# XR4DRAMA

Extended Reality For DisasteR management And Media planning

H2020-952133

# D5.1

# Roadmap towards the implementation of the XR4DRAMA platform

| | |
|---|---|
| **Dissemination level:** | Public |
| **Contractual date of delivery:** | Month 4, 28 Feb 2021 |
| **Actual date of delivery:** | Month 5, 02 Mar 2021 |
| **Work package:** | WP5 Platform Development |
| **Task:** | T5.1 Technical requirements and system architecture |
| **Type:** | Report |
| **Approval Status:** | Final version |
| **Version:** | 1.0 |
| **Number of pages:** | 40 |
| **Filename:** | D5.1_XR4DRAMA_RoadMaptowardsImplementation_20210302_v1.0.pdf |

**Abstract**

D5.1 provides an overview on the components of the XR4DRAMA platform. The deliverable provides the functionalities of the modules that will be developed within the XR4DRAMA and the platform as a whole. Specifications for the technical infrastructure will be provided. The deliverable finally provides the time-plan for the development of the platform and the integration timeline.

co-funded by the European Union

# History

| Version | Date | Reason | Revised by |
|---|---|---|---|
| V0.1 | 15.01.2021 | Table of contents | Yash Shekhawat |
| V0.3 | 05.02.2021 | Inputs from partners | All technical partners |
| V0.5 | 15.02.2021 | Consolidated version with inputs | All technical partners |
| V0.9 | 25.02.2021 | Submitted for review to U2M | Yash Shekhawat |
| V1.0 | 02.03.2021 | Changes based on the internal review by Christos Stentoumis | Yash Shekhawat |

# Author list


| Organization | Name | Contact Information |
|---|---|---|
| NURO | Yash Shekhawat | Yash.shekhawat@nurogames.com |
| NURO | Sebastian Freitag | Sebastian.Freitag@nurogames.com |
| CERTH | Spyridon Symeonidis | spyridons@iti.gr |
| CERTH | Sotiris Diplaris | diplaris@iti.gr |
| U2M | Christos Stentoumis | christos@up2metric.com |
| STX | Maria Pacelli | m.pacelli@smartex.it |
| UPF | Jens Grivolla | jens.grivolla@upf.edu |

## Executive Summary

The deliverable is connected to the setup of the project and the platform development.

First a high level overview of the entire XR4DRAMA is provided. Then a high-level view of the envisioned architecture for the XR4DRAMA platform is presented, illustrating the conceptual architecture foreseen in the DoA document. A description of the modules that will be developed within XR4DRAMA follows. The functionality and input/output specifications, per system component are described. In addition, for each module of the system, to support the aforementioned functionalities development milestones, are provided.

To enable the integration of XR4DRAMA services, the available tools, practices and integration patterns are described and the contextualization of XR4DRAMA system is depicted. The XR4DRAMA platform concept is introduced in a manner that facilitates an early abstraction and classification of XR4DRAMA components. A development plan and integration model are included.

A global project timeline, describing the platform's scheduled iterations and the levels of functionality at the project milestones is presented.

Finally a short overview on the use cases is provided.

This roadmap represents a common accord and a technical agreement between the partners responsible for developing and deploying services and components in the XR4DRAMA platform. It defines the context for the implementation of the platform. Addresses all the main concerns related to this type of development and should serve as a guide for developing and integrating any component.

# Abbreviations and Acronyms

| | |
|---|---|
| **API** | Application Programming interface |
| **AR** | Augmented Reality |
| **DoA** | Description of Action |
| **ECG** | Electrocardiogram |
| **GIS** | Geographic Information System |
| **GPU** | Graphical Processing Unit |
| **JSON** | JavaScript Object Notation |
| **KB** | Knowledge Base |
| **RDF** | Resource Description Framework |
| **REST** | Representational State Transfer |
| **SoA/SoTA** | State of the Art |
| **URL** | Uniform Resource Locator |
| **UTF** | Unicode Transformation Format |
| **VR** | Virtual Reality |
| **XR** | Extended Reality |

# Table of Figures



# Table of Tables

# Table of Contents

# 1 INTRODUCTION

The document provides a roadmap and the planning of the implementation and integration of the XR4DRAMA platform. The objective of the deliverable is to specify in detail: (i) the progressively increasing functionality (in particular at the time points of the milestones) of the XR4DRAMA modules/activities as well as of the platform as a whole, (ii) the integration of the modules to ensure their functionality; and (iii) the technical/research methodologies that will be taken into consideration.

Section 2 of the deliverable provides the overview of the Architecture as it was foreseen in the description of action of the project. This gives a good idea into how to start the process on implementation. The section also contains an overview on the data exchange formats envisioned currently inside the project as well as how the communication will happen. The section also shows how each component is categorized inside the system.

Section 3, describes the functional descriptions of the modules. The modules are first divided into the roles they play inside the platform. The current roles for each component includes: Data Acquisition Modules, Data Analysis and Processing Modules, Data Linking and Understanding Modules, Data Storage Modules, and Data Visualisation Modules. Each particular component is described as well as input/outputs for each of those, the dependencies to other tasks and components are described as well as the timeline of development for each of the module is showcased.

Section 4 of the deliverable provides the information about the platform integration plan. The section defines the development cycles and the prototypes that will be created in the project to manage the implementation. The section also provides an overview on the integration timelines for all the components in the project

The final section provides a brief overview about the use case demonstration in the project.

# 2 ARCHITECTURE FORESEEN IN THE DOA

In this section we will present the architecture foreseen in the DoA and in the current state based on user requirements. The draft of the architecture was conceived during the creation of project proposal aiming to leverage the technical knowledge of the partners in a single platform.

## 2.1 Description

The main objectives of the project and the technical knowledge of the partners in the field is to define the set of components, services and activities. The architecture envisioned takes into account the capabilities of the partner and future innovations on a conceptual and technological level. The architecture also envisions scalability and modularity for the system as well as user requirements.

The entire components of the system can be categorized in the following categories:

1. Data Acquisition Modules (Data Collection Modules in Figure 1)
2. Data Analysis and processing Modules (Analysis and Reconstruction; and Linking data in Figure 1)
3. Data Storage Modules
4. Data Visualisation Modules (XR Tools in Figure 1)

The following diagram shows the categorization of the system components into the categories:

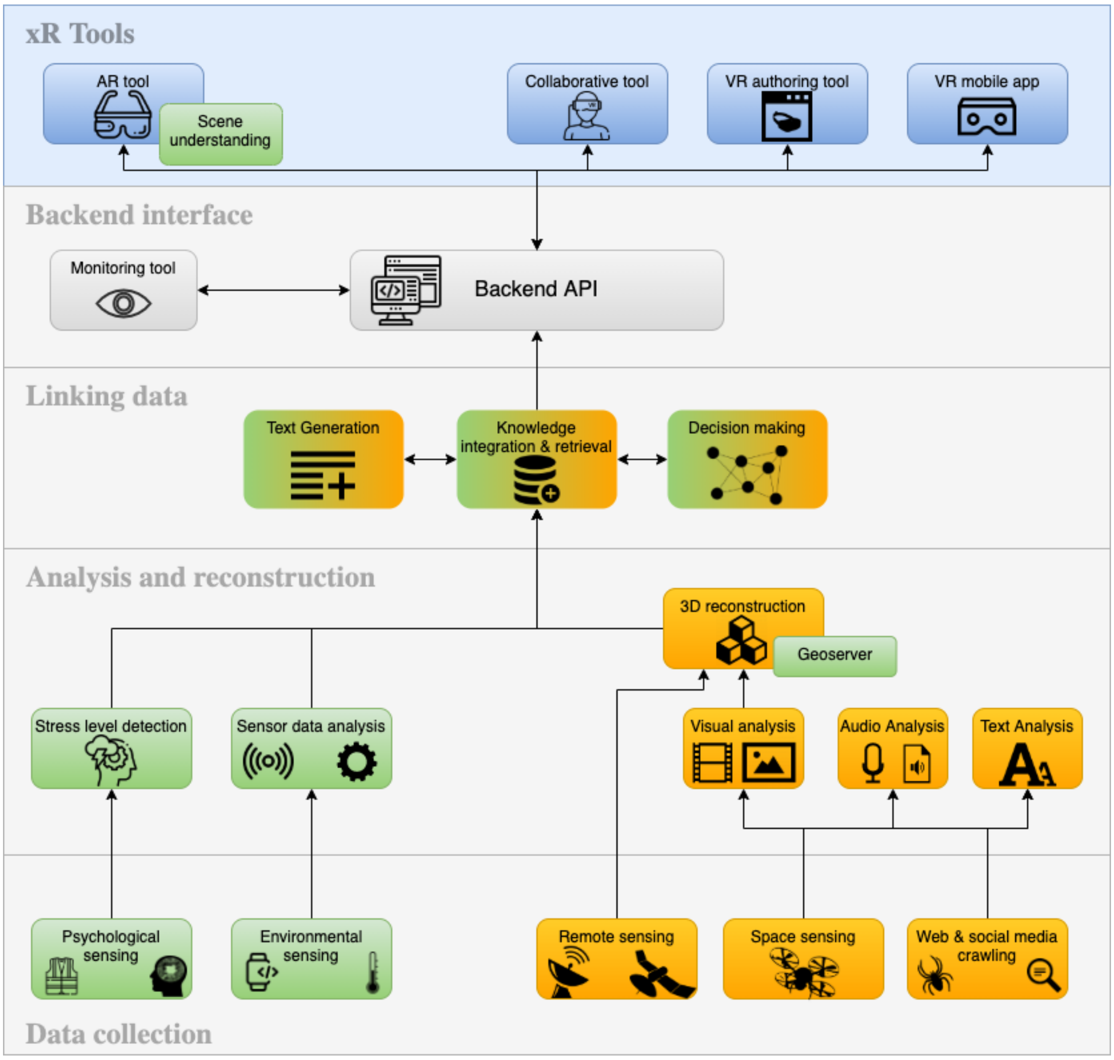


Figure 1 Components categorization

The platforms flow is started by the ingestion of content/data from the data collection modules which includes various sensors (physiological, environmental, remote sensors, space sensors and web & social media crawling). The data is ingested based on the user's requirements about the project they create in the platform.

The ingested data is then analysed and processed by the various modules which help in sorting of data in relevant and non-relevant data. The output of these modules makes sure that the platform does not contain irrelevant data and is easily scalable and fast. The components included are visual analysis, audio analysis, text analysis, 3D Reconstruction with Geoserver included, stress level detection and sensor data analysis.

Following the analysis of data from various components the data is linked and stored to the knowledge integration and retrieval module. The linking of the data also allows the decision making by the platform and the generation of text to be shown to the users for better situational awareness. The flow also allows in creation of scene understanding (Running on the XR app.

The final flow of the data ends-up to the data visualisation components; the various data visualisations help the specific users of each of those to view data and manipulate the data in the system, but also to interact with them in real-time and establish a bidirectional communication between the remote "control room" and the in-situ "1st responders". The data visualisation modules are connected using a middleware API which helps in abstracting the frontend components to the backend components of the system.

## 2.2 Technology Stack

A controlled set of technologies will be employed to implement the XR4DRAMA components, having as a goal to provide a robust, maintainable and manageable system. The technology stack that will be employed has to be defined by taking into consideration the compatibility between the system components. Although, the XR4DRAMA platform includes independent components there are dependencies for a subset of modules which impose the use of a certain technology. The technology stack could determine the success of a project and there are a couple of aspects that have to be taken into account. Sometimes, the selection of the technology stack is determined by the type of a module that you aim to build. For example, it is well-known that Python is an effective and powerful language for Machine Learning and Big Data services. In the following sections a list of the utilized programming languages, databases, technologies and exchange formats are presented

### 2.2.1 Data Exchange Formats

The data exchange formats set a standard throughout the entire platform to easily manage the information flow between different components of the system. For this we will be using the following the formats:

- REST APIs for communication with web services. REST determines how the API looks like. It stands for "Representational State Transfer". It is a set of rules that developers follow when they create their API. One of these rules states that you should be able to get a piece of data (called a resource) when you link to a specific URL. JSON (JavaScript Object Notation) will be used for formatting of data/messages inside the system.
- OBJ / FBX / PLY for 3D reconstruction meshes. These formats are natively supported by all of the major 3D modelling engines and renderers.
- **UTF-8** [1] encoding will be used and enforced throughout the platform to ensure correct and consistent handling of multilingual content.

# 3 FUNCTIONAL DESCRIPTION OF THE MODULES

## 3.1 Data Acquisition Modules

These modules help in acquiring the data for various projects that the users can start in the platform.

### 3.1.1 Data Acquisition from Web and Social Media

The data collection module implements the requirements of Task 2.1 for bringing and providing data to the XR4DRAMA system. In particular, it carries out the task of collecting and storing multimedia content that exists online. The target resources are websites, social media and open repositories that provide access using an application programming interface (API). Several different mechanisms are applied in order to accommodate for different types of sources.

The data collection process is performed in 4 phases. For each phase, different techniques may be applied according to the use case:

- *Phase 1 - Requirements*: In this phase, the input of the module is defined. The data collection needs can be expressed in the form of *URL addresses* or keywords forming *textual queries*.
- *Phase 2 - Discovery*: In the situations where the exact web resources to be gathered are not known, discovery has to be conducted. Detection of relevant web resources can be made in 3 ways: a) using *web crawling*, when the input is an entry URL, b) using *search*, when the input is a textual query, c) *streaming* in real-time, instead of search, where an existing API (e.g. Twitter) provides such capability. This phase is bypassed when the indicated URL resource in Phase 1 is the only one to be integrated.
- *Phase 3 - Content Extraction*: The procedure is straightforward when data collection is done using existing APIs as it corresponds to a simple *retrieval* action. However, if there is no available API, *web scraping* techniques have to be applied.
- *Phase 4 - Storage & Integration:* As a last step, the retrieved content from the previous phase is parsed and stored using a *unified representation* model able to aggregate various types of multimedia. The base data model will be the SIMMO ( https://github.com/MKLab-ITI/simmo ) one.

