BACK END REFERENCE ARCHITECTURE AND SPECIFICATIONS

Project Title	RESOLUTE
Project number	653460
Deliverable number	D4.1
Version	10.1
State	FINAL
Confidentially Level	PU
WP contributing to the Deliverable	WP2, WP3, WP4
Contractual Date of Delivery	M14 (30/06/2016)
Finally approved by coordinator	02.08.2016
Actual Date of Delivery	05.08.2016
Authors	Laura Coconea
Email	laura.coconea@swarco.com
Affiliation	SWARCO Mizar
Contributors	Leonora Buzio (SWARCO), Claudio Griglione (SWARCO), Gianni Canepari (SWARCO), Domenico Zagari (SWARCO), Emanuele Bellini, Paolo Nesi, Michela Paolucci (UNIFI), Antonio Candelieri (CMR), Andrea Grifoni (THALIT), Gianluca Mando (THALIT), Pedro Ferreira (COFAC), Anabela Simoes (COFAC), Anastasis Drosou (CERTH), Alexandros Zamichos (CERTH), Jan-Paul Leuteritz (FRAUNHOFER)

Funded by the Horizon 2020
Framework Programme of the European Union

EXECUTIVE SUMMARY

This document is the outcome of the efforts made by the RESOLUTE consortium within the context of WP4 “Platform back-end”, aiming to define the system architecture and specifications of the network models for UTS resilience, while operationalizing the ERMG guidelines for UTS, as defined in Task 3.4. Towards the definition of the RESOLUTE system architecture, the SotA set in WP2 and the ERMG defined in WP3 were analysed in order to provide a complete set of requirements enabling an open architecture, easily adaptable to frameworks addressing similar to the RESOLUTE end-user applications.

The outcome of T4.1, described in this document, presents a generic system architecture structure that has been designed, which will allow the addition of different elements and modules, in order to address the needs of each RESOLUTE subsystem and application. It will allow and validate the easy interaction between the different modules and subsystems and, at the same time, provide back-up solutions and other alternatives related to technological risks. Thus, the resulting architecture focuses on flexible integration of different functionalities, data and processing resources. Towards this result, the logical structure of the RESOLUTE modules and their communication is provided, following a standardized and interoperable way of interchanging messages among them. Furthermore, all units that constitute the CRAMSS (the core of RESOLUTE system) are linked together by an Integration Enterprise Service Bus (i.e. in scope of T4.5), characterized by a communication protocol along with the corresponding data model for the exchange of information regarding resilience strategies, meant to ensure their synchronization/ coordination based on prioritization.

In more detail, a separate section is dedicated to provide further software and hardware specifications for all components that compose the RESOLUTE prototype to be implemented and deployed in RESOLUTE pilot sites.

Finally, the methodology used to provide risk analysis for the critical components of the RESOLUTE system is described, along with the corresponding risk results on a number of identified potential risks.

It should be noted that, although this deliverable “D4.1 Back end reference architecture and specifications” is delivered according to the RESOLUTE DoW in Month 14, the system architecture will remain an open issue till all its subcomponents and subsystems are built and all modules have been integrated to them. Thus, this document can be considered as a living document.

PROJECT CONTEXT

Workpackage	WP4: Platform back-end
Task	T4.1: Back end reference architecture and specifications
Dependencies	The reference architecture influences implementation and validation activities in WP5 and WP6. At the same time the successful implementation of the system ensures the validation of the guidelines defined in WP3.

Contributors and Reviewers

Contributors	Reviewers
Laura Coconea (SWARCO)	Paolo Nesi (UNIFI)
Leonora Buzio (SWARCO)	Evangelia Gaitanidou (CERTH HIT)
Claudio Griglione (SWARCO)	Vivien Melcher (FhG)
Gianni Canepari (SWARCO)	
Domenico Zagari (SWARCO)	
Emanuele Bellini (UNIFI)	
Michela Paolucci (UNIFI)	
Antonio Candelieri (CMR)	
Andrea Grifoni (THALIT)	
Gianluca Mando (THALIT)	
Pedro Ferreira (COFAC)	
Anabela Simoes (COFAC)	
Anastasis Drosou (CERTH)	
Alexandros Zamichos (CERTH)	
Jan-Paul Leuteritz (FRAUNHOFER)	

Version History

Version	Date	Authors	Sections Affected
1.0	13.05.2016	Laura Coconea	ToC, Introduction
2.0	03.06.2016	Laura Coconea, Gianni Canepari, Emanuele Bellini	Chapter 4: first draft of Functional Architecture
3.0	15.06.2016	All	Chapter 3 completed
4.0	23.06.2016	All	Chapter 4 first draft
5.0	22.07.2016	All	Chapter 4 completed, Chapters 5&6 first draft
6.0	29.07.2016	All	All chapters finalized, document ready for internal review
10.0	02.07.2016	Laura Coconea, Emanuele Bellini	Comments from internal review addressed
10.1	04.07.2016	Paolo Nesi	All

Copyright Statement – Restricted Content

Table of Contents

List of Figures

List of Tables

1 INTRODUCTION

1.1 Scope

This document is the outcome of the efforts made by the RESOLUTE Consortium within the context of T4.1 "Collaborative Risk Assessment and Management Support System Reference Architecture design and specification" for the definition of a modular, extendable and efficient architecture that addresses all critical components that it is composed of and provides information about their interaction in all architectural layers. In that context, the deliverable provides an introductive part containing the methodology followed to derive the system architecture that takes into account also the semantic interoperability of all components of the system, based on the common ontological framework defined, concerning hazard recognition in transportation disasters and related emergencies, as well as on the outcomes of the work performed in WP2 and WP3.

Towards the definition of the RESOLUTE system architecture, the system requirements were analysed in order to provide an open architecture, easily adaptable to frameworks addressing similar to the REOSLUTE end-user applications.

The designed architecture aims to operationalize the European Resilience Management Guidelines (ERMG, D3.5) in UTS (Urban Transport Systems) in order to enhance infrastructure resilience in daily operations. In fact, according to the equivalence principle of success and failures introduced in the FRAM, it is not necessary to identify different modalities of operation (routine vs emergency). The scope is to focus on why the things go right, which means the understanding of the everyday adjustment performed by a system to achieve the intended outcomes (e.g. delivery service). Operators should manage (dampen) functional variability increment generated by unexpected conditions (e.g. water bomb, terrorist attack) using the same tools used in daily operations.

In order to do that, the architecture should address several critical aspects. In particular, as identified in D2.2 Conceptual Framework, the sources of operational variability and the mechanisms through which it may potentially propagate and impact on system performance, as well as the resources and system capacities needed to manage and cope with operational variability, are critical elements that need to be addressed by the present software solution. Moreover, the presence of multiple decision makers acting at the same time in a system as a city (e.g. urban police, fire brigades, mobility department), requires the implementation of a strong information system able to support distributed decision making.

Thus, the designed architecture aims at supporting decisions of CI Managers and decision makers under varying conditions using data generated by the environment, so that the intended and acceptable outcomes in everyday activities makes operation as efficient and resilient as possible.

It should be noted that although this deliverable "D4.1 Back end reference architecture and specifications" is delivered according to the REOSLUTE DoW in Month 14, the system architecture will remain an open issue until all its subcomponents and subsystems are built and all modules have been integrated to them. Thus, this document can be considered as a living document that would be further enhanced/updated, if needed, during implementation/ testing process.

1.2 Target audience

Audience of this deliverable are Pilot Site Leaders plus technological partners involved in the deployment of the RESOLUTE system in each of the pilot sites.

Besides, the deliverable is written for all stakeholders interested to learn about the support architecture required for the deployment of the RESOLUTE framework as an application of the ERMG defined in the WP3.

1.3 Methodology

The architecture of a system describes its different components and the way in which they interact in order to carry out the required functionality. A system *architecture* is a description which forms the basis for a class of systems and, hence, for a set of designs.

The architecture should have the following characteristics:

- The architecture must be open; this means that all suppliers, operators and users will be able to make use of its components.
- The architecture should be technology independent and promote (when it is possible) the use of generic/ standard solutions/ data models.

The objective of a system architecture is to provide a stable basis for a working and workable system. A working system is the system that has a set of fully functioning subsystems, which cooperate in order to provide the full functionality required by the goals of the system. A workable system is pleasant to use, easy to manage and maintain during its planned lifetime.

The system architecture provides the structure around which a class of systems may be developed. It is the level at which the basis for "working and workable systems" is set up; the overall system structure.

The architecture of a system can be described at different levels: the functional and the physical level. On the functional level, which is the objective of this document, the individual subsystems are identified and the way they interact is sketched (by highlighting not only own functionalities but also interdependencies among modules defined as inputs/ outputs), while for the physical components of the system, only generic guidelines are being traced for their implementation/ deployment.

The design of the RESOLUTE system is based on the results of previous WPs. The first step in the design and development of the architecture has been to cluster and refine the requirements derived from SotA (WP2), ERMG (WP3) and user needs (mainly through the involvement in the process of local stakeholders at Pilot Sites).

On this basis, the system architecture was defined, taking into account all the components of the system in all architectural levels (functional and physical). The rough procedure followed to derive the RESOLUTE system architecture is shown in the figure below:

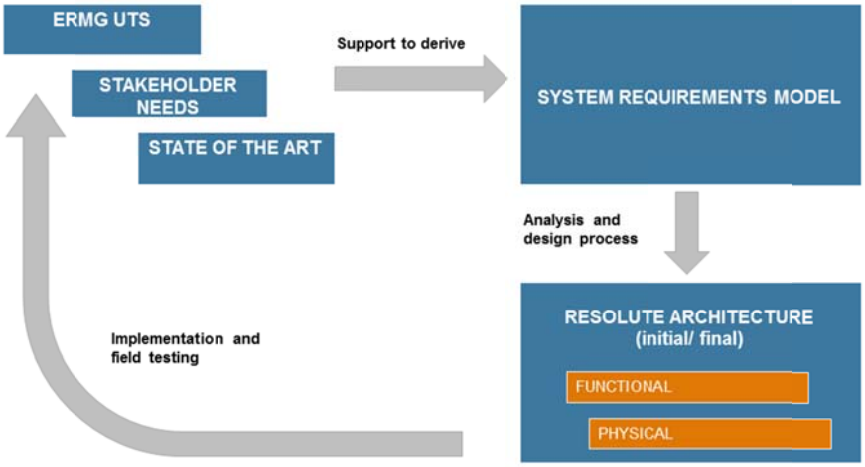


Figure 1 Methodology for the definition of RESOLUTE architecture

1.4 Structure of the document

Chapter 2 of the document inserts this work within the overall RESOLUTE concept, while chapter 3 identifies the requirements for the RESOLUTE system through the procedure described above. Chapter 4 represents the core of the document and addresses the components of the System by describing them from a functional point of view. Finally, Chapter 6 presents the methodology followed to identify and categorize risks related to the actual implementation of the RESOLUTE system, along with a detailed risk analysis and countermeasures to minimize the identified risks.

2 RESOLUTE CONCEPT

2.1 RESOLUTE framework

RESOLUTE is grounded on a wide diversity of expertise and knowledge domains. The project methodology was designed in such a way that this diversity could be investigated and analysed towards the production of a comprehensive support to the definition of requirements and specifications for project outputs. The RESOLUTE framework provides a set of criteria and a rationale for the architecture and functionalities of the tools to be developed. This is detailed in Deliverable D2.2 and aims to produce the necessary alignment between the development of the ERMG, the CRAMSS and its RESOLUTE support system.

As stated in D2.2, the endeavour of RESOLUTE is to address the system capacities and resources that are critical to ensure all operational and managerial needs under highly variable and unpredictable operational environments. Generating and managing such capacities and resources requires foremost the provision of information that supports the needs of local decision making processes. This means that the RESOLUTE systems must be capable of integrating data at system level and deliver information that is up-to-date and operationalizable in view of multiple and specific local operational needs and constraints. The system should be designed as an applied operational support that relates management guidance and requirements derived from the ERMG with multiple user constraints and needs. From a system's perspective, it should facilitate the management of potentially conflicting goals that may emerge from different local operational conditions and seek a continuous alignment of priorities around system operational goals. In the case of Urban Transport Systems (UTS), ultimately this relates to the continued provision of multiple mobility services under continuously changing

operational conditions. The following key features are derived from RESOLUTE framework and should be used as drivers for the design of RESOLUTE system:

- User jobs or missions should be mapped on to functions modelled with FRAM, which in return relate to the ERMG.
- This should create the basis for the definition of criteria for the selection of information, priorities and guidance to be provided to the user.

2.2 Identification of the users of RESOLUTE system

Target users of the RESOLUTE systems are managers and decision makers at Critical Infrastructures (CI), as well as so-called "operators" for the back-end part and the front-end of the expert User Interface (CRAMSS), while for the front-end part of the systems the target users are represented by the travellers, that interact with the system both collectively (i.e. interaction with urban public transport system or urban traffic management systems) and individually (i.e. personal interaction with their own mobile apps).

The term "operators" is used here referring to all persons working at a CI who are in charge of monitoring the flow of vehicles or passengers and whose responsibility it is in a case of emergency to redirect traffic or person flow or to coordinate mitigation attempts. These professionals are trained to know the procedures to be followed in different types of emergency, usually based on a handbook with clear instructions. They are trained at using the tools available at their workplace, including IT-infrastructure for monitoring or steering the flow of traffic.

Managers or other decision makers are persons with a higher responsibility and executive power, yet they are not necessarily involved directly in ground level operations. They are not necessarily aware of the detailed procedures to be followed in emergency situations. Their primary task is to provide the correct resources and make strategic rather than tactical decisions, thus ensuring the functioning of the CI is constantly maintained at an appropriate level.

The target groups meant to use the RESOLUTE APPs are civilian users and rescue professionals; these two main user groups may be divided into several sub-groups that may use the app for different motives, or that may play different contents on the app, according to their role in real disruptive events. Civilian users of the RESOLUTE mobile app, may in turn vary in age from 14 years to old age. Younger users would supposedly need a slightly different typology of app, which is not foreseen in the context of this project. Some civilian users are characterized as ad-hoc volunteers if they have a certain profile; they constitute a subgroup of the civilian users. Rescue professionals can belong to different organizations yet they share common characteristics: a professional training with respect to rescue activities and a more restricted age span (from 18 years of age to retirement age).

A more detailed description of System users and their profiles, as well as a specific list of identificated users in case of the two RESOLUTE Pilots, are presented in Deliverable 5.1.

3 SYSTEM REQUIREMENTS

3.1 Requirements derived from State of Art

This section highlights the key aspects of the State of the Art (SotA – Deliverable D2.1) that should be placed at the core of the development of the RESOLUTE back end platform. References are given to facilitate the identification of these key aspects by providing the relevant section numbers in Deliverable D2.1.

One of the fundamental challenges for resilience is the alignment and coordination amongst multiple local needs that are prone to continuous change. Managing high dynamics and its inherent uncertainty requires enhanced coordination, not just top-down and bottom-up, but also "laterally" amongst different local actors (D2.1: 2.5.2 and 2.5.3). Coordination and operational control requires information and communication (D2.1: 2.7.4). However, as operations become increasingly complex, the use of information and communication resources towards coordination becomes equally complex. In fact, understanding coordination needs and mechanisms also becomes increasingly challenging (D2.1: 2.5.1). Hence, as highlighted by the RESOLUTE conceptual framework (D2.2: 2.1), an in-depth system understanding, particularly in terms of interdependencies, is foremost required.

In particular, where system interdependencies are built on tighter couplings (hence less tolerant to variability), information and communication coordination mechanisms must support the continuous identification of the limits for adaptation to local needs. (D2.1: 2.5.4). The challenge is then to identify what information is relevant and to whom and when it should be communicated. Although computational capabilities have drastically improved access to information and the ability to generate data, humans are unable to keep up with such evolutions. People cannot process and make sense of the volumes of information that currently flow across complex systems (D2.1: 2.7.1) Any coordination support must therefore make use of a system understanding to generate information and communication that is actionable in the face of multiple different local realities.

From the perspective of the end-user, the system should support two fundamental system operational features:

- Common understanding of system operation and organisational structures, which will support the coordination of actions and the cohesion of decision-making processes. This also constitutes a critical support to operational control.
- The use of the system should align the requirements for overall operation coordination with local needs to adjust to continuously changing operational conditions. Context dependent factors must be taken into account, namely local availability and limitation of resources, among others.

Table 1 Requirements derived from SotA

ID	Requirement	Description	Source	Category	Validation method	Priority
SoA_01	Coordination	Information and tools should support coordinated and synchronised action and decision making. In view of a given context and target of operation, coordination needs should be identified and supported.	This should be based on the understanding of the specific roles and functions that each actor fulfils towards achieving a given operational goal. As outlined in Deliverable D2.2, the understanding of system interdependencies, namely through the development of FRAM models, provides the means to place each local reality within the scope of system operation and the fulfilment of its operational goals. This system level understanding is produced in Deliverables D3.5 and D3.7.	F	Verify how functional interdependencies are aligned with observed communication and other coordination oriented actions among actors	H (Coordinated and synchronised action and decision-making are critical for both the safety and the efficiency of operations)
SoA_02	Applicability	While supporting overall coordination, tools must also provide context dependent information, so as to produce meaningful support in view of local specific needs and requirements	Adjusting coordination principles to local context dependent features should be based on real-time feedback from local actors regarding operational conditions. This is at the core of the highly distributed nature of decision making processes within complex systems (D2.1: 2.6.1) Establishing the need for feedback could be based on a functional understanding of interdependencies, as provided by Deliverables D3.5 and D3.7.	F	Assess the extent to which local action and decision-making is aligned with overall requirements for achieving operational goals	H (Local adjustment is constant and unavoidable and therefore, should be taken into account when establishing information and action support principles. This is often undermined and can compromise credibility of the support provided and its applicability)

ID	Requirement	Description	Source	Category	Validation method	Priority
SoA_03	Validity	Knowing under what circumstances, at what time and for how long a given set of information and action support may be considered valid is critical.	This should be based on the understanding of the boundaries and limits of variability that can be sustained by various critical operational elements. Variability must be assessed in terms of change of local conditions but placed in the scope of system interdependencies in terms of the impacts that may emerge from the coupling of multiple different local variability trends (D2.1: 2.7.4 and 2.7.5).	NF	Estimating the extent to which local action and decision-making are aligned with local operational needs and requirements (i.e. if objectives are aligned with available resources)	H (Out of time or out of context execution of actions or decision-making constitutes a significant source of operational variability and of resource misuse or waste)
SoA_04	Control	Monitoring and assessing the degree of accomplishment of planned operational objectives	Risk management systems, safety and operation procedures. Should be based on operational and structural elements but control loops should be aligned with coordination needs and critical interdependencies, as previously outlined. The requirements for aligning organisational structures and processes with system interdependencies are outlined in D3.5 and D3.7.	F	Management auditing	H (Critical to support feedback needs at all levels of operation and management)

ID	Requirement	Description	Source	Category	Validation method	Priority
SoA_05	Assessing and measuring	Models, methods and tools are crucial to model, analyse and evaluate resilience of critical infrastructures, combining actual data (historical and real-time, from devices/systems as well social/human sensors) and software simulation.	As previously outlined, coordination mechanisms must target the production of locally actionable information (D2.1: 2.5). This should be built on data from monitoring and control systems; user-generated data produced by citizens/users (social network and social media data); other ICT systems such as GIS, asset management system, customer relationship management system, Supervisory Control And Data Acquisition (SCADA) system.	F	Validation is mainly performed through comparison between data generated through simulation of several scenarios and actual data – collected from historical databases, real-time sensing data or training activities	H (Crucial to support operations during the emergency as well as planning (recovery and adaptation), addressing the three layers (physical, service and cognitive)

3.2 Requirements derived from user needs

This section describes the functional requirements for the RESOLUTE system from the user perspective, as derived from interviews with stakeholders held both in the first RESOLUTE workshop and in dedicated RESOLUTE local meetings.

The RESOLUTE system should permit to create, update and delete co-operative operation control strategies for all contributing UTS applications and for all controlled objects (i.e. in this case one example for objects are urban transportation field systems but at a general level it refers to all actuation channels). Therefore, it can be said that the RESOLUTE system is a Strategy Management tool.

Every strategy is defined providing:

- control points
- activation conditions for the control points
- activation conditions for the strategy
- activation type (automatic, semi-automatic, manual)
- actions (operations control for assigned objects)
- priority

Each strategy is being activated at the moment when alert is generated at one or more control points. Several control points can be added to each strategy. A control point can be defined for every object capable of providing

measurement data to the RESOLUTE platform. The analysis of available data sources in RESOLUTE is being performed on Task 4.2 and therefore the associated report is Deliverable 4.2.

Control points are points where potential hazards can be monitored and controlled: they represent sub-systems with own structure and functions (within the main process or system) from where full or partial control can be exercised over the entire process or system. Possible control points for the UTS are:

- traffic measures (traffic flow, speed, occupancy, journey time, …)
- weather measures (road surface status, weather conditions, …)
- crowdsourcing derived measures
- pollution measures (pollutants concentration, …)

A strategy without control points can be activated only manually. Strategies activation rules are provided defining the activations conditions to be verified on every assigned control point.

