Socially-acceptable Extended Reality Models and Systems

D3.1 Description of the modular architecture of the SERMAS XR Agent and design of its modules

31/03/2023

Funded by the European Union

DELIVERABLE INFORMATION	
Deliverable leader	UNIMORE
Document type	R — Document, report
Document code	D3.1
Document name	D3.1 Description of the modular architecture of the SERMAS XR Agent and design of its modules
Work Package / Task	WP3 Design of the SERMAS XR Agent
Delivery Date (DoA)	31/03/2023
Actual Delivery Date	March 2023
Reviewers	Valeria Di Pasquale, Rosario De Chiara, Francesco Zurolo (POSTE)

DELIVERABLE HISTORY			
Date	Version	Author	Summary of main changes
26/01/2023	0	Valeria Villani, Lorenzo Sabattini	Table of contents
02/03/2023	0	All the partners	Details about modules
09/03/2023	0	Valeria Villani Rosario de Chiara	Modules vs. requirements
15/03/2023	0	Valeria Villani	Overview of the modules
16/03/2023	0	Valeria Villani	Introduction and executive summary
22/03/2023	0	Luca Capra	Description of system architecture and interoperability among modules
23/03/2023	1	Valeria Villani	Integration of input from partners about the description of the modules
29/03/2023	2	Valeria Villani	Integration of input from partners

DISSEMINATION LEVEL		
PU	Public	X
PP	Restricted to other programme participants (including the EC services)	
RE	Restricted to a group specified by the consortium (including the EC services)	
CO	Confidential, only for the members of the consortium (including the EC)	

SERMAS partners

Disclaimer

This project has received funding from the Horizon Europe programme under the Grant Agreement 101070351.

SERMAS ● Grant Agreement: 101070351 ● 2022 – 2025 | Duration: 36 months
Topic: HORIZON-CL4-2021-HUMAN-01-13

Public Executive Summary

The aim of this deliverable is to present a first version of the SERMAS XR Agent modular architecture. In particular, it addresses project objective OBJ 2 *Define the SERMAS XR Agent*.

The document will, hence, present and discuss the architecture of the system and the functionalities and specifications for its modules. The approach leading to a modular architecture is detailed, focusing on general applicability and possibility to integrate external input.

Table of contents

1. Introduction 1

2. Overview of modules and architecture 2

3. SERMAS XR Agent modules 6

3.1. 3D modeling and representation 7

3.1.1. Functionalities 7

3.1.2. Hardware specifications 7

3.1.3. Software specifications 7

3.1.4. Input/output from/for the other modules 7

3.1.5. Addressed requirements 7

3.2. Occlusion management 8

3.2.1. Functionalities 8

3.2.2. Hardware specifications 8

3.2.3. Software specifications 8

3.2.4. Input/output from/for the other modules 8

3.2.5. Addressed requirements 8

3.3. User detection 9

3.3.1. Functionalities 9

3.3.2. Hardware specifications 9

3.3.3. Software specifications 9

3.3.4. Input/output from/for the other modules 9

3.3.5. Addressed requirements 9

3.4. User identification 10

3.4.1. Functionalities 10

3.4.2. Hardware specifications 10

3.4.3. Software specifications 10

3.4.4. Input/output from/for the other modules 10
3.4.5. Addressed requirements 11
3.5. User classification 12
3.5.1. Functionalities 12
3.5.2. Hardware specifications 12
3.5.3. Software specifications 12
3.5.4. Input/output from/for the other modules 12
3.5.5. Addressed requirements 12
3.6. User adaptation 13
3.6.1. Functionalities 13
3.6.2. Hardware specifications 13
3.6.3. Software specifications 13
3.6.4. Input/output from/for the other modules 13
3.6.5. Addressed requirements 13
3.7. Account management 14
3.7.1. Functionalities 14
3.7.2. Hardware specifications 14
3.7.3. Software specifications 14
3.7.4. Input/output from/for the other modules 14
3.7.5. Addressed requirements 14
3.8. Saved game management 15
3.8.1. Functionalities 15
3.8.2. Hardware specifications 15
3.8.3. Software specifications 15
3.8.4. Input/output from/for the other modules 15
3.8.5. Addressed requirements 15
3.9. Interaction initiation 16

3.9.1. Functionalities 16
3.9.2. Hardware specifications 16
3.9.3. Software specifications 16
3.9.4. Input/output from/for the other modules 16
3.9.5. Addressed requirements 17
3.10. Physical interaction 18
3.10.1. Functionalities 18
3.10.2. Hardware specifications 18
3.10.3. Software specifications 18
3.10.4. Input/output from/for the other modules 18
3.10.5. Addressed requirements 18
3.11. Multiple users handling 19
3.11.1. Functionalities 19
3.11.2. Hardware specifications 19
3.11.3. Software specifications 19
3.11.4. Input/output from/for the other modules 19
3.11.5. Addressed requirements 19
3.12. Communication in multiple languages 20
3.12.1. Functionalities 20
3.12.2. Hardware specifications 20
3.12.3. Software specifications 20
3.12.4. Input/output from/for the other modules 20
3.12.5. Addressed requirements 20
3.13. Multimodal interaction 21
3.13.1. Functionalities 21
3.13.2. Hardware specifications 21
3.13.3. Software specifications 22

3.13.4. Input/output from/for the other modules 22
3.13.5. Addressed requirements 22
3.14. Dialogue management 24
3.14.1. Functionalities 24
3.14.2. Hardware specifications 24
3.14.3. Software specifications 24
3.14.4. Input/output from/for the other modules 24
3.14.5. Addressed requirements 24
3.15. Multiple choice quizzes management 25
3.15.1. Functionalities 25
3.15.2. Hardware specifications 25
3.15.3. Software specifications 25
3.15.4. Input/output from/for the other modules 25
3.15.5. Addressed requirements 25
3.16. Context and intent understanding 26
3.16.1. Functionalities 26
3.16.2. Hardware specifications 26
3.16.3. Software specifications 26
3.16.4. Input/output from/for the other modules 26
3.16.5. Addressed requirements 26
3.17. User sentiment detection and interpretation 28
3.17.1. Functionalities 28
3.17.2. Hardware specifications 28
3.17.3. Software specifications 28
3.17.4. Input/output from/for the other modules 28
3.17.5. Addressed requirements 28
3.18. Disturbances suppression and environment resilience 30

3.18.1. Functionalities 30
3.18.2. Hardware specifications 30
3.18.3. Software specifications 30
3.18.4. Input/output from/for the other modules 30
3.18.5. Addressed requirements 30
3.19. Object detection 31
3.19.1. Functionalities 31
3.19.2. Hardware specifications 31
3.19.3. Software specifications 31
3.19.4. Input/output from/for the other modules 31
3.19.5. Addressed requirements 31
3.20. QR codes scanning 32
3.20.1. Functionalities 32
3.20.2. Hardware specifications 32
3.20.3. Software specifications 32
3.20.4. Input/output from/for the other modules 32
3.20.5. Addressed requirements 32
3.21. Payment handling 33
3.21.1. Functionalities 33
3.21.2. Hardware specifications 33
3.21.3. Software specifications 33
3.21.4. Input/output from/for the other modules 33
3.21.5. Addressed requirements 33
3.22. Text, images, videos and 3D models display 34
3.22.1. Functionalities 34
3.22.2. Hardware specifications 34
3.22.3. Software specifications 34

