17th International Conference on Space Operations, Dubai, United Arab Emirates, 6 - 10 March 2023.

SpaceOps-2023, ID # 271

Return Link Service Test Bed: a demonstrator platform for new future SAR Galileo Services

C. Cuevas Garcia [a], S. Delattre [b], H. Ruiz [c], M. Ottaviani [d]
CNES, Toulouse, 31401, France
S. Ayachi [e], L. Dubois [f]
Telespazio, France
[a] *SAR RLSP Technical Manager, CNES DTN/TSA/LMC, Cristobal.CuevasGarcia@cnes.fr*
[b] *SAR/Galileo Project Manager, CNES DOA/NT/CS, Sylvain.delattre@cnes.fr*
[c] *Head of software and common means department, DTN/TSA/LMC, Helene.Ruiz@cnes.fr*
[d] *Space ground segment expert, CNES DTN/ISA/SSM, Matthieu.Ottaviani@cnes.fr*
[e] *Software Engineer, Telespazio France, SatNav Department, Soufian.Ayachi@telespazio.com*
[f] *Technical Head of SSA and IOS Services, Telespazio France, Loic.Dubois@telespazio.com*

Abstract

Within the international Search and Rescue COSPAS-SARSAT system, the Return Link Service Provider or RLSP, is described as an operational system responsible for the dissemination of Return Link Messages (RLM) to compatible beacons transmitting distress messages. To accomplish that, the RLSP interfaces are established on one side, with COSPAS-SARSAT system to receive the requests for acknowledgment and on the other side, with the Galileo system to transmit the acknowledgments (Return Link Messages) using the Galileo satellites Signal in Space broadcasting capabilities.

The operations of the SAR/Galileo Service (including the RLSP) are entrusted by the European Union Agency for the Space Program (EUSPA) to CNES acting as the SAR/Galileo Data Service Provider (SGDSP). In this context, the European Commission and EUSPA have established a roadmap to incorporate additional Services exploiting the RLSP capability to establish a downlink communication with RLS compatible beacons or any device equipped with a GNSS receptor.

Within these future Galileo Services, the most relevant ones are the Remote Beacon Activation (RBA), the Emergency Warning System (EWS) and the Two-Way Communication (TWC). Different initiatives have been launched to develop demonstration projects with a limited duration and scope aiming to perform a proof of concept of the above mentioned services, including over-the-air transmissions. Such demonstrators will provide invaluable information to determine the final design; however, they imply a peril for the RLSP operations and thus, a peril to the current service provision.

To cope with this demand, a dedicated infrastructure to host any proof of concept services in safe, controlled and secure environment is being procured. Such infrastructure, also refereed to RLS Test Bed, will be based on a virtualized system capable to handle web-based external requests while securely interfacing with the operational systems using a harmonized approach in line with the existing operational processes.

This paper will present the Agile development methodology chosen to implement the RLS Test Bed in a short-term period and with a design driven by the real needs of the RLS services. The objective is to manage a demonstration platform that could evolve based on its integration with the client web applications and that could integrate all the feedback provided by different stakeholders at each product increment.

The paper will also present the architecture based on container technologies to ease deployment, development and compatibility with clients. The architecture will showcase how manage the interface with the RLSP, a critical system connected to a closed secured environment such as Galileo and the security constraints put in place to avoid any risk to it. In addition, the RLS Test Bed will also provide an interface with other SAR/Galileo sub-systems allowing the SAR data collected by the distributed ground infrastructure to be channeled back to the Service requestors. Then, the paper will introduce the development planning split in several sprints and major releases.

To conclude, the paper will touch upon the roadmap and implementation plans for the future Galileo services integration at the RLS Test Bed.

Keywords: SAR/Galileo, Return Link Service, Return Link Service Test Bed, Future SAR services.

Acronyms/Abbreviations

API	=	Application Programming Interface	MTCF	=	MEOLUT Tracking Coordination Facility
AU	=	Authorised User	RBA	=	Remote Beacon Activation
C/S, CS	=	Cospas/Sarsat			
CNES	=	French National Space Agency	RCC	=	Rescue Coordination Centre
EC	=	European Commission			
EWS	=	Emergency Warning Service	RLM	=	Return Link Message
EUSPA	=	European Union Agency for the Space Program	RLS	=	Return Link Service
			RLSP	=	Return Link Service Provider
GLONASS	=	Global Navigation Satellite System	RTB	=	RLS Test Bed
GNSS	=	Global Navigation Satellite System	SAR	=	Search and Rescue
			SDH	=	SAR Data Hub
GMS	=	Ground Mission Segment	SGSC	=	SAR/Galileo Service Centre
GPS	=	Global Positioning System			
GUI	=	Graphical User Interface	SGDSP	=	SAR/Galileo Data Service Provider
KPI	=	Key Performance Indicator			
MCC	=	Mission Control Centre	SSO	=	Single Sign-On
MEO	=	Medium-altitude Earth Orbit	TOA	=	Time Of Arrival
MEOLUT	=	MEO Local User Terminal	TWC	=	Two Way Communication
MEOSAR	=	MEO Search And Rescue			

1. Introduction

The Search and Rescue (SAR) service of Galileo contains the Return Link Service Provider or RLSP, which is an operational system that feedback the beacons in distress from the SAR system. For that purpose, the RLSP is interfaced with the COSPAS-SARSAT system to receive the requests for acknowledgment and with the GMS to transmit the acknowledgments (Return Link messages) by the Galileo satellites Signal in Space broadcasting capabilities.

The operations of the SAR/Galileo Service (including the RLSP) are entrusted by the European Union Agency for the Space Program (EUSPA) to CNES acting as the SAR/Galileo Data Service Provider (SGDSP). A roadmap has been set up to introduce additional services exploiting the RLSP capability to establish downlink communication with the RLS compatible beacons or any device provided with a GNSS receptor.

For that purpose, a dedicated infrastructure is being procured, in order to host any proof of concept services in safe, controlled and secured environment. Among those proof of concept services, the most relevant ones are the Remote Beacon Activation (RBA), the Emergency Warning System (EWS) and the Two-Way Communication (TWC). Such demonstrators will provide unique feedback and inputs for the real services procurement process, but they also represent a relevant threat for the RLSP system, which is a system connected to a very closed secured

environment such as GMS. Thence, this infrastructure referred to RLS Test Bed is based on a virtualised system capable to handle web-based external requests while securely interfacing with the operational systems.

The RLS Test Bed is being developed following the SCRUM method based on the Agile methodology. This methodology has been chosen to adapt to a short-time period development and to design the product based on the real needs of the future RLS services.

The RLS Test Bed development is carried out under a contract between EUSPA, CNES and Telespazio France.

2. Galileo SAR Service

Galileo supports the **Search and Rescue service** by providing on its satellites a SAR payload that relays beacon distress signal towards COSPAS/SARSAT (1).

Relayed signals from the different satellites are received by one or several Medium altitude Earth Orbit Local User Terminals (MEOLUTs) (2). The MEOLUT determine the location of the beacons, either by demodulating the beacon message or by processing the times of arrival (TOA) and Doppler shifts of the received signals (FOA).

Then, the MEOLUT sends the estimated beacon position and other relevant data to a Mission Control Centre (MCC) (3).

This MCC communicates with the Rescue Coordination Centre (RCC) (4) and the Return Link Service Provider (RLSP) through the French MCC (5). Then, the FMCC sends a Return Link Message Request.

The RCC is in charge of launching a rescue operation (6).

RLSP sends the Return Link Message to GMS (7).

GMS inserts the RLM data in the C band mission uplink (8).

The Return Link Message is downlinked to the beacon in the I/NAV signal in E1B (9).

