PROJECT DELIVERABLE REPORT

Introducing advanced ICT
and Mass Evacuation Vessel design
to ship evacuation and rescue systems

D7.2 Uniform Data Exchange Modules – Interoperability Layer (v2)

A holistic passenger ship evacuation and rescue ecosystem

MG-2-2-2018

Marine Accident Response

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 814962"

Document Information

Grant Agreement Number	814962	Acronym	PALAEMON	
Full Title	A holistic passenger ship evacuation and rescue ecosystem			
Topic	MG-2-2-2018: Marine Accident Response			
Funding scheme	RIA - Research and Innovation action			
Start Date	1st JUNE 2019	Duration	36 months	
Project URL	www.palaemonproject.eu			
EU Project Officer	Georgios CHARALAMPOUS			
Project Coordinator	AIRBUS DEFENCE AND SPACE SAS			
Deliverable	D7.2 Uniform Data Exchange Modules – Interoperability Layer (v1)			
Work Package	WP7 – PALAEMON Integrated System and Technology Validation Trials			
Date of Delivery	Contractual	M32	Actual	M37
Nature	R - Report	Dissemination Level	PU-PUBLIC	
Lead Beneficiary	Atos Spain S.A.			
Responsible Author	Julia Ruiz (ATOS)	Email	julia.ruiz@atos.net	
		Phone		
Reviewer(s):	Alfredo Gardel (UAH), Alexandros Koimtzoglou (NTUA)			
Keywords	ICT, PALAEMON Architecture, Data Models, Interoperability, NGSI-LD			

Authors List

Name	Organization
Julia Ruiz, David Gómez, Carmen Perea	ATOS
Fotis Oikonomou, Panagiota Arampatzi	DANAOS
Kyriakos Giannakis, Sotiris Moustogiannis	KT
Jens Hübel	JU
Nikos Triantafyllou, Petros Kavassalis	UAEG
Manuel Ramiro	ADV
Alfredo Gardel, Ignacio Bravo	UAH
Alexandros Koimtzoglou, Natasa Danopoulou	NTUA
Elias Chatzidouros	ESI
Francesco Piantini, Marco Merlini	THALIT
Julia Ruiz, David Gómez, Carmen Perea	ATOS
Fotis Oikonomou, Panagiota Arampatzi	DANAOS
Kyriakos Giannakis, Sotiris Moustogiannis	KT

Revision History

Version	Date	Responsible	Description/Remarks/Reason for changes
0.1	2021/01/04	ATOS	ToC first draft and initial list of contributors
0.2	2021/10/25	ATOS	First round of contributions
0.3	2022/04/26	ALL	Partners' contributions
0.4	2022/05/03	ATOS	Section 3
0.5	2022/06/13	ATOS	Section 4 (compiled tables)
0.6	2022/06/15	ATOS	Section 5
0.7	2022/06/23	ATOS	Section 2
0.8	2022/06/28	ATOS	Intro + conclusions
0.9	2022/06/29	UAH + NTUA	Internal review
0.95	2022/06/30	ATOS	Update with internal reviewers' feedback
1.0	2022/06/30	ATOS	Review and Release (submitted to EC portal)

Contents

List of Figures

List of Tables

Abbreviations

AIS	Automatic Identification System
API	Application Programming Interface
AR	Augmented Reality
CIM	Context Information Management
DFB	Data Fusion Bus
DSS	Decision Support System
ETSI	European Telecommunications Standardization Institute
GE	Generic Enabler
GPS	Global Positioning System
GUI	Graphical User Interface
ICT	Information and Communication Technologies
IoT	Internet of Things
MQTT	MQ Telemetry Transport
NGSI-LD	Next Generation Sensors Initiative – Linked Data
NMEA	National Marine Electronics Association
PaMEAS	PALAEMON Mustering and Evacuation Process Automation System
PIMM	PALAEMON Incident Management Module
SHM	Ship Health Monitoring
SRAP	Smart Risk Assessment Platform
SSL	Secure Sockets Layer
SSS	Smart Safety System
SST	Ship Stability Toolkit
UAV	Unmanned Aerial Vehicle
URN	Uniform Resource Name
VDES	VHF Data Exchange System
VDR	Voyage Data Recorder
VRG	Voyage Report Generator
WFT	Weather Forecast Toolkit

1 Introduction

> *DISCLAIMER: The reader must know beforehand that, due to an error in WP7's deliverables numbering, this document will be officially referred herein to as D7.2 (and not D7.3, as it should be). This is a bug that we drag from the Grant Agreement and, unfortunately, we have not fixed in form of amendment. A direct consequence is that all WP7's deliverables numbering may be erroneous as well and the reader might see a mismatch between the references of this document and others belonging to different WPs.*

One of the main challenges of today's (Internet of Things) IoT infrastructures is the lack of off-the-shelf interoperability among components. Unless all sensors and devices come from a common manufacturer (or a number of them belonging to the same group), the typical situation is that each one follows its own data model, leading as a result to a so-called vertical lock-in, where a thorough adaptation work is deemed necessary. Nonetheless, it is worth acknowledging the efforts of a handful of initiatives, which are striving to settle down – and even standardize – a common vision that guarantees a complete and transparent interconnection and understanding among devices. Namely, we have chosen FIWARE [1] as the base framework to stick to, not only for defining the formats, but also to leverage some of their core components to offer a reliable way to stream context information.

In this document we focus on the specification of the PALAEMON Information Model, the "common language" defined in the scope of the project to support a seamless communication among all the components that shape the PALAEMON system. In past reports we introduced and tackled relevant aspects, as the Reference Architecture, split into two iterations or releases (D2.6 – PALAEMON Architecture v1 [2] - coped with the so-called v1 and in D2.7 – PALAEMON Architecture v2 [3] - all the updates gave rise to v2), where we designed all the logical connections. Going a step atop this, in D7.1 (D6.6 following the official-and-wrong numbering) we addressed the concept of Communications Platform [4], where we "landed" the abstraction level fostered in the aforementioned documents, underpinning aspects like the physical connections or the security layer between services and the outer world. Besides these core documents, in D6.2 – PALAEMON Data Fusion Middleware [5] – we delved into the cornerstone of the system, the so-called Data Fusion Bus (DFB), responsible for the real-time data exchange and storage of the system.

The purpose of this deliverable is to define the steps that were followed to make all the information in the DFB to be presented in a uniform format. To achieve this goal, we have stuck to an open-source framework (i.e., FIWARE and its underlying NGSI-LD information model). We have also defined an information pipeline that permitted us to transform data on-the-fly, thus the outgoing information shall be understandable by every component in the pipeline.

The document is structured as follows: Section 2 introduces the FIWARE initiative and outlines NGSI-LD and the information model we have chosen to format all PALAEMON-related data. Section 3 zooms into the part of the Reference Architecture that homogenizes the data to be seamlessly understandable by all the components. Section 4 compiles the messages that flow across the PALAEMON Communications Platform. Section 5 applies all the above points and connects with the way components handle an evacuation process by modifying their own operation models. Finally, Section 6 properly closes the document and links with the subsequent WP7's deliverables.

2 FIWARE Overview

Before presenting the information model we have chosen to homogenize the data that goes across PALAEMON's components, it is deemed necessary to introduce FIWARE [1]. FIWARE is an open-source software platform, promoted by the European Union, which was developed to facilitate the creation of applications and services spanning different domains, such as Internet of Things (IoT), Big Data, smart cities, e-health, smart manufacturing, etc. On this regard, it is worth mentioning the Smart Data Models initiative [6], a collaborative approach to define a wide spectrum of common data models to guarantee replicable and standardized smart solutions.

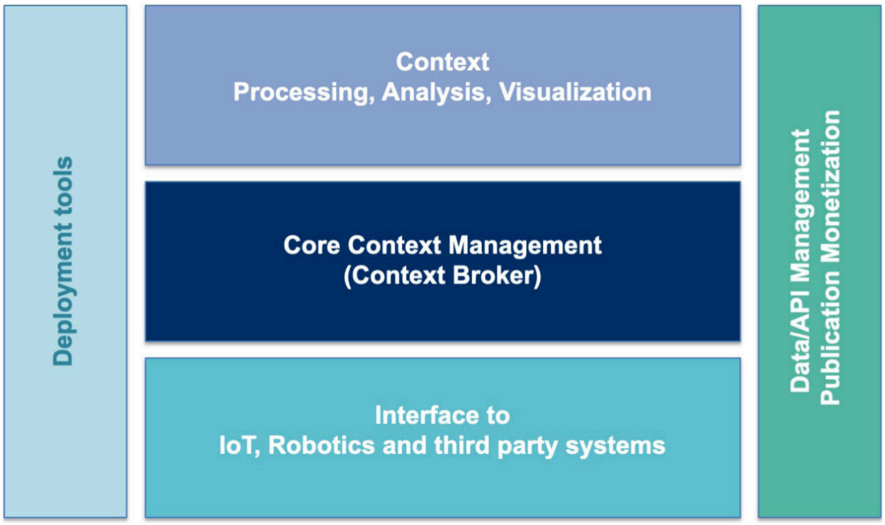


Figure 1. FIWARE Catalogue chapters. Source: FIWARE Foundation [7]

From a technological standpoint and for the sake of illustration (see Figure *1*), FIWARE can be interpreted as a collection of components that can be interconnected, combined and deployed. In a nutshell, its primary objective is to manage large-scale context information and to build an open and interoperable ecosystem on top of that. As shown in the figure, the cornerstone of the whole FIWARE system is the so-called "Context Broker", core component that leans on the Next Generation Service Interfaces (NGSI) standard. This specification nails down the information model(s) and Application Programming Interfaces (APIs) that data providers and consumers must follow when it comes to interact with the Context Broker. Hinging around this broker, FIWARE curates and compiles [8] a handful of Generic Enablers (GEs), that is, a list of software modules that can be assembled to build a smart solution "powered by" FIWARE.

Shifting to a slightly different topic and touching upon a strategical point of view, the main actor behind the platform is the FIWARE Foundation, a legal-and-independent body that aims to help in the promotion and growth of the platform. Alongside this open community, we can stand out active members like ATOS, Telefónica, Orange or NEC. As a matter of fact, at the time of writing this deliverable, the list of members spans up to 127 affiliates [9], clustered in four main categories: Platinum, Gold, Gold Strategic End User and Associates.

