Ref. Ares(2023)14044 - 03/01/2023

**Grant Agreement N°:** 952189
**Topic:** ICT-53-2020

Next generation connectivity for enhanced, safe & efficient transport & logistics

# D6.3: Description of enabling functions architecture (Technical)

Revision: v.1.2

| | |
|---|---|
| Work package | WP 6 |
| Task | Task 6.3 |
| Due date | 31/05/2021 |
| Submission date | 03/01/2023 (resubmission rejection EC) |
| Deliverable lead | Be-Mobile |
| Version | 1.2 |

# Abstract

Based on the functional architecture and using feedback and lessons learned from Deliverable 6.2, the technical architecture is developed for each enabling function, including a description of interfaces, secure communication protocols, hardware and software requirements. The development of an architecture for the integrated package of enabling functions will be part of Task 7.1.


**Keywords:** Teleoperation, Enabling Functions

**Document Revision History**

| Version | Date | Description of change | List of contributor(s) |
|---|---|---|---|
| V0.1 | 20/01/2021 | First draft of the document based on the EF summaries | Evert Gellynck (Be-Mobile), Joost Vandenbossche (Be-Mobile), Tom Van de Ven (Locatienet), Maarten van de Pas (Sweco), Martin Kralik (Roboauto), Sander Maas (Sentors), Wim Van den Broeck (R40) |
| V0.2 | 23/04/2021 | Update iTLC cluster, VRU cluster and ETA cluster | Evert Gellynck (Be-Mobile), Joost Vandenbossche (Be-Mobile), Tom Van de Ven (Locatienet), Maarten van de Pas (Sweco) |
| V0.3 | 03/05/2021 | Update executive summary, conclusion and final adjustment of the architecture diagrams | Evert Gellynck (Be-Mobile), Joost Vandenbossche (Be-Mobile) |
| V0.4 | 04/05/2021 | Review and update of iTLC cluster | Maarten van de Pas (Sweco) |
| V0.4 | 12/05/2021 | Internal review | Wim Vandenberghe (MIW), Maria Chiara Campodonico (MAR), Rakshith Kusumakar (V-Tron) |
| V0.5 | 21/05/2021 | Revision (with track changes) | Evert Gellynck (Be-Mobile) |
| V0.6 | 26/05/2021 | Revision (with track changes) | Sander Maas (Sentors), Martin Kralik (Roboauto), Ahmed Ahmed (IMEC) |
| V1.0 | 28/05/2021 | Final version ready for submission | Johann Marquez Barja (imec)<br>Wim Vandenberghe (MIW) |
| V1.1 | 24/12/2021 | Integrated improvements based on general project review after M10 | Joost Vandenbossche (Be-Mobile), Johann Marquez Barja (imec), Wim Vandenberghe (MIW) |
| V1.2 | 24/12/2022 | Clarifications on the use of the 5G network within the Enabling Functions | Freek van Der Valk (Swarco), Tom van de Ven (Locatienet), Besian Kuka (Room40), Ahmed Ahmed (IMEC), Sander Maas (Santers), Linde Vande Velde, Bart Lannoo (Be-Mobile) |

**Disclaimer**



| Project co-funded by the European Commission under H2020-ICT-2018-20 | | |
|---|---|---|
| Nature of the deliverable: | R | |
| Dissemination Level | | |
| PU | Public, fully open, e.g. web | X |
| CI | Classified, information as referred to in Commission Decision 2001/844/EC | |
| CO | Confidential to 5G-Blueprint project and Commission Services | |

* R: Document, report (excluding the periodic and final reports)

DEM: Demonstrator, pilot, prototype, plan designs

DEC: Websites, patents filing, press & media actions, videos, etc.

OTHER: Software, technical diagram, etc

# EXECUTIVE SUMMARY

This document builds further on the previous deliverable (D6.2. functional architecture of enabling functions) and aims to provide a more technical description of the enabling functions. In the previous deliverable a functional description and clustering was made for the enabling function. With D6.3. we start from the 6 EF clusters (as defined in D6.2.) and we will describe for every cluster the application architecture, together with the interfaces, components, protocols,…

Each chapter in this document handles an EF-cluster and consist of a short description followed by the application architecture and a component definition. The application architecture serves as a blueprint for the EF-cluster and identifies the requirements of the different components and establish communication and coordination mechanism among the partners involved in the EF-cluster. In the component definition, the interfaces, authentication and the requirements are defined. This chapter also addresses interoperability possibilities or problems if present.

In the final chapter of the document the 6 application architectures are merged into one final application architecture in order to have an overview of the major components, their relationships and how they interact with each other.

# TABLE OF CONTENTS



# LIST OF FIGURES

# LIST OF TABLES

# ABBREVIATIONS

| | |
|---|---|
| **ACA** | Active Collision Avoidance |
| **AEB** | Advanced Emergency Breaking |
| **API** | Application programming Interface |
| **ASN 1** | Abstract Syntax Notation One |
| **BIC** | Bureau of International Containers |
| **CACC** | Cooperative Adaptive Cruise Control |
| **CAM** | cooperative awareness messages |
| **CAN** | Controller Area Network |
| **CAS** | Collision Avoidance System |
| **CCTV** | Closed-Circuit Tele-Vision |
| **CPM** | Collective Perception Message |
| **CRA** | Collision Risk Analyser |
| **C-V2X** | Connected Vehicle to everything |
| **CVG** | CAM / VAM Generator |
| **DENM** | Decentralized Environmental Notification Message |
| **DBW** | Drive by Wire |
| **EBA** | Emergency Breaking |
| **EF** | Enabling Function |
| **EF1 EAD** | Enabling Function 1: Enhanced Awareness Dashboard |
| **EF2 VRU** | Enabling Function 2: Vulnerable Road Users |
| **EF3 iTLC** | Enabling Function 3: Time slot reservation at iTLC |
| **EF4 DP** | Enabling Function 4: Distributed Perception |
| **EF5 ACA** | Enabling Function 5: Active Collision Avoidance |
| **EF6 CID** | Enabling Function 6: Container ID recognition |
| **EF7 ETA** | Enabling Function 7: ETA sharing |
| **EF8 SA** | Enabling Function 8: Scene Analytics |
| **ELKS** | Emergency Lane Keeping Device |
| **ETA** | Estimated Time of Arrival |
| **ETSI** | Europees Telecommunicatie en Standaardisatie Instituut |
| **GNSS** | Global Navigation Satellite System |
| **GPS** | Global Positioning System |
| **HDOP** | horizontal dilution of precision |
| **HMI** | Human Machine Interface |
| **HUD** | Heads-Up Display |
| **ISY** | I Saw You message |

| | |
|---|---|
| **iTLC** | intelligent Traffic Light Controller |
| **JSON** | JavaScript Object Notation |
| **MQTT** | Message Queuing Telemetry Transport |
| **PKI** | Public Key Infrastructure |
| **REST** | Representational State Transfer |
| **RPC** | remote procedure call |
| **RIS** | Roadside ITS Station |
| **RTSP** | Real Time Streaming Protocol |
| **SGD** | Safety Gateway Device |
| **SPAT** | Signal Phase and Timing |
| **SRTI** | Safety-Related Traffic Information |
| **TAS** | Truck Assignment System |
| **TCP/IP** | Transmission Control Protocol/Internet Protocol |
| **TOV** | Tele-Operated Vehicle |
| **TLC** | Traffic Light Controller |
| **TTR/TTG** | Time to Red/Time to Green |
| **UIC** | Union Internationale des Chemins de fer (International Union of Railways) |
| **UUID** | universally unique identifier |
| **V2I** | Communication from Vehicle to Infrastructure |
| **V2X** | Communication from Vehicle to Everything |
| **VAM** | Vulnerable Road User Awareness Message |
| **VRU** | Vulnerable Road User |
| **WGS 84** | World Geodetic System 1984 |
| **XML** | Extensible Markup Language |

# 1 INTRODUCTION

## 1.1 Overall Functional Architecture EFs

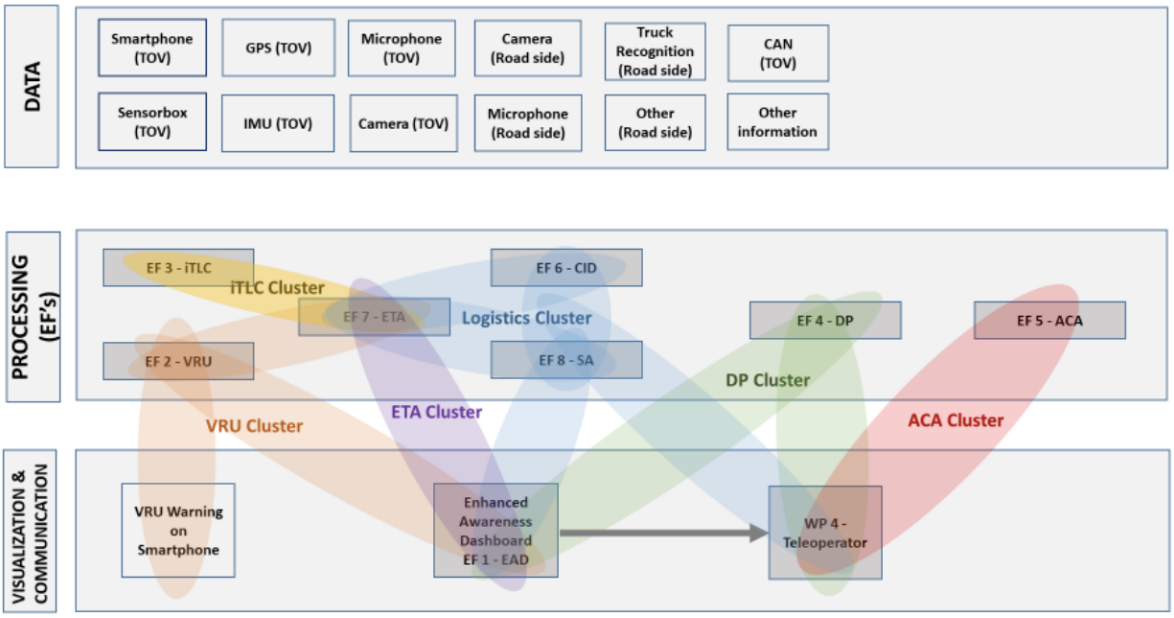



*Figure 1: Functional architecture EF-clusters*

The overall functional architecture (D 6.2.) of the EF's is taken as start for designing and describing the technical application architectures for every EF cluster. As already fully explained in D6.2., the functional architecture of the EF's consists of three layers. The first and top layer is the **data ingestion layer**. The processing from raw data into useful information will be done in the second layer, **the processing layer** by the enabling functions. The **visualization and communication** to end-users (teleoperators, VRU's), other work packages and other interested parties will be done in the last layer. These three layers will be found again in the application architecture, together with following clusters:

- **VRU cluster** (EF2, EF1, and EF7): This cluster is aimed at warning VRUs and teleoperators for potential VRU/TOV conflicts. Part of this cluster also involves communication back to the VRU via smartphone;
- **iTLC cluster** (EF3 and EF7): This cluster aims to provide an appropriate time slot to the tele-operated transport;
- **DP cluster** (EF4 and EF1): This cluster is aimed at providing an extended perceptive range to the remote teleoperator for making the appropriate decisions.
- **ACA cluster** (EF5): This cluster is aimed at providing active collision avoidance tools to the teleoperated transport.
- **Logistics cluster** (EF6, EF7 and EF8): This cluster is aimed at increasing efficiency of tele-operated transport.
- **ETA cluster** (EF7 and EF1): This cluster is aimed at providing detailed ETA estimates to the teleoperator and other interested parties. Inputs from other clusters will be used within this cluster; similarly, the output from this cluster will also be important for other clusters. This explains the central position of EF7 in the architecture scheme.

# 2 TECHNICAL ARCHITECTURE AT EF - CLUSTER LEVEL

## 2.1 ETA - Cluster (EF 7 and EF 1)

### 2.1.1 Short description

This cluster is aimed at providing detailed ETA estimates to the teleoperator and other interested parties. Be-Mobile (as EF 7 ETA provider) will collect speed, heading and the position from the onboard GPS unit (WP4) in order to calculate the ETA of the TOV. The ETA of the teleoperated transport will be calculated on a continuous real-time basis and shared with the teleoperator (via the EF1 EAD dashboard, showing also the route and turn-by-turn navigation) and other interested parties (for example a Driver app).

This ETA will be based on the fastest route from the current position, speed and heading of the TOV to the point of destination. Furthermore, dynamic information from other EF's will also be taken into account. The route of the TOV will also be divided into waypoints. For each waypoint an ETA will be calculated. These intermediate ETAs will be used by EF2 VRU to calculate more accurate potential collision or by EF3 iTLC to assign a time slot.

Information taken into account for calculating the ETA:

- The standard SRTI feed (Be-Mobile)
- Time to red/green (Be-Mobile)
- Warnings picked up from extended perceptive range (EF4 DP) and the continuous monitoring of the TOV and its environment (EF8 SA).
- Warnings on path conflicts with VRUs (coming from EF2 VRU):
- Time slot reservation (coming from EF3 iTLC):
- If available: time to green/time to red from TLCs

The output of EF 7 will be collected by Be-Mobile (as EF 1 EAD provider) to visualise it in the different user interfaces for the teleoperator or driver.

EF 1 EAD will create an **"enhanced awareness dashboard"** on which three types of information will be displayed:

- Speed advice.
- Warnings.
- Navigation and routing features.

The consolidated information may be displayed in three ways:

- **Heads-up-display (HUD):** This is a small widget displayed directly in the line of sight of the teleoperator. This HUD will present only key operational information, that is: advised speed and current speed (based on GPS-speed); textual, symbolic warnings; and turn-by-turn navigation with ETA.
- **Dynamic Map:** On a primary screen, a real-time dynamic map with information on the route, long-distance obstacles, VRU's distribution perception and logistic optimization will be shown.
- **Driver app:** a web based interface will be made so the Safety driver in the TOV will have his own dynamic map and dashboard.

