Ref. Ares(2021)7332306 - 29/11/2021

SHared automation Operating models for Worldwide adoption

SHOW

Grant Agreement Number: 875530

D4.2: SHOW Dashboard

This report is part of a project that has received funding by the European Union's Horizon 2020 research and innovation programme under Grant Agreement number 875530

Legal Disclaimer

The information in this document is provided “as is”, and no guarantee or warranty is given that the information is fit for any particular purpose. The above-referenced consortium members shall have no liability to third parties for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials subject to any liability which is mandatory due to applicable law. © 2020 by SHOW Consortium.

This report is subject to a disclaimer and copyright. This report has been carried out under a contract awarded by the European Commission, contract number: 875530. The content of this publication is the sole responsibility of the SHOW project.

Executive Summary

SHOW project aims to advance sustainable urban transport through the introduction of automated vehicles (AVs). Within the project, many different OEM's, operators, local authorities, and technology providers are involved. Inconsistent AV fleet management practices, together with safety and sustainability concerns, makes it an area with a need for consolidation of resources and processes and with potential for automation. The global project impacts and lessons learnt sharing between the project demo sites will benefit from a consolidated view and approach to monitor all fleets, vehicles, and related impacts. With this consolidated view, we are hoping to be able to access various data, to better optimize operations and assist management decisions, get an insight in the expected impacts and, thus, contribute to the sustainability of the solution.

SHOW Dashboard is a dashboard of SHOW project that provides a unified view of all SHOW sites in terms of performance monitoring and visualization of realtime vehicle information during the project demo phases.

The PoC instance of the SHOW Dashboard platform is available at the following URL:https://demo.innovationcloud.ericsson.net/show-project/home.

SHOW Dashboard interfaces with the centralized SHOW Mobility Data Platform (SMDP[1]) through the data APIs for the collection of SHOW demo site's KPI related data and through the realtime broker for vehicle sensor data (if available). Optionally, it can also leverage the same input interfaces for on-boarding SHOW pilot site data directly from local dashboards or local data collectors during the interim and testing period.

SHOW Dashboard will be used by a diverse user base interested in different features of the platform. These users include core group members, communicators, and the public. Users need to be authorized to access any site data.

The current version corresponds to the revised version emerging as of the interim review (M18) of the project. The comment received in the Interim review was as follows: "However it is not clear what the functional scope of the dashboard is intended to be. There is for instance limited explanation of exception alerting at a tactical level; or of potential exploitability by future CAV sites at a strategic level. It would be helpful to explain the role of the dashboard more clearly". To address this comment, the functional scope of the SHOW Dashboard within the project and its recommended usage are clarified in section 3.2 and the new section 3.4.

Document Control Sheet

Start date of project:	01 January 2020
Duration:	48 months
SHOW Del. ID & Title:	D4.2: SHOW Dashboard
Dissemination level:	CO
Relevant Activities:	A4.3: SHOW Dashboard
Work package:	WP4: System architecture & tools
Lead authors:	Thanh Hai Bui, Per Bröms, Rego Granlund, Valentina Ivanova (RISE), Dhruv Shah, Jagannath Gouda, Stig Persson (Ericsson)
Other authors involved:	CERTH, Ericsson Sweden, Combitech, Keolis, Transdev, UPSTREAM, DATIK, Sensible, STIB, VALEO, TECHNALIA
Internal Reviewers:	Maria Gkemou (CERTH), Anastasia Bolovinou (ICCS), Lucia Isasi (Tecnalia), Paulo Miranda, Daniele Sachetti, Mihai Chirca, Sam Lysons (Transdev), Nikolaos Tsampieris (ERTICO)
Actual submission date:	07/10/2021
Resubmission date as of M18 project review:	29/11/2021
Status:	SUBMITTED
File Name:	SHOW_D4.2_SHOW Dashboard_REVISED

Document Revision History

Version	Date	Reason	Editor
0.1	05/11/2020	Table of contents	Thanh Hai Bui (RISE)
0.2	06/04/2021	Update with suggestions from WP4 leader	Thanh Hai Bui (RISE)
0.7	21/05/2021	Update contents of functionalities and messaging	Dhruv Shah, Jagannath Gouda, Rego Grandlund, Valentina Ivanova
0.8	27/05/2021	Updates from system integration workshop	Thanh Hai Bui (RISE)
1.0	28/05/2021	Pre-final ready for internal review	Thanh Hai Bui (RISE)
1.1	28/06/2021	First reviewer comments addressed (from CERTH, ICCS, TECNALIA, TRANSDEV)	Thanh Hai Bui, Jagannath Gouda, Valentina Ivanova, Dhruv Shah, Per Bröms
1.2	30/06/2021	Review by the Quality Manager	Nikolaos tsampieris (ERTICO)
1.3	30/06/2021	Peer reviewed version sent for submission.	Thanh Hai Bui (RISE)
2.0	07/07/2021	Submitted version	Management team
2.1	17/11/2021	Revised version as of M18 interim review comments	Thanh Hai Bui (RISE
3.0	29/11/2021	Final revised version for resubmission by Technical Manager	Maria Gkemou (CERTH/HIT)

Table of Contents

Executive Summary ... 3
Table of Contents ... 5
List of Tables ... 8
List of Figures ... 9
Abbreviation List ... 11
1 Introduction ... 12
1.1 Purpose and structure of the document ... 12
1.2 Intended Audience ... 12
1.3 Interrelations ... 12
2 Relevant initiatives and standards ... 15
2.1 Drive Sweden innovation projects ... 15
2.1.1 KRABAT ... 15
2.1.2 AD Aware Traffic Control ... 15
2.1.3 AVTCT ... 16
2.2 Data standards ... 16
2.2.1 Transmodel ... 16
2.2.2 NeTEx ... 16
2.2.3 SIRI ... 16
2.2.4 NOPTIS ... 16
2.2.5 GTFS ... 17
2.2.6 The formulation of SHOW's vehicle data model ... 17
2.3 Map-based visualization best practices ... 18
2.3.1 Map visualisation toolkit (Leaflet) ... 18
2.3.2 Map Tile Server (Mapbox) ... 19
2.3.3 Map in SHOW Dashboard ... 19
3 SHOW Dashboard architecture and design ... 22
3.1 SHOW Dashboard system requirements ... 22
3.2 SHOW Dashboard functionalities ... 23
3.2.1 Performance monitoring and visualization ... 23
3.2.2 Experimental/optional functionalities ... 24
3.3 SHOW Dashboard in the overall architecture ... 24
3.4 Functional scope and recommended usage ... 26

3.4.1 Functional scope of SHOW Dashboard: 26
3.4.2 Recommended usage of the reference SHOW Dashboard 27
3.5 SHOW Dashboard Component diagram 28
3.6 SHOW Dashboard Component descriptions 30
3.7 SHOW Dashboard interfaces 30
4 SHOW Dashboard integration and development 32
4.1 SHOW Dashboard development methodology 32
4.2 Ericsson Innovation Cloud 32
4.2.1 Introduction 32
4.2.2 Innovation Cloud Model 33
4.3 Dashboard PoC instance and the encapsulated components 34
4.4 SHOW Dashboard Users 36
4.5 SHOW Dashboard User interfaces 37
4.6 SHOW Dashboard data 38
4.6.1 Site metadata 38
4.6.2 Vehicle data 39
4.6.3 KPI Query Data Structures 40
4.7 SHOW Dashboard KPI implementation 41
4.7.1 KPI Broad category 41
4.7.2 List of KPIs 41
4.8 SHOW Dashboard API interfaces 41
4.8.1 MQTT 42
4.8.2 REST API 43
4.8.3 HTTP/MQTT proxy 43
4.9 KPI query structure 43
4.10 Local data collector prototype 46
4.10.1 Architecture 47
4.10.2 Local data collector messaging API 47
4.11 SHOW Dashboard self-service on-boarding 48
4.11.1 SHOW Dashboard entity hierarchy 48
4.11.2 Community on-boarding 48
4.11.3 Partner on-boarding 49
5 SHOW Dashboard reference PoC 50
5.1 Site management 50

5.2 Geofence management ... 52
5.2.1 Geofence ... 52
5.2.2 Geofencing rules ... 53
5.3 Multi-site dashboard ... 54
5.4 Site Dashboard ... 55
5.5 Site data setup ... 57
5.5.1 Site metadata ... 57
5.5.2 Telemetry data ... 58
5.5.3 KPI data ... 59
6 Conclusions and outlook ... 60
References ... 61
Appendix I: Demo sites support for Local Dashboards or SHOW Dashboard ... 63
Appendix II: Specifications for KPI Data representation ... 67
Appendix III: Swagger OpenAPI specifications ... 70

List of Tables

Table 1: D4.2 interrelation to other projects. 13

Table 2: D4.2 interrelation with other SHOW work items. 14

Table 3: SHOW Dashboard system requirements. 22

Table 4: Recommended usages of the SHOW Dashboard. 27

Table 5: SHOW Dashboard component descriptions. 30

Table 6: Data sources interfaces. 31

Table 7: Existing and newly developed components. 35

Table 8: SHOW Dashboard users and authorizations 36

Table 9: Vehicle data model. 39

Table 10: Local Dashboards and intended usages of SHOW reference Dashboard current status (the Mega sites) 63

Table 11: Local Dashboards and intended usages of SHOW reference Dashboard current status (the Satellite sites) 65

List of Figures

Figure 1: A4.3 Interrelations with other WPs. 13

Figure 2: NOPTIS interfaces [11]. 17

Figure 3: The map system showing automated vehicles (marked circles). 18

Figure 4: Map constructed by a set of static images generated by a tile-server. 19

Figure 5: Map layers in SHOW Dashboard (1: Site layer, 2: Digital twin layer, 3: Route layer, 4: Geofence layer). 20

Figure 6: SHOW Dashboard map view with layers. 20

Figure 7: SHOW Global Dashboard map indicating Sites in Europe. 21

Figure 8: SHOW Gothenburg/ Lindholmen site with route and real-time positions of AVs. 21

Figure 9: SHOW Dashboard in the SHOW reference architecture. 25

Figure 10: SHOW Dashboard integration diagram. 26

Figure 11: SHOW Dashboard architecture diagram (Component level). 29

Figure 12: SHOW Dashboard PoC development methodology. 32

Figure 13: Innovation Cloud (EIC) data lifecycle [3]. 33

Figure 14: Innovation cloud (EIC) model [3]. 34

Figure 15: Data frequencies. 40

Figure 16: Available measurement units of KPIs. 41

Figure 17: SHOW Dashboard component and its interfaces to external components/systems. 42

Figure 18: OpenAPI Swagger in path parameters. 44

Figure 19: OpenAPI Swagger query parameters example. 45

Figure 20: OpenAPI Swagger response example. 46

Figure 21: Local data collector prototype architecture. 47

Figure 22: SHOW Dashboard entity relationship diagram. 48

Figure 23: SHOW Project community. 49

Figure 24: Site management. 50

Figure 25: Fleet data registration. 51

Figure 26: Vehicle data registration. 52

Figure 27: Geofence creation. 53

Figure 28: Geofence management. 54

Figure 29: SHOW Dashboard multi-site view screenshot. 55

Figure 30: SHOW Dashboard, bright theme. 55

Figure 31: SHOW Dashboard site view (KPI screen).. 56
Figure 32: SHOW Dashboard site view (vehicle real-time position)........................ 57
Figure 33: Site metadata.. 58
Figure 34: Vehicle specification.. 58
Figure 35: MQTT vehicle data topics.. 59

Abbreviation List

Abbreviation	Definition
AD	Automated Driving
ADS	Automated Driving System
API	Application Programming Interface
AV	Autonomous Vehicles
AVxPT	AVs for PT (UITP, SPACE project)
AVL	Automatic Vehicle Location
CAV	Connected and (fully) automated vehicle
CCAV	Collaborative Connected Autonomous Vehicles
C-ITS	Co-operative Intelligent Transport Systems
SMDP	SHOW Mobility Data Platform
DRT	Demand Responsive Transport
EC	European Commission
ETA	Estimated Time of Arrival
EU	European Union
HTTP	Hypertext Transfer Protocol
HW	Hardware
I2I	Infrastructure To Infrastructure (communications)
IoT	Internet of Things
IP	Internet Protocol
KPI	Key Performance Indicator
MQTT	Message Queuing Telemetry Transport
NAP	National Access Point
NOU	National Organization Unit
OBU	On Board Unit
OEM	Original Equipment Manufacturer
PoC	Proof of Concept
PT	Public Transport
PTA	Public Transport Authority
PTO	Public Transport Operators
REST	REpresentational State Transfer
RSU	Roadside unit
SDK	Software Development toolkit
SP	Sub Project
SW	Software
TLS	Transport Layer Security
TMC	Traffic management center
TSMO	Transportation Systems Management and Operations
URI	Uniform Resource Identifier
V2C	Vehicle to Cloud
V2D	Vehicle to Device
V2G	Vehicle to Grid
V2I	Vehicle to Infrastructure
V2N	Vehicle to Network
V2P	Vehicle to Pedestrians
V2V	Vehicle to Vehicle
V2X	Vehicle-to-X (X represents any entity capable of receiving C-ITS communications)
VRU	Vulnerable Road User
W3C	World Wide Web Consortium
WG	Working Group
WoT	Web of Things
Wp	Work package (of the project)

1 Introduction

1.1 Purpose and structure of the document

This document complements the initial description provided in "*D4.1- Chapter 5- Functional preview of the SHOW Dashboard: SHOW operational Dashboard*"[2] with further technical and implementation details. The document is structured as follows:

- ***Chapter 1 Introduction***: This chapter positions the A4.3 task and the SHOW Dashboard within the SHOW project.
- ***Chapter 2 Relevant initiatives and standards***: Description of relevant initiatives and standards that are used as references and inspiration for the Dashboard architecture, design and development.
- ***Chapter 3 SHOW Dashboard architecture and design***: Architecture and functionality of the Dashboard and its relationships with other systems.
- ***Chapter 4 SHOW Dashboard integration and development***: Description of the integration interfaces and implementation details.
- ***Chapter 5 SHOW Dashboard reference PoC***: Screenshots and descriptions of the Dashboard functionalities.
- ***Chapter 6 Conclusions and outlook***
- ***Appendices*** provide the status of the local site dashboards and initial intentions of site's usages of the SHOW Dashboard (I)**,** the specifications for KPI Data representation (II) and the Swagger OpenAPI specifications (III)

1.2 Intended Audience

The deliverable is confidential, the intended audience of this work is limited to:

SHOW SP3 demo sites' technical teams responsible either for the CAVs operation, the service design and evaluation or/and any local systems' integration with the SHOW system (representing the CAV fleet, the demo sites' infrastructure, and any local backend cloud system involved). More specifically the following groups are addressed:

- Evaluation team (WP9)
- Technical validation team of WP11
- Experimenters of WP11 and WP12 (Real-life demonstrations in pre-demo and final demo phase of the project respectively)
- Simulation analysts of WP10

Apart from the above, the Data Collection and Management tools of the project are inevitably concerned in this work.

1.3 Interrelations

Deliverable's main internal interrelations to other WPs/Activities have been developed throughout the first one and a half year of the project and have been supported by the WP4 interviews with the demo sites and the SP2 development teams and are presented in Figure 1 and outlined hereafter:

- Other WP4 architectural activities that progress in parallel and in interactive mode to ensure the integrity of the SHOW Dashboard as part of the overall architecture. SHOW Dashboard is also included as a part of the reference architecture described in D4.1.
- WP5: SHOW Dashboard collects its near real-time and historical KPI data through SHOW's SMDP (A5.1). A4.3 is closely connected to WP5 in data definition, data registry, data APIs.

- SP2 activities regarding technical aspects of vehicles, infrastructure, services and negotiating the availability of data; WP7, with vehicle-related data, on-board architecture and connectivity, experiments with other road users.
- A1.3 SHOW Use cases, provide the high-level requirements that all the systems within the projects shall support.
- A9.4, provides the KPI definitions that will be implemented in the SHOW Dashboard visualizations.
- WP11: Pilot implementation and integration (with two early adopter sites Linköping and Gothenburg, Sweden megasite).
- WP12: SHOW Dashboard will be used during the real-life demonstration.

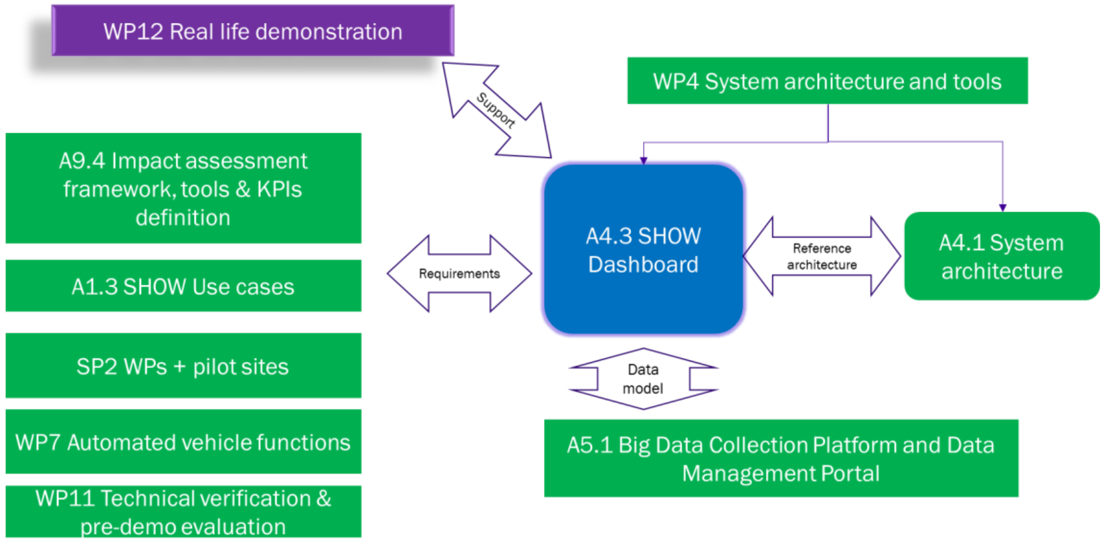


Figure 1: A4.3 Interrelations with other WPs.

The D4.2's interrelations to external projects, initiatives, platforms are indicated in Table 1.

Table 1: D4.2 interrelation to other projects.

External initiative	Item of interest for SHOW Dashboard
ITxPT organization	Technical specs data interfaces
Drive Sweden, Innovation Cloud[1]	Digital innovation platform that provides tools, experiences and lessons learned from its connected research projects in highly automated mobility and control center.
Data4PT	Insights from workshops with PT stakeholders, Data models to be promoted
GTFS-RT	Google Transit feed specs
SIRI/NETEX/NOPTIS	Data standards

The relations between this task A4.3 (D4.2) and other work packages within the SHOW project are provided in Table 2.

[1] https://smartcitysweden.com/best-practice/370/drive-sweden-innovation-cloud/

Table 2: D4.2 interrelation with other SHOW work items.

Other WP	Interrelation
A4.1	Reference system architecture
A9.4	KPI data definition and computation formulas
WP5	Data collection platform (SMDP), data models, messaging standards, KPI computation, integrations
WP7	Vehicle data availability
WP11	Site onboarding (pilot test)
WP12	

2 Relevant initiatives and standards

In the design process of the dashboard the team early identified the necessity of relying on existing standards for an efficient design and implementation process. Especially data models and communication interface standards are of importance.

2.1 Drive Sweden innovation projects

SHOW Dashboard is developed in connection with an innovation platform [3] that hosts multiple research projects within the DriveSweden [4] program. The platform inherits the latest innovations from the program in the forms of knowhow of the Dashboard core team and existing technical components and design concept. The following representative innovation projects are the key inspirations of this task.

2.1.1 KRABAT

The mission of KRABAT [5] from the Swedish government innovation partnership program, “The next generation’s travel and transport,” is to run a self-driving, electric and shared vehicle project in a system solution that can push Sweden against a shift in the transport system. To implement this on a large and full scale, huge investments are needed from all involved players, but demonstrations and reality-driven pilots are initially critical to creating a sigh among users to radically change the way in which they have so far solved their mobility. The project consists of several relevant sub-projects:

- Autopilot: build on the project Autopilot in Kista to create a whole that is integrated into the existing public transport and to continue driving Sweden's leading position for integrated, shared and electric self-driving vehicles.
- 5G Ride: Pilot how 5G and control towers can help manage public transport in the future with AVs.
- Connected Traffic Signals: connect a number of traffic signals to Drive Sweden's Innovation Cloud to enable optimization of connected and automated vehicles.
- System Solutions for Shared Cars.

The lessons and experiences regarding 5G communications, connected IoT are valuable reference assets for the SHOW Dashboard.

2.1.2 AD Aware Traffic Control

AD aware traffic control project [6] handles a digitized traffic control for connected and autonomous vehicles with traffic control based on different information flows. Within the project, an interface will also be proposed and implementation of information flow module for implementation in a cloud-based environment, where information from several different public and/or private sources can be collected and consolidated. The goal of the project was to define and propose a traffic control cloud for automated vehicles with interfaces to vehicles, road authorities, and city authorities, along with the associated information flows for connected vehicles. In addition, the project proposed solutions on required services including traffic control and information sharing.

AD aware traffic control project demonstrates a working system with information flow from the car to different clouds with different communication protocols and message structures.

The Dashboard benefits from this project the experiences and experimental lessons for messaging protocols, network topology, and multi-actor interaction schemes.

2.1.3 AVTCT

Automated Vehicles Traffic Control Tower project [7] investigates the effects of traffic on automated vehicles and traffic management of commercially operated fleets and public transport.

This project provides insights for the SHOW Dashboard regarding the remote operation of AVs and different design perspectives with regards to safety and comfort.

2.2 Data standards

Several models for the representation of public transportation data have been evaluated and reviewed to build a common vehicle model that is suitable for SHOW project. The subset of different datatypes from these standards that are selected as SHOW Dashboard I/O data and will be described in Section 4.6. This section briefly describes the data models that have been considered.