The following functionalities have been considered as a basis for the definition of functional requirements:

- Resilience Strategy Definition
- Resilience Strategy Activation
- Resilience Strategy Monitoring
- Resilience Strategy Termination

While for non-functional requirements, the following categories have been addressed:

- Configuration requirements
- User Interface requirements (Note: User Interface requirements are treated in detail in D5.1.)
- Monitoring requirements

Table 2 Requirements derived form user needs

ID	Requirement	Description	Source	Category	Validation method	Priority
USR_01	User privileges	Access to strategy definition depends on user privileges	User needs/ system access	F	Lab testing	H
USR_02	Add a control point	Users can add a new control point to a strategy according to their user privileges	User needs/ strategy definition	F	Lab testing	H
USR_03	Update a control point	Users can update a control point for an existing strategy according to their user privileges	User needs/ strategy definition	F	Lab testing	M
USR_04	Enable a control point	Users access can enable a control point for an existing strategy according to their user privileges	User needs/ strategy definition	F	Lab testing	M

ID	Requirement	Description	Source	Category	Validation method	Priority
USR_05	Disable a control point	Users can disable a control point for an existing strategy according to their user privileges	User needs/ strategy definition	F	Lab testing	M
USR_06	Delete a control point	Users can delete a control point for an existing strategy according to their user privileges	User needs/ strategy definition	F	Lab testing	H
USR_07	Display of control points	Users can display control points for existing strategies	User needs/ strategy definition	F	Lab testing	M
USR_08	Semi-automatic behaviour	Every strategy should have at least a control point to be activated automatically or semi-automatically	User needs/ strategy definition	F	Lab testing	H
USR_09	Manual actuation	Strategies without control points can be activated manually	User needs/ strategy definition	F	Lab testing	H
USR_10	Control points definition	Control points can be defined for every object providing measurement data to the RESOLUTE platform	User needs/ strategy definition	F	Lab testing	H
USR_11	Activation conditions	Activation conditions should be defined for every control point added to the strategy either manually or automatically	User needs/ strategy definition	F	Lab testing	H
USR_12	Semi-automatic control	Every control point should have at least an activation condition to enable automatic or semi-automatic control of the strategy	User needs/ strategy definition	F	Lab testing	H
USR_13	Activation conditions definition	Activation type and value should be specified for every activation condition	User needs/ strategy definition	F	Lab testing	H
USR_14	Semi-automatic activation	Every control point should have at least an activation condition to be activated automatically or semi-automatically	User needs/ strategy definition	F	Lab testing	M
USR_15	Manual activation	Strategies without control points can be activated manually	User needs/ strategy definition	F	Lab testing	H

ID	Requirement	Description	Source	Category	Validation method	Priority
USR_16	Control actions definition	Control actions should be defined for every strategy	User needs/ strategy definition	F	Lab testing	H
USR_17	Control action existence	Every strategy should have at least a control action	User needs/ strategy definition	F	Lab testing	M
USR_18	Control actions assignment	Several control actions can be added to a single strategy	User needs/ strategy definition	F	Lab testing	M
USR_19	Actuation channels	Control actions can be defined for every object controllable by the common linking platform	User needs/ strategy definition	F	Lab testing	M
USR_20	Activation trigger monitoring	The system continuously monitors the activation status for active strategies	User needs/ activation	F	Lab testing	H
USR_21	Activation trigger	Strategy monitoring is performed evaluating the activation indicator of every active strategy	User needs/ activation	F	Lab testing	M
USR_22	Activation conditions	Control point measure is compared with control point activation conditions (membership) to obtain an activation indicator for every control point (0 – 100%)	User needs/ activation	F	Lab testing	H
USR_23	Combined activation conditions	Control point activation indicators are combined according to strategy activation rules to obtain an activation indicator for every strategy (0 – 100%)	User needs/ activation	F	Lab testing	H
USR_24	Actuation actions	Control actions for activated strategies are issued to controlled object	User needs/ activation	F	Lab testing	M
USR_25	Deactivation conditions	When an active strategy activation indicator becomes smaller than the activation threshold, corresponding strategy is deactivated	User needs/ termination	F	Lab testing	M
USR_26	Minimum rest time	To avoid high frequency strategy activation / deactivation in case of activation indicator values in the range around the activation threshold, minimum activation time and minimum rest time should be defined	User needs/ termination	F	Lab testing	M

ID	Requirement	Description	Source	Category	Validation method	Priority
USR_27	Strategy monitoring	Strategy monitoring makes available data on current status for all defined strategies	User needs/ monitoring	NF	Documentation	M
USR_28	Available status	Strategy monitoring makes available data on strategy activation status (active, enabled, disabled)	User needs/ monitoring	NF	Documentation	M
USR_29	Available control point information	Strategy monitoring makes available data on control point activation indicator status (0-100%)	User needs/ monitoring	NF	Documentation	M
USR_30	Activation time configuration	The system supports the configuration of the minimum activation time for a strategy that has been activated	User needs/ configuration	NF	Documentation	M
USR_31	Minimum rest time configuration	The system supports the configuration of the minimum rest time for a strategy that has been deactivated	User needs/ configuration	NF	Documentation	M
USR_32	System modularity	Strategy Manager is modular	User needs/ performance	NF	Documentation	H
USR_33	System scalability	Strategy Manager is scalable	User needs/ performance	NF	Documentation	H
USR_34	User connectivity	Users might be connected simultaneously (apart from the limits eventually set by common linking platform)	User needs/ performance	NF	Documentation	M
USR_35	Heterogeneous data sources	The system needs to integrate data coming from multiple and heterogeneous data sources	User needs/ performances	F	Lab testing	H
USR_36	Multi-decision (distributed decision) making	The system needs to allow specific views and needs to support multi-decision making	User needs/ system management	F	Lab testing	H

3.3 Requirements derived from ERMG

The requirements presented in this section have derived from the work performed by the Consortium in WP3 and are a direct consequence of the ERMG.

Within the WP3, the aim of D3.5 was to produce a first version of the European Resilience Management Guidelines (ERMG), as a product of work performed within T3.2. Following the project structure, these guidelines were operationalized for the Urban Transport System (D3.7) that will further be tested in the pilots of WP5. In order to reach this objective there was a need to translate the ERMG definition into concrete aspects that impact on the design and future implementation of the RESOLUTE system itself. This section is meant to ensure this connection and has been produced by analysing each function described in D3.5/ D3.7 with the scope of deriving concrete system requirements to be considered in the design phase of RESOLUTE, considering both characteristics of single components and also interdependencies among them.

Table 3 Requirements derived from ERMG

ID	Requirement	Description	Source	Category	Validation method	Priority
ERMG_01	Monitor road traffic	The system should keep information about the road traffic	Monitor Urban Transport Safety and Security	F	Lab testing & field testing	H
ERMG_02	Monitor extreme weather conditions	The system should keep information about the weather and especially for extreme weather conditions	Monitor Urban Transport Safety and Security	F	Lab testing & field testing	H
ERMG_03	Monitor unexpected events	The system should monitor for unexpected events such as terrorist attacks, traffic incidents etc.	Monitor Urban Transport Safety and Security	F	Lab testing & field testing	H
ERMG_04	Monitor the urban transport system	The system should monitor: 1.The number of boarding/alighting travelers at stops 2.the position of every urban transport vehicle 3.the travelling time for a route	Monitor Urban Transport Safety and Security	F	Lab testing & field testing	H
ERMG_05	Safety and security alerts.	The system should be able to generate triggering and communicating safety and security alerts for the appropriate authorities or the general public.	Monitor Urban Transport Safety and Security	F	Lab testing & field testing	H
ERMG_06	Recognize the most unsecure places	The system should collect every day data related to unsecure conditions and reported violent behaviors at transport spots, and map the most unsecure places/routes and terminals in order to prioritize secure movements in cases of emergencies.	Monitor Urban Transport Safety and Security	F	Lab testing & field testing	H
ERMG_07	Recognize the safe places	The system should collect every day data for safe movements (roads and paths well maintained and visible) in order to be able to provide safe and secure guidance in emergency conditions.	Monitor Urban Transport Safety and Security	F	Lab testing & field testing	H
ERMG_08	Data security	The system should have secure authentication mechanisms in order to protect private data from unauthorized persons	Monitor Urban Transport Safety and Security	F	Lab testing & field testing	H

ID	Requirement	Description	Source	Category	Validation method	Priority
ERMG_09	Data privacy	Personal data should be carefully treated, according to existing standards and regulations	Monitor Urban Transport Safety and Security	F	Documentation	L
ERMG_10	Triggering alerts without causing panic.	In case of an emergency the system has to communicate with the general public in a way that does not produce panic movements and actions	Monitor Urban Transport Safety and Security	F	Lab testing & field testing	H
ERMG_11	Operate according to the pre-defined safety and security plan	The system should comply with the guidelines of the safety and security program/plan	Monitor Urban Transport Safety and Security	F	Lab testing & field testing	H
ERMG_12	Prioritize protection guidance	The procedures for protection guidance should be prioritized in order to secure most vulnerable travellers, with specific instructions for kids, the elderly, and people with reduced mobility	Monitor Urban Transport Safety and Security	F	Lab testing & field testing	M
ERMG_14	Guide the travelers.	The system should provide safety and security information to travellers, in a time responsive way and with alternative paths to follow, along with other appropriate information (means and form) covering different travellers needs	Monitor Urban Transport Safety and Security	F	Lab testing & field testing	H
ERMG_15	Automated system operation	The required tasks should be executed in an automatic manner so that they would not require additional time from the employees and in order employees to be protected as well if needed.	Monitor Urban Transport Safety and Security	F	Lab testing & field testing	H
ERMG_17	Extract meaningful knowledge from the data.	The system should be able to extract knowledge from the data and translate such knowledge into a meaningful mobility dashboard for supporting decisions and consequently actuation of road network management strategies.	Monitor Operations	F	Lab testing & field testing	H
ERMG_18	Check the data quality profile for each data source	The system should check the data quality profile defined for each data source addressing the following dimensions: Relevance (Fitness), completeness, consistency, accuracy, timeliness, integrity, accessibility and clarity, comparability, and coherence.	Monitor Operations	F	Lab testing & field testing	M
ERMG_19	Detect failure of the monitoring infrastructure	The system should be able to detect and inform about any fault in the monitoring infrastructure	Monitor Operations	F	Lab testing & field testing	H

ID	Requirement	Description	Source	Category	Validation method	Priority
ERMG_20	Monitor resources status and supply	The system should keep updated information on the status and supply of all resources.	Monitor Resource availability	F	Lab testing & field testing	H
ERMG_21	Monitor data and services access	The system should continuously monitor the access to data and services.	Monitor Resource availability	F	Lab testing	H
ERMG_22	Report resources use	The system should report updated information on resources use.	Monitor Resource availability	F	Lab testing	H
ERMG_23	Inform on resource allocation and fLs	The system should generate information on resource allocation and the understanding of their fLs.	Monitor Resource availability	F	Lab testing & field testing	H
ERMG_24	Inform about resource supply failure	The system should be able to promptly inform about resource supply failure defining also the related causes according to the specified protocol and procedures.	Monitor Resource availability	F	Lab testing & field testing	H
ERMG_25	Activate alarm in case of resource unavailability	The system should activate suitable alarms when data and/or services resources are no longer available.	Monitor Resource availability	F	Lab testing & field testing	H
ERMG_26	Deploy contingency resources in case of failure	The system should be able to deploy contingency resources in case of failures or disruptions in critical resources.	Monitor Resource availability	F	Lab testing & field testing	H
ERMG_27	Track resources dysfunctions	The system should keep track of all resources dysfunctions.	Monitor Resource availability	F	Lab testing	M
ERMG_28	UI usability	The UI of the system should fulfil usability requests.	Monitor Resource availability	Non F	Documentation	M
ERMG_29	Total user accessibility	The information displayed by the system should be totally accessible to users.	Monitor Resource availability	Non F	Documentation	M
ERMG_30	Easy, secure and successful HMI	The system should alL for easy, comfortable and secure interactions and successful human-machine dialogues.	Monitor Resource availability	Non F	Documentation	M

ID	Requirement	Description	Source	Category	Validation method	Priority
ERMG_31	Crowd sensing of user generated data	The system should monitor social media and smartphones data through crowd sensing in order to collect user generated feedback in terms of comments, scores, images on the services and specific geolocalised elements as bus stop, rack, red light, plates, underpass, bridges, etc.	Monitor user generated feedback	F	Lab testing & field testing	H
ERMG_32	Analyse crawled user data to extract opinions and sentiments	The system should be able to analyse crawled user data towards opinion/sentiment analysis.	Monitor user generated feedback	F	Lab testing	H
ERMG_33	Infer citizens' security and safety after an event	The system should be able to infer the perceived level of security and safety of the citizens after an event.	Monitor user generated feedback	F	Lab testing & field testing	H
ERMG_34	Identify criticalities reported by the users	The system should be able to identify possible criticalities reported by the users which represent barriers to the acceptance/usage of a specific service.	Monitor user generated feedback	F	Lab testing & field testing	M
ERMG_35	Analyse data for incident detection	The system should be able to analyse smart devices' geolocation data (crawling detection) and tweet contents (cause of crawling) towards detecting transport incidents.	Monitor user generated feedback	F	Lab testing & field testing	H
ERMG_36	Provide social/human sensing data to involved stakeholders	The social/human sensing data collected by the system should be made available for all stakeholders involved.	Monitor user generated feedback	F	Lab testing & field testing	H
ERMG_37	Ensure privacy and security of public crawled data	The system operation should guarantee privacy and security of the public data crawled from the web and social networks according to the internal and local policies.	Monitor user generated feedback	Non F	Lab testing & field testing	H
ERMG_38	Support transportation service during standard operation	Coordinate urban transport service during ordinary/normal operations.	Coordinate Service delivery	F	Lab testing & field testing	H
ERMG_39	Support post-event transportation service coordination	Coordinate urban transport service during or after an accident	Coordinate Service delivery	F	Lab testing & field testing	H
ERMG_40	Provide evacuation information	Provide information about delays or evacuation information to the users through the p.a. or the public information display	Manage awareness and user behaviour	F	Lab testing & field testing	H

ID	Requirement	Description	Source	Category	Validation method	Priority
ERMG_41	Integration with alternative sensory channels	Integrate alternative sensory channels for visual & hearing impairments along with instructions for handling wheelchairs during emergencies	Manage awareness and user behaviour	F	Lab testing & field testing	H
ERMG_42	Provide information to external software apps	Provide emergency information to external software apps for mass alerts and bulk SMS that may instruct people on the move in cases of emergencies	Manage awareness and user behaviour	F	Lab testing & field testing	H
ERMG_43	Integration with station/ terminal/ road advertising	Integration with station/ terminal/ road advertising to provide information	Manage awareness and user behaviour	F	Lab testing & field testing	H
ERMG_44	Support SOP	Provide specific SOP in case of event	Develop / update procedures	F	Lab testing & field testing	H
ERMG_45	Integrate Intelligent Transportation Systems (ITS)	Integrate information from transport and traffic management systems	Manage ICT resources	F	Lab testing & field testing	H
ERMG_46	Integrate CCTV	Integrate Video Surveillance Systems to display live and recorded video streams to the user	Manage ICT resources	F	Lab testing & field testing	H
ERMG_47	Integrate Physical infrastructures device diagnostic	Receive diagnostic information from physical infrastructure devices for ordinary and extraordinary maintenance	Maintain UTS physical/cyber infrastructures	F	Lab testing & field testing	H
ERMG_48	Monitor Cyber infrastructure	Receive diagnostic information from cyber infrastructure for ordinary and extraordinary maintenance	Maintain UTS physical/cyber infrastructures	F	Lab testing & field testing	H
ERMG_49	Support cooperation during the emergency	Respond effectively by coordinating actions during the emergency by implementing preliminary actions and facilitating the cooperation between the different involved actors, according to the minimum availability of service (e.g. to evacuate population or re-direct traffic fL)	Coordinate emergency actions in UTS	F	Lab testing & field testing	M
EMRG_50	Support the quick recovery of the service and level of usage	Respond quickly services and procedures after a disruption in order to avoid propagation (within the same infrastructure or towards interconnected infrastructures) as well as avoid/limit/mitigate social, economic and psychological impacts	Restore/repair operations	Non F	Lab testing	M

ID	Requirement	Description	Source	Category	Validation method	Priority
ERMG_51	Learn, directly from data which is related to features of the event, operations performed (and timing), characteristics of the infrastructure	Perform ex-post analysis to learn, directly from data, which could be the most appropriate adaptations and improvements to perform in order to improve resilience of the critical infrastructure to events. Data are related to: • the event "per-se" (as characterized by data collected through ICT systems and information reported by people involved • the status and properties of the critical infrastructure, with respect both to the physical of the system and the service (hours, day, season when the event occurred) • the actions performed by the actors as well as the behaviour of the citizens/users involved in the disruptive event Analysis of this data allS for a better identification of criticalities/vulnerabilities in the physical infrastructure and in processes/procedures. Corrections may be analysed – ex-post – in order to define the most suitable improvements to adapt the critical infrastructure	Provide adaptation & improvement insights	F	Lab testing & field testing	H

ID	Requirement	Description	Source	Category	Validation method	Priority
ERMG_52	Integrate data from ICT systems and social/human sensing to obtain a more holistic, comprehensive and complete characterization of the event, its impact and the effectiveness of actions	The ICT based systems, platforms and solutions usually used to monitor the condition of the infrastructure and the level of service, as well as those used during the emergency, allL to collect and store in-home data (usually structured) which can be stored in order to be analysed – ex-post – to better understand the features of the event, its impact and the effectiveness of the current guidelines and good practices. However, a huge amount of external information is usually lost, even if it could be extremely relevant to deeply understand, model and analyse the event and evaluate the current resilience capabilities of the UTS. People involved in the event can for sure provide a lot of information which can complete the data and information collected through ICT systems and reported by the operators, respectively. In some cases the "human/social sensing" could be the only solution to collected information (e.g. about a specific area not covered by monitoring systems and not yet reached by the emergency management operators).	Collect event information	F	Lab testing & field testing	H

4 FUNCTIONAL REFERENCE ARCHITECTURE

This section aims to illustrate the functional blocks of the RESOLUTE architecture and their interaction.

The RESOLUTE system has been designed using as a starting point the requirements presented in Chapter 3 and coherently with the preliminary high-level design presented in the DoW. The RESOLUTE system is meant to offer a complete suite of functionalities for resilience management in terms of identification and mitigation of emergency situations.

The architecture relies on open common communication interfaces used to collect data and run control actions on respective actuation system. For the definition of the functional architecture it has been chosen the three-layer architecture model.

The three-layer architecture model, which is the fundamental framework for the logical design model, segments an application's components into three tiers of functionalities. These tiers do not necessarily correspond to physical locations on various computers on a network, but rather to logical layers of the application. In otherHow the pieces of an application are distributed in a physical topology can change, depending on the system requirements.

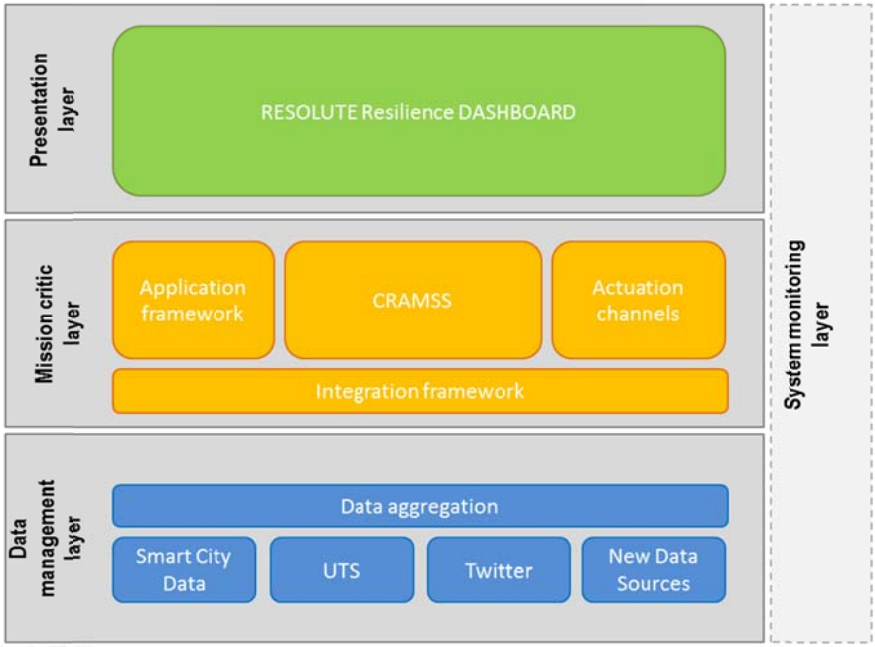


Figure 2 RESOLUTE Three-tier architecture

Following are brief descriptions of the functionalities allocated to each layer:

- The presentation layer, or user services layer, gives a user access to the application. This layer presents data to the user through the resilience DASHBOARD and optionally permits data manipulation and data entry. The two main types of user interface for this layer are the traditional application and the Web-based application. Web-based applications now often contain most of the data manipulation features that traditional applications use. Also part of this layer is the functionality of events/information publication in a semi-automatic way towards all relevant actors involved in the management of the addressed Critical Infrastructure (CI).