## 2.1.2 Architecture diagram

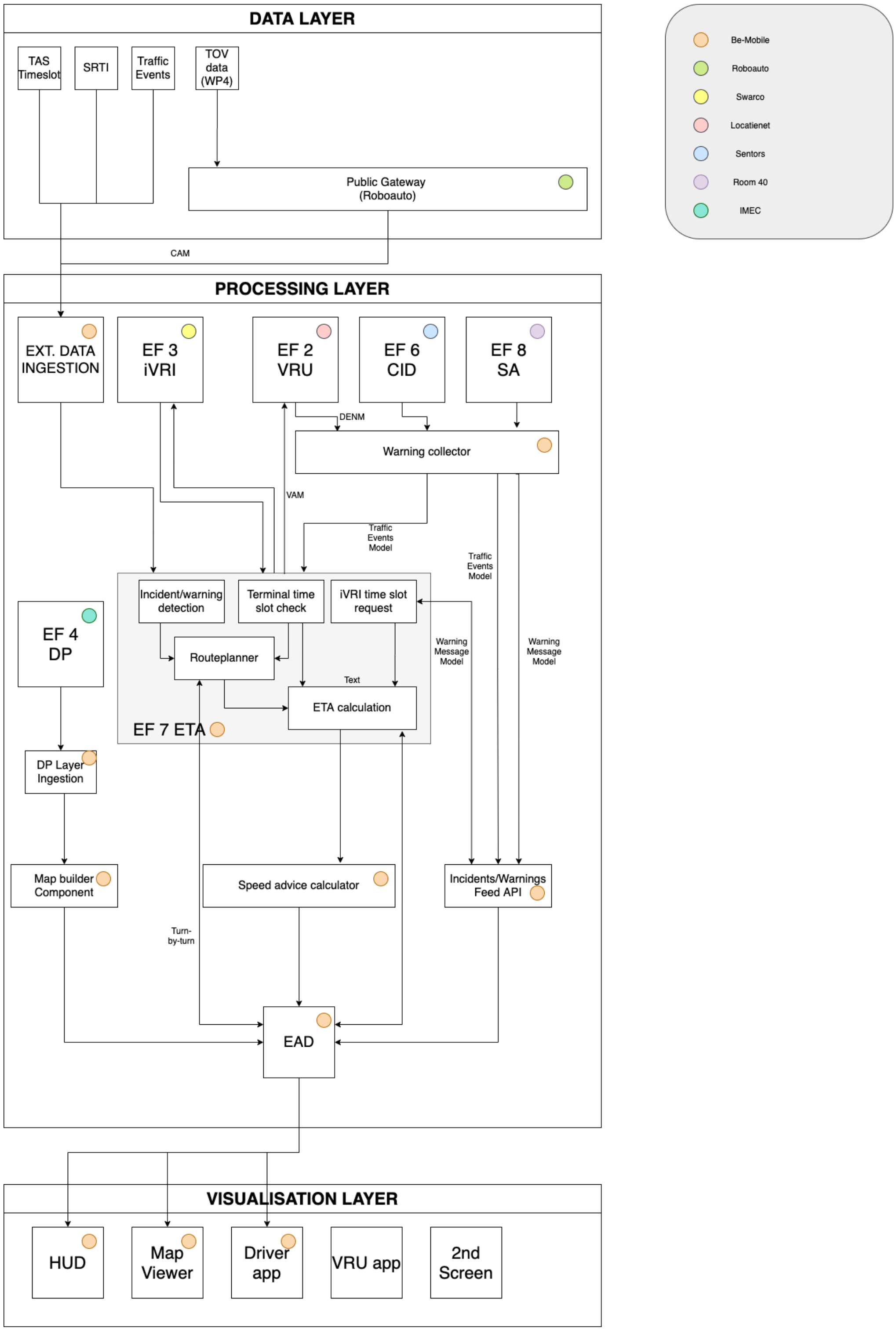



*Figure 2: Application architecture ETA-Cluster*

### 2.1.3 Component Definition

**General information components**

| Name | Description | Input metadata description & Format | Output metadata description & Format |
|---|---|---|---|
| Ext. data ingestion | Component that aggregates incoming datastreams. These streams can both be internal or external. For external feeds, this component will also be responsible for converting data to internal Be-Mobile dataformats. | Traffic Events coming from Be-Mobile (using internal traffic events model). Connection will be made via rest API.<br><br>CAM messages coming from TOV (Connection made through MQTT broker of Roboauto)<br><br>TAS destination + timeslot | Traffic events will be forwarded<br>- Payload: Traffic Events V2 (Be-Mobile model)<br>- Encoding: JSON<br>- Protocol: Server-sent-events/websocket<br><br>CAMs will be converted into internal Be-Mobile format that then will be used for live positions<br>- Payload: GNSS positions containing<br>  ○ Lat/lon (WGS84)<br>  ○ Heading (Degrees)<br>  ○ Speed (km/h)<br>  ○ Confidence (HDOP)<br>  ○ Position sampling timestamp (RFC3339)<br>  ○ Unique Vehicle ID (string/UUIDV5)<br>  ○ Convoy ID<br>- Encoding: flatbuffer<br>- Protocol: kafka<br>- Message pattern: Pub-Sub<br>Tas destination + timeslot will be forwarded to routeplanner. |
| Warning collector | Collection of adapters that will transform incoming feeds into internal Be-Mobile formats. Two different formats will be used : the Be-Mobile internal traffic events format and a new format : 'warningsmessage'. The latter can be used to send messages including hyperlinks to images or videos to the front-end. | Input from EF2: Denm messages from VRU.<br><br>Input from EF6: json message with hyperlink to images.<br><br>Input from EF8:<br>- json message with relevant info that can be used for our traffic events. This will be converted to traffic events model of Be-Mobile<br>- link to a video file in a standard format containing a recording of the triggering event. | Traffic events model v2<br>- Payload: Traffic Events V2 (Be-Mobile model)<br>- Encoding: JSON<br><br>Warnings message<br>- Payload: Warnings message format (Be-Mobile model)<br>- Encoding: JSON |
| Incident/warnings API | Two REST API's that will forward the warnings from the warning collector to the EAD. There is an API for each message type (traffic event and warning messages) | Output of Warning collector<br>- Traffic Events V2<br>- Warning messages | Traffic events will be forwarded<br>- Payload: Traffic Events V2 (Be-Mobile model)<br>- Encoding: JSON<br>- Protocol: Server-sent-events/websocket<br>Warning messages will be forwarded<br>- Payload: Warning message format (Be-Mobile model)<br>- Encoding: JSON |

| | | | |
|---|---|---|---|
| | | | - Protocol: Server-sent-events/websocket |
| ETA Calculator | The central component of EF7, which calculates the ETA over the entire route as well as to individual waypoints along the route such as intersections.<br><br>Using the determined ETA for arriving at the intersection, the ETA calculator sends a time slot reservation request for the intersection. The resulting time slot reservation contain a different time than the originally request time slot.<br><br>The timeslot is used to provide a speed advice which aims at letting the TOV pass the intersection during the provided timeslot. | The ETA calculator receives GNSS information from the public gateway (WP4). Data format:<br>- Requirements from our side:<br>○ Accuracy of <= 2m (negotiable)<br>○ frequency of minimal 1 Hz (hard constraint)<br>○ unique vehicle ID (UUIDV5)<br>○ stationID used in CAM to RIS or predetermined deterministic function to calculate stationID based on vehicle and intersection characteristics<br>- Encoding:<br>○ JSON<br>- Protocol: WS<br><br>The ETA calculator receives intersection topology (MAP) from the EF3 communication broker<br>- Payload: ETSI MAP<br>- Encoding: ASN.1 UPER<br>- Protocol: UDAP<br>- Message pattern: pub-sub | The ETA calculator will send position updates to EF1<br>Format:<br>- Payload: GNSS positions containing<br>○ Lat/lon (WGS84)<br>○ Heading (Degrees)<br>○ Speed (km/h)<br>○ Confidence (HDOP)<br>○ Position sampling timestamp (RFC3339)<br>○ Unique Vehicle ID (string/UUIDV5)<br>- Encoding: JSON (protofbuf/flatbuffer is also possible)<br>- Protocol: MQTT<br>- Message pattern: Pub-Sub<br><br>ETA to VRIs on the route using the MAP topology to detect stop line locations<br>- Payload: ETA (in seconds) to stopline of passing VRI. Using MAP topology<br>- Encoding: Json<br>- Protocol: MQTT/HTTP (REST)<br>- Message Pattern: pub-sub/request-response<br><br>Once within 2 minute mark, send time slot request to broker. Maintain until VRI is passed<br>- Payload: ETSI SRM (No PKI headers)<br>- Encoding: ASN.1 UPER<br>- Protocol: UDAP<br>- Message Pattern: pub-sub<br>- Frequency: 0.1HZ or with 10% ETA difference compared to last request<br><br>The ETA calculator will propagate the Time Slot Reservation to EF1 to calculate speed advice<br>- Payload: raw SSM<br>- Encoding: JSON (can also be Protobuf)<br>- Message Pattern: pub/sub<br>- Frequency:1-to-1 with incoming SSM frequency<br><br>The ETA calculator propagates current signal group statuses and TTG/TTR to for passing intersection approaches EF1 |

| | | | |
|---|---|---|---|
| | | Timeslot reservation responses on timeslot requests (SSM) in ASN.1 UPER encoded messages from the broker<br>- Payload: ETSI SSM<br>- Encoding: ASN.1 UPER<br>- Protocol: UDAP<br>- Message pattern: pub/sub<br><br>The ETA calculator receives SPAT messages indicating TTG/TTR (time to green/red) for traffic lights<br>- Payload: SPAT<br>- Encoding: ASN.1 UPER<br>- Protocol: UDAP<br>- Message pattern: pub/sub | - Payload: Traffic Events V2 (Be-Mobile model)<br>- Encoding: JSON<br>- Protocol: Server-sent-events/websocket<br>- Message pattern: pub-sub |
| Speed advice calculation | Module that will calculate precise speed advice in order to arrive at given waypoints in a desired time window. | From EF3: SSM message that will provide the timeslot at a iVRI<br>From EF7: array of waypoints with ETA<br>From EF7: route itself in standard routeplanner output | Speed advice<br>Warningmessage (timewindow & priority on iVRI) |
| Map Builder component | This module will collect with the DP layer ingestion component the information from IMEC (EF 4 – DP). After the ingestion a layer will be build which will be used for the EAD. | GeoJSON with location of warnings | Be-Mobile traffic Events Model V2 |
| EAD | This component will gather all relevant info that will need to be visualized on the front end. Most of the data gathered by the EAD will be generated by EF7. However, also other EF's will deliver data to the EAD in order to be visualized (EF1 – speed advice, EF4 – additional map views, EF6 – images of the cargo) | Input from Incidents/warningAPI (traffic events format or warningsmessageformat)<br>Inpunt from EF4 : DP (maplayers/geojsons)<br>Input from Speed advise calculation<br>From EF7 : array of waypoints with their ETA<br>From EF7 : json with metadata of the waypoints. Metadata can consist of traffic events model or of static data of iVRIs<br>From EF7 : Extra metadata of iVRI (SPaT – BM)<br>Turn-by-turn route coming from EF7 mapbox format | Route to routeplanner (EF7) + start commando (TBD) – depends on input of TOV-roboauto<br><br>To HUD<br>- Payload : HUDMessage<br>- Encoding: JSON<br>- Protocol : WS<br>To mapviewer :<br>- Payload : with Waypoints and metadata & turn by turn in mapbox format<br>- Protocol : WS |

| | | | |
|---|---|---|---|
| HUD | The HUD will receive information of the AED that will be shown to the front end. The hub shows instructions and metrics (speed, warnings, ..) as opposite to the mapviewer, that will show a maplayer. | Current speed<br>Waypoints array with ETA and metadata<br>Turn-by-turn (ETA)<br>Current speed<br>most important alerts/traffic events | Visuals |
| Map viewer | This component is responsible for the map visualisation of the route and the environment of the route. | Turn by turn in mapbox format<br>Visualisationlayer with waypoints and metadata<br>Background tiles<br>Additional layers processed from the data coming from EF4 | Atlas url with maplayers |
| App | This component is responsible for the mapvisualisation of the route and the environment of te route. | Turn by turn in mapbox format<br>Visualisationlayer with waypoints and metadata<br>Background tiles<br>Additional layers processed from the data coming from EF4 | Slimmed down version of the mapviewer |

*Table 1: Component definition ETA-Cluster*

### Interfaces

| Connection | Interface protocol | Documentation |
|---|---|---|
| TAS, SRTI, Traffic Events – EXT.DATA INGESTION | Traffic Events coming from Be-Mobile (using internal traffic events model). Connection will be made via rest API.<br>CAM messages coming from TOV (Connection made through MQTT broker of Roboauto)<br>TAS destination + timeslot | See annex |
| EXT.DATA INGESTION – EF7 ETA | Be-Mobile Traffic Events model | See annex |
| EF 2 VRU – Warning collector | DENM Messages | https://www.etsi.org/deliver/etsi_en/302600_302699/30263703/01.02.01_30/en_30263703v010201v.pdf |
| EF 6 CID – Warning collector | Custom JSON | / |
| EF 8 SA – Warning collector | Datex 2 or Be-Mobile traffic events model | See annex |
| EF 4 DP – DP layer Ingestion | GeoJSON | / |

| | | |
|---|---|---|
| DP Layer ingestion – Map Builder Component | GeoJSON | Internal Be-Mobile format |
| Map Builder Component – EAD | GeoJSON | Internal Be-Mobile format |
| EAD – HUD | Payload : HUDMessage<br>Encoding: JSON<br>Protocol : WS | Internal Be-Mobile format |
| EAD – Map Viewer | Payload : with Waypoints and metadata & turn by turn in mapbox format<br>Protocol : WS | Internal Be-Mobile format |
| EAD – Driver app | Payload : with Waypoints and metadata & turn by turn in mapbox format<br>Protocol : WS | Internal Be-Mobile format |

*Table 2: Interfaces ETA-Cluster*

**Availability & authentication**

The API's and WS are secured by authorization keys.