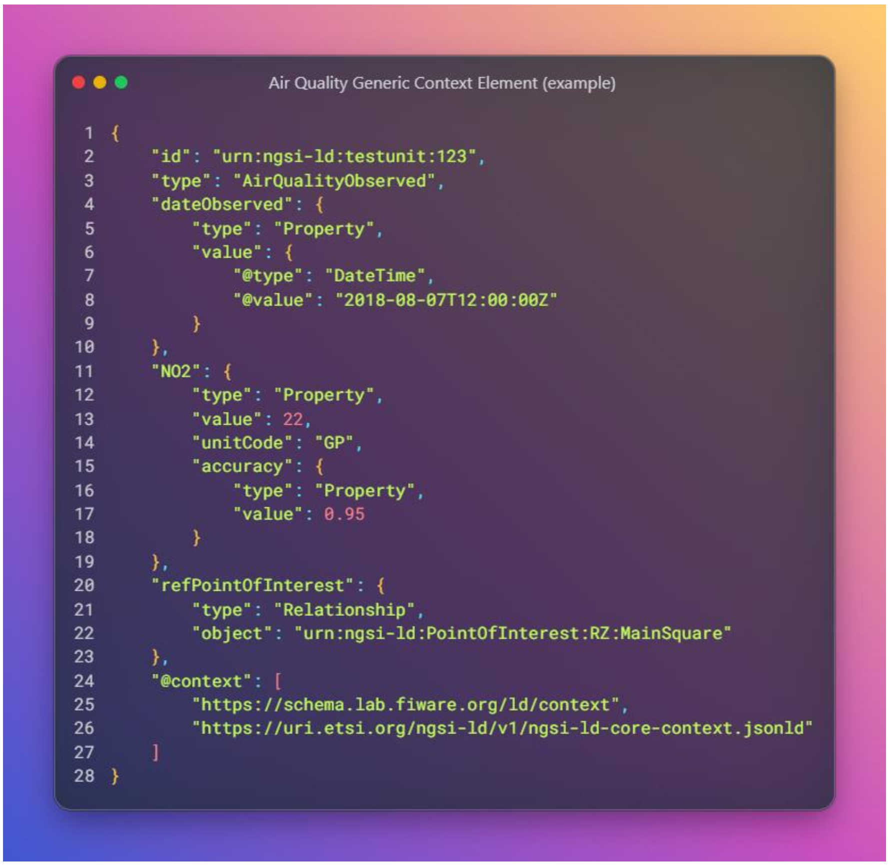
Linking back to PALAEMON, FIWARE was chosen due to its decentralized data ecosystem, fruit of the holistic operation between building blocks (i.e., Generic Enablers) and smart data models, thus offering a cradle for interoperability, sovereignty and trust over the whole information workflow.

2.1 NGSI-LD standard & Scorpio Broker

As hinted above, FIWARE relies on NGSI as the standard that describes both information models and APIs. They, altogether, settle down a well-defined communication ecosystem between Generic Enablers. Since the birth of FIWARE back in 2021, at least three version of NGSI have been specified. Whilst v1 and v2 (mainly the latest one) still keep a good ratio of popularity these days, in the scope of PALAEMON NGSI-LD (Linked Data) is. preferred. The reason behind this choice is threefold:

1. NGSI-LD is actually an European Telecommunications Standard Institute (ETSI) [10] standard, abridged as ETSI Context Information Management (CIM) NGSI-LD API [11].
2. Beside the definition (and standardization) of a common information model, NGSI-LD opens the door to the utilization of link data, thus binding with the potential utilization of semantic technologies. Despite this is not the objective of PALAEMON and it is out of the scope of this task, it is worth remarking the potential of this technologies, mainly when it comes to identity relationships among entities and properties. For the sake of illustration, link data could be seen as a humungous graph where the different parts of the messages are interconnected.
3. Scorpio Broker [12], the core context management we have chosen (this is not the only implementation for NGSI-LD), is based on Kafka, thus sharing a common communication channel with our Reference Architecture. In consequence, we could use the same instance of DFB's Kafka Broker could be utilised for both pipelines.

To help the reader land the concept of how an NGSI-LD data model would look like, Figure *2* presents an arbitrary example of context information, i.e., a sensor measuring air quality data.




```
{
    "id": "urn:ngsi-ld:testunit:123",
    "type": "AirQualityObserved",
    "dateObserved": {
        "type": "Property",
        "value": {
            "@type": "DateTime",
            "@value": "2018-08-07T12:00:00Z"
        }
    },
    "NO2": {
        "type": "Property",
        "value": 22,
        "unitCode": "GP",
        "accuracy": {
            "type": "Property",
            "value": 0.95
        }
    },
    "refPointOfInterest": {
        "type": "Relationship",
        "object": "urn:ngsi-ld:PointOfInterest:RZ:MainSquare"
    },
    "@context": [
        "https://schema.lab.fiware.org/ld/context",
        "https://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.jsonld"
    ]
}
```


Figure 2. NGSI-LD Context Element example. Source: Scorpio GitHub repository [13]

Shortly, the part of the message can be easily identified and the common elements that will be presented in all of the data models can be guessed.

- **id**: Unique ID or Uniform Resource Name (URN), that is, as string that serve as main identifier of the context entity (e.g., sensor, actuator, etc.).
- **type**: Easy to infer, this one refers to the type of context entity (and underlying data model). A list of context entities can be found in the Smart Data Models repository [6].
- **dateObserved**: Timestamp of, in this case, the observation of the physical phenomena. We can also infer, from the non-straightforward notation, the possibility of using metadata for advanced and additional features.
- **NO2**: For this case, this key is particular to the type of entity. This could be extended to temperature, humidity (i.e., weather stations), location (for those entities whose positioning brings relevant information), wave height (e.g., tide sensors, etc.).
- **refPointOfInterest**: Rather than a particular property of the air quality sensor (i.e., this example), this field is used to bind to another entity, in this case, a Point of Interest, which may encompass a number of different context elements. As a matter of fact, this is something we will not explore in depth in PALAEMON.
- **@context**: this last property is used to link to the ontology that is used to define the information used in the NGSI-LD notation. Again, this is completely out of the scope of the project.

After having shallowly tackled the information model, it is important to touch upon some basic characteristics of the Scorpio Broker, component that we have seen present as part of the PALAEMON Reference Architecture. We will cover this in the next section. Namely, sticking to the nomenclature of the Figure *1*, Scorpio Broker would correspond to the “Context Broker” block. With a purely illustrative purpose and without entering into the details, Figure 3 presents the inner architecture of Scorpio Broker.

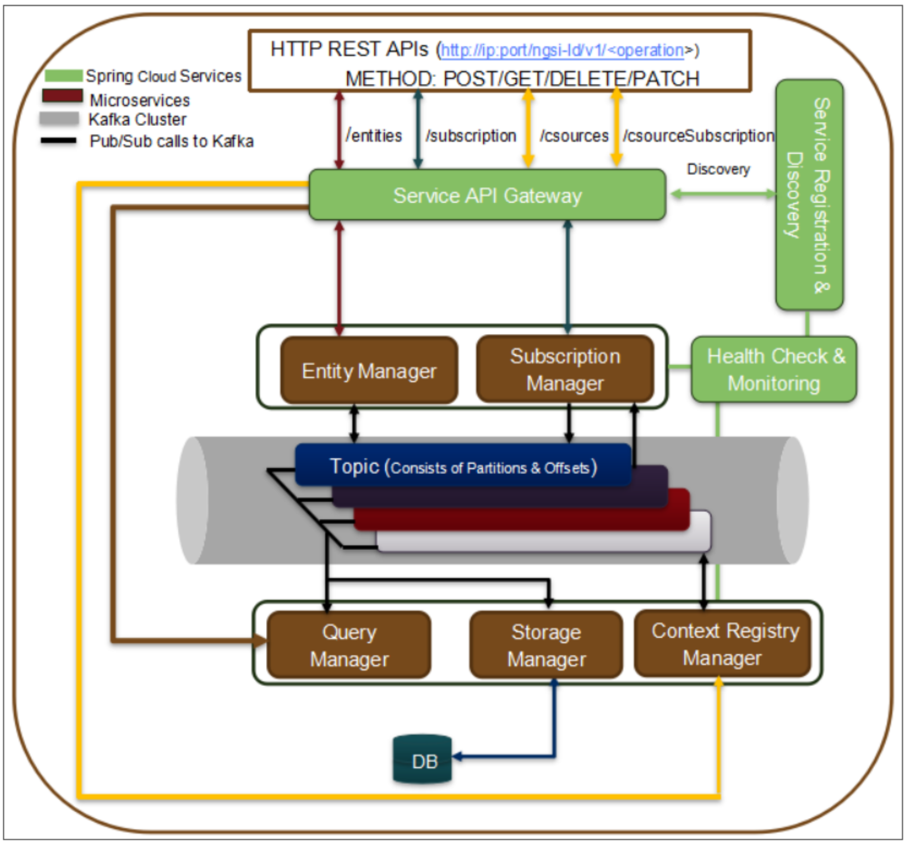


Figure 3. Scorpio Broker Deployment Architecture. Source *[12]*

Amongst all the blocks and connections we can see there, we must highlight two: the Kafka cluster, displayed as the grey pipe in the figure, responsible for the handling of the events streaming. On the other hand, the Service API (green boxes), direct interface with external applications, where they can perform a handful of operations, like the ones gathered below, which have a direct utilization in the project.

- Entity creation
- Entity update/append
- Entity subscription
- Query

Technically speaking, the underlying technology behind this interface is RESTful, which will coexist with Kafka to query, fetch and deliver all the context information that goes across the PALAEMON platform.

3 PALAEMON Information model

It is worth recalling that, till this point in time, the concepts of PALEMON Reference Architecture and PALAEMON Communications Platform have been addressed in a series of deliverables. As a matter of fact, in D2.6 (PALAEMON Architecture v1) [2] and D2.7 (PALAEMON Architecture v2) [3], the path we followed to nail down the logical connections between the components that intervene during an evacuation process (in the eyes of the PALAEMON system, of course) has been reported. Beside this vision, D7.1 (PALAEMON Communications Platform) [4] completes the picture and defines the physical connections among the software components.

For the very context of this deliverable, it is deemed necessary that we bring back once again the PALAEMON Reference Architecture, illustrated in Figure 4. For the sake of refreshment, the figure spans, from left to right, the information flow between the sources that generate the information and the outputs of the system, that is, the points from which stakeholders (e.g., Master, crew, passengers, etc.) do interact with the platform.

