Horizon 2020
European Union funding
for Research & Innovation

Big Data technologies and extreme-scale analytics

Multimodal Extreme Scale Data Analytics for Smart Cities Environments

D6.1: Demonstrators execution - initial version[†]

Abstract: This deliverable provides a detailed description of the experiments' implementation following the line of D1.2 and taking into consideration the progress of the work done from month 10 (October 2021) to month 18 (June 2022) concerning the experimental protocol alignment of Task 6.1 and the configuration of the framework and execution of real-life societal use cases of Task 6.2. Each use case is described in terms of integration and deployment, configuration of the framework and execution of real-life societal experiments, as well as demonstration by reporting the experimental indicators and associated metrics for all experiments (Task 6.3).

Contractual Date of Delivery	30/06/2022
Actual Date of Delivery	19/07/2022
Deliverable Security Class	Public
Editor	*Thomas Festi (MT)*
Contributors	All *MARVEL* partners
Quality Assurance	*Adrian Muscat (GRN)* *Pawel Bratek (PSNC)* *Nikola Simic (UNS)*

[†] *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. *Adrian Muscat, (GRN)*
2. *Pawel Bratek, (PSNC)*
3. *Nikola Simic (UNS)*

Revisions

Version	Date	By	Overview
1.0.4	19/07/2022	Thomas Festi	Addressing final comments from PC
1.0.3	15/07/2022	Thomas Festi	Final draft submitted to PC for quality check
1.0.2	11/07/2022	Thomas Festi Alessio Brutti Adrian Muscat, (GRN) Pawel Bratek, (PSNC) Nikola Simić (UNS)	Include all partners contributions Comments on pre-final draft and approval
1.0.1	03/07/2022	Thomas Festi Dragana Bajovic	Revision on the deliverable 1st draft
1.0.0	20/06/2022	Thomas Festi	1st draft of the Deliverable
0.0.6	10/06/2022	Thomas Festi ATOS	Final ToC Incorporating ATOS suggestions
0.0.5	10/06/2022	Thomas Festi	ToC -4th draft
0.0.4	18/05/2022	Thomas Festi	ToC – 3rd draft
0.0.3	16/05/202	Thomas Festi Alessio Brutti	Revision on the ToC – 2nd draft
0.0.2	05/05/2022	Thomas Festi Alessio Brutti	Revision on the ToC – 2nd draft.
0.0.1	22/04/2022	Thomas Festi, Dragana Bajovic	Comments on the ToC.
0.0.0	01/04/2022	Thomas Festi	ToC – 1st draft.

Disclaimer

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

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

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AAC	Automated Audio Captioning
AI	Artificial Intelligence
API	Application Programming Interface
ARM	Advanced RISC Machines
ASC	Acoustic Scene Classification
AudioAnony	GANs for audio anonymisation
AV	Audio-Visual
AVAD	Audio-Visual Anomaly Detection
AVCC	Audio-Visual Crowd Counting
AVDrone	Drone-based Audio-Visual data collection
AT	Audio tagging
AUC	Area Under the ROC Curve
CATFlow	Data Acquisition Framework
CCTV	Closed-Circuit Television
CIA	Confidentiality, Integrity, and Availability
CI/CD	Continuous Integration / Continuous Delivery
CLI	Command-Line Interface
COVID-19	Coronavirus Disease – 2019
CPU	Central Processing Unit
D#.#	Deliverable #.#
DatAna	Data Acquisition Framework
DFB	Data Fusion Bus
DISCO	auDiovISual Crowd cOunting dataset
DL	Deep Learning
DMT	Decision Making Toolkit
DNN	Deep Neural Network
DNS	Domain Name System
DPIA	Data Protection Impact Assessment
DPO	Data Protection Officer
DT#	Dataset
DynHP	Compressed Models
E2F2C	Edge to Fog to Cloud
EAB	External Advisory Board

EC	European Commission
EdgeSec	Security Services at the edge
ELAN	EUDICO Linguistic Annotator
FedL	Framework and implementation of ML algorithms – Federated learning
FL	Federated Learning
FLOPS	Floating Point Operations Per Seconds
FPS	Frames Per Second
GA	Grant Agreement
GAN	Generative Adversarial Network
GB	Gigabyte
GDPR	General Data Protection Regulation
GPS	Global Positioning System
GPU	Graphics Processing Unit
GPURegex	GPU Pattern Matching Framework
GUI	Graphical User Interface
H2020	Horizon 2020 Programme
HDD	Hierarchical Data Distribution
HDFS	Hadoop Distributed File System
HPC	High Performance Computing
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
HW	Hardware
ICT	Information and Communication Technology
IoT	Internet of Things
IP	Internet Protocol
IT	Information Technology
JSON	JavaScript Object Notation
Karvdash	Kubernetes CARV dashboard
KPI	Key Performance Indicator
M#	Month #
mAP	mean Average Precision
MAE	Mean Absolute Error
MAVD	Monte-Video Audio-Video Data
MB	Megabyte
MEMS	Micro Electro-Mechanical Systems

ML	Machine Learning
MP3	Moving Picture Experts Group Layer-3 Audio
MP4	MPEG-4 Part 14 digital multimedia container format
MPEG	Moving Picture Experts Group
MQTT	Message Queuing Telemetry Transport
MVP	Minimum Viable Product
NAT	Network Address Translation
NFS	Network File System
O#	Objective #
OEI	Other Ethics Issues
openSMILE	open-source Speech and Music Interpretation by Large-space Extraction
OS	Operating System
PC	Project Coordinator
POE	Power Over Ethernet
POPD	Protection of Personal Data
R#	Release
R	Report
RAID	Redundant Array of Independent Disks
RAM	Random Access Memory
REST	REpresentational State Transfer
RISC	Reduced Instruction Set Computing
RPi	Raspberry Pi
RTSP	Real-time Streaming Protocol
S2S	Site-to-Site
S3	Simple Storage Service
SD	Secure Digital
sec	second
SED	Sound Event Detection
SED@Edge	Sound Event Detection at the Edge
SELD	Sound Event Localisation and Detection
SET	Sound Event Tagging
SmartViz	Advanced Visualisation Toolkit
SOTA	State-of-the-Art
SSD	Solid State Drive
SSL	Secure Sockets Layer

T#.#	Task #.#
TAD	Text Anomaly Detection
TLS	Transport Layer Security
TOC	Table of Contents
TXT	Plain Text
UC#	Use Case
UCSD	User Centered System Design
UI	User Interface
URL	Uniform Resource Locator
USB	Universal Serial Bus
UX	User experience
VAD	Voice Activity Detection
VCC	Visual Crowd Counting
ViAD	Visual Anomaly Detection
VideoAnony	GANs for video anonymisation
VM	Virtual Machine
VPN	Virtual Private Network
WAV	Waveform Audio File Format
WiFi	Wireless Fidelity
WP#	Work Package #
Y#	Year #

Executive Summary

MARVEL aspires the convergence of a set of technologies in the areas of AI, analytics, multimodal perception, software engineering, HPC as part of an Edge-Fog-Cloud Computing Continuum paradigm, to support data-driven real-time application workflows and decision-making in modern cities, showcasing the potential to address in an effective way societal challenges.

This deliverable is the baseline of the MARVEL framework for the implementation of the real-life societal experiments in smart city environments (from month 10 to month 36).

In particular, it reports on the progress of the experimental protocol alignment as well as on the configuration of the framework and its execution in real-life societal experiments in each use case selected for implementation for month 18. In addition, it provides a detailed list of indicators to be measured during the experiments in order to validate the technical, functional and non-functional performance of the MARVEL platform.

Special emphasis is placed on ensuring the alignment of the operational experiments to foster innovation in audio-visual analytics and sound recognition as well as on addressing societal and industrial requirements in the smart city domain expressed in the context of the defined use cases.

Finally, it provides a highly detailed report of the implementation, problems encountered and solutions developed with respect to the expected project innovations and achievements, as well as an overview of the current and prospective activities. The operational experiments' specifications reported in this deliverable will be further refined and adjusted during the experimentation phase to reflect data availability and platform functionality.

1 Introduction

1.1 Purpose and scope of this document

This deliverable presents the initial results of WP6 – Real-life societal experiments in smart cities environments. In particular, it describes the experimental protocol alignment process aiming to ensure a smooth and adequate running of the experiments according to the experimental protocol. Furthermore, it reports on the progress of preparatory actions for the experiments (execution time plans, evaluation scenarios and selection of framework tools to be tested) and on the drivers for adapting the system modules for the trial execution.

In addition, it provides an updated specification of the execution of the experiments (through an iterative process), which will take place over the entire time frame of WP6. Early versions of the MARVEL framework have limited functionality, while further testing/validation will be required as more functionality is added.

Finally, it presents the outputs from all real-life smart city experiments that have been conducted. The outputs will be analysed to determine the efficiency, operability, usability, robustness, performance, accountability, transparency, and privacy awareness of the framework.

1.2 Contribution to WP6 and project objectives

This document is the main output of Task 6.1 – Experimental protocol alignment and Task 6.2 – Configuration of the framework and execution of real-life societal experiments, as well as the first overview of the work done under Task 6.3 – Evaluation and Impact analysis.

The deliverable is directly related to the achievement of MARVEL Objective 4: *"Realise societal opportunities in a smart city environment by validating tools and techniques in real-world settings"*. As modern cities face numerous societal challenges, it is again critical for technologies like Big Data, IoT and Edge/Fog/Cloud computing to offer solutions that increase citizens' wellness and wellbeing. However, it is critical to encapsulate the complexity of a city and support accurate, cross-scale and in-time predictions across different application scenarios.

This involves:

- ensuring smooth and adequate execution of the experiments according to the experimental protocol (T6.1 and T6.2);
- demonstrating how the MARVEL framework can quickly and effectively aggregate, process and visualise extremely-large-scale audio-visual data at the edge, fog, and cloud (T6.1 and T6.2);
- fostering innovation in audio-visual analytics and sound recognition and addressing societal and industrial requirements in the smart city domain (T6.1, T6.2, and T6.3);
- providing structured feedback, both from the data providers and the technology owners to the development process (T6.3);
- assessing project impact to drive actions for the framework's long-term sustainability (T6.3 and related to T7.5).

Finally, the work done under the WP6 – Real-life societal experiments in smart cities environment contributes to achieving the following project-related KPIs:

- KPI-O4-E1-1: More than 10 trial cases to showcase framework's capabilities.
- KPI-O4-E2-1: Identify at least 20 dependent and independent verification and validation variables for the system.
- KPI-O4-E3-1: Execute the trial cases in at least two real life smart city environments.

1.3 Relation to other WPs and deliverables

This deliverable synthesises the initial results of all WP6 activities, and as such, there is a close interrelation between this deliverable and all the tasks in the current work package.

Furthermore, there is also a close dependency between this deliverable and all other work packages:

- WP1: Setting the scene: Project set up:
  - T1.1: The critical role of multimodal analytics in addressing societal challenges, T1.2: Extreme-scale multimodal analytics: progress beyond the state-of-the art and D1.1 - Project set-up;
  - T1.3: Experimental protocol - real life societal trial cases in smart cities environments and D1.2 – MARVEL's experimental protocol;
  - T1.4: Technology convergence: specifications and E2F2C distributed architecture and D1.3 – Architecture definition for MARVEL framework.
- WP2: MARVEL multimodal data Corpus-as-a-Service for smart cities:
  - T2.1: Collection and analysis of MARVEL experimental distributed data assets and D2.1 – Collection and analysis of experimental data;
  - T2.2: Data management and distribution and D2.2 – Management and distribution Toolkit – initial version;
  - T2.3: Incremental scheme: continuous augmentation of the dataset.
- WP3: AI-based distributed algorithms for multimodal perception and situational awareness:
  - T3.1: AI-based methods for audio-visual data privacy;
  - T3.2: MARVEL's personalised federated learning realisation for extreme-scale analytics;
  - T3.3: Multimodal audio-visual intelligence and D3.1 – Multimodal and privacy-aware audio-visual intelligence – initial version;
  - T3.4: Adaptive E2F2C distribution and optimisation of AI tasks, T3.5: Edge-optimal ML/DL deployment for multimodal processing and D3.2 – Efficient deployment of AI-optimised ML/DL models – initial version.
- WP4: MARVEL E2F2C distributed ubiquitous computing framework:
  - T4.1: Optimised audio capturing through MEMS devices and T4.2: openSMILE platform for audio-visual analysis and voice anonymisation and D4.1 – Optimal audio-visual capturing, analysis and voice anonymisation – initial version;
  - T4.3: Security and acceleration in the complete E2F2C and D4.2 – Security assurance and acceleration in E2F2C framework – initial version;
  - T4.4: MARVEL's decision-making toolkit and D4.3 – MARVEL's decision-making toolkit – initial version.
- WP5: Infrastructure Management and Integration, mainly regarding:
  - T5.1: HPC infrastructure, T5.2: Resource management and optimized automatic usage of external computational and storage resources and D5.3 – HPC infrastructure and resource management for audio-visual data analytics – initial version;
  - T5.3: Continuous integration towards MARVEL's framework realisation, D5.1 – MARVEL Minimum Viable Product and D5.4 – MARVEL Integrated framework – initial version.

1.4 Structure of the document

The structure of this document is as follows:

- Section 2 – The experimental protocol definition is revisited, considering: the outcome of the MVP deployment at M12 in terms of architecture, component interaction; any new insight derived from the data collections; hardware procurement and deployment;
- Section 3 – The aspects that characterise all the use cases selected for R1 are described, specifically: the Edge-to-Fog-to-Cloud (E2F2C) infrastructure developed; the inference pipeline conceived; the batch processing pipelines for system optimisation created; the process implemented in order to guarantee ethics and privacy assurance and to anonymise the data collected; the realisation of the Data Corpus and the integration process overview;
- Section 4, 5, 6, 7 and 8 – The integration and framework configuration is presented; the multimodal and privacy-aware intelligence applied; the analysis of the outputs from real-life smart city experiments and the demonstration of the implementation realised for GRN3: Traffic Conditions and Anomalous Events (Section 4); GRN4: Junction Traffic Trajectory Collection (Section 5); MT1: Monitoring of Crowded Areas (Section 6); MT3: Monitoring of Parking Places (Section 7) and UNS1: Drone Experiment (Section 8);
- Section 9 – The summary and conclusions of the document are included.

2 Experimental protocol alignment

2.1 Rationale

Deliverable D1.2, released at M8, provided a first definition of the experimental protocol and the benchmarking strategies for the MARVEL project, considering the framework as a whole as well as each individual component. D1.2 was organised in two directions:

1. **Use cases.** The overall project is designed around the use cases provided by the pilots; therefore, a relevant part of the experimental protocol definition was devoted to identifying and defining ten use cases. In particular, task T1.3 was focused on:
   - definition of each use case in terms of goals, location, hardware, and societal challenges to be addressed;
   - definition of the experimental subjects, internal and external to the consortium, involved in both the experimentation and the use cases assessment;
   - definition of the functional and non-functional evaluation parameters for the use cases with related KPIs;
   - identification of the datasets that will be made available for the prototype development;
   - definition of an execution plan based on the project development phases, as described in the GA.
2. **Technological components.** In order to monitor the progress of all components and assets employed in MARVEL, D1.2 also defined KPIs for each technological component/asset of the project, identifying metrics, datasets, benchmarks, and KPIs.

The results of these two parts led in D1.2 to the definition of the benchmarking strategies, which are addressed and finalised in D5.2[1] and D5.4.

2.1.1 Scope of task T6.1

The goal of the experimental protocol alignment carried out under T6.1 is to revisit the experimental protocol definition considering:

- the outcome of the MVP deployment at M12 in terms of architecture and component interaction;
- any new insight derived from the data collections;
- and the hardware purchasing and deployment.

In addition, within T6.1, pilots were expected to finalise the definition of the use cases scheduled for deployment at M18. In particular, the goals of T6.1, as per GA, are:

- to ensure the efficient execution of the experiments;
- to provide more details about the execution time plans;
- to provide a better definition of the evaluation scenarios;
- to perform the final selection of framework tools to be tested;
- to drive the adaptation of the system modules for the trials' execution.

This section reports the results of the efforts devoted under T6.1 focusing, in particular, on the use cases and components involved in the R1 prototype deployment.

As a result of the experimental protocol alignment:

- two new components were introduced with related KPIs;

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

- seven asset KPIs were revised;
- three use case KPIs were redefined;
- three non-functional use case requirements were slightly revised.

Note that for what concerns the “final selection of framework tools to be tested”, the results of this effort are reported in detail in each use case architecture (both in this deliverable and in D5.4).

2.1.2 Alignment process

The activities carried out for the experimental protocol alignment were highly entangled with efforts devoted under WP3, WP4 and WP5 for the architecture definition and for the use case deployment, in particular, for what concerns the use cases finalisation. All the use cases related issues were addressed in a series of weekly focused meetings organised under WP6 and reported in a living technical report.

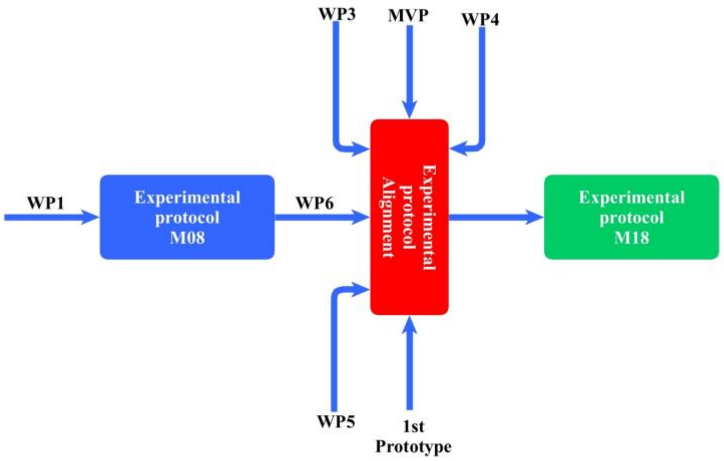

WP3
MVP
WP4
WP1
Experimental protocol M08
WP6
Experimental protocol Alignment
Experimental protocol M18
WP5
1st Prototype

Figure 1: Experimental protocol alignment

2.2 Update and Revision of the execution plan

One of the goals of T6.1 is the revision of the execution plan defined a year earlier in D1.2, in particular for the time span covering the final part of the project, which was inevitably less accurate a year earlier. The next sections provide the revised execution plans for each pilot.

2.2.1 Green Roads time plan

Table 1 reports the updated time plan for GRN use cases.

Table 1: Green Roads time plan

Phase	Activity	Period	Status
Baseline	Definition of the use cases	M6	Completed
	First data release	M8	Completed
Innovation	HW procurement	M8-M12	Completed
	HW computing devices installation	M10 – M22	Ongoing
	HW sensing devices installation	M10 – M22	Ongoing
	First release of the MARVEL’s infrastructure	M12	Completed

	Minimum Viable Product	M12	Completed
	MARVEL prototype 1st versions	M18	Ongoing
Experimentation	Internal Evaluation of the R1	M22	Not Started
	Revision of the data flow	M18-M24	Not Started
	Execute field trials	M20-M30	Not Started
Consolidation	Release the multimodal audio-visual corpus	M22-M30	Not Started
	Fix issues	M22-M30	Not Started
	Develop a business plan	M22-M30	Not Started

With reference to the activity "HW procurement" in the "Innovation" row, all equipment needed has been purchased, but not all of it has been installed. The focus was given to installing the sensors and computational devices required for the R1 integration. The remainder of the hardware will be installed by M20. Some sensing devices, namely microphones, will be installed upon delivery by IFAG.

From M18 onwards, the first task will be testing and installing purchased hardware such that these can be added to the MARVEL framework during the next integration. The data flow and user stories for GRN1 and GRN2 will be developed and discussed with the rest of the partners. In addition, the data flow for GRN3 and GRN4 will be revised if found to be necessary during the R1 integration internal evaluation. In the second half of the project, tests to confirm the functionality and reliability of the infrastructure and data flow will be performed as part of the field trials. Issues found will then be addressed in future integrations.

2.2.2 Municipality of Trento time plan

Table 2 reports the updated time plan for the MT pilots.

Table 2: Municipality of Trento time plan

Phase	Activity	Period	Status
Baseline	Definition of the use cases	M6	Completed
	First data release	M8-M12	Completed
Innovation	HW procurement	M8-M22	Ongoing
	HW computing devices installation	M15 – M22	Ongoing
	HW sensing devices installation	M15 – M22	Ongoing
	First release of the MARVEL's infrastructure	M18	Ongoing
	Minimum Viable Product	M12	Not Started
	MARVEL prototype 1st versions	M18	Ongoing
Experimentation	Internal Evaluation of the R1	M22	Not Started
	Revision of the data flow	M18-M24	Not Started
	Execute field trials	M20-M30	Not Started
Consolidation	Release the multimodal audio-visual corpus	M22-M30	Not Started
	Fix issues	M22-M30	Not Started

	Develop a business plan	M22-M30	Not Started

With reference to the activity “HW procurement” in the “Innovation” row, other equipment needed to improve the MARVEL framework (e.g., run more AI models at the edge) will be purchased and all of it will be installed for the R2 phase.

Regarding the activity “HW sensing devices installation” and “HW computing devices installation” in the “Innovation” row, the focus was given to installing the sensors and computational devices required for the R1 integration. Specifically, the microphones and Raspberry Pi devices needed for MT3 are not at the definitive installation stage; they first need to be tested because once they are installed it will be hard to physically gain access to these devices.

The remainder of the hardware will be installed by M20; some sensing devices, namely microphones, will be installed when delivered by the partners from IFAG.

From M18 onwards, the data flow and user stories for MT2 and MT4 will be developed and discussed with the rest of the partners, and the data flow for MT1 and MT3 will be improved if necessary. Issues found will then be addressed in future integrations.

2.2.3 Novi Sad time plan

Table 3 reports the updated execution plan for the UNS use cases.

Table 3: Novi Sad time plan

Phase	Activity	Period	Status
Baseline	Definition of the use cases	M8	Completed
	HW testing	M7	Completed
Innovation	Recording app development	M10	Completed
	Databases recording	M22	Ongoing
	First data release	M14	Finished
Experimentation	Algorithm development	M20	Not started
	MARVEL prototype 1st version	M18	Ongoing
	Execute field trials	M18-M20	Not started
Consolidation	Public release of recorded databases	M22	Not started
	Fix issues	M22-M24	Not started

IFAG MEMS microphones received, following a delay due to customs process. Thus, the first small dataset was recorded in November 2021. However, the recording was paused during the winter period due to sensitive equipment and data augmentation resumed in the late spring of 2022. Drone-based data collection can only be performed when weather conditions are suitable (e.g., no precipitation, no wind). All these lead to the delay related to the Database recording.

2.3 Use Cases for the 1st release of the MARVEL Framework

This section describes the use cases selected for the R1 prototype deployment, highlighting changes with respect to what was reported in D1.2. Changes mainly concern:

- the final definition of the evaluation scenarios and the target events;
- the actual hardware deployment and the tools to be tested;
- the dataset made available for model training and for component evaluation.

2.3.1 GRN3: Traffic Conditions and Anomalous Events

The use case is useful to monitor traffic conditions and detect anomalous events, for example, traffic jams, accidents, cars stuck and obstructing a junction, very slow vehicles and service vehicles parked on the side or obstructing a carriageway. The latter event is frequent in Malta's narrow one-way urban streets, often causing ripple effects that extend beyond the immediate area. In general, this output would find application in for example systems intended to inform drivers near the detected anomaly or to infer possible issues in adjacent areas and inform drivers of obstacles ahead. In addition, the detection of anomalous events can be used to alert personnel stationed at traffic management control rooms, who can then interpret the data and take the necessary action.

The aim of this use case, as described in deliverable D1.2, is to detect anomalous road conditions which may be related (in a passive or active way) to obstructions. In terms of evaluation metrics, accuracy and detection time are two crucial features of the use case and these can be used to benchmark the system as follows:

- correct detection of the cause of obstructions on the road 70% of the time, as verified by manual processing.
- detection of anomalous events within 2 minutes from the onset, as verified by manual processing.

The R1 integration is designed such that both goals can be addressed. For the R1 integration, whenever an anomaly occurs, the anomaly is flagged typically at the control room such that action can be taken. In this case, when an anomaly is detected, a video clip of the anomaly is saved and could be made available to the traffic control personnel, thus gaining visual insight into the anomaly, which helps in determining the right course of the action. This feature is essential for the evaluation of both goals mentioned above since the traffic personnel would be able to flag the accuracy of the anomaly detection models. In addition, the time stamps on the video stream will give an indication of the latency with which the anomaly was detected.

The users of this system are intended to be traffic managers who can give directives to authorities to react to a traffic incident.

Execution time plan

The execution time plan up to the M18 and evaluation scenarios are reported in Table 4 for GRN3. These were the plans laid out prior to the beginning of the integration.

Table 4: GRN3 execution time plan

Phase	Activity	Period	Status
Baseline	Definition of the use cases	M6	Completed
	First data release	M8	Completed
Innovation	HW procurement	M8-M12	Completed
	HW computing devices installation *Equipment – releases in M14:* • Connection to fog, processing in real-time ○ Processing and anonymisation at the edge	M10 – M22	Ongoing

	○ Processing and anonymisation at the fog ○ Only anonymised streams reach the cloud layer ○ Fog and edge infrastructure require GPUs to be able to process streams in real-time.		
	HW sensing devices installation *Equipment – releases in M14:* • Sensors ○ Fixed camera (feeds CATFlow + anomaly detector, traffic state) ○ Fixed microphone integrated in CCTV camera • Reported in T3.5	M10 – M22	Ongoing
	First release of the MARVEL's infrastructure	M12	Completed
	Minimum Viable Product	M12	Completed
	MARVEL prototype 1st versions *Availability of Training Data to fine-tune models – releases in M16:* • GRN provided labelled data for the following models: ○ SED dataset ○ Audio Tagging Dataset ○ Anomaly Detection Dataset ○ Number Plate Annotations *User interface – releases in M15:* ○ Implemented in SmartViz ○ Requires user journey *Testing of use case – releases in M18:* • Effort to test each component deployed in pairwise tests and end-to-end tests were carried out by partners involved in the implementation of use cases. *Deployment of use case – releases in M18:* • This is a collective effort between all the partners involved in the use cases	M18	Ongoing
Experimentation	Internal Evaluation of the R1	M22	Not Started
	Revision of the data flow	M18-M24	Not Started
	Execute field trials	M20-M30	Not Started

The outline laid out by this time plan was followed. The updated execution time plan for the coming months till the completion of the R1 integration by M18 can be seen in Table 5.

Table 5: Updates to the GRN3 execution time plan

Phase	Activity	Period	Status
Innovation	MARVEL prototype 1st versions *Completion of the E2F2C integration – releases in M17:* • This task needs to be completed before progressing to the next tasks	M18	Ongoing
Experimentation	Internal Evaluation *Selection of evaluation scenarios or otherwise – releases in M17-M18:*	M17-M22	Initiated

	• This involves planning and executing scenarios to evaluate the use case • The implementation of this task depends on the completion of the integration		

Final Evaluation Scenarios

Two evaluation scenarios have been defined for this use case related to two use case KPIs as reported in Table 6.

Table 6: Evaluation scenario and relevant KPIs for the GRN3

Evaluation Scenario	Target	Relevant KPI
Testing the various AI models on a labelled dataset to determine the detection rate and F1 score achieved by the models.	The aim is to obtain a 70% detection of anomalous events.	**GRN-KPI4: Correctly detect various anomalous events on the road**
Observing the time taken to detect an anomaly through measuring the system's latency.	The aim is to detect anomalies 2 minutes after the start of the event.	**GNR-KPI5: Detection time**

Modifications to the E/F/C infrastructure

The GRN Infrastructure provided for the R1 integration of GRN3 consists of 3 IP cameras:

- one camera is at Mgarr (a rural town in the north-west region);
- the other two cameras are at Zejtun (an urban town close to the southern inner-harbour region).

GRN has deployed a PC at the Mgarr location directly connected to the Mgarr IP Camera that simulates an Edge device. GRN has also deployed a workstation with an integrated GPU as part of the Fog layer. The AV streams from the Zejtun Cameras will be processed at the Fog Layer.

Datasets

In D1.2, GRN had planned to deliver three datasets. Two of them (GRN-AV-traffic-entity and GRN-AV-traffic-state) are meant to be used to either train or fine-tune a selection of the AI models, and as such are annotated manually, whilst the third (GRN-TXT-traffic-data) is the output from the CATFlow AI model and can be used further along the AI pipeline in some of the use cases. In addition to these three pre-planned datasets, three extra datasets have been added; the annotated video number plate recognition dataset, the annotated AV anomaly detection dataset and the AVCC dataset.

GRN provided all AI providers with the necessary data required for the use cases after discussions with the partners. All these datasets are reported in Section 4.2.1.1 and Section 5.2.1.1.

For the GRN-TXT-traffic-data, the data is collected and shared in two ways and saved to the Data Corpus. The first is using the data GRN has been collecting from CATFlow throughout the project from the three static CCTV cameras. This data is more detailed than what was planned in D1.2 and as such contains extra fields, i.e., trajectories and speed of traffic entities. In addition, this data is augmented and uploaded with audio recordings from the same CCTV cameras, essentially providing a large, queryable audio dataset. The second part of this dataset will be collected during the implementation of the use cases, where inference results from all the components will be saved on the MARVEL Data Corpus.

2.3.2 GRN4: Junction Traffic Trajectory Collection

Junction Traffic Trajectory collection is focused on the requirement of long-term data analytics that shed light on both the behaviour of road users (e.g., car drivers, motorcyclists, cyclists, pedestrians, etc.) and on gathering traffic statistics at road network junctions. This use case is of interest for long-term transport planning and evaluation. In particular, there is currently significant interest in studying active travel modes, such as cycling, walking, and micro-mobility, more generally. Authorities in Malta are interested in, for example, finding the optimal position of pedestrian crossings, whether provisions for cyclists at complex junctions are adequate, and whether installed provisions are being used as intended.

This use case requires entity detection and its trajectory across a junction or road segment and descriptive statistics of network junction traffic. It, therefore, follows that entity detection and tracking models can be potentially used as a first processing stage, followed by further processing to generate descriptive statistics.

The innovation that we are targeting, as described in D1.2, with this use case is the construction of a queryable database that can be used to look up historical data on the turning ratios of vehicles and pedestrians in two complex junctions (e.g., roundabouts), with sufficient accuracy to detect anomalous patterns autonomously 50% of the time.

This goal will be partially addressed in the R1 integration. The trajectories and data generated from the CATFlow algorithm are saved on the MARVEL Data Corpus such that the data can be accessed and processed by the end user. Currently, the anomalous paths can be detected through visual inspection of the trajectories, which is a feature of the system. Future integrations will have tools that automatically detect anomalous paths. This use case was also partially implemented for the MVP in M12. The progress since the MVP implementation included the incorporation of all the components as part of the integrated system.

The users for this system are intended to be traffic engineers who need data to make informed decisions about infrastructure changes and upkeep, as well as transport researchers.

Execution time plan

The execution time plans up to the M18 evaluation scenarios are reported in Table 7 for GRN4. These were the plans laid out prior to the beginning of the integration.

Table 7: GRN4 execution time plan

Phase	Activity	Period	Status
Baseline	Definition of the use cases	M6	Completed
	First data release	M8	Completed
Innovation	HW procurement	M8-M12	Completed
	HW computing devices installation *Equipment – releases in M12:* • Connection to cloud or fog tier	M10–M22	Ongoing
	HW sensing devices installation *Equipment – releases in M12:*	M10–M22	Ongoing

	• Fixed camera and fixed microphone		
	First release of the MARVEL's infrastructure	M12	Completed
	Minimum Viable Product	M12	Completed
	MARVEL prototype 1st versions *Availability of Training Data to fine-tune models – releases in M16:* • Ongoing progress with AU • Ongoing progress with TAU Availability of annotators *User interface – releases in M15:* • Implement in SmartViz • Requires user journeys - may need some refinement *Testing of use case – releases in M17* *Deployment of use case – releases in M18*	M18	Ongoing
Experimentation	Internal Evaluation of the R1	M22	Not Started
	Revision of the data flow	M18-M24	Not Started
	Execute field trials	M20-M30	Not Started

The outline laid out by this time plan was followed. The updated execution time plan for the coming months till the completion of the R1 integration by M18 is given in Table 8.

Table 8: Updates of GRN4 execution time plan

Phase	Activity	Period	Status
Innovation	MARVEL prototype 1st versions *Completion of the E2F2C integration – releases in M17:* • This task needs to be completed before progress to the next tasks	M18	Ongoing
Experimentation	Internal Evaluation *Selection of evaluation scenarios or otherwise – releases in M17-M18:* • This involves planning and executing scenarios to evaluate the use case • The implementation of this task depends on the completion of the integration	M22	Initiated

Final Evaluation Scenarios

Table 9 describes the planned evaluation scenarios and relevant KPIs.

Table 9: Evaluation scenarios and relevant KPIs for the GRN4

Evaluation Scenario	Target	Relevant KPI
Detection of the trajectories and the storage of these trajectories to be used later in data-driven decision-making.	The aim is to have a 50% detection rate of the trajectories.	**GRN-KPI6 Availability of historical video samples of pedestrian and vehicle trajectories at two junctions.**

Surveys with relevant traffic experts to determine if this data will help in decreasing the planning time.	The aim is to have confirmation that this data is helpful.	**GRN-KPI7 Increased efficiency in the planning of roads**

Modifications to the E/F/C infrastructure

The GRN infrastructure provided for the R1 integration of GRN4 is the same as that provided for GRN3. It consists of 3 IP cameras; One camera is at Mgarr (a rural town in the north-west region) and the other two cameras are at Zejtun (an urban town close to the southern inner-harbour region. GRN has deployed a PC at the Mgarr location directly connected to the Mgarr IP Camera. The PC simulates the availability of an Edge device. GRN has also deployed a workstation with an integrated GPU as part of the Fog layer. The AV streams from the Zejtun Cameras will be processed at the Fog Layer.

Datasets

GRN provided all AI providers with the necessary data required for the use cases after discussions with them. All these datasets are reported in Section 4.2.1.1 and Section 5.2.1.1.

2.3.3 MT1: Monitoring of Crowded Areas

The goal is to select views of relevant areas for reasons such as exceptional crowd, suspect or unusual crowd movements, etc.

The area for this scenario is a square hosting the “Christmas Markets”. Every year, from November to the first day of January, in Trento, some of the main squares of the city host the “Christmas Markets” which are visited by thousands of people during the opening period. Particularly, during the weekend and holidays, these areas are highly crowded. Due to these situations, the number of robberies and aggressions can increase. In addition, first aid may be needed for people who are unwell or faint. In order to be informed of these actions in time, thanks to the cameras already installed in the square, the MARVEL framework is adopted to prevent them. The alert is sent to a control room managed by the local police.

Another place is a square hosting the weekly market located in the city centre. This is also a scenario where crowding can occur, therefore closer monitoring is necessary. This situation is more challenging due to the presence of the market operators' awnings. MARVEL framework is deployed to prevent these situations by alerting the operational centre of the local police and consequently the policeman/woman on-site or near the market.

The situations analysed will refer to robberies, aggressions, people who are unwell or faint, and gatherings, which can be found in a presence of more than four persons per square meter (or even one person per square meter if COVID-19 restriction persists) in the reference squares. The target locations are the fenced area in Piazza Fiera for “Christmas Markets” and Piazza Duomo and the proximity area for the weekly market. The visual analysis must be carried out in real-time.

Execution time plan

The execution time plans up to the M18 evaluation scenarios are reported in Table 10 for MT1. These were the plans laid out prior to the beginning of the integration.

Table 10: MT1 execution time plan

Phase	Activity	Period	Status
Baseline	Definition of the use cases	M6	Completed
	First data release • Dataset has been collected at M11, anonymised and shared with the technological partner in order to train and improve the AI models	M8-M12	Completed
Innovation	HW procurement	M8-M22	Ongoing
	HW computing devices installation	M15–M22	Ongoing
	HW sensing devices installation	M15–M22	Ongoing
	First release of MARVEL's infrastructure • Due to the GDPR constraints, MT decided at M16 that edge tier will not be part of the Kubernetes cluster.	M18	Ongoing
	Minimum Viable Product	M12	Not Started
	MARVEL prototype 1st versions • Starting from M15, annotated data has been provided to AI partners • User interface – releases in M15: ○ Implement in SmartViz ○ Requires user journeys - may need some refinement	M18	Ongoing
Experimentation	Internal Evaluation of the R1	M22	Not Started
	Revision of the data flow	M18-M24	Not Started
	Execute field trials	M20-M30	Not Started

Final Evaluation Scenarios

Table 11 describes the planned evaluation scenarios and relevant KPIs

Table 11: Evaluation scenarios and relevant KPIs for the MT1

Evaluation Scenario	Target	Relevant KPI
The evaluation scenario includes testing the various AI models on a labelled dataset to determine the detection rate and score achieved by the models.	Single person observing multiple cameras improve at least 10% the detection of anomalous events.	**MT-KPI1 increase the accuracy in detecting targeted events in crowds**
The evaluation scenario here involves observing the time taken to detect an anomaly through measuring the system's latency.	The aim is to detect anomalies 5 minutes from the start of the anomalous event.	**MT-KPI2 Detection time reduction.**

Modifications to the E/F/C infrastructure

The MT1 Infrastructure provided for the R1 integration consists of:

- three IP cameras are at Piazza Fiera (one of the main squares in Trento centre);

- three IP cameras are at Piazza Duomo (one of the main squares in Trento centre).

The upload function is a secure transmission by VPN access between MT and FBK in which raw video will be sent to the data lake in FBK.

FBK provides the Fog tier for the MT use cases. In order to comply with the constraints in the agreement that granted FBK access to the raw data of the MT's sensors and to satisfy the requirements of the MARVdash Kubernetes cluster, FBK deploys two workstations, both with GPU.

Datasets

For the development and evaluation of the MT1, a dataset was collected using the feeds from the set of selected cameras. The dataset was annotated, anonymised and made available to the partners:

- TrentoOutdoor – real recording (as defined in D2.1 and D8.2).

A dataset was collected at M11 and shared with the technical partner in order to improve the AI models. This process will be repeated until the end of the project.

At the end of the project, the dataset, or part of it, will be made publicly available to the research community.

The datasets have been collected at M11 and shared with the technical partners in order to improve the AI models. This process will be repeated until the end of the project.

At the end of the project, the two datasets, or part of them, will be made publicly available to the research community via MARVEL Data Corpus.

2.3.4 MT3: Monitoring of Parking Places

This use case concerns audio-visual monitoring of a parking place, including analysis of car trajectories, detection of cars out of the parking slots, car damages, car robberies, obstructions, etc. The target of this use case is the "Ex Zuffo" Parking Area which is one of the largest parking lots in Trento (around 1000 parking places). It is typically used by citizens who park their cars and then move around the city centre using public transportation, bike-sharing services or e-scooters.

The MARVEL framework will support the prevention of robberies or damages to parked cars through the audio-video analysis of the existing cameras and the microphones that are installed thanks to the MARVEL project. The system will analyse the audio-visual data to detect potential issues that may refer to the scenarios described above.

The aim of the use case is to detect anomalies, the timeline distribution of parking activity and anomalous behaviour, and the clustering of vehicles behaviour or events. When an anomaly is detected, an alert is sent to the Local Police headquarter, which should check the live feed from the camera where the event is occurring. In addition, the Local Police needs to check the feed of a few minutes before the anomaly happened, to accurately assess the cause.

The audio-visual analysis must be carried out in real-time and on recording data saved on the servers of the Local Police.