**Interoperability possibilities or problems**

Several adapters need to be built for consuming data from different suppliers.

**Hardware and software requirements**

-

**Latency requirements**

The latency of the incoming positions of the TOV will have an impact on the accuracy of the ETA calculations. However, there is no hard requirement on the latency of the positions coming from the TOV.

| Connection | Latency | Reasoning |
|---|---|---|
| TAS, SRTI, Traffic Events – EXT.DATA INGESTION | SRTI, Traffic events: Latency < 2 sec<br>TAS < 2 min | |
| EXT.DATA INGESTION – EF7 ETA | < 1s | |
| EF 2 VRU – Warning collector | < 1s | DENM messages should be obtained with minimal delay |
| EF 6 CID – Warning collector | < 1 min | Receiving BIC codes |
| EF 8 SA – Warning collector | < 5 s | Obstacles/warnings detected from sensors on the vehicle should be received almost real-time with a minimal latency. These warning could potentially indicate a hazardous situation where the teleoperator should be notified immediately. |
| EF 4 DP – DP layer Ingestion | < 1 min | Obstacles detected from other vehicles that are ahead of the TOV, needs to be received fast so a warning can be sent on time. |

| | | |
|---|---|---|
| DP Layer ingestion – Map Builder Component | < 1s | This should be as fast as possible |
| Map Builder Component – EAD | < 1 s | This should be as fast as possible |
| EAD – HUD | < 1 s | Update frequency 1s because we are sending speed advise and real-time speed |
| EAD – Map Viewer | < 1 min | Update frequency will be 1min, so the latency should not be more than 1 min |
| EAD – Driver app | < 1min | Update frequency will be 1min, so the latency should not be more than 1 min |

*Table 3: Latency requirements ETA-Cluster*

### Other

-

## 2.2 VRU cluster – (EF 2, EF1 and EF7)

### 2.2.1 Short description

EF2 has a dual objective:

First, it **calculates the likely path of all vulnerable road users** (VRU) that participate in the service, based on data obtained from these VRUs and the topology of the surroundings. How this works is the following: the speed of the VRU will be calculated based on both the actual and historical data of the device the VRU is using. Based on this speed, an estimation of the possible horizontal movement for the next X (X=10) seconds is obtained. By extrapolation, the expected location can be found. When there is a map available (on the public road for instance) mapmatching will be done to determine in which street and on which line the VRU is and will be moving. The map will be created as a graph of nodes that refer to information about the geometry of the road. This way, the likely path will be calculated by recursive node traversal. The most likely path will take the ratio effective vs possible displacement and the angle between the current heading and the final location into account.

Second, EF2 VRU aims to **provide early warnings** (up to 2 minutes in advance) to VRUs and teleoperators about potential collisions between VRUs and TOVs. The likely paths of the relevant VRUs (the first objective of this function) are compared with the anticipated paths of the relevant TOVs (provided by the EF7 ETA provider, based on position and routing data).

The EF2 VRU provider will also relay these warnings to the teleoperator of the relevant TOV through the EF1 EAD provider by publishing likely path messages to a central location.

EF2 architecture fundamentals:

- In the pilot a single cloud based MQTT broker (MB) is used to exchange messages between all actors.
- As suggested in ETSI TS 123 286 V16.5.0, message formats are compliant with the ETSI ITS standards, specifically on 'Vulnerable Road Users (VRU) awareness' (ETSI TR 103 300). In the field trials 5G V2N connections are used (there is no 5G C-V2X V2V (PC5) yet as this has not been standardised in R16, the release we are using) using a 5G high priority low latency slice. Note that the software design supports both V2N and V2V connections. In a V2V setup the message broker moves from the cloud to the individual devices (geographic message filtering is then disabled)
- Quadkeys and tiles are used to filter and route relevant messages.

General functioning of the information exchange:

- All VRU probes publish CAM messages with 1Hz, over the 5G slice, on the MB
- EF1 publishes VAM messages for all TOVs with 1Hz on the MB, based on live position and speed data, and the planned route of the TOV
- All probes (TOV and VRU) subscribe to the CAM, VAM and DENM topics in their vicinity (using quadkeys)
- If a VRU probe receives CAM or VAM of another probe it starts publishing VAM messages, over the 5G slice, that describe the likely path of the probe
- Each VRU probe determines its collision risk with other probes and publishes these as DENM messages over the 5G slice
- Each probe uses the DENM information to warn the end-user (VRU, TO) and suggest appropriate action to avoid the collision or mitigate the impact.

### 2.2.2 Architecture diagram

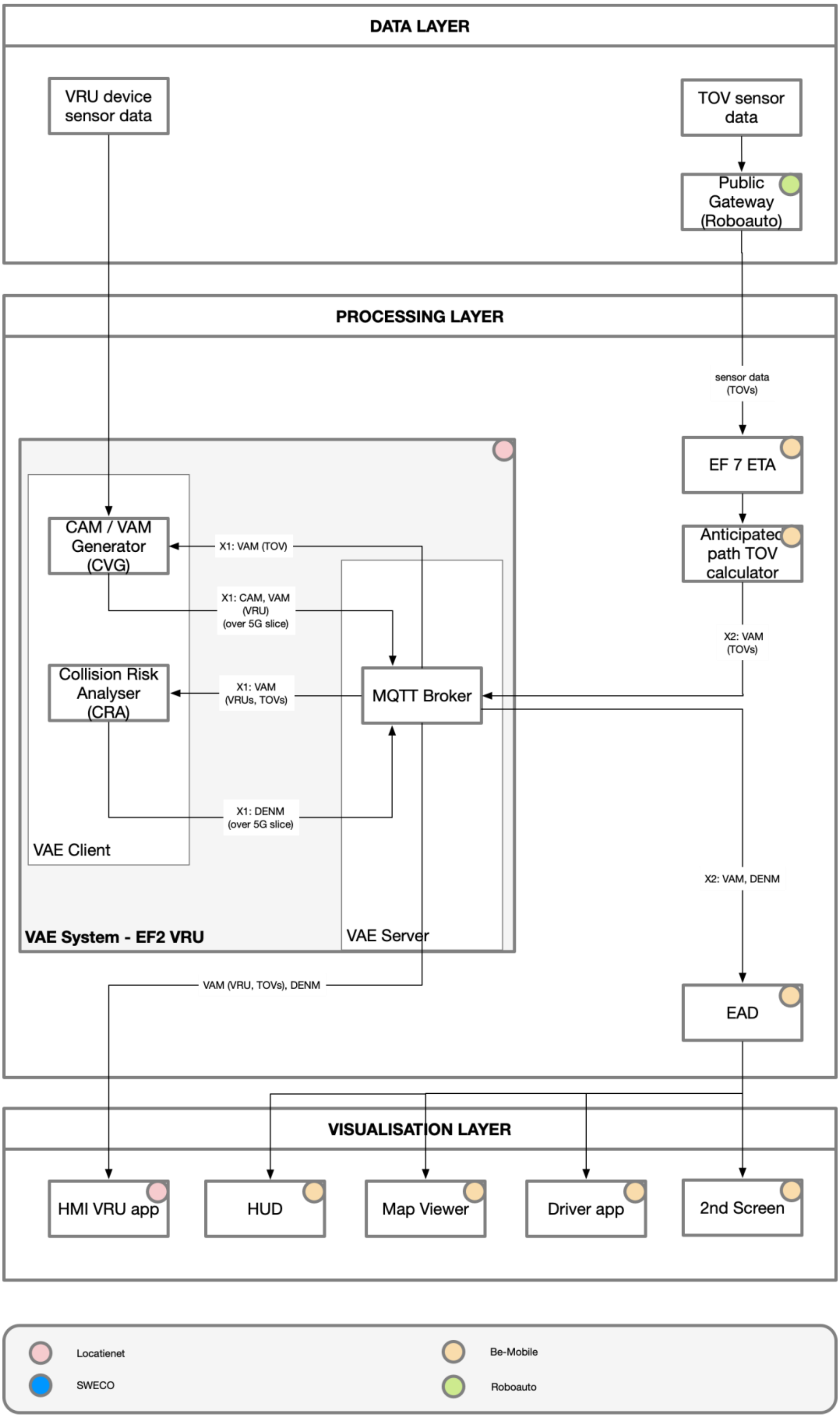



*Figure 3: Application Architecture VRU-Cluster*

## 2.2.3 Component Definition

### General information components

| Name | Description | Input metadata description & Format | Output metadata description & Format |
|---|---|---|---|
| MQTT Broker (MB) | Central communication gateway. Allows publication of real-time data from VRUs and TOVs. Allows nodes to subscribe to information of a certain topic in a specific area. | ETSI ITS messages: CAM, SPAT, VAM, DENM) | ETSI ITS messages: CAM, SPAT, VAM, DENM |
| CAM / VAM Generator (CVG) | Generates CAM messages every second and publishes these on the MB. Subscribes to nearby CAM and VAM of other probes (such as TOVs) on the MQTT Broker, based on the current GNSS position of the device. When another vehicle enters the detection zone it estimates the VRU path, generates and publishes VAM messages to the MB until no other probe is nearby any longer. | ETSI ITS messages: CAM/VAM from TOVs in the area | ETSI ITS messages of the VRU: CAM, VAM |
| Collision Risk Analyser (CRA) | Collects all VAM relevant to the VRU, determines the collision risk and generates a DENM message. | ETSI ITS messages: VAMs of VRU and TOVs. | ETSI ITS messages: DENM<br>(with causeCodecollisionRisk (97), CollisionRiskSubCause Code vulnerableRoadUser (4), position and trace of potential collision) |

*Table 4: Component definition VRU-cluster*

### Interfaces

| Name | Description |
|---|---|
| X1 | Interface between all VRU probes and the MB: CAMs (VRU), VAMs (all probes), DENM (ETSI ITS), over 5G high priority low latency slice. |
| X2 | From Anticipated TOV path calculator of Be-Mobile to MB: TOV VAMs (ETSI ITS), MQTT, over TCP/IP |
| X3 | From MB to EAD of Be-Mobile: VAM describing VRU path, DENM describing collision risks (ETSI ITS), MQTT, over TCP/IP |

*Table 5: Interfaces VRU-cluster*

### Availability & authentication

| Name | Where | How / when to use | Authentication / secure communication protocols |
|---|---|---|---|
| MQTT Broker (MB) | Cloud service | Continuously to exchange real-time information on VRUs, TOVs, and collision risks | Standard internet security (TLS/SSL), MQTT authentication, payload encryption |
| VRU app | Smartphone application and background app service containing CVG and CRA | Continuously when smartphone is active and app has been started. | Access is restricted to authenticated smartphone user |

*Table 6: Authentication components VRU-cluster*

### Interoperability possibilities or problems

/

### Hardware and software requirements

5G handsets for the VRUs.

### Latency requirements

100ms for all interfaces.

### Other

/

## 2.3 iTLC Cluster (EF 3 and EF 7)

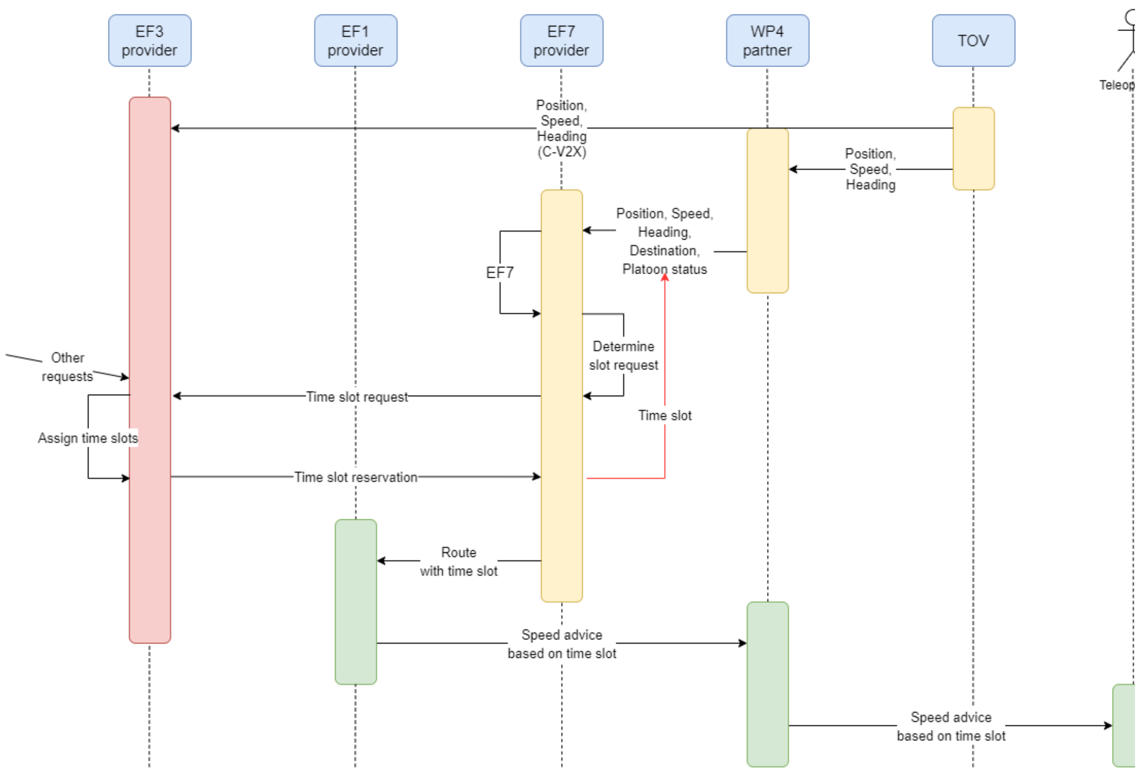
### 2.3.1 Short description

**The iTLC cluster ensures conflictless crossings for teleoperated transport at intersections with intelligent TLC's**. Teleoperators can request a timeslot during which the TOV (and possibly, all TOVs part of the same platoon) is guaranteed a green-lighted passage over the intersection, without there being any possibility that conflicting traffic has green light at the same time. We must strive to keep the whole platoon together. The teleoperator receives a time slot and an advised speed which ensures the time slot can be made. If the assigned time slot can no longer be made by the TOV (or any other TOV in the platoon), a new time slot is provided to the teleoperator. Simultaneously, TOVs will send out standardized CAM messages towards the iTLC's roadside information systems. This system forms a backup mechanism for guaranteeing platoon integrity in case the timeslot reservation system is inoperable. The connection between the iTLC and the RIS is established via a 5G connection

## D6.2 – EF3 DESCRIPTION ERRATUM

D6.2 EF3 describes a secondary channel through which TOVs remain capable of requesting and maintaining a time slot reservation in case of a disruption between EF7 and EF3, by allowing the TOVs to submit their own requests. After further consideration, this approach to a secondary reservation channel was no longer deemed favourable and has since been replaced by the following backup mechanism.

