MindSpaces

Art-driven adaptive outdoors and indoors design
H2020- 825079

D6.2
Technical Requirements and Architecture

Dissemination level:	Public
Contractual date of delivery:	Month 12, 31.12.2019
Actual date of delivery:	Month 13, 18.1.2020
Workpackage:	WP6 MindSpaces platform development and integration
Task:	T6.4, System Integration
Type:	Report
Approval Status:	Final Version
Version:	1.1
Number of pages:	78
Filename:	D6.2 -MindSpaces _WP6_v1.1.docx

Abstract

This deliverable details the technical requirements and the first UI/UX Prototypes of the MindSpaces platform. It describes all the services of the platform; in addition, it describes the functionalities that will be supported by the platform. Finally, it details the system architecture and the communication methodology of the platform.

co-funded by the European Union

History

Version	Date	Reason	Revised by
V0.1	30.09.2019	ToC	Yash Shekhawat
V0.8	13.12.2019	All partner inputs	All partners in WP6
V0.9	23.12.2019	First integrated document	Yash Shekhawat
V1.0	25.12.2019	Final Version	Yash Shekhawat
V1.1	18.1.2020	Reviewed and updated version	Petros Alavanitopoulos

Author list

Organization	Name	Contact Information
Nurogames	Yash Shekhawat	Yash.shekhawat@nurogames.com
Nurogames	Andrew Pomazanskyi	Andrew.pomazanskyi@nurogames.com
Nurogames	Sebastian Freitag	Sebastian.Freitag@nurogames.com
Nurogames	Lena Werthmann	Lena.werthmann@nurogames.com
CERTH	Stefanos Vrochidis	stefanos@iti.gr
CERTH	Yiannis Kompatsiaris	ikom@iti.gr
CERTH	Sotiris Diplaris	Diplaris@iti.gr
CERTH	Petros Alvanitopoulos	palvanitopoulos@iti.gr
UPF	Alexander Shvets	alexander.shvets@upf.edu
UPF	Jens Grivolla	Jens.gricolla@upf.edu
UPF	Leo Wanner	Leo.wanner@upf.edu
Up2Metric	Christos Stentoumis	christos@up2metric.com
McNeel	Ayman Moghnieh	ayman@mcneel.com
McNeel	Kostas Avgerinakis	kostas@mcneel.com
McNeel	Luis Fraguada	luis@mcneel.com
McNeel	Verena Vogler	verena@mcneel.com
MU	Alexia Briassouli	alexia.briassouli@maastrichtuniversity.nl
MU	Rico Montulet	rico@montulet.nl
Zaha Hadid	Tyson Hosmer	Tyson.Hosmer@zaha-hadid.com

Executive Summary

The document details the technical requirements and specifications of the MindSpaces platform. It discusses the architecture and the various services, End user tools and the communication methods that will be used in the platform.
The deliverable starts by introducing the tools and mechanisms used to extract and analyse these technical requirements, including documents analysed, use-cases studied, interviews carried on, and surveys conducted. This gives us an overview on how the user requirements were collected. In the following section, the deliverable focuses on the pilot use cases of the MindSpaces platform and the user requirements attached to those pilot use cases. This helps us define the technical requirements of the platform.
Then, it discusses the main technical concerns of the MindSpaces platform architecture, being its architecture design and machine-based deployment model, the logical design and elements of a MindSpaces services, the communication model, and the data management policy established in accordance with the platform's expected data processing flow.
Furthermore, the deliverable outlines the each of the service of the platform. Each of the service is introduced along with the service owner that will be responsible for implementation of the service. For each service, the system requirements to run the service, functionalities that it will be implementing, the logical design of the internal components of the service and the data flow in the service. This is done for 13 MindSpaces services including the Knowledge Base and the Data Storage.
Finally, the deliverable outlines the 3 user tools and their functionalities as well as the UI/UX of the tools where user interaction is required. We also define the components and the development method for the user tools as well as the installation requirements for the machines. The deliverable gives an overview of the technical requirements, the architecture and each of the component of the system in detail.

Abbreviations and Acronyms

ABPS	Agent Based Parametric Semiology
ACME	Automated Certificate Management Environment
AE	Aesthetic Extraction
API	Application Programming Interface
AR	Augmented Reality
BA	Behavioural Analysis
CAD	Computer-Aided Design
DS	Data Storage
EA	Emotional Analysis
EEG	Electroencephalography
FR	Functional Requirement
FURPS	Functionality, Usability, Reliability, Performance
GH	GrassHandler
GH	Gigahertz
GPU	Graphic Processing Unit
GRPC	google Remote Procedure Calls
GSR	Galvanic Skin Response
GUI	Graphical User Interface
HLUR	High-level User Requirement
KB	Knowledge Base
LSL	Lab Streaming Layer
LTSC	"Long Term Servicing Channel" by Microsoft
MDT	Mindspaces Design Tool
NFR / N-FR	Non-functional Requirement
NIC2	Neuroelectronic Instrument Controller 2
PUC	Pilot Use Case
RDF	Resource Description Framework
REST	Representational State Transfer
SDK	Software Development Kit
SSL	Secure Sockets Layer
TA	Textual Analysis
TG	Text Generation
TP	Texture Proposal
TR	Technical Requirement
UI	User Interface
UR	User Requirement
URL	Uniform Resource Locator (A website)
VR	Virtual Reality

Table of Figures

Table of Tables

Table of Contents

1 INTRODUCTION

MindSpaces project aims to create a new way to urban and architectural design using the physiological psychology and by generating 3D-VR immersive and emotion-adaptive “neuro-environments”. The MindSpaces project helps architects to use the knowledge of artists and citizens to help in the design process with physiological inputs for outdoor environments, indoor environments and inspiring workplaces. The project will demonstrate the technology in 3 Pilot Use Cases:

- Design of Outdoor urban environments
- Inspiring workplaces
- Emotionally-sensitive functional interior design

This deliverable documents the technical requirements to develop the system as well as the architecture of the system to perform the tasks for the development of the platform.