- The middle layer, or the mission critic layer, consists of the core system intelligence. This is the layer where developers can solve mission-critical business problems and achieve major productivity advantages and it is hosting the CRAMSS component, which represents the core of the RESOLUTE system. These components can be used to enforce business rules, such as business algorithms and legal or governmental regulations, and data rules, which are designed to keep the data structures consistent. Because these middle-layer components are not tied to a specific client, they can be used by all applications and can be moved to different locations, as response time and other rules require.

- The data management layer, or data services layer, deals with the data acquisition and aggregation from various field sources: both existing and RESOLUTE-specific. Following it also deals with the persistent data usually stored in a database or in permanent storage. This is the actual DBMS access layer. It can be accessed through the integration framework present in the mission-critical layer.

During an application's life cycle, the three-layer approach provides benefits such as reusability, flexibility, manageability, maintainability, and scalability. You can share and reuse the components and services you create, and you can distribute them across a network of computers as needed. You can divide large and complex projects into simpler projects and assign them to different programmers or programming teams. You can also deploy components and services on a server to help keep up with changes, and you can redeploy them as growth of the application's user base, data, and transaction volume increases.

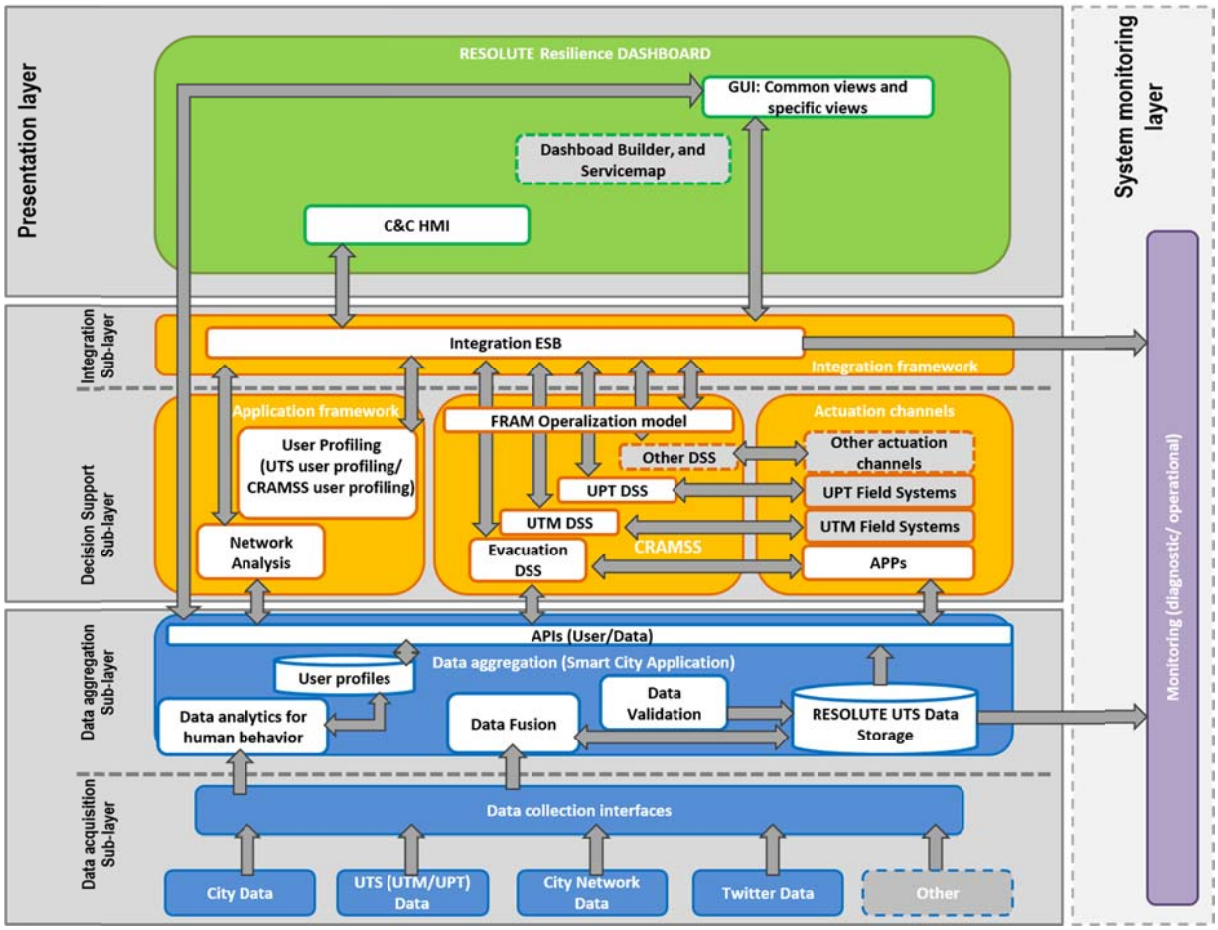


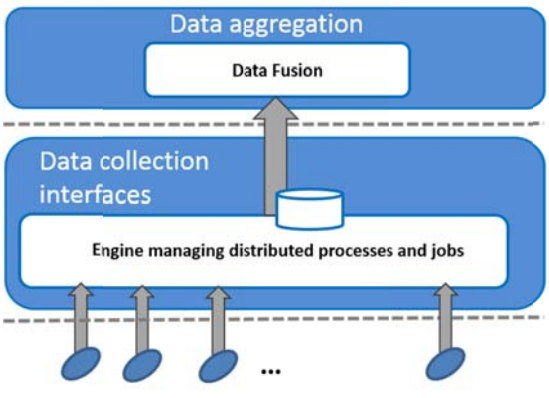
Figure 3 Functional blocks in the RESOLUTE three-layer architecture

The following sections describe the functional blocks of the system.

4.1 Data management layer

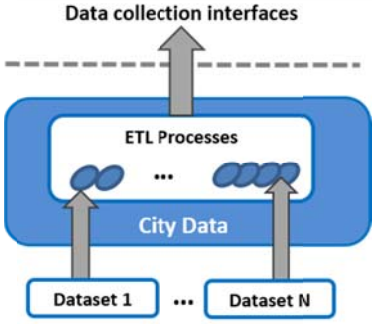
4.1.1 Data acquisition sub-layer

4.1.1.1 Data collection interfaces

Component	Data Collection Interfaces
Requirements Matching	SoA_01, SoA_04, SoA_05, USR_35, ERMG_01, ERMG_02, ERMG_03, ERMG_04, ERMG_20, ERMG_21, ERMG_31, ERMG_51, ERMG_27
Function	This Component plays the role of manager of all the different processes and information ingested from the blocks: Smart City Data, UTS Data, Apps Data, Other. It consists of an engine for distributed task scheduling and manages a set of distributed instances of running agents performing concurrent tasks. This engine with cluster functionality allows adding distributed processes and defining jobs, without service downtime, feature that is particularly relevant for managing the real time data and the semi-Real Time (periodical) data. In order to realize this module, the following tools will be used, already developed by Disit Lab and at disposal of the RESOLUTE consortium, (https://github.com/disit): • DIM tool, Data Ingestion Manager: it allows the creation of Open Data records, setup and management of the ingestion processes. The setup and management of ingestion processes allows selecting tasks to execute both in the creation step and during the life of data for update purposes. A set of different tasks is available and could be executed singularly or joined, (http://www.disit.org/6732, http://www.disit.org/drupal/?q=node/6764). o Actions realised in RESOLUTE and involving this tool: i) tool usage; ii) allocation of ETL processes; iii) Programming of the DIM; iv) DIM maintenance. • DISCHES tool, Distributed Scheduler, https://github.com/disit/sce-frontend. It consists of a set of distributed instances of running agents performing concurrent tasks. DISCES engine with cluster functionality allows adding distributed nodes and defining jobs, without service downtime. o Actions realised in RESOLUTE and involving this tool:This tool will be fundamental also for the 'Data Fusion' and the 'Data Validation' blocks, in the Data aggregation layer.
Component architecture	
Input	• ETL Processes, with the respective associated descriptions (e.g.: validity, data licence, data source info, etc.), necessary to manage the information flow. • Other kinds of jobs
Output	• Ingested information, ready for the Data Fusion step

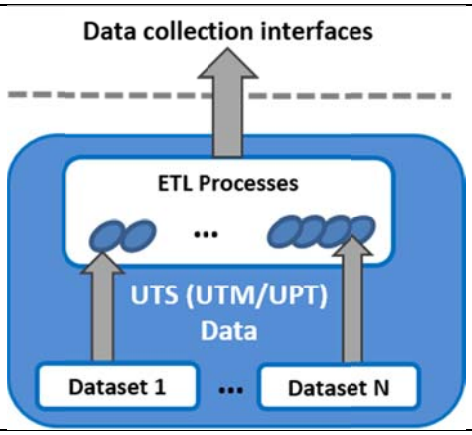
4.1.1.2 City Data

Component	City Data

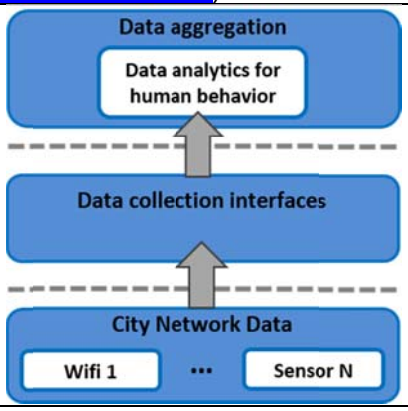
Requirements matching	SoA_01, SoA_04, SoA_05, USR_35, ERMG_01, ERMG_02, ERMG_03, ERMG_04, ERMG_20, ERMG_21, ERMG_31, ERMG_51
Function	This component is realized in order to collect Open and Private Data coming from the city and territory. The major part of these kinds of information is published by governmental organizations as Open Data, in different file formats such as html, xml, csv, shp, etc., and typically provides information that may present links to web resources. Moreover, the information is usually static, but can also be distributed in real time or semi-real time modality (e.g. number of visitors in a museum, city tour that are starting in a specific time and are going to make a guided visit of the city, weather forecasts for the different municipalities, events in the city, etc.). In order to acquire a wide range of these different kinds of data, as described above, a set of ETL (Extract Transform and Load) processes will be realized. The velocity of data ingestion is related to the frequency of data update, and it allows distinguishing static from dynamic data. Actually, data are available in the City of Florence (and in some cases in all the Tuscany) coming from the PAs and typically covering a set of different aspects and can be reused in RESOLUTE: location of points of interest (POIs) on the territory (including, museums, tourism attractions, restaurants, shops, hotels, etc.), major GOV services, ambient data, weather status and forecast, etc. In addition to the above dataset other ETL processes will be developed in order to integrate datasets strictly connected with the resilience aspects, such as: i) underpasses, bridges, etc. ii) risks and vulnerability analysis; iii) relations with Critical Infrastructures; iv) Wi-Fi flows (people); v) flood levels; vi) landslides and earthquakes, etc. vii) accidents, flooding, presences at schools, etc.
Component architecture	
Input	• Datasets coming from City
Output	• ETL Processes to be automatically launched by the data collection interfaces block and having the following features: o collecting the data coming from different sources o capable to manage both static and real time data o associated to a description, sent to the 'Data collection interfaces' block to launch each process in the right modality (e.g.: validity, data licence, data source info, etc.) and correctly manage the information flow.

4.1.1.3 UTS Data

Component	UTS Data
Requirements matching	SoA_01, SoA_04, SoA_05, USR_35, ERMG_01, ERMG_02, ERMG_03, ERMG_04, ERMG_20, ERMG_21, ERMG_31, ERMG_51
Function	This Component is realized in order to collect a wide set of data regarding mobility and transport aspects in a smart city. Actually there are data available, already existing in the City of Florence (and in some cases in all the Tuscany) mainly coming from Intelligent Transportation Systems, ITS, for bus/train/etc. management. Moreover, they also describe and model solutions for managing and control. These data can be reused in RESOLUTE and will be integrated with new information related to traffic flows, and other static or real time information.

Component architecture	
Input	• Dataset related to the city mobility aspects (mainly coming from Public Transport Agencies and PAs)
Output	• ETL Processes to be automatically launched by the data collection interfaces block and having the following features: o collecting the data coming from different sources o capable to manage both static and real time data o associated with a description, sent to the 'Data collection interfaces' block to launch each process in the write modality (e.g.: validity, data licence, data source info, etc.) and correctly manage the information flow.

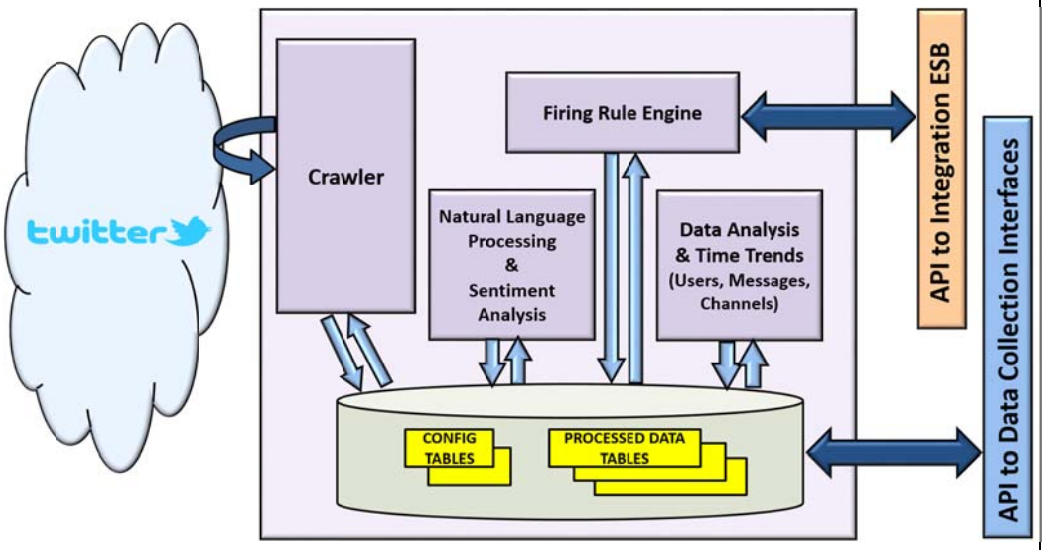
4.1.1.4 City Network Data

Component	City Network Data
Requirements matching	SoA_01, SoA_04, SoA_05, USR_35, ERMG_01, ERMG_02, ERMG_03, ERMG_04, ERMG_20, ERMG_21, ERMG_31, ERMG_51, ERMG_31, ERMG_32
Function	The mobile apps play an important role of our lives, for this reason in a smart city context it is fundamental to take care of their presence. It can be useful to receive real time information coming from the Wi-Fi, or sensors networks having information related to the citizens' habits, collecting their activities and that are fundamental, for example to study the citizen's flows, to establish what the citizens' preferred services are, what can be improved for them, to collect their ideas and necessities, etc. The data collected through this component will be directly used and managed to elaborate new knowledge in the 'Data Analytics for human behaviour block' (in the data aggregation layer). In order to ingest this kind of information, the following tools are at disposal of CERTH and DISIA for developing new Apps and instruments: • An Engine for the users' tracking: that can be integrated in other Apps • An Engine to realized Apps such as the 'Firenze dove, cosa... Km4City', (http://app.comune.fi.it/app/a0025.html, http://www.disit.org/6780) • Development of: Testing Applications, such as the one realized for the ' Mugnone 2016 ' simulation (http://www.km4city.org/app)
	

Input	• Information coming from wifi or sensor networks
Output	• Statistical data about user profile, describing their activities done by the users in a city, etc.

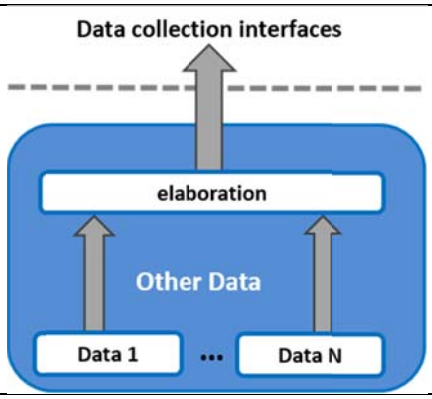
4.1.1.5 Twitter Data

Component	Twitter Data
Requirements matching	SoA_01, SoA_04, SoA_05, USR_35, ERMG_01, ERMG_02, ERMG_03, ERMG_04, ERMG_20, ERMG_21, ERMG_31, ERMG_51, ERMG_36, ERMG_32
Function	This functional component is devoted to collect and analyze data streams from social media (specifically, Twitter data) through the creation of dedicated thematic channels, which can be tuned to monitor one or more Search Queries on Twitter with a sophisticated and expressive syntax. The twitter Data Module is realized thanks to the use of the tool, already developed and at disposal for the RESOLUTE project: Twitter Vigilance (http://www.disit.org/tv) having the above described features. This tool continuously operates in two different working modalities at the same time: offline and real time. In the former mode, it accumulates Twitter data and performs different types of analysis, such as statistics and trends about messages and users, Natural Language Processing (NLP) and Sentiment Analysis (SA) of text messages, in order to provide analysis results, temporal trends and statistics (at level of channel and search, users etc.), sentiment polarity at channel and search level for keywords, adjectives and verbs. By this way, the contextual knowledge of the RESOLUTE Data Storage will be enriched. In the latter mode, this component performs a real-time analysis (both statistics/trends and NLP-SA) on specifically defined channels, in order to monitor and detect critical conditions and providing alert signals or other type of actions, suggestions etc. This component will interface, through dedicated APIs, with the Integration ESB layer and the Data Collection Interfaces. In more detail, the Twitter Vigilance tool presents the following functionalities: • Create one or more channels to monitor and collect Twitter data; • Create and activate multiple channels, that may use new or the same Search Queries; • Provide public access to their channels analysis (as in the channels accessible without registration); • Download data sets for refined analysis; • Provide full access at the channel history of User's content per channel, per search, per users, etc.; • Perform visual view throughput graphs and export them in different graphic format; • Perform analyses at level of channel, search, users, tweets, retweets, etc.: o trends of the Search Queries as reported in the above figure; o distributions on population and activities of users; o distributions about other tags/keywords; o geographic distribution of twitters of single or multiple channels; o distributions regarding tweet and re-tweets. • Perform Natural Language Processing (NLP) and Sentiment Analysis (SA) at level of channel and search, showing temporal trends and providing separate analysis for extracted keywords, nouns, adjectives, verbs, hashtags and mentions; An instance of the Twitter Vigilance Real Time tool, realized for RESOLUTE and presented in draft at the ' Mugnone 2016 ' simulation, studied and created in collaboration with the Civil Protection of Florence, is at disposal in order to be taken as a sample for possible future actions related to resilience activities in the RESOLUTE context, (http://disit.org/rttv/index.php?p=chart_singlechannel&canale=mugnone2016). Moreover, the tool, having the above described features, will be enriched with the following aspects and services that will be added and will complete the RESOLUTE 'Twitter Data' block,

	thus developing a new tool (Twitter Vigilance Real Time tool): • Twitter Vigilance tool: o development of the tool realized to calculate firing rules, periodic calculation; o Adding of rules for resiliency • Twitter Vigilance Real Time tool: o development of the tool realized to calculate firing rules, real time calculation; o Providing real time analytics on processed data about messages, users, data processed through NLP and SA; o Monitoring and detecting critical conditions, through the Firing Rule Engine operating on the Real Time module, and consequently providing alerts, message via email or mobile. o Adding of rules of firing for resilience and risk management o Integration with Enterprise Service Bus (ESB) and the FRAM
	
Input	• Twitter Data Stream
Output	• Statistical data about user profiles and messages grouped by thematic channels and searches (e.g. hashtags, mentions) • Natural Language Processing and Sentiment analysis of messages • Temporal trends and resumes

4.1.1.6 Other

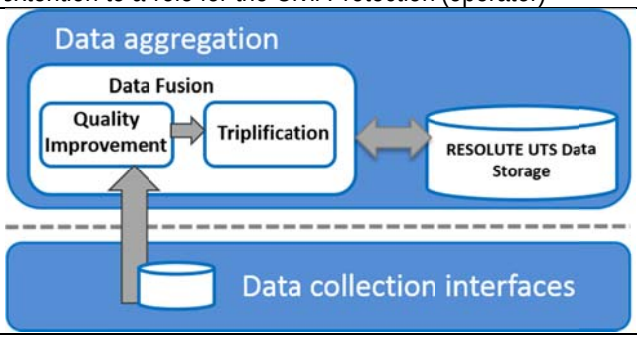
Component	Other
Requirements matching	SoA_01, SoA_04, SoA_05, USR_35, ERMG_01, ERMG_02, ERMG_03, ERMG_04, ERMG_20, ERMG_21, ERMG_31, ERMG_51
Function	This Component provides a set or tools to create ETL processes to provide data related to the city. Many different kinds of datasets are relevant to analyze and solve problems connected to resilience aspects. In this sense the Platform is Open to ingest all the data relevant to reach to RESOLUTE goals.

Component architecture	
Input	• Other Datasets useful to reach the RESOLUTE goals
Output	• ETL Processes or Algorithms to be automatically launched/processed by the data collection interfaces block and having the following features: o collecting the data coming from different sources o capable to manage both static and real time data o associated with a description, sent to the 'Data collection interfaces' block to launch each process in the write modality (e.g.: validity, data licence, data source info, etc.) and correctly manage the information flow.

