European Commission

Horizon 2020
European Union funding
for Research & Innovation

Big Data technologies and extreme-scale analytics

MARVEL

Multimodal Extreme Scale Data Analytics for Smart Cities Environments

D5.2: Technical evaluation and progress against benchmarks – initial version[†]

Abstract: The purpose of this deliverable is to describe in detail the technical evaluation and progress against benchmarks. The benchmarking strategy was defined in WP1, and this document describes how the benchmarking is implemented for the components in the Minimum Viable Product (MVP) of the MARVEL project. The role of each component in the MVP together with the description of the development status is discussed in detail before the benchmarking process of components is described. This process involves defining the measurement metrics and data, as well as the state-of-the-art baselines, and reporting the measurement results together with the observations about the results. In addition to this, the contribution to MARVEL KPIs per component is described and expected future results are discussed. The final version of this document will be delivered by the end of the project, and it will contain the benchmarking of the full MARVEL framework.

Contractual Date of Delivery	28/02/2022
Actual Date of Delivery	28/02/2022
Deliverable Security Class	Public
Editors	*Toni Heittola and Tuomas Virtanen (TAU)*
Contributors	FORTH, IFAG, AU, ATOS, CNR, FBK, TAU, STS, ITML, GRN, ZELUS, PSNC
Quality Assurance	*Alessio Brutti (FBK)* *Manolis Falelakis (INTRA)*

[†] *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. *Manolis Falelakis (INTRA)*
2. *Alessio Brutti (FBK)*

Revisions

Version	Date	By	Overview
0.7.1	28/02/2022	Editors	Addressed comments from the PC
0.7.0	28/02/2022	Editors	Addressed comments from the PC
0.6.0	23/02/2022	Editors	Revised document after internal reviews
0.5.0	22/02/2022	Editors	Revised document after internal reviews
0.4.0	15/02/2022	Editors	Addressed comments from IR1 and IR2.
0.3.1	14/02/2022	IR1 (INTRA), IR2 (FBK)	Comments from IR1 (INTRA) Comments from IR2 (FBK)
0.3.0	08/02/2022	Editors	Phase 2 consolidated version (for internal review).
0.2.0	01/02/2022	Editors	Phase 1 consolidated version.
0.1.0	27/12/2021	Editors	Final version of the ToC.
0.0.1	27/12/2021	WPL (INTRA), STPM (UNS)	Comments from WPL and STPM on the ToC.
0.0.0	09/12/2021	Editors	ToC.

Disclaimer

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

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

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AI	Artificial Intelligence
API	Application Programming Interface
AV	Audio-Visual
AVCC	Audio-Visual Crowd Counting
CATFlow	Data acquisition Framework
CCTV	Closed-Circuit Television
CNCF	Cloud Native Computing Foundation
CNN	Convolutional Neural Network
CPU	Central Processing Unit
D#.#	Deliverable
DatAna	Data acquisition Framework
DFB	Data Fusion Bus
DL	Deep Learning
DMP	Data Management Platform
E2F2C	Edge-to-Fog-to-Cloud
EC	European Commission
ER	Error Rate
FedL	Framework and implementation of ML algorithms – Federated learning
FID	Fréchet Inception Distance
FLOPs	Floating Point Operators
GAN	Generative Adversarial Network
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HDFS	Hadoop Distributed File System
HTTPS	Hypertext Transfer Protocol Secure
IoT	Internet of Things
IOU	Intersection over Union
IP	Internet Protocol
ISO	International Organisation for Standardisation
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
M#	Month #
MAE	Mean Absolute Error

mAP	Mean Average Precision
MARVdash	Kubernetes CARV dashboard
MEMS	Micro-Electro-Mechanical Systems
MFLOPS	Million Floating-Point Operations Per Second
ML	Machine Learning
MP	Megapixels
MSE	Mean Squared Error
NVR	Network Video Recorder
OpenCV	Open Source Computer Vision Library
POE	Power over Ethernet
RAM	Random Access Memory
REST	Representational State Transfer
RPi	Raspberry Pi
RTSP	Real Time Streaming Protocol
S3	Simple Storage Service
SD	Secure Digital
SED	Sound Event Detection
SLA	Service Level Agreement
SL-ViT	Single-Layer Vision Transformer
SmartViz	Advanced visualisation toolkit
SNR	Signal-to-Noise-Ratio
SotA	State-of-the-Art
SSD	Solid State Drive
SUS	System Usability Scale
T#	Task #
THD	Total Harmonic Distortion
UEQ	User Experience Questionnaire
UI	User Interface
URL	Uniform Resource Locator
VCPU	Virtual Central Processing Unit
VideoAnony	GANs for video anonymisation
VM	Virtual Machine
WiFi	Wireless Fidelity
WP	Work Package

Executive Summary

This document presents in detail the technical evaluation and progress against benchmarks in the MARVEL project. This is the initial version of the document, and it deals with the components in the Minimum Viable Product (MVP). The final version of the benchmarking document will be prepared closer to the end of the project (M30), and it will contain benchmarking for the full MARVEL framework. The benchmarking strategy defined in WP1 (reported in D1.2) is implemented in this document for each component involved with the MVP. For each component, the role in the MVP is defined together with the information about the current status of the component. The state-of-the-art baseline is defined, and benchmarking process, measurement results, and result observations are presented. Lastly, the contribution to the project's KPIs is described and expected future results are discussed.

The document first introduces the benchmarking strategy of MARVEL and discusses different types of benchmarking. Components related to sensing and perception included in this document are GRNEdge, a portable device used to record data, CATFlow, an algorithm to detect the different types of vehicles and track them, MEMS, hardware used for audio data acquisition, and SED@Edge, a component to detect sound events using low-resource devices. VideoAnony component included in this document is related to security, privacy, and data protection, and it performs video anonymisation for captured video content. Components related to data management and distribution included in this document are DFB, a component allowing a trustworthy way of transferring data between components and the permanent storage, and DatAna, a data management platform allowing the creation of data processing pipelines graphically. Audio, visual, and multimodal AI-related components included in this document are audio-visual crowd counting, a component to estimate the number of people being present in the field of view of a camera, and sound event detection, a component recognising vehicle types based on an audio signal. MARVdash, a component related to optimised E2F2C processing and deployment implements a dashboard for instantiating and deploying services. Components related to the E2F2C infrastructure included in this document are GRN edge infrastructure, GRN fog infrastructure, and PSNC cloud infrastructure. Components related to decision making and user interaction included are SmartViz, a data visualisation component that constitutes the user interface of the decision-making toolkit, and MARVEL Data Corpus-as-a-Service, a component where processed multimodal audio-visual data is stored and released free of charge, as a service.

Results from this document (D5.2) together with its final version (D5.5) will be used within *'WP6 – Real-life societal experiments in smart cities environment'*. In the 'Task 6.3 – *Evaluation and Impact analysis'* (task active M12-M36), KPIs are monitored and evaluated for each experiment in operational terms as well as in technical terms, and benchmarking results will be essential part of this process.

1. Introduction

1.1. Purpose and scope of this document

The purpose of this document is to describe in detail the technical evaluation and progress against benchmarks. This initial version of the document includes all components in the Minimum Viable Product (MVP). The final version will be prepared by the end of the project (M30), and it will contain the benchmarking results of all components from the project. The benchmarking strategy was defined in WP1 and reported in D1.2 "MARVEL's experimental protocol" (M8). This document defines in detail the benchmarking process for components involved in the MVP and reports the progress made against benchmarks.

1.2. Contribution to WP5 and project objectives

This deliverable has been produced within the context of '*WP5 – Infrastructure Management and Integration*', and more specifically under '*Task 5.4 – Quantifiable progress against societal, academic and industry validated benchmarks'.* Work package 5 focuses on the Edge-to-Fog-to-Cloud (E2F2C) framework that allows for powerful, scalable and real-time processing of multimodal audio-visual data on top of distributed deployment of MARVEL Machine Learning (ML) models. The main objective of this WP is to ensure successful E2F2C framework delivery. This includes 1) provision and configuration of an HPC infrastructure, 2) orchestration of infrastructure resource management and optimised automatic usage of external resources (computational and storage), 3) integration and quality assurance of software releases, 4) quantifiable progress against societal, academic and industry-validated benchmarks, and 5) continuous alignment with the responsible AI planning and guidelines. This document contributes directly to the fourth objective.

Benchmarking also contributes to three objectives of MARVEL:

- Objective 1: "Leverage innovative technologies for data acquisition, management and distribution to develop a privacy-aware engineering solution for revealing valuable and hidden societal knowledge in a smart city environment",

- Objective 2: "Deliver AI-based multimodal perception and intelligence for audio-visual scene recognition, event detection and situational awareness in a smart city environment",

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

For these objectives, multiple KPIs are defined and benchmarking is used to verify that they are met.

1.3. Relation to other WPs and deliverables

This deliverable relies on the foundational work on the benchmarking strategy conducted within '*WP1 – Setting the scene: Project setup'* and documented in deliverable *'D1.2 – MARVEL's Experimental protocol*'. Results from this deliverable (D5.2) and the final version of it (D5.5) will be used within *'WP6 – Real-life societal experiments in smart cities environments'* and more specifically in the ‘Task 6.3 – *Evaluation and Impact analysis’* (task active M12-M36). In this task, KPIs are monitored and evaluated for each experiment in operational terms as well as in technical terms, and benchmarking results will be essential part of this process.

1.4. Structure of the document

The structure of this document is as follows:

- Section 2 introduces the benchmarking strategy of the MARVEL and discusses different types of benchmarking.

- Section 3 describes benchmarks for the components related to sensing and perception subsystems of the MARVEL framework. Components dealt with are GRNEdge, CATFlow, MEMS microphones, and SED@Edge.

- Section 4 describes benchmarks for the security, privacy and data protection subsystems of the MARVEL framework. Benchmarking for video anonymisation component VideoAnony is described in this section.

- Section 5 describes benchmarks for subsystems related to data management and distribution. Components dealt with are DFB and DatAna.

- Section 6 describes benchmarks for audio, visual and multimodal AI components. Components dealt with are audio-visual crowd counting and sound event detection.

- Section 7 describes benchmarks for the optimised E2F2C processing and deployment component MARVdash.

- Section 8 describes benchmarks for E2F2C infrastructure components: GRN edge infrastructure, GRN fog infrastructure, and PSNC cloud infrastructure.

- Section 9 describes benchmarks for decision making and user interaction subsystems of the MARVEL framework. Components dealt with are SmartViz and MARVEL Data Corpus-as-a-Service.

- Section 10 summaries benchmarks.

- Section 11 concludes this document.

2. Benchmarking strategy

Benchmarking is perceived as one of the pillars of MARVEL. In fact, quoting GA, "MARVEL will systematically, qualitatively and quantitatively assess the proposed approaches in Pillars I-III, through a thorough exploration and adoption of benchmarks". Therefore, in MARVEL we are looking at a coherent strategy to make this wish a reality helping to assess the MARVEL components and platform.

In this sense, this document presents the first step towards this goal using the MARVEL MVP artefacts. This section reports on the scope of benchmarking the MVP components and the current view on the benchmarking strategy for the upcoming months. To this extent, a Benchmarking workshop was held on the 25th of January 2022 to assess the status of the MVP benchmarking and discuss the strategy. This section also provides the outcomes of the workshop in terms of strategy.

2.1. Benchmarking types in MARVEL and scoping the MVP

There are many different definitions of benchmarking. One of the most accepted is the one provided by Wikipedia in the scope of benchmarking computing programs: "In computing, a benchmark is the act of running a computer program, a set of programs, or other operations, in order to assess the relative performance of an object, normally by running a number of standard tests and trials against it."[1]

Deliverable D1.2 (MARVEL / D1.2, 2021) reported in Section 7 the initial ideas related to benchmarking. In this sense, WP1 paved the way towards setting the initial benchmarking strategy, especially for the MVP and provided an overview of the different types of benchmarks. The objective of this section is to summarise the types of benchmarks that are going to be used in MARVEL and provide an initial overview of them, as well as explain what has been done in the scope of benchmarking the MVP results.

2.1.1. Types of benchmarks envisaged in MARVEL

2.1.1.1. *Business benchmarking*

As explained in D1.2, "**Business Benchmarking** will help to validate and assess the MARVEL process, sustainability, and financial perspectives, especially related to the usage of AV data in the scope of Smart Cities.".

In this context, D1.2 provided a table of 6 suggested metrics (Revenues increase, Profit increase, Cost reduction, Time efficiency, Product service quality, and Business model innovation) to evaluate the business perspective based on the work on business benchmarking carried out in the DataBench project (DataBench / D4.3, 2020). In order to assess these indicators, several means have been proposed, via surveys, market studies, or usability questionnaires for the different MARVEL pilots. These initial ideas have to be worked out in the coming months in collaboration with WP7 (especially with tasks T7.3 and T7.5). These business KPIs will serve as input for future business and exploitation perspectives.

2.1.1.2. *Societal benchmarking*

As in the previous case, D1.2 provided a first set of societal benchmarking and means to assess them from the pilots' perspective, mostly in the form of targeted surveys and usefulness for the different stakeholders. These societal objectives are clearly related to the work done in WP6 in

[1] Wikipedia: https://en.wikipedia.org/wiki/Benchmark_(computing)

relation to the outputs and evaluation of the results of the pilots. Therefore, the benchmarking of these societal objectives carried out in WP5 will provide inputs to WP6 to analyse the results and provide the final takeovers for the pilots.

2.1.1.3. *Technical benchmarking*

Technical benchmarking is one of the main goals and undoubtedly the main bulk of the work on benchmarking in MARVEL. There are several categories of technical benchmarks in the computing arena. In MARVEL, we follow the categories from the DataBench project explained in D1.2, dividing them into micro (or component), application, and user experience benchmarks:

- **Micro Benchmarks, sometimes known as component benchmarks,** are used to assess very specific low-level or component functionalities. These types of benchmarks are useful to assess the performance of the MARVEL components in isolation, and in many cases are using very specific benchmarks or well-known datasets or settings provided by the community.
- **Application Benchmarks** are used to benchmark end-to-end scenarios or applications. These are typically designed to assess system-wide performance metrics and therefore they are on the radar of MARVEL. Examples of these types of benchmarks are BenchCouncil AIBench suite[2], MLCommons (former MLPerf)[3], DeFog[4], LEAF[5] or TCP-x-IoT[6] among others.
- **User Experience or usability benchmarks** help in the assessment of user-centric indicators and the user experience. They are in many cases qualitative and used in the scope of user interfaces, dashboards, or elements of a system that offer any user-oriented functionality. Typically, usability is measured using dedicated surveys and the technical capabilities using performance stress tools simulating concurrent users and user-oriented tasks.

2.1.2. Scope of benchmarking the MARVEL MVP

The MARVEL MVP is the initial realisation of the MARVEL architecture for a set of selected scenarios based on the Malta pilot. In this deliverable, the focus is on technical benchmarking of the MVP realisation, as the business and societal aspects require more maturity of the solution and further iterations with the pilots and potential exploitation outcomes.

From the technical perspective, micro or component benchmarks are appropriate to test individual components or part of the functionalities of the components, which is useful at the current state of the components (MVP). The components used in the MVP can be seen highlighted in yellow in the architecture of MARVEL depicted in Figure 1.

[2] https://www.benchcouncil.org/aibench/

[3] https://mlcommons.org/

[4] https://github.com/qub-blesson/DeFog

[5] https://leaf.cmu.edu/

[6] http://tpc.org/tpcx-iot/default5.asp

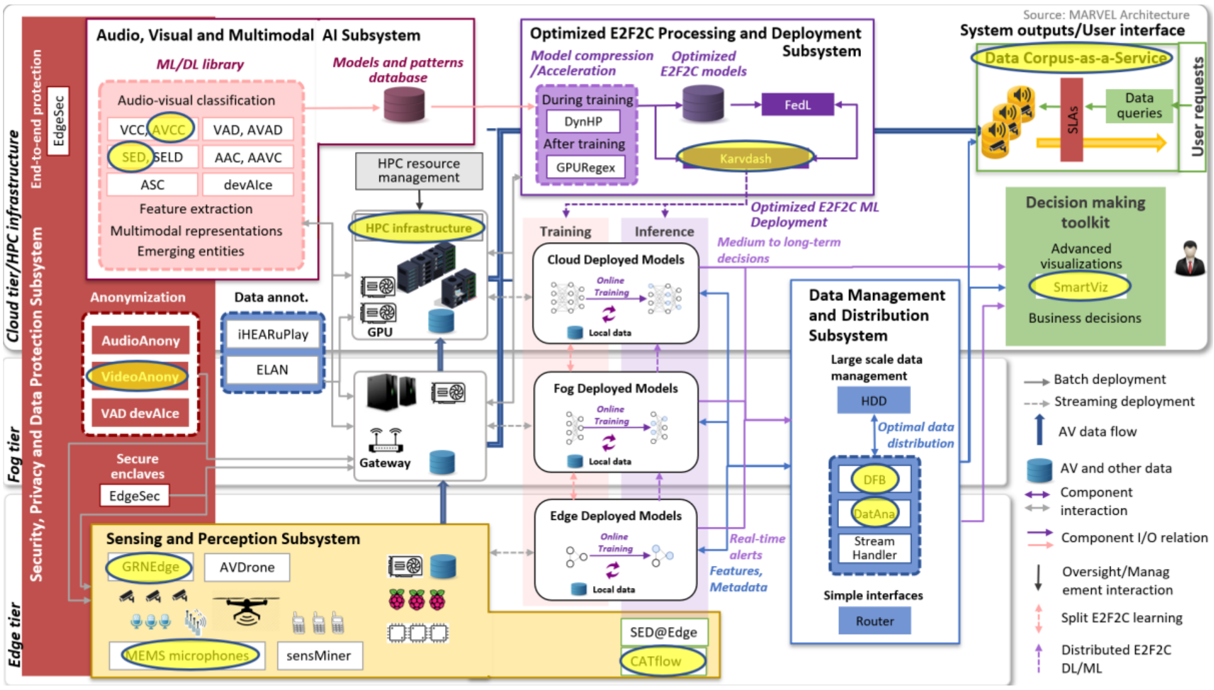


Figure 1. MARVEL architecture highlighting the components used for the MVP

Deliverable D5.1 provided an overview of the MARVEL components used in the three scenarios selected (Identify vehicle type and trajectories, Sound event detection and crowd counting, and Populate the MARVEL data corpus with AV data). For clarity's sake, the components used in these three scenarios can be seen in the following figures, with the related sub-architectures, based on the figures presented in D5.1.

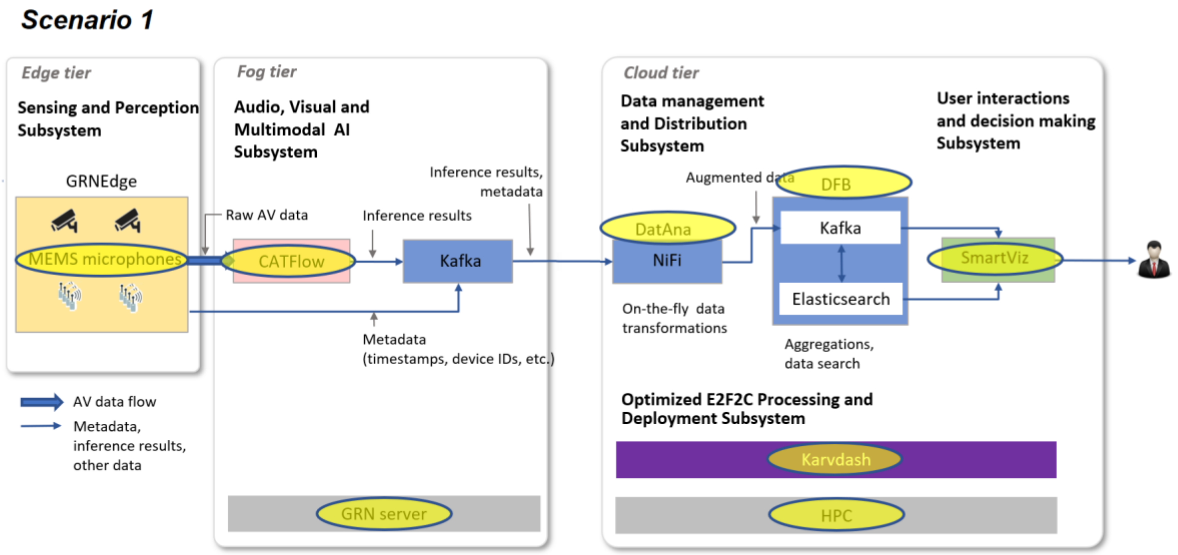

Scenario 1
Edge tier
Sensing and Perception Subsystem
GRNEdge
MEMS microphones
Fog tier
Audio, Visual and Multimodal AI Subsystem
Raw AV data
Inference results
Inference results, metadata
CATFlow
Kafka
Metadata (timestamps, device IDs, etc.)
GRN server
Cloud tier
Data management and Distribution Subsystem
Augmented data
DFB
DatAna
NiFi
Kafka
Elasticsearch
On-the-fly data transformations
Aggregations, data search
User interactions and decision making Subsystem
SmartViz
Optimized E2F2C Processing and Deployment Subsystem
Karvdash
HPC
AV data flow
Metadata, inference results, other data

Figure 2. Scenario 1 from the Malta pilot implemented for the MVP highlighting the components used

Figure 2 presents the first scenario of the MVP. In this scenario, the data collected at the edge is processed and analysed by the CATFlow component at the Fog server located in the GRN premises. The output is stored in a Kafka topic in a Microsoft Azure cloud service managed by GRN. From there, the Data Management platform components located at the MARVEL cloud managed by Karvdash over the PSCN infrastructure, retrieve the data (DatAna) and pass and process it (DFB) to make it available for the SmartViz component for visualisation.