Execution time plan

The execution time plans up to the M18 and evaluation scenarios are reported in Table 12 for MT3. These were the plans laid out prior to the beginning of the integration.

Table 12: MT3 execution time plan

Phase	Activity	Period	Status
Baseline	Definition of the use cases	M6	Completed
	First data release • A dataset was collected at M11, anonymised and shared with the technical partners in order to train and improve the AI models	M8-M12	Completed
Innovation	HW procurement • Raspberry Pi devices purchased at M16	M8-M22	Ongoing
	HW computing devices installation • Raspberry Pi devices first stage of installation at M16	M15–M22	Ongoing
	HW sensing devices installation • Microphones combined with Raspberry Pi devices first stage of installation at M16	M15–M22	Ongoing
	First release of the MARVEL's infrastructure • Due to the GDPR constraints, MT decided at M16 that edge tier is no part of the Kubernetes cluster.	M18	Ongoing
	Minimum Viable Product	M12	Not Started
	MARVEL prototype 1st versions • Staged recording done at M13 • Starting from M15, annotated data has been provided to AI partners	M18	Ongoing
Experimentation	Internal Evaluation of the R1	M22	Not Started
	Revision of the data flow	M18-M24	Not Started
	Execute field trials	M20-M30	Not Started

Final Evaluation Scenarios

Table 13 describes the planned evaluation scenarios and relevant KPIs.

Table 13: Evaluation scenarios and relevant KPIs for the MT3

Evaluation Scenario	Target	Relevant KPI
The evaluation scenario involved the increased detection of targeted events, like car damages, checking the number of campers, and the average length of stay.	Single person observing multiple cameras improve at least 50% the detection of anomalous events.	**MT-KPI5 Increase the detection of targeted events**

The evaluation scenario here involves observing the reduction of detection time needed to identify the events mentioned above.	The aim is to detect anomalies 5 minutes from the start of the anomalous event.	**MT-KPI6 Detection time reduction.**

Modifications to the E/F/C infrastructure

The MT3 Infrastructure provided for the R1 integration consists of:

- two IP cameras are at Piazzale ex Zuffo (one of the largest parking places in Trento);
- two Microphones IFAG-MEMS at Piazzale ex Zuffo (one of the largest parking places in Trento);
- two Raspberry Pi 4B – for audio elaboration – at Piazzale ex Zuffo (one of the largest parking places in Trento).

The upload function is a secure transmission by VPN access between MT and FBK in which raw audio-video will be sent to the data lake in FBK.

FBK provides the Fog tier for the MT use cases. In order to comply with the constraints in the agreement that granted FBK access to the raw data of the MT's sensors and to satisfy the requirements of the MARVdash Kubernetes cluster, FBK deploys 2 workstations, both with GPU.

Datasets

For the development and evaluation of the MT3, a dataset was collected using the actual feeds from the set of selected cameras and microphones listed. The dataset was annotated, anonymised and made available to the partners:

- TrentoOutdoor – real recording (as defined in D2.1);
- TrentoOutdoor – staged recording (as defined in D2.1).

The datasets have been collected at M11 and shared with the technical partners in order to improve the AI models. This process will be repeated until the end of the project.

At the end of the project, the two datasets, or part of them, will be made publicly available to the research community via MARVEL Data Corpus.

2.3.5 UNS1: Drone Experiment

Monitoring and surveillance of large public events could be difficult due to the lack of infrastructure and sometimes unpredictable behaviour of the crowds. Fixed street cameras can provide frontal views of the crowd, but inner details could not be checked accurately. Furthermore, there are angles or even whole spaces that are not covered using cameras. The purpose of the Drone experiment use case is to evaluate the potential of drones in the monitoring of large public events. The utilisation of drones equipped with cameras and additional ground-based microphones and computational resources could help to check much faster whether some problematic behaviour has occurred among the crowds. The drone needs to fly over the main event points and recording from above could help to recognise if some anomalous and potentially dangerous event is happening. If the camera on the drone spots a problem, the drone can move closer to the identified location or inform the event organisers about the problem occurrence.

The focus of the use case is also on federated learning for crowd counting and onboard real-time processing. The idea of the pilot is to perform crowd classification and crowd counting.

Regarding classification, we will focus on the distinction between three classes of crowd behaviour: Neutral, Party, and Anomalous/dangerous behaviour. Later, classes can be further refined/multiplied as needed.

UNS members had several meetings with the organisers of the EXIT festival, which is one of the largest open-space music events in Europe, held each summer in Novi Sad. Their feedback was used as guidance for upgrading evaluation scenarios, as they are potential end-user. As a result, crowd counting was identified as the most desirable task to work on and their feedback was used while performing staged recordings.

Execution time plan

The execution time plan up to M18 is updated comparing to the plan from D1.2, by incorporating details related to hardware purchase and installation and availability of training data. Details are reported in Table 14.

Table 14: UNS1 execution time plan

Phase	Activity	Period	Status
Baseline	Definition of the use cases	M8	Finished
	HW testing	M7	Finished
Innovation	Recording app development	M10	Finished
	Databases recording • Staged recording performed in M16.	M16	Finished
	First data release • First experimental dataset release in M12; • Annotated dataset for VCC is a part of MARVEL Data Corpus from M18.	M18	Finished
Experimentation	Algorithm development for UNS Use Case X	M20	Initiated
	MARVEL prototype 1st version *Infrastructure enablers:* • Purchased Intel NUC to be secured for MARVEL purposes only; • Edge devices (RPi and Intel NUC) connected via a Wi-Fi access point to the Fog server; • Internal configuration of Edge devices and Fog server (VM); • Secured a separate internet connection to enable Kubernetes deployment. The E2F2C infrastructure, with a drone-mounted camera and a microphone, was fully configured and incorporated within the Kubernetes cluster and MARVEL platform.	M18	Finished
	Execute field trials Experimental tests carried out in M18.	M18-M20	Initiated
Consolidation	Public release of recorded databases	M22	Not started
	Fix issues	M22-M24	Not started

Final Evaluation Scenarios

Table 15 reports the evaluation scenarios of the UNS1 with the relevant KPIs.

Table 15: Evaluation scenario and relevant KPIs for the UNS1

Evaluation Scenario	Target	Relevant KPI
Detection of anomalous events and alerting event organisers about them,	The goal is to achieve 5% improvement comparing to the frontal camera views only or vision only.	**UNS-KPI1** **Increase the average accuracy for the drone-based audio-visual anomaly detection.**
System performance evaluation against, e.g., accuracy, latency, packet loss, for varying operating system conditions	The goal is to achieve at most 10% performance degradation comparing to the nominal system operation	**UNS-KPI2** **Robustness to different operating conditions (e.g., distance, fps rate, camera resolution, modality dropout)**
Monitoring (e.g., by security crew) of streaming data.	An improvement of 5% in detection time is expected in comparison to human-based monitoring.	**UNS-KPI3** **Decrease the time needed to identify an event using audio-visual monitoring comparing to the human (manual) detection.**
System performance evaluation with presence of both audio and visual modality and modality dropout.	The goal is to achieve 5% improvement comparing to the video or audio-based only system operation.	**UNS-KPI4 Multimodality: Different data modalities successfully accommodated in the platform (audio, video, GPS, etc.)**

Modifications to the E/F/C infrastructure

The infrastructure of the UNS1 use case covers all three layers: Edge, Fog, and Cloud. Within R1, recordings are performed using a drone-mounted camera and ground-based IFAG microphones. Intel NUC is used as a computing device at the Edge and it is mounted on the drone. Video anonymisation is performed at the Edge and after that data is streamed to the Fog layer using RTSP. The fog layer consists of the UNS server, which is used for running VCC in the inference pipeline and also for FedL-based VCC model training (prior to the inference). For the staged recording purposes, we have used lighter drone DJI P4 Multispectral Drone instead of DJI M600 Pro, due to lower noise level from propellers and less restrictive flying regulations. This way, the UNS team would like to acknowledge Prof. Boris Antić for providing us with the drone for staged recording.

Datasets

An audio-visual dataset was prepared containing audio-visual snippets that capture various sound or visual anomalies in crowds:

- sound anomalies: gunshot, broken glass causing people to disperse/run away.

The above-mentioned dataset features snippets capturing variations in several aspects: distances, camera angles and number of people in the scene. The data were acquired during UNS staged recordings in M16 at the Petrovaradin fortress, where UNS staff and students were gathered to simulate crowd movements and relevant crowd events. This dataset contributes to the *UNS drone dataset* described in D2.1 (Section 4.2.6) and also D1.2 (Section 4.6.1) and D8.2.

The main distinctions with respect to the experimental protocol dataset description (D1.2) are: 1) the dataset is fully annotated for visual crowd counting (VCC); 2) change of the recording location – from UNS Campus to Petrovaradin fortress; 3) only drone-camera views were used (i.e., no frontal camera views were acquired at this stage).

The change of location from the UNS campus to the Petrovaradin fortress was motivated for alignment with the EXIT festival, taking place at the same location, as the main tentative end-user for the experimental UNS1 use case. Due to the drone recording limitations (specifically, the light angle needs to satisfy certain requirements that prevent early recordings), the recordings had to occur in times of day where the likelihood of accidental observers increases. Hence, to ensure that no sensitive data were collected, no frontal cameras were used. We note that the height from which drone-camera recordings were collected was such that it was ensured that no sensitive information is present in the dataset (i.e., faces are not recognisable). We also note that all experimental participants have signed written consent to participate in the staged recordings.

Further details on this dataset and annotations can be found in Section 8.2.1.1 and Section 8.2.1.2 of the current document.

2.4 Use Cases for the 2nd release of the MARVEL Framework

Although the focus of T6.1 and of this section of D6.1 is on the deployment of the first prototype at M18, the remaining five use cases not involved in this intermediate deployment were addressed and revised, in particular in terms of the user journey and evaluation procedure.

2.4.1 GRN1: Safer Roads

This use case addresses the need to increase safety on urban roads for vulnerable road users, with the aim of encouraging the uptake of active travel modes in Malta. More specifically, this use case targets cycling. Malta has witnessed a significant effort, from both the authorities and the bicycle commuting lobby, in encouraging cycling and walking, mainly through infrastructural changes. The use case takes this effort further and aims at detecting cyclists, including e-bikes and possibly other motorised micro-mobility modes, exiting a junction and alert car and motorised-vehicle drivers of their presence via variable message signs in the hope that car drivers take greater care and concentrate more in such circumstances.

In addition, detecting cyclists is a particularly interesting task in low visibility conditions because it is both more dangerous for these entities and more challenging from a technology point of view.

From an AI task point of view, this use case requires detectors for traffic entities (cycles, pedestrians, cars, etc.) that are typically present at a junction. It is also necessary to resolve the exit carriageway taken by the vulnerable road users such that the respective message boards on that carriageway are triggered whilst avoiding false positives, the occurrence of which can reduce the system's impact in the long term. This can be achieved by sampling the entrance to the road. However, it will also increase the time taken for the system to respond.

In addition, it would be interesting to study whether driver behaviour improves when such a system is implemented. To do so, it is necessary to monitor a set of motorised-vehicle variables, such as speed along the stretch of road that the detected vulnerable road users use. Invariably, this necessitates the temporary installation of an additional data collection system that records variables such as speed and vehicle trajectory along the road. However, if this proves not to be possible, which is highly likely, interviews will be conducted instead.

The detection and classification of entities are typically implemented using computer vision techniques. Detecting the cyclist is a known hard problem, and at face value, the addition of the audio signal would not help. However, sound cues may potentially disambiguate a bicycle from a motorcycle or moped. In addition, audio-visual models may differentiate between bicycles and motorised bicycles, which is a desired function in use case IV.

2.4.2 GRN2: Road User Behaviour

This use case addresses the need to monitor the behaviour of road users at a junction. An application of this use case is in education campaigns targeting responsible driving, cycling, and other actions on roads. Malta has experienced fast changes in the transport landscape to which human response often lags behind technical progress. Educational campaigns are one way to close the gap and have been shown to be effective in the past. This use case involves the classification of actions into a spectrum of examples demonstrating good to bad behaviour. This use case will not be implementing the latter campaigns or policies; however, it could be tried in different places and its output could be observed. Surveys will be used to find how this tool will be able to help local authorities.

Examples of actions include the way pedestrians cross over the intended crossings, whether cyclists dismount at pedestrian crossings, and whether car drivers stop in the delineated zone at junctions. The system will be able to count the number of times bad behaviour is detected before and after the execution of education campaigns or policy changes.

At a high level, the system requires models that take as input multi-modal data (audio-visual streams) and output a summary of interesting AV segments. At a lower level, desired models need to compute approximate speed, detect anomalous sounds (e.g., vehicle horns, bicycle bells, excessive speed, etc.), work out the instantaneous location of an entity, and track the trajectory of the entities across the junction.

2.4.3 MT2: Detecting Criminal and Anti-Social Behaviours

The goal is to monitor some areas to detect criminal or anti-social behaviours. The system would trigger an alarm or a custom view in the control room.

MARVEL framework will be deployed to detect possible dangerous situations, during the night-time especially, such as gatherings, robberies, aggressions, and drug trafficking. The system has to analyse the visual and audio data streams of the cameras already installed in the selected location and send an alert to the local police operational centre in order to send the squad to the location. Moreover, the stream is saved into the local server of the local police for any further investigation.

The audio-visual analysis must be carried out in real-time and on recording data saved on the servers of the Local Police. The real-time analysis considers a daily time span approximately from 18:00 until 8:00 of the next day. If the infrastructure allows it, 24/7. In the second hypothesis, the data, in accordance with the provisions of the GDPR 2016/679 on privacy, will be saved for seven days and then deleted if no requests are received for timely investigations by the police.

As previously described, the actions to be monitored will be the presence of bothersome gangs (to detect groups, noises, actions), aggressions or robberies, gang fights, and drug dealing.

The evaluation parameters for the use case will be:

- the time of reaction in case of issues, in terms of intervention by Local Police and other authorities (First aid, Carabinieri, etc.).

- the reduction of citizens reporting, in terms of the effectiveness of the solution for improving the perceived safety of the citizens.
- the increased detection of targeted events, like the presence of bothersome gangs (to detect group, noises, actions), aggressions or robberies, and gang fights. It is expected that 50% of dangerous situations will be correctly noted.

The Local Police will use the MARVEL framework and the police officers will analyse the results obtained by the system in collaboration with MT managers.

Non-functional evaluation variables like scalability, end-user experience, data protection, and preservation of privacy will be validated by MT managers, Local Police, DPO, and IT managers/infrastructure managers.

2.4.4 MT4: Analysis of a Specific Area

The Municipality of Trento wants to monitor the city's main places to support the Administration's decision-making. In order to do this, the MARVEL framework can help with the counting of persons, cars, buses, taxis and bikes, calculate their trajectories and calculate any notable event during a specific timeframe (for example, from 7.30 h to 8.30 h for school or 17.00 h to 19.00 h for the train station) or the entire day. From a technical point of view, the goal of this use case is the collection and the following analysis of data (number of persons, cars, trajectories, events).

We have identified the area in front of the train station, comprising the road and part of Piazza Dante (from Via Dogana traffic lights to Via Pozzo traffic lights). Most probably, this scenario will be integrated into the project for the creation of the "Smart City Control Room" being launched in the municipality of Trento, an initiative designed to prepare and collect the data necessary for the creation and monitoring of the urban plan for sustainable mobility, and sustainable energy action plan for the energy transition.

The audio-video analysis must be carried out in real-time and on recording data saved on the servers of the Local Police.

The evaluation parameters for the use case will be:

- the creation of a queryable database that can be used to look up historical data on the turning ratios and other statistics such as the number of persons, cars, trajectories, events, that we can summarise as “detection of habits” in a specific area of the town to support long-term decision-making by public authorities. The MT managers will use the MARVEL framework and the officers will analyse the results obtained by the system in collaboration with policy-makers;
- the increase of efficiency in urban planning in terms of traffic management and traffic city planning. The MT managers will use the MARVEL framework and the officers will analyse the results obtained by the system in collaboration with policy-makers.

Non-functional evaluation variables like efficacy, scalability, end-user experience, data protection, privacy preservation and citizens’ satisfaction will be validated by MT managers, Local Police, DPO, and IT managers/infrastructure managers and policy-makers.

2.4.5 UNS2: Audio-Visual Emotion Recognition

With this use case, the goal will be to develop methods for distinguishing between several basic emotional states, including neutral, happy, angry, sad, and scared, based on both audio and visual modalities. These emotional states are recognised in the psychological sciences as universally experienced in all human cultures. Within this use case, we will perform audio-visual emotion detection from a close-up camera and a microphone. These two modalities are complementary to each other. Although facial expressions clearly express some emotions – like smiling for happiness and frowning for anger, recordings from different angles and low level of expressivity might lead to emotion detection mismatch as there are similarities among facial expressions of different emotions. Audio signal can help differentiate emotions that appear similar in the video or images since it is proved that voice characteristics greatly depend on the underlying emotional state. For example, happiness will be expressed by higher a pitch and usually faster speech, while sadness is typically expressed by slower speech. Some accompanying expressions like laughter, crying, inhales or similar, can also help in distinguishing emotions.

The development of such multimodal algorithms should lead to the increase of classification accuracy, providing algorithms that could be applied in smart cities, for various applications. Detecting fear on someone's face in the crowd (in subway, market or shopping mall) can indicate that something is wrong – the person can be lost or even kidnapped. Anger on someone's face can indicate that the person can be the cause of some fight or similar action. Also, such scenario can for example prevent a bank to let the burglar in.

2.5 Development of MARVEL framework for selected use cases for M18

2.5.1 Revision of MARVEL conceptual architecture

This section provides a detailed report on revisions of the MARVEL conceptual architecture, as initially devised for D1.3[2]. First, we provide a general overview and list revisions across MARVEL subsystems, while the details on the revisions of each component (where applicable) can be found in Section 2.5.2. Figure 3 presents the revised MARVEL conceptual architecture.

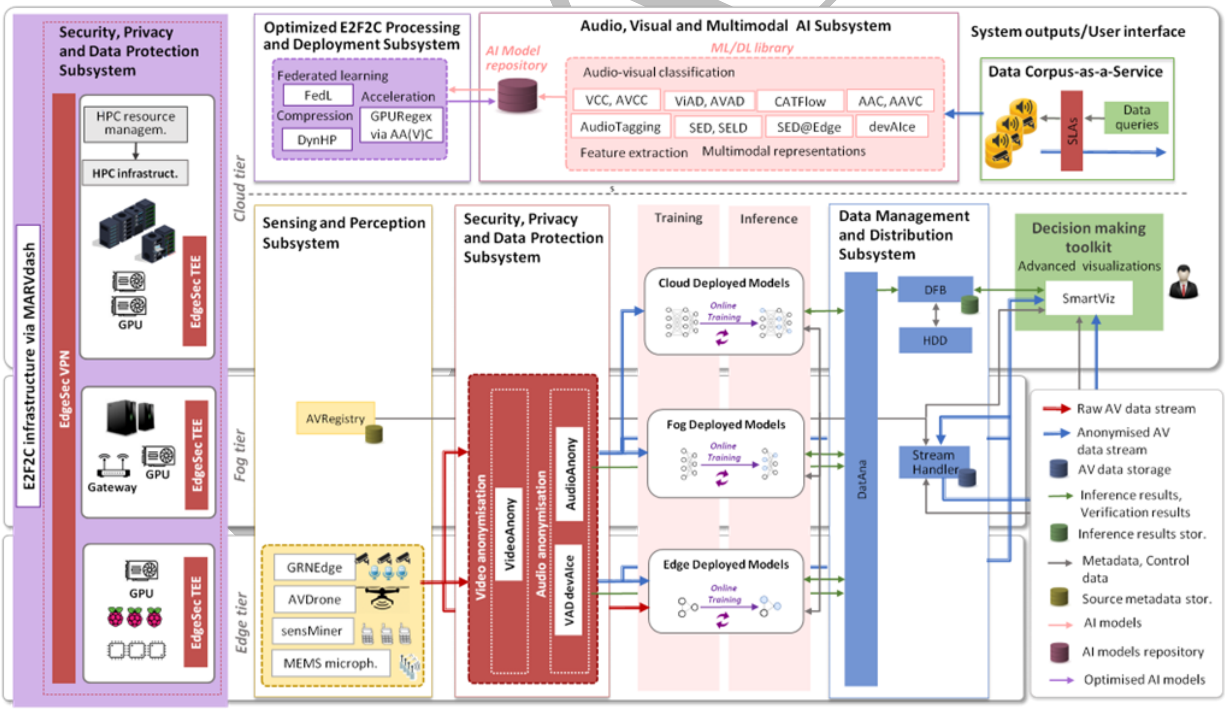


Figure 2: MARVEL conceptual architecture

1. **Sensing and perception subsystem** – As before, this subsystem includes all AV and possibly other data sources producing raw audio, visual and audio-visual streams, possibly

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

with the presence of other modalities, and, more generally, all hardware and software components that directly contribute to data collection and acquisition. The distinction with respect to the architecture version in D1.3 is the inclusion of a novel component – AV Registry, serving as a repository of metadata of all AV sources in the system. Further, components – SED@Edge and CATFlow have been shifted to the Multimodal AI subsystem, due to the changed nature and role of these two components following their developments and status since M08 (more details on this change are provided below).

2. **Security, privacy and data protection subsystem** – The role and functionalities of this subsystem have not changed: to enable secure communication and code execution between and within all elements of the platform and ensure data protection by proper anonymisation. With regards to the changes, the former EdgeSec component has been split into two independent components, namely EdgeSec VPN and EdgeSec TEE, addressing, respectively, secure end-to-end communication and secure code execution, to facilitate components' individual developments. With regards to anonymisation components – VideoAnony, VAD, and AudioAnony, the allowed deployment layers – edge and fog, compliant with the data protection requirements, are now explicitly presented in the MARVEL conceptual architecture figure.

3. **Data management and distribution subsystem** – This subsystem collects all MARVEL components that manage data, including both binary and textual data. With respect to D1.3, several components have been extended, fine-tuned, or fully revised in terms of functionality and role within the platform: DFB has sharpened its roles to enable Kafka topics for AI components and persistent storage of inference results; having the capability of running at the edge, DatAna is acting as the main E2F2C information "bus" of the platform, through which all textual (non-binary) data are managed (most notably, inference results) and transformed for compliance with relevant data model standards; StreamHandler has extended its role by incorporating a novel functionality, namely, persistent storage of AV data; and HDD has been adapted by liaising with DFB to address optimal Kafka topics partition.

4. **Audio, visual and multimodal AI subsystem** – As indicated above, following the more flexible status of CATFlow within the platform, operating now as a deployable AI component (as opposed to an AI-based system in M08), this component has been reclassified in the Audio, visual and multimodal AI subsystem. Similar reclassification occurred for SED@Edge as well. Further, as an important addition, the subsystem includes now an AI model repository, with persistent storage of trained AI models for AI tasks relevant to MARVEL, including different subtypes, compression levels, and other model variations. Finally, the Acoustic Scene Classification (ASC) component has been replaced by Audio Tagging (AT) as a more suited functionality for acoustic analysis within MARVEL pilots.

5. **Optimised E2F2C processing and deployment** – This subsystem consists of two types of components: 1) components that aim at optimising the platform's operation, in terms of model accuracy – FedL, model size – DynHP, and GPU-based pattern matching acceleration – GPURegex, and 2) Kubernetes-based deployment through MARVdash platform (previously Karvdash). Besides internal components' developments, most notable change occurred with GPURegex, the application of which in MARVEL has been enabled through coupling with Automated Audio (Visual) Captioning.

6. **E2F2C Infrastructure** – The developments in this subsystem consist of concrete instantiations of E2F2C infrastructure with each of the MARVEL pilots and R1 use cases,

as described in Sections 4-8.1.2, and PSNC HPC-based cloud delivery. The subsystem did not exhibit significant revisions in terms of roles and functionalities.

7. **System outputs** – As in the D1.3 version of the MARVEL conceptual architecture, this subsystem consists of two main elements of MARVEL's UI: SmartViz, for results visualisations, and MARVEL Data Corpus, for access to labelled, anonymised training data from MARVEL pilots. Both components have undergone significant enrichment of functionalities since M08.

2.5.2 Components

In this section, all technology providers that offer components for the M18 use cases selected report modifications in their components, evaluation protocol or other relevant revisions.

2.5.2.1. Audio tagging (AT)

Audio tagging (AT) component was introduced to complete the set of audio analysis components available in the project. This component is related to the acoustic scene classification (ASC) component and sound event detection (SED) component. In the ASC component, segments of an audio signal are classified into one of the predefined sound classes, and in the SED component, the start and end time stamps of each active sound event within the audio signal are outputted. In the case of AT component, a segment of audio can be assigned with multiple tag labels at the same time. Usually, sound classes used in ASC are defined such that they characterise the acoustic scene at high-level (e.g., "busy street"), and tag labels used in AT are related to sound sources or sound events active at the acoustic scene (e.g., "people talking", "car alarm"). Audio tagging is used instead of SED in use cases that require recognition of multiple sound classes at time, but the exact start or end timestamps of these sounds are not important. AT component will be applied in use cases (GRN, MT, UNS) instead of ASC or SED components when these requirements are met. In use case GRN3, the AT component is taking as input fixed-length audio segments (five seconds) captured using traffic monitoring cameras, and the component outputs per segment labels describing level of traffic and level of speed of the traffic at the location. Audio tagging is evaluated using mean average precision (mAP).

As a baseline with respect to which AT component will be compared and evaluated, the DCASE Challenge audio tagging task baseline will be used. The improvement over the indicated baseline can be measured using mAP metric. If no data from MARVEL can be provided, then the improvement over the baseline can be measured using the standard freely available AT data used at the AT task of DCASE. Additionally, there are other publicly available datasets that can be used for the tagging task, such as AudioSet. We expect to improve the baseline by a relative 10% over the DCASE baseline system in the DCASE Challenge development setup (using the same metric, dataset and cross-validation setup). In terms of KPIs, the AT component can target the addressing of KPI-O2-E2-1, KPI-O2-E3-1, and iKPI-3.2.

2.5.2.2. Text Anomaly Detection (TAD)

Text Anomaly Detection (TAD) was introduced as part of GRNs contribution to Task 4.4 as a means of post-processing the output from the CATFlow algorithm. TAD is a component that automatically detects anomalous events in data, for example anomalous vehicle velocities and trajectories. TAD takes as input the JSON messages outputted from CATFlow, and after processing, flags any anomalous behaviour. The TAD component requires access to the dataset which CATFlow outputs such that a model of the normal behaviour on the scene being observed is developed and updated. The current TAD version considers the speed of vehicles and flags anomalous low or high values.

The current implementation of the TAD component makes use of the CATFlow output, specifically the vehicle speed calculation. In addition, TAD accesses the GRN CATFlow database to model the vehicle speed normally observed on the road segment being monitored. The TAD can also use a preloaded model instead of accessing the database, such that the anomaly detection tool can still be used in the event of the database being inaccessible. During the last step, TAD performs a z-score test for any new vehicle speed value extracted from the CATFlow output and if the new data point is not within the range of the Z-score test, the TAD flags or raises an alarm to indicate the occurrence of the anomalous event.

In the current implementation, two types of anomalies can be detected:

- anomalously low speeds usually indicate that either a vehicle has stopped moving, possibly creating an obstruction or unusual traffic jams;
- anomalously high speeds usually indicate vehicles that are over speeding (velocity beyond sign posted limit) and are useful in estimating the safety of the road segment under observation.

Table 16: Asset specific KPIs for the "Audio visual and multimodal AI" subsystem

Asset	KPI	Metric	Baseline SOTA	Datasets / Benchmarks	Expected result	Relevant project KPIs
AT	Accuracy	Mean average precision	Baseline of the AT task at DCASE Challenge	Publicly available datasets, e.g., data used at the AT task of DCASE, and dataset available in MARVEL	10% relative improvement on metrics in DCASE setup (dataset, cross-validation setup)	KPI-O2-E2-1 KPI-O2-E3-1 iKPI-3-2
TAD	Accuracy	ROC Curve, Area Under ROC Curve	Not Available	Possible Publicly Available Datasets and MARVEL Datasets	>53% accuracy in detecting relevant anomalies	KPI-O2-E2-1 KPI-O2-E3-1 KPI-O2-E3-3

2.5.3 MARVEL framework

During the period M8-M18 no major scientific, industrial or technological shifts have been observed, therefore no revision of the MARVEL framework is necessary. The most noticeable modification is the introduction of the two new AI components, which, however, do not affect the requirements of the MARVEL frameworks. The KPI revision reported in the next section is mostly concerned with minor changes in the metrics and in the evaluation subjects.

2.6 KPI revision

This section provides a refinement of some of the KPIs defined in D1.2. Most of the revisions are minor and mostly related to the measurability of the KPIs. The tables reported here updates part of the tables in D1.2.

2.6.1 Sensing and Perception

Table 17 presents the specific KPIs relating to the asset selected for implementing use cases developed for R1. Below, these assets are described referring to the KPIs identified.

Table 17: Asset specific KPIs for the "Sensing and perception" subsystem

Asset	KPI	Metric	Baseline SOTA	Dataset/ Benchmarks	Expected result	Relevant Project KPIs
GRN Edge	Robustness	Sustains performance in various weather conditions, (Power source, sensor, and CPU board operation)	Experiments during MARVEL project	GRN AV data from field trials	Performance sustained in most weather conditions	KPI-O1-E1-1 KPI-O1-E1-2 KPI-O5-E1-1
	Reliability	Percentage of transmitted packets lost	Experiments during MARVEL project	GRN AV data from field trials	Downtime is minimised to 10% or less	

2.6.1.1. GRNEdge

With reference to the GRNEdge components, the metric for evaluating the reliability of the components was changed from the percentage of transmitted packets lost to downtime as a percentage of total time. The motivation for this change was that throughout the first half of the project, it was noted that the lost packets are not critical to the performance of the AI models, whilst downtime is. The expected result was also altered to reflect this change in the metric. The expected result is to minimise the downtime to 10% or less.

2.6.2 Multimodal AI

Table 18 present the specific KPIs relating to the asset selected for the implementation of use cases developed for R1. These assets are described and refer to the KPIs identified.

Table 18: Asset specific KPIs for the "Multimodal AI" subsystem

Asset	KPI	Metric	Baseline SOTA	Dataset/ Benchmarks	Expected result	Relevant Project KPIs
CATFLOW	Performance	Classification Accuracy Latency	Experiments on MARVEL data	GRN Dataset	GRN project managers and developers	KPI-O2-E3-1
	Scalability	Computational resources needed	Use a set of benchmarks (designed within project MARVEL)	GRN Dataset	GRN project managers and developers	

2.6.2.1. CATFlow

With reference to the CATFlow component, the expected result was changed for both mentioned KPIs which are Performance and Scalability. For the performance KPI, the expected result was changed to: Classification Accuracy increased by 5% and adequate Latency that satisfies real-time use cases. For the Scalability KPI, the expected result is that more than one instance of components is running on a fog device. The reason for both changes was to better reflect and describe the goals of the MARVEL project.

2.6.3 Data Management and Distribution

Table 19 present the specific KPIs relating to the asset selected for the implementation of use cases developed for R1. These assets are described and refer to the KPIs identified.

Table 19: Asset specific KPIs for the “Data Management and Distribution” subsystem

Asset	KPI	Metric	Baseline SOTA	Dataset/ Benchmarks	Expected result	Relevant Project KPIs
DFB	Data Integrity	Data loss rate	Isolated execution of DFB	Synthetic data streams	Confirm that advanced encryption mechanisms over end-to-end data transfer will guarantee data integrity	KPI-O1-E2-1 KPI-O1-E2-2 KPI-O1-E2-3 KPI-O1-E2-4
	Scalability	Hardware resources utilisation, speedup			Increase the number of modality data streams and verify that performance metrics improve or at least stay the same	
	Availability	Service availability-failed request, data access restriction			Verify that DFB resources are available and discoverable	
	Performance for high volume, heterogeneous data streams	Data transfer latency, data throughput, response time, number of cluster nodes			Thoroughly measure different performance metrics under different execution conditions	
StreamHandler	Performance for high volume, heterogeneous data stream	Data transfer latency, data throughput, response time	Isolated execution of StreamHandler	Synthetic audio-visual streams of different formats and quality	Ensure small latency in - storage of segments, - response time to API requests (including compilation of edited files). Validate with streams of different formats and degrees of quality	KPI-O1-E2-3

2.6.3.1. DFB

DFB stores the inference results it receives, and it makes the data available in real-time. Additionally, the DFB persists all inference results in an Elasticsearch database and exposes a REST API. The KPIs for the DFB defined in D1.2 (KPI-O1-E1-1, KPI-O1-E1-2, KPI-O1-E2-2, KPI-O1-E3-2) are replaced by KPI-O1-E2-1, KPI-O1-E2-2, KPI-O1-E2-3, KPI-O1-E2-4.

2.6.3.2. StreamHandler [INTRA]

StreamHandler is responsible for the satisfaction of KPI-O1-E2-2, which is associated with increasing the number of different modality data streams. By providing the ability to handle audio-visual data, modalities are clearly increased serving both the project and the component itself that did not have this capacity before. StreamHandler also contributes to KPI-O1-E1-1, KPI-O1-E2-1, KPI-O1-E2-3 and KPI-O1-E2-4.

2.6.4 Audio Visual and Multimodal AI

Table 20 present the specific KPIs relating to the asset selected for the implementation of use cases developed for R1. These assets are described and refer to the KPIs identified.

Table 20: Asset specific KPIs for the "Audio visual and multimodal AI" subsystem – addendum with respect to D1.2

Asset	KPI	Metric	Baseline SOTA	Datasets / Benchmarks	Expected result	Relevant project KPIs
AT	Accuracy	Mean average precision	Baseline of the AT task at DCASE Challenge	Publicly available datasets, e.g., data used at the AT task of DCASE, and dataset available in MARVEL	10% relative improvement on metrics in DCASE setup (dataset, cross-validation setup)	KPI-O2-E2-1 KPI-O2-E3-1 iKPI-3-2
VCC	Accuracy	MAE MSE	M-SFANet Transformer-based crowd counting	Shanghai Tech Parts A and B, World Expo 10 DISCO	11.00 MAE on average for sub-regions of the image	KPI-O2-E3-1 KPI-O2-E3-2 KPI-O2-E3-3
	Speed	FLOPS			30% speedup while retaining 90% accuracy	

2.6.4.1. *VCC (Video Crowd Counting)*

The KPIs for VCC are updated as follows: "11.00 MAE on average for sub-regions of the image on DISCO dataset."

Reason for update: previous set KPIs were "62% frame AUC on Street Scene, 97.5% frame AUC on UCSD" referring to a Visual Anomaly Detection functionality. The updated values refer to the actual VCC functionality measured on a standard public benchmark.

2.6.4.2. *SED (Sound Event Detection)*

The KPI metrics definitions for SED are updated as follows: "The macro-averaged (averaged over sound event classes) segment-based F1-score metric will be used in addition to default micro-averaged F1-score metric to take better account of the unbalanced nature of data".

2.6.4.3. *SELD (Sound Event Localisation and Detection)*

The KPI metrics definitions for SELD are updated as follows: "The location-dependent F1-score metric will be macro-averaged (averaged over sound event classes) instead of micro-averaged to take better account of the unbalanced nature of data."

2.6.5 Use case KPIs

Table 21 reports the revised use case KPIs with respect to D1.2. The unmodified KPIs are not reported here.

Table 21: Use case KPIs

Use case	KPI	Metric	Baseline	Expected result/ Improvement	Evaluators
GRN3	GNR-KPI4: Correctly detecting various anomalies on the road	Detection rate/F1	Labelled Dataset	70% detection of anomalous events	GRN Developers External traffic experts

GRN4	GRN-KPI6 Availability of historical video samples of pedestrian and vehicle trajectories at two junctions.	Automatic detection of patterns	No baseline	50% of the trajectories detected	GRN managers External traffic experts
	GRN-KPI7 Increased efficiency in the planning of roads	Surveys with road planners	No baseline	potential decrease in time for planning through the availability of data	External traffic experts
UNS1	UNS-KPI1 Increase the average accuracy for the drone-based audio-visual anomaly detection	Classification accuracy; Multimodality	Vision only case	5% improvement	UNS staff
	UNS-KPI2 Robustness to different operating conditions (e.g., distance, fps rate, camera resolution, modality dropout)	Accuracy (Light intensity) Latency (distance) Packet loss (distance)	Current System operation	At most 10% performance degradation compared to the nominal system operation	UNS staff
	UNS-KPI3 Decrease the time needed to identify an event using audio-visual monitoring comparing to the human (manual) detection	Processing time	Human visual detection (e.g., by security crew)	Improvement of 5% in detection time is expected in comparison to human-based monitoring.	UNS staff
	UNS-KPI4 Multimodality: Different data modalities successfully accommodated in the platform (audio, video, GPS, etc.)	Classification accuracy	Video-only or audio-only system operation	The goal is to achieve 5% improvement compared to the video or audio-based only system operation	UNS staff

2.6.5.1. GRN3: Traffic Conditions and Anomalous Events

For this use case, the GNR-KPI4 was altered from *Correctly detecting the cause of obstructions on the road* to *Correctly detecting various anomalies on the road*. The reason behind this change was to expand the possibilities of anomalies detected on the road. Obstructions are a type of anomaly and therefore will still be included amongst the anomalies. The baseline for this KPI was also changed from *Possible manual detection in the control room* to *Labelled Dataset*. This change makes the evaluation of the use case easier and just as effective.

2.6.5.2. GRN4: Junction Traffic Trajectory Collection

For the GRN-KPI6, the expected result was altered from *50% of the patterns detected* to *50% of trajectories detected*. This change better reflects the expected outcome for detecting the vehicle trajectories and collecting historical data.

For the GRN-KPI7 the expected result was clarified from *decreased time for planning* to *potential decrease in time for planning through the availability of data*. This clarification was required since it is not possible to accurately measure the decrease in planning time, however, it would be possible to obtain feedback from the survey with road planners on if and how this data could help in decision-making during planning and maintenance of road infrastructure.

2.6.5.3. UNS1: Drone Experiment

The UNS-KPI2 has been modified from "*Robustness to different operating conditions (e.g., distance, light intensity)*" to "*Robustness to different operating conditions (e.g., distance, fps rate, camera resolution, modality dropout)*" in order to account to new possible operation challenges identified during the hardware deployment and the finalisation of the evaluation scenarios.

The target has been revised from "*5% improvement in detection time with respect to the current state*" to "*10% degradation with respect to nominal operating conditions*" to better align with the KPI nature, targeting system robustness.

The UNS-KPI3 "*Decrease the time needed to identify an event using audio-visual monitoring*" has been modified as "*Decrease the time needed to identify an event using audio-visual monitoring comparing to the human (manual) detection*", to better quantify the comparison baseline. The target has been revised as: "*Improvement of 5% in detection time is expected in comparison to human-based monitoring*" to provide a better-defined metric.

2.6.6 Use case non-functional KPIs

Table 22 reports the revised use case KPIs with respect to D1.2. The unmodified KPIs are not reported here.

Table 22: Use case specific non-functional KPIs

Use case	Evaluation variable	How to measure	Internal evaluators	External Evaluators
GRN1	End-user Experience Safer cycling System welcomed by cyclists	Survey	GRN managers	Cyclists
	Efficacy	Survey	GRN managers	
	Scalability	Cost to add new devices/junctions	GRN managers	
GRN2	Potential end-user	Survey	GRN Managers	Transport authorities Road users
	Efficacy	Survey	No baseline	Road and Transport Experts

2.6.6.1. GRN1: Safer Roads

For the evaluation variable *Efficacy*, the measurement method was altered from *Car drivers demonstrate increased awareness* to *Survey*. This necessitated the addition of *car drivers* as an external evaluator. The plan is still to note the driver's awareness, however, the evaluation method is a survey. Car drivers were chosen as the external evaluators as driving with a personal vehicle is the most common method of transport in Malta.

