Y	Y	M	Y	N
	VCC	Y	N	Y	N	Y	Y	Y	Y	Y	N
	AVCC	Y	N	Y	N	M	Y	Y	M	Y	N
	AAC	N	Y	N	N	M	Y	Y	M	Y	N
	SED	Y	Y	Y	Y	N	Y	Y	M	Y	N
	ASC	Y	Y	Y	Y	M	Y	Y	M	Y	N
	SELD	Y	Y	Y	N	N	Y	Y	M	Y	N
	devAIce	Y	Y	Y	Y	N	Y	Y	M	Y	Y

Optimised E2F2C processing and deployment	GPURegex	Y	Y	Y	N	M	Y	N	M	N	N
	DynHP	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
	FedL	N	N	M	M	Y	M	N	M	Y	Y
	Karvdash	Y	Y	Y	Y	Y	Y	Y	Y	M	M
E2F2C infrastructure	HPC	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
	HPC management and orchestration	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
	Cloud tier	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
	Fog tier	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
	Edge tier	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
User interactions and the decision-making toolkit	SmartViz	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
	Marvel Data Corpus-as-a-Service	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

12.2 HPC customisation across the use case experiments

All defined trial cases may take advantage of HPC infrastructure to perform various tasks within the MARVEL project. HPC and cloud infrastructure can be used for tasks that require high computing power like model training or cloud-based inference (e.g., long-term decision making). The pilot and pre-production testbeds will be composed of existing computing and storage resources. The customisation at this stage was based on an analysis of the initial needs of the pilots performed in D2.1 Collection and analysis of experimental data [9] (e.g., Table 13 Estimates of data rates per device, layer and location, for devices that are deployed in the streaming of data in [9]) and also within the current deliverable. Sufficient resources have been reserved to guarantee meeting the requirements of each use case. At the moment, the HPC infrastructure does not require particular adaptation to any of the use cases. PSNC will perform further customisation continuously as needed at a later time.

12.3 Data value chains

In the scope of the projects funded under the Big Data Value PPP[23], the project DataBench devised what they called a generic Big Data Architectural Blueprint[24]. This blueprint is mapped to the main four steps defined in DataBench for a typical data value chain, namely (i) Data Acquisition/Collection (covering aspects related to data ingestion, extraction, processing, streaming for different data types); (ii) Data Storage/Preparation (referring to storage, retrieval/access/queries, data protection, data curation, data fusion, and publication processes); (iii) Data Analytics/ML (including AI, ML, DL, data processing for analysis, training); and (iv) Data Visualisation, Action/Interaction (covering aspects about visualisation, data presentation, interaction with humans and other systems).

[23] Big Data Value Public Private Partnership, https://ec.europa.eu/digital-single-market/en/big-data-value-public-private-partnership

[24] https://toolbox.databench.eu/knowledgeNugget/nugget/84

This blueprint and its mappings to the data value chain are showed in Figure 53, providing a sequence of steps that is in principle generic enough to fit most of the big data and AI systems and applications. Note that the blueprint is abstract and can be applied to different data types. For instance, processing AV data will follow similar steps to processing textual or structured data, although the technologies involved might not be the same. Therefore, the mapping of the blueprint to a generic architecture might stay at an abstract level, but it might be more specific when dealing with concrete use cases and data types. The figure can be found operational for searching purposes in the DataBench Toolbox[25].

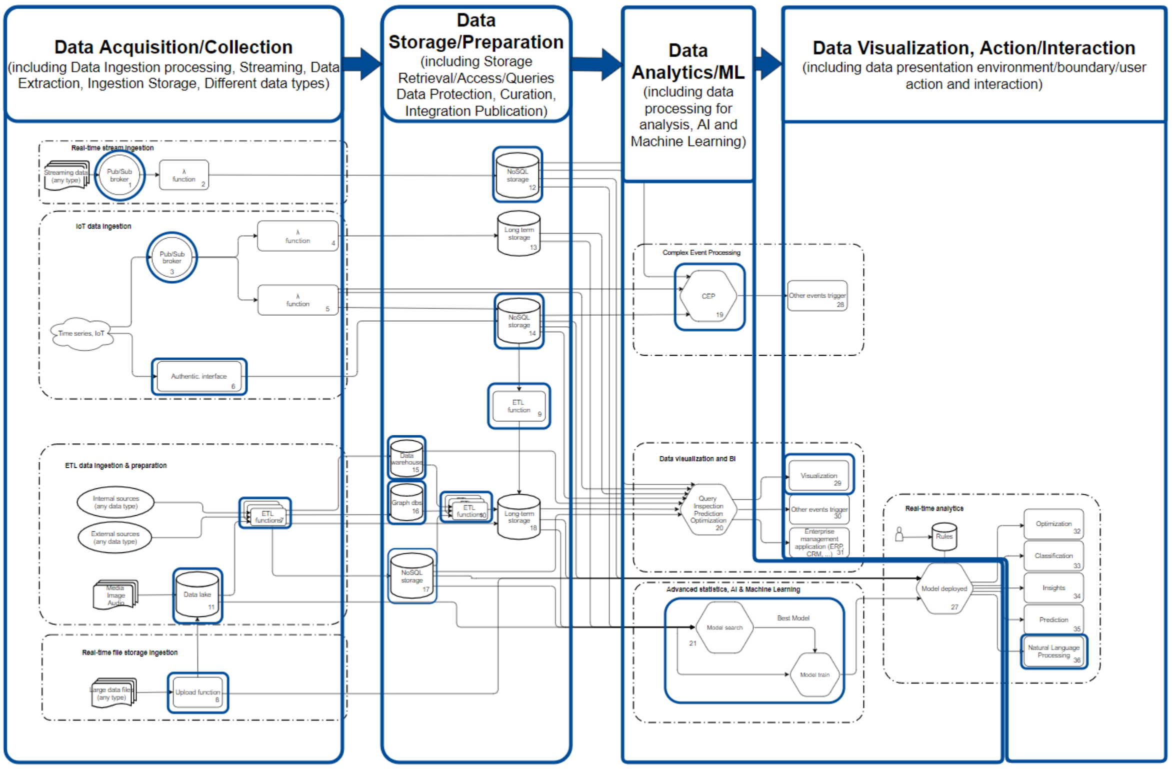

Data Acquisition/Collection
(including Data Ingestion processing, Streaming, Data Extraction, Ingestion Storage, Different data types)
Data Storage/Preparation
(including Storage Retrieval/Access/Queries Data Protection, Curation, Integration Publication)
Data Analytics/ML
(including data processing for analysis, AI and Machine Learning)
Data Visualization, Action/Interaction
(including data presentation environment/boundary/user action and interaction)
Real-time stream ingestion
Streaming data (any type)
Pub/Sub broker 1
λ function 2
IoT data ingestion
Pub/Sub broker 3
λ function 4
λ function 5
Time series, IoT
Authentic. interface 6
ETL data ingestion & preparation
Internal sources (any data type)
External sources (any data type)
ETL functions 7
Media Image Audio
Data lake 11
Real-time file storage ingestion
Large data files (any type)
Upload function 8
NoSQL storage 12
Long term storage 13
NoSQL storage 14
ETL function 9
Data warehouse 15
Graph dbs 16
ETL functions 10
Long-term storage 18
NoSQL storage 17
Complex Event Processing
CEP 19
Other events trigger 28
Data visualization and BI
Query Inspection Prediction Optimization 20
Visualization 29
Other events trigger 30
Enterprise management application (ERP, CRM, ...) 31
Advanced statistics, AI & Machine Learning
Model search 21
Best Model
Model train
Real-time analytics
Rules
Model deployed 27
Optimization 32
Classification 33
Insights 34
Prediction 35
Natural Language Processing 36

Figure 53. Generic Big Data and AI Pipeline blueprint from DataBench [2]

This blueprint has been customised by several EU-funded projects to map their own architecture and use cases, therefore providing a clear link among different initiatives and a standard way of the technological choices selected in each project. Some of the results of these mappings can be seen also in the DataBench Toolbox searching by "R&D Project"[26]. Information about the methodology followed to provide the mappings can be seen in the DataBench deliverable D5.5 [10].

In the case of the MARVEL use cases, the result of this work of mapping the use case pipelines to the DataBench is presented in the following subsections.