The deliverable starts by defining the Technical requirements analysis procedures. For that purposes, this deliverable starts in the first section by introducing the practice of technical requirements collection and analysis, arguing about its scope and applicability in the context of MindSpaces, before discussing the related methods in gathering requirements and those selected and applied in the context of the project such as document analysis, surveys, one-on-one interviews, use case analysis.

In the next section, we discuss the various pilot use cases of the MindSpaces project and the user requirements gathered from closely analysing the use cases and with discussions with relevant stakeholders. This provides an overview of the role performed by the platform as a service-oriented architecture geared to perform user-driven tasks. This also helps us achieve a first set of technical requirements that the system needs to be able fulfil.

In the fourth section of the deliverable we discuss the Architecture design methodology and the specifications. We start by focusing on the methodology that is used in the development of the architecture of the platform. Following that we work on identifying the elemental technologies that will be used in the system. The next sub-section deals with discussing the machine-based deployment and the three machines of the platform a) The behavioural machine; b) The VR-EEG Machine; c) The Cloud Machine. Then we focus on the communication model and the use GRPC for communication between different components in the system. Finally, this section describes the Data Model and the Data Storage of the system.

The fifth section of the deliverable documents the technical specifications of each of the platform services, this includes the functionality of the service, the requirements from the service, the logical design of the service and the data the service will handle. This is done for each of the 13 currently identified services of the platform, each performing separate tasks and addressing different requirements.

The sixth section deals with the 3 end user tools and the UI/UX of the end user tools, which include the Design tool, ABPS Generative design tool and the VR Tool.

Finally, we conclude by summarizing the findings discussed in this deliverable.

2 INTRODUCTION TO TECHNICAL REQUIREMENTS

In system engineering, the definition of the technical requirements (TRs) is essential for a smooth design, development and deployment of a system. In this section the fundamentals, principles and methods of requirements engineering is introduced. Then, the methodologies of technical requirements engineering are examined in the context of MindSpaces, aiming to highlight the need for a thorough analysis of the platform by taking into consideration all the aspects for a solid definition of the MindSpaces platform.

More specifically, the next section starts with the requirements engineering by introducing the definition of the Requirement Analysis and the purpose of the requirements. Then the principles governing the requirements assessment are presented. The analysis approach follows with the included activities and characteristics of requirements. The last section of the present chapter is dedicated to the MindSpaces context. It highlights the significance of the application of requirements analysis to the MindSpaces platform and it ends with a brief description of the methodology followed by the technical partners.

2.1 Requirements Engineering

Requirements Analysis[1] (or Requirements Engineering) "*is the process of eliciting stakeholder needs and desires and developing them into an agreed set of detailed requirements that can serve as a basis for all other subsequent development activities*". A requirement is a statement aiming to identify a product or processes operational, functional, or design feature or constraint, which is unequivocal, measurable or testable and essential for product or process acceptability. At the beginning of the requirements analysis the understanding of the system is not clear. The analysis starts with the description of initial ideas, concepts and expectations from end users and continues with the involvement of technical experts. The goal is to produce a "desired output", which is a set of detailed statements, on which all people involved agree, and serves as a basis for the kick-off of development activities. Getting to a deeper understanding of the process the principles governing systems requirements are described in the following section.

2.1.1 Principles Governing System Requirements

Relation to User Requirements and Logical Architecture

In software engineering the requirements engineering impose a clear distinction between user requirements and system (technical) requirements. First, a set of stakeholder/user statements (user requirements) written in natural language are provided and clarified in order to be understandable by users who do not have any technical knowledge. Then these statements of need are translated into technical requirements usually written in engineering-oriented language in a standard form or template. The technical requirements could be consider as an extended version of user requirements, used by software engineers to enable proper system design.

The system requirements take into consideration factors such as performance, quality and other measures having as a goal to identify the modules and functionalities of any solution system. Close coordination between stakeholders/users and engineers is indispensable in

[1] **Pohl, Klaus, and Nelufar Ulfat-Bunyadi. "The three dimensions of requirements engineering: 20 years later." Seminal contributions to information systems engineering. Springer, Berlin, Heidelberg, 2013. 81-87.**

order to produce system requirements that describe the functions that the system supports in an unambiguous, consistent and verifiable way.
The logical architecture activity also defines system boundary and functions, from which a refinement of system requirements could be elicited. The logical architecture activity decomposes the system to logical components without imposing any implementation constraint. Regardless of whether the system is an evolution of a previous product/service, or a new and unprecedented solution it is important to ensure that the user requirements, system requirements, and logical architecture are all complete and consistent with each other.

Classification of Technical Requirements
Several classifications of system requirements are possible. An example of system requirements classification based on ISO 2011[2] is summarized in the following table:

Types of System Requirement	
Functional Requirements	Performance Requirements
Usability Requirements	Interface Requirements
Operational Requirements	Modes and/or States Requirements
Adaptability Requirements	Physical Constraints
Design Constraints	Environmental Conditions
Logistical Requirements	Policies and Regulations
Cost and Schedule Constraints	

Table 1 System requirements classification

Characteristics of Requirements.
For an efficient definition of the requirements several characteristics are proposed in the literature. The following table includes a set of recommended characteristics according to INCOSE 2011[3].

Characteristic	Description
Necessary	The requirement defines an essential feature. If it is not included in the list of requirements, it causes the deficiency of conformance with regards to the need of a desired characteristic of the system, defined by users.
Unambiguous	The requirement is written in a way that produces a single and the same interpretation to all readers.
Complete	The requirement has to be sufficient without the need for additional information in order to understand it.
Singular	The requirement should describe a single need.
Feasible	The requirement has to take into account potential constraints with acceptable risk
Verifiable	The requirement has to be written in a such way that its realization can be

[2] ISO/IEC/IEEE. 2011. Systems and Software Engineering - Requirements Engineering. Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission/ Institute of Electrical and Electronics Engineers (IEEE), (IEC), ISO/IEC/IEEE 29148.
[3] INCOSE. 2011. Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, version 3.2.1. San Diego, CA, USA: International Council on Systems Engineering (INCOSE), INCOSE-TP-2003-002-03.2.1.

	measurable or testable in order to verify it.
Correct	The requirement must be an accurate representation of the entity need from which it was transformed.
Conforming	Each requirement has to conform to an approved standard template and style for writing requirements, when applicable.