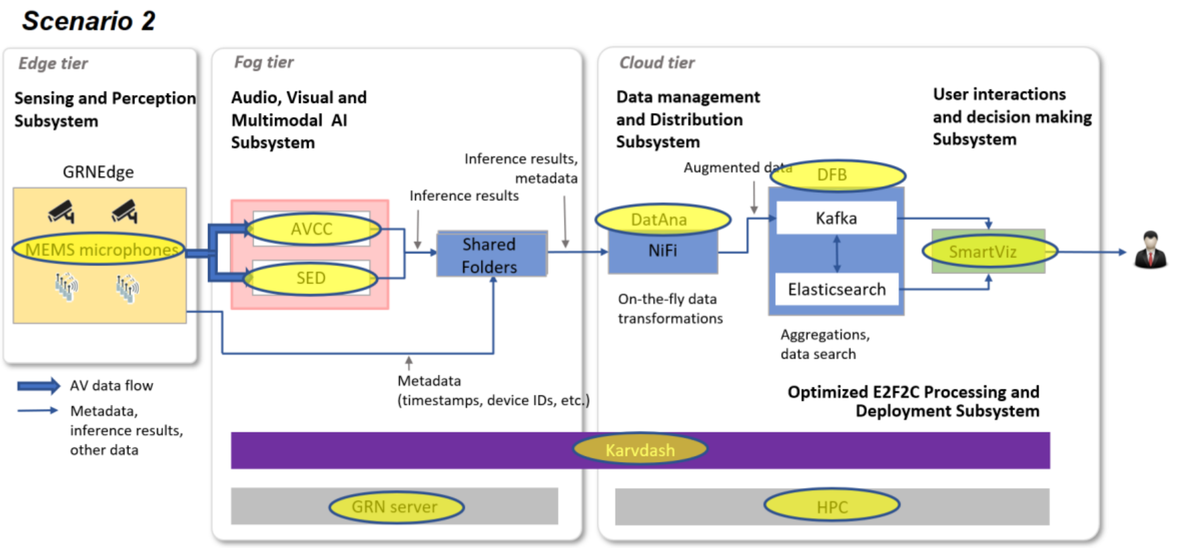

Scenario 2
Edge tier
Sensing and Perception Subsystem
GRNEdge
MEMS microphones
Fog tier
Audio, Visual and Multimodal AI Subsystem
AVCC
SED
Inference results
Inference results, metadata
Shared Folders
Metadata (timestamps, device IDs, etc.)
Cloud tier
Data management and Distribution Subsystem
DatAna
NiFi
On-the-fly data transformations
Augmented data
DFB
Kafka
Elasticsearch
Aggregations, data search
User interactions and decision making Subsystem
SmartViz
Optimized E2F2C Processing and Deployment Subsystem
Karvdash
GRN server
HPC
AV data flow
Metadata, inference results, other data

Figure 3. Scenario 2 from the Malta pilot implemented for the MVP highlighting the components used

Similar to the previous scenario, the second one shown in Figure 3 gets the data from the edge tier, but in this case, two of the AI components in the Audio, Visual, and Multimodal AI subsystem of the architecture have been implemented. AVCC infers the number of people in the video frames taken from a road crossing in Malta, while SED uses audio data to detect different types of vehicles. The outputs of these AI models are stored in shared folders of the Kubernetes cluster managed by Karvdash, and from there taken by DatAna, processed applying data transformation and normalisation and passed to the DFB, where the data is made available for the SmartViz component for further visualisation.

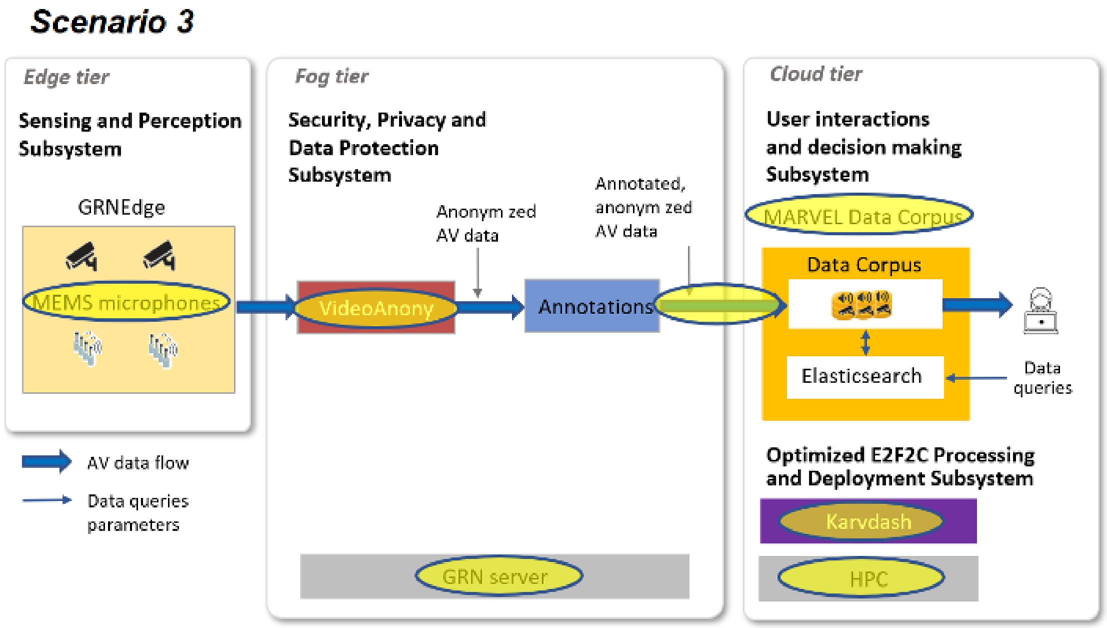


Figure 4. Scenario 3 (MARVEL Corpus) from the Malta pilot implemented for the MVP highlighting the components used

Finally, the third scenario shown in Figure 4 provides an overview of the way AV data flows to the MARVEL Corpus. The data captured at the edge is anonymised at the fog layer by the VideoAnony component and the manual annotations collected at GRN over this video data are collected by a set of scripts that move the data to the MARVEL Corpus stored in the MARVEL Cloud.

These three scenarios show the interactions of several MARVEL components composing a set of data-driven pipelines. The components have been integrated into a coherent MVP as reported in D5.1. In this document, we report on the performance evaluation of the different MARVEL components used in the MVP without looking in detail at the system-wide approach. This aspect will be further analysed in future work in WP5, but for the MVP the view is more into analysing the performance of the individual components using specific components or micro-benchmarks. This is in line with the strategy discussed in Section 2.2.2. However, benchmarking AV data is yet not very well catered for in the benchmarking community, although for most of the MARVEL's components established validation protocols exist, at least partially covering the MARVEL's target scenarios. In summary, for the MVP, we focus on component benchmarking.

2.2. Benchmarking strategy after the MVP

As mentioned in the previous section, component benchmarking is the choice selected to assess the MVP results. In this section, we discuss the current benchmarking strategy views not only for the present but mostly looking at the future releases of the platform.

2.2.1. Results from the first benchmarking workshop

One of the first milestones of the work done in T5.4 in WP5 is the organisation of the internal benchmarking workshop that was held on January 25, 2022. This workshop was a contractual obligation aiming at assessing the status of benchmarking the MVP and discussing the strategy on benchmarking for the second year of the project. It was a four-hour workshop organised by TAU and ATOS, devoting the first half to the MVP benchmarking aspects and the second to the discussion of the benchmarking strategy. All MARVEL partners were represented in the sessions (39 attendees).

For the first half of the workshop, the main objectives were the following:

- Common understanding of benchmarking.
- Status of benchmarking of MVP components.
- Solving doubts.
- Steps towards D5.2.

The first objective aimed at consolidating the understanding about the work to be done in benchmarking in MARVEL, by providing definitions, motivation, and revisiting the previous work done so far in the scope of WP1 (D1.2) and WP5 (T5.4). For the last three objectives above, the MVP component owners provided a brief presentation of their respective status and issues, if any, towards the finalisation of the first round of benchmarking results to be delivered in this document. The outcomes of this session were quite positive, showing that the component benchmarking is progressing in the right way.

The main objectives of the second half of the workshop were:

- Revising the benchmarking strategy for Y2.
- Looking at the different benchmarking dimensions: technical, business, societal.
- Focusing on the strategy technical system-wide (application) benchmarking.

The different elements of benchmarking were discussed in this second session of the workshop. The discussion was centred around the different strategies to assess MARVEL results from a system perspective. Some initial thoughts were highlighted since the beginning of the discussion:

- Staying at the component benchmarking level as done for the MVP could lead to inconsistent conclusions (i.e., an AI model optimisation component may show a decrease of latency if assessed alone, while assessed in the scope of a realistic application scenario may not show this decrease).
- Benchmarking real-world applications in the E2F2C is still in its infancy. There are some benchmarks already in place, but not so suitable for the purpose of MARVEL. Most of the application benchmarks provide workloads for testing specific implementations at different infrastructures/layers, rather than providing an assessment of a new application.
- Implementing new benchmarks is not an easy task, even if they are an extension of an existing benchmark (e.g., a new scenario in the AIBench Scenario benchmark). This normally implies the provision of datasets (i.e., a subset of MARVEL Corpus), different workloads, specific implementation of scenarios and pipelines, automation of the deployment for the community, etc. The effort to do so is very high and requires the engagement and support of the community behind the benchmark (e.g., the Bench Council for AIBench Scenario).

Several alternatives for benchmarking MARVEL were discussed, and their advantages and disadvantages are highlighted in the mind map shown in Figure 5.

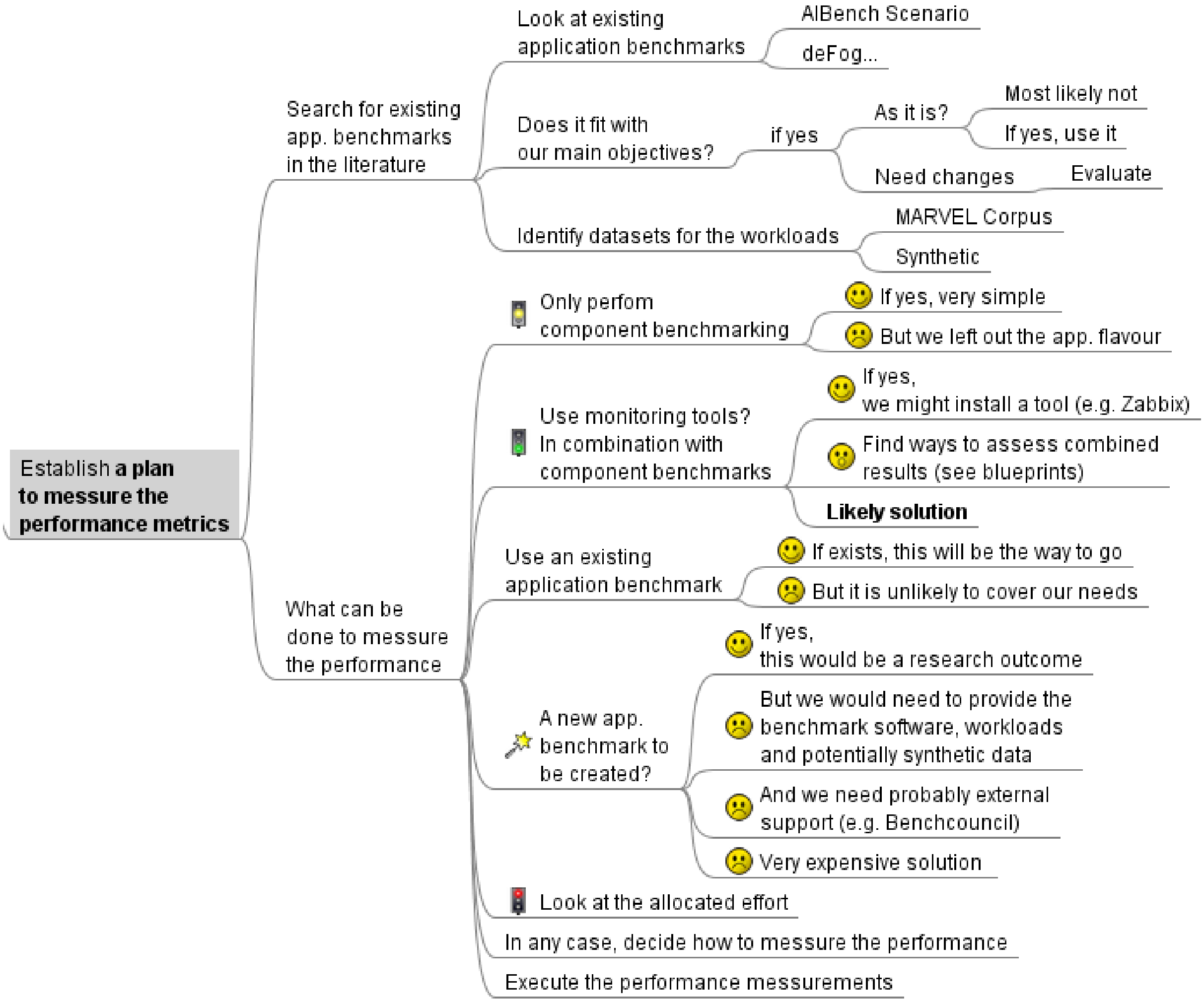
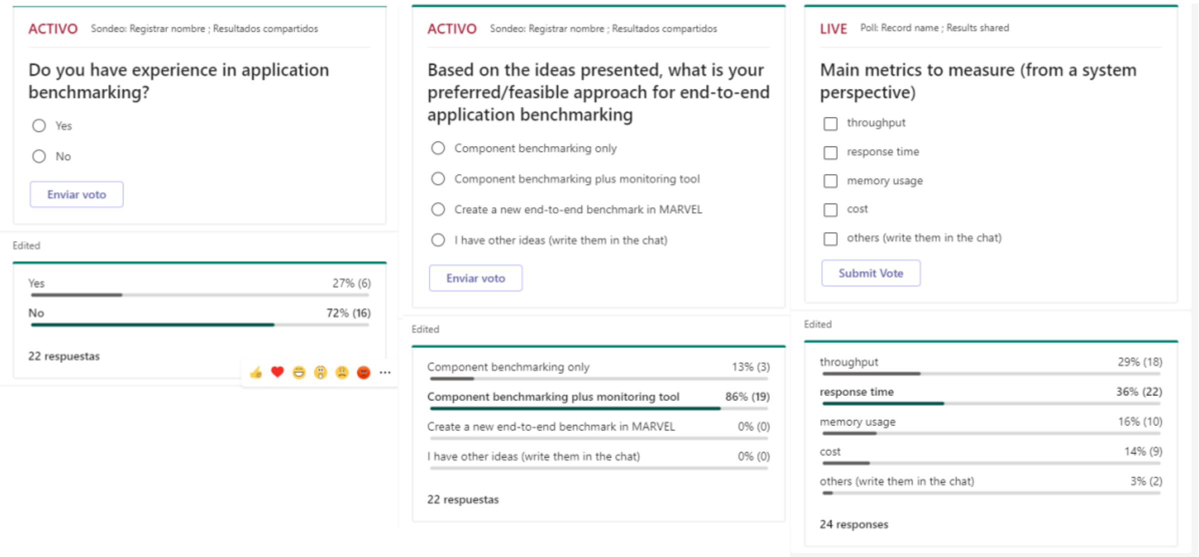

Establish a plan to messure the performance metrics
Search for existing app. benchmarks in the literature
Look at existing application benchmarks
AIBench Scenario
deFog...
Does it fit with our main objectives?
if yes
As it is?
Most likely not
If yes, use it
Need changes
Evaluate
Identify datasets for the workloads
MARVEL Corpus
Synthetic
What can be done to messure the performance
Only perfom component benchmarking
If yes, very simple
But we left out the app. flavour
Use monitoring tools? In combination with component benchmarks
If yes, we might install a tool (e.g. Zabbix)
Find ways to assess combined results (see blueprints)
Likely solution
Use an existing application benchmark
If exists, this will be the way to go
But it is unlikely to cover our needs
A new app. benchmark to be created?
If yes, this would be a research outcome
But we would need to provide the benchmark software, workloads and potentially synthetic data
And we need probably external support (e.g. Benchcouncil)
Very expensive solution
Look at the allocated effort
In any case, decide how to messure the performance
Execute the performance messurements

Figure 5. Ideas presented for system-wide application benchmarking during the workshop

The discussion led to the selection of the second option proposed to measure the performance, at least for the second year, without completely ruling out the last two options. The use of component benchmarking only was deemed insufficient, as expected, while the creation of a new benchmark, although being potentially a good research result, was considered probably too complex and beyond the scope of the project for the time being. Nevertheless, this decision will be revisited at the end of Y2 in the upcoming second benchmarking workshop expected after M24.

Figure 6 shows the results of the polls that were posed to the workshop attendees.


ACTIVO
Sondeo: Registrar nombre ; Resultados compartidos
Enviar voto
Edited
ACTIVO
Sondeo: Registrar nombre ; Resultados compartidos
Enviar voto
Edited
LIVE
Poll: Record name ; Results shared
Submit Vote
Edited

Do you have experience in application benchmarking?

○ Yes
○ No

Yes	27% (6)
No	72% (16)

22 respuestas

Based on the ideas presented, what is your preferred/feasible approach for end-to-end application benchmarking

○ Component benchmarking only
○ Component benchmarking plus monitoring tool
○ Create a new end-to-end benchmark in MARVEL
○ I have other ideas (write them in the chat)

Component benchmarking only	13% (3)
Component benchmarking plus monitoring tool	86% (19)
Create a new end-to-end benchmark in MARVEL	0% (0)
I have other ideas (write them in the chat)	0% (0)

22 respuestas

Main metrics to measure (from a system perspective)

□ throughput
□ response time
□ memory usage
□ cost
□ others (write them in the chat)

throughput	29% (18)
response time	36% (22)
memory usage	16% (10)
cost	14% (9)
others (write them in the chat)	3% (2)

24 responses

Figure 6. Results from the polls issued during the workshop

The results show that most of the partners do not have experience in application or system-wide end-to-end benchmarking, which makes harder the possibility of developing new benchmarks due to this lack of expertise. In fact, the second question shows evidence that the usage of component benchmarking in combination with some monitoring tools, to help assessing system-wide monitoring metrics, is the choice of 86% of the participants of the workshop. As for the metrics, the main one is related to assessing the response time of the complete system (26%), while measuring the throughput is second (36%) and memory usage third (16%), close to the assessment of costs (14%).

2.2.2. Initial benchmarking strategy

As explained in Section 2.1, deliverable D1.2 (MARVEL / D1.2, 2021) provided a summary of the benchmarks that might be useful from the micro (component) benchmarking. The work carried out so far follows closely the initial strategy consisting of applying micro (component) benchmarks for each of the MVP components, therefore assessing them in an isolated manner. Each component is performing its own assessment based on the State-of-the-art (SotA) perceived in its own domains. This is in line with the expectations of the MVP and it is the main content of this deliverable. The results can be checked in the following sections of the document. For this initial MVP benchmarking, an excel file has been provided internally as a living document used to collect the benchmarking results.