[25] https://toolbox.databench.eu/unicum

[26] https://toolbox.databench.eu/benchmarks/searchByFeature/482

12.3.1 GRN1 use case: Safer roads

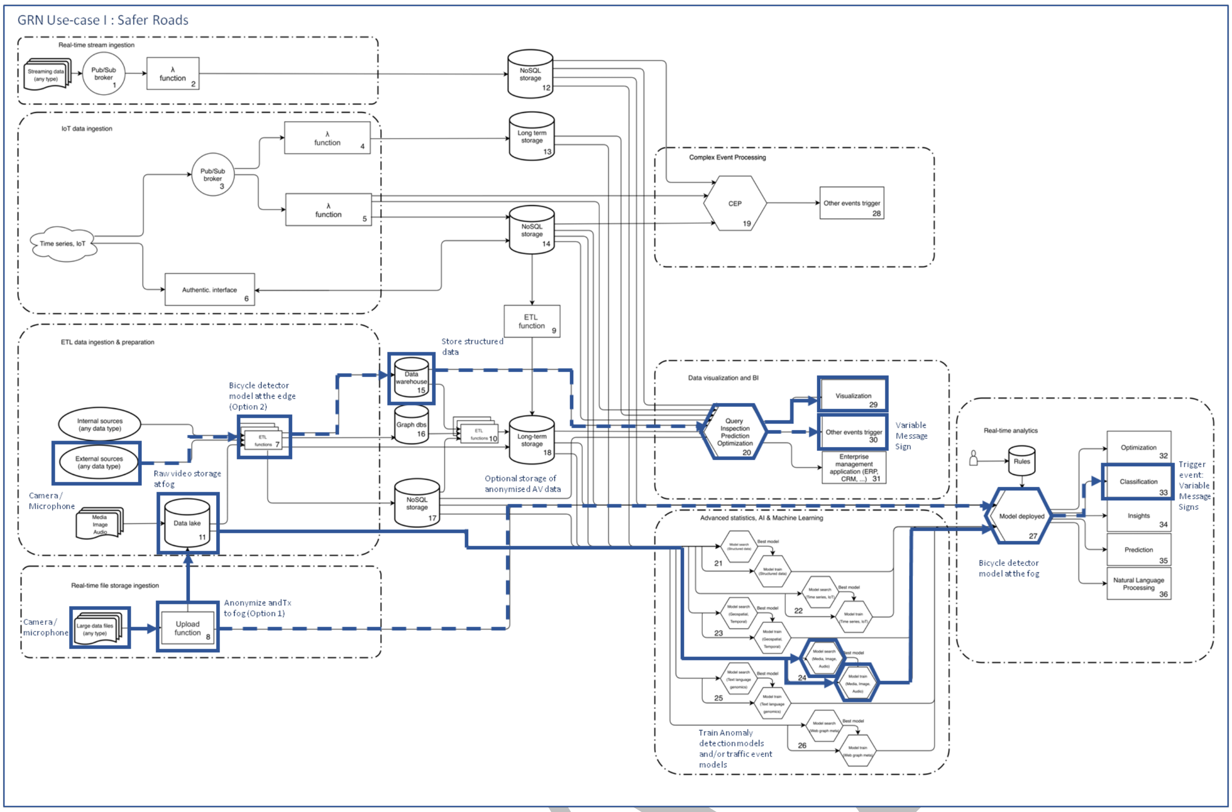

GRN Use-case I : Safer Roads
Real-time stream ingestion
Streaming data (any type)
Pub/Sub broker 1
λ function 2
IoT data ingestion
λ function 4
Pub/Sub broker 3
λ function 5
Time series, IoT
Authentic. interface 6
NoSQL storage 12
Long term storage 13
NoSQL storage 14
ETL function 9
ETL data ingestion & preparation
Store structured data
Data warehouse 15
Graph dbs 16
Bicycle detector model at the edge (Option 2)
Internal sources (any data type)
External sources (any data type)
ETL functions 7
ETL functions 10
Long-term storage 18
Raw video storage at fog
Camera / Microphone
Media Image Audio
Data lake 11
NoSQL storage 17
Optional storage of anonymised AV data
Real-time file storage ingestion
Camera / microphone
Large data files (any type)
Upload function 8
Anonymize and Tx to fog (Option 1)
Complex Event Processing
CEP 19
Other events trigger 28
Data visualization and BI
Query Inspection Prediction Optimization 20
Visualization 29
Other events trigger 30
Variable Message Sign
Enterprise management application (ERP, CRM, ...) 31
Advanced statistics, AI & Machine Learning
Model search (Structured data)
Best model
Model train (Structured data)
21
Model search (Time series, IoT)
Model train (Time series, IoT)
22
Model search (Geospatial, Temporal)
Model train (Geospatial, Temporal)
23
Model search (Media, Image, Audio)
Model train (Media, Image, Audio)
24
Model search (Text language genomics)
Model train (Text language genomics)
25
Model search (Web graph meta)
Model train (Web graph meta)
26
Train Anomaly detection models and/or traffic event models
Real-time analytics
Rules
Model deployed 27
Bicycle detector model at the fog
Optimization 32
Classification 33
Trigger event: Variable Message Signs
Insights 34
Prediction 35
Natural Language Processing 36

Figure 54. Specification of the generic Big Data and AI pipeline blueprint for the GRN1 use case: Safer roads

Two options are planned for implementation. Option 1 starts with the real-time file storage ingestion block, the camera and microphone are used to obtain the large data files. The upload function (8) represents the anonymisation function to remove PD as well as transmit the data to fog. In the fog, this data which is a raw anonymised video is stored in the data lake (11). This data is transferred to train anomaly detection and/or traffic event models (24). The model is deployed (27) as a bicycle detector model in the fog. The model is able to perform classification (33) and trigger an event to inform drivers via the variable message signs that a cyclist is on the same road.

Once the model is trained, the data collected is transferred in real-time (after it has been anonymised, 8), via the link connected directly to (27) in the fog where real-time (or almost) classification occurs and alerts the drivers instantly.

In option 2 the bicycle detector is deployed at the edge, as seen in the path starting at the ETL data ingestion and preparation block. Here the data is obtained from a camera and microphone. This data is processed in (7) at the edge to detect bicycles. The data obtained is structured data and is stored in a data warehouse (15) and transferred to (20) such that it can be visualised (29) or trigger an event (30) to display a message on the variable message sign to alert drivers of nearby cyclists in real-time.

12.3.2 GRN2 use case: Road user behaviour

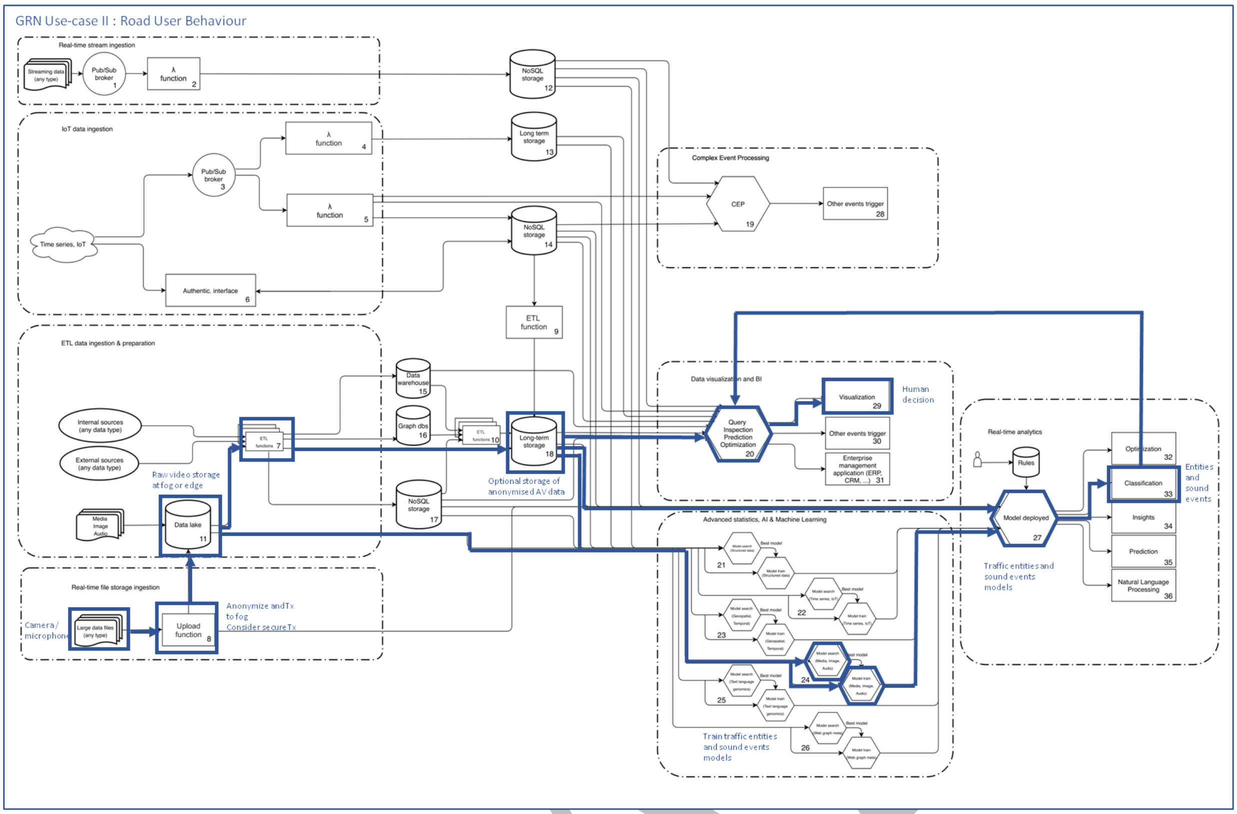

GRN Use-case II : Road User Behaviour
Real-time stream ingestion
Streaming data (any type)
Pub/Sub broker 1
λ function 2
NoSQL storage 12
IoT data ingestion
λ function 4
Long term storage 13
Pub/Sub broker 3
λ function 5
NoSQL storage 14
Time series, IoT
Authentic. interface 6
ETL function 9
ETL data ingestion & preparation
Data warehouse 15
Graph dbs 16
Internal sources (any data type)
External sources (any data type)
ETL functions 7
ETL functions 10
Long-term storage 18
Raw video storage at fog or edge
Optional storage of anonymised AV data
NoSQL storage 17
Media Image Audio
Data lake 11
Real-time file storage ingestion
Camera / microphone
Large data files (any type)
Upload function 8
Anonymize and Tx to fog Consider secure Tx
Complex Event Processing
CEP 19
Other events trigger 28
Data visualization and BI
Query Inspection Prediction Optimization 20
Visualization 29
Human decision
Other events trigger 30
Enterprise management application (ERP, CRM, ...) 31
Real-time analytics
Rules
Model deployed 27
Optimization 32
Classification 33
Entities and sound events
Insights 34
Prediction 35
Natural Language Processing 36
Traffic entities and sound events models
Advanced statistics, AI & Machine Learning
Model search (Structured data)
Best model
Model train (Structured data)
21
Model search (Time series, IoT)
Model train (Time series, IoT)
22
Model search (Geospatial, Temporal)
Model train (Geospatial, Temporal)
23
Model search (Media, Image, Audio)
Model train (Media, Image, Audio)
24
Model search (Text language genomics)
Model train (Text language genomics)
25
Model search (Web graph meta)
Model train (Web graph meta)
26
Train traffic entities and sound events models

Figure 55. Specification of the generic Big Data and AI pipeline blueprint for the GRN2 use case: Road user behaviour

For the *GRN Use-case II: Road User Behaviour,* the links for model training are the same as in *GRN Use-case I.* The collected AV data is first anonymised and transferred to the data lake and then moved to long-term storage (18). This data can be used to train the *anomaly detection and classification models.* When deployed (27), the models are used as filters to process the queries set by the human, i.e., the models operate on the data stored in (18). The retrieved answers to the queries are made available to the human via the visualisation (29) and the human then processes the results.