Table 2 Characteristics of requirements

2.2 Requirements analysis in MindSpaces

MindSpaces is a distributed system. Several types of data will be collected by a set of sensors. Moreover, multiple services will also be implemented to support the intelligent analysis of raw signals from the sensing environments. Cloud based and machine-based deployment are taken into consideration. The MindSpaces platform is a service oriented platform. GRPC[4] communication is utilized for the communication between the modules of the system. MindSpaces platform includes different types of storages. A SOLR Database will be used to create the discourse and emotions graphs. The SOLR instance's API will handle the communication between these graphs. A MongoDB will be examined to store the 3D models and other data which can be accessed by the KB and other components of the system through GRPC requests. The platform integrates several user tools for EEG information gathering, for designers and other visualisations that together represent the platform's user interface. These tools communicate with the platform using GRPC to the Knowledge Base and then to the central data storage. A distributed system design is not a straightforward task especially when it involves many factors that impact the architecture stability, scalability and performance in the long run and with multiple users. The MindSpaces platform is a complex system. Thus, the importance of a complete, consistent, validated and agreed set of technical requirements is self-evident. The purpose of the requirements analysis in MindSpaces aims to:

1) Develop a document for the definition of technical requirements and specifications that can push off the development process and can be agreed upon by all the stakeholders.
2) Define the tasks that the platform will perform for the users and any concerns, both technical and non-technical about it.
3) Define the functionalities for each component of the system
4) Define the constrains and the implications of the architecture of the system and evaluate any such constraints for the integration of the components.
5) Define the KPIs of the platform.

A close collaboration between the users-partners of the consortium and the technical partners was necessary. A set of activities are undertaken during the requirement analysis process in the MindSpaces project. These activities include:

- Analyzing the user requirements to check completeness of expected services and operational scenarios, conditions, operational modes, and constraints.
- Defining the technical requirements and their rationale.
- Classifying the technical requirements
- Incorporating the derived technical requirements (coming from architecture and design)

[4] https://grpc.io/docs/

- Establishing the upward traceability with the user needs and requirements.
- Validating the content and relevance of each system requirement against the set of stakeholder requirements.
- Identifying potential risks

Technical requirements specification is an indispensable process that greatly facilitates the platform development. Moreover, it is useful for users and the development team. The former define their need with accuracy and the latter understand the characteristics that have to be supported. Investing time in defining the technical requirements reduces future work, since this action is the reference and the guide for a successful development. Good practices lead to good documentation. In MindSpaces our approach for gathering the requirements includes:

- Document Analysis
- Use cases and user stories
- A survey for the definition of the core modules of the system
- Interviews
- Drafts of System Architecture

Following the document analysis, an initial outline of the technical specifications is documented. From the examination of the DoA and the technical specifications sheet that was shared in the wiki site the system components and their functionalities were identified.
Following the user stories and use cases approach, some additional functional requirements are obtained. Such requirements capture initial presumptions of the users towards future platform, which ideally shall be met in full and evaluated at later stages. Moreover, risks and the related impact related to cost, schedule and technical constraints are also examined.
Following the component definition survey, which was introduced upon the project initiation phase, offers a comprehensive definition of each subsystem envisioned within MindSpaces architecture. This helps in defining the owners for each of the components and address the specific technical specifications. The survey also helps in detailing the concept of each of the components, like the logical design, data handling capacity, the requirements and the functionalities. This gives us a starting point in relation to the technical requirements and how they are being met.
Furthermore, in order to get more information and address more specific questions with respect to component-level requirements, a number of interviews have been performed with tools and services owners in an unstructured way using open-ended questionnaires. These interviews were useful to complete the requirements gathering exercise at this stage in the project.
The architecture diagrams that were also produced and discussed in the meetings and the ethical discussions speeded up the identification of constraints and helped us in a proactive way to include new functionalities and consider the performance issues.
A combination of these methods allowed the consortium to gather a variety of technical requirements (TRs) for the platform that pertains a list of functional and non-functional, macro- (generic) and micro- (specific) level requirements, as well as those related to the platform as a whole, individual subsystems, components and their services. Nevertheless, it is acknowledged that future re-evaluation and further elicitation on the requirements will be required throughout the implementation of the project. The requirements analysis is an essential step that has to be revisited after the evaluation of the platform and before the start of the next iteration of the development phase. Following this approach our goal is to

meet the A crucial milestone though would be to perform such activities after initial platform evaluation deemed to verify that incremental versions of MindSpaces platform meet expectations of the end users and various stakeholders.

Throughout the project technical development, requirements determined within the elicitation process shall serve as a handbook towards eventual evaluation. Each development iteration shall be governed by the mentioned guidelines set in such handbook. This in turn can be achieved by isolating **platform level requirements** (which include mostly architecture design specifications) and **component level requirements** (simplistic functions of subsystems)[5], and thus independent teams will be able to set apart individual components and work on them.

[5] https://www.eso.org/sci/facilities/eelt/owl/Blue_Book/5_System_engineering.pdf

3 PILOT USE CASES AND TECHNICAL REQUIREMENTS

There are three pilot use cases that we plan to implement during the project lifetime. The user expectations are related to these three PUCs. In the following sections, we briefly describe the pilot use cases analysing their direct implications for the platform in terms of requirements, especially what pertains to the user expectations and the role of the platform in the pilots.