2.6.6.2. GRN2: Road User Behaviour

For this use case, the measurement methods for the *Efficacy* variable were updated. The measurement method of *Integrating the system in at least one safety campaign* was removed as it is unlikely GRN to have the opportunity to participate in such a campaign. The other measurement method *Noting an increase of 50% in awareness of bad behaviour* was changed to *Survey*. The external evaluators were also altered to *Road and Transport Experts*. In summary, the surveys collected from the experts will allow the effectiveness of this system to be evaluated.

3 Cross-cutting aspects for R1 use case realisations

This section shows the aspects that characterise all the use cases selected for R1, specifically: the Edge-to-Fog-to-Cloud infrastructure developed, the inference pipeline conceived, the batch processing pipelines for system optimisation created and the process implemented in order to guarantee ethics and privacy assurance and to anonymise the data collected, the realisation of the Data Corpus and the integration process overview.

3.1 E2F2C infrastructure via MARVdash

MARVdash provides a backbone, exploiting Kubernetes, for the deployment of all components within the MARVEL platform. MARVdash provides a dashboard for requesting resources and specifying other parameters related to service execution. The main goal is to make it straightforward for domain experts to interact with resources in the MARVEL platform without having to understand lower-level tools and interfaces. The relationship between MARVdash and Kubernetes can be seen in Figure 3. Deployed applications in the Kubernetes cluster are represented as Pods. A pod is the smallest deployable object of the Kubernetes ecosystem. It specifically represents a group of one or more applications running together on a cluster.

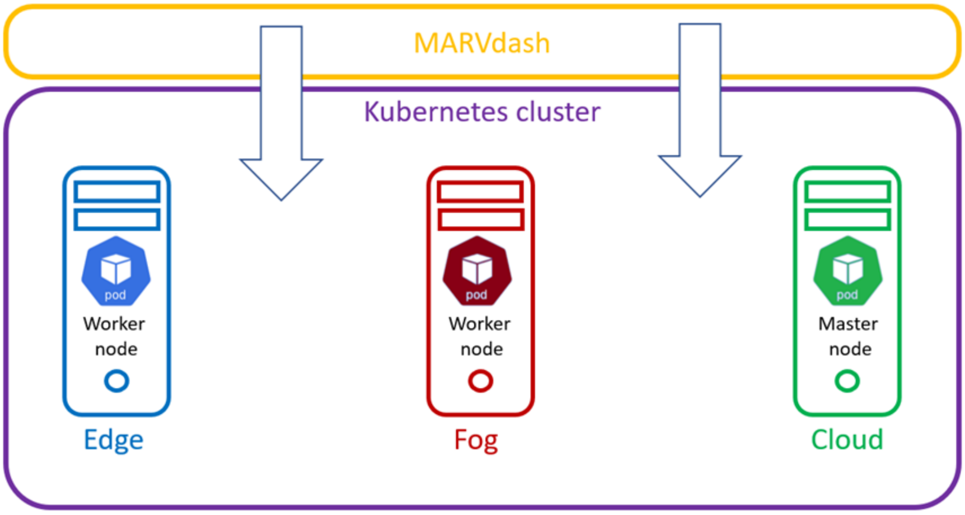


Figure 3: High level view of the E2F2C MARVEL infrastructure

Kubernetes is an open-source container management tool. Its functionalities include container deployment, scaling and descaling of containers, and container load balancing on computing nodes. The whole set of nodes constitutes the Kubernetes cluster.

Mentioning the advantages of containers is the first step toward convincing stakeholders about the value of Kubernetes. Containers are an easy way to package and deliver code, they are a good way to run applications, and it seems that there is no other technology that could be considered a worthy competitor. The container technology is here to stay, but in a production environment, containers must be managed and run with no downtime. This is exactly what is offered by Kubernetes, an interface for running a distributed system as smoothly as possible. Kubernetes will bring to the table quick scaling, resilience of the workflows, and consistent deployment management. A more comprehensive list of Kubernetes offerings[3] is presented below:

[3] https://kubernetes.io/

- **automated rollouts and rollbacks** – In case you need to apply any changes to your deployed application, Kubernetes progressively rolls out changes, while monitoring your application;
- **service discovery and load balancing** – Kubernetes assigns to running pods on their own IP addresses and a single DNS name for a set of Pods, and can load-balance across them;
- **storage orchestration** – Kubernetes allows you to mount the storage system of your choice;
- **secret and configuration management** – Kubernetes gives you the ability to handle sensitive data such as a password, a token, or a key, without including confidential data in your application code;
- **IPv4/IPv6 dual-stack** – Allocation of IPv4 and IPv6 addresses to Pods and Services;
- **horizontal scaling** – Kubernetes gives you the option to scale an application up and down with a simple command, with a UI, or automatically based on CPU usage;
- **self-healing** – One of the major advantages of Kubernetes is that it restarts containers that fail, replaces and reschedules containers when nodes die, kills containers that don't respond to your user-defined health check, and doesn't advertise them to clients until they are ready to serve;
- **designed for extensibility** – Add features to your Kubernetes cluster without changing upstream source code.

Kubernetes by design requires that all pods can communicate with other pods on any node, even if some of the nodes are part of local private sub-networks. This fact comes to direct contradiction with the actual setup of having remote nodes in different layers, as the MARVEL platform does.

The MARVEL Framework requires nodes from all different layers (edge, fog, and cloud) to become part of the Kubernetes Cluster. The aforementioned requirement can be fulfilled with the use of EdgeSec VPN, which brings together all the participating nodes as if they were under the same local network. What EdgeSec VPN essentially does is to create a new virtual network, where every node becomes its part, by assigning to it a VPN IP. In that way, two distant machines join the same network with MARVdash and become part of the existing Kubernetes cluster. Nodes are able to directly announce themselves and discover other nodes on the virtual network.

The communication between the participating nodes is limited to the traffic that matches the network subnet defined by the EdgeSec VPN. This means that only the traffic that is relevant to the communication of the MARVEL services, placed in the individual nodes, is routed through the VPN channel. All unrelated traffic such as browsing the internet or downloading updates used other mediums, limits the overhead of the VPN channel.

To ensure that pods are placed on the appropriate nodes, we use mechanisms offered by Kubernetes such as taints and tolerations. Taints are used on the nodes and tolerations are used on the pods. When we add a taint to a node, we practically exclude all the pods that do not have a matching toleration. When we add a toleration to a pod, we allow (but do not require) the pod to be deployed on nodes with matching taints. Another mechanism, Node affinity, is a set of rules used to determine where a pod can be placed. Node affinity allows a pod to specify an affinity (or anti-affinity) towards a group of nodes it can be placed on. The above mechanisms allow nodes to accept only pods that must be instantiated in that layer.

Furthermore, several of the pilot's devices and machines have an NVIDIA GPU that the Kubernetes cluster needs to access. The official NVIDIA GPU device plugin is installed for that reason. This plugin has the following requirements for the GPU Kubernetes nodes: i) nodes to be pre-installed with NVIDIA drivers and nvidia-docker 2.0; ii) kubelet must use Docker as its container runtime; iii) nvidia-container-runtime must be configured as the default runtime for Docker, instead of runc.

For certain use cases, such as GRN3, there was also the need to expose services outside Kubernetes. One of these services was an RTSP Server, which enables clients to publish and consume streams. More specifically, there is a need for components existing outside as well as inside the Kubernetes cluster to be able to interact with the RTSP Server. To achieve this, we used the Kubernetes NodePort type of service. A NodePort is a way to enable Kubernetes services to receive traffic from outside the cluster.

Similarly, there was also the need to expose an MQTT Broker outside Kubernetes. Components, acting as publishers/subscribers, existing outside as well as inside the Kubernetes cluster, need to be able to interact with the MQTT Broker. Similarly, as with the RTSP service, we used here as well the Kubernetes NodePort type of service.

Furthermore, there was also a need for the usage of confidential data in cases such as VideoAnony and CATFlow. Specifically, in the case of the VideoAnony the URLs of the streams are confidential and for this reason, they are exposed to the Pods via secrets. A Secret is an object that contains a small amount of sensitive data, such as a password, a token, or a key. The usage of Secret allows you not to include confidential data in your application code. Likewise, to safeguard GRN's IP on the CATFlow software asset, the CATFlow configurator was uploaded as part of the MARVEL registry. This configurator is then responsible for pulling the CATFlow image onto the device from GRN's Azure registry. This configurator also uses secrets.

MARVdash provides a backbone, exploiting Kubernetes, for the deployment of all components within the MARVEL platform. Kubernetes is an open-source container management tool. Its functionalities include container deployment, scaling and descaling of containers, and container load balancing on computing nodes. The whole set of nodes constitutes the Kubernetes cluster.

Kubernetes by design requires that all pods can communicate with other pods on any node without NAT which comes in direct contradiction with the actual setup of having remote nodes. The aforementioned requirement can be fulfilled with the use of EdgeSec VPN, which brings together all the participating nodes as if they were under the same local network. A Super Node that assigns a VPN IP for both the server and the workstation is used at the cloud in the PSNC's infrastructure. In that way, two distant machines join the same network with MARVdash and become part of the existing Kubernetes cluster. Nodes are able to directly announce themselves and discover other nodes via the Super Node.

The communication between the participating nodes is limited to the traffic that matches the network subnet defined by the EdgeSec VPN. This means that all unrelated traffic such as browsing the internet or downloading updates is not routing through the VPN thus limiting the overhead of the VPN channel.

After becoming parts of the Kubernetes cluster these nodes are tainted and labelled. This happens in order the nodes to accept only pods that must be instantiated in that layer.

Furthermore, several of the pilot's devices and machines have an NVIDIA GPU that the Kubernetes cluster needs to access. The official NVIDIA GPU device plugin is installed for that reason. This plugin has the following requirements for the GPU Kubernetes nodes: i) nodes to be pre-installed with NVIDIA drivers and nvidia-docker 2.0; ii) kubelet must use Docker as its container runtime; iii) nvidia-container-runtime must be configured as the default runtime for Docker, instead of runc.

For certain use cases, such as GRN3, there was also the need to expose services outside Kubernetes. One of these services was an RTSP Server, which enables clients to publish and consume streams. More specifically, there is a need for components existing outside as well as inside the Kubernetes cluster to be able to interact with the RTSP Server. To achieve this, we used the Kubernetes NodePort type of service. A NodePort is a way to enable Kubernetes services to receive traffic from outside the cluster.

Similarly, there was also the need to expose an MQTT Broker outside Kubernetes. Components, acting as publishers/subscribers, existing outside as well as inside the Kubernetes cluster need to be able to interact with the MQTT Broker. Similarly, as with the RTSP service, we used here as well the Kubernetes NodePort type of service.

Furthermore, there was also a need for the usage of confidential data, specifically for the CATFlow component (within use cases GRN3, GRN4 and MT1) the handling of which was limited due to the associated IP rights. To overcome these limitations, Kubernetes secrets were used for the pods of CATFlow. A Secret is an object that contains a small amount of sensitive data such as a password, a token, or a key. Using a Secret allows you not to include confidential data in your application code. For the case of CATFlow, the Secret was used specifically for the CATFlow model.

3.2 The inference pipeline

In the context of the execution of the use cases using AV data from cameras and microphones, a set of real-time streaming data pipelines has been set up. It involves many of the components and elements of MARVEL, which can be divided into different categories.

Deployment support tools

- **MARVdash** – To facilitate the deployment of the streaming pipeline components and its integration, the project partners agreed on using Kubernetes. The core element to facilitate this deployment is MARVdash. As explained in D3.2, MARVdash is a dashboard that allows an easy deployment and management of Docker containers and services using customisable templates in Kubernetes environments, as presented in more detail in the preceding section (Section 3.1). Some issues to connect servers behind NAT via Kubernetes have been solved in MARVEL using VPN, as explained in D3.2. It is worth noting that for R1, due to legal restrictions in accessing raw audio and video streams and also restrictions associated with access to MT network and devices, some of these (specifically, in MT1 and MT3 use cases) are not deployed in Kubernetes and therefore not managed by MARVdash. Nevertheless, these nodes are part of the streaming pipeline.
- **HPC and Cloud infrastructure** – The infrastructure provided by PSNC is the final destination of the streaming data pipeline. PSNC offers computing and storage resources in terms of HPC infrastructure (Eagle cluster) and virtualised private cloud (LabITaaS), allowing efficient deployment of the MARVEL framework and its validation. The role of the Eagle cluster is to provide the HPC environment accessing GPU for training AI models of various MARVEL components. While the most computation-intensive tasks, such as model training, are performed on the Eagle supercomputer, we use cloud infrastructure as a base for deploying the MARVEL software stack. Managing these resources is possible using the OpenStack web interface, which allows for a flexible allocation of computing resources and various class storage services connected with the HPC system. While cloud and HPC services

are separate entities from hardware and software perspectives, a fast, dedicated (multiple 100Gbit) network connection is established, allowing for efficient data exchange between components running on either service. A detailed description of both infrastructures and the storage and data analytics system is included in the dedicated document D5.3 presenting the report on the HPC infrastructure and resource management aspects used in the first integrated version of the MARVEL framework.

AV data processing

- **Anonymisation close to the source** – In order to comply with legal and ethical requirements regarding privacy, the raw data from cameras and microphones passes a set of anonymisation processes. By executing the VideoAnony and AudioAnony+VAD components over the raw video and audio streams respectively, the result of the process is a set of anonymised videos (RTSP) and/or audio streams to which the different inference models subscribe. This is the start of the streaming data pipeline. More information about VideoAnony and AudioAnony can be found in Section 3.4 and in deliverables D3.2 and D5.4 and in the future deliverable D3.3.
- **Management of AV data – StreamHandler** – The output of the Anonymisation components is also directed to the StreamHandler for AV data management. This component receives anonymised AV streams, segments them with user-defined configurations (segmentations), and stores in MinIO database for on-request future inspection by the end user.
- **Inference models execution** – The inference models (AT, AVAD, AVCC, CATFlow, TAD, SED, VCC and VAD) used in the R1 of MARVEL are AI components included in the streaming data pipeline by connecting to the AV streams that are the output of the Anonymisation components (RTSP or audio streams). An exception to this is the VAD inference model, which uses the non-anonymised (raw) audio data in order to provide onset and offset times of speech segments, which are fed as inputs to the AudioAnony component to indicate which audio segments need to be anonymised. The inference models instances are typically deployed in Kubernetes using MARVdash per use case and AV stream. The results of the execution of each model in real-time is a message per element, such as per frame processed (e.g., in the case of VCC/AVCC), or per sequence of frames (e.g., in the case of AVAD/ViAD, to the nearest MQTT broker in the layer the component is deployed).

Data Management Platform streaming pipeline

- **MQTT brokers** – In order to facilitate a loosely coupled integration between the outputs of the inference models and the Data Management Platform, the streaming inference results pipeline relies on the usage of MQTT brokers. MQTT offers a lightweight message brokerage that enables fast communication following the publish-subscription paradigm. All inference models participating in the streaming pipelines have an instance in the specific layer (edge, fog or cloud) and output their messages to the specific instance of the MQTT broker deployed typically in the same layer. This way, the output data of the inference models is published in the broker and available for any other component by subscription.
- **DatAna** – DatAna is a Data Management Platform that enables data gathering, transformation and transmission based on the Apache NiFi ecosystem (more information available in D2.2). The role of DatAna in the streaming inference pipeline

is that of processing and moving the outputs from the inference models from the edge to the cloud. A DatAna instance typically runs in the same layer as the MQTT where the inference models outputted their results, subscribes to the broker and process and transform the data to a specific data model for MediaEvent, Alert, or Anomaly defined for MARVEL. This is done in each specific layer of the E2F2C according to the architecture of the use cases. Once the data is transformed, the resulting outputs are transmitted ultimately via a taxonomy of instances of Apache NiFi residing on each layer to the cloud using the NiFi S2S protocol. Once in the cloud, the results are moved to specific Kafka topics provided by the DFB for further processing and storage prior to visualisation. All these processes are done in real-time in a stream fashion due to the ability of NiFi technology to process multiple concurrent data flows.

- **DFB** – The DFB resides on the MARVEL cloud and receives the data from DatAna in dedicated Kafka topics and stores the inference results it receives as Alert, Anomaly or MediaEvent. It makes the data available in real-time to SmartViz and Data Corpus completing the real-time aspects of the streaming data pipeline. Additionally, the DFB persists all inference results in an Elasticsearch database and exposes a REST API used currently by SmartViz to enable access to the historical data stored in Elasticsearch.

Visualisation

- **Decision-making toolkit** – The Decision-making toolkit (DMT) is the key means of interaction with MARVEL end-users. It serves as the key interface of all processes performed in the MARVEL data pipeline so that the users can consume the extracted insights. It receives data input from the DFB, as well as insights from some E2F2C deployed models that allow the visualisation of detected events and anomalies. The DMT can be envisioned as a dashboard with pre-configured widgets according to the use cases and the preferences of users. SmartViz is a data visualisation toolkit that constitutes the UI of the DMT. SmartViz enables the end-users to interact with and gain a solid understanding of data, drill into more detailed information, discover patterns and correlations of data items and to lead them in making data-driven decisions. SmartViz functionalities include advanced visualisations of detected events, demonstration of the related statistical data via different visualisation widgets, data filtering options, and flexible interfaces.
- **Data Corpus** – Data from the piloting smart city environment is maintained in the Data Corpus. There, the user can interact with the Corpus via a flexible and user-friendly graphical interface. With web-based technologies, the user stories and handles audio-video data (e.g., viewing, adding, deleting, etc.). The user story starts with the main front page of the user interface (UI) where the MARVEL Data Corpus user can have an overall overview of the current status of the uploaded datasets. Moreover, the latter works as a starting point for the user in order to add any new dataset, view, or alternate existing ones. The user can obtain on a single page, a complete synopsis of the uploaded data inside the Corpus and the corresponding actions over them.

3.3 Batch processing pipelines for system optimisation

3.3.1 AI model training

To a large extent, the operation of the MARVEL framework in the ‘AI Inference Pipeline’ relies on AI components that require Machine Learning (ML), Deep Learning (DL), Federated Learning (FL) and compression mechanisms for the formulation and optimisation of their associated AI models. While each component may adopt a different approach to perform such AI Training processes according to its nature and objectives, all these processes depend on the availability of relevant datasets and annotation information. In addition, some of the MARVEL AI models are significantly case-specific, i.e., they require training to occur on the datasets that are representative of the data that this model will analyse in the inference phase, in order to produce models that achieve accurate inference (e.g., streams of a camera employed in a given use case should share the same FoV for both training and inference etc.).

AI Training of MARVEL AI components deployed using each component's own resources and using private or publicly available datasets that have not necessarily emerged from the MARVEL data collection activities. However, in the context of the MARVEL R1 integration activities, dedicated processes were established to support AI training of MARVEL components that took the form of structured system architecture. The AI Training processes foreseen by this architecture can be activated on an on-demand basis or be driven by a periodical schedule, subject to the availability of new datasets in the Data Corpus. The associated AI Training architecture ensures that:

- AI components in training mode can gain access to the data that is aggregated at the Data Corpus for AI training purposes;
- AI components can store and retrieve different versions of their associated AI models that are produced through AI Training with different datasets;
- AI models can be further enhanced using special compression and optimisation techniques to provide improved performance in the 'AI Inference Pipeline' (DynHP):
- Federated Learning can be applied in AI training processes to enable model training over sensitive or private datasets (FedL).

AI components in training mode deployed at the fog or cloud infrastructure nodes may access the Data Corpus, deployed at the cloud (PSNC HPC via OpenStack) to retrieve raw (anonymised) datasets and annotation information and use these data to feed their own AI training processes. The training process typically also involves a quality control phase (benchmarking), which can be automated or manually operated to ensure that the trained model meets certain functional and performance criteria.

After concluding AI training tasks, AI components in training mode can store the updated AI models that they produce in a central AI Model Repository that is hosted in the cloud (PSNC HPC via OpenStack) using specified data structures and naming conventions. Similarly, each AI component to be deployed in inference mode (at an arbitrary infrastructure layer) can access and retrieve the AI model of choice, associated with their functions, from the AI Model Repository.

Figure 4 provides the deployment and runtime view of the MARVEL reference architecture for AI Training that is applied to support the 'AI Inference Pipeline' in all R1 use cases.

The following list includes the descriptions of the annotation labels associated with each I/O interface connection in the architecture diagram of Figure 4:

1. an AI component in training mode receives AV data from the Data Corpus and performs ML/DL training with the received dataset;
2. an AI component in training mode sends an updated model to the AI Model Repository after a training procedure is complete;
3. an AI component in inference mode receives an updated AI model from the AI Model Repository;

4. DynHP receives a (supported) AI model from the Repository and processes it to compress/optimise it. Subsequently, DynHP sends a compressed/optimised AI model to the Repository; more details on DynHP are provided in Section 3.3.2.;
5. the FedL server receives a compressed/optimised AI model from the AI Model Repository and returns an updated version of the model after it has been processed using federated learning;
6. the FedL server exchanges information bidirectionally with the FedL client to process a compressed/optimised AI model and update it after performing federated learning; more details on FedL are provided in Section 3.3.3.

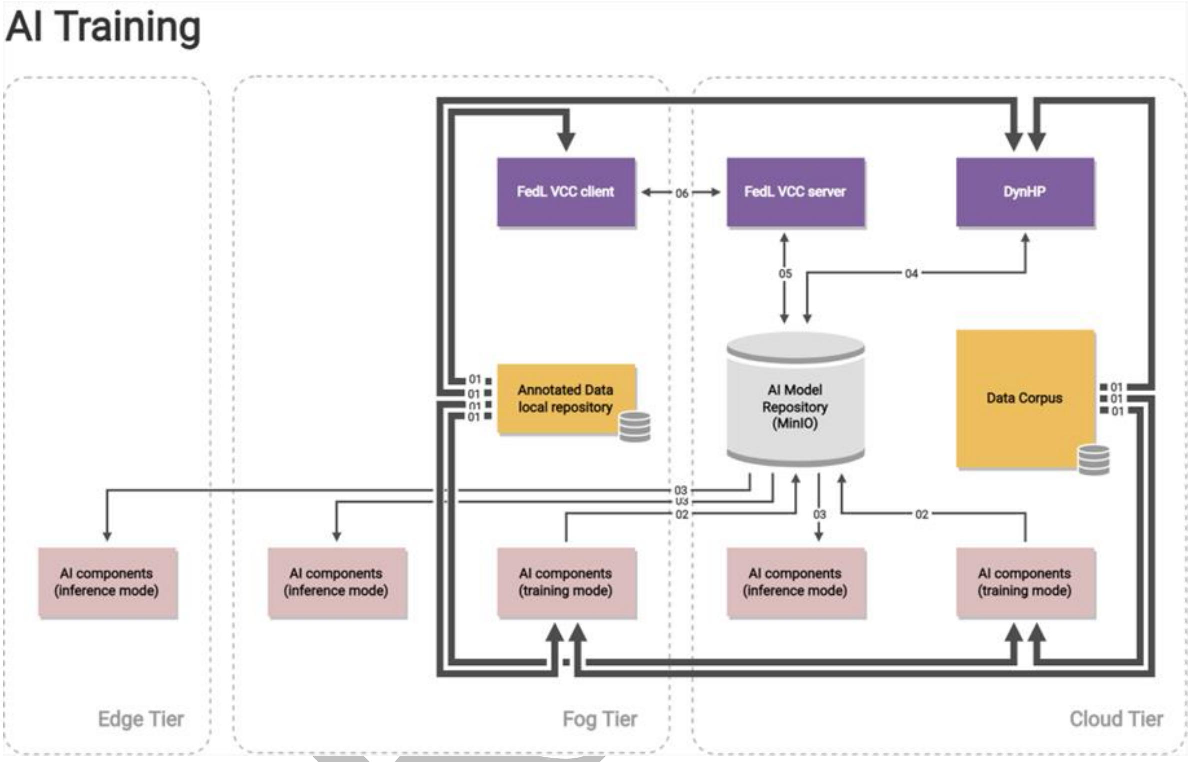

AI Training
FedL VCC client
06
FedL VCC server
DynHP
05
04
01
01
01
01
Annotated Data local repository
AI Model Repository (MinIO)
Data Corpus
01
01
01
03
03
02
03
02
AI components (inference mode)
AI components (inference mode)
AI components (training mode)
AI components (inference mode)
AI components (training mode)
Edge Tier
Fog Tier
Cloud Tier

Figure 4: AI training pipeline – runtime and deployment view

3.3.2 AI model compression: DynHP

DynHP is a methodology for training a deep neural network model and compressing them simultaneously. The type of compression operated by DynHP is pruning, i.e., the parameters of a DNN are zero-ed at training time. DynHP operates structured pruning where the idea is to “remove” the entire neurons of a DNN.

Compression-wise, structured pruning is more effective since it allows for removing entire groups of parameters. DynHP was initially designed with memory constraints in mind, i.e., the main idea behind it was to develop a methodology to free the memory as the training and compression proceed incrementally. This approach is called incremental hard pruning. With “hard pruning”, the parameters that are “switched off” during training cannot be recovered afterwards. In this way, at least in principle, it would be possible to free the memory occupied by the zeroed parameters (that should be removed from the network topology). This procedure might result in a slightly more complex and unstable training process than the one performed on an “uncompressed” model. DynHP tries to mitigate these additional difficulties by adaptively tuning some of its training parameters (i.e., the mini-batch size).

The main benefits coming from this procedure are connected to the reason behind its development: having a way to reduce the size of a Deep Neural Network model directly on the device that is training it. Resource-constrained devices such as edge or fog devices can benefit from such a methodology as they can, at least in principle, operate model compression without resorting to remote cloud facilities. Finally, it might also help preserve data privacy and ownership because the process can be performed without the intervention of cloud-based procedures.

For the first MARVEL prototype (R1), DynHP will be used to compress the Visual Crowd Counting (VCC) Deep Learning model. VCC takes as input a video frame and returns the number of people detected in the frame. The training and compression are performed using the data collected in the UNS1 use case.

3.3.3 Federated model training: FedL

Federated Learning allows for distributed privacy-preserving training of Deep Learning models. In that, the data never leaves its source, and the training is performed near the data collection point. Another benefit is that there can be multiple Federated Learning clients which perform the training with the local data, which is then passed to a central orchestration point – Federated Learning server which is used to average all the models and provide a global model which is created by combining all the client models. In that way, we obtain a model similar to a model which is trained on all the data, but the client data never leaves its source. Only the internal parameters of the Deep Learning models are shared with the server and not the training/input data. From a deployment standpoint, in a federated or a distributed system, it is then easy to add new data collection points (for example new camera location hubs in smart cities) to the existing architecture.

To incorporate federated learning, we use MARVEL developed FedL component. FedL contains an implementation of the Federated Learning training paradigm for various Deep Learning models. For the MARVEL project needs, the FedL component also develops a custom Federated Learning strategy which optimises the learning process for flaky client-server communication. This issue with communication is expected in large heterogeneous systems and the custom model merging strategy is to address it. The custom strategy (names NUS – non-uniform sampling strategy) allows for clients to be temporarily unavailable during learning. It also only requests client training results if the client data is valuable to the global model, based on several metrics such as the number of client data points, model metrics such as accuracy, model gradient variance, client availability history and other information.

For the first MARVEL prototype (R1), the Visual Crowd Counting (VCC) Deep Learning model was used. This VCC model uses an image (video frame) as input data and outputs the number of people detected in that input. The model was adapted without architectural changes for use within the FedL component; we refer to the resulting FedL component instantiation as VCC-FedL. The cameras collecting the data then do not need to share the images with the entire MARVEL architecture. They can only share the images with their data collection point which can be used as a federated learning client in the MARVEL system (FedL client), thus preserving privacy of the people in the images.

For R1, three different use cases were selected for the VCC-FedL implementation: GRN4, MT1 and UNS1. Hence, each of the selected use cases implements one FedL client being locally fed with training data from the respective data collection sites; the details on the data collection sites and the data used within VCC-FedL training, for the three use cases/sites, can be found, respectively, in Sections 5.2.1, 6.2.1 and 8.2.1.

3.4 Ethics and privacy assurance and anonymisation

3.4.1 Hierarchical data distribution: HDD

HDD is a set of distributed algorithmic schemes for guaranteeing latency requirements while effectively prolonging network lifetime in networked settings. The current MARVEL design of HDD considers the problem of Apache Kafka data topic partitioning optimisation. Apache Kafka uses partitions to scale a topic across many brokers for producers to write data in parallel, and also to facilitate the parallel reading of consumers. Even though Apache Kafka provides some out-of-the-box optimisations, it does not strictly define how each topic shall be efficiently distributed into partitions. The well-formulated fine-tuning that is needed in order to improve an Apache Kafka cluster performance is still an open research problem. HDD first models the Apache Kafka topic partitioning process for a given topic. Then, given the set of brokers, constraints and application requirements on throughput, OS load, replication latency and unavailability, HDD formulates the optimisation problem of finding how many partitions are needed and shows that it is computationally intractable, being an integer program. Furthermore, HDD implements two simple, yet efficient heuristics to solve the problem: the first tries to minimise and the second to maximise the number of brokers used in the cluster.

The HDD approach comes with a significant user benefit: its paradigm is more suitable for modern streaming application developers that consist of vertically separated engineering teams, each one managing a loosely coupled sub-module. In comparison to old-fashioned data-driven applications with centralised, shared data management systems, the distributed and non-synchronous nature of stream handling enables stakeholders to construct their sub-modules in an event-driven way which heavily relies on optimised event-data passing. Also, a significant user-tailored added value is the semi-automated responses that HDD gives to fundamental Apache Kafka setting questions, such as how many partitions shall be allocated to a topic, or how many brokers shall be used, or how to effectively respect the application constraints.

3.4.2 Ethics and privacy assurance

Ethics, Privacy and Data Protection compliance management is the backbone for the successful realisation of MARVEL. It enables ethical and privacy functions to be implemented throughout the project lifecycle. Transversal by its nature, ethics, and privacy are relevant for all project outcomes and activities across all tasks and WPs. As such, a review of general governance and the way privacy is managed in MARVEL allows for the identification of the most efficient ways to implement compliance requirements in practice.

The ethics, privacy and data protection compliance challenges and how they have been handled within the project realisation are outlined in the D2.3 as well as addressed within WP9 and ethics deliverables.

The monitoring of compliance activities has been focused on the identification of regulatory requirements as well as on the technological developments within MARVEL. Implementation of the requirements in practice is governed by strong cooperation among the project partners and led by the Project Coordinator (PC). Compliance monitoring is the niche for ensuring legal and ethical compliance during the entire project life cycle. It is the strong integrative factor for converging security, privacy, and data protection requirements in project outcomes. It ensures further collaboration among project partners around ethical and legal issues/challenges. Therefore, activities within this task ensure a coherent approach to legal and ethical aspects and support the project's risk management activities through ongoing compliance monitoring and support as per need.

The compliance management was particularly engaged about legal grounds for ensuring lawful processing of personal data, satisfaction of transparency requirements, implementation of Trustworthy AI standards, and implementation of appropriate technical and organisational measures to secure data. The opinions about the necessity for Data Protection Impact Assessments (DPIA) are provided, as well as the DPIA has been conducted where it has been needed.

3.4.2.1. Lawful processing of personal data

The project consortium takes with utmost care the needed steps to ensure the lawful processing of personal data within MARVEL. Concerning data processing within the project pilots, the legal ground for lawful data processing is data subject consent. The consent form will include a written information sheet that will present all relevant information about the project and pilots as well as collected data and protocols, data protection rights, and contact details. The consent declaration text is provided in clear, easy-to-understand, and plain language. The participants are asked to read the sheet and subsequently provide their consents. Participants get an opportunity to ask questions about the project and data processing practice, and they are allowed to withdraw their consent without any consequence.

Apart from data subject consent, there are secondary legal grounds for processing personal data. A significant portion of the participants that are involved in the project pilot activities and project activities are members of the MARVEL project partners (internal staff). In such cases, informed consent is requested, and lawful processing of personal data relies on legitimate interest grounds.

More details about lawfulness of personal data processing are available in D9.2 – POPD

3.4.2.2. Transparency

Within the MARVEL requirements regarding the principle of transparency are met. Any information or communication relating to the processing of personal data within MARVEL is easily accessible as well as easy to understand. That is a particular case with the information sheets that provide information about project pilots. Where it was requested (by ethics requirements) project partners have developed project-specific by presenting all needed information. Information sheets are given in ethics deliverable D9.1-Humans.

The MARVEL project website also contains relevant policies that provide information in accordance with Art. 13 of the GDPR about data processing practices at the website.

3.4.2.3. Trustworthy AI

The MARVEL framework includes a wide variety of AI components. Thus, one of the MARVEL goals is to develop an AI system in accordance with relevant ethical standards and following the best practice. Therefore, it was decided to rely on the principles of diversity, non-discrimination, and fairness for building a Trustworthy AI system. One of the challenges was the elimination or mitigation of potential AI bias in the developed AI models. For that purpose, the analysis of all AI components has been done and potential risks were assessed. The analysis demonstrated that the likelihood of AI bias within the MARVEL AI system is considered as low. However, MARVEL plans to consider all potential risks and hence the set of precautions that will be applied in the context of AI development.

The analysis of AI components and precautionary measures are presented in ethics deliverable D9.4-OEI.

3.4.2.4. Data security

Concerning data security, the greatest efforts were on the development of anonymisation/pseudonymisation strategies, tactics and techniques, Edge-to-Fog-to-Cloud security as well as other safeguards to protect data.

All use cases and pilot executions within the MARVEL project involve audio and video recordings of real-life events in public spaces such as town squares, and road & transport physical infrastructures, such as road network junctions, streets, and parking lots (more detail about the use cases and pilot are given in ethics deliverable D9.3-POPD). To achieve this objective, anonymisation techniques were considered and developed as the project progressed. The MARVEL anonymisation/pseudonymisation strategy was developed and presented in the ethics deliverable D9.2. This strategy contains specific tactics concerning processing different types of data.

To preserve confidentiality, integrity and availability of data (CIA triad) that is the essence of information security as well as to properly enforce information security management, MARVEL applies additional security measures. Combining standard cryptographic protocols like Secure Sockets Layer (SSL) and/or Transport Layer Security (TLS) to provide endpoint encryption with a hash algorithm (used to add an additional layer of security) MARVEL leverages the security features of Trusted Execution Environments (TEE) and apply specific measures to protect the CIA of data at rest. Explained combination of measures composes the infrastructure known as Edge-to-Fog-to-Cloud (E2F2C). More details are given in the ethics deliverable D9.2 - POPD and D2.3.

Finally, for the purpose of satisfying the requirements concerning the implementation of appropriate technical and organisational measures for securing data additional measures are implemented and they are presented in D2.3 and in more details in ethics deliverable 9.2-POPD.

3.4.2.5. Data Protection Impact Assessment

One of the requirements imposed by the ethics evaluator is to provide opinions on whether data protection impact assessments are needed with regard to data processing practices within MARVEL pilots. The opinions were provided in ethics deliverable D9.3 – POPD. Provided opinions present the joint position of the PC and project partners including the legal and ethical manager. The DPIA with regard to data processing conducted in the Municipality of Trento is available in D2.3.

3.4.2.6. Ethics requirements

The ethics requirements received as part of the EC Ethics Appraisal Scheme have been successfully met (WP9). Therefore, all pertinent deliverables, concerning, for instance, anonymisation, identification of participants, data processing, and surveillance risk management are expected to be conceived in synergy with ethics deliverables and compliance management report (D2.3). This report takes into consideration WP9 and considers it an integral part of the ethics, privacy, and data protection strategy of MARVEL. Finally, the project has benefited from the established Ethics Advisory Board (EAB). This independent external body has ensured adherence to privacy laws and regulations, handling of anonymisation processes where applicable, handling of informed consent processes, processes associated with data security, secondary use of data, involvement of third countries in the project, and more. The EAB report is available in D9.6 and D2.3.

3.4.3 Audio-visual anonymisation: VideoAnony, VAD and AudioAnony

Visual anonymisation – The raw video feeds coming from the CCTV cameras from each pilot site are anonymised with VideoAnony. This component first detects faces and car number plates and then achieves identity obfuscation via image redaction methods, starting from classic image processing techniques, such as blurring, towards the more advanced GAN-based face swapping techniques, which are under development within the MARVEL project. VideoAnony component receives the incoming raw video stream either via RTSP or direct cable access, and performs the face/car number plate detection with customised Yolov5 models and then anonymisation of the detected region of interest, and finally streams the anonymised videos via the RTSP server. The component will be deployed in either edge (e.g., the use cases in GRN and UNS) or fog device (e.g., the use cases in MT and GRN) depending on the infrastructure design. Its real-time performance depends on the available on-board processing power of devices at each pilot site. VideoAnony was used in the MVP for providing anonymised videos into the Data Corpus in an offline manner. In R1, VideoAnony is integrated into the MARVEL platform in all selected pilot cases where visual anonymisation is deemed necessary.

Audio anonymisation – In R1 use cases, and also in the whole project, the audio anonymisation pipeline is made up of the composite component AudioAnony and VAD. These two low-level components are highly correlated and operate together in order to ensure privacy-compliance when it comes to audio, among those privacy compliance measures is the anonymisation of the speech in the audio streams. Speech often contains sensitive features that can be used to identify the speaker, and if those features are transferred like they are, then we will have a violation of privacy constraints. Therefore, the goal of this composite is to detect in the first stage the speech boundaries in the audio stream and anonymise them in the second stage. To ensure the non-transfer of sensitive features to the fog and cloud layers, the anonymisation must take place on premise. That is why the composite component will be deployed on edge devices (Raspberry Pi devices for the relevant use cases).