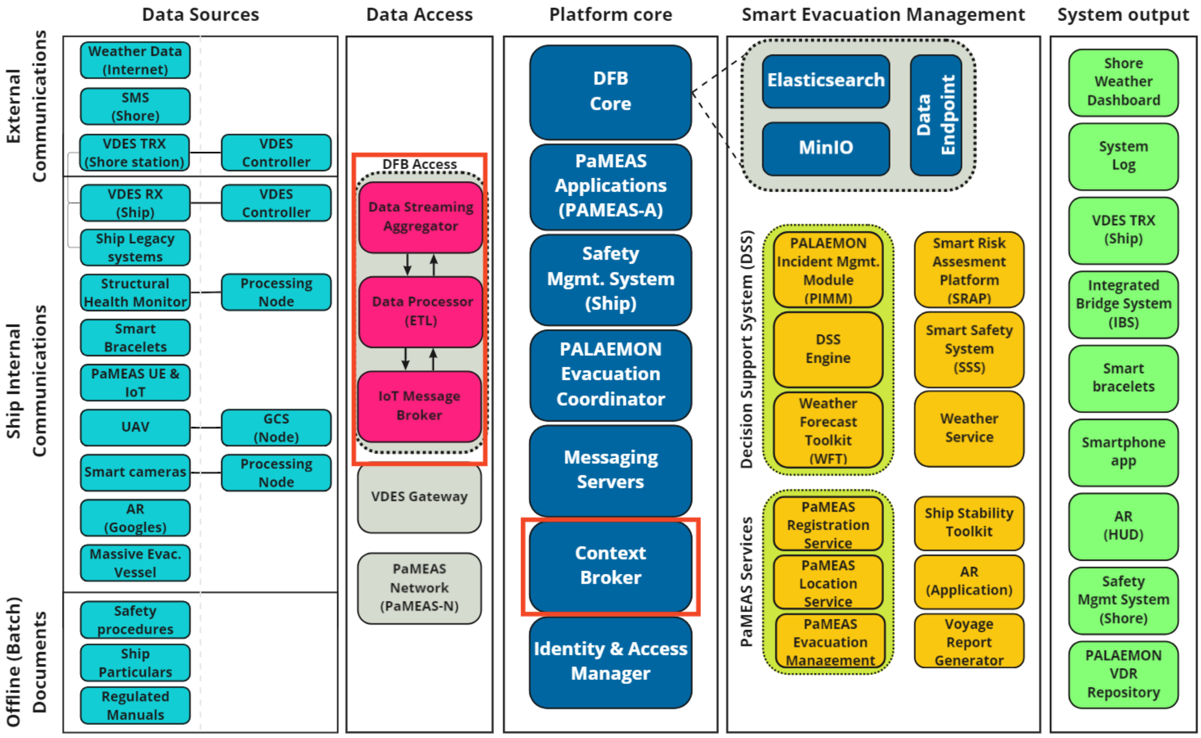


Figure 4. PALAEMON Reference Architecture v2.1, highlighting (framed within red rectangles) the elements responsible for the data interoperability

Among all this handful of blocks, we have highlighted in red the parts that undertake the tasks framed in this deliverable. Namely, the access part of the Data Fusion Bus is responsible for gathering and transforming data on-the-fly, in order to adequate it as an NGSI-LD-compliant data model. In addition, the Context Broker (i.e., Scorpio) directly binds PALAEMON and FIWARE ecosystems, thus making it possible to leverage its main features, such as the possibility of publishing, subscribing or even querying context information (i.e., data coming from the various sources).

In the next part of the section, we describe how we have managed to deal with this homogenization exercise, thus guaranteeing the seamless interoperability (and communication) among PALAEMON components.

3.1 Data Fusion Bus Access

The PALAEMON Communications Platform hinges around the Data Fusion Bus. As we have already described in depth in former deliverables, like D2.6 [2], D2.7 [3] or (mainly) D6.2 PALAEMON Data Fusion Middleware) [5], the focus was mainly on the ingestion and aggregation/combination of data, coming from multiple (and potentially heterogeneous) sources. Obviously, we can apply this diversity to the formats (i.e., information models) the information reaches the platform. If we just dump the data and proceed to store it as it is, upper layer components (i.e., smart evacuation services) will have to handle each format individually. In consequence, every component/service would have to carry out an adapter per module it would like to communicate with. For this reason, the data pipeline displayed in Figure 5 was developed.

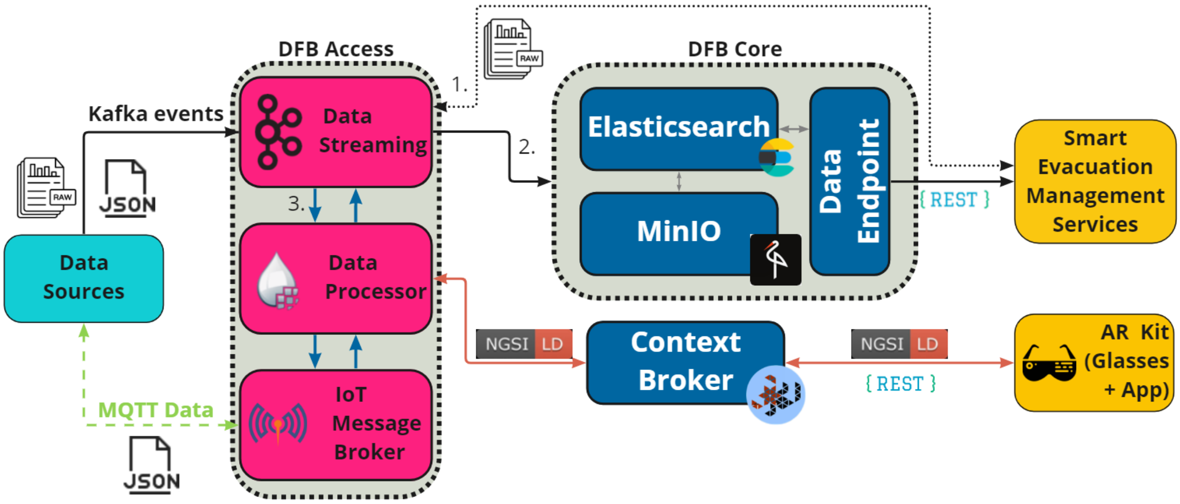


Figure 5. Data Fusion Bus interoperability flow.

At the left side, Data Sources (we do not need to differentiate them here, as the treatment is the same for all) send the raw information (in this case, JSON, but we could extend the analysis to any kind of serialization format) to DFB Access, heading the Kafka Broker [14] (Data Streaming in the context of the picture). In essence, most of the information generated in the PALAEMON Communication System passes through the Kafka Broker. As main outputs, the Broker opens up to three different paths, whose starting points are explicitly marked in the figure:

1. (Dotted black line) The utilization of Kafka-based streaming notifications is also a possibility for smart evacuation services, which have the possibility to either gather data from the sources, or to exchange data among them. To illustrate this with an example, the Evacuation Coordinator broadcasts an Evacuation Status Change through a common and well-known Kafka topic. The particularity of this first option, compared to the second one, is that the information goes near real-time, and a negligible latency is expected.
2. (Straight black line) The raw information is directly forwarded to the core of the DFB, where it will be properly stored in the different databases (i.e., Elasticsearch and MinIO). From this, if services want to retrieve the information, they will have to query the Data Endpoint via a tailored (and protected) RESTful API. As a matter of fact, the process of mapping from Kafka and saving the data on Elasticsearch introduces a

notable latency, meaning that the data would not be visible from the API for some seconds (between 5 and 10 seconds[1]).

3. (Straight blue line) For this last option, there is a direct communication between Kafka and Apache NiFi (i.e., Data Processor) [15], a so-called Extract, Transform and Load (ETL) tool that connects (i.e., subscribes) to the Kafka events. As described in Section 3.2, NiFi takes the raw information as input and generates an NGSI-LD output, which feeds the second broker of the system, Scorpio. At this point, services could use regular RESTful APIs to subscribe to event notifications and get the data (NGSI-LD-ish). As a matter of fact, the main use case of this pipeline has to do with Augmented Reality (AR) glasses, whose development libraries are not natively compatible with the utilization of Kafka alongside Secure Sockets Layer (SSL) certificates. As alternative and to harness the benefits of this second and standardized interface, the glasses will get the data without having to send periodic polls to DFB's Data Endpoint (which would have led to a significative delay). Moreover, it should be noted that the transformed data might be sent back to Kafka on a different topic, so evacuation services could also get the events on NGSI-LD format.

Last but not least, we must define a special case, illustrated with a dashed red arrow between the IoT Message Broker and Data Sources. From a technical point of view (this aspect will be explained in depth in D7.2 – Encryption and Authentication Mechanisms [16]), PALAEMON introduces some lightweight or embedded devices (e.g., smart bracelets), whose limited computational capacity impedes the utilization of Kafka combined with SSL for the authentication (likewise the AR glasses). For this reason, we activated a plan B that relied on MQ Telemetry Data (MQTT) [17], and an authentication schema based on user and password, a simpler option that allows a secure communication with these elements. To achieve this, Apache NiFi gets the data from Kafka, transforms it (if needed) and delivers these new messages on a MQTT channel, whose traffic is centralized in this case with a Mosquitto Broker [18]. As the communication is bidirectional, Kafka is translated/forwarded to MQTT and the other way around, always following the same pipeline: Data sources ⇄ MQTT ⇄ NiFi ⇄ Kafka.

3.2 Data transformation flow

One of the implicit advantages of using a platform like DFB is the flexibility it offers across its main components (i.e., Kafka/MQTT and NiFi). As we described in the last paragraph of the previous section, Apache NiFi is an ace up the sleeve that allows to straightforwardly define on-the-fly transformations, thus easily converting from raw data to standardized NGSI-LD context information. This way, components would be able to hook at this new data format if they want/need to. In fact, what can be called Data Transformation flow is displayed in Figure 6, where the raw inputs mainly come from Kafka or MQTT events, and after a transformation process (we will introduce how below), the outgoing NGSI-LD data is forwarded to a number of destinations (i.e., Apache Kafka, MQTT, Scorpio Broker, Elasticsearch and MinIO).

[1] These figures are approximate and based on our own experience.