Nearby TOVs will continuously emit floating call data via C-V2X. This data will include platooning information. In the event the time slot reservation channel becomes unavailable, the platoon of TOVs will likely not be able to cross the intersection without deceleration or coming to a complete stop. In this case the platooning information in local floating data will allow the iVRI Controller to let the platoon clear the intersection in its entirety. This prevents unsafe and costly situations where the last vehicle(s) in the platoon might end up stranded at an intersection, or violate red lights.



### 2.3.2 Architecture diagram

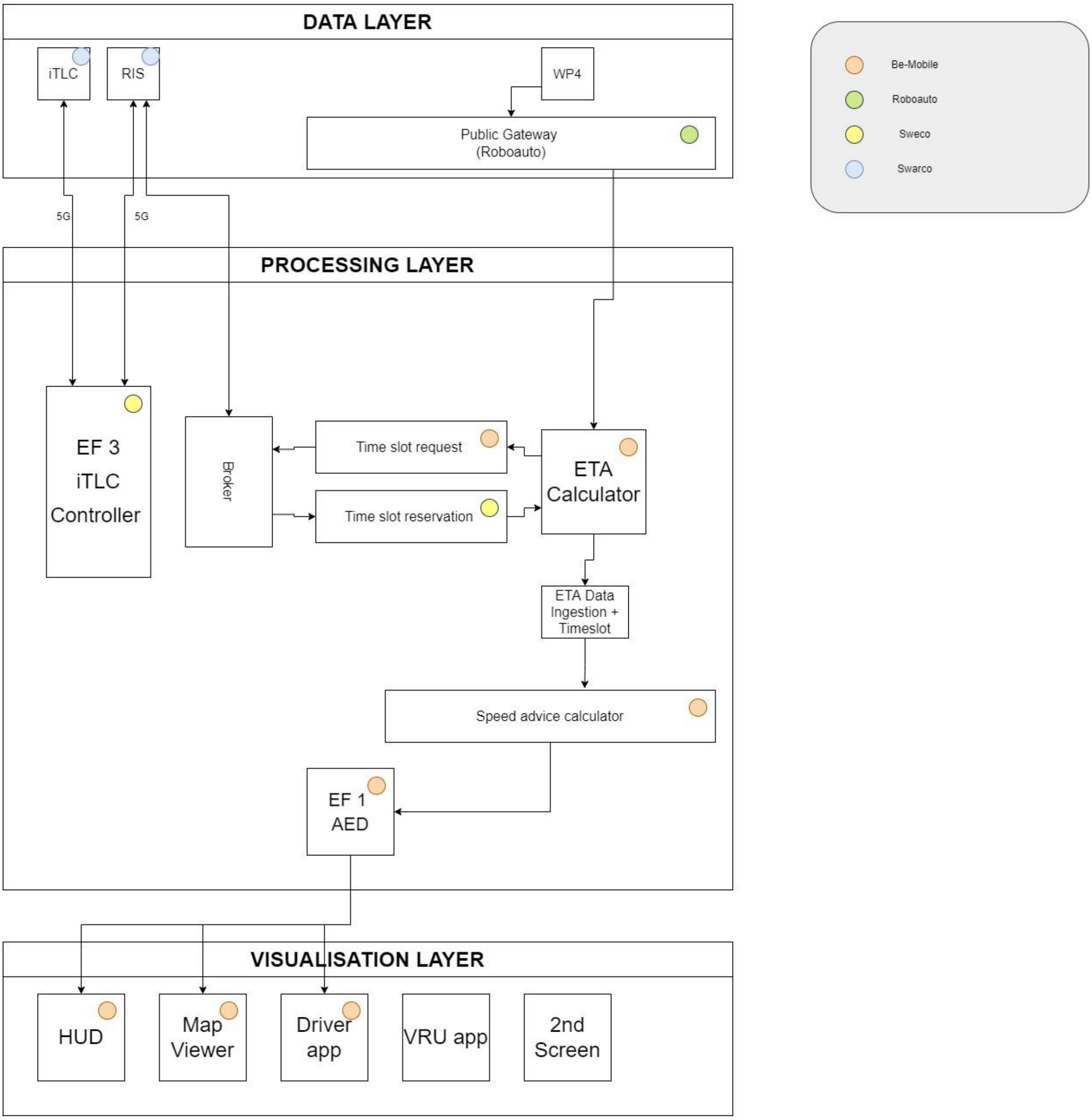



*Figure 4: Application architecture iTLC-cluster*

### 2.3.3 Component Definition

**General information components**

<table>
<tr><th>Name</th><th>Description</th><th>Input metadata description & Format</th><th>Output metadata description & Format</th></tr>
<tr><td rowspan="3">iVRI Controller</td><td rowspan="3">The iVRI controller is the application which schedules and instructs the iVRI to change signals according to internal logic which is fed with iVRI detector information as well as additional data streams received from the RIS in the form of floating car data and priority/time slot reservation requests.<br><br>Besides the controlling the iVRI, it also provides information via the RIS through 5G cellular channels on intersection topology, signal phase timing predictions as well as responses to incoming incoming priority/time slot reservation requests.<br><br>For the enabling function EF3, the controller provides and additional individualized Signal Phase and Timing message (iSPAT) which denote the time slots at the intersection assigned in response to a time slot reservation request.<br><br>The controller may be collocated with the physical iVRI or be hosted in a data centre/cloud.</td><td>iVRI state (change) data<br><br>JSON-RPC objects conforming to TLC-FI specifications.<br><br>These messages include information on state of various aspects of the iVRI such as:<br>• application control state<br>• intersection control state<br>• signal state and predictions<br>• detector state<br>• input state<br>• output state<br><br>Special one off notifications exist to denote:<br>• traditional priority requests<br>• special detector events</td><td>iVRI control requests<br><br>JSON-RPC objects conforming to TLC-FI specifications.<br><br>These are realtime update requests (actuations) to the state of the iVRI. Controllable aspects include:<br>• application control state<br>• intersection control state<br>• signal state and predictions<br>• output state</td></tr>
<tr><td>Floating Car Data<br><br>Received in JSON-RPC according to the RIS-FI standard, the contents map to an extension of the ETSI CAM message standard, containing platooning information.<br><br>This is used to guarantee platoon integrity, even if the time slot reservation service chain is unavailable.</td><td>Priority and Time Slot Reservation Responses<br><br>Sent in JSON-RPC according to the RIS-FI standard, the contents map to an extension of the ETSI SSEM message standard.<br><br>Sent in response to incoming priority and time slot reservation requests, it denotes the status of the request as well as the assigned time slot. It is updated frequently (ranging from several seconds up to a minute depending on the ETA) as currency of the response is vital to properly update the requestor the state of their request as well as providing a correct and actionable speed advice.</td></tr>
<tr><td>Priority and Timeslot Reservation Request messages.<br><br>Received in JSON-RPC according to the RIS-FI standard, the contents map to an extension of the ETSI SRM message standard, containing platooning information.<br><br>This is used to request priority or a time slot at the intersection and forms the basis of the time slot reservation function.</td><td>Signal predictions (SPAT)<br><br>Sent in JSON-RPC according to the RIS-FI standard, the contents map to the ETSI SPAT message standard.<br><br>Sent whenever the timing predictions or signal states change.</td></tr>
</table>

| | | | |
|---|---|---|---|
| iVRI | The iVRI is the physical traffic light controller which reads the input from detectors and controls the signals. It contains a backup application which will control the signals should the iVRI controller lose control due to connection or other issues. | iVRI control requests<br>JSON-RPC objects conforming to TLC-FI specifications.<br>These are realtime update requests (actuations) to the state of the iVRI. Controllable aspects include:<br>• application control state<br>• intersection control state<br>• signal state and predictions<br>• output state | iVRI state (change) data<br>JSON-RPC objects conforming to TLC-FI specifications.<br>These messages include information on state of various aspects of the iVRI such as:<br>• application control state<br>• intersection control state<br>• signal state and predictions<br>• detector state<br>• input state<br>• output state<br>Special one off notifications exist to denote:<br>• traditional priority requests<br>• special detector events |
| RIS | The RIS is a physical roadside component which handles the gateway towards the centralized broker for the iVRI controller. As such, it merges message streams coming from both short range and long range sources.<br><br>Within this function, the stream of individual CAM messages arriving via C-V2X are used to provide backup platoon integrity measures, in case of failure of the reservation mechanism. | Floating car data from broker<br>Extension of ETSI CAM messages in ASN.1 UPER encoding, wrapped with protocol specific headers. | Priority and Time Slot Reservation Requests to iVRI Controller.<br>*See iVRI controller input for details* |
| | | Priority and Time Slot Reservation Request from broker.<br>Extension ETSI SRM messages in ASN.1 UPER encoding, wrapped with protocol specific headers. | Priority and Time Slot Reservation Responses towards broker.<br>*See iVRI controller output for functional details*<br>Extension on the ETSI SSEM messages in ASN.1 UPER encoding, wrapped with protocol specific headers. |
| | | Priority and Time Slot Reseration responses from iVRI Controller.<br>*See iVRI controller output for details.* | |
| | | Time Slot Reservations from iVRI Controller.<br>*See iVRI controller output for details*. | Signal Predictions towards broker .<br>*See iVRI controller output for functional details*<br>ETSI SPAT messages.<br>With ETSI PKI security headers for ASN.1 UPER encoded with UDAP specific headers towards broker. |
| | | Signal predictions (SPAT) from iVRI Controller<br>*See iVRI controller output for details.* | Intersection topology towards broker.<br>*See iVRI controller output for functional details*<br>ETSI MAP messages.<br>With ETSI PKI security headers for ASN.1 UPER encoded with UDAP specific headers towards broker. |

| | | | |
|---|---|---|---|
| Broker | The broker serves as a central access point for all service providers in EF3 and EF7, allowing for a multi-vendor ecosystem. | The broker receives and sends all of the aforementioned messages in ASN.1 UPER encoded format. These messages are wrapped with a protocol specific header to signal that these in fact messages with a meaningful payload. | The broker receives and sends all of the aforementioned messages in ASN.1 UPER encoded format. These messages are wrapped with a protocol specific header to signal that these in fact messages with a meaningful payload. |
| ETA Calculator | The central component of EF7, which calculates the ETA over the entire route as well as to individual waypoints along the route such as intersections.<br><br>Using the determined ETA for arriving at the intersection, the ETA calculator sends a time slot reservation request for the intersection. The resulting time slot reservation contain a different time than the originally request time slot.<br><br>The timeslot is used to provide a speed advice which aims at letting the TOV pass the intersection during the provided timeslot. | The ETA calculator receives GNSS information from the public gateway (WP4). Data format:<br>- Requirements from our side:<br>  ○ Accuracy of <= 2m (negotiable)<br>  ○ frequency of minimal 1 Hz (hard constraint)<br>  ○ unique vehicle ID (UUIDV5)<br>  ○ stationID used in CAM to RIS or predetermined deterministic function to calculate stationID based on vehicle and intersection characteristics<br>- Encoding:<br>  ○ JSON<br>- Protocol: MQTT | The ETA calculator will send position updates to EF1<br>Format:<br>- Payload: GNSS positions containing<br>  ○ Lat/lon (WGS84)<br>  ○ Heading (Degrees)<br>  ○ Speed (km/h)<br>  ○ Confidence (HDOP)<br>  ○ Position sampling timestamp (RFC3339)<br>  ○ Unique Vehicle ID (string/UUIDV5)<br>- Encoding: JSON (protofbuf/flatbuffer is also possible)<br>- Protocol: MQTT<br>- Message pattern: Pub-Sub |
| | | The ETA calculator receives intersection topology (MAP) from the EF3 communication broker<br>- Payload: ETSI MAP<br>- Encoding: ASN.1 UPER<br>- Protocol: UDAP<br>- Message pattern: pub-sub | ETA to VRIs on the route using the MAP topology to detect stop line locations<br>- Payload: ETA (in seconds) to stopline of passing VRI. Using MAP topology<br>- Encoding: Json<br>- Protocol: MQTT/HTTP (REST)<br>- Message Pattern: pub-sub/request-response |
| | | | Once within 2 minute mark, send time slot request to broker. Maintain until VRI is passed<br>- Payload: ETSI SRM (No PKI headers)<br>- Encoding: ASN.1 UPER<br>- Protocol: UDAP<br>- Message Pattern: pub-sub<br>- Frequency: 0.1HZ or with 10% difference compared to last request |

| | | | |
|---|---|---|---|
| | | Timeslot reservation responses on timeslot requests (SSM) in ASN.1 UPER encoded messages from the broker<br>- Payload: ETSI SSM<br>- Encoding: ASN.1 UPER<br>- Protocol: UDAP<br>- Message pattern: pub/sub | The ETA calculator will propagate the Time Slot Reservation to EF1 to calculate speed advice<br>- Payload: raw SSM<br>- Encoding: JSON (can also be Protobuf)<br>- Message Pattern: pub/sub<br>- Frequency:1-to-1 with incoming SSM frequency |
| | | The ETA calculator receives SPAT messages indicating TTG/TTR (time to green/red) for traffic lights<br>- Payload: SPAT<br>- Encoding: ASN.1 UPER<br>- Protocol: UDAP<br>- Message pattern: pub/sub | The ETA calculator propagates current signal group statuses and TTG/TTR to for passing intersection approaches EF1<br>- Payload: Traffic Events V2 (Be-Mobile model)<br>- Encoding: JSON<br>- Protocol: Server-sent-events/websocket<br>- Message pattern: pub-sub |