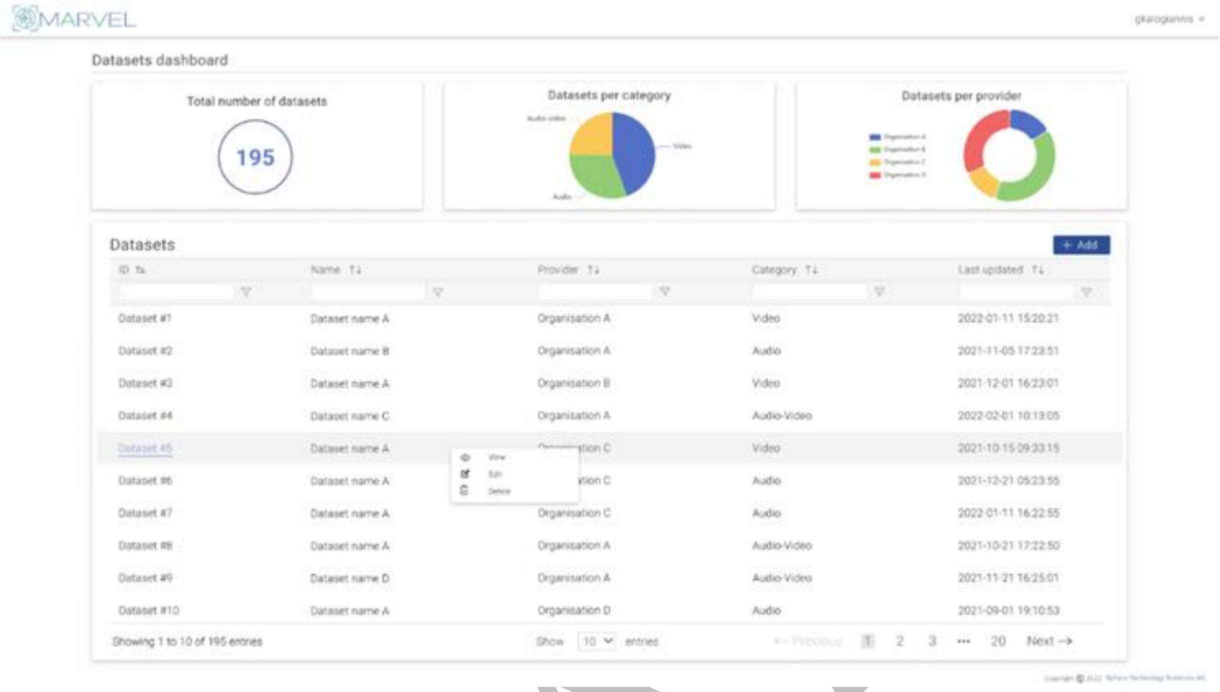
3.5 Data Corpus realisation

The Data Corpus implements a Big Data repository that can be used by all use cases. The pilots collect data in real-time, which is then anonymised and annotated by MARVEL components (e.g., VideoAnony). The resulting datasets can be stored in the Data Corpus, which is deployed in MARVEL's backend/cloud infrastructure. Storing of the audio/video files themselves is performed by the Hadoop Distributed File System (HDFS) while the administration of this repository is supported by the HBase distributed database (an open-source non-relational distributed database). Data streams sent to StreamHandler for persistent storage can also be ingested to the Corpus automatically. Alternatively, users can upload datasets via a graphical user interface (GUI). Typically, annotated (labelled) data are uploaded via the Data Corpus GUI, while AV data processed by anonymisation only (unlabelled) arrive at Data Corpus from StreamHandler. As an additional mechanism and data source for Data Corpus, MARVEL also enables storing inference results acquired during the executions of the streaming pipeline. These are transmitted to the Data Corpus from the DFB Elasticsearch database.

The user can interact with the Corpus via a flexible and user-friendly graphical interface that has been developed in Angular[4] and can be utilised by users to review and retrieve the ingested datasets. With web-based technologies the user stores and handles audio-video data (e.g., viewing, adding, deleting, etc.). The user story starts with the main front page of the user interface (UI), where the MARVEL Data Corpus user can have an overall overview of the

[4] https://angular.io/

current status of the uploaded datasets. Moreover, the latter works as a starting point for the user, in order to add any new dataset, view, or alternate existing ones. The user can have on a single page, a complete synopsis of the uploaded data inside the Corpus and the corresponding actions over them.



Datasets dashboard

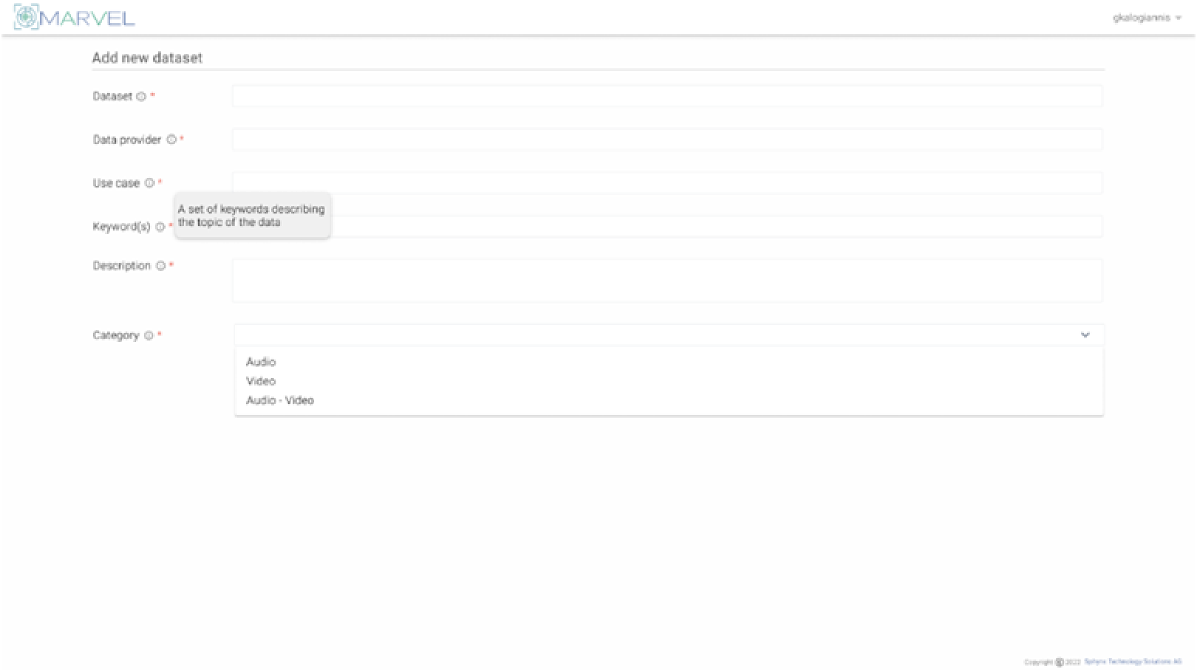
Datasets

ID	Name	Provider	Category	Last updated
Dataset #1	Dataset name A	Organisation A	Video	2022-01-11 15:20:21
Dataset #2	Dataset name B	Organisation A	Audio	2021-11-05 17:23:51
Dataset #3	Dataset name A	Organisation B	Video	2021-12-01 16:23:01
Dataset #4	Dataset name C	Organisation A	Audio-Video	2022-02-01 10:13:05
Dataset #5	Dataset name A	Organisation C	Video	2021-10-15 09:33:15
Dataset #6	Dataset name A	ation C	Audio	2021-12-21 05:23:55
Dataset #7	Dataset name A	Organisation C	Audio	2022-01-11 16:22:55
Dataset #8	Dataset name A	Organisation A	Audio-Video	2021-10-21 17:22:50
Dataset #9	Dataset name D	Organisation A	Audio-Video	2021-11-21 16:25:01
Dataset #10	Dataset name A	Organisation D	Audio	2021-09-01 19:10:53

Showing 1 to 10 of 195 entries   Show 10 entries   ← Previous 1 2 3 ... 20 Next →

Figure 5: Data Corpus – Datasets overview

From this point, the user can add, edit, view, or delete the selected dataset with a simple click since the main page of the UI will redirect him/her to the corresponding page of the interface. When it comes to adding data to the Corpus, the interface will guide him/her via a single page where a series of related fields must be filled.



Add new dataset

Dataset ⓘ *

Data provider ⓘ *

Use case ⓘ *

Keyword(s) ⓘ *

A set of keywords describing the topic of the data

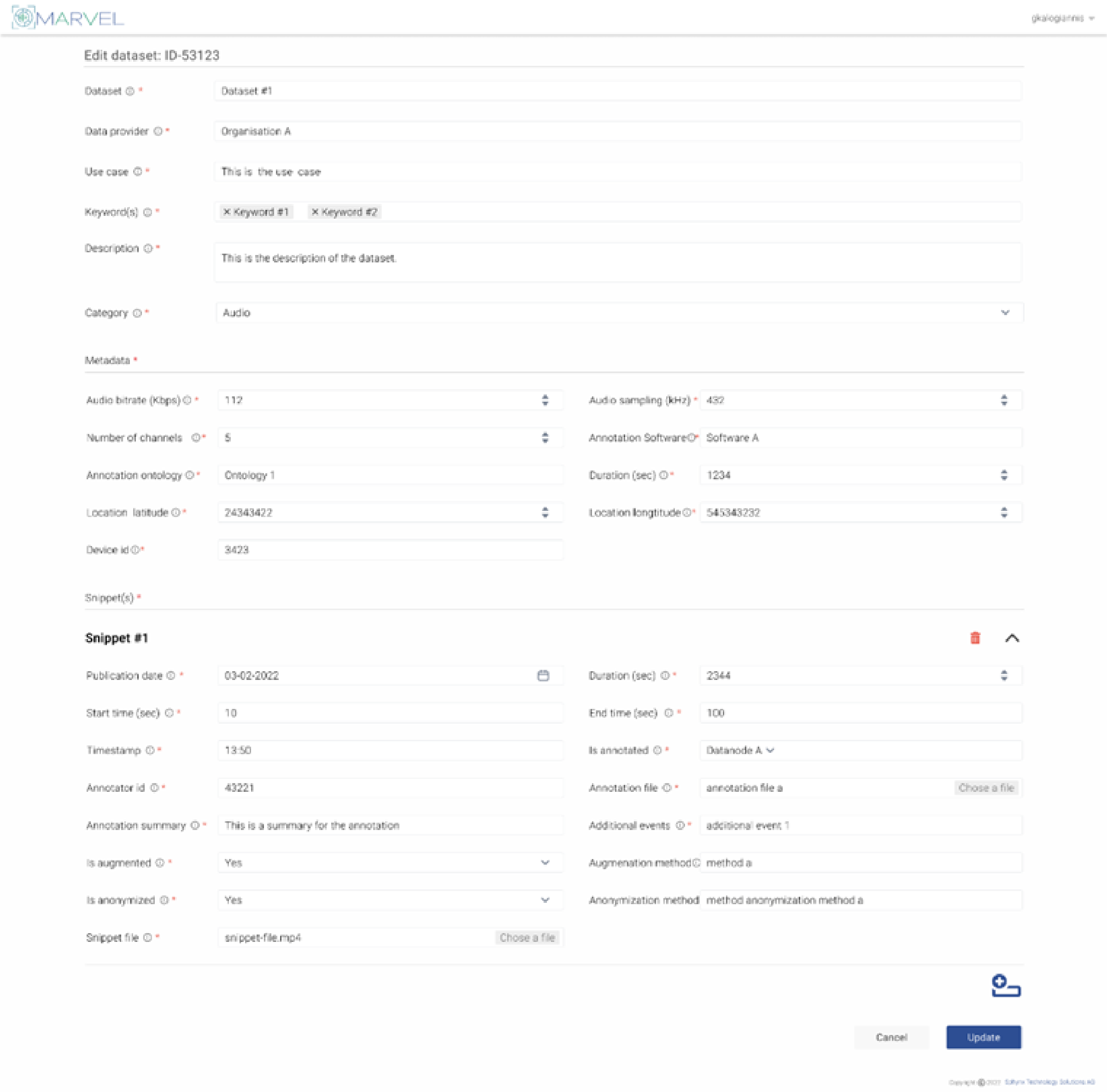
Description ⓘ *

Category ⓘ *

Audio
Video
Audio - Video

Figure 6: Data Corpus – Add new dataset

Editing a dataset is as simple as it can be and can be done through a single page also. Since the relative correlated information needed to be filled by the user is quite a lot, the UI of the Corpus, via a uniform view, gives the ability to have an overall control of his/her entries.



Edit dataset: ID-53123

Dataset *	Dataset #1
Data provider *	Organisation A
Use case *	This is the use case
Keyword(s) *	× Keyword #1 × Keyword #2
Description *	This is the description of the dataset.
Category *	Audio

Metadata *

Audio bitrate (Kbps) *	112	Audio sampling (kHz) *	432
Number of channels *	5	Annotation Software*	Software A
Annotation ontology *	Ontology 1	Duration (sec) *	1234
Location latitude *	24343422	Location longtitude *	545343232
Device id *	3423		

Snippet(s) *

Snippet #1

Publication date *	03-02-2022	Duration (sec) *	2344
Start time (sec) *	10	End time (sec) *	100
Timestamp *	13:50	Is annotated *	Datanode A
Annotator id *	43221	Annotation file *	annotation file a Chose a file
Annotation summary *	This is a summary for the annotation	Additional events *	additional event 1
Is augmented *	Yes	Augmenation method	method a
Is anonymized *	Yes	Anonymization method	method anonymization method a
Snippet file *	snippet-file.mp4 Chose a file		

Figure 7: Data Corpus – Update existing dataset

Finally, the MARVEL Data Corpus user can view and delete a specific dataset by just selecting it and performing the relative action. Upon successful deletion, the dataset list presented on the front page of the UI will be automatically refreshed.

Datasets can be used either internally by AI/ML components to improve their operation and analysis of piloting live streams or externally by research and industrial communities to facilitate their research and developed product/algorithms. The user can search for datasets and download them (manually or via a programmable interface).

Moreover, the MARVEL user can also apply video and audio processing techniques to the original data and produce augmented versions of it. Several Python-based augmentation scripts were developed in order to automatically augment the datasets provided by the pilots using

Keras[5], TensorFlow[6] and imagaug[7] open-source software libraries. The imgaug library, which was used for image-based tasks, supports a large range of augmentation techniques, allows the techniques to be easily combined and can execute the techniques in random order or on multiple CPU cores. The supported video and audio augmentation techniques are referred below.

- Video processing/augmentation techniques
  - Geometric transformations
  - Flipping
  - Colour space
  - Cropping
  - Rotation
  - Translation
  - Noise injection
  - Colour space transformations
  - Kernel filters
  - Mixing images
  - Random erasing
  - Style transfer
- Audio processing/augmentation techniques
  - Adding Gaussian noise
  - Time stretch
  - Pitch shift
  - Mixup
  - SpecAugment
  - MixedSpecAugment
  - SamplePairing

3.6 Integration process overview

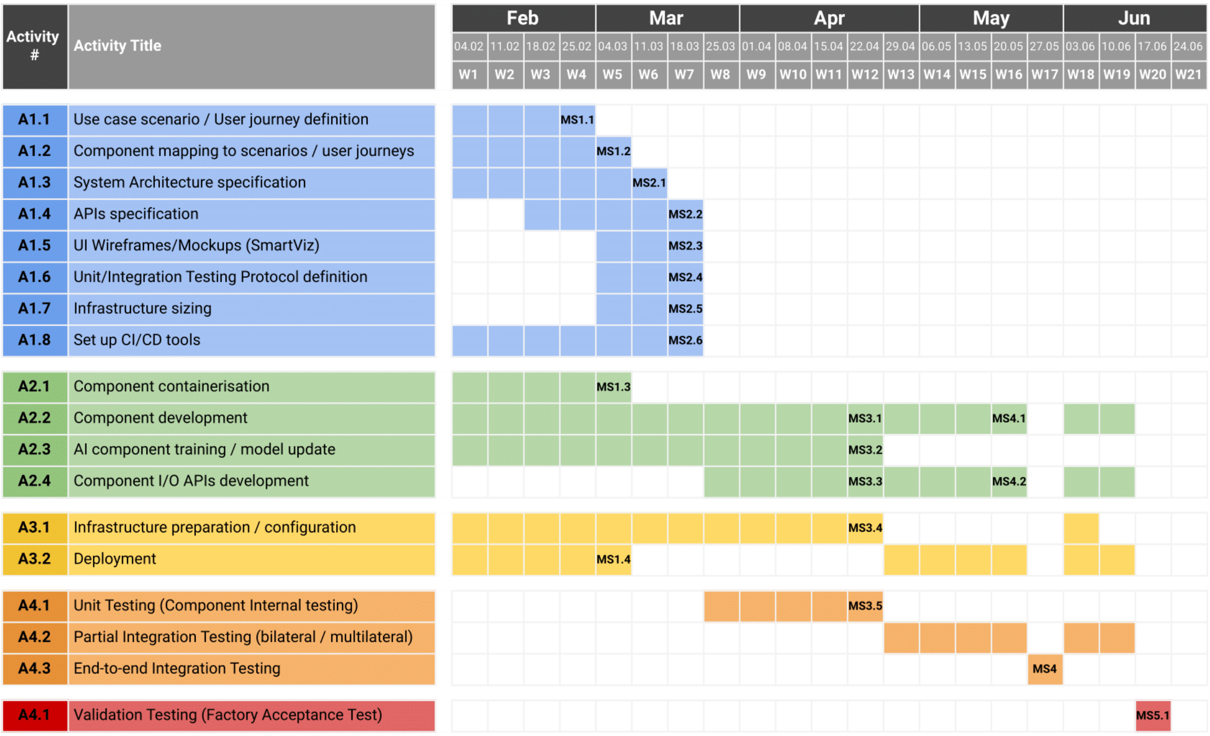
The integration process of R1 was responsible for delivering a unified, operational system that could address the needs of the selected use cases. This was a challenging task due to the large number (32) of independent, heterogenous components developed by 15 different partners that were required to be integrated and communicate effectively with each other, and together achieve a goal of greater scope than their individual functionalities offer. Furthermore, R1 integration faced several complex aspects, as R1 needed to address five different use cases belonging to three pilots by implementing different system designs, access and manage multiple AV data sources from the pilots, manage the computational infrastructure resources at each pilot distributed across three layers (Edge-Fog-Cloud) and organise all the associated deployment and testing activities.

In order to structure the various integration activities and deliver R1 as an operational prototype within the allocated time frame, the R1 integration process was organised according to a detailed time plan that was prepared at the outset of R1 integration activities in the form of a Gantt Chart (Figure 8).

[5] https://keras.io/

[6] https://www.tensorflow.org/

[7] https://imgaug.readthedocs.io/en/latest/



Activity #	Activity Title	Feb				Mar				Apr					May				Jun			
		04.02	11.02	18.02	25.02	04.03	11.03	18.03	25.03	01.04	08.04	15.04	22.04	29.04	06.05	13.05	20.05	27.05	03.06	10.06	17.06	24.06
		W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15	W16	W17	W18	W19	W20	W21
A1.1	Use case scenario / User journey definition	■	■	■	MS1.1																	
A1.2	Component mapping to scenarios / user journeys	■	■	■	■	MS1.2																
A1.3	System Architecture specification	■	■	■	■	■	MS2.1															
A1.4	APIs specification			■	■	■	■	MS2.2														
A1.5	UI Wireframes/Mockups (SmartViz)					■	■	MS2.3														
A1.6	Unit/Integration Testing Protocol definition					■	■	MS2.4														
A1.7	Infrastructure sizing					■	■	MS2.5														
A1.8	Set up CI/CD tools	■	■	■	■	■	■	MS2.6														
A2.1	Component containerisation	■	■	■	■	MS1.3																
A2.2	Component development	■	■	■	■	■	■	■	■	■	■	■	MS3.1	■	■	■	MS4.1		■	■		
A2.3	AI component training / model update	■	■	■	■	■	■	■	■	■	■	■	MS3.2									
A2.4	Component I/O APIs development								■	■	■	■	MS3.3	■	■	■	MS4.2		■	■		
A3.1	Infrastructure preparation / configuration	■	■	■	■	■	■	■	■	■	■	■	MS3.4						■			
A3.2	Deployment	■	■	■	■	MS1.4								■	■	■	■		■	■		
A4.1	Unit Testing (Component Internal testing)								■	■	■	■	MS3.5									
A4.2	Partial Integration Testing (bilateral / multilateral)													■	■	■	■		■	■		
A4.3	End-to-end Integration Testing																	MS4				
A4.1	Validation Testing (Factory Acceptance Test)																				MS5.1	

Figure 8: Week-based time plan in the form of a Gantt chart for organising R1 integration activities

The R1 integration began with design activities that involved the selection of the R1 use cases, complemented by an analysis of the available infrastructure and provisional mappings of components to use cases and infrastructure layers and nodes. This process gave way to the definition of distinct system architectures for each use case, which was coupled with the identification, specification and documentation of the necessary protocols, I/O interfaces and data models to ensure communication between components. In parallel, the user interfaces and overall user experience were designed and tailored to the needs of the involved end-users in the R1 use cases. At each step, the various aspects of design (e.g., complying with use case requirements, infrastructure sizing, component mapping, interface and data model specification, UI/UX) were juxtaposed against each other to ensure that all necessary sides of R1 design were being considered and addressed and that conflicting or underdefined aspects were resolved.

During the design process, the various possibilities were being visually represented using architectural diagrams. Separate diagrams were prepared for the AI inference pipeline of each addressed use case, while a single diagram was prepared for the AI training pipeline, which referred to all use cases. The diagrams were being regularly updated, as the R1 design was progressing to provide a common ground for all parties involved in the design process.

In parallel to the design activities, the development and configuration of individual components were proceeding. Initially, efforts were made to ensure that all components that were foreseen to be involved in R1, were appropriately configured to be delivered in a containerised form, suitable for deployment in a Kubernetes cluster environment through the MARVdash tool. Further development and optimisation of individual components for achieving the necessary functionalities for R1 took place throughout the entire period of R1 integration activities. Following the specification of APIs for communications between components, efforts were also allocated to implementing the associated interfaces and operations at the level of individual components. In parallel, activities included training of the ML models of MARVEL AI components that were applicable in R1, using datasets that were provided by pilot owners.

Efforts were also allocated to properly configuring all infrastructure devices to be seamlessly added to the Kubernetes cluster and be able to host the MARVEL services. In parallel, issues related to data privacy and security concerns were resolved, especially in cases of configuring Edge and Fog host devices. Finally, MARVEL technical partners were active in preparing and updating the container images of their components as well as providing necessary configuration documents to achieve the deployment of the respective services.

R1 was successfully delivered following the deployment of MARVEL components in a unified environment (Kubernetes cluster), bridging multiple infrastructure nodes that were available across the three pilots with the help of the MARVdash tool. R1 was thoroughly tested following procedures of unit testing, partial integration testing, end-to-end integration testing and technical validation testing.

The development, integration and delivery of R1 also depended on organisational mechanisms that were introduced for R1, namely the realisation of weekly technical integration meetings and the establishment of an Integration Board, but also on the implementation of other support tools, mechanisms and methods (version control system, issue tracking system, specification documentation, infrastructure sizing, quality assurance).

- **Weekly R1 Technical Integration Meetings** – Focused on the coordination of integration activities from a technical perspective. The purpose of this series of meetings was to (i) assist in aligning the efforts of technical Partners contributing technologies to be integrated into the MARVEL 1st Prototype (R1), (ii) track the progress of the integration process, (iii) identify and resolve issues that arose.
- **Establishment of an Integration Board** – Introduced to facilitate a more efficient management of integration activities by taking a leading role and initiatives in the relevant processes and helping in swiftly resolving central issues that would arise without needing to involve all technical Partners, whenever possible.
- **Continuous Integration / Continuous Delivery (CI/CD)** – MARVEL adopted an iterative approach for the development and integration of the technical framework based on incremental releases and an agile methodology.
- **Infrastructure sizing** – In order to achieve a functional operation of the MARVEL system, it was necessary to perform estimations of the infrastructure computational resources that would be required at the Edge, Fog and Cloud layers to host the foreseen services to be deployed. At the same time, the pool of available infrastructure resources at the pilot sites (Edge and Fog layers) was also taken into consideration and the foreseen services were also scaled to better match the capabilities of the available resources.
- **Source Version Control system** – In the context of R1 development and integration activities, a source code repository with version control capabilities (GitLab) was set up. The repository was configured to be able to host the source code of individual R1 components, provide a central issue tracking system, maintain a repository of open-source code that was useful to multiple parties, retain the documentation of the integration specifications (I/O Interfaces and data models) as well store deployment and operational information required by some R1 components.
- **Issue Tracking System** – A dedicated Issue Tracking System (ITS) was set up within the MARVEL GitLab repository. Its purpose was to (i) document open issues that required actions from single or multiple parties, (ii) assign issues with actionable items to specific Partners/persons, (iii) serve as a communication channel, (iv) organise and

prioritise issues, including deadlines and association to milestones., (v) monitor and manage the progress of integration activities, (vi) maintain a backlog of unresolved items for long-term planning, (vii) link issues with parts of code, API and data model specifications, hosted on the GitLab repository.

- **Specification Documentation** – Particular attention was paid to the documentation of I/O Interfaces and data models that were applied for the needs of R1 integration. The documentation was maintained on the GitLab repository and was regularly updated.
- **Quality Assurance** – A quality assurance plan was prepared and implemented for R1, involving (i) unit testing (internal testing of individual components), (ii) partial integration testing (integration tests between pairs or groups of components), (iii) end-to-end integration testing (integration tests involving components of the entire pipeline that are deployed at the foreseen infrastructure nodes) and (iv) a final stage of technical validation testing (final and formal end-to-end integration test of the entire pipeline in each use case).

4 GRN3: Traffic Conditions and Anomalous Events

4.1 Integration and framework configuration for GRN3

This section describes the steps taken to integrate the systems such that the GRN3: Traffic Conditions and Anomalous Events could be set up and tested. The following sections describe the architecture and components, the pilot infrastructure and the integration, deployment and testing phase.

4.1.1 Architecture, components and E2F2C streaming data pipeline

The GRN3 realisation of MARVEL architecture is given in Figure 9, while the employed MARVEL components together with their configurations for GRN3 are given in Table 23.

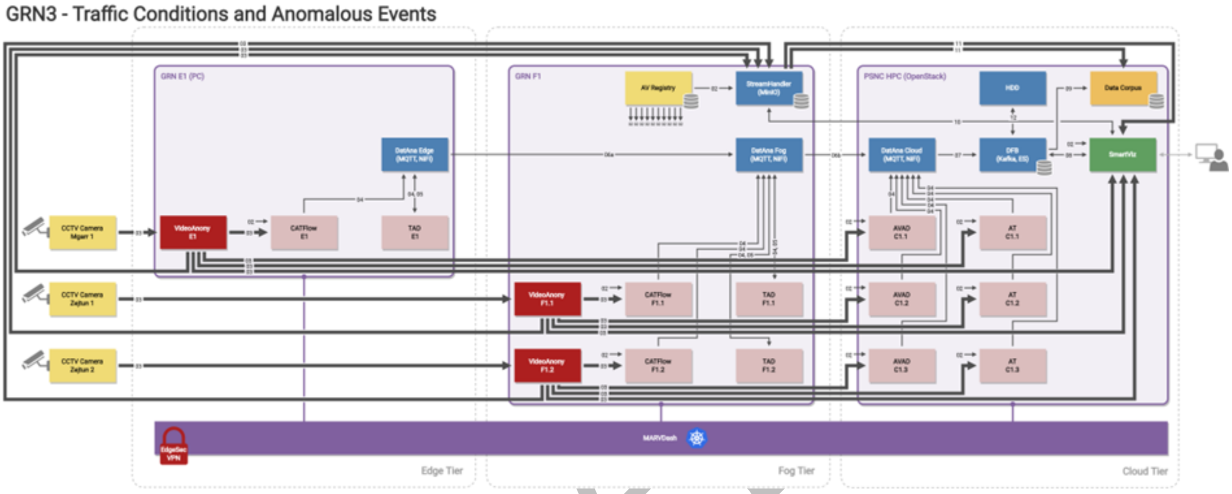


Figure 9: MARVEL R1 deployment and runtime view of the MARVEL architecture for GRN3: Traffic Anomalous Events

Table 23: MARVEL components in the GRN3

GRN3: Traffic Conditions and Anomalous Events			
Component owner	**Subsystem /Component**	**Comments on how the component is used in GRN3 for R1**	**Deployment location**
Sensing and perception subsystem			Edge and fog
ITML	AV Registry	AV Registry contains metadata information of all AV sources present in GRN3. These include the information on the raw streams produced by the cameras (fps rate, resolution, etc.), but also on the anonymised streams produced by the VideoAnony anonymisation component.	GRN fog server (GRN F1)
GRN	Cameras with integrated microphones	Three cameras with integrated microphones will be used in GRN3 to produce audio-visual streams. Two cameras are used at the Zejtun and one at the Mgarr location.	Edge cameras
Security, privacy, and data protection subsystem			Edge, fog and cloud
FORTH	EdgeSec VPN	EdgeSec VPN creates a secure E2F2C VPN traffic backbone for all communications within	PC-simulated edge (GRN E1), GRN fog server (GRN F1) and

		the elements of the MARVEL platform by 100% encryption of the traffic.	cloud (PSNC HPC via OpenStack)
FBK	VideoAnony	VideoAnony will detect the cyclists and pedestrians that appear in the scene and anonymise their faces. In GRN3, the component is present with three instances, once for each of the three camera streams.	PC-simulated edge (GRN E1) and GRN fog server (GRN F1)
Data management and distribution subsystem			Edge, fog and cloud
ITML	Data Fusion Bus (DFB)	DFB stores inference results of the AI components that participate in GRN3.	Cloud (PSNC HPC via OpenStack)
INTRA	StreamHandler	StreamHandler receives continuously anonymised AV data streams and segments them for temporary storage, for later on-request visual inspection through SmartViz.	GRN fog server (GRN F1)
ATOS	DatAna Edge, Fog and Cloud	DatAna for GRN3 consists of DatAna Edge, DatAna Fog and DatAna Cloud components, each continuously consuming, through MQTT, the results of AI inference components deployed at the corresponding layer (Edge/Fog/Cloud) and sending to the relevant Kafka topics at the DFB.	PC-simulated edge (GRN E1), GRN fog server (GRN F1) and cloud (PSNC HPC)
CNR	HDD	For the predefined GRN3 AI inference components, HDD optimises the allocation of their output streams across a given set of DFB Kafka topics.	Cloud (PSNC HPC via OpenStack)
Audio, visual, and multimodal AI subsystem			Edge, fog and cloud
GRN	CATFlow+TAD	CATFlow classifies traffic entities (different vehicle classes and pedestrians) and provides their trajectories. TAD detects traffic anomalies, specifically, high and low traffic speeds, after processing CATFlow outputs.	PC-simulated edge (GRN E1) and GRN fog server (GRN F1)
AU	Audio-Visual Anomaly Detection (AVAD)	In GRN3, AVAD detects traffic anomalies (unseen data) from the AV streams.	Cloud (PSNC HPC via OpenStack)
TAU	Audio Tagging (AT)	In GRN3, AT outputs audio tags for the acoustic environment of the monitored locations.	Cloud (PSNC HPC via OpenStack)
Optimised E2F2C processing and deployment subsystem			Cloud
FORTH	MARVdash	MARVdash provides a Kubernetes-based deployment environment of all the GRN3 components. In GRN3, all nodes operate under Kubernetes/ MARVdash.	Cloud (PSNC HPC via OpenStack)
System outputs: User interactions and the decision-making toolkit			Cloud
ZELUS	SmartViz	SmartViz visualises detected anomalous road conditions which may be related (passively or actively) to obstructions.	Cloud (PSNC HPC via OpenStack)
STS	Data Corpus-as-a-Service	Data Corpus contains all data collected for GRN3 use cases, used for training of the relevant AI models.	Cloud (PSNC HPC via OpenStack)

AV data acquisition and anonymisation

The inference data flow starts with the three cameras, one at the Mgarr and two at the Zejtun locations. Each of the cameras produces continuous AV streams with both modalities present (audio, video). The stream from the Mgarr camera is transmitted to the GRN PC (GRN E1) and then subsequently consumed by the VideoAnony component. (It was noted that audio anonymisation is not relevant for GRN3 due to the very low likelihood of speech content presence). Upon anonymisation, specifically, by blurring faces and car plates of the received frames, VideoAnony produces an RTSP stream compiled from the anonymised frames. In parallel, i.e., concurrently in time, the two cameras in Zejtun produce similar AV streams, both of which are transmitted now to the GRN fog server (GRN F1), where two different instances of VideoAnony receive the respective streams and anonymise them, frame-by-frame. Each of the three in total VideoAnony instances produces an RTSP stream that is consumed by several components further up in the pipeline: 1) by the AI components, to produce inference results; 2) by StreamHandler, for AV data storage; and 3) by SmartViz, for real-time views of the AV streams. Each of these is further described below.

Real-time AI inference

The anonymised streams from the three locations are each processed by CATFlow+TAD, AVAD and AT, each instantiated three times.

- **CATFlow+TAD** – In GRN3, one instance of CATFlow+TAD is running at the GRN Edge (GRN E1), processing the anonymised stream from the Mgarr camera, while the two (anonymised) streams from the Zejtun cameras are processed by CATFlow+TAD instances running at the GRN fog server (GRN F1). The goal of this component is to produce trajectories of traffic entities in the anonymised AV streams, received from VideoAnony, and use these to detect traffic anomalies. Specifically, CATFlow first outputs trajectories of traffic entities and sends these results to the appropriate MQTT topics of the respective DatAna MQTT brokers. DatAna agent then relays the received results in the form of textual data to TAD for subsequent analysis. TAD processes the received results, outputting anomalies in the form of low and high traffic speeds. Results of TAD are submitted to the appropriate MQTT topics of the respective DatAna MQTT brokers, which, after the appropriate transformations detailed below, are applied are sent further up the chain.
- **AVAD** – The three instances of AVAD are executed on the cloud (PSNC HPC via OpenStack). This component is trained on regular traffic situations such that it can recognise any deviation from normal, i.e., events from previously unseen data. After the results are generated, they are transmitted to appropriate MQTT brokers of the Cloud DatAna agent (PSNC HPC via OpenStack).
- **AT** – Similarly as with AVAD, the three instances of AT are running on the cloud (PSNC HPC via OpenStack). The component is employed in GRN3 to enable tagging of audio traffic events on consecutive time intervals. In each component instance, after being generated, the results are transmitted to appropriate MQTT brokers of the Cloud DatAna agent (PSNC HPC via OpenStack).

More details on each of the AI components in GRN3 can be found in Section 4.2.2.

Inference results management and on-the-fly transformations

As indicated in the paragraph above, after the inference results have been generated, they are submitted to the respective MQTT broker of the DatAna agent residing at the same node. Specifically, CATFlow and TAD from GRN edge (GRN E1), which process the stream from the Mgarr camera, submit the results to the MQTT broker of DatAna edge (GRN E1), while the instances of CATFlow and TAD at GRN fog (GRN F1), which process the streams from the two Zejtun cameras, submit their results to the MQTT broker of DatAna fog (GRN F1). Similarly, each of the three instances of AVAD and AT, residing on the cloud (PSNC HPC via OpenStack), submit their results to the MQTT brokers of the DatAna cloud agent (PSNC HPC via OpenStack). Each of the DatAna agents, after receiving the inference results, applies appropriate transformation to standard data models, referred to in Section 3.2, and relays the results to the next DatAna agent in the infrastructure, in the case of DatAna edge or DatAna fog, or sends these to the appropriate Kafka topics of DFB (PSNC HPC via OpenStack). Specifically:

- DatAna edge (GRN E1) receives inference results of CATFlow+TAD at edge and sends the transformed results to DatAna fog;
- DatAna fog (GRN F1) receives the results from DatAna edge and forwards them to DatAna cloud;
- DatAna fog (GRN F1) concurrently receives inference results of CATFlow+TAD at fog and sends the transformed results to DatAna cloud;
- DatAna cloud (PSNC HPC via OpenStack) receives the results from DatAna fog and sends them to the Kafka topics of DFB;
- DatAna cloud (PSNC HPC via OpenStack) receives the results from AVAD and AT instances at the cloud, transforms them, and sends results to the appropriate Kafka topics of DFB.

Inference results fusion and storage

After being gathered in the appropriate Kafka topics in DFB (PSNC HPC via OpenStack), where one Kafka topic per each AI component is configured and enabled, results are consumed by SmartViz and Data Corpus, as detailed below. Finally, DFB passes also results from Kafka topics to Elasticsearch for persistent storage.

AV data storage

Concurrently with processing the RTSP (anonymised) streams for AI inference, another component, StreamHandler, is also subscribed to receive these streams. StreamHandler buffers each of the streams for further segmentation that occurs at regular time intervals, and finally stores the respective AV snippets in the MinIO database. The buffering intervals and segmentation parameters are configurable by the end-user. When a user wishes to inspect the AV data in an interval of interest (e.g., upon detection of an anomaly or raised alert by SmartViz), the user will submit a request to StreamHandler with onset and offset time intervals. StreamHandler then compiles the relevant AV data segments into a single file and sends back to SmartViz the compiled AV data section.

Visualizations and UI interface

The last component of the real-time inference pipeline, also serving as the UI of the platform, is SmartViz. This component runs on the cloud (PSNC HPC via OpenStack). Several functionalities are supported: 1) advanced visualisations, with a dashboard for user configurations and user interactions, 2) live AV feed for real-time inspection of the monitored areas, 3) on-request inspection of stored AV data, and 4) user-based verification of AI inference results. First, visualisations of AI inference results are provided in the form most suitable to the end user. For GRN3, this consists of a map indicating locations of detected anomalies and audio events and related time statistics; further details can be found in Section 4.4. Live AV feed from the three cameras (Mgarr and Zejtun), arriving from the RTSP (anonymised) stream from the three VideoAnony instances is shown upon user request. Also, for inspection of historical AV data or near real-time, e.g., in case an event has been detected by the system and the user was not able to follow the AV stream in real-time, the user may submit a request to the StreamHandler with onset and offset times indicating the time interval of interest. SmartViz receives the respective AV data segment, as described in the above, and displays it for user inspection. Finally, after inspecting an AV data segment, either through live feed or from StreamHandler, the user may verify the relevant inference result. This is done by submitting the verification result to the corresponding Kafka topic in DFB, upon which the relevant entries in the Elasticsearch database are changed.

4.1.2 Pilot E2F2C infrastructure

The GRN Infrastructure provided for the R1 integration consists of 3 IP cameras transmitting audio and video, a PC to simulate the edge layers and a workstation at the fog layer. The following subsections will describe each component in detail.

IP Cameras - Edge

The GRN pilot includes three IP cameras; one camera is at Mgarr (a rural town in the north-west region), and the other two cameras are at Zejtun (an urban town close to the southern inner-harbour region). These cameras have no ability to process data at the edge and can only transmit audio and video via an IP connection. The camera at Mgarr has slightly different specifications than the IP cameras at Zejtun. All three components will be used in both of the GRN use cases employed for R1, namely GRN3 and GRN4. Table 24 lists the specifications for each camera.

Table 24: Specifications for GRN IP Cameras

Specifications Type	Mgarr Camera	Zejtun Cameras
IP camera model	Safire 5MP Bullet Outdoor/Indoor IP Camera With PoE	ANNKE outdoor 5MP PoE security cameras, Model I51DL, Lens: 2.8mm
Resolution	1920 x 1080 P	1920 x 1080 P
Frame Rate	25 fps	20 fps
Video Encoding	H.265	H.265
Audio Encoding	MP2L2	MP2L2
Audio Sampling Rate	32kHz	16kHz
Audio Stream Bitrate	64kbps	128kbps

Edge PC

GRN has deployed a PC at the Mgarr location directly connected to the Mgarr IP Camera. Table 25 lists the specifications of the GRN Edge PC. This PC will be used to carry out processing at the edge.

Table 25: GRN Edge PC specifications

Component	Specifications
CPU	Intel Core i7-3770
GPU	GTX1650 4GB
Hard Drive	1TB
RAM	32GB

Fog Workstation

GRN has deployed a workstation with integrated GPU as part of the Fog layer. The AV streams from the Zejtun Cameras will be processed on the Fog Layer. Table 26 shows the specifications of the GRN Fog workstation.

Table 26: GRN Fog workstation specifications

Component	Specifications
CPU	Intel core i7 11700K
GPU	RTX3070 8GB
Hard Drive	2TB
RAM	32GB

4.1.3 Deployment

The infrastructure of GRN consists of a server located at the fog layer and a workstation located at the edge layer (Figure 10). Both these nodes need to become part of the Kubernetes cluster. Taking advantage of the EdgeSec VPN, the aforementioned nodes in two different layers are assigned a VPN IP. After that, nodes are able to directly announce themselves and discover other nodes in the virtual network they belong to.