4.1.2 Data aggregation sub-layer

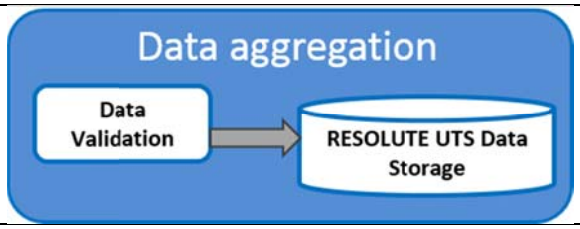
4.1.2.1 Data fusion

Component	Data Fusion
Requirements matching	SoA_01, SoA_02, USR_35, RMG_01, ERMG_02, ERMG_03, ERMG_04, ERMG_20, ERMG_21, ERMG_31, ERMG_51, ERMG_17,
Function	The work done in this Component can be divided in the following phases: • Quality Improvement (QI): the datasets ingested can contain a set of different errors and inaccuracies. For this reason it is necessary to increase the quality of the datasets, if necessary, in order to produce reliable and useful information for next applications, taking into account the followng aspects: completeness, consistency, accuracy, absence of duplication, integrity, etc. • Triple generation and Aggregation (T): this phase generates RDF triples from QI data for every dataset that has been processed. The triples created are stored in the RESOLUTE Semantic Data Storage and are based on a model built on relationships defined within referring to a City Ontology (that will be realized starting from the multi-domain and Open Source Ontology Km4City, http://www.disit.org/6506) with the purpose to semantically integrate all the different information coming from the Data Acquisition sub-layer in an RDF Data store (Resolute Data Storage). • Reconciliation (R) task tries to solve the lack of coherence among indexed entities referring to the same concept but coming from different data sets. The starting ontology model (Km4City, http://www.disit.org/6506), will be enriched taking into account the following aspects related to the resilience activities: i) underpasses, bridges, etc. ii) risks and vulnerability analysis; iii) relations with Critical Infractures; iv) wi-fi flows (people); v) flood levels; vi) landslides and earthquakes, etc. In order to realize this module, the following tools will be used, already developed by Disit Lab and at disposal of the RESOLUTE consortium, (https://github.com/disit): • RIM tool, RDF Indexing Manager: for the RDF index generation the RDF Index

	Manager produces a script according to the index descriptor and the RDF store target, (http://www.disit.org/6708, http://www.disit.org/drupal/?q=node/6764). o Actions realised in RESOLUTE and involving this tool: i) tool usage; ii) rebuilding the index (sometimes takes twice) • RDF Licensing CAS, IPR management (http://www.disit.org/drupal/?q=node/6877). Services on smart city are typically based on a large amount of data sets (static and real - time) coming from different sources. Therefore, a data aggregation service providing access to aggregated data with the aim of setting up mobile and web applications has to be very careful and provide support for managing the different licensing models by: i) Aggregating data; ii) Providing aggregated data and services via API with coherent licenses. o Actions realised in RESOLUTE and involving this tool: i) tool usage; ii) extention to a role for the Civil Protection (operator)
Component architecture	
Input	• Ingested dataset
Output	• Triples for the RDF Store (Resolute Semantic Data Storage)

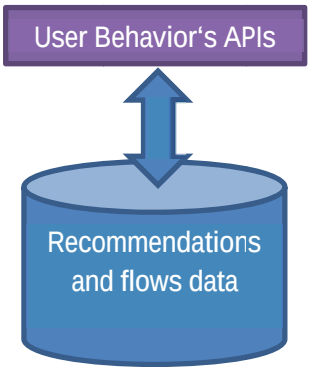
4.1.2.2 Data validation

Component	Data Validation
Requirements matching	SoA_03, RMG_01, ERMG_02, ERMG_03, ERMG_04, ERMG_20, ERMG_21, ERMG_31, ERMG_51, ERMG_18,
Function	This component is created in order to apply validation and verification techniques that allow checking the correctness of the system. Given the complexity of structure implemented for each dataset, a validation of the triples has been executed in order to verify that the realized DataCube is well-formed. This validation is executed by running a set of SPARQL queries that verifies the consistency of the fundamental constraints. In order to realize this module, the following tools will be used, already developed by Disit Lab and at disposal of The RESOLUTE consortium: • SPARQL frontend tool, realized to make guided sparQL queries on the RDF Km4City Storage (http://log.disit.org/sparql_query_frontend) • Service Map (http://servicemap.disit.org/WebAppGrafo/mappa.jsp). It is a map based query interface to access at the Transport Systems and Smart City Ontology and reasoning tools (get the document) populated by more than 300 open data coming from Florence, and Tuscany, about: trasportation, points of interest, services, educational instititions, bus stops, parking, and much more. o Actions realised in RESOLUTE and involving this tool: A) tool usage, with the purpose of developing new Apps. B) visualization of the added RESOLUTE datasets: i) underpasses, bridges, etc. ii) risks and vulnerability analysis; iii) relations with Citrical Infractures; iv) wi-fi flows (people); v) flood levels; vi) landslides and earthquakes, etc. • LOG / LOD. This is a web interface tool connected with the Service Map: the users may select the bus lines, and the sevice to discover the correlations and to access at

	the semantic model expressed and represented as Linked Open Graph. o Actions realised in RESOLUTE and involving this tool: i) tool usage, with the purpose of developing new Apps.
Component architecture	
Input	• Query made on the RESLUTE Semantic Data Storage
Output	• Statistics and the System validity

4.1.2.3 RESOLUTE User Data Storage

Component	RESOLUTE User Data Storage
Requirements matching	SoA_01, USR_34, ERMG_08, ERMG_09
Function	A set of services related to Data Analytics for Human behavior APIs it already avalilable and at disposal for RESOLUTE to provide (http://km4city.org): • Information about flows of people configuration from Apps tracking and from wi-fi among geographical clusters • Information about active users • Information about recommendations (sent, viewed, disliked) • Information about users or profile activity • Information about time of the day activity • Information about users' speed • Engagement tool • Recommended tool: it allows to access all configurations and data produced and referred to: i) automatic production of recommendations for users of the App, machine learning solutions for users' behavior; ii) users' behavior trends, general statistics, single user statistics, etc. iii) heatmap for category of users, etc. dynamic maps are available, many types with many parameters. The access to the dynamic maps is allowed only under specific request. Moreover, the following will be developed: • APIs involving resilience aspects • APIs capable to provide data and hints to the CERTH and DISIA Apps • People Flow Mobile tool located on Apps of different kinds • OD Matrix, People flow monitoring from Apps and wifi • Recommendation tool and suggestion production via computing algorithms • Security related aspects and user behavior for safety • Engagement tool: o Addition of rules to the virtuous behavior resilience and disaster relief o Addition of rules for civil protection operators

Component architecture	This component will be implemented using standard APIs for accessing human behavior data (clustered and specific) from the database. 
API	Flows
Input	Profiles or none, clusters, time range
Output	User related flows in the provided time range among clusters
API	Active users
Input	Profiles or none, time range
Output	Active users in the provided time range
API	Recommendations
Input	Profiles or none, time range
Output	Recommendations sent in the provided time range
API	User/Profile activity
Input	Profiles or none, time range
Output	User/profiles activity in the provided time range
API	Time range
Input	Profiles or none, time range
Output	Users activity in the provided time range
API	Users speed
Input	Profiles or none, time range
Output	Users speed in the provided time range

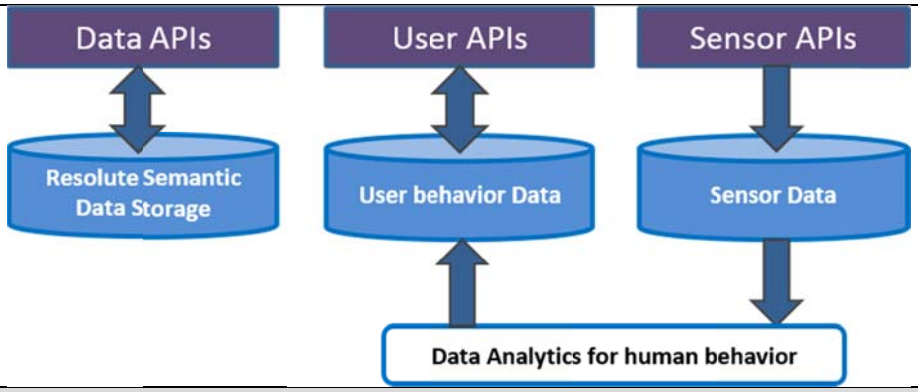
4.1.2.4 RESOLUTE Data Storage

Component	RESOLUTE Data Storage
Requirements matching	SoA_05, USR_32, USR_33, USR_35, ERMG_08, ERMG_09
Function	The RESOLUTE Semantic Data Storage, will be realized based on the Storage already available in the Disit Data center and containing all the information related with the City (not only information on the Urban Transport System). It will be realized based on the RDF tool and store already realized and deployed in the DISIT center, (http://km4city.org). It will contain static information as the road graph, with details on how the roads are connected, the services, monuments and facilities that are present in the city, but also the public transport information as bus routes, bus stops, time tables etc. The store will contain also dynamic data as the bus positions, the status of road sensors, the status of parkings etc. The data will be integrated using km4city ontology and other base ontologies (dcterms, foaf, schema, wgs84, etc.). The

	use of ontologies will allow inference. Moreover, using semantic queries supported by inference, other information can be discovered. The following features will be added: • Management of Data coming from crowd sourcing tools: alert and damage information
Component architecture	This component will be implemented using an RDF Store available as open source, supporting geographic queries, SPARQL 1.1 queries with at least RDFS reasoning and full text search.
Input	• Triples to be stored on the store (as RDF/XML, turtle, NTriple) • SPARQL queries (via standard SPARQL 1.1 HTTP Protocol)
Output	• Query result as XML or JSON

4.1.2.5 APIs (on Data)

Component	APIs (on Data)
Requirements matching	SoA_05, USR_35, ERMG_08, ERMG_09
Function	Data APIs A set of Data access APIs is already available and provides access to the Km4City RFD Semantic data Storage: • information on infrastructure present on a geographic area (circle, rectangle, path, polygon); • information on a specific service with realtime information (e.g. cars/hours passing on a road sensor); • information on public transport services as the last position of buses, the routes of the bus lines, the passing hours on a bus stop etc.; • other information using direct semantic SPARQL queries when not available with other APIs Moreover, additional APIs will be developed and added to those above described: • APIs for specific resilience aspects • APIs for the integration with a set of DSSs • APIs for the integration with a set of Mobile Apps In general, the APIs could be authenticated using HTTPS protocol with client certificates. User APIs • See the component 'Data Analytics for Human behavior, User Profile DB and User APIs' Sensor APIs: • Information coming from Wifi and sensor networks • Information coming from mobile APPs Moreover, additional APIs will be developed and added to those above described: • Information related to resilience sensors • Developing of a more detailed confirmation output

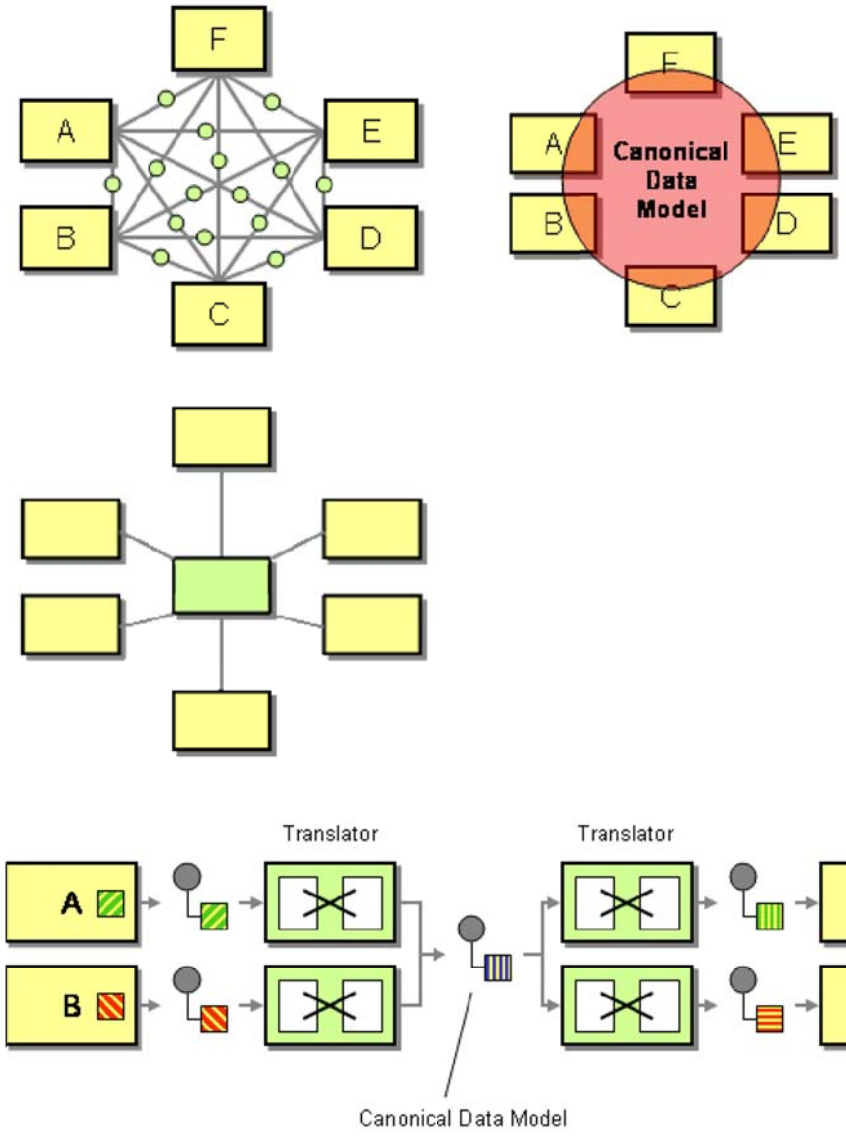
Component architecture	
API	Get information on infrastructures on a geographic area
Input	• the geographic area expressed as: o a GPS position with a max distance o a rectangualar area o a generic WKT (Well Known Text) path or polygonal area • the categories of entities to be located in the area (Bus Stops, Sensors, etc.)
Output	• the list of entities matching the geographic area with minimal information (name, type, position, uri)
API	Get information on a specific service
Input	• the identifier of the service
Output	• detailed information on the service and for some service also realtime specific information
API	Information on Public Transport line, route or bus stop
Input	• the bus stop, line or route of which provide information
Output	• the lines/routes passing at a bus stop or the bus stops of a line/route
API	SPARQL Query
Input	• the SPARQL query text
Output	• the SPARQL query result as XML or JSON
API sensors	Sensor data submission
Input	• Json related to: a GPS position, Wifi identifier, user role, time, etc.
Output	• Submission confirmation

4.2 Mission critic layer

4.2.1 Integration framework

4.2.1.1 Integration ESB

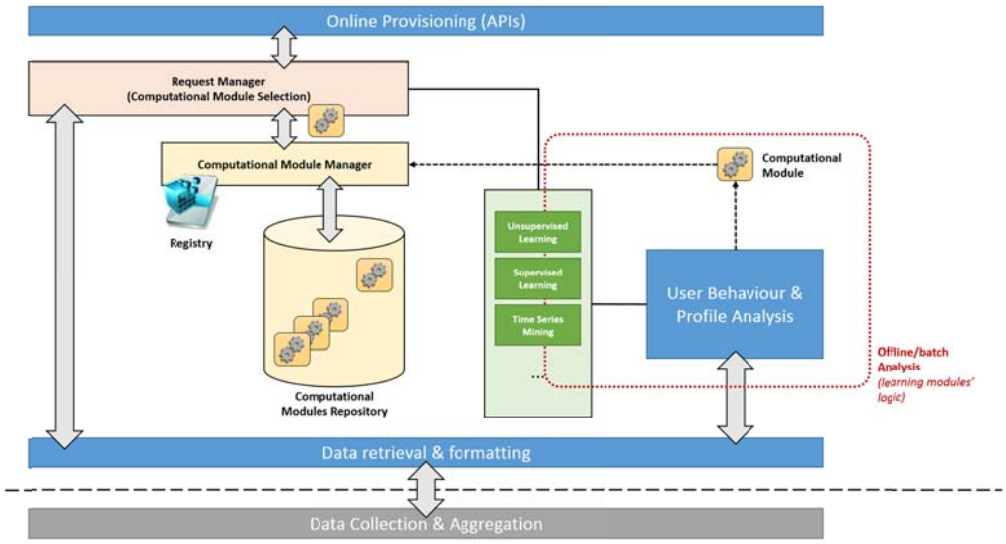
Component	Integration ESB
Requirements matching	SoA_01, SoA_02. SoA_03. SoA_04, USR_32, USR_33, ERMG_05, ERMG_08, ERMG_09, ERMG_38, ERMG_39, ERMG_40, ERMG_44, ERMG_45, ERMG_46, ERMG_47, ERMG
Functionality	RESOLUTE shall be capable to manage events and alarms in real-time, thus it is necessary to adopt a system that allows a publish/subscribe mechanism. The Integration ESB will be in charge of this functionality, providing a message broker where other RESOLUTE components (FRAM, DSSs) could connect to receive events. Moreover, given the high number of eterogenous connected applications, each with its own internal data format, the Integration ESB will adopt the Canonical Data Model Pattern to minimize inter-dependencies.

	An appropriate data format will be defined during the development of the Integration ESB.
Component architecture	
Input	• RESOLUTE collected and processed field data
Output	• RESOLUTE canonical data model

4.2.2 Application framework

4.2.2.1 User profiler

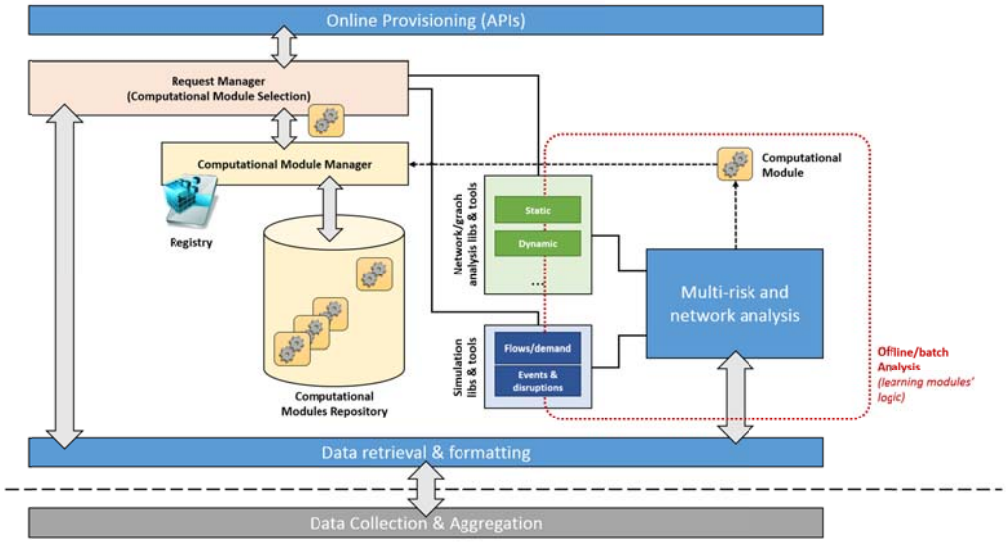
Component	User Profiler
Requirements matching	SoA_05,ERMG_05, ERMG_15, ERMG_17 Also supporting the following CRAMSS' specific requirements: ERMG_06, ERMG_07, ERMG_14, ERMG_50, ERMG_51

Functionality	Profiling management is a functionality aimed at supporting the analysis and a deeper understanding of the typical behaviours of people, in particular UTS users, during normal service conditions, emergency situations or repair/recovery phases. According to the available set of data, algorithms, software libraries and computational modules will be implemented – or further developed, in case specific approaches are available in the literature – to infer the possible categories of behaviours, their relation with one or more specific conditions (normality, emergency, etc.) and the specific "features" of each behaviour. Although most of the literature is focused on the analysis of the behaviour of users of urban mobility and UTS services, in RESOLUTE also other relevant stakeholders will be taken into account, in particular UTS staff and external teams involved in the emergency management. With respect to this consideration, also skills of people involved and availability of information – provided in time and in place – will be considered as relevant features for the profiling management. This data will permit to have a more comprehensive and complete overview with respect to the classic analysis of "movement data", usually aimed at identifying typical mobility paths and patterns. As final result, profiling management will collect and offer a set of computational modules to support decisions and facilitate the provision of information, in particular to UTS users, to manage flow both in normal conditions and during emergency.
Component architecture	
Input	• Passenger-counts over time, even at different time scales, and for every line • Origin-Destination (time-dependent) matrices • Structure of the overall UTS, timetable of the transportation service (scheduled and real-time), car/bike sharing availability, usage and location over time • Position of people over time, for instance acquired through Wi-Fi networks as well as smartphone apps (in any case by using pseudonymised data) – in particular referred to both citizens (UTS users) and relevant actors involved during the emergency management • Data from traffic management systems and other monitoring and control systems • Information about procedures and skills of people involved during the management of an emergency or a service disruption
Output	• Categories of behaviours (typical and anomalous) directly inferred from data • Automatic identification of modifications in typical behaviours ("concept shift/drift"), possibly at different time scales according to the availability of data at high sampling rate • Modelling and forecasting of transportation service demand over time – at different

	time scales and horizons (short, medium long term) according to data • Estimating response of people, in terms of variations in transport service demand and flows over the network and over time, even according to some actions (e.g. information provision during the emergency or after the recovery) • Information provision – to both UTS users and relevant stakeholders – for a more effective management of service demand and flows with respect to the current condition (normality as well as emergency/disruption)