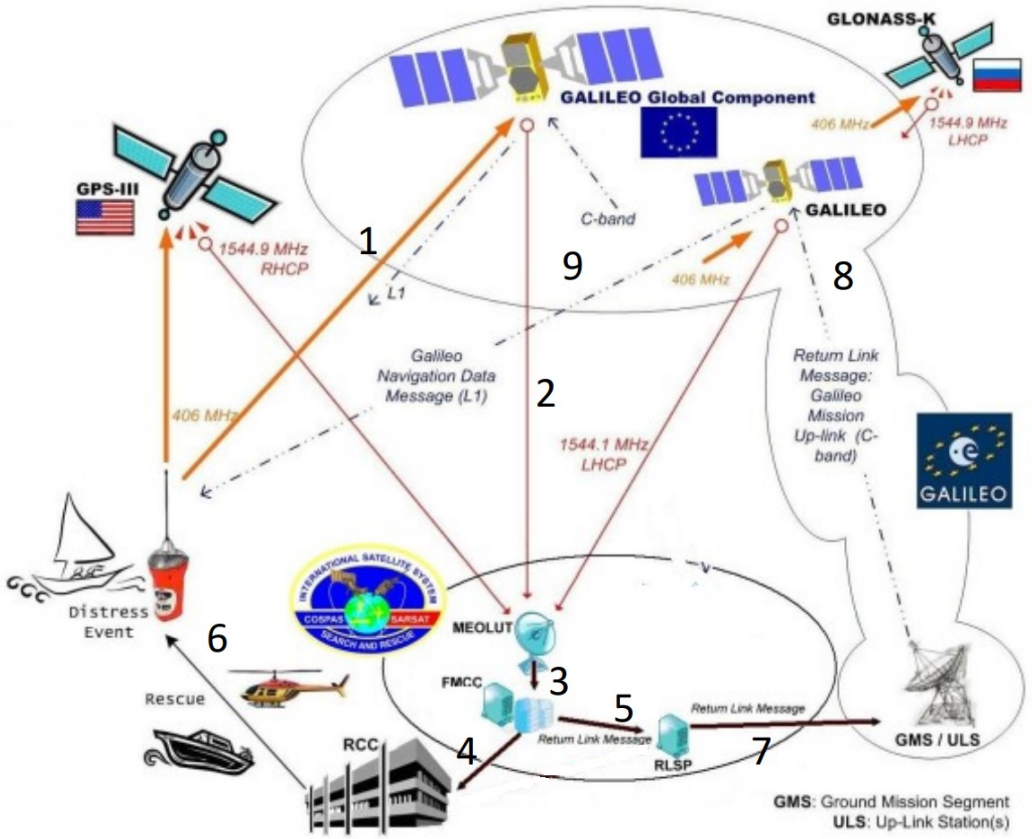

Fig. 1. Galileo SAR Service Concept

3. RLS Test Bed main functions

The RLS Test Bed infrastructure has several purposes that are evolving during the development of the platform. As introduced in section 1, it is a virtualised platform whose objective is to interface in a secured manner with the SAR Service operational means in order to host proof of concept demonstrations projects for future SAR services. Based on this need, the following functions are identified:

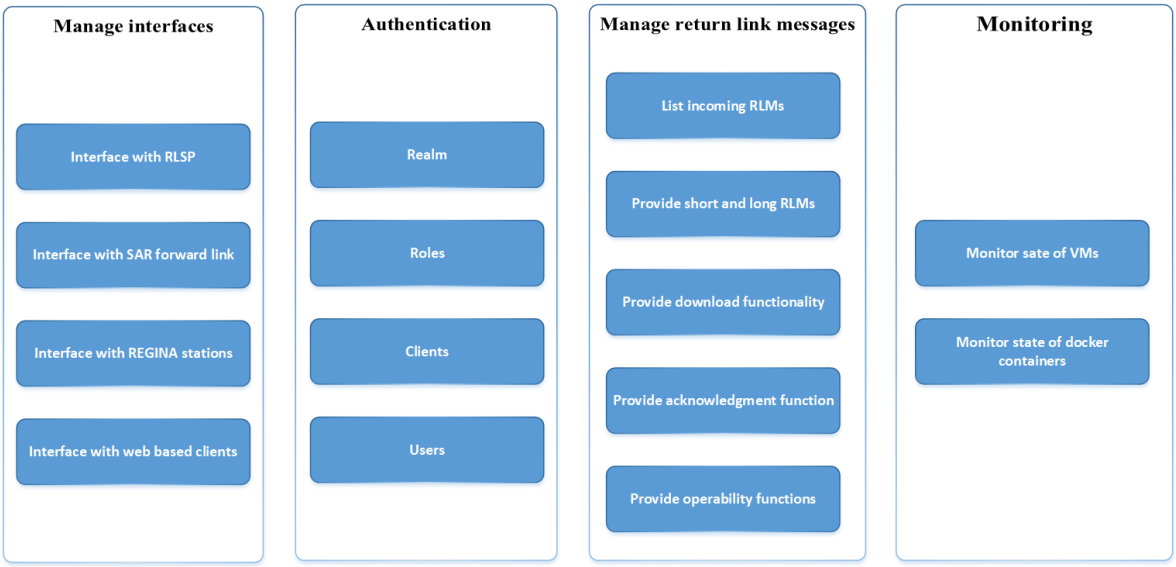

Fig. 2. RLS Test Bed main functions

3.1. *Manage interfaces*

The RLS Test Bed is in interface with the following operational means of the SAR GALILEO service:

- Interface with the RLSP: this interface is responsible for the dissemination of information through the return link service provided by the RLSP through the Galileo Satellites Signal in Space broadcasting.
- Interface with the forward link SAR service: this interface is responsible to retrieve the information coming from triggered SAR beacons through the MEOLUT stations and making accessible to external users of the RLS Test Bed.
- Interface with REGINA stations network: this interface is responsible to connect the RLS Test Bed with the REGINA infrastructure, which is a global network of more than 30 GNSS stations capable of retrieving the signal from current and future navigation systems such as GPS, GLONASS, GALILEO and BEIDOU. The interface is focused on collecting data regarding transmitted RLMs through the RLSP in order to measure the performance of the system during demonstrations.
- Interface with web-based clients: this interface is responsible of making the RLS Test Bed accessible from client web services that need to use the RLS Test Bed for future service demonstrations. For that purpose, three different APIs are developed at the RLS Test Bed: RLSP API, REGINA API and Forward Link API.

