Horizon 2020
European Union funding
for Research & Innovation

Big Data technologies and extreme-scale analytics

Multimodal Extreme Scale Data Analytics for Smart Cities Environments

D5.7: MARVEL's framework large scale deployment[†]

Abstract: This deliverable showcases the capabilities of the MARVEL framework for large-scale deployments, as well as extending it with third-party components. The deliverable first presents an overview of the main concepts and the use cases exploited within the MARVEL project. In addition, the deliverable reports on the scalability and extensibility requirements for a large deployment in a Smart City, which is the main target use case of MARVEL. Then, the approach considered for scalability and extensibility, organised based on the requirements previously described, demonstrates in detail the actions taken from the various components towards the completion of this goal. Subsequently, an in-depth analysis of the legal, ethical, and privacy related concerns, followed with operational and user-oriented considerations.

Contractual Date of Delivery	31/12/2023
Actual Date of Delivery	23/01/2024
Deliverable Security Class	Public
Editor	*Ilias Nektarios Seitanidis (INTRA)*
Contributors	FORTH, CNR, INTRA, ATOS, FBK, AU, ITML, ZELUS, TAU, GRN, PSNC, STS, UNS
Quality Assurance	*Manos Papoutsakis (FORTH)* *Lorenzo Valerio (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. *Manos Papoutsakis (FORTH)*
2. *Lorenzo Valerio (CNR)*

Revisions

Version	Date	By	Overview
0.7	23/01/2024	INTRA	Final version ready for submission
0.6	23/01/2024	INTRA	Addressing comments from the PC
0.5	09/01/2024	INTRA	Reviewers' Comments addressed
0.4	29/12/2023	INTRA	Submitted for review
0.3	28/12/2023	INTRA	Final inputs incorporated
0.3	20/12/2023	INTRA	Partner's input incorporated
0.2	13/11/2023	INTRA	TOC draft
0.1	13/10/2023	INTRA	Initial TOC draft

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.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AI	Artificial Intelligence
API	Application Programming Interface
AV	Audio-Visual
AVCC	Audio-Visual Crowd Counting
BDVA	Big Data Value Association
DFB	Data Fusion Bus
DMP	Data Management Platform
DPO	Data Protection Officer
E2F2C	Edge-to-Fog-to-Cloud
EAB	Ethics Advisory Board
E/F/C	Edge/Fog/Cloud
FFMPEG	Fast Forward MPEG
FL	Federated Learning
GDPR	General Data Protection Regulation
GUI	Graphical User Interface
HDD	Hierarchical Data Distribution System
HDFS	Hadoop Distributed Files System
HPA	HorizontalPodAutoscaler
HPC	High Performance Computing
IoT	Internet of Things
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
MEMS	Micro Electro-Mechanical Systems
ML	Machine Learning
MQTT	Message Queuing Telemetry Transport
MVP	Minimum Viable Product
PC	Project Coordinator
PPG-VC	Phonetic PosteriorGram-Based Voice Conversion
R1	1st Release of the MARVEL Integrated Framework
R2	2nd Release of the MARVEL Integrated Framework
RAM	Random Access Memory
REST	Representational State Transfer
RTSP	Real-Time Streaming Protocol

SED	Sound Event Detection
SSH	Secure Sockets Shell
TB	Terabyte
TRL	Technology Readiness Level
UI	User Interface
URL	Uniform Resource Locator
UTC	Coordinated Universal Time
VM	Virtual Machine
VPN	Virtual Private Network
WP	Work Package
XML	Extensible Markup Language

Executive Summary

This deliverable outlines the capabilities of the MARVEL framework in the context of large-scale deployments, highlighting its extension with third-party components. The deliverable has been developed in the context of Task T5.5, “From the prototype to the final solution” within WP5 “Infrastructure Management and Integration”, under Grant Agreement No. 957337.

The work presented in this report relies significantly on previously submitted deliverables D5.4 “MARVEL Integrated framework – initial version” and D5.6 “MARVEL Integrated framework – final version”, which reported the work carried out by task T5.3 “Continuous integration towards MARVEL’s framework realisation”. The integrated analytics framework developed under the framework of T5.3 was the main input for T5.5. The latter exposed the integrated analytics to Big data sources and extreme-scale operation environments.

The document begins with an overview of the fundamental concepts, inference pipelines, and diverse use cases explored within the MARVEL project. Emphasising its focus on Smart City applications, the deliverable elucidates the scalability and extensibility requirements integral to the project's inception.

The strategy for achieving scalability and extensibility is meticulously presented, organised in alignment with the pre-defined project requirements. This includes a comprehensive examination of the measures undertaken across various components to meet the objectives of large-scale deployment in a Smart City environment. To this end, experimental results or theoretical approaches based on the characteristics of the components have been documented.

Legal, ethical, and privacy considerations are subjected to an in-depth analysis, addressing concerns surrounding data usage, algorithmic decision-making, and user privacy. Operational and user-oriented considerations are then explored, ensuring a holistic approach to deployment that aligns with regulatory frameworks and user expectations.

The deliverable concludes with a succinct summary of key findings and conclusions drawn from the comprehensive exploration of MARVEL's capabilities, scalability, and ethical considerations. It serves as a vital resource for stakeholders, providing insights into the framework's potential for impactful large-scale deployment in Smart City environments while ensuring adherence to legal and ethical standards.

1 Introduction

1.1 Purpose and scope of this document

Deliverable D5.7 entitled "MARVEL's framework large scale deployment", reports on the progress made through the relative activities carried out within the Task T5.5 framework. The key areas that T5.5, "From the prototype to the final solution" covers are: a) explore and demonstrate the scalability features of the various building blocks that compose the MARVEL framework and b) examine and report all the legal, ethical, privacy and accessibility concerns related to the utilisation of the MARVEL framework in real-world large scale scenarios. The actions related to these two key points cover all the high-level and technical insights that prove how MARVEL can adapt to any Smart City scenario in the real world. This leads to the fulfilment of Objectives 3, 4, and 5 entitled "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", "Realise societal opportunities in a smart city environment by validating tools and techniques in real-world setting" and "Foster the European Data Economy vision and create new scientific and business opportunities by offering the MARVEL Data Corpus as a free service and contributing to BDVA (Big Data Value Association) standards" respectively.

1.2 Contribution to WP5 and project objectives

The work presented in this deliverable was conducted under the T5.5 framework and some parts of this work are reported in deliverables of other Work Packages (WP) as well. The work reported in the following sections regards the contribution of each of the MARVEL components in a flexible, highly scalable and extensible framework that can meet the needs of a large-scale deployment such as a modern Smart City. During the course of the project, the components presented in this document fulfilled their goals, defined through the project's Key Performance Indicators (KPIs). More details regarding the components' KPIs and metrics can be found in WP2, WP5, and WP6 deliverables D2.4[1], D2.5[2], D5.2[3], D5.5[4], D6.2[5], and D6.4[6]. Some of the project's KPIs related to the scalability and extensibility of the MARVEL framework are:

- KPI-O3-E1-1: Number of novel algorithms and tools utilised from diverse multi-domain technological areas ≥ 3.
- KPI-O3-E3-1: Realise a secure computing framework at all the processing layers.
- KPI-O4-E3-1: Execute the trial cases in at least 2 real-life smart city environments.
- KPI-O5-E2-1: Release Service Level Agreements and consider all the relevant aspects, namely accessibility, operability, managing streaming and network, legal considerations, security, privacy and technical concerns.

[1] "D2.4: Management and distribution Toolkit – final version," Project MARVEL, 2023. https://doi.org/10.5281/zenodo.8147109

[2] "D2.5: Corpus-as-a-Service specifications and business continuity," Project MARVEL, 2023. To appear.

[3] "D5.2: Technical evaluation and progress against benchmarks – initial version," Project MARVEL, 2022. https://doi.org/10.5281/zenodo.6322699

[4] "D5.5: Technical evaluation and progress against benchmarks – final version," Project MARVEL, 2023. https://doi.org/10.5281/zenodo.10438311

[5] "D6.2: Evaluation report," Project MARVEL, 2022. https://doi.org/10.5281/zenodo.7296312

[6] "D6.4: Final assessment report and impact analysis," Project MARVEL, 2023. To appear.

1.3 Relation to other WPs and deliverables

D5.7 is strongly related to the tasks of WPs that refer to the development of the MARVEL components, i.e., WP2, WP3, WP4, and WP5. More precisely, this deliverable relies on the work carried out in T5.3 and reported in the deliverables D5.1, D5.4, and D5.6. The foundational work conducted within '*WP1 – Setting the scene: Project setup'*. More specifically, the selection of use cases for demonstration draws from the detailed material on Use Case descriptions of deliverable *'D1.2 – MARVEL's Experimental protocol*'[7]. Additionally, deliverable '*D1.3 – Architecture definition for MARVEL framework*'[8] is an important source for this work, as it contains the refined architecture, which is the blueprint for this release, as well as subsequent releases. The work carried out in the WP1, was updated based on the requirements of the project as it progressed.

This deliverable is also coupled with work in the context of '*WP2 - MARVEL multimodal data Corpus-as-a-Service for smart cities', 'WP3 – AI-based distributed algorithms for multimodal perception and situational awareness',* and *'WP4 - MARVEL E2F2C distributed ubiquitous computing framework'*.

Within WP5, apart from T5.3, there has been a close collaboration with T5.1 and T5.2, with regard to the underlying infrastructure and resource management, respectively. Additionally, there is a close connection to T5.4 that aimed to set the integration and benchmarking environments of the MARVEL framework.

1.4 Structure of the document

The structure of this document is as follows:

- **Section 2**: In this section, an overview of the main aspects and requirements for scaling up the MARVEL framework in a real-world implementation is presented. In Section 2.1, the main concepts and use cases investigated through the MARVEL project are presented as well as the possible scalability considerations in each of the E2F2C layers. In Section 2.2, the requirements that a modern scalable and extensible framework should have are presented. The way that these requirements are met is presented in detail in Section 3. In Section 2.3, the limitations of the MARVEL project are described.
- **Section 3:** This section presents all the measures, methods and technologies that were applied in the design and implementation of the MARVEL framework to enable its scalability and extensibility potential, including the relevant work carried out by each individual component. The sub-sections are organised based on the main aspects of the MARVEL framework that contribute to the achievement of scalability and extensibility. In each sub-section, the relevant components present their main features that contribute to a more scalable framework, elaborated with implementation examples.
- **Section 4:** In this section, all the concerns and considerations of scaling up and extending to third-party systems are presented. More specifically, Section 4.1 covers all the ethical, legal and privacy-related aspects, while Sections 4.2 and 4.3 provide more technical details as well as the issues arising through the achievement of a scalable and extensible framework.
- **Section 5:** This section summarises and concludes the work presented in this document.

[7] "D1.2: MARVEL's experimental protocol," Project MARVEL, 2020. Confidential.

[8] "D1.3: Architecture definition for MARVEL framework," Project MARVEL, 2020. https://doi.org/10.5281/zenodo.5463897

2 Scalability considerations

In the domain of system architecture design, a continuous debate between horizontal and vertical scalability continues to shape the strategies adopted, seeking to meet the ever-increasing demands of modern applications and systems. Each approach has its strengths and weaknesses and it is crucial for the system designers to understand the differences between horizontal and vertical scalability for making decisions about system design and resource allocation.

Vertical scalability involves enhancing the computational capacity of a single machine by adding more resources, such as CPU, RAM, or storage. This is achieved through hardware upgrades to make the existing system more powerful. On the other hand, horizontal scalability focuses on distributing the workload across multiple machines or nodes. Instead of enhancing the power of a single machine, organisations add more machines to the infrastructure, allowing for the parallel processing of tasks. In Table 1, a brief comparison of the two approaches is presented in the areas of cost, performance, fault tolerance, and maintenance.

Table 1: Overview of vertical and horizontal scalability

	Vertical scaling	Horizontal scaling
Cost	Upfront investment in high-end hardware.	Pay-as-you-go model aligns infrastructure costs with actual demand.
Performance	Limited to how much a single machine can be upgraded.	Handles increased workloads by distributing tasks across multiple nodes/instances.
Fault tolerance	Not inherently enhance fault tolerance.	Enhances fault tolerance by distributing the workload across multiple nodes/instances.
Maintenance	Complex procedure.	Simple procedure.

Both approaches, vertical and horizontal scalability, have a series of advantages and disadvantages. An in-depth analysis of the scenario to be used is of paramount importance. In Section 2, the main pipelines and use cases of the MARVEL framework are introduced. While checking the insights derived from the framework analysis, we have identified the key requirements necessary for expanding the project to a greater extent, potentially evolving into an entire "smart municipality". The rationale behind this deliverable is to examine the requirements that need to be satisfied to implement the current MARVEL solution to a Smart City that will introduce the need for hundreds of cameras and microphones across the span of a large municipality.

2.1 Overview of the MARVEL main concepts, inference pipelines, use cases

The operation of the MARVEL framework fundamentally relies on the 'AI Inference Pipeline'. In essence, this pipeline and its reference architecture constitute the backbone of the MARVEL framework and provide the most significant features of the MARVEL solution.

The MARVEL 'AI Inference Pipeline' reference architecture was conceived as a universal scheme that could be applied to all MARVEL use cases, including the existing 10 use cases addressed by the 2nd Release of the MARVEL Integrated framework (R2), but also any potential future use cases, with suitable adaptations on a case-by-case basis. This reference architecture was the outcome of the distillation of multiple requirements elicited from the analysis of use cases and ensures that there is a consistent and coherent approach in all applications of

MARVEL for extracting and delivering inference results from the processing of multimodal Audio-Visual (AV) data. This architecture is illustrated in Figure 1 below.

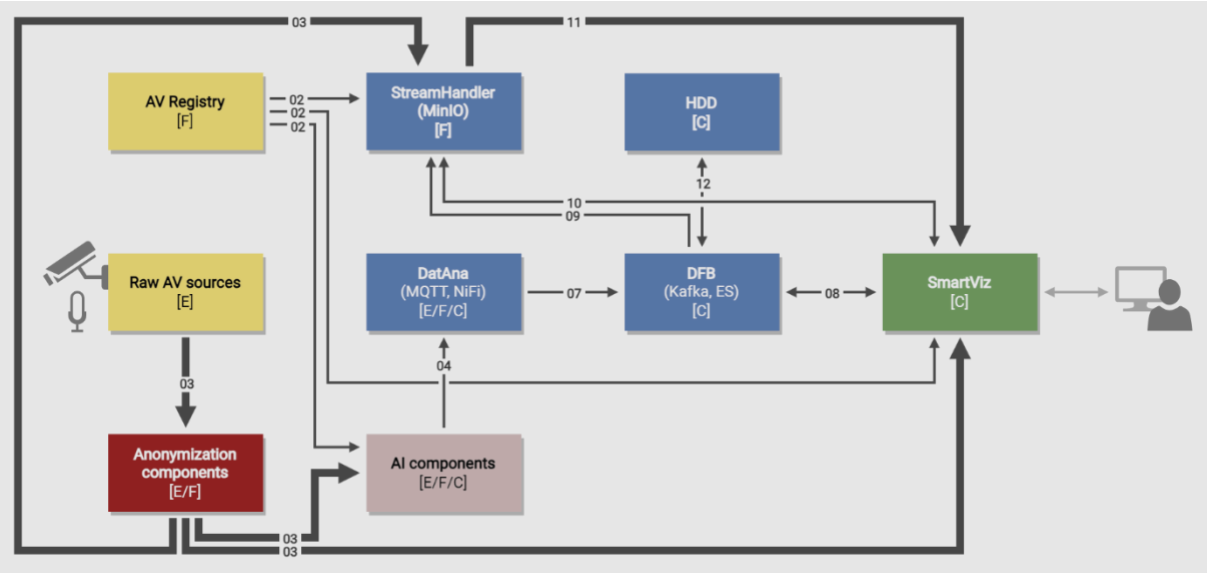

Figure 1: MARVEL 'AI Inference Pipeline' reference architecture diagram

The MARVEL 'AI Inference Pipeline' reference architecture and the design process that led to its definition are elaborated in D5.6 'MARVEL Integrated framework – final version'[9], which presents the details of the R2 and associated integration activities. Specifically, inside D5.6:

- Section 2 presents the R2 use cases that drove the design of the MARVEL 'AI Inference Pipeline' reference architecture,
- Section 3.2 presents the design methodology,
- Section 5.1 presents the details of the architecture,
- Section 5.2 presents the Application Programming Interfaces (APIs) and IO interfaces used in the reference architecture,
- Section 5.3 presents the data models used in the reference architecture,
- Section 5.4 presents the instantiations of the MARVEL 'AI Inference Pipeline' reference architecture in each R2 use case.

The 'AI Inference Pipeline' reference architecture diagram presents the key building blocks and components of the MARVEL framework along with their interactions in the form of data exchange communications. It contains abstractions for certain components with similar functionalities that are grouped together. More specifically, the **Anonymisation components** refer to the ***AudioAnony, VAD, and VideoAnony*** components and the **AI components** refer to the ***CATFlow, TAD, VAD, ViAD, AVAD, VCC, AVCC, AT, SED, AAC, YOLO-SED, RBAD,*** and ***SELD*** components. A full description of all components referenced in the 'AI Inference Pipeline' is provided in Section 4 of D5.6. A complete description of all interactions between the building blocks and components in the diagram is provided in Section 5.1 of D5.6. The diagram also features a unified annotation with labels that refers to the indexing (ranging from #01 to #12) of the consolidated I/O interface and API types that were implemented (described

[9] "D5.6: MARVEL Integrated framework – final version," Project MARVEL, 2023, https://doi.org/10.5281/zenodo.8315386

in detail in Section 5.2 of D5.6). Thin arrows represent the exchange of text-based data and thick arrows represent the exchange of AV (binary) data.

The main operation of the MARVEL 'AI Inference Pipeline' can briefly be described through the following interactions:

- **Anonymisation components** receive AV data streams from raw **AV sources** (cameras and microphones).
- **Artificial Intelligence (AI) components** request the metadata of available AV sources (including anonymisation components) from the **AV Registry**.
- **AI components** receive anonymised AV data streams from the **Anonymisation components**.
- **AI components** publish their raw inference results to dedicated topics on the **DatAna MessageQueuing Telemetry Transport** (**MQTT)** broker which is hosted on the same layer (Edge/Fog/Cloud (E/F/C)) as the AI components.
- A **DatAna NiFi** node receives the raw inference results from the **DatAna MQTT** broker, which is hosted on the same layer (E/F/C) as the DatAna NiFi node. DatAna NiFi nodes transform the raw inference results they collect to SDM-compliant inference results and push them to a DatAna NiFi node at a higher layer (DatAna NiFi Edge pushes results to DatAna NiFi Fog and DatAna NiFi Fog pushes results to DatAna NiFi Cloud).
- **DatAna NiFi** (Cloud node) publishes the Smart Data Models (SDM)[10] compliant inference results it collects to a topic on a Kafka[11] broker of the **Data Fusion Bus (DFB)**. The DFB persistently stores the received results on an Elasticsearch (ES)[12] database.
- **SmartViz** accesses the incoming inference results at the **DFB** in real-time by subscribing to the Kafka topics where they are published by DatAna. SmartViz also accesses the historical inference result data stored in the ES repository of the DFB. SmartViz displays the inference results to the user through appropriate visualisations.

In parallel, the following interactions take place for SmartViz to gain access to AV data:

- **SmartViz** and **StreamHandler** request the metadata of available AV sources (including anonymisation components) from the **AV Registry**.
- **StreamHandler** receives anonymised AV data streams from the **Anonymisation components**. StreamHandler segments and stores the anonymised AV data in a MinIO repository for subsequent access.
- **StreamHandler** accesses the incoming inference results at the **DFB** in real-time by subscribing to the Kafka topics where they are published by DatAna to produce AV files that correspond to these results.
- **SmartViz** requests and receives AV data from **StreamHandler** that refers to a particular inference result.
- **SmartViz** receives an anonymised AV data stream from an **Anonymisation component** upon user demand. The connection is not continuous and is initiated only

[10] https://smartdatamodels.org/

[11] https://kafka.apache.org/

[12] https://www.elastic.co/

when a user requests a live AV feed from the location where an event has previously been detected.

The 'AI Inference Pipeline' operation is also complemented by the following interaction that serves the purposes of performance optimisation:

- The **Hierarchical Data Distribution System (HDD)** receives the current Kafka topic partition configuration from the **Data Fusion Bus** (**DFB)** and returns a recommendation for an optimised Kafka topic partition configuration.

When considering a real-world implementation of the MARVEL 'AI Inference Pipeline', one of the initial aspects that needs to be addressed is the mapping of its constituent building blocks and components to the Edge-Fog-Cloud continuum. The 'AI Inference Pipeline' architecture proposes specific deployment layers for certain components but also incorporates inherent flexibility as far as other components are concerned. These are denoted by the [E], [F], [C] annotation in the diagram of Figure 1. For some of the components, the possibilities of the deployment layer are more limited due to their nature (e.g., AV sources are located at the Edge and SmartViz is located at the Cloud). In other cases, there are more deployment possibilities (e.g., AI components can be deployed on any layer). More details as well as the rationale behind the definition of possible mappings of the various building blocks to Edge-Fog-Cloud layers are provided in Section 5.1 of D5.6.

Another initial aspect to consider in the design of a real-world implementation of the MARVEL 'AI Inference Pipeline' is the available surveillance and computational infrastructure on each of the three E/F/C layers. Regarding the surveillance infrastructure, the specific AV sources (cameras, microphones or other sensors) that are to be used need to be specified along with other associated specifications and requirements (e.g., location, access modality and details, etc). In terms of the computational infrastructure, the specific nodes that can be made available can determine the mapping of components to E/F/C layers and to specific devices that are part of the infrastructure that hosts the MARVEL framework. Due to the inherent design of the MARVEL framework, virtually any computational device that can be attached to a Kubernetes[13] cluster can potentially be added to the devices hosting the deployed services of the MARVEL framework.

Finally, the exact use case requirements from end-users and the purpose of the MARVEL framework implementation need to be defined, which can affect the selection of anonymisation and AI components and their deployment location.

Eventually, the consideration of all aforementioned aspects should lead to an instantiation of the 'AI Inference Pipeline' reference architecture in any new use case of a real-world implementation. Similarly, the same aspects need to be considered in any scalability analysis and large-scale implementation since they are intrinsic to the MARVEL framework and serve as its foundations.

2.2 MARVEL implementation at city scale and associated requirements

In the previous sub-section, the MARVEL 'AI Inference pipeline' reference architecture, its main concepts and operation were presented as well as a brief overview of the mechanisms that need to be considered in any scale-up procedure of the framework. Another important aspect

[13] https://kubernetes.io/

for an efficient scale-up path of the MARVEL framework is to understand why this needs to be realised.

MARVEL predominantly targets the operational environment of Smart Cities. When we refer to real-world implementation scenarios of MARVEL, essentially we refer to the adoption of the MARVEL framework by municipalities and relevant local competent authorities. This implementation could potentially be limited in scale and involve only a few locations that need to be monitored. However, the MARVEL solution needed to be designed and fully equipped to handle city-wide large-scale implementations, should such an interest be expressed by potential future customers and end-users.

In this context, the possibility of implementing MARVEL in use cases that refer to the full scale of a large city was analysed to examine the critical aspects that need to be considered when performing such a scale-up from the use cases that have already been implemented in the context of the MARVEL project. This process has led to the identification of a set of relevant requirements. This section will present these requirements, alongside the possible associated challenges to be faced and the mitigation solutions to these challenges.

2.2.1 Expanding the number of sensor nodes

The first and most critical aspect that needs to be considered in a potential implementation of the MARVEL solution in the full scale of a large city is the increase in the number of sensor nodes that need to be integrated into the system. Essentially, this means that a significantly larger number of AV sources needs to be managed and accessed and the resulting AV data feeds analysed. This main requirement leads to a series of other consequent second-order requirements that need to be met, analysed below.

AV Resource Discovery, Management, Access, and Storage. This requirement concerns all the interactions needed to enable the AV resource discovery and management as well as accessing the produced data and storing them in an efficient manner. As the number of sensor nodes increases, the complexity of AV resource discovery, management, access, and storage is amplified. In addition, another challenge may arise in ensuring seamless integration with existing systems, leading to potential bottlenecks in resource handling. Another aspect to consider is the potential increase in the variance in the types and models of AV sources that need to be integrated into the MARVEL framework.

Increased number of streams to be anonymised. This requirement refers to the additional workload that will be introduced by the anonymisation components for handling more streams. Anonymising additional data streams poses computational challenges, potentially impacting real-time processing, while the increasing volume of streams to be anonymised may introduce additional data privacy and General Data Protection Regulation (GDPR) related concerns. In addition, the anonymisation process needs to take place as close to the source nodes as possible, i.e., at the edge or fog layers of the computational infrastructure.

Increased number of streams to be processed by the AI components. This requirement regards the additional workload to be introduced in the inference pipelines and more specifically the AI components' processing. Feeding more AV data streams into AI components requires enhanced processing capabilities. In addition, as the AI components cause an intense resource consumption, a scale-up of these components may lead to a significant increase in terms of required resources like processing power and memory.

Management of higher volumes of inference results. This requirement refers to components handling the inference results in the AI inference pipeline. With more sensor nodes and AI components, managing efficiently and mitigating the inference results becomes crucial to

maintaining data accuracy and reliability. Furthermore, the increased volume of inference results may lead to a higher production of false positives, affecting the overall effectiveness of the system.

AI training demands for more datasets. An increased number of AV sources leads to additional requirements in relation to training the AI components that need to be applied for the analysis of the AV data feeds from sources at new locations. To guarantee the effectiveness and level of accuracy in the AI inference results, the set of training data needs to be increased significantly to include samples from the new locations. For this reason, a demand for more computational and storage resources may come up. In addition, as the datasets may not be consistent and vary for each deployment even in the same city or municipality, there is the need to ensure that the datasets will be of good quality and include enough samples for each use case.

Increased demand for AV data segmentation and storage. An increased demand for AV data segmentation is the outcome of the additional AV source nodes introduced to the system. With more AV sources the number of AV streams to be segmented is significantly increased resulting in a system overhead. In addition to the need to segment more AV streams, the newly created segments have to be stored somewhere. Thus, the need for an efficient storage space that will handle the needs of the increased demand of AV segments is introduced.