4.2.2.2 Network Analysis

Component	Network Analysis
Requirements matching	SoA_05, ERMG_05, ERMG_10, ERMG_15, ERMG_17 Also supporting the following CRAMSS' specific requirements: ERMG_06, ERMG_07, ERMG_14, ERMG_38, ERMG_39, ERMG_50, ERMG_51
Functionality	This component of the functional architecture is devoted to store, organize, manage and offer several algorithms and models, mostly based on network science and graph analysis, to support and operationalize resilience concepts in the management of UTS. Network/graph analysis applications, tools and software libraries are very useful to model different aspects of the UTS. The first obvious category of analysis is related to the infrastructure (physical layer), which can be naturally modelled as a graph: vertices are stations, edges are connections between two stations. Starting from this "basic" representation is possible to retrieve simple, but in any case relevant, graph-based measures which can be used to identify, even in very complex networks, structural vulnerabilities (e.g. links whose interruption may induce a disconnection between different areas of the city: impairment in terms of reachability). However, resilience is related to more than physical disconnection. Edges of the graph representing the UTS may be weigthed according to the flow of passengers in order to evaluate the impact on adverse events in terms of impairment of the service (service level), a more subtile and useful information which may also support the comprehension of the "social response" of the UTS users and citizens in general. Different types of data may be used to weight edges to provide different insights: • aggregated data may offer an overall evaluation of the ability of the UTS to absorb an event; this is usually useful for planning modifications and expansions of the transport network; • actual stored data, with a frequent sampling rate, can be used for modelling flow in the network over time (according to the available sampling rate); this is very useful to identify typical as well as anomalous conditions in the level of service and to evaluate the impact of adverse events on the level of service depending on the demand-supply over time; • actual (near) real-time data from monitoring and control systems may be used to characterize the current condition of the UTS with respect to those identified by using historical data • simulated/forecasted data to evaluate the impact of possible events starting from specific condition; in this case useful information can be inferred such as simulation of different scenarios – created or forecasted – to understand how flow should be redirected depending on current condition of the UTS, event and level of demand; in particular to understand which information – and where/when – should be provided to users in order to manage demand while increasing satisfaction and perceived quality/security (cognitive level). Since UTS usually consists of different transportation sub-networks (e.g. bus, underground, railways, etc.) the tools and software libraries for graph/network analysis will be also used to

	model multi-modal graphs, where removing edges/nodes represents disruptive events, while generating "virtual" edges represents actions to suggest redirection of flows (e.g. transfer from an underground station to a nearby bus stop). In order to enable a multi-risk analysis, different events have to be mapped into "basic" modifications on the graph representing the UTS (e.g. removal of one or more edges/nodes, reduction of the maximum flow admitted on an edge, etc.) Finally, the application layer will also allow for modelling/simulating cascading effects at different levels: • within the physical infrastructure of the UTS (e.g. how delays modify flows overtime inducing congestion and increasing/propagating delays) • from interconnected (ICT) systems and critical infrastructures.
Component architecture	
Input	• Structure of the overall UTS, timetable of the transportation service (scheduled and real-time), car/bike sharing availability, usage and location over time • Passengers counts over time, even at different time scales, and for every line • Origin-Destination (time-dependent) matrices and flows data • Position of people over time, for instance acquired through wi-fi networks as well as smartphone apps (in any case by using pseudonymised data) – in particular referred to both citizens (UTS users) and relevant actors involved in the emenrgency management • Data from traffic management systems and other monitoring and control systems • Information about procedures during the management of an emergency or a service disruption – in particular alternative mobility options or information provided to manage the flow of UTS users • Data from UTS simulation software applications
Output	• Vulnerabilities at physical level (from structural static analysis) • Vulnerabilities at service layers (from the combined analysis of structural and services data) – in this case vulnerabilities vary over time according to time-dependent variations in demand and flows • Graph analysis based resilience measures, both static and dynamic (i.e. varying over

	time) • Simulation of impact (at both physical and service level) of an event affecting a set of nodes and/or edges in the graph model • Predicting the impact of an event by starting from actual real-time data about demand, flows and events • Estimating, through simulation, the cost-benefit of alternative actions which could be performed to manage demand and flows • Modelling the interconnections and propagations of disruptive events – within the UTS and from other interconnected urban Critical Infrastructure

4.2.3 Actuation channels

4.2.3.1 UPT field systems

Component	UPT field systems
Requirements matching	N/A (Existing system)
Functionality	The following Florence tramway integrated subsystem could be exploited to cover RESOLUTE requirements: • Public Information Displays • Public Announcement System • Public WiFi (both on-board and on-ground)
Component architecture	N/A (Existing system)
Input	• RESOLUTE procedure, Text information, Voice information
Output	• Text information displayed on the Public Information displays, Voice announcements broadcasted through the public announcement system

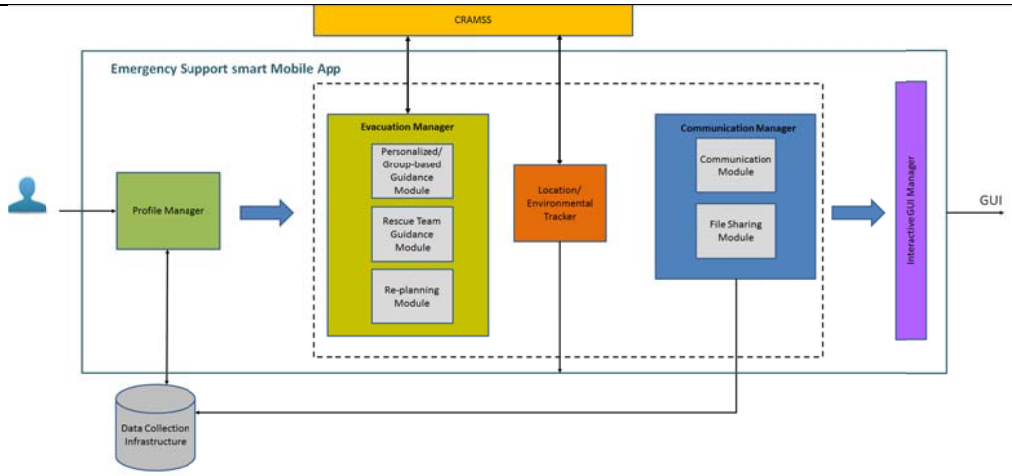
4.2.3.2 UTM field systems

Component	UTM field systems
Requirements matching	N/A (Existing system)
Functionality	The UTM system integrates various sub-systems meant to ensure the traffic control at city level. The specific scope of this component in RESOLUTE is to ensure the collective guidance/ actuation of road network users (pedestrians, public transport travelers and drivers) in relation to a certain modification of the road network. In case of such a modification, this would automatically have an impact on the controlled field systems: • VMS • UTC • LTZ
Component architecture	N/A (Existing system)
Input	• RESOLUTE UTM DSS
Output	• Text and graphics displayed on the VMS • Change of the management rules of the LTZ • Urban Traffic Control Systems

4.2.3.3 APPs

Component	APPs

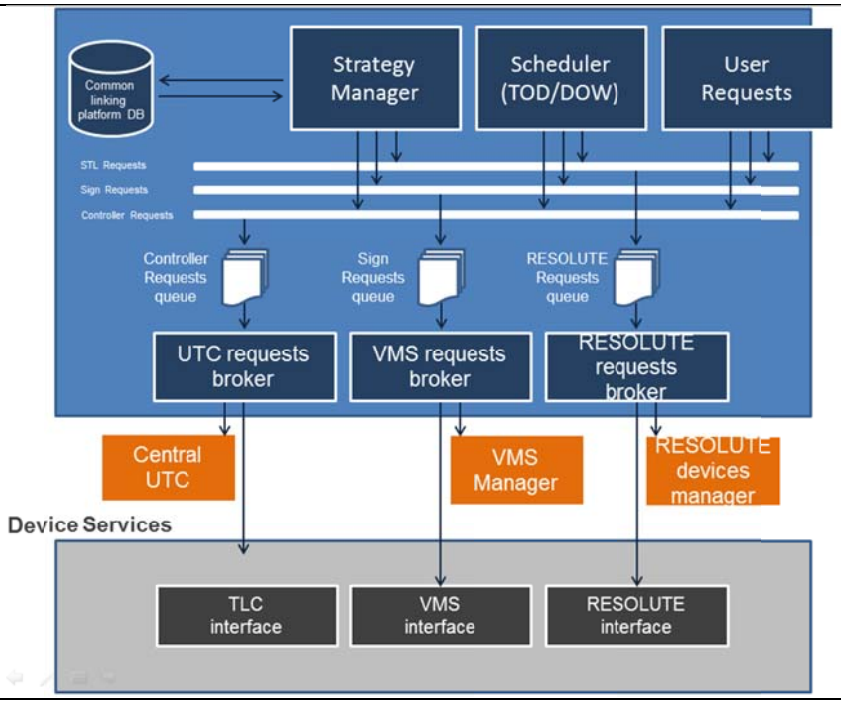
Requirements matching	SoA_01, SoA_02, SoA_03, SoA_04, USR_28, USR_29, USR_30, USR_31, USR_32, USR_34, ERMG_08, ERMG_09, ERMG_10, ERMG_11, ERMG_14, ERMG_15, ERMG_17, ERMG_18, ERMG_28, ERMG_29, ERMG_30, ERMG_31, ERMG_33, ERMG_34, ERMG_36, ERMG_37, ERMG_39, ERMG_40, ERMG_42, ERMG_49
Functionality	The main role of the Emergency Support smart Mobile App (ESMA) is to optimally utilize the CRAMSS back-end functionalities (T5.2), in order to facilitate the real-time evacuation guidance interface, and to provide information to the back-end (e.g. environmental, behavioural, sensorial network, etc.). The app is developed to be available on every platform and to support diverse end-users roles (i.e. rescue teams, individual travelers, groups of citizens, etc.). The ESMA takes as input data: • Users' profiles • CRAMSS output Every user has the opportunity to create an account and to formulate his/her profile by adding personal details, demographics and/or by declaring his/her abilities. Moreover, the system supports dynamic profiling of every user (e.g. often visited POIs, often followed routes, average speed of walking, etc.). Users are also able to update their profiles and to customize the layout of the app based on their preferences. The created profiles are stored in the Data Collection Infrastructure. The responsible module for the above functions is the Profile Manager, which is also responsible for the users' authentication and authorization. The core module of the ESMA is the Evacuation Manager, which take the following inputs from the CRAMSS: • Personalized routing evacuation guidelines • Group routing evacuation guidelines • Rescue team information & action planning The evacuation manager receives the above inputs and manages them according to the users' profiles/needs. The Evacuation Manager is divided into three sub-modules: • The Personalized/Group-based Guidance Module: real time personalized or grouped navigation/evacuation (indoor/outdoor) guidance, in order to support self-rescue or to divert passenger/group flow in emergencies. • The rescue Team Guidance Module: I. Assigns dynamically roles to users. II. Formulates rescue teams III. Provide guidance to the rescue teams in order to reach points of interest (POIs) in time. • The Re-planning Module: Allows user to accept or reject the proposed (by the CRAMSS) rescue plan and/or to make a re-planning request. The Communication Manager is responsible for communication-related data collection and data mining and is divided into the: • Communication Module: Users (rescuers/citizens) are able to post messages about their or the emergency's situation during a crisis, thus creating location-aware hit maps that describe the level of emergency, sentiment, etc. • File Sharing Module: The exchange of multimedia content is available, facilitating thus, the enhanced communication of the resilience strategy to be followed and the descriptive communication/illustration of the ongoing situation by rescuers/citizens. Except from the communication-related data collection, the app extracts location and environmental (e.g. temperature, humidity, etc.) information from user's devices. Background monitoring is applied in order to extract hints (e.g. correlations, etc.) about the level of

	resilience by combining both individual- and group-related information. All the above modules communicate with the Interactive GUI Manager which produces the interactive graphical interface displayed to the user.
Component architecture	
Input	• Peronalizes navigation guidance • Group navigation guidance • Rescue team information and action planning • User's profile
Output	• Re-planning request • Information collection from the app users (e.g. location, environmental, behavioural, posts, etc.) • Graphical User Interface (GUI)

4.3 CRAMSS

4.3.1 UTM Decision Support System

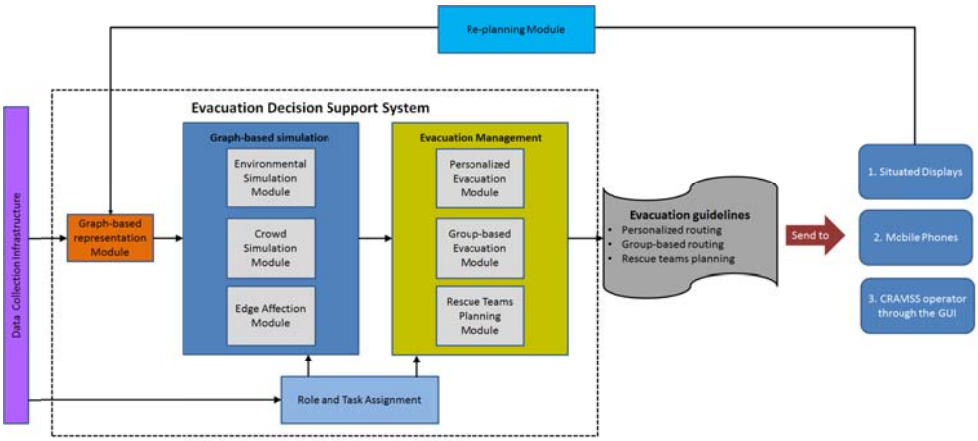
Component	UTM Decision Support System
Requirements matching	SoA_01, SoA_02, SoA_03, SoA_04, USR_02, USR_03, USR_04, USR_05, USR_06, USR_07, USR_08, USR_09, USR_10, USR_11, USR_12, USR_13, USR_14, USR_15, USR_16, USR_17, USR_18, USR_19, ERMG_01, ERMG_11, ERMG_15, ERMG_35, ERMG_39, ERMG_49
Functionality	The ITS DSS is one of the components of the CRAMSS which, through the implementation of strategic traffic management, enables co-operative operations control by means of definition and automatic identification of control strategies for both daily-life and emergency situations. Every strategy is defined providing: • control points • activation conditions for the control points • activation conditions for the strategy • activation type (automatic, semi-automatic, manual) • actions (operations control for assigned objects) • priority

	ITS DSS performs continuous monitoring of selected control points to evaluate the activation conditions for all the defined strategies. When one (or more) strategy activation conditions are verified, all corresponding control actions are activated. Strategy activation can be: • automatic: control actions are activated directly without prompting the user • semi–automatic: user is prompted before activating control actions • manual: strategy actions are started upon user request regardless the activation conditions ITS DSS offers strategy monitoring services to easily identify: • running strategies • strategy condition (activation status, control points status, …) Running strategies can be terminated: • automatically, when activation conditions are no longer verified; • on user request. Strategic traffic management operations are logged in the RESOLUTE common linking platform logbook where they can be retrieved for offline analysis and statistics. Strategy manager covers the following ITS applications (in terms of monitoring, and decision-support system when identifying pre-set network situations): • Urban traffic control • VMS control • Parking management • Streetlight control According to the foreseen architecture solution, the ITS DSS sends all required commands to proper device brokers (ITS actuation channels).
Component architecture	

Input	• UTS Field Systems: traffic-related data/ information • FRAM operalization module: emergency actuation strategy, feedback on proposed strategies
Output	• UTS Field Systems: control action for ITS applications • FRAM operalization module: traffic management strategies

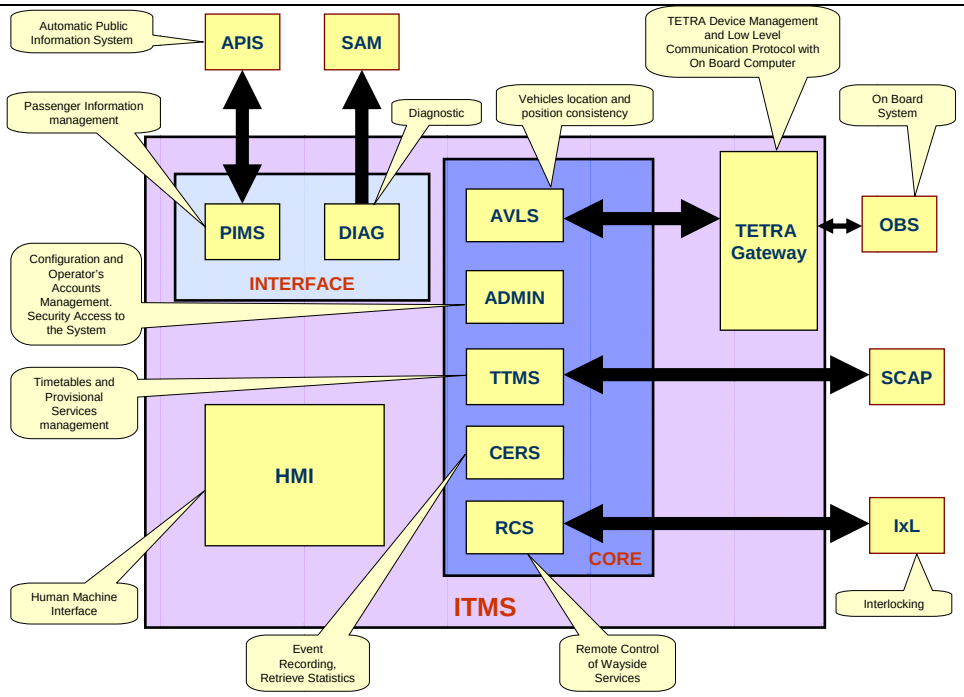
4.3.2 Evacuation Decision Support System

Component	Evacuation Decision Support System
Requirements matching	SoA_01, SoA_02, SoA_03, SoA_04, USR_01, USR_02, USR_03, USR_04, USR_05, USR_06, USR_07, USR_18, USR_27, USR_28, USR_30, USR_31, USR_32, USR_33, ERMG_01, ERMG_04, ERMG_08, ERMG_09, ERMG_11, ERMG_12, ERMG_14, ERMG_20, ERMG_21, ERMG_24, ERMG_25, ERMG_28, ERMG_29, ERMG_30
Function	The Evacuation Decision Support System is a major building block of the CRAMSS application. Its main role is to provide evacuation guidelines to individuals (personalized guidance) and/or groups of people in case of emergencies. These guidelines include both the optimal evacuation routes for people in danger, but also to the optimal allocation of the rescuers and the rescue resources (e.g. cranes, tracks, dozers etc.) with respect to the current emergency situation. A key characteristic of the Evacuation Decision Support System is that it will take into account information from other systems developed in WP4 (e.g. Behavioral Gathering and Processing modules), in order to understand the current situation and possibly predict future outcomes, such as the population distribution over space and time, or the state of road network. Artificial intelligence algorithms will be applied in order to efficiently combine and analyze the information from multiple data sources in real-time, and provide the best evacuation guidelines, from a set of Pareto optimal strategies, with the ultimate goal of reducing the total number of human life casualties. Special emphasis will be given on the modeling of the emergency situation as a search tree, where nodes represent different states and edges represent different decisions. Multiple artificial intelligence algorithms will be considered and tested, in order to efficiently traverse the modeling tree and identify the best decisions, so as to minimize the number of casualties. The Evacuation Decision Support System takes as input data from the Data Collection Infrastructure of RESOLUTE for analysis using the implemented methods. Specifically, it takes as input: • Road network • Traffic-related data/ information • Location/Situation of people and resources • Emergency actuation strategy The module also can take as input specific role assignments (rescuers or to-be-rescued) for people in the area in order to be used by the analysis methods included in it. These data are used to formulate a graph representation of the environment and the actors that can act in it (people, rescue teams, trucks etc.). Based on this graph representation, the Evacuation Decision Support System utilizes three analytical modules for the generation of Evacuation Guidelines, which are responsible for identifying the new network situation based on the why the involved actors act: • Environmental Simulation Module: Estimates the new situation based on the vehicle movements in the area. • Crowd Simulation Module: Estimates the new situation based on the movements of the Crowd of people involved in the hazardous situation, including both rescuers and to-be-rescued people. • Edge/Arc Affection Module: Estimates the availability of each of the network

	edges/arcs (e.g. roads, bridges etc.) based on the current and future situation in the area. Based in these three graph-based simulation modules, three types of analysis are performed by three modules: • Personalized Evacuation Module: A personalized guidance module that takes into account the characteristics (e.g. disabilities, age, etc.) of each individual in order to provide optimal evacuation guidelines to the specific individual. • Group-based Evacuation Module: A guidance module that analyzes an entire group of people as a unique actor, and provides the same evacuation guidelines to all of them at the same time. • Rescue Teams Planning Module: A plan that contains an order to actions that must be performed by the rescue teams in order to minimize the number of casualties. The goal of this module is provide Evacuation Guidelines, and more specifically: • Personalized routing • Group routing • Rescue teams planning The Evacuation Guidelines are communicated to the involved parties through situated displays or/and by SMS to their mobile phones. All the Guidelines are also available to the CRAMSS operator through the use of the application GUI.
Component architecture	
Input	• Road network,Traffic-related data/ information, Location/Situation of people and resurces, Role assignments, Emergency actuation strategy
Output	• Evacuation guidelines, i.e. Personalized/Group routing and Rescue teams planning.
Impact on other modules	•