The main resources the data collection will consume are the network bandwidth for downloading and the hard disk space (~100GB) for preserving the web resources that include multimedia content of high quality and resolution.

| | |
|---|---|
| INPUT(S) | • Web entry points (URL addresses)<br>• Text queries |
| OUTPUT(S) | Multimedia records (and linked files) in a predefined data model |

| | |
|---|---|
| PROGRAMMING LANGUAGES/TOOLS | JAVA, MongoDB |
| INTEGRATION WITH OTHER COMPONENTS | Output of data collection module will be given to:<br>● Text & audio analysis (T3.3)<br>● Visual analysis (T3.2) |
| DEPENDENCIES | External APIs (e.g. Twitter) and tools (e.g. Youtube-dl), according to the desired resources |
| CRITICAL FACTORS | ● Availability of free data<br>● Internet speed |
| TIMELINE | Implementation plan:<br>● State-of-the-art study and definition of raw input (web entry points and queries) (M3, M13)<br>● Implementation of skeleton service (M4-M6)<br>● Implementation of first version of crawling, scraping & search (M7-M10)<br>● Data collection (M8-M20)<br>● Upgrade of every submodule – Inclusion of more resources (M14-M18)<br>● Integration & Evaluation (M11-M12, M19-M20) |

### 3.1.2 Space Sensing: Data Acquisition from drones and satellites

The data for modelling the space will be collected from drone campaigns in the site of the scenario and from web services that provide satellite data of different levels. Preferably a DIAS will be selected. All visual data will be preprocessed in a cloud service in XR4DRAMA platform. The expected computational load requires 32-64 GB of RAM and a GPU.

| | |
|---|---|
| INPUT(S) | Region of interest |
| OUTPUT(S) | Available satellite data (image and DEM) and collected drone data after preprocessing. |
| PROGRAMMING LANGUAGES/TOOLS | python, C++ |
| INTEGRATION WITH OTHER COMPONENTS | Space Modelling component |

| | |
|---|---|
| DEPENDENCIES | Available satellite services |
| CRITICAL FACTORS | - Selection of the demo area for collecting drone images<br>- Selection of a competitive web service for remote sensing data with adequate spatio-temporal resolution |
| TIMELINE | Overall (M3-M12)<br>Drone data preprocessing service (M6)<br>Drone data acquisition and in situ tests (M10)<br>Remote sensing data market analysis (M4)<br>1st implementation of remote sensing data acquisition service (M6-M10)<br>2nd implementation of remote sensing data acquisition service (M10-M16) |

### 3.1.3 Physiological and environmental Data acquisition

Systems dedicated to acquire physiological and environmental data will be treated separately as they have different levels of technological readiness.

The wearable system for monitoring physiological parameters is more advanced and it will be adapted to the working environment and it will be integrated in the platform. Instead, the wearable (or portable system) will be developed to acquire environmental data, according to the requirements of the scenarios, it will be tested at laboratory level to evaluate the mock up as a proof of concept.

**Physiological Data acquisition**

The physiological data will be acquired using a sensing platform based on textile sensors fully integrated in the vest layer in contact with the skin and a data logger that is able to recorder data on board, to process and to send data via Bluetooth 2.1.
In the sensorized garment (smart vest) are integrated

- two textile electrodes
- one textile piezoresistive sensor
- one jack connector to plug the garment to the electronic device
- a pocket to hold the electronic during activity

and in the data logger (RUSA Device) are integrated sensors to detect the trunk movements (Mag(X,Y,Z), Gyro (X,Y,Z), Acc(X,Y,Z), Quaternion (X,Y,Z,W)).

| | |
|---|---|
| INPUT(S) | Sensor readings, biopotential and biomechanical input |

| | |
|---|---|
| OUTPUT(S) | Raw Data (ECG, BR, Mag(X,Y,Z), Gyro (X,Y,Z), Acc(X,Y,Z), Quaternion (X,Y,Z,W) |
| PROGRAMMING LANGUAGES/TOOLS | C# Libraries for integration in x4Drama platform,<br>Software visualization and data extraction from internal memory of the data logger |
| INTEGRATION WITH OTHER COMPONENTS | Sensor data analysis (T3.1)<br>Stress level detection (T3.4) |
| DEPENDENCIES | |
| CRITICAL FACTORS | Smart vest has to fit the body of the users (size development needs) |
| TIMELINE | Physiological sensing with smart clothes (M05-M20) |

**Environmental Data acquisition**

A feasibility study will be conducted to design the system for the environmental data acquisition according to the T2.5, in which a wearable (or portable) proof of concept mock-up with sensors capable to acquire data correlated to the vulnerability curves, will be implemented and used to collect environmental parameters. The mock-up will be tested at laboratory level.

## 3.2 Data Analysis and Processing Modules

These modules analyse and process the data in the platform.

### 3.2.1 Sensor Data Analysis

Sensor data analysis (T3.1) will process and analyze various physiological signals, as well as inputs from environmental sensors, in order to determine stress levels and/or activity recognition. The physiological signals will be extracted from a wearable smartvest, with embedded sensors such as inertial ones, ECG and respiratory piezoresistive sensor. These sensors can be analyzed separately or combined. For the individual analysis of the sensors, the framework that will be followed, is the following:

- Extract the raw measurements (online or offline, depending on whether the analysis will be real time or not)
- Filtering of the raw signals
- Extraction of relevant features. The features depend on the initial signal.
- Feeding of the features to a classifier in order to detect stress or activities

During this task, the physiological parameters that are strongly correlated with stress detection levels will also be identified. For this subtask we will employ typical correlation methods. This procedure will be conducted offline, after collecting an adequate amount of relevant data.

For the multimodal processing of all the sensors embedded in the smartvest or the combination of those signals with the environmental sensors, we will use neural networks or fuzzy logic systems. The fusion of sensor measurements with other types of data, like video and audio, will be in Task 3.5 and will be achieved with the use of deep learning algorithms.

| | |
|---|---|
| Input(s) | Sensor readings, which are usually time series data |
| Output(s) | Classification labels that will respond to the categories that will be defined regarding stress levels |
| Programming Languages/Tools | R and/or Python |
| Integration with other components | -Stress level detection (T3.4)<br>-Multimodal information fusion (T3.5) |
| Dependencies | Libraries (eg Caret, Numpy) |
| Critical Factors | Retrieval of real-time data and real-time results |
| Timeline | • Study the SoA methods for the analysis of project's sensors (M3-M5)<br>• Evaluation of methodologies (M6-M10)<br>• Implementation for the first prototype (M11-M13)<br>• Adjust the sensor data analysis framework after the 1st prototype (M13-M20)<br>• Implementation for the 2nd and final prototype (M20-M22) |

### 3.2.2 Audio based Stress level Detection Analysis

The audio stress detection component (T3.4) serves to analyze the stress level of a person (e.g. a First Responder in the field) based on their speech and voice. It is based on a deep learning model that takes into account acoustic and advanced prosodic features such features as Mel-frequency cepstrum coefficients, pitch, loudness, and timbre.

From a set of audio records that contain stress level information a general stress level is computed after a training process. This model can be adjusted for every user if there is a recording from that user in a basal (low stress level) status, allowing the generation of an individual model that can be applied afterwards for the same user.

On online operation audio recordings are analyzed. Two cases are considered: First case is when there is an individual stress model, then the output of the model can be considered the user's stress level. In the second case, the system does not have the basal audio from the user, and will use the general model. In that case the absolute stress level can include a strong bias, depending on the user, but it can be used when compared to new/previous recordings, to detect changes in the stress level (if the user evolves to a worse or better situation).

The information is later combined with the sensor-based stress level estimations described above.

| | |
|---|---|
| INPUT(S) | Audio recordings (e.g. from FRs in the field) |
| OUTPUT(S) | JSON structure including stress level estimation |
| PROGRAMMING LANGUAGES/TOOLS | TBD |
| INTEGRATION WITH OTHER COMPONENTS | The functionality will be exposed through REST services, packaged with Docker.<br>- Stress level detection will receive audio from crawling services<br>- Outputs are provided to the KB for combining with sensor-based stress level detection |
| DEPENDENCIES | |
| CRITICAL FACTORS | |
| TIMELINE | M12: 1st Version<br>M18: 2nd Version<br>M22: Final Version |

### 3.2.3 Visual Analysis

Visual Analysis services (Task 3.2) will analyse archival images or videos depicting places of interest to be 3D reconstructed.