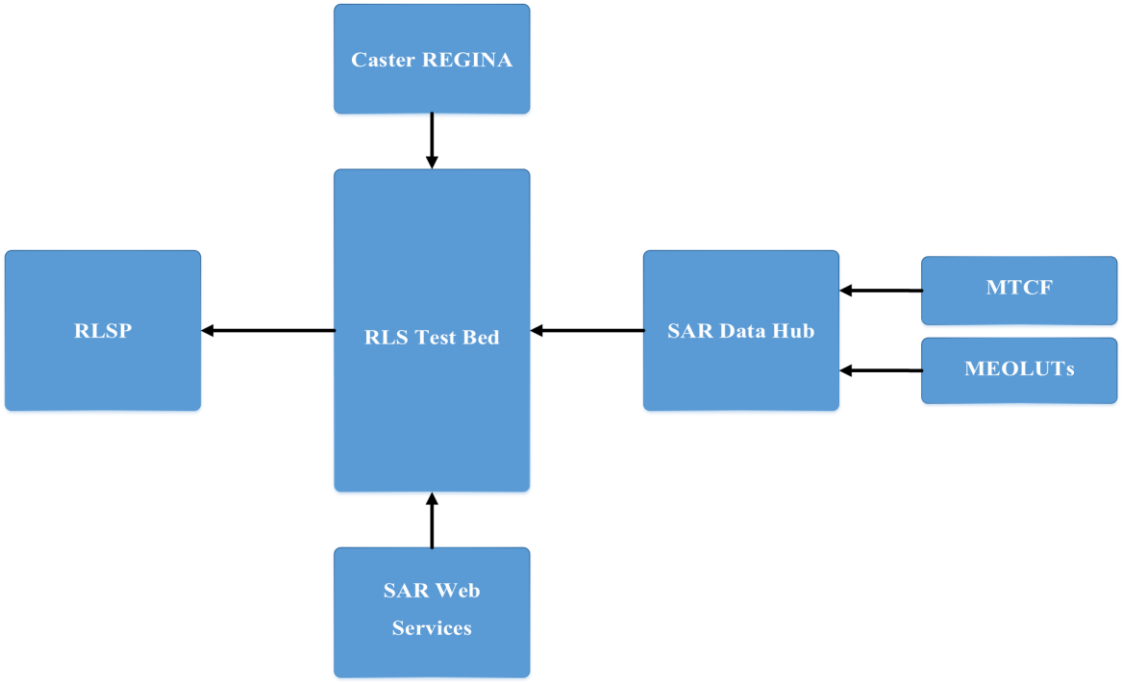

Fig. 3. RLS Test Bed functional external interfaces

3.2. *Authentication*

Authentication is a major issue for the RLS Test Bed, given the fact that is a system in interface with operational means and specially with the RLSP, a strongly secured environment. The RLS Test Bed provides a dedicated module to ensure the authentication of its users by a SSO software. The following concepts are identified:

- Realm: it manages a set of users, credentials, roles and groups.
- Roles: each role has limited access to the three different functionalities of the RLS Test Bed: RLSP API, Forward Link API and REGINA API.
- Clients: web services that are allowed connecting to the RLS Test Bed in order to use the different interfaces that are provided. Each client is assigned with one or several roles.
- User: physical operators that can have access to the RLS Test Bed in order to administrate the system or just use limited functionalities.

3.3. *Manage return link messages*

The RLS Test Bed provides an operational graphical user interface for the SAR GALILEO operators, in order to export RLMs messages to be injected into the RLSP and disseminated through the GMS.

- List the incoming RLMs from web services connected to the RLS Test Bed.
- Provide an API capable of processing short and long RLMs.
- Provide the download functionality for operators.
- Provide the acknowledgment function to alert the external web service about the transmission of an RLM through the RLSP.
- Provide operability functions at the GUI: show priority level, beacon message, creation date, RLM type and classification functions.

3.4. *Monitoring*

The RLS Test Bed provides a monitoring function that allows operators and administrators monitoring the state of the VMs and Docker containers that are part of its core infrastructure. Since it will be a pre-operational mean that will act as a key central point for future SAR services demonstrations, its availability needs to be guaranteed during those crucial points: the web-based service integration and the service demonstrations.

4. RLS Test Bed development methodology

4.1. *SCRUM Method*

SCRUM Method (Agile) has been chosen as the development methodology. Due to the size of the project (one year of development) and the evolving needs during the development of the product, this is the method that better suits with the RLS Test Bed. The following figure recalls the SCRUM Framework:

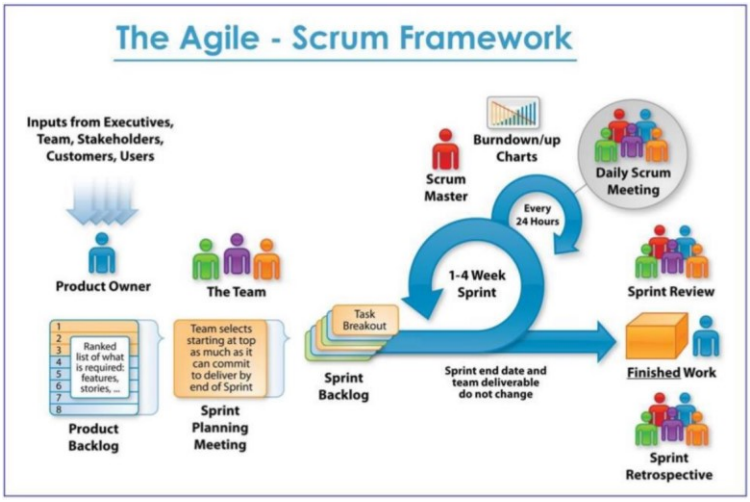

Fig. 4. SCRUM framework

The product owner is responsible of the backlog that contains the necessary items to implement the release of the need. The backlog is prioritized with at the top the most valuable and detailed items. The bottom items can be some imprecise or uncertain needs to be refined for later iterations. The rest of the team is responsible to implement the needs present in the backlog with the help of the Scrum Master that prevents the team to be disturbed by any impediments. This collaborative work between Product Owner and the rest of the team ensures the definition of release and Sprint to remain compliant with the delivery of the most valuable part of the product after each Sprint.

The agile ceremonies are key elements to supervise the project and ensure constant improvement of the team:

Table 1. Agile Ceremonies

Agile ceromony	Definition
Sprint Planning	The perimeter of the Sprint is defined with all the detailed tasks to be realised by the team during the Sprint.
Daily Meeting	The supervision of the Sprint is done by this very short meeting where each member answers to three questions: “What have you done since the last meeting?”, “What do you plan to realise until the next meeting?” and “What are the impediments that you encounter?”
Refinement Meeting	This regular weekly meeting enables the team to estimate the complexity of the new or updated backlog items.
Sprint Demonstration	The realised items are shown to the user in order to validate the end of the Sprint and to collect the user feedback.
Sprint Retrospective	The team analyses the realised Sprint and thinks about the way to improve the next Sprints.

The following indicators are used to supervise the progress of the project:

Table 2. SCRUM indicators

SCRUM indicator	Definition
Burnup or burndown charts	Virtual indicators that enable the Scrum Master to supervise the Sprint in progress.
Velocity	This indicator measures the performance of the team and enable to refine the release plan. It is based on the sum of the complexity for the realised items in the Sprint.

The following RLS Test Bed elements have been tailored for the project:

Table 3. SCRUM RLS Test Bed elements

SCRUM element	Definition
Sprint duration	Three weeks.
Backlog items	The type of the backlog items is defined at the beginning of the project. An Agile customised Jira configuration is used for that purpose, with the following elements present in the backlog: • User Story: element created by the Product Owner that describes a functional need of the RLS Test Bed. • Task: element created by the Product Owner, the Scrum Master or development team that describes specific tasks to carry out to advance in the project. • Sub-task: element created by the development team to split the user stories created by the Product Owner into technical tasks to be carried out and to implement the need on the RLS Test Bed. • Epic: need created by the Product Owner that is very wide and that needs to be split in user stories to be injected in a Sprint backlog of three weeks. • Risk: risks can be also integrated in the product backlog in order to identify mitigation actions. • Bug: anomaly of the product once the RLS Test Bed is mature enough.
Complexity estimation	The items of the backlog are estimated at the beginning of the sprint in order to quantify the effort of the workload. The Fibonacci sequence method is used to estimate the user stories, with the following grades: 1, 2, 3, 5, 8, 13.
Definition of done	The definition of done of the RLS Test Bed is composed of the following elements and is applicable to each User story in the backlog: • Code required to perform user stories ready and committed. • Impacted documentation updated. • If needed for demonstration, dataset available for validation purposes. • If API development, the documentation is automatically and correctly generated after implementation. • Source code is deployed on the CNES platform. • Presentation prepared for demonstration. • Acceptance criteria tested and validated manually. • Compliance to security CNES standard practices. • Compliance to quality standard at CNES.