3.22.4. Input/output from/for the other modules 34
3.22.5. Addressed requirements 35
3.23. Path planning for the user 36
3.23.1. Functionalities 36
3.23.2. Hardware specifications 36
3.23.3. Software specifications 36
3.23.4. Input/output from/for the other modules 36
3.23.5. Addressed requirements 36
3.24. Motion in physical space 37
3.24.1. Functionalities 37
3.24.2. Hardware specifications 37
3.24.3. Software specifications 37
3.24.4. Input/output from/for the other modules 37
3.24.5. Addressed requirements 37
3.25. Motion in virtual space 38
3.25.1. Functionalities 38
3.25.2. Hardware specifications 38
3.25.3. Software specifications 38
3.25.4. Input/output from/for the other modules 38
3.25.5. Addressed requirements 38
3.26. Connection with staff 39
3.26.1. Functionalities 39
3.26.2. Hardware specifications 39
3.26.3. Software specifications 39
3.26.4. Input/output from/for the other modules 39
3.26.5. Addressed requirements 39
3.27. CMS account management 40

3.27.1. Functionalities 40
3.27.2. Hardware specifications 40
3.27.3. Software specifications 40
3.27.4. Input/output from/for the other modules 40
3.27.5. Addressed requirements 40
3.28. Training management 41
3.28.1. Functionalities 41
3.28.2. Hardware specifications 41
3.28.3. Software specifications 41
3.28.4. Input/output from/for the other modules 41
3.28.5. Addressed requirements 41
3.29. Media import 42
3.29.1. Functionalities 42
3.29.2. Hardware specifications 42
3.29.3. Software specifications 42
3.29.4. Input/output from/for the other modules 42
3.29.5. Addressed requirements 42
3.30. Dashboard management 43
3.30.1. Functionalities 43
3.30.2. Hardware specifications 43
3.30.3. Software specifications 43
3.30.4. Input/output from/for the other modules 43
3.30.5. Addressed requirements 43
4. Interoperability among modules 44
5. Mapping of SERMAS pilot requirements to modules 45
6. Conclusions 51

List of Figures

Figure 1: Architecture of the SERMAS agent. 2
Figure 2: High-level module interaction modalities 4

List of Tables

Table 1: Mapping of the requirements of the SERMAS pilots to the proposed modules for the SERMAS architecture 46

1. Introduction

One of the major goals of the SERMAS project is to design its XR Agent as prototypical, general-purpose system realizing an XR model through a combination of HW, SW and algorithmic modules, allowing socially acceptable interaction with non-specialized users, adapting to the context. To achieve this goal, it is fundamental that the Agent is built upon a modular architecture that ensures general applicability and opens it for integration of external input. The aim of this document is, hence, to present the architecture of the SERMAS XR Agent, with a specific focus on its modular organization. As discussed in Section 2, the proposed architecture relies on cloud infrastructure resources supported with edge computing frameworks and shared communication protocols to allow for scalability, while the modules are derived abstracting from system requirements and will be instantiated in the considered application scenarios.

The document is organized as follows. Section 2 presents an overview of the proposed modular architecture, together with a description of the design approach. In Section 3, the details of the foreseen modules are provided, while Section 4 discusses how communication among modules and scalability to other modules will be handled. Then, Section 5 provides a clear overview of the relationships among the proposed modules and SERMAS requirements discussed in D2.1. Finally, Section 6 follows with some concluding remarks.

It is noteworthy mentioning that the content of this deliverable is preliminary and will undergo an update at M12, with a second release. In the current version, the design of the architecture and its modules relies on the analysis of requirements for socially acceptable XR systems (from D2.1) and initial implementation efforts.

2. Overview of modules and architecture

The foreseen modular architecture for the SERMAS XR Agent is depicted in Figure 1.

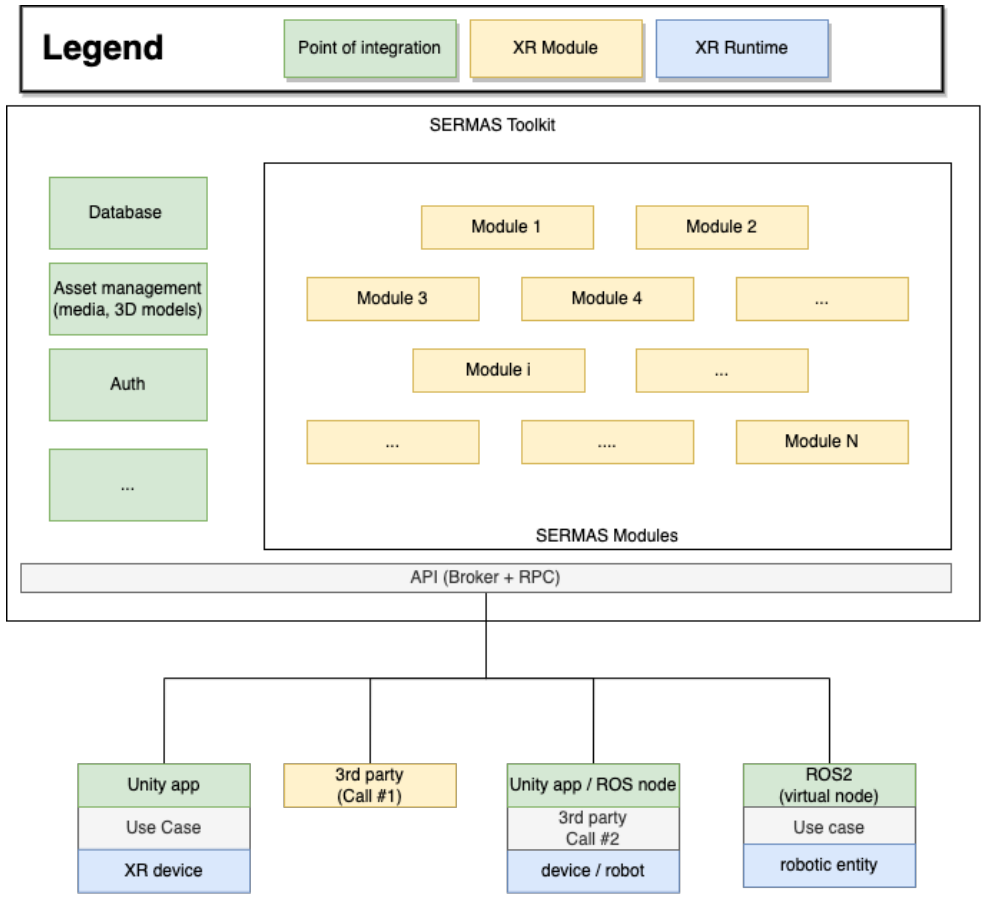

Figure 1: Architecture of the SERMAS agent.

The overall architecture enables fully distributed scenarios, regardless of the location of the involved runtime. XR runtimes can leverage transparently on remote resources to enable complex services compositions and a mesh of different runtimes, eventually enabling multi-user and multi-domain interaction models. To tackle security within this architecture, a possible solution is to use a JSON Web Token (*JWT*) based message-signing approach, which will contain non-private information of an acting user (or impersonation of a user, such as a module or application runtime). Information consistency can be cryptographically verified to avoid JWT data being altered during transport over the network. Inspired by Zero

Trust Architecture[1], a centralized component in charge of authentication and authorization will identify and decide which user or impersonation may retrieve information or perform action on protected resources.

To ensure interoperability between modules, a unified communication model at module/application level is proposed, as depicted in Figure 2. Such model covers common exchange patterns:

Request-Response model over HTTP+JSON

This communication model offers request / response patterns where a client "calls" a server. A simple use case is to fetch information or trigger an instantaneous synchronous task (e.g., turn on a light, open a gate).

Event-driven messaging model over MQTT

This communication approach offers a publish / subscribe model and is suited for event passing and near real-time notifications from the system. The process is in general asynchronous and is best suited for use cases where there is no specific timing between exchanges or where the amount of data sent does not require explicit feedback from the receiving part(s).