*Table 7: Component definition iTLC-cluster*

### Interfaces

Interactions between the components in this cluster is mostly done via standardized protocols, some of which were developed in earlier projects such as the Dutch Talking Traffic program.

| Connection | Interface protocol | Documentation |
|---|---|---|
| iVRI – iVRI Controller | TLC-FI | TLC-FI IDD<br>TLC-FI IRS |
| RIS – iVRI Controller | RIS-FI | RIS-FI IDD<br>RIS-FI IRS |
| RIS – Broker | UDAP/TLEX Protocol using ETSI ITS Messages plus an extension for Time Slot Reservations | UDAP/TLEX [1], [2], [3], [4] |
| ETA Calculator – Broker | UDAP/TLEX Protocol using ETSI ITS Messages plus an extension for Time Slot Reservations | UDAP/TLEX [1], [2], [3], [4] |
| Gateway – ETA Calculator | Encoding:JSON<br>Protocol: MQTT | |

*Table 8: Interfaces iTLC-cluster*

### Availability & authentication

| Connection | Availability | Security |
|---|---|---|
| iVRI – iVRI Controller | The iVRI is available at a preconfigured address, usually within a VPN owned by the road operator. In rural area's a cellulair communication is used. | Connections to the iVRI can be secured via TLS, or by use of a VPN.<br>Access to the iVRI is controlled by a username and |

| | | |
|---|---|---|
| | For Zelzate as well Vlissingen we use a 5G connection.<br><br>This address is configured within the iVRI controller, which connects directly to this component. | password combination, accompanied by authorization controls which allow or disallow indidual iVRI controllers from controlling the iVRI. |
| RIS – iVRI Controller | The RIS is available at a preconfigured address, usually within a VPN owned by the road operator.<br><br>This address is configured within the iVRI controller, which connects directly to this component. | Connections to the iVRI can be secured via TLS, or by use of a VPN.<br><br>Access to the iVRI is controlled by a username and password combination, accompanied by authorization controls which allow or disallow indidual iVRI controllers from controlling the iVRI. |
| RIS – Broker | The RIS connects towards a centralized broker. For this, the broker address is configured in the RIS.<br><br>With the broker, the RIS registers a session for a given intersection ID, which effectively is a subscription to all relevant messages. It also allows for the publishing of messages from the RIS towards that broker on that session. | Connections towards the broker are secured via TLS.<br><br>RIS units need have their egress IP's towards the broker whitelisted for access.<br><br>Access to the broker is further controlled via API keys, which serves as a method of authentication and authorization to access or create a set of resourced. |
| ETA Calculator - Broker | The ETA Calculator connects towards a centralized broker. For this, the broker address is configured in the ETA Calculator service. | Connections towards the broker are secured via TLS.<br><br>The services needs to have their egress IP's towards the broker whitelisted for access.<br><br>Access to the broker is further controlled via API keys, which serves as a method of authentication and authorization to access or create a set of resourced. |
| ETA calculator – WP 4 Gateway | The WP 4 gateway needs to guarantee a 99.99% (= daily 8 seconds timeout) uptime in order to facilitate a constant and real-time eta calculation which results in a time slot reservation with constantly updating arrival time estimation | Connections towards WP4 gateway are secured via TLS. Gateway is protected using API keys or other authentication mechanism applicable for B-2-B communication.<br><br>Possbile Firewall rules for accessing gateway at WP4 side with egress IP of ETA calculator? -> Depends on WP4 preference |

*Table 9: Authentication components iTLC-cluster*

Please note that, in this architecture, the EF3 function provider component cannot be accessed directly. Instead, it initiates a connection towards the RIS, which in turn initiates a connection towards a central broker. As a result, this component can in fact be located anywhere, be it in the cloud, in the edge or in situ with the iVRI and RIS hardware. The only limiting factor, is that there must be some form of connectivity between those 3 components. The connectivity between the iTLC and the RIS is established via a 5G connection.

**Interoperability possibilities or problems**

The introduction of (optional) extensions in the SREM, SSEM and CAM data types may require a new version of the RIS-FI protocol, which may be managed via backward compatibility to guarantee interoperability with older versions.

The proposed broker component (UDAP/TLEX) is outside the scope of the project and it may not be compatible with the aforementioned extensions of SREM, SSM and CAM. If this can not be remedied with the proposed broker an alternative broker may be required for the duration of this project.

Similarly, the introduction of a new message type towards the RIS will require a new version of the RIS-FI protocol, which may be managed via backward compatibility to guarantee interoperability with older versions.

The ETA cluster currently acts as a proxy for TTG/TTR information for passing VRIs as EF7 is the only enabling function that interacts with the EF3 broker. This adds extra latency to the input information of the speed advice calculator in EF1 which can influence the advice itself. It might prove to be more feasible to let EF1 also consume the SPAT info itself.

**Hardware and software requirements**

The proposed broker component (UDAP) is outside the scope of the project and it may not be compatible with the aforementioned extensions of SREM, SSM and CAM. If this can not be remedied with the proposed broker an alternative broker may be required for the duration of this project.

Similarly, the introduction of a new message type towards the RIS will require a new version of the RIS-FI protocol, which may be managed via backward compatibility to guarantee interoperability with older versions.

**Latency requirements**

| Connection | Latency Requirement | Reasoning |
|---|---|---|
| iVRI – iVRI Controller | < 150ms | The iVRI controller needs to control the signal in real time, excessive latencies will interfere with internal controlling logic. |
| RIS – iVRI Controller | < 150ms | In this architecture, the RIS functions as an extra hop from the broker towards the iVRI controller. For this reason, and for processing incoming C-V2X communication in real time, latency should be minimized. Here we use a 5G connection. |
| RIS – Broker | < 100ms | As determined in the latency requirements table of Dutch Talking Traffic RFP. |
| ETA Calculator - Broker | <= 100ms | As determined in the latency requirements table of Dutch Talking Traffic RFP. |
| TOV – ETA calculator | < 400ms | As determined in the latency requirements table of Dutch Talking Traffic RFP section 5.1. We take the combination of latency requirements for sending from TOV to gateway and gateway to ETA calculator. Since 5G will greatly lower the sending latency, a total budget of 400 ms seems feasible and will allow ETA calculator 600ms budget for processing location and sending CAM + time slot requests since position sampling. |

*Table 10: Latency requirements iTLC-cluster*

**Other**

-

## 2.4 DP Cluster (EF 4 and EF 1)

**This cluster will provide an extended perceptive range to the remote teleoperator for making the appropriate decisions.** A system of connected sensors on a distinct and diverse range of vehicles each with its limited field of view will be foreseen. The information gathered from LiDARs, and GNNS/IMU will be first compressed and shared over a 5G-based MQTT server (provided by KPN), then aggregated and fused in locally on the TOV to detect surrounding objects and create a global world model (a “map”) which will be used by the teleoperator for better decision making. Since the ego sensors of the TOV will have data that will provide a localized (and incomplete) view of the environment, aggregating real-time data received speedily over the 5G network will allow the teleoperator to have a better sense of the surrounding environments since other vehicles in the immediate environment may have a slightly different and complementary view of that same environment. By fusing the information available from within the TOV with the information collected by other vehicles in its vicinity, a consolidated view of the local environment can be created which is richer in terms of the amount of information than the original view.

The fused “map” will be shared with the teleoperator as a layer on the map-based viewer. Consequently, the detected objects are then sent to EF1 over the 5G network, so to be displayed over the "Enhanced Awareness Dashboard (EAD)", which makes it easier for the teleoperator to relate the position of the detected objects to the surrounding vehicles. This will mainly be adapted by the CACC based platooning. Where trucks in the platoon will broadcast their predictions about the surrounding environment (using locally mounted perception and localization sensors), which will be fused with other predictions broadcasted from trucks in the platoon to create a global map representation. This will ensure that platoons can be kept intact for the full length of the platoon route.

### 2.4.1 Architecture diagram

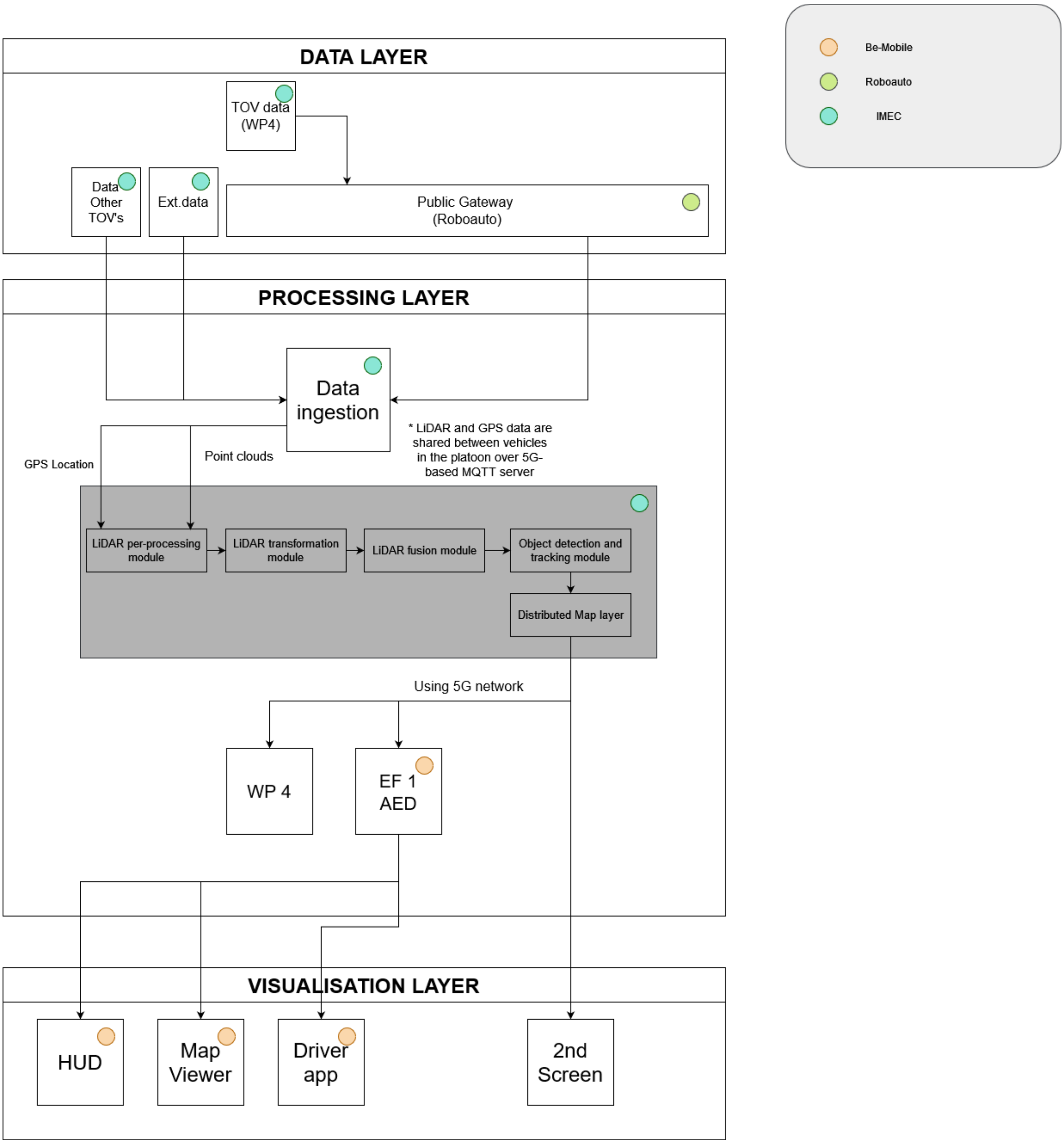



*Figure 5: Application architecture DP-cluster*

### 2.4.2 Component Definition

**General information components**

| Name | Description | Input metadata description & Format | Output metadata description & Format |
|---|---|---|---|
| LiDAR_imec | LiDAR point clouds are fused to create a map and representation of the surroundings. LiDAR surpasses the camera's performance during environmental variations like rain, fog, snow, shades, or low illumination. | Source : IMEC | Sensor type: Mechanical/Solid state<br>Vertical resolution: atleast 64 channels<br>Horizontal FOV: 360 degrees<br>Frequency: 10 Hz<br>Format: bin |
| GPS/IMU | Will be used along with the generated map to enhance the estimates poses and location on the map. | Source: Public gateway (Roboauto) | Output: Location coordinates<br>Format: json |
| Distibuted_fusion_module | Shared world model of the surrounding environment including detection of surrounding objects, **including data received from surrounding TOVs, vehicles etc.** | Input: Cooperative perception message (CPM) containing detected object's dimensions, location, heading and time.<br>Frequecny: 400ms (max)<br>Format: geojson | Shared world model (Map)<br>Output: Detected objects and their locations<br>Format: geojson |

*Table 11: Component definition DP-cluster*

**Availability & authentication**

There will be no authentication (log in, keys) is required.