4.2. *CNES software factory*

The RLS Test Bed is developed and integrated in the software factory of CNES, and a DevOps chain has been set-up for that purpose. The following table summarizes the tools and technologies used during the development:

Table 4. CNES software factory tools

Tool	Description	
Gitlab	Tool used to register the produced code and configuration files and manage their configuration.	GitLab
Artifactory	Tool used as repository to download and upload all the necessary elements to make work the RTB such as Docker Images, Test Files and needed COTS.	JFrog Artifactory
Docker	Container technology that allows to developers building, deploying, running and updating their applications. The containers are components that combine application source code with operating system libraries and dependencies required to run the code in any environment.	docker
PostgreSQL	Database available at CNES infrastructure to register all data SAR coming from MTCF and REGINA and the generated RLMs to be injected into the RLSP.	PostgreSQL
Jenkins	Open-source automation tool to build and test the developed software within the RTB. It allows developers testing their production continuously making easier to integrate changes to the project.	Jenkins
SonarQube	Open-source platform for continuous inspection of code quality.	sonarqube
Jira software	Tool to follow issues and tickets customised to follow an Agile development methodology.	Jira
Confluence	Web-based corporate wiki to manage the documentation of the project.	Confluence

4.3. *Development Planning*

The development of the RLS Test Bed is divided in two major releases and 13 sprints of 3 weeks. The perimeter of the backlog introduced on each Sprint is designed so that the RLS Test Bed starts integrating functional elements from the very beginning. Moreover, the activities related to the documentation production, quality assurance, security issues, validation tests and DevOps integration are also refined during the whole lifecycle of the project and each product increment. Figure 5 illustrates the development roadmap that has been followed.

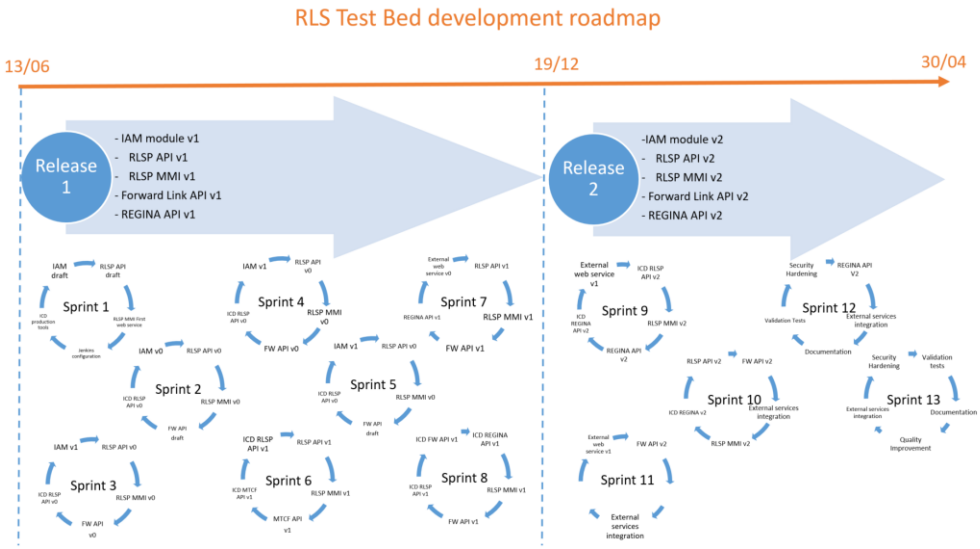

Fig. 5. RLS Test Bed development roadmap

5. RLS Test Bed technologies and design

In this section, the selected tools that make the RLS Test Bed state of the art are presented, together with the design of the interfaces. Those tools have been chosen at the 'Sprint 0' preparatory phase at the beginning of the project, together with the definition of the architecture.

5.1. *RLS Test Bed high level architecture*

The architecture of the RLS Test Bed has been designed taking into consideration the following constraints:

- The RLS Test Bed needs to be hosted in CNES virtualised means.
- The interface with the RLSP cannot be logical due the high security constraints.
- An interface needs to be set-up between the RLS Test Bed and the SAR means in order to recuperate information from the forward link (triggered beacons).
- An interface needs to be set-up with the external web-based clients so that they can access to the SAR information provided by the RLS Test Bed. The external web-based services connect to the RLS Test Bed from internet.

Based on those constraints, the architecture shown on figure 6 was defined. The architecture is segregated in the following elements:

- RLS test bed virtualised environment (1):
  - RLS Test Bed VM: it contains all the elements developed at the RLS Test Bed providing functionalities – the docker Compose containing the APIs, database, Keycloak and NGINX.
  - Internal Web APPs VMs: it contains the external web-based services when the external projects desire to host their applications in the CNES infrastructure.
  - Database PostgreSQL: mutualised database to store SAR information from forward link and incoming RLMs from external web-based services.
- Forward link interface (2):
  - SAR Data Hub: SFTP server to exchange SAR data coming from MEOLUTs and MTCF.
  - MTCF (Meolut Tracking Coordination Facility): it is an operational mean deployed at CNES and is the designated SAR/GALILEO Facility for monitoring and controlling the SGS infrastructure and service.
  - MEOLUTs: European Medium altitude Earth Orbit Local user Terminals. Those are the stations providing the forward link SAR data coming from triggered beacons. There are four stations located in Maspalomas (Spain), Spitsbergen (Norway), Larnaca (Cyprus) and La Réunion (France overseas land).
- RLSP interface (3):
  - CNES office PC: CNES PC used by RLSP operator in order to connect to the RLS Test Bed VM and download RLMs or administrate the platform.
  - Air gap: physical interface consisting on an operator that exchanges RLM files between the RLS Test Bed and the RLSP.
  - RLSP.
- REGINA interface (4):
  - CASTER REGINA: REGINA mission centre component that provides real-time information about RLMs disseminated by GALILEO satellites.
- Internet – External web-based applications (5):
  - CNES firewalls: firewalls to protect the different connections between the different elements at CNES internal and external network.
  - WAF: web application firewall to filter connections coming from internet to the RLS test bed.
  - External Users/External Web apps: physical users or external web-based services that connect to the RLS Test Bed VMs to access to its functionalities.