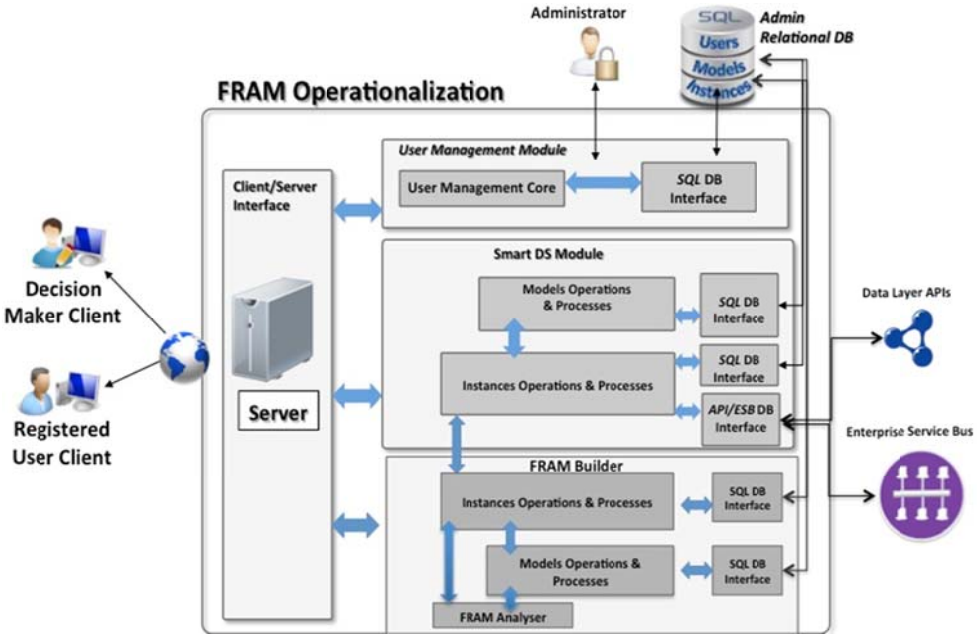
4.3.3 UPT Decision Support System

Component	UPT Decision support System
Requirements matching	SoA_01, SoA_02. SoA_03. SoA_04, ERMG_04, ERMG_05, ERMG_08, ERMG_09, ERMG_010, ERMG_11, ERMG_12, ERMG_14, ERMG_15, ERMG_19, ERMG_38, ERMG_39, ERMG_40, ERMG_44, ERMG_45, ERMG_49
Functionality	UPT Decision support system is a software application located at the Operational Control Centre able to combine data flowing from Signalling Controllers and On Board Units in an integrated manner so as to provide a fully featured system to monitor and control the tramway

	operations. Localization and monitoring of the vehicles and supervisory of the tramway Signalling system is achieved through a "distributed" architecture. At the OCC, the UPT DSS application is in charge of collecting, managing and dispatching data coming from the Wayside Controllers and On Board Units. Each of the typical Tramway functions can be treated allowing the operators to generate timetables, track and retrieve events, visualize and follow on the display the trams' position and their current status, manage train regulation, remotely control wayside devices, receive and acknowledge warnings and alarms. UPT DSS is a system that manages the tram localization. It is able to detect the tram position on the line and to manage the tram regulation by early/delay messages. It manages the ground equipment status, sets main line and depot route requests for tram parking. UPT DSS displays the status of equipment like signals, switches, rail track circuits, trams, wayside controller and road signals and it provides the interface to manage the Timetable. The system provides all functionalities to manage the entire tramway in a single integrated HMI.
Component architecture	
Input	• Train speed, Train id, Train number, Train diagnostic status, Train operative status, Ahead/Delay, Train service
Output	• Train emergency services, Public information display information, Public announce control

4.3.4 FRAM operalization

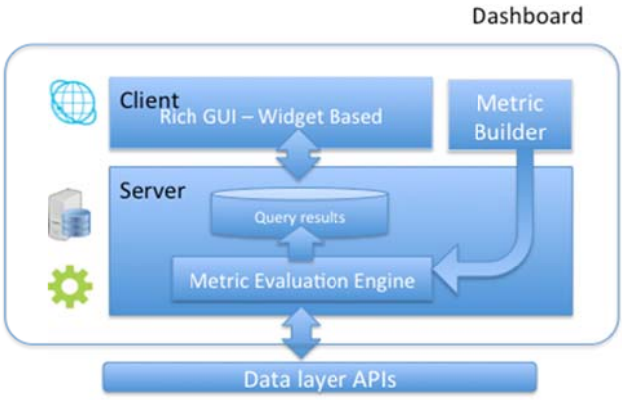
Component	FRAM operalization model
Requirements matching	SoA_01, SoA_4, SoA_05, ERMG_04, ERMG_03, ERMG_10, ERMG_19,ERMG_20, ERMG_22, ERMG_23, ERMG_24, ERMG_25, ERMG_26, ERMG_27, ERMG_28
Functionality	It allows users in modelling and representing the UTS using the FRAM notation and to connect data to the model instantation in order to monitor the entire socio-technical system from the

	function perspective. Moreover, using the data, the tool is able to evaluate the funtional variability and the system resonance at a quantitative level. The component is organised in three macro modules: User management, FRAM Builder and Smart DS. In particular the Smart DS and the User management are branch of exiting applications while FRAM Builder is a new software tool capable to manage FRAM modelling (function and interdependency creations, function grouping, etc.) and FRAM instantiation (connection to data) to quantify/verify functional variability and resonance quantification. In order to connect data to FRAM instantiation, the Smart DS has been exploited, an Analytic Hierarchy Process-based tool adapted to be fuelled with data retrived by different data sources (DB, Triple stores, ESB, etc.). Thus each FRAM Instantiation has its own representation into the Smart DS that, connecting to the data, is able to calculate the FRAM parameters. In FRAM Builder a model can be created, modified, cloned, removed, imported from the FRAM Visualiser tool (a 3rd party software) and exported in XML. Each model can be instantiated many times according to the scenario under investigation (e.g. different inputs can trigger different function and generate functional resonance in multiple ways). The existing Smart DS tool needs to be adapted to read data from ESB and to be connected to the FRAM builder. In this way FRAM builder can "export" a FRAM model into Smart DS automatically.
Component architecture	This component is a WEB App based on a REST Client-Server architecture  SMART DS Adaptation - ESB Interfaces - FRAM Builder connection FRAM Builder - Instantiation Operation & Process: Manage Instantation (delete, creation, clone, update) - Models Operation & Process: Manage model (delete, creation, clone, update) - FRAM Analyser : Implements analysis models to be applided to the FRAM graph.

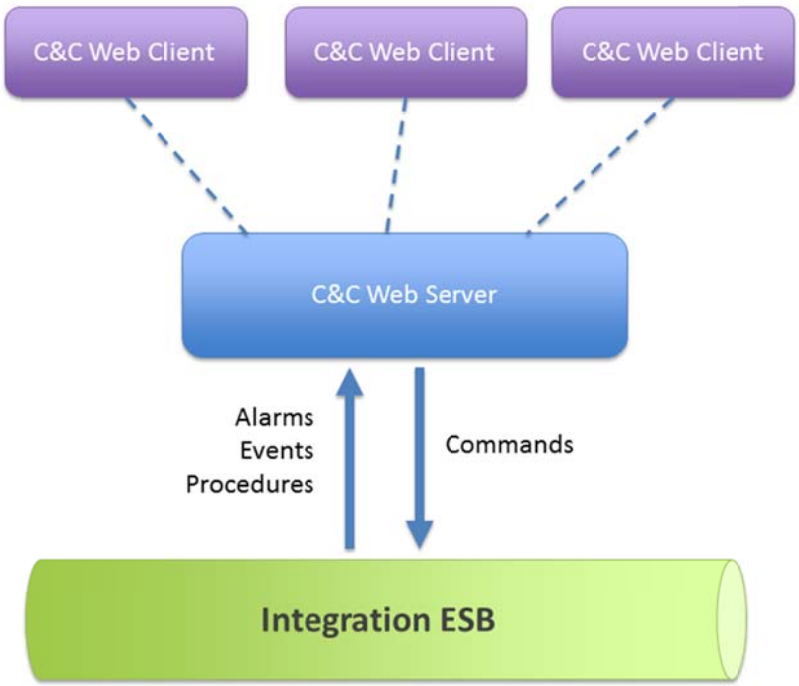
	Examples are: critical functions identification, resonance and functional variailbity analysis, quantification and propagation, etc. - Web Interfaces: to design and manage FRAM models and instantations
Input	• Data from other DSS through ESB as decisions, status, etc.
Output	• Graphical representation of the system • System Analysis (critical functions, resonance, variability, capacity assessment, resource availability)

4.4 Presentation layer

4.4.1 GUI: Common views and specific views

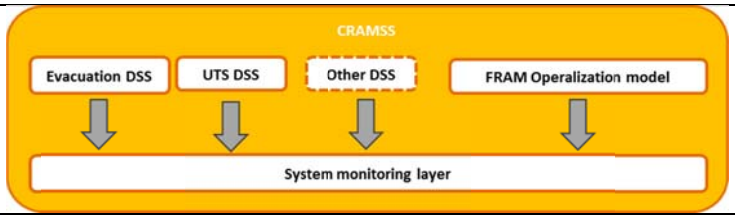
Component	GUI: Common views and specific views
Requirements matching	SoA_01; ERMG_28; ERMG_29; ERMG_30
Functionality	The RESOLUTE dashboard aims at showing the relevant parameters of the system to the operators in order to support monitoring and decisions. The dashboard is shaped with The Dashboard builder, a UNIFI existing application that allows the visualization of the desired metrics considered needed to be monitored. Such metrics are specific queries on the Data layer. Once the metrics are defined they can be selected from a list by the operator that assigns to them the preferred visualization style. It is also possible to constrain the possibility of visualising the metrics to specific user profiles. In this way it is possibile to manage different access grant to information.
Component architecture	The RESOLUTE dashboard is a web application based on the Dashboard builder, a widget based solution that is fed with data obtained through available SPARQL endpoints. The Metric Evaluation Engine implements a process that periodically queries the Data Layer to collect the defined values into the Query results database. Such process is defined using the Metric Builder, a dedicated component able to support users in defining its own metrics using SPARQL.  A reference of the Dashboard builider can be seen at the following link: http://dashboard.km4city.org/
Input	• Inputs are defind on the base of the operator requirements and then will be translated into SPARQL
Output	• Multiple Graphical representation of the information

4.4.2 C&C HMI

Component	C&C HMI
Requirements matching	SoA_01, SoA_02. SoA_03. SoA_04, USR_34, ERMG_08, ERMG_28, ERMG_29, ERMG_30
Functionality	Through the C&C HMI the operator will be able to monitor all the events and alarms passing through the Integration ESB, as well the procedures generated by the FRAM to be executed (manually by an operator or automatically) through the specific DSS.
Component architecture	
Input	• Events • Alarms • Procedures
Output	• Commands

4.5 System monitoring layer

Component	System monitoring layer
Requirements matching	SoA_01, SoA_02, SoA_03, SoA_04, ERMG_08, ERMG_09, ERMG_11, ERMG_15, ERMG_18, ERMG_19, ERMG_20, ERMG_21, ERMG_22, ERMG_23, ERMG_24, ERMG_25, ERMG_27
Function	The goal of the system monitoring layer is to collect and monitor information related to the operational status of each module in the CRAMSS application. Using this information the analyst can decide if the system is in good operational status, or if there are certain actions that must be taken in order to ensure efficient and accurate operation. In order to perform the logging and monitoring, the system uses a list of operational specifications for each module of the CRAMSS application. These specifications are used in

	order to identify the operational status of each Module. Example specifications include, but are not limited to: • Frequency of the sent Guidelines • Format of the Guidelines (e.g. JSON, XML etc.) • Content of the Guidelines • etc. The actual specifications for each module will be specified during the second year of the RESOLUTE project. Three types of operational modes are considered for each module in the context of RESOLUTE, based on their specifications: • Normal operation: This mode specifies that all the modules work normally and as expected. • Warning: This mode specifies that there are some errors in the module, but the module is still able to operate and send useful guidelines. • Error: This mode specifies that the module is not able to operate for various reasons, e.g. input data are missing or the system crushed.
Component architecture	
Input	• Normal operational signals from all components • Warning signals from all components • Error signals from all components
Output	• Log files

5 PHYSICAL REFERENCE ARCHITECTURE

5.1 Software architecture

The RESOLUTE software architecture is a fast, scalable and reliable solution that includes several software modules; each module is delegated to perform a well-specified functionality, as described in the section above. Each and every software module implemented to carry out RESOLUTE software architecture is interfaced with other RESOLUTE software modules by means of interface protocols.

The adopted approach permits to:

- control independently (start, stop, restart) any RESOLUTE software module, without affecting the working conditions of other RESOLUTE software modules,
- maintain and evolve independently (upgrade, replace) any RESOLUTE software module, without involving other RESOLUTE software modules,
- prevent any malfunction on any external system (e.g. UTC System) to impact on RESOLUTE functionalities.

Details about RESOLUTE building and modularisation are available in chapter 4, where all RESOLUTE components are described together with their internal architecture and external interdependencies. In general terms, the whole solution allows for scaling and distributing the different core functionalities among different servers in order to comply with performance and system dimension requirements.

RESOLUTE software provides a protected login where the access to the system is subordinated to the verification of the operator access rights. The authentication to the system is based on three parameters: username, password and roles (access level, which specifies the authorisations needed to make some actions inside the system. The different rights correspond to different roles).

The system provides:

- Local Authentication: the users and the passwords (protected) are stored in the system database.
- Authentication Server: allowing to use external authentication servers (active directories – LDAP) for users' authentication.

5.2 Communication model

5.2.1 Interfaces for Data Layer

The RESOLUTE System interface implements a RESTful architecture which relies on the HTTP protocol.

5.2.1.1 RESTful Architecture

General concept and constraints of a RESTful architecture are reported here for completeness.

Client–server

A uniform interface separates clients from servers. This separation of concerns means that, for example, clients are not concerned with data storage, which remains internal to each server, so that the portability of client code is improved. Servers are not concerned with the user interface or user state, so that they can be simpler and more scalable. Servers and clients may also be replaced and developed independently, as long as the interface between them is not altered.

Stateless

The client–server communication is further constrained by no client context being stored on the server between requests. Each request from any client contains all the information necessary to service the request, and session state is held in the client. Important to note is that the session state can be transferred by the server to another service such as a database to maintain a persistent state for a period and allow authentication. The client begins sending requests when it is ready to make the transition to a new state. While one or more requests are outstanding, the client is considered to be in transition. The representation of each application state contains links that may be used the next time the client chooses to initiate a new state-transition.

Cacheable

As on the World Wide Web, clients can cache responses. Responses must therefore, implicitly or explicitly, define themselves as cacheable, or not, to prevent clients from reusing stale or inappropriate data in response to further requests. Well-managed caching partially or completely eliminates some client–server interactions, further improving scalability and performance.

Layered system

A client cannot ordinarily tell whether it is connected directly to the end server, or to an intermediary along the way. Intermediary servers may improve system scalability by enabling load balancing and by providing shared caches. They may also enforce security policies.

Code on demand (optional)

Servers can temporarily extend or customize the functionality of a client by the transfer of executable code. Examples of this may include compiled components, such as Java applets and client-side scripts, such as JavaScript. "Code on demand" is the only optional constraint of the REST architecture.

Uniform interface

The uniform interface constraint is fundamental to the design of any REST service. The uniform interface simplifies and decouples the architecture, which enables each part to evolve independently. The four guiding principles of this interface are:

- Identification of resources: Individual resources are identified in requests, for example using URIs in web-based REST systems. The resources themselves are conceptually separate from the representations that are returned to the client. For example, the server may send data from its database as HTML, XML or JSON, none of which are the server's internal representation, and it is the same one regardless the resource.

- Manipulation of resources through these representations: When a client holds a representation of a resource, including any metadata attached, it has enough information to modify or delete the resource.

- Self-descriptive messages: Each message includes enough information to describe how to process the message. For example, which parser to invoke may be specified by an Internet media type (previously known as a MIME type). Responses also explicitly indicate their cacheability.

- Hypermedia as the engine of application state: Clients make state transitions only through actions that are dynamically identified within hypermedia by the server (e.g., by hyperlinks within hypertext). Except of simple fixed entry points to the application, a client does not assume that any particular action is available for any particular resources beyond those described in representations previously received from the server.

5.2.1.2 Communication Protocol

The communication protocol used in a RESTful architecture is the Hypertext Transfer Protocol (HTTP)

The standard HTTP methods used are:

- GET: to retrieve a resource or a list of resources
- POST: to add/create a new resource
- PUT: to update/replace an existing resource
- DELETE: to delete an existing resource

Some methods require parameters in order to function properly; each call to an HTTP method shall return an HTTP return code.

All interfaces consist of HTTP-methods which can be called up on the specified resources via network. Each method requires a server responsible for execution which is identified by a Uniform Identifying Name (UIN). Objects within the RESOLUTE System shall be identified by its GUID (Globally Unique IDentifier): the GUID identifies univocally an object in all the sub-systems.

5.2.1.3 System Behavior / Interaction

This section gives a short description of the "typical" interactions between two or more components that are collaborating within the RESOLUTE System.

1. Get data from other components – Polling Approach

   Every component within the RESOLUTE System shall be able to expose an endpoint at which the resource can be retrieved by using the "polling approach".

   If the resource is frequently updated or there is the need of having the updated resource straightaway, then the component shall implement a "push approach" (see subscription below).

2. Get data via Subscription – Push Approach

   Every component in the system that wants to receive notifications/data from another object of the system shall be able to subscribe to it.
   Every component with "real time capabilities" shall be able to manage subscription from one or more subscribers for at least some basic functionalities (i.e. a controller shall at least be able to publish signal group data to its subscribers). Every component shall also expose its "publication capabilities" in order to let the other components know how the subscription should be expected.

3. Get big data – Batch Approach

   In some cases it has to be taken into consideration that some requests could be quite long (i.e. retrieving traffic data offline for a certain time interval). For this reason, the interface should expose an endpoint for those operations; the endpoint URL shall have the same name of the "classic" endpoint with the addition of a specific prefix / suffix or parameter.

The Data Model for the Data Management Layer is not within the scope of this document, being fully detailed in deliverable D4.2 Multi source data acquisition.

5.2.2 Interfaces for Mission Critic Layer

All units that constitute the core of the Mission Critic Layer are linked together by an Integration Enterprise Service Bus (i.e. aim of T4.5), characterized by a dedicated communication protocol along with the corresponding data model for the exchange of information regarding resilience strategies. The full specification of the interfaces and data model for this component does not fit the scope of this document and can be therefore found in D4.5 Integration Framework Implementation.

5.3 Resilience of the Hardware architecture

The RESOLUTE system designed in this document needs to respond to various user needs in both normal and emergency situations, and therefore it needs to relay on a HW architecture that is characterized by:

- Usability: the system needs to be available to its users according to the functional specification in Chapter 4 in both normal and emergency situations
- Security: security of the HW architecture is applicable to both servers/ hard drives and removable/ portable devices and the following considerations should be taken into account:
    - Local servers and intranet: in case of a shared network that is contained in one building (LAN) or across several locations (WAN) the IT department should identify a person responsible for maintenance and security of this network. The IT person/department will also write, own and manage policy for the server and intranet permission.
    - Third party servers and services including Cloud storage: areas that need to be addressed include:
        - website hosting
        - location of data servers
        - security policy of the 3rd party providers
    - Protection from hardware security breach: similar to individual computers/stations, security programs should be used to protect all hardware systems (i.e. not allowing auto-run, security programs including intranet firewalls)
    - Intranet Firewalls: can be implemented in both hardware and software, or a combination of both. Firewalls need to be used to prevent unauthorized Internet users from accessing private networks connected to the Internet, especially *intranets*.

- Reliability: the RESOLUTE system should be available in a continuous manner 24hrs/ day. Some specific parameters need to be considered in the operation phase:
    - Mean Time Between Failures
    - Mean Time to Repair
    - Maximum Defect Rate

- Performance: the system response time should be near to real-time and, most important the RESOLUTE system should be able to operate in a degraded mode: in case one of the components results out of function this should not impact the overall system performance.
- Adaptability: The HW architecture of RESOLUTE system must be able to scale and de-scale on-demand in order to answer specific crisis situations.

- Availability: emergencies happen, environments go down, but the business process must go on and therefore the RESOLUTE system needs to be characterized by both aspects of availability:
    - Business Continuity
    - Disaster Recovery

First things first: it's important to understand that disaster recovery and business continuity are two different business objectives. However, they can certainly overlap. Cloud technology could be one way to answer this requirement. There are specific cloud technologies that have become driving factors for better business IT redundancy, and the following considerations need to be taken into account:

- Cloud-based traffic management. This is all about global traffic management (GTM) and global server load balancing (GSLB). The ability to logically control traffic over vast distances has allowed the cloud to become a true hub for many different kinds of DR strategies.
- Software-defined approach. A lot of physical resources can be abstracted directly into the cloud layer.
- Virtualization. Working with agile technologies like virtualization allows the sharing, replication, and backup of various kinds of images and workloads.
- New types of storage and hardware intelligence. Cloud-based storage solutions have come a long way. Data deduplication, controller multi-tenancy, and fast site-to-site replication make cloud storage systems a powerful part of the DR process.