**Interoperability possibilities or problems**

**2x LiDAR**

- Model: Livox Horizon
- Vertical Resolution: 32 channels
- Range: 120m
- Horizontal Resolution: 512, 1024, or 2048
- Rotation Rate 10 or 20 Hz
- Points Per Second: 655,360
- Connectivity: Ethernet

**GPU**

- Model: Nvidia Jetson Orin

**CPU**

- Model: Intel NUC Kit NUC7i7DNKE

- Intel(R) Core(TM) i7-8650U (4 cores, 8 threads)
- 8 GB RAM, 250 GB SSD
- Number of USB ports: 4
- Number of ethernet ports: 1

**Latency requirements:**

**Bandwidth:** 27 mbps (10 MHz channel)

**Latency in 99% of the packets (one way on application level):** 99% of the packets should have a latency less than 150ms. (packets arriving later than this are considered to be 'lost')

**Max. discontinuity cross border**: 200 ms

**Other**

-

## 2.5 ACA Cluster (EF 5)

### 2.5.1 Short description

**ACA cluster (EF5) will provide a set of teleoperation tailored ADAS features mitigating hazardous situations** caused by a vehicle collision associated with a possible loss of connection and the impaired perception of the environment.

A sensor set containing Lidar, Radar, GNSS-INS and cameras will be installed on TOV. The task of this sensor system is to register obstacles on the TOV's route. The position GNSS-INS module will be used to locate the vehicle in a digital map that contains the planned path of the vehicle.

The main safety features are:

- **Emergency braking in the event of a loss of connection (EBA):** The SGD monitors a heartbeat signal indicating the connection between the vehicle and the operator. If the heartbeat signal does not arrive within the time limit, the vehicle emergency stop procedure will be started.
- **Advanced emergency braking (AEB) system** which can automatically detect a potential forward collision and perform appropriate braking. AEB primarily uses a radar system to detect and track obstacles in the route of a teleoperated vehicle.
- **Adaptive cruise control (ACC) system** that automatically adjusts the vehicle speed to maintain a safe distance from vehicles ahead.
- **Emergency lane keeping system (ELKS)** which can ensure that the vehicle remains in the lane (preplanned virtual track) in the event of a lost connection.

The EF5 system will run on a separate computer system. Each feature will function as a particular software module that connects to the DBW interface with a predefined priority.

EBA and AEB systems are CAS features and therefore have a higher control priority than the teleoperation module. Both systems are manifested only by influencing the longitudinal direction by activating the brakes and deactivation the acceleration. These modules communicate directly with the DBW interface. The EBA system is not aware of its surroundings (it does not process any sensory data) therefore, its priority is lower than that of the AEB.

ACC and ELKS systems are in the function of assistants and therefore have a lower priority than input from the teleoperator or EBA and AEB.

The ELKS system affects the vehicle's movement in the lateral direction, and its goal is to keep the vehicle in the planned virtual lane.

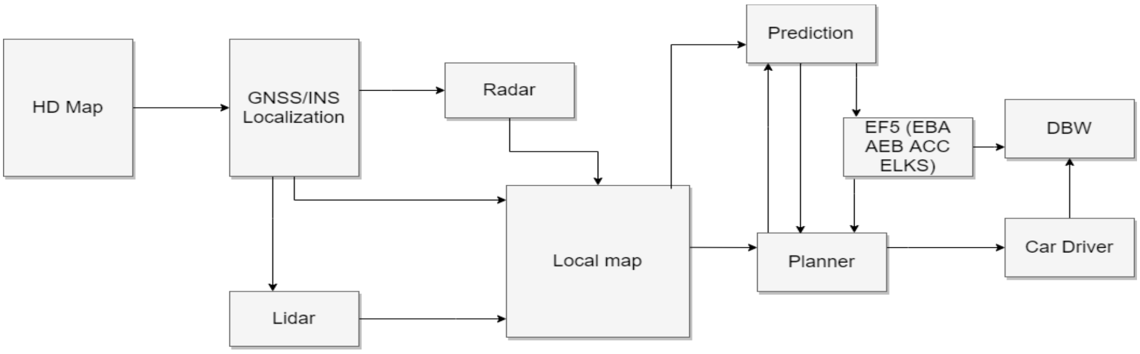



*Figure 6: Functional architecture*

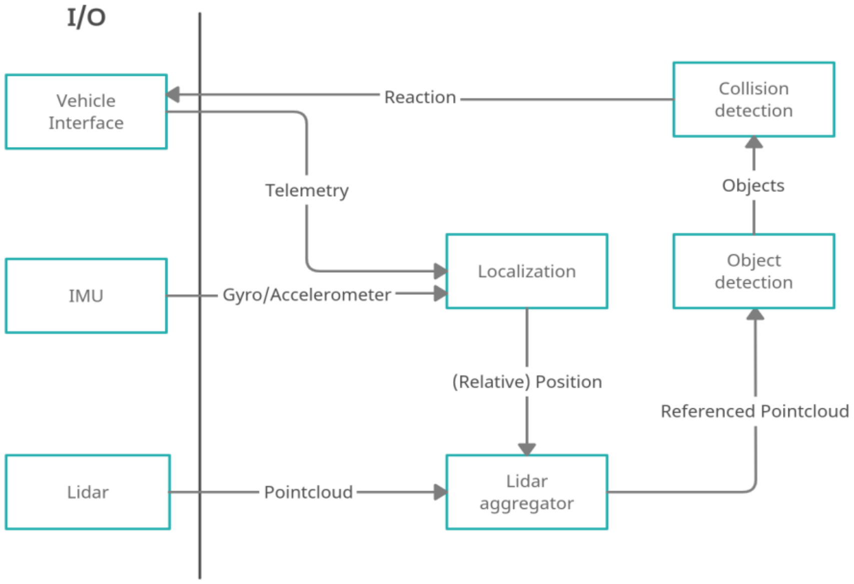



*Figure 7: In/Outs ACA-cluster*

### 2.5.2 Architecture diagram

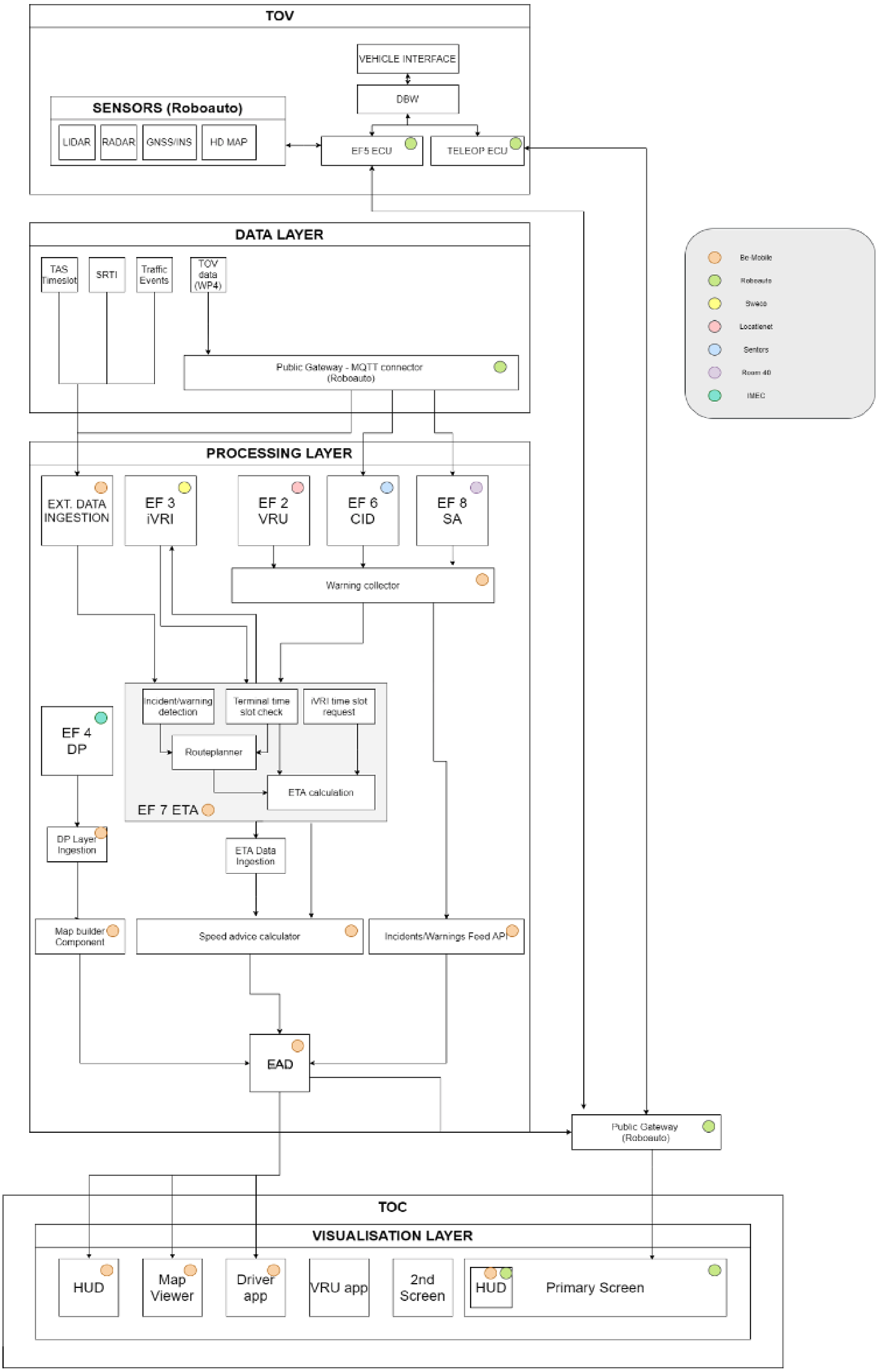



*Figure 8: Application architecture ACA-cluster*

### 2.5.3 Component Definition

**General information components**

| Name | Description | Input metadata description & Format | Output metadata description & Format |
|---|---|---|---|
| EBA | Emergency braking in the event of a loss of connection. | Heartbeat signal with 50 Hz frequency. | Brake activation signal.<br>Acceleration deactivation signal. |
| AEB | Advanced emergency braking (AEB) system which can automatically detect a potential forward collision and perform appropriate braking. | Detected objects in collision course:<br>Object class.<br>Object dimensions.<br>Time to collision.<br>Collision probability.<br>Sensors state. | Brake activation signal.<br>Acceleration deactivation signal. |
| ACC | Adaptive cruise control (ACC) system that automatically adjusts the vehicle speed to maintain a safe distance from vehicles ahead. | Detected objects in collision course:<br>Object class.<br>Object dimensions.<br>Time to collision.<br>Collision probability.<br>Minimal safe distance.<br>Sensors state. | Brake activation signal.<br>Acceleration change signal. |
| ELKS | Emergency lane keeping system (ELKS) which can ensure that the vehicle remains in the lane (preplanned virtual track) in the event of a lost connection. | Current vehicle position.<br>Projected trajectory.<br>Virtual track localization. | Required delta correction (passed to the planner.) |
| GW MQTT Connector | Distributes telemetry from the vehicle | | Current vehicle position.<br>Current speed.<br>Fuel / battery level.<br>Vehicle ID. |

*Table 12: Components definition ACA-cluster*

**Interfaces**

| Connection | Interface protocol | Documentation |
|---|---|---|
| GW MQTT connector – ext.data ingestor | MQTT broker | - CAM messages on MQTT broker of Roboauto, as documented on *https://www.etsi.org/deliver/etsi_en/302600_302699/30263702/01.03.01_30/en_30263702v010301v.pdf* |
| | | - Cas message – messages related to Collision Avoidance System, containing following parameters (retrievable by using the topic structure on the MQTT |

| | | |
|---|---|---|
| | | broker of Roboauto):<br>- CAS status<br>- Objects (that can be detected on the path of the TOV)<br><br>- Teleoperation messages containing the following parameters (retrievable by using the topic structure on the MQTT broker of Roboauto):<br>- Teleoperting status e.g. driving<br>- Station of the teleoperator<br>- Vehicle type |
| EAD – public gateway | MQTT broker | URL to visualisation tool of Be-Mobile. |

Topic structure of MQTT broker of Roboauto :

***vehicle/<vehicle_id>***

*/**cam** - CAM as specified by https://www.etsi.org/deliver/etsi_en/302600_302699/30263702/01.03.01_30/en_30263702v010301v.pdf, the JSON format is specified at the end of this document*

***/telemetry***

*/**fuel** **-** current fuel level - [percent]*
*- format: float represented by a plain string. Example: “47.1” would mean 47.1% of fuel*
*/**speed** - current speed - [m/s]*
*- format: double represented by string. Ex.: “3” - 3m/s*
*/**position/latitude** - current GPS latitude - [degrees]*
*/**position/longitude** - current GPS longitude - [degrees]*
*- format: WGS double represented by string - Ex.: “49.13” - 49°7’48’’*
*/**position/altitude** - current altitude - [m]*
*- format: double represented by string. Ex.: 900 would mean 900 meters above sea level*

*/**environmental** - environmental sensors*

*/**tempext** - external temperature sensor*

*/**<sensor#>** - temperature reading [°C]*
*- format is float represented by string*
*- # indicates the index of the sensor (indexed from 0)*

*/**vibrationext** - external vibration sensor*

*/<sensor#>*

*/**humidityext** - external humidity sensor*

*/<sensor#> - humidity reading [RH%]*
*- format is float represented by string*
*- # indicates the index of the sensor (indexed from 0)*

*/**tempcpu** - CPU temperature*
*- format is float represented by string*

*/**tempgpu** - GPU temperature*
*- format is float represented by string*

*/**cas** - messages related to the Collision Avoidance System*

*/**status** - CAS status*

*/**objects** - parameters of object(s) recognized in collision path*

*/**teleoperation** - messages related to teleoperation*

*/**status** - teleoperation status (Eg.: “driving” or “rsConnectionLost” ...)*

*/**station** - string containing an ID of the remote station that is operating the vehicle (Eg.: “BringAuto RS E01”)*

*/**type** - string containing a vehicle type*

***NOTE:** The MQTT communication is encrypted using TLS. The required certificate will be provided upon request.*

### Availability & authentication

To subscribe to the MQTT broker, a client certificate is required. The certificate will be provided to partners upon request.