3.1 MindSpaces pilot use cases

In the following section, we offer a summarized description of the MindSpaces pilot use cases and the High-level User Requirements (HLURs):

PUC 1: Outdoor Urban environment

The pilot use case for “Designing of improved, attractive city spaces”, intents to improve urban design in a rapidly expanding city by addressing new challenges that may arise related to the city’s functionality, mobility, attractiveness, protection of culture and environment. MindSpaces will aim to increase sensitivity and awareness towards the cultural significance and current issues of the city, related to the environment and mobility, through innovative art installations in key locations. The HLURs for the PUC, as noted down in D7.2, are the following:

HLUR	HLUR Title	HLUR Description
HLUR_1.1	User interaction and control	Architects can collect onsite, geolocate and aggregate biometric/behavioural data in 3D reconstructed environments.
HLUR_1.2	Manipulation of spatial conditions	Architects and artists can use spatial conditions to increase social interactions and communicate artistic concepts
HLUR_1.3	Data Analysis for understanding social needs and human values	Artists can use data Analysis for understanding social needs and human values
HLUR_1.4	Adaptable public spaces	Citizens can have to have adaptable public spaces depending on their needs.
HLUR_1.5	Space use prediction	An architect/designer can predict the potential uses for new spaces by analysing previous behavioural data.
HLUR_1.6	Intelligent projects based on feedback	An architect/designer can produce social intelligent projects based on feedback (emotional and rational: opinion on the internet)

Table 3 High-level user requirements for PUC 1, D7.1 section 3.1.5

PUC 2: Inspiring workspaces

Inspiring Workplaces is a Pilot Use Case designed to test and develop the MindSpaces platform specifically for designing better quality workplace environments. MindSpaces research partners collect and analyse behavioural, emotional, and textual data from people inhabiting workplace environments physically and virtually (AR/VR environments) to develop design and analysis tools used in designing workplaces. Artists, architects, and designers will leverage the tools and data insights to explore and envision improved workplace environments. The HLURs for the PUC, as noted down in D7.2, are the following:

HLUR	HLUR Title	HLUR Description
HLUR_2.1	User interaction and control	Architects can collect onsite, geolocate and aggregate biometric/behavioural data in 3D reconstructed environments
HLUR_2.2	Correlation of a user's space attributes with their emotional state	An architect can correlate spatial quality and environmental attributes of space with the emotional state and behaviour of the users.
HLUR_2.3	Adaptable workplaces	An office worker can have an adaptable workplace depending on their needs
HLUR_2.4	Data Analysis for understanding social needs and human values	An artist/designer can use Data Analysis for understanding social needs and human values through social interaction with spaces
HLUR_2.5	Space use prediction	An architect/designer can predict the potential uses for new spaces by analysing previous behavioural data.
HLUR_2.6	Intelligent projects based on feedback	An architect/designer can produce social intelligent projects based on feedback (emotional and rational

Table 4 High-level user requirements for PUC 2, D7.1 section 3.2.5

PUC 3: Emotionally-sensitive functional interior design

Architectural and interior design has always aimed at creating emotionally appealing and functional environments. But it is only in recent years that emotional effects and the usability/functionality of a designed space are being assessed in an objective and quantifiable manner. Quantitative data using multiple sensors are now showing great potential to support design. Additionally, the widespread availability of digital representations of past aesthetic trends and features enables their innovative re-use and integration in new designs. The HLURs for the PUC, as noted down in D7.2, are the following:

HLUR	HLUR Title	HLUR Description
HLUR_3.1	User interaction and control	Architects can collect onsite, geolocate and aggregate biometric/behavioural data in 3D reconstructed environments
HLUR_3.2	Manipulation of spatial conditions	Architects and artists can use spatial conditions to increase social interactions and communicate artistic concepts
HLUR_3.3	Data Analysis for understanding social needs and human values	An artist/designer can use Data Analysis for understanding social needs and human values through social interaction with spaces
HLUR_3.4	Adaptable houses	A senior can have an adaptable house depending on their needs
HLUR_3.5	Space use prediction	An architect/designer can predict the potential uses for new spaces by analysing previous behavioural data

Table 5 High-level user requirements for PUC 3, D7.1 section 3.3.5

Each of the PUC has defined several high-level user requirements from the initial definition from the partners involved in the PUCs. These high-level requirements correlate with more elementary requirements that describe specific functions and functionalities that ought to be achieved by the system due to the broad nature of high-level user requirements, these can share several elementary requirements, or in other words each elementary user requirements can be associated with one or more high level requirements.
The complete set of identified and described user requirements are documented in D7.1. The user requirements are further categorized using the MoSCoW framework[6]. The must-haves and the should-haves from the categorization of the requirements are the basis of the Minimal Viable Product for the platform. We therefore qualify these essential requirements as critical and important and prioritize the assessment and evaluation of the developed components and services accordingly.

3.2 Technical Requirements Overview

This section documents all the Specific User Requirements (UR) identified based on the HLURs identified by the users. The URs are then distinguished as a Functional or a Non-Functional requirement based on the classification Software Engineering principles[7].

[6] https://www.productplan.com/glossary/moscow-prioritization/
[7] https://www.guru99.com/functional-vs-non-functional-requirements.html

TUR	Associated HLUR	Detailed description	Functional or Non-Functional (FR/N-FR)
UR_1	HLUR_1	As an architect I want to be able to connect parametrized models with live data streams and follow their morphing process in different conditions	FR
UR_2	HLUR_1	As an architect I want to reconstruct a 3D environment in order to contextualize my work in realistic settings	FR
UR_3	HLUR_1	As an architect I want to be able to apply aesthetic and structural transformations or modifications on the reconstructed 3D environment	FR
UR_4	HLUR_1	As an Architect I want to acquire biometric and behavioural data from subjects thought onsite experiments, through sensor-based data acquisition mechanisms	FR
UR_5	HLUR_1	As an Architect I want to be able to geolocate the biometric and behavioural data in virtual space	FR
UR_6	HLUR_1	As an Architect I want to be able to aggregate biometric and social media/biographies data by subject, subject groups, location, and any possible configuration of meaningful parameters	FR
UR_7	HLUR_1	As an Architect/artist I want | to have a self-explanatory system that would describe what changes and alterations were applied in VR and why, for their empirical evaluation and studying of correlations between behavioural and emotional patterns, and visual, aesthetic, and structural aspects	FR
TUR-8	HLUR_1	As an Architect I want to identify chokepoints, stress points, and other focal points of an environment by analysing biometric and social media/biographies data acquired from subjects dwelling in this environment	FR
UR_9	HLUR_2	As an Architect/ artist/ designer I want to have a gist of a particular aspect or topic summarized based on opinion data from the internet	FR
UR_10	HLUR_2	As an architect I want to correlate material palettes and colours with the emotional state of users in designed workplace environments	N-FR
UR_11	HLUR_2	As an architect I want to correlate the amount / quality of light with the behaviour and/or emotional state of users in designed workplace environments	N-FR