[1] Rose, S. , Borchert, O. , Mitchell, S. and Connelly, S. (2020), Zero Trust Architecture, Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD, [online], https://doi.org/10.6028/NIST.SP.800-207,
https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=930420 (Accessed March 27, 2023)

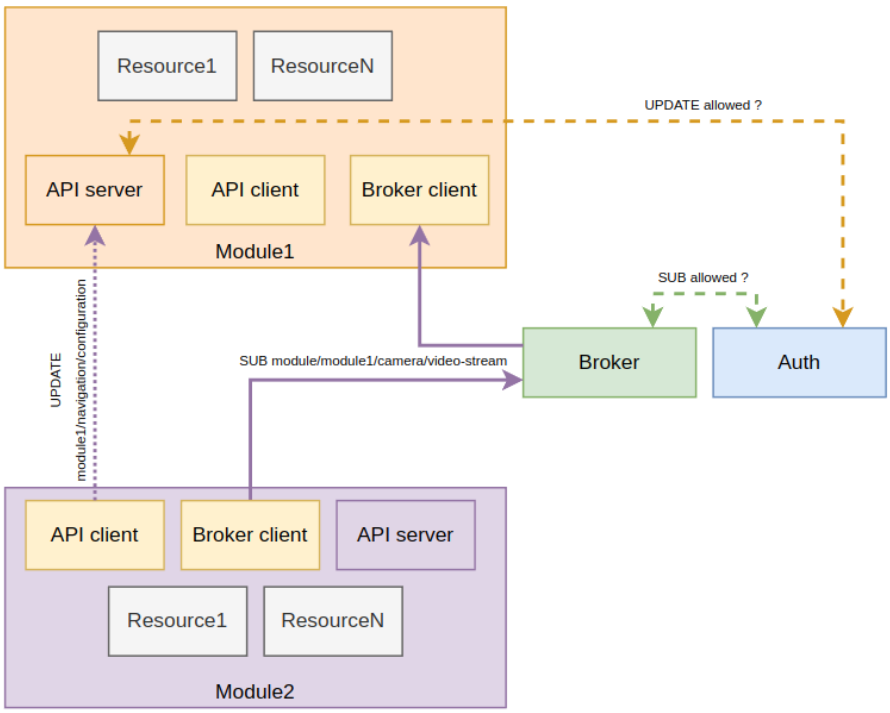

Figure 2: High-level module interaction modalities

As regards the modules, they are defined with the aim of distributing responsibilities for the implementation of the required functionalities, as described in Deliverable D2.1, and simplifying the implementation of new requirements (identified in later stages of the project or related to other use cases and application scenarios). To this end, to identify the required modules for the SERMAS agent, we considered a bottom-up approach, which started from the requirements and, when possible, merged and abstracted them to identify the corresponding required functionality. As a result, a module will represent a (software and/or hardware) piece of the SERMAS agent implementing a set of requirements, or a single requirement in some cases.

It is noteworthy discussing that we considered this approach for the following advantages. First, it allows to clearly identify the responsibilities of each module and among project partners, thus identifying teams in charge of addressing each requirement. As a result, the risk of leaving some requirements not addressed is reduced. Second, it allows scalability towards new application scenarios and use cases, which will follow with the second open call and after the end of the project. Indeed, any additional requirements introduced by a new use case can be addressed by the SERMAS agent by implementing a corresponding module that

does not conflict or overlap with the existing ones. This will be supported also by the presence of an underlying architecture that allows simple integration of new modules and implements a communication protocol shared among the modules.

As a result of this approach, a list of 30 modules has been derived that define the functionalities that the SERMAS agent has to implement to satisfy the requirements of the project use cases. It is important to point out that such modules do represent the functionalities required to the SERMAS agent and do not represent the functionalities that will be implemented in the project. In other words, some of these modules are already available on the market or as part of suited development platforms and will be simply integrated in the SERMAS architecture, with minor changes and adaptations; conversely, some other modules will be developed from the early stage in the SERMAS toolkit. As a result, such modules will require different levels of implementation effort in the SERMAS project. Such effort will be defined in conjunction with WP6, which is charge of the implementation of the SERMAS agent, and WP2, which is in charge of the definition and evaluation of the proofs of concept for the project.

In the following sections, a detailed view of the identified modules is provided and for each of them the expected functionalities, the related hardware and software specifications, together with the connections with the other modules are specified. Moreover, responsibilities for each module are reported. Finally, 5it is discussed how the selected modules address the requirements from the use cases, establishing a correspondence between modules and requirements.

3. SERMAS XR Agent modules

The foreseen modules for the SERMAS XR agent are listed below and described in detail in the following sections:

1. 3D modeling and representation
2. Occlusion management
3. User detection
4. User identification
5. User classification
6. User adaptation
7. Account management
8. Saved game management
9. Interaction initiation
10. Physical interaction
11. Multiple users handling
12. Communication in multiple languages
13. Multimodal interaction
14. Dialogue management
15. Multiple choice quizzes management
16. Context and intent understanding
17. User sentiment detection and interpretation
18. Disturbances suppression and environment resilience
19. Object detection
20. QR codes scanning
21. Payment handling
22. Text, images, videos and 3D models display
23. Path planning for the user
24. Motion in physical space
25. Motion in virtual space
26. Connection with staff
27. CMS account management
28. Training management
29. Media import
30. Dashboard management

3.1. 3D modeling and representation

Responsible partner(s)	SPXL and DW

3.1.1. Functionalities

- Show an instance of the 3D model of the agent. This includes appearing on any screen, in MR and as physical agent.

3.1.2. Hardware specifications

- A platform supporting the selected 3D engine, that is Unity.

3.1.3. Software specifications

- Load a supported 3D model either from local package or via URI via 3D engine.

3.1.4. Input/output from/for the other modules

- I: the 3D model built statically or imported at runtime
- O: event for the loading status / result
- O: event for the visibility status

3.1.5. Addressed requirements

- FR_ASA01: Manifest in 3D mode (on any screen)
- FR_ASA02: Manifest in 3D mode (in MR / physical space)

This module applies also to the other SERMAS pilots, namely the Post Office Agent (POA) and the Receptionist Agent (RA). Indeed, its representation is considered in the non-functional requirements NFR_POA34 and NFR_RA24.

3.2. Occlusion management

Responsible partner(s)	SPXL and DW

3.2.1. Functionalities

- Hide the agent model when occluded by a physical obstacle

3.2.2. Hardware specifications

- Device supporting an AR toolkit.

3.2.3. Software specifications

- Leverage on underlying AR toolkit and plane detection. This will be handled (more or less) automatically by the SDK.

3.2.4. Input/output from/for the other modules

- O: event indicating the visibility of the agent changed state

3.2.5. Addressed requirements

- FR_ASA03: Follow occlusion standards in MR mode

3.3. User detection

Responsible partner(s)	SUPSI and SPXL

3.3.1. Functionalities

- Detect the presence of a user in the operational area of the agent, such as in a predefined area around the kiosk

3.3.2. Hardware specifications

- An RGB camera able to provide a video stream or RGB-D sensor capable to provide also depth information
- Available computing, GPU computing and memory (if applicable)
- Local processing (for improved performance and limited bandwidth usage)

3.3.3. Software specifications

- An object detection / classification model trained on people, possibly with skeleton detection and tracking capabilities
- Filter by a predefined perimeter the camera view
- Existing SDKs, such as the one of state-of-the-art sensors like Azure Kinect, already provide user's skeleton detection

3.3.4. Input/output from/for the other modules

- I: user's behavior (e.g., standing close to the agent)
- O: event indicating a person entered/exited the area of active monitoring (including, the last N most relevant frames in a predefined time window), to *Interaction initiation*
- O: event indicating a person is stationing in the area of the active monitoring, to *Interaction initiation*

3.3.5. Addressed requirements

- FR_ASA04: Detect user
- FR_POA01: User detection
- FR_RA01: User detection

3.4. User identification

Responsible partner(s)	KCL for managing the entire process, POSTE for integration with enterprise system, and SPXL for implementation

3.4.1. Functionalities

- Capability to identify an external user, who request to enter the building, based on user provided data (e.g., email, ID Card, QR code and combination of sources).