2.2.1 Transmodel

Transmodel [8] is the CEN European Reference Data Model for public transportation information (EN 12896: 2006 (Transmodel V5.1)). Transmodel is a conceptual model providing a unified vocabulary and data structures for public transport-related data. It has a broad scope and includes a wide range of topics such as (the names of the different parts are listed): Part 2: Public Transport Network (EN 12896-2:2016), Part 3: Timing Information and Vehicle Scheduling (EN 12896-3:2016), Part 4: Operations Monitoring and Control (EN 12896-4:2016), Part 5: Fare Management (EN 12896-5:2016), Part 6: Passenger Information (EN 12896-6:2016), Part 7: Driver Management (EN 12896-7:2016) and Part 8: Management Information & Statistics (EN 12896-8:2016).

2.2.2 NeTEx

NeTEx [9] (CEN/TS 16614-1:2020), Network Timetable Exchange, is a technical standard for exchanging rarely changing public transport data. NeTEx provides a XML implementation of several of the Transmodel parts and focuses on the topology of the public transportation network (CEN/TS 16614-1:2014), scheduling information for various transportation modes (CEN/TS 16614-2:2014) as well as fare models (CEN/TS 16614-3:2015).

2.2.3 SIRI

SIRI [10], Service Interface for Real Time Information, is another technical standard based on the Transmodel. Similar to NeTEx, it provides XML implementation but focuses on dynamic/real-time data. It provides three functional services to exchange: 1) planned and real-time timetables, arrivals and departures as well as vehicle positions and connections between trips; 2) real-time state of facilities at stations and stops and 3) messages about the public transportation network and services.

2.2.4 NOPTIS

The Nordic Public Transport Interface Standard (NOPTIS [11]) is a set of aligned Transmodel-based interfaces supporting the interconnection of subsystems within a public transport information system, including planning systems, schedule databases, GIS-systems, real-time vehicle reporting systems, traveller information systems, travel-planning systems, etc.

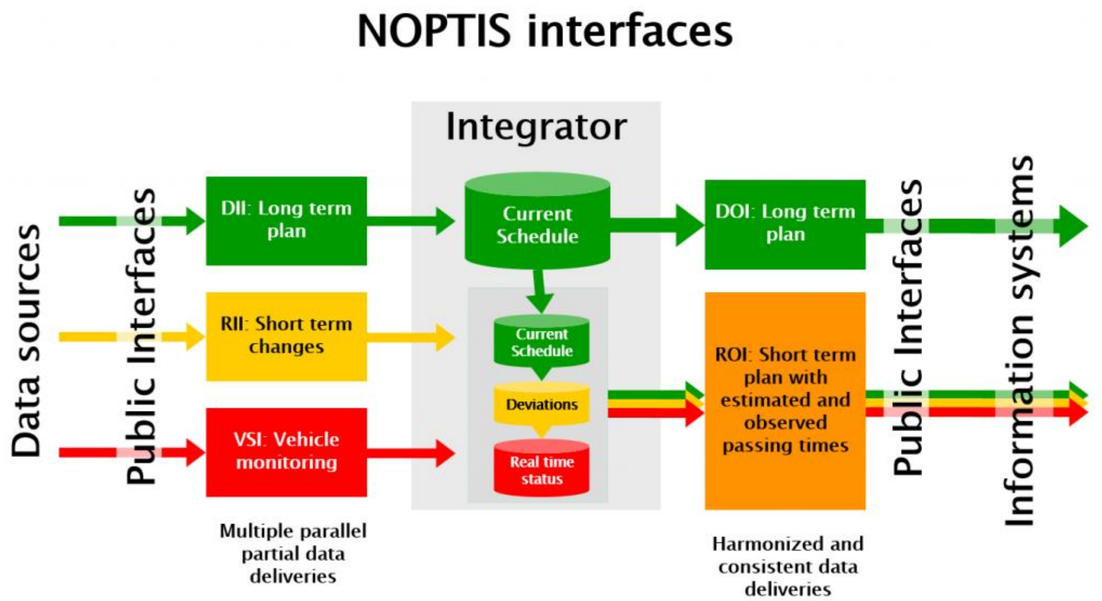


Figure 2: NOPTIS interfaces [11].

2.2.5 GTFS

GTFS [12] (General Transit Feed Specification) format is a widely used standard for public transportation data. The GTFS format has two components – GTFS-static and GTFS-realtime. GTFS-static data are needed to interpret and analyse the GTFS-realtime data. The GTFS-static format describes the setup of the public transportation services including routes, trips, schedules, and transfers between routes and is distributed in an archive containing text files. GTFS-realtime format provides dynamic information about the current position of public transportation vehicles (VehiclePosition feed), real-time trip updates (TripUpdate feed) and disruption of service (Service Alerts feed). The GTFS-realtime is distributed in protocol buffer format. Both – the real-time and static components – have a simpler structure in comparison to the other standards mentioned above. SHOW vehicle model development

2.2.6 The formulation of SHOW's vehicle data model

The development of the SHOW vehicle model started with an evaluation of the aforementioned standards and a number of design workshops with participants from some members of SHOW Sweden megasite (RISE, Ericsson and Combitech). During the workshops, participants discussed relevant aspects of the existing models, SHOW KPIs and combine with diverse experiences in working with public transportation data and automated vehicles. The data model was then communicated to other related tasks and will be finalized by WP5.

The development of the vehicle payload and vehicle equipment parts of the SHOW vehicle model was informed mostly by the respective Transmodel and NeTEx parts regarding vehicle passenger and accessibility equipment. The equipment was further specialized into sensor, actuator, and network equipment to highlight the various aspects of the collected operational data. The vehicle identity data was drawn from NeTEx and NOPTIS vehicle information. The environmental aspects of the SHOW vehicle model, including fuel and emissions, were mostly based on the KPI definitions. Since the SHOW vehicles are largely automated vehicles, a category called intelligent vehicles and infrastructure was created to represent vehicle capabilities in connection to automated driving and communication with automated vehicles and infrastructure.

2.3 Map-based visualization best practices

In the dashboard, both the central SHOW Dashboard but also for local implementation, there is a need for map-based visualisations. One well developed open source system is Open Street Map [13]. Open Street Map has a couple of slightly different implementations of the map APIs. Mapbox [14] and Leaflet [15] are used in this project. Mapbox is used in the Ericson Innovation Cloud system whereas Leaflet has been implemented in the local system in Linköping. Figure 3 illustrates the two SHOW's automated vehicles that are operating at the Linköping site in Sweden and how they are visualized as a map layer.

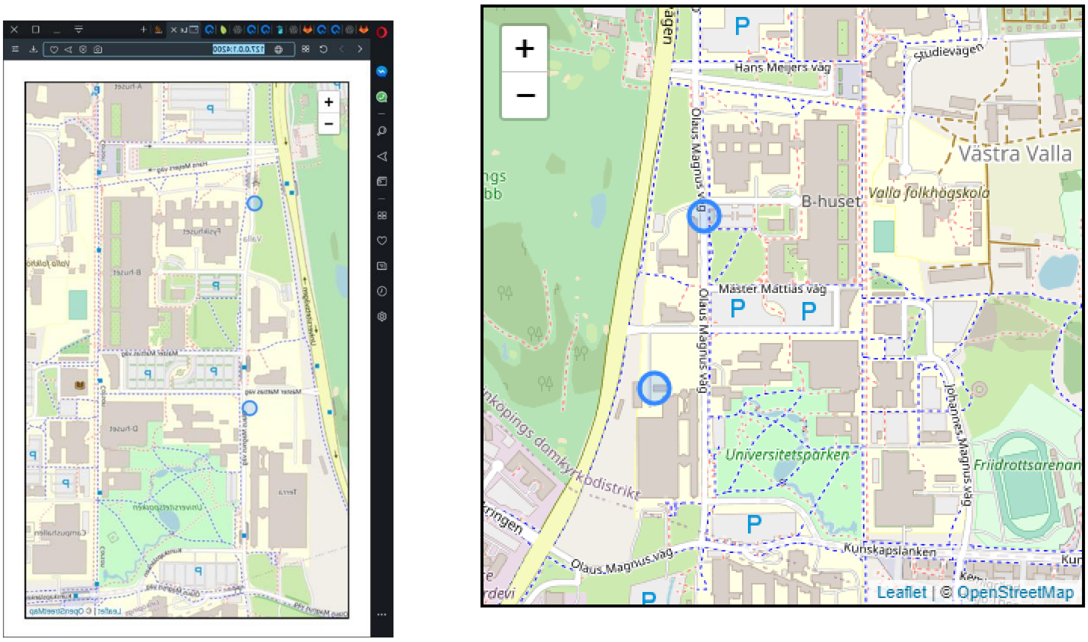


Figure 3: The map system showing automated vehicles (marked circles).

2.3.1 Map visualisation toolkit (Leaflet)

One of the popular map visualisation toolkits is Leaflet [15]. Leaflet an engine rendering the geoinformation on the map. It is a standard web browser toolkit, written in JavaScript, used to display tile-based maps. The toolkit contains two basic components a JavaScript module responsible for displaying different geoinformation layers and a module rendering the background map tiles (images).

The main map layers used in the SHOW Dashboard map are:

- **Tile layer** - the tile layer is the background in the map. It displays the tiles, static images, received from a tile server.
- **Marker layer** – the marker layer is used to display clickable icons on the map. For example, this layer is used to display vehicles and sites on the map.
- **Image layer** – the image layer is used to display image-based markers and information.
- **Line layer** – the line layer is a polyline drawing layer, that makes it possible to draw lines and areas with a set of geographical points; for example, this layer can be used to draw vehicle planned routes, bus lines and fleet's geographical area, etc.
- **Popup layer** – the popup layer is used to display different types of information connected to the marks; for example, displaying information about vehicles and sites.

2.3.2 Map Tile Server (Mapbox)

The web-based map renders often use tile-servers to get the background map image. The SHOW Dashboard map uses Mapbox as its tile-server. A tile-server generates a set of static images that are used by the map render to construct the map. The look and feel of the displayed map depend on the static image tiles. When the map module renders the map, it combines the static images generated by the tile-server for the specific zoom level (see Figure 4). The look and feel of the map can often be configured at the tile-server user configuration.

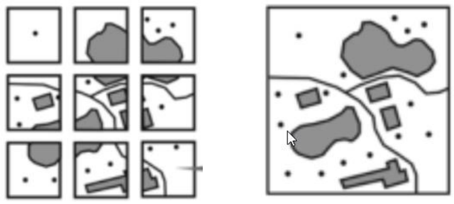


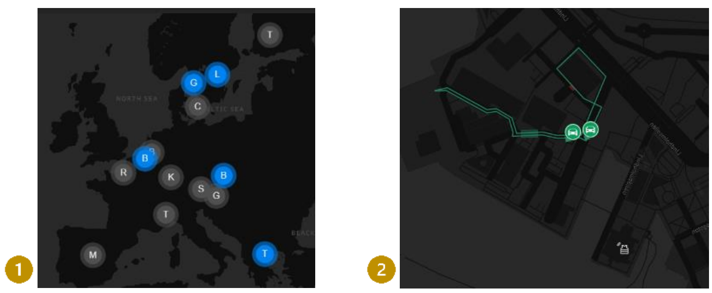
Figure 4: Map constructed by a set of static images generated by a tile-server.

2.3.3 Map in SHOW Dashboard

Map visualization and tracking is a key aspect of web-based dashboards for automotive trials. The availability of geographical data for tracking/reporting vehicle-centric information makes it essential to plan and implement visual cues to represent the relevant data in a map-based visualization. There are multiple map-visualization libraries and tools available for representing geographical data, such as Leaflet, Here Maps [16], Google Maps [17], etc. In SHOW Dashboard, Mapbox GL JS library is used to implement a map layer to visualize different geographical entities.

Map layers included in SHOW Dashboard (Figure 5) are of the following types:

1. Base layer: A base layer (currently uses dark theme) is implemented as the base map layer.
2. Site layer: Custom static markers are used in this layer to visualize different SHOW pilot sites across Europe in the Dashboard.
    a. Grayed markers represent inactive sites (sites that are registered but not yet enabled for retrieving real life data) and
    b. Colored markers represent Active sites.
3. Digital Twin Layer: Custom dynamic markers are used to represent geographical information of real-world vehicles and other traffic participants such as Vulnerable Road User (VRU) on the map.
4. Route layer: Routes and stops are hosted by a separate layer in the Dashboard.
5. Geofence layer: Indicates geofenced areas on the map.



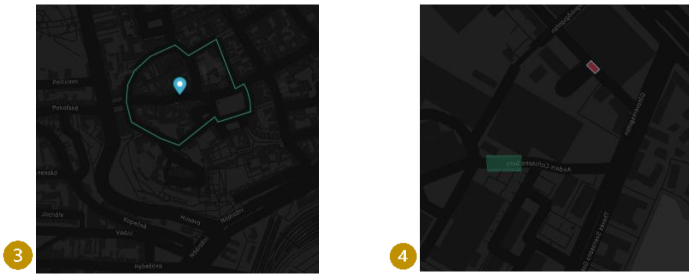


Figure 5: Map layers in SHOW Dashboard (1: Site layer, 2: Digital twin layer, 3: Route layer, 4: Geofence layer).

Figure 6 shows an example of different map layers that can be overlayed on the same map. The layers are representing different information layers that can be selected by user to hide or display on map.

Map layers can also be used to display overview or detailed information, for example, the overview screen of SHOW Dashboard is only displaying locations of SHOW sites (Figure 7) while the site view can also display information such as vehicle routes, stops and current position of the vehicle (Figure 8).

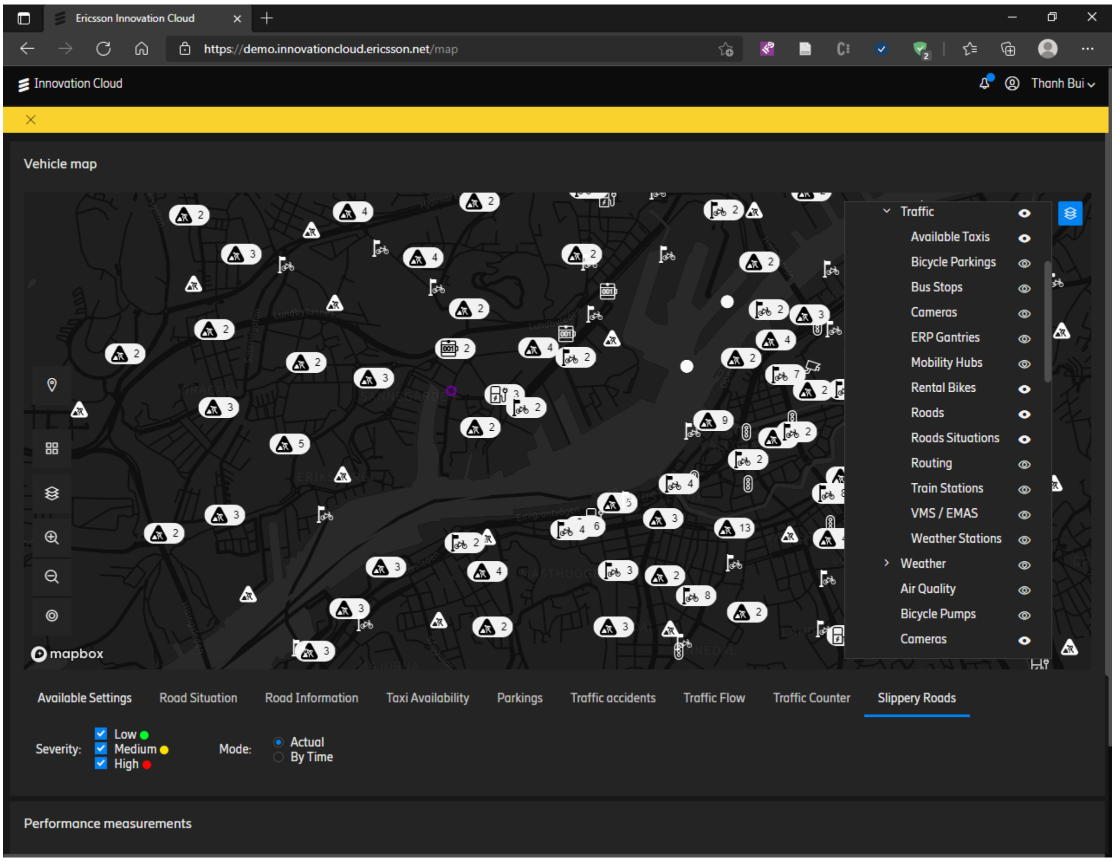


Figure 6: SHOW Dashboard map view with layers.

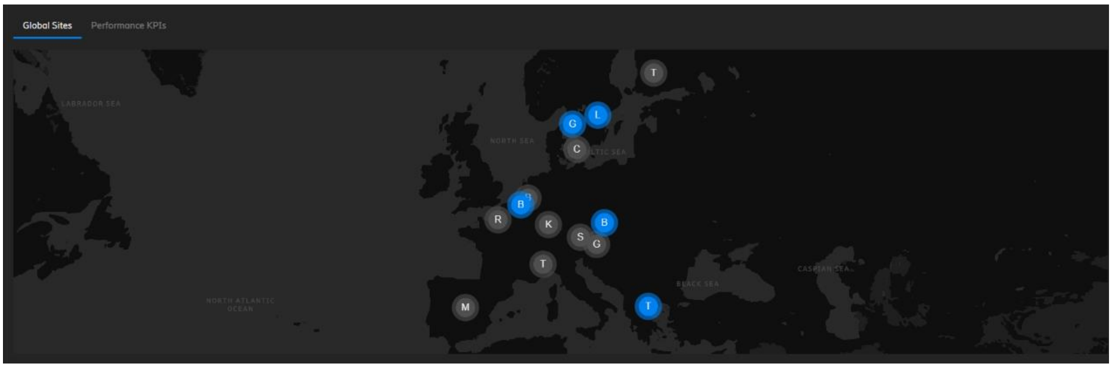


Figure 7: SHOW Global Dashboard map indicating Sites in Europe.

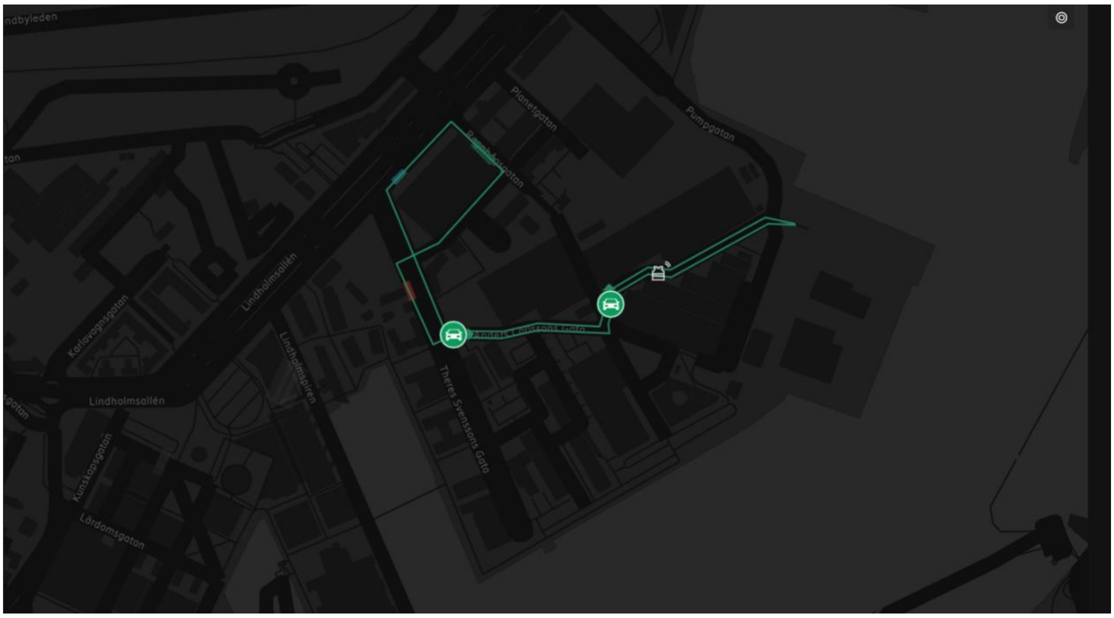


Figure 8: SHOW Gothenburg/ Lindholmen site with route and real-time positions of AVs.

3 SHOW Dashboard architecture and design

3.1 SHOW Dashboard system requirements

The requirement specification for SHOW Dashboard (has been collected during the Explore phase according to the methodology described later in Section 4.1) is provided in the following Table 3.

Table 3: SHOW Dashboard system requirements.