Besides the pure component benchmarks, D1.2 provided a list of widely known system or application-wide benchmarks that might be useful to assess the MARVEL pilot scenarios or at least some of its elements. Within this range are benchmarking suites such as YSBC[7], HiBench[8], defog[9], MLPerf[10], and AIBench[11], among others. However, most of these benchmarks are not suitable in the MARVEL context, as the AV data is not widely represented

[7] https://ycsb.site

[8] https://github.com/Intel-bigdata/HiBench

[9] https://github.com/qub-blesson/DeFog

[10] https://mlcommons.org

[11] https://www.benchcouncil.org/aibench

in the big data and AI perspectives in the benchmarking community. Of especial interest are benchmarks to assess AI and federated learning, such as AIBench, MLPerf, or LEAF[12], as well as benchmarking approaches followed in AIBench and MLPerf, some of them dealing with image and video data. Nevertheless, the matching of these benchmarks with the specific MARVEL usage scenarios and data make these difficult to apply or adapt for the MARVEL needs. With the view of the components and pilot scenarios to be implemented for M18, a review of these benchmarks will be carried out to revisit its functionality and assess its relevance and applicability to the MARVEL context.

Based on the results of the workshop, the strategy for Y2 can be summarised as follows:

- Until M18 several tasks will be performed:
    a. Establish objectives and added value of benchmarking for MARVEL (e.g., focusing on comparisons between executing algorithms and processing data in different layers of the computing continuum, rather than measuring everything).
    b. In parallel, decide on the main system-wide performance metrics to measure and how to address their potential dependencies.
    c. Investigate the scenarios in the pilot use cases for M18 and respective pipelines and blueprints provided in D1.3 (MARVEL / D1.3, 2021).
    d. Once the scenarios and components involved are clear, try to reduce the complexity still matching the objectives (e.g., by focusing on a set of components that affect more the overall performance).
    e. Select the benchmarking approach to follow:
        i. The main approach selected at the benchmarking workshop held in January 2022 (see 2.2.1) consists of performing component benchmarking plus monitoring of significant system-wide performance metrics.
        ii. Study alternative methods, such as the usage or adaptation of existing application benchmarks that may be fit for our needs. Especial attention will be paid to well-known benchmarking suites such as the abovementioned AIBench or MLPerf.
    f. Before M18 the supporting tools for performance assessment should be prepared and in place (e.g., the monitoring tools).
    g. Establish a roadmap to perform the measurements aligned with the work under WP6 related to the MARVEL pilots and to the delivery and integration of the components in T5.3.
    h. Test if the approach works (preliminary examples before M18).
- After M18:
    a. Start the measurement of system-wide metrics over the deployment of the scenarios delivered in M18, in several iterations, if needed.
    b. Collect measurements and interpret them.
    c. Reassess the strategy after the first iteration in the second benchmarking workshop (M25).
    d. Run the final iteration of the benchmarks (M26-M30).
    e. Draw conclusions (M28-M30).
    f. Report in D5.5 (M30).

This plan is the current view of the work to be done especially for Y2, but it will be revisited in the scope of the second benchmarking workshop at the beginning of Y3, as commented above.

[12] https://leaf.cmu.edu/

3. Sensing and perception

In this section, the sensing and perception subsystem of the MARVEL architecture for the MVP is discussed. The GRNEdge V1 portable device was used to record part of the dataset required for training the AI components for the MVP. It is the first version of a device that could be capable of processing at the edge in future versions. As a perception system, GRN also deployed the CATFlow algorithm capable of detecting and tracking vehicles in real-time and providing structured nonbinary data as output. The MEMS microphone provided by IFAG is used for audio data acquisition in the MVP. The device used most frequently is the AudioHub Nano with two connected microphones. The SED@Edge tool can be used to detect target sound events using low-resource devices.

3.1. GRNEdge

3.1.1. Role in the MVP

The GRNEdge V1 portable device was used to record part of the dataset required for training the AI components for the MVP. The data consisted of 3-5 minute audio-video snippets taken at various road locations in Malta. This data was annotated for the SED component and anonymised to be used for the demonstration of the Data Corpus.

However, since the GRNEdge V1 portable device was only able to save data at the edge, an off-the-shelf security camera with an integrated microphone was used for the E2F2C infrastructure. The video data from this camera was streamed to GRN's fog layer and was processed by the CATFlow algorithm. This security camera is a SIPC2MPDOMEM IC Safire 2MP Dome Outdoor/Indoor IP Camera with POE and a built-in microphone.

3.1.2. Current status

GRNEdge V1 portable device is complete. The device is able to record approximately 2 hours of continuous AV data that is suitable for the intended purpose. Future versions of the device could include adding the capability to stream the AV data, improving the hardware such as camera or microphone for better quality, or perhaps removing the video capability to create an audio node.

3.1.3. State-of-the-art baseline

A solution suitable for the purpose of having a portable GRNEdge setup would the use of a smartphone with camera and microphone. This method has been tested by GRN and used as a data collection method alongside the GRNEdge V1 portable device. The smartphone has greater ease of mobility and better quality of audio and video, especially in low light conditions. The drawback of using a smartphone is that the smartphones may use variable microphone models to record audio, which results in audio that is difficult for the AI providers to work with. Thus, an equivalent solution to the GRNEdge V1 portable device would be a smartphone with an external microphone.

3.1.4. Description of the benchmarking process/assessment strategy

3.1.4.1. *Assessment strategy*

The assessment method for the GRNEdge device V1 was to test the device before the deployment for metrics such as latency and synchronisation and then use field testing to test sustained performance in various weather conditions.

3.1.4.2. *Benchmarks used*

The benchmarking approach used was to find suitable key performance metrics and perform tests to confirm that the GRNEdge device is of adequate quality. The tests included taking videos and storing them to assess subjectively their quality.

3.1.4.3. *Infrastructure for testing*

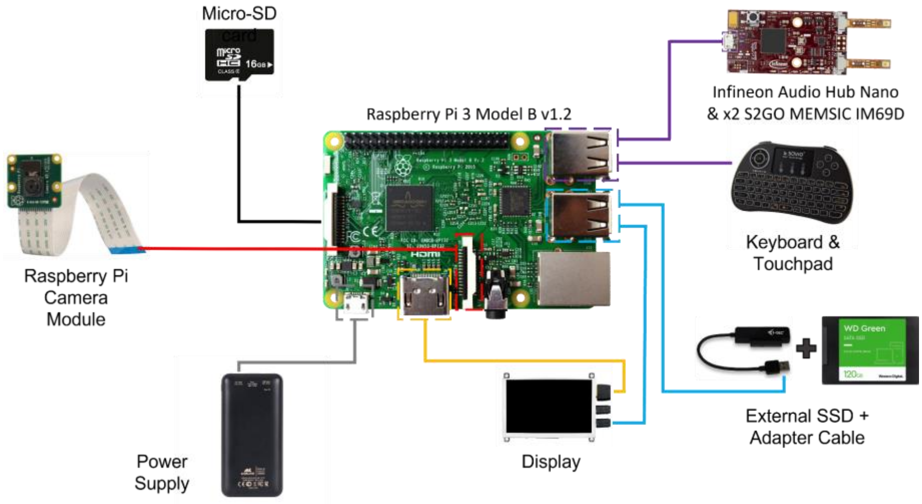


Figure 7. Diagram of GRNEdge V1 portable setup

The GRNEdge device consisted of a Raspberry Pi 3 Model B v1.2, a 16Gb micro-SD card, a Raspberry Pi Camera Module, a 20,000 mAh capacity power supply, display, external SSD and adapter cable, keyboard and touchpad and IFAG Audio Hub Nano microphone as shown in Figure 7. The device was placed in a transparent waterproof case to protect it from the weather conditions.

3.1.4.4. *Metrics*

The metrics for measuring the systems’ robustness are latency, drift, number of lost frames, audio-video synchronisation delay, and throughput, as well as the ability to sustain performance in various weather conditions. The reliability of the system is measured through the visualised data volume.

3.1.4.5. *Results of the measurements*

Latency: The latency was found to be negligible while the GPU of the Raspberry Pi is used.

Drift: To test the drift, a video[13] was played on a screen for 1 hour. This screen was then recorded by the GRNEdge device. The video shows a simple repetitive animation with sound, thus if the audio is drifted from the video it would be visible on the recorded video.

Number of lost frames: No compression or transmission is used for the GRNEdge device; thus, no frames will be lost.

[13] https://www.youtube.com/watch?v=ucZl6vQ_8Uo

Audio-video synchronisation delay: The delay was considered negligible while the GPU of the Raspberry Pi is used.

Throughput: To calculate the throughput three videos were taken as samples and the throughput was calculated as Throughput = data (Kb) / time (s).

Throughput_1 = 1852.5 Kb/s, Throughput_2 = 1971.7Kb/s, Throughput_3 = 2221.9 Kb/s.

Average Throughput = 2015.4Kb/s.

Sustain performance: The GRNEdge V1 device had enough memory and power to run for about 2 hours. The device was placed in a weatherproof case that is able to withstand rain and wind. The device was tested in real-world conditions during the month of August, where temperatures are 22-30^{o}C. These conditions, as well as the inability to have active cooling in the trasparent box made it difficult to use the device for periods of time longer than 15 minutes.

Visualised data volume: During recording, the temperature of the GRNEdge device appeared to reach a constant 70 - 72 °C after only about 15 minutes of capturing. Temperatures above 72°C are not ideal for more than a few minutes, however, temperatures above 80°C are considered harmful. After periods in the sun, the temperature still reached 78°C thus prolonged use was not feasible. Based on previous facts, the data acquired for short period could be reliably captured and visualised.

3.1.4.6. *Results observations*

The GRNEdge V1 portable device performs as intended, recording video for a short time. However, the difficulties with overheating due to the weatherproofing case used to protect the device make it improbable that a similar setup can be used for long-term data collection.

3.1.5. Contribution to MARVEL KPIs

The component contributes to the following KPIs:

KPI-O1-E1-2: Increase of data throughput and access latency by 10%. Later versions of the GRNEdge will contribute towards this KPI.

KPI-O5-E1-1: More than 3.3PB of data made available through a Corpus-as-a-Service. The GRNEdge V1 partially contributed to this KPI since the GRNEdge device produced training data that will be released in the Data Corpus after being properly anonymised.

3.1.6. Expected future results

GRN expects the GRNEdge device to have various different versions according to the needs of the use cases. It is probable that future versions of the GRNEdge device would incorporate more off-the-shelf devices to increase sustained performance in different weather conditions. For example, the GRNEdge surveillance camera devices used in the E2F2G framework for the MVP could be included to increase weather resistance. Altered versions of the GRNEdge device to capture only audio are also possible. These audio nodes could integrate GPU in the device as well to increase quality and reduce latency.

3.2. CATFlow

3.2.1. Role in the MVP

The CATFlow algorithm is used in Scenario 1 – Identification of vehicles and trajectories of the MVP. Video data from the static Mgarr camera is streamed to the GRN fog layer where it is processed by the CATFlow algorithm. The CATFlow algorithm detects the different types of vehicles and tracks them. Thus, the output is structured nonbinary data that contains information such as entry and exit lanes, trajectories, speed information, and time of entry and exit. This data is then sent as a Kafka message to be used in the DMP.

3.2.2. Current status

The CATFlow algorithm is functional and is currently being optimised and maintained to keep improving its performance.

3.2.3. State-of-the-art baseline

The CATFlow algorithm makes use of a purposely trained YOLOv4 model (Bochkovskiy, et al., 2020) which is a SotA Object Detector that can perform object detection in real-time with good accuracy. YOLOv5 and other expansions using the YOLO framework also exist. However, it has not yet been integrated into OpenCV (Bradski, 2000) and there is not enough research to support it. In the case of tracking, the trackers tested are from OpenCV. Novel standalone trackers exist but OpenCV is a regularly updated library with generally good tools. Research on trackers is currently ongoing.

3.2.4. Description of the benchmarking process/assessment strategy

3.2.4.1. *Assessment strategy*

Using the open-source MSCOCO dataset (Lin, et al., 2014) the accuracy of the CATFlow algorithm can be assessed in terms of object detection. The open-source datasets are annotated, thus they can be used as benchmarks. The precision and recall are calculated using an (Intersection over Union) IOU (Rezatofighi, et al., 2019) threshold. Then the Mean Average Precision (mAp) is calculated to give a better representation of the model's accuracy.

3.2.4.2. *Benchmarks used*

MSCOCO (Lin, et al., 2014) and MIO-TCD (Luo, et al., 2018) datasets were used to benchmark the CATFlow algorithm in terms of object detection.

3.2.4.3. *Infrastructure for testing*

The CATFlow algorithm was tested on the GRN fog infrastructure. The server on the fog layer has the following specifications:

- Server: HPE ProLiant DL385 Gen10 Plus
- CPU: AMD EPYC 7302 16-Core Processor
- RAM: currently 32GB using 16GB sticks
- GPU: Nvidia Tesla T4 16GB
- Storage: HPE 480GB SATA

3.2.4.4. *Metrics*

The following metrics were measured to determine the quality of the object detection: precision and recall using the IOU threshold, F-score and mAp. Latency of the model is also considered. These metrics will be used to measure the performance of the CATFlow algorithm.

3.2.4.5. *Results of the measurements*

Figure 8 shows the precision, recall and F-score for each confidence threshold.

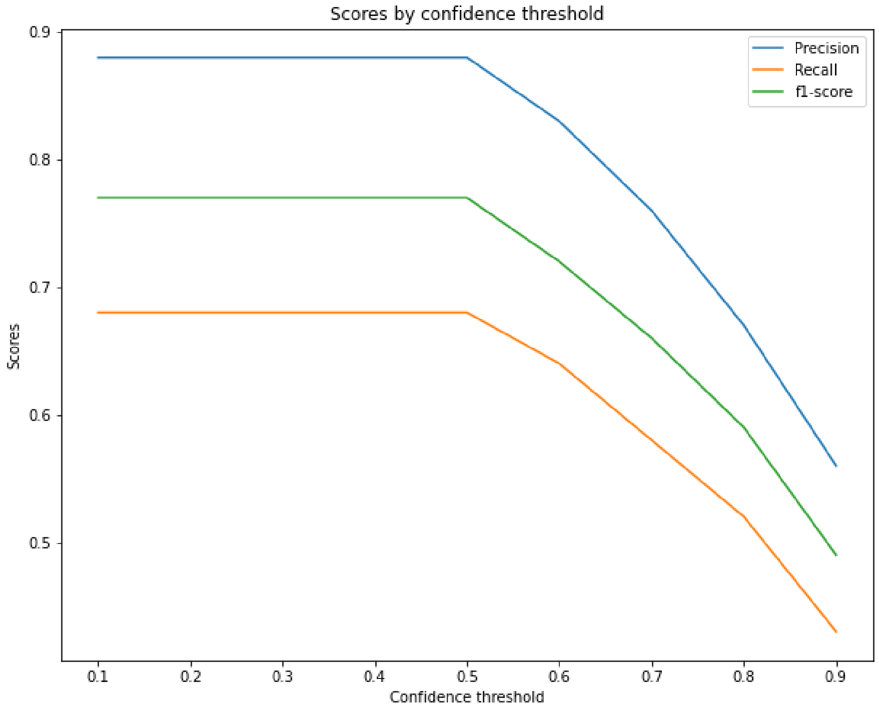


Figure 8. Precision, recall and F-score for each confidence threshold

The mAp value for each class can be viewed in Table 1.

Table 1. Results of the measurements for CATFlow component, mAp values for each class

Class	mAp
pedestrian	0.5517
car	0.7225
bus	0.8048
light goods vehicle	0.7454
heavy goods vehicle	0.7010
motorcycle	0.6593
bicycle	0.6011
average mAp for each class	**0.6927**

Latency: the CATFlow algorithm is able to work at a rate faster than real-time, thus no latency is observed.

3.2.4.6. *Results observations*

Through these results, one can see that object detection of the CATFlow algorithm works well. One can also note the lowest mAp value is observed when detecting pedestrians which signifies that more effort needs to be placed in that area to improve the object detection.

3.2.5. Contribution to MARVEL KPIs

This component does not contribute to any KPIs.

3.2.6. Expected future results

In the future, GRN aims to improve pedestrian, bicycle, and motorcycle detection in the CATFlow model to improve the overall quality of the model. Metrics to evaluate the tracking algorithms are also required to be include in the benchmarking in the future.

3.3. MEMS

3.3.1. Role in the MVP

The hardware provided by IFAG is used for audio data acquisition in the MVP. The most used device is the AudioHub Nano with two connected microphones.

3.3.2. Current status

IFAG has provided the following MEMS-based boards:

- EVAL_IM69D130_FLEXKIT
- IM69D130 Microphone Shield2Go
- IFAG My IoT adapter for Raspberry Pi
- IFAG My IoT adapter for Arduino
- IFAG Audiohub – Nano (2 microphones)
- Audiohub – Nano (4 microphones)
- XMOS XK-USB-MIC-UF216 (7 microphones)

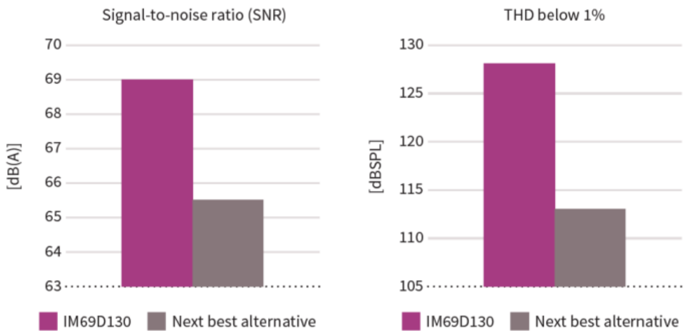
IFAG is currently working on the Dual-PCB 8-microphones Audiohub – Nano. It will be finished in the second quarter of 2022 and as a novelty, it will provide edge processing capabilities directly on the microphone board for cloud offloading.

3.3.3. State-of-the-art baseline

IFAG is a world leader in MEMS microphone technology. The devices described above use the high-end microphone IM69D130. It is known for the following features:

- low self-noise (SNR)
- wide dynamic range
- low distortions
- high acoustic overload point

The microphone uses a dual-backplate technology which allows for those remarkable features. Figure 9 and Figure 10 show a comparison of the IM69D130 to alternative MEMS microphones on the market. In particular, the Total Harmonic Distortion (THD) is kept very low even if exposed to loud audio signal, which allows for a very high-quality recording of the spectral component of the audio data, enabling accurate data analytics, even in high noise and very loud environments.



Clear far field and low volume audio pick-up

Below 1 percent distortion even if exposed to loud audio signals

Figure 9. Comparison of IM69D130 to alternative MEMS microphones

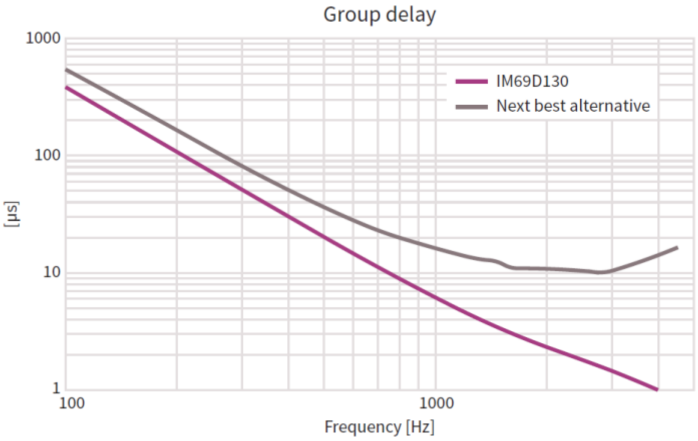


Figure 10. Comparison of IM69D130 to alternative MEMS microphones

3.3.4. Description of the benchmarking process/assessment strategy

3.3.4.1. *Assessment strategy*

For the MVP, the benchmark is to create initial use cases and to carry out initial data acquisition with the dual microphones MEMS boards.

3.3.4.2. *Benchmarks used*

Up to now, IFAG’s audio data acquisition devices are in initial use. Data can be acquired, but some of the data is not yet of the expected quality, for example, caused by surrounding factors like the physical casing around the device. The future goal is to acquire data with higher precision and a larger number of microphones.

3.3.4.3. *Infrastructure for testing*

Up to now, data pre-processing is done on the Audio-Hub-Nano such that the audio stream can be connected to an edge device via USB. Further processing is done on a Raspberry Pi.

In the future, more processing will be possible on the edge node since a more powerful processing unit will be connected to the microphones. Furthermore, data streaming via WiFi will be enabled which allows sending the data directly to the MARVEL’s fog.

3.3.4.4. *Metrics*

The desired metrics are the SNR of the microphones and the quality of the audio samples in a real and noisy outdoor environment.

3.3.4.5. *Results of the measurements*

Impact of the casing in the SNR has been assessed, including suggestions to improve it by leaving a better opening in the casing. As a result, future boards will have more microphones (up to eight) and further processing capabilities to improve the quality of the signal directly at the edge, using the low-distortion microphones characterised in Figure 9 and Figure 10.