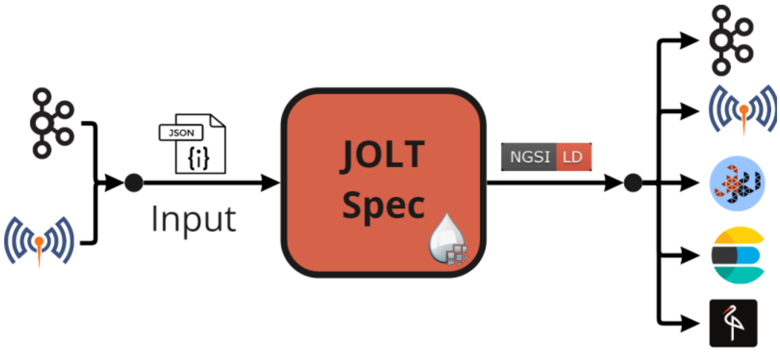


Figure 6. JOLT on-the-fly data transformations flow in PALAEMON.

Among the portfolio of processors available within Apache NiFi, we have found JOLT (JsOn Language for Transform) [19], a library for transforming JSON files on-the-fly. To help the reader understand the process we have followed to make the data interoperable (i.e., NGSI-LD compliant), Figure 7 displays a flow on Apache NiFi that starts (upper node) from a subscription to the ***weather-service*** topic. In the next step, a JOLT processor gets the raw JSON of the previous component as input and generates a new JSON object, this time following NGSI-LD principles. Finally, the flow is split into three parallel tasks: on the left, the data is sent to Kafka (this time through a different topic, ***weather-service-ngsi-ld***), the center processor is responsible to deliver the message through MQTT (***weather-service*** topic); on the right, the data is handed, via RESTful API, to the Scorpio Context Broker. In comparison with Figure 6, we should miss two additional nodes to send the data to Elasticsearch and MinIO; however, DFB has an intrinsic connector that dumps all Kafka events as Elastic indices (and likewise to MinIO), so the process is completely transparent to developers.

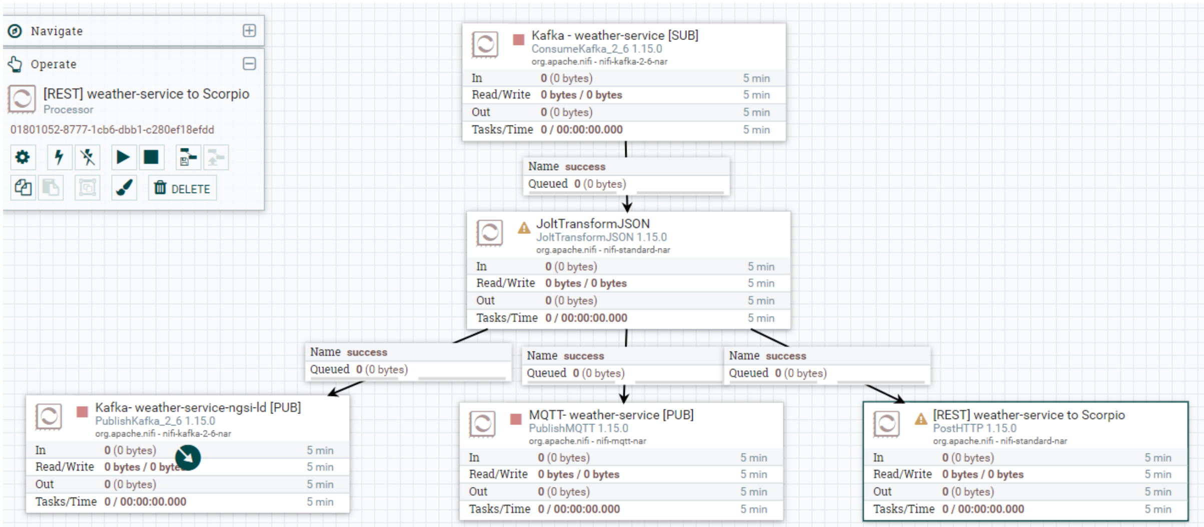


Figure 7 Example of a NiFi transformation flow – Evacuation Coordinator topic

Focusing on the transformation process itself, JOLT provides a set of diverse operations to use, including shift, default, remove, sort, etc. These operations are chained together to form the whole transformation [20]. The main transformations used are:

- **Shift**: changes the structure of the JSON using the same data from the input.
- **Default**: used to add fields or new objects.
- **Remove**: used to delete fields and objects. The field to be removed must be assigned to an empty string.
- **Sort**: sorts fields in the JSON in alphabetical order.

To accomplish the standardization using FIWARE and NGSI-LD several Kafka messages are transformed using JOLT. As an example of transformation, a message from the topic /weather-service is used. The original (raw) message is as shown in Figure 8.

Weather forecast information (raw)


```
{
    "Timestamp": "2020-01-01 17:00:00",
    "combDirectionDegrees": 5.68,
    "combPeriod": 3.83,
    "combSWHMeters": 0.74,
    "currentsDirectionDegrees": 45.0,
    "currentsSpeedKnots": 0.05832,
    "hum%": 64.2,
    "iceCover": 0.0,
    "mslhPa": 1031.24,
    "sea": true,
    "swellDirectionMeters": 17.89,
    "swellPeriod": 4.22,
    "swellSWHMeters": 0.64,
    "tempCelciousDegrees": 13.28,
    "visKm": 24.135,
    "wavesDirectionDegrees": 330.48,
    "wavesPeriod": 2.76,
    "wavesSWHMeters": 0.39,
    "windDirectionDegrees": 322.31,
    "windSpeedKnots": 11.780639999999998
}
```


Figure 8. Weather forecast information (raw)

To this JSON the following JOLT transformation is applied (Figure 9):

JOLT Specification for weather data

```
[{
        "operation": "default",
        "spec": {
            "type": "SeaConditions",
            "id": "urn:ngsi-ld:SeaCondition-Elyros",
            "dataProvider": "DANAOS API",
            "coordinates0": 0,
            "coordinates1": 1,
            "location_type": "Point",
            "context": [
                "https://smaertdatamodels.org/context.jsonld",
                "https://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.jsonld",
                "https://raw.githubusercontent.com/smart-data-models/  \
                  dataModel.Weather/master/context.jsonld"
            ]
        }
    },
    {
        "operation": "modify-overwrite-beta",
        "spec": {
            "str0": "=substring(@(1,Timestamp),0,10)",
            "str1": "=substring(@(1,Timestamp),11,19)",
            "dateObserved": "=concat(@(1,str0),'T',@(1,str1),,'Z')",
            "windSpeedKnots": "=divideAndRound(4,@(1,windSpeedKnots),1)"
        }
    },
    {
        "operation": "shift",
        "spec": {
            "id": "id",
            "type": "type",
            "dataProvider": "dataProvider",
            "dateObserved": "dateObserved",
            "coordinates0": "location.coordinates",
            "coordinates1": "location.coordinates",
            "location_type": "location.type",
            "tempCelciousDegrees": "surfaceTemperature",
            "wavesSWHMeters": "waveHeight",
            "wavesPeriod": "wavePeriod",
            "combDirectionDegrees": "combDirectionDegrees",
            "combPeriod": "combPeriod",
            "combSWHMeters": "combSWHMeters",
            "currentsDirectionDegrees": "currentsDirectionDegrees",
            "currentsSpeedKnots": "currentsSpeedKnots",
            "hum%": "hum%",
            "iceCover": "iceCover",
            "mslhPa": "mslhPa",
            "sea": "sea",
            "swellDirectionMeters": "swellDirectionMeters",
            "swellPeriod": "swellPeriod",
            "swellSWHMeters": "swellSWHMeters",
            "visKm": "visKm",
            "wavesDirectionDegrees": "wavesDirectionDegrees",
            "windDirectionDegrees": "windDirectionDegrees",
            "windSpeedKnots": "windSpeedKnots",
            "context": "\\@context"
        }
    }

```

Figure 9. JOLT Specification for weather data

Finally, the resulting (NGSI-LD compliant) JSON is represented in Figure 10:

Weather forecast information (NGSI-LD output)


```json
{
    "id": "urn:ngsi-ld:SeaCondition-Elyros",
    "type": "SeaConditions",
    "dataProvider": "DANAOS API",
    "dateObserved": "2020-01-01T17:00:00Z",
    "location": {
        "coordinates": [0, 1],
        "type": "Point"
    },
    "surfaceTemperature": 13.28,
    "waveHeight": 0.39,
    "wavePeriod": 2.76,
    "combDirectionDegrees": 5.68,
    "combPeriod": 3.83,
    "combSWHMeters": 0.74,
    "currentsDirectionDegrees": 45,
    "currentsSpeedKnots": 0.05832,
    "hum%": 64.2,
    "iceCover": 0,
    "mslhPa": 1031.24,
    "sea": true,
    "swellDirectionMeters": 17.89,
    "swellPeriod": 4.22,
    "swellSWHMeters": 0.64,
    "visKm": 24.135,
    "wavesDirectionDegrees": 330.48,
    "windDirectionDegrees": 322.31,
    "windSpeedKnots": 11.7806,
    "@context": ["https://smaertdatamodels.org/context.jsonld",
                 "https://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.jsonld",
                 "https://raw.githubusercontent.com/smart-data-models/  \
                     dataModel.Weather/master/context.jsonld"]
}
```


Figure 10. Weather forecast information (NGSI-LD output)

4 PALAEMON Information Model

In this section we compile all the context information elements that circulate across the PALAEMON Communications platform. As part of the exercise, we lean on two different tables per message type (or topic). The first table gathers the following elements:

- **Producer** → Who creates the event
- **Consumer** → Who receives the notification
- **NGSI-LD** → Flag that indicates whether the message is transformed to NGSI-LD or not (via the process we described in Section 3.2)
- **MQTT** → Check if data is forwarded as MQTT notification
- **Elasticsearch** → Name if the messages are saved as Elastic indices (YYYY.MM.DD means that new indices are created daily, so the name explicitly includes the date)

Additionally, the second table deals with the next elements:

- **Field** → Element identifier
- **Description** → One-liner definition
- **Format** → Data Format (e.g., string, integer, date, etc.)
- **Example** → Arbitrary data sample to get an idea of what we could expect

4.1 evacuation-coordinator

Messages from the topic /evacuation-coordinator describe the ship evacuation status.

Table 1 Evacuation-coordinator topic summary

Producer	Consumer	NGSI-LD	MQTT	ElasticSearch
Evacuation Coordinator	ALL	X	X	evacuation-coordinator-YYYY.MM.DD

Table 2 Evacuation-coordinator topic data content

Field	Description	Format/ Range	Example
originator	The originator of the command	str	"evacuation-coordinator"
evacuation-status	Ship evacuation status. Range: (0-5)	number	0
timestamp	The UTC timestamp to which evacuation status was set. Format: "YYY-mm-ddTHH:MM:SS"	string	"2022-03-03T22:26:13.393523"

4.2 evacuation-component-status

Messages from the topic /evacuation-component-status describe the components evacuation status at a specific time.