Req ID	Description	Type
A43-01	SHOW Dashboard should provide tools to input metadata (vehicle, fleet).	Performance
A43-02	SHOW Dashboard should provide a mechanism to navigate through KPIs of different aggregation levels.	Performance
A43-03	SHOW Dashboard should provide multiple views per actor role to illustrate the real-time operations of AVs and project KPIs. This includes map-based real-time view of SHOW's vehicle positions (upon integration and data availability).	Performance
A43-04	SHOW Dashboard should be able to analyze telemetry messages and generate alerts.	Performance
A43-05	SHOW Dashboard should be able to retrieve KPI data in batch mode or near real-time mode.	Performance
A43-06	SHOW Dashboard should be able to define geofences and related alerts	Performance
A43-07	SHOW Dashboard should support a map-based view of vehicle fleets and real-time positions, sensor details of the connected vehicles	Performance
A43-08	SHOW Dashboard should support multi-layer view of retrieved information (e.g. weather, traffic condition, external PT fleets...).	Performance
A43-09	SHOW Dashboard should support delegations of onboarding new user processes (e.g. site leader can invite site member users).	Performance
A43-10	SHOW Dashboard should support view on an aggregated level (SHOW project level) or site level of the KPI data. Interested and authorized users should be able to get the summarized view of important KPIs to allow reliable cross view and comparison.	Performance

Req ID	Description	Type
A43-11	SHOW Dashboard should have role-based user management. Keep and maintain user profiles. Access to different resources is based on role-based privileges. The anticipated roles and users will be in sync with the SHOW Cloud platform if possible. Users will be grouped by sites, partners, OEM, and project team role.	Security
A43-12	SHOW Dashboard should secure site data to only authorized users.	Security
A43-13	Secure and private subscription of all connected Things to SHOW Dashboard shall be managed via authentication, de-anonymization, and other means.	Security
A43-14	SHOW Dashboard should be able to retrieve KPI and related data from SMDP and other 3rd party systems (if needed/existing) via data API interfaces (MQTT and REST).	Communication
A43-15	SHOW Dashboard should have internal API management tools that help developers to onboard new API (Swagger specification), govern API usage, deploy and coordinate API lifecycle, and additionally allow users to access site KPI / geospatial data onboarded on the API Manager.	Communication
A43-16	SHOW Dashboard should support data owners to establish data API to external users up on mutual agreed data security schemes.	Communication

3.2 SHOW Dashboard functionalities

The SHOW Dashboard is used (in conjunction with/without local dashboards) to support demo sites monitoring the performance of their pilots within the SHOW project.

The Dashboard has two main groups of functionalities: (i) performance monitoring and visualization of SHOW pilot runs, and (ii) optional/experimental functionalities[2].

3.2.1 Performance monitoring and visualization

SHOW Dashboard is served as the Dashboard for the SHOW project to monitor the pilot runs at SHOW sites. The following functionalities are available for this purpose:

[2] Not planned to be used for operations by SHOW sites, but available for explorative experiments.

- Provision of SHOW automated vehicles, assets, actors, and relationships. This functionality is a lightweight site/fleet/vehicle management to register site's metadata and manage entity identities for the integration with SHOW SMDP.
- Collect[3] and visualize in real-time[4] data of vehicles and (potentially) connected traffic infrastructure objects.
- Data visualization: Dashboard with multiple views per actor roles to illustrate the real-time operations of AVs and project KPIs. This includes map-based real-time view of SHOW's vehicle positions/sensor data and the project's KPI measures (upon integration and data availability).
- Data collectors: Real-time MQTT based telemetry data and REST messages containing KPI data will be collected from the data interfaces with SHOW SMDP. Similar interfaces are also be used for site local data collector to publish their data to the SHOW SMDP.
- Dashboard customization ability: Full customization options are only available for the developer team. Dashboard users can select themes, KPI gadget view options, hide/unhide KPI panel.

3.2.2 Experimental/optional functionalities

The following functionalities are available to be explored for research initiatives within SHOW project (not intended for site operations):

- Analyse telemetry messages and generate realtime alerts: Realtime rule engine working with telemetry data from the MQTT broker. This function is used for geofencing alert notifications.
- Workflow with life-cycle events: Internal mechanism.
- Data API for external systems: Data owners who are Innovation Cloud users can develop data API to encapsulate and share their data to external parties.
- Geofencing Alert Notifications: Geofences are digitally marked geographical areas/fences that can be marked on any digital tool to symbolize a special area such as "speed limit zone", "green zone", "emission-free zone", etc. The dashboard allows users to create digital geofences. Consequently the system monitors the movement of vehicles in these areas and generates relevant notifications based on its location activity. Notifications are visualized in Dashboard interface and also available via MQTT broker message subscription.
- VRU Protection Notifications: Vulnerable Road Users (VRUs) are digitally synchronized pedestrians/road workers/ cyclists, etc. who share their location activity with the SHOW Dashboard application. Protection notifications for such users are generated by the system whenever there is some pre-defined vehicle activity in their vicinity. This notification can be used to provide alerts to the road workers / pedestrians about e.g. an approaching vehicle.

The dashboard and its components can serve as inspiration and best practice reference for the designs of local dashboards and multi-site/multi-view dashboard (e.g. at European or national transport authority level).

3.3 SHOW Dashboard in the overall architecture

The SHOW Dashboard is a component of the bigger SHOW centralized system. It is described as part of the reference architecture provided in deliverable D4.1[2], and independent from the architecture variations. Figure 9 illustrates the SHOW

[3] Via SHOW SMDP
[4] Via MQTT protocol

Dashboard's position (with thick purple bounding box) in the SHOW reference architecture. The reference architecture enforces no direct input data connection from site's local systems. However, during the interim period (up to M21), to support Dashboard PoC developments and testing, two temporary data interfaces with Gothenburg and Linköping (Sweden mega-site) sites have been maintained (unidirectional from local data collectors to the Dashboard). The temporary data connections are using the same data APIs interfaces between SHOW Dashboard and SHOW SMDP.

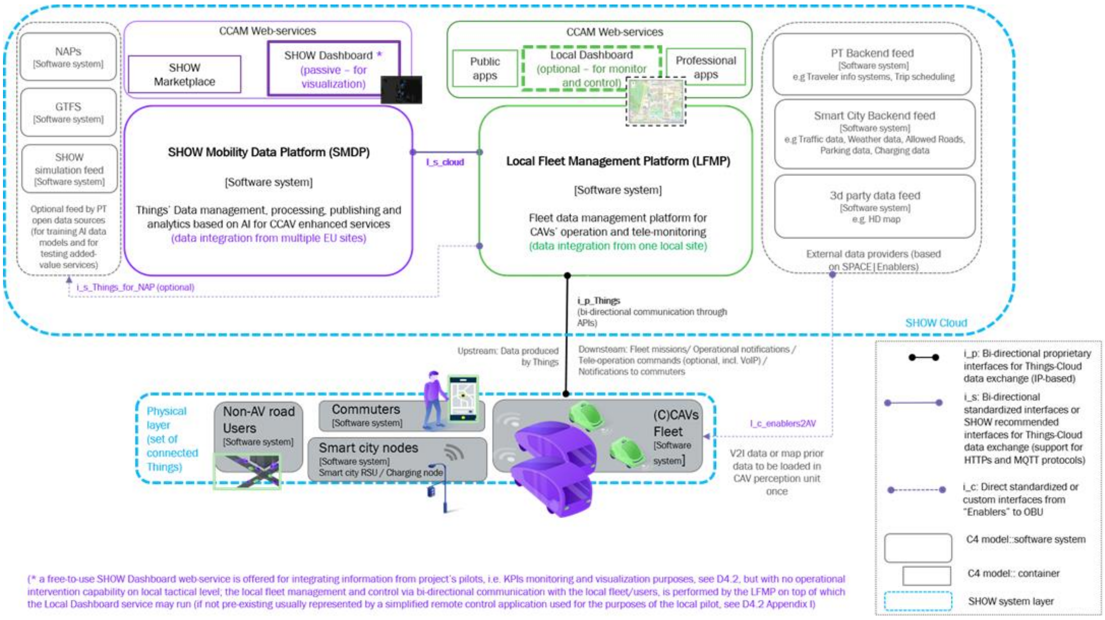


Figure 9: SHOW Dashboard in the SHOW reference architecture.

There is a need for standardization of interfaces and data formats to avoid unnecessary site-specific customizations and ensure platform compatibility.

The SHOW Dashboard team has developed prototypes for the interfaces and data format that should be used for the interaction between SHOW SMDP and the Dashboard. It has also been proposed that these definitions and specifications should be used also for the interfaces between all sites and the SHOW SMDP (except the difference in selected data transmission protocols. E.g., SHOW SMDP will accommodate also offline integration of site's data while SHOW Dashboard does not).

In Figure 10 these interfaces are represented by the yellow boxes.

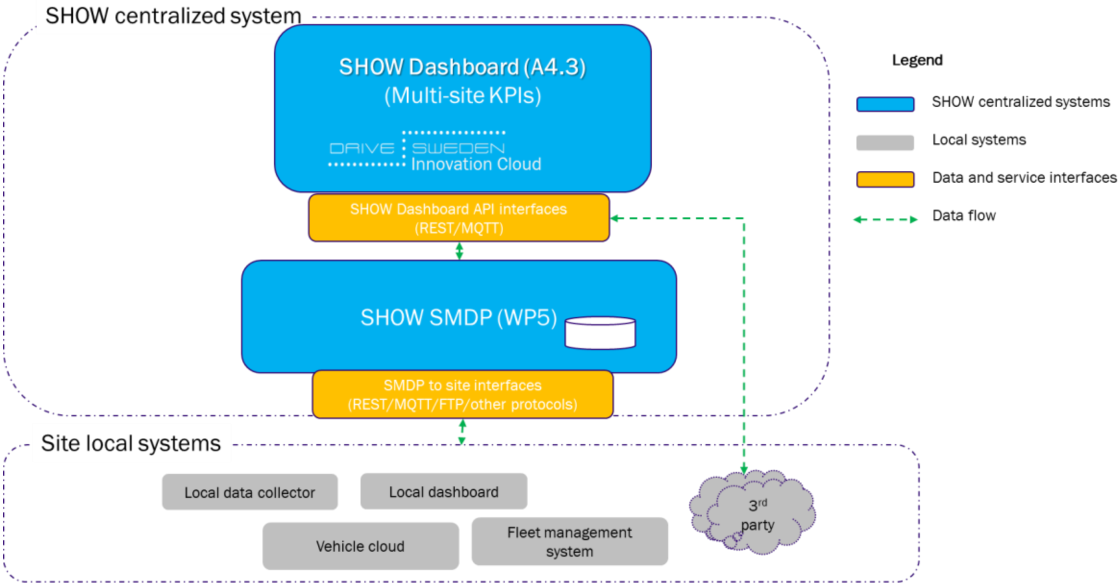


Figure 10: SHOW Dashboard integration diagram.

The SHOW Dashboard has been developed on top of the Ericsson Innovation Could platform (EIC). The EIC platform is essentially a data-brokering platform developed as a sandbox-environment for innovation projects in a multi-partner environment. The EIC Platform inherently provides data-brokering capabilities between different actors along with features such as Authentication, Authorization and Accountability (AAA) for the data that is being brokered. Additionally, the EIC Platform offers the features of User Onboarding, Data Onboarding and Data Consumption in a self-serviced manner. These features make it an ideal choice for supporting dashboards for automotive trials such as the SHOW Dashboard where data exchange and authorization are key elements in a multi-partner infrastructure.

3.4 Functional scope and recommended usage

3.4.1 Functional scope of SHOW Dashboard:

- ***A project dashboard for SHOW***: The Dashboard provides visualization and monitoring tool on the project level, offering a pan-European overview of the SHOW fleets for demonstrating in a coherent way the final results of the project[5]. It is designed as a non-operational dashboard in the overall architecture, as featured in Figure 9, and does not allow operational "remote-control" role/functionality. The motivation behind this was to offer a mature visualization tool[6] mainly for presenting project information[7] specially caring for SHOW sites which did not plan to have local Dashboards integrated as part of the local fleet management platforms (hence visualization of local information would not be possible). This functionality will support SHOW project board and demo site leaders to monitor pilot phases. Following functionalities are available:
    - Integrated condition monitoring of automated vehicles within SHOW pilot sites.

[5] Offering screens for data digestion and KPIs visualization that are activated for each site if selected

[6] The platform is already used in Sweden for smart mobility research thus its early instantiation for SHOW purposes is more feasible.

[7] end-to-end data pipeline

- A lightweight site/fleet/vehicle management to register site's metadata and manage entity identities for the integration with SHOW SMDP.
- Map-based dashboards visualizing the latest updated KPIs as defined within SHOW, with two interfaces (i) reference site dashboard for detailed view at specific pilot site and (ii) global dashboard overview of all SHOW pilot sites.
- Integration interfaces with SHOW SMDP to collect (i) KPI data and (ii) realtime data upon the data availability and accessibility.
- Onboarding function to support user account registration and management.
- Geofencing based real-time alerts capability: Allow site users to define geofences and corresponding real-time alerts on the Dashboard. The triggered alerts are published to the Dashboard realtime message broker and available for the site local systems. However, this feature is experimental and should only be used for research purposes. No access/intervention from the Dashboard to any local site systems is allowed, tactical and operational decisions are performed by site operational systems (e.g. fleet management systems at local sites).
- The consolidated view of site registration data, geospatial data and KPI information are useful for strategical decisions.

- ***A reference design for multi-site dashboard***: the dashboard and its components can serve as inspiration and best practice reference for the designs of local dashboards and multi-site/multi-view dashboard (e.g. at European or national transport authority level).

Out of scope: SHOW Dashboard does not address tactical level. This role resides on the local Fleet Management platforms (that may or may not support a custom Dashboard service) and that can be also totally different in terms of functionality in each site

3.4.2 Recommended usage of the reference SHOW Dashboard

The below Table 4 describes recommendations on the usage of SHOW Dashboard for different actor types.

Table 4: Recommended usages of the SHOW Dashboard.

Actor	Recommended usage
Project demo sites boards	Centralized monitoring of SHOW pilot phases (in both real-time and offline modes)
Project technical work packages	Supporting the end-to-end site-agnostic reference PoC specification by providing an end-point web application where local fleet data and KPIs are visualized (helps in verification of data model, data interfaces/integration, KPI definitions/visualization paradigms)
Project local site partners	- Offering a free-to-use SHOW Dashboard web-service for integrating and visualizing information from project's pilots, i.e. KPIs monitoring and visualization (especially useful when a local Dashboard functionality is missing from the local LFMP) - Supporting end-to-end integration of local data sources with SHOW SMDP (KPI and optionally

Actor	Recommended usage
	realtime data): Relevant data in SHOW SMDP will be visible in SHOW Dashboard automatically. - Offering a best practice reference implementation as a guideline for the designs of local dashboards - Offering a small set of optional experimental features that can be implemented by local LFMP developers for exploring innovative new features (e.g. alerts to connected VRUs)
Future AV PT operators	A reference design for multi-site dashboard: the dashboard and its components can serve as inspiration and best practice reference for the design of a multi-site/multi-view dashboard (e.g. at European or national transport authority level deploying one central Dashboard for multi-location-based fleets).

NOTE: There is no conflict in having both SHOW Dashboard and local dashboard running in parallel as their objectives are disentangled. The SHOW Dashboard is designed as a passive dashboard used for performance monitoring and visualization purposes and, as such, no operational intervention to site operation is allowed. The local fleet management and control via bi-directional communication with the local fleet/users, is performed by the LFMP on top of which the Local Dashboard service may run (if not pre-existing usually represented by a simplified remote control application used for the purposes of the local pilot, see D4.2 Appendix I).

3.5 SHOW Dashboard Component diagram

The component diagram of SHOW Dashboard (using Context, Containers, Components, and Code (C4) nested model [18]) is described in Figure 11. The system interfaces to external systems via Data sink and Data source layers. The API Server and API Manager components manage all the data and other microservice APIs. Telemetry Data Processor/Server are the components to perform real-time assessments of telemetry data collected from the vehicles with pre-defined rules, and generates corresponding real-time alerts (e.g. Geofencing violations).

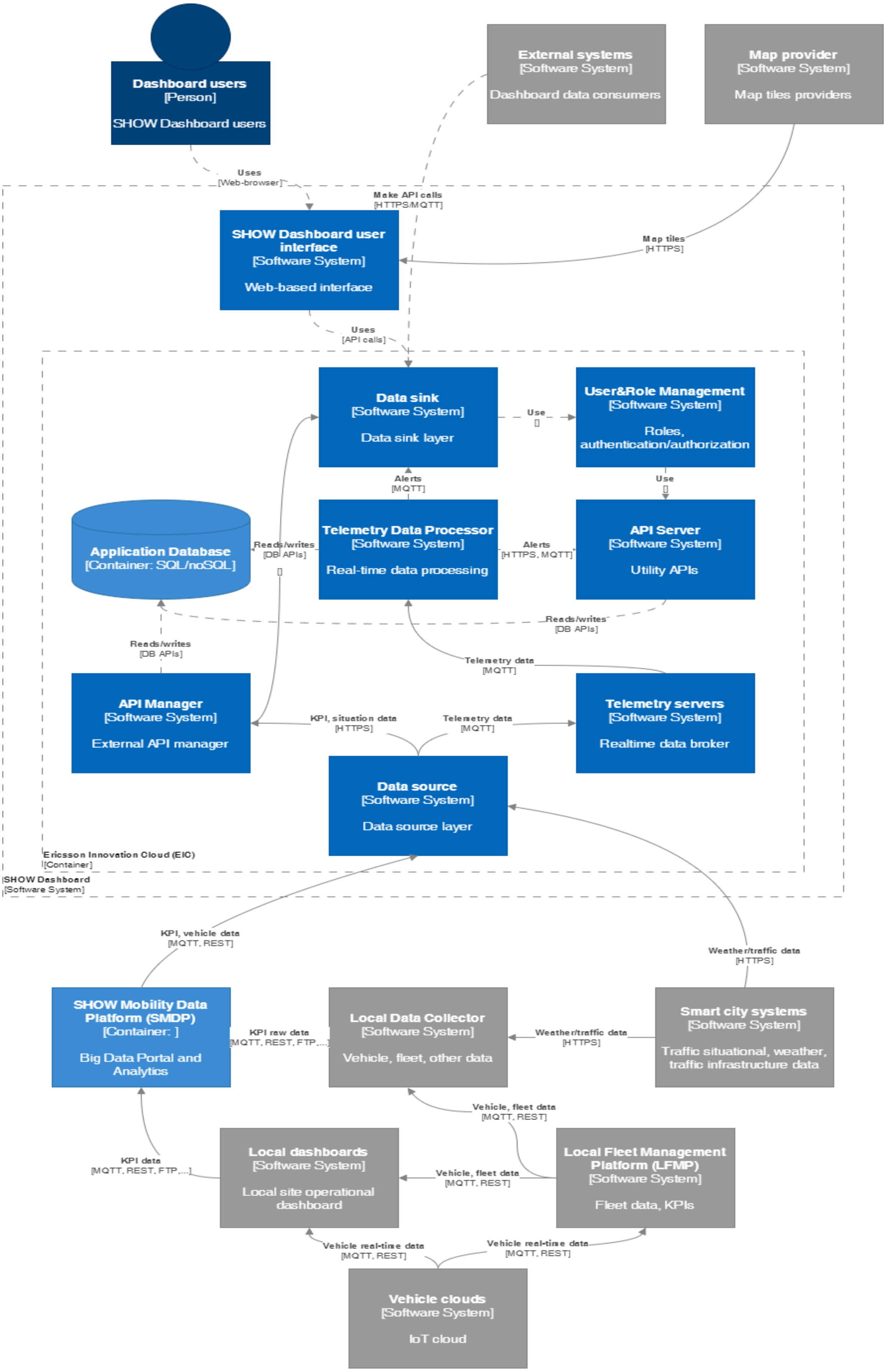


Figure 11: SHOW Dashboard architecture diagram (Component level).

3.6 SHOW Dashboard Component descriptions

The SHOW Dashboard is built on microservice architecture and conceptually grouped into different components as depicted in Figure 11. Descriptions of these components are provided in Table 5.

Table 5: SHOW Dashboard component descriptions.

Component	Description
SHOW Dashboard user interface	Web-based Dashboard interface to the user with SHOW's KPI gadgets including map-based multi-layer real-time visualization of vehicle/fleet geospatial positions. Each layer represents a specific group of objects to be visualized on the map gadget. Depending on the integration and data availability, traffic situations can also be configured as a layer.
Data sink	Broker/gateway service to communicate with Dashboard user interface and Dashboard external data API for external parties to access the geospatial KPIs upon some security schemes. The data-sink services rely on the User & Role Management and API Manager for supporting secure access to the API data.
Data source	Broker/gateway service to facilitate communication with data sources e.g. SHOW SMDP cloud platform, vehicles/sites/IoT devices in both synchronous (HTTP/TLS) and asynchronous (MQTT) modes, and possibly conversion between these modes. This component is used to collect both raw and aggregated data from sites for KPIs.
Application Database	Scalable and secure storage to store all required data needed for Dashboard, such as collected vehicle data (including metadata and payloads) and aggregated KPI data retrieved from SMDP.
Telemetry Data Processor	Application micro-service to facilitate transformation and broadcast of real-time messages (MQTT-based payloads) into a format that is accessible and usable in the different micro-services deployed as a part of SHOW Dashboard Application.
Telemetry Server	Application Server to facilitate the exchange of telemetry data between vehicles/sites/IoT devices with the SHOW Dashboard micro-services. For example, if a traffic system intends to share its traffic notifications to the SHOW Dashboard directly, then, in such cases, the Telemetry Server can be the single point of contact that the traffic systems can subscribe to.
API Manager	Provide management tools that help developers to onboard new API (Swagger specification), govern API usage, deploy and coordinate API lifecycle, and additionally allow users to access site KPI / geospatial data onboarded on the API Manager. This application allows the creation of 'data subscription workflows for users to easily connect to for fetching data out of the SHOW Dashboard.
API Server	An API gateway/broker server that receives/orchestrates API requests, enforces security policies, routes requests to the related services, and handles responses to requesters.
User & Role Management	Keep and maintain user profiles. Access to different resources is based on role-based privileges. The anticipated roles and users are in sync with the SHOW SMDP platform. Users are grouped by sites, partners, OEM, and project team roles.

3.7 SHOW Dashboard interfaces

SHOW Dashboard has the following integration interfaces:

- Input data API interfaces: Collect site raw and aggregated data from SHOW's SMDP platform with the following protocols:
    - HTTP/TLS (KPI related data)
    - MQTT (realtime data)
- Output data API interfaces: Dashboard generated realtime alerts can be retrieved from a topic in the MQTT broker for authorized systems.

SHOW Dashboard connects to SMDP to collect aggregated and/or raw realtime and KPI data from the sites.

Table 6: Data sources interfaces.