The main services will be:

i) Scene Recognition and

ii) Building and Object Localization

Scene Recognition service will receive images' sets and videos as input. Its aim is to extract information about the type of area or building(s) depicted in the analysed images or video frames using a classification deep neural network. In the case of video input, we will first split it into video shots, extract some keyframes and then based on them get the scene-related information. This service can also give information about an emergency event depicted in the current video shot or image set, such as flood or fire. The output of the Scene Recognition service will be the semantic information extracted per video shot that will be saved in the Knowledge Base. This information can enrich the corresponding 3D model reconstructed using the specific image set or video analysed by this service.

Building and Object Localization service will also receive the video shots or images' sets as its input, after they have been analysed by Scene Recognition. Initially, an optimization of the images or video frames will be carried out using a Photorealistic Style Transfer method. Photorealistic Style Transfer transfers the style of one image to another but preserves the original structure and detail outline of the content image. The aim is that the output image looks like in different lighting time of day or weather in order to optimize the results of detection algorithms. For this purpose, a set of selected style images should be available to be used by this service. Next, proper segmentation deep neural networks will be deployed for the detection and localization of buildings and objects of interest existing in the scene. The outputs of the Building and Object Localization service will be semantic information (e.g., if there are buildings in the scene, what other objects are detected) and the binary masks and bounding boxes of the detected buildings and objects.

The semantic information extracted from both Scene Recognition and Building and Object Localization services can enrich the corresponding 3D model that was reconstructed using the specific image set or video analysed by these services. Additionally, the binary masks of the Building and Object Localization service can help the 3D reconstruction process by separating the building or object of interest from its background.

The main resources the visual services will consume are the GPU for the inference mode of the deep learning models and the hard disk space (~20GB) to temporarily save the input images and videos to be analyzed.

| | |
|---|---|
| INPUT(S) | • images/video frames<br>• videos/video shots |
| OUTPUT(S) | • images/video frames<br>• video shots<br>• binary masks<br>• bounding boxes<br>• semantic information (detected classes and labels)<br><br>The output information can be in JSON form. |
| PROGRAMMING LANGUAGES/TOOLS | python / anaconda |
| INTEGRATION WITH OTHER COMPONENTS | Visual Analysis services will receive information from:<br><br>• Data acquisition from social media, web and open repositories (T2.1)<br>• Space sensing using drones and cameras (T2.2) |

| | |
|---|---|
| | Visual Analysis services will send information to:<br>● 3D reconstruction of the area (T4.4)<br>● Multi-modal information fusion (T3.5) |
| DEPENDENCIES | Libraries (e.g., tensorflow, keras, pytorch, caffe, OpenCV, Numpy) |
| CRITICAL FACTORS | ● Find proper datasets to train visual analysis models<br>● Occlusions (e.g., video captions) in input videos/images<br>● Shooting angle in input videos/images |
| TIMELINE | ● Collect initial data and study SoA on scene recognition and object detection/localization (M3-5)<br>● Define initial technical requirements and deploy baseline architecture (M6-M10)<br>● Integrate the module for the operational and 1st prototype (M11-M13)<br>● Collect new data and update literature (M14-M15)<br>● Update technical requirements and set new goals (M16-19)<br>● Integrate the component for the 2nd and final prototype (M20-M22) |

### 3.2.4 Audio and Textual Analysis

Audio and Textual Analysis consist in two main parts: Automatic Speech Recognition (ASR) and Text Analysis (TA).

The ASR component serves to transcribe spoken language (recordings) into written text. The Text Analysis component serves to extract relevant information from written text (e.g. social media, text document, speech transcripts, etc.) and make this information available in a structured (machine processable) form.

This can serve to extract specific information required e.g. for the preparation of on-location shoots or emergency interventions by automatically analyzing large amounts of available background information about the location, or to gather real-time information from a variety of sources during an intervention.

| | |
|---|---|
| INPUT(S) | Audio files and written text |
| OUTPUT(S) | Structured information (JSON) |
| PROGRAMMING LANGUAGES/TOOLS | Java, Python, deep learning frameworks such as Tensorflow, ... |

| | |
|---|---|
| INTEGRATION WITH OTHER COMPONENTS | The functionality will be exposed through REST services, packaged with Docker.<br>● Audio inputs for ASR will be provided by T2.1 (crawling) and from the mobile device<br>● Textual inputs for TA will come from ASR as well as Data acquisition from social media, web and open repositories (T2.1) |
| DEPENDENCIES | |
| CRITICAL FACTORS | |
| TIMELINE | M12: 1st Version<br>M18: 2nd Version<br>M22: Final Version |

### 3.2.5 Space Modelling Module

The scope of this component is to exploit visual data from UAVs, digital archives and web resources, and satellite remote sensing to generate 3D models of urban and country areas. The workflow for visual data combines custom SfM, SLAM and stereo-matching algorithms, commercial and open-source tools. Satellite images and available DEM will be used to rapidly extract the landscape for the xR modules of the authoring platform. The initial rough terrain will be fused with the 3D data produced from in-situ images from drones; the processing and the data fusion of 3D data consist of mutual registration, mesh model creation and photo texturing of the models via the available image and satellite data. All terrain/ 3D models will be geo-referenced.

| | |
|---|---|
| INPUT(S) | Images, DEM and ROI |
| OUTPUT(S) | 3D meshes, 2.5D models in the forms of obj and geotiffs |
| PROGRAMMING LANGUAGES/TOOLS | python, C++, javascript |
| INTEGRATION WITH OTHER COMPONENTS | Space Sensing |
| DEPENDENCIES | Space Sensing |
| CRITICAL FACTORS | Availability of “good” satellite data and reference of models of different resolution |
| TIMELINE | • Overall (M5-M18)<br>• State-of-the-art of 3D reconstruction and 3D data registration |

| | |
|---|---|
| | (M5-M7)<br>• 1st prototype (M12)<br>• Final service (M18) |

## 3.3 Data Linking and understanding Modules

### 3.3.1 Decision Support System

The Decision Support System (DSS) module will be actively used in the process of extraction of relevant information from the Knowledge Base . In the case of XR4DRAMA the need of handling uncertain information leads to the use of fuzzy and/or non monotonic reasoning approaches, in order to resolve incomplete and inconsistent information. In other words, the reasoning process aims to derive facts and higher-level implicit knowledge from information already generated by the aforementioned XR4DRAMA modules, and asserted in the ontologies, preparing the information to be presented to the user. Also the component will be also responsible for query formulation. The query information will consist of all the data extracted from users regarding entities, relation and objects. Depending on the information, the DSS will inaugurate reasoning patterns within the semantic integrated structure.