UR_12	HLUR_2	As an architect I want to correlate the quantity and location of entry points, walls, tables, chairs, desks, and other architectural features with the behaviour and/or emotional state of users in designed workplace environments	N-FR
UR_13	HLUR_2	As an architect I want to correlate the ceiling / spatial height with the emotional state and/or behaviour of users in designed workplace environments	N-FR
UR_14	HLUR_2	As an architect I want to correlate the size/shape of a personal working desk/space with the emotional state and/or behaviour of users in designed workplace environments	N-FR
UR_15	HLUR_2	As an architect I want to correlate amenities with the emotional state and/or behaviour of users in designed workplace environments	N-FR
UR_6	HLUR_2	As an architect I want to correlate the location / type of green or exterior space with the emotional state and/or behaviour of users in designed workplace environments	N-FR
UR_17	HLUR_2	As an architect I want to understand which spatial conditions enable and encourage more effective social interactions and collaborations	N-FR
UR_18	HLUR_2	As an artist I want to be able to use spatial, material and time dimension to communicate artistic concepts	N-FR
UR_19	HLUR_2	As an artist I want to be able to use VR as a simulation environment to test new designs	FR
UR_20	HLUR_3	As an artist I want to able to use data (sensors, camera, internet) to understand social needs	FR
UR_21	HLUR_3	As an artist I want to be able to rethink social systems and human values through social interaction with spaces	N-FR
UR_22	HLUR_4	As a citizen I want to feel comfortable in an open urban public space	N-FR
UR_23	HLUR_4	As a citizen I want to be able to interact socially and maybe technologically with confidence in public space	N-FR
UR_24	HLUR_4	As a citizen I want to empower myself being conscious of my role of citizen shaping and giving sense to open public space	FR
UR_25	HLUR_4	As a citizen I want to see open space as a gate to discover culture and identity	FR
UR_26	HLUR_4	As an architect/designer I want to be able to design/provide for public authorities safe and nice open public spaces	FR
UR_27	HLUR_4	As an architect/designer I want to be able to design spaces that show cultural diversity and richness	FR

UR_28	HLUR_4	As a public authority I want to use public spaces to show public policies	FR
UR_29	HLUR_4	As an office Worker I want a platform to support design parameters for productive collaboration	N-FR
UR_30	HLUR_4	As Worker in an office I want to be more healthy, happy, calm, and productive at work	N-FR
UR_31	HLUR_4	As an office Worker I want to feel I have privacy while not feeling isolated and working close to productive colleagues	N-FR
UR_32	HLUR_4	As an office Worker I want to easily communicate effectively with my colleagues	N-FR
UR_33	HLUR_4	As an office Worker I want access to light and green space	N-FR
UR_34	HLUR_4	As an office Worker I want to understand how to navigate and use a space	N-FR
UR_35	HLUR_4	As an office Worker I want to feel I am in a contemporary and high-quality space	N-FR
UR_36	HLUR_4	As an office Worker I want to not feel bored by the repetition of working	N-FR
UR_37	HLUR_4	As an office Worker I want to maintain a high energy level at work while reducing stress	N-FR
UR_38	HLUR_4	As an office Worker I want to be motivated and productive	N-FR
UR_39	HLUR_4	As a senior I want to communicate with my entourage	FR
UR_40	HLUR_4	As a senior I want to have the opportunity to socialize/feel socially included	N-FR
UR_41	HLUR_4	As a senior I want to experience positive and empowering feelings/emotions	N-FR
UR_42	HLUR_4	As a senior I want to experience a space adapted to potential impairments (visual, hearing, mobility)	N-FR
UR_43	HLUR_4	As a senior I want to have the technological support adapted to my visual or hearing impairments (accessibility of the technology)	FR
UR_44	HLUR_4	As a senior I want to experience an aesthetical pleasant space	Non-FR
UR_45	HLUR_4	As a senior I want to experience a space that evokes positive memories	Non-FR
UR_46	HLUR_4	As a senior I want to feel closeness/proximity/presence/ to repel the feeling of loneliness	Non-FR
UR_47	HLUR_4	As a senior I want to feel comfort (physically) and comforted (emotionally)	Non-FR
UR_48	HLUR_4	As a senior I want to remain independent as long as possible	Non-FR
UR_49	HLUR_4	As a senior I want Privacy by Design	FR

UR_50	HLUR_4	As a senior I want a space responding to my affective and intimacy needs	N-FR
UR_51	HLUR_4	As a senior I want to control the transmission of information (coming from my set and getting to my set)	FR
UR_52	HLUR_4	As a senior I want my living space to be bright	N-FR
UR_53	HLUR_4	As a senior I want to be connected to nature (i.e.: plants) when in my living space	N-FR
UR_54	HLUR_4	As a senior I want that my living space inspires my creativity	Non-FR
UR_55	HLUR_4	As a senior I want to keep objects from the past at home	Non-FR
UR_56	HLUR_4	As a senior I want to feel safe at home	Non-FR
UR_57	HLUR_4	As a senior I want my living space to inspire/be adapted to/ physical activity practice	Non-FR
UR_58	HLUR_4	As a senior I want to my living space to be inspired/be adapted to/ physical activity practice	Non-FR
UR_59	HLUR_4	As a senior I want to be able to welcome people in my house	Non-FR
UR_60	HLUR_4	As a senior I don’t want to be cluttered by too many objects in my living space	Non-FR
UR_61	HLUR_4	As an architect/Designer I want to have the capacity to incorporate data-driven behavioural and emotional analysis to workplace wellbeing.	FR
UR_62	HLUR_4 HLUR_5	As an Architect/Designer I want to be able a process that can be easily pivoted to other domains where managed semi-public spaces are central.	FR
UR_63	HLUR_4	As an Architect/Designer I want highly flexible models compared with approaches using fixed models of wellbeing.	FR
UR_64	HLUR_5	As an architect/designer I want to predict use of the new spaces by previous behavioural data	FR
UR_65	HLUR_6	As an architect designer I want to be able to produce social intelligent projects based on feedback (emotional and rational: opinion on the internet)	N-FR
UR_66	HLUR_6	As an architect/designer I want Intelligent concept extraction taking advantage of both linguistic and statistical parameters	FR
UR_67	HLUR_7 HLUR_4	As an office worker I want a collaborative tool for collective spaces where co-workers can work simultaneously on the same tasks.	FR
UR_68	HLUR_10	As an architect I want an easier way to develop 3D environments than traditional tools	FR
UR_69	HLUR_7	As citizen/office worker I want a software that can help me communicate/interact with other citizens/co-workers	FR