Interface	Source system	Description
Vehicle/IoT telemetry	SMDP	MQTT interface <fleet-id>/<vehicle-id>/<SMDP-attribute>
Map	Map providers such as Mapbox	Map tiles, map objects (traffic road network…)
Traffic situation	Smart city systems	REST API. Weather, traffic situation (roadworks, accidents…)
Trip information	SMDP	REST API
KPI information	SMDP	REST API

4 SHOW Dashboard integration and development

4.1 SHOW Dashboard development methodology

The development of SHOW Dashboard adopts the Proof of Concept (PoC) agile development methodology depicted in Figure 12.

In the **Init** phase, the core team has been set up consisting of key personnel: solution architect, product owner, dashboard, data expert and site team.

In **Explore** phase, the team investigates existing initiatives, literature, and standards, designs the first proof of concept (PoC) instance of a central Dashboard with some basic components from the Innovation Cloud platform. The requirement specification is collected through a series of site surveys, workshop series with "A4.1- System architecture and tools" and consulting the best practices from Drive Sweden[4] research projects. This phase resulted in the first release of SHOW Dashboard proof of concept instance. The instance is used as initial baseline.

The third phase **realizes** the PoC development with 1-week sprints. The requirements are constantly updated every week in connection with the evolvement of KPI definition (WP9), and the updated information from the sites via site surveys and different project workshops (SP2, SP3).

Deploy phase is the last phase of this task, where the Dashboard is fully integrated with SMDP and has onboarded the first set of SHOW sites that are ready to connect (Linköping, Göteborg).

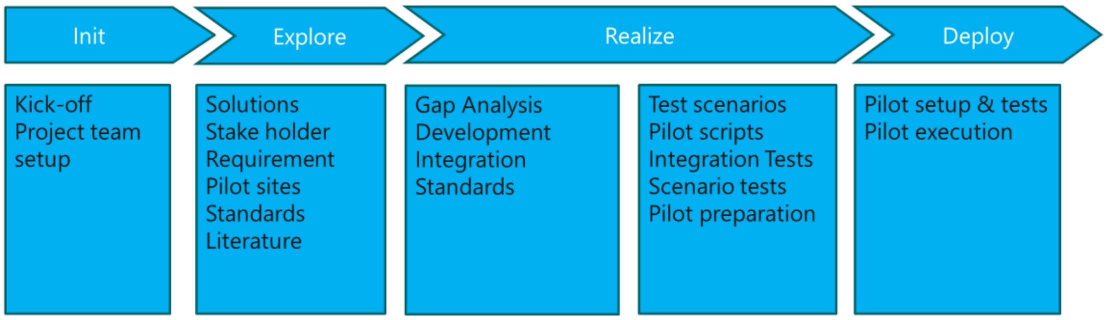


Figure 12: SHOW Dashboard PoC development methodology.

4.2 Ericsson Innovation Cloud

The SHOW Dashboard is built based on Ericsson Innovation Cloud (EIC[3]) platform. This section provides an overview of the platform. Readers can explore more about the platform via its URL https://drivesweden.innovationcloud.ericsson.net/home

4.2.1 Introduction

The EIC is tailored-made for business-to-business (B2B) data and insights sharing. It allows organizations to connect data streams through APIs and trade it with other organizations across the globe on their terms. The flip side is that it also allows organizations to consume and purchase data streams by subscribing to different sources from the marketplace. One can also choose to become an insight creator by subscribing to one or more data streams, adding intelligence and connect the enriched data back to the platform as insights to be traded by other organizations at one's terms. The data lifecycle managed by the platform is briefly described in Figure 13.

The EIC Platform inherently provides data-brokering capabilities between different actors along with features such as Authentication, Authorization and Accountability (AAA) for the data that is being brokered. Additionally, the EIC Platform offers the features of User Onboarding, Data Onboarding and Data Consumption in a self-serviced manner. These features make it an ideal choice for supporting dashboards for automotive trials such as the SHOW Dashboard where data exchange and authorization are key elements in a multi-partner infrastructure.

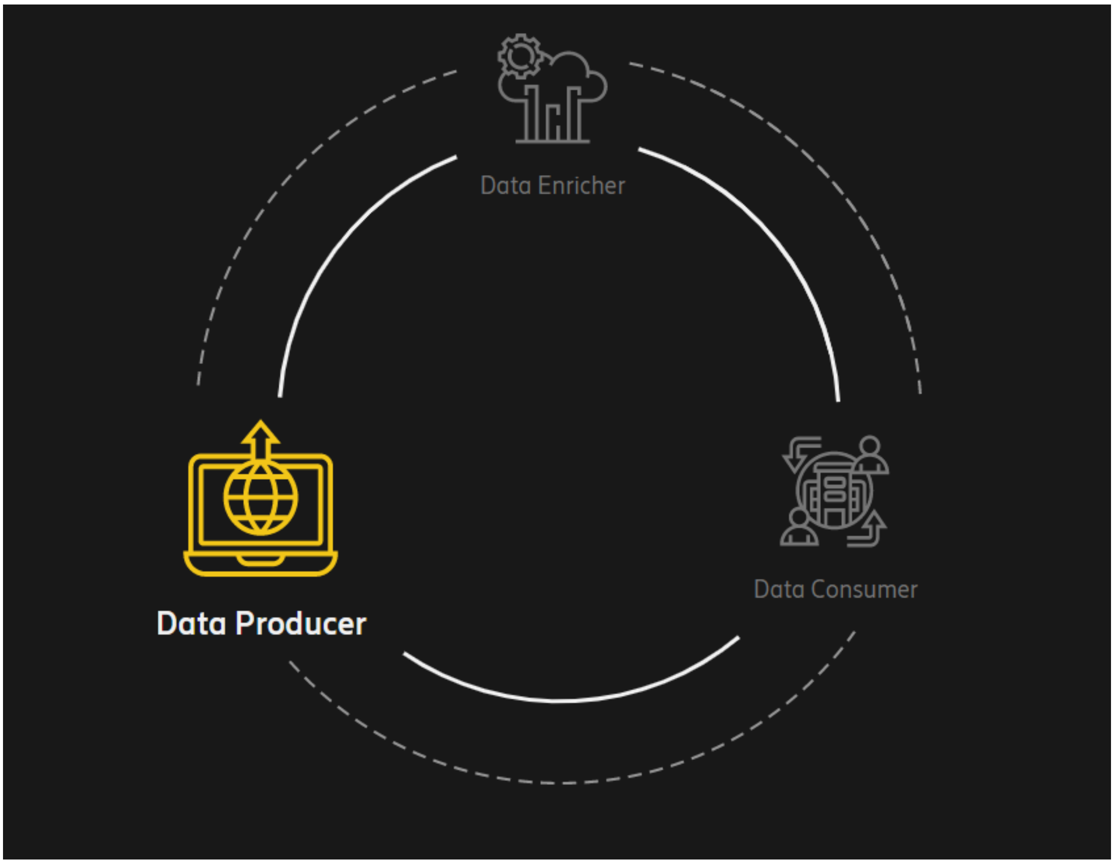


Figure 13: Innovation Cloud (EIC) data lifecycle [3].

4.2.2 Innovation Cloud Model

The platform provides continuous integration/continuous deployment flow service (CI/CD) which is vital for configuration management and quality assurance of the SHOW Dashboard PoC sprint developments. Figure 14 depicts the innovation cloud model as a general conceptual architecture of the platform. The model allows different actors involved in the data lifecycle while still ensuring data security and mutual data sharing agreements.

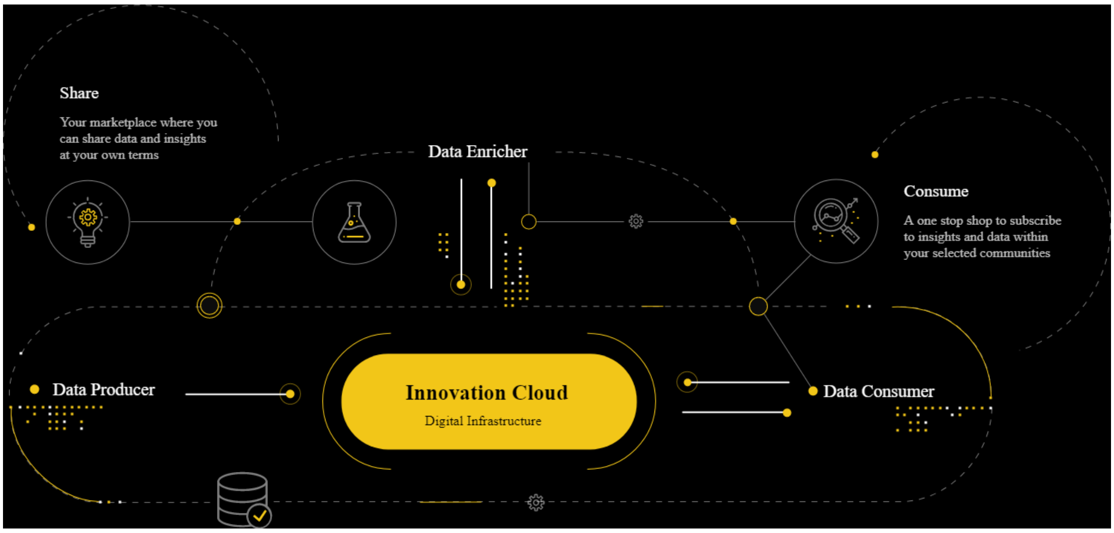


Figure 14: Innovation cloud (EIC) model [3].

4.3 Dashboard PoC instance and the encapsulated components

The Dashboard PoC has been implemented as an instance of the EIC platform, available at the URL https://demo.innovationcloud.ericsson.net/show-dashboard , with the following encapsulated components:

- Data sink/data source: Data broker and APIs to collect and expose data from and to the platform.
- Telemetry data processor: Realtime rule-engine, data management and processing.
- Application database: hosting telemetry and KPI data collected from SMDP.
- API management: API server and API manager that maintain internal and external APIs.
- Map view: Map-based view with Mapbox tiles.
- Authentication: EIC user authentication servers are used to authenticate users.
- User/Role management: Site based user management is implemented in SHOW Dashboard. To access a site the following prerequisites must be satisfied:
    - User must be onboarded on Ericsson Innovation Cloud
    - User should have subscription to SHOW Dashboard
    - User must have access to site.
- Onboarding: A custom self-serviced workflow has been implemented in the dashboard which, an authorized user can use to onboard new users to EIC. Along with sending an invitation, the inviter can specify the sites which the invitee can access after joining the platform. This workflow enables a smooth and automated feature to map new users to SHOW Dashboard and its Sites. Once the invitation is sent and the invitee signs up, the workflow executes the following automated steps:
    - After the new user successfully signs up to EIC, a subscription is created for the user on SHOW Dashboard platform. The user is now authorized to access SHOW Dashboard.
    - The user is provided access to all the sites that the inviter had selected while sending the invitation.

Table 7: Existing and newly developed components.

Functions	Existing component	New developments
Map view	Mapbox basic functionalities	- Global and Site Dashboard interfaces with access to map functionalities - SHOW specific layers representing sites, vehicles, routes, geofences, VRUs.
User onboarding	Direct onboarding workflow for Ericsson Innovation Cloud through SHOW Dashboard	- Self-onboarding to SHOW Dashboard - SHOW sites and user hierarchy - Site access for users while sending invitations
Realtime alerts	Rule engines	- Alert rules for geofences - Alert rules for VRUs - Alert rules for vehicles - Integration to SHOW Dashboard - Alerts are available at MQTT broker
KPI view	KPI gadgets	- 78 KPI gadgets - Integration with SHOW SMDP to collect KPI data. - Displayed KPIs can be mocked up if data is not available. - Show latest update timestamp
Integration	Message broker foundation	- MQTT message broker for near real-time data visualization and digital twin - REST interfaces for batch-processed data
UI		- User interfaces - Global Dashboard for visualization of all sites - KPI visualization components - Digital twin creation of connected devices and their visualization on map - Onboarding and modification of Fleets and Vehicles

Functions	Existing component	New developments
Database	Database	- Design and develop data schemes, data entity relationship and storage of entity details and generated alerts
User/Role	Basic functionalities	- Setup user/role hierarchy - Site Owner – can create, modify or delete all entities in a site. Can authorize other users to access a site. - Users with new user onboarding privilege – can invite other users to the platform
Authentication and authorization	JWT based authentication	- Implementation of user authentication - User authorization for access to Sites

4.4 SHOW Dashboard Users

The SHOW Dashboard provides role-based access management, with pre-defined roles and suggested users as tabulated in Table 8. User onboarding for a site starts with an agreement between a site's nominated leader and SHOW Dashboard platform technical leader (Ericsson). He/she will be granted an account with privileges to register all site related metadata (site, fleet, vehicle, geofences...) and invite other users to access their site dashboard instance.

A public layer of SHOW Dashboard containing publicly accessible KPIs information will be available for public access through SHOW project website [19] as dissemination material. This function will be made available in M21 (Oct 2021).

Table 8: SHOW Dashboard users and authorizations

Roles	Partners	Authorization	Functionality
Demo site leader	Nominated core group of a demo site, can be from different site's partners.	• Requires EIC User account • Requires SHOW Dashboard Subscription	• Can view site dashboard; • Can modify, or delete fleets and connected devices in a site; • Can authorize other users to access the site.

Roles	Partners	Authorization	Functionality
Site User	Invited by Demo site leader. Recommendation: Partners from the related pilot sites, and SHOW project core group involved in WP11, WP12.	• Requires SHOW Dashboard Subscription • Requires authorization to access a site	• Can view site dashboard
Fleet User	Site AV operators	• Requires SHOW Dashboard Subscription • Requires site authorization	• Can view site dashboard; • Can modify, or delete fleet and connected devices in a fleet.
Demo site IT lead	User with invitation privilege selected by Demo site leader to onboard new site users.	• Requires SHOW Dashboard Subscription • Requires site authorization • Requires User Invitation Privilege	• Can view site dashboard; • Can invite new users to the platform; • Can provide site authorization to invitees.
Public	Public	• Not Required	• Can view a public layer of SHOW Dashboard.

4.5 SHOW Dashboard User interfaces

The Dashboard provides the following main user interfaces to support authenticated and authorized users accessing related functionalities and data:

a. **Global Dashboard:** This component consists of a map section and a KPI section. Restricted access to this component will only provide to authorized users appointed by the SHOW project. The aggregated data shown in this component are subjects to mutual data access agreements between the site data owners and the project board.
    i. **Map section:** All Sites are represented on this map using custom markers. Markers can be clicked to enter into the specific site dashboard.
    ii. **KPI section:** Consists of the list of sites and the publicly available and applicable KPIs for each selected site.

b. **Site Dashboard[8]:** This component consists of KPI panel and Site map. Users can use the Fleet dropdown to select a fleet to view. Authorized users to this component are site partners that have mutual data access agreements with the site data owners. The site dashboard can be accessed from the global dashboard view by navigating through the site location markers.
    i. **KPI Panel:** All KPIs classified based on their category can be visualized in this section. The KPIs here represent the average values for the whole fleet that the user has selected.
    ii. **Site Map:** This component consists of a vehicle panel and a map.
        - Vehicle Panel: All the vehicles in a selected fleet can be visualized here as individual cards. Expanding each card, a user can view the last updated vehicle details, vehicle sensor data and vehicle network data.
        - Map: This component renders all the active Mapbox map layers i.e., Vehicle & VRU Digital Twins, Routes, Stops & Geofences.

c. **Geofence Management**: This component is used to manage all geofences of all authorized sites. Authorized users can create, view, edit and delete the geofences of individual sites.

d. **Site Management**: Authorized users can manage all entities of a site from this section. A user can view all sites, create, edit or delete fleets, vehicles, VRU clusters and VRUs from this section. Selected authorized users can use this function to invite other users to register to SHOW Dashboard.

Detailed implementation of these key user interfaces will be provided in related sections 5.2, 5.3, 5.4, and 5.5

4.6 SHOW Dashboard data

The data for SHOW Dashboard includes the following main data entities:

- Site registration metadata: Site location, geospatial data, fleets, routes, vehicles, geofences.
- Vehicle data
    - Static data: vehicle specification
    - Dynamic data: Operational data collected with some update frequency
- KPI data (as defined by A9.4 and realized into the Data Registry of WP5)
- Environmental data: Data collected from 3rd party sites/services including weather, traffic situations.

4.6.1 Site metadata

The site metadata consists of some basic site, fleet, and vehicle information. That includes geographical locations, properties of each site objects (site, fleets, vehicles) and the object relationship. Detailed implementation is provided later in section 5.5.1.

In SHOW Dashboard, site metadata also involves user relationship mapping. Every site maintains a site-user mapping. This site-user mapping enables us to introduce authorization and accountability for all sites and their entities. Only authorized users can access, modify, or delete a site. Only authorized users can provide other users with site access.

[8] Readers please note the difference: The terms Site dashboard refers to the site specific views within SHOW Dashboard, while Local dashboard refers to local dashboard solutions at a site.

4.6.2 Vehicle data

We derived a common vehicle data model to support unified communication interfaces between the site data collectors, SMDP, and the SHOW Dashboard. The data model has been derived as a result of the literature review of public transportation standards (2.2) and the data requirements resulted from the project KPIs. We extended the standards with supports of automated vehicle capabilities and communication software and hardware. The data model consists of two key parts – vehicle metadata (a.k.a. static data) and vehicle operational data (a.k.a. dynamic data) received during the operation of the vehicle. The vehicle data model is structured as in the below Table 9.

Table 9: Vehicle data model.

Category	Attribute	Description
Static data		Technical specification data registered for each vehicle.
	Vehicle identity	Data which is used to identify the vehicle.
	Intelligent vehicle and infrastructure	Information about hardware and software related to autonomous operation and communication between the vehicles and intelligent traffic infrastructure.
	Vehicle payload	Cargo and passenger capacity.
	Vehicle environmental footprint	Description of fuel types and consumptions.
	Vehicle equipment	Equipment and sensors belonging to the vehicle.
	Vehicle metadata	Vehicle manufacturer's specification.
Dynamic data		Operational data consists of dynamic data received during the operation of the vehicle, exported from different vehicle sensors. Usually received with near real-time protocols (such as MQTT).
	Identity/context	Timestamp of data, vehicle ID and fleet ID.
	Position	GPS and GNSS position of the vehicle.
	Intelligent Vehicle and Infrastructure	Current operational values for the different driving modes as well as other related equipment states depending on the equipment.

Category	Attribute	Description
	Trip	Trip, route, stations/stops as well as estimated time of arrival if known.
	Environmental footprint	Currently used fuel consumption, and emissions.
	Vehicle payload	Number of passengers, occupancy, payload or both.
	Vehicle equipment state	Data received from sensors, actuators and network equipment.

4.6.3 KPI Query Data Structures

Beside the categorical structure of the KPIs, the data structure displayed in the dashboard can be viewed based, on object and time. Some KPIs can also be viewed based on mode and service.

Object - The objects in the data structure is the vehicles and fleets of vehicles and sites monitored in the SHOW project. Possible objects:

- Vehicle – the KPI value for a specific vehicle.
- Fleet – the KPI value for a fleet of vehicles.
- Site – the KPI value for all fleets and vehicles in a site.

Time – The time for the displayed value can be a specific time point or a series of values. In both cases the value is an average based on time step frequency. Possible times series frequency (Figure 15):

- Daily – average value for every day.
- Weekly – average value for every week.
- Monthly – average value for every month.

```
dataFrequency  string
Enum:
[ second, minute, daily, weekly, monthly, pre-demo, demo ]
```

Figure 15: Data frequencies.

Measurement units – The measurement units for the KPI are depending on the specific KPI, selected object and time constraint. Example on measurement units in different KPI classes:

- Road Safety - Events/km, Conflicts/km, Accidents/km, m^2/s^4
- Traffic – # of, km, Km/h
- Energy - kWh/km, Liter/km, J/km
- Environment - g/km, mcg/m^3, dB(A)
- Logistics - # of, km, minutes, hours
- Project Success - %, €, €/km; €/shipment; €/vehicle

```
KPIUnit  string
Enum:
[ accidentsPkm, conflictsPkm, eventsPkm, kmPh, m2Ps4, numberPkm, seconds, percent, minutes, vehicleHours, numVehiclePkm, kWhPkm, literPkm, jPkm, gPkm, mcgPm3, dB,
eurPkm, eurPtrip, likert9, eurPshipment, eurPvehicle, eur, number ]
```

Figure 16: Available measurement units of KPIs.

For a full and most updated list of KPIs and detailed definitions, readers can consult WP9's deliverable D9.2 [20] and the coming deliverable D9.3 [21].

4.7 SHOW Dashboard KPI implementation

4.7.1 KPI Broad category

The following KPI broad categories are visualized in SHOW Dashboard:

- "Road Safety",
- "Traffic, Energy, Environment"
- "Societal, employability and equity"
- "Logistics"

The two other broad categories "User Acceptance", "Project Success" which are not requiring frequent updates will be accommodated by the WP13 demonstration and evaluation results as well as the overall project evaluation that will reach their final values at the very end of the project.

4.7.2 List of KPIs

The Dashboard currently supports all 78 KPIs as specified in the KPI spreadsheet (task A9.4 [20]) as of M18. Updates to the KPI list and KPI definitions will be accommodated until M21 as in the current plan. The KPIs are defined over three dimensions:

- Time: second, minute, day, month, pre-demo, demo
- Object: vehicle, fleet, site
- Geospatial: route segment, route, geospatial area.

4.8 SHOW Dashboard API interfaces

The Dashboard platform provides both synchronous (HTTPS) and asynchronous (MQTT) interfaces for the SHOW SMDP, any external systems (such as traffic infrastructure) and any IoT connected devices to communicate with it. The high-level functional layout of the Dashboard depicting all the interactions with SHOW or external components was provided in D4.1 [2] and is also shown in Figure 17 below.