3.4.2. Hardware specifications

- Not applicable.

3.4.3. Software specifications

- An async API to exchange user provided data with the enterprise system and receive the meeting verification and authentication result.

3.4.4. Input/output from/for the other modules

- O: event indicating a user has been identified (or not), to *Interaction initiation*, *Multimodal interaction*, *Context and intent understanding* and possibly other modules
- I: information from *QR codes scanning*, if QR codes are used for this purpose
- I: user provided data from acquisition module(s), from an external unit (the user)
    - Host name - Name of the employee hosting the meeting
    - Alternative host name - Optional alternative host name
    - Host email - The host email corresponds to the account on the Microsoft Teams server to accommodate the guest confirmation call
    - Alternative host email - The alternative host email corresponds to the account on the Microsoft Teams server to accommodate the guest confirmation call
    - Meeting data and time - Date and time of the meeting
    - Meeting room identifier - Unique identifier of the meeting room

3.4.5. Addressed requirements

- FR_RA08: External visitor identification
- FR_RA12: User verification

3.5. User classification

Responsible partner(s)	TUDa, SUPSI and SPXL

3.5.1. Functionalities

- Classify the user based on visual metrics, to recognize their age range
- Classify the user based on voice characteristics and utterances
- Classify the user into different classes (generations), related to age range (e.g., boomer, gen_z, gen_x, millennial)

3.5.2. Hardware specifications

- Available GPU computing and memory (if applicable)
- Camera
- Microphone

3.5.3. Software specifications

- An AI model to classify the user based on face emotions (e.g., age, gender, facial expression, race)
- An AI model to classify the user based on voice information (e.g., surprise, neutral, calm, happy, sad, angry, fear, disgust)
- An AI model to classify the user based on facial expressions and physical characteristics extracted from gestures as input signals for response generation

3.5.4. Input/output from/for the other modules

- I: captured frames from a user, from *User detection*
- O: user classification based on physical characteristics, to *User adaptation*
- O: user classification based on face emotions, to *User adaptation*

3.5.5. Addressed requirements

- FR_POA11: Promotional customized messages (cluster users)
- FR_RA02: User classification

3.6. User adaptation

Responsible partner(s)	Everyone (UNIMORE coordinating)

3.6.1. Functionalities

- Customize the functionalities of the agent based on the user interacting with it, selecting specific functionalities to be enabled or disabled based on the classified user, and selecting parameters for the enabled functionalities (e.g., specific content to be displayed, language to be used, etc.).

3.6.2. Hardware specifications

- Available GPU computing and memory (if applicable)

3.6.3. Software specifications

- Database matching user's classification information with parameters and functionalities of the agent to be adapted. The content of the database will be defined once the specific contents of the use cases is defined.

3.6.4. Input/output from/for the other modules

- I: user classification information from *User classification*: this information will be used as an additional input to our dialog module for language adaptation to the user's generation
- O: functionalities to enable/disable, adapted parameters

3.6.5. Addressed requirements

- FR_POA02: Language interaction
- FR_POA11: Promotional customized messages (cluster users)
- FR_POA13: Promotional messages (recognized user)

3.7. Account management

Responsible partner(s)	SPXL and KCL for security issues

3.7.1. Functionalities

- Allow user to create an account: provide a way for the user to register to the system.
- Allow user to log in to an account: provide a way for the user to identify within the system

3.7.2. Hardware specifications

- Not applicable.

3.7.3. Software specifications

- UI to provide registration information (e.g., email, password or social login)
- UI to provide email validation (if registered via email)
- UI to recover the password
- Management UI to manage users
- Backend system to store the user credentials
- Backend system allowing email validation (where applicable)
- Backend system allowing password recovery
- UI allowing the user to login
- Provide JWT access token providing user ID, roles and operating scopes

3.7.4. Input/output from/for the other modules

- I: user's account information
- O: Event indicating a new user has registered
- O: API providing JWT token exchange and refresh
- O: API providing JWT token validation

3.7.5. Addressed requirements

- FR_ASA05: Allow user to create an account
- FR_ASA06: Allow user to log in to an account

3.8. Saved game management

Responsible partner(s)	DW and KCL for privacy and security

3.8.1. Functionalities

- Allow user to create a saved game
- Allow user to load a saved game

3.8.2. Hardware specifications

- Not applicable

3.8.3. Software specifications

- UI to let the user select the game to load
- Database of saved games

3.8.4. Input/output from/for the other modules

- I: user's request to save the current game
- I: information on the game to load
- O: an event indicating the current game was saved
- O: an event indicating the desired game was loaded

3.8.5. Addressed requirements

- FR_ASA07: Allow user to create a saved game
- FR_ASA08: Allow user to load a saved game

3.9. Interaction initiation

Responsible partner(s)	TUDa, KCL, SUPSI and UNIMORE

3.9.1. Functionalities

- Communicate the availability for potential interaction to the user
- Greet the user via voice, pose (MR agent) and gestures (MR agent and physical agent), and move towards the user (for the physical agent)
- Generate a greeting based on the identified user class

3.9.2. Hardware specifications

- Display
- Speaker
- Available GPU computing and memory (if applicable)
- Robotic platform:
    - mobile base, to move towards the agent
    - arms, to communicate using gestures
    - lights (if any), to communicate through light signals
    - speakers, to communicate through audio signals

3.9.3. Software specifications

- A set of animations for the 3D model to provide a greeting (e.g., waiving hand)
- A set of poses for the 3D model of the head providing a greeting (e.g., smiling)
- Synthetize greeting gestures
- Command robot lights (if any)
- Move the agent towards the user

3.9.4. Input/output from/for the other modules

- I: information about user's intent from *Context and intent understanding* and user's class from *User classification*
- I: greeting content, from *Dialogue management* and *Multimodal interaction*
- O: communicate to *Motion in physical space* to move towards the user

- O: command for the robot actuators

3.9.5. Addressed requirements

- FR_ASA09: Greet user
- FR_ASA23: Propose interaction
- FR_POA05: Greetings to the visitor
- FR_RA04: Greetings to the visitor
- FR_RA21: Proposing for interaction

3.10. Physical interaction

Responsible partner(s)	UNIMORE, SUPSI and KCL

3.10.1. Functionalities

- Issue items such as tickets, paper forms and badges to the user. Items can be handed out by the agent through its arm(s) or printed by the agent, letting the user to pick the, on their own.
- Handle objects carried by the user. Objects can be picked directly from user's hands or from an area where the user has just placed them (no physical contact between the user and the robot)

3.10.2. Hardware specifications

- Robotic platform, with at least an arm
- RGB-D sensor
- Printer

3.10.3. Software specifications

- Control of the robotic arm and the mobile platform for safe physical interaction with the user

3.10.4. Input/output from/for the other modules

- I: information about user presence and position, from *User detection*
- I: information about items presented by the user, from *Object detection*
- I: information about user's request, from *Multimodal interaction*
- O: command for the robot actuators

3.10.5. Addressed requirements

- FR_ASA21: Handle physical objects presented by user
- FR_POA17: Release ticket number
- FR_POA19: Paper forms distribution
- FR_RA13: Badge issuance