6 RISK ASSESSMENT

6.1 Approach

This section presents the proposed contingency plans and actions to countermeasure the potential risks during the final specification and implementation of the RESOLUTE components.

The applied methodology consists in conducting an *a priori* risk assessment for the various modules. This has been done using an 'Expanded Failure Modes and Effects Analysis' (EFMEA) approach, drawing upon previous FMEA risk assessment exercises from the ASK-IT, OASIS and SAVE-ME projects, but expanding this to account for some of the perceived limitations of 'traditional' FMEA approaches. The analysis is done based on the specification of all modules presented in Chapter 4.

The EFMEA classifies risks based upon their nature:

- Technical (physical features of hardware; coding elements of software)
- Legal (based upon existing policies and laws in each nation)
- Behavioural (from users' behaviour)
- Organisational (in relation to disaster mitigation plans and actors' roles)

All these typologies of risks have been addressed for every component in the RESOLUTE system. In addition to the risk analysis, the EFMEA helps to identify critical elements of the overall system architecture, evaluate suitable actions and mitigation strategies, with the overarching goal of contributing to the contingency plans of the project.

6.2 Expanded Failure Modes and Effects Analysis (EFMEA)

This section specifies the actions and analysis to be implemented in the EFMEA of the RESOLUTE, in order to ensure that the project system architecture conforms to the highest standards and any critical areas of potential failure can be identified and solutions developed.

In order to overcome and even avoid malfunctions or errors that are usually observed when developing a new system, the possible errors must be identified at an early stage of the system design. The RESOLUTE system may malfunction due to three reasons: false system readings/outputs, inappropriate use by the users and having a single point of failure that through its malfunctioning could cause the collapse of the whole system. For the first case, the RESOLUTE system developers will need to put more effort into making the system reliable and robust, while for the second, users' personal limits should be the main concern and this will be done through the implementation of User profiling module; also an easy-to-use product must be developed. As regards to the third issue, the design with the physical architecture will overcome this issue.

6.2.1 Failure Modes and Effects Analysis (FMEA)

Failure Mode and Effects Analysis (FMEA) is an analysis technique which facilitates the identification of potential problems in the design or process by examining the effects of lower level failures. Recommended actions or compensating provisions are made to reduce the likelihood of the problem occurring, and mitigate the risk, if in fact, it does occur. The FMEA determines, by failure mode analysis, the effect of each failure and identifies single failure points that are critical. It may also rank each failure according to the criticality of a failure effect and its probability of occurring (FMEA-FMECA.com).

A successful FMEA activity helps to identify potential failure modes based on past experience with similar products or processes, enabling those failures to be designed out of the system with the minimum of effort and resource expenditure, thereby reducing development time and costs.

Failure modes are the ways, or modes, in which something might fail. Failures are any errors or defects, especially ones that affect the user, and can be potential or actual.

Effects analysis refers to studying the consequences of those failures and potential mitigation strategies.

Failures are prioritized according to how serious their consequences are, how frequently they occur and how easily they can be detected through the calculation of a Risk Priority Number (RPN) for each failure mode identified in the system. The purpose of the FMEA is to take actions to eliminate or reduce failures, starting with the highest-priority ones.

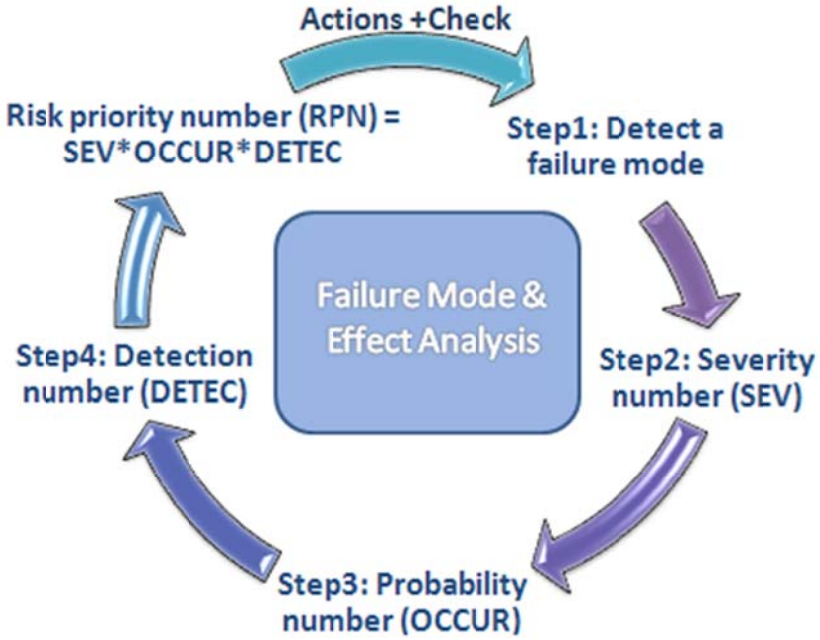


Figure 4: FMEA Process Cycle (by Dieter Vandeun; Source: Wikipedia)

6.2.2 Expanded Failure Modes and Effects Analysis (EFMEA)

Whilst FMEA is a widely accepted methodology, and has been applied to many different projects, it has been criticised for having a number of limitations throughout the various calculation steps – it often misses key failures; if performed too late it cannot affect key product/process decisions; is a tedious process and the Risk Priority Number is not a good measure of risk (Kmenta, 2002). Therefore, it is proposed to use Expanded FMEA (EFMEA) as proposed by Bluvband, Grabov and Nakar (2004), which is designed to develop traditional FMEA processes to account for some of the limitations.

The methodology to be used in RESOLUTE will follow the well-established processes from ASK-IT, OASIS and SAVE-ME EC-funded projects, adding to this where required, using the principles from EFMEA.

6.2.3 Calculation of Risk Priority Numbers

The results of the RESOLUTE risk analysis must be comparable and must be presented in an understandable and comprehensive format – Risk Priority Numbers (RPNs). Such an analysis involves various factors of each

safety-security issue: severity, occurrence probability, detectability and recoverability, not only for technical risks, but also for behavioural, legal and organizational related risks.

Behavioural risks are related to the users' behaviour, regarding their interaction with the system, concentrating on the possible wrong moves or reactions they might perform. Legal risks include the risks that will arise if the system is not compliant with the legislation of the country. Finally, by the term organizational related risks, the risks involved within the organization structure of the service chain are implied.

The overall process is depicted in the figure below:

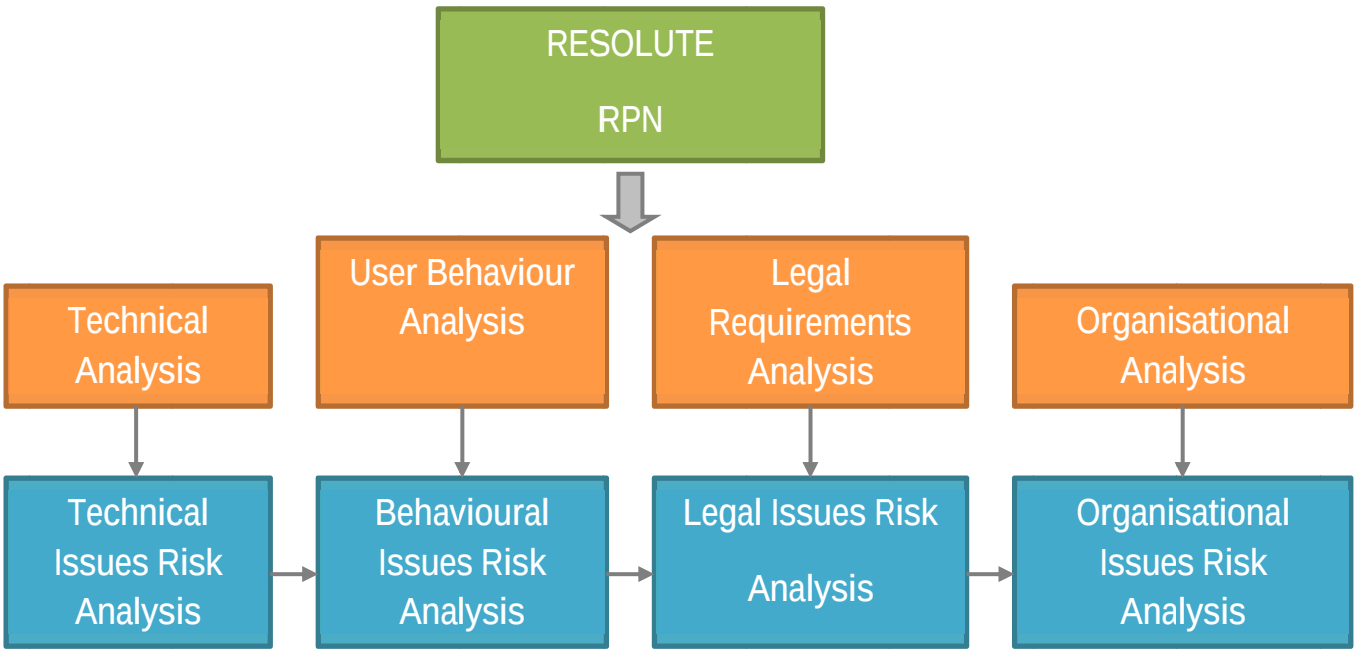


Figure 5 Methodology for the estimation of Risk Priority Numbers

The Risk Priority Number (for each risk) is calculated by the following equation:

$$RPN = S \times O \times \frac{D + R}{2}$$

Where: *S* = Severity; *O* = Occurrence; *D* = Detectabilty; *R* = Recoverability

Whilst many (E)FMEA are carried out by a team of experts, it is important to understand that in RESOLUTE there will be different people from different partners undertaking the calculations and so ways of achieving consistent results from each person/partner are required. The following checklist of 10 key points based upon the question "What can go wrong?" has been developed by Bluvband and Grabov (2009) to assist individuals in identifying possible Failure Modes:

1. The intended function is not performed
2. The intended function is performed, but there are some safety problems, or a problem in meeting a regulation associated with the intended function performance
3. The intended function is performed, but at a wrong time (availability problems)

4. The intended function is performed, but in the wrong place (position in the system)
5. The intended function is performed, but in the wrong way (efficiency problems)
6. The intended function is performed, but the performance level is lower than expected
7. The intended function is performed, but its cost is higher than planned (additional maintenance, repair, power consumption etc.)
8. An unintended/unplanned and/or undesirable function is performed
9. The period of unintended function performance (lifetime) is lower than planned (reliability issues)
10. Support for the intended function performance is impossible or problematic (maintenance, repair, service issues etc.)

From this, the following tables have been developed in project SAVE-ME in order to assist with identifying the degree of each issue and what value should be assigned in the RPN calculation.

Table 4: Severity (S) level analysis

Level of severity	Technical issue	Behavioural issue	Legal issues	Organisational issues
9-10 (extremely severe)	The failure could put user safety at risk, potentially causing injury or fatality	The user error in operating the system could lead to an incident worseness (i.e. safety effects)	Are there laws in each country that do not allow the system to be implemented?	Wide and different organizational framework is needed, that is completely missing (i.e. new services)
7-8 (severe)	The failure implies the total loss of the system functions, resulting in user's dissatisfaction	User behavioural error may abort the system benefits (i.e. safety effects due to changes in ways of acquiring info)	New laws are required for system implementation and no relevant work has been performed yet	Organisational framework adaptation is needed (some initial actions have been taken on this domain)
5-6 (slightly severe)	The failure implies the partial loss of the system function, resulting in user's dissatisfaction	User's behavioural changes may significantly reduce the positive effects of the system	New laws are required for system implementation and work required has already been performed	Organizational framework adaptation is needed which has already started being realised
3-4 (significant)	The failure implies slight dissatisfaction to the user	User's behavioural changes may somehow influence the positive effects of the system	New laws are required for system implementation but consensus on them exist	There is a need for limited and easily realized organizational changes
1-2 (insignificant)	The failure does not imply perceptible effects to the system function and to the	User's behaviour is not expected to reduce the system benefits significantly, or may	No new laws are required for implementation	There is no need at all for organizational changes

Level of severity	Technical issue	Behavioural issue	Legal issues	Organisational issues
	user's satisfaction	even further enhance them		

Table 5: Occurrence (O) level analysis

Occurrence level	Technical issue	Behavioural issue	Legal issues	Organisational issue
9-10 (high)	It is certain that some failures will sometimes occur	It is certain that some behavioural effects will occur (by the system users)	It is certain that some legal problems will occur	It is certain that there will be a need for organizational restructuring
6-8 (medium)	A failure could occasionally occur	Some behavioural effects could occasionally occur	Some legal problems could occasionally occur	A need for organizational restructuring could occasionally occur (depending on the needs of the service, that will arise after the operation of the system)
3-5 (low)	There is only a slight probability that en error/failure will occur	There is only a slight probability that some behavioural effects will occur	There is only a slight probability that some legal problems will occur	There is only a slight probability that a need for organizational restructuring will occur
1-2 (improbable)	It is unlikely that a fault will occur	It is unlikely that some behavioural effects will occur	It is unlikely that some legal problems will occur	It is unlikely that a need for organizational restructuring will occur

Table 6: Detectability (D) level analysis

Detectability level	Technical issue	Behavioural issue	Legal issue	Organisational issue
9-10	It is impossible or improbable that a	It is impossible or improbable that a	It is impossible or improbable that a	It is impossible or improbable that an

Detectability level	Technical issue	Behavioural issue	Legal issue	Organisational issue
(improbable)	problematic area will be detected	user's behavioural effect will be detected	legal problem will be detected	organizational problem will be detected
7-8 (slight)	The problematic area is detected only in particular cases	The user's behavioural effect is detected only in particular cases	The legal problem is detected only in particular cases	The organizational problem is detected only in particular cases
5-6 (moderate)	It is probable that the problem will be detected (depending on the situation)	It is probable that the user's behavioural effect will be detected	It is probable that the legal problem will be detected	It is probable that the organizational problem will be detected
3-4 (high)	It is very probable that a problem will be detected	It is very probable that the user's behavioural effect will be detected	It is very probable that the legal problem will be detected	It is very probable that the organizational problem will be detected
1-2 (very high)	It is certain that a problem will be detected	It is certain that the user's behavioural effect will be detected	It is certain that the legal problem will be detected	It is certain that the organizational problem will be detected

Table 7: Recoverability (R) level analysis

Recoverability level	Technical issue	Behavioural issue	Legal issues	Organisational issues
9-10 (null)	No recovery action is provided	System is (in)flexible to user's behavioural effects	System is either accepted or rejected by the legal framework	System requires a fixed organizational environment to operate
6-8 (low)	The user is only advised on the failure	Behavioural effects are taken into account by the system	System may be slightly adapted to meet legal restrictions	System requires a fixed organizational framework with limited adaptations
3-5 (high)	Effective recovery action is provided	System customization might compensate for user's behavioural effects	System encompasses different versions to meet particular legal demands	System may operate within various organizational frameworks

Recoverability level	Technical issue	Behavioural issue	Legal issues	Organisational issues
1-2 (full recoverability)	The failure effect is completely avoided by the recovery action	System does not allow user's behavioural effects	System is easily reconfigurable to meet legal demands	System does not require organizational changes

Using the values in the above tables, the appropriate RPN must be calculated for each identified risk item in the RESOLUTE system, using one of the individual Technical RPN (TRPN), Behavioural RPN (BRPN), Legal RPN (LRPN) and Organisational RPN (ORPN) value.

6.2.4 Identification of Total Risk Estimate and Critical Items

The calculation of the RPN for each item can highlight potentially problematic areas in which the developers are required to put more effort in resolving (i.e. to offer mitigation strategies). The value of each individual RPN calculated above is initially matched to five levels of severity, as defined in the following table (values are indicative only):

Table 8: Correlation of Overall risk factor with overall risk severity level.

Calculated RPN	Overall severity
512-1000	I- Extremely severe
216-512	II- Severe
64-216	III – Moderate
8-64	IV – Slight
1-8	V – Insignificant

It is also useful to calculate the Total Risk Estimate (TRE) for the overall project, as proposed by Bluvband and Grabov (2009):

$$TRE = \frac{\sum_{i=1}^{n} RPN_i}{1000n} \times 100\%$$

Where: RPN_i are the individual RPN values for each item, and n is the total number of items in the EFMEA analysis.

TRE values range between 0.1% (no risk at all) and 100% (extremely risky), but it unlikely that either of these extreme values will be obtained. Bluvband and Grabov suggest that any TRE>17% indicates a 'risky' project as this is where the individual T/B/L/ORPN values are 5.5 i.e. the middle of the 1 to 10 scale used in the tables, or higher.

The next step is to attempt assessing the risk of each of those issues.

6.2.5 Corrective Actions

Once the risk items have been identified, the next step is to attempt to identify possible corrective actions or mitigating strategies. The possible success of these actions/strategies should also be identified and, where possible, quantified. There may be several possible options for each issue, and any risk reduction is an iterative process involving dependencies between the different issues.

In terms of corrective actions, risk can be reduced in a number of generic ways:

- reducing the magnitude (severity) of the consequences of the potential risk;
- reducing the probability of the risk occurring;
- increasing failure detection speed and probability;
- protecting against the risk, mitigating strategies to compensate for a failure;
- transferring the risk to another Party.

Traditional FMEA does not issue adequate guidance for selecting the optimal choice of corrective actions, as actions required to lower existing RPN values may not be appropriate, achievable or feasible under project constraints (time, resource, budget etc.) Therefore, Bluvband and Grabov (2009) propose a comparison evaluation of each pre- and post-correction RPN, also taking into account the 'feasibility' of each action.

The 'feasibility' of each action is ranked on a scale from 1 (Best Case) to 10 (Worst Case), using the following guidelines (Bluvband, Grabov and Nakar, 2004):

Table 9: Feasibility of Corrective Actions

Feasibility of Corrective Action Implementation	Ranking
Safety problem and/or non-compliance to Government regulations; Unavailable necessary resources; Unacceptable cost/time/resource consumption; Zero chance of success; 100% probability of undesirable impact	10
Very remote availability of necessary resources; Almost unacceptable cost/time/resource consumption; Very low chance of success; ~90% probability of undesirable impact	9
Remote availability of necessary resources; Near unacceptable cost/time/resource consumption; Remote chance of success; ~80% probability of undesirable impact	8
Very low availability of necessary resources; Very high cost/time/resource consumption; Very low chance of success; ~70% probability of undesirable impact	7
Low availability of necessary resources; High cost/time/resource consumption; Low chance of success; ~60% probability of undesirable impact	6
Rather low availability of necessary resources; Relatively high cost/time/resource consumption;	5

Rather low chance of success; ~50% probability of undesirable impact	
Moderate availability of necessary resources; Medium cost/time/resource consumption; Moderate chance of success; ~40% probability of undesirable impact	4
Some availability of necessary resources; Rather low cost/time/resource consumption; Some chance of success; ~30% probability of undesirable impact	3
Good availability of necessary resources; Low cost/time/resource consumption; Good chance of success; ~20% probability of undesirable impact	2
Full availability of necessary resources; Very low cost/time/resource consumption; High chance of success; 0-10% probability of undesirable impact	1

Once the pre- and post-correction RPN, and the feasibility rank have been determined for each item, the following equation is used to identify the most suitable action(s) to apply:

$$\frac{RPN_{iBefore} - RPN_{iAfter}}{F_i} = \frac{\Delta RPN_i}{F_i}$$

Where: $RPN_{iBefore}$ = pre-correction RPN value; RPN_{iAfter} = pre-correction RPN value; F_i = Feasibility Rank (from the table). The corrective action with the largest value of $[RPN_i/F_i]$ is the one that can be said to be most suitable or preferable to implement.

6.2.6 Evaluation of Corrective Actions

Once the initial RPNs have been calculated, and the optimal post-correction RPNs determined for each item, it is useful to return to the overall project and evaluate the effectiveness of the interactions using a normalised improvement estimate (Bluvband and Grabov, 2009):

$$\Delta RPN = \frac{\sum RPN_{iBefore} - \sum RPN_{iAfter}}{\sum RPN_{iBefore}} \times 100\%$$

It has been suggested that a risk reduction of up to 30% can be achieved through the completion of a full EFMEA, but this naturally depends on the initial TRE value.

6.3 A Priori Expanded Failure Modes and Effects Analysis for RESOLUTE

The *a priori* EFMEA was conducted in two consecutive stages.

Stage one identified the various risks associated with the RESOLUTE system, assigning RPN numbers to each risk. Stage two then addressed the risks, including the detailed mitigation strategies and re-calculation for RPN values for all identified risks.