There are five main categories of DSSs

*1. Communications-driven*

*2. Data-driven*

*3. Document-driven*

*4. Knowledge-driven*

*5. Model-driven*

The elicitation of the most significant designing principles regarding media production planning and disaster management will be the core factor for deciding which DSS category is the most appropriate.

| | |
|---|---|
| INPUT(S) | Data in RDF form, relevant to the queries |
| OUTPUT(S) | Data extracted from the Knowledge Base |
| PROGRAMMING LANGUAGES/TOOLS | JAVA |
| INTEGRATION WITH OTHER COMPONENTS | WP4 |

| | |
|---|---|
| DEPENDENCIES | • Use-case needs and requirements<br>• Semantic representation infrastructure |
| CRITICAL FACTORS | • Large Knowledge base might be difficult to handle<br>• The lack of specificity can lead to poor results |
| TIMELINE | Implementation Plan<br>• DSS Requirements elicitation (M7-M8)<br>• Design of the DSS (M9-M10, M15-M16)<br>• Development of the DSS (M11-M12, M17-M20)<br>• Testing and Validation (M13-M14, M21-M22) |

### 3.3.2 Text Generation Module

The Text Generation component serves to transform structured information (from Text Analysis and other sources) into written text, e.g. messages or reports.

| | |
|---|---|
| INPUT(S) | Structured data, e.g. from text analysis, stress detection, visual analysis, etc. |
| OUTPUT(S) | Written text (human readable) |
| PROGRAMMING LANGUAGES/TOOLS | Java |
| INTEGRATION WITH OTHER COMPONENTS | The functionality will be exposed through REST services, packaged with Docker.<br>Inputs can come from (after aggregation in the KB):<br>• Audio and written language analysis (T3.3)<br>• Stress detection (T3.4)<br>• Visual analysis (T3.2)<br>• Sensor data analysis (T3.1) |
| DEPENDENCIES | All input information needs to be aggregated by the KB |
| CRITICAL FACTORS | |
| TIMELINE | M12: 1st Version<br>M18: 2nd Version<br>M22: Final Version |

### 3.3.3 Semantic integration

The scope of this module is to create the appropriate semantic knowledge structures in order to map the incoming (heterogenus) data from the different modules of XR4DRAMA. The fundamental components of the semantic integration module are going to be: a) **XR4DRAMA Knowledge Base (KB)** and b) **KBpopulation and Semantic Enrichment**.

The XR4DRAMA Knowledge Base (or ontology) lay the foundations to represent semantically all the knowledge derived from several modalities like, multi-modal inputs (text, image, video, speech), sensor observations, geo-spatial data and other contextual information (e.g. social media). The vast diversity of incoming data leads to a more modular design approach that will be achieved with the adaptation and the extension of several existing sub-ontologies and standards which will be reviewed during the procedure. Predominant tools for deploying the ontology is the Protégé and for repository is the GraphDB.

The KBpopulation and Semantic Enrichment component will be in charge for updating the base by exploiting the queries made by users, constantly integrating information that is the output of other components, interlinking with external sources (other structural semantic repositories) and make a good use of the knowledge derived from these sources. Additionally objects like finding homogenous data and combining data from different sources will be achieved. Several mapping services will be supported accordingly with the format of the input e.g. XML, JSON.

| | |
|---|---|
| INPUT(S) | Analysis results and fusion of heterogeneous information |
| OUTPUT(S) | RDF representation format |
| PROGRAMMING LANGUAGES/TOOLS | • OWL/OWL2<br>• Protégé |
| INTEGRATION WITH OTHER COMPONENTS | • WP2:use-cases<br>• WP3 & WP4 Knowledge |
| DEPENDENCIES | • Communication Bus<br>• Components/Modules that access the KB |
| CRITICAL FACTORS | • Scalability of ontology.<br>• Early clarification/specification of modeling requirements to avoid the possibility of poor results. |
| TIMELINE | • Subtask 1: Modeling requirements elicitation and study of methodologies (M7-M9) |

| | |
|---|---|
| | • Subtask 2: Design and Development of KB (M10-M14)<br>• Subtask 3: Population and Validation (M15-M19)<br>• Sutask 4 : Fusion and Interlinking (M20-M22) |

### 3.3.4 Scene Understanding

Scene understanding for AR interaction to assist actors in the field from cameras in the field. The main goal of this task is to develop 3D computer vision, object detection and scene segmentation algorithms on top of devices' commercial APIs, to assist user navigation in the real world and also quickly identify potential threats and dangers for the agents in the field. These inference models will run at the edge, where the h/w allows it, to ensure that no personal data leak and so that ethics issues are confronted for the aim of European citizen; this approach will also make the approach fast and scalable.

| | |
|---|---|
| INPUT(S) | images and videos |
| OUTPUT(S) | bounding boxes and classification info |
| PROGRAMMING LANGUAGES/TOOLS | python, C# |
| INTEGRATION WITH OTHER COMPONENTS | AR, GIS |
| DEPENDENCIES | |
| CRITICAL FACTORS | computational efficiency, the availability and usability of in situ cameras |
| TIMELINE | • Overall (M5-M22)<br>• 1st protorype (M12)<br>• Final service (M22) |

### 3.3.5 GIS

This task aims at providing a geospatial database with 2D and 3D content and a reference frame to be the underlying localization platform that will allow all relevant data to be suitably placed in 3D space. The geographic information system (GIS) will connect to all relevant processed data of WP3 and will manage them in the geo-referenced system via geospatial queries, in order to support the AR interaction. Information will be served to users either on demand, or via their spatial coherence and other set of rules implemented through spatial queries. All available 2D information and 3D reconstructed areas/buildings of interest will be organized in layers accessible to the AR module. The GIS will be developed in PostGIS, geoserver, QGIS and leaflet and run as part of the XR4DRAMA platform.

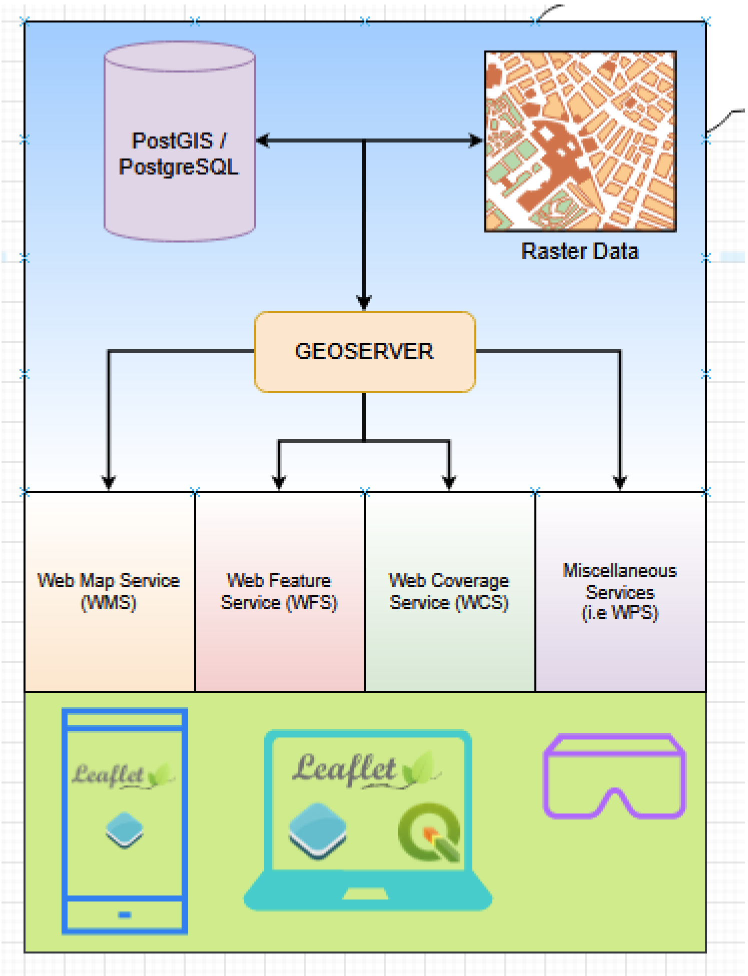


Figure 2 GIS Modules

| | |
|---|---|
| Input(s) | text, vector and raster data (from existing open data services, user input) |
| Output(s) | text, vector and raster data |
| Programming Languages/Tools | python, sql, javascript |
| Integration with other components | Space Modelling, AR, VR authoring tool, data warehouse, knowledge base |
| Dependencies | Space Modelling |
| Critical Factors | |
| Timeline | • Overall (M6-M22) |

| | |
|---|---|
| | • 1st protorype (M12)<br>• Final service (M20) |

## 3.4 Data Storage Modules

These modules store the data in the platform.