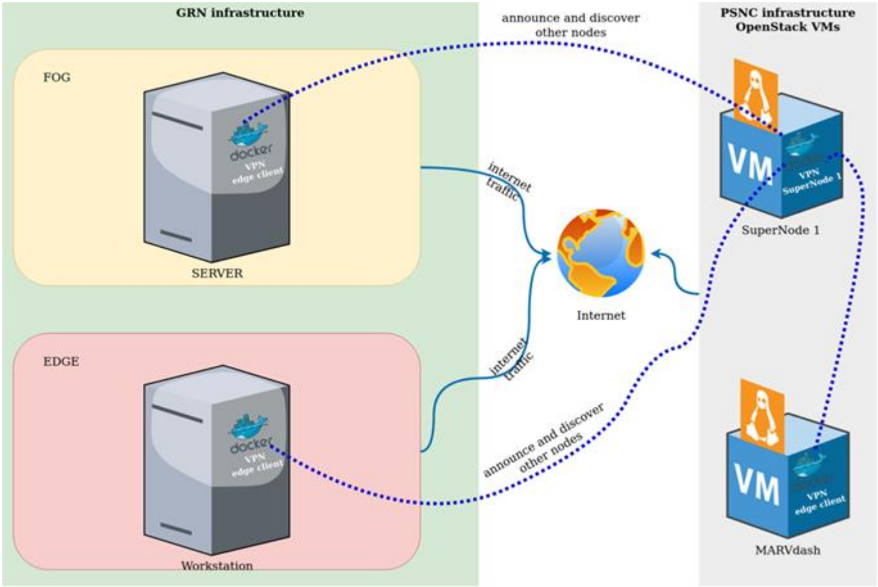


Figure 10: VPN implementation in the GRN use cases

4.1.4 Analysis of real-life data streams

Two types of IP cameras with integrated microphones are being used for both use cases GRN3 and GRN4. In one type, video is captured at a resolution of 1920x1080 at 25 frames per second, whilst in the other type, video is captured at a resolution of 1920x1080 at 20 frames per second. Both camera types encode the video using the H.264 standard and the RTSP protocol is used to live stream the video. Audio is captured by the built-in microphones and the specifications for the two types are different. In one camera the audio sampling rate is 32kHz and encoding is MP2L2 resulting in a bit-rate of 64kbps. The audio sampling rate in the second type is 16kHz and the output bit rate is 128kbps. The audio is also streamed over RTSP to the next node.

4.1.5 Integration and testing

The integration of GRN3 components is based on the reference architecture of the ‘AI Inference Pipeline’. Based on the infrastructure that was available at the pilot site, it was decided to distribute the payload of anonymising and processing the AV data streams from the three cameras with CATFlow and TAD into two infrastructure nodes, namely the GRN E1 and the GRN F1. GRN E1 at the edge hosted the services required for one of the AV data streams (Mgarr) due to more limited available resources, while GRN F1 hosted the services associated with the other two AV data streams (Zejtun). A special configuration was applied to all GRN VideoAnony instances in order to comply with data privacy regulations and security concerns expressed by GRN. An encryption mechanism was applied to the authentication information that was required by VideoAnony to access the raw AV data streams from the three CCTV cameras via RTSP. This mechanism was able to mask the necessary authentication credentials from other services deployed within the MARVEL Kubernetes cluster.

In terms of AI component distribution to the E2F2C layers, due to problems in the supply chain in 2021 and 2022, originally foreseen edge computation infrastructure devices could not be secured (e.g., Nvidia Jetson), which would have allowed the deployment of more demanding AI components with GPU-enabled processing capabilities at the edge. Therefore, such AI components (AVAD, AT) were deployed at the cloud layer.

Each component that needed to access an AV data stream produced by a VideoAnony instance had to make a REST API request to the AV Registry component at GRN F1 to receive the details (e.g., URL, AV metadata) of the AV source (VideoAnony instance) it needed to connect to.

The RTSP protocol was implemented for delivering the AV data streams from the VideoAnony instances to the respective AI components, StreamHandler, and SmartViz.

Each AI component published its output of raw inference results as messages structured in JSON format to an MQTT broker that resided on the same infrastructure node as the AI component.

DatAna agents subscribed to the topics of the MQTT brokers residing on the same layer to receive AI raw inference results. Each DatAna agent transformed the inference results into data models that are compliant with the Smart-Data-Models and relayed them to a DatAna agent at a higher layer (DatAna Edge to DatAna Fog, DatAna Fog to DatAna Cloud). Finally, the DatAna Cloud published the inference results collected from all layers to dedicated Kafka topics at the DFB, which was deployed at the cloud layer.

SmartViz was deployed at the cloud layer and could access the inference results that were aggregated at the DFB (both real-time results published at the Kafka topics and historical results stored in the DFB Elastic Search repository). SmartViz was responsible for visualising these results and for transferring information on results that were verified by the user back to the DFB for updating the respective entries in the Elastic Search.

SmartViz could also request AV data segments from StreamHandler (deployed at GRN F1) that corresponded to specific inference results selected by the user and receive the corresponding AV data segments to be displayed to the user.

The Data Corpus deployed at the cloud layer could access the inference results and verification information published in the Kafka topics in real-time from DatAna Cloud and SmartViz, respectively, by subscribing to these Kafka topics. In addition, the Data Corpus could access AV data that was produced by StreamHandler after processing the live AV data streams it was receiving.

HDD deployed at the cloud layer could exchange information with the DFB through a REST API to provide recommendations for optimising the DFB Kafka topic configuration.

MARVdash was used to orchestrate the deployment of all components at all infrastructure nodes.

During the partial integration testing and end-to-end integration testing activities, several issues were resolved, including the following:

- appropriate configuration of the infrastructure nodes;
- appropriate deployment configuration of the components, service names and exposed ports;
- exposing services so that they are reachable from other services;
- fine-tuning AV data streams;
- fixing inconsistencies in inference result data.

4.2 Multimodal and privacy-aware intelligence for GRN3

This section presents the work done to configure the framework according to the needs of the GRN3 in terms of consolidation and ingestion of audio-visual data, training and testing of AI models, design of the pipeline data flow from the edge to the cloud and analysis of the output obtained.

4.2.1 Datasets for model training and privacy assurance

This section describes the methods and approaches taken towards collecting the AV data, the datasets required for model training, the analysis of datasets and streams and privacy assurance and anonymisation methods used in the use case.

4.2.1.1. Datasets for model training

GRN has provided data for various components throughout the preparation for the M18 R1 integrated version. GRN has contributed to datasets for the training of SED, Audio Tagging, AVAD and VideoAnony for the GRN3 use case. GRN has also contributed with datasets for the training of AVCC for GRN4 and federated training within GRN4, MT1, and UNS1.

The dataset for SED training is the GRN-AV-traffic-entity dataset described in D1.2. For this dataset audio-visual snippets were obtained from various locations around Malta and from the GRN static cameras.

The audio track was manually annotated using ELAN. The ontology used in the annotation is obtained by extending the MAVD (Montevideo Audio and Video Dataset) ontology, which is defined by Entity and Component taxonomies and Actions that link an entity to a component. The ontology sets are the following:

- Entities = {Car, Motorcycle, Light Goods vehicle, Heavy goods vehicle, Bicycle, Pedestrian, Micro-mobility, Bus, Pedelec, Other}
- Component = {Brakes, Engine, Wheel, Door, Horn, Alarm, Compressor, Bell, Voice, Footsteps, Motor, Music}
- Actions = {Rolling, Whining, Screeching, Playing, Sounding, Shouting, Chatter, Closing, Opening, Idling, Accelerating}

Other fields were added according to the needs observed, such as construction noise, wind noise, and ambulance sirens.

All the AV clips recorded were made up of short clips 3-5 minutes in length. Approximately 180 snippets were annotated for this dataset, 44 from different locations around Malta and the rest from the GRN static cameras. This amounts to more than 13 hours of annotated data.

Although this amount of data exceeds the 4 hours planned in D1.2, more data will be required due to the complex requirements for training the SED model.

The GRN-AV-traffic-state dataset is used for training the Audio Tagging model. For this dataset, AV clips from various locations and the GRN static cameras were cropped into 5–second clips and subsequently annotated. During the first iteration of this data collection process, the Ontology used in annotation consisted of four classes and a number of labels per class are defined:

- Speed {Standstill, Low, Moderate, High}
- Collision {Light, Heavy}
- Traffic {Sparse, Moderate, Heavy, Jam}
- Stationary Obstacle {Service Vehicle, Breakdown, Accident}.

For each clip, an annotator was expected to create a metadata file and write down each label. This method proved to be more time-consuming and complicated than necessary. Thus, the more recent versions of the datasets were simplified. First, the ontology was reduced to two classes instead of four. These are:

- Speed {Standstill, Low, Moderate, High}
- Traffic {Sparse, Moderate, Heavy, Jam}.

The other two classes, Collisions and Stationary obstacles, were absorbed by the dataset for AVAD, which will be explained below. Next, the annotators were only asked to change the file name from the AV 5-second long snippet to show the labels. For example, a snippet would be labelled as *SpeedStandstill_TrafficJam_1.mp4*.

This new method allowed the annotators to quickly go through large amounts of data. In fact, 134 5-seconds snippets were annotated with the old method, whilst 4500 5-seconds snippets were annotated with the new method. Thus, more than 6 hours of data were annotated and the Audio Tagging model shows promising results. It is expected that more data will be required.

In addition, data for the AVAD was required. This was not planned in D1.2, but it was required for the R1 integration and other future implementations. This dataset consisted only of videos obtained from the GRN static cameras. The ontology used in the annotation consists of the following labels:

- normal
- pedestrians_crossing
- buses
- obstructions
- heavy_weight_vehicles
- bicycles
- exit_streets
- police car
- ambulance
- U-turns.

However, annotators were instructed to label also any other exceptionally anomalous event they might observe. During the annotation process, the given video is opened with a simple video editing software such as Windows photos. The annotation task entails trimming a video into time sections according to the anomalies, or lack of, as observed in that section of the video. Then that video is labelled according to which anomaly is observed in the trimmed section. Thus, an annotated video can be in the range of 1 second to 5 minutes. If there is more than one anomaly occurring concurrently, the annotator will produce trimmed sections from that period of the video, each labelled with the corresponding anomaly and trimmed to the exact duration of the anomaly it is labelled as. More than 660 video snippets were labelled from videos obtained from the GRN static cameras. More data may be needed in the future.

The training of VideoAnony required the need for an annotated dataset with bounding boxes indicating the vehicles' number plates. The data was obtained from various locations around Malta (D2.1, section 4.2.2), in addition to videos obtained from the GRN Static camera locations. These types of annotations are very time-consuming since a short video involves a large number of frames. The CVAT software was used to streamline the process as much as possible. This software partially automates the annotation process by guessing the place of the bounding box in the next frame. All annotated videos were trimmed to 10-30 seconds snippets such that a variety of videos can be annotated. 41 snippets were annotated from the dynamic locations, whilst 32 snippets were annotated from the static locations. In total, more than 30 minutes of video have been annotated.

4.2.1.2. Analysis of datasets

The analysis of the streams and datasets is performed by the AI model providers. With their feedback, more data for training will be added.

4.2.1.3. Privacy assurance and anonymisation

To ensure privacy, all video streamers are anonymised using VideoAnony. The current version of this component blurs number plates and faces thus the streams contain no identifiable information.

4.2.2 Training and testing of the models

In addition to the CATFlow model, the GRN3 use case employs two AI models, namely Audio-Visual Anomaly Detection (AVAD) and Audio Tagging (AT). Detailed description of the methodologies proposed by MARVEL partners defining the underlying models of these AI functionalities can be found in D3.1. In the following, a description of these models and their training with data from GRN3 use case are provided.

4.2.2.1. Multimodal AI realisation

- **CATFlow+TAD** – In the GRN3, CATFlow is used to extract an estimate of the vehicle's speed. This information is then passed onto the TAD component such that vehicles driving at anomalous low or high speed could be identified.
- **Audio-Visual Anomaly Detection (AVAD)** – The GRN3 use case employs an Audio-Visual Anomaly Detection (AVAD) model to detect previously unseen, novel events in the audio-visual live feed. Audio-Visual Anomaly Detection is the task of identifying novel situations in a scene, based on the audio-visual information captured in an input video or image. Models are trained to learn and replicate situations considered normal in a certain setting, e.g., pedestrians on a pathway. Situations that do not occur often, for example, a car on the pathway, are not present in the data used to train the model to identify as normal and, thus, they will be detected as anomalies. Information related to the underlying machine learning model developed by MARVEL partners for detecting anomalies based on visual information can be found in deliverable D3.1. An anomaly detection model based on audio-visual information will be developed in the second part of the project, and it will be trained and tested on MARVEL data from GRN3.
- **Audio Tagging (AT)** – The GRN3 use case also utilises Audio Tagging (AT) component to label the audio segments with tags related to traffic level and speed. Tagging is done in 5-second consecutive non-overlapping segments, and the audio segments are extracted from a live audio-visual feed. In the GRN3 use case, the audio segments are tagged with two tags, one related to traffic amount (sparse, moderate, heavy, and jam) and one related to the speed of the traffic (standstill, low, moderate, and high). The component is trained and tested with manually annotated material from three locations used in the use case. The component uses multitask approach: the neural network architecture used in the sound event detection component is modified to have two classifying outputs, one for traffic tag and one for speed tag.

4.3 Analysis of the outputs from real-life smart city experiments in GRN3

Table 27 summarises the characteristics or parameters against which the use case will be evaluated. Most of these characteristics are directly dependent on the non-functional and functional use case and asset KPIs as listed in D1.2, tables 6.2-6.3 and 6.5-6.11. Evaluation will be performed later and reported in D6.2.

Table 27: Parameters to determine for use case GRN3

Use case	Parameter	How to measure	Target to be achieved
GRN3: Traffic Conditions and Anomalous Events	Efficiency *Related to the efficiency of the system as used in the use case*	Efficiency is largely dependent on how long it takes to detect and flag an anomaly. The average time in minutes from the start of the anomaly to detection time is measured.	2 minutes from start of the anomalous event.
	Operability *Related to the ability of the components to keep functioning together*	Record downtime for any of the components (assets) along the system pipeline as a percentage of total time.	Downtime is minimised to 10% or less.
	Usability *Related the how well the system helps the users to achieve a task in a given use case*	Interview GRN Managers to measure the end-user experience.	End-users finds the system easy to use
	Robustness *Related to how robust are the system components during the period of operation*	Sustains performance in various weather conditions (Power source, sensor, and CPU board operation).	Performance sustained in most weather conditions
	Performance *Related to how well the system performs the intended task*	Correctly detect various anomalous events on the road.	70% detection of anomalous events
	Accountability *Related to the system being able to explain results or decisions.*	System stores video snippet of anomaly.	Number of times system fails to store video snippet, even though anomaly is flagged.
	Transparency *Related to the description of the processes or algorithms that are used to generate system output*	Decision processes are described in a document.	Document availability
	Privacy awareness *Related to the provision of adequate governance mechanisms that ensure privacy in the use of data.*	Inspect anonymisation in a sample of stored videos detected as anomalies.	Anonymisation of all frames within AV snippet.

4.4 Demonstration

4.4.1 The Decision-making Toolkit

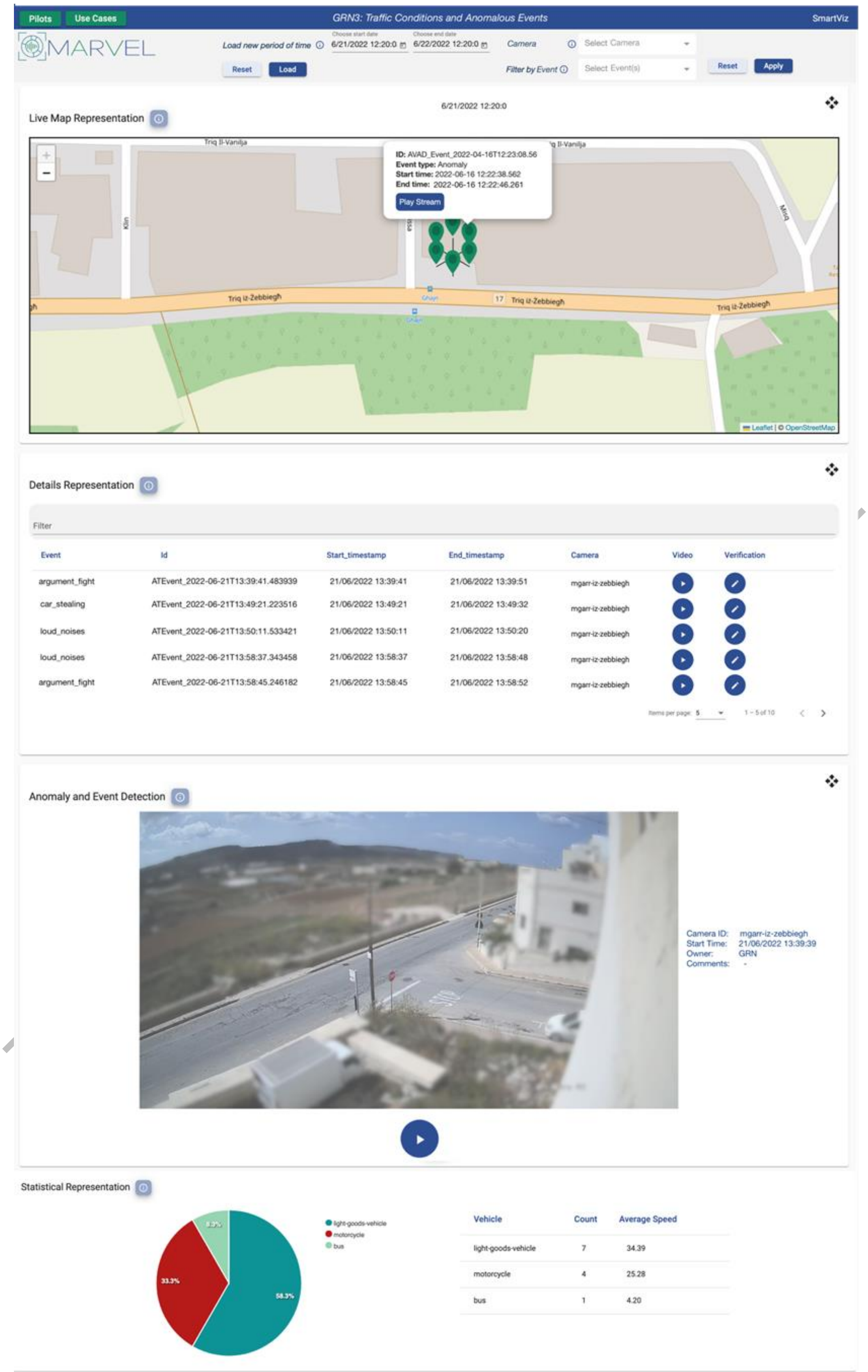
In this use case, traffic conditions are monitored to detect anomalous events, for example, traffic jams, accidents, cars stuck and obstructing a junction, very slow vehicles and service vehicles parked on the side or obstructing a carriageway. The latter event is frequent in Malta’s narrow one-way urban streets, often causing ripple effects that extend beyond the immediate area. In general, this output would find application in systems intended to inform drivers near the detected anomaly or to infer possible issues in adjacent areas, thus informing drivers of obstacles ahead. In addition, the detection of anomalous events can be used to alert personnel stationed at traffic management control rooms, who can then interpret the data and take the necessary action. The latter application is particularly attractive since it promises to reduce the detection time of anomalous events.

The users of this system are intended to be traffic managers who can give directives to authorities to react to a traffic incident. The User Stories for this use case are shown in Table 28 and these give a better understanding of the functionality expected by the intended user.

Table 28: The user stories for the GRN3 – Traffic Conditions and Anomalous Events

Title	Main user stories	Sub user stories and user intent
Anomaly detection	**As a** Traffic Manager, **I want** to be alerted if an anomaly occurs on the road **so that** I can analyse it and determine a course of action.	**As a traffic manager:** • once informed of anomalous events on the roads, I can take appropriate actions i.e.: dispatching of the relevant civil protection authorities. • once an anomaly is flagged, I can see a live feed from the camera where the event occurred, as well as the feed a few minutes before the anomaly happened, to accurately assess the cause. • once I review the camera feed of a flagged event, I can have the option to mark this event as anomalous or not and continuously improve the system and in this case: i. allowing for the augmentation of additional annotated data which is needed for training or testing of AI models, and ii. measuring the accuracy of the models and in general the evaluation of the system. • if anomalies happened in the past or in my absence, I can still view them and access historical data around them, therefore such data are stored and managed. • since I am observing multiple locations, I can see observed anomalies and related data on a map.
Traffic Conditions	**As a** Traffic Manager **I want** to monitor the traffic conditions easily and efficiently on various roads and junctions **so that** I can monitor events such as traffic jams, congestion levels and traffic flow rate.	By monitoring the traffic conditions of various roads simultaneously, I can observe traffic jams, traffic bottlenecks, traffic flow rates and obstructions. Here, as the traffic manager, I can be alerted whenever the traffic conditions or states change substantially. With the knowledge that a road is obstructed, I can alert the appropriate authorities. With long-term data, infrastructural changes that might be needed can be flagged for further processing by the traffic and infrastructure engineers. If more than one location is being observed, this data can be represented on a map.

The dashboard for the GRN3 – Traffic Conditions and Anomalous Event use case links to two implemented user stories: Anomaly detection and Traffic conditions. Both user stories are offered to a Traffic Manager as a user in a single dashboard (Figure 11) to combine information and explore the available data and analysis. The goal of the Anomaly detection user story is to alert the users if an anomaly occurs so they can analyse it and determine a course of action.



GRN3: Traffic Conditions and Anomalous Events

Load new period of time: Choose start date 6/21/2022 12:20:0 Choose end date 6/22/2022 12:20:0

Camera: Select Camera

Filter by Event: Select Event(s)

Live Map Representation

6/21/2022 12:20:0

ID: AVAD_Event_2022-04-16T12:23:08.56
Event type: Anomaly
Start time: 2022-06-16 12:22:38.562
End time: 2022-06-16 12:22:46.261

Details Representation

Filter

Event	Id	Start_timestamp	End_timestamp	Camera	Video	Verification
argument_fight	ATEvent_2022-06-21T13:39:41.483939	21/06/2022 13:39:41	21/06/2022 13:39:51	mgarr-iz-zebbiegh	▶	✎
car_stealing	ATEvent_2022-06-21T13:49:21.223516	21/06/2022 13:49:21	21/06/2022 13:49:32	mgarr-iz-zebbiegh	▶	✎
loud_noises	ATEvent_2022-06-21T13:50:11.533421	21/06/2022 13:50:11	21/06/2022 13:50:20	mgarr-iz-zebbiegh	▶	✎
loud_noises	ATEvent_2022-06-21T13:58:37.343458	21/06/2022 13:58:37	21/06/2022 13:58:48	mgarr-iz-zebbiegh	▶	✎
argument_fight	ATEvent_2022-06-21T13:58:45.246182	21/06/2022 13:58:45	21/06/2022 13:58:52	mgarr-iz-zebbiegh	▶	✎

Items per page: 5 1 – 5 of 10

Anomaly and Event Detection

Camera ID: mgarr-iz-zebbiegh
Start Time: 21/06/2022 13:39:39
Owner: GRN
Comments: -

Statistical Representation

Vehicle	Count	Average Speed
light-goods-vehicle	7	34.39
motorcycle	4	25.28
bus	1	4.20

Figure 11: DMT GRN3 Dashboard

These detected anomalies are visualised in the Real-time Map Representation widget in order to facilitate a quick response from the users' perspective. The Video Player widget can either be used to play a live feed from the selected camera or a static video that corresponds to a detected anomaly upon user request. Finally, the inference results and textual information linked to the detected anomalies are represented by the Details widget.

5 GRN4: Junction Traffic Trajectory Collection

5.1 Integration and framework configuration for GRN4

This section describes the steps taken to integrate the systems such that the GRN4: Junction Traffic Trajectory Collection could be set up and tested. The following sections describe the architecture and components, the pilot infrastructure and the integration, deployment and testing phase.

5.1.1 Architecture, components and E2F2C streaming data pipeline

The GRN4 realisation of MARVEL architecture is given in Figure 12, while the employed MARVEL components together with their configurations for GRN4 are given in Table 29.

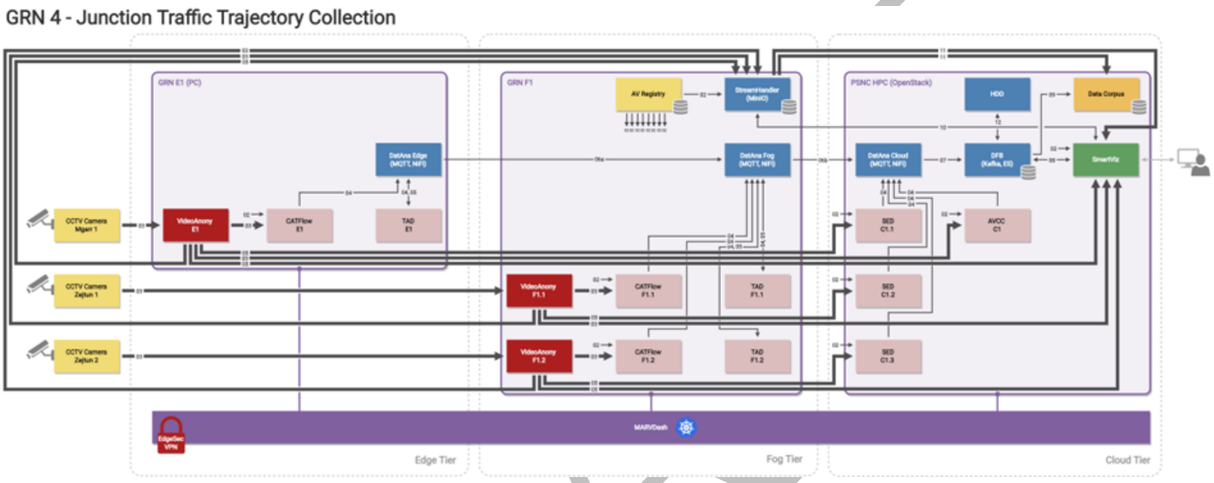


Figure 12: MARVEL R1 deployment and runtime view of the MARVEL architecture for GRN4: Junction Traffic Trajectory Collection

Table 29: MARVEL components in the GRN4

GRN4: Junction Traffic Trajectory Collection			
Component owner	**Subsystem /Component**	**Comments on how the component is used in GRN4 for R1**	**Deployment location**
Sensing and perception subsystem			Edge and fog
ITML	AV Registry	AV Registry contains metadata information of all AV sources present in GRN4. These include the information on the raw streams produced by the cameras (fps rate, resolution, etc.), but also on the anonymised streams produced by the VideoAnony anonymisation component.	GRN fog server (GRN F1)
GRN	Cameras with integrated microphones	Three cameras with integrated microphones will be used in GRN4 to produce audio-visual streams. Two cameras are used at the Zejtun and one at the Mgarr location.	Edge cameras
Security, privacy, and data protection subsystem			Edge, fog and cloud
FORTH	EdgeSec VPN	EdgeSec VPN creates a secure E2F2C VPN traffic backbone for all communications within	PC-simulated edge (GRN E1), GRN fog server (GRN F1) and

		the elements of the MARVEL platform by 100% encryption of the traffic.	cloud (PSNC HPC via OpenStack)
FBK	VideoAnony	VideoAnony will detect the cyclists and pedestrians that appear in the scene and anonymise their faces. In GRN4, the component is present with three instances, once for each of the three camera streams.	PC-simulated edge (GRN E1) and GRN fog server (GRN F1)
Data management and distribution subsystem			Edge, fog and cloud
ITML	Data Fusion Bus (DFB)	DFB stores inference results of the AI components that participate in GRN4.	Cloud (PSNC HPC via OpenStack)
INTRA	StreamHandler	StreamHandler receives continuously anonymised AV data streams and segments them for temporary storage, for later on-request visual inspection through SmartViz.	GRN fog server (GRN F1)
ATOS	DatAna Edge, Fog and Cloud	DatAna for GRN4 consists of DatAna Edge, DatAna Fog and DatAna Cloud components, each continuously consuming, through MQTT, the results of AI inference components deployed at the corresponding layer (Edge/Fog/Cloud) and sending to the relevant Kafka topics at the DFB.	PC-simulated edge (GRN E1), GRN fog server (GRN F1) and cloud (PSNC HPC)
CNR	HDD	For the predefined GRN4 AI inference components, HDD optimises the allocation of their output streams across a given set of DFB Kafka topics.	Cloud (PSNC HPC via OpenStack)
Audio, visual, and multimodal AI subsystem			Edge, fog and cloud
GRN	CATFlow+TAD	In GRN4, CATFlow classifies traffic entities (both vehicles and pedestrians) and provides their trajectories. TAD detects traffic anomalies, specifically, high and low traffic speeds, by processing CATFlow outputs.	PC-simulated edge (GRN E1) and GRN fog server (GRN F1)
AU	Audio-Visual Crowd counting (AVCC)	In GRN4, AVCC is used to provide the number of detected people at the Mgarr location and the respective heatmaps in the camera FoV.	Cloud (PSNC HPC via OpenStack)
TAU	Sound event detection (SED)	In GRN4, SED is used to detect different vehicle types from the audio stream.	Cloud (PSNC HPC via OpenStack)
Optimised E2F2C processing and deployment subsystem			Cloud
FORTH	MARVdash	MARVdash provides a Kubernetes-based deployment environment of all the GRN3 components.	Cloud (PSNC HPC via OpenStack)
UNS	FedL	For GRN4, FedL delivers federated training of the VCC model together with MT1 and UNS1 pilots without explicit exchange of the (raw) data. GRN4 implements a VCC-FedL client that communicates through the FedL server, running at the cloud, with similar MT1 and UNS1 FedL clients.	GRN fog server (GRN F1) and Cloud (PSNC HPC via OpenStack)
System outputs: User interactions and the decision-making toolkit			Cloud
ZELUS	SmartViz	SmartViz shed light on both the behaviour of road users and on gathering traffic statistics at road network junctions across time.	Cloud (PSNC HPC via OpenStack)

STS	Data Corpus-as-a-Service	Data Corpus contains all data collected for GRN4 use cases, used for training of the relevant AI models.	Cloud (PSNC HPC via OpenStack)

The streaming pipeline is instantiated on the same E2F2C infrastructure as with GRN3 and is very similar to the one of the GRN3, with the only difference in the AI components employed and the respective visualisations.

Real-time AI inference

With GRN4, all the three CATFlow+TAD instances are present, and in the same deployment locations: edge (GRN E1), fog (GRN F1), and cloud (PSNC HPC via OpenStack). The distinction with respect to GRN3 is that in GRN4 CATFlow classifies and tracks both vehicles and pedestrians. In addition, for GRN4 CATFlow is internally configured such that it outputs the trajectories of each vehicle as well as the entry and exit point with the camera field of view. For further analysis of pedestrians' presence, AVCC is employed to count the number of people and provide heatmaps indicating densities of pedestrians in the camera FoV. One instance of AVCC is deployed, specifically for the Mgarr camera stream where the relevance of analysing pedestrians' presence is higher. This instance of AVCC is deployed at the cloud (PSNC HPC via OpenStack).

In addition to CATFLow+TAD and AVCC, GRN4 employs also SED for detecting traffic sound events – specifically, different vehicle classes such as cars, buses, motorcycles. In GRN4, three instances of SED are deployed at the cloud (PSNC HPC via OpenStack), processing the Mgarr and the two Zejtun streams. More details on each of the AI components in GRN3 can be found in Section 5.2.2.

Visualisations and UI interface

For GRN4, visualisations consist of heatmaps and people counts for the Mgarr camera stream, and statistics visualisations of the presence of traffic entities over time (vehicles, pedestrians), at each of the three monitored locations (one at Mgarr, and two at Zejtun). More details on GRN4 visualisations can be found in Section 5.4.

5.1.2 Pilot E2F2C infrastructure

The pilot infrastructure is the same as that used in the GRN3 use case and the reader is referred to Section 4.1.2.

5.1.3 Deployment

Since GRN3 and GRN4 share the same infrastructure, the issues faced during component deployment were the same. They are already described in subsection 4.1.3.

5.1.4 Analysis of real-life data streams

The same cameras used in GRN3 are used in GRN4. Details about the cameras can be found in Section 4.1.4.

5.1.5 Integration and testing

The integration of GRN4 components is based on the reference architecture of the 'AI Inference Pipeline' and partially shares a common scheme with GRN3. More specifically, the AV data streams that were used as input were from the same three cameras used in GRN3. In addition, the anonymisation and processing of the AV data by CATFlow and TAD also shared the same configuration with GRN3, as was the distribution of AI components to E2F2C layers. The main difference between GRN4 and GRN3 was the use of different AI components at the cloud layer, namely SED and AVCC. Furthermore, while different instances of SED were used to process each of the three available AV data streams, AVCC was only applied to the stream from the camera at Mgarr, being anonymised at the GRN E1 device. The interfaces that were applied and the types of issues that were resolved as a result of integration testing were also the same as with GRN3 (Section 4.1.5).

5.2 Multimodal and privacy-aware intelligence for GRN4

This section presents the work done to configure the framework according to the needs of the GRN4 in terms of consolidation and ingestion of audio-visual data, training and testing of AI models, design of the pipeline data flow from the edge to the cloud and analysis of the output obtained.

5.2.1 Datasets for model training and privacy assurance

This section describes the methods and approaches taken toward collecting the AV data collection. The datasets required for model training, the analysis of datasets and streams and privacy assurance and anonymisation methods used in the use case.

5.2.1.1. Datasets for model training

GRN has provided data for various components throughout the preparation for the M18 R1 integrated version (find the names in D8.2 of the dataset). GRN has contributed to datasets for the training of SED, Audio Tagging, AVAD, VideoAnony and AVCC. All the datasets except the AVCC dataset will be used in both the GRN3 and the GRN4. GRN's contribution to the datasets used for training SED, Audio Tagging, AVAD and VideoAnony have been explained for use case GRN3 and the reader is referred to section 4.2.1.1. The remaining contributions to the AVCC dataset are explained in this section.

GRN contributed 71 snippets which contain pedestrians to the creation of the AVCC data.

5.2.1.2. Analysis of datasets

The analysis of the streams and datasets is performed by the AI model providers. With their feedback, more data for training will be added.

5.2.1.3. Privacy assurance and anonymisation

The same anonymisation is done as the one in GRN3. Details can be found in Section 4.2.1.3.

5.2.2 Training and testing of the models

The GRN4 use case employs four AI models, namely CATFlow+Text Anomaly Detection (TAD), Visual Crowd Counting (VCC) implemented on a Federated Learning framework, Audio-Visual Crowd Counting (AVCC), and Sound Event Detection (SED). Detailed description of the methodologies proposed by MARVEL partners defining the underlying models of these AI functionalities can be found in D3.1, while details on the Federated Learning framework used for implementing some of the AI functionalities to achieve distributed privacy-preserving training of Deep Learning models involved in the AI functionalities will be included in D3.4. In the following, a description of these models and their training in data of the use case are provided.

5.2.2.1. Multimodal AI realisation

- **CATFlow+TAD** – In the GRN4 use case, CATFlow is used to detect and track the different types of vehicles and pedestrians. In addition, CATFlow outputs the trajectories of each vehicle as well as the entry and exit points within the camera field of view. All this data is then used in transport studies, e.g., to observe how the junction or road segment that is being observed is used by pedestrians and vehicles. The speed information is also extracted and is then passed onto the TAD component such that vehicles driving at anomalous low or high speed could be identified. Anomalously slow vehicles could indicate that a vehicle is causing obstructions on the road, or that a road is unsuitable to drive through, thus a vehicle would have to slow down. An anonymously high speed could indicate reduced safety to pedestrians and thus more adequate infrastructure or enforcement is required.
- **Audio-Visual Crowd Counting (AVCC)** – The GRN4 use case employs an Audio-Visual Crowd Counting (AVCC) model to estimate the number of people present in the audio-visual live feed. Crowd counting is the problem of identifying the total number of people present in a scene who are observed in a given image of that scene. The input to a crowd counting model is a colour image, and the expected output is a number representing the total number of people present in that image. When additional audio signals from the scene are available, the enriched audio-visual information can also be used as input. The additional audio information encodes information related to the ambient sound in the scene, and it can be used to improve performance when the provided image is of low quality. Optionally, the output of a crowd counting model can be a density map which specifies the density of the crowd for each pixel of the input image, and the total count can be calculated by summing up all the density values for all pixels. Information related to the underlying machine learning models developed by MARVEL partners for visual and audio-visual crowd counting can be found in deliverable D3.1. For the GRN4 use case, the audio/visual crowd counting functionality has been tested on a subset of the MARVEL data. The corresponding models receiving as input visual and audio-visual data have not been trained using this data yet.
- **Sound Event Detection (SED)** – The GRN4 use case also utilises Sound Event Detection (SED) component to detect vehicle types (bus, car, truck, and motorcycle) in the audio-visual feed based on audio modality. Sound event detection is the task of recognising what sound class is active and when it is active within the analysed audio signal. The input to the SED component is a continuous audio signal, and the output contains sound event labels along with the start and stop timestamps of the sound events. SED component implemented for GRN4 is providing start and stop timestamps with a 1-second time resolution. The component is trained with manually annotated material where start and stop timestamps of the sound events are indicated. The training material was collected with the same setup in the same locations as in the use case and with a mobile setup in additional locations to diversify the material. The component was tested with data collected from the same locations as is used in the use case. Information related to the underlying machine learning models developed by MARVEL partners for sound event detection can be found in deliverable D3.1.

5.2.2.2. Federated learning realisation

GRN4 participates in MARVEL's federated learning R1 realisation with the FedL component, as described in detail in Section 3.3.3. At GRN4, the corresponding VCC-FedL client runs on the GPU of the GRN fog server (GRN F1). The client also communicates with the VCC-FedL server that runs at the MARVEL cloud (PSNC HPC via OpenStack).

5.3 Analysis of the outputs from all real-life smart city experiments

Table 30 summarises the characteristics or parameters against which the use case will be evaluated. Most of these characteristics are directly dependent on the non-functional and functional use case and asset KPIs as listed in D1.2, tables 6.2-6.3 and 6.5-6.11. Evaluation will be performed later and reported in D6.2.

Table 30: Parameters to determine for use case GRN4

Use case	Parameter	How to measure	target to be achieved
GRN4: Traffic Conditions and Anomalous Events	Efficiency *Related to the efficiency of the system as used in the use case*	Increased efficiency in the planning of roads and network infrastructure. The feedback from external traffic experts will be collected through a survey to determine the increase in efficiency	Potential decrease in time for planning through the availability of data
	Operability *Related to the ability of the components to keep functioning together*	Record downtime for any of the components along the system pipeline as a percentage of total time	Downtime is minimised to 10% or less
	Usability *Related the how well the system helps the users to achieve a task in a given use case*	Interview GRN Managers to measure the end-user experience	End-user finds the system easy to use
	Robustness *Related to how robust are the system components during the period of operation*	Sustains performance in various weather conditions, (Power source, sensor, and CPU board operation)	Performance sustained in most weather conditions
	Performance *Related to how well the system performs the intended task*	Successfully detect the various events of interest on the road, including trajectories. Availability of the required historical video samples	90% detection of events. 50% detection of trajectories
	Accountability *Related to the system being able to explain results or decisions.*	N/A	N/A
	Transparency *Related to the description of the processes or algorithms that are used to generate system output*	Decision processes are described in a document	Document availability
	Privacy awareness *Related to the provision of adequate governance mechanisms that ensure privacy in the use of data.*	Manually evaluate a sample of anonymised AV data to determine privacy protection. Draw a list of secure data characteristics and evaluate each.	Minimise anonymisation misses

5.4 Demonstration

5.4.1 The Decision-making Toolkit

Junction Traffic Trajectory collection is focused on the requirement of long-term data analytics that shed light on both the behaviour of road users (e.g., car drivers, motorcyclists, cyclists, pedestrians, etc.) and on gathering traffic statistics at road network junctions. This use case is of interest for long-term transport planning and evaluation. In particular, there is currently significant interest in studying active travel modes, such as cycling, walking, and micro-mobility more generally. Authorities in Malta are interested in, for example, finding the optimal position of pedestrian crossings, whether provisions for cyclists at complex junctions are adequate, and whether installed provisions are being used as intended.