6.3.1 Stage One: Risk Identification

In total, 39 risks were identified for the RESOLUTE system, clustered by layers/sub-layers of the system itself, as shown in the following table:

Table 10: Initial Risk Items and RPN Calculations

Risk Description	S	O	D	R	RPN	Risk Level
DATA ACQUISITION						
Some Data source unavailable (temporarly)	6	6	1	7	144	III - Moderate
Data quality degraded	4	5	3	3	60	IV – Slight
Data availability rate from data sources higher respect to data collection rate	4	3	4	3	42	IV – Slight
Data publication APIs or data model changed	6	5	1	6	105	III – Moderate
Data collection process failure	8	4	1	2	48	IV – Slight
DATA AGGREGATION						
Volume of data managed affects the system performance	6	6	2	4	108	III – Moderate
Data model/ontology not optimised for the given queries	6	5	5	3	120	III – Moderate
Data management HW/SW failure	8	3	1	3	36	IV – Slight
Volume of data requested through the APIs affects the system performance	6	5	4	4	120	III – Moderate
INTEGRATION FRAMEWORK						
Malfunctioning in Integration ESB	5	2	1	3	27	IV – slight
Not enough troughput in Integration ESB	3	2	3	3	27	IV – slight
APPLICATION FRAMEWORK						
Incorrect profiling for an UTS user	1	6	1	8	27	IV – slight

Risk Description	S	O	D	R	RPN	Risk Level
Incorrect profiling for a CRAMSS user	6	5	6	8	210	III – moderate
Malfunctioning in Network analysis (physical layer)	5	2	1	8	45	IV – slight
Malfunctioning Network analysis (service layer)	5	4	1	8	90	III – moderate
Malfunctioning in Network analysis (cognitive level)	5	2	1	8	45	IV – slight
Malfuntioning in Network analysis (integrated layers)	5	4	1	8	90	III – moderate
Malfunctioning in Network analysis (cascading effects)	5	4	1	8	90	III – moderate
CRAMSS						
Lack of documentation for candidate technologies	6	4	6	5	132	III – Moderate
Performed requirements analysis is ineffective, result-ing in a situation where pro-ject drifts into wrong direc-tion and subsequently ef-forts and performed actions yield poor results as op-posed on the proposed R&D	7	4	5	6	154	III – Moderate
The use case definition could result in use cases and sce-narios which cannot be fully technically implemented.	7	5	4	6	175	III – Moderate
Research on the technolo-gies addressed within the project is limited.	6	4	6	5	132	III – Moderate
Integration of the CRAMSS multisensorial platform with different sensors and devic-es might lead to inconsisten-cies	6	5	2	3	75	III – Moderate
Technological or operational constraints may affect the deployment and operation of the planned project pilots, limiting the amount and/or the quality of pro-duced results.	5	6	4	4	120	III – Moderate
Not properly defined KPI's and thus not optimal evalua-tion of project results	6	5	7	3	150	III – Moderate
Possible damaged sensor or meter device - false data acquisition	5	5	3	2	62.5	IV – Slight
Performance issues (e.g. too slow)	6	6	5	5	180	III – Moderate

Risk Description	S	O	D	R	RPN	Risk Level
RESILIENCE DASHBOARD						
Metrics selected are not enough informative to support situation awareness	7	5	5	4	157,5	III – Moderate
The human computer interaction designed increase the cognitive workload and errors under stressful condition	7	4	4	4	112	III – Moderate
Interfaces widget failure	6	4	2	4	72	III – Moderate
The dashboard is not accessibile	6	3	3	3	54	IV – Slight
Browser compatibility issues	7	4	2	4	84	III – Moderate
ACTUATION CHANNELS						
Interoperability problems between components that have been built on heterogeneous frameworks or programming tools.	6	5	2	3	75	III – Moderate
Performed requirements analysis is ineffective, resulting in a situation where project drifts into wrong direction and subsequently efforts and performed actions yield poor results as opposed on the proposed R&D	7	4	5	6	154	III – Moderate
Over fitting of System Models and Analysis Services to the RESOLUTE Pilot Sites & Business Cases	6	3	5	6	99	III – Moderate
Technological or operational constraints may affect the deployment and operation of the two planned project pilots, limiting the amount and/or the quality of produced results.	5	6	4	4	120	III – Moderate
Not properly defined KPI's and thus not optimal evaluation of project results	6	5	7	3	150	III – Moderate
Possible damaged sensor or meter device - false data acquisition	5	5	3	2	62.5	IV – Slight
Communication failure among RESOLUTE sub-systems	8	6		6	288	II- Severe
Interference of wireless networks	8	5	3	5	160	III – Moderate
Performance issues (e.g. too slow)	6	7	5	5	210	III – Moderate

Risk Description	S	O	D	R	RPN	Risk Level
Loss of Saved Data	8	6	3	2	120	III – Moderate
Lack of documentation for candidate technologies	6	4	6	5	132	III – Moderate

Calculation of the TRE value for this first stage, using the formula given in section 6.2.4, is as follows:

100% x [(RPN)] / [1000*n(RPN)] = 10.82

6.3.2 Stage Two: Risk Mitigation

Once all risks had been ranked, the next stage was to establish mitigation strategies for each item. Furthermore, a single mitigation strategy and the possibility of implementing this strategy was required, based upon the following table:

Table 11: Definition of Mitigation Possibility Levels for Assigning to Risk Items

Mitigation Possibility	*Definition*
High	A solution is available at relatively little cost
Medium	An achievable solution may be possible at reasonable cost, or a reasonable solution is available at modest cost
Low	An expensive solution may be possible, but system benefits may not justify these, and/or a solution needs further investigation or is highly complicated
Improbable	Solutions are too expensive (likely to remain so) in relation to the reduction of risk(s) and the benefits gained from the functionality of the system, and/or a solution is not available for the (extremely) severe risk that has been identified

Furthermore, the RPN values were calculated for all identified risks considering the provided mitigation strategies. Before values are highlighted in the following table in red, while after values are highlighted in green.

Table 12: Mitigation Strategies for Each Risk Item and new RPN

Risk Description	S	O	D	R	RPN	Risk Level	Mitigation Strategy	Possibility	S	O	D	R	RPN	Risk Level
DATA ACQUISITION														
Some Data source unavailable (temporarly)	6	6	1	7	144	III - Moderate	Preliminary identification of different data providers for the same data	Medium	6	6	1	3	72	III - Moderate
Data quality degraded	4	5	3	3	60	IV – Slight	Implementation of Algorithms to identify outlayers in the data streams	Medium	4	5	2	2	40	IV – Slight
Data availability rate from data sources higher respect to data collection rate	4	3	4	4	48	IV – Slight	Improvement of the communication protocols	Low	4	3	3	2	30	IV – Slight
Data publication APIs or data model changed	6	5	1	6	105	III – Moderate	Preliminary agreement between data providers and data aggregators to establish an alerting protocol when data format or API are going to be changed. This alert sent in due tme, allows a prompt adaptation of the data aggregator to the fortcoming conditions.	Medium	6	2	1	3	24	IV – Slight
Data collection process failure	8	4	1	2	48	IV – Slight	Hardware/Software distributed architecture implementation	High	8	2	1	1	16	IV – Slight
DATA AGGREGATION														
Volume of data managed affects the system performance	6	6	2	4	108	III – Moderate	Increment of HW capability applying cloud based solution	Medium	6	3	2	2	36	IV – Slight
Data model/ontology not optimised for the given queries	6	5	5	3	120	III – Moderate	Revision of the queries. Improvement of the Ontology (e.g. creation of semantic connection among entities to speed up the queries)	High	6	4	4	2	72	III – Moderate

Risk Description	S	O	D	R	RPN	Risk Level	Mitigation Strategy	Possibility	S	O	D	R	RPN	Risk Level
Data management HW/SW failure	8	3	1	3	48	IV – Slight	Business continuity and Disaster Recovery strategy application	Medium	8	2	1	2	24	IV – Slight
Volume of data requested through the APIs affects the system performance	6	5	4	4	120	III – Moderate	Load balance strategy, increment of system resources on demand	High	6	2	2	2	24	IV – Slight
INTEGRATION FRAMEWORK														
Malfunctioning in Integration ESB	5	2	1	3	20	IV – slight	Deploy the Integration ESB in “High availability”	Medium	5	1	1	2	7,5	V – Insignificant
Not enough troughput in Integration ESB	3	2	3	3	18	IV – slight	Deploy the Integration ESB in “High availability”	Medium	3	1	2	2	6	V – Insignificant
APPLICATION FRAMEWORK														
Incorrect profiling for an UTS user	1	6	1	8	27	IV – slight	Improving accuracy and reliability of the profiling and behavioural nalysis through more (historical) data	High	1	3	1	8	13,5	IV – Slight
Incorrect profiling for a CRAMSS user	6	5	6	8	210	III – moderate	Improving accuracy and reliability of the profiling and behavioural nalysis through more (historical) data	High	6	3	6	8	126	IV – Slight
Malfunctioning in Network analysis (physical layer)	5	2	2	8	50	IV – slight	Frequent monitoring and control of the service availability and functioning	High	5	2	1	8	45	IV – Slight
Malfunctioning Network analysis (service layer)	5	4	2	8	100	III – moderate	Frequent monitoring and control of the service availability and functioning, Improving reliability of the access to data related to flow and counts of passangers	High	5	3	1	8	67,5	III – Moderate
Malfunctioning in Network analysis	5	2	2	8	50	IV – slight	Frequent monitoring and control of the service	High	5	2	1	8	45	IV – Slight

Risk Description	S	O	D	R	RPN	Risk Level	Mitigation Strategy	Possibility	S	O	D	R	RPN	Risk Level
(cognitive level)							availability and functioning							
Malfuntioning in Network analysis (integrated layers)	5	4	2	8	100	III – moderate	Frequent monitoring and control of the service availability and functioning Improving reliability of the access to data related to flow and counts of passangers	High	5	3	1	8	67,5	III – Moderate
Malfunctioning in Network analysis (cascading effects)	5	4	2	8	100	III – moderate	Frequent monitoring and control of the service availability and functioning Improving reliability of the access to data related to flow and counts of passangers Performing frequent analysis and collect data/information about the possible propagation of effects from/to other interconnected urban critical infrastructures	Medium	5	3	1	8	67,5	III – Moderate
CRAMSS														
Lack of documentation for candidate technologies	6	4	6	5	132	III – Moderate	The documentation provided for candidate technologies is insufficient for appropriate evaluation will be mitigated by contacting manufacturer to get more information.	Medium	6	4	6	3	108	III – Moderate
Performed requirements analysis is ineffective, resulting in a situation where project drifts into wrong direction and subsequently efforts and performed actions yield poor results as opposed on the proposed R&D	7	4	5	6	154	III – Moderate	The Project Coordinator has high expertise in the architectural definition process, being the project man-ager and coordinator in several national and European projects, while the Quality Manager has already designed a similar system (InfrArt-Sonic). In addition, all task leaders have the necessary know-how and the procedures to follow towards fulfilling the objectives and expected results from each task. The involved academic and industry	Medium	7	3	3	5	84	III – Moderate

Risk Description	S	O	D	R	RPN	Risk Level	Mitigation Strategy	Possibility	S	O	D	R	RPN	Risk Level
							partners will provide the necessary combination of knowledge towards conducting a detailed and valuable SoA analysis as well as an extensive Market feasibility study. Finally, the WP participants and Task Leaders will be with close contact through periodic aligment calls with all other R&D WP leaders in order to ensure the minimisation of risks for the whole project lifecycle.							
The use case definition could result in use cases and scenarios which cannot be fully technically implemented.	7	5	4	6	175	III – Moderate	The use cases will be tailored to the technical possibilities of the technologies envisioned by the consortium. However, a feasibility assessment is established in order to evaluate if the scenario implementation requires an R&D advancement that exeed project resources, scope or consortium capability. In this case the scenario will be re-aligned to the project boundaries.	Medium	7	3	3	4	73,5	III – Moderate
Research on the technologies addressed within the project is limited.	6	4	6	5	132	III – Moderate	The WP leaders and the partners participating in the definition and analysis of corresponding technologies have profound experience in pioneering research & development in the domains addressed within each WP. Furthermore, given the fact that there is no consensus available on setting thresholds and attributes related to environmental conditions for user authentication, the foreseen high data collection in WP4 would be one of the high priorities of the first year in order to ensure the desired knowledge and consensus for the algorithms development in this WP.	Medium	6	3	5	4	81	III – Moderate
Integration of the CRAMSS multisensorial platform with different sensors and devic-es might lead to inconsistencies	6	5	2	3	75	III – Moderate	The Platform of CRAMSS will support cross-platform interoperability of the pro-duced solutions for different platforms and devices. Without re-nventing the wheel, redesign of portion of code will be held whenever necessary.	High	6	4	2	2	48	IV – Slight

Risk Description	S	O	D	R	RPN	Risk Level	Mitigation Strategy	Possibility	S	O	D	R	RPN	Risk Level
Technological or operational constraints may affect the deployment and operation of the planned project pilots, limiting the amount and/or the quality of pro-duced results.	5	6	4	4	120	III – Moderate	The two pilot cases will be lead by project partners that have an extensive knowledge of their characteristics. Preliminary reasearch is conducted, in order to identify potential obstacles to the pilot implementation, as well as possible solutions. In addition, pilot leaders own or manage other objects that can be used as alternative.	Medium	5	3	3	3	45	IV – Slight
Not properly defined KPI's and thus not optimal evaluation of project results	6	5	7	3	150	III – Moderate	All partners will deliver a full list of KPI's and performance measurements that will be carefully examined/evaluated and constantly updated throughout the project.	High	6	4	5	3	96	III – Moderate
Possible damaged sensor or meter device - false data acquisition	5	5	3	2	62,5	IV – Slight	The heterogeneous sensors are an essential part of the RESOLUTE project. Regular check of sensors and devices will be performed so that damaged equipment will be replaced as soon as possible.	Medium	5	5	2	1	37,5	IV – Slight
Performance issues (e.g. too slow)	6	6	5	5	180	III – Moderate	During the development of each module but also after the implementation of the Pilot tests, performance issues will be dealt with real time data. The performance will be evaluated and if needed the time and effort will be estimate to improve the system	Medium	6	4	4	4	96	III – Moderate
RESILIENCE DASHBOARD														
Metrics selected are not enough informative to support situation awareness	7	5	5	4	157,5	III – Moderate	Designing and revising metrics with stakholders and experts in dedicated sessions	High	7	3	3	3	63	IV – Slight
The human computer interaction designed increase the cognitive workload and errors under stressful condition	7	4	4	4	112	III – Moderate	The Usability and UX desing will be tested in a lab-settings with end users	Medium	7	4	3	3	84	III – Moderate

Risk Description	S	O	D	R	RPN	Risk Level	Mitigation Strategy	Possibility	S	O	D	R	RPN	Risk Level
Interfaces widget failure	6	4	2	4	72	III – Moderate	Bug reportings and fix cycle	High	6	4	1	2	36	IV – Slight
The dashboard is not accessibile	6	3	3	3	54	IV – Slight	Verification of the interface accessibility through manual and automatic tools (e.g. WCAG2.0)	High	6	3	2	2	36	IV – Slight
Browser compatibility issues	7	4	2	4	84	III – Moderate	In the development phase, a compatibility check with multiple browser is included.	High	7	3	1	2	31,5	IV – Slight
ACTUATION CHANNELS														
Interoperability problems between components that have been built on heterogeneous frameworks or programming tools.	6	5	2	3	75	III – Moderate	The WP leaders will use best practices to prevent interoperability thrust while assessing the design specifications for each component. Interfaces, exchange formats etc. will be defined and tracked in cooperation with all technical WP leaders.	High	6	3	2	3	45	IV – Slight
Over fitting of System Models and Analysis Services to the RESOLUTE Pilot Sites & Business Cases	6	3	5	6	99	III – Moderate	Pilot Sites premises are based in 2 different cities and provide a fully-fledged and representative test case for real- life calibration & evaluation of the system performance on alternative business cases. The size of the premises, their multifaceted aspects as well as their wide geographical dispersion provides also a sound basis for the pilot simulation approach proposed by RESOLUTE. Additionally, the RESOLUTE models will be open and developed in such a way that they will allow for calibration and testing of the system on different business environments.	Medium	6	3	4	5	81	III – Moderate
Communication failure among RESOLUTE	8	6	6	6	288	II- Severe	Each partner is specialized in his component and	Medium	8	5	4	4	160	III – Moderate

Risk Description	S	O	D	R	RPN	Risk Level	Mitigation Strategy	Possibility	S	O	D	R	RPN	Risk Level
sub-systems							should define during the project the communication problems that could arise along with their possible solution. Partners should also define which components communication with each other and what are the standards of this communication. A series of tests could be performed among the sub-systems during their development. Also agile practices could be used.							
Interference of wireless networks	8	5	3	5	160	III – Moderate	A series of measurements and tests will exclude possible scrambling and will give us details about the support of the wireless equipment.	Medium	8	5	2	4	120	III – Moderate
Performance issues (e.g. too slow)	6	7	5	5	210	III – Moderate	After the implementation of the Pilots, performance issues will be dealt with real time data. The performance will be evaluated and if needed the time and effort will be estimate to improve the system	Medium	6	5	3	4	105	III – Moderate
Loss of Saved Data	8	6	3	2	120	III – Moderate	If data are lost, the system will delay to respond to user needs and learn their habits and preferences. A back up storage of the total amount of data will ensure data security.	Medium	8	4	2	2	64	IV – Slight
Lack of documentation for candidate technologies	6	4	6	5	132	III – Moderate	The documentation provided for candidate technologies is insufficient for appropriate evaluation will be mitigated by contacting manufacturer to get more information.	Medium	6	4	5	4	108	III – Moderate

Calculation of the TRE value following this second stage, using the formula given in section 6.2.4, is as follows:

100% x [(RPN)] / [1000*n(RPN)] = 6.11

Therefore, the risk reduction if all mitigation strategies are correctly applied results of 43%.

7 CONCLUSIONS

This deliverable presents the architecture principles for the RESOLUTE system, through an extensive analysis regarding the various architectural views and layers. In the context of WP4 and collaboration with other RESOLUTE WPs (WP2 and WP3), the layers of RESOLUTE architecture have been presented, namely the Data Management Layer, Mission Critic Layer and Presentation Layer, with a special attention to the CRAMSS component that represents the core of the RESOLUTE system. Towards the final specification, different components and subsystems have been defined and their functional and other critical requirements have been taken into account for deriving the system architecture, presented in this report.

The architectural views presented in this deliverable will further drive the design and implementations during the project lifetime as:

- The use cases and their functional requirements.
- The high level conceptual architecture along with the detailed architectural analysis for each module that constitutes the RESOLUTE system.
- The common ontological framework addresses the lack of ontologies concerning transportation disasters and emergency situations in the domains that RESOLUTE targets and will form the basis for several updates of the ontology.
- The risk analysis provides the necessary feedback to the developers during the design of the system in order to identify potential risks and to further minimize the risk of their appearance.

Conclusively, this deliverable provides a flexible, modular and open architecture which could provide valuable input for standardization work in the domains and systems similar to the RESOLUTE system. Furthermore, this document references and aggregates refined content from all previous WPs through the operationalization of the ERMG output of WP3, in order to orchestrate the specification guidelines for the further modules implementation and their interoperable integration. As also stated in the beginning of this document, during the project lifetime and based on the outcomes of the foreseen pilots, the content presented in this document may be updated and revised to better reflect the final system specifications and architecture.

8 REFERENCES

1. RESOLUTE D2.1 : State of the Art review and assessment report
2. RESOLUTE D2.2 : Synthesis and scoping for RESOLUTE
3. RESOLUTE D3.7 : ERMG adaptation to UTS
4. https://www.techopedia.com/definition/24649/three-layer-architecture
5. http://opensdlc.org/support-files/SDLC-RUP-Hardware-Requirements-Specification.pdf
6. https://en.wikipedia.org/wiki/Representational_state_transfer
7. Bluvband, Grabov and Nakar, Expanding FMEA (EFMEA), IEEE Reliability and Maintainability 2004 Annual Symposium RAMS, 10.1109/RAMS.2004.1285419

ANNEX A: ACRONYM TABLE

API	Application Programming Interface
C&C	Command & Control
CI	Critical Infrastructure
CRAMSS	Collaborative Risk Assessment and Management Support System
DBMS	Data Base Management System
DIM	Data Ingestion Manager
DoW	Description of Work
DR	Disaster Recovery
DSS	Decision Support System
ECA	Event Condition Actions
EFMEA	Extended Failure Modes and Effects Analysis
ERMG	European Resilience Management Guidelines
ESB	Enterprise Service Bus
ESMA	Emergency Support Mobile App
ETL	Extract Transform Load
FRAM	Functional Resonance Analysis Method
GPS	Global Positioning System
GSLB	Global Server Load Balancing
GTM	Global Traffic Management
GUI	Graphic User Interface
HMI	Human Machine Interface
HTTP	Hyper Text Transfer Protocol
IPR	Intellectual Property Rights
JSON	Java Script Object Notation
LAN	Local Area Network
LTZ	Limited Traffic Zones
MDR	Maximum Defect Rate
MTBF	Mean Time Between Failures
MTR	Minimum Time to Recovery
NLP	Neuro Linguistic Programming
OD	Origin Destination
PA	Public Authority
PTA	Public Transport Authority
RDF	Resource Description Framework
RPN	Risk Priority Number
SA	Sentiment Analysis
SotA	State of the Art
SPARQL	SPARQL Protocol and Query Language
SQL	Structured Query Language
TRE	Total Risk Estimate
UPT	Urban Public Transport
UTC	Urban Traffic Control
UTM	Urban Traffic Management
UTS	Urban Transportation System
VMS	Variable Message Sign
WAN	Wide Area Network
WKT	Well Known Text
WP	Work Package
XML	eXtended Mark-up Language