12.3.3 GRN3 use case: Traffic conditions and anomalous events

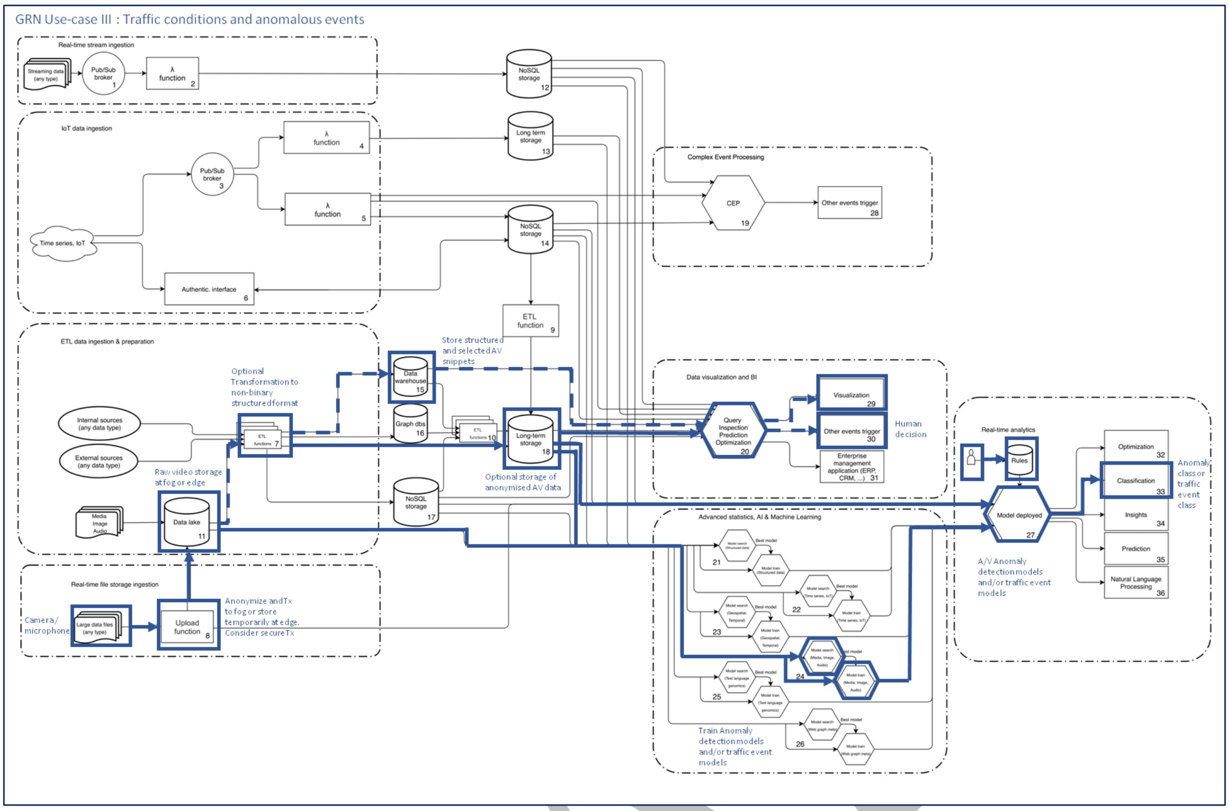

GRN Use-case III : Traffic conditions and anomalous events
Real-time stream ingestion
Streaming data (any type)
Pub/Sub broker 1
λ function 2
NoSQL storage 12
IoT data ingestion
λ function 4
Long term storage 13
Pub/Sub broker 3
λ function 5
NoSQL storage 14
Time series, IoT
Authentic. interface 6
ETL function 9
Complex Event Processing
CEP 19
Other events trigger 28
ETL data ingestion & preparation
Optional Transformation to non-binary structured format
Store structured and selected AV snippets
Data warehouse 15
Graph dbs 16
ETL functions 10
Long-term storage 18
Optional storage of anonymised AV data
Internal sources (any data type)
External sources (any data type)
ETL functions 7
Raw video storage at fog or edge
Media Image Audio
Data lake 11
NoSQL storage 17
Real-time file storage ingestion
Camera / microphone
Large data files (any type)
Upload function 8
Anonymize and Tx to fog or store temporarily at edge. Consider secure Tx
Data visualization and BI
Query Inspection Prediction Optimization 20
Visualization 29
Other events trigger 30
Human decision
Enterprise management application (ERP, CRM, ...) 31
Real-time analytics
Rules
Model deployed 27
Optimization 32
Classification 33
Insights 34
Prediction 35
Natural Language Processing 36
Anomaly class or traffic event class
A/V Anomaly detection models and/or traffic event models
Advanced statistics, AI & Machine Learning
Model search (Structured data)
Best model
Model train (Structured data)
21
Model search (Time series, IoT)
Model train (Time series, IoT)
22
Model search (Geospatial, Temporal)
Model train (Geospatial, Temporal)
23
Model search (Media, Image, Audio)
Model train (Media, Image, Audio)
24
Model search (Text language genomics)
Model train (Text language genomics)
25
Model search (Web graph meta)
Model train (Web graph meta)
26
Train Anomaly detection models and/or traffic event models

Figure 56. Specification of the generic Big Data and AI pipeline blueprint for the GRN3 use case: Traffic conditions and anomalous events

In this use case, traffic conditions, such as light or heavy traffic, and anomalous events, such as anomalous queue lengths and roadside breakdown are detected in the AV data. This use case can be deployed in real-time, for example, to alert traffic control room personnel of anomalous events and can also be used offline to collect statistics on anomalous events. The links to train the anomaly detection and classification models are the same as in *GRN use-case I*.

In the real-time application (after the model is trained), the data is extracted by a camera and microphone, anonymised (8) and transmitted to raw video storage (11) which could be at the fog or edge. This is then loaded (7) where an optional transformation to a non-binary structured format can be made. From here the data is stored and selected into AV snippets (15) and transferred to (20) where the data is inspected and can then be visualised (29) or trigger an event (30) such that a human is alerted.

From (7) the data can be transferred to (18) where there is the option of storing the anonymised AV data which can be used to train the model (24), passed to (20) for data visualisation and BI or to (27) for real-time analytics.

12.3.4 GRN4 use case: Junction traffic trajectory collection

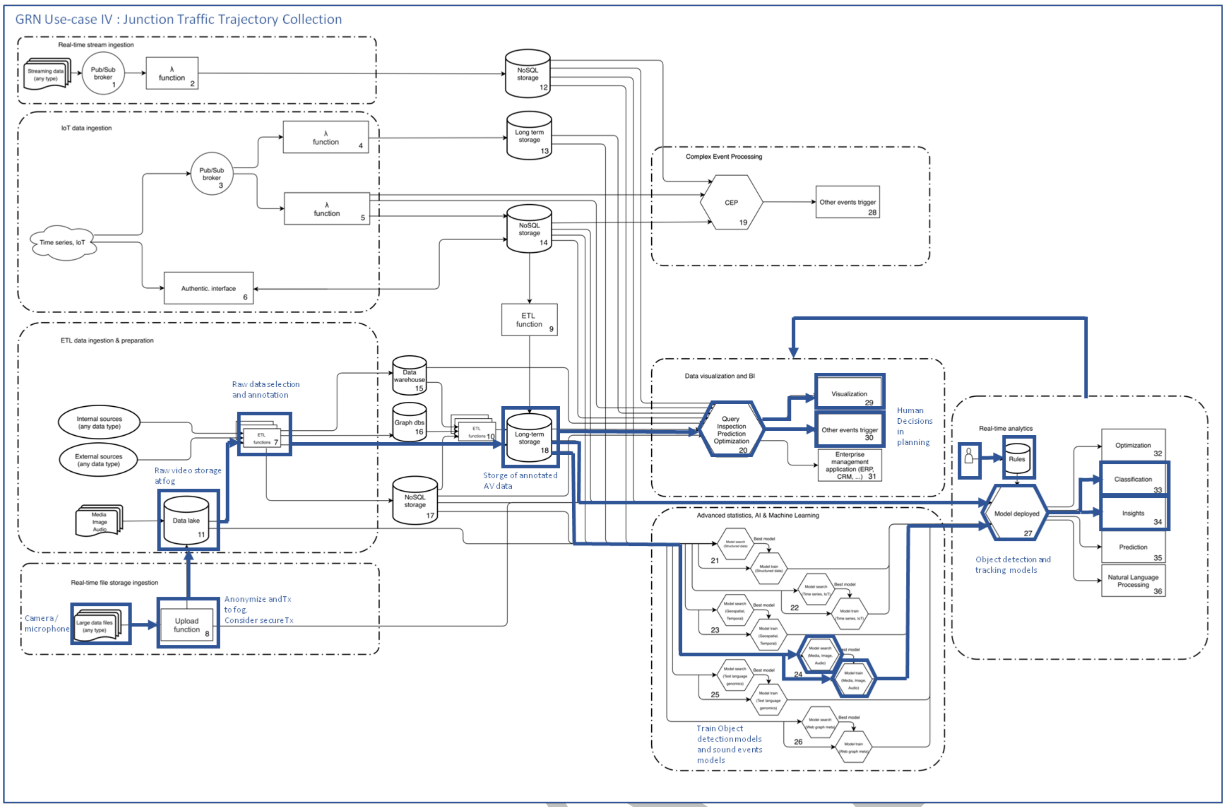

GRN Use-case IV : Junction Traffic Trajectory Collection
Real-time stream ingestion
Streaming data (any type)
Pub/Sub broker 1
λ function 2
NoSQL storage 12
IoT data ingestion
λ function 4
Long term storage 13
Pub/Sub broker 3
λ function 5
Time series, IoT
NoSQL storage 14
Authentic. interface 6
ETL function 9
ETL data ingestion & preparation
Raw data selection and annotation
Data warehouse 15
Graph dbs 16
Internal sources (any data type)
External sources (any data type)
ETL functions 7
ETL functions 10
Long-term storage 18
Raw video storage at fog
Storge of annotated AV data
NoSQL storage 17
Media Image Audio
Data lake 11
Real-time file storage ingestion
Camera / microphone
Large data files (any type)
Upload function 8
Anonymize and Tx to fog. Consider secure Tx
Complex Event Processing
CEP 19
Other events trigger 28
Data visualization and BI
Query Inspection Prediction Optimization 20
Visualization 29
Other events trigger 30
Enterprise management application (ERP, CRM, ...) 31
Human Decisions in planning
Real-time analytics
Rules
Model deployed 27
Optimization 32
Classification 33
Insights 34
Prediction 35
Natural Language Processing 36
Object detection and tracking models
Advanced statistics, AI & Machine Learning
Model search (Structured data)
Best model
Model train (Structured data)
21
Model search (Time series, IoT)
Model train (Time series, IoT)
22
Model search (Geospatial, Temporal)
Model train (Geospatial, Temporal)
23
Model search (Media, Image, Audio)
Model train (Media, Image, Audio)
24
Model search (Text language genomics)
Model train (Text language genomics)
25
Model search (Web graph meta)
Model train (Web graph meta)
26
Train Object detection models and sound events models

Figure 57. Specification of the generic Big Data and AI pipeline blueprint for the GRN4 use case: Junction traffic trajectory collection

Here the application is intended to collect long-term data for long-term decision-making, for example in planning infrastructure upgrades or to study how vulnerable road users make use of facilities provided to them. The data is extracted from a camera and microphone setup, anonymised and transmitted to the fog (8). The video is stored in its raw format (11). Optionally, this raw data can be selected and annotated (7) and moved into long-term storage (18) for model training.