2.2.2 Expanding the number of end users

In a real-world, scaled-up implementation of the MARVEL framework in the context of a large city, it is anticipated that the system may need to be accessed by multiple human operators concurrently. In particular, granted that SmartViz is the main system front-end and the single point of contact between users and the MARVEL framework, this practically means that a high number of SmartViz users can be expected. Therefore, all the necessary provisions need to be made to ensure that the additional load is sufficiently handled and that the overall MARVEL framework can serve an increased number of operators. This primarily refers to internal scalability issues of the SmartViz application that needs to guarantee uninterrupted and continuous availability (24x7) as well as a fluid user experience, while allowing a fault-tolerant operation in case of sub-system failure or reception of invalid data. In addition, the increased demands resulting from an increased number of SmartViz users also refer to scalability issues of the back-end system and especially all other MARVEL components that SmartViz needs to access and interact with, such as the DFB and StreamHandler. These back-end components should also make all necessary provisions to be able to handle the respective increased number of requests and associated computational load.

2.2.3 MARVEL extensibility

In its current form, the MARVEL framework has successfully integrated multiple technological components that participate in the MARVEL project. These components and their integration into a unified, seamless framework have managed to fully address the needs of the 10 use cases, where they were implemented and offer significant added value through the offered technological achievements and innovations.

However, in the context of a long-term sustainability plan, MARVEL needs to ensure that it is able to address new demands and requirements that have not yet been identified through the implementation of MARVEL in the existing 10 use cases but may present themselves when implementing MARVEL in **new Use Cases** formulated by **prospective future customers**. In the same context, MARVEL cannot remain static as a framework but should keep up with all major technological developments in associated fields and be able to incorporate all respective

main technological achievements and progress so as to preserve its efficiency, effectiveness, and competitiveness.

Therefore, to maximise the **(1) applicability** and **(2) future-proofness** of the MARVEL framework, it is critical to ensure that it is able to incorporate and/or interface with new and additional components, systems, and technologies. These considerations refer to the potential for **extensibility** of the MARVEL framework. The following four cases have been identified as the most prominent ones for incorporating and/or interfacing with new third-party tools and components for the further extension of MARVEL:

- **New input data sources and modalities, such as external databases and non-AV sensor data**. In its current form, MARVEL has been implemented to receive real-time input data from AV sources, i.e., camera and microphone sensors in the 'AI Inference Pipeline' and this has proven to be sufficient for addressing the needs of the ten use cases. However, future implementations of MARVEL might demand the use of other live data types and modalities, obtained from different sensors, such as sensors that measure environmental conditions, e.g., temperature, atmospheric pressure, and air quality. In addition, a common requirement that may arise in future use cases is the interfacing of the MARVEL framework with external databases to access and obtain static or dynamic data that can be analysed within the MARVEL framework and potentially juxtaposed with data collected from other sources.
- **New third-party analytical and AI components**. Currently, 13 individual AI components have been integrated within the MARVEL framework and have been applied in the 10 MARVEL use cases. These AI components share a common operation mode that consists of receiving live AV data and processing these data in real-time to produce AI inference results. Even though these components were sufficient to meet the requirements of the currently implemented use cases, new future use cases may demand (i) the use of different data modalities (non-AV), (ii) the production of different types of inference results that match the use case context, (iii) different priorities on the performance thresholds to be met. Furthermore, in consideration of the rapid evolution of the field of AI-based analysis in recent years, the MARVEL framework needs to be able to incorporate all technological developments that can improve its efficiency. Therefore, the MARVEL framework needs to be able to incorporate or interface with new AI components that have not already been integrated. There are **two main types of new AI components that can be integrated**: (a) AI components that process live data streams (either raw or anonymised) and (b) AI components that can post-process the AI inference results produced by AI components in the former category for long-term analysis purposes. The type of the AI component affects the integration parameters as these two types are positioned differently in the MARVEL 'AI Inference Pipeline'.
- **New front-end UI and data visualisation components**. SmartViz has been built to fulfil the purpose of visualising the inference results produced by MARVEL and will therefore remain the main and recommended front-end UI tool of the MARVEL framework. However, data visualisation and interaction requirements may change in the future or there may be specific cases where other, third-party front-end applications are required to be used for visualising the inference results produced by the MARVEL 'AI Inference Pipeline'. For example, a potential customer may require that a specific software already used by the customer's organisation be used instead of SmartViz for interacting with MARVEL and visualising the inference results. MARVEL should be equipped to be able to deal with such possible future requirements.

- **External, third-party data sinks**. In the current form of the MARVEL 'AI Inference Pipeline', all produced inference results are aggregated at the DFB, where they are also stored permanently. However, in future use cases that MARVEL may be called to address, there may be a need to relay produced inference results to external data stores for permanent storage or feeding the produced results to other systems for further processing or for controlling aspects of such systems. For example, potential customers may need the inference results stored on their own databases that may be already established and connected with other processes of their organisation. For such reasons, MARVEL needs to provide the necessary endpoints that allow interfacing with external data sinks to propagate the produced inference results.

Considering the aforementioned future possible demands, in this section, we will demonstrate how third-party/ external systems can be connected to the MARVEL framework, e.g., external components (AI components, data sources, post-processing tools, visualisation tools) connect and be utilised alongside with the MARVEL framework.

One of the primary objectives of the project, as described also in T7.3, is the long-term sustainability and the use of the MARVEL E2F2C framework and Data Corpus-as-a-Service by third-parties in Europe. The integration of third-party systems into an existing framework is by default a complex and intensive process yet essential for the extensibility of a framework. In this context, within the MARVEL framework, an in-depth analysis was carried out and a set of requirements that a system should follow was derived. More specifically, the main areas considered include system analytics, standardisation, security, data management, monitoring, quality assurance, scalability plan and support.

System Analytics. Prior to any extensibility attempt a detailed analysis of the existing system architecture has to be carried out, including components' analysis, data flows and dependencies just to name a few. The purpose of this step is crucial as it will identify the possible integration points of the system with the third-party components as well as provide an estimation of the impact that this integration may have on the system. In addition, part of the system analytics is the study of systems flexibility and adaptability to additional workload. To this end, it has to be considered how to maintain the same performance while increasing the various data processing and management processes. This analysis was made for each use case examined within the MARVEL and was reported in several deliverables with the latest being D2.4 and D5.6.

Standardisation. For a system to be able to support extensibility, it is of paramount importance to define and standardise the communication interfaces and data formats used within the framework. To this end, all the communication interfaces should be well documented and aligned with industrial standards, i.e., RESTful Application Programming Interfaces (APIs). In addition to the communication interfaces, the data exchanged through them must also be well-defined and use a common, well-established format such as JavaScript Object Notation (JSON) and Extensible Markup Language (XML). In MARVEL, several standards were used to guarantee the interoperability of the MARVEL components, among others ISO 8086 and RTSP just to name a couple.

Security. Security is an important aspect for every system itself, when considering extensibility several aspects have to be considered. The data exchange has to be safe and resilient to guarantee the data integrity and confidentiality. There are several mechanisms for a system to meet this requirement, one of them is the use of security authentication methods, such as OAuth or API tokens. Another way to guarantee the basic security aspects is by encrypting the communication channel/medium with the use of secure network communication channels such

as Virtual Private Network (VPN). In MARVEL, the security aspects were tackled with the use of EdgeSec VPN and with strict access policies to the MARVEL system.

Data management. One of the main concerns of extensibility is the data management. As the volume of data increases rapidly with the integration of third-party systems, a few aspects have to be considered. First of all, the additional data that will flow in the system has to be managed in a way that ensures data consistency across the various nodes, Edge/Fog/Cloud and the necessary mechanisms that can handle possible failures and guarantee the data consistency. In addition, based on the needs of the system, the real-time aspect of data consistency across the various nodes has to be addressed as well. Apart from the technical aspects of data management, there is the policy aspect. MARVEL's primary target is the European market and it is already compliant with the EU regulations regarding data management. However, each individual organisation may have additional policies on data management and an alignment should be considered.

Monitoring. During the first stages of integrating a third-party service into the existing system, errors may appear. In addition, it is possible that during the regular operational period of the third-party service and the system, errors may occur as well. To this end, it is of paramount importance to integrate logging mechanisms that will be able to capture errors and system activities taking place in both systems. Another critical aspect is the continuous monitoring of the two systems' performance. In addition to the logging mechanism, a monitoring mechanism should also be present. This mechanism should be able to monitor key performance metrics such as data throughput, error rates, response times and any other anomaly that would degrade the performance of the two systems. In MARVEL several logging and monitoring mechanisms have been integrated that allow the seamless supervision of all the components throughout the entire Edge-Fog-Cloud continuum.

Quality assurance. Prior to any extensibility attempt, it is crucial to guarantee that each component itself operates as intended and the whole system as well. This would reveal any errors before integrating a third-party system which could make the investigation and the solution of the arising errors very difficult. In this context, within the MARVEL framework, a series of system-level tests were carried out several times with the most notable on RP1 and RP2, reported on the deliverables D5.4[14] and D5.6 respectively. In addition, a series of benchmark tests took place to measure the system performance under high load, reported in deliverable D5.5.

Scalability plan. Extensibility sometimes is tightly connected with scalability as extending an existing system with a third-party system is considered as an addition of extra workload. A system that anticipates future growth and expansion should consider the initial design steps scalability, to seamlessly integrate the additional workload introduced by integrating with third-party systems. Apart from the individual components' scalability, the infrastructure used should support scalability by enabling dynamic resource allocation in terms of computation power, storage and network resources. For this reason, a scalability plan has to be prepared prior to any integration of a system with third-party systems. In MARVEL, this was achieved by considering high scalability and throughput mechanisms from the initial design steps as well as by performing infrastructure and component upgrades through the lifespan of the project always considering scalability and performance improvement. Finally, a scalability plan was developed by identifying the possible requirements needed, Section 2.2.1.

[14] "D5.4: MARVEL Integrated framework – initial version," Project MARVEL, 2022. Confidential.

Support. Support is a crucial aspect of extensibility as it regards all the necessary instructions and insights required for an external system to be integrated with another system. Support is mainly focused on two aspects, user training and documentation. The new users should familiarise themselves with the system‘s features and functionalities. This means that a user training guideline should be written for the users and administrators in a coherent way. In addition, all the system interactions should be well-defined and documented with all the necessary details. In general, a well-documented system can be integrated more easily and faster and enable ongoing maintenance. In MARVEL, the documentation process was adopted from the beginning of the project with the creation of a GitLab repository where all the components had to document all the details required for inter-component. As the project was involving, this documentation was continuously receiving updates enhancing the whole system documentation and support material.

A successful integration of third-party systems requires a meticulous approach that addresses technical, security and operational considerations. By adhering to these essential requirements, organisations can minimise the challenges associated with integrating disparate systems and create a cohesive, high-performance environment. A well-executed integration enhances efficiency, fosters innovation, and positions the organisation for sustained success in an increasingly interconnected digital landscape.

2.3 Limitations within the MARVEL context

The MARVEL framework is conceived and designed to serve the needs of entire Smart Cities and is therefore meant to operate in the form of large-scale deployments when released as a commercial product. However, it should be considered that MARVEL is a 36-month RIA project starting from basic concepts and prototypes of a low Technology Readiness Level (TRL) and aiming to reach a TRL of 5-6 at its end. During the development of the MARVEL Integrated Framework and its three consecutive releases (Minimum Viable Product (MVP) in D5.1[15], 1st Release of the MARVEL Integrated framework (R1) in D5.4, R2 in D5.6), it was possible to apply and test it in multiple use cases that reflect contemporary and future needs of Smart Cities. Considering that the components integrated into the MARVEL framework were of a low maturity level in most cases and that the overall Integrated Framework was initiated from raw and non-validated concepts, one of the main purposes of the MARVEL project was to design and develop a prototype of the MARVEL integrated framework in order to be able to test it under operational conditions.

Within its 36-month period, MARVEL managed to achieve the goal of validating the MARVEL integrated framework under operational conditions, starting from the early MVP release that was deployed in a staging environment, using offline data as input and being applied in a single use case and delivering the final release of the MARVEL Integrated Framework (R2) towards the project end that was deployed in on-site infrastructure spanning the entire E2F2C continuum, using live data as input from actual AV sources and being applied in 10 use cases in three pilot sites. In this context, MARVEL was applied as a whole in small-scale experiments that could validate the operation of the framework in real-life conditions. At the same time, the overall design and configuration of the Integrated Framework and constituent components and mechanisms, as well as the deployment, testing, monitoring, benchmarking and optimisation activities, were performed with a view towards enabling future versions of MARVEL to be applied in large-scale deployments. Therefore, a series of embedded and validated mechanisms enable the framework’s scalability and extensibility by design and set the stage for the further

[15] “D5.1: MARVEL Minimum Viable Product,” Project MARVEL, 2021. https://doi.org/10.5281/zenodo.5833310

evolution of MARVEL into a production stage and ultimately its release as a commercial product. The following section (Section 3) elaborates on all such mechanisms that allow MARVEL to meet the scalability and extensibility requirements set out in Section 2.

In the context of deployments and tests in operational conditions, several attempts were made, however, a combination of the available time, resources and starting point in the project posed certain realistic limitations with regard to the scale to which the framework could be applied and tested. Other limiting factors included the availability of input data and computational infrastructure. In the three pilot sites where MARVEL was applied, it was only possible to access the data streams from a few AV sources (cameras and microphones) due to the assets that were made available and also due to data privacy issues and other legal implications. Even in the few cases where potentially slightly more AV sources could be made available (e.g., in MT), another limiting factor was the computational infrastructure, especially at the edge and fog. A main restriction set by the project was that any AV data from public locations would need to be anonymised close to the source before being further processed and analysed, meaning that the relevant anonymisation procedures needed to run on edge and fog devices. The fact that there was a delay in securing suitable edge devices in the project due to problems in the supply chain during 2021-2022 alongside with the increased cost of the hardware, led to limitations in the available options and activities. In addition, in the case of the MT pilot in particular, the solution for accessing AV data involved the use of the FBK infrastructure at the fog level resulting from a legal agreement between MT and FBK, nominating FBK as a Data Processor. The specific infrastructure configuration implemented at FBK for dealing with legal, data privacy, and security issues (e.g., dedicated VPN connection to MT network, dual segregated fog servers at FBK in different subnets) complicated matters further also imposed restrictions in terms of the number of AV sources whose streams could be processed at the fog level in the MT pilot.

3 Scalability approach in MARVEL

The purpose of this section is to show how the overall MARVEL framework and individual components help to address the requirements of Section 2. This will be achieved with the description of the design approach for the overall framework, the mechanisms embedded into it, the adopted protocols, the features of each component and their contribution towards achieving scalability and extensibility. Where available, relevant tests made for each component and the related findings will be reported. This information is organised in sub-sections that group together elements that refer to a central common theme or aspect of the MARVEL system.

3.1 Overview of the scalability approach and system design

In the dynamic landscape of technology, the demand for scalable and robust systems is ever-growing. Horizontal scalability, also known as scaling out, has emerged as a key strategy to meet these demands. Unlike vertical scalability, which involves increasing the power of a single machine, horizontal scalability focuses on distributing the workload across multiple machines or nodes. This approach offers a range of benefits, making it a preferred choice for modern, high-performance systems.

Horizontal scalability is the ability of a system to handle an increasing amount of work by adding more resources, typically in the form of additional machines or nodes. This is in contrast to vertical scalability, where a single machine is upgraded to handle more load. Horizontal scalability is achieved by adding more instances of a resource, such as servers, to a system, which allows it to distribute the workload and handle more concurrent users or transactions.

One of the key advantages of horizontal scalability lies in its cost-effectiveness. Instead of investing in expensive high-end hardware to upgrade a single machine, organisations can use cost-effective commodity hardware and add more machines as needed. This not only reduces upfront costs but also provides a more flexible and scalable infrastructure that can adapt to changing workloads. Horizontal scalability offers a cost-effective approach to increasing system capacity. Instead of investing in expensive, high-capacity hardware, organisations can start with a smaller setup and scale horizontally by adding more machines as needed. This pay-as-you-go model allows businesses to align their infrastructure costs with actual demand, optimising resource utilisation and minimising unnecessary expenses.

In addition to the financial benefit of horizontal scalability, there are numerous more technical benefits of scaling up horizontally. Horizontal scalability enhances performance by allowing the system to handle a larger number of requests simultaneously. As more machines are added to the infrastructure, the system can distribute the workload across these nodes, leading to improved throughput and reduced response times. This is particularly beneficial for applications with fluctuating workloads or those experiencing rapid growth. Moreover, distributing the workload across multiple nodes enhances fault tolerance and reliability. If one machine fails, the remaining nodes can continue to handle the workload, ensuring minimal impact on overall system performance. This inherent redundancy contributes to increased system availability and resilience, critical factors for applications that require high levels of uptime. Finally, horizontal scalability enables organisations to scale their infrastructure dynamically in response to changing workloads. With the ability to add or remove nodes based on demand, organisations can achieve elasticity, ensuring that resources are allocated efficiently. This is particularly valuable for applications with variable workloads, such as e-commerce platforms experiencing peak traffic during sales events. Managing a horizontally scalable system is often more straightforward than dealing with a monolithic, vertically scalable

infrastructure. Upgrades and maintenance can be performed on individual nodes without affecting the entire system. This modular approach simplifies system management and reduces the risk of downtime during maintenance activities.

In conclusion, horizontal scalability offers a flexible and cost-effective solution to meet the increasing demands of modern applications. By distributing workloads across multiple nodes, organisations can achieve improved performance, cost-efficiency, fault tolerance, and scalability on demand. Embracing horizontal scalability is a strategic decision that positions businesses to adapt to evolving requirements and deliver a reliable, high-performance user experience. MARVEL generally adopts the horizontal scalability approach but also introduces elements of vertical scalability in the choice of infrastructure for edge and fog layers. In the following subsections, it will be demonstrated how each of the various MARVEL components contributes to the horizontal scalability, enabling MARVEL for large-scale deployments.

Within the MARVEL project, a complete systems analysis with details regarding the components and the dataflows among them has been carried out and updated through the project's lifespan. This analysis was made for each use case examined within the MARVEL and was reported in several deliverables with the latest being D2.4 and D5.6. For the needs of the MARVEL framework, a combination of the two approaches is adopted, but with a dominance of the horizontal scalability approach due to certain limitations of the vertical scale-up. More specifically, vertical scalability has an upper limitation to the upgrades that can be performed on a single machine. In addition, the node has to be equipped with the latest components in terms of hardware to cope with extreme workload and after reaching the node's upper limit another one of the same specifications has to be deployed. On the other hand, considering that the number of AV source nodes in MARVEL is varying and that this variation is the main driver of the MARVEL framework scalability demands, the horizontal approach is more preferable due to its pay-as-you-go model and the reduced maintenance effort required. In addition, the horizontal scalability approach complies with the overall deployment strategy of the MARVEL framework that is meant to occur within the E2F2C continuum and therefore involves devices across different locations. That said, elements of a vertical scalability approach are also adopted, where required. For example, options of computational devices at the edge with higher computational capacity can be selected when there is a high computational demand that needs to be served locally and only by a single node. From both technological and financial perspectives, horizontal scalability aligns better with the needs of MARVEL.

A key point of scalability and extensibility is the use of well-established and widely used interfaces. In this context, the MARVEL partners have identified and established common interfaces based on industrial standards. The first example of a communication interface is the Real Time Streaming Protocol (RTSP) (Schulzrinne, Lanphier, & Henning, 1998) which is an application-level network protocol for multiplexing AV streams, and it is widely used in the domains of entertainment and communications. A second example is the use of RESTful APIs for components' communication as well as Apache Kafka messages used for the inference result message exchange. Another example of standardisation within the MARVEL is the use of a common data format, ISO 8601 (Houston, 1993), and the establishment of a common time reference for all the components regardless of their deployment location, Coordinated Universal Time (UTC). Finally, regarding the message payload of the inference result messages the SDM specification was adopted.

In MARVEL, apart from the previously mentioned initiatives towards scalability and extensibility, the networking part was considered as well. Networking plays a pivotal role in facilitating the seamless deployment of large-scale systems across E/F/C layers, offering enhanced scalability and extensibility. In the context of Edge computing, where computing

resources are decentralised and positioned closer to the data source, efficient networking ensures quick data transfer and reduced latency. In MARVEL, as the main data transfer from the Edge is AV content, it is crucial to ensure the seamless and real-time propagation of them to the Fog. In this context, high bandwidth and low-latency network technologies were used for interconnecting the Edge and the Fog layers. Due to the nature of the Edge devices, state-of-the-art wireless technologies, such as the fifth generation of cellular communications protocol, were used. In the Fog layer, which acts as an intermediary between Edge and Cloud, networking enables the efficient distribution of computational resources. In MARVEL, the Fog layer is interconnected with the Cloud layer using broadband, high speed and low-latency wired technologies, such as Fiber optics networks. This is crucial as the inference information has to be propagated to the services residing in the Cloud in real-time and then provide the information gathered to the end user. As for the Cloud layer, a robust network infrastructure is crucial for managing and coordinating the overall system. Networking facilitates the seamless transfer of data between Fog and Cloud, enabling centralised data storage, complex analytics, and the utilisation of scalable cloud resources. In MARVEL, the Cloud layer utilises the 10x Gigabit lines provided by PSNC for employing a high bandwidth and low-latency network that will ensure a) the seamless and real-time information exchange among the components located in the Cloud as well as the Fog-Cloud interconnection and b) the seamless integration of more Fog and Cloud nodes as well as third-party services towards the extreme scale deployments and the extensibility.

3.2 Overall framework design and embedded mechanisms and protocols for scalability

The MARVEL ‘AI Inference Pipeline’ is built to be scalable by design. The generic reference architecture of this pipeline (presented in Section 2.1), may appear abstract and conceptual to some extent, but it also incorporates the necessary elements and the potential for the deployment of the MARVEL framework in the real world. First off, as described in Section 2.1, the general logic of mapping the constituent parts of the ‘AI Inference Pipeline’ reference architecture to the Edge-Fog-Cloud continuum is already embedded. The association of the architecture with the E/F/C continuum is coupled with an inherent ability to facilitate horizontal scaling. Practically, this means that the architecture of the ‘AI Inference Pipeline’ is fully compatible with the addition of devices (computational nodes) on each of the three E/F/C layers through a logic of distributing components across available nodes.

The diagram in Figure 2 is the same as the one presented in Section 2.1 for the ‘AI Inference Pipeline’ reference architecture, with the only difference of having an alternative graphic layout of the included components.

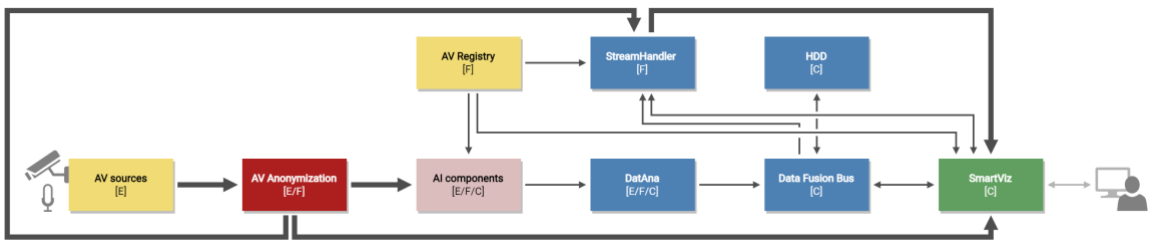

Figure 2: AI Inference Pipeline

The diagram in Figure 3 depicts in isolation the most critical part of the 'AI Inference Pipeline' reference architecture that is essential for producing and delivering AI inference results. In the rest of this section, we will examine this part more closely.

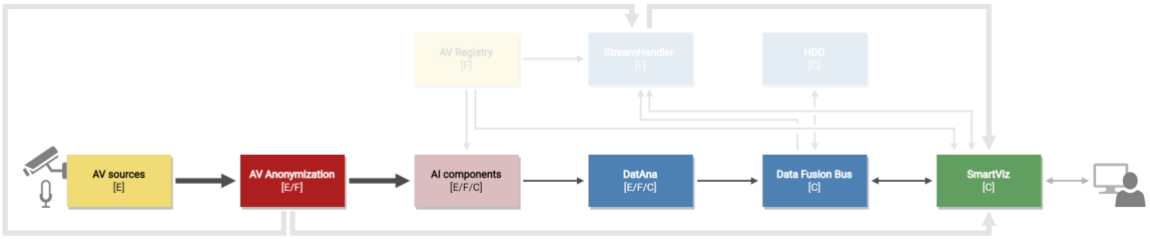

Figure 3: AI Inference Pipeline components

In the next diagram of Figure 4, we move from the more abstract version of the reference architecture to a depiction of a potential MARVEL deployment that is still generalised, but closer to the real world. This architecture diagram is an example of an instantiation of the reference architecture that illustrates (a) the distribution of components to E/F/C layers and (b) the presence of specific infrastructure computational devices (nodes) that host specific components on each layer (rendered as large purple rectangles). This is a very simple case for processing a single AV source with different AI components that are positioned on different devices (one on each E/F/C layer).

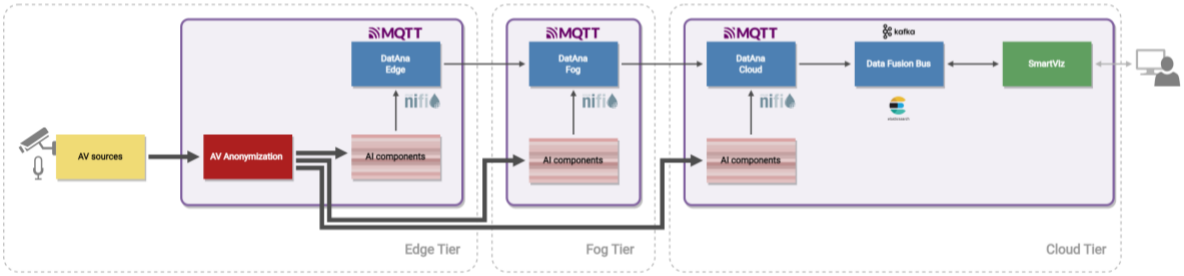

Figure 4: Potential MARVEL Architecture

Following the same rationale, the diagram in Figure 5 illustrates how the architecture from the previous example can be scaled up at the Edge layer. In this case, a second AV source has been added to the system and a second node has been added to the edge layer to accommodate this increase of AV sources. More specifically, the added Edge node can host a second set of components that will be related to the processing of the new AV source. This logic of replication by adding more Edge nodes and corresponding components can be extended to accommodate as many AV sources as is necessary.