### 3.4.1 Data Warehouse

The data warehouse is the main data storage for the entire platform. The data warehouse will be hosted in the cloud and will be accessible through REST APIs to the other components of the system.

The data model of the warehouse will be defined in the upcoming months based on the user requirements as well as the architecture of other components.

| | |
|---|---|
| INPUT(S) | Stores data from components to provide access to other components |
| OUTPUT(S) | Stores data from components to provide access to other components |
| PROGRAMMING LANGUAGES/TOOLS | PHP<br>Appropriate databases (e.g., MariaDB, InfluxDB, etc.) |
| INTEGRATION WITH OTHER COMPONENTS | All components that need access to common data |
| DEPENDENCIES | On all the data collection and manipulating components |
| CRITICAL FACTORS | The data warehouse needs to be scalable and reliable |
| TIMELINE | M1-M6: Conceptualization and architecture design<br>M7-M12: Development of first version<br>M18: 2nd Version<br>M22: Final Version |

| DATABASE | VERSIONS ALLOWED | NOTES |
|---|---|---|
| MongoDB | 2.4.x | For fast, non-relational storage |
| GraphDB | | Non-relational triple store |

Table 1 Table of Databases

## 3.5 Data Visualisation Modules

These are the modules that visualise the data to the various end users.

### 3.5.1 Collaborative tool for VR

The collaborative tool will help in the users manipulating 3D models as well as displaying situational awareness inside a 3D environment to multiple users at the same time. The tool will work over virtual reality devices connected to the same system.

The collaborative tool for VR will be made using unity3D[1] game engine. The tool will use 3D environments created using the authoring tool and would display data from various sensors integrated in the entire platform.

| | |
|---|---|
| INPUT(S) | • User Input (User-facing application)<br>• 3D reconstructions (from data warehouse)<br>• Location metadata and user metadata (created by authoring tool, retrieved from data warehouse)<br>• Live data from sensors etc. (via data warehouse) |
| OUTPUT(S) | • none foreseen currently |
| PROGRAMMING LANGUAGES/TOOLS | • C#<br>• Unity<br>• Custom networking tools |
| INTEGRATION WITH OTHER COMPONENTS | • Data warehouse: get project data<br>• Middleware APIs to access data |
| DEPENDENCIES | Other end user applications and middleware API |
| CRITICAL FACTORS | the tool consume and display a lot of data and has to be fast |
| TIMELINE | M1-M6: Conceptualization and architecture design<br>M6-M12: Development of first version<br>M18: 2nd Version<br>M22: Final Version |

### 3.5.2 VR Authoring Tool

The VR authoring tool will be the main user tool that will start a project. The user will be able to use the tool to define the location where they would like to start the project and the tool would gather information from various sources as well as trigger the other components that work towards data gathering to start gathering the data required for the project. The authoring tool will be able to display 3D models as well as Different types of data collected from the modules and would help the user create a simulated environment of the location of the project.

[1] https://unity.com/

The VR authoring tool will be developed using Unity3D and will be a standalone desktop applications for the users.

| | |
|---|---|
| INPUT(S) | • User Input (User-facing application)<br>• OpenStreetMap Data<br>• 3D reconstructions<br>• Location metadata (from user and possibly external sources) |
| OUTPUT(S) | • New projects, configurations etc. (on the data warehouse)<br>• User-created metadata |
| PROGRAMMING LANGUAGES/TOOLS | • C#<br>• Unity |
| INTEGRATION WITH OTHER COMPONENTS | • Data warehouse: create projects, get data |
| DEPENDENCIES | All data gathering and processing components |
| CRITICAL FACTORS | The tool will have to be fast and responsive |
| TIMELINE | M1-M6: Conceptualization and architecture design<br>M6-M12: Development of first version<br>M18: 2nd Version<br>M22: Final Version |

### 3.5.3 AR Tool

This task will deploy the appropriate applications that will allow the xR device to project and manage the available information from the XR4DRAMA platform on the real world to offer the control centre perception to 1st responder The module will be implemented for a dedicated device AR/MR (potentially Hololens, or industry standard AR/MR device). Different scenarios will be developed based on user requirements in T6.2 such as i) view and examine data from the platform in the real environment, ii) provide paths to specific areas of interest, iii) perform measurement (dimensions, volumes etc.) via the AR/MR device, iv) show guidelines for dangerous situations, v) capture images and videos and send them back to the XR4DRAMA platform. The augmented and mixed reality scenarios will be optimized and finalized through the evaluation and validation procedures of WP6. AR capabilities will be supported by a relevant framework (e.g. SolAR) and Unity3D will be used for rendering.

Navigation for AR will be a submodule, whose scope is to register, via the MR/AR device (potentially Hololens, or industry standard MR device) movements in real-world into a Geographic Information System (GIS), which will be treated as the underlying localization platform for augmented (or even mixed reality). To perform localization, a model of object detection and multi-view matching via convolutional neural network (CNN) architectures will be trained to identify the user’s position in the 2D/3D map also exploiting GNSS, IMU sensors and matching the videos from the AR device and to the pre reconstructed 3D models of the environment. The localization algorithm will exploit all the available XR4DRAMA platform information, on top of sensors and visual navigation.

| | |
|---|---|
| INPUT(S) | Text, vector data, Images, 3D (?) |
| OUTPUT(S) | On-Screen Display |
| PROGRAMMING LANGUAGES/TOOLS | python, C++, C#, java |
| INTEGRATION WITH OTHER COMPONENTS | ● Desktop/Web App of the Control Room<br>● GIS Database and server<br>● awareness app<br>● scene understanding<br>● decision support system<br>● text generation |
| DEPENDENCIES | |
| CRITICAL FACTORS | Get all types of data and interactions on schedule to appropriately design the AR app |
| TIMELINE | • Overall (M5-M22)<br>• 1st protorype (M12)<br>• Final service (M22) |

#### 3.5.4 Awareness Apps for end users

This activity will develop a situation awareness mobile application for the wider public and/or first responders teams. The mobile app will be able to detect the user context based on variables such as location, time and proximity to dangerous areas, and will inform the user about the event current status, possible threats and alerts. The data provided by the users of the mobile app are fed into the XR4DRAMA platform, where the analytic modules process the input and the results are integrated into the overview of the current situation. The end-user application is going to be part of the external layer of XR4DRAMA platform and some of its functionalities are: Displaying info to users, capturing data using mobile phone's sensors , alerting every time there is a need to notify the user and interacting with other components (e.g receiving and sending data before and after an incident for updating the situation).

| | |
|---|---|
| INPUT(S) | • Data regarding each situation from XR4DRAMA control room(e.g alerts,disaster type)<br>• GIS-Location based data<br>• Authentication and authorization data |
| OUTPUT(S) | • Multimodal data from mobile sensors<br>• Reports |
| PROGRAMMING LANGUAGES/TOOLS | C# / Unity |
| INTEGRATION WITH OTHER COMPONENTS | • WP3<br>• WP5 |
| DEPENDENCIES | • Use cases<br>• Communication bus |
| CRITICAL FACTORS | • Bad reception on the mobile network<br>• Bad GPS reception |
| TIMELINE | • Overall (M6-M22) |

# 4 PLATFORM INTEGRATION

The main outcome of the WP5 of XR4DRAMA will be an integrated system with all the components working in coherence with each other.

As it can be concluded from the review of the defined technical components of XR4DRAMA in the previous sections, the integration task conceived for the project faces the prospect of connecting a heterogeneous set of modules, some oriented to user interaction, some to data processing, some to data gathering, and some to the development of the situational awareness in the platform.