From here the data can be used to train the object detection models and sound event detection models (24) and then use in (27). After the model has been trained the link can be directly from (18) to (27) and used to query (20) the database for the purpose of collecting statistics. The long-term data can also be visualised (29) and used by humans in long term decision making.

12.3.5 MT1 use case: Monitoring of crowded areas

In general, for the MT use cases the architectural blueprint is more or less the same for the first 3 cases, MT1-3, and for MT4 a slight change on the blueprint has been introduced to reflect the specificities of this use case.

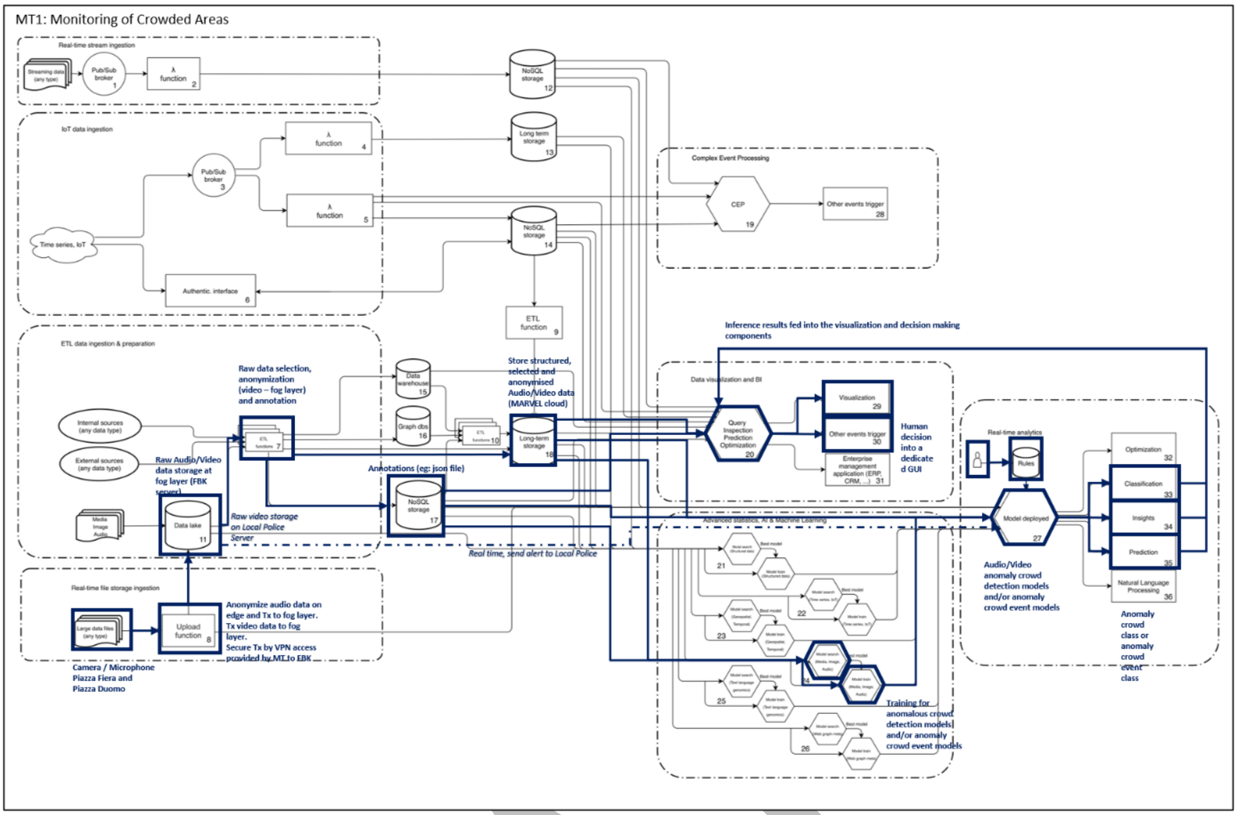

MT1: Monitoring of Crowded Areas
Real-time stream ingestion
Streaming data (any type)
Pub/Sub broker 1
λ function 2
NoSQL storage 12
IoT data ingestion
λ function 4
Long term storage 13
Pub/Sub broker 3
λ function 5
Time series, IoT
NoSQL storage 14
Authentic. interface 6
ETL function 9
ETL data ingestion & preparation
Raw data selection, anonymization (video – fog layer) and annotation
Internal sources (any data type)
External sources (any data type)
ETL functions 7
Data warehouse 15
Graph dbs 16
ETL functions 10
Store structured, selected and anonymised Audio/Video data (MARVEL cloud)
Long-term storage 18
Raw Audio/Video data storage at fog layer (FBK server)
Annotations (eg: json file)
NoSQL storage 17
Media Image Audio
Data lake 11
Raw video storage on Local Police Server
Real time, send alert to Local Police
Real-time file storage ingestion
Large data files (any type)
Upload function 8
Anonymize audio data on edge and Tx to fog layer. Tx video data to fog layer. Secure Tx by VPN access provided by MT to FBK
Camera / Microphone Piazza Fiera and Piazza Duomo
Complex Event Processing
CEP 19
Other events trigger 28
Inference results fed into the visualization and decision making components
Data visualization and BI
Query Inspection Prediction Optimization 20
Visualization 29
Other events trigger 30
Enterprise management application (ERP, CRM, ...) 31
Human decision into a dedicated GUI
Real-time analytics
Rules
Model deployed 27
Optimization 32
Classification 33
Insights 34
Prediction 35
Natural Language Processing 36
Audio/Video anomaly crowd detection models and/or anomaly crowd event models
Anomaly crowd class or anomaly crowd event class
Advanced statistics, AI & Machine Learning
Model search (Structured data)
Best model
Model train (Structured data)
21
Model search (Time series, IoT)
Model train (Time series, IoT)
22
Model search (Geospatial, Temporal)
Model train (Geospatial, Temporal)
23
Model search (Media, Image, Audio)
Model train (Media, Image, Audio)
24
Model search (Text language genomics)
Model train (Text language genomics)
25
Model search (Web graph meta)
Model train (Web graph meta)
26
Training for anomalous crowd detection models and/or anomaly crowd event models

Figure 58. Specification of the generic Big Data and AI pipeline blueprint for the MT1 use case: Monitoring of crowded areas

The routes are divided into two different situations: the normal lines represent the phase of training and retraining based on some selected parts of videos, instead, the dashed lines represent the phase of the real-time monitoring of crowded areas, in particular, Piazza Fiera and Piazza Duomo, the main cities of Trento City Center.

During the phase of training, videos and audio will be collected from the cameras in Piazza Duomo and Piazza Fiera. Due to privacy audio will be anonymised on the edge in order to remove speech and other sounds that cannot be collected. The upload function (8) is a secure transmission by VPN Access between MT and FBK in which anonymised audio and raw video will be sent to the data lake (11a) in FBK. In parallel, the Local Police have to store the raw video in the current surveillance system (11b) and data will be deleted after 7 days (this by regulation on the investigation). Since the videos also contain parts that are not interesting for the use case, videos are cut and then anonymised by using ETL function (7) and at the same time video will be annotated (partially automated process). The annotations will be produced in JSON format (or other standard format selected by the MARVEL partners) and stored in a no-SQL DB (17). The selected videos and audio are stored in long-term storage (18) provided by PSNC inside the MARVEL framework.

At this point, we have videos and audio with annotations, and they can be used for developing the best model for detecting anomalous crowding situations (24a) and also for the training (24b). The model now must be deployed (27) by ingesting the annotated videos and audios and the model created. The results of the deployment will produce a classification of the event selected (33), insights (34) related to the number of events, and other information requested by Local Police and prediction (35) in order to understand if a possible crowed situation is dangerous or not.

All the outputs will be used in the phase of Query Inspection (20) in other to visualise the results of classification, insights and prediction. The visualisation will be done in ad-hoc GUI interfaces in which Local Police and Policy Maker can access and view the results.

Let's see now the path for the real-time analysis (that will be done during the project duration). Data from cameras and microphones will be uploaded in the data lake of the local police (11b) and at the same time video will be directly sent to the model deployment/ingestion (27). If a dangerous situation will happen, in the dedicated GUI (29) an alarm will be triggered and the video will be shown, also because the Local Police server has stored the registration.