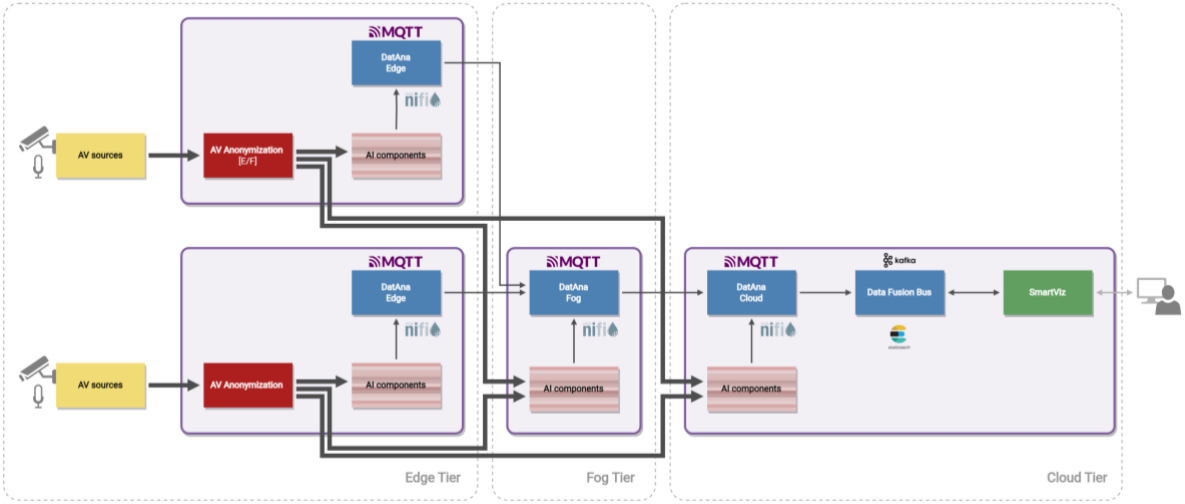

Figure 5: MARVEL scale-up at the Edge layer

In general, the nodes at the Fog layer are associated with specific AV sources and corresponding Edge nodes, i.e., the Fog nodes host components that process the data coming from specific AV sources and from components hosted on the corresponding Edge nodes. When AV sources and corresponding edge nodes increase, a maximum processing capacity can be reached at the associated Fog node. In such cases, the MARVEL framework can be extended by adding new Fog nodes. Each new Fog node in turn can be associated with its own AV sources and Edge nodes. The diagram in Figure 6 illustrates such an example, where two Fog nodes are present in the system, each of which is associated with two AV sources and Edge nodes. Also, when necessary, Fog nodes can be directly associated with specific AV sources without requiring an Edge node in between for performing first-order processing of the corresponding data stream.

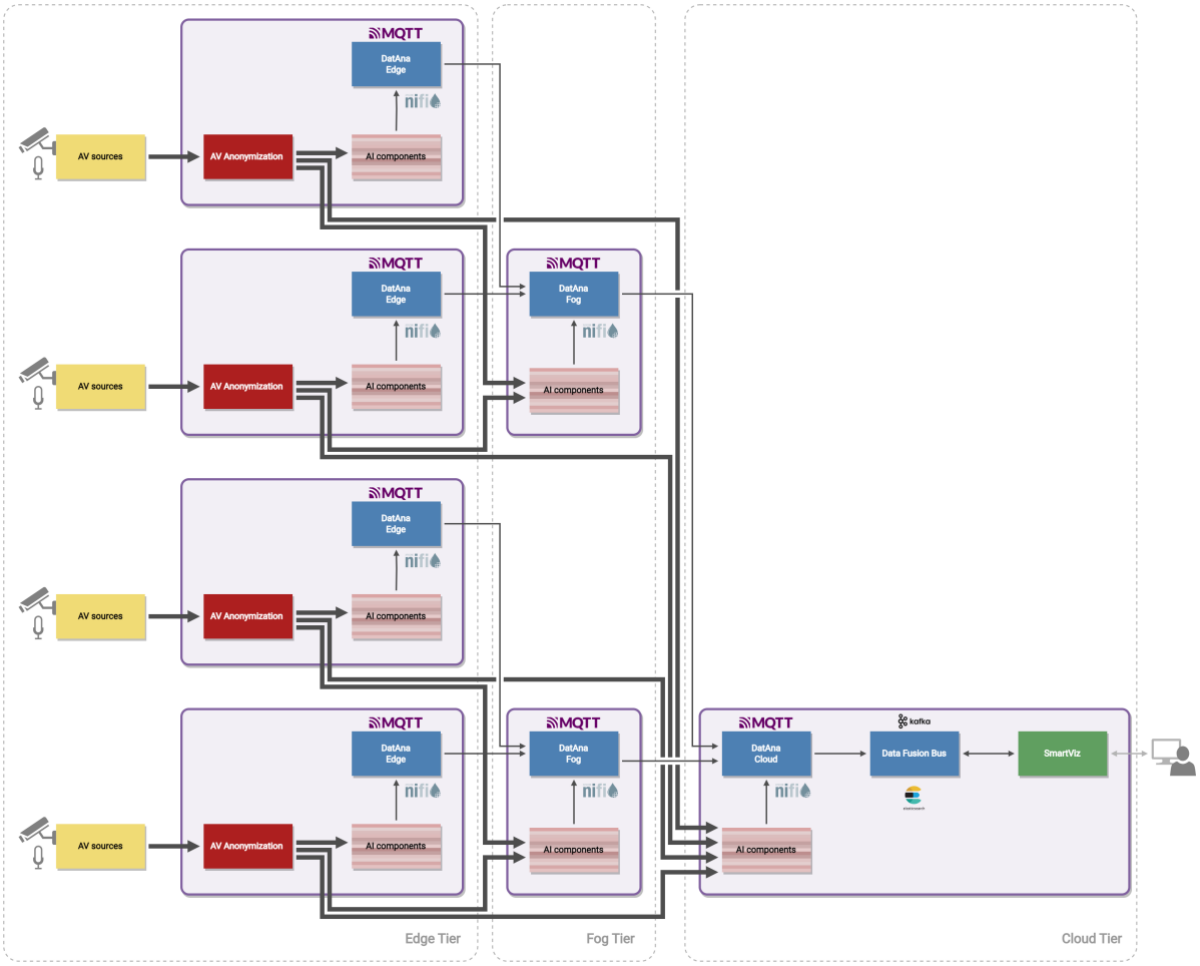

Figure 6: MARVEL scale-up at Fog layer

Based on the experiments conducted so far in the 10 use cases where the MARVEL framework has been implemented, a rough estimation can be made for a recommended sizing of the infrastructure at the Edge and Fog Layers as follows:

- Each Edge node can support 1-3 AV sources
- Each Fog node can support 3-12 Edge nodes

This estimation is based on the specific hardware devices used so far in MARVEL and the demands of the specific components that were implemented on the Edge and Fog layers. Any significant change in the computational resources of the devices positioned on these layers or demands generated by the hosted services will of course affect the infrastructure sizing.

Finally, at the Cloud level, horizontal scaling operates by adding Virtual Machines in the Cloud computational infrastructure where the MARVEL framework is deployed, such as the High-Performance Computing (HPC) provided by PSNC in the context of the MARVEL project. Each Virtual Machine acts as a different node in the system. In this case, there is a qualitative difference to the rationale followed in the Edge and Fog layers as there is no direct association between specific Cloud nodes and specific Fog nodes.

All Fog nodes are able to communicate with any of the available Cloud nodes. Components that are deployed in the cloud are distributed by the Kubernetes scheduler in an efficient way taking into account the defined requirements of the components in combination with the available resources of the nodes.

As an example, in the case of AI components hosted on the Cloud layer, the decision of which node will be used, is a result of the hardware requirements (CPU, RAM, GPU) defined in their respective template in MARVdash. Kubernetes' scheduler takes them into account and chooses the most appropriate node to deploy the component.

The general rule is that a single instance of an AI component processes the data stream originating from a single AV source.

The 'AI Inference Pipeline' reference architecture is designed to be versatile and adaptable to a very broad range of use cases both in terms of requirements related to the scope of data analysis and in terms of scalability regarding the accommodated AV sources and computational infrastructure while maintaining its consistency and cohesion. This is demonstrated by the application of the same reference architecture in different use cases in the context of the MARVEL project through the instantiation of specific fit-for-purpose architectures that are all based on the same underlying reference architecture. The logic described in this section for the horizontal scaling at each of the three E/F/C layers has been applied on a limited scale in several of the 10 use cases where the MARVEL framework was implemented.

For example, in the GRN4 use case (Figure 7), there are 2 Edge nodes, one of which is associated with a single AV source while the second is associated with two AV sources. A single Fog node is associated with both Edge nodes. The Cloud Layer hosts different instances of an AI component (SED), each of which is applied on a different AV source.

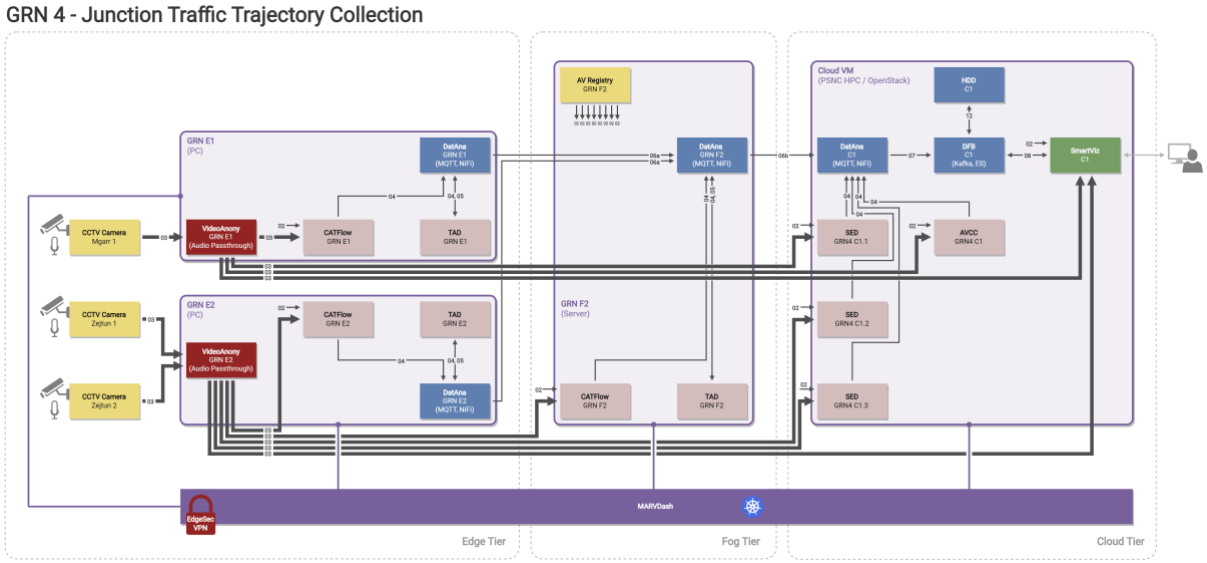

Figure 7: GRN example use case

In another example in the MT2 use case (Figure 8), there are 2 Edge nodes, each of which is associated with a single AV source. There are two more AV sources that are processed directly at the Fog node. Difference instances of AI components at the Cloud Layer (AVAD, AT, SED) process the streams of different AV sources.

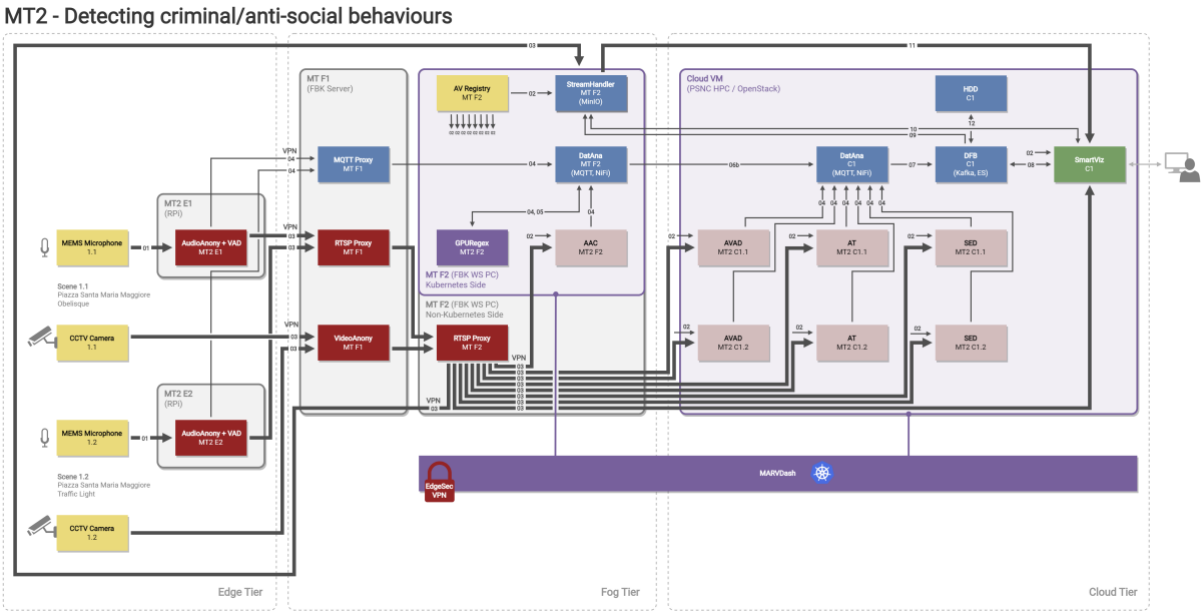

Figure 8: MT example use case

Due to the reasons and limitations described in Section 2.3, the deployment of the MARVEL framework at a large scale was not possible. However, the foundations have been set for future deployments of MARVEL at larger scales and ultimately at the scale of an entire city. Figure 9, Figure 10, and Figure 11 illustrate the logic of the horizontal scalability in the MARVEL framework in terms of computational infrastructure. They are based on an abstract representation of the infrastructure nodes that can be added to the MARVEL infrastructure cluster. The following symbology is used:

- Red circles represent AV sources.
- Yellow circles represent Edge computational nodes.
- Green circles represent Fog computational nodes.
- Blue circles represent the Cloud computational nodes (Virtual Machines (VMs)) and the large light blue circle represents the infrastructure cluster in the cloud.

Figure 9 presents a very simple case involving a single AV source and a single computational node at each E/F/C layer.

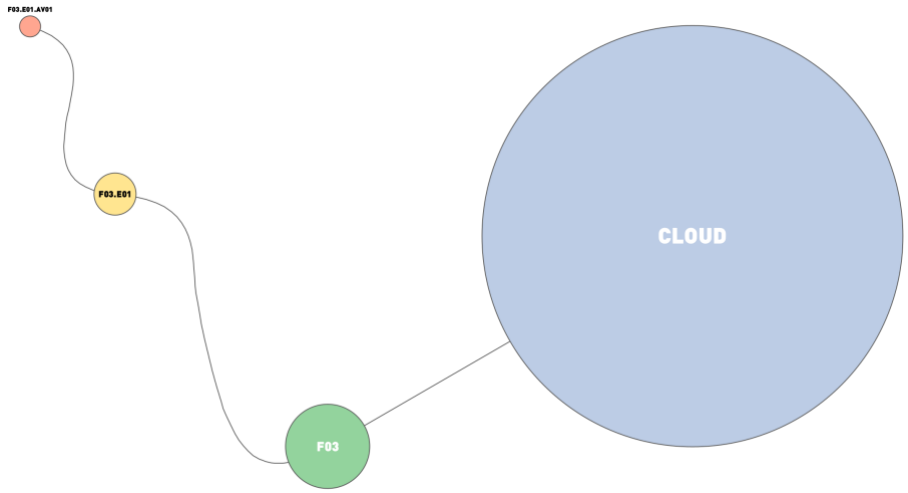

Figure 9: Single use case of E2F2C

Figure 10 presents a more evolved case of MARVEL framework deployment at a larger scale. This case closely approximates the scale level of the R2 release at the end of the MARVEL project, i.e., one Fog node at each of the pilot sites (GRN, MT, UNS) that is in turn associated with a series of Edge nodes associated with some AV sources, combined with a series of Cloud VM nodes.

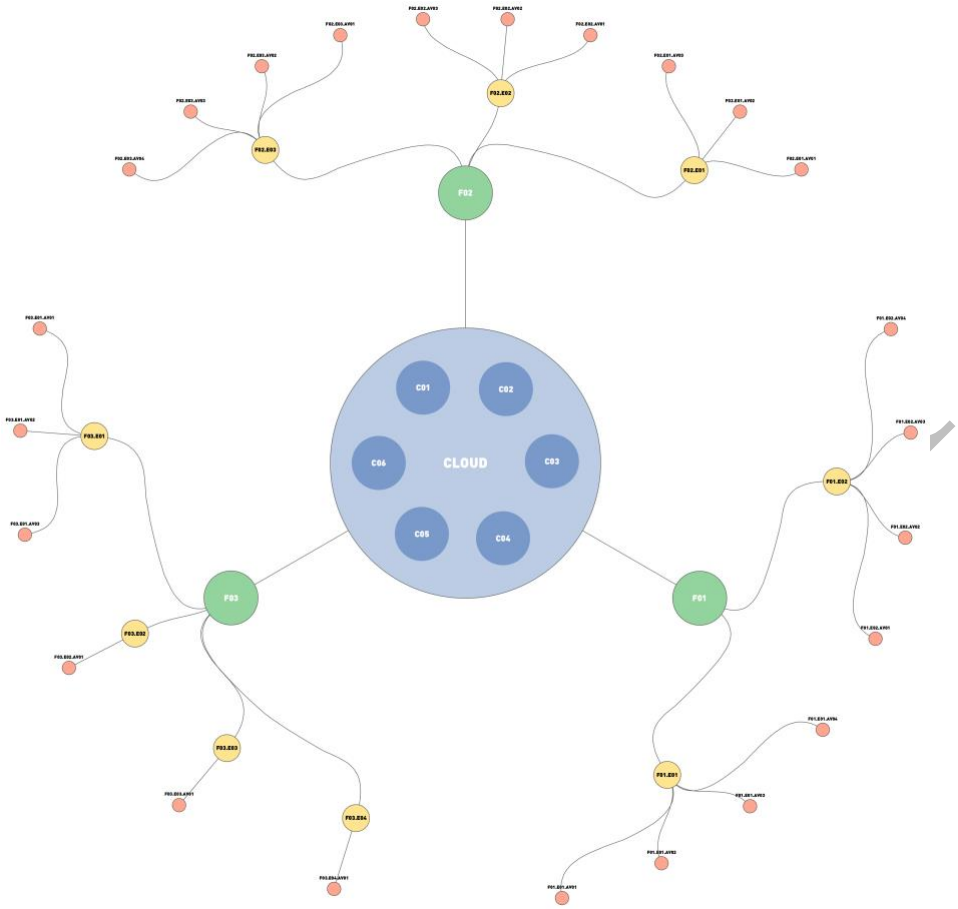

Figure 10: R2 use case example

Figure 11 presents a theoretical case of what can be anticipated if MARVEL were to be implemented at the scale of an entire city, resulting in potentially hundreds of AV sources and Edge nodes and tens of Fog and Cloud nodes.

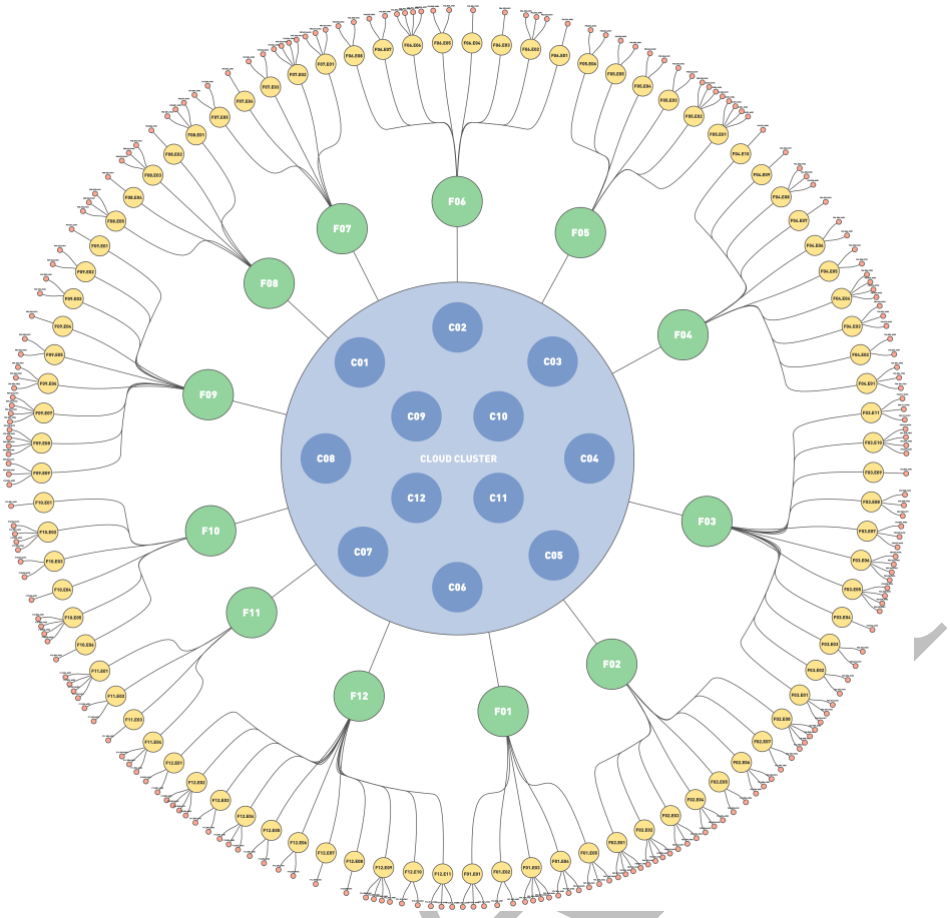

Figure 11: Extreme-scale use case

The aforementioned cases illustrated by the diagrams in the preceding figures demonstrate the rationale for the horizontal upscale of the infrastructure on which the MARVEL framework can be implemented.

Cases such as the first two were part of the current MARVEL implementation and have been extensively tested and validated, as reported in D5.5 and D5.6. The first case essentially corresponds to a single "branch" of the E/F/C structure of the MARVEL framework infrastructure. Several such "branches" are present in the second case and have been implemented in the actual current R2 deployment through the 10 use cases where MARVEL has been applied.

It could be argued that the overall horizontal scaling approach of MARVEL largely consists of adding E/F/C infrastructure branches with the possibility of sub-branching at each E/F/C level of nodes, i.e., having more than one AV source corresponding to each Edge node and having more than one Edge node corresponding to each Fog node.

Considering that the horizontal scaling logic is based on the proportional addition of nodes/devices and components/services in direct relation to the addition of AV sources while following the aforementioned "branching" structure, it can be concluded that **any further upscale from the current implementation that conforms to the same logic and structure should be feasible** (i.e., in the direction described by the third case in Figure 11), since this essentially corresponds to adding more "branches" to the E/F/C structure of the MARVEL framework infrastructure, a process that has in principle been validated by the R2 deployment.

Nevertheless, even though the scale-up of the MARVEL framework is well-defined from a methodological and structural point of view, in practice, any scale-up that consists of adding infrastructure nodes and component instances, also causes a significant increase in the complexity in terms of deployment management and may lead to risks associated with performance and stability if a proper scalability analysis had not been carried out in prior. The following sub-sections present the mechanisms, features, protocols and tools that have been implemented on an overall framework and on a per-component level that enable the scale-up methodology and support the management of the framework deployment while mitigating risks that emerge from scaling up.

3.3 Infrastructure and orchestration for horizontal scalability

3.3.1 Infrastructure and orchestration of the MARVEL components

The architecture of the MARVEL framework is centred around containers that are orchestrated by Kubernetes. Applications packaged in containers run consistently across different devices, ensuring uniformity in behaviour and reducing compatibility issues.

Containers provide process isolation, ensuring that each container operates independently. This isolation makes it easier to scale parts of an application without affecting the entire system. In the MARVEL framework, Kubernetes plays a fundamental role in scaling up operations seamlessly across the Edge, Fog, and Cloud layers. By utilising Kubernetes, MARVEL can dynamically and efficiently scale resources and workloads to meet varying demands.

Kubernetes' best practices[16] dictate that each node is allowed to handle 110 pods. Usually, one pod accommodates at least one container. This means that for every new node that we add to the Kubernetes cluster, we increase our pod availability by 110.

Each node has its own resources (CPU, RAM, etc.) that pods can consume. Based on the resources that each pod requests, it is deployed on the appropriate node. Kubernetes offers the object HorizontalPodAutoscaler[17] (HPA) that allows us to automatically scale up the number of pods based on the load. This synergy between Docker's containerisation and Kubernetes' orchestration enables MARVEL to adapt and scale responsively across its entire ecosystem, ensuring optimal performance and resource utilisation in all layers.

3.3.2 Component deployment with MARVdash

MARVdash, a Kubernetes dashboard, aims to automate the selection of deployment targets within the MARVEL E2F2C framework, utilising available resources across its layers. It allows YAML file creation by removing the necessity to declare deployment node specifics. Users now have the flexibility to choose a particular node based on deployment layers and component resource needs via the MARVdash User Interface (UI), including options like GPU selection. This choice instantiates the deployment on the specific node. Alternatively, users can opt for only the desired deployment layer without node specification making the deployment process agnostic to the scale of available nodes of the whole infrastructure. In this scenario, components are deployed on any suitable node within the layer that meets resource criteria.

[16] https://kubernetes.io/docs/setup/best-practices/cluster-large/

[17] https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale-walkthrough/

Furthermore, users can use the Template section of MARVdash to make use of the HorizontalPodAutoscaler feature of Kubernetes. Any template can be enriched with the HorizontalPodAutoscaler feature as shown in Figure 12. More specifically Figure 12 defines that at least one replica of the pod will exist at all times and when the CPU utilisation is above the defined threshold more pods will be instantiated until the maximum number of replicas is reached.

```
---
apiVersion: autoscaling/v2beta2
kind: HorizontalPodAutoscaler
metadata:
  name: $NAME
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: $NAME
  minReplicas: 1
  maxReplicas: 10
  metrics:
  - type: Resource
    resource:
      name: cpu
      target:
        type: Utilization
        averageUtilization: 20
```

Figure 12: HorizontalPodAutoscaler feature template snippet

To demonstrate the above functionality, we deployed a simple HTTP server and attempted to increase the load by making requests at a high rate. Using the script shown in Figure 13, we performed requests every 0.01 seconds to trigger a spike in the CPU of the HTTP server.

```
#/bin/bash

while sleep 0.1;
do wget -q -O- http://autoscale.karvdash-anthimpa.svc:8080;
done
```

Figure 13: Script for increasing load

Before we start the aforementioned script, the pod number is only one (Figure 14) because the workload is 0%. However, as soon as we start our script, the workload will gradually increase (Figure 15). Eventually, our script will create a workload of more than 20%. This will trigger

the HPA to increase the number of pods to match the workload (Figure 16). After the new pods have been created, the workload caused by our script is distributed among them and the average workload falls below the threshold of 20% (Figure 17).