### Interoperability possibilities or problems

Data from all EF5 components will be propagated to the Roboauto Gateway, from where it is further distributed using the MQTT protocol. As MQTT is a widely used and well-described communication protocol, we do not foresee any significant problems. Minor issues may arise when accommodating to the MQTT over TLS with client certificate authentication method that is used for increased security when compared to the standard MQTT.

### Hardware and software requirements

### Latency requirements

The latency of communication between the EF5 modules and the DBW must not exceed 100ms, otherwise, the communication line is put into a fault state. The latency of the control commands from the remote station must not exceed 150 ms. The required latency of the sensor communication depends on the given sensor. For lidar data it’s 100 ms, for GNSS-INS data it’s 100 ms, for radar data it’s 150 ms, and for the camera data streamed to the teleoperator it is 200 ms.

## 2.6 Logistics cluster (EF 6, EF 7 and EF 8)

### 2.6.1 Short description

EF6 provides the capability to identify shipping containers and/or rail wagons on video streams. Shipping containers are identified by BIC codes (Bureau of Internation Containers), which is an identification method globally standardized in ISO6346. The BIC code appears on all visible angles at any shipping container. Rail wagons are identified using so-called UIC codes (Union Internationale des Chemins de fer, or in English the International Union of Railways). The UIC code appears on the both sides of rail wagons.

The recognized code can directly be used for automation purposes, e.g. to automatically register containers or rail wagons as being arrived/departed or loaded/unloaded.

The tests are focused on testing the 5G network for Mobile Edge Computing (MEC), so that the software runs in an edge node in the network, rather than physically next to the camera. Figure 9 shows an incoming train at the testing site of Verbrugge (Scaldia site) that is equipped with 5G Stand-Alone coverage.


*Figure 9: Train loaded with containers arriving at the testing site of Verbrugge*

**EF8 foresees in a continuous monitoring of several key areas relevant to teleoperation**, making teleoperation safer and more efficient. Key areas include the port terminal area, the buffer parking and the TOV itself. Monitoring will be done by processing real-time streams from multiple sources through a 5G enabled local edge gateway(s) at the location or in the truck, detecting anomalies in real time. Used data sources are:

- Existing and new 5G connected CCTV camera's with sound capabilities
- New to be installed sound detection equipment with geo-localization
- Other available real-time streams and additional external algorithms working on the same streams can be added at a later stage if deemed interesting.

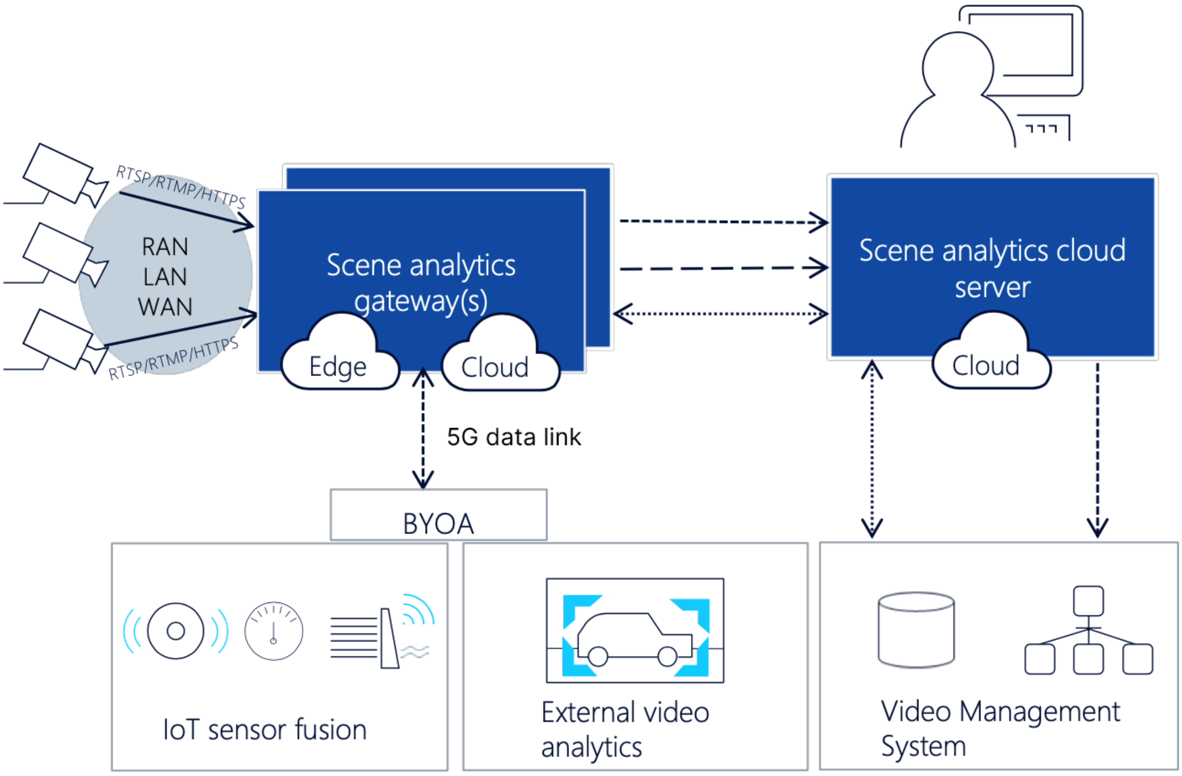



*Figure 10: Functional architecture SA-analytics*

Use-cases include the following:

- Inspection of the TOV: Operators carry out inspection at the start and end of their trip. This inspection becomes more challenging in the physical absence of the teleoperator. The use-case therefore foresees in inspection of the truck by analysing and processing the streams available. Camera and sound streams may pick up on anomalies which point to for example (external or even internal) damage to the vehicle. Anomaly detection may also take place while driving.
- Security breach of the TOV: When the TOV is parked on the port premises or on a buffer parking, the state of the TOV can be continuously monitored. If a security breach is detected (e.g. an attempt to open the door), the EF can warn the teleoperation centre of the breach.
- Suboptimal operations: The continuous monitoring of the TOV's and key areas makes it also possible to detect parts of the transport itinerary that are not going as planned (i.e. that deviate from how the transport should normally go). For example, CCTV camera's at the entry gates of port terminals may pick up that the queue at a terminal entry is longer than typical. This may trigger a message to the teleoperator and to the EF7 provider that

the arrival time of the payload will be delayed.

- The integration of additional data sources or externally developed algorithms that can be hosted on the platform allow for a high degree of flexibility in stitching together different solutions dependant on the requirements of the users. The following high level components will be implemented:

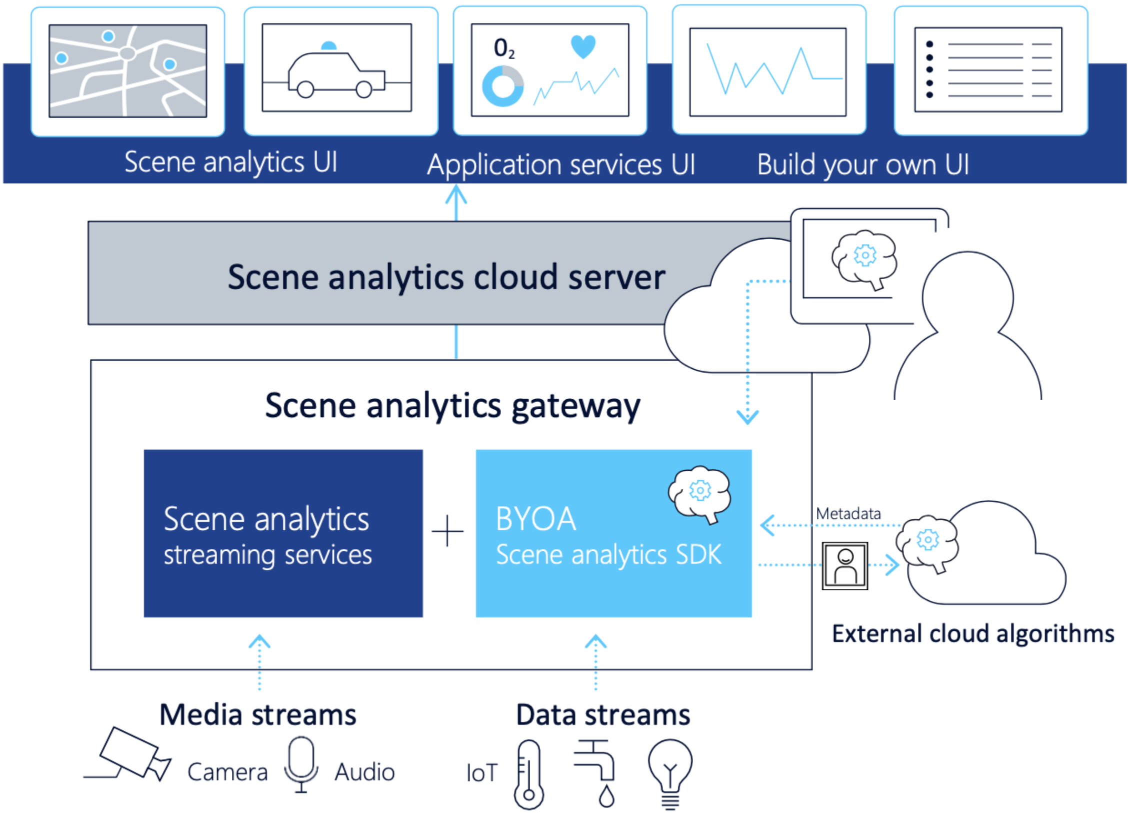



*Figure 11: Components SA-analytics*

### 2.6.2 Architecture diagram

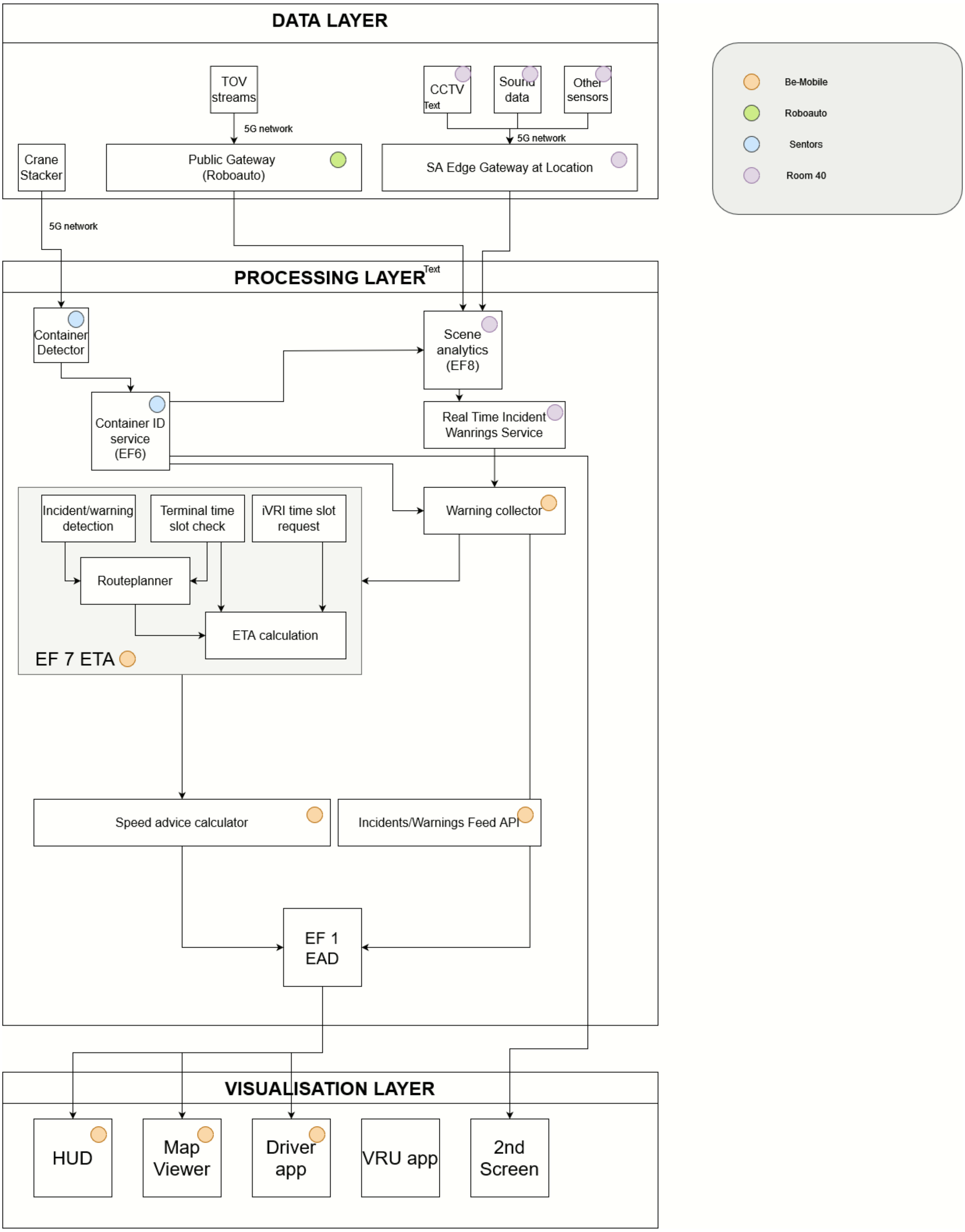



*Figure 12: Application architecture logistics cluster*

### 2.6.3 Component Definition

**General information components**

| Name | Description | Input metadata description & Format | Output metadata description & Format |
|---|---|---|---|
| Container Detector | Continually analyses incoming video data and when a container has been detected a message is sent | IP video stream, interpretable by FFMPEG<br>minimal resolution: 640x480<br>preferred resolution: 1280x960<br>preferred protocol: RTSP<br>preferred transport: TCP<br>preferred framerate: 25 | Digital representation of the detected Container ID (BIC code) as a JSON-message (Container Seen Message). The following fields will be provided:<br>- BIC code<br>- Name: identification of the video stream<br>- Timestamp: indicating when the container was seen<br>- Confidence: measure of confidence for the detected BIC code<br>- Location: the position of the container ID relative to the original video frame (to be able to draw an overlay)<br>Currency:<br>the data will be current; it will be sent as soon as the detection has been made.<br>Credibility:<br>a confidence-measure is provided with each detection.<br>Granularity:<br>no aggregation is done; detections are sent individually. |
| SA Edge Gateway | Local edge compute installed near by the (video) streams or VMS system | IP video stream<br>minimal resolution: 640x480<br>preferred protocol: RTSP; h.264 prefferably | Metadata to be defined |
| Scene Analytics | Cloud (AWS) environment | Onboarding of SA Edge Gateway | Metadata (description of anomalies), Video and sound clips |
| Real Time Incident Warning Service | Anomaly detection in video and sound streams | SA Edge gateway metadata | Metadata with clip or image |
| Warning collector | Collection of adapters that will transform incoming feeds into internal Be-Mobile formats. Two different formats will be used : the Be-Mobile internal traffic events format and a new format : 'warningsmessage'. The latter can be used to send messages including hyperlinks to images or videos to the front-end. | Input from EF6: json message with hyperlink to images.<br><br>Input from EF8: json message with relevant info that can be used for our traffic events. This will be converted to traffic events model of Be-Mobile | Traffic events model v2<br>- Payload: Traffic Events V2 (Be-Mobile model)<br>- Encoding: JSON<br><br>Warnings message<br>- Payload: Warnings message format (Be-Mobile model)<br>- Encoding: JSON |

*Table 13: Components definition Logistics cluster*

## Interfaces

| Connection | Interface protocol | Documentation |
|---|---|---|
| Public Gateway – Container Detector | Digital (IP) video input. Can be any popular format, criterium is that is must be interpretable by FFMPEG | Minimum:<br>resolution: 640x480 pixels<br>framerate: 5 fps<br>bitrate: 5 Mbit/sec<br><br>Preferred:<br>resolution: 1280x960 pixels<br>framerate: 25 fps<br>bitrate: 35 Mbit/sec<br>protocol: RTSP over TCP |
| Crane Stacker – Container Detector | Digital (IP) video input. Can be any popular format, criterium is that is must be interpretable by FFMPEG | Minimum:<br>resolution: 640x480 pixels<br>framerate: 5 fps<br>bitrate: 5 Mbit/sec<br><br>Preferred:<br>resolution: 1280x960 pixels<br>framerate: 25 fps<br>bitrate: 35 Mbit/sec<br>protocol: RTSP over TCP |
| Container Detector – Container ID Service | Proprietary format message detailing the frames where a container was potentially detected. | / |
| Container ID Service – Warning collector | MQTT or WS, depending on the availibity of MQTT broker | / |
| Warning collector - ETA | Websocket | Internal API at Be-Mobile |
| SA Edge Gateway – Scene Analytics | Proprietary Software running under Ubuntu OS | |
| Public Gateway – Scene Analytics | Subscription management API | A 3rd party application server (3PAS) cab subscribe to alerts or post messages to SA |
| Scene Analytics – Real Time Warnings service | Meta data posted to 3PAS at event time | TBD |
| Real Time Warnings Service – ETA Calculation | MQTT or WS, depending on the availibity of MQTT broker | Only anomalies are provided and a description of this if possible. |
| Real Time Warnings Service – EAD | MQTT or WS, depending on the availibity of MQTT broker | Only anomalies are provided and a description of this if possible. |

*Table 14: Interfaces Logistics cluster*

### Availability & authentication

| Name | Availability | Authentication |
|---|---|---|
| Container Detector | The instances of this component will be deployed on the EF-8 platform. Each instance will be configured to work autonomously and pull and process the available live video. | The instances of this component will be deployed on the EF-8 platform. No specific authentication is required; |
| Container ID Service | A single instance of this container will be deployed on the EF-8 platform. It will send a message to preconfigured parties (for each incoming video feed) as soon as a container is detected. | Here also, no specific authentication is required to use the service; it will function autonomously. |
| SA Edge Gateway | An Edge Gateway device will be deployed on site to collect data from sensors such as cameras | Authorized users can either operate within the SA Cloud environment or access can be granted for using the SDK API's. |
| Scene Analytics | SA is the EF-8 hybrid platform spanning over the Edge Gateway for data collection and a Cloud environement to facilitate remote access. | Authorized users can either operate within the SA Cloud environment or access can be granted for using the SDK API's. |
| Real Time Incident Warning Service | A service running on the EF-8 platform sending mqtt messages as soon as an anomaly is detected. | access can be granted for using the SDK API's. |

*Table 15: Authentication Logistics cluster*

### Interoperability possibilities or problems

/

### Hardware and software requirements

The main potential issue that is foreseen is the availability of NVIDIA GPU resources in the EF8 platform. This is needed in order to gain real-time behaviour for the machine learning algorithms that are deployed. Also, a powerful CPU is needed since the image/video algorithms will work on live video feeds.

### Latency requirements

| Connection | Latency Requirements | Reasoning |
|---|---|---|
| Public Gateway – Container Detector | Minimal latency for the transport of the camera-feed to the Container Detector component | Packet loss is very much dependant upon the resolution, encoding, fps and available bandwidth for the stream. Without these variables, not much can be said about this latency requirement. |
| Crane Stacker – Container Detector | Minimal latency for the transport of the camera-feed to the Container Detector component | Packet loss is very much dependant upon the resolution, encoding, fps and available bandwidth for the stream. Without these variables, not much can be said about this latency requirement. |
| Container Detector – Container ID Service | Close to zero latency (milliseconds) for communication between the Container | Bandwidth per stream (for a typical 1280x960 / 15 fps stream), must be |

| | | |
|---|---|---|
| | Detector component and the Container ID Service | more then 20 Mbit/second |
| Container ID Service – ETA Calculation | < 1 min | Container ID recognition is mostly done pre-trip. |
| SA Edge Gateway – Scene Analytics | Close to zero latency (milliseconds) | In order to give time relevant alerts, data needs to be captured with the smallest possible delay. |
| Public Gateway – Scene Analytics | Close to zero latency (milliseconds) | The Public Gateway ought to cause minimal delays in data transmition. |
| Scene Analytics – Real Time Warnings service | Close to zero latency (milliseconds) | The Real Time Warning service needs to be constantly avilible and not cause any additional latency on detections |
| Real Time Warnings Service – ETA Calculation | < 20 s | Due to traffic variability conditons delay in ETA communication is less critical |
| Real Time Warnings Service – EAD | Close to zero latency (milliseconds) | Real Time Warnings need to be transmited with the lowest possible delay due to possible safety implications |

*Table 16: Latency requirements Logistics cluster*

**Other**

The components used for EF6 (Container Detector) rely on neural networks and deep learning algorithms. This means that they are computational components.

For these components to provide results in a timely manner they require the availability of a NVIDIA GPU resources to these components in EF8. To be more detailed, as a minimum they require:

- Linux-based OS (Ubuntu preferred)
- NVIDIA GPU (CUDA) available
- Quad-core CPU
- 16GB of memory
- 2 Gb diskspace

# 3 OVERALL APPLICATION ARCHITECTURE

## 3.1 Architecture diagram

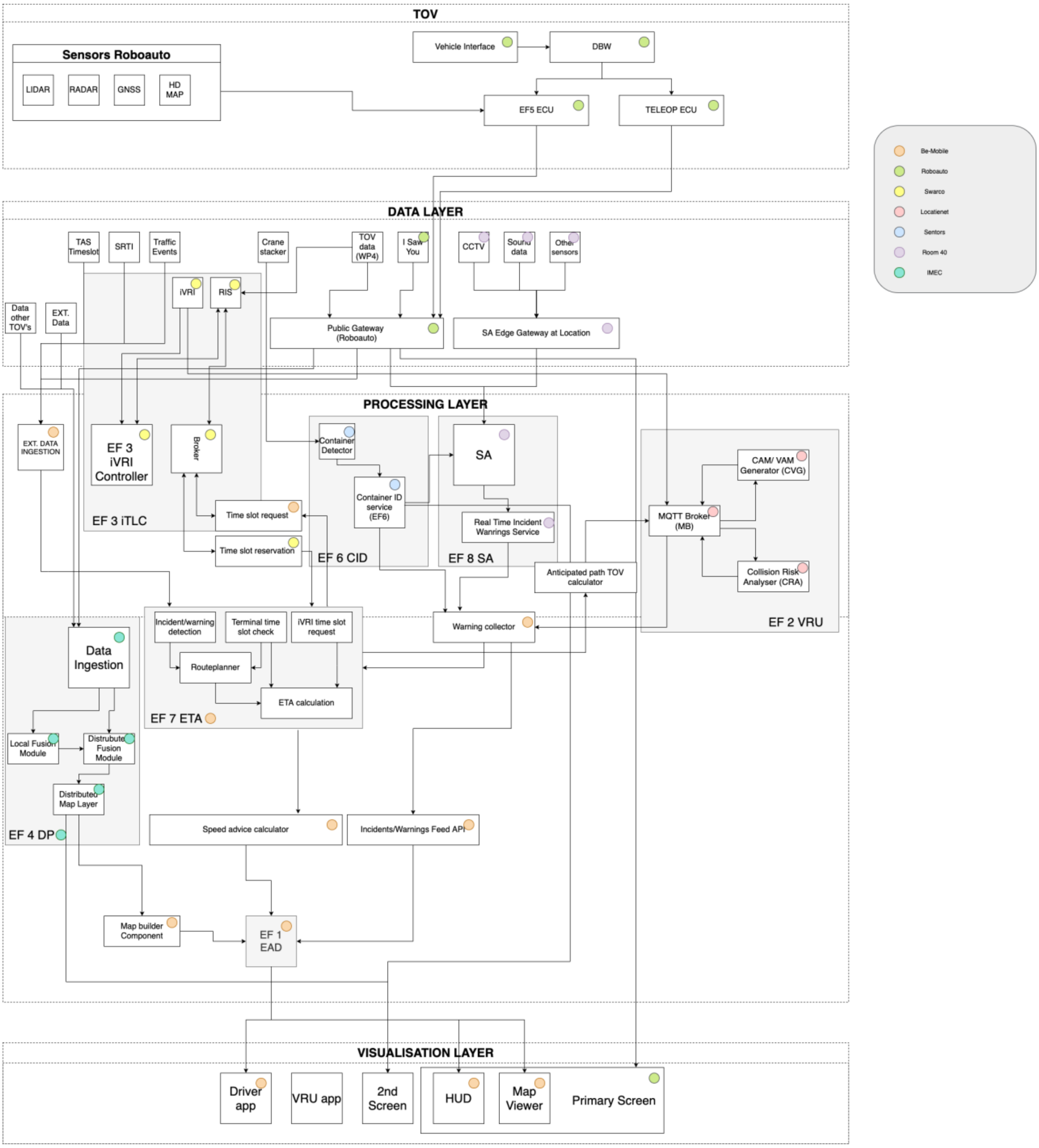



*Figure 13: Overall application architecture*

## 3.2 Interactions between EF Clusters

The diagram above shows how all of the EF’s will cooperate together in order to provide the teleoperated driver with information on both the primary and the secondary screen. As you can see in the overall architecture, the warning collector and the ETA calulculator are two of the most important building blocks of this WP. The ETA calculator because the VRU, the iTLC and the logistics cluster need ETA’s as input and the final ETA will also be visible for the teleoperated driver. The technical architecture is setup so that the teleoperation can continue if one of the EF’s would be unavailable for some reason, but losing the ETA block would mean an immediate loss of several EF's.

The warning collector gets input from VRU, iTLC and logistics cluster, and will collect the warnings in order to be shown to the teleoperator. Because of it's role, it is important that the warning collector is performant and does not add a lot to the latency of the messages. The advantage of this architecture is that the warning collector acts as a hatch of incoming messages, thus it is safe to say that the latency of this component can be minimilised. On the other hand, the component is crucial to enable the functions. Therefore, we need to aim for an uptime of 100% of this component. This will be closely monitored as it is needed in order to provide the teleoperator with curcial warning information, but also the logging of the messages passing through this component can be used for the evaluation of several EF's.

Last but not least, is the EAD of course an important building block because this collects all of the data and ist the last block before the visualisation layer. As this can also be seen as a hatch, here again, it is important to add little or no additional latency to the system and has aimed uptime of 100%.

With the three components described above, we have determined the critical path. The ETA calculation and his input of GPS positions coming from the TOV of WP4, the warning collector that adds information from several other EF's and the EAD that is the hatch to the visualisation layer. All of this components are the responsibility of Be-Mobile, which is also consistent with the fact that they are the leader of this Workpackage.