12.3.6 MT2 use case: Detecting criminal and antisocial behaviour

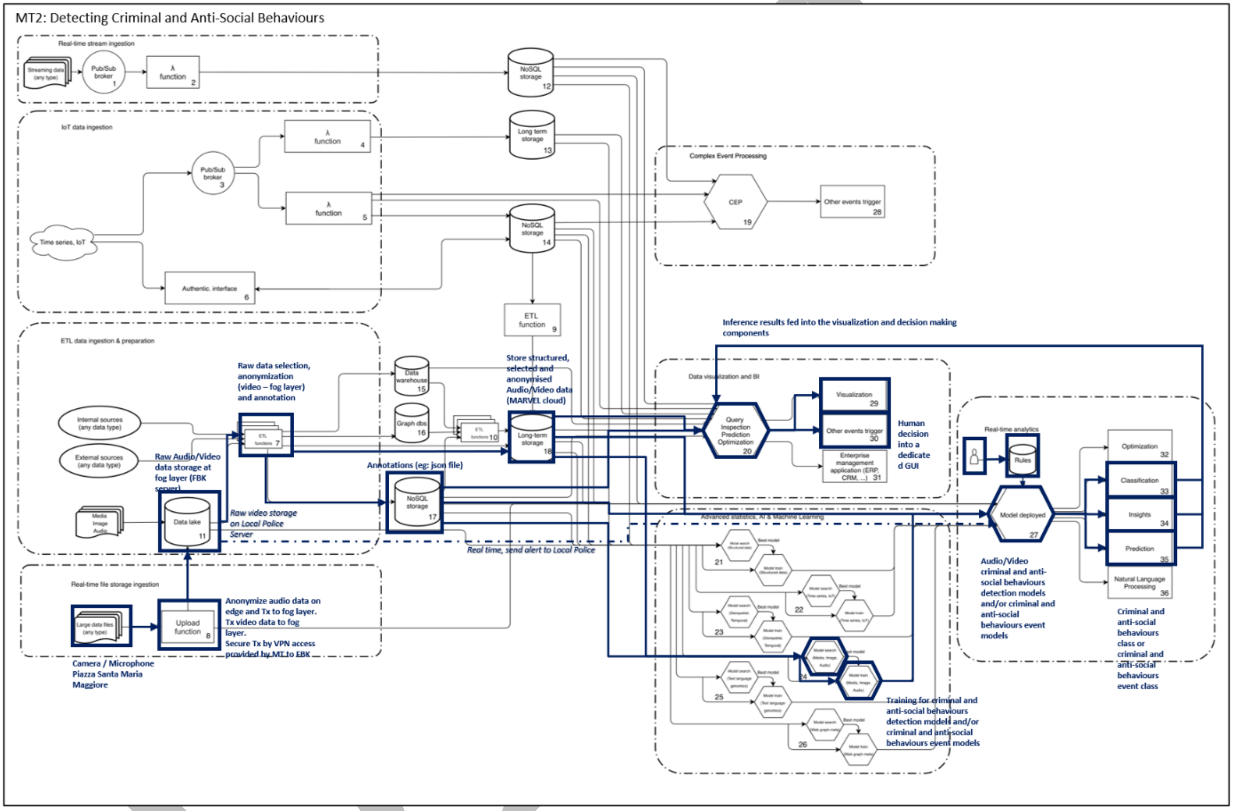

MT2: Detecting Criminal and Anti-Social Behaviours
Real-time stream ingestion
Streaming data (any type)
Pub/Sub broker 1
λ function 2
NoSQL storage 12
IoT data ingestion
λ function 4
Long term storage 13
Pub/Sub broker 3
λ function 5
NoSQL storage 14
Time series, IoT
Authentic. interface 6
Complex Event Processing
CEP 19
Other events trigger 28
ETL function 9
ETL data ingestion & preparation
Raw data selection, anonymization (video – fog layer) and annotation
Data warehouse 15
Graph dbs 16
Store structured, selected and anonymised Audio/Video data (MARVEL cloud)
Internal sources (any data type)
External sources (any data type)
ETL functions 7
ETL functions 10
Long-term storage 18
Raw Audio/Video data storage at fog layer (FBK server)
Annotations (eg: json file)
NoSQL storage 17
Media Image Audio
Data lake 11
Raw video storage on Local Police Server
Real time, send alert to Local Police
Real-time file storage ingestion
Large data files (any type)
Upload function 8
Anonymize audio data on edge and Tx to fog layer. Tx video data to fog layer. Secure Tx by VPN access provided by MT to FBK
Camera / Microphone Piazza Santa Maria Maggiore
Inference results fed into the visualization and decision making components
Data visualization and BI
Query Inspection Prediction Optimization 20
Visualization 29
Other events trigger 30
Enterprise management application (ERP, CRM, ...) 31
Human decision into a dedicated GUI
Real-time analytics
Rules
Model deployed 27
Optimization 32
Classification 33
Insights 34
Prediction 35
Natural Language Processing 36
Advanced statistics, AI & Machine Learning
Model search (Structured data)
Best model
Model train (Structured data)
21
Model search (Time series, IoT)
Model train (Time series, IoT)
22
Model search (Geospatial, Temporal)
Model train (Geospatial, Temporal)
23
Model search (Media, Image, Audio)
Model train (Media, Image, Audio)
24
Model search (Text language genomics)
Model train (Text language genomics)
25
Model search (Web graph meta)
Model train (Web graph meta)
26
Audio/Video criminal and anti-social behaviours detection models and/or criminal and anti-social behaviours event models
Criminal and anti-social behaviours class or criminal and anti-social behaviours event class
Training for criminal and anti-social behaviours detection models and/or criminal and anti-social behaviours event models

Figure 59. Specification of the generic Big Data and AI pipeline blueprint for the MT2 use case: Detecting criminal and antisocial behaviour

The routes are divided into two different situations: the normal lines represent the phase of training and retraining based on some selected parts of videos, instead, the dashed lines represent the phase of the real-time analysis for detecting criminal and anti-social behaviours in Piazza Santa Maria Maggiore in the Trento City Center in which the number of criminal acts has increased in the recent years.

During the phase of training, videos and audio will be collected from the cameras in Piazza Duomo and Piazza Fiera. Due to privacy audio will be anonymised on the edge in order to remove speech and other sounds that cannot be collected. The upload function (8) is a secure transmission by VPN Access between MT and FBK in which anonymised audio and raw video will be sent to the data lake (11a) in FBK. In parallel, the Local Police have to store the raw video in the current surveillance system (11b) and data will be deleted after 7 days (this by regulation on the investigation). Since the videos also contain parts that are not interesting for the use case, videos are cut and then anonymised by using ETL function (7) and at the same time video will be annotated (partially automated process). The annotations will be produced in JSON format (or other standard format selected by the Marvel partners) and stored in a no-SQL DB (17). The selected videos and audio are stored in a long-term storage (18) provided by PSNC inside the MARVEL framework.

At this point, we have videos and audio with annotations, and they can be used for developing the best model for detecting criminal and anti-social behaviours (24a) and also for the training (24b). The model now must be deployed (27) by ingesting the annotated videos and audios and the model created. The results of the deployment will produce a classification of the event selected (33), insights (34) related to the number of events, and other information requested by Local Police and prediction (35) in order to understand if a possible crowed situation is dangerous or not.

All the outputs will be used in the phase of Query Inspection (20) in other to visualise the results of classification, insights, and prediction. The visualisation will be done in ad-hoc GUI interfaces in which Local Police and Policy Maker can access and view the results.

Let's see now the path for the real-time analysis (that will be done during the project duration). Data from cameras and microphones will be uploaded in the data lake of the local police (11b) and at the same time video will be directly sent to the model deployment/ingestion (27). If a dangerous situation will happen, in the dedicated GUI (29) an alarm will be triggered and the video will be shown, also because the Local Police server has stored the registration.

12.3.7 MT3 use case: Monitoring of parking places

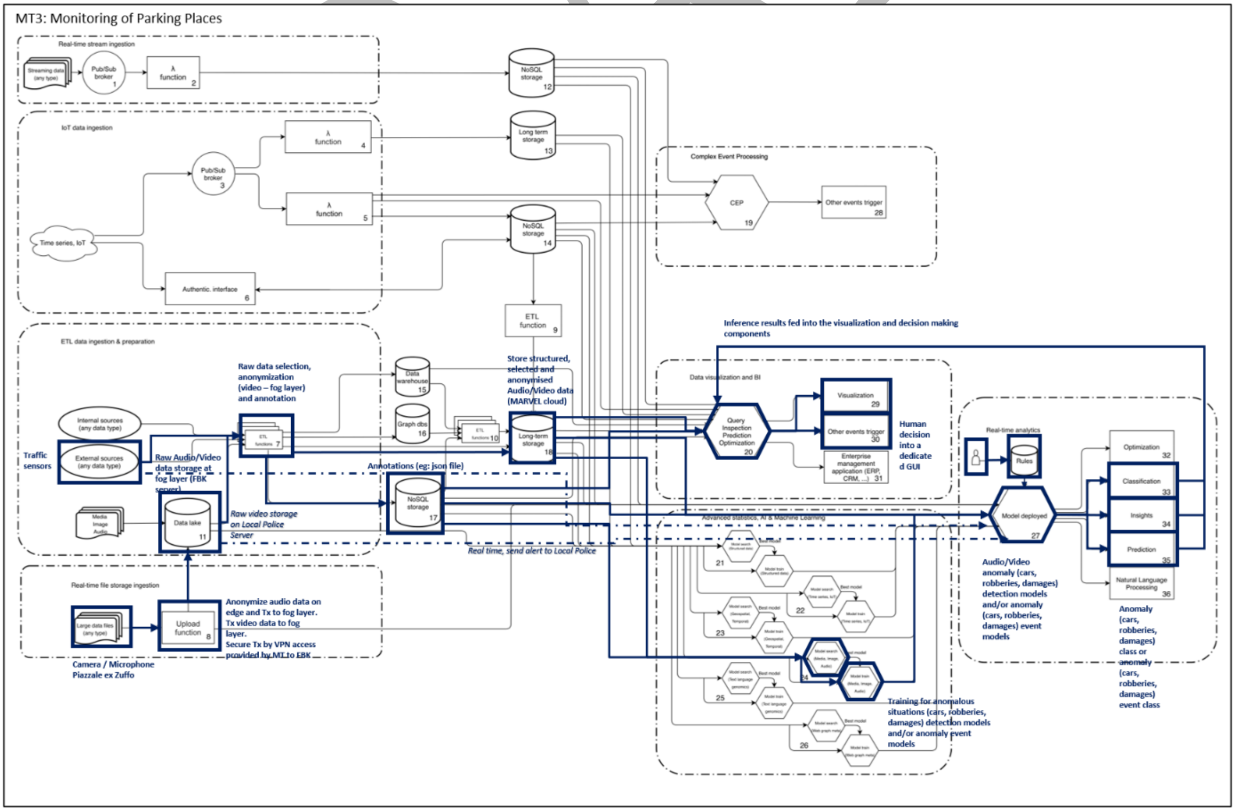

MT3: Monitoring of Parking Places
Real-time stream ingestion
Streaming data (any type)
Pub/Sub broker 1
λ function 2
NoSQL storage 12
IoT data ingestion
λ function 4
Long term storage 13
Pub/Sub broker 3
λ function 5
NoSQL storage 14
Time series, IoT
Authentic. interface 6
ETL function 9
Complex Event Processing
CEP 19
Other events trigger 28
ETL data ingestion & preparation
Raw data selection, anonymization (video – fog layer) and annotation
Data warehouse 15
Graph dbs 16
Store structured, selected and anonymised Audio/Video data (MARVEL cloud)
Internal sources (any data type)
Traffic sensors
External sources (any data type)
ETL functions 7
ETL functions 10
Long-term storage 18
Raw Audio/Video data storage at fog layer (FBK server)
Annotations (eg: json file)
Media Image Audio
Data lake 11
Raw video storage on Local Police Server
NoSQL storage 17
Real time, send alert to Local Police
Real-time file storage ingestion
Large data files (any type)
Upload function 8
Anonymize audio data on edge and Tx to fog layer. Tx video data to fog layer. Secure Tx by VPN access provided by MT to FBK
Camera / Microphone Piazzale ex Zuffo
Inference results fed into the visualization and decision making components
Data visualization and BI
Query Inspection Prediction Optimization 20
Visualization 29
Other events trigger 30
Enterprise management application (ERP, CRM, ...) 31
Human decision into a dedicate d GUI
Real-time analytics
Rules
Model deployed 27
Optimization 32
Classification 33
Insights 34
Prediction 35
Natural Language Processing 36
Audio/Video anomaly (cars, robberies, damages) detection models and/or anomaly (cars, robberies, damages) event models
Anomaly (cars, robberies, damages) class or anomaly (cars, robberies, damages) event class
Advanced statistics, AI & Machine Learning
Model search (Structured data)
Best model
Model train (Structured data)
21
Model search (Time series, IoT)
Model train (Time series, IoT)
22
Model search (Geospatial, Temporal)
Model train (Geospatial, Temporal)
23
Model search (Media, Image, Audio)
Model train (Media, Image, Audio)
24
Model search (Text language genomics)
Model train (Text language genomics)
25
Model search (Web graph meta)
Model train (Web graph meta)
26
Training for anomalous situations (cars, robberies, damages) detection models and/or anomaly event models