3.11. Multiple users handling

Responsible partner(s)	SPXL, SUPSI and TUDa

3.11.1. Functionalities

- The agent should identify a relevant group of people with a principal actor to interface for
- The agent should identify the user among many
- In the presence of a group of people, the agent should not ask sensitive information to the user

3.11.2. Hardware specifications

- Not applicable

3.11.3. Software specifications

- Identify a group of users inside the operational perimeter of the agent, and recognize that they are in a group

3.11.4. Input/output from/for the other modules

- I: captured frames from a user, from *User detection*
- I: chunks of audio, from *Multimodal interaction*
- O: event indicating a (tentative) count of people in the group

3.11.5. Addressed requirements

- FR_POA21: Recognize user speaking
- FR_RA09: More guests at once

3.12. Communication in multiple languages

Responsible partner(s)	TUDa and SPXL

3.12.1. Functionalities

- Allow user to select (main) language
- Allow the user to interact in the preferred language within the supported ones

3.12.2. Hardware specifications

- Available GPU computing and memory

Note: the use of multilingual models requires larger memory and more GPU resource for real-time inference.

3.12.3. Software specifications

- UI to select the language for the dialogue session
- Ability for the user to set the language via voice input
- Availability of a model trained on multilingual data or ability to switch the dialogue model based on the selected

3.12.4. Input/output from/for the other modules

- I: the selected language by the user
- O: event indicating the selected language for a session, to *Multimodal interaction* and *Dialogue management*

3.12.5. Addressed requirements

- FR_POA04: Language selection
- FR_RA03: Multi language communication

3.13. Multimodal interaction

Responsible partner(s)	SPXL, UNIMORE for management of multimodal interaction and gestures, TUDa for dialogue-based interaction, with either UI or voice, + SUPSI for gesture based interaction and KCL for security ad privacy

3.13.1. Functionalities

- Allow user to select (main) mode of interaction: users may prefer to use a dialogue-based interaction model (via a text appearing on a UI or voice) against a dialogue-based interaction model, and/or gestures
- Allow user to interact via a UI based on text/touch (non-voice based)
- Allow user to interact via voice
- Allow user to interact via gestures
- Provide textual representation of user's voice-based dialogue
- Provide the agent with the ability to interact via voice, UI and gestures
- Provide the agent with the ability to produce context-aware dialogues

Note: This module refers to the possibility, for the user, to communicate with an XR agent in a socially acceptable manner. To this end, different communication modalities should be made available. Although, from a technical perspective, each of them consists in a different implementation effort, they are all merged in a single module since, from the user's point of view, they all relate to communication with the agent.

3.13.2. Hardware specifications

- Display
- Keyboard (virtual keyboard)
- Touch screen (UI and/or virtual keyboard)
- Speaker
- Microphone
- Camera
- Robot

3.13.3. Software specifications

- UI or voice command to switch to UI interaction (voice-based by default)
- UI to allow to switch back to voice -based interaction
- UI with a dialogue like interface
- UI to show a navigation tree from a website or internal app
- Collect chunks of audio from the user and automatic chunking on silence
- Integrate dialogue-based interaction
- Integrate to speech-to-text API
- Integrate to text-to-speech API
- AI model to interpret gestures from the user
- Synthetize gestures

3.13.4. Input/output from/for the other modules

- I: noise cancelled audio from the user, from *Disturbances suppression and environment resilience*
- O: an event indicating the interaction model for the session
- I: agent responses to dialogue, from *Dialogue management*
- O: transcription of speech chunks as text, to *Dialogue management* and *Context and intent understanding*
- O: an event indicating a gesture happened, based on a list of supported one (e.g., waiving)
- O: command for the robot actuators, to produce a gesture
- O: information to be communicated to the user, as a result of interaction, to *Text, images, videos and 3D models display*

3.13.5. Addressed requirements

- FR_ASA11: Allow user to select (main) mode of interaction
- FR_ASA12: Allow user to interact via text/touch
- FR_ASA13: Allow user to interact via voice
- FR_ASA14: Allow user to interact via gestures
- FR_ASA17: Transcribe user requests (voice)
- FR_POA03: Interaction selection
- FR_POA22: Provide needed information
- FR_POA23: Context

- FR_POA24: Engaged and effective communication
- FR_POA25: Gesture
- FR_RA07: Request understanding

3.14. Dialogue management

Responsible partner(s)	TUDa + SPXL

3.14.1. Functionalities

- Provide the agent with conversational capabilities

3.14.2. Hardware specifications

- Available GPU computing and memory

3.14.3. Software specifications

- Provide text-to-dialogue AI model

3.14.4. Input/output from/for the other modules

- I: information about user's intent to interact, from *Context and intent understanding*
- I: language selected by the user, from *Communication in multiple languages*
- I: information about the user, from *User sentiment detection and interpretation* and *User classification*
- I: transcription of speech chunks as text, from Multimodal interaction
- O: generated dialogue, to *Multimodal interaction*, *Interaction initiation*

3.14.5. Addressed requirements

- FR_ASA09: Greet user
- FR_ASA12: Allow user to interact via text/touch
- FR_ASA13: Allow user to interact via voice
- FR_ASA34: Offer help / background / safety info
- FR_POA07: Detect help and Understand request
- FR_POA15: Question answering
- FR_POA22: Provide needed information
- FR_POA23: Context
- FR_RA06: Provide brief help
- FR_RA07: Request understanding

3.15. Multiple choice quizzes management

Responsible partner(s)	DW and TUDa

3.15.1. Functionalities

- Provide multiple choice quizzes
- Evaluate user quiz input and provide brief explanation of mistakes

3.15.2. Hardware specifications

- Available computational resources, CPU, GPU and memory (if applicable)
- Local processing (for speed and limited bandwidth usage)

3.15.3. Software specifications

- A trained model that returns a list of potential related but incorrect answers as multiple choices given a question and its expected answer
- A tool that evaluates the user's answer (choice selection)

3.15.4. Input/output from/for the other modules

- I: user's answer, from *Multimodal interaction*
- O: present the user with a quiz and a list of possible answers

3.15.5. Addressed requirements

- FR_ASA15: Offer multiple choice quizzes
- FR_ASA16: Evaluate user quiz input

3.16. Context and intent understanding

Responsible partner(s)	TUDa, SUPSI, KCL and UNIMORE

3.16.1. Functionalities

- Interpret and predict user's intention to interact

3.16.2. Hardware specifications

- RGB-D (video and depth) sensor
- Microphone
- Computational resources (CPU, GPU, memory) for training and serving the trained models

3.16.3. Software specifications

- Collect user's motion data
- Get access to the sensor SDK (e.g., to be able to use the information of the user's skeleton detected by built-it functionalities)
- Collect user's textual data
- An intent detection model that detects user's intent based on the textual data

3.16.4. Input/output from/for the other modules

- I: detected user nearby the agent, from *User detection*
- I: information extracted from any QR code carried by the user, from *QR codes scanning*
- I: information about any physical objects carried by user, from *Object detection*
- I: user's requests, from *Multimodal interaction*, in the case interaction is initiated by the user (e.g., with a waiving gesture or a voice request)
- O: event indicating a person may be interested in interacting, to *Interaction initiation*, *Multimodal interaction* and *Dialogue management*
- O: communicate to *Motion in physical space* to move towards the user

3.16.5. Addressed requirements

- FR_ASA18: Understand context/intent

- FR_POA06: Detect user's indecision
- FR_POA07: Detect help and Understand request
- FR_RA05: Detect user's indecision
- FR_RA07: Request understanding