3.3.4.6. *Results observations*

Up to now, the devices are in initial use for data acquisition in connection with a Raspberry Pi. Recordings were performed in the scope of the drone experiment and in open space experiments in places (the Campus of Novi Sad) and on the road. The casing of the devices was a problem since it muffled the sound, but this problem can be fixed by using a different casing.

3.3.5. Contribution to MARVEL KPIs

IFAG's components contribute to the following KPIs:

KPI-O5-E1-1: More than 3.3PB of data made available through a Corpus-as-a-Service. The components described above contribute to the KPIs by allowing for innovative audio data acquisition and thereby to the composition of the MARVEL datasets.

KPI-O1-E1-1: Different kind of resources to be discoverable: ≥ 3. The data acquired allows for a handy connection of MEMS microphones to the data acquisition framework.

KPI-O4-E1-1 - More than 10 trial cases to showcase framework's capabilities**, KPI-O3-E4-1** - Detailed insights to more than 5 hidden correlations**,** and **KPI-O4-E2-1** - Identify at least 20 dependent and independent verification and validation variables for the system: Since the AudioHub Nano with two microphones is already in use in several locations for data acquisition, it contributes to showcasing the framework's capabilities, correlations in the data and verification and validation of the variables for the system.

KPI-O3-E4-1: Detailed insights to more than 5 hidden correlations. With the 8-microphones board under development, it will be possible to use innovative data processing on the edge.

KPI-O5-E3-1: More than 5 SMEs used in the Corpus. The data acquired with IFAG's devices will enable further collaborations with SMEs.

3.3.6. Expected future results

The Dual-PCB 8-microphones Audiohub – Nano, currently under development will feature 8 microphones, a strong processing unit (PSoC 6) and data streaming via WiFi. More processing will be feasible directly on the network edge node. Even ML models can run on the PSoC 6. The higher number of microphones will allow for more precise measurements and better sound source localisation. Custom firmware will allow for very high, very fast data throughput.

3.4. SED@Edge

3.4.1. Role in the MVP

The goal of the tool is to detect target sound events using low-resource devices. Although this component is not used in the MVP because such edge devices are not installed yet, it can be used to detect different types of vehicles.

3.4.2. Current status

The tool has been improved with respect to the original model based on distillation and it now implements PhiNets architectures (Paissan, et al., 2021) operating both on the spectrograms and the waveforms. The current implementation is in PyTorch, using the lightning module, and runs on a Linux workstation both for training and inference.

3.4.3. State-of-the-art baseline

The peculiarity of the SED@Edge tool is the fact that it requires few computational and energy resources. Typical SotA baseline does not consider these limitations. For the purpose of our work, we need to consider two types of baselines:

- Unconstrained: to assess our KPI on limited performance reduction when reducing the computational power.
- Constrained by the device limitations.

3.4.4. Description of the benchmarking process/assessment strategy

3.4.4.1. *Assessment strategy*

In this iteration, we want to evaluate the performance of the current SED@Edge tool to detect areas of improvement in terms of training and architecture. The tool is evaluated in an offline fashion using pre-recorded audio files. The goal is to evaluate the performance in comparison with SotA unconstrained approaches but also to understand if we can develop a solution suitable for the MARVEL edge platforms with limited performance deterioration.

3.4.4.2. *Benchmarks used*

The tool has been evaluated using publicly available datasets with urban and traffic events: UrbanSound8K (Salamon, et al., 2014) and MAVD (Zinemanas, et al., 2019). For UrbanSound8K we adopted the official cross-fold validation of the dataset. Further results are available in (Paissan, et al., 2022). For MAVD, we considered only vehicle classification (4 classes: car, truck, bus, motorcycle) on 1-sec segments. We considered a single class task (i.e., no overlapping events).

In addition, we also performed a very preliminary analysis using part of the first set of data with traffic sound released by GRN for the MVP deployment. We considered the audio files recorded with GRNEdge and we used the annotation labelled as “Leonel”. Overall, 26 recordings are available. Similarly to MAVD, we considered only vehicle classification (car, bicycle, bus, heavygoodsvehicle, lightgoodsvehicle, micromobility, motorcycle) and we split the recordings into 1-sec segments.

3.4.4.3. *Infrastructure for testing*

The current evaluation is performed on a workstation with a GPU (a Dell precision 7920 server with 3 RTX5000 NVIDIA GPU). Note that the evaluation involves also the training stage which does not have to be at the edge. Note also that the current configuration of the evaluation platform is well above what would be required for real-time processing.

3.4.4.4. *Metrics*

Given that the tool is designed to operate on edge devices, its benchmarking must also consider aspects related to computational requirements and power consumption, which are, of course, device-dependent. To define the size of the model, we considered as target edge device an STM32 micro-controller. As metrics, we consider the event classification accuracy (comparable with the SotA), the number of multi-accumulation (MAC) operation and the memory needed to store the model weights. Only the former depends on the datasets, while the two latter are characteristics of the algorithm. We report the averaged results on 10-fold cross-validation for US8K and GRN and on 10 runs for MAVD.

3.4.4.5. *Results of the measurements*

The results of the first evaluation are reported in Table 2. The number after PhiNets refers to the number of parameters that slightly depends on the number of output classes (and hence on the dataset). Note that we used MAVD in a different way from what is typically done in literature (i.e., single class problem) so a direct comparison is not possible. For the GRN data, we also report the standard deviation of the accuracy on the 10-fold as we observed a very high variability.

Table 2. Results of the measurements for SED@Edge component

Dataset	Method	Accuracy (%)	MMAC	Memory
MAVD	PhiNets 19.9K	67%	37.02	20KB
UrbanSound8K	AudioClip	90%	-	-
	PhiNets 27.1K	76.3%	43.00	27KB
GRN	PhiNets 19.9K	50% (24%)	37.02	20KB

3.4.4.6. *Results observations*

The current architecture is suitable for deployment on very small edge devices and would definitely fit on the Raspberry PI 4B that will be deployed in the MARVEL infrastructure. Although at the moment we do not have a direct comparison with a fair upper bound, the performance on the current benchmarks is below the target KPIs, which, for this component, requires a 5% max drop with respect to the current SotA upper-bound on RPi, and 10% drop on MCU.

Note however that AudioClip is a very challenging upper bound which would require very high computational resources for online inference. Nevertheless, the component is only 4-5 % point shorter on UrbanSound8K and there is a margin for improvement.

3.4.5. Contribution to MARVEL KPIs

The component contributes to the following KPIs:

KPI-O3-E2-1: model compression algorithms to achieve 70% compression rates, without a noticeable degradation of accuracy. The current architectures achieve a higher compression but performance is still a little lower than the baseline in some benchmarks.

KPI-O3-E2-2: optimise performance (prediction accuracy, time-to-decision) of DL deployment by 20%. The current tool allows increasing the time-to-decision by operating directly on edge devices.

KPI-O3-E2-3: increase accuracy levels of real-time observations at the edge devices by 20%. The current architecture improves with respect to our previous edge approach.

3.4.6. Expected future results

Currently, the performance of SED@Edge on the selected benchmarks is a bit low in terms of accuracy, although all other constraints in terms of resources are met. We expect that by further optimising the architecture and improving the training protocol we will be able to achieve the project goals. Future results will investigate a more detailed comparison with SotA methods in order to measure the related KPIs.

4. Security, Privacy, and data protection

Privacy preservation in visual data is often achieved by means of redaction techniques, e.g., obfuscation on the personal data. Classical anonymisation techniques, such as blurring for faces can successfully remove Personal Data. Nevertheless, this comes at a high cost of deteriorating further audio-visual analysis. Advances in Generative Adversarial Networks (GAN) have allowed proposing different GAN-based video anonymisation solutions by swapping the original face with natural-looking faces of another identity with preserved facial pose and expression. VideoAnony component serves the video anonymisation in the MARVEL framework with the ambition to develop GAN-based face swapping techniques addressing challenges in the Closed-Circuit Television (CCTV) footage.

4.1. VideoAnony

4.1.1. Role in the MVP

The first version of VideoAnnoy is incorporated in the MVP to anonymise the pilot data, which is then stored in the Data Corpus of MARVEL. The current anonymisation module first detects car plates and human faces and then blurs them to remove any identifiable attributes.

4.1.2. Current status

The deployed version of the VideoAnony at the time of MVP is in its first version, where the anonymisation is achieved by the classic blurring technique. In this version, the most challenging part is to detect the car plates and human faces in the CCTV footage, which suffers from domain shift when the testing scenarios differ from the one where the detector is trained. Due to the limited amount of available annotated pilot data, we mitigate the domain shift by exploiting existing public datasets that resemble crowded public places.

In parallel, we are developing more advanced GAN-based face-swapping techniques that can de-identify the detected person while maintaining high image fidelity and decent pose-preserving capability in challenging CCTV footage.

4.1.3. State-of-the-art baseline

The SotA methods, e.g., CIAGAN (Maximov, et al., 2020), SimSwap (Chen, et al., 2020), FaceShifter (Li, et al., 2020), and HiFiFace (Wang, et al., 2021) mostly focus on swapping faces in centred and frontal-faced visual content, whose faces are presented in high-quality images, such as CelebA-HQ (Karras, et al., 2018) and FFHQ (Kazemi & Sullivan, 2014). In general, with the provided models, most of the methods can generate natural and pose-preserving faces of the specified identity with the high-quality face images, apart from CIAGAN which tends to produce less natural faces. However, for CCTV videos, faces are often captured with varying resolutions due to their varying distances towards the camera with some extreme poses. When applying SotA methods on faces with varying resolutions, their performance often decreases.

4.1.4. Description of the benchmarking process/assessment strategy

4.1.4.1. *Assessment strategy*

We plan to benchmark recent GAN-based face-swapping methods, including CIAGAN, SimSwap, FaceShifter, and HiFiFace, on a new dataset we create on top of CelebA-HQ in order to study how the SotA methods perform on faces with varying resolutions, in terms of the face naturalness, face detection, face identification and pose preservation performances.

4.1.4.2. *Benchmarks used*

We created an extended dataset on top of CelebA-HQ by downsampling the original high-quality images to a set of different resolutions, including 512 by 512, 256 by 256, 128 by 128, 64 by 64 and 32 by 32. We follow the same train/test split of the original images. When benchmarking different face swapping methods, we evaluate the generated faces in terms of face naturalness, face detection, face identification and pose preservation by inputting the target image of different resolutions, i.e., the image containing the face we aim to swap, while the source image, i.e., the image containing the face of the new identity to be swapped, can be the original high-resolution image.

4.1.4.3. *Infrastructure for testing*

We use a server equipped with an RTX 3080 GPU for the evaluation of all the methods. The setting for the current stage of the evaluation is appropriate. Note that for the later-on stage when we evaluate the performance of our component VideoAnony on the edge device, such a server will not be optimal anymore, while an embedded GPU board, such as NVIDIA Jetson Nano or NVIDIA Xavier NX could be exploited.

4.1.4.4. *Metrics*

We will measure the *face naturalness* performance in terms of Fréchet Inception Distance (FID) score (Heusel, et al., 2017), which compares the distribution of generated images with the distribution of real images (the lower, the better), the *face detection* performance in terms of percentage of faces being detected after swapping, the *face identification* performance in terms of the percentage of swapped faces being matched to its original face, and the *pose preservation* performances in terms of the normalised mean error (NME) of the detected landmarks.

4.1.4.5. *Results of the measurements*

The numerical results are shown in Table 3. We presented the classic blurring for anonymisation, together with the above-listed SotA face-swapping methods when evaluated with input target images of different resolutions. Our VideoAnony component is based on the architecture of FaceShifter with a re-training with our extended dataset.

Table 3. Results of the measurements for VideoAnony component

Method	FID ↓	Face Detection (dlib/Facenet) (%) ↑	Pose NME (%) ↓	Re-Identification ↓
Blurring	153.76	10/86	-	-
SimSwap (128x128)	26.59	99/100	3.42	0.224±0.140
SimSwap (64x64)	37.10	98/100	3.45	0.126±0.099
SimSwap (32x32)	83.21	95/100	4.22	0.301±0.186
CIAGAN (128x128)	96.88	99/100	9.25	0.097±0.097
CIAGAN (64x64)	100.98	76/99	9.21	0.108±0.095
HiFiFace (256x256)	15.48	99/100	5.17	0.085±0.143
HiFiFace (64x64)	44.22	98/100	5.21	0.085±0.104
HiFiFace (32x32)	111.76	96/100	5.88	0.089±0.087
FaceShifter (256x256)	16.28	98/100	2.87	0.523±0.254
FaceShifter (64x64)	52.86	97/99	3.38	0.368±0.225
FaceShifter (32x32)	127.61	94/99	4.78	0.240±0.142
VideoAnony (256x256)	27.22	100/100	4.16	0.284±0.198
VideoAnony (64x64)	28.38	99/100	4.44	0.224±0.163
VideoAnony (32x32)	83.04	93/100	5.31	0.366±0.135

4.1.4.6. *Results observations*

In general, by checking the quantitative results, blurring has the worst FID performance, and it also discourages meaningful analysis in pose preservation and re-identification. Current SotA methods with the pre-trained models on high-quality images suffer from generating natural swapped faces when the image resolution decreases, leading to an increased FID score, which results in inferior performances in face detection and pose preservation. In general, they can maintain the identity swapping capability even with a low-resolution input face, however, its swapped faces tend to preserve fine details which are not natural for their respective resolution, leading to a high FID score. Moreover, the pre-trained model of SimSwap seems to suffer the least performance deterioration among all methods in terms of the face generation naturalness; however, it tends to not perform face-swapping when the input source faces are at low resolutions, which does not fit our purpose. Our VideoAnony is based on the FaceShifter architecture and it can achieve comparable numerical results compared to SimSwap, while being able to actually swap faces even when fed with low-resolution source faces.

4.1.5. Contribution to MARVEL KPIs

KPI-O1-E3-3: This KPI aims to reduce 10% the computational complexity for video anonymisation. At the moment of MVP, we are still progressing to achieve a model that can handle face-swapping for anonymising CCTV videos. Once the model is mature, we will further reduce its computational complexity, in order to realise this MARVEL KPI.

4.1.6. Expected future results

We expect to achieve a face-swapping model that can achieve improved image naturalness and de-identification performance while maintaining the poses regardless of the face resolution. For the long-term, we will work on the model reduction, realising the KPI requirement, towards running the component on the edge device.

5. Data Management and Distribution

For the MVP version, the subset of the Data Management and Distribution subsystem that was implanted consists of two components: a) the Data Fusion Bus (DFB) and b) DatAna. In the context of Use case Scenarios 1 and 2, DatAna was used to retrieve data from the fog layer when they were available, more specifically, the results of CATFlow from a Kafka service on the GRN cloud infrastructure, and the results of SED and AVCC from a shared folder where these two components stored their inference results. DatAna streamed all collected data to DFB's Kafka service. Then DFB indexed those data into its Elasticsearch instance. Those stored data were then made available to MARVEL's UI component.

5.1. DFB

5.1.1. Role in the MVP

The Data Fusion Bus (DFB) is a customisable component that implements a trustworthy way of transferring large volumes of heterogeneous data between several connected components and the permanent storage. The key modules of DFB are:

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

DFB is one of the two core components of the Data Management Platform (DMP) for the MVP. It interfaces with the other core DMP component, DatAna within the context of Scenario 1 and Scenario 2. More specifically, after DataAna has collected inference results or other metadata from other layers of the MARVEL architecture, it streams these data to DFB's Kafka, using a predefined JSON data format and a specific Kafka topic for each data producing component, namely CATFlow, AVCC, and SED.

Internally, DFB maintains the content of these topics for a limited time, as the purpose of the Kafka topics is to provide them to consumers in real-time. Upon receiving a data stream, DFB maps this information to Elasticsearch indexes, so that data may be later available for querying and filtering.

The main consumer for data stored in DFB is the SmartViz component. For the scenarios selected for demonstration, SmartViz accesses the stored data by querying DFB's Elasticserach. In the upcoming version, SmartViz will also subscribe to Kafka topics for real-time visualisations.

5.1.2. Current status

For the MVP version of DFB, the included and tested modules are a) the Kafka instance, b) the ES-connector, and c) the Elasticsearch index. The modules that were not necessary and therefore not included are: a) the DFB Core & UI, and b) the Keycloak authenticator. For future releases, we will examine if these modules are relevant to the needs of the implemented use cases.

For this actual version, the included DFB modules are shipped with their full functionality. In terms of deployment, we have setup DFB in a minimum configuration, using 1-3 cluster nodes, for isolated testing. In future releases, we will deploy DFB in a more complex configuration to cover the increasing data volumes to be handled.

5.1.3. State-of-the-art baseline

DFB is a custom, distributed solution for handling and storing large amounts of input data streams. Because of the nature of this solution, there are no standard baseline measurements to compare against.

5.1.4. Description of the benchmarking process/assessment strategy

5.1.4.1. *Assessment strategy*

As mentioned in the previous subsection, there is no standard baseline to assess the performance of a custom distributed solution such as DFB. Our approach to assessing DFB's performance is based on comparative measurement of the evolution of DFB within the MARVEL project against a DFB isolated, minimum deployment. This way we can assess the "out-of-the-box" performance of this component that is used in the context of the MVP and use this measurement as a reference for testing DFB with large amounts of data in more complex deployment configurations.

5.1.4.2. *Benchmarks used*

As it is typical to performance evaluation of distributed solutions, we have created synthetic data that was streamed to DFB. These data were generated from actual sample data used in the MVP demonstrators, recorded from cameras and microphones at the edge level of the GRN infrastructure. We followed the same JSON data structure and modified the size of the input data by generating the same structure with varying, dummy data.

5.1.4.3. *Infrastructure for testing*

All tests were performed on a DFP deployment on 3 VMs with 4-core CPUs and 32 GB of data. Input test data were streamed from a distinct network node, in order to simulate normal network connectivity.

5.1.4.4. *Metrics*

Overall, in order to assess the contribution of DFB to the MARVEL framework, we need to address the following high-level performance indicators: a) Data Integrity, b) Scalability, c) Availability, and d) Performance for high volume, heterogeneous data streams. Below, a list of specific, measurable metrics is associated with the above indicators, along with a brief description of the purpose of each indicator:

- Data Integrity
  - **Metrics**: Data loss rate
  - **Description**: Confirm that advanced encryption mechanisms over end-to-end data transfer will guarantee data integrity.
- Scalability
  - **Metrics**: HW speed up
  - **Description**: Increase the number of modality data streams and verify that performance metrics improve or at least stay the same
- Availability
  - **Metrics**: Service availability-failed request, data access restriction

  - **Description**: Verify that DFB resources are available and discoverable
- Performance for high volume, heterogeneous data streams
  - **Metrics**: Data transfer latency, data throughput, response time, number of cluster nodes

Description: Thoroughly measure different performance metrics under different execution conditions

5.1.4.5. *Results of the measurements*

Table 4 summarises the collected measurements for the specified metrics.

Table 4. Results of the measurements for DFB component

Metric	Value
Data loss rate	0
HW speed up	-
Service availability-failed request	100% availability
Data access restriction	None
Data transfer latency	5 ms (200 MB/s load)
Data throughput	605 MB/s
Response time	5 ms (200 MB/s load)
Number of cluster nodes	3

5.1.4.6. *Results observations*

At this initial stage, DFB performs as expected. The modules that comprise DFB have been developed with a rigorous engineering process and tested in isolation. Overall, DFB is high performant in terms of availability, speed, and resilience, in the proof-of-concept context of the MVP. This evaluation will be used as a baseline to evaluate DFB’s performance on later releases when data and infrastructure will scale significantly.

5.1.5. Contribution to MARVEL KPIs

The KPIs affected by DFB’s performance are listed below, accompanied by a comment on the way that each KPI is addressed by the MVP version of DFB.

KPI-O1-E1-2: Increase of data throughput and access latency by 10%. Initial measurements of data throughput, data transfer latency, and response time directly contribute to this KPI.

KPI-O1-E2-2: Increase the number of different modality data streams that can be handled by 30%. In the MVP, DFB contributes to the management of data streams. In future releases, the volume of data streams will grow significantly. In that direction, all measured metrics in this evaluation are relevant to the efficiency of the handling of those data, as well as the infrastructure that supports data management.

KPI-O1-E2-3: Increase the speed of the fusion process by at least 20%. This KPI is directly related to DFB, as one of its core capabilities is to gather and fuse diverse data streams for various sources. All the above-mentioned metrics are relevant to the speed of the fusion process.