Figure 60. Specification of the generic Big Data and AI pipeline blueprint for the MT3 use case: Monitoring of parking places

The routes are divided into two different situations: the normal lines represent the phase of training and retraining based on some selected parts of videos, instead, the dashed lines represent the phase of the real-time analysis for detecting robberies or damages to the cars parked in Park Zuffo in the Trento City Center. Moreover, an analysis on the usage of the parking occupancy also in dedicated parks for disabled people and load/unload will be done in the same parking zone.

During the phase of training, videos and audio will be collected from the cameras in Park Zuffo. Due to privacy audio will be anonymised on the edge in order to remove speech and other sounds that cannot be collected. The upload function (8) is a secure transmission by VPN Access between MT and FBK in which anonymised audio and raw video will be sent to the data lake (11a) in FBK. In parallel, the Local Police have to store the raw video in the current surveillance system (11b) and data will be deleted after 7 days (this by regulation on the investigation). Since the videos also contain parts that are not interesting for the use case, videos are cut and then anonymised by using ETL function (7) and at the same time video will be annotated (partially automated process). In parallel data coming from the traffic sensors are ingested into the ETL function (7) in order to evaluate the number of cars inside the parking area. The annotations will be produced in JSON format (or other standard format selected by the Marvel partners) and stored in a no-SQL DB (17). The selected videos and audio are stored in long-term storage (18) provided by PSNC inside the MARVEL framework.

At this point, we have videos and audio with annotations, and they can be used for developing the best model for detecting criminal and anti-social behaviours (24a) and also for the training (24b). The model now must be deployed (27) by ingesting the annotated videos and audios and the model created. The results of the deployment will produce a classification of the event selected (33), insights (34) related to the number of events, and other information requested by Local Police and prediction (35).

All the outputs will be used in the phase of Query Inspection (20) in other to visualise the results of classification, insights, and prediction. The visualisation will be done in ad-hoc GUI interfaces in which Local Police and Policy Maker can access and view the results.

Let’s see now the path for the real-time analysis (that will be done during the project duration). Data from cameras and microphones will be uploaded in the data lake of the local police (11b) and at the same time video will be directly sent to the model deployment/ingestion (27) that returns in output information for the optimisation (32), classification (33) such as the number of cars parked in the disabled slots, insights (34) and prediction (35) in order to calculate for example the occupancy in determined hour or day. Also, data coming from the traffic sensors will be sent directly to the model deployment/ingestion (27). If a dangerous situation will happen, in the dedicated GUI (29) an alarm will be triggered and the video will be shown, also because the Local Police server has stored the registration. Moreover, information related to an occupation on the parking areas and other data can be viewed in a dedicated dashboard.

12.3.8 MT4 use case: Analysis of a specific area

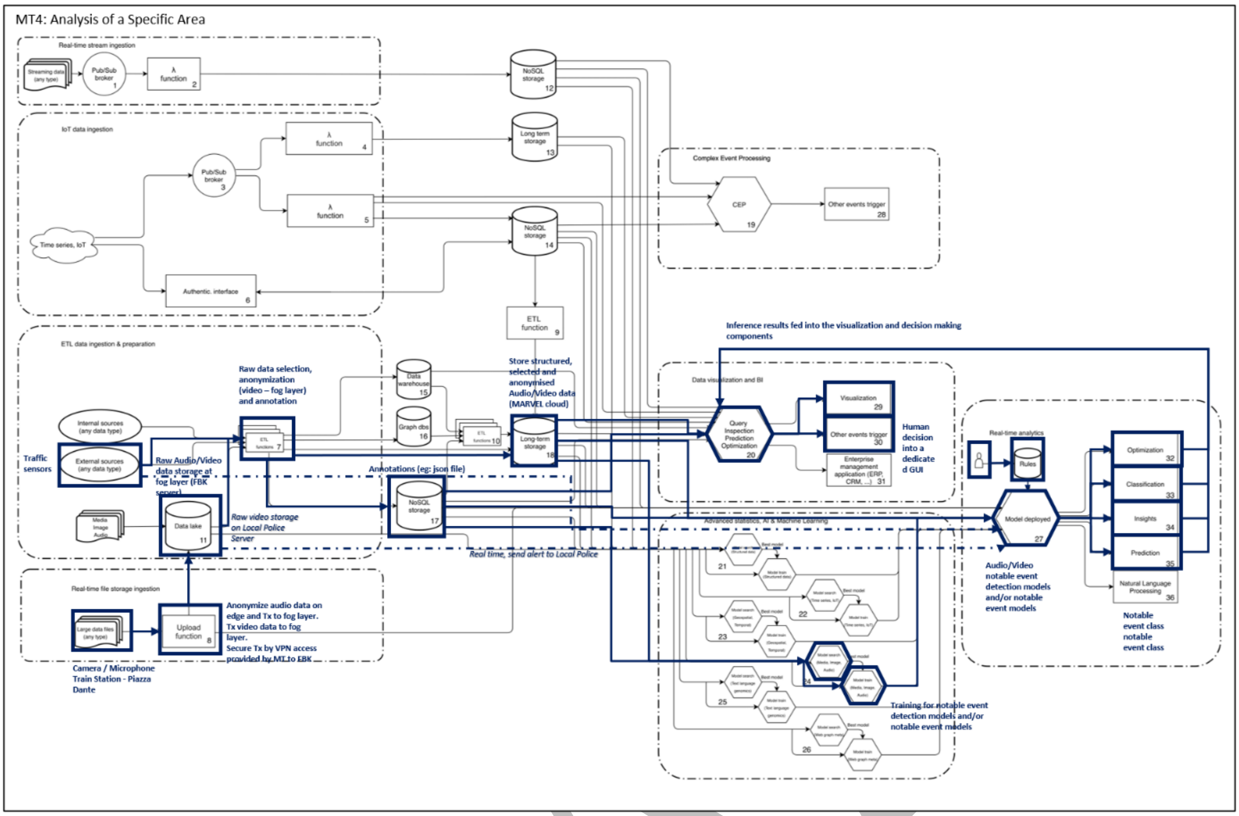

MT4: Analysis of a Specific Area
Real-time stream ingestion
Streaming data (any type)
Pub/Sub broker 1
λ function 2
IoT data ingestion
Pub/Sub broker 3
λ function 4
λ function 5
Time series, IoT
Authentic. interface 6
NoSQL storage 12
Long term storage 13
NoSQL storage 14
ETL function 9
ETL data ingestion & preparation
Raw data selection, anonymization (video – fog layer) and annotation
Internal sources (any data type)
Traffic sensors
External sources (any data type)
ETL functions 7
Raw Audio/Video data storage at fog layer (FBK server)
Media Image Audio
Data lake 11
Raw video storage on Local Police Server
Data warehouse 15
Graph dbs 16
ETL functions 10
Store structured, selected and anonymised Audio/Video data (MARVEL cloud)
Long-term storage 18
Annotations (eg: json file)
NoSQL storage 17
Real time, send alert to Local Police
Real-time file storage ingestion
Large data files (any type)
Upload function 8
Anonymize audio data on edge and Tx to fog layer. Tx video data to fog layer. Secure Tx by VPN access provided by MT to FBK
Camera / Microphone Train Station - Piazza Dante
Complex Event Processing
CEP 19
Other events trigger 28
Inference results fed into the visualization and decision making components
Data visualization and BI
Query Inspection Prediction Optimization 20
Visualization 29
Other events trigger 30
Enterprise management application (ERP, CRM, ...) 31
Human decision into a dedicated GUI
Real-time analytics
Rules
Model deployed 27
Optimization 32
Classification 33
Insights 34
Prediction 35
Natural Language Processing 36
Audio/Video notable event detection models and/or notable event models
Notable event class notable event class
Advanced statistics, AI & Machine Learning
Model search (Structured data)
Best model
Model train (Structured data)
21
Model search (Time series, IoT)
Model train (Time series, IoT)
22
Model search (Geospatial, Temporal)
Model train (Geospatial, Temporal)
23
Model search (Media, Image, Audio)
Model train (Media, Image, Audio)
24
Model search (Text language genomics)
Model train (Text language genomics)
25
Model search (Web graph meta)
Model train (Web graph meta)
26
Training for notable event detection models and/or notable event models

Figure 61. Specification of the generic Big Data and AI pipeline blueprint for the MT4 use case: Analysis of a specific area

The routes are divided into two different situations: the normal lines represent the phase of training and retraining based on some selected parts of videos, instead, the dashed lines represent the phase of the real-time analysis for collecting data (number of persons, cars, trajectories, events) in Piazza Dante in front of the train station of Trento.