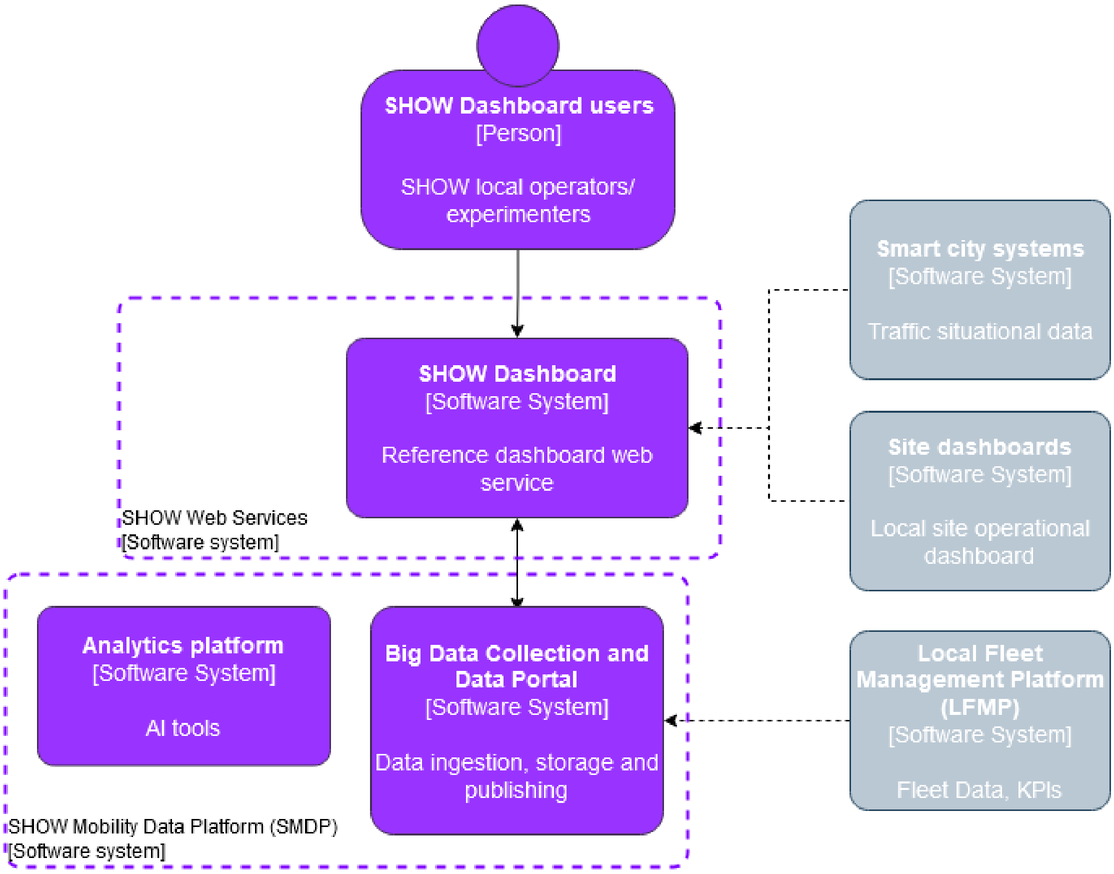


Figure 17: SHOW Dashboard component and its interfaces to external components/systems.

4.8.1 MQTT

MQTT interfaces are used for “real-time” data collection and processing. The interface is built in the form of MQTT, with a security mechanism to authenticate and authorize vehicles and/or site’s systems to publish and subscribe to MQTT topics.

The security mechanism also includes access tokens which a client can include in the MQTT requests. Only authorized users having access to a site, can publish data over the topics of that site on the SHOW MQTT Broker server. Messages published by unauthorized users are rejected by the broker.

The MQTT component also supports message payload parser to transform from one format to the others.

Currently, the interfaces are used to deal with the following data topics:

i. Location
ii. Speed
iii. Tire Pressure
iv. Mileage
v. Temperature
vi. Seat
vii. Vehicle Actuator Equipment
viii. Network

A detailed message payload format of each topic can be downloaded from Manage Vehicle section in Site Management in SHOW Dashboard.

4.8.2 REST API

REST API has been implemented in SHOW Dashboard API to facilitate data transmission related to SHOW entities which can be consumed by automotive applications to synchronise entity data. This helps automotive applications to fetch and sync details about the digital twins of real-world entities, implemented in SHOW Dashboard. Authentication has been implemented to secure these APIs.

SHOW APIs expose the data for the following entities:

- Sites
- Fleets
- Vehicles
- Vulnerable Road Users (VRUs)
- Geofences
- Routes

These APIs are used for performing CRUD (Create, Read, Update & Delete) operations in SHOW Dashboard. Relevant data is stored in MongoDB database. Dashboard retrieves all dependent data from DB using these REST APIs.

4.8.3 HTTP/MQTT proxy

The MQTT proxy server translates all MQTT messages into web-socket messages and forwards this real-time data over a proxy publisher-subscriber(pub-sub) web-socket server to the SHOW Dashboard. The pub-sub server also uses the REST APIs to store the latest received message in the database. Multiple dashboard clients connect with this pub-sub server and subscribe to relevant web-socket topics to consume data.

4.9 KPI query structure

The dashboard KPI message structure is mainly based on the object and time defined in Section 4.6.3. To get the KPI data the dashboard uses the REST-based queries described below.

The main request is defined on the objects:

- Get vehicle information.
- Get fleet information.
- Get site information.

Object request definitions

The object request definition is defined with in REST in path parameters show in OpenAPI [22] Swagger definition in Figure 18.

```
get - /vehicle_info/<site_id>/<fleet_id>/<vehicle_id>/<kpi_id>/

get - /fleet_info/<site_id>/<fleet_id>/<kpi_id>/

get - /site_info/<site_id>/<kpi_id>/
```

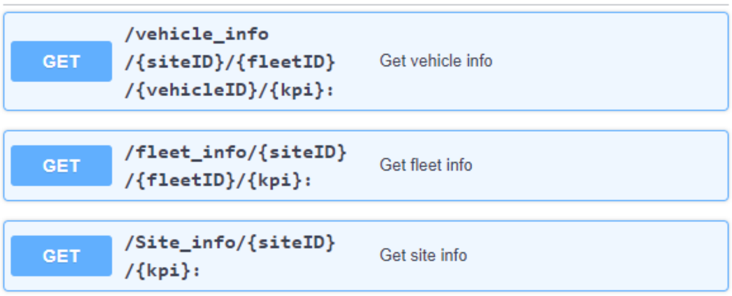


Figure 18: OpenAPI Swagger in path parameters.

Time request definitions

The time request definition is defined within REST in-query parameters show in OpenAPI Swagger definition in Figure 20.

```
/?time_from=<time_from>&time_to=<time_to>&frequency=<frequency>
```

Value types for the time parameters are:

- time_from – time in UTC format, example “2021-01-04T21:36:50Z”
- time_to – time in UTC format, example “2021-03-06T21:36:50Z”
- frequency – possible values, <daily / weekly / monthly>

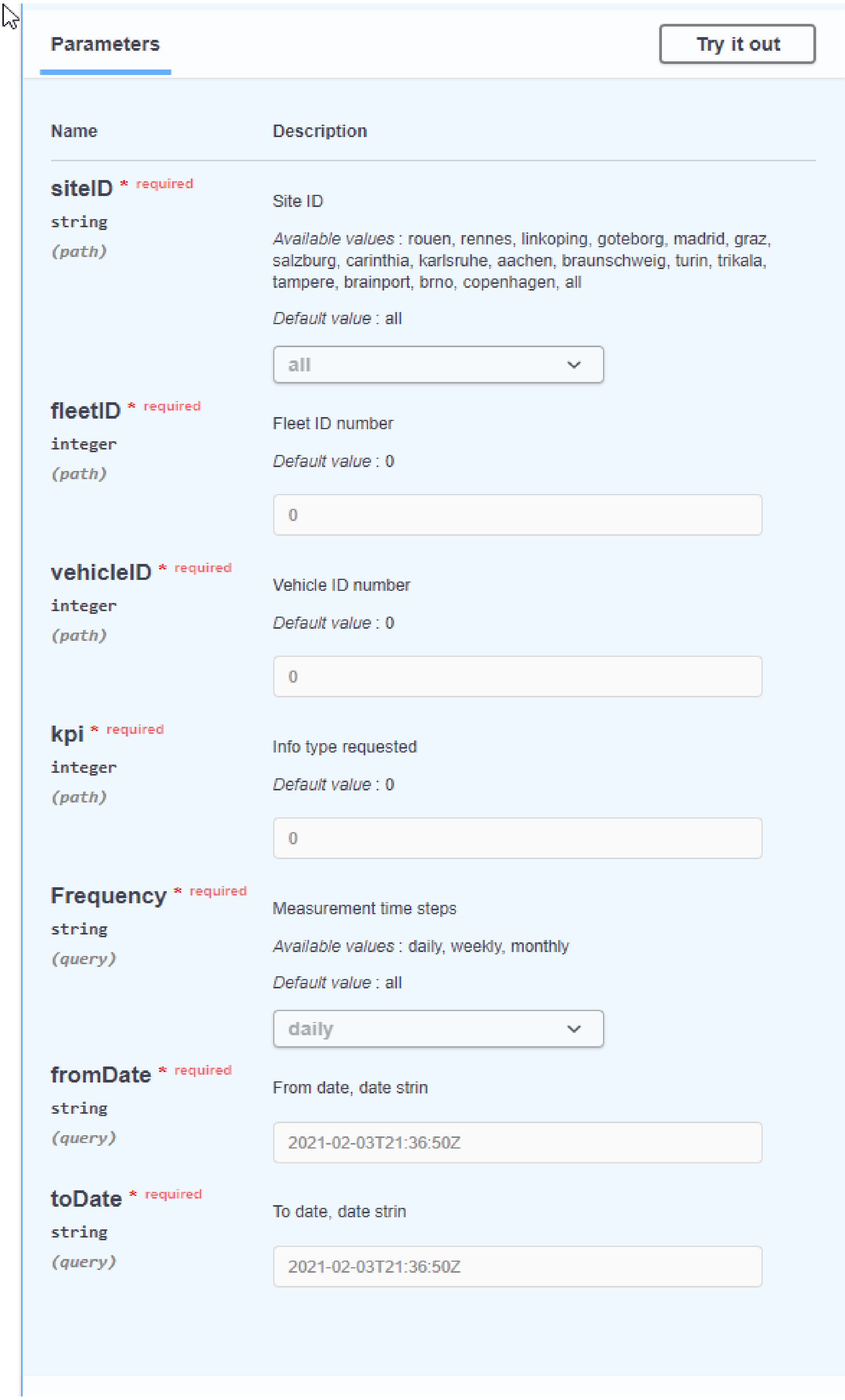


Figure 19: OpenAPI Swagger query parameters example.

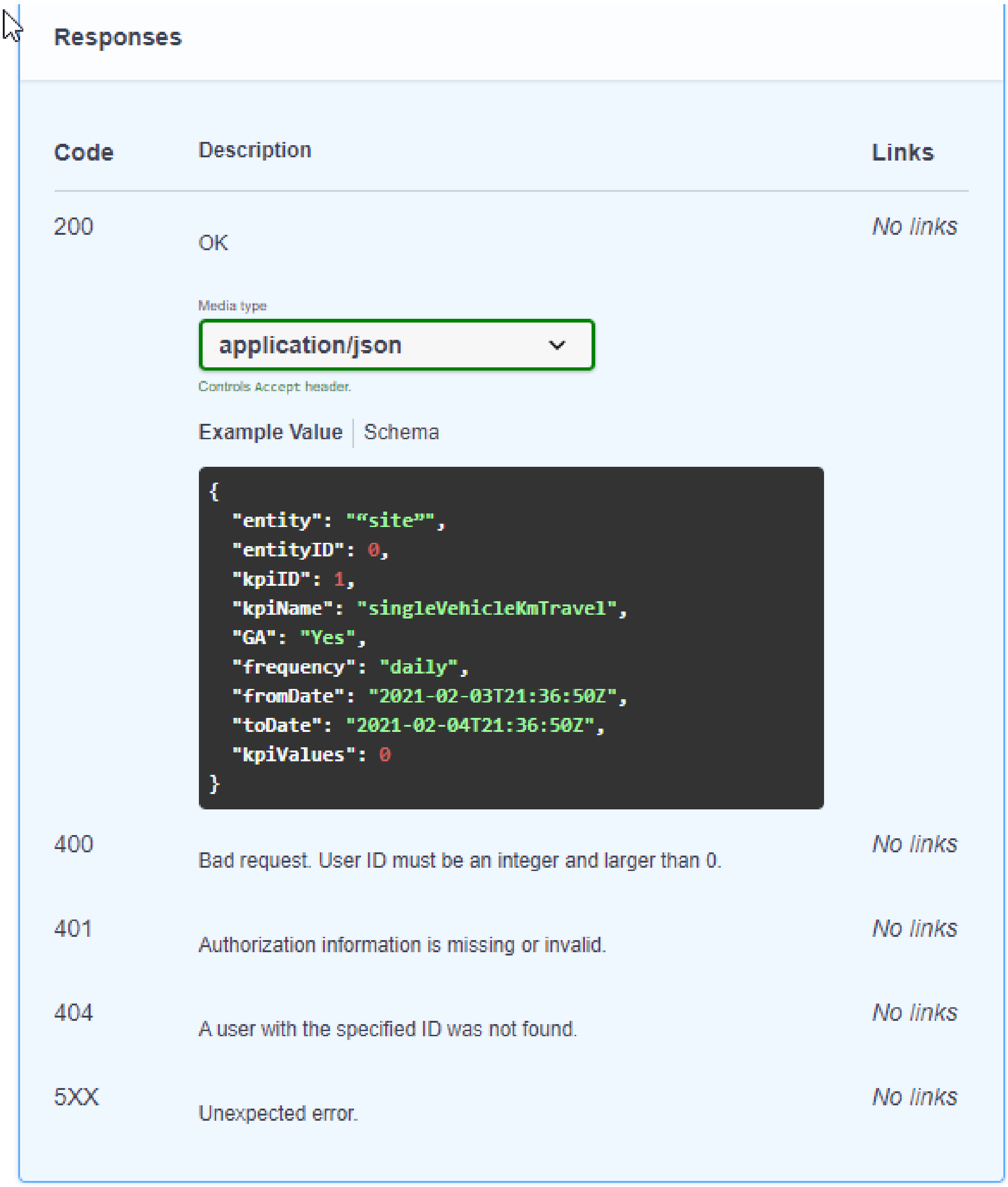


Figure 20: OpenAPI Swagger response example.

Examples of REST API queries are provided below:

https://elin.linkoping-ri.se/vehicle_info/1/1/1/2/?time_from=2021-03-03T21:36:50Z&time_to=2021-03-06T21:36:50Z&frequency=daily

https://elin.linkoping-ri.se/fleet_info/1/1/2/?time_from=2021-01-04T21:36:50Z&time_to=2021-03-06T21:36:50Z&frequency=weekly

https://elin.linkoping-ri.se/site_info/1/2/?time_from=2020-01-04T21:36:50Z&time_to=2020-08-06T21:36:50Z&frequency=monthly

4.10 Local data collector prototype

This section describes a local data collector prototype that has been implemented in Linköping site. It presents an example of how such a solution works with SHOW SMDP and SHOW Dashboard. Other pilot sites may opt for similar solutions or build their solution.

4.10.1 Architecture

To test the vehicle map system in the dashboard a vehicle position REST Server was developed. The server architecture is based on a back end consisting of Apache NiFi data collection system, a database, a REST server and a map server (Figure 21).

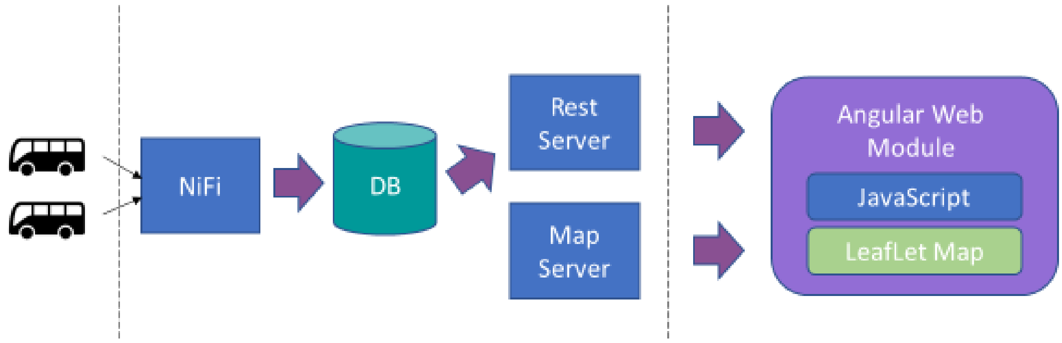


Figure 21: Local data collector prototype architecture.

NiFi: The Apache NiFi [23] is a data routing, information transformation, and system mediation logic system that is used for collecting the streamed data from autonomous vehicles. The NiFi processes the data for future usage and stores the data in a database.

Database: The database is a standard SQL database (MySQL [24]), storing processed collected and parsed data.

Vehicle position server: The Vehicle position server is representational state transfer (REST) server written in NodeJS [25]. It is responsible to provide the current vehicles' position at the Linköping site. The REST server API is described below.

Map Tile Server: The map tile server is an OpenStreetMap server providing map the tiles (images) that is used by the map system to render the map. The map server gets the geodata from the OpenStreetMap foundation.

Angular Web Module: Is the front end that displays the map. It consists of two basic components a JavaScript module responsible for the logic displaying the vehicles in the angular module, and a Leaflet Map Visualisation module responsible for the map rendering.

4.10.2 Local data collector messaging API

The vehicle position REST Server implements a standard GET method that generates a response given a list of all vehicle positions for a specific site.

GET: /getpos/

It returns an array of all vehicles position information, at the site, formatted as a JSON object containing an array with vehicle information.

[{vehicle info}, …, {vehicle info}]

The vehicle information contains:

- vehicle_id: The id for the vehicle.
- record_created_date: Time when the data was received from the vehicle.
- lon: Longitude position.
- lat: Latitude position.

- speed: The speed of the vehicle in km/h

  {"vehicle_id","record_created_date","lon","lat","speed"}

Example on response string:


```
[{"vehicle_id":"VJRD1A10224000036","record_created_date":"2021-05-21T12:41:04.000Z","lon":15.573281585060013,"lat":58.395842934238985,"speed":0},
{"vehicle_id":"VG9A2CB2CJV019145","record_created_date":"2021-05-21T11:18:50.000Z","lon":15.571910987480612,"lat":58.396150599245495,"speed":null}]
```


4.11 SHOW Dashboard self-service on-boarding

4.11.1 SHOW Dashboard entity hierarchy

SHOW Dashboard entity hierarchy is illustrated in Figure 22. To avoid confusion, readers please note that the entities here are virtual entities used within the Dashboard platform. The global dashboard (Figure 7) provides the overview dashboard consisting of all SHOW pilot sites, while SHOW site dashboards (e.g. Figure 8) provide detailed information about specific sites. There are SHOW sites having their own local dashboards and thus only populate the precomputed KPIs to the SHOW SMDP to be visualized in SHOW Dashboard. These sites include (at the time of this writing): Rouen (FR), Tampere (FI), and Turin (IT). The preliminary list of existing local dashboards and intended usage of SHOW Dashboard is provided in Table 15, D4.1 [2]. Development and integration statuses of these local systems will be updated in the next releases of D4.1.

Several fleets are linked as child objects of a site, while vehicles are the lowest entity level that belongs to fleet objects.

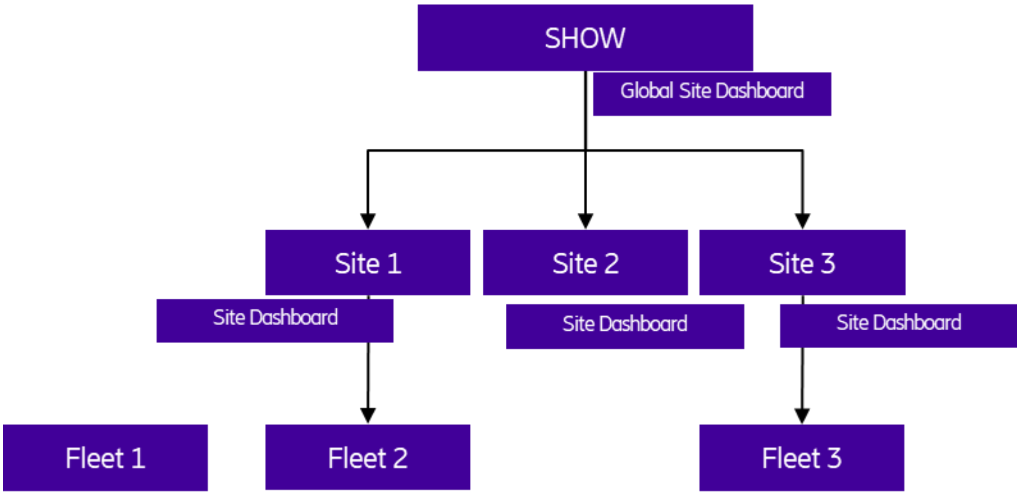


Figure 22: SHOW Dashboard entity relationship diagram.

4.11.2 Community on-boarding

The innovation cloud hosts several communities sharing the same infrastructure but isolated in terms of access and authorization to functionalities and data.

All users of SHOW Dashboard are belonging to a community namely SHOW Automated Urban Mobility. The community has its URL access at: https://demo.innovationcloud.ericsson.net/show-project

A community is formed from a number of collaborative partners, which are child objects of the community entity. Community enables its partners to share knowledge and data along different collaborative projects.

A community can be either public or private. Within the SHOW project, we have created the SHOW Automated Urban Mobility community, where the members are partners of the SHOW project consortium (Figure 23).