5.1.6. Expected future results

In the upcoming releases of the MARVEL framework, the scale and scope of the implemented solution will grow significantly, both in terms of implemented use cases, and the amount of data to be handled. As DFB is a high-performance, scalable component, its added value will show in those releases. To that end, we should make sure that DFB is performant and responds well to an increasing number of connected components and data volume. The above-mentioned metrics will be re-evaluated and closely monitored to meet the desired goals, with respect to data management.

5.2. DatAna

5.2.1. Role in the MVP

DatAna is an Apache NiFi-based Data Management Platform that allows the creation of data processing pipelines graphically. In the scope of the MVP, DatAna plays a role in scenarios 1 and 2.

- Scenario 1: DatAna takes the outputs of the CATFlow component located in a remote Kafka in the GRN fog layer, and moves them to the MARVEL cloud, to the Kafka service provided by the DFB.
- Scenario 2: Similarly, DatAna provides two data flows to gather the results of the two analytical models (SED and AVCC), processes these data to match with agreed data models, and pass the resulting data to Kafka provided by the DFB.

5.2.2. Current status

DatAna has been deployed as a service in MARVdash as a simple Apache NiFi 1.12 version in the cloud. In order to do that, a dockerized version of NiFi has been provided as well as a specific Helm Chart[14] to help MARVdash to prepare the deployment of the service.

An instance NiFi service has been deployed in the MARVEL cloud that handles the data flows created for the two scenarios of the MARVEL MVP.

5.2.3. State-of-the-art baseline

There are many potential strategies for data processing. Apache NiFi is one of the SotA solutions to serve as middleware to perform data management and processing, offering many off-the-shelf processors to connect with data sources as well as extract and transform and load data in other systems using a friendly graphical user interface which in many cases allows to create data flows with no coding involved. Besides, the NiFi ecosystem provides ways to deploy MiNiFi agents at the edge making data processing a possibility in the E2F2C continuum. These, along with some other technical features (backpressure, load balancing, high reliability, etc.) make NiFi a very good solution for MARVEL.

Regarding benchmarking NiFi-based systems, there are not so many examples in the literature. Some vendors provide measurements made in specific settings. This is for instance the case of Cloudera[15], which reports how NiFi behaves in terms of scalability and performance (data rates) using very demanding workloads.

[14] https://github.com/cetic/helm-nifi

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

5.2.4. Description of the benchmarking process/assessment strategy

5.2.4.1. *Assessment strategy*

The setting of the MVP for DatAna is quite limited, with a single NiFi in a very limited infrastructure chosen on purpose to test how NiFi behaves in a very minimal environment. Therefore, the assessment strategy for the MVP is also quite limited, focusing mainly on assessing if the current installation is enough to handle the data provided in the project. The scalability is therefore no subject of the assessment in the MVP setting.

5.2.4.2. *Benchmarks used*

No specific benchmarks were used. The data workloads are MARVEL data used in the MVP.

5.2.4.3. *Infrastructure for testing*

An instance NiFi service has been deployed in MARVdash and it is being used for the testing of the MVP. This instance is running under minimal resources, to assess their validity for the MVP:

- 1 single node for Apache NiFi v1.12.1 in Linux.
- Deployed in MARVdash as a simple NiFi service[16]
- 1 core
- 2GB Java heap space
- Internal repositories located inside the docker deployment directories
- Access to the GUI is protected by username and password

5.2.4.4. *Metrics*

As mentioned, scalability cannot be measured in this setting with a single node of NiFi. The main metric is therefore related to the latency and throughput, in particular, the data processing rates.

5.2.4.5. *Results of the measurements*

Table 5 shows the results for NiFi in a similar manner to the DFB shown in Table 4. Figure 11 shows the results for scenario 1 and Figure 12 shows the results for scenario 2.

Table 5. Results of the measurements for DatAna component

Metric	Value
Data loss rate	0
Service availability-failed request	100% availability
Data access restriction	None
Data transfer latency	-
Data throughput	Scenario 1: 0,54 MB/s (25,6 KB for 234 Kafka entries) Scenario 2: 1,1 MB/s (11 files, total of 76,6KB from SED data, corresponding to 1731 Kafka entries)
Response time	Scenario 1: 47,1 ms Scenario 2: 67 ms

[16] https://nifi-javiervillazan.marvel-platform.eu/nifi/

Number of cluster nodes	1

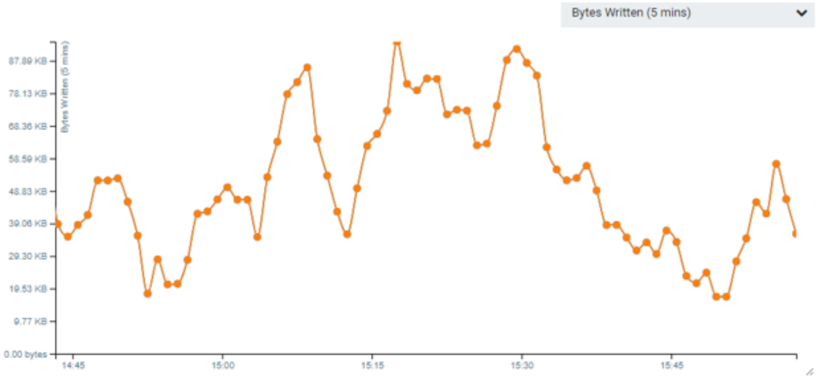


Figure 11. Scenario 1. Benchmarking test, bytes written

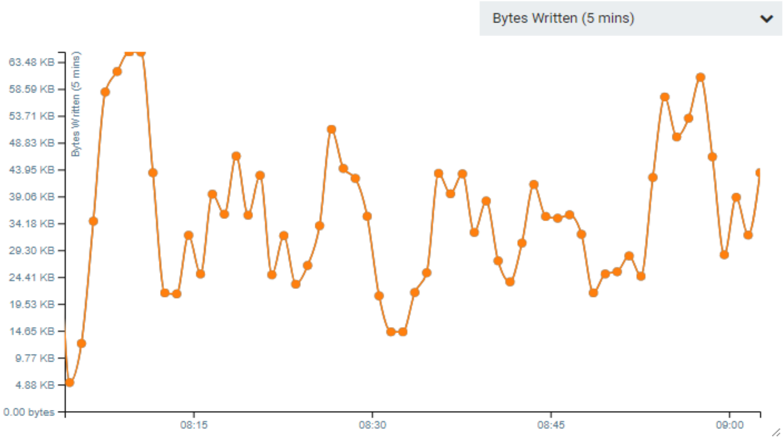


Figure 12. Scenario 2. Benchmarking test, bytes written

5.2.4.6. *Results observations*

DatAna in the MVP is only operating as a single deployment in the MARVdash cloud. No usage of MiNiFi at the edge is required or NiFi topologies, so no scalability metrics can be measured.

The scale of the data collected is very small so far. Only data coming from CATFlow is growing over time, but no sustained data collection is in place for the MVP. As a bigger scale assessment of NiFi-based systems has been provided by other vendors (see the Cloudera example above), no extra benchmarking at a similar scale has been performed in MARVEL for the MVP.

In general, DatAna has performed well with the data processed in the MVP. The latency to retrieve the results either from Kafka (scenario 1) or Kubernetes shared folders (scenario 2) is not relevant to the scenarios, while the data rates have been managed properly, taking into account the usage of just one core, introducing a very small delay. For instance, the delay introduced is on the scale of tenth of milliseconds in the overall data pipeline of the MVP for the whole set of data collected for scenario 1.

5.2.5. Contribution to MARVEL KPIs

DatAna contributes to KPIs **KPI-O1-E2-1** - Execution time of data management and distribution improved at least 15% and **KPI-O1-E2-2** - Increase the number of different modality data streams that can be handled by 30%, and **KPI-01-E2-4** - Improve data distribution in relevant device resource usage at least 15%. The contribution to those will be subject to future assessments, due to the limited scope of the deployment of DatAna in the MVP.

5.2.6. Expected future results

The current assessment can be considered as a baseline for future benchmarking. However, the scale of the data collected is very small so far. Only data coming from CATFlow is growing over time, but not a sustained data collection is in place for the MVP. The limited deployment of DatAna (only one node in the MARVEL cloud with limited resources) will be extended to the fog and edge in further iterations of the system. This will allow the possibility of measuring the data rates and the scalability of DatAna in a more realistic setting.

However, during the execution of the MVP, we have observed some disruption of the service apparently due to the excessive disk space consumed by logs in the data management platforms. This issue has been solved but would require some close monitoring in the future to avoid such issues, as well as the correct configuration of logs and internal storage (distribute the internal storage outside the Docker container in future releases) and providing enough disk space to handle growing workloads.

We expect also to migrate to NiFi versions 1.14+ in future releases in order to use new NiFi features (e.g., MiNiFi and NiFi common code base and enhanced security) available since that version.

6. Audio, Visual, and multimodal AI

AI components utilised in the MVP were audio-visual crowd counting (AVCC) and sound event detection (SED). AVCC component was used to estimate the number of people in the field of view of a camera. In this estimation process, visual information is combined with information from audio signal to produce more robust estimation. SED component was used to estimate the type of passing vehicles based on audio signal.

6.1. Audio-Visual crowd counting (AVCC)

6.1.1. Role in the MVP

The AVCC component contributes to the audio-visual data analytics of the MARVEL framework by providing an estimate of the number of people being present in the field of view of a camera. Visual information is combined with audio to provide a more robust estimation of the result in case of low visual quality. The component receives as input video frames from a camera and a synchronised audio clip and provides as output an estimated crowd count and a heatmap that is presented to the user as one of the system's results. Since for the AVCC component the objective is to achieve high performance with low computational power, the current implementation is based on our recent method improving the accuracy and processing speed by introducing a novel architecture for early exit branches.

6.1.2. Current status

At the time of the benchmark, the component provides its key functionalities. It estimates the number of people in an input image (by fusing audio and visual information), and additionally, it creates a heatmap that can be visualised. The AVCC component is not optimised for MARVEL data. It uses a model trained on a public dataset, with different camera placement, etc.

6.1.3. State-of-the-art baseline

The current SotA for audio-visual crowd counting which we use as baseline is AudioCSRNet (Hu, et al., 2020) which, on the DISCO dataset (Hu, et al., 2020), obtains a mean absolute error (MAE) of 14.24 for high-resolution images, 16.88 for low-resolution images, 13.70 for images with gaussian noise, and 27.33 for images with low illumination.

6.1.4. Description of the benchmarking process/assessment strategy

6.1.4.1. *Assessment strategy*

The component was benchmarked on a public dataset, DISCO. We aim to assess the performance of the component on data provided by the MARVEL use case providers once annotations for crowd counting are made available.

6.1.4.2. *Benchmarks used*

As a benchmark for our method, we used the DISCO dataset, which is a recent public audio-visual dataset for crowd counting containing around 2500 high-resolution images from various scenes and settings (including both day and night) and a 1-second ambient audio clip from the scene for each of the images. We evaluate the accuracy on raw images of the dataset, that is, without introducing noise or modifying the resolution or illumination.

6.1.4.3. *Infrastructure for testing*

The component runs locally on machine with 32 GB of RAM, AMD Ryzen 5 CPU, and Nvidia GeForce RTX 2060 Super GPU. The results are then shared with other components in a form of a JSON file that contains the crowd count, heatmap, and other data.

The current setup is not optimal for benchmarking, as it is run locally.

6.1.4.4. *Metrics*

Two metrics are commonly used for the evaluation of crowd counting methods: mean absolute error (MAE) and mean squared error (MSE). MAE is often used as a measure of accuracy, whereas MSE is used as a measure of robustness. For each input, we subtract the total number of people predicted by the model from the true number of people given by the label, then take the absolute of this value for MAE and the square root for MSE, then average over all values for all inputs. MSE highlights the difference between prediction and truth, so if the method performs well for dense crowds but not well for sparse ones (or vice versa), it is reflected in MSE but not necessarily in MAE, and that is why MSE is a measure of robustness.

Since the focus of our component is on improving the accuracy in audio-visual settings, the primary metric selected for the evaluation of our method is MAE. Moreover, the efficiency of the method is assessed by using the number of Floating Point Operators (FLOPs).

6.1.4.5. *Results of the measurements*

The component produces two different early exit locations, where the first location only uses visual features, and the second location utilises both audio and visual features of the input. The first early exit location has a speedup of 1.49x while achieving an MAE of 15.04 compared to the SotA MAE of 16.99. The second early exit location has a speedup of 1.48x while obtaining an MAE of 14.58 compared to the SotA MAE of 17.00. We also train the original AudioCSRNet backbone differently, which improves the MAE from 14.24 to 13.63. In terms of Floating Point Operators, the network using the first early exit has 329 BFLOPs (two variants, 329.77B using CNN early exit architecture, and 328.72B using single-layer vision transformer (SL-ViT)), while the network using the second early exit has 331 BFLOPs (331.37B using CNN early exit architecture, and 330.31B using SL-ViT).

6.1.4.6. *Results observations*

Our method significantly improves the accuracy of early exit branches compared to the SotA. Moreover, since we achieve an accuracy improvement in the second exit location by utilising the audio features in combination with the visual ones, while the SotA approach gains no improvements between the two exit locations, we show that our novel method of combining audio and visual features is superior to the original approach introduced by AudioCSRNet. In addition, our training strategy improves the accuracy of the original backbone network.

On the DISCO dataset, our method achieves SotA performance. On sample data provided for the GRN use case, the model achieves promising performance. The model outputs are not far from the ground truth even without fine-tuning on the project's specific data.

However, until annotated data for training and testing for the MARVEL use cases is available, the good performance of the model can be described as promising, but not binding. Additionally, the performance will be highly dependent on whether the model is expected to produce heatmaps or just the crowd count.

6.1.5. Contribution to MARVEL KPIs

The component contributes to the following KPIs:

KPI-O2-E2-1: Average accuracy enhancement for audio-visual representations and models at least 20%. Performance improvement of the component using an early exit compared to using an early exit at the same layer of the baseline AudioCSRNet is 14.23% (14.58 MAE against 17 MAE of the baseline).

KPI-O2-E3-1: Increase the average accuracy for audio-visual event detection by at least 10%. Performance improvement of the component using early exits with the corresponding early exits of the baseline AudioCSRNet is 14.23% (14.58 MAE against 17 MAE of the baseline).

KPI-O2-E3-3: Decrease the time needed to identify an event by at least 30% of current time. Compared to the baseline AudioCSRNet, the component achieves comparable performance (14.58 MAE against 13.63 MAE for the AudioCSRNet) while improving speed by 48%.

iKPI-2.2: At least 20% reduction in lines of code required due to new deep learning models. The component contributes to this KPI by minimising the amount of code needed for implementing audio-visual crowd counting for other use cases later in the project. The code reduction amount will be benchmarked for the final version of the benchmarking documentation.

iKPI-3.3: At least 3 approaches tested for ML training algorithms. The implementation of this component does not contribute to this iKPI yet.

6.1.6. Expected future results

The expected next steps in the evolution of this component are dependent on the availability of MARVEL data. Once the annotated data is created, we will start the process of fine-tuning our models for the specific use cases. We will aim to achieve results that are, taking into account the differences in crowd counts in those datasets, closer to the results published on the DISCO dataset.

6.2. Sound event detection (SED)

6.2.1. Role in the MVP

Sound event detection component is used in MVP for recognising vehicle types based on audio. Information about the vehicle types provided by the SED component can be used to aid the junction traffic trajectory analysis. In the MVP, the aim is to recognise segments of the captured audio signal into four vehicle types: small vehicle (car), large vehicle (truck), bus, and motorcycle. The analysis is done in short segments of audio.

6.2.2. Current status

The current SED component is based on a transfer learning approach where audio embeddings pre-trained with a large dataset are used as acoustic features for the sound activity detector. The component uses SotA audio embeddings, called PANN (Qiugiang, et al., 2020) which are pre-trained with the large-scale AudioSet dataset (Gemmeke, et al., 2017). These embeddings have shown SotA performance on audio tagging before. The embeddings are used as acoustic features for a classifier based on neural networks (bidirectional gated recurrent units and fully connected layers).

The development of the component was hindered by the delayed availability of the learning material from the MVP setup. The early development was based on public datasets such as the MAVD-traffic dataset (Zinemanas, et al., 2019) and the IDMT-traffic dataset (Abeßer, et al., 2021). These datasets contain audio material from all specified vehicle sound classes. The benchmarking in the initial stage is done with the MAVD-traffic dataset, with three vehicle types instead of four used in MVP. Later, the benchmarking will be extended to use MARVEL datasets as well. The component is not yet fully optimised and there is room for improvement in the performance.

The MVP setup has a fixed camera setup installed relatively far from the junction (e.g., 7-10 meters), and in this setup, the audio capturing microphone is installed in the video camera housing. In the MAVD-traffic dataset, the material is collected with a microphone that is installed close to the road (1-3m). The recording setup in MVP can be considered more challenging in comparison to one used in MAVD-traffic dataset collection. Because of this difference in the recording setup, the benchmarking with the MAVD-traffic dataset will give a best-case estimate of the component's performance.

6.2.3. State-of-the-art baseline

Zinemanas et al. (Zinemanas, et al., 2019) presented the MAVD-traffic dataset along with two solutions for vehicle recognition where analysis is done in one-second segments. The first proposed solution is based on MFCC acoustic features extracted on segments and random forest classifiers and the second solution using mel spectrograms as acoustic features and classification is utilising neural networks with CNN architecture. The CNN-based approach has been proposed by Salamon et al. (Salamon, et al., 2017) and the system is first trained with material from the URBAN-SED dataset and after that, the acoustic model is fine-tuned with the MAVD-traffic dataset for the vehicle recognition task. These baseline systems were evaluated with MAVD-traffic dataset (test set in cross-validation setup) and for vehicle recognition task three classes, car, bus, and motorcycle were evaluated.

6.2.4. Description of the benchmarking process/assessment strategy

6.2.4.1. *Assessment strategy*

The performance assessment is done by using the test set from MAVD-dataset's cross-validation setup for evaluating the system performance. The training set from the dataset is used for the training. Other data is allowed as training material, as was used in the CNN-based approach in (Zinemanas, et al., 2019) as well. The quantitative evaluation of the audio analysis component performance is done by comparing the system output with a reference. The reference is created by manually annotating the audio material. In this process, the sounds are labelled and their activity with start and end times are marked.

6.2.4.2. *Benchmarks used*

The MAVD-traffic dataset is used as an application-specific benchmark dataset. The entire dataset contains 2.5 hours of audio, with 4718 manually annotated passing vehicles. The dataset is released with a cross-validation setup and the test set with three vehicle types is used in the benchmarking. Statistics of the test set are shown in Table 6.

Table 6. Vehicle event statistics of the SED benchmarking dataset (MAVD-traffic dataset)

	Event instances	Total length
Bus	309	~33min
Car	627	~80min
Motorcycle	102	~12min
Total	**1038**	**~125min**

6.2.4.3. *Infrastructure for testing*

The testing is implemented in a supercomputer environment in the initial benchmarking with the following computing resources allocated:

- CPU: 10 cores from one computing node (Intel Xeon Gold 6230 2x20 cores, 2.1GHz)
- RAM: 32GB
- GPU: Nvidia Volta V100 with 32GB memory

6.2.4.4. *Metrics*

Segment-based F-score and error rate (ER) calculated in one-second segments are used as metrics (Mesaros, et al., 2016). The sound activity is compared between reference annotation and the output from the analysis component. Intermediate statistics such as true positive (TP), true negative (TN), false positive (FP), and false negative (FN) are calculated within one-second segments.

F-score is calculated by first accumulating these intermediate statistics over the evaluated segments for all classes and summing them up to get overall intermediate statistics. Based on this F-score is calculated as P = TP / (TP+FP), R = TP / (TP + FN), F = (2*P*R)/(P+R). F-score is commonly used to evaluate the system performance in sound event detection.

Error rate (ER) measures the number of errors in terms of substitutions (S), insertions (I), and deletions (D) that are calculated per segment from intermediate statistics. ER is calculated by summing the segment-wise counts for S, D, and I over the evaluated segments and normalising with the total number of segments. This metric is commonly used to evaluate the system performance in speech recognition, speaker diarisation, and sound event detection.

6.2.4.5. *Results of the measurements*

Results of the benchmarking are shown in Table 7.

Table 7. Results of the measurements for SED component

	Reference	System description	F-score (higher the better)	ER (lower the better)
Baseline	(Zinemanas, et al., 2019)	Random forest (RF)	63.1 %	0.54
Baseline	(Zinemanas, et al., 2019)	S-CNN	55.5 %	0.51
Benchmarked	MARVEL	Audio embeddings (PANN) and RNN-based classifier (2-sec)	58.6 %	0.93

6.2.4.6. *Results observations*

The measured performance of the current audio analysis component is slightly below the SotA. However, the results show already that the system is working respectable well although there is still room for improvement.