3.17. User sentiment detection and interpretation

Responsible partner(s)	UNIMORE, TUDa, SUPSI and KCL

3.17.1. Functionalities

- Detect end-user satisfaction during the interaction, highlighting negative feelings, such as frustration, discomfort, idleness and indecision
- Incorporate this information in the dialogue system for response generation

3.17.2. Hardware specifications

- Camera to detect user's facial expressions
- Microphone
- GPU, to speed up inference

3.17.3. Software specifications

- Estimate emotions from user utterances (e.g., happiness, neutral) during data collection, map these emotions to facial expressions available from publicly available datasets
- Estimate emotions from facial expressions, relying on publicly available datasets to train the model
- Combine emotions estimated from different sources of information (e.g., utterances and facial expressions)
- Dialog component, based, for instance, on Transformers, Pytorch and Python

3.17.4. Input/output from/for the other modules

- I: user's chunks of audio, from *Disturbances suppression and environment resilience*
- I: video of user's face
- O: user's sentiment to *User adaptation* and *Multimodal interaction*, in order to shape the interaction accordingly

3.17.5. Addressed requirements

- FR_ASA22: Detect user idleness/indecision
- FR_POA18: Capturing end-user feedback

- FR_POA26: Sentiment analysis

3.18. Disturbances suppression and environment resilience

Responsible partner(s)	SPXL and POSTE for technology scouting

3.18.1. Functionalities

- Limit the noises from environment
- Remove echoes

Note: This module may or may not rely on specific hardware. Effective solutions are available both as full software solutions (e.g., relying on neural networks, require some calibration) or hardware plugin that could be applied common microphones.

3.18.2. Hardware specifications

- Microphone with hardware cancellation
- Multiple microphones
- Effective solutions (need to be tested in realistic environments) can be pluggable devices that, in cascade to a common microphone, will provide noise cancellation functionalities

3.18.3. Software specifications

- Software based noise cancellation
- Software based multi-microphone syncing and de-noising
- Full software solutions are available as library easy to integrate in other software stacks; they may need calibration, in terms of average noise level

3.18.4. Input/output from/for the other modules

- O: user's noise cancelled audio, to *Multimodal interaction* and *Context and intent understanding*

3.18.5. Addressed requirements

- FR_ASA19: Ignore standard disturbances (environment resilience)
- FR_POA21: Recognize user speaking

3.19. Object detection

Responsible partner(s)	SUPSI and UNIMORE

3.19.1. Functionalities

- Be able to recognize that a user is carrying an object, whose detection is helpful to develop the interaction. For example, if a parcel is detected, it can be easily inferred the reason why the user is there.

3.19.2. Hardware specifications

- RGB-D sensor

3.19.3. Software specifications

- Detection of standard everyday-life objects

3.19.4. Input/output from/for the other modules

- I: raw sensor signals
- I: user's intent, from *Context and intent understanding*
- O: notification of the object that can be used to understand user intent to start interaction, to *Context and intent understanding*, to ask specific questions to the user, to *Multimodal interaction*, and to let the agent handle the object, to *Physical interaction*

3.19.5. Addressed requirements

- FR_ASA20: Detect physical objects presented by user
- FR_POA08: Detect object
- FR_POA16: Information procedure

3.20. QR codes scanning

Responsible partner(s)	SPXL and TUDa

3.20.1. Functionalities

- Read the content of a QR code

3.20.2. Hardware specifications

- QR code scanner, possibly embedded in a camera

3.20.3. Software specifications

- QR code software reader / parser

3.20.4. Input/output from/for the other modules

- I: frames from a camera
- O: content from QR code, if needed in a structured format (txt, json, yaml)
- O: information about (potential) users, to *Context and intent understanding*, *User identification*, *Interaction initiation* and *Multimodal interaction*

3.20.5. Addressed requirements

- QR code scan

3.21. Payment handling

Responsible partner(s)	POSTE

3.21.1. Functionalities

- Manage the payment service for mobile phone credit charge or credit card

Note: The functionalities of this module will be implemented through a mockup that will be tested in a simulated scenario.

3.21.2. Hardware specifications

- Card reader

3.21.3. Software specifications

- A software that manages payment services

3.21.4. Input/output from/for the other modules

- I: information from the user about service to pay for (e.g., for mobile phone top-up: phone number, mobile carrier identifier, amount)
- I: credit card details
- O: feedback to the user about the outcome of the transaction

3.21.5. Addressed requirements

- FR_POA14: Payment services
- FR_POA20: Card payment

3.22. Text, images, videos and 3D models display

Responsible partner(s)	SPXL, TUDa, POSTE and DW

3.22.1. Functionalities

- Ability to display to the user various contents (text, images and 3D models)
- Provide customized ads: provide commercial ads based on the user's age range, the user's identity or based on organizational policies
- Provide non-customize content, such as mandatory safety brief: provide information in the form of a web page or as an audio message about safety information, mandatory to be provided before the interaction with the agent or before accessing the structure
- Provide other content, such as help/ background /safety info

3.22.2. Hardware specifications

- Compatible XR Runtime
- Screen/touchscreen
- Audio speaker
- Microphone

3.22.3. Software specifications

- An API providing a multimedia content URI
- Collect user confirmation to dismiss the message as UI input
- Collect user confirmation to dismiss the message as predefined statement (e.g., "OK", "Yes" o "I accept")

3.22.4. Input/output from/for the other modules

- I: API and/or event to display a 3D object with relevant metadata (eg. Position or interaction modalities)
- I: for content upon request, information about user's request, from *Multimodal interaction*
- I: for customized content, age range, user's identity, from *User classification*
- O: event indicating an interaction result (e.g., event indicating the safety brief has been confirmed by the user)

Note: This module is connected to all the others, since the above mentioned input and output are exchanged with all the other modules.

3.22.5. Addressed requirements

- FR_ASA10: Do (mandatory) safety brief
- FR_ASA24: Display text (on any screen)
- FR_ASA25: Display text (in AR/MR)
- FR_ASA26: Display images (on any screen)
- FR_ASA27: Display images (in AR, MR)
- FR_ASA28: Display videos (on any screen)
- FR_ASA29: Display videos (in AR, MR)
- FR_ASA30: Display 3D models (on a screen)
- FR_ASA31: Display 3D models (in MR, scalable)
- FR_ASA34: Offer help / background / safety info
- FR_POA11: Promotional customized messages (cluster users)
- FR_POA12: Promotional messages
- FR_POA13: Promotional messages (recognized user)
- FR_POA15: Question Answering
- FR_POA16: Information procedure
- FR_POA22: Provide needed information
- FR_RA06: Provide brief help
- FR_RA17: Mandatory safety brief
- FR_RA18: On request safety info
- FR_RA20: Commercial advertising

3.23. Path planning for the user

Responsible partner(s)	POSTE, UNIMORE and TUDa

3.23.1. Functionalities

- Show the position to be reached (e.g., meeting room, or host office)
- Show how to get to a position to be reached
- Provide textual direction to guide the user

3.23.2. Hardware specifications

- Screen where to show the map or path

3.23.3. Software specifications

- Algorithm for path planning

3.23.4. Input/output from/for the other modules

- I: user's request about the position they need to reach, from *Multimodal interaction*
- O: map of the building with indication of where the position to be reached is located, to *Text, images, videos and 3D models display*
- O: path to reach the desired position, either shown on a map or provided as textual information, to *Text, images, videos and 3D models display*

3.23.5. Addressed requirements

- FR_ASA32: Show user how to get to a specific location
- FR_RA14: Instructions for reaching a location
- FR_RA15: Host location
- FR_RA16: Meeting rooms location