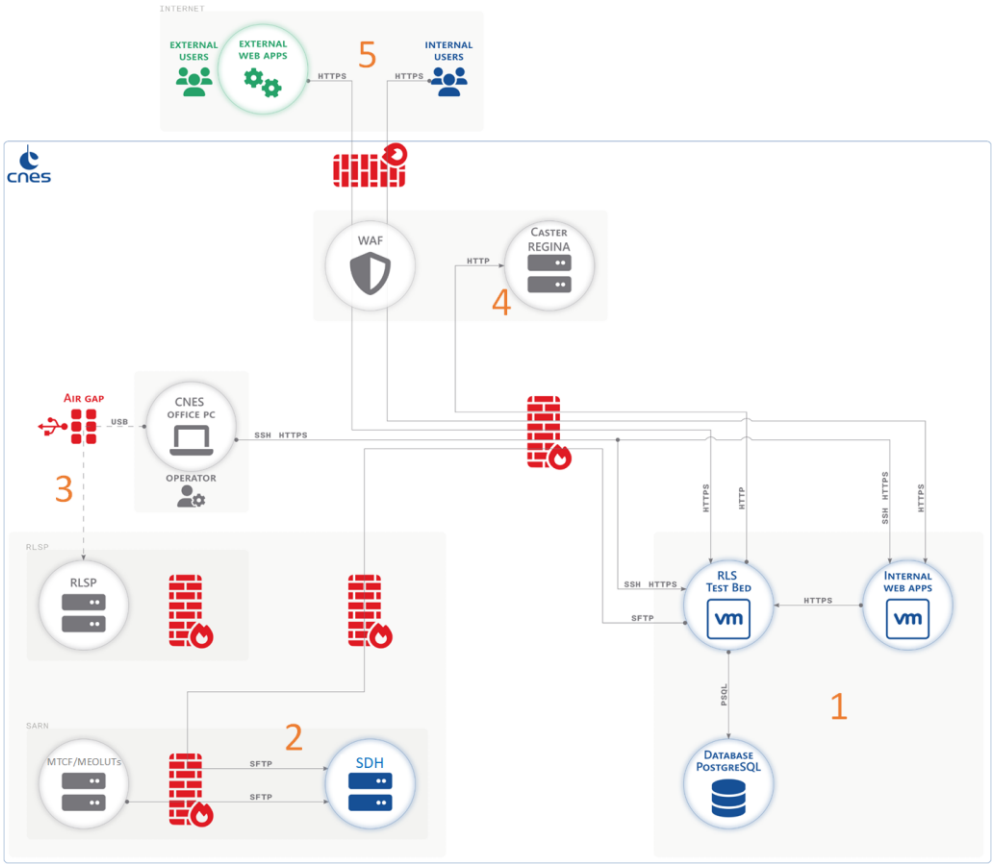

Fig. 6. RLS Test bed high level architecture

5.2. *Docker principle and architecture*

All the RLS Test Bed subsystems run using Docker. Docker is a container technology that allows running multiple services on one host. In the traditional virtualised environments, the service-based infrastructures used to have each service running independently with its own operating service. Containers technologies allow using a host OS and pack everything needed for the service to run: framework, dependencies and container runtime.

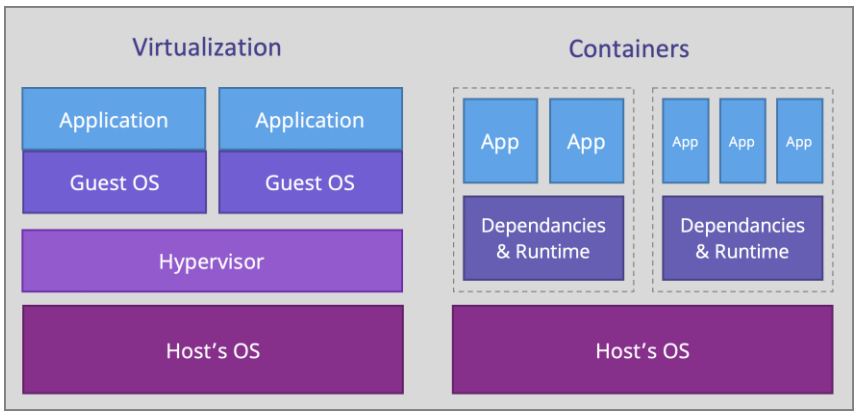

Fig. 7. Containers technology

This technology allows running all the RLS Test Bed functionalities in a single virtual machine on the CNES IT infrastructure. Each functionality (APIs, NGINX, Authentication and database) are running in separate containers that communicate between each other by exposing ports and using a NGINX reverse proxy. As introduced on section

3.1, a dedicated API is present at RLS Test Bed for each of its external interfaces. In the procured micro-service infrastructure, each API is running as a micro service in one single container as depicted in figure 8:

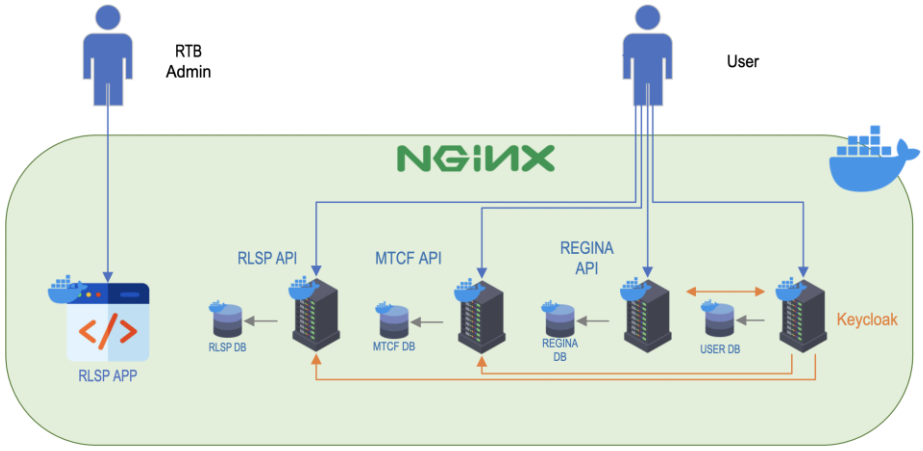

Fig. 8. RLS Test Bed Docker architecture

For the deployment process of the infrastructure, the selected tool is Docker compose. The system uses a single YAML file to build automatically the whole RLS Test Bed system into the CNES VMs.

5.3. *RLS Test Bed subsystems*

5.3.1. *Authentication module*

The selected tool for the authentication purposes is the open source identity and access management solution Keycloak. The operation of Keycloak and more generally the SSO system works in the following way: the client tries to connect to a secure application via an SSO system. For that purpose, a token is retrieved from the SSO software and attach it to all the requests that he would like to make at API. The interest of such a system makes possible to protect several APIs with different rights of use according to the customer accounts.

In the case of the RLS Test Bed, some clients are able to make requests to the RLSP API but not to the Forward Link API and vice versa. This is why the different roles mentioned on section 3.2 have been defined. Exchanges to Keycloak are made by HTTPS requests. The web-based clients are able to request a token using the credentials that they received and use this token to access to the APIs. The following diagram describes the authentication mechanism:

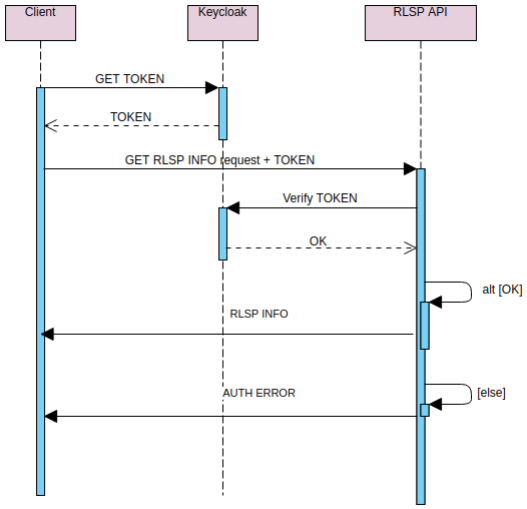

Fig. 9. SSO authentication mechanism

The following table presents the different functionalities related to account management:

Table 4. Keycloak functionalities

Action	Description
Add account for RLSP operator	Create an account for an RLSP operator so that he can have access to the RLS Test Bed MMI functionality and download the incoming RLMs.
Remove account for RLSP operator	Delete existing account for an RLSP operator.
Add external client application	Add an external client application by providing a client ID.
Remove external client application	Remove an external client application.
Add a new role	Add a new role with particular permissions for each API of the RLS Test Bed.
Add role to a client application	Assign a role to an external client application client ID so that it can access the necessary RLS test bed APIs.
Add role to a physical user	Assign a role to a physical user (such as RLSP operator) so that he can access to the RLSP MMI.
Add administration permissions to a physical user	Assign administration permissions to a physical RLSP user so that he can administrate the RLS Test Bed.

The Keycloak open source software is running in a container at the RLS Test Bed, and can be operated through a Graphical User Interface accessible from a browser as depicted in figure 10.