After the initial benchmarking, the component development will focus on increasing the performance with the MAVD benchmarking dataset. The next step after this will be to incorporate MARVEL data from MVP setup for the benchmarking and focus the development on dealing with the mismatches of datasets (recording setup, general acoustic characteristics).

6.2.5. Contribution to MARVEL KPIs

The component contributes to the following KPIs:

KPI-O2-E2-1: Average accuracy enhancement for audio-visual representations and models at least 20%. The performance level in the initial benchmarking is not yet contributing to this KPI.

KPI-O2-E3-1: Increase the average accuracy for audio-visual event detection by at least 10%. The performance level in the initial benchmarking is not yet contributing to this KPI.

iKPI-3.2: At least 20% reduction in lines of code required due to new deep learning models. The component contributes to this KPI by minimising the amount of code needed for implementing audio analysis systems for other use cases later in the project. The code reduction amount will be benchmarked for the final version of the benchmarking documentation.

6.2.6. Expected future results

Currently, the benchmarking is implemented with an open dataset and it gives only a best-case estimate of the component's performance. Later the benchmarking will be extended to include MARVEL datasets to get more accurate performance estimate. The SED component is currently in the initial stage, and the performance is not fully optimised. By the end of the project, specified KPIs will be fulfilled by continuing the component development. Possible ways to achieve the KPIs are using a different approach (neural network architecture), by optimising the model learning procedure. Furthermore, data augmentation strategies during the model learning procedure have not been applied yet. All these should result in an increase in performance based on the prior work on related analysis applications.

7. Optimised E2F2C processing and deployment

MARVdash (former known as Karvdash) is a dashboard for instantiating services as orchestrated containers and deployed via appropriate automation to execution sites selected by a dynamic online optimisation strategy. It enables user-friendly, consolidated management of distributed services deployed in the entire execution-site continuum, spanning from the Edge to the Cloud. It provides a web-based graphical frontend to coordinate accesses to the E2F2C execution platform, orchestrate service execution in containers from pre-defined templates, and interact with collections of data, which are made available automatically to application containers when launched.

7.1. MARVdash

7.1.1. Role in the MVP

The role of MARVdash in the MVP is manifold. It coordinates the execution of the data management platforms on the E2F2C testbed and allows for services that need to be exposed outside to MARVEL infrastructure to become accessible. Moreover, MARVdash provides the implementation basis for the optimised deployment of the AI tasks. Furthermore, MARVdash allows corresponding services of the DatAna and DFB data platforms to be launched thanks to the creation of MARVdash-compatible templates.

All containers are launched within the user's private namespace, while respective frontends (DatAna's NiFi GUI and DFB's Kibana GUI) are exposed via Internet-accessible URLs, over HTTPS, and protected using the MARVdash authentication infrastructure. These services can access any resource in the E2F2C-private network, while produced data is directed to MARVdash-supplied storage that can also be accessed via the MARVdash web interface.

Finally, MARVdash also assists in the deployment of the Data Corpus, integrating a web proxy service that allows external, secure access to the Data Corpus API services. According to the proxy implementation, a new ingress endpoint is registered at the Kubernetes side and is configured to require authentication using the MARVdash-registered credentials before forwarding requests to the backend.

7.1.2. Current status

Many of the main functionalities of the MARVdash are already available such as running of a service, creation and usage of a template, uploading of a container image, usage of Jupyter notebook, creation of an Argo workflow, and management of user files. What is still missing is some optimisation that is planned for the second half of the project, such as the creation of containers in pods close to the data they use as input.

7.1.3. State-of-the-art baseline

Since MARVdash is an interactive software, user experience metrics will be incorporated in its evaluation, such as the User Experience Questionnaire (UEQ). UEQ is a questionnaire that is used for the assessment of interactive software. There are two versions of the said questionnaire, a short and an extended one. UEQ includes 26 items, with each one being a pair of antonyms and a scale between them from -3 (fully agree with negative term) to 3 (fully agree with positive term). These items are grouped in 6 scales. (Hinderks, et al., 2018)

7.1.4. Description of the benchmarking process/assessment strategy

7.1.4.1. *Assessment strategy*

MARVdash does not do any processing but provides the mechanisms to efficiently deploy the AI models and related support services included in the MARVEL data management and distribution toolkit in all computing continuum layers that support container-based execution. In that sense, MARVdash plays the role of a dashboard for requesting resources and specifying other parameters related to service execution. Those dashboard aims to make it straightforward for domain experts to interact with resources in the E2F2C platform without the need for understanding lower-level tools and interfaces.

The overall assessment will be conducted through a survey, where the users will rate their experience. Without MARVdash, users would be required to write deployment files, specify the individual placement of containers in the continuum, and apply these configurations through the command-line interfaces of each system component. The survey will include questions regarding the main functionalities of the MARVdash, its advantages and disadvantages compared to other Kubernetes dashboards, and the whole user experience.

7.1.4.2. *Benchmarks used*

User Experience Questionnaire includes a benchmark that allows for assessment of how good or bad is a given product, in comparison to other products. Measured scale means as a result of the answers to the UEQ for a large number of products are available and can be used for comparison.

The feedback that the benchmark returns ranks the given product into 5 categories: i) Excellent, meaning that the evaluated product is among the best 10% of results; ii) Good, meaning that 10% of the results in the benchmark are better and 75% of the results are worse; iii) Above average, meaning that 25% of the results in the benchmark are better and 50% of the results are worse; iv) Below average, meaning that 50% of the results in the benchmark are better and 25% of the results are worse; and v) Bad, meaning that the evaluated product is among the worst 25% of results (Schrepp, et al., 2017).

In that way, we can have an idea of how good or bad MARVdash is in comparison with other software.

7.1.4.3. *Infrastructure for testing*

The current version of the questionnaire for benchmarking MARVdash is available online[17]. Additional software that is needed for the processing of the responses, can be found at the website of the User Experience Questionnaire.

7.1.4.4. *Metrics*

The questions of the questionnaire are grouped into three sections. There is a number of questions that refer to MARVdash specific functionalities such as starting a service, using a template, uploading an image, etc. that constitute the first section.

The second section includes questions that try to compare MARVdash with other Kubernetes dashboards, asking the user to mention what are the MARVdash advantages and disadvantages compared to those other dashboards.

[17] https://docs.google.com/forms/d/e/1FAIpQLSe-xpswq4Vwxts_NVaMJt5JFiKjNLiZb2oI7gMA1XDZQiajxg/viewform

Finally, the third section includes all the questions from the UEQ that will help us compare MARVdash with other products regarding the user experience. Said questions are further grouped into six scales: Attractiveness, Perspicuity, Efficiency, Dependability, Stimulation, and Novelty. For each of those, a number will be produced as an output of the benchmark.

7.1.4.5. *Results of the measurements*

The created questionnaire was made available to the MARVEL partners for completion. Individuals that were already familiar with MARVdash were asked to answer the predefined questions. In this subsection, we present the results derived from the given answers.

First section

The results of the answers given for the first section of the questionnaire are presented in Figure 13 and Figure 14.

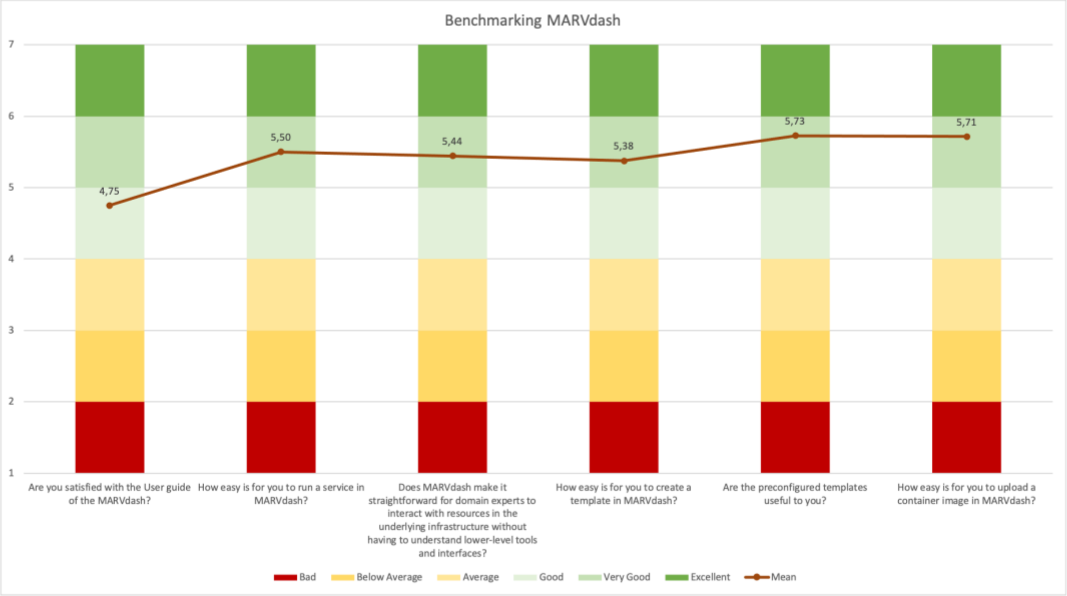


Figure 13. Averages for the 1st section of questionnaire - part A

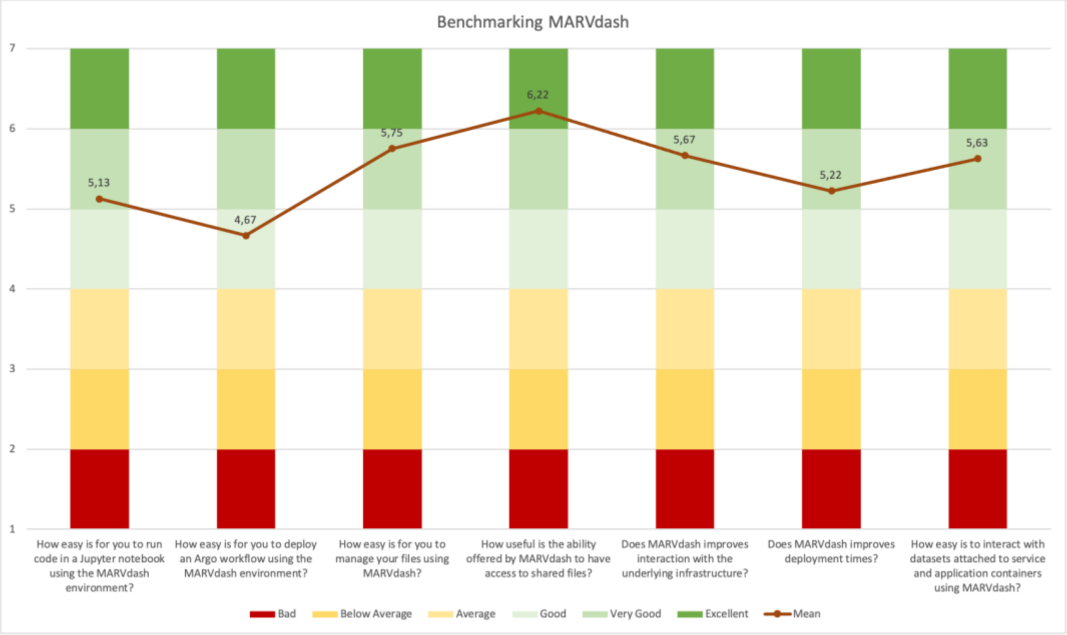


Figure 14. Averages for the 1st section of questionnaire - part B

The users could give their answers by choosing one point of a 7-point Likert scale. The above figures show the calculated average number for each of the given questions. The averages from the received answers range from 4.75 the lowest to 6.22 the highest. All these numbers belong to the "Very good" category (one of them belongs to the "Excellent") of the corresponding qualitative assessment.

Second section

Three other choices for interacting with Kubernetes were mentioned by two of the participants, named Red Hat OpenShift Kubernetes Engine, Kubernetes Dashboard, and Rancher Kubernetes Engine (RKE).

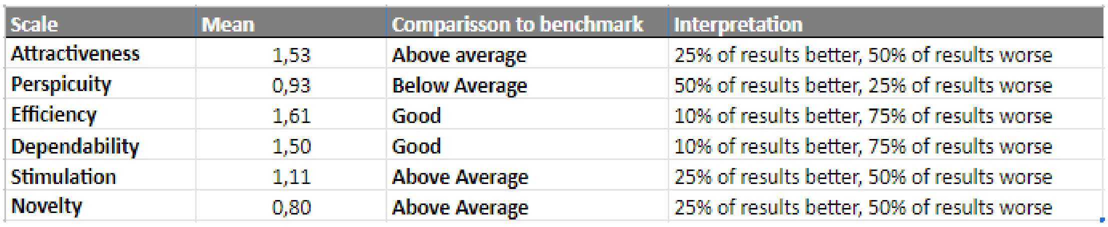
Regarding Red Hat OpenShift Kubernetes Engine, its dashboard has kubectl built into oc, the command-line interface for Red Hat OpenShift. Based on the received answers OpenShift offers advanced logging with historical data.

Kubernetes Dashboard is a native web-based Kubernetes user interface. According to the responses, one of its advantages, compared to MARVdash, is the visual representation of architecture (with networking).

RKE is a CNCF-certified Kubernetes distribution that runs entirely within Docker containers, making the installation of Kubernetes less complex. Its dashboard allows for different Kubernetes clusters to communicate with each other. This was mentioned as a feature that is missing from MARVdash.

Third section

As we have already mentioned, part of our questionnaire included questions from the UEQ. The results of this part are presented in Figure 15 and Figure 16. Both representations of the results show how good MARVdash is compared to the products in the benchmark dataset. This comparison allows for the identification of advantages and disadvantages of MARVdash, indicating whether it offers sufficient user experience.



Scale	Mean	Comparisson to benchmark	Interpretation
Attractiveness	1,53	Above average	25% of results better, 50% of results worse
Perspicuity	0,93	Below Average	50% of results better, 25% of results worse
Efficiency	1,61	Good	10% of results better, 75% of results worse
Dependability	1,50	Good	10% of results better, 75% of results worse
Stimulation	1,11	Above Average	25% of results better, 50% of results worse
Novelty	0,80	Above Average	25% of results better, 50% of results worse

Figure 15. Means for the 6 scales of UEQ for MARVdash

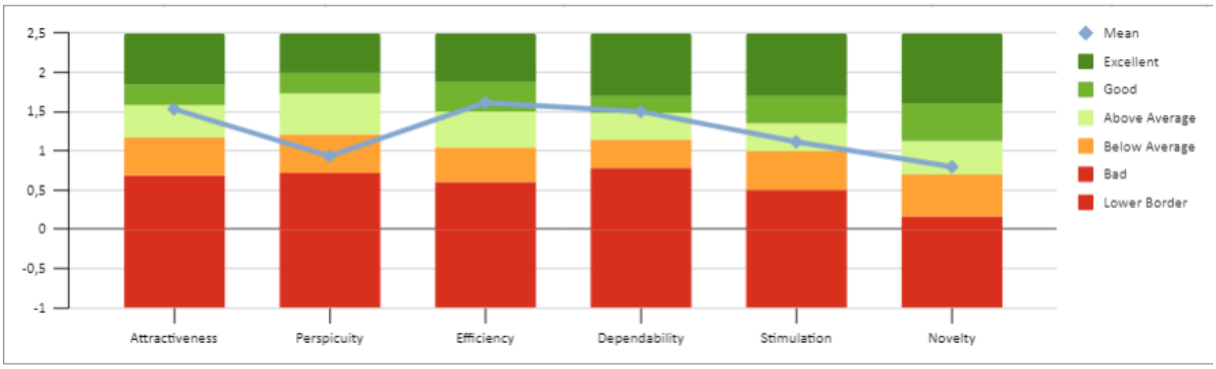


Figure 16. Visualisation of the benchmark results for the 6 scales of UEQ for MARVdash

The first column of Figure 15 includes the names of the six scales that the 26 items of the UEQ questionnaire are grouped into 6 categories, namely, Attractiveness, Perspicuity, Efficiency, Dependability, Stimulation, Novelty. The second column presents the corresponding mean values. The third column of the table shows the comparison to other products included in the benchmark dataset, while the fourth column presents the interpretation of the means. The per scale feedback is grouped into five categories: Excellent, Good, Above average, Below average, Bad. These categories are already explained in Subsection 7.1.4.2 above. As it can be seen, two out of the four MARVdash's means are characterised as "Good", three of them are placed in the "Above Average" category, and only one is considered "Below Average".

Figure 16 is a visualisation of the benchmark and shows for each scale, how the MARVdash results are related to the products in the benchmark dataset. The line represents the calculated means of MARVdash, while the coloured bars represent the ranges for the scales' mean values. The legend at the right side of the chart includes the five feedback categories and their corresponding colours.

7.1.4.6. *Results observations*

According to the answers from the first section of the questionnaire, the users seem to be satisfied with the way the main functionalities of MARVdash are delivered to them. They rated MARVdash with high average scores.

The questions regarding the comparison with other Kubernetes dashboards were answered by a very small set of participants. Although this was good initial feedback, we could use more answers to make comprehensive observations.

Regarding user experience, the first round of questionnaires showed that MARVdash's means are above average compared with a large number of other products. Based on that MARVdash could be successful in the market.

We should say that a larger number of responses will give a more clear view of what are the strengths and weaknesses of MARVdash. We anticipate that during a future second round of questionnaires, more individuals will be familiarised with MARVdash and could provide their feedback.

7.1.5. Contribution to MARVEL KPIs

The KPIs that are affected by MARVdash include:

KPI-O3-E2-2: Optimise performance (prediction accuracy, time-to-decision) of DL deployment by 20%. This KPI is linked with the distributed execution of DL tasks. Towards that end, software architecture has been defined, spanning from E2F2C/HPC that enables the said distributed execution of DL tasks, and with efficient use of execution resources as a major concern. MARVdash contributes to the ability to match the task resource requirements to the various execution sites available in the MARVEL distributed environment. Consequently, it is possible to enable improvements both in performance, particularly time-to-decision, as well as in the sophistication of the DL models being deployed, thus enhancing prediction accuracy.

iKPI-1.1: At least three (3) tools for complex/federated/distributed systems handling extremely large volumes and streams of data. MARVdash contributes to this KPI not directly but indirectly, by enabling the instantiation of the first version of component FedL. This component is scalable to a large number of FL clients and capable of handling data from multiple sites arriving in a streaming fashion. As the FedL component matures new requirements may rise from the side of MARVdash however no blocking issue is foreseen.

iKPI-12.2: Increased performance in terms of response time, throughput, and reliability compared to a standard approach. According to the roadmap towards meeting this KPI, experiments will start within FedL to compare the response time and reliability of the FedL protocol compared to a baseline approach. MARVdash allows for the deployment of FedL and the execution of said experiments.

7.1.6. Expected future results

The results of the benchmarking will give feedback regarding the main functionalities of the MARVdash, its advantages and disadvantages compared to other Kubernetes dashboards, and the whole user experience. This feedback will be extremely useful for the further design of MARVdash. We will try to tackle any shortages in its functionality and correct any flaws in its design. As far as the user experience is considered, the comparison with a large number of products will give us feedback for future design decisions.

8. E2F2C infrastructure

GRN has two components in the E2F2C infrastructure: the GRN edge infrastructure and the GRN fog infrastructure. The GRN edge infrastructure is meant to capture AV data which is then streamed to the GRN fog infrastructure where it is processed by the CATFlow algorithm. From the fog layer the output of the CATFlow algorithm, that is structured non-binary data, is pushed as Kafka messages to the cloud infrastructure. The cloud infrastructure is handled by PSNC.

8.1. GRN edge infrastructure

8.1.1. Role in the MVP

The GRN edge infrastructure is the static surveillance camera with an integrated microphone located in Mgarr, connected to the mobile internet. The role of the edge infrastructure in the MVP was to capture AV data. This data was transferred in real-time to the GRN fog layer to be then processed by the CATFow algorithm or stored so that it could be used to train AI models (after annotations).

8.1.2. Current status

The edge infrastructure is able to continuously stream AV data to the server. The next addition to the edge infrastructure is having the ability to process at the edge.

8.1.3. State-of-the-art baseline

The solution used for the MVP is mostly an off-the-shelf solution, thus this solution is at the SotA.

8.1.4. Description of the benchmarking process/assessment strategy

8.1.4.1. *Assessment strategy*

The assessment strategy for the off-the-shelf components was to observe that the camera is robust and reliable. The edge infrastructure must be able to stream continuously, ideally not missing any anomalous events and being robust in adverse weather conditions, including storms.