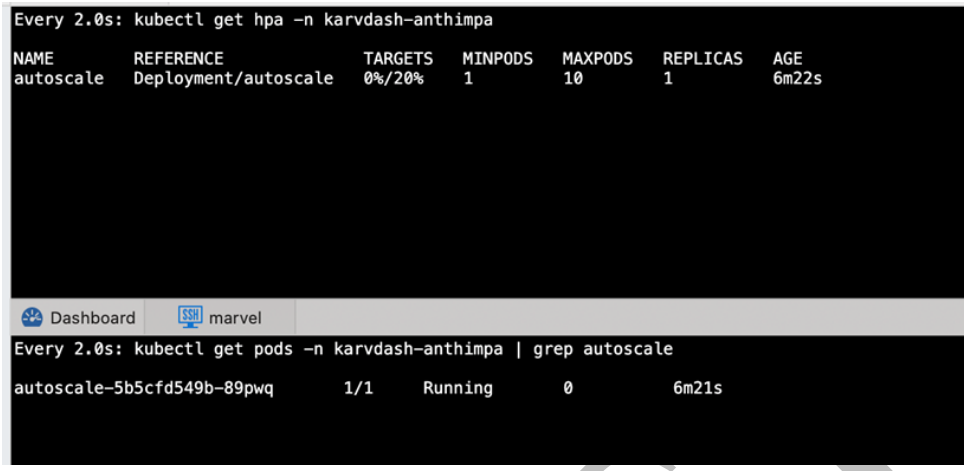

Figure 14: Status of HPA without load

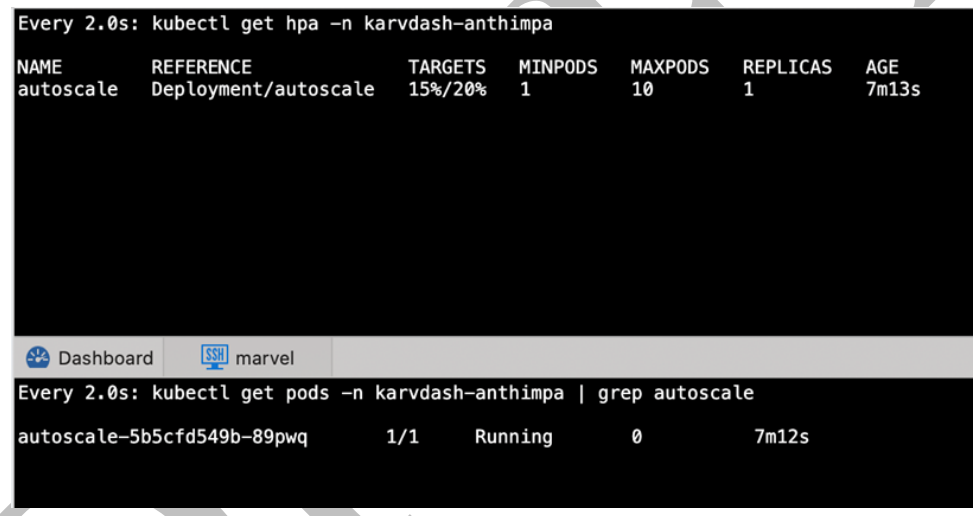

Figure 15: Status of HPA with load step 1

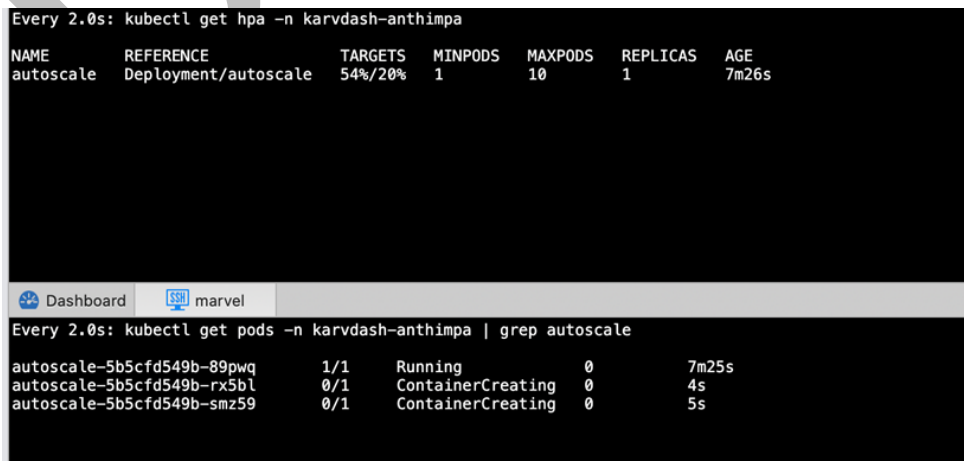

Figure 16: Status of HPA with load step 2

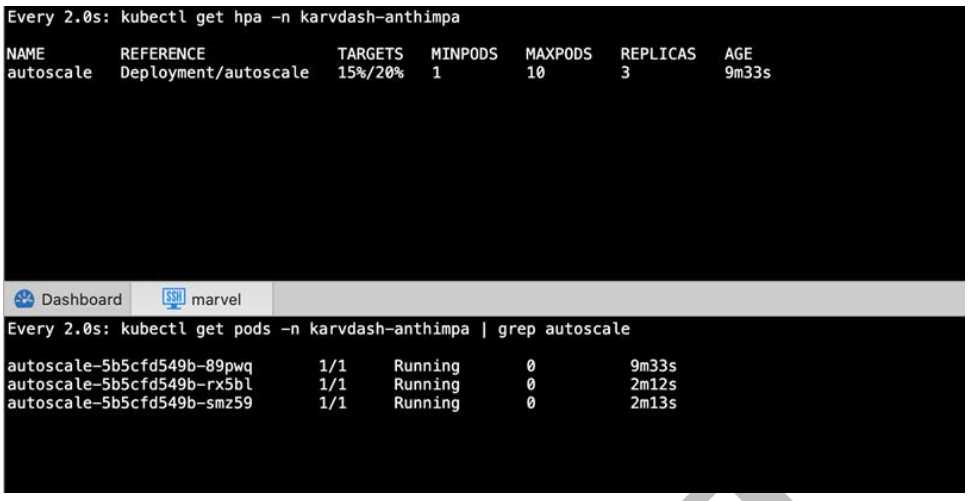

Figure 17: Status of HPA with load step 3

3.3.3 Monitoring and logging in MARVEL

In MARVEL, several logging and monitoring mechanisms have been integrated that allow the seamless supervision of all the components throughout the entire Edge-Fog-Cloud continuum. To this end, MARVEL's Kubernetes cluster is enriched with monitoring tools like Prometheus[18], Loki[19], Grafana[20], and Zabbix[21] and they are thoroughly described in D3.6[22]. These tools provide real-time insights, performance metrics, and proactive alerts, enabling efficient resource management and informed decision-making for optimised scalability and extensibility.

Prometheus is an open-source monitoring and alerting system specifically designed for monitoring highly dynamic and distributed environments using metrics. Loki is also open-source focuses on logs of the Kubernetes cluster, rather than metrics and is capable of horizontal scalability. Grafana is a popular open-source data visualisation and analytics platform known for its rich features and user-friendly interface. It is widely used to create interactive dashboards that help users gain deep insights from their data. Grafana complements Prometheus and Loki since it integrates seamlessly to retrieve and display the collected metrics and logs in visually appealing dashboards.

By selecting Grafana from the MARVdash menu, users can select from a list of prebuilt dashboards or create custom dashboards accordingly. By choosing for example a pre-built dashboard like "Compute Resources per namespace" (Figure 18), MARVdash users gain visibility into all instantiated pods, presenting metrics such as CPU Quota, Memory Quota, Memory Usage, Current Network Usage, Bandwidth, Packet Rate, Dropped Packet Rate, and Storage IO for each. Additionally, through Grafana, users can access the MARVdash dashboard displaying pod logs related to the MARVdash deployment. The MQTT dashboard, equally

[18] https://prometheus.io/

[19] https://grafana.com/oss/loki/

[20] https://grafana.com/

[21] https://www.zabbix.com/

[22] "D3.6: Efficient deployment of AI-optimised ML/DL models – final version," Project MARVEL, 2023. https://doi.org/10.5281/zenodo.8147021

valuable, visualises logs from MQTT brokers instantiated via MARVdash within the Kubernetes cluster.

Figure 18: GRAFANA dashboard

To enhance MARVEL's Kubernetes cluster monitoring capabilities, we deployed a Zabbix agent on each node of the Kubernetes cluster. Zabbix is a widely adopted open-source monitoring solution that delivers comprehensive monitoring, alerting, and visualisation functionalities. Zabbix provides real-time monitoring of crucial performance metrics based on templates. CPU usage, RAM utilisation, Network traffic, Disk space and GPU usage are just a few metrics that are collected and provide valuable insights into the infrastructure's health and performance (Figure 19).

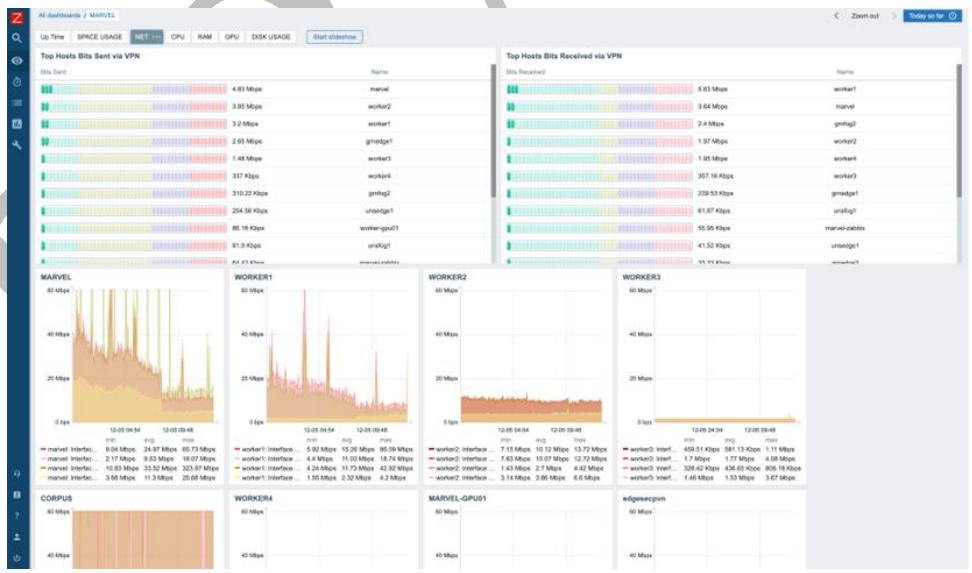

Figure 19: Zabbix MARVEL dashboard

3.3.4 Security in MARVEL

One of the requirements for successful extensibility is Security. Within the MARVEL project, data security is one of the major concerns and the Confidentiality, Integrity, and Availability (CIA triad) (Kim, 2022) aspects are covered with the use of secure communications channels by using EdgeSec VPN and with strict access policies to the MARVEL system. EdgeSec VPN,

as an n2n VPN solution, utilised within the Kubernetes cluster, contributes to scalability and extensibility by satisfying the fundamental requirement of the Kubernetes network model that all pods must communicate on any other node without the presence of NAT[23]. It employs a peer-to-peer (n2n) architecture, allowing nodes to seamlessly join the network. New nodes can easily connect to the VPN, enabling effortless scalability without intricate configuration changes. As nodes, in any layer, join the EdgeSec VPN network, they become part of a unified private network, enabling secure and direct communication between them. This setup ensures that scalability is feasible and does not compromise network security or create communication barriers. The use of n2n VPN facilitates the extension of the Kubernetes cluster across diverse environments or geographical locations as described in D4.5[24].

In the EdgeSec VPN, the architecture is based on two components: the Edge Node and the Super Node. The Super Node is responsible for coordinating communication between the Edge Nodes. To address scalability, the EdgeSec VPN supports load balancing and distributing the VPN connection across multiple Super Nodes.

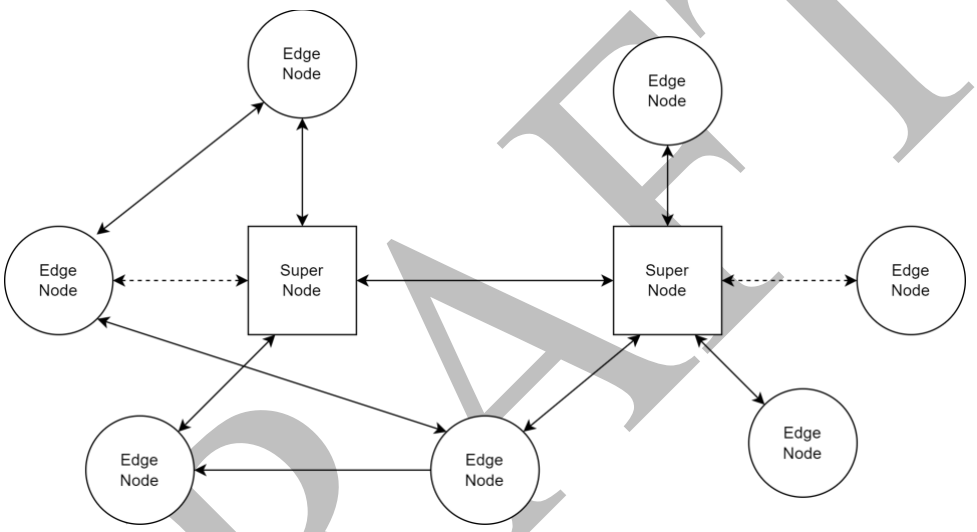

Figure 20[25]: Load balancing across multiple Super Nodes

3.3.5 Exploitation of GPU resources

The MARVEL infrastructure, spanning across the Cloud, Fog, and Edge layers, contains GPU-enabled virtual machines (VMs). Enabling GPU utilisation in a Kubernetes cluster enhances computational capabilities by leveraging GPU-accelerated resources for high-performance computing tasks. GPUs excel at handling certain workloads more efficiently than traditional CPUs, empowering the system to accommodate an increased workload without compromising performance. This scalability allows the execution of more services on the existing infrastructure and accelerates the completion of ongoing tasks. GPUs' parallel processing enables quicker analysis, training, and inference for AI models, leading to faster iterations, improved accuracy, and enhanced model development. Particularly in domains like artificial intelligence and machine learning, where scalability is crucial due to the growing complexity of algorithms and data volumes, leveraging GPUs in Kubernetes facilitates swift iterations and

[23] https://kubernetes.io/docs/concepts/services-networking/

[24] "D4.5: Security assurance and acceleration in E2F2C framework – final version," Project MARVEL, 2023. https://doi.org/10.5281/zenodo.8147058

[25] https://www.ntop.org/products/n2n/

improved model training. The parallel processing power of GPUs not only accelerates computations but also supports the seamless scaling of resources, making them ideal for scalable and efficient AI and ML tasks within Kubernetes clusters.

Furthermore, within the MARVEL Kubernetes cluster, the configuration is optimised for more than just GPU usage; it is designed for shared GPU access among multiple services. This design scales effectively, offering enhanced flexibility as the number of services grows. MARVdash, functioning as the Kubernetes dashboard, streamlines the creation of GPU-enabled services. This feature allows users to leverage the cluster's GPU resources, ensuring that as services expand, computational performance scales seamlessly across the board. This setup not only enables efficient GPU utilisation but also promotes scalability by facilitating shared access to these high-performance resources, thereby accommodating increased computational demands without compromising performance.

In addition, several optimisations dedicated to the cloud infrastructure for efficiently utilising GPU resources to enhance scalability and extensibility carried out for integrating cloud and High-Performance Computing (HPC) resources. Initially, accessing GPUs via the HPC queuing system proved impractical for MARVEL's needs. To address this, a dedicated GPU server was connected to the cloud through Kubernetes (MARVdash), enabling efficient resource sharing. To overcome HPC constraints, a VM was deployed on a dedicated server, establishing a bridge between cloud and HPC environments (Figure 21). Network communication challenges were tackled by exposing the HPC VLAN to Cloud Network Nodes, and creating a special router for seamless traffic flow. Key benefits of the developed solution include a uniform software stack for MARVEL VMs, direct data access for GPU applications, and connectivity through MARVEL Kubernetes for task submission. Despite its semi-static nature, the system facilitates the straightforward addition of new GPU resources. For more details, please refer to D5.8[26].

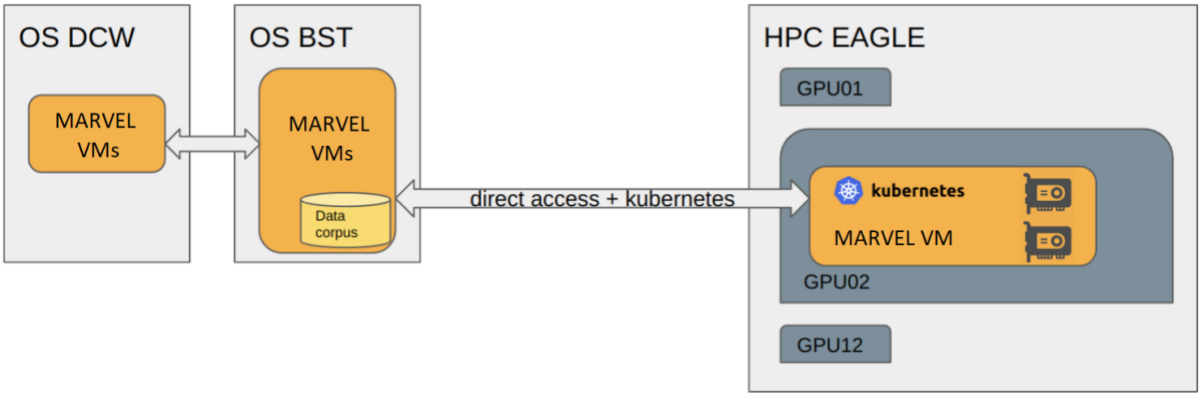

Figure 21: Schematic of cloud-HPC bridge

3.3.6 Management of Cloud Infrastructure

OpenStack[27] within the MARVEL framework is a robust open-source cloud computing platform, that is exemplary for scalability within modern cloud infrastructures. Its architecture

[26] "D5.8: HPC infrastructure and resource management for audio-visual data analytics – final version," Project MARVEL, 2023. To appear.

[27] https://www.openstack.org/

is designed to facilitate scalability, offering a versatile framework that allows organisations to expand their resources in response to evolving demands.

One of the main aspects of OpenStack's scalability lies in its ability to horizontally scale resources. This means that as workload demands increase, rather than merely boosting the capacity of existing infrastructure, OpenStack permits the addition of new nodes or servers, allowing for seamless expansion. This horizontal scaling approach ensures that organisations can accommodate growing workloads efficiently by augmenting compute, storage, or networking resources as required, without encountering the limitations of vertically scaling single components.

Moreover, the modular nature of OpenStack contributes significantly to its ability to scale. This modular design allows for the independent scaling of individual components based on specific needs. For instance, if there is a surge in computational requirements but storage needs to remain constant, organisations can scale up the compute component without impacting the storage infrastructure, ensuring resource allocation precisely where it is needed without unnecessary expansions or constraints.

Furthermore, OpenStack ability to be adopted across diverse cloud environments boosts its scalability. Whether it is expanding within a private cloud environment or integrating with public cloud resources, OpenStack offers the flexibility to scale resources dynamically to address fluctuating workloads effectively.

3.4 Sensor node resource management

In this section, the scalability and extensibility of the MARVEL components related to the AV source handling, management and storage will be described. Particularly, the issues related to the Edge, AV sources, and their streaming to the Fog layer, alongside the mechanisms used to manage these devices will be discussed in terms of scaling up. Additionally, the AV Registry and storing the streaming content, StreamHandler, will be explored for their contributions to supporting a large number of sensor devices.

This section is related to the AV Source Discovery, Management, and Access described in Section 2. Possible mitigation solutions for this requirement include the implementation of a centralised management system utilising metadata for efficient AV resource discovery and access. Moreover, with the use of distributed storage solutions such as object storage, Content Delivery Networks (CDNs), or similar state-of-the-art, data-efficient storage mechanisms becomes imperative to accommodate the growing volume of AV data. In addition, to the storage mechanisms, a data lifecycle management strategy can be used to optimise the storage utilisation, e.g., a periodic elimination of the AV files that are not required anymore to be present in a specific machine. Within the MARVEL framework, these solutions were considered and adopted to the final system. Below the technologies and how they were used are detailed.

The RTSP enables an easy and efficient way of utilising a vast number of AV devices in the surveillance and streaming era. The versatility of the protocol enables seamless integration, remote access, and efficient management of a diversity of AV hardware, leading to a robust and efficient surveillance system.

RTSP significantly contributes to the scalability of the MARVEL framework at the Edge layer by providing a comprehensive framework for handlining simultaneously an extremely large number of AV streams. This is achieved by the protocol’s inherent characteristics. First of all, RTSP allows the feed of numerous AV sources to be collected by a single node. This centralised approach is important in large-scale deployments such as smart cities, where several cameras

from a specific region of interest, district X of a municipality, can be connected to a single Fog server. Another benefit of RTSP is that significantly contributes to a large-scale deployment where the underlying hardware, cameras, microphones, etc, comes from different vendors and the AV specifications vary. By adopting a protocol, such as RTSP, the burden of supporting the various proprietary protocols by each hardware manufacturer is addressed. This leads to a more scalable and flexible system that easily can adopt the newly added AV sensor nodes without the need for extensive maintenance. In addition to the large number of devices supported, RTSP can be easily integrated with any AV management system, such as StreamHandler due to its standardised interfaces.

AV Registry is a component developed by ITML and it serves the purpose of collecting and making available to the relevant components all the information required for accessing and managing the various AV resources. This is efficiently achieved by exploiting two major technologies. The information on the AV resources is stored in a JSON-oriented space by the AV Registry. This allows housing a vast number of records with a minimal resource footprint. In addition to that, this solution also enables an easy addition of new nodes' information as the new record following a specific template can be added to the existing storage and become available to the relevant components. AV Registry makes the information stored available to other components through a RESTful Application Programming Interface (API), a technology that is widely used in the inter-component communication. More specifically, two endpoints are available through the AV Registry service, one containing all the available AV resources and their information and one that returns the information of a specific AV resource. An example of the AV Registry's payload for a specific resource is presented in Figure 22. AV Registry contributes to the scalability and extensibility of the MARVEL framework by storing and making available critical information for the Edge resources and it is built in a way that scales up either in one single node, by its low footprint or in a horizontal scalability scenario by deploying one instance of AV Registry per each Fog node. More information regarding the insights of AV Registry can be found in the deliverables D2.4 and D5.6.


```
{
  "id": "Cam-GRN-VA-01",
  "type": "Camera",
  "AVSourceType": "VA",
  "owner": "GRN",
  "cameraNum": "VA-01",
  "cameraName": "GRN-VA-01",
  "streamName": "GRN-VA-01_Mgarr_VideoAnony",
  "streamURL": "rtsp://rtspserver-grnedge1-nodeport.karvdash-lucadv1fbk.svc:8554/GRN-VA-01_Mgarr_VideoAnony",
  "cameraManufacturer": "FBK",
  "cameraModel": "VideoAnony_GRN_E1",
  "location": {
    "type": "Point",
    "coordinates": [
      14.36671,
      35.920308
    ]
  },
  "cameraOrientation": {
    "comments": "Camera facing East"
  },
  "contentType": "V",
  "videoAnonymized": true,
  "originalAVSourceId": "Cam-GRN-CCTV-01",
  "videoResolutionWidth": 1920,
  "videoResolutionHeight": 1080,
  "videoFramerate": 8,
  "videoCodec": "H264",
  "videoBitrate": 1000,
  "AVContainer": "MP4"
},
```


Figure 22: Sample Information stored and published by AV Registry

StreamHandler developed by INTRA is an AV management system deployed in the Fog layer. Due to its design, StreamHandler has inherent features that contribute to the scalability and

extensibility of the MARVEL platform. StreamHandler is designed based on Microservices architecture, which allows splitting each functionality to a separate microservice thus achieving a high degree of scalability. Figure 23 demonstrates StreamHandler's architecture. To this end, if a bottleneck is detected, another instance of that service can be deployed to guarantee the performance of StreamHandler in an increasing workload scenario. StreamHandler can be scaled up using horizontal scalability in two ways, either by replicating the bottleneck service in the same node or by replicating the entire StreamHandler system to several Fog nodes. In a per microservice breakdown, StreamHandler consists of state-of-the-art highly scalable and flexible technologies. To this end, the segmentation microservice is making use of a relatively low footprint solution to capture the AV streams and segment them into files. These files are stored in a storage pool that can efficiently handle input/output interactions of a large volume of files. The stored files can be accessed by other components through a RESTful API that provides on-demand AV file generation and offers a uniform resource locator (URL) of the file that can later be processed in several ways, such as being displayed by a Graphical User Interface component such as SmartViz. Further details on the architecture and functionality of the StreamHandler can be found in the deliverables D2.4 and D5.6.

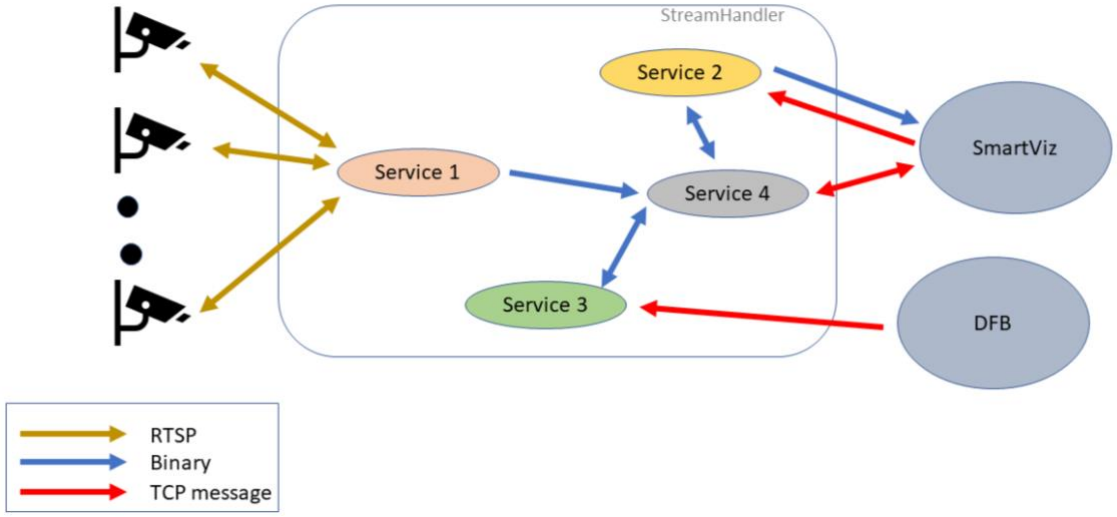

Figure 23: StreamHandler's microservices schematic

3.5 Anonymisation and AI optimisations

In this section, the scalability of the anonymisation and the AI components' optimisations will be described. The requirements that are covered in this section are the **Increased number of streams to be anonymised and the Increased number of streams to be processed by the AI components. With the addition of a large number of cameras, the Anonymisation and AI components are the first part of the 'MARVEL inference pipeline' that will be overloaded.** Due to the design of anonymisation components, a different instance of such components needs to be applied for the anonymisation of each AV source to guarantee that the performance will be of a high level while eliminating the risk of non-anonymised data leakage and the burden introduced by using and maintaining advanced encryption and privacy-preserving techniques. After anonymising the various streams, the AI components will process the feed which may create several issues. The issues can be overcome with the use of parallel processing and distributed computing frameworks that enhance the scalability of AI components. Moreover, techniques like containerisation, placement of the AI components as close as possible to the

AV sources and the modularity of the AI components can significantly increase the performance and handle the additional workload.