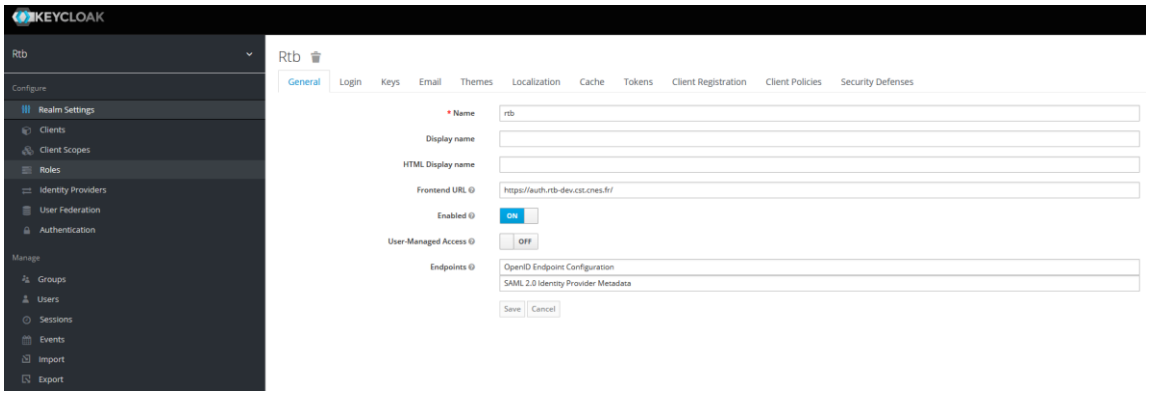

Fig. 10. Keycloak GUI

5.3.2. *RLSP RTB GUI*

The RLSP RTB GUI is a web service running on a single Docker container at the RLS Test Bed that allows operators retrieving incoming RLMs from external web-based services. This graphical interface allows establishing the physical interface between the RLS Test Bed and the RLSP, which is performed by an 'air gap', since no logical connection is allowed due to the security level of RLSP. The air gap is performed by an accredited operator that downloads the incoming RLMs (XML files) and transfer them to the RLSP with a secured USB key.

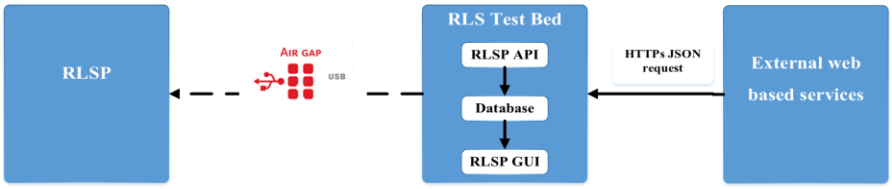

Fig. 11. RLSP GUI interface with GMS

The RLSP GUI provides the functionalities presented in section 3.3. The following picture shows the appearance of the RLSP GUI.

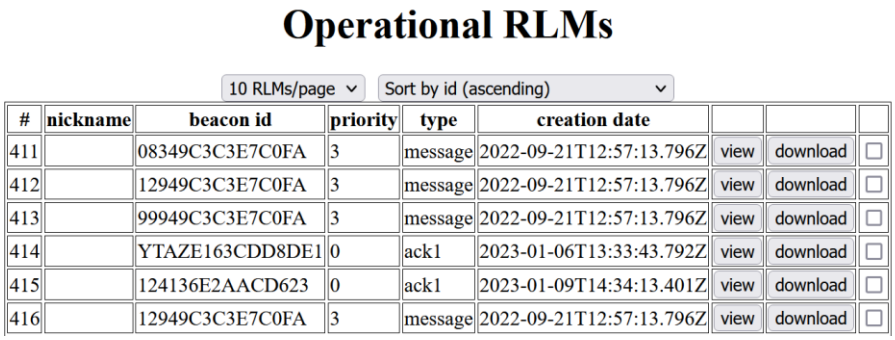

Fig. 12. RLSP GUI

5.3.3. *RLSP API*

The RLSP API is an application running on a single Docker container at the RLS Test Bed and exposed directly to the external web-based services clients. It allows receiving short and long RLM messages and register them at RLS Test Bed database to make them available to the RLSP operators through the GUI.

A set of different HTTPs requests is available for the external applications in order to encode and launch the requests to the RLSP API following a JSON format. The short and long RLMs, provide the possibility to encode 16 and 96 bits of useful information, respectively; and disseminate it through the Galileo satellites.

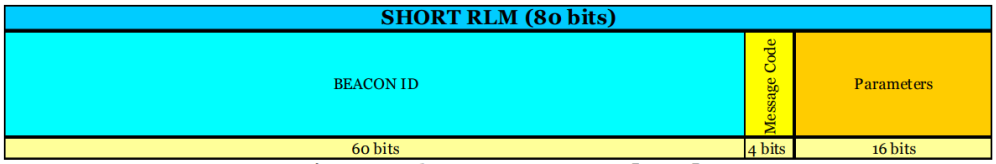

Fig. 13. RLSP short RLM structure

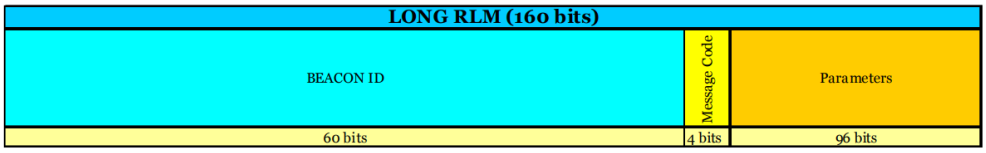

Fig. 14. RLSP long RLM structure

5.3.4. *Forward Link API*

The Forward Link API is an application running on a single Docker container at RLS Test Bed and exposed directly to the external web-based service clients. It allows registering at database and exposing to the clients data from SAR beacons coming from the MEOLUTs and the MTCF. The file exchange between the MEOLUTs/MTCF and the RLS Test Bed is performed through an exchange SFTP server called SAR Data Hub (SDH). The following table presents the data retrieved through this interface:

Table 5. Exchanged information on Forward Link API

File Type	Description	Parameters	Format
Calibrated detections	Information on beacon detections at MEOLUTs.	Beacon ID Beacon Message Calculated Longitude Calculated Latitude Encoded Longitude Encoded Latitude Encoded Altitude Start Time Time of arrival (TOA)	XML
Raw locations	Information on beacon locations at MEOLUTs.		

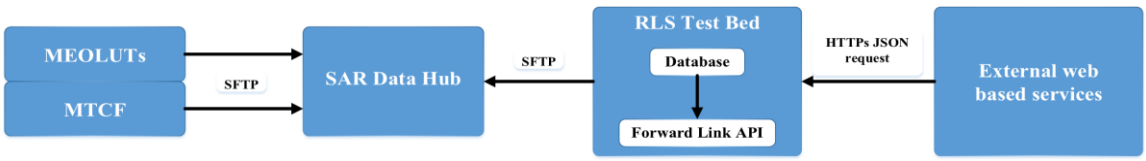

Fig. 15. Forward Link Interface

5.3.5. *REGINA API*

The REGINA API is an application running on a single Docker container at RLS Test Bed and exposed directly to the external web-based service clients. It allows registering at database and exposing to the clients data coming from the compatible RLS REGINA stations around the globe. The objective is to provide to the clients the information about the dissemination of RLMs in order to measure the performance of the services. The information is recuperated at the database of the RLS Test Bed from the CASTER component of the Mission Center REGINA, through a HTTP flow.