During the phase of training, videos and audio will be collected from the cameras in Piazza Dante and the surrounding areas. Due to privacy audio will be anonymised on the edge in order to remove speech and other sounds that cannot be collected. The upload function (8) is a secure transmission by VPN Access between MT and FBK in which anonymised audio and raw video will be sent to the data lake (11a) in FBK. In parallel, the Local Police have to store the raw video in the current surveillance system (11b) and data will be deleted after 7 days (this by regulation on the investigation). Since the videos also contain parts that are not interesting for the use case, videos are cut and then anonymised by using ETL function (7) and at the same time video will be annotated (partially automated process). In parallel data coming from the traffic sensors are ingested into the ETL function (7) in order to evaluate the number of cars inside the parking area. The annotations will be produced in JSON format (or other standard format selected by the Marvel partners) and stored in a no-SQL DB (17). The selected videos and audio are stored in long-term storage (18) provided by PSNC inside the MARVEL framework.

At this point, we have videos and audio with annotations, and they can be used for developing the best model for detecting criminal and anti-social behaviours (24a) and also for the training (24b). The model now must be deployed (27) by ingesting the annotated videos and audios and the model created. The results of the deployment will produce a classification of the event selected (33), insights (34) related to the number of events, and other information requested by Local Police and Policy Makers and prediction (35) in order to take the appropriate decisions.

All the outputs will be used in the phase of Query Inspection (20) in other to visualise the results of classification, insights, and prediction. The visualisation will be done in ad-hoc GUI interfaces in which Local Police and Policy Maker can access and view the results.

Let's see now the path for the real-time analysis (that will be done during the project duration). Data from cameras and microphones will be uploaded in the data lake of the local police (11b) and at the same time video will be directly sent to the model deployment/ingestion (27) that returns in output information for the optimisation (32), classification (33) such as the number of cars parked in the disabled slots, insights (34) and prediction (35) in order to calculate for example the occupancy in determined hour or day. Also, data coming from the traffic sensors will be sent directly to the model deployment/ingestion (27). If a dangerous situation will happen, in the dedicated GUI (29) an alarm will be triggered and the video will be shown, also because the Local Police server has stored the registration. Moreover, information related to cars passage, trajectories, number of pedestrians, etc. can be viewed in a dedicated dashboard.

12.3.9 UNS1 use case: Drone experiment

Figure 62 presents the instance of the generic Big Data and AI pipeline blueprint for the UNS Drone experiment use case. The blueprint and the accompanying description are only indicative and are to be revised following the developments and specific technical approaches within individual MARVEL components and subsystems that are relevant for this use case.

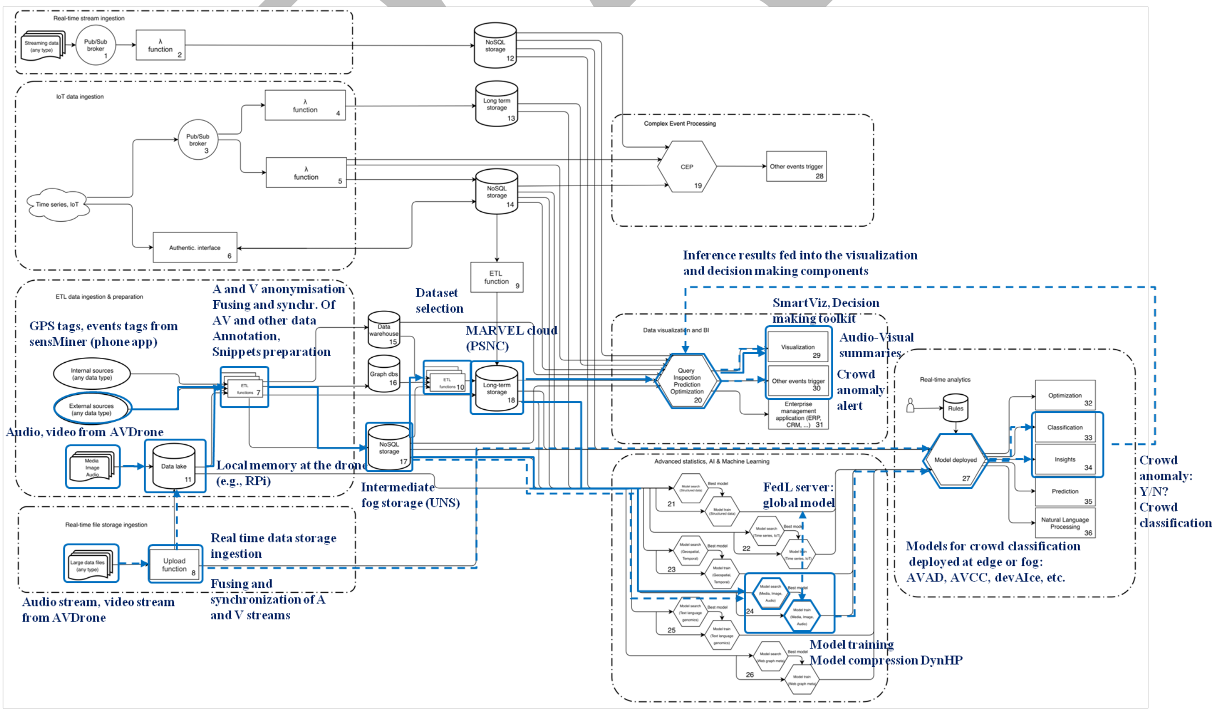

Real-time stream ingestion
Streaming data (any type)
Pub/Sub broker 1
λ function 2
NoSQL storage 12
IoT data ingestion
Pub/Sub broker 3
λ function 4
λ function 5
Long term storage 13
NoSQL storage 14
Time series, IoT
Authentic. interface 6
ETL function 9
Complex Event Processing
CEP 19
Other events trigger 28
ETL data ingestion & preparation
GPS tags, events tags from sensMiner (phone app)
A and V anonymisation Fusing and synchr. Of AV and other data Annotation, Snippets preparation
Internal sources (any data type)
External sources (any data type)
ETL functions 7
Data warehouse 15
Graph dbs 16
Dataset selection
MARVEL cloud (PSNC)
ETL functions 10
Long-term storage 18
Audio, video from AVDrone
Media Image Audio
Data lake 11
Local memory at the drone (e.g., RPi)
NoSQL storage 17
Intermediate fog storage (UNS)
Real-time file storage ingestion
Large data files (any type)
Upload function 8
Real time data storage ingestion
Fusing and synchronization of A and V streams
Audio stream, video stream from AVDrone
Inference results fed into the visualization and decision making components
Data visualization and BI
SmartViz, Decision making toolkit
Query Inspection Prediction Optimization 20
Visualization 29
Other events trigger 30
Enterprise management application (ERP, CRM, ...) 31
Audio-Visual summaries
Crowd anomaly alert
Real-time analytics
Rules
Model deployed 27
Optimization 32
Classification 33
Insights 34
Prediction 35
Natural Language Processing 36
Crowd anomaly: Y/N? Crowd classification
Models for crowd classification deployed at edge or fog: AVAD, AVCC, devAIce, etc.
Advanced statistics, AI & Machine Learning
Model search (Structured data)
Best model
Model train (Structured data)
21
FedL server: global model
Model search (Time series, IoT)
Model train (Time series, IoT)
22
Model search (Geospatial, Temporal)
Model train (Geospatial, Temporal)
23
Model search (Media, Image, Audio)
Model train (Media, Image, Audio)
24
Model search (Text language genomics)
Model train (Text language genomics)
25
Model search (Web graph meta)
Model train (Web graph meta)
26
Model training Model compression DynHP

Figure 62. Specification of the generic Big Data and AI pipeline blueprint for the UNS1 use case: Drone experiment

In the UNS Drone experiment, we differentiate between two workflows: (i) workflow for data collection and model building, indicated by full lines/arrows; and (ii) workflow for inference, decision-making, and user interaction, indicated by dashed lines/arrows.

In the workflow for data collection and model building, the use case will collect data using AVDrone hardware setup, with drone-mounted cameras and microphones, and additional ground-based AV capturing, together with the sensMiner Android application for GPS data and (coarse) events tags. This is indicated by the two bottom fields marked in blue in the *ETL data ingestion and preparation* block of the blueprint. The AV data is collected in the local memory of the drone - field 11 of the blueprint ("Data lake"), and then transmitted wirelessly for later processing in field 7 ("ETL functions"). Processing that occurs in field 7 consists of several steps. First, the batches of AV and other data are anonymised and then fused and synchronised. Second, the snippets of targeted events are extracted from the data and appropriately annotated, taking into account the needs of ML/DL models to be trained. The data is then stored at the local storage of UNS (fog), represented by field 17 of the blueprint ("NoSQL storage"). A selection of the training data is also stored at the MARVEL cloud, enabled by PSNC – represented by field 18 ("Long-term storage"). From the long-term storage 18, the data is consumed by MARVEL visualisation components (SmartViz) – represented by fields 20 ("Query Inspection") and 29 ("Visualisation") of the *Data visualisation and BI* block; this for example includes advanced visualisations for offline forensics, such as audio-visual summaries. The training data from 17 is fed into the models to be trained in field 24 of the *Advanced statistics, AI & Machine Learning* block of the blueprint; both sub-fields of 24 are relevant to the use case: "model train (Video, Audio)" – for training a given model, and then subsequently – or simultaneously (as in DynHP), performing "Model search (Video, Audio)" – for model compression. The trained model is then deployed at the edge (drone-mounted RPi or Intel NUC) or fog (UNS server).

In the workflow for inference, decision-making and user interaction, the flow starts with the real-time ingestion of AV data streams from the AVDrone, including also preprocessing for time alignment/synchronisation. This is represented by the first two fields marked in blue of the *Real-time storage ingestion* block of the blueprint – i.e., "Large data files (any type)" and field 8 ("Upload function"). The flow then branches from 8 to fields 11 and 27. The flow from 8 to 27 indicates the data that is fed into deployed models for real-time inference in fields 33 ("Classification"/Anomaly detection") and 34 ("Insights") of the *Real-time analytics* block of the blueprint. The obtained inference results are fed into the backend of MARVEL's visualisation and decision-making toolkit, enabled by SmartViz, for raising real-time alerts and customised visualisations; this is indicated by fields 20, 29, and 30 ("Other events trigger") of the *Data visualisation and BI* block of the blueprint. The second branch of the real-time AV data flow goes from 8 to 11 and then passes through the same processing stages as described above in the case of the model training workflow. Finally, annotated audio-visual snippets are transferred from 17 to 24 for online training using federated learning (FedL). In addition to the AV data flow from AVDrone (UNS), models in the Drone experiment use case will be benefitting from partners' datasets where similar crowd-related events are observed/available (e.g., as in the MT1 use case – Crowd monitoring)., enabled by the federated learning (FedL). This is achieved by ingesting the FedL server model parameters into the local training, updating the model based on the local data, and uploading the newly optimised model parameters to the FedL server, to iterate until the desired accuracy is reached. The described process is indicated by the bidirectional arrow from the "Model train" sub-field of 24 to the "FedL server: global model" textbox of the figure.