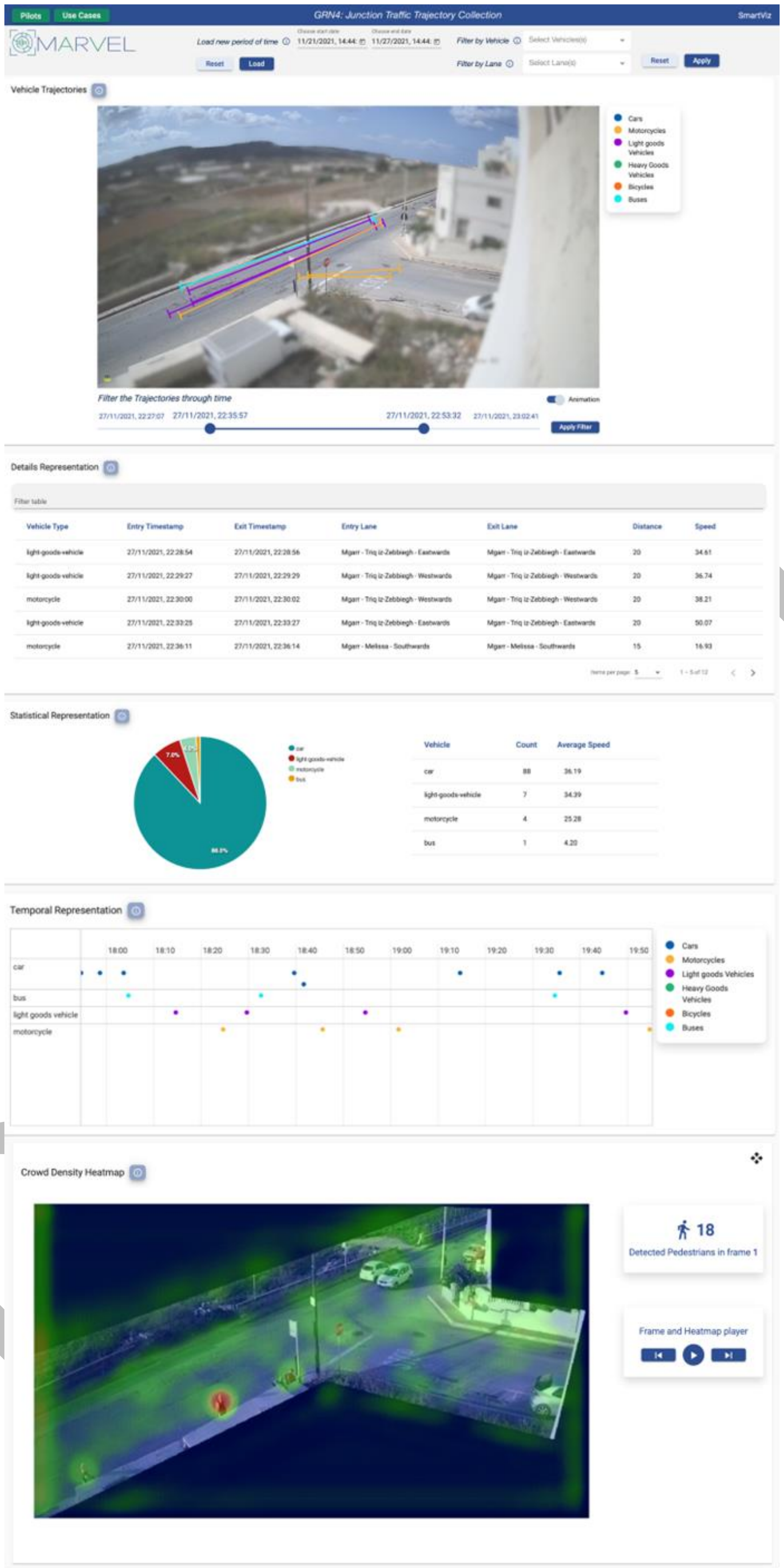
The users for this system are intended to be traffic engineers who need data to make informed decisions about infrastructure changes and upkeep, as well as transport researchers who are interested in user behaviours.

The User Stories for this use case are shown in Table 31 and give a better understanding of the functionality expected.

Table 31: The user stories for the GRN4 – Junction Traffic Trajectory Collection

Title	Main user stories	Sub user stories and user intent
Vehicle and Pedestrian Trajectories	**As a** Traffic Engineer, **I want to** view the trajectories of vehicles across a junction, **so that** I can analyse the most preferred paths vehicles take and make decisions on infrastructure for example, whether large vehicles should be allowed in a particular street.	**As a traffic engineer:** • once I view the most common trajectories taken by each type of vehicle or entity, I am able to make decisions on any infrastructure upgrade needed at each location. For example, I can note that pedestrians cross the street frequently from one spot, thus I can plan a crossroad at a certain point. In another case I can note that cyclists take a certain path, thus plan a cycling lane accordingly. The trajectories can also give insights into frequent recurring obstructions (i.e., cars parked at an inappropriate spot), which force temporal changes in the trajectories.
Vehicle and Pedestrian Counting	**As a** Traffic Engineer, **I want to** view the number of different types of vehicles that use certain types of roads, **so that** I am able to track the frequency of road usage per vehicle.	**As a traffic engineer:** • I want to easily view the traffic counts of all types of vehicles per vehicle, per time period and per road or junction, so that I can track the usage of a road per vehicle and estimate when and how the infrastructure needs to be upgraded. For example, stronger materials might be needed if the roads are used frequently by heavy vehicles or maybe wider pavements are needed if the road is a frequent walking route.

Figure 13 is the dashboard that combines the two user stories and contains all the widgets and visualisation schemas described below.



Load new period of time: 11/21/2021, 14:44 – 11/27/2021, 14:44

Filter by Vehicle: Select Vehicle(s)

Filter by Lane: Select Lane(s)

Filter the Trajectories through time

27/11/2021, 22:27:07 27/11/2021, 22:35:57 27/11/2021, 22:53:32 27/11/2021, 23:02:41

Vehicle Type	Entry Timestamp	Exit Timestamp	Entry Lane	Exit Lane	Distance	Speed
light-goods-vehicle	27/11/2021, 22:28:54	27/11/2021, 22:28:56	Mgarr - Triq iz-Zebbiegh - Eastwards	Mgarr - Triq iz-Zebbiegh - Eastwards	20	34.61
light-goods-vehicle	27/11/2021, 22:29:27	27/11/2021, 22:29:29	Mgarr - Triq iz-Zebbiegh - Westwards	Mgarr - Triq iz-Zebbiegh - Westwards	20	36.74
motorcycle	27/11/2021, 22:30:00	27/11/2021, 22:30:02	Mgarr - Triq iz-Zebbiegh - Westwards	Mgarr - Triq iz-Zebbiegh - Westwards	20	38.21
light-goods-vehicle	27/11/2021, 22:33:25	27/11/2021, 22:33:27	Mgarr - Triq iz-Zebbiegh - Eastwards	Mgarr - Triq iz-Zebbiegh - Eastwards	20	50.07
motorcycle	27/11/2021, 22:36:11	27/11/2021, 22:36:14	Mgarr - Melissa - Southwards	Mgarr - Melissa - Southwards	15	16.93

Items per page: 5 1 – 5 of 12

Vehicle	Count	Average Speed
car	88	36.19
light-goods-vehicle	7	34.39
motorcycle	4	25.28
bus	1	4.20

18 Detected Pedestrians in frame 1

Figure 13: DMT GRN4 Dashboard

This use case will allow the traffic engineers to make informed decisions on issues such as the most preferred location of a path or if there needs to be more enforcement on issues such as heavyweight vehicles staying on the left side of the road. This use case was also selected for the MVP release in M12.

The data output of CATFlow is used to illustrate the trajectories of the detected vehicles and detailed information on the paths of the passing vehicles is drawn in the image of the camera feed that is recording them. The paths are grouped and colour-coded depicting different types of vehicles. The users are also able to change the time period and filter by the type of vehicle or lane to further investigate. Furthermore, detailed available information about the detected incoming events is represented in a tabular form in the Details Widget.

Regarding the data feeding the DMT, the AVCC component outputs a JSON file with points on each frame of a video with a likelihood value for a pedestrian to be present at that point. SmartViz uses that data to represent them as a heatmap on top of the appropriate camera frame. This use case also presents the distribution of the different types of vehicles passing through a road junction through a widget containing charts and the Temporal Representation widget fed by information of detected vehicles that is the outcome of the SED component.

6 MT1: Monitoring of Crowded Areas

6.1 Integration and deployment of the selected use cases for M18

This section describes the steps taken to integrate the systems such that the MT1: Monitoring of crowded areas could be set up and tested. The following sections describe the architecture and components, the pilot infrastructure and the integration, deployment and testing phase.

6.1.1 Architecture and Components

The MT1 realisation of MARVEL architecture is given in Figure 14, while the employed MARVEL components together with their configurations for MT1 are given in Table 32.

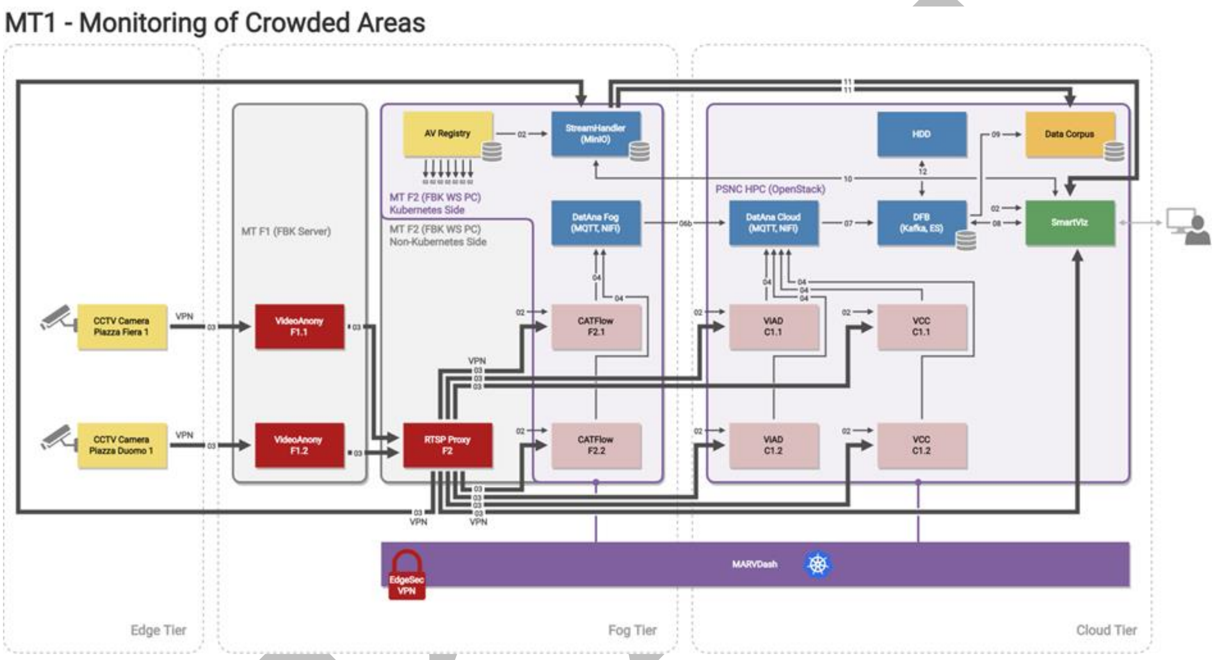


Figure 14: MARVEL R1 deployment and runtime view of the MARVEL architecture for MT1: Monitoring of crowded areas

Table 32: MARVEL components in the MT1

MT1: Monitoring of Crowded Areas			
Component owner	**Subsystem /Component**	**Comments on how the component is used in MT1 for R1**	**Deployment location**
Sensing and perception subsystem			Edge and fog
ITML	AV Registry	AV Registry contains metadata information of all AV sources present in MT1. These include the information on the raw streams produced by the cameras (fps rate, resolution, etc.), but also on the anonymised streams produced by the VideoAnony anonymisation component.	FBK fog WS PC (MT F2-Kubernetes)
MT	Cameras	Two cameras with video footage only and enabled streaming will be used in MT1. One camera will be located at Piazza Fiera and another one at Piazza Duomo.	Edge cameras
Security, privacy, and data protection subsystem			Fog and cloud

FORTH	EdgeSec VPN	EdgeSec VPN creates a secure F2C VPN traffic backbone for all communications within the elements of the MARVEL platform by 100% encryption of the traffic. For MT1, edge layer is not part of the EdgeSec VPN network.	FBK fog WS PC (MT F2-Kubernetes) and cloud (PSNC HPC via OpenStack)
FBK	VideoAnony	VideoAnony will detect individuals that appear in the video footages from the two squares and anonymise their faces. In MT1, the component is present with two instances, once for each of the two camera streams.	FBK fog server (MT F1)
Data management and distribution subsystem			Fog and cloud
ITML	Data Fusion Bus (DFB)	DFB stores inference results of the AI components that participate in MT1.	Cloud (PSNC HPC via OpenStack)
INTRA	StreamHandler	StreamHandler receives continuously anonymised AV data streams and segments them for temporary storage, for later, on-request visual inspection through SmartViz.	FBK fog WS PC (MT F2-Kubernetes)
ATOS	DatAna Fog and Cloud	DatAna for MT1 consists of DatAna Fog and DatAna Cloud components, each continuously consuming, through MQTT, the results of AI inference components deployed at the corresponding layer (Fog/Cloud) and sending to the relevant Kafka topics at the DFB.	FBK fog WS PC (MT F2-Kubernetes) and cloud (PSNC HPC via OpenStack)
CNR	HDD	For the predefined MT1 AI inference components, HDD optimises the allocation of their output streams across a given set of DFB Kafka topics.	Cloud (PSNC HPC via OpenStack)
Audio, visual, and multimodal AI subsystem			Fog and cloud
GRN	CATFlow	In MT1, CATFlow detects pedestrians and provides their trajectories for each of the two camera streams.	FBK fog WS PC (MT F2-Kubernetes)
AU	Visual Anomaly Detection (ViAD)	In MT1, ViAD detects anomalies in crowds and public areas (unseen data) from each of the two camera streams.	Cloud (PSNC HPC via OpenStack)
AU	Visual Crowd Counting (VCC)	In MT1, VCC provides the number of detected people and the associated heatmaps in each of the two camera streams.	Cloud (PSNC HPC via OpenStack)
Optimised E2F2C processing and deployment subsystem			Cloud
FORTH	MARVdash	MARVdash provides a Kubernetes-based deployment environment of all the MT1 components.	Cloud (PSNC HPC via OpenStack)
CNR	DynHP	DynHP trains and compresses the VCC model.	Cloud (PSNC HPC via OpenStack)
UNS	FedL	For MT1, FedL delivers federated training of the VCC model together with GRN4 and UNS1 pilots without explicit exchange of the (raw) data. MT1 implements a VCC-FedL client that communicates through the FedL server, running at the cloud, with similar GRN4 and UNS1 FedL clients.	FBK fog server (MTF1) and Cloud (PSNC HPC via OpenStack)
System outputs: User interactions and the decision-making toolkit			Cloud

ZELUS	SmartViz	SmartViz is focused on visualising the monitoring situations and events in areas of interest, such as exceptional crowd, suspect or unusual crowd movements.	Cloud (PSNC HPC via OpenStack)
STS	Data Corpus-as-a-Service	Data Corpus contains all data collected for MT1 use cases, used for training the relevant AI models.	Cloud (PSNC HPC via OpenStack)

AV data acquisition and anonymisation

The inference data flow starts with the two cameras, one at Piazza Duomo and another one at Piazza Fiera. Each of the cameras produces continuous video footage. We remark that due to the restrictions associated with monitoring of public areas, no audio recording is allowed for this use case. Both streams are transmitted to the FBK fog server (MT F1), and then subsequently consumed by the VideoAnony component located at the same server. Upon anonymisation, specifically, by blurring faces present in the received frames, each of the two VideoAnony instances produces an RTSP stream compiled of anonymised frames. These two streams arrive at the second fog device in the MT pilot infrastructure, FBK WS PC (MT F2-non-Kubernetes). This workstation is connected to both the local FBK network (thus enabling communication with the FBK fog server) but also enabled as a Kubernetes node, i.e., as part of the Kubernetes-managed infrastructure. To enable consumption of the VideoAnony streams by the components residing in the Kubernetes-operated part of FBK WS PC (MT F2), an additional streaming service is implemented at the non-Kubernetes part of FBK WS PC (MT F2) – RTSP Proxy service.

Real-time AI inference

The anonymised streams arriving from the RTSP Proxy service at FBK WS PC (MT F2-non-Kubernetes) are received and processed by CATFlow, ViAD and VCC, each instantiated two times.

- **CATFlow** – In MT1, both instances of CATFlow are running at the Kubernetes-managed part of the FBK WS PS (MT F2 - Kubernetes), processing the two streams from Piazza Duomo and Piazza Fiera, respectively. The goal of this component is to detect pedestrians in the two public squares and track their trajectories from the anonymised stream produced by VideoAnony. After the results are generated, they are transmitted to appropriate MQTT brokers of the DatAna fog agent residing at the same node, FBK WS PS (MT F2 - Kubernetes).
- **ViAD** – Each of the two instances of ViAD is running at the cloud (PSNC HPC via OpenStack). Each of the two instances is trained using typical video footages from the corresponding square where the component instance will be applied, and it recognises any deviation from normal, i.e., events and such that differ from previously seen data. After the results are generated, they are transmitted to appropriate MQTT brokers of the DatAna cloud agent (PSNC HPC via OpenStack).
- **VCC** – Similarly as with ViAD, the two instances of VCC are running at the cloud (PSNC HPC via OpenStack). The component is employed in MT1 to provide the number of detected people and the corresponding heatmaps, indicating the crowd density. In each component instance, after the results are generated, they are transmitted to appropriate MQTT brokers of the DatAna cloud agent (PSNC HPC via OpenStack).

More details on each of the AI components in MT1 can be found in Section 6.2.2.

Inference results management and on-the-fly transformations

As indicated in the paragraph above, after the inference results have been generated, they are submitted to the respective MQTT broker of the DatAna agent residing at the same node. Specifically, the CATFlow instances running at FBK fog WS PC edge submit their results to the MQTT broker of DatAna fog agent running at the same fog node – FBK fog WS PC (MT F1-Kubernetes). The two instances of ViAD and VCC submit their results to the MQTT broker of DatAna cloud (PSNC HPC via OpenStack). DatAna fog agent, after receiving the inference results, applies an appropriate transformation to standard data models, referred to in Section 3.2. Specifically:

- DatAna fog (MT F2 – Kubernetes) receives inference results of CATFlow and sends the transformed results to DatAna Cloud;
- DatAna cloud (PSNC HPC via OpenStack) receives the results from DatAna fog and sends them to the Kafka topics of DFB;
- DatAna cloud (PSNC HPC via OpenStack) receives the results from ViAD and VCC instances, transforms them, and sends results to the appropriate Kafka topics of DFB.

Inference results fusion and storage

After being gathered in the appropriate Kafka topics in DFB (PSNC HPC via OpenStack), where one Kafka topic per each AI component is configured and enabled, results are consumed by SmartViz and Data Corpus, as detailed below. Finally, DFB passes also results from Kafka topics to Elasticsearch for persistent storage.

AV data storage

Concurrently with processing the RTSP (anonymised) streams for AI inference, another component, StreamHandler, is also subscribed to receive these streams. StreamHandler buffers each of the streams for further segmentation that occurs at regular time intervals, and finally stores the respective AV snippets in the MinIO database. The buffering intervals and segmentation parameters are configurable by the end-user. When a user wishes to inspect the AV data in an interval of interest (e.g., upon detection of an anomaly or raised alert by SmartViz), the user will submit a request to StreamHandler with the onset and offset interval times. StreamHandler then compiles the relevant AV data segments into a single file and sends back to SmartViz the compiled AV data section.

Visualisations and UI interface

The last component of the real-time inference pipeline, also serving as the UI of the platform, is SmartViz. This component runs at the cloud (PSNC HPC via OpenStack). Several functionalities are supported: 1) advanced visualisations, with a dashboard for user configurations and user interactions, 2) live AV feed for real-time inspection of the monitored areas, 3) on-request inspection of stored AV data, and 4) user-based verification of AI inference results. First, visualisations of AI inference results are provided in the form most suitable to the end-user. For MT1, this consists of a map indicating locations of detected anomalies and other visualisations, as described in Section 6.4. Live, anonymised feed from the two cameras (Piazza Fiera and Piazza Duomo), arriving from the RTSP Proxy service, is shown upon user request. Also, for inspection of historical AV data or near real-time, e.g., in case an event has been detected by the system and the user was not able to follow the streams in real-time, the user may submit a request to the StreamHandler with onset and offset times indicating the time interval of interest. SmartViz receives the respective AV data segment, as described in the above, and displays it for user inspection. Finally, after inspecting an AV data segment, either through live feed or from StreamHandler, the user may verify the relevant inference result. This is done by submitting the verification result to the corresponding Kafka topic in DFB, upon which the relevant entries in the Elasticsearch database are changed.

6.1.2 Pilot E2F2C infrastructure

The MT1 infrastructure provided for the R1 integration consists of 6 IP cameras transmitting video and 2 workstations, managed by FBK.

The following subsections will describe each component in detail.

Edge devices

The MT1 infrastructure consists of 6 IP cameras transmitting video for the edge layer, 3 in Piazza Fiera and 3 in Piazza Duomo. For the R1 streaming pipeline demonstration, we will be using one camera stream from Piazza Fiera and one camera stream from Piazza Duomo. The upload function is a secure transmission by VPN access between MT and FBK in which raw video will be sent to the data lake in FBK.

Table 33 lists the specifications for each camera.

Table 33: MT1 Sensing devices

Specifications Type	Piazza Fiera Cameras	Piazza Duomo Cameras
IP camera model	Digital cameras - Basler BIP2-1600c-dn	Digital cameras - Basler BIP2-1600c-dn
Resolution	1600 x 1200	1600 x 1200
Frame Rate	12 fps	12 fps
Video Encoding	H.264	H.264

Fog workstations

FBK provides the Fog tier for the MT use cases. In order to comply with the constraints in the agreement that granted FBK access to the raw data of the MT's sensors and to satisfy the requirements of the MARVdash Kubernetes cluster, FBK deploys two workstations, both with GPU.

Table 34 lists the specifications of the workstation 2. Workstation 1 is an FBK internal machine which may change and will not be accessed by users external to the research organisation.

Table 34: Specification of the FBK Fog workstation in the Kubernetes cluster

Component	Specifications
CPU	Intel Xeon E5-1620
GPU	Tesla K40 11GB

Hard Drive	512GB
RAM	20GB

6.1.3 Deployment

As mentioned in Section 3.1, MARVdash and, by extension, Kubernetes are also present in this use case.

The fog layer of the infrastructure for the MT use case is actually hosted in FBK's infrastructure. FBK is hosting two workstations that are located at the fog layer. Only one of them is meant to be part of the Kubernetes cluster. The aforementioned setup introduced new requirements in order to comply with security policies present at the FBK network. The first requirement is to route all the traffic via the EdgeSec VPN and not only the traffic that matches the network subnet defined by the EdgeSec VPN. The second requirement is that EdgeSec VPN should not interfere with the communication between the two workstations internally.

The first requirement is addressed by introducing a new VM at the cloud to act as a gateway to the internet for the workstation at the FBK. This gateway will route all the traffic originating from the workstation through PSNC's infrastructure. Regarding the second requirement, modifications made to the routing table of the workstation did not interfere with the internal communication of the two workstations.

The final outcome is depicted in Figure 15. Workstation 1 and Workstation 2 are connected internally via a switch. Workstation 1 accesses the internet via the main router of the FBK's network, whereas Workstation 2 accesses the internet via the Virtual Machine hosting the VPN gateway at the cloud in PSNC's infrastructure.

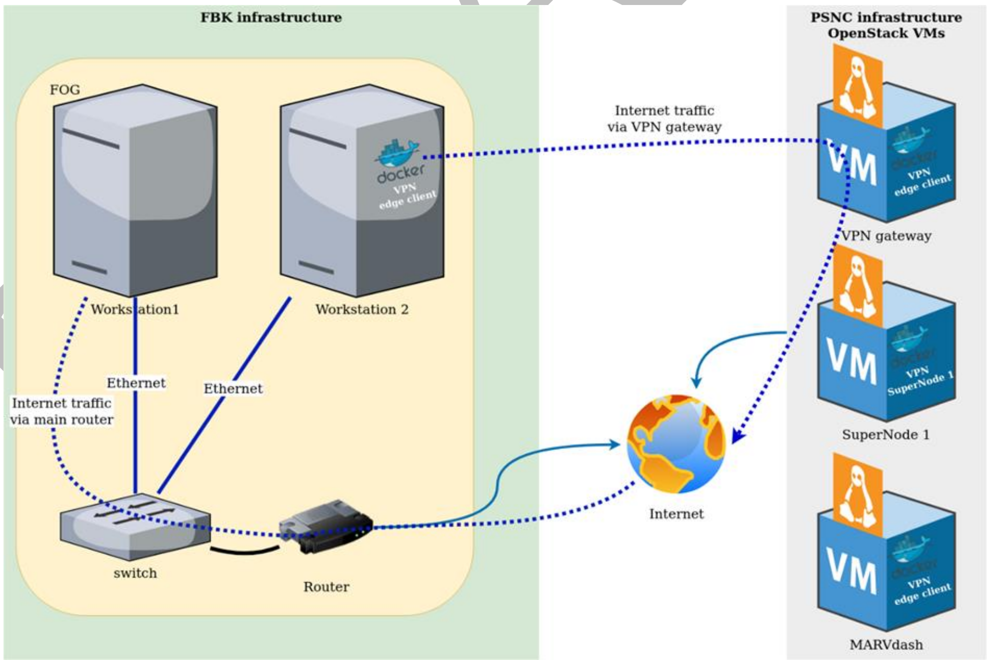


Figure 15: EdgeSec VPN in MT use cases

6.1.4 Analysis of real-life data streams

In the case of MT1, video data is being recorded using 6 IP cameras of resolution 1600x1200 pixels, 3 in Piazza Fiera and 3 in Piazza Duomo. For the R1 streaming pipeline demonstration, we will be using one camera stream from Piazza Fiera and one camera stream from Piazza

Duomo. Data capturing is performed using 12 frames per second, H.264 codec is used and data is stored using the MP4 format.

6.1.5 Integration and testing

The integration of MT1 components is based on the reference architecture of the ‘AI Inference Pipeline’.

Due to restrictions associated with data privacy regulations and network security concerns expressed by the Municipality of Trento (MT), a customised solution was implemented to allow MARVEL components to access the AV data streams of the MT1 CCTV cameras at the Piazza Duomo and Piazza Fiera locations. According to data privacy regulations, access to the raw AV data feeds from the CCTV cameras that are part of the MT network is restricted only to authorised personnel and to trusted devices from within the network. Therefore, it was not possible to provide MARVEL components with direct access to the raw AV data, but only to anonymised versions of it. Following deliberations with the MT authorities and based on an agreement that nominates FBK as a data processor under certain constraints, a solution was reached that involved the use of an external server (MT F1), which is managed by FBK. However, this server (MT F1) could not be attached to the MARVEL Kubernetes cluster, because such an action would violate the conditions and security policies foreseen by the agreement between MT and FBK. Therefore, it was necessary to deploy VideoAnony services directly on the MT F1 server at the Fog, without them being hosted on the MARVEL Kubernetes cluster. These VideoAnony services could consume the raw AV streams through a VPN tunnel. In addition, a second Fog server was set up at FBK (MT F2), which was allowed to (i) be attached to the MARVEL Kubernetes cluster and (ii) gain access to the VideoAnony services hosted on the first infrastructure node of FBK (MT F1) through an RTSP Proxy service that was not part of the Kubernetes cluster, but could re-stream the output of VideoAnony towards the Kubernetes cluster using a VPN tunnel. Therefore, using this configuration, it was possible for all components that are hosted within the MARVEL Kubernetes cluster to gain access to anonymised AV data streams produced by VideoAnony at MT F1.

Based on the infrastructure that was available at the pilot site and at the FBK premises, certain MARVEL components were selected to be deployed at the fog infrastructure node MT F2, within the Kubernetes cluster and through MARVdash. Due to the ability of CATFlow to operate with reduced computational resources, it was selected for deployment at MT F2 among the AI components that were foreseen to be applied for MT1. Two CATFlow instances were deployed, with each one processing a different anonymised AV data stream being produced by VideoAnony instances at MT F1 and relayed by the RTSP Proxy service at MT F2. StreamHandler and AV Registry were also deployed at the fog layer (MT F2), as this is the recommended layer for their operation according to the reference ‘AI Inference Pipeline’ architecture. A DatAna MQTT and DatAna Nifi services were also deployed at MT F2 to collect and process the output of the CATFlow instances.

Each component that needed to access an AV data stream produced by a VideoAnony instance and relayed by the RTSP Proxy service (MT F2) had to make a REST API request to the AV Registry component at MT F2 to receive the details (e.g., url, AV metadata) of the AV source (VideoAnony instance) it needed to connect to.

The RTSP protocol was implemented for delivering the AV data streams from the VideoAnony instances to the respective AI components, StreamHandler and SmartViz.

Each AI component published its output of raw inference results as messages structured in JSON format to an MQTT broker that resided on the same infrastructure node as the AI component.

DatAna agents subscribed to the topics of the MQTT brokers residing on the same layer to receive AI raw inference results. Each DatAna agent transformed the inference results into data models that are compliant with the Smart-Data-Models and relayed them to a DatAna agent at a higher layer (DatAna Edge to DatAna Fog, DatAna Fog to DatAna Cloud). Finally, the DatAna Cloud published the inference results collected from all layers to dedicated Kafka topics at the DFB, which was deployed at the cloud layer.

SmartViz was deployed at the cloud layer and could access the inference results that were aggregated at the DFB (both real-time results published at the Kafka topics and historical results stored in the DFB Elastic Search repository). SmartViz was responsible for visualising these results and for transferring information on results that were verified by the user back to the DFB for updating the respective entries in the Elastic Search.

SmartViz could also request AV data segments from StreamHandler (deployed at MT F2) that corresponded to specific inference results selected by the user and receive the corresponding AV data segments to be displayed to the user.

The Data Corpus deployed at the cloud layer could access the inference results and verification information published in the Kafka topics in real-time from DatAna Cloud and SmartViz, respectively, by subscribing to these Kafka topics. In addition, the Data Corpus could access AV data that were produced by StreamHandler after processing the live AV data streams it was receiving.

HDD deployed at the cloud layer could exchange information with the DFB through a REST API to provide recommendations for optimising the DFB Kafka topic configuration.

MARVdash was used to orchestrate the deployment of all components at all infrastructure nodes.

During the partial integration testing and end-to-end integration testing activities, several issues were resolved, including the following:

- appropriate configuration of the infrastructure nodes;
- appropriate deployment configuration of the components, service names and exposed ports;
- exposing services so that they are reachable from other services;
- fine-tuning AV data streams;
- fixing inconsistencies in inference result data.

6.2 Multimodal and privacy aware intelligence for MT1

This section presents the work done to configure the framework according to the needs of the MT1 in terms of consolidation and ingestion of audio-visual data, training and testing of AI models, design of the pipeline data flow from the edge to the cloud and analysis of the output obtained.

6.2.1 Datasets for model training and privacy assurance

This section describes the methods and approaches taken towards collecting the AV data, the datasets required for model training, the analysis of datasets and streams and privacy assurance and anonymisation methods used in the use case.

6.2.1.1. *Datasets for model training*

MT has provided data for various components. MT has provided the dataset "TrentoOutdoor – real recording" (as defined in D2.1) processing the streams of cameras from Piazza Duomo and Piazza Fiera, respectively.

MT, in collaboration with FBK, collected and anonymised more than 500 videos of 3 minutes each, in total 165 GB used for the training of:

- CATFlow – MT contributed 70 videos which contain pedestrians in crowded and not crowded situations;
- ViAD –MT contributed 70 videos which contain anomaly and not anomaly situations (crowded situation mostly);
- VCC – MT annotated 170 videos, 700 frames in total using the CVAT software to provide the number of detected people.

6.2.1.2. *Analysis of datasets*

The analysis of the streams and datasets is performed by the AI model providers. With their feedback, more data for training will be added.

6.2.1.3. *Privacy assurance and anonymisation*

This use case involves the use of cameras only. The collected data are anonymised at FBK premises using a batch version of the VideoAnony tool. This is compliant with the request of the MT DPO.

6.2.2 Training and testing of the models

The MT1 use case employs three AI models, namely Visual Crowd Counting (VCC) and Visual Anomaly Detection (ViAD), and CATFlow. The VCC model has two variants, one trained using the standard centralised training approach, and another one which is implemented within a Federated Learning framework to achieve distributed privacy-preserving training of the underlying Deep Learning model. Detailed description of the methodologies proposed by MARVEL partners defining the underlying models of these AI functionalities can be found in D3.1, while details on the Federated Learning framework used for implementing some of the AI functionalities to achieve distributed privacy-preserving training of Deep Learning models involved in the AI functionalities will be included in D3.4. In the following, a description of these models and their training using data of the use case are provided.

6.2.2.1. *Multimodal AI realisation*

- **CATFlow** – In the MT1 use case, CATFlow was identified to be a good component to track pedestrians for the Piazza Fiera camera. Following a number of tests, it was concluded that CATFlow would also be able to provide the trajectories of these pedestrians. Another camera from this use case at Piazza Duomo is characterised by a less than ideal angle and distance from the pedestrians. Despite this challenge to the AI component, CATFlow will still be used for this camera. The vehicle tracking aspect of CATFlow is still employed to track the occasional cyclist passing through or service vehicles that could pass through the area, even if marked as a pedestrian area.
- **Visual Crowd Counting (VCC)** – The MT1 use case employs a Visual Crowd Counting (VCC) model to estimate the number of people present in the visual live feed. Crowd counting is the problem of identifying the total number of people present in a scene, who can be observed in a given image of that scene. The input to a crowd counting model is a colour image, and the expected output is a number estimating the

total number of people present in that image. Optionally, the output of a crowd counting model can be a density map which specifies the density of the crowd for each pixel of the input image, and the total count can be calculated by summing up all the density values for all pixels. Information related to the underlying machine learning models developed by MARVEL partners for visual and audio-visual crowd counting can be found in D3.1. For the MT1 use case, the visual crowd counting functionality has not been yet tested on a subset of the MARVEL data. The corresponding model receiving as input visual data has not been trained using this data yet.

- **Visual Anomaly Detection (ViAD)** – The MT1 use case also employs a Visual Anomaly Detection (ViAD) model to detect previously unseen, novel events in the visual live feed. Visual Anomaly Detection is the task of identifying novel situations in a scene, based on the visual information captured in an input video or image. Models are trained to learn and replicate situations considered normal in a certain setting, e.g., pedestrians on a pathway. Situations that do not occur often, for example, a car on the pathway, are not present in the data used to train the model to identify as normal and, thus, they will be detected as anomalies. Information related to the underlying machine learning model developed by MARVEL partners for detecting anomalies based on visual information can be found in D3.1. An anomaly detection model based on visual information will be developed in the second part of the project, and it will be trained and tested on MARVEL data from MT1.

6.2.2.2. *Federated learning realisation*

As explained in Section 3.3.3, MT1 also implements a VCC-FedL client that participates in federated training of the VCC model. The MT1 VCC-FedL client runs on a GPU of the MT fog workstation (MT F1-Kubernetes). The client also communicates with the VCC-FedL server that runs at the PSNC-based MARVEL cloud (PSNC HPC via OpenStack).

6.3 Analysis of the outputs from all real-life smart city experiments in MT1

Table 35 summarises the characteristics or parameters against which the use case will be evaluated. Most of these characteristics are directly dependent on the non-functional and functional use case and asset KPIs as listed in D1.2, tables 6.2-6.3 and 6.5-6.11. Evaluation will be performed later and reported in D6.2.

Table 35: Parameters to determine for use case MT1

Use case	Parameter	How to measure	target to be achieved
MT1: Monitoring of crowded areas	Efficiency *Related to the efficiency of the system as used in the use case*	Efficiency is largely dependent on how long it takes to detect and flag targeted events. The average time taken to detect and flag the event	On average less time taken when compared to a single person observing multiple cameras
	Operability *Related to the ability of the components to keep functioning together*	Install additional cameras and microphones	System discovers sensors or adding sensors does not disrupt operation
	Usability *Related the how well the system helps the users to achieve a task in a given use case*	Interview MT staff and local police to measure the end-user experience	End-users finds the system easy to use

	Robustness *Related to how robust are the system components during the period of operation*	Scale system to other cameras and microphones	Performance sustained as additional sensors are added
	Performance *Related to how well the system performs the intended task*	Counting the number of detected targeted events in crowds	10% increase in the detection of events when compared to a single person observing multiple cameras
	Accountability *Related to the system being able to explain results or decisions.*	System stores video snippet of targeted event	Number of times system fails to store video snippet
	Transparency *Related to the description of the processes or algorithms that are used to generate system output*	Decision processes are described in a document.	Document availability
	Privacy awareness *Related to the provision of adequate governance mechanisms that ensure privacy in the use of data.*	Count data breach reports.	Minimisation of count.