The video anonymisation component is particularly important in the MARVEL framework as it is the first processing block, and it is typically deployed in the first processing node of the pipeline. The most demanding processing component is a face and plate detector based on Yolo[28] which requires a GPU for processing video streams with a 10-12 frames per second rate. The component has been optimised in three ways:

- The code has been implemented in a way that the same component can process multiple streams simultaneously, reducing the memory requirement, while the computation is not affected.
- The code was optimised with the Nvidia TensorRT[29] library. This library aims to optimise and run high-performance deep learning models, to fit the computation available on the Jetson Nano. However, this effort was only partially successful as it was able to achieve a maximum of 2 frames per second.
- A very lightweight version was developed, also featuring face-swapping instead of blurring, suitable for micro-controllers. Details are reported in (Ancilotto, Paissan, & Farella, 2023) and in the deliverable D3.5[30].

For audio anonymisation, two research directions were followed during the development of the AudioAnony component: developing cutting-edge neural voice conversion algorithms and implementing lightweight signal processing-based voice modification methods. The latter, in combination with VAD from devAIce, is used in the prototype and can comfortably run on a Raspberry PI. Given that the Micro Electro-Mechanical Systems (MEMS) microphone arrays used in MARVEL require a device for audio acquisition, AudioAnony allows extending the network of audio sensors without the need for any additional hardware.

In the same domain, AU optimised the inference (capabilities) of the AI components by replacing the disk-based readers of audio and video streams input, reducing the models' latency at the first frame from 23.26 seconds to 1.5 seconds and the latency at the 10th frame is 0.35 seconds. Furthermore, AU improved the AI models by introducing smaller models. This includes a compressed AV Crowd Counting (AVCC) model developed by CNR and integrated by AU. The compressed AVCC model performs 1.8 times faster than the uncompressed model. Additionally, AU developed a Visual Crowd Counting (VCC) model with Early Exit Branches which reduces the number of parameters in the model and the required GPU memory to use the model. AU developed an Unsupervised Visual Anomaly Detection (ViAD) method based on structured parameter pruning on Memory-augmented Deep Autoencoder which improves the model performance by 1.2% and reduces the model size by 71.78% and the number of floating-point operations (FLOPs) by 21%. In this context, AU contributes to the MARVEL scalability by reducing the number of required resources and allowing for the horizontal scale-up of the relevant pipeline by replicating the models' instances to handle a higher number of AV inputs. Alongside the improved models' performance, the relevant inference pipeline is capable of handling efficiently an enhanced number of AV input streams.

The architecture for the AI components was kept small to account for deployment in all different layers, cloud, fog and edge. For the SED and AT components, the deployed

[28] https://pjreddie.com/darknet/yolo/

[29] https://developer.nvidia.com/tensorrt

[30] "D3.5: Multimodal and privacy-aware audio-visual intelligence – final version," Project MARVEL, 2023. https://doi.org/10.5281/zenodo.8147164

architecture relies on a state-of-the-art Convolutional Neural Network (CNN), specifically implementing a CNN10 architecture with 4.95 million parameters. In the case of SELD, the system features three CNN blocks, followed by two bi-directional recurrent layers and a Multi-Head Self-Attention (MHSA) block, resulting in a total of 740 thousand parameters. These components have undergone testing across various layers of the MARVEL architecture and demonstrated successful deployment on both GPU and CPU platforms. However, the AAC component's compatibility with different hardware configurations on the CPU has not been experimentally verified. When addressing Automated Audio Captioning (AAC), contemporary systems leverage cutting-edge deep learning methodologies, employing neural networks with notably large parameters exceeding 100 million. The large number of parameters makes it difficult to deploy the component on the CPU.

The process of anonymisation has minimum impact on the SED, AT, SELD, or AAC components, given that audio data itself, and not personal information or sensitive data like speech, is being processed. As a result, the performance of these components remains unaffected by the anonymisation step. The absence of sensitivity to anonymisation contributes to the adaptability and reliability of the MARVEL framework in handling audio data processing requirements. In a scaled-up version of a MARVEL framework deployment, this contributes to increased stability of the system, as potential points of failure are eliminated (e.g., audio AI components yielding unexpected behaviour when fed with anonymised data).

CATFlow uses on average two to three GB of memory, one to two logical CPU cores and 1.0 to 1.5GB of GPU memory per camera. The resource allocation estimates were obtained by monitoring the resource usage using htop and nvidia-smi tools for different cameras. The fluctuation on different machines for the GPU and the CPU depends on the characteristics of the hardware. If a low-resourced CPU is used more cores are required to satisfy the demand. Similarly, if a low-resourced GPU is used more VRAM will be used to satisfy demand. The code is compiled using Cython to improve speed and some sections were refactored to reduce code duplication. The TAD component accesses the text output from the CATFlow component and sends a message if a speed value is considered anomalous. Since the TAD component performs simple numerical operations the component can be considered light weight. If the CATFlow component is already part of the architecture, the addition of the TAD component would comparably not consume a considerable amount of resources, thus it can be considered easily scalable.

3.6 Handling real-world impure data in large scale

In this section, we analyse the fault tolerance aspects of MARVEL components against inconsistent AV streams and/or inference results that would contribute to the application of the components in real-world scaled-up scenarios.

In real-life deployments of the MARVEL framework, it should be anticipated that there may be failures at the level of data sources, failures in data transfer channels or other inconsistencies in the transmitted data that in many cases cannot be foreseen. When scaling up, the potential relevant points of failure increase significantly. Considering that the system’s operation is based on the processing and distribution of data, it is of paramount importance that the system can be robust and absorb the impact of any such data failures without risking a complete failure of the overall system operation and integrity while being able to recover from any individual failures that occur. This sub-section presents the relevant mechanisms that have been implemented to address such issues.

3.6.1 Anonymisation and AI components

AU implemented Fast Forward MPEG (FFMPEG) and GStreamer audio and video RTSP readers. To battle errors in RTSP streams, such as unstable connection, broken packets or unsupported data frames, the readers are implemented in separate threads and are monitored by a controller thread. If one of the readers fails due to any reason, the other reader is asked to be closed and then both readers are restarted after the streams probed with FFProbe. If one of the streams does not produce enough data, there can be a situation where not enough AV pairs are created for a long period of time. In this case, the stream readers are forcibly restarted. AU tested FFMPEG and GStreamer readers and the latter option were shown to be more stable, which is the reason it is selected as a default option.

In a real-world Smart City scenario, a large number of stream inputs is taken for granted. However, due to the versatility of the hardware to be used as well as the networking technologies to be used, several issues may arise, with the most critical being the fluctuations in the frame transmission process that may lead to several frames being dropped. In the case of minor glitches like a few frames missing in the stream, the CATFlow component can self-adjust and tracking is sustained. However, if a large number of frames are missing then CATFlow will not handle them as the scene would be too different and tracking restarts soon after. The CATFlow component can be executed at low frame rates to simulate a lack of info between frames. The TAD component is not affected by glitches in the video stream. If CATFlow is able to produce a message, TAD can process it.

3.6.2 Management of inference results

Regarding the management of inference results from the AI components, the approach followed in MARVEL allows a common way of interacting via one of the Data Management Platforms: DatAna. DatAna is an Apache NiFi, MQTT broker-based tool deployed in each of the layers of the MARVEL infrastructure where inference takes place. Deliverable D2.4 includes the diagram shown in Figure 24 depicting an example of topologies of DatAna nodes used in the MARVEL pilots.

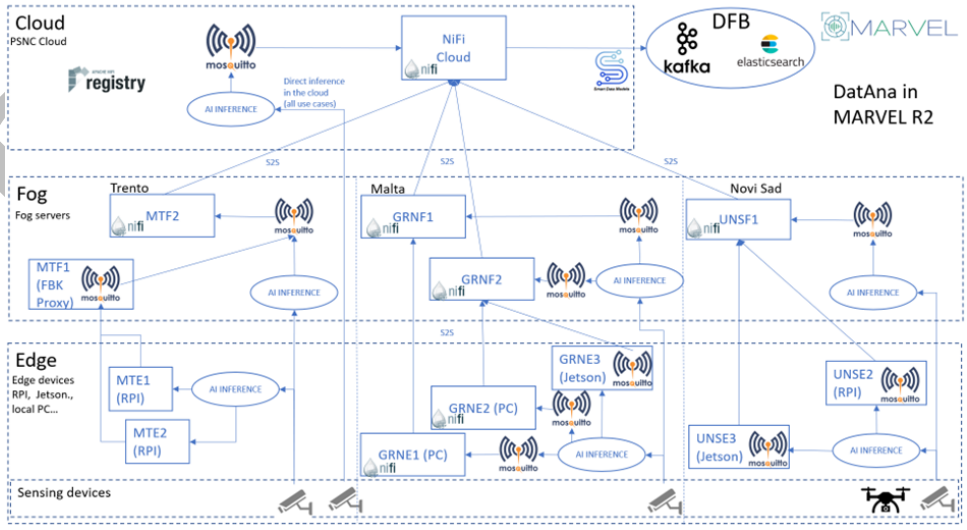

Figure 24: DatAna topologies for R2 (from MARVEL D2.4)

The process followed to manage the inference results is the following:

- The inference components output their results as JSON messages to the "nearest" MQTT broker to where the inference data is produced. For example, if the inference is

taking place in the GRN fog layer, the data from that inference component will be sent as a JSON message to the MQTT broker deployed in that fog server, allowing the process of inference data to remain at the device, reducing the risk of information loss.

- Then, an Apache NiFi instance of DatAna is usually deployed in the same machine, at the Edge, Fog or Cloud layers. Specific data flows are running in the Nifi instance subscribed to the MQTT messages from the inference components.
- These data flows proceed with the validation and transformation of the inference data to the agreed data models in MARVEL for Alerts, Anomalies or MediaEvents. The standardised inference messages are then sent to the cloud via NiFi secure protocols (S2S) and then transmitted to the DFB for further processing. Note that in some cases, the NiFi instance is not deployed at the edge to minimise the footprint in the edge devices, and the NiFi in the fog layer is performing the same functionality.

The architecture of the inference data collection provides a high degree of flexibility, fault tolerance and scalability properties to the system. The main properties are the following:

- The system is highly decoupled both horizontally and vertically, meaning that all the inference and Data Management Platform (DMP) components in the data processing pipeline run as autonomous services relying on Pub-Sub paradigm over messaging systems at each of the E2F2C layers.
    - If one of the inference components fails, the system will continue the processing of the rest of the components.
    - If one of the DatAna nodes fails, the rest of the nodes will continue the data processing.
    - The only critical point is in the cloud. If NiFi is down, then the data collection would stop after the NiFi queues become full in the fog and edge layers. If the components are restored to run before that happens, there will be no data loss. Another solution for mitigating such risks is for the DatAna agents at the fog level to communicate directly with the DFB Apache Kafka cluster. Furthermore, Nifi in the cloud can be deployed in cluster mode, exploiting all available VM resources that would significantly increase its reliability and scalability.
- Data transfer between DatAna instances is secured and monitored. This is managed via the NiFi Site-to-Site protocol (S2S)[31], an internal and secure data transfer mechanism enabling communication among NiFi instances.
    - If the communication fails, an error is raised in the NiFi dataflow that can be monitored via the NiFi user interface. Recovery actions such as reinitiating the data flow or digging into logs to figure out the problem can be done from the graphical user interface (GUI).

The data is routed according to the dataflow depicted in the NiFi GUI, replicating the topology depicted in Figure 24.

3.7 AI training

In Section 2.2.1, the requirement of more datasets for the AI training process and its issues were presented. To overcome these issues that may arise, several steps need to be taken. First of all, instead of performing the training process on a single node, distributed training frameworks can be used to efficiently distribute the workload over several nodes. In addition, this procedure can take place in the cloud-based environments where computation and storage are inherited scalable. To tackle the dataset-originated issue, data augmentation techniques can be used to

[31] https://nifi.apache.org/docs/nifi-docs/html/administration-guide.html#site-to-site-protocol-sequence

enhance the datasets' diversity of the training data. In this section, we will describe how the training of the AI components pipelines can be optimised for large-scale implementations. In particular, the contribution of FedL, DynHP, and MARVEL Data Corpus are reported.

In a system comprising numerous interconnected heterogeneous components within the same network, and distributed within edge, fog, and cloud layers, such as MARVEL, one of the major goals is to minimise communication while retaining high performance of the deployed models. As edge devices are usually constrained and may not have a stable communication channel with other components in a Federated Learning (FL) setup, a non-uniform sampling (NUS) federated learning strategy was implemented in the FedL component of the MARVEL architecture as an alternative to the widely used federated averaging (FedAvg) strategy.

The FedL component of MARVEL enables scalable solutions, with potentially large-scale configurations, by the inherent design of federated learning algorithms. This is achieved by the following inherent features of federated learning: parallel execution and incremental training.

Parallel execution: Each training round of the overall FL method (and hence of FedL) works in such a way that all currently active (selected) clients at the current round perform their updates simultaneously, in parallel. Therefore, adding new clients does not significantly increase execution time per FL round, except for the usually inexpensive model aggregation (averaging) step done at the server. However, to fully ensure scalability as new clients join the system, communication efficiency (avoiding communication bottleneck) has to be taken care of. The non-uniform sampling strategy of FedL, described in more detail below, contributes to achieving communication efficiency of the overall FL system.

Incremental training: An additional design aspect of the federated learning training (and thus of FedL) is that the models are naturally trained incrementally. In other words, given a current model, each training round incrementally improves (changes) the model, essentially by doing a gradient descent over the loss function that is being minimised. Hence, when a new client (or a group of new clients) joins the FL system, the model does not have to be trained from scratch, but rather only a few additional training rounds (as required) can be added.

In addition to parallel and incremental training capabilities of FedL, in a system comprising numerous interconnected heterogeneous components within a same network, and distributed within edge, fog and cloud layers, such as MARVEL, one of the major goals is to minimise communication retaining high performance of the deployed models, hence taking care of the communication bottleneck challenge. As edge devices are usually constrained and may not have a stable communication channel with other components in a federated learning setup, non-uniform sampling (NUS) federated learning strategy was implemented in the FedL component of the MARVEL architecture as an alternative to the widely used federated averaging (FedAvg) strategy. The proposed strategy not only provides data savings, thus leading to the narrowed required bandwidth for communication but also enables the possibility to continue training in the case if some of the clients are offline due to the unstable connection. Implementation is done using Flower (flwr) library, as it represents an adaptable and extensible federated learning library for Python, which enables custom strategies.

The adaptive FedL NUS protocol achieved improvements regarding training speed compared to the standard protocol. Experiments were performed on the LEAF dataset, which represents a standard benchmark, and on the MARVEL data on the example of the visual crowd counting (VCC). For clients that exhibit heterogeneities, the proposed strategy achieved significant improvements in terms of training speed compared to the baseline FedAvg. In the case of the LEAF benchmark, 33% of data saving was achieved, i.e., FedL NUS model required transmission of 80 megabytes (MB) compared to the 120 MB, required by FedAvg. The

accuracy achieved using the proposed FedL NUS is 88%, which is only 2.3% relatively lower compared to the accuracy of 90%, which is achieved using the baseline model.

In the VCC experiment, we have implemented federated learning considering that each MARVEL pilot is a single edge node. It was observed that the proposed NUS strategy outperformed non-federated setup (local training) as well as implementation of the baseline FedAvg. The achieved reduction of the data transfer over one training process of 100 epochs is quite large, as the VCC model is large as well. We notice that in the case of the proposed NUS strategy, the reduction in model parameter transfers is around 23 gigabytes (GB). Besides this, achieved accuracy was increased by using the global model in two out of three clients compared to the non-federated setup, which is one of the main goals of implementing federated learning.

Another component that contributes to the training pipeline is DynHP from CNR. DynHP is a technique for compressing AI models, suitable to run at different levels for the Edge-to-Fog-to-Cloud continuum. Its internals have been already presented in D3.2[32] and in a related publication in the Elsevier Journal of Computer and Network Applications (Lorenzo, Nardini, Andrea, & Raffaele, 2022). Here we report only the details that are pertinent to this Deliverable.

Differently from the typical approaches for model compression that typically perform it once a Deep Neural Network Model is already trained offline and often in Cloud, with DynHP, it is possible to do both, i.e., training and compression, at the same time. Such an approach brings several advantages because, in this way, it is possible to move the training process closer to the data sources, e.g., the Fog or the Edge. Edge or Fog devices can train and reduce the size of a model using the local data, getting in the end a smaller and more efficient model that can be deployed to resource-limited edge devices. From the experimental results obtained from the MARVEL use case GRN4, where the task is AV crowd-counting, we observed a 1.8x speed up of inference time when we used the model compressed through DynHP, compared to the time needed by the original and uncompressed AVCC model.

Beyond improving over the single inference task, enabling Edge or Fog-friendly compression in combination with distributed training approaches such as Federated Learning might also impact the traffic generated for data transmission and collection. Specifically, the combination of the two approaches (distributed training + compression) not only prevents the transmission of raw training data through the network infrastructure, but it might also reduce traffic overhead generated by the exchange of models during the Federated Learning process, that is proportional to the model compression obtained during the process.

Finally, a component that is not directly contributing to the training pipeline as FedL and DynHP do, is Data Corpus. The MARVEL Data Corpus can be also engaged in the ML procedures. The core offering is to act as the main repository for the datasets that have been produced during MARVEL. These include both private datasets (internal use for the project's AI components) and public ones (use by external stakeholders). Corpus has been designed to store data in the PB scale. If a large virtual hard disk had been created, it would exhibit very low performance due to high delays in seek, read, and write operations. Therefore, the core repository was built upon the Hadoop Distributed File System (HDFS). Virtual Machines (VMs) with 100-250 TBs hard disk can be added to the HDFS dynamically, acting as Data Nodes and fulfilling the current Corpus needs for storage. At M36, around 8 VMs were added to the Corpus, reaching a total size of around 1.1PBs. The size can be easily extended to reach 3.3PBs or more.

[32] D3.2: Efficient deployment of AI-optimised ML/DL models – initial version, Project MARVEL, 2022. https://doi.org/10.5281/zenodo.6821232

The Corpus supports a scalable Big Data repository, where high volumes of data can be stored and retrieved in high volumes. A stored file can be replicated in different Data Node VMs. Thereupon, different AI processes can read the data in parallel and independently from different nodes. Those read requests are made towards the HDFS's Master Node, which performs load balancing and forwards the request to a related Data Node, with the process being transparent to the requester.

Moreover, the Corpus component offers an Augmentation Engine. With this, the user can produce augmented versions of the ingested datasets. This is a very useful feature for ML experts, who can enhance the training data with conditions that are not included in the original datasets (e.g., different brightness or weather conditions) and improve the effectiveness of their AI solutions. The Augmentation Engine supports a series of augmentation libraries for audio and video. These libraries support internal parallelization of the augmentation algorithms in multi-core CPUs and GPUs. Also, a scalable augmentation strategy has been implemented, where augmentation of different files and different augmenters for each selected file can be computed in parallel.

3.8 Distributed Inference Results collection

3.8.1 Inference results collection and distribution process

This section will be focused on the collection mechanisms for inference results and specifically how they can respond to a high number of incoming data. In particular, the contribution of DatAna and the MARVEL architecture configuration will be reported.

To overcome these challenges, several approaches can be used. First of all, a robust and fault-resilient mechanism capable of transferring high volumes of inference results can be used. As an alternative, filtering mechanisms can be deployed close to the AI components to compress the inference results messages exchanged, i.e., the inference data can be sent into batches.

Section 3.6 explained the process followed in the DMP to manage the results from the inference components. DatAna plays a vital role in it, as the main component that receives all inference data in each of the devices of the MARVEL E2F2C infrastructure, transforms the data to comply with the agreed data models and moves it to the cloud and the DFB for further fusion, re-distribution and storage.

This functionality provided by DatAna is quite scalable and fault-tolerant. Both MQTT and NiFi provide queuing mechanisms to avoid data loss if one of the components of the system is not responding or there is a spike in the amount of data received. The data are therefore queued and processed once the fault components are back online. Needless to say, if the number of messages in the queues becomes very high, at a certain point the system could not cope with the data retention. This would happen only in rare cases and can be easily monitored to increase the capacity of the queues or to react to put back on line the faulty components.

In MARVEL, it was decided to use a relatively simple installation of NiFi in a single node. This is going to remain this way, especially at the edge layer, and probably at the fog layer, as a cluster deployment of NiFi might be an overkill for relatively limited devices or servers. If the number of edge devices increases or the inference data becomes too high, instead of routing all the messages from the devices to the same fog server it would be advisable to set up a new fog server with a DatAna instance able to handle the scale of messages expected from a limited set of devices.

As a central point of inference data collection, the NiFi instance in the cloud should be deployed in cluster mode in production environments, where a high amount of processing and messages

from many fog and edge deployments are likely to happen. According to some benchmarks[33], a NiFi cluster might be set to serve as much as a billion events per second, giving ample room for future scalability. Setting up a NiFi cluster involves the provision of new hardware elements (VMs or servers) in the cloud and the deployment of the NiFi instance in cluster mode. The documentation for administrator provided by Apache NiFi[34] explains in detail how to deploy NiFi in cluster mode. In the case of MARVEL, this will involve updates to the configuration file (YAML file) of the Kubernetes NiFi service managed by MARVdash, to ensure that the service is deployed in cluster mode. The security credentials must be updated to enable secure communication using the NiFi Site-to-Site protocol (STS)[35] between the new cloud NiFi cluster with the underlying NiFi nodes, currently pointing to the single node NiFi service. This is standard NiFi configuration and the guidelines on how to do so can be found in the NiFi administration guide as well.

NiFi provides a graphical user interface to monitor the status of the different data flows as well as allow changes in the configuration of the flows and some parameters even during execution time in production. After the delivery of the messages to the DFB, this component relies on Apache Kafka. Kafka also provides mechanisms for queueing, scaling up and parameters for fine-tuning the data management.

3.8.2 Extensibility of the approach

From an architectural perspective, the DMP provides extensibility points to ensure that the MARVEL framework could be enhanced in the future to incorporate results from new AI inference modules as well as manage other types of data with minimal development effort.

Extensibility to incorporate new AI inference component results

The usage of a highly decoupled architecture for the DMP enables the possibility of incorporating or updating inference components with no or minimal changes to the framework. As explained, the collection of inference results involves the publication of these results by the inference components in an MQTT broker-specific topic, from which a dataflow in NiFi is subscribed and processes the data. The data is finally transformed into the MARVEL-agreed data models and sent to the DFB for further fusion and storage.

Within MARVEL, the extensibility approach has been proved by including new AI inference components (for example, several new components were added to the architecture in the R2). The addition of a new inference component entails the following requirements:

- Addition of a new MQTT and Kafka topic: The component should drop the results of the inference in an MQTT-specific topic. In MARVEL, we followed the pattern of naming the topic as the component in lowercase (e.g., “avad” topic for the AVAD component).
- A new dataflow should be designed in the Apache NiFi close to the data source (at the edge, fog, or cloud). The new dataflow can be copied from existing ones, as the structure of the flow would be very similar. The processors should be configured to read from the specific topic of the new inference model, and the transformation

[33] https://blog.cloudera.com/benchmarking-nifi-performance-and-scalability/

[34] https://nifi.apache.org/docs/nifi-docs/html/administration-guide.html#clustering

[35] https://nifi.apache.org/docs/nifi-docs/html/user-guide.html#site-to-site

processors configured to perform the necessary changes to the data to fit the MARVEL data models.

The addition of a new inference component in DatAna usually can be done in a few hours of configuration and testing, with zero-code effort, providing that the developer is familiar with the NiFi user interface.

Extensibility to integrate new data source types

Section 3.9 explains potential extensibility points provided by the DFB and its underlying technologies (Kafka and Eleasticsearch) to new data sources. The same applies to DatAna. While the DFB may be used to get new data already processed and formatted with minimal development effort, **DatAna offers off-the-shelf processors to get data from multiple data sources** and easily provide acquisition, transformation, communication, and connection with multiple storage mechanisms **with almost zero-code effort**. This is done via the NiFi graphical user interface, which enables the creation of updates of data flows by simply dragging, dropping and configuring processors that provide functionalities to get and manage the data. The NiFi user interface is quite easy to use by developers and on most occasions only requires chaining and configuring the available data processors to set up a new data flow, with no need of developing any code. This can be very useful if the need arises to include data in MARVEL that needs to be processed. These data may come from new sensors, databases or services. DatAna also offers plenty of possibilities to store the results of the process in other repositories, databases, or communicate with other systems besides the DFB.

The complete list of NiFi off-the-shelf processors is available in the NiFi documentation[36], but there are even more processors that can be added to the NiFi default deployment by adding specific .nar files, each of them containing processors to handle data from even more systems (e.g., from Elasticsearch, Flume, Datadog, Confluent, Cassandra, MongoDB, etc.). Developers could even add new .nar files to the deployment with new processors to enable the usage of other systems, even proprietary ones. A non-exhaustive list of examples of these extensibility mechanisms provided by DatAna are the following:

- **Getting data from multiple data sources**: Apache NiFi provides dozens of processors to get incoming data from multiple data sources. Examples are:
  - From data brokers: In the case of MARVEL, the usage of specific processors for MQTT (ConsumeMQTT) or Kafka (ConsumeKafka) have been used. Other processors available enable communication with other systems such as RabbitMQ, JMS, Amazon Kinesis, Web Sokets, IMAP or EWS email servers.
  - From storage systems: Specific NiFi processors are available for SQL databases, such as issuing SQL statements, list database tables, converting JSON to SQL, executing SQL, etc. Specific connections to other big data storage systems such as Hive. CouchBase, DynamoDB or Kudu are also available.
  - From files or web servers; Several processors are available to get data from files, such as accessible files (GetFile) and FTP files (GetFTP), among others. HTTP and HTTPs calls are also enabled to get or put data of Representational State Transfer (REST) services (e.g., InvokeHTTP, ListenHTTP, etc.).
  - Other processors: Other processors are available to interface with external systems, such as the ones to connect with the Amazon Web Service (AWS) ecosystem (GetSQS, FetchS3Object, etc.).