8.1.4.2. *Benchmarks used*

The assessment strategy used is to confirm that the system is robust and reliable by using it for an extended period and noting the failures or outage times.

8.1.4.3. *Infrastructure for testing*

The hardware setup of the system is as shown in Figure 17 with a Safire 2MP Dome Outdoor/Indoor IP camera with POE and built-in microphone, connected to a Network Video Recorder (NVR) through POE Ethernet cable. The NVR is connected to the internet through a direct connection to a router.

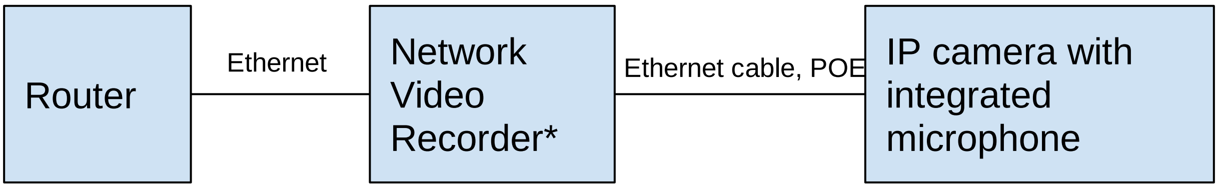


Figure 17. Diagram of Camera Setup. *NVR is only used to supply power and as a way to connect the IP camera to internet

A live RTSP stream was set up from the IP camera with an integrated microphone. The frame rate is 25fps, max bitrate is 2048Kbps, H265 video encoding, and G.711 ulaw audio encoding.

Through port forwarding on the router, the stream can be accessed from anywhere. Thus, the stream was accessed by the fog server and by personal computers to assess the quality and benchmark.

In addition, a visual privacy mask was applied to the video stream which covers areas that are unnecessary to the traffic surveillance and hide private properties.

8.1.4.4. *Metrics*

The metrics considered are sustained performance in various weather and ambient lighting conditions to confirm robustness. The reliability of the system is measured by observing when the system stops transmitting. The throughput of the system is also recorded to measure the data produced.

8.1.4.5. *Results of the measurements*

Robustness: The edge infrastructure was found to work in all weather conditions experienced throughout the four months of testing, exhibiting no signs of water damage or loss of quality. These weather conditions range from very hot September days to rainy, windy winter days in January. The only observable loss of quality is the audio quality on very windy days.

The edge infrastructure also streamed in all lighting conditions. The poorer quality video was seen during the dusk hours which might affect the quality of the AI models using this data.

Reliability: The edge infrastructure streamed reliably through most conditions. One artefact that might be considered is that the router restarts itself automatically during early morning hours, thus any system that uses this stream must account for this.

Throughput: Through observation of AV snippets recorded from the GRN edge setup, the throughput was estimated to be approximately 100Kb/s.

8.1.4.6. *Results observations*

The GRN Edge infrastructure has shown to be reliable and robust in most conditions, thus future additions to the GRNEdge infrastructure would be similar.

8.1.5. Contribution to MARVEL KPIs

The component contributes to the following KPIs:

KPI-O1-E1-1: Different kind of resources to be discoverable: ≥ 3. This KPI explores the need to discover different kinds of devices to connect with the MARVEL framework. Thus, this infrastructure is contributing to the KPI.

KPI-O5-E1-1: More than 3.3PB of data made available through a Corpus-as-a-Service**.** This setup is a contribution to this goal.

8.1.6. Expected future results

Future work on the GRN edge infrastructure could include the addition of edge processing so that the need to transmit sensitive data is diminished or achieve lower latency. This would also reduce the bandwidth required to transmit information making the GRN edge infrastructure more efficient.

8.2. GRN fog infrastructure

8.2.1. Role in the MVP

The GRN fog infrastructure plays two roles in the MVP. The first role is to receive the AV stream from the edge infrastructure and process it with the CATFlow algorithm. The CATFlow algorithm produces structured non-binary data which is then sent to the MARVEL Cloud. The other role is to host a virtual machine for the MARVEL infrastructure.

8.2.2. Current status

The fog infrastructure is functionally complete. The CATFlow algorithm is functional and the VM can be accessed. Currently, the VM is being altered such that more IPs can be used to access it.

8.2.3. State-of-the-art baseline

The device is an off-the-shelf solution thus cannot be compared to a SotA of similar solutions.

8.2.4. Description of the benchmarking process/assessment strategy

8.2.4.1. *Assessment strategy*

To test the ability of the GRN fog infrastructure before using it for the CATFlow algorithm a script was generated to measure the load that the server's GPU can handle by running the YOLO algorithm (Redmon, et al., 2016) on it.

8.2.4.2. *Benchmarks used*

The benchmark used was to run the YOLO4 algorithm (Bochkovskiy, et al., 2020) on the server to measure the load that the GPU can handle. With this approach, the frames per second that can be handled by our detector were determined.

8.2.4.3. *Infrastructure for testing*

The server has the following specifications:

- Server: HPE ProLiant DL385 Gen10 Plus
- CPU: AMD EPYC 7302 16-Core Processor
- RAM: currently 32GB using 16GB sticks
- GPU: Nvidia Tesla T4 16GB
- Storage: HPE 480GB SATA

8.2.4.4. *Metrics*

The metrics being measured are the frames per second the server can handle and the amount of GPU memory used by each AV stream input.

8.2.4.5. *Results of the measurements*

The detector was found to use 867MiB out of the 16GB (1/16th of the GPU memory) provided when YOLO runs at 5fps using a detector size of 416.

8.2.4.6. *Results observations*

The results give an indication of what processing power is needed at the edge and the volume of real-time processing that the fog layer can handle.

8.2.5. Contribution to MARVEL KPIs

The component contributes to the following KPIs:

KPI-O1-E2-3: Increase the speed of the fusion process by at least 20%. The fog layer contributes to a reduction in latency and to speeding up the fusion process.

KPI-O1-E3-1: Number of incorporated safety mechanisms (e.g. for privacy, voice anonymization) ≥ 3. The fog layer contributes to a number of safety mechanisms via the implementation of secure edge-fog data transfer and anonymised data when transferring from fog to cloud.

KPI-O2-E3-3: Decrease the time needed to identify an event by at least 30% of current time. The fog layer contributes to a reduction of time taken to identify an event in the video through its computational power and its proximity to the data source.

8.2.6. Expected future results

In the future, it is expected that the same requirements are necessary for the Fog infrastructure, and the hardware required for the server increases as the number of streams that need to be processed increases.

8.3. PSNC cloud infrastructure

8.3.1. Role in the MVP

Cloud infrastructure is used as a base for the deployment of the MARVEL software stack.

8.3.2. Current status

The component consists of existing infrastructure and is therefore perceived as complete. However, there may be some modifications or optimisations during the project.

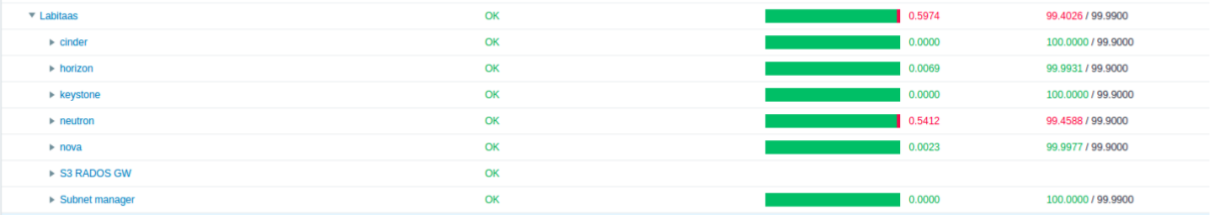
8.3.3. State-of-the-art baseline

Since with similar services, the exact scope and items monitored depend on the specific SLA between infrastructure operator and user, there is a “standard” way of monitoring this kind of services.

8.3.4. Description of the benchmarking process/assessment strategy

8.3.4.1. *Assessment strategy*

The infrastructure is being monitored on physical (servers, switches, state of links, etc.) and logical levels. From MARVEL’s perspective, more important is logical level monitoring on which all basic components are tested and checked for proper behaviour. Monitoring and checking against agreed SLA levels is done using Zabbix software. A sample screen with a summary of current and agreed SLA levels for the tested period of time is presented in Figure 18.



▾ Labitaas	OK	0.5974	99.4026 / 99.9900
▸ cinder	OK	0.0000	100.0000 / 99.9000
▸ horizon	OK	0.0069	99.9931 / 99.9000
▸ keystone	OK	0.0000	100.0000 / 99.9000
▸ neutron	OK	0.5412	99.4588 / 99.9000
▸ nova	OK	0.0023	99.9977 / 99.9000
▸ S3 RADOS GW	OK		
▸ Subnet manager	OK	0.0000	100.0000 / 99.9900

Figure 18. Sample screen with summary of SLA levels for the tested period

One can see that there are several columns listed. The first column holds the name of the component followed by the current state. The next column shows unavailability levels both in graphical and numerical format. The last column presents current and threshold values of the SLA for a given module.

The tested modules are:

- LabITaaS: aggregates values for the cloud infrastructure calculated on the basis of all components. This aggregate considers the whole cloud as non 100% functional if any of the components is not available.
- cinder: block storage service for all virtual machines and volumes used in the cloud environment. Unavailability of this service may affect running VM (virtual machines) as it may (but not have to) indicate that VM lost access to the underlying storage
- horizon: web interface used for human interaction for creation or manipulation of virtualised resources. Unavailability of this service does not affect running VMs but prevents the possibility of manipulation of the resources using the WWW interface but does not affect any already configured services.
- keystone: basic identity service used by OpenStack components. Unavailability mostly affects the ability of components to communicate with each other but does not imply malfunction of VMs.
- neutron: network virtualisation service. Unavailability affects the possibility to change or create networking components in Cloud and may also mean some problems with accessing some or all virtual machines from the external networks.
- nova: service responsible for creation and manipulation of the VMs. Unavailability prevents new VMs creation.
- S3 RADOS GW: access to object storage via S3 gateway. This represents the availability of the S3 storage. Does not directly affect VMs but may mean a lack of data access for services that use this storage.
- subnet manager – management service for Infiniband network. Because PSNC cloud relies on high speed, low-latency Infiniband network for storage and networking, malfunction of this service means severe problems with all other services and VMs running in the environment.

8.3.4.2. *Benchmarks used*

All components are tested at least on two levels:

- Operating system check: if all needed processes are running and are not in “zombie” or other states that indicate problems.
- Logical check: most services (with exception of subnet manager) can be queried using REST interface. In this way, Zabbix can determine if each and every one of the

components report status “OK”. For the subnet manager, a “visibility test” is executed – Zabbix queries compute nodes about the visibility of the subnet manager.

All components are configured in high availability configurations meaning each of the components consists of at least two services. Service is considered inactive if all services of a given component are being detected as faulty.

8.3.4.3. *Infrastructure for testing*

All systems that build up the cloud infrastructure are being monitored, this means more than 200 servers, multiple Ethernet and Infiniband switches, etc.

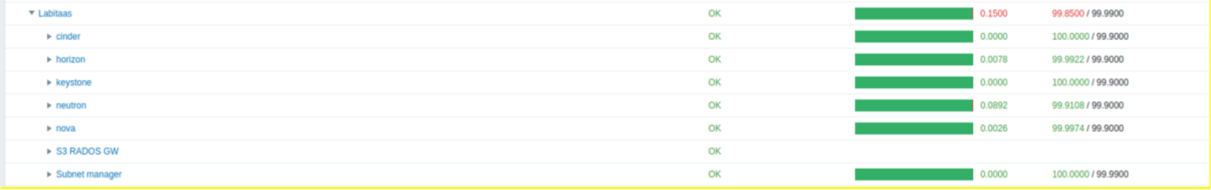
In addition, a set of virtual machines is used for monitoring, these consume 40 VCPUS, 64GB of memory and 20TB of fast storage. Virtual machines host zabbix services are located on a separate, simplified cloud environment managed by Proxmox software.

8.3.4.4. *Metrics*

The basic measure metric is the availability of the LabITaaS service. The SLA level PSNC offers is 99,9% in a 30-day period.

8.3.4.5. *Results of the measurements*

Values for 01/2022 are shown in Figure 19.



▾ Labitaas	OK	🟩	0.1500	99.8500 / 99.9900
▸ cinder	OK	🟩	0.0000	100.0000 / 99.9000
▸ horizon	OK	🟩	0.0078	99.9922 / 99.9000
▸ keystone	OK	🟩	0.0000	100.0000 / 99.9000
▸ neutron	OK	🟩	0.0892	99.9108 / 99.9000
▸ nova	OK	🟩	0.0026	99.9974 / 99.9000
▸ S3 RADOS GW	OK			
▸ Subnet manager	OK	🟩	0.0000	100.0000 / 99.9900

Figure 19. Measurement results for 01/2022

8.3.4.6. *Results observations*

To better reflect the “real” availability of the cloud environment we should create a more elaborate aggregation algorithm that takes into account that some of the metrics are critical from point of view of clients like the MARVEL software stack.

8.3.5. Contribution to MARVEL KPIs

The component contributes to the following KPIs:

KPI-O5-E4-1: Maintain the corpus for at least one year after the end of the project.

8.3.6. Expected future results

To be more informative from a project perspective the monitoring should take into account also components that are not yet integrated, such as data store for the corpus service.

9. Decision making and user interactions

In this section, SmartViz and the MARVEL Data Corpus components are discussed. SmartViz is the final component in the MVP's pipeline and it has an important role in the use case scenarios by providing the decision-making tools for users. In the MARVEL Data Corpus component, the processed multimodal audio-visual data is stored. The component is released as a service from which the data can be obtained free of charge.

9.1. SmartViz

9.1.1. Role in the MVP

SmartViz is a data visualisation toolkit that constitutes the user interface of the Decision-Making Toolkit (DMT). It plays a significant role for two of the use case scenarios selected for the MVP. SmartViz is the final component in the MVP's pipeline and its purpose is to utilise the incoming detected events, deriving from the analysis of data coming through the edge layer, and to give meaningful insights and interaction capabilities to the end-users. The decision-making toolkit consumes a variety of data coming through different workflows and pipelines inside the MARVEL infrastructure. Data are displayed using a pool of visualisations that include charts, graphs, temporal representation, and geospatial depictions carefully selected for the respective data and the users' needs. SmartViz facilitates the discovery of patterns, behaviours, and correlations of data items via a visual data exploration that fosters the decision-making process for urban planning.

9.1.2. Current status

At the time of the benchmarking, SmartViz is enabling the key functionalities of the MARVEL decision-making toolkit. It consumes a variety of data coming through different workflows inside the MARVEL infrastructure and it serves them in visualisation schemas and widgets that facilitate decision-making and satisfy the end-users' needs in terms of getting situational awareness. Limited filtering mechanisms are also available to allow further data presentation and exploratory analysis.

The decision-making toolkit does not have full functionality yet, however it can be used to explore a limited set of data and receive the user's feedback in order to proceed with improvements and better design the next features.

9.1.3. State-of-the-art baseline

Visualisation benchmarks are quite difficult to be identified since they are dependent on the calculations and data processing supporting them. Therefore, ZELUS decided to focus only on SmartViz usability regarding SotA benchmarking, using 2 datasets leveraged in the study of (Lewis & Sauro, 2009) as well as 500 evaluations from the private research of one of the authors, the results of which can be found online[18]. The data were collected from the SUS usability scale (Bangor, et al., 2008), a 10-item System Usability Scale, which since its introduction in 1986, has been assumed to be unidimensional. A perfect SUS score is a 100 and anything above 68 is considered to be above average and anything below 68 is below average, with the top-performing, scoring above 80.2 points.

We will be tracking more parameters for benchmarking, namely: scalability, availability, reliability, and performance for which we will be comparing the experience from the SmartViz

[18] https://measuringu.com/sus/

and by extend from the Decision-Making Toolkit with the one the users have with their current ways of working in terms of SotA, but also with measurements from internal tests of using the tool in other research programs run by ZELUS.

9.1.4. Description of the benchmarking process/assessment strategy

9.1.4.1. *Assessment strategy*

The MARVEL MVP provides the possibility to assess a limited set of functionalities, given the limited data it processes. However, usability, availability, and reliability of the Decision-Making Toolkit (and therefore SmartViz) are still possible to be measured setting a baseline for future improvements and minimum values for the functionalities that will be added in future releases. ZELUS has prepared a questionnaire to collect the input from the consortium partners, leveraging their role as end-users but also as collaborators especially as far as the reliability parameter is concerned, ensuring that no faulty information is delivered by the tool.

The questionnaire is based on the widely used System Usability Scale (SUS), as mentioned in paragraph 9.1.3, all answered at a scale of 1 to 5 where 1 stands for “Strongly Disagree” and 5 for “Strongly Agree” and is enriched with questions for the rest of the KPIs:

1. I think that I would like to use the MARVEL Decision Making Toolkit (DMT) frequently. (scale 1-Strongly Disagree to 5-Strongly Agree)
2. I found the MARVEL DMT unnecessarily complex. (scale 1-Strongly Disagree to 5-Strongly Agree)
3. I thought the MARVEL DMT was easy to use. (scale 1-Strongly Disagree to 5-Strongly Agree)
4. I think that I would need the support of a technical person to be able to use the MARVEL DMT. (scale 1-Strongly Disagree to 5-Strongly Agree)
5. I found the various functions in the MARVEL DMT were well integrated. (scale 1-Strongly Disagree to 5-Strongly Agree)
6. I thought there was too much inconsistency in the MARVEL DMT. (scale 1-Strongly Disagree to 5-Strongly Agree)
7. I would imagine that most people would learn to use the MARVEL DMT very quickly. (scale 1-Strongly Disagree to 5-Strongly Agree)
8. I found the MARVEL DMT very cumbersome to use. (scale 1-Strongly Disagree to 5-Strongly Agree)
9. I felt very confident using the MARVEL DMT. (scale 1-Strongly Disagree to 5-Strongly Agree)
10. I needed to learn a lot of things before I could get going with the MARVEL DMT. (scale 1-Strongly Disagree to 5-Strongly Agree)
11. Did you experience any crash or malfunction while using the MARVEL DMT? (Yes – No, and open text for elaboration in case of Yes)
12. I prefer using the MARVEL DMT compared to my current way of working for accessing traffic data at a specific road junction (scale 1-Strongly Disagree to 5-Strongly Agree and open text for elaboration in case of responses in the range of 1 and 2)
13. I prefer using the MARVEL DMT compared to my current way of working for accessing road users’ behavior data at a specific road junction (scale 1-Strongly Disagree to 5-Strongly Agree and open text for elaboration in case of responses in the range of 1 and 2)

To measure the performance of the component regarding load times of landing and internal pages once it becomes available to the wider audience, we will use an open-source Java application, the Apache JMeter[19], that is designed to implement stress test scenarios. Indicative scenarios were selected that are of interest in terms of performance, due to the stress that they can bring to the component. These scenarios are aiming to observe the performance of the load times of the pages in the component while the number of users that use it is increasing from 1 to 100.

Once the demo becomes available to the wider public (at the moment data privacy restrictions allow us to circulate the benchmarking questionnaire only among the partners) we are planning to add usability based on the User-Centric Design (ISO 9241-210:2019, 2019). Once more data become available after the real-life experimentation (WP6), we will be also assessing the scalability of the DMT and its capability to support browser-friendly data volumes with no errors that prevent its reliability and performance.

Based on the internal research project where SmartViz was used, we expect to score above 80.2 regarding usability (questions 1-10) positioning us together with the top-performing applications regarding usability, no downtime during the operation of the tool (availability and reliability: question 11) and an average load time of maximum 9.9 seconds for the loading of the landing page and 1.2 seconds for the loading of the internal pages.

9.1.4.2. *Benchmarks used*

All functionalities will be benchmarked from one hand with the current experience of the users with the tool in terms of usability, availability, and reliability compared to the tools and processes they are used to (questions 12 and 13) and on the other hand with the widely accepted norm according to user experience metrics (wherever available – questions 1-10 should score above 68[20]). The baseline for all these metrics is the performance of the tool in internal research projects (see paragraph 9.1.4.5), which were validated after the MVP demonstration, based on the users' feedback.

9.1.4.3. *Infrastructure for testing*

A local desktop computer was used for the initial benchmarking for the performance and availability parameters. The component is running locally on machines with 16 GB of RAM, 2GHz Quad-Core Intel Core i5.

9.1.4.4. *Metrics*

Since the nature of the SmartViz component is directly dependent on the users' interactions and needs, the metrics we would like to measure are mainly user experience-oriented.