6.4 Demonstration

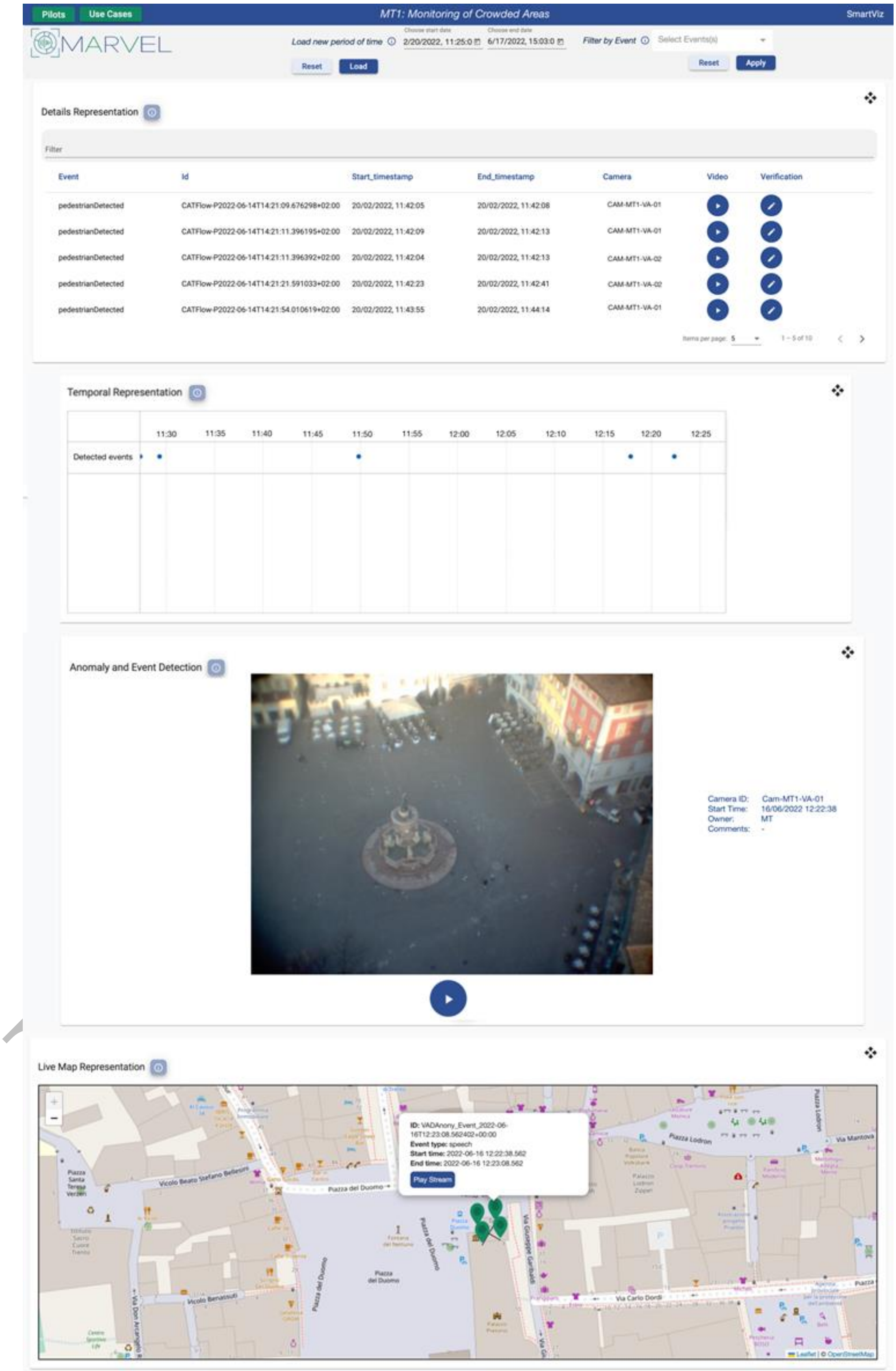
6.4.1 The Decision-making Toolkit

This use case is focused on the selection of views of areas of interest for monitoring situations such as exceptional crowd, suspect or unusual crowd movements, etc. The areas for this scenario are Piazza Fiera, a square hosting the “Christmas Markets” in Trento and Piazza Duomo, where the weekly market is located. Both locations host other crowded events. On these occasions (when the squares are crowded) the number of robberies and aggressions can increase. In addition, first aid may be needed for people who are unwell or faint. In order to be informed of these actions in time, alerts should be sent to the local police operational centre as quickly as possible. Visual analysis should be carried out in real-time and on recording data saved on the servers of the Local Police. The User Stories for this use case are shown in Table 36 and give a better understanding of the functionality expected.

Table 36: The user stories for the MT1 – Monitoring of Crowded Areas

Title	Main user stories	Sub user stories and user intent
Exceptional crowd, suspect or unusual crowd movements	**As a** Local Police officer, **I want** to be alerted if an anomaly, such as an overly-dense crowd, and unusual crowd movement, occurs in the squares, **so that** I can further focus on the relative views, analyse them and determine a course of action.	**As a Local Police officer,** • once informed of anomalous events in crowded areas, I can activate appropriate actions such as dispatching the relevant authorities. • once there is an anomaly flagged, I am alerted and notified which live feed from which camera (the one capturing the location of the event) should be checked with attention and I should be able to view the feed a few minutes before the anomaly happened, to accurately assess the cause. • once I review the camera feed of the flagged anomaly, I have the option to verify if this event's analysis is indeed correct or not. This option, similar to GRN3 use case, allows for continuously improving the system. • I also want to: ○ view the anomalous events on the map, ○ compare events in different contexts (e.g.: weekly market vs Christmas market), ○ view the timeline of events, ○ have a clustering of events, • I can prevent or be better prepared for a similar situation in the future.

Figure 16 depicts the dashboard that combines the user stories of MT1 and contains all the widgets and visualisation schemas described below.



Load new period of time: Choose start date 2/20/2022, 11:25:0 Choose end date 6/17/2022, 15:03:0

Filter by Event: Select Events(s)

Filter

Event	Id	Start_timestamp	End_timestamp	Camera	Video	Verification
pedestrianDetected	CATFlow-P2022-06-14T14:21:09.676298+02:00	20/02/2022, 11:42:05	20/02/2022, 11:42:08	CAM-MT1-VA-01	▶	✎
pedestrianDetected	CATFlow-P2022-06-14T14:21:11.396195+02:00	20/02/2022, 11:42:09	20/02/2022, 11:42:13	CAM-MT1-VA-01	▶	✎
pedestrianDetected	CATFlow-P2022-06-14T14:21:11.396392+02:00	20/02/2022, 11:42:04	20/02/2022, 11:42:13	CAM-MT1-VA-02	▶	✎
pedestrianDetected	CATFlow-P2022-06-14T14:21:21.591033+02:00	20/02/2022, 11:42:23	20/02/2022, 11:42:41	CAM-MT1-VA-02	▶	✎
pedestrianDetected	CATFlow-P2022-06-14T14:21:54.010619+02:00	20/02/2022, 11:43:55	20/02/2022, 11:44:14	CAM-MT1-VA-01	▶	✎

Items per page: 5 1 – 5 of 10

	11:30	11:35	11:40	11:45	11:50	11:55	12:00	12:05	12:10	12:15	12:20	12:25
Detected events	●				●					●	●	

Camera ID: Cam-MT1-VA-01
Start Time: 16/06/2022 12:22:38
Owner: MT
Comments: -

ID: VADAnony_Event_2022-06-16T12:23:08.562402+00:00
Event type: speech
Start time: 2022-06-16 12:22:38.562
End time: 2022-06-16 12:23:08.562

Figure 16: DMT MT1 Dashboard

The users of this use case are local police officers that want to be alerted if an anomaly, such as an overly dense crowd, and unusual crowd movement, occurs in the squares, so they can further focus on the relative views, analyse it and determine a course of action. In order to notify the users of the detected alerts, their location and any available information, we chose the Real-time Map Representation widget. The users are also given the opportunity to see either the video stream of the camera, a video corresponding to an event, or a video that occurred in a selected time frame through the Video Player widget. The inference results and all the available information linked to the detected alerts are also represented in the Details widget, where the user can also verify their validity, and in the Temporal Representation widget.

7 MT3: Monitoring of Parking Places

7.1 Integration and deployment of the selected use cases for M18

This section describes the steps taken to integrate the systems such that the MT3: Monitoring of parking places could be set up and tested. The following sections describe the architecture and components, the pilot infrastructure and the integration, deployment and testing phase.

7.1.1 Architecture, components and E2F2C streaming data pipeline

The MT3 realisation of MARVEL architecture is given in Figure 17, while the employed MARVEL components together with their configurations for MT3 are given in Table 37.

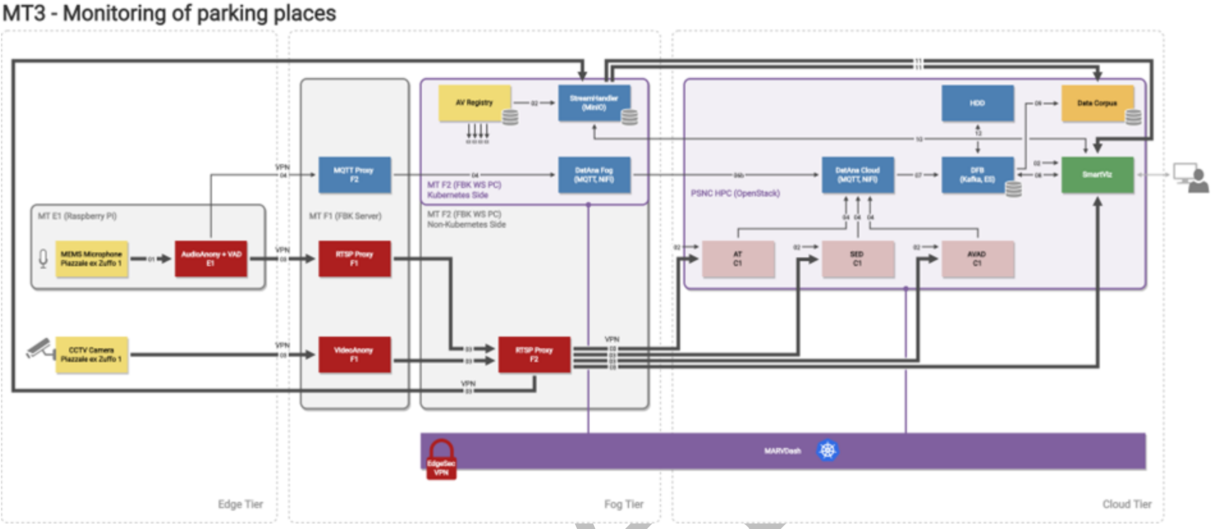


Figure 17: MARVEL R1 deployment and runtime view of the MARVEL architecture for MT3: Monitoring of parking places

Table 37: MARVEL components in the MT3

MT3: Monitoring of Parking Places			
Component owner	**Subsystem /Component**	**Comments on how the component is used in MT3 for R1**	**Deployment location**
Sensing and perception subsystem			Edge and fog
ITML	AV Registry	AV Registry contains metadata information of all AV sources present in MT3. These include the information on the raw streams produced by the camera and the microphone (fps rate, resolution, etc.), but also on the anonymised streams produced by the VideoAnony and AudioAnony anonymisation components.	FBK fog WS PC (MT F2-Kubernetes)
IFAG	MEMS microphone	One MEMS microphone connected to a Raspberry Pi will be used in MT3, specifically at Piazzale ex Zuffo.	Edge microphone
MT	Camera	One camera with enabled streaming will be used in MT3, specifically at Piazzale ex Zuffo.	Edge camera
Security, privacy, and data protection subsystem			Edge, fog and cloud

FORTH	EdgeSec VPN	In MT3, EdgeSec VPN creates a secure F2C VPN traffic backbone for all communications within the elements of the MARVEL platform by 100% encryption of the traffic. For MT3, the edge layer is not part of the EdgeSec VPN network.	FBK fog WS PC (MT F2-Kubernetes) and cloud (PSNC HPC via OpenStack)
FBK	VideoAnony	VideoAnony will detect individuals that appear in the video footages from the Piazzale Ex Zuffo parking place and anonymise their faces. As MT3 uses one camera stream, the component is present with one instance.	FBK fog server (MT F1)
AUD	VAD	VAD detects speech segments in an audio stream and outputs the respective onset and offset times.	Raspberry Pi (MT E1)
FBK	AudioAnony	Based on the onset and offset times of speech segments detected by VAD, AudioAnony will anonymise the respective segments of the audio. As MT3 uses one audio stream, the component is present with one instance.	Raspberry Pi (MT E1)
Data management and distribution subsystem			Edge, fog and cloud
ITML	Data Fusion Bus (DFB)	DFB stores inference results of the AI components that participate in MT3.	Cloud (PSNC HPC via OpenStack)
INTRA	StreamHandler	StreamHandler receives continuously anonymised AV data streams and segments them for temporary storage, for later, on-request visual inspection through SmartViz.	FBK fog WS PC (MT F2-Kubernetes)
ATOS	DatAna Fog and Cloud, MQTT Edge	DatAna for MT3 consists of DatAna Fog and DatAna Cloud components, each continuously consuming, through MQTT, the results of AI inference components deployed at the corresponding layer (Fog/Cloud) and sending to the relevant Kafka topics at the DFB. An additional MQTT service is provided at the Raspberry Pi to collect VAD outputs.	Raspberry Pi (MT E1), FBK fog WS PC (MT F2-Kubernetes) and cloud (PSNC HPC via OpenStack)
CNR	HDD	For the predefined MT3 AI inference components, HDD optimises the allocation of their output streams across a given set of DFB Kafka topics.	Cloud (PSNC HPC via OpenStack)
Audio, visual, and multimodal AI subsystem			Cloud
AU	Audio-Visual Anomaly Detection (AVAD)	In MT3, AVAD detects anomalies in the monitored parking place (e.g., wrongly parked cars).	Cloud (PSNC HPC via OpenStack)
TAU	Sound Event Detection (SED)	In MT3, SED detects audio events that occur in parking places (e.g., parking breaks).	Cloud (PSNC HPC via OpenStack)
TAU	Audio Tagging (AT)	Similarly, as with SED, AT detects events in consecutive time intervals.	Cloud (PSNC HPC via OpenStack)
Optimised E2F2C processing and deployment subsystem			Cloud
FORTH	MARVdash	MARVdash provides a Kubernetes-based deployment environment for all MT3 components operating under part of the MT infrastructure managed by Kubernetes.	Cloud (PSNC HPC via OpenStack)
System outputs: User interactions and the decision-making toolkit			Cloud

ZELUS	SmartViz	SmartViz indicates the presence of anomalies in the Parking place Ex Zuffo.	Cloud (PSNC HPC via OpenStack)
STS	Data Corpus-as-a-Service	Data Corpus contains all data collected for MT3 use cases, used for training the relevant AI models.	Cloud (PSNC HPC via OpenStack)

AV data acquisition and anonymisation

The inference data flow starts with the AV data of MT3 consisting of the separate streams from the camera and the microphone at the Piazzale ex Zuffo parking place. The camera live feed is streamed to the FBK server (MT F1), and then subsequently consumed by the VideoAnony component, running on the same machine. Upon anonymisation, specifically, by blurring faces and car number plates, VideoAnony produces an RTSP stream compiled of anonymised frames to be consumed by several components. In parallel, the microphone on the ground acquires an audio stream that is passed to the Raspberry Pi (MT E1) and processed by VAD, running on the same device, for detection of speech segments. VAD outputs onset and offset times of the detected speech segments and passes them to AudioAnony to indicate which intervals in the microphone stream require audio anonymisation. AudioAnony anonymises the corresponding audio segments, compiles them together with the remaining audio segments, and finally produces a (single) RTSP stream. The outputs of VAD are submitted to MQTT DatAna brokers residing at the FBK server (MT F1) to be further utilised in the platform (e.g., for audio content inspection by SmartViz). The RTSP stream from VideoAnony is received by an RTSP proxy service residing at the FBK WS PC (MT F2-non-Kubernetes), from where the stream is further forwarded to the AI components for inference. The RTSP stream from AudioAnony travels one more hop than the video counterpart: prior to the RTSP proxy service at the FBK WS PC, this stream goes to an RTSP proxy "relay" located at the FBK server (MT F1), as indicated in the figure. The reason for employing the two RTSP proxy services is to enable data handover between the non-Kubernetes and Kubernetes part of the MT infrastructure.

Real-time AI inference

The anonymised streams arriving from the RTSP Proxy service at FBK WS PC (MT F2-non-Kubernetes) are received and processed by AVAD, SED and AT, each instantiated once, and running at the cloud (PSNC HPC via OpenStack).

- **AVAD** – In MT3, the component is trained using typical video footages from the Piazzale ex Zuffo parking place, and it recognises any deviation from normal, i.e., events and such that differ from previously seen data. After the results are generated, they are transmitted to the MQTT brokers of the DatAna cloud agent (PSNC HPC via OpenStack).
- **SED** – The component is employed in MT3 for the detection of events from the audio stream relevant to the parking place that is monitored. After the results are generated, they are transmitted to the MQTT broker of the Cloud DatAna agent (PSNC HPC via OpenStack).
- **AT** – Similarly as with SED, the component is employed in MT3 to enable tagging of parking audio events at consecutive time intervals. After the results are generated, they are transmitted to the MQTT broker of the Cloud DatAna agent (PSNC HPC via OpenStack).

More details on each of the AI components in MT3 can be found in Section 7.2.2.

Inference results management and on-the-fly transformations

The data management pipeline for the inference results of VAD, AVAD, SED and AT is outlined next:

- MQTT broker running at the FBK fog server (MT F1) receives inference results of VAD (running at the edge) and sends the transformed results to DatAna fog;
- DatAna fog running at the FBK fog WS PC (MT F2-Kubernetes) receives VAD results from MQTT, transforms them, and forwards to DatAna Cloud;
- DatAna cloud (PSNC HPC via OpenStack) receives the transformed VAD results from DatAna fog and sends them to the appropriate Kafka topics of DFB;
- DatAna cloud (PSNC HPC via OpenStack) concurrently receives the AVAD, SED, and AT results, transforms them and sends to the appropriate Kafka topics of DFB.

Inference results fusion and storage

After being gathered in the appropriate Kafka topics in DFB (PSNC HPC via OpenStack), where one Kafka topic per each AI component is configured and enabled, results are consumed by SmartViz and Data Corpus, as detailed below. Finally, DFB passes also results from Kafka topics to Elasticsearch for persistent storage.

AV data storage

Concurrently with processing the RTSP (anonymised) streams for AI inference, another component, StreamHandler, is also subscribed to receive these streams. StreamHandler buffers each of the streams for further segmentation that occurs at regular time intervals, and finally stores the respective AV snippets in the MinIO database. The buffering intervals and segmentation parameters are configurable by the end-user. When a user wishes to inspect the AV data in an interval of interest (e.g., upon detection of an anomaly or raised alert by SmartViz), the user will submit a request to StreamHandler with the onset and offset time intervals. StreamHandler then compiles the relevant AV data segments into a single file and sends back to SmartViz the compiled AV data section.

Visualisations and UI inference

The last component of the real-time inference pipeline, also serving as the UI of the platform, is SmartViz. This component runs at the cloud (PSNC HPC via OpenStack). Several functionalities are supported: 1) advanced visualisations, with a dashboard for user configurations and user interactions, 2) live AV feed for real-time inspection of the monitored areas, 3) on-request inspection of stored AV data, and 4) user-based verification of AI inference results. First, visualisations of AI inference results are provided in the form most suitable to the end user. For MT3, this consists of a map indicating locations of detected anomalies and other visualisations, as described in Section 7.4. Live, anonymised feeds from Piazzale ex Zuffo camera and microphone, arriving from the RTSP Proxy service, are shown upon user request. Also, for inspection of historical AV data or near real-time, e.g., in case an event has been detected by the system and the user was not able to follow the streams in real-time, the user may submit a request to the StreamHandler with the onset and offset times indicating the time interval of interest. SmartViz receives the respective AV data segment, as described in the above, and displays it for user inspection. Finally, after inspecting an AV data segment, either through live feed or from StreamHandler, the user may verify the relevant inference result. This is done by submitting the verification result to the corresponding Kafka topic in DFB, upon which the relevant entries in the Elasticsearch database are changed.

7.1.2 Pilot E2F2C infrastructure

The MT3 infrastructure provided for the R1 integration consists of two IP cameras transmitting video, two devices consisting of IFAG microphones and a Raspberry Pi transmitting audio data, and two workstations at the fog layer, managed by FBK.

The following subsections will describe each component in detail.

Edge devices

The MT3 edge layer infrastructure comprises of two IP cameras transmitting video, two devices consisting of IFAG microphones and a Raspberry Pi, the latter runs a voice activity detection and a speech anonymiser and streams via RTSP anonymised data. The upload function is a secure transmission by VPN access between MT and FBK in which raw data will be sent to the data lake in FBK.

IFAG has provided the boards AudioHub Nano and Audiohub - Nano 4 Mic Version as the hardware necessary to collect the audio. Those devices stream mono, stereo, or 4channels audio data. Both use the standard USB Audio protocol, supported by the edge devices planned to be used in this demonstrator (Intel NUCs and Raspberry Pi). No driver installation is required, and recording can be performed by using most audio recording software and libraries. User can choose the desired number of channels to record from and the sampling rate. With the on-board switch the operating mode and gain configuration can be selected (from 0 to 24dB) to better suit the recording scenario. An extended technical explanation of the devices can be found in deliverable D4.1[8], section 2.2.

Table 38 lists the specifications for each device.

Table 38: MT3 Sensing devices

Specifications Type	Piazzale ex Zuffo
IP camera model	Digital cameras - Basler BIP-1600c
Resolution	1600 x 1200
Frame Rate	2 fps
Video Encoding	H.264
Microphones	IFAG-MEMS
Audio Encoding	ACC (LC)
Audio Sampling Rate	16kHz
Audio Stream Bitrate	Mono 69 kbps
RPi	Raspberry Pi 4 Model B 8GB RAM – Micro SD 32GB

[8] "D4.1: Optimal audio-visual capturing, analysis and voice anonymisation – initial version," Project MARVEL, 2020. https://doi.org/10.5281/zenodo.5833277

Fog workstations

The pilot infrastructure is the same as that used in MT1 and the reader is referred to section 6.1.2.

7.1.3 Deployment

Since MT1 and MT3 share the same infrastructure, the issues faced during component deployment were the same. They are already described in subsection 6.1.3.

7.1.4 Analysis of real-life data streams

In the case of MT3, video data is being recorded using two IP cameras of resolution 1600x1200 pixels. Data capturing is performed using two frames per second, H.264 codec is used and data is stored using the MP4 format.

Audio is being captured using IFAG-MEMS at sampling frequency equal to 16kHz whereas bit depth is 16 bits per sample. Taking into account that mono audio is recorded, the bit-rate is 69 kbps. Audio data is recorded using WAV format.

7.1.5 Integration and testing

The integration of MT3 components is based on the reference architecture of the ‘AI Inference Pipeline’.

Similar restrictions that applied in MT1 were also present in MT3, related to data privacy regulations and network security concerns expressed by the Municipality of Trento (MT). In order to overcome these restrictions, a similar solution as in MT1 was employed for MT3. In the case of MT3, access of MARVEL components to the AV data streams of the MT3 MEMS microphone and CCTV camera at the Piazzale ex Zuffo location was required. According to data privacy regulations, access to the raw AV data feeds (microphone and CCTV camera) that are part of the MT network is restricted only to authorised personnel and to trusted devices within the network. Therefore, it was not possible to provide MARVEL components with direct access to the raw AV data, but only to anonymised versions of it. Following deliberations with the MT authorities and based on an agreement that nominates FBK as a data processor under certain constraints, a solution was reached that involved the use of an external server (MT F1), which is managed by FBK. However, this server (MT F1) could not be attached to the MARVEL Kubernetes cluster, because such an action would violate the conditions and security policies foreseen by the agreement between MT and FBK. Regarding audio, it was decided to deploy AudioAnony (coupled with VAD) directly on the MT E1 Raspberry Pi device in order to provide anonymisation at the Edge, considering that the RPi could provide sufficient computational power for this service. However, since MT E1 was part of the MT network, it could not be attached as a node to the MARVEL Kubernetes cluster. Furthermore, due to the agreement between MT and FBK, MT E1 could only be accessed from MT F1 using VPN. In order to gain access to the anonymised audio from AudioAnony on MT E1 from other endpoints within the MARVEL Kubernetes cluster, an RTSP Proxy F1 service was deployed directly on MT F1 that could re-stream the output audio stream from AudioAnony on MT E1. Regarding video, it was necessary to deploy a VideoAnony service directly on the MT F1 server at the Fog, without it being hosted on the MARVEL Kubernetes cluster. This VideoAnony service could consume the raw AV stream through a VPN tunnel. In addition, a second Fog server was set up at FBK (MT F2), which was allowed to (i) be attached to the MARVEL Kubernetes cluster and (ii) gain access to the RTSP Proxy F1 and VideoAnony services hosted on the first infrastructure node of FBK (MT F1) through an RTSP Proxy F2 service that was not part of the Kubernetes cluster, but could re-stream the output of AudioAnony and VideoAnony towards the Kubernetes cluster using a VPN tunnel. Therefore, using this configuration, it was possible for all components that are hosted within the MARVEL Kubernetes cluster to gain access to anonymised AV data streams produced by AudioAnony at MT E1 and VideoAnony at MT F1. In addition, in order to receive the inference result output of VAD from MT E1, an MQTT Proxy service was deployed on MT F1 to which VAD could publish MQTT messages via VPN, which were restreamed through another VPN tunnel towards the DatAna Fog service, deployed on MT F2 within the Kubernetes cluster.

Based on the infrastructure that was available at the pilot site and at the FBK premises, certain MARVEL components were selected to be deployed at the fog infrastructure node MT F2, within the Kubernetes cluster and through MARVdash. StreamHandler and AV Registry were deployed at the fog layer (MT F2), as this is the recommended layer for their operation according to the reference ‘AI Inference Pipeline’ architecture. A DatAna MQTT and DatAna Nifi services were also deployed at MT F2 to collect and process the output of VAD deployed within the same container as AudioAnony at the MT E1 device (Raspberry Pi).

Each component that needed to access an AV data stream produced by AudioAnony or VideoAnony and relayed by the RTSP Proxy service (MT F2) had to make a REST API request to the AV Registry component at MT F2 to receive the details (e.g., url, AV metadata) of the AV source (AudioAnony or VideoAnony instance) it needed to connect to.

The RTSP protocol was implemented for delivering the AV data streams from AudioAnony and VideoAnony to the respective AI components, StreamHandler and SmartViz.

The AI components deployed at the cloud (AT, SED, AVAD) published their output of raw inference results as messages structured in JSON format to an MQTT broker that also resided at the same cloud infrastructure node.

DatAna agents subscribed to the topics of the MQTT brokers residing on the same layer to receive AI raw inference results. Each DatAna agent transformed the inference results into data models that are compliant with the Smart-Data-Models and relayed them to a DatAna agent at a higher layer (DatAna Fog to DatAna Cloud). Finally, the DatAna Cloud published the inference results collected from all layers to dedicated Kafka topics at the DFB, which was deployed at the cloud layer.

SmartViz was deployed at the cloud layer and could access the inference results that were aggregated at the DFB (both real-time results published at the Kafka topics and historical results stored in the DFB Elastic Search repository). SmartViz was responsible for visualising these results and for transferring information on results that were verified by the user back to the DFB for updating the respective entries in the Elastic Search.

SmartViz could also request AV data segments from StreamHandler (deployed at MT F2) that corresponded to specific inference results selected by the user and receive the corresponding AV data segments to be displayed to the user.

The Data Corpus deployed at the cloud layer could access the inference results and verification information published in the Kafka topics in real-time from DatAna Cloud and SmartViz, respectively, by subscribing to these Kafka topics. In addition, the Data Corpus could access AV data that were produced by StreamHandler after processing the live AV data streams it was receiving.

HDD deployed at the cloud layer could exchange information with the DFB through a REST API to provide recommendations for optimising the DFB Kafka topic configuration.

MARVdash was used to orchestrate the deployment of all components at all infrastructure nodes.

During the partial integration testing and end-to-end integration testing activities, several issues were resolved, including the following:

- appropriate configuration of the infrastructure nodes;
- appropriate deployment configuration of the components, service names and exposed ports;
- exposing services so that they are reachable from other services;
- fine-tuning AV data streams;
- fixing inconsistencies in inference result data.

7.2 Multimodal and privacy aware intelligence for MT3

This section presents the work done to configure the framework according to the needs of the MT3 in terms of consolidation and ingestion of audio-visual data, training and testing of AI models, design of the pipeline data flow from the edge to the cloud and analysis of the output obtained.

7.2.1 Datasets for model training and privacy assurance

This section describes the methods and approaches taken towards collecting the AV data, the datasets required for model training, the analysis of datasets and streams and privacy assurance and anonymisation methods used in the use case.

7.2.1.1. Datasets for model training

MT has provided data for various components. MT has provided the dataset "TrentoOutdoor – real recording" and "TrentoOutdoor – staged recording" (as defined in D2.1) processing the streams of cameras and microphones from Piazzale ex Zuffo.

MT, in collaboration with FBK, collected and anonymised more than 184 videos of 3 minutes each, in total 55 GB used for the training of:

- AVAD – MT contributed 52 videos which contain anomaly and not anomaly situations.

Thanks to the staged recording done during M13, MT and FBK have provided 38 audio-videos of 30 seconds, each manually annotated using ELAN (see Section 4.2.1.1. for a comprehensive explanation of the software and the ontology used in the annotation) and used for training of SED and AT.

7.2.1.2. Analysis of datasets

The analysis of the streams and datasets is performed by the AI model providers. With their feedback, more data for training will be added.

7.2.1.3. Privacy assurance and anonymisation

The video collected data are anonymised at FBK premises using a batch version of the VideoAnony tool. Instead, the audio collected data are anonymised by the composite component AudioAnony+VAD on edge devices (Raspberry Pis). This is compliant with the request of the MT DPO.

For the staged recordings, anonymisation is not necessary as all involved subjects signed the informed consent.

7.2.2 Training and testing of the models

The MT3 use case employs three AI models, namely Audio-Visual Anomaly Detection (AVAD), Sound Event Detection (SED), and Audio Tagging (AT). Detailed description of the methodologies proposed by MARVEL partners defining the underlying models of these AI functionalities can be found in D3.1. In the following, a description of these models and their training in data of the use case are provided.

7.2.2.1. Multimodal AI realisation

- **Audio-Visual Anomaly Detection (AVAD)** – The MT3 use case employs an Audio-Visual Anomaly Detection (AVAD) model to detect previously unseen, novel events in the audio-visual live feed. Audio-Visual Anomaly Detection is the task of identifying novel situations in a scene, based on the audio-visual information captured in an input video or image. Such models are trained to learn and replicate situations considered normal in a certain setting, e.g., pedestrians on a pathway. Situations that do not occur often, for example, a car on the pathway, are not present in the data used to train the model to identify as normal and, thus, they will be detected as anomalies. The use of enriched audio-visual information is expected to lead to increased performance in cases where anomalies are caused by factors which are not visible in the image, but they can be detected based on the available audio information. Information related to the underlying machine learning model developed by MARVEL partners for detecting anomalies based on visual information can be found in D3.1. An anomaly detection model based on audio-visual information will be developed in the second part of the project, and it will be trained and tested on MARVEL data from MT3.
- **Sound Event Detection (SED)** – The MT3 use case also utilises a Sound Event Detection (SED) component to detect human actions in parking places based on the audio modality in the live audio-visual feed. The aim of the task is to recognise what sound class is active and when it is active within the analysed audio signal, and the input to the component is a continuous audio signal and the output contains sound event labels along with the start and stop timestamps of the sound events. The sound event classes focused on in the component development are events related to dangerous situations such as alarms, horns, exploding bombs, engine accelerating, gun shooting, people running, people shouting, and tyres skidding. The implemented SED component provides start and stop timestamps with a 1-second time resolution. The component is trained and tested with manually annotated material where the start and stop timestamps of the sound events are indicated. The material was collected in staged recording sessions with the same setup in the same location as in the use case. Information related to the underlying machine learning models developed by MARVEL partners for sound event detection can be found in D3.1.
- **Audio Tagging (AT)** – Finally, the MT3 use case utilises an Audio Tagging (AT) component to recognise anomalous scenes in a parking place based on the audio modality in the live audio-visual feed. The aim of the task is to assign predefined tag classes for a segment of audio. The classes focused on component development are argument fight, bad parking, car stealing, loud noises, and normal activity. The neural network architecture used in the sound event detection component is modified for the audio tagging task. The component is trained and tested with manually annotated data that was collected in staged recording sessions with the same setup in the same location as in the use case.

7.3 Analysis of the outputs from all real-life smart city experiments in MT3

Table 39 summarises the characteristics or parameters against which the use case will be evaluated. Most of these characteristics are directly dependent on the non-functional and functional use case and asset KPIs as listed in D1.2, tables 6.2-6.3 and 6.5-6.11. Evaluation will be performed later and reported in D6.2.

Table 39: Parameters to determine for use case MT3

Use case	Parameter	How to measure	target to be achieved
MT3: Monitoring of parking places	Efficiency *Related to the efficiency of the system as used in the use case*	Efficiency is largely dependent on how long it takes to detect and flag dangerous events. The average time taken to detect and flag the event will be measured.	On average less time taken is 5min since onset of dangerous event.
	Operability *Related to the ability of the components to keep functioning together*	Install additional cameras and microphones.	System discovers sensors or adding sensors does not disrupt operation.
	Usability *Related the how well the system helps the users to achieve a task in a given use case*	Interview MT staff and local police to measure the end-user experience via periodic surveys.	End-user finds the system easy to use.
	Robustness *Related to how robust are the system components during the period of operation*	Scale system to other cameras and microphones.	Performance sustained as additional sensors are added
	Performance *Related to how well the system performs the intended task*	Compute accuracy in the detection of targeted events in parking spaces.	50% of dangerous events are correctly noted
	Accountability *Related to the system being able to explain results or decisions.*	System stores video snippet of targeted event.	Number of times system fails to store video snippet.
	Transparency *Related to the description of the processes or algorithms that are used to generate system output*	Decision processes are described in a document.	Document availability.
	Privacy awareness *Related to the provision of adequate governance mechanisms that ensure privacy in the use of data.*	Count data breach reports	Minimisation of count.

7.4 Demonstration

7.4.1 The Decision-making Toolkit

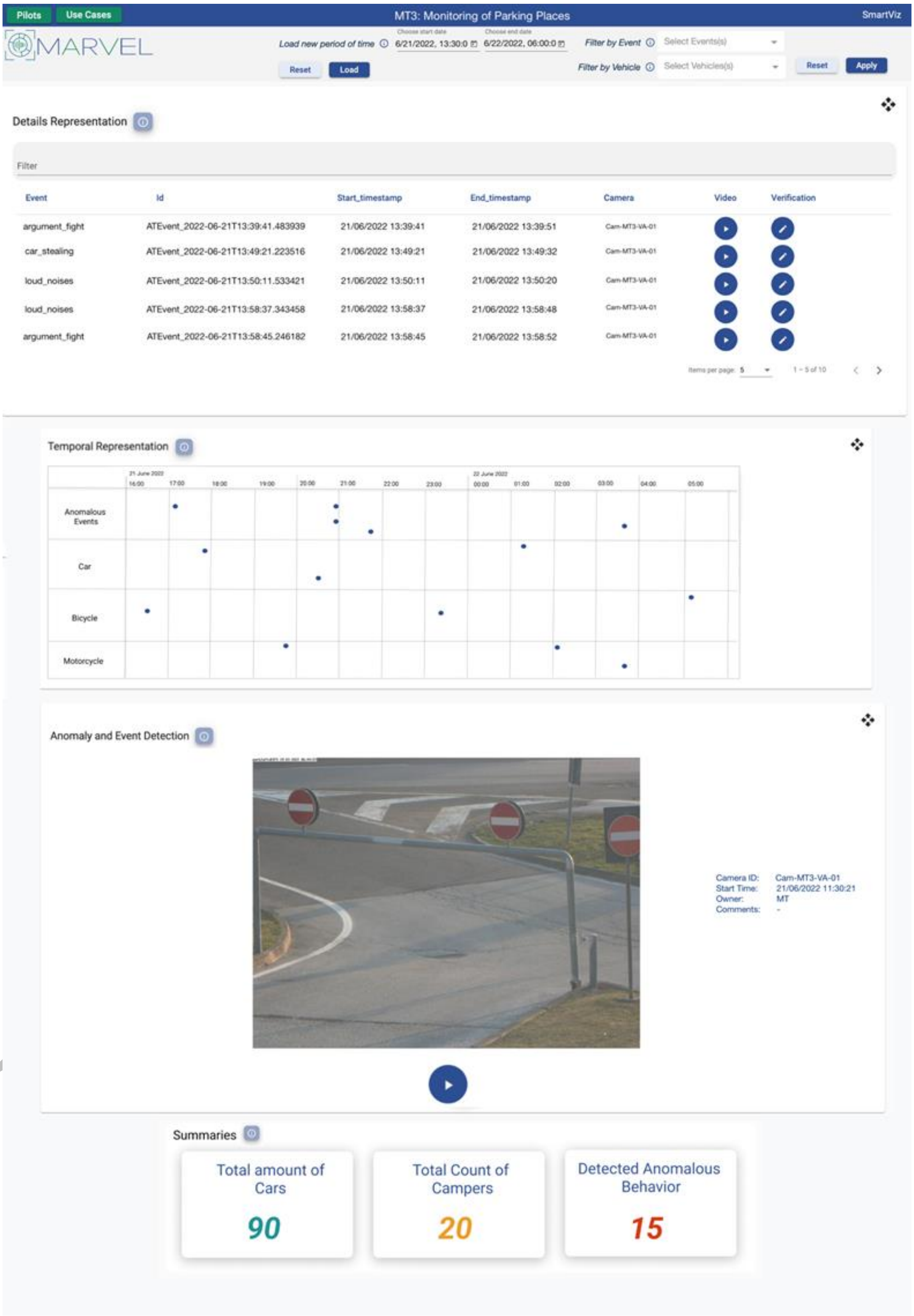
The target of this use case is the so-called “Ex Zuffo” Parking Area which is one of the largest parking lots in Trento (around 1000 parking spaces). It is used by the citizens and works well as an interchange car park, e.g., to leave the car and reach the city centre by public transportation, rentable bikes, and e-scooters.

To prevent robberies or damages to the cars parked, MARVEL framework will support prevention activities with the audio-visual analysis of the existing cameras and the microphones that can be installed thanks to the MARVEL project. Furthermore, anomalous behaviours will be examined, such as the correct use of parking spaces reserved for taxis, the occupation of spaces reserved for the vehicles of disabled people, the number of parked campers and their time of stay, the average parking time of vehicles, the detection of possible damage and other occurrences that will emerge during the execution of the experimentation. The audio-visual analysis must be carried out in real-time and on recording data saved on the servers of the Local Police. The users of this system are intended to be Local Police officers. The User Stories for this use case are shown in Table 40 and give a better understanding of the functionality expected.

Table 40: The user stories for the MT3 – Monitoring of parking places

Title	Main user stories	Sub user stories and user intent
Monitoring of parking places, in terms of occupancy, the correct usage of the parking lots and anomalous events.	**As a** Local Police officer, **I want to** monitor the usage of the parking lot and detect anomaly activity in parking spaces, **so that** I can take corrective /supporting actions.	**As a Local Police officer**, • in order to properly monitor the parking area, I want to view: ○ the timeline distribution and duration of parking activity ○ the total number of vehicles ○ the clustering of vehicles and/or events ○ the severity and type of anomalies observed. • once an anomaly is flagged, I am alerted and notified which live feed from which camera should be checked with attention and I should be able to access the feed a few minutes before the anomaly happened, to accurately assess the event. • once I review the feed of the flagged anomaly, I have the option to mark this event as anomalous or not. This option, similar to use case MT1, allows for continuously improving the system.

In Figure 18 depicts the constructed dashboard containing the widgets, filters and features described below for the Monitoring of Parking Places use case is represented.



Details Representation

Filter

Event	Id	Start_timestamp	End_timestamp	Camera	Video	Verification
argument_fight	ATEvent_2022-06-21T13:39:41.483939	21/06/2022 13:39:41	21/06/2022 13:39:51	Cam-MT3-VA-01	▶	✎
car_stealing	ATEvent_2022-06-21T13:49:21.223516	21/06/2022 13:49:21	21/06/2022 13:49:32	Cam-MT3-VA-01	▶	✎
loud_noises	ATEvent_2022-06-21T13:50:11.533421	21/06/2022 13:50:11	21/06/2022 13:50:20	Cam-MT3-VA-01	▶	✎
loud_noises	ATEvent_2022-06-21T13:58:37.343458	21/06/2022 13:58:37	21/06/2022 13:58:48	Cam-MT3-VA-01	▶	✎
argument_fight	ATEvent_2022-06-21T13:58:45.246182	21/06/2022 13:58:45	21/06/2022 13:58:52	Cam-MT3-VA-01	▶	✎

Items per page: 5 1 – 5 of 10

Temporal Representation

Anomaly and Event Detection

Camera ID: Cam-MT3-VA-01
Start Time: 21/06/2022 11:30:21
Owner: MT
Comments: -

Summaries

Total amount of Cars	Total Count of Campers	Detected Anomalous Behavior
90	20	15

Figure 18: DMT MT3 Dashboard

In this particular use case, four visualisation widgets carefully selected by SmartViz pool of widgets, are being utilised. First of all, the inference results and all the available information linked to the detected vehicles and anomalous events by AT and SED are represented in the Details widget where the user can also verify their validity, and in the Temporal Representation widget hence, a temporal analysis of the historical data can be made by the user. The user is also given the opportunity to see either the video stream of the camera, a video corresponding to an event, or a video that occurred in a selected time frame through the Video Player widget. Finally, some information regarding the total amount of each vehicle type detected in a parking lot and the detected anomalies or behaviours are represented in the Summaries widget.