[36] https://nifi.apache.org/docs/nifi-docs/

- **Processing the data**: NiFi provide a set of processors that enable data processing capabilities without programming. These capabilities are also complemented by a simple scripting language that enables easy data transformation and management capabilities.
    - Attribute extraction processors: These processors support the process of inspecting and changing the attributes (metadata) in a dataflow. Some of the processors that belong to this category are UpdateAttribute, EvaluateJSONPath, ExtractText, AttributesToJSON, etc.
    - Routing and mediation processors to enable control of the data flows, such as routing on values of attributes, on content, based on rate, based on text found, etc.
    - Aggregation and data splitting processors: Processors that enable the splitting and aggregation of content in the data flow, such as SplitJson, SplitContent, MergeContent, etc.
    - System interaction processors: These processors enable running processes, commands or scripts in a variety of operating systems. These examples are ExecuteScript, ExecuteProcess, ExecuteStreamCommand, etc.
    - Data Transformation processors: These processors are able to modify the content of the data in a flowfile, and are one of the key aspects provided by NiFi. This functionality is used in MARVEL to enable the transformation of the inference results from their raw format to the desired data model used in the project. Examples of these processors are ReplaceText or JoltTransformJSON.
- **Communication with external systems**: These are processors that typically are located at the edge of a data flow, providing connection with external systems that store or receive the data in a destination server. In the same way, NiFi provides processors for getting data, it also boosts a similar list of processors that enable communication with external systems, such as data brokers, storage systems, files, or other systems.

Extensibility to add data from other sensors

It is worth noticing that the usage of MQTT messages as an entry point for new data coming from external systems or sensors could be easily implemented in NiFi by creating new data flows similar to the ones developed within MARVEL. Other types of data ingestion may be implemented as well using NiFi-specific data processors to get data from external data sources, as explained before.

As the data coming from other sensors could be very different in nature and data model to the one coming from MARVEL inference results, the new NiFi dataflows managing this process should be carefully designed to interface with the right storage systems and/or with the DFB, if this is deemed necessary.

3.9 Data aggregation, storage, and distribution

The Data Fusion Bus (DFB) has a central role in the data management of the ‘AI Inference Pipeline’. It is responsible for receiving and aggregating the inference results from all sources via DatAna, permanently storing them and re-distributing them both in real-time and asynchronously as historical data.

In addition, the DFB can also serve as the central hub for interfacing with external components that are not part of the ‘AI Inference Pipeline’, thus having a **significant contribution to the extensibility aspects** of the system. As suggested in Section 2.2.3, there might be several cases, where the MARVEL framework might need to interface with external, third-party components.

Such components can be distinguished into external data sources and external data sinks. Due to the widespread use of the DFB underlying technologies in the industry (Apache Kafka and Elasticsearch), the DFB can expose corresponding interfaces to external components.

In the case of external data sources, the DFB can provide access to the Apache Kafka interface and the data sources can act as producer clients. Dedicated Kafka topics can easily be configured depending on the use case. In cases where it is needed to connect with external data sinks, both main DFB interfaces can be exposed, i.e., the Apache Kafka and Elasticsearch APIs (via the ES-Proxy REST API) so that external components can access real-time and historical inference results respectively. Such a connection has already been established with the Data Corpus, which implements a Kafka consumer client for receiving inference results by subscribing to Apache Kafka topics.

Examples of external **data sources** may include the following:

- **External databases**. Organisations that adopt MARVEL may have proprietary data repositories, whose data may need to be combined with other data collected by MARVEL and analysed jointly. These databases or data stores may contain static data or be updated dynamically with new data. In such cases, a data extractor or a new entry watcher can be implemented to obtain the necessary data, which can then be transferred to the DFB by implementing a Kafka producer client.
- **Streams of non-AV data** (e.g., weather or other environmental data from sensors) that are not integrated in the MARVEL 'AI Inference Pipeline' via DatAna. Smart City use cases may require the analysis of data that originate from non-AV types of sensors such as temperature, humidity, air quality, wind speed, motion, flow, pressure, proximity. Third-party Internet of Things (IoT) platforms that collect and aggregate such data typically rely on messaging system implementations (e.g., MQTT, DDS, Azure Event Hubs, Amazon MQ, Google Cloud pub/sub, Apache Kafka). Data from such systems can be integrated to MARVEL by establishing an interface between the external messaging system and the DFB Kafka cluster. In the particular, cases where the external system implements Kafka technology, a Kafka mirroring configuration can also be established [37].
- **Streams of inference or other analysis results from third-party components** that are not integrated in the MARVEL 'AI Inference Pipeline' via DatAna. Third-party data analytics and AI components that operate outside of the MARVEL framework may produce analysis/inference results that may be required by potential future use cases to be addressed by MARVEL. In such cases, these third-party components hosted on external infrastructure can act as Kafka producer clients and publish their results directly to dedicated topics of the DFB Kafka cluster.

Examples of external **data sinks** may include:

- **Front-end UI and data visualisation components**. In cases where there are demands for presenting and visualising data managed by MARVEL in external applications, the DFB may allow access to the data via its two main interfaces, i.e., Apache Kafka for real-time data and Elasticsearch (via the ES-Proxy REST API) for historical data.
- **Components for post-analytics that process first-order inference results**. Future use cases may require third-party data analytics components hosted on external

[37] https://cwiki.apache.org/confluence/pages/viewpage.action?pageId=27846330

infrastructure to access the inference results produced by the MARVEL 'AI Inference Pipeline' so that they can be post-processed (e.g., for long-term analysis or integration with external data). In such cases, the DFB may allow access to the inference results via its two main interfaces, i.e., Apache Kafka for real-time data and Elasticsearch (via the ES-Proxy REST API) for historical data.

- **Control mechanisms in external systems**. Future use cases may require the reception of the MARVEL inference results in real-time by the control mechanism external systems that may use them as input for controlling procedures. For example, the detection of multiple traffic bottlenecks or accidents in the road network by MARVEL may be fed to an external system controlling traffic lights so that it can perform necessary adjustments in real-time. In such cases, the external systems can subscribe to the required DFB Kafka topics by implementing Kafka consumer clients.

Figure 25 illustrates the pivotal role of the DFB in the 'AI Inference Pipeline' to operate as a hub for connecting external data sources and data sinks.

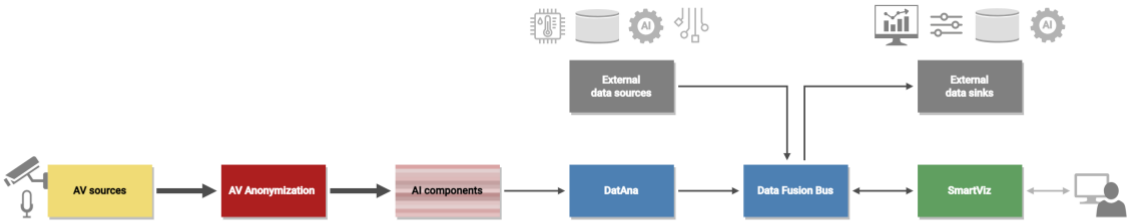

Figure 25: Use of the DFB as an interface for external data sources and data sinks

The integration of external data sources and data sinks via the DFB Kafka interface can be further facilitated by the availability of multiple ready Kafka connectors[38] in the Apache Kafka ecosystem that are based on the Apache Kafka Connect technology[39]. Furthermore, the REST API that has been developed for exposing selected query types in the Elasticsearch data repository (ES-Proxy) can further simplify the access to historical data (inference results). Besides, both the Kafka and Elasticsearch technologies are both widely adopted by the industry and as such they provide interfaces that are familiar to most external technical organisations that may need to integrate their solution with the MARVEL framework. Furthermore, the fact that they are open source or free and open (in the case of Elasticsearch) technologies maximises the potential for usage in extensibility scenarios, as they are more accessible to SMEs in terms of cost and they also allow a much higher customisation potential when compared to commercial closed source solutions (e.g., MS Azure, Amazon MQ, Google Cloud pub/sub).

Due to the central role of the DFB in data management both in relation to the internal 'AI Inference Pipeline' and to interfacing with external systems and components, as described above, it is greatly affected by changes of scale. For example, an addition of sensor nodes (AV sources) and/or AI component types to a deployment of the MARVEL framework is bound to generate proportionally more inference results that will need to be handled by the DFB. Similarly, the addition of SmartViz instances that may be needed to serve an increased number of operators will increase the demand for serving data. In the context of extensibility, the load

[38] https://docs.confluent.io/platform/current/connect/kafka_connectors.html

[39] https://kafka.apache.org/documentation/#connect

incurred by the DFB can also be greatly increased by the demands of requests from external data sources and data sinks that can potentially be significant when considering real-world scenarios of adoption of the MARVEL framework at a city scale.

The DFB is best suited to deal with scenarios of scaling up and is able to serve requests of extreme scale, as it is built upon technologies that are inherently scalable by definition. Both the Apache Kafka and the Elasticsearch technologies have been developed as fully scalable solutions by design and are meant to operate in cluster mode on cloud-based infrastructure.

Elasticsearch is a very well-established solution for data storage and fast data retrieval that has a proven track record in multiple industry domains in real-world applications. Through careful planning of an Elasticsearch cluster composed of nodes and resource allocation, significant gains can be had in terms of resilience and performance[40]. In its 2017 release, Elasticsearch was able to provide 1 million writes per second in a cluster set-up[41], while it remains ahead of competitive solutions (Redisearch, PostgreSQL, TypeSense, MeiliSearch) in a comparison benchmark held in 2021[42] and ahead of OpenSearch according to tests held in 2023[43]. Furthermore, Elasticsearch continuously publishes performance evaluation benchmark data of its current release for a diverse range of scenarios[44], while there is ongoing active work working towards improving its scalability[45] and performance[46].

The DFB ES-Connector is the component that connects Kafka to Elasticsearch and has therefore a significant role in the performance of the DFB. It is also scalable by design and built to operate with multiple replica instances of the service in parallel. A noteworthy observation reported in D5.5 (Section 3.3.1 and 8.3.1 / Annex 2) was that the replication of the ES-Connector service led to significant gains in terms of latency during the framework benchmarking activities.

On a server level, Kafka can be run as a cluster of one or more servers that can span multiple datacenters or cloud regions. To implement mission-critical use cases, a Kafka cluster is highly scalable and fault-tolerant: if any of its servers fails, the other servers will take over their work to ensure continuous operations without any data loss[47]. Kafka can consistently deliver very high performance when deployed in cluster mode in suitable infrastructure[48] and provides significant gains compared to competitive solutions such as Pulsar and RabbitMQ[49]. The high performance along with the high fault tolerance and reliability features of Kafka with a proven track record in multiple industry domains where it has been widely adopted for mission-critical real-world scenarios, render it a very suitable and efficient solution in a scaled-up and real-world implementation of the MARVEL framework.

[40] https://www.elastic.co/blog/maximizing-elasticsearch-performance-when-adding-nodes-to-a-cluster

[41] https://medium.appbase.io/benchmarking-elasticsearch-1-million-writes-per-sec-bf37e7ca8a4c

[42] https://medium.com/gigasearch/benchmarking-performance-elasticsearch-vs-competitors-d4778ef75639

[43] https://www.elastic.co/blog/elasticsearch-opensearch-performance-gap

[44] https://elasticsearch-benchmarks.elastic.co/

[45] https://www.elastic.co/blog/benchmark-driven-optimizations-scalability-elasticsearch-8

[46] https://www.elastic.co/blog/whats-new-elasticsearch-platform-8-9-0

[47] https://kafka.apache.org/documentation/

[48] https://engineering.linkedin.com/kafka/benchmarking-apache-kafka-2-million-writes-second-three-cheap-machines

[49] https://www.confluent.io/blog/kafka-fastest-messaging-system/

The results presented in D5.5 (Section 3.3.1) suggest that the DFB is able to cope with the load generated by the R2 deployment. However, in the case of R2, a small Kafka cluster composed of 3 brokers was implemented. In the case of large-scale deployments of the MARVEL framework, clusters of larger sizes will be demanded. Considering that the Kafka interface of the DFB may be the one to possibly sustain the most critical loads while serving the real-time needs of the internal 'AI Inference Pipeline' and of external data sources and sinks, additional steps have been taken to optimise the sizing of the DFB Kafka cluster through the implementation of the HDD component. The relevant activities are documented in Section 3.10.

3.10 Scalable data distribution with Apache Kafka

In this section, it will be described how volumes of big data can be managed and aggregated at a large scale. In particular, the horizontal scaling features of the DFB will be reported along with the optimisations brought forward by HDD. The addressed use cases will include high volumes of incoming data from DatAna as well as from external sources (e.g., external databases, external AI components that do not follow the MARVEL inference pipeline). The connection to multiple data sinks will also be addressed (e.g., multiple SmartViz instances, external data post-processing and visualisation tools). In the case of data originating from DatAna and external data sources, we can consider an increase in Kafka producer clients. Conversely, in the case of data consumed by SmartViz instances and by external data sinks, we can consider an increase in Kafka consumer clients.

The current version of HDD expands the one described in D2.4 in several key ways related to addressing increased scalability and showcasing its potential in this respect. Firstly, we have expanded the methodology for the automated performance evaluation of Apache Kafka clusters by including also production, consumption and end-to-end experimental approaches. This methodology allowed us to run additional, scalability-focused experiments in the automatised prototype to compare BroMax and BroMin not only with different replication factors, and numbers of consumers but also with different numbers of producers. Secondly, we have included the results of these experiments, which have highlighted some insights with respect to the scalability support obtained by HDD. We have made our evaluation framework available to the community so that others can use it for their own performance evaluations. Overall, these additions provide a more complete and robust analysis of Apache Kafka topic partitioning and demonstrate the scalability potential of the methodology.

The experiments have been carried out in controlled and repeatable conditions with the reusable framework that we made publicly available to the research community on GitHub[50]: the repository includes the detailed steps to prepare the prototype, the full set of well-documented scripts to execute the experiments, and the dataset of results obtained on our servers. Our framework enables three types of experiments, each stimulating different aspects of the system performance, which can be more relevant for a specific target production scenario:

1. Consumption experiment, which focuses on the consumption throughput T_c that can be obtained by a set of clients from an Apache Kafka cluster for messages of a given size M, where the data injection is not a choke point. For consumption, the rebalance time, i.e., the time needed for the assignment of the partitions to the consumers to converge to its final configuration, can be also relevant and, thus, it is produced as output by this type of experiment. Consumption experiments have been extensively presented in D2.4, so we omit their presentation in the current deliverable.

[50] https://github.com/ccicconetti/kafka-hdd

2. Production experiment, which is complementary to the previous one and focuses on the production throughput T_p that can be obtained, together with the production latency l_p, i.e., the time between when a given message is generated by a client and when it is committed by the cluster
3. End-to-end experiment, where there are both producers and consumers concurrently generating/reading data in a real-time streaming fashion, with the end-to-end latency l_{e2e} being the main performance metric.

Our prototype consists of two high-end servers: one hosting the client-side scripts and tools and another handling the cluster-side, i.e., the Apache Kafka brokers and a Zookeeper instance for leader election among them. The software on the cluster-side server runs within Docker containers configured with Docker Compose, which is a tool to start/stop/manage applications consisting of multiple containers defined in a single YAML file. This approach is suitable for running the entire cluster within a single physical server, but the scripts we developed can be adapted to match the specific characteristics of the target deployment under test, e.g., a distributed environment where Apache Kafka is run within a Kubernetes (K8s) cluster. The methodology would remain the same and it is illustrated by means of the sequence diagram in Figure 26, which is entirely managed through the execution of a single Bash script on the client-side server as detailed in the following. Note that the sequence is a significantly extended and improved version of the one presented in D2.4. At the beginning of the experiment, there is no Apache Kafka cluster running. In principle, there are situations where we could reuse a running cluster from the previous experiment, i.e., when the cluster parameters *P, b* remain the same but we decided to start with clean conditions to ensure independence and repeatability. The input of the experiment is: the Apache Kafka cluster provisioning algorithm *A*, the replication factor *r*, the number of consumers *c*, and the size of the Apache Kafka messages exchanged *M*, in bytes On the client side, the algorithm *A*, i.e., BroMin or BroMax (described in D2.2), is run to determine the number of topic partitions *P* and brokers *b*. Then, the client issues the commands required to remotely start the Apache Kafka cluster of *b* brokers via Docker Compose, creates the topic that will be used in the current experiment, and configures it with the given replication factor *r* and the number of partitions *P*.

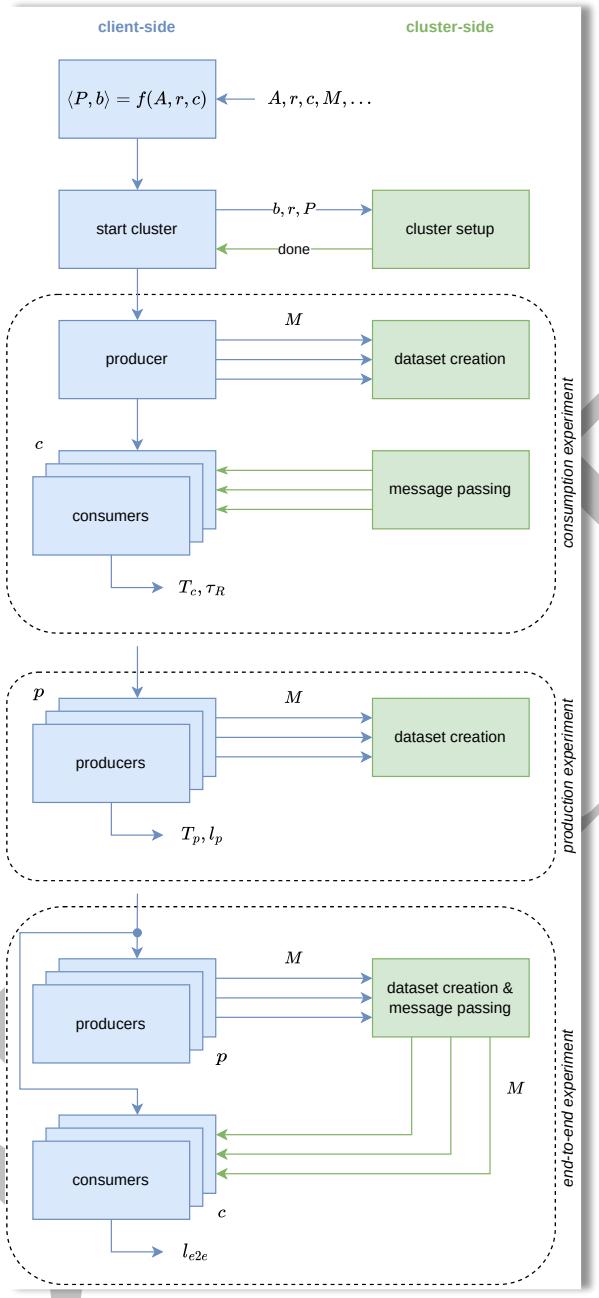

Figure 26: Methodology adopted for the execution of automated experiments

In the experiments illustrated hereafter, we have set the maximum number of brokers available to 16, based on the hardware characteristics of the cluster-side server and we varied the number of consumers c in the range [25,125]. For each combination of factors, we have run multiple iterations: in the plots, we show only the average value across them for better readability; the interested reader can find the plots with confidence intervals in the GitHub repository.

We first present a batch of production experiments, obtained by increasing the message size from 1kB to 100kB to highlight the scalability of this aspect. In this type of experiment, the client-side server uses all possible resources to produce data as fast as possible in an attempt to estimate the maximum production throughput of the cluster, therefore the number of producers is not relevant. By looking at the two plots in Figure 27, which show the production throughput T_p measured in MB/s vs. record rate, we can see that the qualitative trends are opposite: this is

because larger messages consume more resources than small ones to be transferred/serialised/replicated, thus the record rate decreases as *M* increases, but the throughput in MB/s is more than compensated by decreasing per-message overhead and it increases significantly overall.

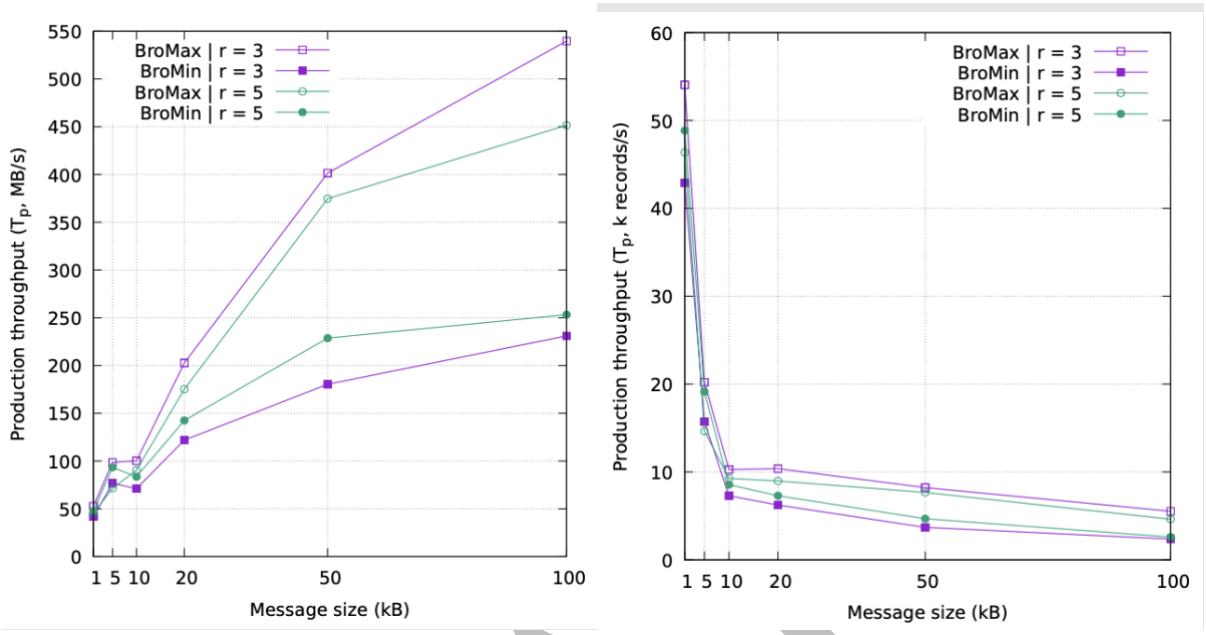

Figure 27: Production scalability experiments: Production throughput in MB/s and record rate

To better understand the results for what concerns the other parameters, i.e., the replication factor and the resource allocation policy, we explain the results of Figure 27; the production throughput with BroMax is much higher than that with BroMin for all the values of *M*: this is because the former allocates way more resources than the latter, in terms of brokers, i.e., 16 instead of 3 or 5. As a result, the Apache Kafka cluster can increase the level of parallelism when ingesting data, which leads to a higher rate of produced messages. T_p is slightly smaller with a higher replication factor because the amount of resources remains the same but each message must be replicated 5 times instead of only 3 in the cluster. On the other hand, with BroMin the opposite happens: the throughput with $r = 5$ is higher than that with $r = 3$ because, while BroMin is conservative in the use of resources, it cannot allocate a number of brokers that is less than the replication factor, which leads to improved performance thanks to the extra resources available with $r = 5$.

We also investigate the production latency, whose average value is shown in Figure 28. Counter-intuitively, the average latency decreases as the message size increases, even if the mere transmission time from client to server obviously increases with *M*. This is because the production latency takes into account also the time for the cluster to replicate the data and such an operation is done in batches: with larger messages, the brokers need to pack together a smaller amount of messages to perform some of their housekeeping operations, which eventually leads to lower latency. On the other hand, the BroMax curves are lower than the BroMin ones, since the latency is inversely proportional to the throughput. Finally, with BroMax, the difference in the average latency for the two replication factors is negligible, while it is noticeable with BroMin, for which $r = 5$ enjoys a relatively smaller average latency, because of the availability of more brokers, as discussed above.

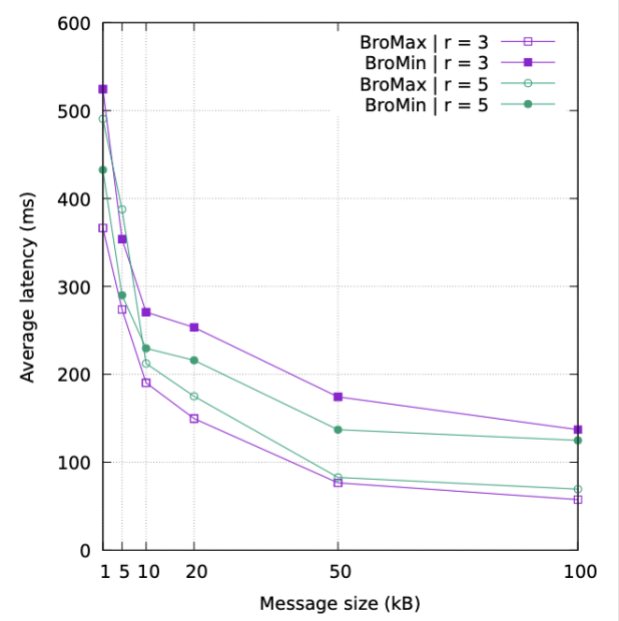

Figure 28: Production scalability experiments: Average production latency

We conclude the prototype performance evaluation with an end-to-end experiment carried out with *p* producers generating 30 messages/s with variable message size in *U*[1kB, 100kB]. Initially, we keep the number of consumers equal to 5 to best emulate a real production scenario where there is a fixed number of services that consume data, e.g., for analytics or monitoring purposes, while the number of producers may change over time due to, e.g., organic growth of the infrastructure. We consider only a replication factor of 3, which is adequate for most production systems deployed with nodes having industry-standard mean time before failure. We report the Cumulative Distribution Function (CDF) of the latency, with 10, 15, and 20 producers, in Figure 29. First, we observe that the latency increases significantly with the number of producers: this is expected because each producer generates data independently from the others, so increasing *p* corresponds to increasing the overall cluster load proportionally, which yields higher latencies. This suggests that resource allocation should also take the number of producers as an input variable of the problem; when a value for this parameter cannot be estimated, the latter can be monitored in the field and used to periodically re-run an appropriate resource allocation algorithm; such a study is complementary to this work and will be considered as part of our future research activities. Second, BroMax exhibits a higher latency than BroMin; in other words, unlike in the production experiment above, allocating more resources to the cluster, that is brokers and partitions, leads to worse performance, especially in terms of the high tail latency. This effect is due to the increased overhead for cluster management, and it implies that, contrary to general intuition, overprovisioning does not necessarily produce optimal results, when the key performance is maintaining a tight pipeline from producers to consumers.