Table 3 Evacuation-component-status topic summary

Producer	Consumer	NGSI-LD	MQTT	ElasticSearch
ALL	Evacuation Coordinator			evacuation-component-status-YYYY.MM.DD

Table 4 Evacuation-coordinator-status topic data content

Field	Description	Format/Range	Example
timestamp	The timestamp of the command. Format: “YYYY-mm-ddTHH:MM:SS+TZ”	string	“2020-11-09T10:42:58.790517+00:00”
component_id	Name of the component	string	“PIMM”
operation_mode	Operation mode of the component. Range: (0-5)	number	1

4.3 resource-discovery-request

Initialization message request to know all active devices.

Table 5 Resource-discovery-request topic summary

Producer	Consumer	NGSI-LD	MQTT	ElasticSearch
Evacuation Coordinator	ALL	X	X	resource-discovery-request-YYYY.MM.DD

Table 6 Resource-discovery-request topic data content

Field	Description	Format/Range	Example
originator	The originator of the command	string	“evacuation-coordinator”
evacuation-status	Ship evacuation status. Range: (0-5)	number	0
timestamp	The timestamp of the command. Format: “YYYY-mm-ddTHH:MM:SS”	string	“2022-03-03T22:26:13.393523”

4.4 resource-discovery-response

Initialization message request to know all active devices, response of the component.

Table 7 Resource-discovery-response topic summary

Producer	Consumer	NGSI-LD	MQTT	ElasticSearch
ALL	Evacuation Coordinator			resource-discovery-response-YYYY.MM.DD

Table 8 Resource-discovery-response topic data content

Field	Description	Format/Range	Example
timestamp	The timestamp of the command. Format: "YYYY-mm-ddTHH:MM:SS+TZ"	string	"2020-11-09T10:42:58.790517+00:00"
component_id	Name of the component	string	"PIMM"
operation_mode	Operation mode of the component	number	1

4.5 heartbeat-request

Periodic check of components health.

Table 9 Heartbeat-request topic summary

Producer	Consumer	NGSI-LD	MQTT	ElasticSearch
Evacuation Coordinator	ALL	X	X	heartbeat-request-YYYY.MM.DD

Table 10 Heartbeat-request topic data content

Field	Description	Format/Range	Example
originator	The originator of the command	string	"evacuation-coordinator"
evacuation-status	Ship evacuation status. Range: (0-5)	number	0
timestamp	The timestamp of the command. Format: "YYYY-mm-ddTHH:MM:SS"	string	"2022-03-03T22:26:13.393523"

4.6 heartbeat-response

Periodic check of components health, response of the component.

Table 11 Heartbeat-response topic summary

Producer	Consumer	NGSI-LD	MQTT	ElasticSearch
ALL	Evacuation Coordinator			heartbeat-response-YYYY.MM.DD

Table 12 Heartbeat-response topic data content

Field	Description	Format/Range	Example
timestamp	The timestamp of the command. Format: “YYYY-mm-ddTHH:MM:SS”	string	"2020-11-09T10:42:58.790517+00:00"
component_id	Name of the component	string	“PIMM”
operation_mode	Operation mode of the component	number	1

4.7 smart-safety-system

Transcription of the Drag’n Drop events created on the SSS dashboard.

Table 13 Smart-safety-system topic summary

Producer	Consumer	NGSI-LD	MQTT	ElasticSearch
SSS	DFB			smart-safety-system-YYYY.MM.DD

Table 14 Smart-safety-system topic data content

Field	Description	Format/Range	Example
type		string	“Fall/Fire/Grounding”
timestamp	The timestamp of the command. Format: “YYYY-mm-ddTHH:MM:SS+TZ”	string	“2021-01-13T11:30:16.825950+00:00”
deck	Number of the deck	string	'5'
position_x	X Position of the event in m	number	13.72
position_y	Y Position of the event in m	number	5.47

4.8 ship-stability-toolkit

Ship motion prediction.

Table 15 Ship-stability-toolkit topic summary

Producer	Consumer	NGSI-LD	MQTT	ElasticSearch
SST	DFB			smart-camera-YYYY.MM.DD

Table 16 Ship-stability-toolkit topic data content

Field	Description	Format/Range	Example
state	Vessels initial state / heeling angle	number	0
timestamp	The timestamp of the command. Format: "YYYY-mm-ddTHH:MM:SS+TZ"	string	"2021-01-13T11:30:16.825950+00:00"
fn	Vessels speed over ground	number	0
t	Wave peak period	number	12
Hs	Significant wave height	number	8
L0	Wave length	number	225
beta	Wave direction	number	120
head	Vessels heading	number	180
surge	Expected vessels surge	number	3.087589
sway	Expected vessels sway	number	5.388004
heave	Expected vessels heave	number	7.445331
roll	Expected roll angle	number	32.617599
pitch	Expected pitch angle	number	7.419556
yaw	Expected yaw angel	number	3.533860

4.9 smart-camera

Table 17 Smart-camera topic summary

Producer	Consumer	NGSI-LD	MQTT	ElasticSearch
ADV	DFB	X	X	smart-camera-YYYY.MM.DD

Table 18 Smart-camera topic data content

Field	Description	Format/Range	Example
component_id	Id of the camera	string	"camera-01"

Field	Description	Format/Range	Example
people_count	Number of people counted by the camera	number	5
timestamp	The timestamp of the command. Format: “YYYY-mm-ddTHH:MM:SS”	string	“2022-03-16T13:05:14.948000”

4.10 smart-camera-alarm

Table 19 Smart-camera-alarm topic summary

Producer	Consumer	NGSI-LD	MQTT	ElasticSearch
ADV	DFB			smart-camera-alarm

Table 20 Smart-camera-alarm topic data content

Field	Description	Format/Range	Example
event-code	Code of the event	number	0
ID	Id of the camara	string	“camera-01”
timestamp	The UTC timestamp of the command. Format: “YYYY-mm-ddTHH:MM:SSUTC”	string)	"2020-12-21T16:02:46.793000UTC"

4.11 shm

Data gathered from the Ship Health Monitoring (SHM) sensors

Table 21 Shm topic summary

Producer	Consumer	NGSI-LD	MQTT	ElasticSearch
ESI	DFB			shm-YYYY.MM.DD

Table 22 Shm topic data content

Field	Description	Format/Range	Example
timestamp	The timestamp of the command. Format: “YYYY-mm-ddTHH:MM:SSZ”	string	“2021-01-21T12:40:03.176Z”
component_id	Unique device ID	string	“shm_01”

Field	Description	Format/Range	Example
accelerometer_x	Acceleration (X axis)	number	0.1
accelerometer_y	Acceleration (Y axis)	number	-0.0
accelerometer_z	Acceleration (Z axis)	number	-9.8
heave_velocity	Vessel's heave velocity	number	0.0
heave_acceleration	Vessel's heace acceleration	number	0.0
heave_ship_motion	Vessel's ship motion detected	number	0.0
yaw	Vessel's yaw observed	number	0.0
pitch	Vessel's pitch observed	number	0.0
roll	Vessel's roll observed	number	0.0

4.12 shm-alarm

Table 23 Shm-alarm topic summary

Producer	Consumer	NGSI-LD	MQTT	ElasticSearch
ESI	DFB	X	X	shm-alarm-YYYY.MM.DD

Table 24 Shm-alarm topic data content

Field	Description	Format/Range	Example
timestamp	The timestamp of the command. Format: "YYYY-mm-ddTHH:MM:SSZ"	timestamp	"2021-01-21T12:40:03.176Z"
type	Alarm type	string	"Grounding"
component_id	Unique device ID	string	"shm_01"
accelerometer_x	Acceleration (X axis)	number	0.1
accelerometer_y	Acceleration (Y axis)	number	-0.0
accelerometer_z	Acceleration (Z axis)	number	-9.8
heave_velocity	Vessel's heave velocity	number	0.0
heave_acceleration	Vessel's heace acceleration	number	0.0

Field	Description	Format/Range	Example
heave_ship_motion	Vessel's ship motion detected	number	0.0
yaw	Vessel's yaw observed	number	0.0
pitch	Vessel's pitch observed	number	0.0
roll	Vessel's roll observed	number	0.0

4.13 smart-bracelet-event-notification

Asynchronous event generated by the bracelets when e.g., a passenger falls down or pushes the alarm button.

Table 25 Smart-bracelet-event-notification topic summary

Producer	Consumer	NGSI-LD	MQTT	ElasticSearch
ADV	DFB		X	smart-bracelet-event-notification-YYYY.MM.DD

Table 26 Smart-bracelet-event-notification topic data content

Field	Description	Format/Range	Example
timestamp	The timestamp of the command. To be added by platform at reception.	string	" "
ID	Smart bracelet ID	number	"SB0000"
event_id	ID of the event	number	0

4.14 smart-bracelet-sensor-data

Periodic message generated by smart bracelets that mainly contain biometric data of the wearer.

Table 27 Smart-bracelet-sensor-data topic summary

Producer	Consumer	NGSI-LD	MQTT	ElasticSearch
ADV	DFB		X	smart-bracelet-sensor-data-YYYY.MM.DD

Table 28 Smart-bracelet-sensor-data topic data content

Field	Description	Format/Range	Example
timestamp	The timestamp of the command. To be added at reception	string	" "
ID	Smart bracelet ID	string	"SB0000"
"HR"	Heart rate (bpm)	number	79
"SpO2"	Oxygen Sat. (0-100%)	number	50
"Temp"	Temperature (ºC)	number	25
"Pitch"	Rotation of Pitch angle (X)	number	0.0
"Roll"	Rotation of Roll angle (Y)	number	0.0
"Angle"	Rotation of Z angle	number	0

4.15 smart-bracelet-pameas-evac-msg

Asynchronous message from event generated by PaMEAS.