UR_70	HLUR_11	As an architect/Designer I want to be able to create novel and inspiring textures that are based on the aesthetics of famous paintings and other images of artwork that do not exist in current 3D modelling market.	FR
UR_71	HLUR_12	As an architect/designer I want to be able to extract the aesthetics of design structures (i.e. interior objects, buildings, materials etc.) and have them as a gallery.	FR
UR_72	HLUR_8	As an architect/designer I want a simple and clear visual UI (User Interface). Simple enough for non-specialised users	FR
UR_73	HLUR_9	As an Architect/designer I want to have a tool that can assist in formulating new, innovative architectural ideas	FR
UR_74	HLUR_9	As an architect I want to be able to have Intelligent concept extraction taking advantage of both linguistic and statistical parameters	FR
UR_75	HLUR_7	As an Architect/Designer I want to Easily share, reuse and configure implemented framework.	N-FR
UR_76	HLUR_13	As an architect I would like to be able to define/change the topography of an urban space in MindSpaces platform	FR
UR_77	HLUR_13	As an architect I would like to be able to define/change the material of an urban/interior space in MindSpaces platform	FR
UR_78	HLUR_13	As an architect I would like to be able to define/change the light of an interior space in MindSpaces platform	FR
UR_79	HLUR_13	As an architect I would like to be able to introduce the user movements to the platform	FR
UR_80	HLUR_13	As an architect, I would like to be able to introduce the environmental sensor input to the platform.	FR
UR_81	HLUR_13	As an architect, I would like to be able to introduce Environmental/Climatic (latitude, average temperature, nearby water, humidity, heating/cooling) input to the platform.	FR
UR_82	HLUR_13	As an architect I would like to be able to have the historic context of the urban space	FR

Table 6 User Requirements Identified

These user requirements are translated to the Technical Requirements (TRs) to develop a specific plan for KPIs that the system should fulfil and the implementation plan of the technical Requirements. First set of Technical Requirements analysed from the User Requirements are the following:

TRs	Related URs
TR 1. Create a colour palette generator service.	UR_3, UR_7, UR_12, UR_71
TR 2. Create a style transfer technique in order to change the style of 3D objects	UR_3, UR_7, UR_70, UR_71

TR3. Collection of visual data from video cameras to support the visual behaviour analysis	UR_4, UR_5, UR_11, UR_16, UR_20, UR_80
TR 4. Create a crawling service to gather online content from social media	UR_6, UR_9, UR_20, UR_65, UR_72
TR 5. Create a scrapping service to gather online content from websites	UR_6, UR_9, UR_20, UR_65, UR_72
TR 6. Acquire biometric data from subjects through light weight biometric devices (EEG, GSR)	UR_4
TR 7. Create an emotional state recognition service in order to recognize the emotional states of the users	UR_8, UR_9, UR_10, UR_11, UR_12, UR_13, UR_14, UR_15, UR_16, UR_30
TR 8. Create a knowledge base that will fuse multimodal data and unify them into one ontological model	UR_5, UR_10, UR_11, UR_12, UR_13, UR_14, UR_15, UR_16, UR_51, UR_61, UR_62
TR 9. Create a Reasoning Service providing the changes that are wanted to occur	UR_24, UR_41, UR_42, UR_44, UR_50, UR_57, UR_58, UR_64
TR 10. Design environments and artistic or architectural interventions	UR_2, UR_10, UR_11, UR_12, UR_13, UR_14, UR_15, UR_16, UR_65, UR_66, UR_73, UR_76, UR_787 UR_78
TR 11. Define and simulate environmental transformations	UR_3, UR_6, UR_10, UR_11, UR_12, UR_13, UR_14, UR_15, UR_16, UR_17, UR_18, UR_63, UR_64, UR_66, UR_76, UR_77, UR_78, UR_79
TR 12. Visualize the emotional and behavioural maps of the environment	UR_4, UR_5, UR_8, UR_!7, UR_18, UR_20, UR_61, UR_63, UR_73, UR_79, UR_80, UR_81
TR 13 Simulating changes and user experiences	UR_1, UR_4, UR_5, UR_7, UR_8, UR_10, UR_11, UR_12, UR_13, UR_14, , UR_15, UR_16, UR_!9, UR_61, UR_63, UR_73
TR 14 Simulating human behaviour	UR-8, UR_11, UR_12,UR_14, UR_15,UR_16,UR_17,UR_29,UR_30,UR_31, UR_32,UR_33,UR_34,UR_36,UR_37,UR_38, UR_61, UR_63,UR_64, UR_69,UR_79
TR 15 Visualizing behavioural simulation data	UR-8, UR_11, UR_12,UR_14,UR_15,UR_16,UR_17, UR_29,UR_30,UR_31,UR_32,UR_33,UR_34,UR_36,UR_37,UR_38,UR_61, UR_63,UR_64,UR_69,UR_79
TR 16. Create semantic parsing technique to capture the underlying semantics of the textual material	UR_6, UR_8, UR_20, UR_65, UR_72, UR_82