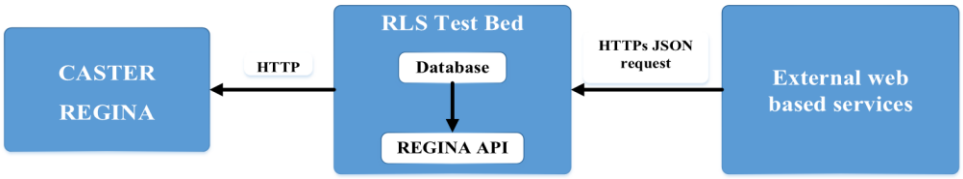

Fig. 16. REGINA Interface

Fig. 17. REGINA station network

6. Future SAR services integration into the RLS Test Bed

The RLS Test Bed will be operated as an interface between external web-based services and the SAR operational means for future service demonstrations. This section aims to present the applications of the RLS test bed to those future services and the functional concept. For further details about the different future SAR services, [1] presents and details the service concepts and their implementation.

6.1. *Emergency Warning System (EWS)*

Warning citizens emergency situations is a crucial objective for national authorities to save lives and reduce the risk induced to emergency services workforce. EU member states are eager to implement the Galileo Emergency Warning System and a roadmap have been established to carry out several demonstrations projects that will showcase the end-to-end process to deliver alert with prototype interface to Galileo System and consolidate the service concepts before the implementation of the operational service.

A solution of opportunity for the implementation of the service consist on setting up an interface with the RLSP to disseminate the Emergency Warning Messages through the Galileo Satellites Signal in Space broadcasting. This solution has many advantages since the RLSP provides very high operational availability (99.95 %) and allows broadcasting the Emergency Warning Message (122 bits) embedded in the structure of a long RLM presented on Figure 14.

On the other hand, the EWS must provide a monitoring function that allows computing KPIs by measuring the performance of the dissemination of the EWMs through the Galileo constellation. For that purpose, the system needs to be in interface with a Ground Stations Network capable of collecting real time Galileo observation and navigation data. REGINA stations is a good candidate since it provides a global coverage to perform demonstrations.

The RLS Test Bed allows integrating web-based services while providing them with those interfaces, as presented in sections 5.3.3 and 5.3.5. The following figure illustrates the system context diagram of the EWS service integrated in the RLS Test Bed.

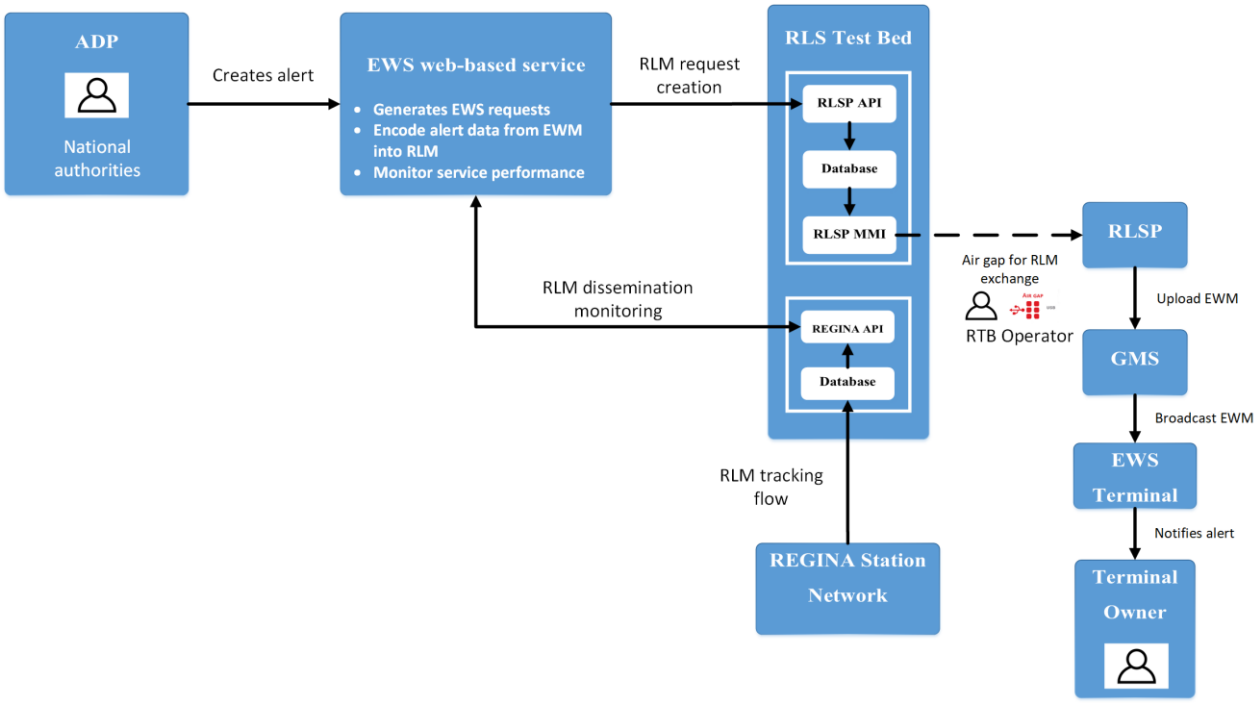

Fig. 18. EWS RLS Test Bed application

6.2. *Two Way Communication service (TWC)*

The two-way communication is a service that will allow the user in distress exchanging short messages with the RCCs to facilitate rescue operations. By establishing an interface with the RLSP, the return link will allow the system disseminating questions and messages to the person in distress through the Galileo Satellites Signal in Space broadcasting.

For the forward link reception, the service needs an interface with the MEOLUTs, so that the RCCs are able to receive the answers from the person in distress and the acknowledgement of reception of the different signals coming from the RCCs.

The RLS Test Bed allows the possibility to integrate a link between the web-based services implementing the TWC service to RCCs and the necessary SAR operational means, by providing the interfaces presented in sections 5.3.3 and 5.3.4. Figure 19 illustrates the system context diagram of the TWC service integrated in the RLS Test Bed.