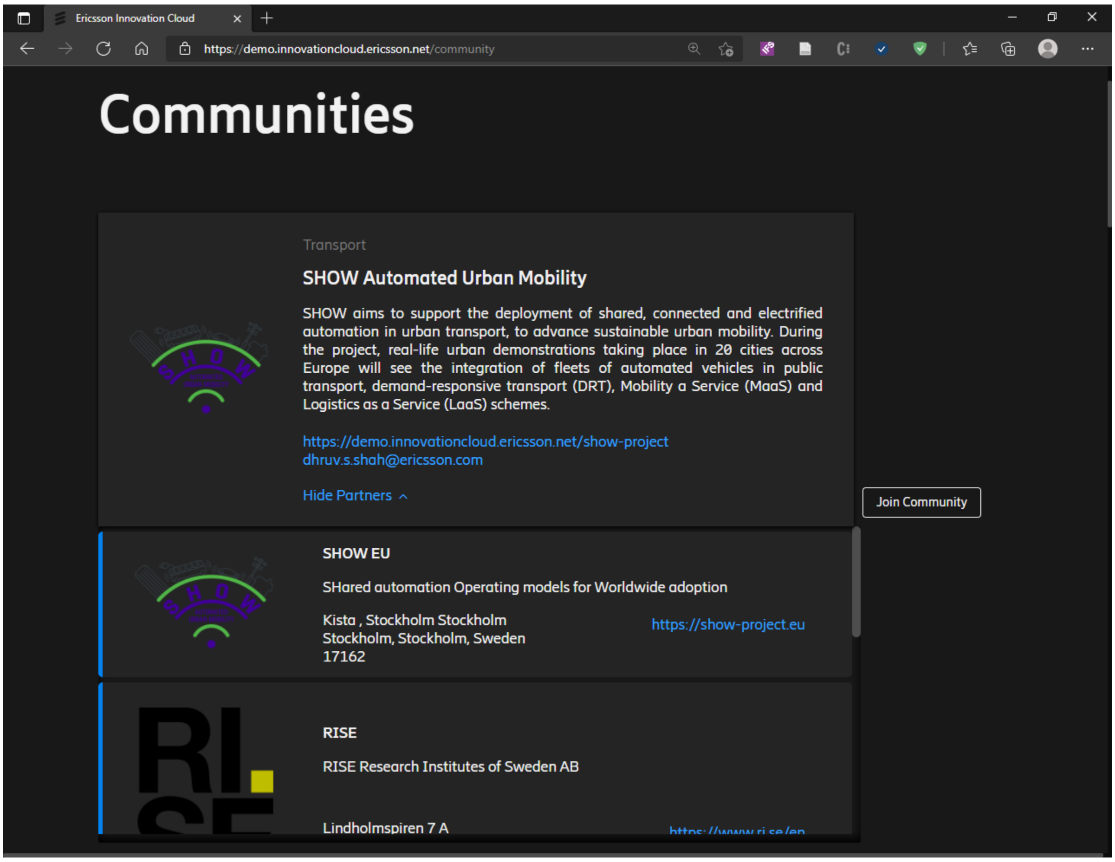


Figure 23: SHOW Project community.

4.11.3 Partner on-boarding

Once created, a community admin can invite partners to on-board the platform using their corporate emails. Each partner will assign a partner admin, who can then involve in the self-on-boarding process by inviting and approving users from their organization to access the platform.

5 SHOW Dashboard reference PoC

SHOW Dashboard reference PoC is available at the following URL: https://demo.innovationcloud.ericsson.net/show-project/view/dashboard. This section provides the screenshots of the PoC implementation.

5.1 Site management

All sites in SHOW are created within the SHOW Dashboard platform with an autogenerated unique ID. Basic information of a site is illustrated in the Site management screen (Figure 24).

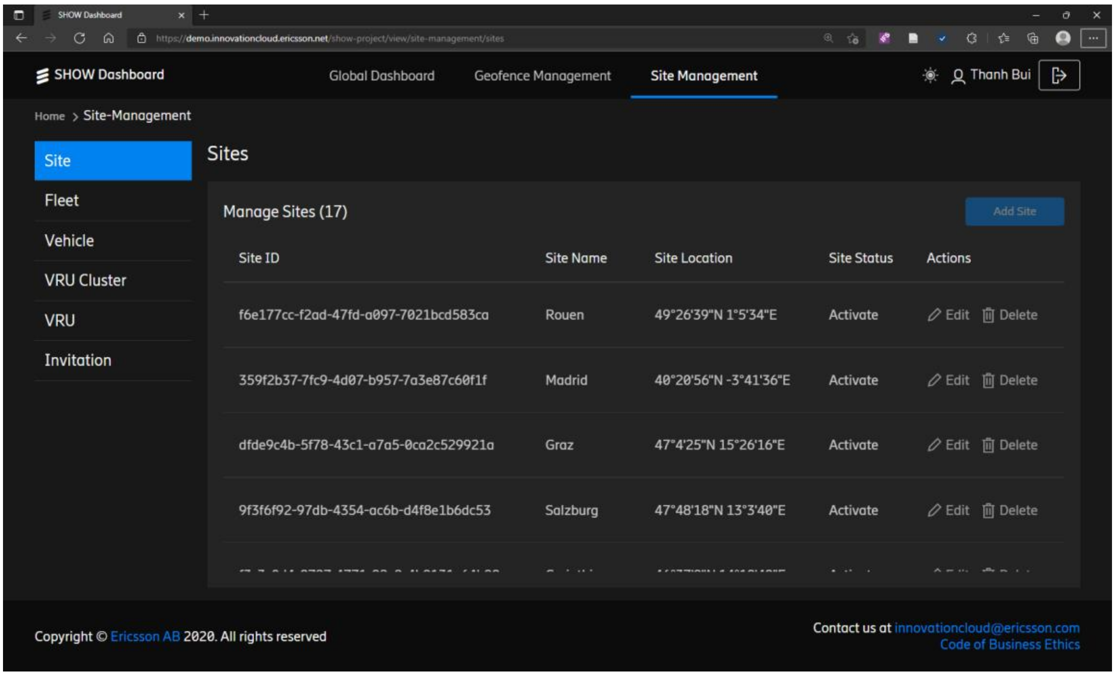


Figure 24: Site management.

Child objects of a site are fleets. The fleets are created with descriptions and geospatial objects such as routes, stops. Figure 25 shows the capability to define routes for registered fleets. The routes can either be manually plotted directly on map or imported from file with GeoJSON [26] format.

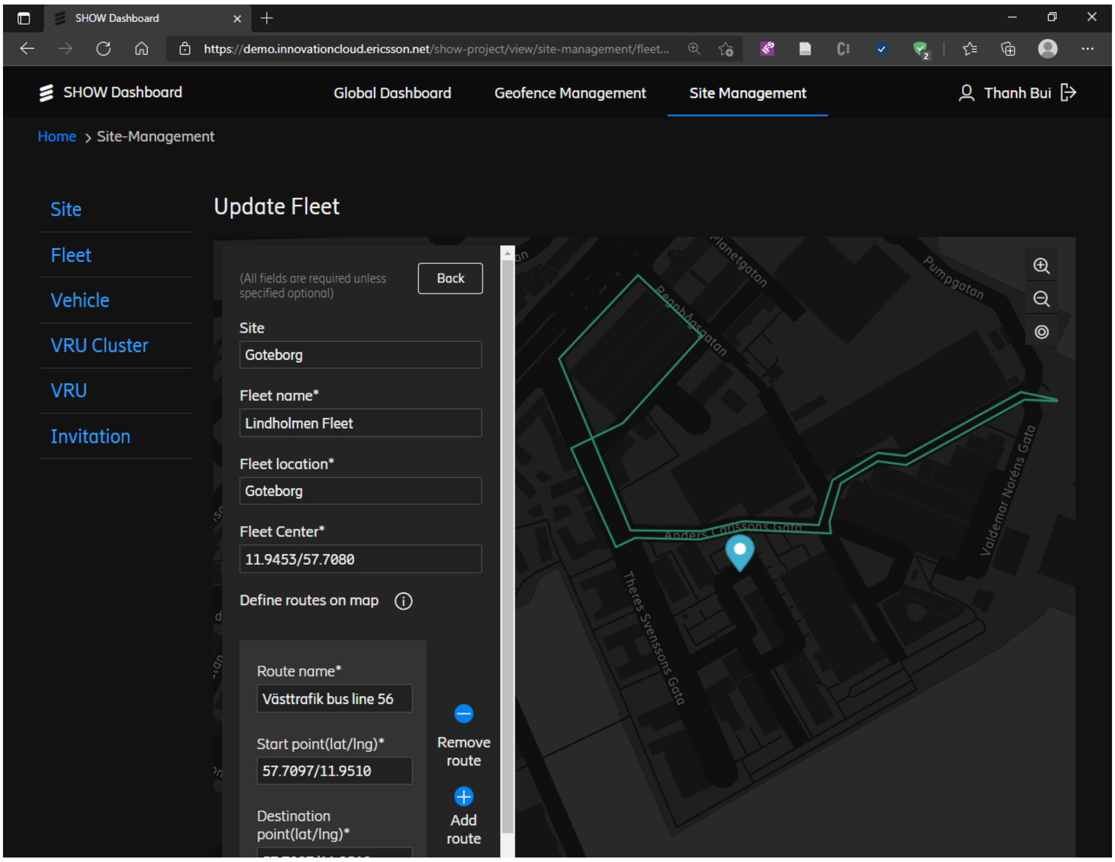


Figure 25: Fleet data registration.

Vehicles are then registered as child objects of fleets. The vehicle specification information is also registered in this step. Some mandatory attributes of the vehicle data are required to compute KPIs related to e.g. energy use, emission and mobility service type. Vehicle registration screen is illustrated in Figure 26.

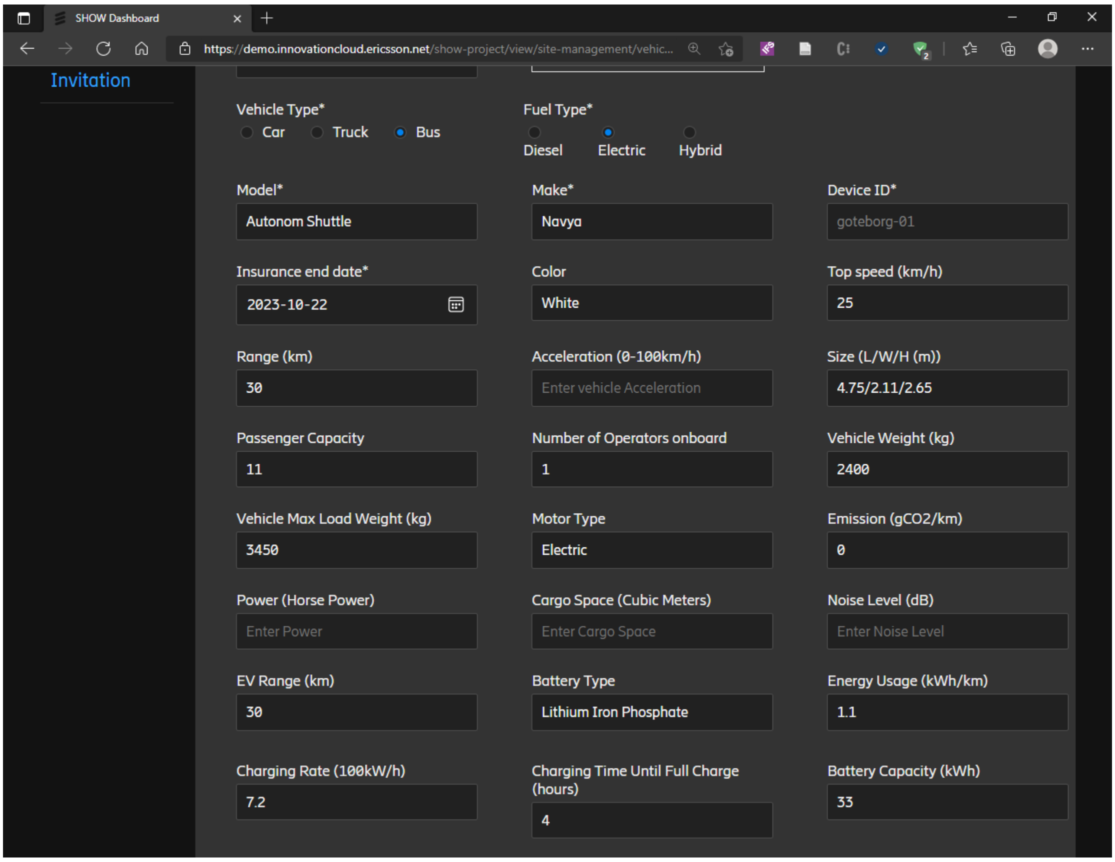


Figure 26: Vehicle data registration.

5.2 Geofence management

5.2.1 Geofence

Geofences are created as geospatial objects to allow geofencing functionalities. Each geofence is defined by a closed region on map and the corresponding geofence rules will be triggered when a vehicle enters or exits a geofence. Figure 27 shows the functionality to create a geofence object belonging to a site.

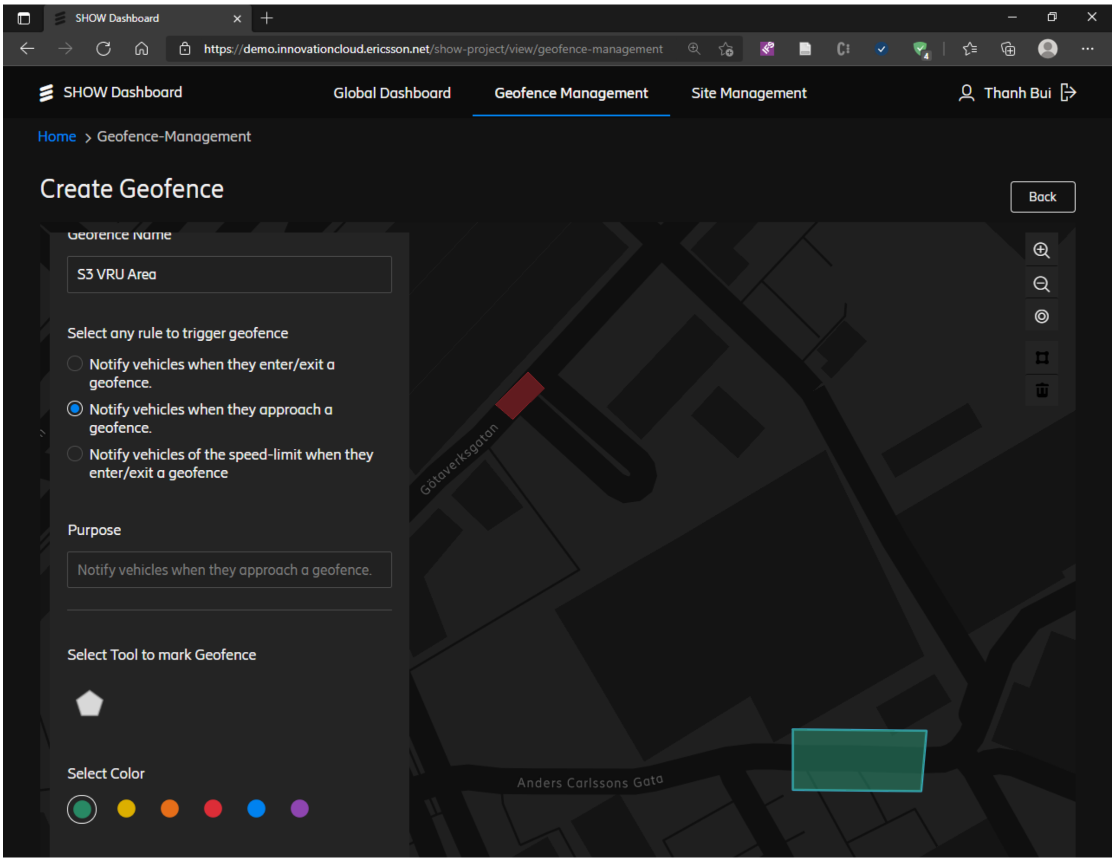


Figure 27: Geofence creation.

5.2.2 Geofencing rules

Once the geofences are created, different rules can be defined to be triggered by the events generated: enter/exit or approach a specific geofence. The rules will be executed in the real-time rule engine to generate real-time alerts that can be shown in Dashboard and available for retrieval by external predefined systems via the Dashboard MQTT broker (e.g. vehicle's HMI). A screenshot showing geofence rules is provided in Figure 28.

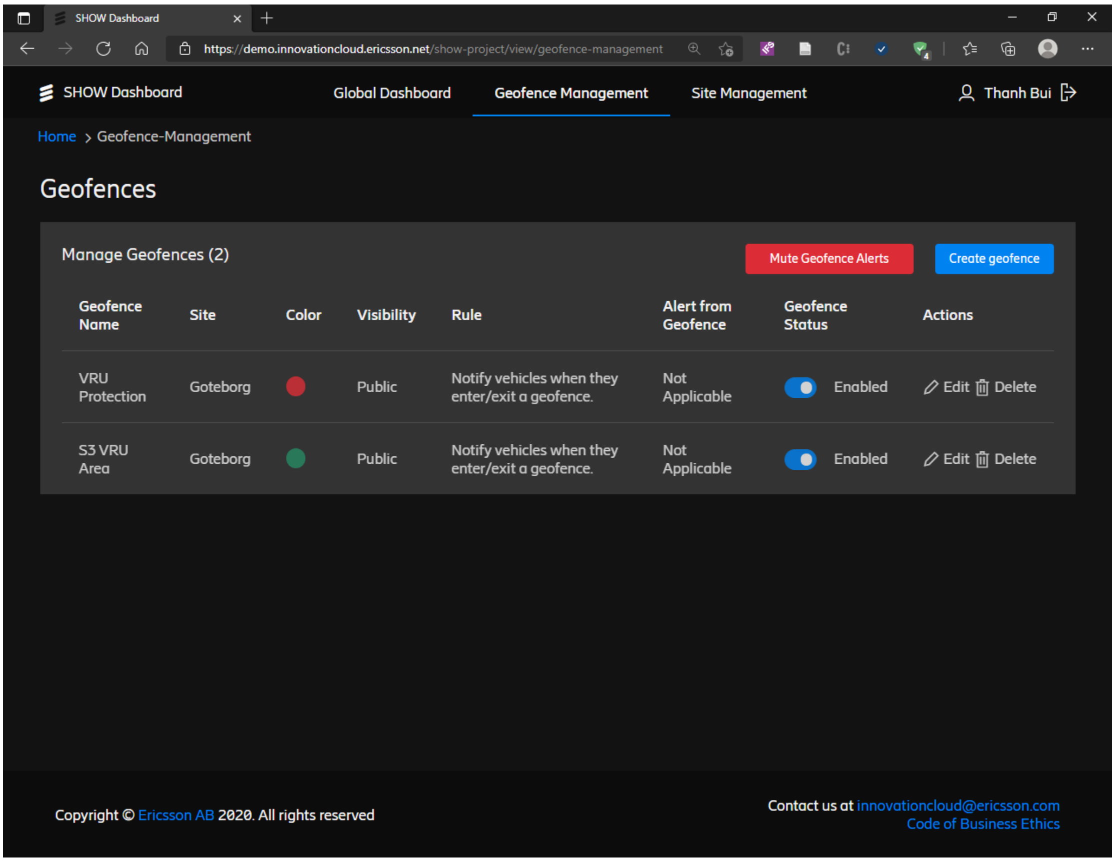


Figure 28: Geofence management.

5.3 Multi-site dashboard

The multi-site view provides the summary view of all SHOW demo sites (Figure 29). In this summary view, some selected representative KPIs are displayed together with map buttons corresponding to SHOW pilot sites. The buttons represent SHOW pilot sites named with site's initial letter, the gray-coded buttons are follower sites that are currently disabled. Sites belonging to SHOW megasites (Sweden, Germany, France, Switzerland, Spain) are marked with color coded border.

A switch between dark/bright themes is available for user to select upon preference (Figure 30).

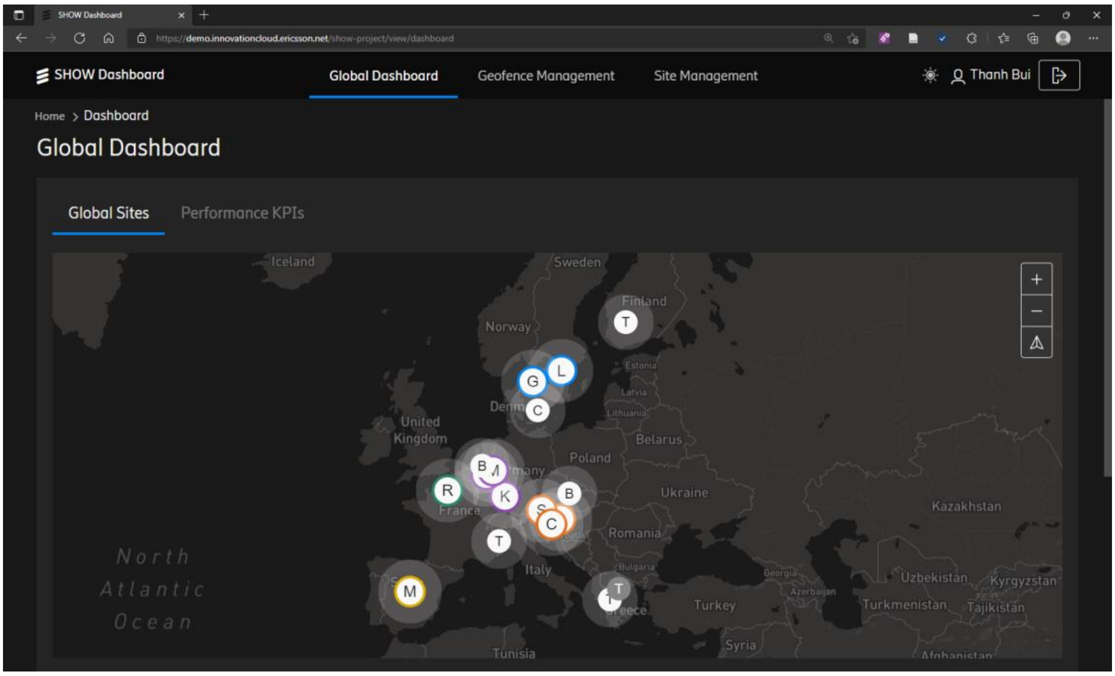


Figure 29: SHOW Dashboard multi-site view screenshot.