Table 29 Smart-bracelet-pameas-evac-msg topic summary

Producer	Consumer	NGSI-LD	MQTT	ElasticSearch
UAEG	DFB		X	smart-bracelet-pameas-evac-msg-YYYY.MM.DD

Table 30 Smart-bracelet-pameas-evac-msg topic data content

Field	Description	Format/Range	Example
message_code	denotes a string code that is mapped by the smart-bracelet service to a specific notification displayable in the bracelet. Internally the PaMEAS messaging service maps this code to a text message that is semantically mapped to the aforementioned notification notification displayed by the bracelet.	string	“message_1”
timestamp	The timestamp of the command. Format: “YYYY-mm-ddTHH:MM:SS”	timestamp	“2021-01-21T12:40:03.176Z”
braceletId	the id of the bracelet that a specific user is in possession on and was registered during embarkation	string	"SB0000"
broadcast	If it is set to true, the message will be displayed in all bracelets regardless of their braceltId	boolean	True

4.16 decision-support-system

Set of recommendations generated by the DSS.

Table 31 Decision-support-system topic summary

Producer	Consumer	NGSI-LD	MQTT	ElasticSearch
KT	DFB			decision-support-system-YYYY.MM.DD

Table 32 Decision-support-system topic data content

Field	Description	Format/Range	Example
Incident	Type of incident	String	Fire-Explosion

Field	Description	Format/Range	Example
Actions	Requires response from the Master in order to proceed	String	YES/NO FIRE ON KITCHEN/FIRE ON ENGINE ROOM
Suggestions	The suggestions of the DSS to the Master, depending on the Actions	String	A visual inspection must be performed. All fuel tanks, ballast tanks and hulls must be counted.
Final Text	DSS questions to the Master	String	Which is the point of the fire or explosion that broke out?
Target	The target point of the suggestions	String	Master, Bridge Officer, Engine officer, etc.
Info	Indicates the danger of taking the suggestion	Number	Hull status: 0.45, Stability likelihood: 0.9, Ability to communicate : 0.7, Structural integrity: 0.3

4.17 mayday-message

When the evacuation process is triggered, various mayday signals are spread to e.g., Port Authorities, Search and Rescue Units, close-by vessels, etc.

Table 33 Mayday-message topic summary

Producer	Consumer	NGSI-LD	MQTT	ElasticSearch
ATOS	DFB	X		mayday-message-YYYY.MM.DD

Table 34 Mayday-message topic data content

Field	Description	Format/Range	Example
via	Outgoing channel through which the message is sent	string	"Inmarsat"
ship_name	Name of the ship	string	"Costa Concordia"
ship_position	Coordinates of ship position	array	[42.3717,10.92602]
timestamp	The timestamp of the command. Format: "YYYY-mm-ddTHH:MM:SS+TZ"	string	"2021-01-26T15:25:00.60661+01:00"
incident_type	Description of incident	string	"Contact - Breach - Blackout"

Field	Description	Format/Range	Example
assistance_required	In case any assistance is required	string	"None"
number_passengers	Number of passengers on board	number	4229
injured_people	Number of people that were injured	number	0
crew_on_board	Number of crew people on board	number	1023
weather_conditions	Summary of whether conditions	string	"ROUGH SEA - NE 4; WIND 7 KNOTS E-NE; VISIBILITY PARTLY CLOUDY"
damage_extent	Percentage of damage	string	"75%"

4.18 srap

Information coming from the Smart Risk Assessment Platform (SRAP).

Table 35 Srap topic summary

Field	Description	Format/Range	Example
messageId	ID random number	number	65454
timestamp	The timestamp of the command. Format: "YYYY-mm-ddTHH:MM:SS"	string	
sender	-	string	"SRAP"
Effectiveness of mitigation measures	The likelihood of the mitigation measures (i.e. response teams and vessel's mitigation systems) to be effective.	string	"Effective/ Not effective"
Passengers proximity to hazards	The probability of passengers to be close to danger due to the occurred incident.	string	"Low/ Medium/ High"
Status of Passive containment	The likelihood whether passive containment of the incident is effective given the status of the detectors and the containment doors.	string	"Effective/ Not effective"

Field	Description	Format/Range	Example
Spreading	The probability of incident to be spread to other parts/areas of the ship other than the incident location.	string	“Contained/ Not contained”
Structural Integrity	The probability of the vessel’s structural integrity to be compromised.	string	“Not compromised/ Compromised”
Stability	The probability of the vessel’s stability to be sufficient (i.e. the vessel remains in a stable, and safe position).	string	“Sufficient/ Not sufficient”
Hull Status	The probability of hull to be in a safe or unsafe state.	string	“Safe/ Unsafe”
Ability to communicate	The vessel’s ability to communicate, within the ship and with the outside world.	string	“Fully operational/ Degraded/ Not operational”
Critical system status	The probability of critical systems to be operational.	string	“Fully operational/ Degraded/ Not operational”
Vessel Status	The probability the vessel to be in a safe or unsafe state.	string	“Safe/ Unsafe”
Pax vulnerability onboard	Probability that indicates the vulnerability for the crew and passengers according to the current conditions.	string	“High/ Moderate/ Low”
Situation Assessment	The outcome of the situation assessment model (i.e. proposed action to sound the GA or not).	string	“Sound GA/ No sound GA”
Individual status	The likelihood of individual passengers requiring assistance for getting to the muster station and/or medical assistance.	string	"2549": 'Assistance required', "2552": 'Movement delayed', "2553": 'Free movement'
Escape routes	Indicates whether escape routes on the ship area are free of obstacles and congestion.	string	"Z1D1": 'Open', "Z2D1": 'Disrupted', "Z1D2": Closed,

Field	Description	Format/Range	Example
Group Performance	Assessment of passengers group performance (on the area under investigation)	string	"Z1D1": 'Low', "Z2D1": 'Medium', "Z1D2": 'High'
Risk of delay	The outcome of the mustering model, i.e. the probability of risk of delay for the areas of the vessel.	string	"Z1D1": 'Low' "Z2D1": 'Medium', "Z1D2": 'High'
Abandon Vessel (Preabandonment)	The outcome of the Preabandonment model (i.e., proposed action to Stay or Abandon ship).	string	“Stay/ Abandon”
Timeout	-	number	60

Table 36 Srap topic data content

Field	Description	Format/Range	Example
messageId	ID random number	number	65454
timestamp	The timestamp of the command. Format: “YYYY-mm-ddTHH:MM:SS”	string	
sender	-	string	“SRAP”
Effectiveness of mitigation measures	The likelihood of the mitigation measures (i.e. response teams and vessel’s mitigation systems) to be effective.	string	“Effective/ Not effective”
Passengers proximity to hazards	The probability of passengers to be close to danger due to the occurred incident.	string	“Low/ Medium/ High”
Status of Passive containment	The likelihood whether passive containment of the incident is effective given the status of the detectors and the containment doors.	string	“Effective/ Not effective”
Spreading	The probability of incident to be spread to other parts/areas of the ship other than the incident location.	string	“Contained/ Not contained”

Field	Description	Format/Range	Example
Structural Integrity	The probability of the vessel's structural integrity to be compromised.	string	"Not compromised/ Compromised"
Stability	The probability of the vessel's stability to be sufficient (i.e. the vessel remains in a stable, and safe position).	string	"Sufficient/ Not sufficient"
Hull Status	The probability of hull to be in a safe or unsafe state.	string	"Safe/ Unsafe"
Ability to communicate	The vessel's ability to communicate, within the ship and with the outside world.	string	"Fully operational/ Degraded/ Not operational"
Critical system status	The probability of critical systems to be operational.	string	"Fully operational/ Degraded/ Not operational"
Vessel Status	The probability the vessel to be in a safe or unsafe state.	string	"Safe/ Unsafe"
Pax vulnerability onboard	Probability that indicates the vulnerability for the crew and passengers according to the current conditions.	string	"High/ Moderate/ Low"
Situation Assessment	The outcome of the situation assessment model (i.e., proposed action to sound the GA or not).	string	"Sound GA/ No sound GA"
Risk of delay	The outcome of the mustering model, i.e. the probability of risk of delay for the areas of the vessel.	string	"Z1D1": 'Low', "Z2D1": 'Medium', "Z1D2": 'High'

4.19 pameas-location

Anonymized location data of a passenger.

Table 37 Pameas-location topic summary

Producer	Consumer	NGSI-LD	MQTT	ElasticSearch
UAEG	PaMEAS, SRAP			N/A

Table 38 Pameas-location topic data content

Field	Description	Format/ Range	Example
hashedMac Address	the hash mac address of the user's device	string	dabf99b54f86bcc4f5949a8bcd7e29961081b36d42cb0905e1e52a131652adf9
geofence	json object containing the details of the geofence of the user	JSON	{"gfEvent":"2132121", "gfId":"1.1", "gfName":"geofence1.1", "macAddress":"58:37:8B:DE:42:F8", "isAssociated":"false", "dwellTime":"1600807918", "hashedMacAddress":"dabf99b54f86bcc4f5949a8bcd7e29961081b36d42cb0905e1e52a131652adf9", "timestamp":"1600807918"}
location	json object containing the specific coordinates of the location of the user	JSON	{"xLocation":"12321", "yLocation":"12312312", "errorLevel":"0", "isAssociated":"false", "campusId":"deckA", "buildingId":"shipA", "floorId":"floor0", "hashedMacAddress":"dabf99b54f86bcc4f5949a8bcd7e29961081b36d42cb0905e1e52a131652adf9", "geofenceId":"1.1", "geofenceNames": ["geofence1.1"], "timestamp":"1600807918"}

4.20 pameas-notification

Table 39 Pameas-notification topic summary

Producer	Consumer	NGSI-LD	MQTT	ElasticSearch
UAEG	DFB			evacuation-coordinator-YYYY.MM.DD

Table 40 Pameas-notification topic data content

Field	Description	Format /Range	Example
type	notification type	string	“PASSENGER_ISSUE”
id	notification id	string	
incident	passenger related issue/incident	json	{“passenger_name”:”testName”, “passenger_surname”:”testSurname”, “health_issues”:”none”, “mobility_issues”:”none”, “pregnancy issues”:” complicated pregnancy”, “xloc”:”1234”, “yloc”:”15421”, “geofence”:” geofence1.1”, “timestamp”:”1600807918”,}
assignedCrewMembers	list of assigned crew members to incident	array of json	[{“id”:”18”, “name”:”CrewtestName”, “surname”:”CrewTestSurname”, “emergencyRole”:”first_aid_unit”, “hashedMacAddress”:”dabf99b54f86bcc4f5949a8bcd7e29961081b36d42cb0905e1e52a131652adf9”}]
status	status of the incident	string	“OPEN”

4.21 pameas-person

Record of person (passenger or crew member) registered to the system. As a matter of fact, the outgoing object is split into 5 different subcategories: personalInfo, networkInfo, locationInfo.