3.24. Motion in physical space

Responsible partner(s)	UNIMORE

3.24.1. Functionalities

- Move in the environment, avoiding obstacles
- Reach a desired position and orientation
- Guide the user to specific location in physical space, keeping a fixed distance from the user

3.24.2. Hardware specifications

- Wheeled robotic platform
- RGB-D camera for obstacle detection

3.24.3. Software specifications

- ROS
- Algorithm for robot control
- Algorithm for path planning

3.24.4. Input/output from/for the other modules

- I: target position (e.g., from user's request, from *Multimodal interaction*)
- I: user's position tracking, from *User detection*
- I: user's indecision, from *Context and intent understanding*
- O: command for the robot actuators, to implement the motion

3.24.5. Addressed requirements

- FR_POA10: Movement in environment
- FR_RA19: Guiding

3.25. Motion in virtual space

Responsible partner(s)	DW and SPXL

3.25.1. Functionalities

- The user can move in a space in AR/MR leveraging on markers that indicate the location and link to a context / state of the application

3.25.2. Hardware specifications

- Supported XR Runtime

3.25.3. Software specifications

- Detect a marker based on a database

3.25.4. Input/output from/for the other modules

- I: API to manage markers and context (e.g., training section or relevant/interactive area in a physical place)
- O: event indicating the activation/deactivation of a context based on a marker

3.25.5. Addressed requirements

- FR_ASA33: Move around freely in virtual space (within boundaries)

3.26. Connection with staff

Responsible partner(s)	POSTE, SPXL, TUDa and DW

3.26.1. Functionalities

- Call for staff upon either user's request or decision from the agent itself, when it is not able to satisfy a request from the user
- Allow to get in contact with staff, for example to get approval on visitor access

3.26.2. Hardware specifications

- Microphone
- Speaker
- Touchscreen
- Camera

3.26.3. Software specifications

- UI to call for staff
- API to trigger call for staff in the enterprise system
- Integration of a videoconferencing tool

3.26.4. Input/output from/for the other modules

- I: user's request, from UI or chunks of audio from *Multimodal interaction*
- I: contact of the person to be reached
- O: API to trigger call for staff
- O: event indicating the user a call for staff
- O: video and audio connection

3.26.5. Addressed requirements

- FR_ASA35: Allow user to call for staff
- FR_POA09: Request operator intervention
- FR_RA11 Video-call

3.27. CMS account management

Responsible partner(s)	SPXL, DW and KCL

3.27.1. Functionalities

- Offer registration and login to users

3.27.2. Hardware specifications

- Not applicable

3.27.3. Software specifications

- Integrate an authentication system offering registration, login, password recovery, authorization capabilities (eg. roles) and management interfaces

3.27.4. Input/output from/for the other modules

- I: user's input
- O: API for authentication and authorization
- O: UI for account and policy management

3.27.5. Addressed requirements

- FR_ASA36: Allow user to create CMS account
- FR_ASA37: Allow users to log in to CMS account

3.28. Training management

Responsible partner(s)	SPXL, DW and KCL

3.28.1. Functionalities

- Offer an interface to manage XR training session (a training session is a consequent series of interactions a user has to complete to reach an objective)
- Potentially integrate to a learning management system to reuse clear patterns and existing APIs or UIs to deliver parts of the system.

3.28.2. Hardware specifications

- Not applicable

3.28.3. Software specifications

- Handle and model training sessions
- Handle “levels” or “training steps” enabling a dedicated interaction, connected to a functional experience that has one or more expected goal or result

3.28.4. Input/output from/for the other modules

- I: API to handle training sessions and track user progresses
- I: UI to manage training sessions
- O: an event indicating a user reached or not a goal
- O: an event indicating a user started or finished a training session

3.28.5. Addressed requirements

- FR_ASA38: Allow users to create training(s)
- FR_ASA39: Allow users to edit and save training(s)

3.29. Media import

Responsible partner(s)	SPXL

3.29.1. Functionalities

- Manage different types of media assets including 3D models

3.29.2. Hardware specifications

- Not applicable

3.29.3. Software specifications

- Store assets such as media as retrievable content

3.29.4. Input/output from/for the other modules

- O: API to store and load assets by an identifier

3.29.5. Addressed requirements

- FR_ASA40: Allow import of all standard media types (incl. 3D models)

3.30. Dashboard management

Responsible partner(s)	SPXL

3.30.1. Functionalities

- Users should be able to interact with the application via a management dashboard

3.30.2. Hardware specifications

- Not applicable

3.30.3. Software specifications

- Implement a web-based UI to interact with the components and modules composing the application

3.30.4. Input/output from/for the other modules

- Not applicable

3.30.5. Addressed requirements

- FR_ASA41: Provide dashboard

4. Interoperability among modules

To ensure interoperability between modules and applications, the SERMAS project could adopt two open standards that offer a formalized way to define endpoints and message payload. Using these standards, the level of abstraction is high. It doesn't matter how a module is implemented (i.e., different programming languages).

From the produced specification, other tools can create wrappers with client or server code to be used during development, formal validation of the messages sent and received, generation of UIs (user interfaces) used as documentation and in some cases to interact with the server-side endpoints. Examples in this regard are:

- OpenAPI *https://www.openapis.org* for Request-Response model
- AsyncAPI *https://www.asyncapi.com* for Event-driven messaging model.

For those modules running on the same XR runtime platform (e.g., game engine) hosting the SERMAS toolkit, communication among modules will be managed within the platform itself, with no need for wrapping.

5. Mapping of SERMAS pilot requirements to modules

As mentioned in section Section 1, the above described modules have been derived abstracting the requirements of the three use cases considered in the project. In this section, we verify whether there is actual correspondence among the selected modules and the specific use cases requirements. To this end, we refer to the list of functional requirements in D2.1, for the three SERMAS pilots:

- ASA (Artificial Security Agent), by DW, whose functional requirements are listed in Table 2 of D2.1;
- POA (Post Office Agent), by POSTE, whose functional requirements are listed in Table 4 of D2.1;
- RA (Receptionist Agent), by POSTE, whose functional requirements are listed in Table 7 of D2.1.

Table 1 shows how the proposed modules address the functional requirements of the SERMAS pilots. Comparing the table with the requirements in D2.1, it can be easily verified that all the requirements are associated to (at least) a module, or, in other words, the proposed modules allow to address all the requirements. Exception is made for requirement FR_POA27 "*Monitor environment*", which will not be implemented in the SERMAS project and, hence, is not associated to any requirement. On the other side, all the modules address at least one requirement. Some comments to Table 1 are noteworthy. First, module 1. *3D modeling and representation* applies, by definition, to the three use cases, although there is not explicit requirement for POA and RA referring to it. Second, with respect to requirement FR_POA24 "*Engaged and effective communication*" (Table 4 of D.2.1), it should be pointed out that the proposed architecture will implement it in terms of contents, through context-aware dialogue. For this reason, the requirement FR_POA24 is linked to module 13. *Multimodal interaction* in Table 1. Finally, the last comment refers to the non-functional requirements, which are listed in Table 5, Table 8 and Section 5 of D2.1. Such requirements are not

addressed by the above described modules, and, hence, are not reported in Table 1. The reason for this is the fact that the modules have been derived considering the need to implement the set of functions required to a socially acceptable XR system. Conversely, non-functional requirements are, by definition, not linked to a functionality. Rather, non-functional requirements will provide guidance for the implementation of system modules, defining the boundaries for the achieved outcomes.