A strategy has been devised in order to integrate the technical modules defined in the previous sections effectively and successfully. The main aim is to reduce the risks associated with the integration of various modules with different envisioned applications, developed by different partners in the project, and uses different frameworks. This helps us ensure that the development of all these modules can proceed in a near ad hoc manner, therefore meeting all the milestones for integration and delivery of various prototypes that will be mentioned later on.

A microservices based architecture is used to facilitate the integration. This helps us in producing highly maintainable, testable, independently deployable and independently owned modules in the entire system.

This type of architecture model has been chosen by taking into account the role of XR4DRAMA platform in supporting do use cases as well as situational awareness for the user. The architecture takes into account both web-based services running on the cloud as well as user machines for interaction with the user.

The following diagram shows the first version of the architecture as defined by the various components described in the DoA and their functionalities envisioned in the system for now.

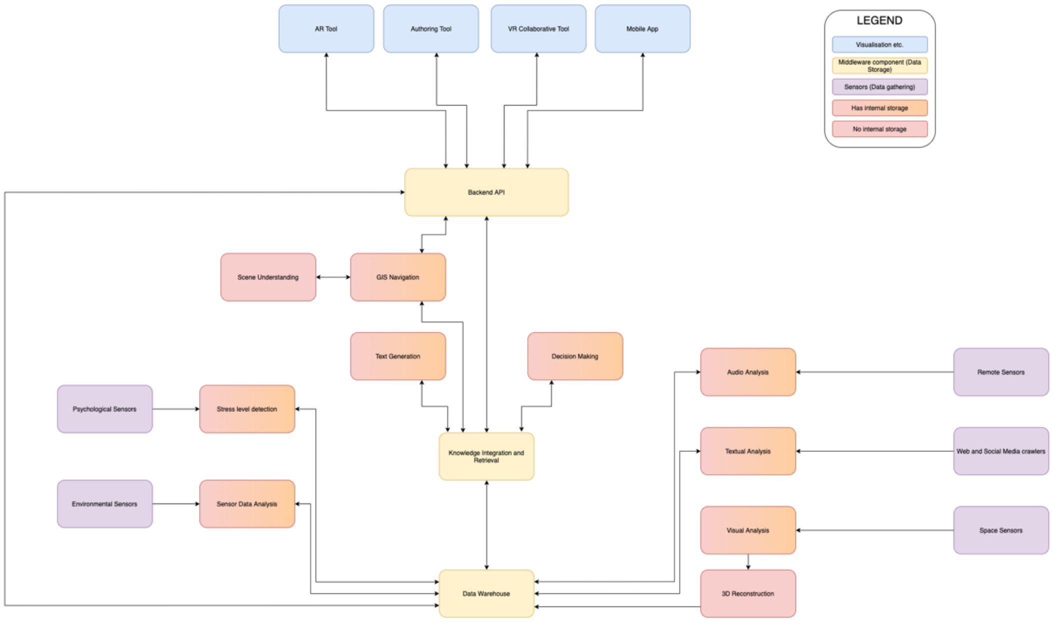


Figure 3 Dependency diagram and data flow

The diagram shows the dependency of the different components and how they will be communicating in the entire system. The diagram also shows the storages and local storages of various components as imagined in a micro service-based architecture.

The strategy adopted to integrate the technical modules into this architecture model rests on a triage method that allows to classify, separate, and address modules according to how and when they execute. For this purpose, the following classes have been defined:

- **Configuration processes**: or processes responsible for generating an output that will be used as a background element in the project (e.g. the data needed from the user to create a project).
- **One-time processes**: or processes executed a single time upon the definition of a new project, or at any point in the project life from definition to implementation (e.g. crawling of data from the maps system to have a 3D environment).
- **Real-time processes**: or processes that execute continuously during a prolonged period of time in the project's life, for instance data crawlers, VR, and other similar applications.
- **Asynchronous processes**: or processes that execute independently or asynchronously from the user interaction with the tool. Their output is delivered upon finalization of the execution in an ad-hoc manner.
- **Chained processes**: or processes that each is composed of single elementary modules, the purpose of each module is to feed the following one in the process.

The integration strategy conceives a call for the following approach to integrating various modules based on their classification:

- Modules working *on configuration processes* and not considered as an integral part of the platform although their output is incorporated by default in the integrated platform. The same shows that the configuration processes execute outside the platform but maintain the compatibility with the platform.
- Major efforts in development and integration would be put towards *one time processes* and *real time processes*. To maintain the simplicity in the integration efforts would be made to move these processes as configuration processes as well as maintain synchronicity between these processes.
- *Asynchronous processes* that also classify as one time processes such as injection of data from OpenStreetMap[2] will be called remotely from the user terminal. No resources are allocated on the cloud platform for such processes since the interaction happens between third party services and the user machine.
- The modules of *chained processes* will be tightly integrated into single larger modules. These composite modules could integrate elements with different ownerships. Also, it is expected that each chained process would map closely to aspects addressed in a single work package. Therefore it is safe to assume that collaboration among partners responsible for building composite modules is already established under the corresponding work package umbrella

To implement the strategy described above we will first work on creation of a machine design, where each machine would work as a separate entity based on the tasks and would consist of components that help achieve the task results. Following that a deployment model would be developed which would help connect the different machines as well as to the cloud infrastructure, this would help in the development of the final architecture which would be deployed in the machine-based model.

## 4.1 Development cycles and integration Milestones (NURO)

The plan for the development of XR4DRAMA platform is to start by creating basic scenarios which can be supported by the first integration of the system and proceed stepwise by adding more features towards the final system. The detailed timelines of the individual modules which are presented in previous sections will follow the general cycle: after the completion of each prototype version based on the timings defined by the project milestones, The functioning of the prototype is assessed, the needs for further development are identified, a agreed upon work plan for the next prototype version is finalized (also based on user inputs) And the development cycle towards the next prototype is initiated. Example driven development plan includes the following prototypes:

• **Operational Prototype**: It is connected to the completion of the setup of the operational infrastructure of XR4DRAMA system. This version is evaluated by the users to look at the basic needs.

[2] https://www.openstreetmap.org/

- **Functional Prototype**: It includes the 1st version of the XR4DRAMA platform by integrating the basic features for the supported services/modules. This version is thoroughly evaluated by the users
- **Final System**: The final versions of the XR4DRAMA modules with all functionalities fully available as planned and new user requirements/needs incorporated

### 4.1.1 Operational Prototype

The operational prototype will be the first integrated version of the entire platform. The operational prototype will be made live in Month 12 of the project. In this prototype the following concerns will be kept in mind:

1. All the initial requirements from the user to make the basic version of the system is assured.
2. The integration of all the components (albeit in their 1st versions) is possible and works as envisioned during the architecture development
3. All the components are able to receive desired inputs and can produce a version of the desired outputs as mentioned in the previous sections.
4. The user is able to create a project and view the data related to the project.
5. The data produced is secure and is GDPR[3] compliant.

### 4.1.2 Functional Prototype

The functional prototype of the system will be the first feedback-based prototype. This prototype will be developed in the Month 18 of the project. In this prototype the following concerns will be kept in mind:

1. The user feedback is taken into account from the operational prototype for the development of this prototype.
2. Each separate component is able to achieve its entire functionality as envisioned during the project proposal and the lifecycle of the project.
3. The communications between each of the separate component is robust and secure
4. The user is able to use all the functionality as described by the use cases and user requirements.
5. New functionalities are added in case new requirements arise after M12.