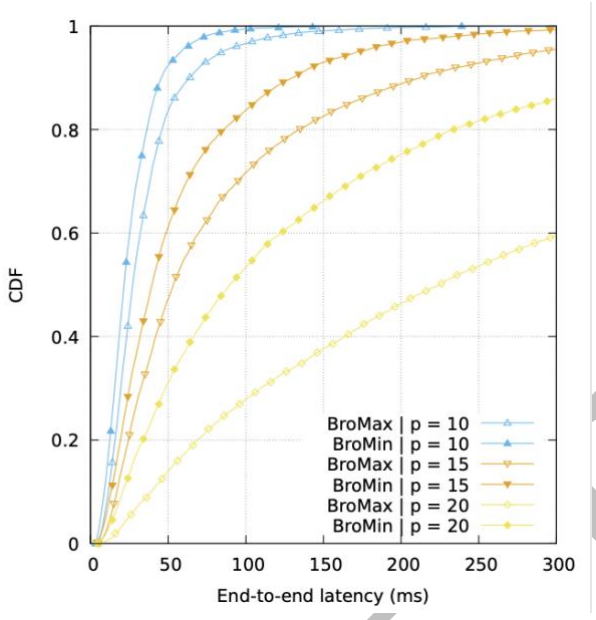

Figure 29: End-to-end scalability experiments: Cumulative distribution of the end-to-end latency with different resource allocation policies (BroMax vs. BroMin) and number of producers (p=10, 15, 20), with 5 consumers (c=5).

We conclude with an extreme scenario, where we increase further the number of producers, to 25 and 50, and the number of consumers, to 10 and 15. As shown in Figure 30, the latency increases significantly, due to the high total throughput (310Mb/s for *p*=25 and 620Mb/s for *p*=50) compared to the resources available in the cluster. Unlike in previous consumer-/producer-only experiments, here the number of consumers affects significantly the latency, which increases steeply from *c*=10 to *c*=15. Furthermore, we can see that BroMax generally exhibits better performance than BroMin, especially in terms of the high quantiles of latency. This is because the performance bottleneck in this scenario is created by how fast the messages can be dispatched by the brokers, hence having more brokers helps in reducing the congestion.

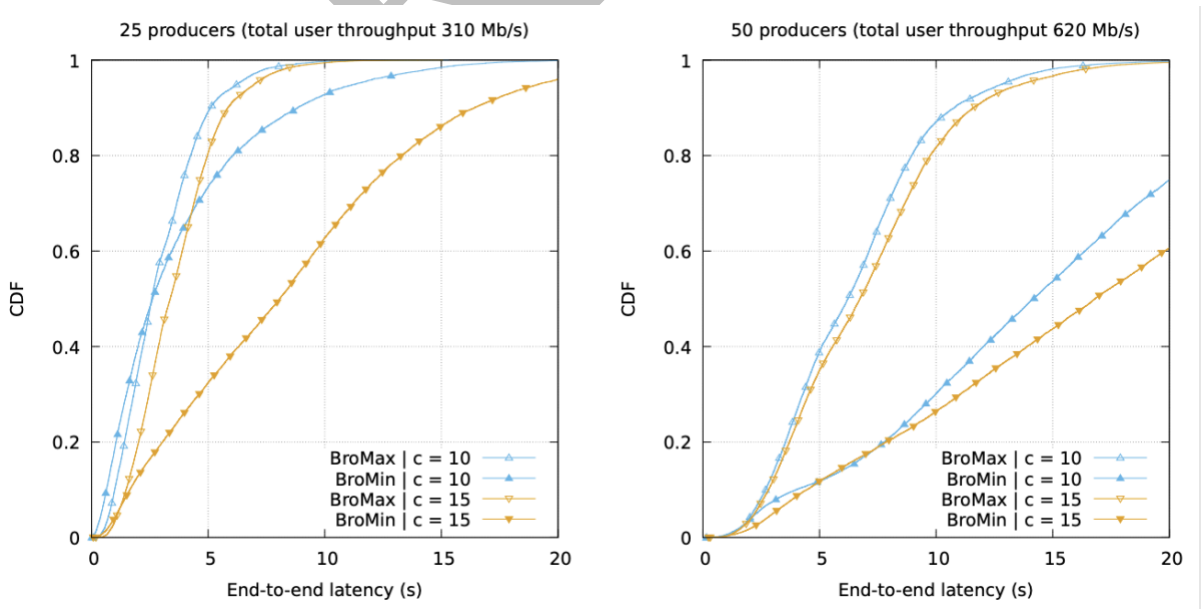

Figure 30: End-to-end scalability experiments: Cumulative distribution of the end-to-end latency with different resource allocation policies (BroMax vs. BroMin), number of consumers (*c*=10,15)}, and number of producers (*p*=25,50)}.

The main outcomes of scalability of the analysis of prototype experiments are as follows:

- The scalability performance in the cluster depends significantly on the size of the messages exchanged, as well as on the replication factor, which can significantly degrade the consumption throughput with large messages.
- With small messages, the advantages of parallelising data ingestion with multiple consumers can be offset by the increased communication overhead, thus leading to a reduced cluster throughput; on the other hand, larger size messages incur smaller cluster management overhead and more efficient batching, which lead to improved production throughput and latency.
- A more aggressive allocation of resources (BroMax), in terms of the number of brokers and topic partitions, does not always lead to improved performance: it does so for production throughput but, on the contrary, the consumption throughput can be lower, and it can take longer for the rebalancing procedure to converge compared a more conservative use of resources (BroMin).
- In a real-time streaming scenario, BroMin achieves significantly lower end-to-end latency due to the reduced complexity of the cluster management procedures, but BroMax reduces the congestion in extremely loaded conditions.

In essence, the scalability of data distribution in prototype experiments hinges on a delicate balance between message size, replication factors, and resource allocation within a cluster. When dealing with small messages, the benefits of parallelising data ingestion can be overshadowed by increased communication overhead, resulting in a diminished cluster throughput. On the flip side, larger messages, while incurring smaller management overhead, facilitate more efficient batching and ultimately lead to improved production throughput and lower latency. Interestingly, a more aggressive allocation of resources, as exemplified by BroMax, doesn't universally translate to enhanced performance. While it boosts production throughput, consumption throughput may suffer, and the rebalancing procedure may take longer compared to a more conservative approach like BroMin. In real-time streaming scenarios, BroMin emerges as the latency champion, owing to simplified cluster management procedures. However, BroMax shines in extremely loaded conditions by mitigating congestion. The key takeaway is that achieving scalable data distribution involves a nuanced consideration of message size, replication factors, and resource allocation, with no one-size-fits-all solution but rather a careful calibration tailored to specific operational conditions.

4 Considerations and Challenges during the implementation of the MARVEL Solution

During the implementation of the MARVEL solution several challenges arose, and they were tackled. In Section 3, the scalability and extensibility aspects from a technical point of view were presented. As described in Section 2, MARVEL handles AV data from public places, for this reason, a pivotal aspect of MARVEL is data protection. More specifically, as the volume of AV data is expected to increase through extensibility, compliance with the latest data protection regulations is of paramount importance. In this section, the aforementioned aspect is presented alongside the actions taken within the MARVEL framework to tackle it. Moreover, Section 4 provides an overview of the operational challenges identified during MARVEL's lifespan and how they can affect the scalability, extensibility, and maintenance of the framework. Finally, the main aspects and patterns used for the realisation of MARVEL's User Interface are presented.

4.1 Legal, ethical, and privacy concerns

In this subsection, the main concerns related to the utilisation of the MARVEL solution in real-world scenarios and on real datasets from the legal, ethical, and privacy perspectives will be described.

Data Privacy and Compliance

Strong privacy protection is established through effective privacy governance, consisting of practices that ensure compliance with privacy laws and regulations. This involves setting a vision for the privacy strategy, sharing responsibilities across project partners, and identifying the scope of collected personal information. Privacy governance includes rationalizing legal requirements for specific contexts, understanding the personal data lifecycle, and addressing cultural and personal expectations and challenges. Identifying and mitigating privacy risks, implementing controlling and monitoring mechanisms, following procedures, and using privacy software tools are crucial components of proper privacy governance. Success in privacy governance is measured by the use of appropriate metrics.

Ensuring compliance with data protection laws when handling sensitive user data within the MARVEL solution is a concern that the MARVEL project had to handle. Compliance monitoring in the MARVEL project focused on identifying regulatory requirements and technological developments. This monitoring serves as a vital factor in ensuring legal and ethical compliance throughout the project life cycle, acting as a unifying element for security, privacy, and data protection requirements, and supporting the project's risk management through a coherent approach to legal and ethical considerations. The most important compliance requirements include the following:

- Lawful processing of personal data: The project consortium prioritises the lawful processing of personal data in MARVEL, particularly within project pilots, where consent from data subjects serves as the primary legal basis.
- Transparency: Compliance with transparency requirements is ensured by making information and communication regarding the processing of personal data easily accessible and understandable.
- Trustworthy AI: Prioritising principles of diversity, non-discrimination, and fairness, the project addresses the challenge of building trustworthy AI.

- Data security: The greatest efforts towards data security were on the development of anonymisation/pseudonymisation strategies, tactics and techniques, Edge-to-Fog-to-Cloud security as well as other safeguards to protect data.
- Data protection impact assessment: The Project Coordinator (PC) and project partners collaborated to provide opinions on the necessity of Data Protection Impact Assessments for data processing in project pilots.

Moreover, an additional strategy regarding compliance was the creation of the Ethics Advisory Board (EAB). It comprises three external members with expertise in ethics and the EU Ethics Appraisal Process, alongside three internal members representing legal, ethical, and technical aspects, including the Project Coordinator. The EAB overseeded adherence to privacy laws, anonymisation processes, informed consent, data security, and other ethical considerations throughout the project's lifetime.

Participation of humans

The participation of humans in the project raises various considerations and potential challenges and necessitates careful consideration of ethical, legal, and procedural aspects, including privacy, consent, and ethical treatment. Below, we summarise the strategies and initiatives undertaken to meet ethical and legal obligations concerning human involvement in the project, providing a comprehensive overview of measures taken in each of the three project pilots (GNR in Malta, MT in Trento, and UNS in Novi Sad), including use case details, participant identification, recruitment, and the informed consent process. More details about each of the pilots can be found in D6.1[51], D6.2[52], and D6.3[53].

- In the GRN pilot, which focused on observing road use through audio and video recordings, participants were not directly selected, but the chosen areas, intersections in residential and industrial zones, influenced the decision on participants appearing in the recordings. The recordings included drivers of private vehicles, motorcycles, and cyclists, reflecting the diversity of road users. Traffic experts played a crucial role in testing and validating use cases, representing decision-makers in smart road system procurement. Anonymisation measures were implemented to protect personal privacy of individuals appearing in recordings without their knowledge. While no informed consent was needed for such individuals, those participating in testing and evaluation activities, typically traffic experts or citizens assessing societal impact, provided consent with information sheets. Records of obtained consent were maintained by GRN.
- The MT use cases in MARVEL involved monitoring busy areas for selecting camera views, detecting criminal/anti-social behaviour, monitoring parking lots, and supporting administrative decision-making. Two types of AV recordings were conducted: real recordings, featuring people present at selected locations without specific criteria, and staged recordings, involving staff from MT and FBK organisations and students from the University of Trento. All participants volunteered, and recorded data, after anonymisation, was shared with MARVEL partners. The participant selection emphasised diversity in gender, ethnicity, age, and other physical characteristics, with experts and decision-makers actively involved in testing, evaluating, and validating the use cases. Recruitment involved direct email invitations, outlining project goals and recording details, and participation was based on personal ability and willingness. For

[51] "D6.1: Demonstrators execution - initial version," MARVEL Project, 2022. https://doi.org/10.5281/zenodo.6862995

[52] "D6.2: Evaluation Report," MARVEL Project, 2022. https://doi.org/10.5281/zenodo.7296312

[53] "D6.3: Demonstrator execution - final version," MARVEL Project, 2023. https://doi.org/10.5281/zenodo.8315360

non-staged recordings in MT, no individual consent was obtained, and the public was informed through privacy policies, signs, and additional information on the Municipality of Trento website. Participants in staged recordings signed a consent and privacy declaration form prepared by MT's Data Protection Officer, with the documents available in Italian and English.

- The UNS use cases focused on monitoring public events and localising audio events within crowds, utilising staged recordings with participants voluntarily selected from UNS employees and students. Due to the specific nature of the use case, there was no formal recruitment procedure, and participants volunteered for data collection, testing, and evaluation. Informed consent for the UNS pilot, involving staged recordings, included briefings on the project details, provision of consent forms and information sheets in Serbian and English, ensuring participants were adequately informed.

Data processing

Researchers in MARVEL took into consideration the preservation of the privacy of research subjects and the protection of their personal data. This concern becomes particularly evident when dealing with special categories of data, such as those revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, trade union membership, genetic data to uniquely identify a natural person, biometric data, and data concerning health or a natural person's sex life or sexual orientation. Within the course of the project, several categories of personal data were processed.

- Audio and video data. The collection of data from IoT devices within MARVEL involved the recording of individuals in public spaces, capturing their activities, interactions, and presence. While this data could provide valuable insights for project analysis and documentation, its handling requires careful consideration to ensure compliance with privacy and ethical standards. Additionally, the incorporation of audio data from individuals in public areas introduces an additional layer of information that contributes auditory context to the project. These types of data raise privacy concerns.
- Data of experts and stakeholders. Within MARVEL, the engagement of experts and stakeholders spanned across various stages of the project. Their involvement resulted in the collection of details such as names, professional affiliations, roles, and contact information. Ensuring the confidentiality and security of this information is paramount to protect the personal and professional identities of these individuals. Furthermore, the collection of email addresses from individuals expressing interest in newsletters signifies a direct interaction with the public. While email addresses may be considered less sensitive than other personal data, emphasising consent and adhering to communication preferences are indispensable components of proficient data management and compliance with privacy regulations.
- Data collection during the events. Personal information for event registration, including names and contact details, was processed, necessitating secure management for participant privacy and smooth event organisation. Additionally, communication data of individuals and project partners, encompassing emails and messages, required protection to maintain the integrity and confidentiality of project-related discussions and collaborations.

Concerns associated with the collection and transfer of data outside the EU present a complex landscape, necessitating adherence to the principles outlined in the GDPR. When transferring data, researchers must address ethics risks by securing informed consent, ensuring an equivalent data protection framework, or confirming the destination country's compliance with GDPR

standards. Similar ethical considerations arise when collecting data from non-EU countries. It is the researchers' responsibility to navigate through these complexities.

Ensuring lawful research ethics depends on obtaining informed consent, a critical aspect defined in Article 4 of the GDPR. Emphasising real choice and control, freely given consent necessitates providing detailed information to the data subject. This includes the identity of the data controller and the DPO, the specific purposes of data processing, the subject's rights, information on data sharing or transfers to third parties, and the duration of data retention.

Finally, to safeguard the rights of data subjects and mitigate ethical risks associated with data processing, measures must be implemented to protect participants' information, as mandated by the GDPR. These measures encompass pseudonymisation, encryption, system integrity, availability, resilience, timely data restoration, and ongoing evaluation of security effectiveness. Researchers and controllers of gathered data are advised to employ GDPR-compliant tools for data collection, processing, and storage.

MARVEL acknowledges and addresses the ethical concerns regarding the processing of sensitive data, reflecting a commitment to standards of research ethics and data protection. MARVEL project aligns with GDPR guidelines on data transfer from outside the EU, places a strong emphasis on data privacy by anonymising data as close to the edge as possible, on data security by implementing a comprehensive set of technical and organisational measures, and makes use of informed consent to ensure the data subjects understanding.

Mass surveillance

Mass surveillance is yet another concern that the MARVEL consortium addressed. Mass surveillance was never the goal of the MARVEL project or the developed MARVEL framework. However, since it was identified as a potential risk, each component of the MARVEL framework underwent evaluation concerning mass surveillance. Given its modular structure with numerous subsystems and tools, we assessed the potential risk associated with each element contributing to mass surveillance within the MARVEL project. The outcome of the evaluation is presented below grouped in the MARVEL subsystems.

The Sensing and Perception Subsystem, comprised of six distinct components, has been evaluated, with each component contributing to an overall low risk of mass surveillance within the MARVEL project. The Advanced MEMS Microphones (IFAG) and the SensMiner (AUD) components pose no risk due to their lack of data storage or processing. Similarly, the Sound Event Detection at the Edge - SED@Edge Kit (FBK) is considered to have a low-risk profile, sharing only non-sensitive events through the network. The Data Acquisition Framework - GRNEdge (GRN) introduces a low risk, taking advantage of limited local storage, physical security, and algorithm development for anonymisation and secure transmission. The Drone-based AV Data Collection - AVDrone (UNS) component presents a low risk, emphasising participant consent, high-level analysis of the collected data, and strict data storage security. Finally, the Data Acquisition Framework - CATFlow (GRN) carries a low risk, incorporating measures like pseudo-IDs and selective video data storage or anonymisation.

The Security, Privacy, and Data Protection Subsystem, comprising three key components, has undergone thorough evaluation, with each component contributing to an overall low risk of mass surveillance. The Security Services at the Edge - EdgeSec (FORTH) component mitigates potential risks by neither storing data nor engaging in processing, aligning with a secure and privacy-conscious design. The Video Anonymisation - VideoAnony (FBK) component introduces a low risk primarily associated with dataset annotations used in training. However, the absence of biometric or similar data usage contributes to a low-risk profile. Similarly, the

Audio Anonymisation - AudioAnony (FBK) component entails minimal risk, as signal processing solutions do not require training data, and potential imperfections during deployment are acknowledged.

The Data Management and Distribution Toolkit has also undergone comprehensive evaluation, with each component contributing to an overall low risk of mass surveillance. The Data Acquisition Framework - DatAna (ATOS) component, utilising NiFi for data processing, introduces minimal risk as NiFi itself does not add network risks, and security measures, including encryption and dedicated storage, are in place. The StreamHandler Platform (INTRA) primarily focused on data management and collection, incorporates robust security mechanisms such as encryption, authentication, and authorisation, further minimising the risk of mass surveillance. The Data Fusion Bus - DFB (ITML), designed with a core focus on security and trustworthiness, implements state-of-the-art solutions for secure data transfer and authorised access, reducing the likelihood of potential misuse. Integrated alerting mechanisms serve as additional safeguards. Lastly, the HDD (CNR) component carries a low risk as it processes data on the edge and fog without storing raw data or extracting individual information.

The Audio, Visual, and Multimodal AI Subsystem, collectively demonstrates a low risk of contributing to mass surveillance within the MARVEL project. The devAIce (AUD) component, designed for voice anonymisation, prioritises public privacy, with mitigation strategies emphasising minimal availability of essential components. Visual Anomaly Detection - VAD (AU), AV Anomaly Detection - AVAD (AU), Visual Crowd Counting - VCC (AU), and Audio-Visual Crowd Counting - AVCC (AU) components share a low-risk profile, processing raw data exclusively on the edge and fog without extracting individual information, thereby minimising potential mass surveillance concerns. Automated Audio Captioning - AAC (TAU) introduces a low risk mitigated by anonymised data usage and the absence of biometric or similar data. Acoustic Scene Classification Algorithms - ASC (TAU) and Sound Event Localisation and Detection Algorithms - SELD (TAU) present no risk of mass surveillance, emphasising secure functionality and privacy-conscious design.

The Optimised E2F2C Processing and Deployment Subsystem, featuring four distinct components, collectively showcases a low risk of contributing to mass surveillance within the MARVEL project. The GPU Pattern Matching Framework – GPURegex (FORTH) enables user-selectable signatures for comparison, with a low probability of mass surveillance, backed by the capacity to isolate and remove problematic signatures. Compressed models - DynHP (CNR) component emphasises secure neural network processing without data storage, ensuring a low-risk profile. Personalised Federated Learning - FedL (UNS) commits to processing anonymised features or raw data with quantifiable privacy guarantees, aligning with project objectives and contributing to a low risk of mass surveillance. The MARVdash Kubernetes dashboard (FORTH) introduces a low risk, with concerns primarily focused on potential password leaks, mitigated by enforced criteria and user isolation.

The System Outputs subsystem, comprising the Advanced Visualisation Toolkit - SmartViz (ZELUS) and MARVEL Data Corpus-as-a-Service (STS), collectively demonstrates a low risk of contributing to mass surveillance within the MARVEL project. SmartViz operates on anonymised data from previous analyses and components, ensuring that its advanced visualisations do not function as a surveillance platform capable of tracking or identifying individuals, thereby maintaining a low-risk profile. Similarly, the MARVEL Data Corpus-as-a-Service module introduces a low risk of mass surveillance by making datasets public and accessible to anyone interested, with rigorous anonymisation processes in place before integration.

In summary, the comprehensive evaluation of the MARVEL project components across its subsystems consistently reveals a low risk of contributing to mass surveillance. Each subsystem, including Sensing and Perception, Security, Privacy, and Data Protection, Data Management and Distribution, AV AI, Optimised E2F2C Processing and Deployment, and System Outputs, demonstrates a collective commitment to secure, privacy-conscious design and meticulous risk mitigation measures. The project's emphasis on anonymisation, secure data handling, and minimal data extraction at various stages ensures an overall low risk of mass surveillance throughout the MARVEL framework.

Data misuse

The MARVEL project identifies several potential risks associated with the data misuse concern, estimating a low likelihood for each. Firstly, there is a risk related to the tracking of pedestrians and vehicles using city-wide deployed cameras and microphones. However, the project clarifies that its focus is on detecting specific events in traffic or city areas, such as traffic anomalies and security threats, excluding methodologies like face or voice detection. The privacy-by-design approach is emphasised, ensuring the protection of generated/processed data.

Secondly, the use of drones in the MARVEL drone experiment for tracking pedestrians and vehicles is identified as a low-risk activity. This experiment is planned for a strictly controlled environment with participants' written consent, limited to small areas and a short duration.

The third risk involves aggregating and matching data from MARVEL devices and external sources, potentially leading to inference attacks on personal data. To mitigate this, the project assures the removal of personal data before any processing activity. Anonymisation is the first step of data processing, avoiding matching the MARVEL-generated data with data from external sources.

Another identified risk is associated with automated decision-making. Although decision-making is part of the functionality of the MARVEL framework, all components that have that role produce alerts that require human intervention rather than deterministic, legally binding results.

Data theft and potential misuse in a project pilot, including hijacking devices for data theft, is acknowledged as a low-risk concern. The project has emphasised on data security, incorporating secure data storage, transmission, and processing, taking advantage of technologies such as remote attestation and trusted execution environments.

The risk of interception of data in communication networks during a project pilot is also considered low. The project ensures security mechanisms at all layers, enforcing 100% encryption in the end-to-end data flow from edge devices to the cloud.

Lastly, the risk of personal data violation, specifically data theft from a data provider's or partner's storage, is identified with a low likelihood. The project addresses this by anonymising all data collected by MARVEL data providers and implementing strong technical and organisational measures, including secure servers, password protection, and minimal access rights, to protect the processed data within the project.

Data security

Data security is another legal, ethical and privacy concern. MARVEL project established security measures to protect personal data, providing two options for storing and accessing datasets—either hosting by the data provider or utilising a cloud under the data provider's direction. Data and Cloud providers detail specific technical and organisational measures implemented to ensure data security. Key security measures include regular backups of project-

related data, secure storage of devices and data with user-level access controls (e.g., encryption, password protection), and support for good security practices, including anti-malware software installation and updating. Each partner within MARVEL is committed to upholding these measures to safeguard the security of the collected personal data throughout the project realisation process.

Regarding **GRN** (Data Provider/Controller), the GRN staff strictly adheres to a comprehensive data privacy policy overseen by the GRN Data Protection Officer (DPO). This policy ensures the secure handling, anonymisation, and storage of MARVEL audio and video data. Personal data is encrypted using AES-256 algorithms when at rest. Dedicated hardware owned by GRN is employed, and internal servers, hosted in a compliant data centre with stringent physical access controls, store original datasets securely. Data is organised in batches, and when no longer needed, batches are permanently deleted. The transfer of data from source to GRN servers occurs via a secure VPN, and access to internal servers is restricted through VPN, Secure Sockets Shell (SSH), and passwords, demonstrating a robust commitment to data security throughout the entire data lifecycle.

MT (Data Provider/Controller) makes use of devices centrally managed, with IT staff overseeing software and antivirus updates. Strict access controls, password protection, and prohibition of software installations by users are enforced. MT utilises its Data Centre and a Disaster Recovery site with restricted access to hosting servers and networking devices. Breaches of personal data trigger immediate reporting to the DPO, activating the Data Breach Management Procedure. The DPO records such events in the Data Breach Register, monitoring their progression regardless of the decision to communicate the breach. MT's commitment to security practices and breach management exemplifies a proactive approach to data protection throughout the project.

FBK (Data Processor) processes data for MT's Data Controller using anonymisation techniques. Complying with GDPR, FBK's privacy policy guides behaviour for employees and collaborators involved in personal and business data processing. Strict device management includes centralised oversight, password protection, and access restrictions, with FBK's Data Centre and Disaster Recovery site ensuring data integrity. Cloud services from Google and Microsoft Azure are utilised. Personal data breaches trigger the Data Breach Management Procedure, overseen by MT's DPO, with events recorded in the Data Breach Register for ongoing monitoring.

UNS (Data Provider/Controller) handles MARVEL data, ensuring security by storing it on a protected server. Access is limited based on the minimisation principle, granting rights only to personnel essential for specific tasks. UNS's data server employs distributed network storage with RAID 1+1 protection for enhanced data security.

PSNC (Data Processor) conducts all data storage and processing within its controlled premises, ensuring strict access control in its Data Centres. Access to services requires identity confirmation through contracts, prohibiting anonymous usage. PSNC personnel are GDPR-aware, trained, and comply with legal obligations on data privacy. ISO27001 certification for data security covers both Cloud and HPC services, with procedures to maintain secure, patched, and up-to-date infrastructure. PSNC's implementation of Polish GDPR aligns with EU regulations. Any personal data breaches trigger the Data Breach Management Procedure, reported to the DPO. The DPO records events in the Data Breach Register, monitoring their evolution over time. On top of that, the HPC system is fortified with multiple layers of data privacy protection. Access is restricted to secure, encrypted protocols (SSH, SFTP, SCP), and data stored in both general purpose (/home filesystem) and fast computation cache (/scratch

filesystem using Lustre) is safeguarded with UNIX file access rights. Data owners retain control over access permissions. Users can only access servers where their processing occurs. Access to the system is granted through personal commercial contracts or scientific grants, both requiring the identification of authorised individuals. All user activity on the PSNC HPC system is logged and monitored for any illicit activities.