TR 17. Create concept extraction technique to support the contextual interpretation within the reasoning	UR_6, UR_8, UR_20, UR_65, UR_66, UR_72, UR_74, UR_82
TR 18. Create techniques for the projection of linguistic structures onto formal abstract representations that will be factored into knowledge graphs	UR_6, UR_8, UR_20, UR_65, UR_66, UR_72, UR_74, UR_82
TR 19. Adapt existing sentiment analysis tools to analyse emotions in citizens' contributions to social media and opinions in online content on websites	UR_6, UR_8, UR_20, UR_65, UR_72, UR_82
TR 20. Create a technique of the projection of ontology constructs to respective lexicalized semantic structures to support knowledge-driven content selection	UR_7, UR_9, UR_10, UR_11, UR_12, UR_13, UR_14, UR_15, UR_16, UR_17, UR_19, UR_21
TR 21. Create a linguistic generation service for the realization of acquired knowledge as natural language sentences	UR_7, UR_9, UR_10, UR_11, UR_12, UR_13, UR_14, UR_15, UR_16, UR_17, UR_19, UR_21
TR 22. Generate workplace environments semi-autonomously from user defined objectives, constraints, and parameters	UR_2, UR_10, UR_11, UR_12, UR_13, UR_14, , UR_15, UR_16, UR_17, UR_29, UR_30, UR_31, UR_32, UR_33, UR_34, UR_35, UR_36, UR_37, UR_38, UR_62, UR_65, UR_66, UR_68, UR_72, UR_73, UR_77, UR_78

Table 7 TRs v1.0

4 ARCHITECTURE DESIGN METHODOLOGY AND SPECIFICATIONS

The Mindspaces platform is a system that brings together a series of concerns, from across different disciplines and practices. It is destined to support artists and architects to plan their interventions in private, semi-public, and public spaces. For that purpose, it acquires data from ***scanning and mapping equipment***, to recreate and reconstruct environments in digital format. It also connects to ***online sources*** to acquire knowledge about the discourses associated with specified concerns. It includes ***design platforms*** that allow to create elaborate CAD-based designs for the conceived interventions, as well as ***EEG analysis*** *in order to* enable the analysis of emotions experiences in specific design settings[8]. The MindSpaces system also connects to on-site cameras and capture devices to ***analyse the behaviour*** of people in the targeted space.

This ecosystem of concerns that the platform brings together is represented in the following block diagram.

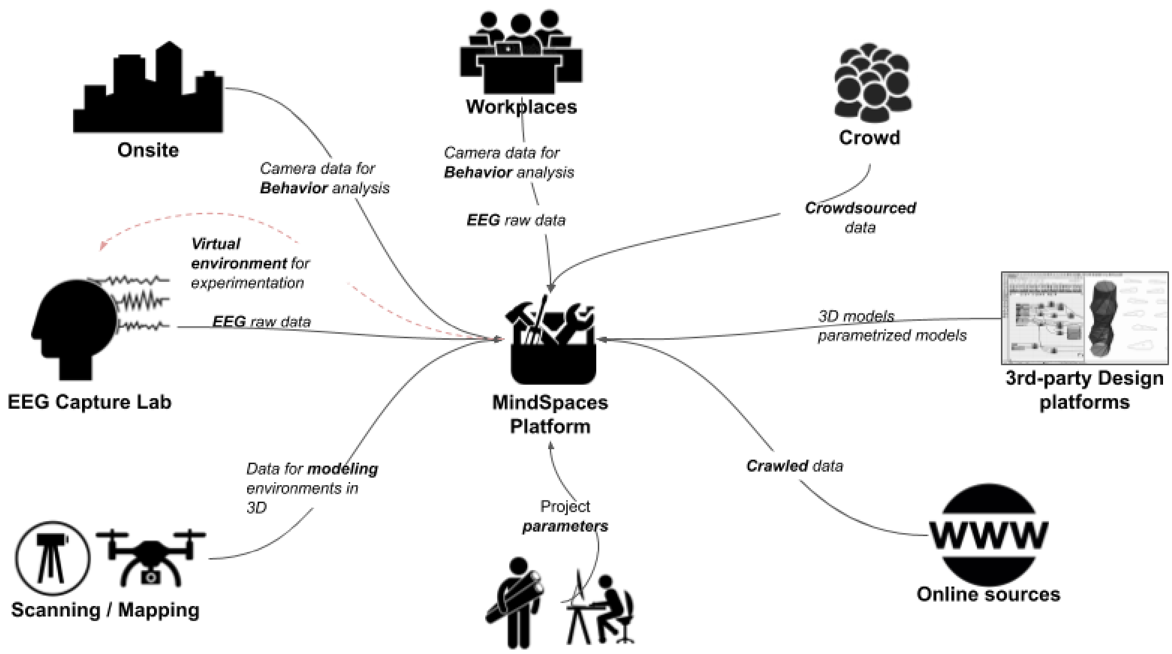

Figure 1 MindSpaces Platform Overview

In this chapter, we introduce the preliminary design of the Mindspaces platform. This complex design brings together different technologies and practices and connects them in a coherent manner that corresponds to the expected role and functionality of advanced technologies in supporting the design of artistic and architectural interventions.

In order to create this architecture model, a bottom-up approach was followed, starting from identifying and describing the elemental modules that form the basic services of a service-oriented-architecture. Following this, a machine design exercise facilitates the separation of concerns in order to consolidate each main aspect of the platform. Consequently, four conceptual machines were defined and analysed, addressing behavioural

[8] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3990628/

analysis, EEG-VR experimentation, design, and generic services as four separate concerns. Following this exercise, the technologies were analysed from the viewpoint of deployment, or in other words, under the envisioned usage scenarios. The knowledge and information gathered facilitated the conceptualization of the first version of the architecture, which was consequently described through its main pipelines in this chapter.

4.1 Methodology for designing the architecture

In order to develop the architecture of the Mindspaces platform, a method has been devised that follows a bottom-up approach in defining the elements of this platform, chaining and connecting them together in order to create an integrated system.