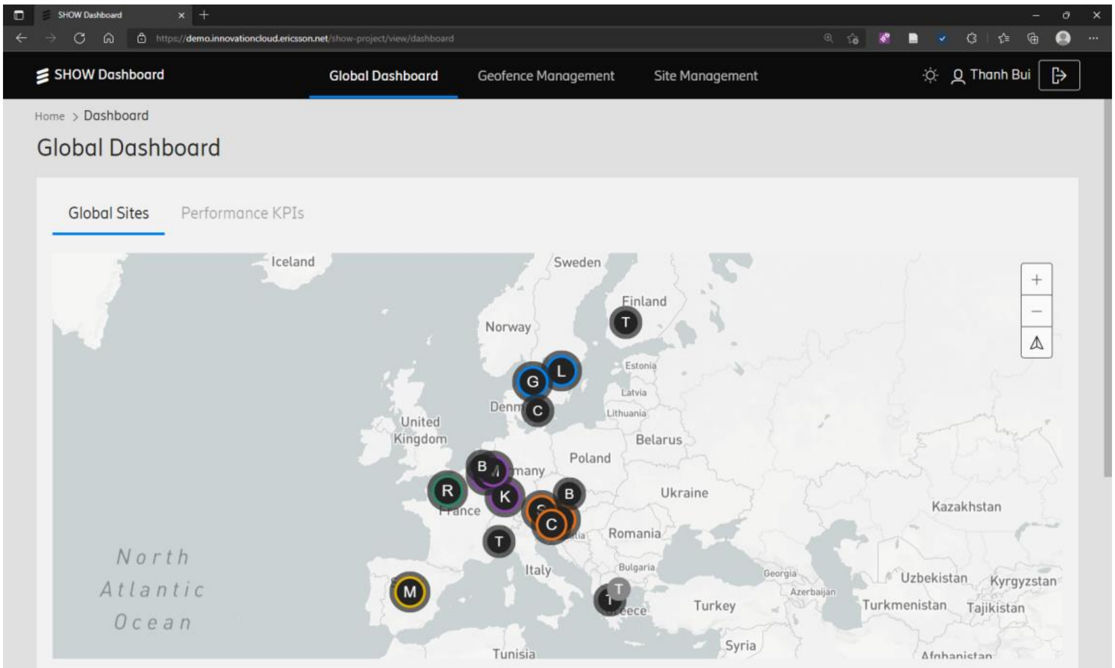


Figure 30: SHOW Dashboard, bright theme.

5.4 Site Dashboard

Partners belonging to each SHOW pilot site can get access to their site Dashboard view in SHOW Dashboard. The Site Dashboard view provides (the latest updated) live information of vehicles/fleets and all KPIs relevant to the site.

The KPIs are grouped into 6 broader categories[20]:

- KPIs under the 02 broad categories (i) “user acceptance/experience” and (i) “project success” are only updated at the end of a demonstration phase, with data collected from the online surveys and reports.

- KPIs under other 04 broad categories are retrieved from the SHOW SMDP platform with more frequent updates.

The broad categories are used as menu items to access KPI gadget views belonging to each category. Figure 31 is the screenshot illustrating KPI views of a site.

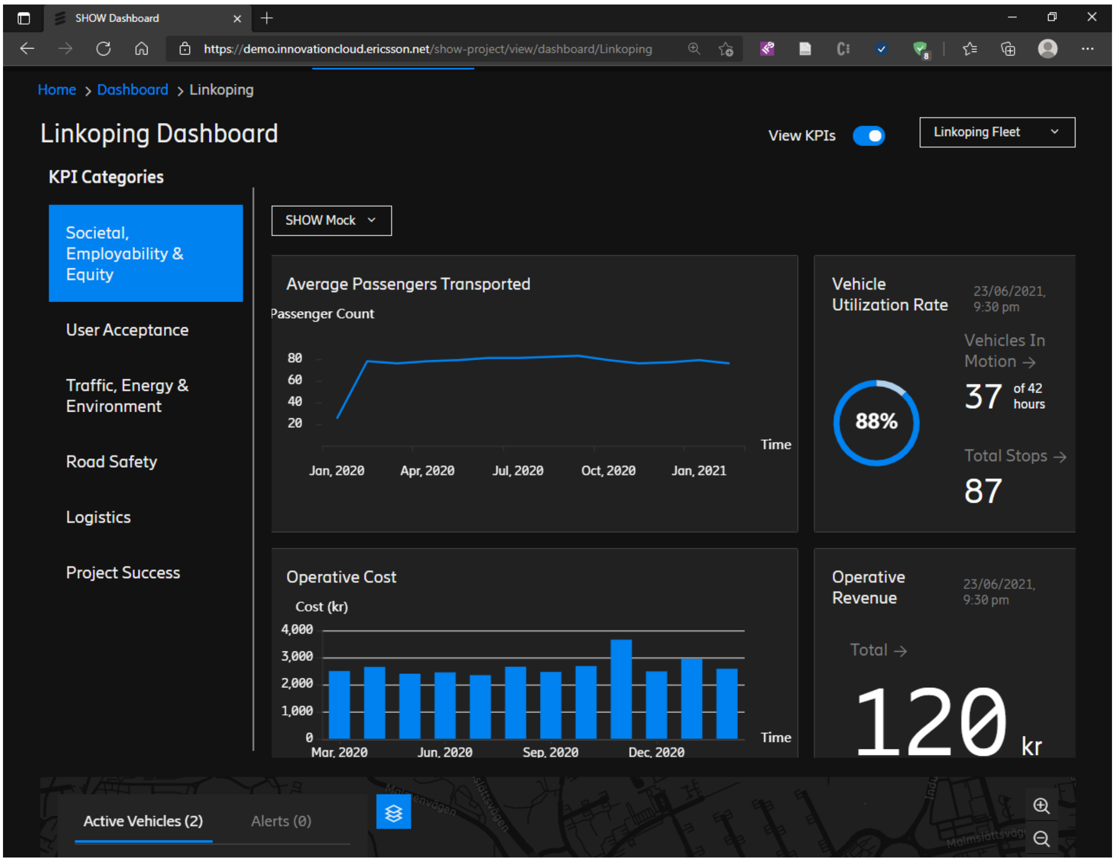


Figure 31: SHOW Dashboard site view (KPI screen).

Streaming real-time information collected from the site is shown in real-time[9], including vehicle positions, status, services, etc. Figure 32 shows the real-time positions of SHOW vehicles together with available real-time information in the left panel.

[9] The terms real-time here refers to the IoT real-time protocol MQTT that is used to stream vehicle data all the way to the Dashboard. The protocol provides real-time communication with QoS support, however since the SHOW Dashboard is not dealing with time-critical application in this function, there is no strict requirement on end-to-end network latency that need to be complied. The visualized “real-time” data are thus in best-effort mode and can be considered near real-time to avoid confusions.

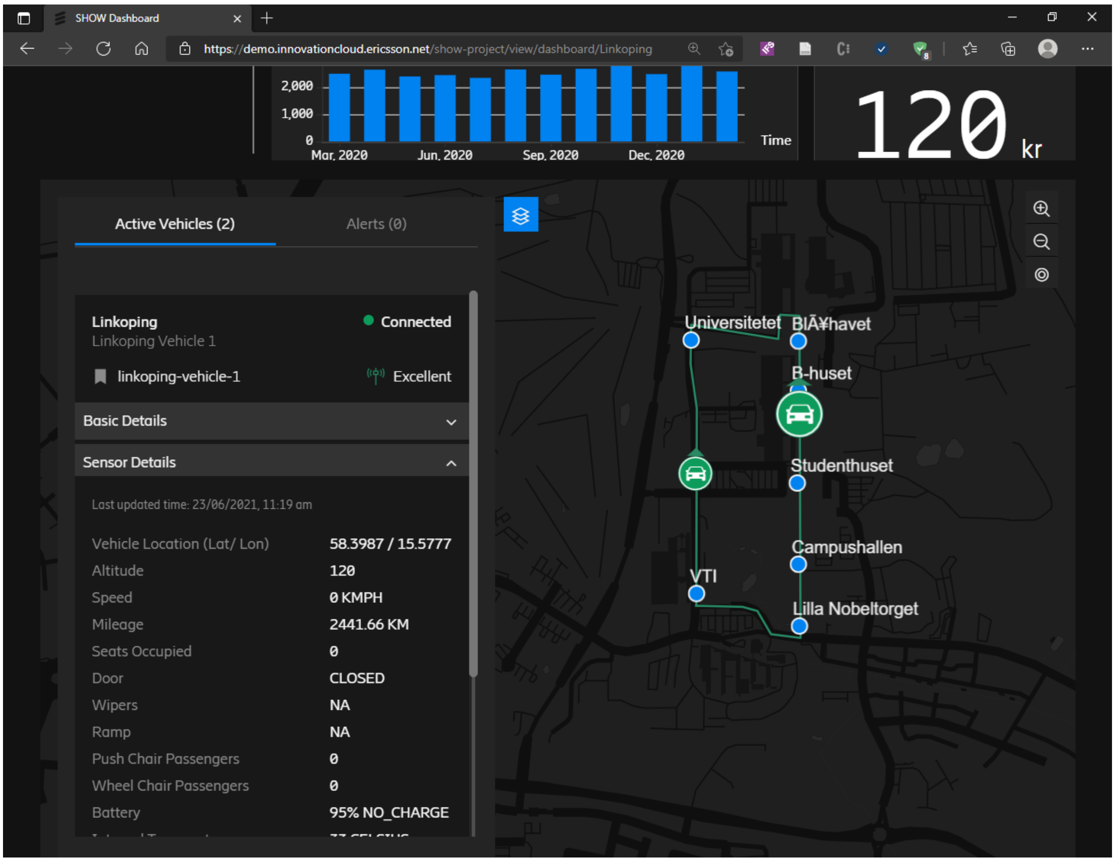


Figure 32: SHOW Dashboard site view (vehicle real-time position).

5.5 Site data setup

5.5.1 Site metadata

Each pilot site in SHOW is on-boarding with the description of the site in form of metadata. The metadata of the site includes the following information:

- Site ID: An unique ID automatically generated and assigned by the system using Universally Unique Identifier UUID [27].
- Site name: Location name of the pilot site
- Site location: geographic coordinates of the site (longitude, latitude)
- Site status
- Fleets belong to site
  - Fleet ID (UUID)
  - Fleet name
  - Fleet location
  - Routes (including stops)
- Vehicles assigned to a fleet
  - Vehicle ID
  - Other specification information (Figure 34)
- VRU cluster
- VRU
- User invitations to access site data

Figure 33, Figure 34 are example screenshots of site metadata.

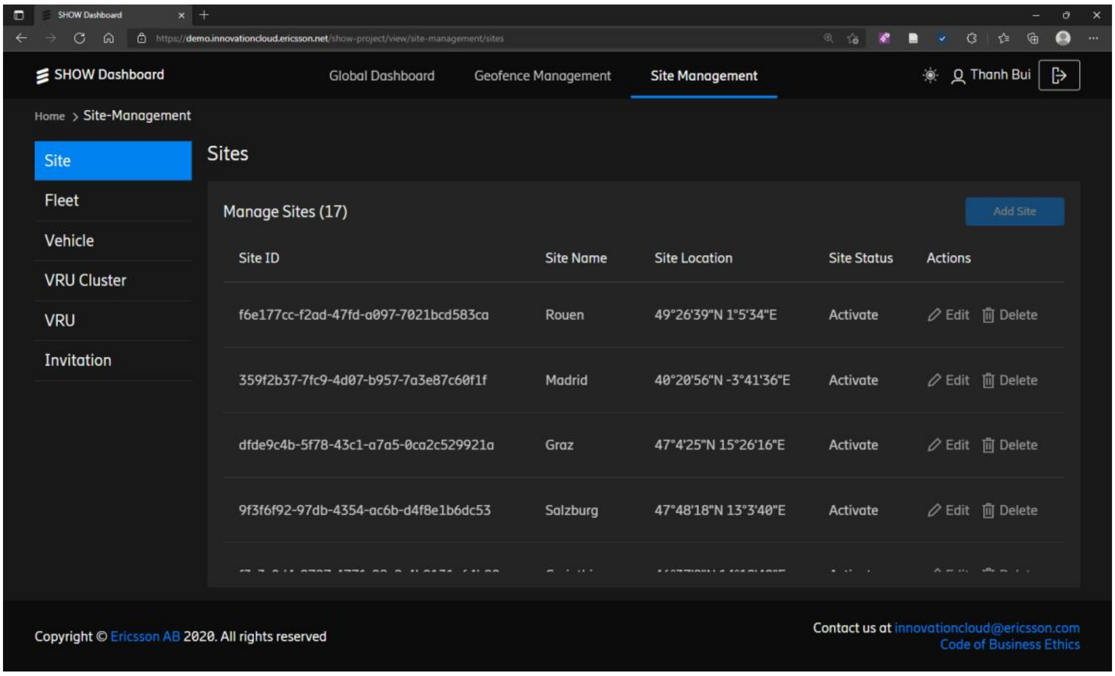


Figure 33: Site metadata.

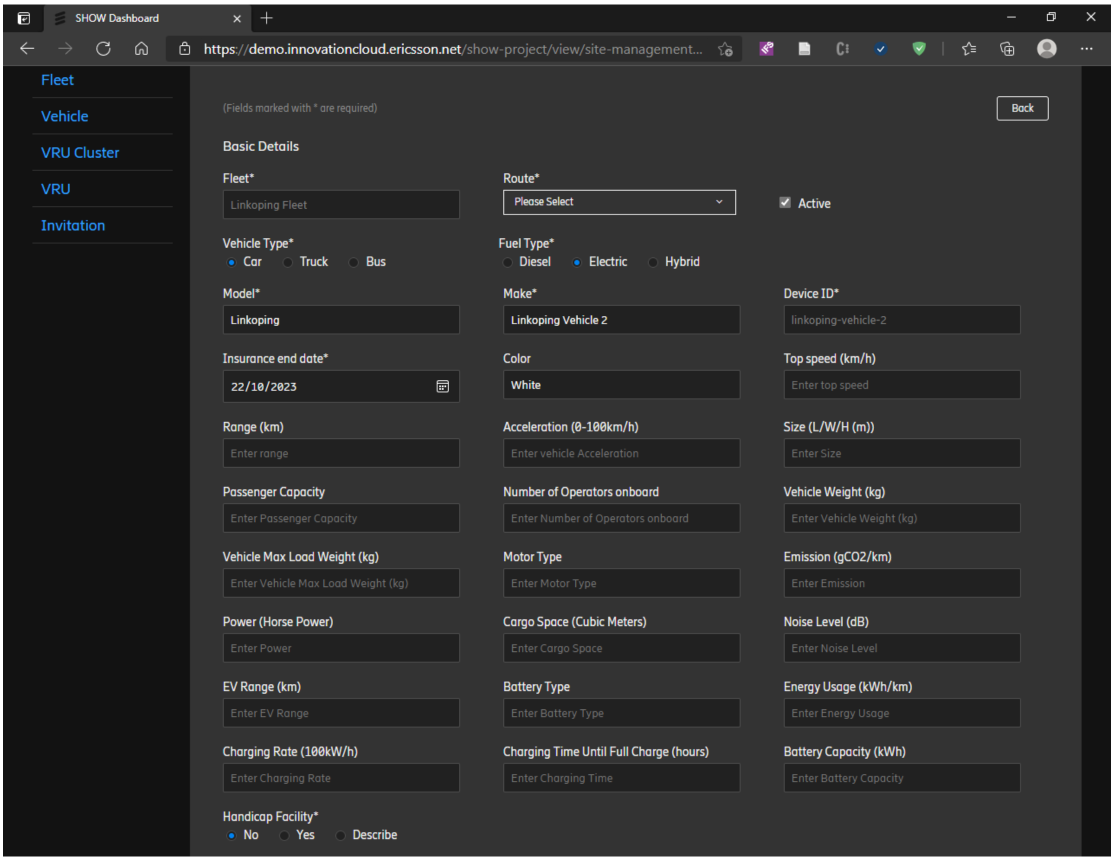


Figure 34: Vehicle specification.

5.5.2 Telemetry data

Near real-time data from the site is collected via MQTT protocol through the MQTT broker. The real-time rules (e.g. geofence rules) can be applied to these data. The data is collected with different topics connected to different sensors or input systems at the site. Figure 35 shows the MQTT data topics provided by a vehicle in the Linköping site.

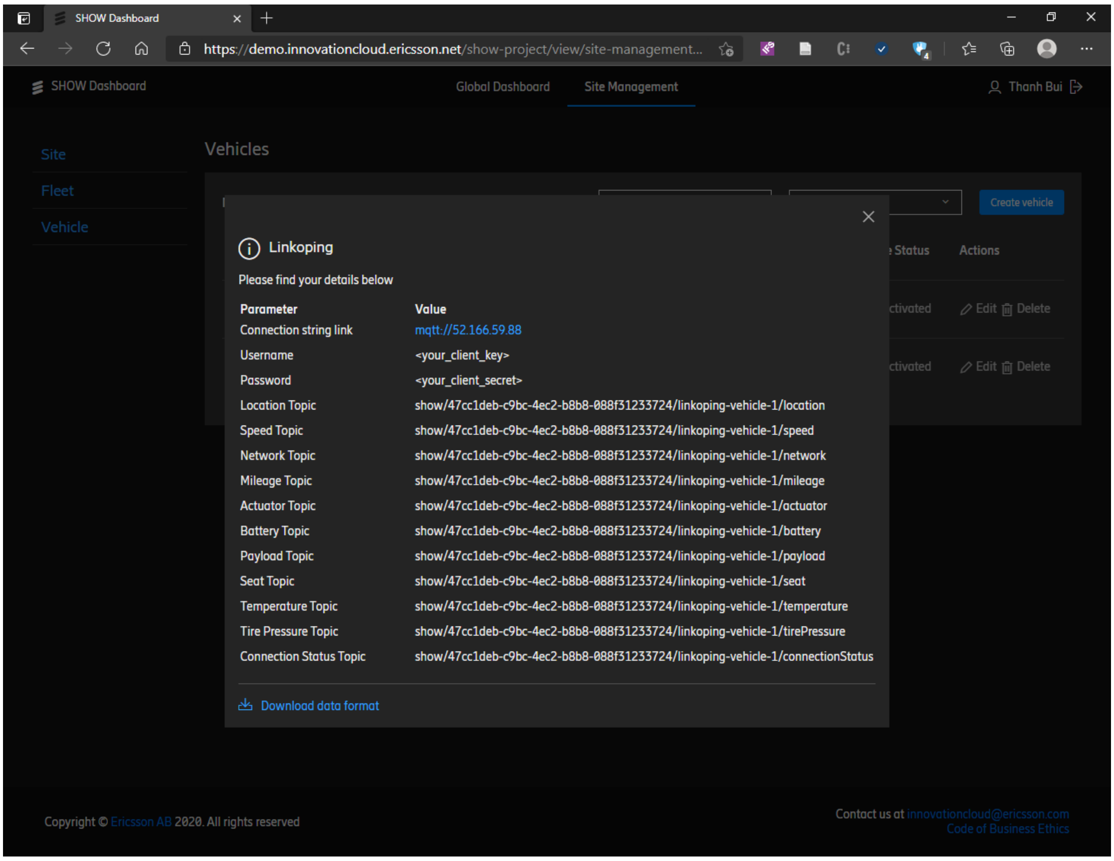


Figure 35: MQTT vehicle data topics.

5.5.3 KPI data

KPI data is collected to SHOW Dashboard via REST interface from SHOW SMDP. The preliminary specification and examples of REST KPI requests and responses are provided in Appendix. The final specification will be provided in the WP5's deliverables, which will also include the final specification of KPIs.

6 Conclusions and outlook

SHOW project together with its large number and diversity of European pilot sites, vehicle OEMs, local systems, legal frameworks, user experiences, and technology developments create a unique opportunity for the research team to integrate different innovation ideas into one single technological platform.

This document provides a snapshot view of the SHOW Dashboard PoC system and briefly describes how it has been implemented. The SHOW Dashboard PoC system will be used for the evaluation and demonstration tasks within SHOW to illustrate the pilot results at both Pre-demo and Demo phases and showcase the actual achievement and impact of the project.

The SHOW Dashboard has been designed and developed through an agile PoC approach and inherits the best practices from an existing innovation platform. The approach allows the SHOW Dashboard PoC system to reflect the latest updates of all interrelated tasks in the project. These leads, finally, to the full SHOW Dashboard ready and operational by the end of October 2021 (according to the plan of pre-demo and demo phases of SHOW pilot sites).

The following supporting work items are foreseen and planned for the coming period:

- Continue support of onboarding all SHOW pilot sites (following the plan).
- A public layer containing data that can be accessible to the public has been planned to allow public access from SHOW official website[19] as a dissemination result. This layer will be made available when sufficient KPIs data are received from SHOW pilot sites during the demo phases.

References

[1] SHOW, “D5.1 SHOW big data collection platform and data management portal.” Deliverable of the Horizon-2020 SHOW project, Grant Agreement No. 875530., 2020.