Table 1: Mapping of the requirements of the SERMAS pilots to the proposed modules for the SERMAS architecture

Requirements from SERMAS pilots			SERMAS modules
ASA (Artificial Security Agent)	FR_ASA01	Manifest in 3D mode (on any screen)	1. 3D modeling and representation
	FR_ASA02	Manifest in 3D mode (in MR / physical space)	1. 3D modeling and representation
	FR_ASA03	Follow occlusion standards in MR mode	2. Occlusion management
	FR_ASA04	Detect user	3. User detection
	FR_ASA05	Allow user to create an account	7. Account management
	FR_ASA06	Allow user to log in to an account	7. Account management
	FR_ASA07	Allow user to create a saved game	8. Saved game management
	FR_ASA08	Allow user to load a saved game	8. Saved game management
	FR_ASA09	Greet user	9. Interaction initiation 14. Dialogue management
	FR_ASA10	Do (mandatory) safety brief	22. Text, images, videos and 3D models display
	FR_ASA11	Allow user to select (main) mode of interaction	13. Multimodal interaction
	FR_ASA12	Allow user to interact via text/touch	13. Multimodal interaction 14. Dialogue management
	FR_ASA13	Allow user to interact via voice	13. Multimodal interaction 14. Dialogue management

	FR_ASA14	Allow user to interact via gestures	13. Multimodal interaction
	FR_ASA15	Offer multiple choice quizzes	15. Multiple choice quizzes management
	FR_ASA16	Evaluate user quiz input	15. Multiple choice quizzes management
	FR_ASA17	Transcribe user requests (voice)	13. Multimodal interaction
	FR_ASA18	Understand context/intent	16. Context and intent understanding
	FR_ASA19	Ignore standard disturbances (environment resilience)	18. Disturbances suppression and environment resilience
	FR_ASA20	Detect physical objects presented by user	19. Object detection
	FR_ASA21	Handle physical objects presented by user	10. Physical interaction
	FR_ASA22	Detect user idleness/indecision	17. User sentiment detection and interpretation
	FR_ASA23	Propose interaction	9. Interaction initiation
	FR_ASA24	Display text (on any screen)	22. Text, images, videos and 3D models display
	FR_ASA25	Display text (in AR/MR)	22. Text, images, videos and 3D models display
	FR_ASA26	Display images (on any screen)	22. Text, images, videos and 3D models display
	FR_ASA27	Display images (in AR, MR)	22. Text, images, videos and 3D models display
	FR_ASA28	Display videos (on any screen)	22. Text, images, videos and 3D models display
	FR_ASA29	Display videos (in AR, MR)	22. Text, images, videos and 3D models display
	FR_ASA30	Display 3D models (on a screen)	22. Text, images, videos and 3D models display
	FR_ASA31	Display 3D models (in MR, scalable)	22. Text, images, videos and 3D models display
	FR_ASA32	Show user how to get to a specific location	22. Path planning for the user
	FR_ASA33	Move around freely in virtual space (within boundaries)	25. Motion in virtual space

	FR_ASA34	Offer help / background / safety info	22. Text, images, videos and 3D models display 14. Dialogue management
	FR_ASA35	Allow user to call for staff	26. Connection with staff
	FR_ASA36	Allow user to create CMS account	27. CMS account management
	FR_ASA37	Allow users to log in to CMS account	27. CMS account management
	FR_ASA38	Allow users to create training(s)	28. Training management
	FR_ASA39	Allow users to edit and save training(s)	28. Training management
	FR_ASA40	Allow import of all standard media types (incl. 3D models)	29. Media import
	FR_ASA41	Provide dashboard	30. Dashboard management
POA (Post Office Agent)	FR_POA01	User detection	3. User detection
	FR_POA02	Language interaction	6. User adaptation
	FR_POA03	Interaction selection	13. Multimodal interaction
	FR_POA04	Language selection	12. Communication in multiple languages
	FR_POA05	Greetings to the visitor	9. Interaction initiation
	FR_POA06	Detect user's indecision	16. Context and intent understanding
	FR_POA07	Detect help and Understand request	16. Context and intent understanding 14. Dialogue management
	FR_POA08	Detect object	19. Object detection
	FR_POA09	Request operator intervention	26. Connection with staff
	FR_POA10	Movement in environment	24. Motion in physical space
	FR_POA11	Promotional customized messages (cluster users)	5. User classification 6. User adaptation 22. Text, images, videos and 3D models display
	FR_POA12	Promotional messages	22. Text, images, videos and 3D models display
	FR_POA13	Promotional messages (recognized user)	6. User adaptation 22. Text, images, videos and 3D models display
	FR_POA14	Payment services	21. Payment handling

	FR_POA15	Question Answering	22. Text, images, videos and 3D models display 14. Dialogue management
	FR_POA16	Information procedure	19. Object detection 22. Text, images, videos and 3D models display
	FR_POA17	Release ticket number	10. Physical interaction
	FR_POA18	Capturing end-user feedback	17. User sentiment detection and interpretation
	FR_POA19	Paper forms distribution	10. Physical interaction
	FR_POA20	Card payment	21. Payment handling
	FR_POA21	Recognize user speaking	11. Multiple users handling, 18. Disturbances suppression and environment resilience
	FR_POA22	Provide needed information	13. Multimodal interaction 14. Dialogue management 22. Text, images, videos and 3D models display
	FR_POA23	Context	13. Multimodal interaction 14. Dialogue management
	FR_POA24	Engaged and effective communication	13. Multimodal interaction
	FR _POA25	Gesture	13. Multimodal interaction
	FR _POA26	Sentiment analysis	17. User sentiment detection and interpretation
	FR_POA27	Monitor environment	*Not considered in SERMAS architecture*
RA (Receptionist Agent)	FR_RA01	User detection	3. User detection
	FR_RA02	User classification	5. User classification
	FR_RA03	Multi language communication	12. Communication in multiple languages
	FR_RA04	Greetings to the visitor	9. Interaction initiation
	FR_RA05	Detect user's indecision	16. Context and intent understanding
	FR_RA06	Provide brief help	14. Dialogue management 22. Text, images, videos and 3D models display
	FR_RA07	Request understanding	13. Multimodal interaction 14. Dialogue management 16. Context and intent understanding

	FR_RA08	External visitor identification	1. User identification
	FR_RA09	More guests at once	11. Multiple users handling
	FR_RA10	QR code scan	20. QR codes scanning
	FR_RA11	Video-call	26. Connection with staff
	FR_RA12	User verification	4. User identification
	FR_RA13	Badge issuance	10. Physical interaction
	FR_RA14	Instructions for reaching a location	23. Path planning for the user
	FR_RA15	Host location	23. Path planning for the user
	FR_RA16	Meeting rooms location	23. Path planning for the user
	FR_RA17	Mandatory safety brief	22. Text, images, videos and 3D models display
	FR_RA18	On request safety info	22. Text, images, videos and 3D models display
	FR_RA19	Guiding	24. Motion in physical space
	FR_RA20	Commercial advertising	22. Text, images, videos and 3D models display
	FR_RA21	Proposing for interaction	9. Interaction initiation

6. Conclusions

This deliverable defined the architecture of the SERMAS agent, focusing on its modular organization. Indeed, to allow for scalability towards new application scenarios and use cases, the functionalities of the SERMAS XR agent will be implemented by dedicated modules. Interoperability among modules will be guaranteed through appropriate communication protocols, which will also preserve security. The list of foreseen modules was derived from a deep analysis of the requirements of the SERMAS pilots, discussed in the deliverable D2.1. In this document, such modules were extensively discussed, highlighting, for each module, its expected functionalities, high-level hardware and software specifications, information to be exchanged with the other modules. Moreover, it was verified that all the requirements from SERMAS pilots are properly addressed by the foreseen modules.

It is noteworthy to underline that this is a first release of this document, which will undergo through revision and updates in the following months, to take into account the project progress.