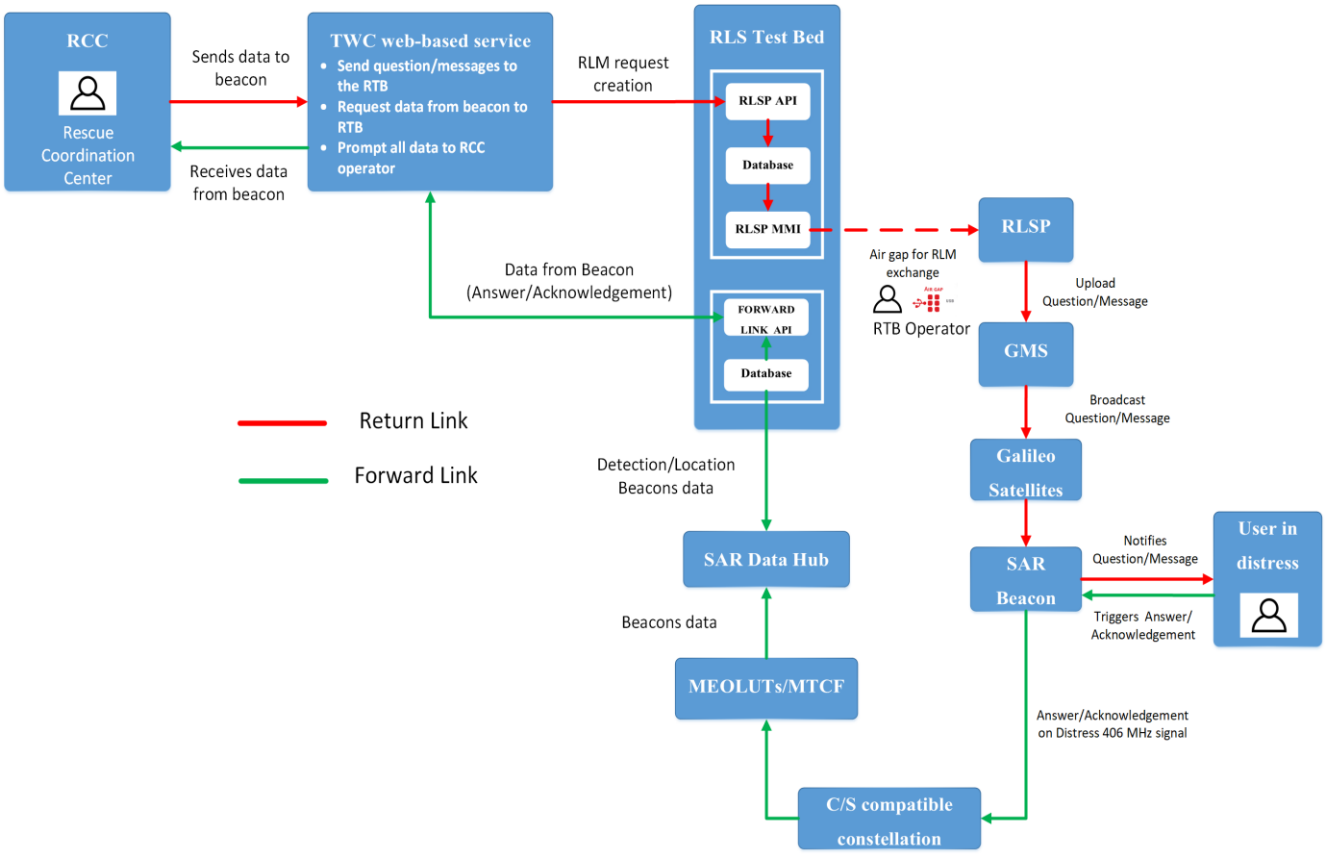

Fig. 19. TWC RLS Test Bed application

6.3. *Remote Beacon Activation (RBA)*

The remote beacon activation is a service that will allow Rescue Coordination Centers or Aircraft companies triggering a SAR beacon on board of a vessel or an aircraft, as an additional (and optional) means of activation. By setting up an interface with the RLSP, the return link will allow the system disseminating the signals to trigger the beacons.

In order to acquire the position of the beacons once they are activated, the system needs to be in interface with the MEOLUTs, so that the RCCs are able to receive the information.

The RLS Test Bed allows the possibility to integrate a link between the web-based services implementing the RBA service and to RCCs and the necessary SAR operational means, by providing the interfaces presented in sections 5.2.3 and 5.2.4. Figure 20 illustrates the system context diagram of the RBA service integrated in the RLS Test Bed.

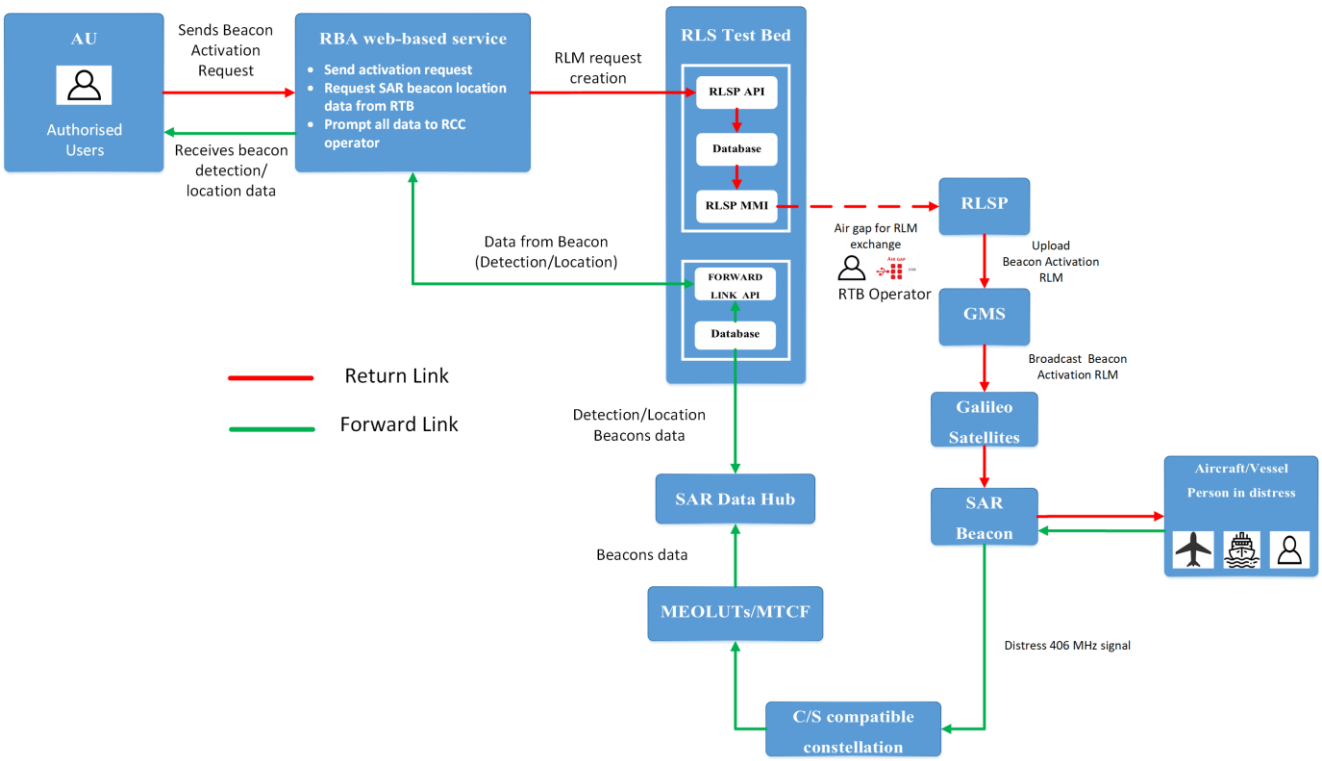

Fig. 20. RBA RLS Test Bed application

7. Conclusions

At the time being, the RLS Test Bed is at the end of its development stage. Most of the functionalities presented on this paper are already operational and the implementation of some of the interfaces needs still to be terminated. The integration of the first external web-based services into the RLS Test Bed has already started in order to perform the first demonstrations on the EWS service and to provide Forward Link real-time data to SAR users.

The RLS Test Bed will play a key role on the definition and integration of future SAR services into the SAR operational means. The feedback obtained during the development and test phase will grant extremely valuable information to SGDSP team that will provide support to European Institutions to properly implement and qualify the future SAR services. The next version of RLSP will englobe on its infrastructure the necessary means to integrate the future SAR services in the SGSC. In the mean-time, given the virtualised nature and scalability of the RLS Test Bed platform, it may act if required as a pre-operational mean to integrate services such as the EWS, RBA and TWC into the SGSC for a preliminary beta phase.

The Agile methodology chosen for its development has eased the possibility to drive the design of the platform based on the needs of the users, and the result is a test bed that establishes an interface in a secured manner between web-based clients and SAR operational means operated and maintained in a closed secured environment.

Acknowledgements

The authors thank other members of the SGDSP project: European Union Agency for the Space Programme, Telespazio France and Cospas/Sarsat team for providing useful information from their various publications.

References

[1] M. Fontanier, S. Delattre, P. Novell, A. Rolla, E. Guyader, Galileo RETURN LINK SERVICE Evolutions, 17th International Conference on Space Operations, Dubai, United Arab Emirates, 6 - 10 March 2023.