The first metric is the perception of the required time to accomplish a task and the quality of the task. This metric will give us an overall understanding of the satisfaction of the user's needs. The volume of the visualised data is another important metric that is giving valuable insights into the functionality and ability of the component to handle big data volumes. Another metric we would like to measure is the accuracy and data error resilience that succours the data and error handling of the component. The response time of SmartViz to user actions is also an important metric that will be measured.

Last but not least, starting from the MARVEL prototype where the possibility to realise more tasks will be available for the user, we are planning to perform usability tests and track task completion and user satisfaction from the process of completing concrete tasks. The results will

[19] https://jmeter.apache.org/

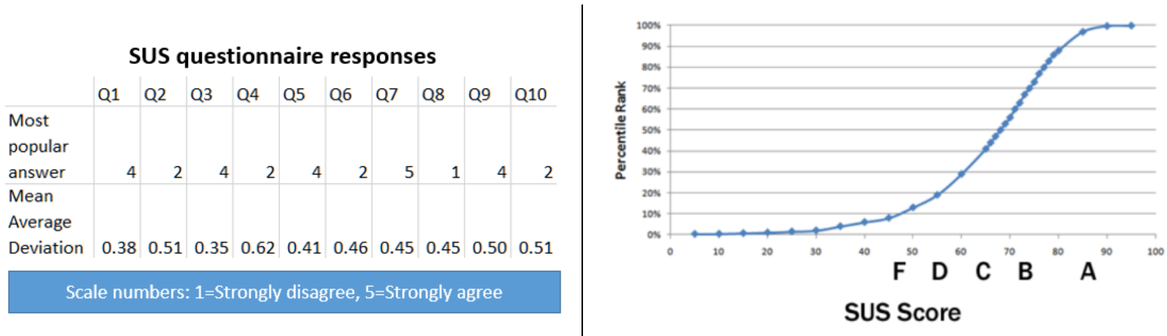
[20] https://measuringu.com/sus/

be compared to the final version of the DMT as well as the users' experience from the way they work now to perform a similar task.

9.1.4.5. *Results of the measurements*

The measurements regarding the MVP were only performed internally in the consortium, therefore we aim to get at least the same or improved results once data privacy issues are resolved and the tool becomes available for testing by a wider tester base. Having mentioned that, according to the SUS scale, the average **usability** score of the MVP of the MARVEL Decision-Making Toolkit is 80.625, which gives it an A grade according to the results of 500 studies[5] (see also Figure 20). The majority of users though were not sure yet whether they would prefer using the MARVEL DMT compared to their existing ways of working, which is something we consider understandable since it is only the MVP version with limited capabilities.

Unfortunately, availability presented only 91,7% success, which is something we are investigating together with the rest component owners, in order to identify the root causes. Reliability of what the tool was showing and the input it was receiving was based on internal checks and was 100%



SUS questionnaire responses

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
Most popular answer	4	2	4	2	4	2	5	1	4	2
Mean Average Deviation	0.38	0.51	0.35	0.62	0.41	0.46	0.45	0.45	0.50	0.51

Scale numbers: 1=Strongly disagree, 5=Strongly agree

a) b)

Figure 20. On the right: DMT usability data. On the left: percentile ranks association with SUS scores and letter grades

9.1.4.6. *Results observations*

The results of the initial baseline met our expectations with the exception of the availability measurement which is under investigation at the moment this report was being written. Overall, the MARVEL DMT got a positive head start with a few features such as the animation option when the user is observing vehicle trajectories in order to understand the traffic trend in particularly short timeframes seems to have left better impression than we even expected. On the other hand, we also received constructive feedback for improving the tool such as items that help the user understand what they see e.g., better explanation texts or help buttons, etc. Everything will be taken into consideration before the next release.

9.1.5. Contribution to MARVEL KPIs

KPI-O3-E4-1: Detailed insights to more than 5 hidden correlations. This KPI is linked with the design and development of the DMT (started at M10) and the visualisations depicted there. It is naturally linked to Enabler 4 - Complex decision-making and insights, the goal of which is to discover hidden correlations by detailed data analysis and advanced visualisations and direct output of T4.4. However, T4.2, T3.3 as well as the whole WP6 where the real-life experiments are being performed are of key importance since they feed the visualisations (T3.3 and T4.2) and evaluate them (WP6) respectively. Therefore, this KPI will start being closely monitored after the MVP release, where end-users will have the opportunity to experiment with the DMT, however a few hypotheses regarding where to find hidden correlations are already being shaped. For example, we are considering using visualisations:

- to compare the results among the outputs of different tools and algorithms,
- to investigate the role of time and weather in the appearance of traffic anomalies as well as in the appearance of unusual events in the use cases related to public safety,
- to investigate what happens at the same period of time across different parts of the city based on historical data.

More areas for investigation and use of visualisations in order to find hidden correlation will be identified as we use the tool and further develop it to address all the pilot use cases instead of just the one we explored in the context of the MVP. Progress will be checked every three months, in order to be able to judge whether to insist on investigating an area or change direction and try a different hypothesis.

9.1.6. Expected future results

SmartViz is the key component of interaction with the user, contributing to the creation of the MARVEL DMT. By the end of the project, we expect it to support all functionalities envisioned for the DMT, namely:

- allowing the user to perform multi-layered queries related to data collected across various timeframes, types of objects, and different types of sources;
- reveal hidden insights facilitated by the advanced interactive visualisations of data;
- demonstrate text-annotated attention maps which will enhance video streams with textual information and indications of associated audio events;
- showcase multisource, multimodal summaries, that allow users to explore and understand AV, sensor, and other context-enriching data (e.g., weather data, information from incident reporting systems, parking sensors, etc.) and interact with them;
- Real-time visualisations of alerts and detected events for short-term decisions and monitoring, supported by a rule-based engine.

As already mentioned in paragraph 9.1.5, all functionalities will be benchmarked with the current experience of the users in terms of usability, availability, and reliability of the tools and processes they are used to. ZELUS aims to have more than 85% of users with a positive perception regarding the usability and overall user experience. Moreover, our target is to have no downtime during the operation of the tool therefore flawless and continuous availability as well as no errors in responsiveness or in delivering faulty information, therefore 100% trustworthiness and reliability. In terms of scalability and performance as soon as more data become available, ZELUS plans to be supporting Big Data without errors that hinder the usability and performance of the interface, allowing max 8-10 seconds response time.

9.2. MARVEL Data Corpus-as-a-Service

9.2.1. Role in the MVP

MARVEL Data Corpus is the component where processed multimodal audio-visual data are stored and are obtained free of charge and released as a service. In the scope of MVP, MARVEL Data Corpus plays roles in scenarios 2 and 3. More specifically:

- Scenario 2: Pre-recorded video snippets (~5min each) from Malta (GRN) pilot are stored inside the Data Corpus and are accessible through the relative REST API calls.
- Scenario 3: MARVEL Data Corpus at the current stage stores several datasets from each of the three pilot sites: Trento (MT/FBK), Malta (GRN), and Novi Sad (UNS), with the majority of data originating from Malta (GRN), as the MVP pilot provider. All data are accessible through the relative REST API calls.

9.2.2. Current status

MARVEL Data Corpus has been deployed as the distributed dockerized environment inside MARVdash. The deployment is based on an Hbase and HDFS system in which the current setup includes one DataNode (which is responsible for serving read and write requests from the file system's clients), one NameNode (which maintains and manages the blocks present on the DataNodes), and one region server (which is responsible for several things, such as handling, managing, executing as well as read and write HBase operations).

9.2.3. State-of-the-art baseline

Most of the SotA baselines focus on measurements regarding the network performance of the data transfer process.

9.2.4. Description of the benchmarking process/assessment strategy

9.2.4.1. *Assessment strategy*

The assessment strategy for the MVP focuses on measuring the average performance metrics of data ingestion to the Corpus by performing a series of ingestion requests in parallel using the HDFS rest calls of the Corpus.

9.2.4.2. *Benchmarks used*

The benchmarks used for this iteration focus on the performance metrics, including parameters such as loss rate and service availability per request, as part of the Hadoop Benchmark Suite (HiBench)[21] and specially the enhanced DFSIO (dfsioe) HDFS Benchmark.

9.2.4.3. *Infrastructure for testing*

The current infrastructure for performing the benchmarks consists of one VM under MARVdash that contains the distributed setup of Data Corpus. This container has 16 AMD CPU cores running at 2299.998 MHz with 32 GB of RAM and a 50 GB hard disk for data storage. The benchmarks have been also tested on an exactly similar VM under STS premises. On both systems, these settings can be considered sufficient for the MVP purposes, in terms of benchmarking, because they exceed the minimum development requirements of the HBase and Hadoop.

[21] https://github.com/hibench/HiBench-2.1

9.2.4.4. *Metrics*

The metrics that were measured are the average data loss rate, the service availability per request, the average data transfer latency and throughput, and the average request response time.

9.2.4.5. *Results of the measurements*

Table 8 summarises the results of the measurements.

Table 8. Results of the measurements for MARVEL Data Corpus-as-a-Service component

Metric	Value
Data loss rate	None
Service availability-failed request	None
Data access restriction	Accessing the HDFS with no access restriction in place (http)
Data transfer latency	~15ms (average)
Data throughput	~120MB/s (average)
Response time	~0.7ms (average)

9.2.4.6. *Results observations*

On several occasions, the data transfer latency may increase due to the normal network round trip time (RTT). Under this consideration, we also have to include the internal work that HBase has to do to retrieve or write the data to the HDFS file system. Overall, Data Corpus performs as expected.

9.2.5. Contribution to MARVEL KPIs

KPI-O5-E1-1: More than 3.3PB of data were made available through a Corpus-as-a-Service. The choice of developing the MARVEL Data Corpus on the basis of Hbase and Hadoop technologies affects the latter KPI as these technologies are optimised for handling/managing large-scale data. By using a distributed file system such as the Hadoop Distributed File System, the data is split into chunks and saved across clusters of commodity servers. As these servers are built with simple hardware configurations, they are inexpensive and easily scalable as the data grows. Hadoop also uses the MapReduce functional programming model to perform parallel processing across datasets. So, when a query is sent to the database, instead of handling data sequentially, tasks are split and executed concurrently across distributed servers. Finally, the output of all tasks is collated and sent back to the application, drastically improving the processing speed.

9.2.6. Expected future results

The MARVEL Data Corpus will be expanded in future versions to include a GUI to manage data ingestion and data augmentation, respectively. Furthermore, it will include the development of a streaming protocol to accept data streams for the pilot side. Additionally, it will include a search/query mechanism to query stored according to user needs. For all these new developments and features, a new series of benchmarks must be performed; however, we do not expect serious variation in terms of networks performance metrics.

10. Summary of the benchmarks

Table 9 summarises the benchmarks for each component in the MVP.

Table 9. Summary of the benchmarks

Component	Benchmarks	Metrics	Observations	Future improvements
GRNEdge	Workloads from the MVP	Latency, Drift, Number of lost frames, Audio-video synchronisation delay, and Throughput	Suitable for short term use. Overheating could prevent the setup from being used for long term data collection.	Incorporate more off-the-shelf devices to increase sustained performance in different weather conditions.
CATFlow	MSCOCO and MIO-TCD datasets	Precision and recall using the IOU threshold, F-score and mAp	Lowest mAp value is when detecting pedestrians which signifies that more effort needs to be placed in that area.	Improve pedestrian, bicycle, and motorcycle detection. Include metrics for the tracking algorithm.
MEMS	Workloads from the MVP	Signal-to-Noise ratio, Total harmonic distortion	Microphones feature SotA performance. Casing affects the quality of the recording. A larger number of microphones is required to perform complex sound analytics.	Casing and audio hole placement will be optimised. A new board with more microphones (up to 8) and edge audio processing capabilities is being designed.
SED@Edge	MAVD-traffic, UrbanSound8K and GRN datasets	Accuracy, MMAC	Expected model compression achieved, performance still a bit below target on some benchmarks.	Improve the accuracy working on the architecture or training strategies.
VideoAnony	CelebA-HQ dataset	Fréchet Inception Distance, Percentage of faces being detected after swapping, Percentage of swapped faces being matched to its original face, Normalised mean error of the detected landmarks	Improved FID of the swapped faces (face naturalness) when handling low-resolution input target faces, without comprising much of the other metrics.	To compress the model by reducing its complexity by at least 10%.
DFB	Synthetic data	Data Integrity, Scalability, Availability, and Performance for	Tested in isolation and performs as expected. Observed as high performant in terms of	DFB will leverage its scaling (clustering) capabilities. It will be measured against the baseline evaluation of

		high volume, heterogeneous data stream	availability, speed, and resilience.	this phase and is expected to perform well under increasing amount/rate of input data.
DatAna	Data workloads from the MVP	Latency, and Throughput	Only benchmarked in a minimal setting of one node in the cloud.	Deployment in fog and edge, and improvements in scalability and throughput.
AVCC	DISCO dataset	Mean absolute error and Mean squared error	Performance at the SotA levels (improved for more efficient version with early exits). Not yet benchmarked with MVP data.	System benchmarking and optimisation with MARVEL datasets.
SED	MAVD-traffic dataset	F1-score and error rate (in one-second segment)	Performance close to the SotA. Not yet benchmarked with MVP data.	System benchmarking and optimisation with MARVEL datasets.
MARVdash	User Experience Questionnaire	Attractiveness, Perspicuity, Efficiency, Dependability, Stimulation, and Novelty	The main functionalities of MARVdash have rated it with high average scores. User Experience metrics are above average compared with other products in the UEQ database.	FORTH aims to tackle any shortages in its functionality (resource allocation, deployment at execution environments) and correct any flaws in its design (user experience) based on the benchmarking results.
GRN edge infrastructure	Workloads from the MVP	Robustness, Reliability, Throughput	Reliable and robust in most conditions.	GRN edge infrastructure could include the addition of edge processing.
GRN fog infrastructure	YOLO4 algorithm	Frames per second the server can handle, Amount of GPU memory	Results give an indication of what processing power is needed at the edge and the volume of real-time processing that the fog layer can handle.	The same requirements are necessary for the fog infrastructure.
PSNC cloud infrastructure	Operating system check, Logical check	Availability	More elaborate aggregation algorithm taking into account metrics critical from point of view MARVEL should be created to better reflect the “real” availability of the cloud environment.	Monitoring should take into account components that are not yet integrated, such as data store for corpus service.

SmartViz	User Experience Questionnaire	Usability, Availability, and Reliability	Performance very close to expectations.	Eliminate availability issues, add usability tests, measure scalability and performance once MARVEL prototype is ready & more functionalities are available.
MARVEL Data Corpus-as-a-Service	Parallel ingestion requests	Loss rate and service availability per request, Average data transfer latency, Throughput and Average response time	At this stage, Data Corpus performs as expected. Data transfer latency may increase due to normal network round trip time (RTT) plus internal work that HBase has to do to retrieve or write the data.	Usage of more DataNodes will improve the service availability per request and expected to serve also the average transfer latency.

11. Conclusions

In this document, we presented in detail the technical evaluation and progress against benchmarks in the MARVEL project. This is the initial version of the document, and it deals with benchmarking in the MVP stage only. The final version of the benchmarking document will be prepared by the end of the project (M30), and it will contain benchmarking of the full MARVEL framework. The benchmarking strategy defined in WP1 (reported in D1.2) was implemented in this document for each component involved in the MVP. For each component, the role in the MVP was defined together with the information about the current status of the component. The state-of-the-art baseline was defined, and benchmarking process, measurement results, and result observations was presented. Lastly, the contribution to MARVEL KPI's was described and expected future results were discussed.

Results from this document (D5.2) together with the final version (D5.5) will be used within *'WP6 – Real-life societal experiments in smart cities environments'*. In the 'Task 6.3 – *Evaluation and Impact analysis'* (task active M12-M36), KPIs are monitored and evaluated for each experiment in operational terms as well as in technical terms, and benchmarking results will be essential part of this process.

12. References

Abeßer, J. et al., 2021. *IDMT-Traffic: An Open Benchmark Dataset for Acoustic Traffic Monitoring Research.* s.l., EUSIPCO.

Bangor, A., Kortum, P. T. & Miller, J. T., 2008. An empirical evaluation of the system usability scale. *International Journal of Human–Computer Interaction,* 24(6), pp. 574-594.

Bochkovskiy, A., Wang, C.-Y. & Liao, H.-Y. M., 2020. *YOLOv4: Optimal Speed and Accuracy of Object Detection,* s.l.: arXiv preprint arXiv:2004.10934.

Bradski, G., 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools.*

Chen, R., Chen, X., Ni, B. & Ge, Y., 2020. *SimSwap: An Efficient Framework For High Fidelity Face Swapping.* s.l., s.n., pp. 2003-2011.

DataBench / D4.3, 2020. *D4.3 Evaluation of Business Performance,* s.l.: DataBench Project.

Gemmeke, J. F. et al., 2017. *Audio Set: An ontology and human-labeled dataset for audio events.* s.l., s.n., pp. 776-780.

Heusel, M. et al., 2017. *GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium.* s.l., s.n., pp. 6629-6640.

Hinderks, A., Schrepp, M. & Thomaschewski, J., 2018. *A Benchmark for the Short Version of the User Experience Questionnaire.* s.l., s.n., pp. 373-377.

Hu, D. et al., 2020. *Ambient Sound Helps: Audiovisual Crowd Counting in Extreme Conditions,* s.l.: arXiv: 2005.07097.

ISO 9241-210:2019, I. O. f. S., 2019. “Ergonomics of human-system interaction -- Part 210: Human-centred design for interactive systems. July.

Karras, T., Aila, T., Laine, S. & Lehtinen, J., 2018. *Progressive Growing of GANs for Improved Quality, Stability, and Variation.* s.l., s.n.

Kazemi, V. & Sullivan, J., 2014. *One Millisecond face alignment with an ensemble of regression trees.* s.l., s.n., pp. 1867-1874.

Lewis, J. R. & Sauro, J., 2009. *The Factor Structure of the System Usability Scale.* s.l., Springer Berlin Heidelberg, pp. 94-103.

Li, L. et al., 2020. *FaceShifter: Towards High Fidelity And Occlusion Aware Face Swapping.* s.l., s.n.

Lin, T.-Y.et al., 2014. *Microsoft COCO: Common Objects in Context.* Cham, Springer International Publishing, pp. 740-755.

Luo, Z. et al., 2018. MIO-TCD: A New Benchmark Dataset for Vehicle Classification and Localization. *IEEE Transactions on Image Processing,* 27(10), pp. 5129-5141.

MARVEL / D1.2, 2021. *D1.2 MARVEL's Experimental Protocol,* s.l.: Project MARVEL.

MARVEL / D1.3, 2021. *D1.3 Architecture definition for MARVEL framework,* s.l.: Project MARVEL.

Maximov, M., Elezi, I. & Leal-Taixe, L., 2020. *CIAGAN: Conditional Identity Anonymization Generative Adversarial Networks.* s.l., s.n., pp. 5447-5456.

Mesaros, A., Heittola, T. & Virtanen, T., 2016. Metrics for Polyphonic Sound Event Detection. *Applied Sciences,* Volume 6, p. 162.

Paissan, F., Ancilotto, A., Brutti, A. & Farella, E., 2022. *Scalable Neural Architectures for End-to-End Environmental Sound Classification.* s.l., s.n.

Paissan, F., Ancilotto, A. & Farella, E., 2021. PhiNets: a scalable backbone for low-power AI at the edge. *arXiv:2110.00337.*

Qiugiang, K. et al., 2020. PANNs: Large-Scale Pretrained Audio Neural Networks for Audio Pattern Recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* Volume 28, pp. 2880-2894.

Redmon, J., Divvala, S., Girshick, R. & Farhadi, A., 2016. *You Only Look Once: Unified, Real-Time Object Detection.* s.l., s.n., pp. 779-788.

Rezatofighi, H. et al., 2019. *Generalized Intersection over Union.* s.l., s.n.

Salamon, J., Jacoby, C. & Bello, J. P., 2014. *A Dataset and Taxonomy for Urban Sound Research.* s.l., s.n., pp. 1041-1044.

Salamon, J. et al., 2017. *Scaper: A library for soundscape synthesis and augmentation.* s.l., IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), pp. 344-348.

Schrepp, M., Thomaschewski, J. & Hinderks, A., 2017. Construction of a Benchmark for the User Experience Questionnaire (UEQ). *International Journal of Interactive Multimedia and Artificial Intelligence,* 4(4), pp. 40-44.

Wang, Y. et al., 2021. *HifiFace: 3D Shape and Semantic Prior Guided High Fidelity Face Swapping.* s.l., s.n., pp. 1136-1142.

Zinemanas, P., Cancela, P. & Rocamora, M., 2019. *MAVD: A Dataset for Sound Event Detection in Urban Environments.* New York University, NY, USA, Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019), pp. 263-267.