### 4.1.3 Final prototype

The final prototype will be the final version of the XR4DRAMA platform. The final prototype will be developed on Month 22 of the project. This will be the prototype which will be the final result of the project. The following concerns will be kept in mind:

1. User feedback from functional protype is taken into account for the development of this version
2. Focus is put on optimisation of system functionalities
3. Focus on packaging for easier deployment for future exploitation

[3] https://gdpr-info.eu

4. Focus on IP Management inside the system on a technological level
5. The security of the system is tested and validated
6. Optimisation of the speed of the system.

#### 4.1.4 Integration Timelines

<table>
<tr><td></td><td></td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td></tr>
<tr><td></td><td>Platform</td><td colspan="6">Requirements Gathering</td><td colspan="6">1st Development Cycle</td><td colspan="6">2nd Development Cycle</td><td colspan="4">Final development cycle</td><td colspan="2"></td></tr>
<tr><td></td><td></td><td colspan="11"></td><td>Operational prototype</td><td colspan="5"></td><td>Functional Prototype</td><td colspan="3"></td><td>Final Version</td><td colspan="2"></td></tr>
<tr><td colspan="2">Data Acquistion Modules</td><td colspan="24"></td></tr>
<tr><td></td><td>Data Acquisition from Web and Social Media</td><td colspan="3"></td><td colspan="3">Skeleton service</td><td colspan="4">First Version</td><td colspan="10">Data Collection</td><td colspan="4"></td></tr>
<tr><td></td><td>Data Acquisition from drones and satellites</td><td colspan="6"></td><td colspan="4">First Version</td><td colspan="6">Second Version</td><td colspan="8"></td></tr>
<tr><td></td><td>Physiological and environmental Data acquisition</td><td colspan="4"></td><td colspan="16">Continuous Development and Integration</td><td colspan="4"></td></tr>
<tr><td colspan="2">Data Analysis Modules</td><td colspan="24"></td></tr>
<tr><td></td><td>Sensor Data Analysis</td><td colspan="10"></td><td colspan="3">1st Prototype</td><td colspan="7">Adjust framework</td><td colspan="2">Final Version</td><td colspan="2"></td></tr>
<tr><td></td><td>Audio based Stress level</td><td colspan="6"></td><td colspan="6">1st Prototype</td><td colspan="6">2nd Prototype</td><td colspan="4">Final Version</td><td colspan="2"></td></tr>
<tr><td></td><td>Visual Analysis</td><td colspan="5"></td><td colspan="5">Initial technical requirements</td><td colspan="3">1st Prototype</td><td colspan="2">New data</td><td colspan="4">Requirement gathering</td><td colspan="3">Final Version</td><td colspan="2"></td></tr>
<tr><td></td><td>Audio and Textual Analysis</td><td colspan="6"></td><td colspan="6">1st Prototype</td><td colspan="6">2nd Prototype</td><td colspan="4">Final Version</td><td colspan="2"></td></tr>
<tr><td></td><td>Space Modelling Module</td><td colspan="4"></td><td colspan="3">SoTA</td><td colspan="5">1st Prototype</td><td colspan="6">Final Prototype</td><td colspan="6"></td></tr>
<tr><td colspan="2">Data Linking Modules</td><td colspan="24"></td></tr>
<tr><td></td><td>Decision Support System</td><td colspan="6"></td><td colspan="2">Requirements</td><td colspan="2">Design</td><td colspan="2">Development</td><td colspan="2">Testing</td><td colspan="2">Requirements</td><td colspan="2">Design</td><td colspan="2">Development</td><td colspan="2">Testing</td><td colspan="2"></td></tr>
<tr><td></td><td>Text Generation Module</td><td colspan="6"></td><td colspan="6">1st Prototype</td><td colspan="6">2nd Prototype</td><td colspan="4">Final Version</td><td colspan="2"></td></tr>
<tr><td></td><td>Semantic integration</td><td colspan="6"></td><td colspan="3">Requirements</td><td colspan="5">Design & Development</td><td colspan="5">Population & Validation</td><td colspan="3">Fusion</td><td colspan="2"></td></tr>
<tr><td></td><td>Scene Understanding</td><td colspan="4"></td><td colspan="8">1st Prototype</td><td colspan="10">Final Version</td><td colspan="2"></td></tr>
<tr><td></td><td>GIS</td><td colspan="5"></td><td colspan="7">1st Prototype</td><td colspan="8">Final Version</td><td colspan="4"></td></tr>
<tr><td colspan="2">Data Storage Modules</td><td colspan="24"></td></tr>
<tr><td></td><td>Data Warehouse</td><td colspan="6"></td><td colspan="6">1st Prototype</td><td colspan="6">2nd Prototype</td><td colspan="4">Final Version</td><td colspan="2"></td></tr>
<tr><td colspan="2">Data Visualisation Modules</td><td colspan="24"></td></tr>
<tr><td></td><td>Collaborative tool for VR</td><td colspan="6"></td><td colspan="6">1st Prototype</td><td colspan="6">2nd Prototype</td><td colspan="4">Final Version</td><td colspan="2"></td></tr>
<tr><td></td><td>VR Authoring Tool</td><td colspan="6"></td><td colspan="6">1st Prototype</td><td colspan="6">2nd Prototype</td><td colspan="4">Final Version</td><td colspan="2"></td></tr>
<tr><td></td><td>AR Tool</td><td colspan="4"></td><td colspan="8">1st Prototype</td><td colspan="10">Final Version</td><td colspan="2"></td></tr>
<tr><td></td><td>Awareness Apps for end users</td><td colspan="5"></td><td colspan="17">Continous Development and Integration</td><td colspan="2"></td></tr>
</table>

Figure 4 Implementation Timeline

# 5 PILOT USE CASES DEMONSTRATIONS

The project envisions 2 use cases which help align the technological development as well as will be used to evaluate the platform.

## 5.1 Disaster Management

The disaster management use case in the project requires the system to be fast and can work in real-time data gathering and visualisation as the use case will work in emergency situations.

The use case will have 2 phases during operations:

1. Preparation of the interventions: This will include simulations of the scenario in VR and help decision makers. More information can be found in D6.1 (Heise, et al., 2021)
2. In the fields: This will include the use of XR Tools and physiological and environmental sensors. More information can be found in D6.1 (Heise, et al., 2021)

## 5.2 Media Production Planning

The media production planning use case helps test the simulations provided by the system and the collaborative approach provided by the tools.

The use case will have various steps to plan media production:

1. Examination of the chosen location for media production,  inspection inside VR as well as use the system to gather data about the location
2. Planning of the production, creating various simulations as well as collaborating to provide a better overview to different stakeholders.
3. Supervision and Management during the production process from a remote position.

More information can be found in D6.1 (Heise, et al., 2021).

# CONCLUSIONS

The deliverable presents the technical roadmap of XR4DRAMA platform. It includes the required information for the development of the project with a view to attaining the scientific and the technical objectives envisaged. However, since the project is following an iterative development procedure (use cases requirements-development-evaluation), it is expected that small adaptations and deviations on the initial technical specifications and the module functionalities foreseen by this document will be needed to satisfy the final user requirements. Potential changes/adaptations might be required at component or subcomponent level. Further updates will be provided in D5.2 with the full architecture of the platform.

# REFERENCES

[1] F. Yergeau, "UTF-8, a transformation format of ISO 10646, No. RFC 3629.," 2003.

[2] N. Heise, A. Primavesi, S. Gensch, S. Diplaris, S. Symeonidis, M. Monego, D. Norbiato and F. Zaffanella, "D6.1, Pilot use cases and initial user requirements," 2021.