Table 41 Pameas-person topic summary

Producer	Consumer	NGSI-LD	MQTT	ElasticSearch
UAEG	DFB			pameas-person -YYYY.MM.DD

Table 42 PersonalInfo field from topic Pameas-person

Field	Description	Format/Range	Example
name	First name	string	“Nikos”
surname	Last name	string	“Triantafyllou”
dateOfBirth	Date of birth formatted as yyyy-MM-dd	string	“05-10-1983”
gender	Gender	string	“Male”
personalID		string	“123”
ticketNumber	Ticket number	string	“5567432”
ticketInfo	Information about the passenger purchased ticket	array	(Table 43)
crew		string	“true”
embarkationPort	Port of embarkation	string	“PIREAUS”
disembarkationPort	Port of disembarkation	string	“CHIOS”
email	Users email address	string	“triantafyllou.ni@gmail.com”
postalAddress		string	”TEST ADDRESS 15771 GR”
emergencyContact	Phone number of user’s emergency contact	string	”6943808730”
countryOfResidence	Code of the country of residence of the user, e.g., GR	string	“GR”
medicalCondition		string	“none”
mobilityIssues		string	“none”
pregnancyData		string	“none”
role		string	“seaman”
emergencyDuty		string	“first_response_unit”
preferredLanguage	List of languages the		

Field	Description	Format/Range	Example
	person is fluent in		
inPosition		string	" "
assignmentStatus		string	" "
dutyScheduleList			
assignedMusteringStation		string	" "
assignedPath		string	" "

Table 43. Pameas-person Ticket information

Field	Description	Format/Range	Example
name	First name	string	"Nikos"
surname	Last name	string	"Triantafyllou"
gender	Gender	string	"Female"
ticketNumber	Ticket number	string	"5567432"

Field	Description	Format/Range	Example
name	First name	string	"Nikos"
surname	Last name	string	"Triantafyllou"
dateOfBirth	Date of birth formatted as yyyy-MM-dd	string	"05-10-1983"
gender	Gender	string	"Male"
personalID		string	"123"
ticketNumber	Ticket number	string	"5567432"
ticketInfo	Information about the passenger purchased ticket	array	(Table 43)
crew		string	"true"
embarkationPort	Port of embarkation	string	"PIREAUS"
disembarkationPort	Port of disembarkation	string	"CHIOS"

Field	Description	Format/Range	Example
email	Users email address	string	“triantafyllou.ni@gmail.com”
postalAddress		string	”TEST ADDRESS 15771 GR”
emergencyContact	Phone number of user’s emergency contact	string	”6943808730”
countryOfResidence	Code of the country of residence of the user, e.g. GR	string	“GR”
medicalCondition		string	“none”
mobilityIssues		string	“none”
pregnancyData		string	“none”
role		string	“seaman”
emergencyDuty		string	“first_response_unit”
preferredLanguage	List of languages the person is fluent in		
inPosition		string	“ “
assignmentStatus		string	“ “
dutyScheduleList			
assignedMusteringStation		string	“ ”
assignedPath		string	“ ”

4.22 weather-service

Weather forecast generated every 3 hours.

Table 44 Weather-service topic summary

Producer	Consumer	NGSI-LD	MQTT	ElasticSearch
DANAOS	Weather API	X		weather-api-YYYY.MM.DD

Table 45 Weather-service topic data content

Field	Description	Format/Range	Example
Timestamp	The UTC timestamp to which the weather data were collected. Format: 'YYYY-MM-DDTHH:MM:SS'	string	"2019-09-10 00:00:00"
combDirectionDegrees	Combined Direction for sea state in degrees: A sea state combination factor including direction of wind, wave, swell	number	173.28
combPeriod	Combined period for sea state in seconds: A sea state combination factor including frequency (period) of wind, wave, swell	number	3.9
combSWHMeters	Combined significant wave height in meters. The output is calculated by means of the following formula [21]	number	1.07
currentsDirectionDegrees	The currents direction of the current (0-359, 0=N, 90=E, etc.)	number	45.0
currentsSpeedKnots	The currents speed in knots	number	0.05832
hum%	The percentage of humidity	number	68.5
iceCover	The ice coverage (from 0.0 = no ice to 1.0 = full ice coverage)	number	0.0
lat	The latitude of the geographical position to which the weather data refers. In decimal degree format.	number	54
lon	The longitude of the geographical position to which the weather data refers. In decimal degree format.	number	21
mslhPa	Mean sea level pressure in hectopascal (hPa)	number	1016.26
sea	Boolean, true if position is in sea, false otherwise	boolean	true
swellDirectionMeters	The swell direction	number	117.32
swellPeriod	The swell period	number	4.17
swellSWHMeters	Significant swell height in meters	number	0.13

Field	Description	Format/Range	Example
tempCelciousDegrees	The temperature in Celsius degrees	number	17.5
visKm		number	24.135
wavesDirectionDegrees	The waves direction	number	173.89
wavesPeriod	The waves period	number	3.9
wavesSWHMeters	Significant waves height in meters	number	1.06
windDirectionDegrees	The wind direction (0-359, 0=N, 90=E, etc.)	number	204.63
windSpeedKnots	The wind speed in knots	number	22.20048

4.23 ais-position

Position generated every 30 seconds.

Table 46 Ais-position topic summary

Producer	Consumer	NGSI-LD	MQTT	ElasticSearch
VDES_GW		X		ais-position-YYYY.MM.DD

Table 47 Ais-position topic data content

Field	Description	Format/Range	Example
user_id	Unique identifier such as MMSI number. Nine digit number	Number	123456789

Field	Description	Format/Range	Example
nav_status	Navigational Status. Range: (0-15) 0 = under way using engine 1 = at anchor 2 = not under command 3 = restricted manoeuvrability 4 = constrained by her draught 5 = moored 6 = aground 7 = engaged in fishing 8 = under way sailing, 9 = reserved for future amendment of navigational status for ships carrying DG, HS, or MP, or IMO hazard or pollutant category C, high speed craft (HSC) 10 = reserved for future amendment of navigational status for ships carrying dangerous goods (DG), harmful substances (HS) or marine pollutants (MP), or IMO hazard or pollutant category A, wing in ground (WIG) 11 = power-driven vessel towing astern (regional use), 12 = power-driven vessel pushing ahead or towing alongside (regional use); 13 = reserved for future use, 14 = AIS-SART (active), MOB-AIS, EPIRB-AIS 15 = undefined (default)	Number	1

Field	Description	Format/Range	Example
nav_status	Rate of turn ROTais as taken from the AIS data. 0 to +126 = turning right at up to 708° per min or higher 0 to –126 = turning left at up to 708° per min or higher Values between 0 and 708° per min coded by ROTAIS = 4.733 SQRT(ROTsensor) degrees per min where ROTsensor is the Rate of Turn as input by an external Rate of Turn Indicator (TI). ROTAIS is rounded to the nearest integer value. +127 = turning right at more than 5° per 30 s (No TI available) –127 = turning left at more than 5° per 30 s (No TI available) 128 (80 hex) indicates no turn information available (default) ROT data should not be derived from COG information.	Number	100
sog_available	True if Speed Over Ground (SOG) is available, false otherwise	Boolean	True
sog	Speed over ground in knots. This field is set to 0 and shall be ignored if sog available is false	Number	10.5
pos_accuracy	The position accuracy (PA) flag:1 = high (≤ 10 m) 0 = low (>10 m, default	Number	1
longitude	Longitude in decimal degrees format and range (-180 to 180)	Number	9.8291289
latitude	Latitude in decimal degrees format and range (-90 to 90)	Number	44.104060
cog_available	True if Course Over Ground (COG) is available, false otherwise	Boolean	True

Field	Description	Format/Range	Example
cog	Course Over Ground, if available. Possible values from 0.0 to 359.0. This field is set to 0 and shall be ignored if cog_available is false	Number	90.5
th_available	True if True Heading is available, false otherwise	Boolean	True
true_heading	True Heading value. Possible values from 0 to 359. This field is set to 0 and shall be ignored if th_available is false	Number	90
timestamp	UTC Timestamp when the report was received from the VDES Ship radio. Format: 'YYYY-MM-DDTHH:MM:SS.mmm'	String	2021-05-12T16:10:23.000
manoeuvre_ind	Special Maneuverer Indicator. Range: (0-2) 0 = not available = default 1 = not engaged in special maneuver 2 = engaged in special maneuver (i.e., regional passing arrangement on Inland Waterway)	Number	1

5 Evacuation Operation Modes

Till this point, we have stated in numerous deliverables and reports that many (most) components do have a tight dependency with the ship evacuation status. Namely, they alter their operation mode according to the current evacuation phase. This behaviour spans from a normal situation, where the voyage elapses without any awkward situation, to the (in the worst case) moment in which the ship is abandoned. This chain of events is illustrated in Figure 11 (upper part of the picture), where we can observe the 6 phases.

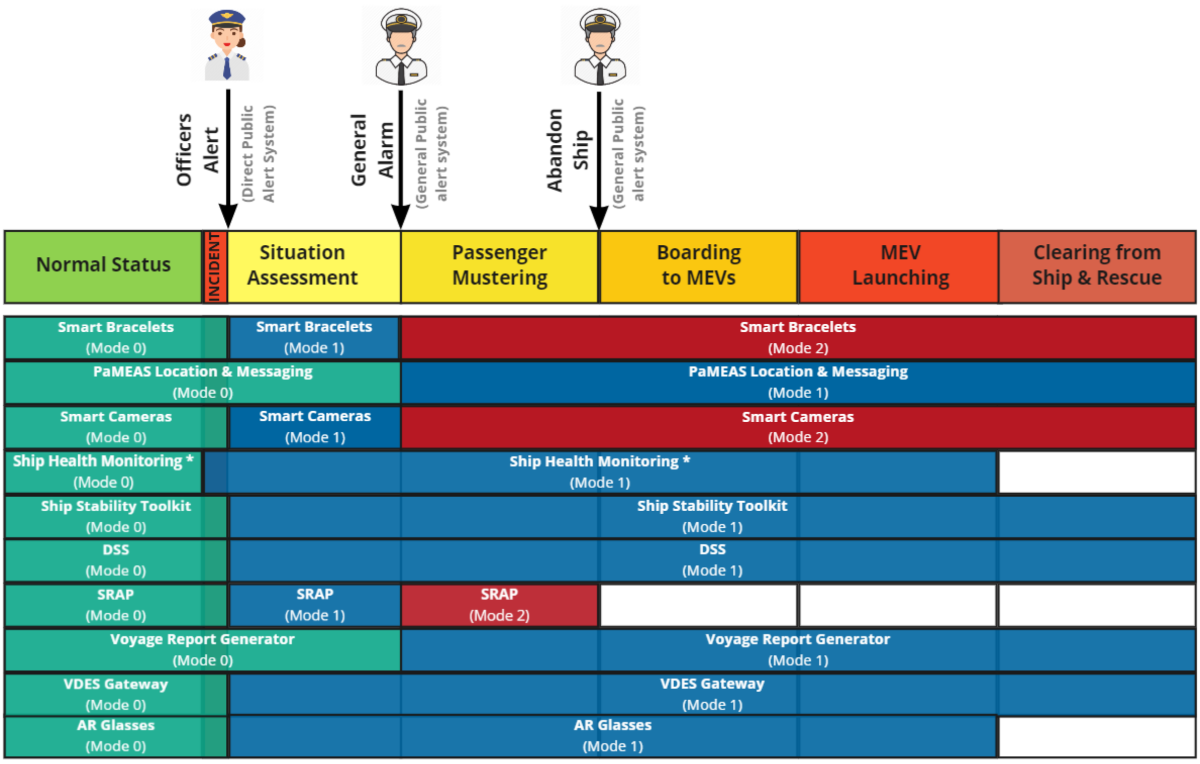


Figure 11. Components operation modes vs. Evacuation Status Phase.