8 UNS1: Drone Experiment

8.1 Integration and deployment of the selected use cases for M18

This section describes the steps taken to integrate the systems such that the UNS1: Drone Experiment could be set up and tested. The following sections describe the architecture and components, the pilot infrastructure and the integration, deployment and testing phase.

8.1.1 Architecture, components and E2F2C streaming data pipeline

The UNS1 realisation of MARVEL architecture is given in Figure 19, while the employed MARVEL components together with their configurations for UNS1 are given in Table 41.

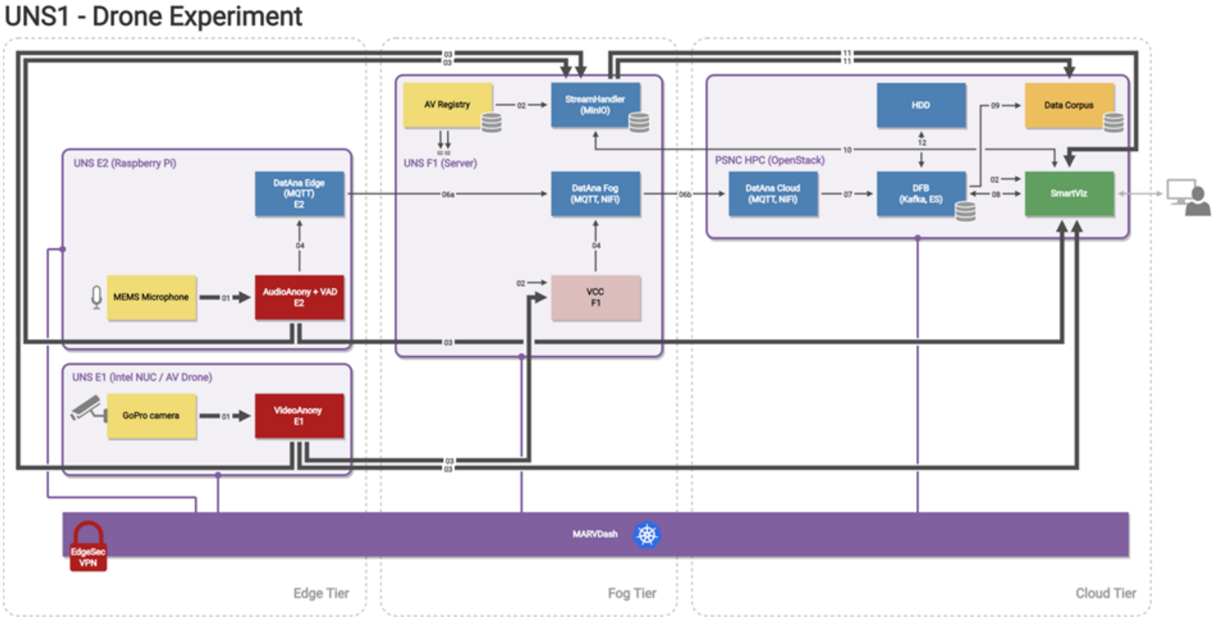


Figure 19: MARVEL R1 deployment and runtime view of the MARVEL architecture for UNS1: Drone Experiment

Table 41: MARVEL components in the UNS1

UNS1:Drone experiment			
Component owner	**Subsystem /Component**	**Comments on how the component is used in UNS1 for R1**	**Deployment location**
Sensing and perception subsystem			Edge and fog
ITML	AV Registry	AV Registry contains metadata information of all AV sources present in UNS1. These include the information on the raw streams produced by the camera and the microphone (fps rate, resolution, etc.), but also on the anonymised streams produced by the VideoAnony and AudioAnony anonymisation components.	UNS fog server (UNS F1)
IFAG	MEMS microphone	In UNS1, one microphone will be used to provide an audio recording of simulated crowds.	Edge microphone
UNS	Drone-mounted camera	UNS1 uses a drone-mounted camera for aerial monitoring of simulated crowds.	Edge camera

Security, privacy, and data protection subsystem			Edge, fog and cloud
FORTH	EdgeSec VPN	In UNS1, EdgeSec VPN creates a secure E2F2C VPN traffic backbone for all communications within the elements of the MARVEL platform by 100% encryption of the traffic.	AVDrone (UNS E1), UNS Raspberry Pi (UNS E2), UNS fog server (UNS F1), cloud (PSNC HPC via OpenStack)
FBK	VideoAnony	VideoAnony will detect individuals that appear in the video footages from the staged recordings at Petrovaradin fortress and anonymise their faces. As UNS1 uses one camera stream, the component is present with one instance.	AVDrone (UNS E1)
AUD	VAD	VAD detects speech segments in an audio stream and outputs the respective onset and offset times.	UNS Raspberry Pi (UNS E2)
FBK	AudioAnony	Based on the onset and offset times of speech segments detected by VAD, AudioAnony will anonymise the respective segments of the audio. As UNS1 uses one audio stream, the component is present with one instance.	UNS Raspberry Pi (UNS E2)
Data management and distribution subsystem			Edge, fog and cloud
ITML	Data Fusion Bus (DFB)	DFB stores inference results of the AI components that participate in UNS1.	Cloud (PSNC HPC via OpenStack)
INTRA	StreamHandler	StreamHandler receives continuously anonymised AV data streams and segments them for temporary storage, for later, on-request visual inspection through SmartViz.	FBK fog WS PC (MT F2-Kubernetes)
ATOS	DatAna Edge, Fog and Cloud	DatAna for UNS1 consists of DatAna Edge, Fog and Cloud agents, each continuously consuming, through MQTT, the results of AI inference components deployed at the corresponding layer (Edge/Fog/Cloud) and sending to the relevant Kafka topics at the DFB. DatAna Edge consists of MQTT only, while DatAna Fog and Cloud implement also the NiFi functionality.	UNS Raspberry Pi (UNS E2), UNS fog server (UNS F1) and cloud (PSNC HPC via OpenStack)
CNR	HDD	For the predefined UNS1 AI inference components, HDD optimises the allocation of their output streams across a given set of DFB Kafka topics.	Cloud (PSNC HPC via OpenStack)
Audio, visual, and multimodal AI subsystem			Fog
AU	Visual Crowd counting (VCC)	In UNS1, VCC provides number of detected people and the corresponding crowd density heatmaps.	UNS fog server (UNS F1)
Optimised E2F2C processing and deployment subsystem			Fog and Cloud
FORTH	MARVdash	MARVdash provides a Kubernetes-based deployment environment of all the UNS1 components.	Cloud (PSNC HPC via OpenStack)
CNR	DynHP	DynHP trains and compresses the VCC model.	Cloud (PSNC HPC via OpenStack)
UNS	FedL	For UNS1, FedL delivers federated training of the VCC model together with GRN41 and MT1 pilots without explicit exchange of the (raw) data. UNS1 implements a VCC-FedL client that, through the	UNS fog server (UNS F1) and Cloud (PSNC HPC via OpenStack)

		FedL server, running at the cloud, enables model exchanges with GRN4 and MT1 FedL clients.	
System outputs: User interactions and the decision-making toolkit			Cloud
ZELUS	SmartViz	SmartViz presents crowd heat maps from drone-mounted camera stream and, upon request, a real-time stream for crowd inspection.	Cloud (PSNC HPC via OpenStack)
STS	Data Corpus-as-a-Service	Data Corpus contains all data collected for UNS1 use cases, used for training of the relevant AI models.	Cloud (PSNC HPC via OpenStack)

AV data acquisition and anonymisation

The inference data flow starts with the AV data acquisition using the drone-mounted camera connected to the Intel NUC within the AVDrone hardware setup (UNS E1) and ground-based microphone attached to the Raspberry Pi (UNS E2). The stream from the camera is transmitted to the Intel NUC (UNS E1) and then subsequently consumed by the VideoAnony component, running also on the Intel NUC. Upon anonymisation, specifically, by blurring faces of the detected people, VideoAnony produces an RTSP stream compiled of anonymised frames to be consumed by several components. In parallel, the microphone on the ground acquires an audio stream that is passed to the Raspberry Pi (UNS E2) and processed by VAD, running on the same device, for detection of speech segments. VAD outputs onset and offset times of the detected speech segments and passes them to AudioAnony to indicate which intervals in the microphone stream require audio anonymisation. AudioAnony anonymises the corresponding audio segments, compiles them together with the remaining audio segments, and finally produces a (single) RTSP stream. The outputs of VAD are submitted to MQTT DatAna brokers, also residing at the Raspberry Pi (UNS E2), to be further utilised in the platform (e.g., for audio content inspection by SmartViz).

Real-time AI inference

The anonymised video stream is next processed by VCC.

- **VCC**. The VCC instance in UNS1 is running at the UNS fog server (UNS F1). The goal of this component is to produce continuous estimates on the number of detected people, together with the accompanying heatmaps indicating crowd/people location densities. Results of VCC are submitted to the appropriate MQTT topic of the DatAna fog MQTT broker, which, after appropriate transformations, sends further up the results.

More details on the application of VCC in UNS1 can be found in Section 8.2.2.

Inference results management and on-the-fly transformations

The data management pipeline for VAD and VCC results is outlined next.

- DatAna edge (UNS E1) receives inference results of VAD (at edge) and sends the transformed results to DatAna fog;
- DatAna fog (UNS F1) receives the transformed VAD results from DatAna edge and forwards them to DatAna Cloud;
- DatAna fog (UNS F1) concurrently receives inference results of VCC (at the fog) and sends the transformed results to DatAna cloud;

- DatAna cloud (PSNC HPC via OpenStack) receives the VAD and VCC results from DatAna fog and sends them to the appropriate Kafka topics of DFB.

Inference results fusion and storage

After being gathered in the appropriate Kafka topics in DFB (PSNC HPC via OpenStack), where one Kafka topic per each AI component is configured and enabled, results are consumed by SmartViz and Data Corpus, as detailed below. Finally, DFB passes also results from Kafka topics to Elasticsearch for persistent storage.

AV data storage

Concurrently with processing the RTSP (anonymised) audio and video streams for AI inference, another component, StreamHandler, is also subscribed to receive these streams. StreamHandler buffers each of the streams for further segmentation that occurs at regular time intervals, and finally stores the respective audio and visual snippets in the MinIO database. The buffering intervals and segmentation parameters are configurable by the end-user. When a user wishes to inspect the AV data in an interval of interest (e.g., upon detection of an anomaly or raised alert by SmartViz), the user will submit a request to StreamHandler with the onset and offset time intervals. StreamHandler then compiles the relevant AV data segments into a single file and sends back to SmartViz the compiled AV data section.

Visualisations and UI interface

The last component of the streaming data pipeline, also serving as the UI of the platform, is SmartViz. This component runs at the cloud (PSNC HPC via OpenStack). Several functionalities are supported: 1) advanced visualisations, with a dashboard for user configurations and user interactions, 2) live AV feed for real-time inspection of the monitored areas, 3) on-request inspection of stored AV data, and 4) user-based verification of AI inference results. First, visualisations of AI inference results are provided in the form most suitable to the end user. For UNS1, this consists of a crowd density heat map and the number of detected people displayed on the screen, and related time statistics; further details can be found in Section 8.4. Live AV feed from the microphone and the camera, arriving from the RTSP (anonymised) AudioAnony and VideoAnony streams, is shown upon user request. Also, for inspection of historical AV data or near real-time, e.g., in case an event has been detected by the system and the user was not able to follow the AV stream in real-time, the user may submit a request to the StreamHandler with onset and offset times indicating the time interval of interest. SmartViz receives the respective AV data segment, as described in the above, and displays it for user inspection. Finally, after inspecting an AV data segment, either through live feed or from StreamHandler, the user may verify the relevant inference result. This is done by submitting the verification result to the corresponding Kafka topic in DFB, upon which the relevant entries in the Elasticsearch database are changed.

8.1.2 Pilot E2F2C infrastructure

The infrastructure of the UNS Drone experiment consists of the Edge, Fog, and Cloud tiers. The core elements of the Edge tier are data capturing and computational devices and they are organised within the AVDrone component. The edge tier consists of drone-mounted devices and ground-based devices and there is an infrastructure for a multi-point recording in space. The aim of the drone-mounted devices is to perform video recording from the drone using

GoPro camera, anonymise video and send secure video streams to the Fog tier. Video anonymisation and streaming are enabled by installing VideoAnony and RTSP server on the Intel NUC, respectively. Additional data capturing is performed using the ground-based microphone. For this task, four IFAG AudioHub Nano microphone boards are used. These boards are attached to the RPi v4 boards, which further stream data to the Fog tier. The description of the device at the Edge tier of the UNS pilot are described in Table 42.

IFAG has provided the boards AudioHub Nano and Audiohub - Nano 4 Mic Version as the hardware necessary to collect the audio. Those devices stream mono, stereo, or 4ch audio data. Both use the standard USB Audio protocol, supported by the edge devices planned to be used in this demonstrator (Intel NUCs and Raspberry Pi). No driver installation is required, and recording can be performed by using most audio recording software and libraries. User can choose the desired number of channels to record from and the sampling rate. With the on-board switch the operating mode and gain configuration can be selected (from 0 to 24dB) to better suit the recording scenario. An extended technical explanation of the devices can be found in D4.1, section 2.2.

Table 42: UNS1 devices at the Edge

Device	Specifications
Intel NUC10i5FNH Mini PC	CPU: Intel Core i5-10210U @4.2GHz GPU: Integrated; Hard drive: 1TB M.2 NVMe RAM: 16GB
Raspberry Pi version 4	CPU: Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
DJI Matrice 600 Pro drone	Used for demonstration purposes
DJI P4 Multispectral drone	Used for data collection due to its smaller size and less restrictive flight permissions. It is equipped with integrated HD camera, which can record data using H.264 codec and 30 frames per second.
AudioHub Nano microphone board	Channels: 2 Sampling rate: 48kHz Rate: 24-bit audio data (stereo)
GoPro Hero 8 camera	Resolution: HD, Full HD, 4K Codec: H.264+HEVC, HEVC Frame rate: 24, 25, 30, 60

We note that the real-life demonstration for UNS1 is not feasible due to the experimental nature of the UNS pilot and drone flight restrictions. The video and audio are taken from the staged recordings.

AI models will be executed at the Fog tier. Visual crowd counting is chosen as the model of interest within R1. Besides VCC, FedL-based VCC model training is implemented at the Fog. VCC-FedL client runs on a GPU of the UNS Fog server and it also communicates with the VCC-FedL server that runs at the PSNC-based MARVEL cloud. Specification of the UNS Fog server is provided in Table 43.

Table 43: UNS1 Devices at the Fog

Device	Specifications
SUPERMICRO SYS-7049A Server SuperWorkstation	CPU: 2 x Intel Xeon Silver 4110 2.1 GHz 8C/16T CPU GPU: Nvidia TitanXP Hard drive: 3 x 300GB SSD and 1 x 1TB SSD RAM: 128GB DDR4

Additionally, there is a local network storage device:

- QNAP TVS-682-8G with 4 x 3TB SATA3 disks under RAID protection, with NFS/SMB/S3 protocols enabled.

All infrastructure components are locally connected with multiple redundant 1G Ethernet links. UNS infrastructure is currently hosted "behind" a HTTP(S) proxy for Internet connectivity.

Cloud tier serves for the model training and visualisation, which is supported by using SmartViz. SmartViz enables end-users to monitor large public events and perform a quick check if some unexpected behaviour occurs. Besides SmartViz, DFB and DatAna Cloud are incorporated at the Cloud tier of this use case.

The infrastructure scheme of the UNS1 pilot is presented in Figure 20.

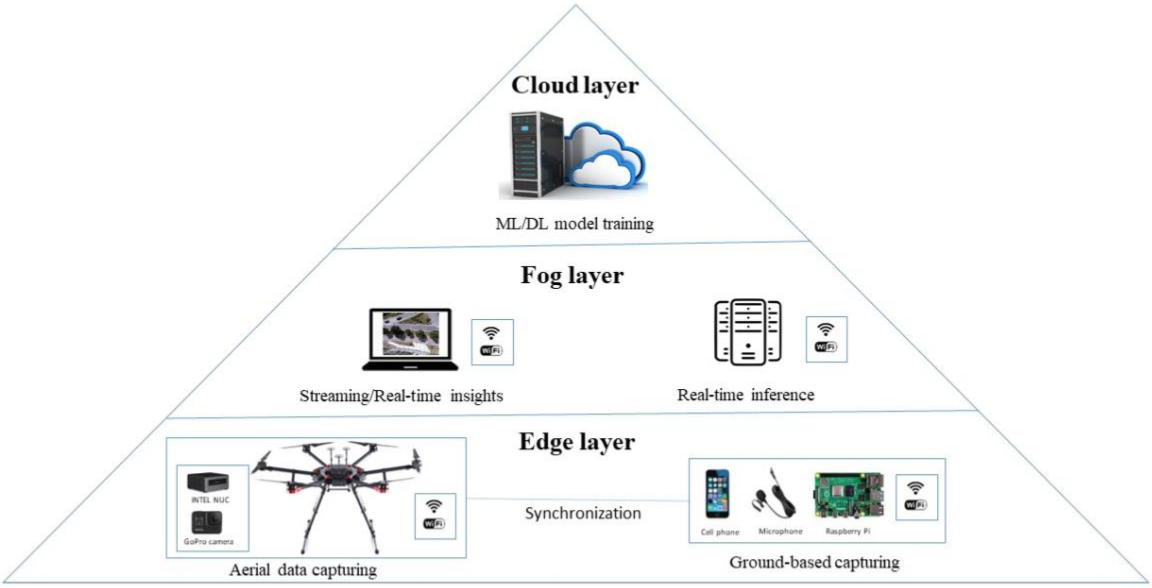


Figure 20: UNS1 pilot infrastructure

8.1.3 Deployment

Once again, MARVdash and, by extension, Kubernetes are present in this use case.

For the UNS1, the framework had to deal with the fact that all the involved computing devices are accessing the internet via proxy. The infrastructure of UNS consists of a server that is located at the fog layer, a Raspberry Pi that is located at the edge layer, and an Intel NUC that is mounted on a drone also located at the edge. The proxy that is used in the UNS infrastructure created unexpected communication issues between the edge nodes and the Super Node that is located at the cloud in PSNC's infrastructure.

The adopted solution was to instantiate a secondary Super Node at the fog layer and configure the edge nodes to connect to this secondary Super Node. The two Super Nodes form a special community, called federation. When a Super Node is part of a federation, it propagates its knowledge about all the edges, to the other Super Nodes in the federation (Figure 21).

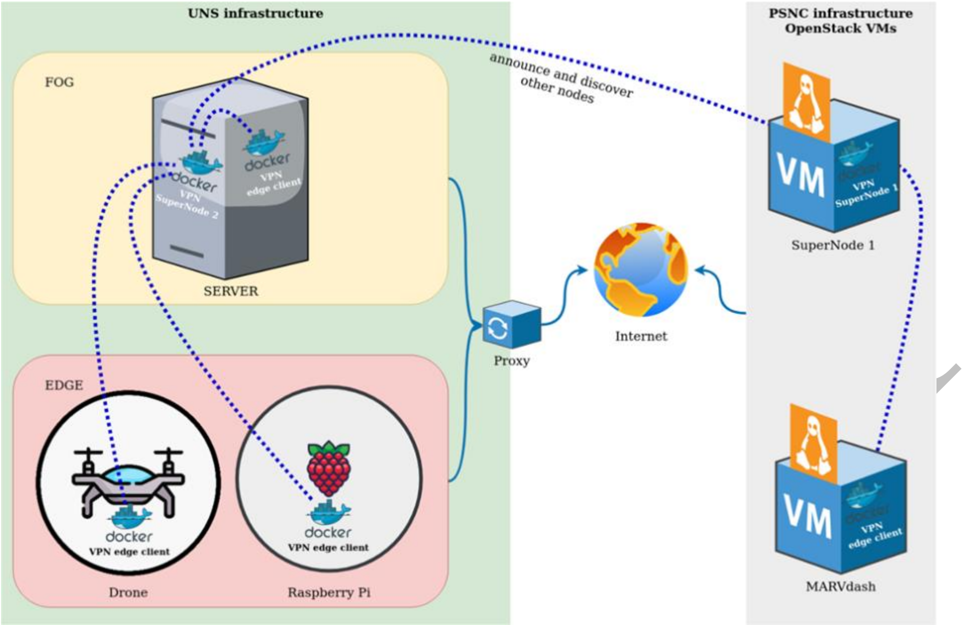


Figure 21: EdgeSec VPN in UNS1 use case

8.1.4 Analysis of data streams

In the case of UNS1, audio is being captured using a sampling frequency equal to 44.1kHz, whereas bit depth is 16 bits per sample. Taking into account that stereo audio is recorded, the bit-rate is 1411 kbps, i.e., ~1.4 Mbps. Audio data is recorded using WAV format.

Video data is being recorded using a camera of resolution 1280x720 pixels. Data capturing is performed using 30 frames per second, H.264 codec is used and data is stored using the MP4 format. The total bit-rate is 30000 kbps, i.e., ~30Mbps. Taking into account a separate audio ground-based stream, previously described, approximately 31.4Mbps is required bit-rate for this use case. Additionally, restricted video streams will be analysed considering lower frame-rates and bit-rates.

8.1.5 Integration and testing

The integration of UNS1 components is based on the reference architecture of the 'AI Inference Pipeline'.

In this use case, the available infrastructure and foreseen AV data sources allowed the anonymisation of AV data to be carried out exclusively at the edge layer, which is considered by MARVEL to be the ideal solution from a data privacy perspective. The AVDrone component by UNS was employed in this use case, which featured a 'DJI Matrice 600 Pro' UAV drone that is able to carry a significant payload (5kg). The AVDrone was also equipped with an Intel NUC Mini-PC that was designated as the UNS E1 edge infrastructure node and a GoPro camera that served as a raw video data capturing sensor. The UNS E1 was able to host a VideoAnony instance for anonymising the AV data stream and was connected to a ground station via a WiFi network. A ground station equipped with a Raspberry Pi device hosting a MEMS microphone was also used for collecting audio data and was designated as UNS E2. This device was also able to host an AudioAnony instance coupled with VAD in the same container for anonymising the raw audio stream from the MEMS microphone. Both UNS E1 and UNS E2 edge infrastructure nodes were incorporated within the MARVEL Kubernetes cluster and managed via the MARVdash component.

In terms of employed AI components, VCC was chosen as most suitable for the needs of the USN1 use case and it was decided to be deployed at a high-performing server that acted as the main infrastructure node at the fog layer (UNS F1). This decision was also affected by the suitability of VCC to be coupled with a Federated Learning approach through the implementation of the FedL component, which is typically applied using a FedL server at the cloud layer and a FedL client at the fog layer. UNS F1 also hosted StreamHandler and AV Registry, as the fog is the recommended layer for their operation according to the reference 'AI Inference Pipeline' architecture. A DatAna MQTT and DatAna Nifi services were also deployed at UNS F1 to collect and process the output of the VCC component. Another DatAna MQTT instance was also deployed at UNS E2 providing an MQTT topic, where VAD (UNS E2) could publish messages with its output. The DatAna Nifi agent at UNS F1 was also subscribed to the MQTT topic of the broker at UNS E2 to collect the output of VAD.

Each component that needed to access an AV data stream produced by AudioAnony or VideoAnony had to make a REST API request to the AV Registry component at UNS F1 to receive the details (e.g., url, AV metadata) of the AV source (AudioAnony or VideoAnony instance) it needed to connect to.

The RTSP protocol was implemented for delivering the AV data streams from AudioAnony and VideoAnony to VCC, StreamHandler and SmartViz.

DatAna agents subscribed to the topics of the MQTT brokers residing on the same layer to receive AI raw inference results. Each DatAna agent transformed the inference results into data models that are compliant with the Smart-Data-Models and relayed them to a DatAna agent at a higher layer (DatAna Fog to DatAna Cloud). Finally, the DatAna Cloud published the inference results collected from all layers to dedicated Kafka topics at the DFB, which was deployed at the cloud layer.

SmartViz was deployed at the cloud layer and could access the inference results that were aggregated at the DFB (both real-time results published at the Kafka topics and historical results stored in the DFB Elastic Search repository). SmartViz was responsible for visualising these results and for transferring information on results that were verified by the user back to the DFB for updating the respective entries in the Elastic Search.

SmartViz could also request AV data segments from StreamHandler (deployed at UNS F1) that corresponded to specific inference results selected by the user and receive the corresponding AV data segments to be displayed to the user.

The Data Corpus deployed at the cloud layer could access the inference results and verification information published in the Kafka topics in real-time from DatAna Cloud and SmartViz, respectively, by subscribing to these Kafka topics. In addition, the Data Corpus could access AV data that were produced by StreamHandler after processing the live AV data streams it was receiving.

HDD deployed at the cloud layer could exchange information with the DFB through a REST API to provide recommendations for optimising the DFB Kafka topic configuration.

MARVdash was used to orchestrate the deployment of all components at all infrastructure nodes.

During the partial integration testing and end-to-end integration testing activities, several issues were resolved, including the following:

- appropriate configuration of the infrastructure nodes;
- appropriate deployment configuration of the components, service names and exposed ports;
- exposing services so that they are reachable from other services;
- fine-tuning AV data streams;
- fixing inconsistencies in inference result data.

8.2 Multimodal and privacy-aware intelligence for UNS1

This section presents the work done to configure the framework according to the needs of the UNS1 in terms of consolidation and ingestion of audio-visual data, training and testing of AI models, design of the pipeline data flow from the edge to the cloud and analysis of the output obtained.

8.2.1 Datasets for model training and privacy assurance

This section describes the methods and approaches taken towards collecting the AV data, the datasets required for model training, the analysis of datasets and streams and privacy assurance and anonymisation methods used in the use case.

8.2.1.1. Datasets for model training

UNS has collected audio-visual data using drone-mounted camera and 4 AudioHub Nano microphone boards. Annotated dataset for VCC model training is fully prepared, whereas a dataset for automatic anomaly detection will be in focus after R1. Data were recorded within the staged recording in Petrovaradin fortress, Novi Sad. For recording purposes, DJI P4 Multispectral Drone was used. Video capturing was performed from different heights varying between 10-20 meters. Videos were recorded using the following configuration: HD resolution (720x1280 pixels), 30fps (frames per second), H.264 codec and MP4 format. Recordings were made in the period between 10h and 15h, and the weather was sunny and cloudless. Three out of four ground-based microphones were attached to fence poles in a line, so that the distance between the first and second microphone, as well as the second and third microphone, was set to 4 meters. The fourth microphone was placed in the centre of the recording scene, 8.5 meters far from the nearest microphone. Bluetooth speaker was used to simulate audio anomalies. Audio anomalies were taken from FSD50K database and they were mixed with a background music using Audacity software.

8.2.1.2. Analysis of datasets

The annotated UNS1 VCC dataset will be used for FedL-based training of the VCC model by the UNS FedL client and possibly also for the general training of the VCC, including DynHP VCC model compression. The dataset has 11.4GB and consists of about 56 minutes of recorded raw video data and 818 annotated frames for VCC training. Twenty-eight video snippets with an average length of two minutes were created. Data were recorded within four 20-minutes-long flights with the drone and non-overlapping snippets are after that extracted due to easier processing. Each flight included different drone positions (i.e., camera position while recording), camera angle and height, so that variations were made while recording scenes, supporting UNS-KPI2. Special attention was paid to generate frames with variations regarding number of people, camera angle, drone (i.e., camera) altitude and people behaviour (calm, running, playing volleyball, standing, sitting and walking). We have generated annotated frames ranging from one person on stage to more than ten people standing or actively moving.

8.2.1.3. Privacy assurance and anonymisation

UNS conducted staged recordings and written consent was acquired from participants. Although this data can be processed without anonymisation, within UNS1 involving both audio and video anonymisation components to align with the requirements of the envisioned end-users.

8.2.2 Training, testing and optimisation of the models

The UNS1 use case employs one AI model, namely Visual Crowd Counting (VCC). Detailed description of the methodologies proposed by MARVEL partners defining the underlying model of these AI functionalities can be found in D3.1. In the following, a description of this model and its training with data from the UNS1 use case is provided.

8.2.2.1. Multimodal AI realisation

Visual Crowd Counting (VCC) – The UNS1 use case employs a Visual Crowd Counting (VCC) model to estimate the number of people present in the visual live feed. Crowd counting is the problem of identifying the total number of people present in a scene, who are observed in a given image of that scene. The input to a crowd counting model is a colour image, and the expected output is a number representing the total number of people present in that image. Optionally, the output of a crowd counting model can be a density map which specifies the density of the crowd for each pixel of the input image, and the total count can be calculated by summing up all the density values for all pixels. Information related to the underlying machine learning models developed by MARVEL partners for visual and audio-visual crowd counting can be found in D3.1. For the UNS1 use case, the audio/visual crowd counting functionality has not been yet tested on a subset of the MARVEL data. The corresponding model receiving as input audio/visual data has not been trained using this data yet.

8.2.2.2. Federated learning realisation

As explained in Section 3.3.3, UNS1 also implements a VCC-FedL client that participates in federated training of the VCC model. The UNS1 VCC-FedL client runs on the UNS fog server. The client also communicates with the VCC-FedL server that runs at the PSNC-based MARVEL cloud.

8.2.2.3. Model compression

DynHP – As explained in Section 3.3.2, DynHP is used to compress the VCC model using the training dataset collected by the use case. The training and compression are performed offline and in the cloud. GPUs are needed to complete the process. The compressed model once achieved satisfactory performance, is uploaded to the AI Model Repository.

8.3 Analysis of the outputs from all real-life smart city experiments in UNS1

Table 44 summarises the characteristics or parameters against which the use case will be evaluated. Most of these characteristics are directly dependent on the non-functional and functional use case and asset KPIs as listed in D1.2, tables 6.2-6.3 and 6.5-6.11. Evaluation will be performed later and reported in D6.2.

Table 44: Parameters to determine for use case UNS1

Use case	Parameter	How to measure	target to be achieved
UNS1: Drone Experiment	Efficiency *Related to the efficiency of the system as used in the use case*	Efficiency is largely dependent on how long it takes to detect events. The average time taken to detect and flag the event is measured.	To decrease the time needed to identify an event using audio-visual monitoring as compared to human visual detection (by security crew).
	Operability *Related to the ability of the components to keep functioning together*	Add additional multi-modal input sources.	Different data modalities successfully accommodated in the platform (audio, video, GPS, etc.).
	Usability *Related the how well the system helps the users to achieve a task in a given use case*	Interview security crew of public event organisers to measure the end-user experience via periodic surveys.	End-user finds the system easy to use.
	Robustness *Related to how robust are the system components during the period of operation*	Change operating conditions (namely distance and light intensity) and measure accuracy, latency, and packet loss.	5% improvement on current operating conditions
	Performance *Related to how well the system performs the intended task*	Classification accuracy across multimodal sources.	To increase the average accuracy (5%) for the drone-based audio-visual anomaly detection as compared to base-line (vision only).
	Accountability *Related to the system being able to explain results or decisions.*	System stores video snippet of event detected.	Number of times system fails to store video snippet.
	Transparency *Related to the description of the processes or algorithms that are used to generate system output*	Decision processes are described in a document.	Document availability.
	Privacy awareness *Related to the provision of adequate governance mechanisms that ensure privacy in the use of data.*	Periodically evaluate the secure transmission.	Minimise data breaches.

8.4 Demonstration

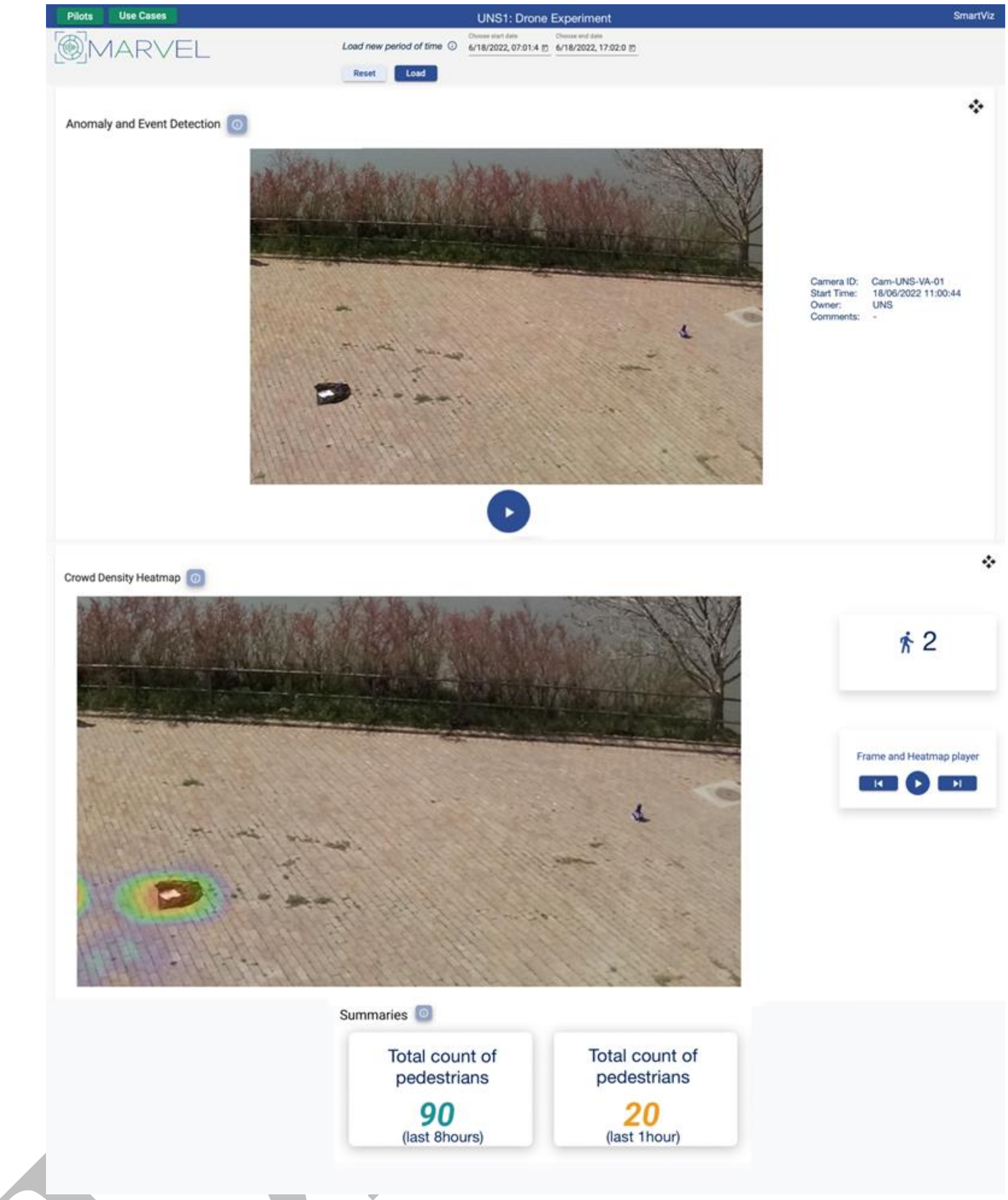
8.4.1 The Decision-making Toolkit

The purpose of the drone experiment is to evaluate the potential of drones for monitoring large open-space public events. The UNS drone experiment setup includes computational resources, video recording from the drone and audio recording from the ground, which serves as a supporting modality for the inference. Such a setup could be useful for a quick security check in the case of crowded spaces. Commonly, street cameras can record only a frontal view of the event and inner details cannot be observed. Furthermore, there are angles or even whole spaces that are not covered using cameras. Flying over the crowd zones of interest, it can be quickly checked if there is some unusual problematic behaviour. The user stories for this use case are shown in Table 45.

Table 45: The user stories for the UNS1 – The drone experiment

Title	Main user stories	Sub user stories and user intent
Crowd counting and anomaly detection	**As an** organiser of a large public event, **I want to**: • have an estimation of the number of people in the event, or a specific part of it (e.g., a stage) • be alerted if there is an overcrowded place and have real-time access to the corresponding AV stream for inspection and determining the corresponding course of action	**As an organiser of a large public event**, being aware of the number of visitors as well as their location at the area of the event but also being informed in real-time if there is an anomalous behavior in a part of the crowds is crucial for alerting people and relevant services. For such tasks, overcrowded places are of special interest due to difficult and slower access to emergency services. As it is difficult to detect anomalous events with ground-based cameras due to the limited visibility, I would like to have an option to have a live stream from drone-based videos, accompanied by audio-visual recordings from the ground. In such a way, decision-making could be accelerated.

In Figure 22, the constructed dashboard containing the widgets, the filters, and the features for the Drone Experiment use case is illustrated.



6/18/2022, 07:01:4

6/18/2022, 17:02:0

Camera ID: Cam-UNS-VA-01
Start Time: 18/06/2022 11:00:44
Owner: UNS
Comments: -

Total count of pedestrians
90
(last 8hours)

Total count of pedestrians
20
(last 1hour)

Figure 22: DMT UNS1 Dashboard

A possible end user of this use case can be an organiser of an event that wants to have an estimation of the number of people in the event or a specific part of it, and furthermore, be alerted if there is an overcrowded place and have real-time access to the corresponding AV stream for inspection and determining the course of action they should follow. The first requirement is met by the Summaries widget, which shows the total amount of visitors, and the detected anomalous events in a selected time frame but is also met by the Crowd Density Heatmap widget that presents the probability of detected pedestrians being present in a video frame. Both visualisation widgets have as input the output analysis of the AVCC component. The latter requirement from the user is met by the Video Player widget, which displays the static video of a detected event or live camera stream.

9 Conclusions

This report presented the initial version of the MARVEL Demonstrators' execution.

In this deliverable, the following achievements are reported (a) the revision of the definition of the experimental protocol considering the outcome of the implementation of the MVP at M12 and any new insights derived from the data collections; (b) a shared definition and consequently the selection and implementation of the aspects that characterise all the use cases selected for Release 1 (Edge-to-Fog-to-Cloud infrastructure, inference pipeline, batch processing pipeline for system optimisation, implemented process to guarantee ethics and privacy and to anonymise the collected data, creation of the Data Corpus and the integration process); (c) the finalisation of the coordinated activities with the aim of implementing the use cases selected for M18 in terms of integration and configuration of the framework, to select the multimodal components and privacy-aware intelligence to be used, analysing the outputs and demonstrating their application; demonstrated the versatility of the MARVEL framework and its components.

The application and integration of the tools made available by the MARVEL framework had to deal with real situations, often not simple, especially due to restrictions related to respect for privacy and the technological components used to collect the data.

The experience gained represents the starting point for the realisation of the remaining five use cases not taken into consideration during the R1 and the improvements of those for which the first version has already been made.

This document, in conclusion, can be used as a basis and reference for planned activities, deliverables and milestones, especially for those related to the final version of the execution of the MARVEL Demonstrator (D6.3 due M30).