[2] SHOW, “D4.2: SHOW Dashboard.” Deliverable of the Horizon-2020 SHOW project, Grant Agreement No. 875530., 2020.

[3] “Drive Sweden Innovation Cloud | Best practice,” *Smart City Sweden*. https://smartcitysweden.com/best-practice/370/drive-sweden-innovation-cloud/ (accessed May 12, 2021).

[4] “Drive Sweden,” *Drive Sweden*. https://www.drivesweden.net/ (accessed May 27, 2021).

[5] “KRABAT,” *Drive Sweden*, Dec. 04, 2018. https://www.drivesweden.net/en/projects-5/krabat (accessed May 27, 2021).

[6] “AD Aware Traffic Control,” *Drive Sweden*, Dec. 03, 2018. https://www.drivesweden.net/en/projects-5/ad-aware-traffic-control (accessed May 27, 2021).

[7] “AVTCT,” *Drive Sweden*, May 15, 2019. https://www.drivesweden.net/en/projects-5/avtct (accessed May 27, 2021).

[8] K. Bouree, H. Roach, L. Staub, and G. Van Der Peet, “TRANSMODEL, A FRAMEWORK FOR INTEROPERABILITY OF SOFTWARE PACKAGES IN PUBLIC TRANSPORT,” *Public Transport International*, vol. 43, no. 4, 1994, Accessed: May 17, 2021. [Online]. Available: https://trid.trb.org/view/425796

[9] “CEN/TS 16614-1:2020 - Public transport - Network and Timetable Exchange (NeTEx) - Part 1: Public transport network topology exchange format,” *iTeh Standards Store*. https://standards.iteh.ai/catalog/standards/cen/b1a3dd12-d026-4b39-9972-c17b4baa245a/cen-ts-16614-1-2020 (accessed May 17, 2021).

[10] “SIRI standard – Transmodel.” http://www.transmodel-cen.eu/standards/siri/ (accessed May 27, 2021).

[11] “Transmodel implementation in Sweden – Transmodel.” http://www.transmodel-cen.eu/implementations/sweden/ (accessed May 17, 2021).

[12] “Reference | Static Transit,” *Google Developers*. https://developers.google.com/transit/gtfs/reference (accessed May 27, 2021).

[13] OpenStreetMap contributors, *Planet dump retrieved from https://planet.osm.org*. 2017.

[14] “About Mapbox Maps | Mapbox.” https://www.mapbox.com/about/maps/ (accessed May 27, 2021).

[15] P. Crickard, *Leaflet.Js Essentials*. Packt Publishing, 2014.

[16] “HERE: the next generation of location services – Nokia Conversations : the official Nokia blog.” https://web.archive.org/web/20130725162540/http://conversations.nokia.com/2012/11/13/here-the-next-generation-of-location-services/ (accessed May 27, 2021).

[17] “Geo-location APIs | Google Maps Platform | Google Cloud.” https://cloud.google.com/maps-platform/ (accessed May 28, 2021).

[18] “The C4 model for visualising software architecture.” https://c4model.com/ (accessed May 27, 2021).

[19] “show-project.eu |.” https://show-project.eu/ (accessed Jun. 28, 2021).

[20] SHOW, “D9.2 Pilot experimental plans, KPIs definition & impact assessment framework for pre-demo evaluation.” Deliverable of the Horizon-2020 SHOW project, Grant Agreement No. 875530., 2020.

[21] SHOW, “D9.3 Pilot experimental plans, KPIs definition & impact assessment framework for final demonstration round.” Deliverable of the Horizon-2020 SHOW project, Grant Agreement No. 875530., 2020.

[22] "OpenAPI Specification - Version 3.0.3 | Swagger." https://swagger.io/specification/ (accessed Jun. 28, 2021).
[23] "Apache NiFi." https://nifi.apache.org/ (accessed May 28, 2021).
[24] "MySQL." https://www.mysql.com/ (accessed May 28, 2021).
[25] Node.js, "Node.js," *Node.js*. https://nodejs.org/en/ (accessed May 28, 2021).
[26] "GeoJSON." https://geojson.org/ (accessed May 28, 2021).
[27] P. Leach, M. Mealling, and R. Salz, "RFC 4122: A Universally Unique IDentifier (UUID) URN Namespace," 2005, [Online]. Available: http://www.ietf.org/rfc/rfc4122.txt

Appendix I: Demo sites support for Local Dashboards or SHOW Dashboard

Table 10: Local Dashboards and intended usages of SHOW reference Dashboard current status (the Mega sites)

ID	The Mega Sites	City	Local dashboard status	TRL (1-9)	Beneficiary operating local dashboard	Short description of key operations of the local dashboard	Readiness to connect to SHOW Dashboard	Remarks
1	France	Rouen	Will build one	6	Transdev	Fleet monitoring & fleet management	Will provide only pre-computed KPIs	Rouen will provide pre computed KPIs and some batch data on a regular basis (frequency TBC).
3	Spain	Madrid	Using solely SHOW Dashboard	N/A	EMT, IRIZAR, TECNALIA	Fleet monitoring, route visualisation, KPIs for user (driver/passenger)	Will provide both real time and batch data.	Madrid mega pilot site fleet/KPI data are still under investigation based on all types of considered vehicles; both real time and batch data integration for feeding the SHOW Dashboard are considered. A local data collector will be developed.
4	Austria	Graz	Using solely SHOW Dashboard	N/A	VIF, AVL	N/A	Will provide "near real-time" data	-
5		Salzburg	Using solely SHOW Dashboard	N/A	N/A	N/A	See next column	The Salzburg Pilot is using the fleet management API “EZ-Fleet” provided by the OEM. Connection to the SHOW Dashboard is possible only under the following prerequisites: data sharing with SHOW cloud platform can be achieved either via OEM-private cloud (OEM to allow) or via SFRG cloud storage (OEM to agree) or directly via communication with the fleet (only if OEM recommends this for some reason). The specific way to be selected is under exploration by WP5.
6		Carinthia (pending Amendment)	Using solely SHOW Dashboard	N/A	n/a	Route visualisation	Will provide batch data upon a frequency to be defined.	Will provide KPIs on a regular basis. Frequency needs to be decided.

ID	The Mega Sites	City	Local dashboard status	TRL (1-9)	Beneficiary operating local dashboard	Short description of key operations of the local dashboard	Readiness to connect to SHOW Dashboard	Remarks
7	Germany	Karlsruhe	Using solely SHOW Dashboard	N/A	FZI	Aggregated/realtime KPIs will be provided via interfaces to the WP5 platform, to be visualised in SHOW Dashboard	Will provide batch data upon a frequency to be defined.	Will provide KPIs on a regular basis. The frequency is not clear yet.
9		Aachen	Using solely SHOW Dashboard	N/A	n.a.	N/A	Will provide "near realtime" data	No local dashboard planned, connection to SHOW Dashboard.
10	Sweden	Linköping	Using SHOW & own local Dashboard	8	Transdev	Fleet monitoring and limited teleoperation	Will provide "near realtime" data	Both local and SHOW Dashboard will be used. To use local data collector solution.
11		Göteborg	Using also SHOW Dashboard	7	Keolis	Fleet management solution from Autofleet	Will provide batch data in CSV	The interface is based on a number of defined and agreed API's between SHOW Dashboard and the Public Transport provider. Keolis are using IT systems from Hogia. Message transfer is done by using MQTT as a mechanism. This is tested and working since early November 2020. The same data collector solution as deployed in Linköping site will be used for integration with SHOW Dashboard.

Table 11: Local Dashboards and intended usages of SHOW reference Dashboard current status (the Satellite sites)

ID	The Satellites sites	City	Local dashboard status	TRL	Beneficiary operating local dashboard	Short description of key operations of the local dashboard	Readiness to connect to SHOW Dashboard	Remarks
12	Finland	Tampere	WIll build one	6	Sensible 4	Fleet monitoring, route visualisation, KPIs for traveller and vehicle efficiency	See next column	Will provide some of the KPIs pre-computed. APIs can be made available. Data to be exchanged to be confirmed. Possibility to also utilise SHOW Dashboard fully is studied.
13	Greece	Trikala	Using also SHOW Dashboard	6	(e-Trikala)	Local SHOW Dashboard may be developed depending on resources availability based on the locally computed KPIs. Additional existing system includes a local remote control center room that supports fleet real time monitoring (and possibly teleoperation) through cameras (driver's view and road view from inside the shuttle) and emergency breaking and immobilisation.	Will provide "near realtime" data	Data from the AVs are not confirmed yet due to late delivery of the shuttles (due for Sept 2021). Plan to share close-to -real time data.
14	Netherlands	Eindhoven (Brainport)	Using SHOW Dashboard most probably	N/A	N/A	N/A	Will provide only pre-computed KPIs	

ID	The Satellites sites	City	Local dashboard status	TRL	Beneficiary operating local dashboard	Short description of key operations of the local dashboard	Readiness to connect to SHOW Dashboard	Remarks
15	Italy	Turin	Existing	9	Bestmile (or replacing partner with similar solution; pending amendment)	"• Observe bookings, automated matching of rides and dispatching of trips, and manage exceptions • Visualize real-time service and vehicle information • Receive, create and edit field logs and incident reports • Design service areas, lines, timetables and frequencies • Set parameters for service constraints and objectives • Plan vehicle, fleet and driver availabilities • Provide traveler, vehicle and fleet efficiency KPIs "	Pending to BESTMILE situation clarification.	Will provide some of the KPIs pre-computed, extracted on a monthly basis. APIs also available to connect directly with our backend platform. Data to be exchanged to be confirmed.
16	Czech Republic	Brno	Will build one.	6	ARTIN	Fleet monitoring and fleet management and teleoperation	Will provide only pre-computed KPI	Will provide KPIs on a regular basis. The frequency will be determined. Full use of SHOW Dashboard.

Appendix II: Specifications for KPI Data representation

1. Frequency: Time Interval Variables for each KPI: Daily / Weekly / Monthly
    a. For URLs default frequency is daily. Otherwise, frequency can be passed as query params.
2. Entity Variables for each KPI: Per Vehicle / Fleet / Site
3. Category: Traffic & Passenger Patterns & Efficiency
4. URLs:
    a. /{site-id}/<attribute> - Returns data for the mentioned site
    b. /{site-id}/{fleet-id}/<attribute> - Returns data for the mentioned fleet
    c. /{site-id}/{fleet-id}/{vehicle-id}/<attribute> - Returns data for mentioned vehicle

Proposed KPI Data Representation

The following description is the initial draft resulted from a join works with WP5. Further details of this work will be reported in the next release of D5.1 [1].

Average Speed (Frequency: Daily)

- Latest value (Gauge Chart)
- Representation:
    - Line Chart / Bar Chart
- Endpoints
    - /{site-id}/average-speed
    - /{site-id}/{fleet-id}/average-speed
    - /{site-id}/{fleet-id}/{vehicle-id}/average-speed
- Queries:
    - □from=<date-time>
    - □to=<date-time>
    - □frequency=<daily, weekly, monthly>
        - Default: daily
- Response Examples:
    - If there are no query params the response need not include fields like **from** and **to.** Example:
    - **GET: /{site-id}/average-speed?**from=2021-02-03T21:36:50Z&to=2021-02-05T21:36:50Z&frequency=daily

      Response:

```
{
        "entity": "<site,fleet,vehicle>",
        "entityId": ""<siteId/fleetId/VehicleId>"",
        "kpiId": "1",
        "kpiValue": 0.0,
        "kpiUnit": "KMPH",
        "frequency": "<daily,weekly,monthly>",
        "category": "traffic-efficiency",
        "fromDate":"2021-02-03T21:36:50Z",
        "toDate":"2021-02-05T21:36:50Z",
        "kpiValues": [
                {
                        "value": 0.0,
                        "time": "<day,week,month>"
```

```
                },
                {
                        "value": 0.0,
                        "time": "<day,week,month>"
                }
        ]
}
```

All other KPIs follow this format except the two KPIs mentioned below which have a few additional attributes:

1. Vehicle Utilization Rate:

    This KPI has the following additional attributes in the responses:

    I. timeInMotion : total time the vehicle was in motion
    II. totalTime: total time the vehicle was active
    III. numberOfStops: total stops taken by the vehicle

    Response:

```
{
        "entity": "<site,fleet,vehicle>",
        "entityId": "”<siteId/fleetId/VehicleId>”",
        "kpiId": "",
        "timeInMotion": 0.0,
        "totalTime": 0.0,
        "kpiUnit": "hours",
        "frequency": "<daily,weekly,monthly>",
        "category": "travel-and-passenger-patterns",
        "numberOfStops": "<Count of stops based on entity>",
        "fromDate": "””",
        "toDate": "””",
        "kpiValues": [
                {
                        "value": 0,
                        "time": "<day,week,month>"
                },
                {
                        "value": 0,
                        "time": "<day,week,month>"
                }
        ]
}
```

2. Kilometers Travelled with/without Passengers:

    This KPI has the following additional attributes:

    I. kmWithPassengers: total km travelled with passengers
    II. kmWithoutPassengers: total km travelled without passengers
    III. relativePercentage: relative percentage a vehicle travelled with passengers with respect to total km travelled.

Response:

```
{
        "entity": "<site,fleet,vehicle>",
```

```
"entityId": "<siteId/fleetId/VehicleId>",
"kpiId": "",
"kmWithPassengers": 0.0,
"kmWithoutPassengers": 0.0,
"kpiUnit": "km",
"relativePercentage":
"<kmWithPassenger/(kmWithourPassenger*0.01)>", // Relative
value
"frequency": "<daily,weekly,monthly>",
"category": "travel-and-passenger-patterns",
"fromDate": "””",
"toDate": "””",
"kpiValues": [
	{
		"value": 0, // Relative value
		"time": "<day,week>"
	},
	{
		"value": 0, // Relative value
		"time": "<day,week>"
	}
]
```

Appendix III: Swagger OpenAPI specifications

SHOW KPI Dashboard API - Site Linköping

This OpenAPI [22] definition contains only the query part. No responses are defined as implemented. Version: '2021-05-26'


```yaml
openapi: '3.0.2'
info:
  title: SHOW KPI Dashboard API - Site Linköping
  description: This OpenaAPI definition contains only the query part. <br>No responces are defined as implemented.
  version: '2021-05-26'
servers:
  - url: 'https://api.server.test/v1'
    description: SHOW site test server
  - url: https://SHOW-SMDP/v1
    description: SHOW SMDP Platform

paths:
  '/vehicle_info/{siteID}/{fleetID}/{vehicleID}/{attribute}:':
    summary: Get vehicle info
    description: Get vehicle info
    get:
      summary: Get vehicle info
      description: Get vehicle info
      operationId: vehicleInfoGet
      tags:
      - Site Linköping
      parameters:
      - in: path
        name: siteID
        schema:
          $ref: '#/components/schemas/SHOWSite'
        required: true
        description: Site ID
      - in: path
        name: fleetID
        schema:
          type: integer
          default: 0
        required: true
        description: Fleet ID number
      - in: path
        name: vehicleID
        schema:
          type: integer
          default: 0
        required: true
        description:  Vehicle ID number
      - in: path
        name: attribute
        schema:
          type: integer
          default: 0
        required: true
        description: Info type requested
      - in: query
        name: Frequency
        schema:
          $ref: '#/components/schemas/frequency'
        required: true
        description: Measurement time steps
      - in: query
        name: fromDate
        schema:
          $ref: '#/components/schemas/fromDate'
        required: true
        description: From date, date strin
      - in: query
        name: toDate
        schema:
          $ref: '#/components/schemas/toDate'
```

```yaml
      required: true
      description: To date, date strin
    responses:
      '200':
        description: OK
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/UserAcceptanceKPIResponse'
      '400':
        description: Bad request. User ID must be an integer and larger than 0.
      '401':
        description: Authorization information is missing or invalid.
      '404':
        description: A user with the specified ID was not found.
      '5XX':
        description: Unexpected error.

'/fleet_info/{siteID}/{fleetID}/{attribute}:':
  summary: Get fleet info
  description: Get fleet info
  get:
    summary: Get fleet info
    description: Get fleet info
    operationId: fleetInfoGet
    tags:
    - Site Linköping
    parameters:
    - in: path
      name: siteID
      schema:
        $ref: '#/components/schemas/SHOWSite'
      required: true
      description: Site ID
    - in: path
      name: fleetID
      schema:
        type: integer
        default: 0
      required: true
      description: Fleet ID, 0 means all
    - in: path
      name: attribute
      schema:
        type: integer
        default: 0
      required: true
      description: Info type requested
    - in: query
      name: Frequency
      schema:
        $ref: '#/components/schemas/frequency'
      required: true
      description: Measurement time steps
    - in: query
      name: fromDate
      schema:
        $ref: '#/components/schemas/fromDate'
      required: true
      description: From date, date strin
    - in: query
      name: toDate
      schema:
        $ref: '#/components/schemas/toDate'
      required: true
      description: To date, date strin
    responses:
      '200':
        description: OK
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/UserAcceptanceKPIResponse'
      '400':
        description: Bad request. User ID must be an integer and larger than 0.
```

```yaml
        '401':
          description: Authorization information is missing or invalid.
        '404':
          description: A user with the specified ID was not found.
        '5XX':
          description: Unexpected error.


  '/Site_info/{siteID}/{attribute}:':
    summary: Get site info
    description: Get site info
    get:
      summary: Get site info
      description: Get site info
      operationId: SiteInfoGet
      tags:
      - Site Linköping
      parameters:
      - in: path
        name: siteID
        schema:
          $ref: '#/components/schemas/SHOWSite'
        required: true
        description: Site ID
      - in: path
        name: attribute
        schema:
          type: integer
          default: 0
        required: true
        description: Info type requested
      - in: query
        name: frequency
        schema:
          $ref: '#/components/schemas/frequency'
        required: true
        description: Measurement time steps
      - in: query
        name: fromDate
        schema:
          $ref: '#/components/schemas/fromDate'
        required: true
        description: From date, date strin
      - in: query
        name: toDate
        schema:
          $ref: '#/components/schemas/toDate'
        required: true
        description: To date, date strin

      responses:
        '200':
          description: OK
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/UserAcceptanceKPIResponse'
        '400':
          description: Bad request. User ID must be an integer and larger than 0.
        '401':
          description: Authorization information is missing or invalid.
        '404':
          description: A user with the specified ID was not found.
        '5XX':
          description: Unexpected error.


components:
  schemas:
    VehicleInfoAttribute:
      type: string
      enum: ['averageSpeed','averagePassengers','vehicleUtilizationRate','operativeCost',
          'roadAccidents','kilometersTravelledWithWithoutPassengers',
          'energyUse','accelerationVariance']
      default: 'all'
```

```
frequency:
  type: string
  enum: ['daily','weekly','monthly']
  default: 'all'
fromDate:
  type: string
  example: '2021-02-03T21:36:50Z'
  description: From date, date string
toDate:
  type: string
  example: '2021-02-03T21:36:50Z'
  description: To date, date string
trafficSafetyEnvKPI:
  type: string
  enum: ['singleVehicleKmTravel','increaseAverageVehicleOccupancy',
      'ptQualityServiceEnhance','reductionCO2','reductionNoise',
      'reductionEnergyConsumption','averageSpeed','accelerationVariance',
      'hardBreakEvent','nonScheduleStop','scheduledStop',
      'serviceReliability','kmTravelWithTraveller','kmTravelWithOutTraveller',
      'avgVehicleDelay','totalIntersectionDelay','totalNetworkTravelTime',
      'modalSplit','totalMileage','totalNetworkDelay','avgNetworkSpeed',
      'numTrip','trafficFlow','LatLonDistance','LatLonHeadway',
      'energyUse','emissions','concentration','noise',
      'all']
  default: 'all'
roadSafetyKPI:
  type: string
  enum: ['safetyEnhancement','roadAccidents','conflicts','illegalOvertaking','all']
  default: 'all'
  description: 'road safety KPI'
  example: "safetyEnhancement"
SHOWSite:
  type: string
  enum: ['rouen','rennes','linkoping','goteborg','madrid','graz','salzburg',
      'carinthia','karlsruhe','aachen','braunschweig','turin','trikala','tampere','brainport','brno','copenhagen','all']
  default: 'all'
KPIUnit:
  type: string
  enum: ['accidentsPkm','conflictsPkm','eventsPkm','kmPh','m2Ps4','numberPkm',
      'seconds','percent','minutes','vehicleHours','numVehiclePkm','kWhPkm','literPkm','jPkm','gPkm',
      'mcgPm3','dB','eurPkm','eurPtrip','likert9','eurPshipment','eurPvehicle',
      'eur','number']
UserAcceptanceKPIResponse:
  type: object
  properties:
    entity:
      type: string
      enum: [“site”,“fleet”,”vehicle”]
      description: <“site”,“fleet”,”vehicle”>
    entityID:
      type: integer
      description: <siteId/fleetId/VehicleId>
    kpiID:
      type: integer
      description: 'KPI id from Spreadsheet 4.0'
      example: 1
    kpiName:
      $ref: '#/components/schemas/UserAcceptanceKPIResponse'
      description: 'Traffic Safety Environment KPI'
      example: "singleVehicleKmTravel"
    GA:
      type: string
      enum: ['Yes','No']
      example: "Yes"
      description: 'GA or Non-GA KPI'
    frequency:
      type: string
      enum: ['daily', 'monthly']
      description: "Data collection frequency"
    fromDate:
      type: string
      example: '2021-02-03T21:36:50Z'
      description: From date  date string
    toDate:
      type: string
```

```
        example: '2021-02-04T21:36:50Z'
        description: To date
      kpiValues:
        type: integer
        default: 0
        description: KPI Measure
    xml:
      name: Order
securitySchemes:
  dashboard_auth:
    type: oauth2
    flows:
      implicit:
        authorizationUrl: http://Dashboard/oauth/dialog
        scopes:
          write:kpi: modify
          read:kpi: read
  api_key:
    type: apiKey
    name: api_key
    in: header}
```