Below we delve into the particularities of the operation mode of each of the components that exhibit a dynamic behavior throughout the evacuation process. Nonetheless, it is worth mentioning that there are other modules whose operation does not vary as long as the evacuation status does (e.g., Evacuation Coordinator, Safety Management System, Smart Safety System), hence they have been left out of this section.

5.1 Smart bracelets

Evacuation status	Mode	Description
0 – Normal	0	Alarm (i.e., triggered by physical button) is switched off. Data is sent every 10 minutes.
1- Situation Assessment	1	Data is sent every minute.
2- Passenger Mustering	2	Data is sent every 10 seconds. Alarms on.
3- Boarding to MEVs		

4- MEV Launching		
5 Clearing from Ship & Rescue		

5.2 PaMEAS

Evacuation status	Mode	Description
0 – Normal	0	Passenger and crew tracing is turned off (no data are generated)
1- Situation Assessment		
2- Passenger Mustering	1	Passenger and crew tracing is enabled together with passenger issue/incident detection and resolution. Data is generated every 10 seconds
3- Boarding to MEVs		
4- MEV Launching		
5 Clearing from Ship & Rescue		

5.3 Smart Cameras

Evacuation status	Mode	Description
0 – Normal	0	Asynchronous data transmission (e.g., people counter)
1- Situation Assessment	1	Async data transmittion (mode 0) + alarm transmission (i.e., stampede, trapped people) + video recording (for the VDR)
2- Passenger Mustering		
3- Boarding to MEVs		
4- MEV Launching		
5 Clearing from Ship & Rescue		

5.4 Ship Health Monitoring (SHM)

Evacuation status	Mode	Description
0 – Normal	0	Status report every 10 min
1- Situation Assessment	1	Alarms activated plus streaming of all data
2- Passenger Mustering		
3- Boarding to MEVs		
4- MEV Launching		

5 Clearing from Ship & Rescue		

5.5 Ship Stability Toolkit (SST)

Evacuation status	Mode	Description
0 – Normal	0	System on standby
1- Situation Assessment	1	System active, data is sent every minute, or when the input parameters have changed.
2- Passenger Mustering		
3- Boarding to MEVs		
4- MEV Launching		
5 Clearing from Ship & Rescue		

5.6 Decision Support System (DSS)

Evacuation status	Mode	Description
0 – Normal	0	DSS is on Standby
1- Situation Assessment	1	DSS is triggered. The DSS receives the type of incident, and it starts suggesting actions to the Master/Crew members.
2- Passenger Mustering		
3- Boarding to MEVs		
4- MEV Launching		
5 Clearing from Ship & Rescue		

5.7 Smart Risk Assessment Platform (SRAP)

Evacuation status	Operation Mode	Description
0 – Normal	0	Sending heartbeat message to /evacuation-coordinator when requested (Stand by)
1- Situation Assessment	1	Module is activated. The situation assessment model operates and provides output based on the data input rate
2- Passenger Mustering	2	The Mustering assessment and Preabandonment models are activated.

		The output rate is based on the data input rate.
3- Boarding to MEVs	N/A	SRAP does not have any explicit role during these phases of the evacuation
4- MEV Launching		
5 Clearing from Ship & Rescue		

5.8 VHF Data Exchange System (VDES) Gateway

Evacuation status	Mode	Description
0 – Normal	0	The component retrieves and forwards data from the Weather (forecast)Service and what VDES TRX overhears from the Automatic Identification System (AIS)
1- Situation Assessment		
2- Passenger Mustering	1	Beside weather forecast and AIS, the VDES gateway generates mayday signals
3- Boarding to MEVs		
4- MEV Launching		
5 Clearing from Ship & Rescue		

5.9 Voyage Report Generator (VRG)

Evacuation status	Mode	Description
0 – Normal	0	Report is generated at the end of the trip, i.e., non-incident report.
1- Situation Assessment		
2- Passenger Mustering	1	When the initiation of the evacuation is triggered, the VRG starts gathering the information from external sources (e.g., smart cameras, etc.). As a matter of fact, it is worth highlighting that the (ICT) systems (e.g., Data Fusion Bus databases, smart cameras' processing nodes video clip folders, etc.) will be flushed after every voyage. All data will be uploaded to a cloud-based repository and synchronized with DANAOS' Safety Management System.
3- Boarding to MEVs		
4- MEV Launching		
5 Clearing from Ship & Rescue		

6 Conclusions

In a nutshell, the main purpose of this deliverable is to nail down a common information model to be used across all the components that shape the PALAEMON platform. The document starts from the main outcomes of the previous deliverable in WP7, which covered the so-called Communications Platform, where we described how and where the different software components are executed, over which hardware (i.e., server or virtual machine in the cloud), and how the information is protected.

Starting from the assumption that the reader has been acquainted with the previous information, in this document we pay attention to the information that circulates across the various modules. The main objective of this task is to settle down a homogenous way to represent the data, hence all blocks could transparently understand each other. To achieve this goal, we have split the breakthrough in two main parts. On the one hand, we have studied and introduced the concept of FIWARE initiative, an open-source framework that does not only describes a standardized format for the data models (i.e., NGSI-LD), but also brings a core component that support a REST-based publish-subscribe communication (i.e., Scorpio Broker). In addition, we have delved into one of the components of PALAEMON's Data Fusion Bus, namely Apache Kafka, which we use to perform real-time on-the-fly transformation over the raw data generated by the data sources. This means that, introducing any format as input (e.g., the information received from the Weather Service), we can configure a "processor" that tampers the object and yields an output compatible with NGSI-LD smart data models.

With this exercise we bring about a double outcome. On the one hand we stick to a well-known and standardized information model, aiming to bridge with similar solutions that may deal with akin datasets; on the other hand, the utilization of Orion Broker also opens the door to a new branch of the data, which could be used by devices that cannot use the primary channel (i.e., Kafka + SSL). In this case, Augmented Reality glasses fall into this category.

Finally, it is worth stating the direct connection this deliverable has with the ones we still have to deliver in the scope of WP7.

- **Deliverable D7.5 – System Integration & Final PALAEMON Prototype** (DEM, CO). In this deliverable we continue with the description of the PALAEMON Communications Platform and focus on the specification of the three main releases or software prototypes we have carried out.
- **Deliverable D7.6 – Test cases and overall system testing results** (DEM, CO). Before the execution of the evacuation trials, this document gathers the final lessons learnt and assessment results, thus serving as connection with WP8's activities.
- **Deliverable D7.7 – Public release WP7** (R, PU). This report basically compiles the main breakthrough and outcomes achieved throughout this WP and open it to the main public.

7 References

[1] “FIWARE homepage.” https://www.fiware.org/.

[2] “PALAEMON Deliverable D2.6 - PALAEMON Architecture v1,” 2020.

[3] “PALAEMON Deliverable D2.7 - PALAEMON Architecture v2,” 2021.

[4] “PALAEMON Deliverable D7.1 - PALAEMON Communications Platform,” 2022.

[5] “PALAEMON Deliverable D6.2 - PALAEMON Data Fusion Middleware,” 2022.

[6] “FIWARE Smart Data Models.” https://www.fiware.org/smart-data-models/.

[7] “FIWARE For Digital Twins (Position Paper),” 2021. [Online]. Available: https://www.fiware.org/wp-content/uploads/FF_PositionPaper_FIWARE4DigitalTwins.pdf.

[8] “FIWARE Catalogue,” [Online]. Available: https://www.fiware.org/catalogue/.

[9] “FIWARE Organizations Directory.” https://www.fiware.org/community/members/organizations-directory/.

[10] “ETSI- European Telecommunications Standardization Institute.” https://www.etsi.org/.

[11] Cim, “GS CIM 009 - V1.2.2 - Context Information Management (CIM); NGSI-LD API,” 2020, Accessed: Nov. 23, 2021. [Online]. Available: https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx.

[12] “Scorpio Broker — ScorpioBroker documentation.” https://scorpio.readthedocs.io/en/latest/ (accessed May 26, 2021).

[13] “Scorpio Broker GitHub main Page.” https://github.com/ScorpioBroker/ScorpioBroker.

[14] “Apache Kafka.” https://kafka.apache.org/ (accessed Jun. 02, 2021).

[15] “Apache NiFi.” https://nifi.apache.org/ (accessed Jun. 02, 2021).

[16] “PALAEMON Deliverable D7.2 - Encryption and Authentication Mechanisms,” 2022.

[17] “MQTT - The Standard for IoT Messaging.” https://mqtt.org/.

[18] “Eclipse Mosquitto - An open source MQTT broker.” https://mosquitto.org/.

[19] “JoltTransformJSON.” https://nifi.apache.org/docs/nifi-docs/components/org.apache.nifi/nifi-standard-nar/1.5.0/org.apache.nifi.processors.standard.JoltTransformJSON/.

[20] “Jolt Specification GitHub Repository.” https://github.com/bazaarvoice/jolt.

[21] “How to calculate the combined sea and swell height.” http://www.bom.gov.au/marine/about/combinedseaswell.shtml.