12.3.10 UNS2 use case: Audio-visual emotion recognition

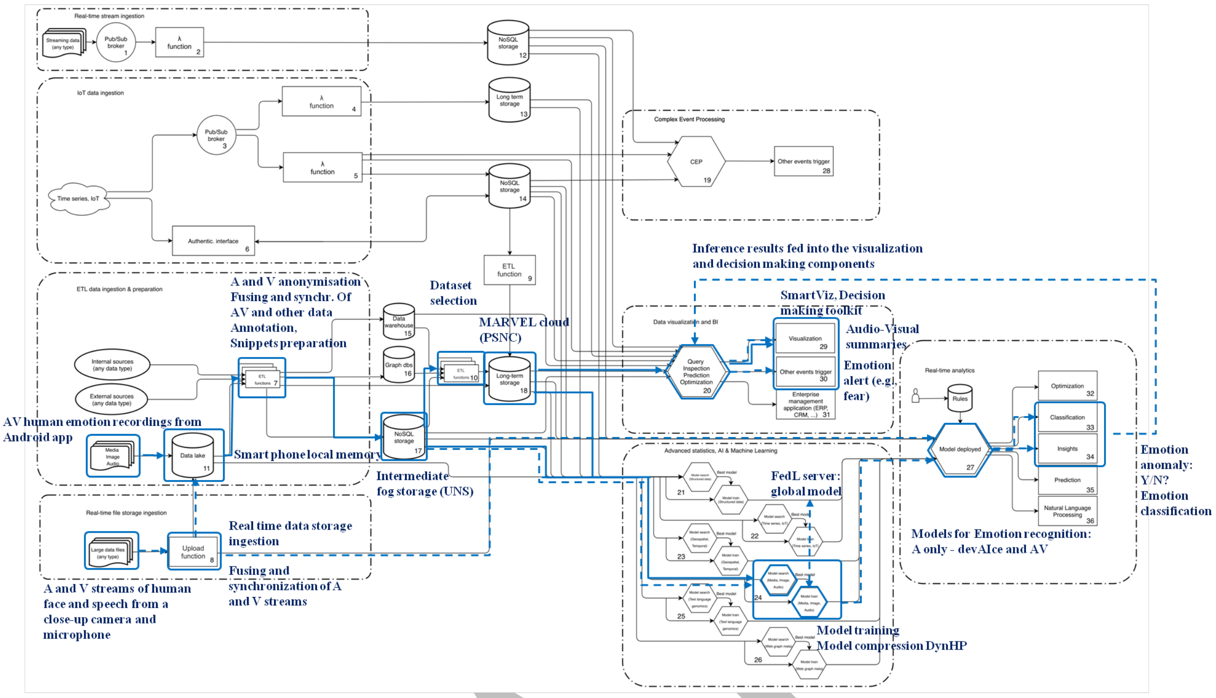

Real-time stream ingestion
Streaming data (any type)
Pub/Sub broker 1
λ function 2
NoSQL storage 12
IoT data ingestion
λ function 4
Long term storage 13
Pub/Sub broker 3
λ function 5
NoSQL storage 14
Time series, IoT
Authentic. interface 6
ETL function 9
Complex Event Processing
CEP 19
Other events trigger 28
Inference results fed into the visualization and decision making components
ETL data ingestion & preparation
A and V anonymisation Fusing and synchr. Of AV and other data Annotation, Snippets preparation
Dataset selection
Data warehouse 15
MARVEL cloud (PSNC)
Graph dbs 16
Internal sources (any data type)
External sources (any data type)
ETL functions 7
ETL functions 10
Long-term storage 18
AV human emotion recordings from Android app
Media Image Audio
Data lake 11
Smart phone local memory
NoSQL storage 17
Intermediate fog storage (UNS)
Real-time file storage ingestion
Large data files (any type)
Upload function 8
Real time data storage ingestion
Fusing and synchronization of A and V streams
A and V streams of human face and speech from a close-up camera and microphone
SmartViz, Decision making toolkit
Data visualization and BI
Query Inspection Prediction Optimization 20
Visualization 29
Other events trigger 30
Enterprise management application (ERP, CRM, ...) 31
Audio-Visual summaries
Emotion alert (e.g., fear)
Real-time analytics
Rules
Optimization 32
Classification 33
Insights 34
Prediction 35
Natural Language Processing 36
Model deployed 27
Emotion anomaly: Y/N? Emotion classification
Advanced statistics, AI & Machine Learning
Model search (Structured data)
Best model
Model train (Structured data)
21
FedL server: global model
Model search (Time series, IoT)
Model train (Time series, IoT)
22
Model search (Geospatial, Temporal)
Model train (Geospatial, Temporal)
23
Model search (Media, Image, Audio)
Model train (Media, Image, Audio)
24
Model search (Text language genomics)
Model train (Text language genomics)
25
Model search (Web graph meta)
Model train (Web graph meta)
26
Model training Model compression DynHP
Models for Emotion recognition: A only - devAIce and AV

Figure 63. Specification of the generic Big Data and AI pipeline blueprint for the UNS2 use case: Audio-visual emotion recognition

Figure 63 shows the instance of the generic AI and Big Data pipeline blueprint for the UNS Emotion recognition use case. The two workflows are very similar as in the Drone experiment use case, with the distinctions on the edge layer and the models and visualisation components. Specifically, for the model building workflow, the edge layer consists of a mobile phone with a custom-made Android app for close-up recordings of human face and speech, while the experimental subject is reading a predefined text under a predefined emotion. The edge layer for the inference workflow consists of a desktop camera and microphone, connected to a PC (laptop or desktop). The models to be built and consecutively deployed are emotion classification ML/DL models. Similarly, visualisation and decision-making components will be customised to raise alerts and the accompanying visualisations when the inference components detect emotions pre-labelled as "anomalous" (e.g., anger, fear).

13 Concluding remarks

This deliverable described in detail the refined specification of the MARVEL conceptual E2F2C architecture grounded on a thorough understanding of the underlying technologies, an updated state-of-the-art review in the relevant project areas, alignment with relevant reference architectures and models, as well as on end-user requirements.

In the process of defining the MARVEL framework and architecture, we first reviewed and analysed functional and non-functional end-user requirements collected in D1.2 – 'MARVEL's Experimental protocol'. This approach allowed us to clearly identify how each of the MARVEL components and their mutual interactions, map to and address the requirements, explaining their roles within the overall framework. Based on their functional similarity, MARVEL components are organised into seven subsystems. Within the identified subsystems, each of the components was then described in full detail, including explanations of their inner workings, inbound and outbound interfaces, and accompanying illustrative figures. Subsystems are described as a synergy of the participating components, focusing on high-level subsystem roles.

User interactions and user interface are addressed both for the overall framework and for each component separately, including component instantiation (configuration and initialization) and also access rights and procedures (authentication and authorisation).

The deliverable presents the current and the expected TRLs and outlines the key innovations across all MARVEL components, serving as an innovations roadmap for the MARVEL framework. The content found in this document aims to guide partners towards the realisation, integration, and deployment of the MARVEL framework and its application in the MARVEL use cases. Towards future deployments that initiate with the Minimum Viable Product (MVP) at M12, this deliverable provides architecture specification for each of the MARVEL use cases. Variations in the architecture for each use case are specified by: (i) defining use case-components mappings, including details on the component application in each specific use case; (ii) HPC customisation for different use case experiments and framework executions; and (iii) presenting data value chain specifications using the architectural blueprint of the DataBench project [2], preparing the ground for future benchmarking tasks within the project.

Finally, the deliverable provides mappings of the MARVEL conceptual architecture to the relevant Big Data, AI, and Fog computing reference architectures, including the Big Data Reference Architecture proposed by BDVA, European AI, Data and, Robotics Framework, and NIST Fog Computing Conceptual Model (Edge-Fog computing), establishing a bridge to efficiently account for further developments in the relevant project areas - Big Data, AI, continuum computing, etc., and future EU strategic agendas in the respective domains.

14 Bibliography

[1] "D1.2 MARVEL's Experimental protocol," Project MARVEL, 2021.

[2] "D5.4 Analytic modelling relationships between metrics, data and project methodologies," Project DataBench, 2020. https://www.databench.eu/wp-content/uploads/2021/01/d5.4-analytic-modelling-relationships-between-metrics-data-and-project-methodologies-ver.-1.0.pdf

[3] BDVA, European Big Data Value Strategic Research and innovation Agenda, Big Data Value Association, 2017. https://www.bdva.eu/sites/default/files/BDVA_SRIA_v4_Ed1.1.pdf

[4] "Strategic Research, Innovation and Deployment Agenda: AI, Data and Robotics Partnership," 2020 September 2020. [Online]. Available: AI-Data-Robotics-Partnership-SRIDA-V3.0.pdf. [Accessed 11 August 2021].

[5] M. Iorga, L. Feldman, R. Barton, M. Martin, N. Goren and C. Mahmoudi, "Fog Computing Conceptual Model, Special Publication (NIST SP)," National Institute of Standards and Technology, Gaithersburg, MD, 2018. https://www.nist.gov/publications/fog-computing-conceptual-model

[6] "ELAN (Version 6.2) [Computer software].," Max Planck Institute for Psycholinguistics, The Language Archive., Nijmegen, 2021. https://archive.mpi.nl/tla/elan

[7] S. Hantke, F. Eyben, T. Appel and B. Schuller, "iHEARu-PLAY: Introducing a game for crowdsourced data collection for affective computing," in *2015 International Conference on Affective Computing and Intelligent Interaction (ACII)*, 2015.

[8] "D9.4: OEI - Requirement 8," Project MARVEL, Confidential, 2021.

[9] "D2.1 Collection and analysis of experimental data," Project MARVEL, 2021. https://www.marvel-project.eu/deliverables/

[10] "D5.5 Final report on methodology for evaluation of industrial analytic project scenarios," DataBench project, 2020. https://www.databench.eu/wp-content/uploads/2021/01/d5.5-final-report-on-methodologies-for-evaluation-of-industrial-analytic-project-scenarios-ver-1.0-.pdf