AUD (Data Processor) uses iHEARu-PLAY (Hantke, Eyben, Appel, & Schuller, 2015), a web-based crowdsourcing data annotation platform, on a dedicated server. To annotate data, it must be temporarily transferred to AUD, with ownership retained by data providers. AUD deletes all data after completion and returns annotated data to providers for consortium sharing. Security measures include authorised web access, global credentials for annotators, and user registration managed by AUD. Annotators, specified by data providers, receive authorised accounts with access limited to their provider's data. Data is promptly deleted post-annotation. While annotating data may compromise participant privacy, AUD does not ensure privacy; it assumes data is anonymised by providers before transfer. Annotators are expected to sign confidentiality agreements. Minimal AUD personnel access is granted, solely to obtain and make data accessible through iHEARu-PLAY. Access is restricted for all other personnel.

Moreover, MARVEL addresses different states of sensitive information named in motion, in use, and at rest. For data in motion, MARVEL emphasises the establishment of end-to-end secure communication channels. Encryption protocols like SSL and TLS are implemented to protect data during transmission, ensuring confidentiality, integrity, and availability. Data in use is safeguarded through the utilisation of trusted execution environments (TEEs), particularly Intel SGX. Leveraging SGX's instructions, MARVEL partitions sensitive information into enclaves within memory, providing enhanced protection against disclosure or modification. Regarding data at rest, which is often stored in the fog or cloud for batch processing, MARVEL introduces features for transparent file encryption. This ensures the integrity and confidentiality of files, allowing for different protection levels—integrity protection only or full encryption—based on specific security needs.

MARVEL project has implemented a robust set of security measures to ensure the confidentiality, integrity, and availability of personal data at different stages—motion, use, and at rest. These measures encompass secure data storage options, strict access controls, encryption protocols, and proactive breach management procedures. Each project partner, including GRN, MT, FBK, UNS, PSNC, and AUD, plays a crucial role in upholding these measures, demonstrating a collective commitment to safeguarding the privacy and security of the data throughout the entire project lifecycle. The use of encryption protocols, secure communication channels, and trusted execution environments reflects MARVEL's comprehensive approach to addressing various states of sensitive information.

Anonymisation and pseudoanonymisation

Anonymisation involves a data processing procedure that permanently eliminates personal identifiers, direct or indirect, that could otherwise identify an individual. MARVEL's approach to anonymisation proves valuable in the context of smart cities. The immense volume of data generated by interconnected IoT devices, coupled with access to high-performance computing resources, presents potential privacy concerns. Consequently, the ethics strategy of the MARVEL project prioritises the avoidance of directly collecting personal data, except when necessary for administrative purposes (such as contact details for dissemination). Anonymising audio and video content involves diverse technical considerations. In both scenarios (offline anonymisation and online anonymisation), the removal or processing of personal data (such as

facial and vocal markers used for identification) is necessary to render the recorded individuals unidentifiable.

In MARVEL, the voice anonymisation technique, inspired by (Liu, et al., 2021), employs a voice conversion model. It adapts the original method by integrating pre-trained speech and speaker representations (WavLM (Chen, et al., 2022), and TitaNet (Koluguri, Park, & Ginsburg, 2022)) in place of the Automatic Speech Recognition (ASR) bottleneck and x-vector speaker representation. This modification aims to enhance performance based on recent studies exploring semi-supervised learning models (Miao, Geng, & Jiang, 2022) for anonymisation tasks. The approach centres on the Phonetic PosteriorGram-Based Voice Conversion (PPG-VC) model, using pre-trained representations in an encoder-decoder module. The encoder integrates content, pitch, and voice details with speaker characteristics from TitaNet. A synthesis model, incorporating multi-speaker attention and logistics attention mechanisms, ensures controlled identity preservation during speech generation.

Moreover, the video anonymisation technique, in response to the need for maintaining various facial attributes while avoiding degradation in video-related tasks, a lighter version of a state-of-the-art GAN-based face-swapping model (Chen, et al., 2020) is being developed on VideoAnony. To address the computational intensity of GAN-based solutions, the focus is on reducing computational complexity. Efforts centre on replacing convolutional blocks with more efficient depth-wise separable convolutions (Howard, et al., 2017) and quantising the trained model from 32-bit to 8-bit integers. Initial outcomes exhibit a significant reduction in model parameters and operations, with a 73% decrease in model size through quantisation. The approach leverages PhiNets (Paissan, Ancilotto, & Farella, 2022) applied to GAN for face-swapping, tailored for resource-constrained platforms such as the Kendryte K210 microcontroller. This platform achieves processing speeds exceeding 15 frames per second with an FID score below 150, consuming less than 300mW. The processing pipeline comprises four steps, employing PhiNets combined with cutting-edge solutions for each task. Face detection, landmark detection using PFLD (Practical Face Landmark Detection) (Guo, et al., 2019), image alignment, and face generation/face swapping utilising a GAN configuration via PhiNets enable high-quality, real-time face generation on resource-constrained devices. The final step, based on FSGAN (Face Swapping GAN) (Nirkin, Keller, & Hassner, 2019), modified by replacing encoder and decoder networks with PhiNets, achieves full convolutional efficiency. This adaptation enables the entire pipeline from face detection to face swapping to fit within the target K210 platform, operating at 46mJ/frame and 6fps@280mW. Nonetheless, this solution remains in the prototype stage, with ongoing efforts to further refine and enhance its performance.

By following the above techniques in MARVEL, anonymisation aims to ensure that sensitive information within audio recordings remains protected by eliminating any speaker-related data, thereby rendering the speaker unidentifiable. In video anonymisation, the described techniques are employed to safeguard sensitive information by erasing identifiable visual cues, thereby ensuring the anonymity of individuals captured in the footage. More details on the anonymisation techniques developed and applied in MARVEL can be found in deliverables D3.3[54] and D3.5[55].

[54] "D3.3: E2F2C Privacy preservation mechanisms," Project MARVEL, 2022. https://doi.org/10.5281/zenodo.7541694

[55] "D3.5 - Multimodal and privacy-aware audio-visual intelligence – final version," Project MARVEL, 2023. https://doi.org/10.5281/zenodo.8147164

Bias and fairness

AI bias refers to recurring errors in computer systems causing unfair outcomes and favouring specific user groups. Bias can arise from various factors like algorithm design, unintended data use, or social expectations, impacting platforms like search engines and social media. It can reinforce biases related to race, gender, sexuality and ethnicity. Key types of AI bias in the MARVEL framework include Training data bias, Algorithmic focus bias, and Transfer context bias. Fairness represents a foundational principle supporting equal treatment in line with societal norms. It is closely linked to justice, striving to offer unbiased and non-discriminatory treatment, thereby guaranteeing equitable access to opportunities for all.

In MARVEL, a comprehensive set of precautions was developed to further ensure the framework's avoidance of unfair treatment, providing guidelines for the development of audio, visual, and multimodal AI subsystems and other AI components.

1. **Careful selection/augmentation of training datasets – no biased training will be used.** The MARVEL pilots initiated data collection for AI model training early in the project, employing diverse and representative datasets. For the GRN pilot, data were randomly sampled from locations where the MARVEL framework operates, focusing on real-life road junctions to detect traffic entities. MT's dataset, recorded in public spaces for anomaly detection, carefully controlled the quantity of samples for targeted anomalous behavior during staged recordings, ensuring a comprehensive representation and avoiding biases associated with incomplete or unrepresentative training examples. In the UNS pilot, a well-balanced dataset was synthetically generated in a controlled scenario, effectively preventing biases associated with skewed or unbalanced data.
2. **Ensure that data will come from diverse and representative set of data subjects.** Data for training and developing methodologies within MARVEL were collected from diverse cities and regions, employing a random sampling approach to ensure representation across all data subjects. This strategy aimed to prevent specific focus on particular subjects, promoting fairness. Furthermore, various AI methodologies were trained using publicly available datasets, as outlined in the related deliverables.
3. **The data acquisition will cover a fair time span.** GRN's data collection spanned from M6 to M30, aligning with the project's ongoing needs, while MT's staged recordings took place at specific intervals (M16, M28, and M29) to strategically cover anomaly detection scenarios. In the case of UNS, staged recordings were limited in duration, customised to the specific goals of the use cases.
4. **No selection or rejection of certain types of input data will be performed.** The data used for training AI models were randomly selected from real-life environments, ensuring the absence of algorithmic focus bias throughout the development of AI components. This methodology avoided the selective inclusion or exclusion of specific types of input data.
5. **Continuous monitoring of the results to identify potential issues related to bias, discrimination, or poor performance of the AI system.** Since the design phase of MARVEL, a crucial aspect has been to enable end-users to observe potential anomalous events, allowing them to assess the accuracy of event analysis by the AI system. This functionality is an integral part of the MARVEL framework, facilitating continuous system enhancement and empowering end-users through the UI to identify possible issues such as poor performance, bias, or discrimination, and take necessary corrective actions.

6. **Ensure that its components will work reliably and efficiently across different cities in different countries.** The AI models created were effectively deployed in the three unique pilots of the MARVEL framework. Datasets from diverse cities were employed for both training and testing the components, and extensive testing and evaluation procedures conducted across these pilot cities demonstrated the adaptability, reliability, and efficiency of the methodologies in real-world scenarios, as detailed in D5.5.
7. **Equal access to services developed for MARVEL's framework.** From the project's inception, a fundamental goal has been to guarantee equal access to services within the MARVEL framework. Offering a diverse range of AI methodologies that can be customised to meet end users' needs, MARVEL prioritises inclusivity, accessibility, and usability through a straightforward and user-friendly interface. The user-centric UI design process, which incorporates feedback from pilots and stakeholders, ensures accessibility for users with varying technical expertise, and ongoing support, including training and education, is provided by MARVEL experts to potential end-users.
8. **The developed AI models/algorithms and the respective datasets will become publicly available.** AI methodologies developed in WP3 and WP4 resulted in several publications featured in top-tier conferences and journals, showcasing methodologies, associated code, and utilised datasets. These resources are openly available to the public through the MARVEL website and repositories such as Zenodo and GitHub, with additional detailed information presented in documents D3.1[56], D3.3, D3.4[57], and D3.5.
9. **A wide range of stakeholders will be used for the design and development of the MARVEL AI system.** In the MARVEL project, ten diverse use cases were implemented across three smart cities, with stakeholders playing a vital role. From the project's start, feedback from end-users like local police and road managers shaped these cases, addressing smart city challenges. Continuous feedback, collected through various events, allowed for refining the framework. External evaluations of the use cases ensured a thorough assessment, with the AI system gaining attention as a crucial subsystem in stakeholder interests.
10. **The impact of the AI system on the potential end-users and/or subjects will be assessed.** The evaluation of the impact of the MARVEL AI subsystem incorporates diverse channels, including feedback from end-users via surveys conducted internally by the pilots and externally by independent evaluators and stakeholders. Comprehensive results and insights from these assessments are documented in D6.4, while the system's performance, measured against predefined benchmarks and KPIs related to end-user satisfaction and system effectiveness, is detailed in D5.5 and D6.4.

Responsible & Trustworthy AI

The EU's AI regulation, introduced in April 2021[58], marks a significant milestone as the first legal framework dedicated to fostering trustworthiness and excellence in AI. Emphasising a human-centric approach, the regulation recognises AI's potential benefits while acknowledging associated risks to rights and interests. The regulation classifies AI applications into four risk levels, with MARVEL being categorised as a Limited-risk system, as its AI subsystem is not

[56] "D3.1: Multimodal and privacy-aware audio-visual intelligence – initial version," Project MARVEL, 2022. https://doi.org/10.5281/zenodo.6821318

[57] "D3.4 - MARVEL's federated learning realization," MARVEL Project 2023. https://doi.org/10.5281/zenodo.7543936

[58] Ethics guidelines for trustworthy AI: https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai

the principal outcome, it is not being used to manage critical infrastructure nor to run public services and decide on individuals' rights. On the contrary, MARVEL relies on AI in order to enhance privacy.

Even though MARVEL is not considered a high-risk system, from the beginning of the project we acknowledged the need to integrate to the degree possible the Responsible and Trustworthy AI principles into the framework. Below, we assess the alignment of MARVEL with the seven pivotal Trustworthy AI requirements, encompassing developers, deployers, end-users, and the broader societal context.

1. **Human agency and oversight:** MARVEL adopted a human-centric design, emphasising on tools that assist rather than replace human decision-making. The system produces alerts, explored by experts, as it avoids automatic decision-making. Throughout development, active engagement with end-users, stakeholders, and experts ensured user needs were considered, granting users the ability to understand, verify, and question AI results, maintaining human oversight and final decision authority.
2. **Technical robustness and safety:** The extensive benchmarking phase of MARVEL's AI methodologies, documented in D5.5, revealed a high likelihood of accurate decision-making. The AI subsystem demonstrated reliable performance in ten pilot use cases across three smart cities, but further investigation is required for result reproducibility, as this aspect was not addressed during benchmarking. To enhance safety, all data undergo anonymisation before processing, reducing potential vulnerabilities to the AI subsystem from malicious actors.
3. **Privacy and data governance:** MARVEL prioritises privacy and data protection by implementing video and audio anonymisation techniques at close to the edge as possible, ensuring that data processing occurs exclusively on anonymised data. The framework maintains secure data transmission and upholds data quality and integrity through established security provisions and a quality assurance process. The project's proactive measures at the project's outset have minimised the likelihood of bias in data and AI methodologies, and access to anonymised data is restricted to authorised users, supported by MARVdash's flexible functionality catering to different roles in data access.
4. **Transparency:** In MARVEL, datasets align with FAIR data management principles, emphasising traceability, while the AI methods' processes are documented in the project’s deliverables. The user-friendly MARVEL UI simplifies technical details, and the structured AI methods offer clear reasoning, providing end-users with accessible and comprehensible insights.
5. **Diversity, non-discrimination, and fairness:** The MARVEL framework prioritises diversity, non-discrimination, and fairness in constructing a Trustworthy AI subsystem, with a low likelihood of AI bias as analysed above. To further mitigate potential biases, the framework employs a set of measures outlined and analysed for the development of audio, visual, and multimodal AI subsystems.
6. **Societal and environmental well-being:** The primary aim of MARVEL was to create a framework that would be used for the benefit of the citizens. Even though not direct end-users, the citizens experience indirect benefits from the adoption of MARVEL from a smart city. MARVEL Pilots’ use cases include enhancing safety in Trento and optimising road traffic management in Malta. The real-time alerts in Trento improve citizen safety by enabling quicker responses to potentially dangerous situations,

explored by local police. Similarly, in Malta, MARVEL contributes to elevating citizens' quality of life by understanding road user behaviour and optimising transportation planning Additionally, MARVEL's E2F2C continuum processes data close to the edge, enhancing energy efficiency and conserving network bandwidth.

7. **Accountability:** AI methodologies in MARVEL are designed to be clear and understandable, with a simplified UI for users without technical background. Detailed documentation fosters traceability, and redress mechanisms, including user review and human oversight, are in place. Further exploration is needed for accountability, reporting, and minimizing negative impacts during operational adoption by a smart city.

4.2 Operational concerns

The successful implementation of a project emphasising scalability and extensibility across edge, fog, and cloud layers necessitates a meticulous consideration of operational concerns. Addressing these concerns ensures the project's robustness, reliability, and ease of maintenance. The main scalability and extensibility requirements that would guarantee the successfulness of both scaling up and extending the MARVEL framework were presented in Section 2, while in Section 3 how these requirements are addressed was described.

In the context of Quality Assurance, within the MARVEL framework, a series of system-level tests were carried out several times with the most notable on R1 and R2, reported on the deliverables D5.4 and D5.6 respectively. In addition, a series of benchmark tests took place to measure the system performance under high load, as reported in deliverable D5.5. Another critical aspect when it comes to extensibility and the maintenance of a large-scale system is documentation. This is of paramount importance as the technicians as well as system architects will need a very detailed view of the system either to maintain it or extend it. In MARVEL, the documentation process was adopted from the beginning of the project with the creation of a GitLab repository where all the components had to document all the details required for inter-component. Throughout the lifespan of the project, the documentation was enhanced with the latest advancements.

The most effort-intensive task in scaling up, extending and maintaining a system relies on system management. As described in Section 3.3, MARVEL has adopted mechanisms that substantially mitigate this challenge. These mechanisms use frameworks that enable automation and provide in a user-friendly and summarised way the monitoring and debugging information required for effortless operational supervision.

4.3 User Interface concerns

SmartViz, the data visualisation toolkit at the end of the MARVEL pipeline, serves as a vital User Interface (UI) for end users and is an integral component of the pipeline. It provides an interactive environment to engage with and gain insights from the data processed via the E2F2C system. This section is dedicated to thoroughly addressing accessibility issues. Our primary focus in this section is to make SmartViz not only accessible but also intuitive for all MARVEL project participants, ensuring its functionalities are easy to use and understandable.

Robust Data Availability

SmartViz resilience is crucial. Its effectiveness depends on continuous data access from the Elasticsearch repository and the uninterrupted operation of the data pipeline. An elaborated overview of SmartViz's architecture (Figure 31) and its integration with various MARVEL

components is detailed in Section 2.2 of D4.6[59]. Proactive monitoring and response mechanisms are implemented to maintain consistent access to SmartViz. Should there be any issues with the DFB, MARVdash, or the pipeline, SmartViz is designed to maintain core functionalities, thereby minimising disruptions to its full range of features.

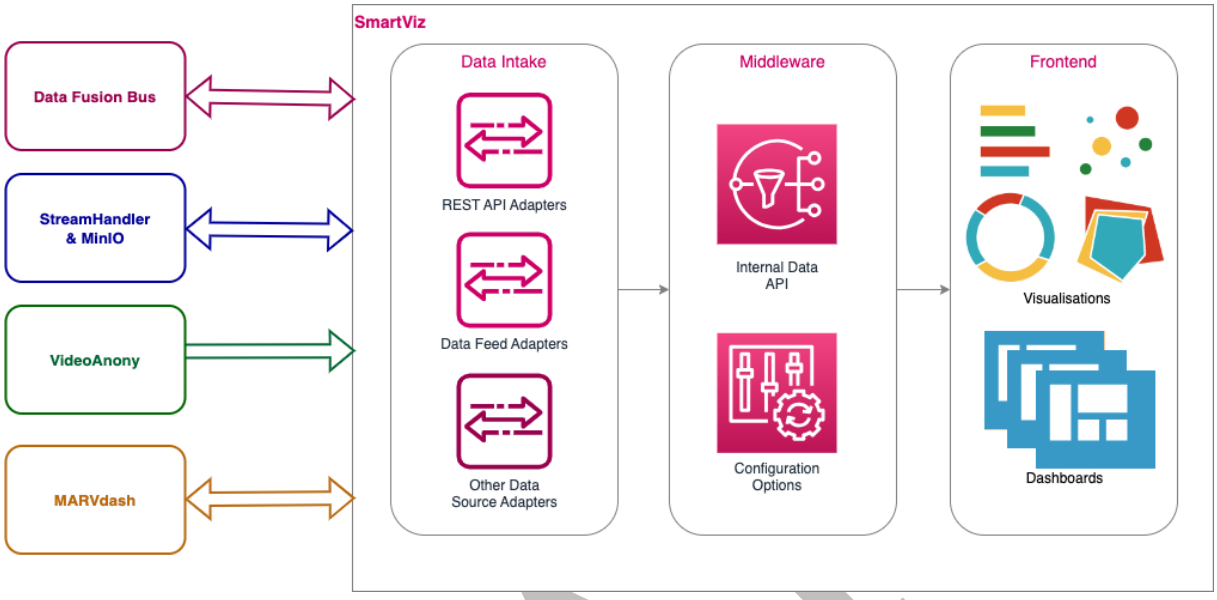

Figure 31: SmartViz internal architecture

User-Friendly Design

SmartViz is designed with the user in mind, focusing on a straightforward and engaging interface. Its design is tailored to meet the diverse needs and preferences of a broad user base, aiming to provide a smooth and productive user experience. A detailed discussion of the design principles and user interface strategy for SmartViz within the MARVEL project is available in Section 2 of D4.3[60].

User Feedback Integration

Incorporating user feedback is a critical aspect of SmartViz, particularly regarding the MARVEL pipeline's output. A user feedback mechanism within SmartViz for the MARVEL pipeline inference results allows users to report data validity, ensuring continuous improvement for the AI models deployed within the project. This feature allows users to comment on the relevancy and accuracy of the data, which is vital for the ongoing enhancement of the AI models within the project. This direct feedback loop ensures that the system evolves and improves based on actual user experiences and observations.

Documentation and Training

SmartViz comes with comprehensive, easy-to-understand documentation. Each visualisation widget is explained in detail, including its functionalities and purpose, directly within the UI.

[59] "D4.6: MARVEL's decision-making toolkit – final version," Project MARVEL, 2023. https://doi.org/10.5281/zenodo.8147077

[60] "D4.3: MARVEL's decision-making toolkit – initial version," Project MARVEL, 2022, https://doi.org/10.5281/zenodo.7543685

Moreover, the project's YouTube channel[61] hosts an array of instructional videos for each use case, aiming to equip users with the knowledge and skills needed to fully utilise SmartViz's features. These resources are crafted to be user-friendly and informative, catering to users with varying levels of technical expertise.

Integration and Validation

Ongoing validation and testing at the integration points with MARVdash, StreamHandler, Minio, and DFB are essential for ensuring smooth and reliable operations. These routine checks are vital for maintaining the integrity and efficiency of communications and operations within SmartViz, ensuring that the system runs smoothly and effectively.

Continuous Improvement

SmartViz adopts an iterative development approach, heavily influenced by feedback from pilot users and end users, as well as the evolving of a use cases guide. This strategy is central to our commitment to continuous improvement, ensuring that SmartViz remains flexible and responsive to the changing needs and preferences of users.

By proactively addressing these accessibility and usability concerns, SmartViz is well-positioned to offer a user-friendly, adaptable, and comprehensive platform. It is designed to cater to the varied needs of all participants in the MARVEL project, enhancing user experience and contributing significantly to the project's overall success.

[61] https://www.youtube.com/channel/UCUqSx3-4anthxkSAGgvQMdQ

5 Conclusions

In the pursuit of a scalable and extensible framework for large-scale real-world implementation in the context of Smart Cities, this deliverable has provided a comprehensive exploration of the MARVEL project's key aspects and considerations. This document presents the work carried out within Task 5.5. To summarise the key findings and insights garnered in this report, the deliverable initiated with an overview of the main concepts and use cases within the MARVEL project, emphasising on scalability considerations across the E2F2C continuum. Then, an overview of the requirements essential for a contemporary scalable and extensible framework, setting the stage for the subsequent detailed examination in Section 3 was unfolded. Section 3 dug into the measures, methods, and technologies incorporated in the design of the highly scalable and extensible MARVEL framework. The organisation of sub-sections mirrored the primary aspects contributing to scalability and extensibility, with detailed insights into the features and implementation examples of each relevant component. Finally, in Section 4 all the legal, ethical, privacy and operational concerns related to a system to be adopted by third parties as well as be deployed on large scale were described as well as how they applied to MARVEL.

In conclusion, the MARVEL project has demonstrated a concerted effort in addressing the intricacies of scalability and extensibility. By combining a nuanced understanding of main concepts, meticulous design and implementation strategies, and a thorough examination of ethical and technical considerations, MARVEL stands as a robust framework poised for impactful real-world deployment. The limitations and challenges identified provide a roadmap for further refinement, ensuring continuous improvement in the pursuit of scalable and extensible solutions within the dynamic landscape of edge, fog, and cloud computing.

6 Bibliography

Houston, G. (1993). ISO 8601: 1988 Date/Time Representations.

Kim, L. (2022). Cybersecurity: Ensuring confidentiality, integrity, and availability of information. In *Nursing Informatics: A Health Informatics, Interprofessional and Global Perspective* (pp. 391-410). Springer.

Liu, S., Cao, Y., Wang, D., Wu, X., Liu, X., & Meng, H. (2021). Any-to-many voice conversion with location-relative sequence-to-sequence modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29*.

Chen, S., Chengyi, W., Chen, Z., Wu, Y., Liu, S., Chen, Z., . . . others. (2022). Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing, 16*(6), 1505-1518.

Koluguri, N. R., Park, T., & Ginsburg, B. (2022). TitaNet: Neural Model for speaker representation with 1D Depth-wise separable convolutions and global context. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 8102-8106). IEEE.

Miao, W., Geng, J., & Jiang, W. (2022). Semi-supervised remote-sensing image scene classification using representation consistency siamese network. *IEEE Transactions on Geoscience and Remote Sensing, 60*, 1-14.

Chen, Z., Rosenberg, A., Zhang, Y., Wang, G., Ramabhadran, B., & Moreno, P. J. (2020). Improving Speech Recognition Using GAN-Based Speech Synthesis and Contrastive Unspoken Text Selection., (pp. 556-560).

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., . . . Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. doi:arXiv:1704.04861

Paissan, F., Ancilotto, A., & Farella, E. (2022). PhiNets: a scalable backbone for low-power AI at the edge. *ACM Transactions on Embedded Computing Systems, 21*(5), 1-18.

Guo, X., Li, S., Yu, J., Zhang, J., Ma, J., Ma, L., . . . Ling, H. (2019). PFLD: A practical facial landmark detector. *arXiv*. doi:arXiv:1902.10859

Nirkin, Y., Keller, Y., & Hassner, T. (2019). Fsgan: Subject agnostic face swapping and reenactment. *Proceedings of the IEEE/CVF international conference on computer vision*, 7184-7193.

European Union. (2010). Charter of Fundamental Rights of the European Union. *Official Journal of the European Union C83, 53*, 380.

Ancilotto, A., Paissan, F., & Farella, E. (2023). XimSwap: many-to-many face swapping for TinyMl. *ACM Transcactions on embedded computing systems*.

Hantke, S., Eyben, F., Appel, T., & Schuller, B. (2015). iHEARu-PLAY: Introducing a game for crowdsourced data collection for affective computing. In *2015 International Conference on Affective Computing and Intelligent Interaction (ACII)* (pp. 891-897). doi:10.1109/ACII.2015.7344680

Schulzrinne, A. R., Lanphier, R., & Henning. (1998, 4 1). Real Time Streaming Protocol (RTSP). *RFC 2326*. RFC Editor. doi:10.17487/RFC2326

Lorenzo, V., Nardini, F. M., Andrea, P., & Raffaele, P. (2022). Dynamic hard pruning of neural networks at the edge of the internet. *Journal of Network and Computer Applications*, *200*, 103330.