This approach consists of the following five steps:

- **Technology definition survey**: a questionnaire-based survey circulated within the technology consortium partners and gather their feedback in order to identify the technologies that will form part of the platform and inquire about their use, give feedback about their technical and system requirements, as well as their data management aspects, internal architecture and other concerns.
- **Technology classificatio**n: once the technologies were identified and described clearly in the preceding step and classified according to technological and system criteria that tried to locate the "region" in the architecture relevant to each technology. As if a puzzle piece is placed in the general area where it belongs, other pieces and service combinations were also examined.
- **Machine design**: at this step, components and technologies that belong to the same region of the architecture and generally concerned with performing complementary roles for a common objective, were grouped and tightly integrated into a "machine design".
- **Deployment model**: when the machine design was consolidated, a deployment model was developed by retracing the pipelines implemented by the platform according to specific use cases and functional requirements. In this step, machines were broken back to their original constituents in order to allow a more flexible and efficient deployment model. Accordingly, some aspects were centralized, common concerns were translated to middleware components, while some services were hosted on their own servers, and some modules other joined into a single component.
- **Architecture design**: the final step of this methodology consists of a consolidation of the architecture design obtained in previous steps, first by revisiting the requirements and making sure that critical requirements were met, and important functionalities were streamlined. In addition, flexibility in the design was introduced by allowing the architecture to evolve iteratively, and morph into paradigm that will meet the project objectives.

The following diagram represents this methodology:

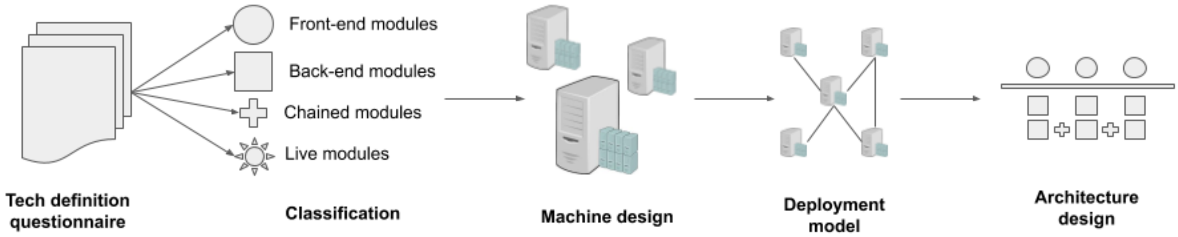

Figure 2 Architecture Design Methodology

The implementation of this approach requires a collaborative effort from the partners involved in building the platform. This was facilitated through frequent meetings and coordination activities (e.g. weekly teleconferences with use and technical partners, meetings to setup the sensors in demonstration places, etc.).

4.2 Identifying the elemental technologies

The technology definition survey revealed that the main components of the platform, which were designated as services, each an independent entity with interfacing capacities. Services were defined as black boxes with specific input/output interfaces, and constraints regarding their processing time and overall behaviour in the platform.

Accordingly, the following three services implemented the main mechanisms for data acquisition in the platform:

- **EEG Acquisition and Analysis service**: responsible for processing EEG signals from EEG devices worn by the subjects. It relays emotional tags back to other modules in 3-to-5 seconds intervals.

- **Crawling service**: responsible for executing queries on social media and online web sources to acquire data relevant and identify the discourse and emotions related to specific topics set by the user[9].

- **Camera footage acquisition service**: responsible for interfacing with several kinds of capturing human behaviour patterns, such as RGB, depth and thermal cameras, deployed in a specific space and acquiring footage from them in order to be processed by other components.

The following services constitute a second layer of components responsible for structuring and processing the data acquired:

- **Knowledge Base (KB) service**: acts as the intelligence hub of the platform, analysing the significance and relations of different signals and their metadata acquired in order to make automated sophisticated decisions to change aspects in the virtual

[9] https://en.wikipedia.org/wiki/Web_crawler

environments supported by the platform and produce the desired crowdsourced immersive environments.

- **Behavioural analysis service**: responsible for analysing the behavioural patterns of people dwelling in the spaces being monitored by the platform. The service would be able to work in different modes, including live analysis.
- **Text analysis service**: performs and in-depth semantic analysis on the data acquired from social media, questionnaires, and other textual materials in order to discern the related discourse.

The following services provide necessary support to the design processes that are supported by the platform, in particular the intended artistic or architectural interventions:

- **Style transfer service**: consists of a predetermined list of emotion-related style transformations that can be applied on any 3D model whose original set of scan images are available.
- **Aesthetic analysis service**: generates colour palettes from artwork to reflect specific emotions. The palettes are used as a design resource to style and change the aesthetics of spaces in order to evoke specific emotions while inhibiting others.
- **Language generation service**: generates reach textual descriptions of abstract concepts and allows designers to understand and interpret complex configurations of associated data and metadata fields.
- **GrassHandler service**: connect the design tools with live and offline data streams, creates visual structures to represent abstract datasets, and implements a series of CAD transformation in a manner that suits the purpose of the project.

The following components are tools defined to provide interfacing capacities for the user to work and operate the platform:

- **VR tool**: the virtual reality tool is the core component of the VR-EEG experimental setting envisioned in the project, in which the artists and architects can test different configurations of the design space and change them automatically and in a parametrized way in order to maximise specific aspects of the experience being created.
- **Design tool**: the main purpose of the design tool is to support artists and architects in the creation of the parametric design space of projects, the configuration and the simulation of the VR-EEG experiment, and the monitoring of virtual and physical settings related to each project.
- **ABPS Generative Design and Behavioural Simulation Tool**: This tool allows the user to conduct elaborate simulations to understand how the behaviour of the space's occupants is affected by changes in the environment and provides semi-autonomous generative design of workplaces from user defined parameters and constraints. It supports agent-based simulations with predictive capacities.

- **Cloud interface**: the platform also has a web-based interface that allows users to address functional aspects unrelated to design, monitoring, and experimentation, such as seeding the crawler and exploring the discourses and emotional graphs mined from social media.

In addition to these services and components, a **centralized Data Storage (DS)** component is conceived to allow services and tools to exchange datasets and files among themselves, and also to allow users to upload their assets to the platform and use them in their projects.

Together, these fourteen components form the core modules of the Mindspaces platform and address all the main technical concerns related to the objectives and requirements of Mindspaces system. It is worthwhile noting that scanning and virtualizing physical spaces, namely for the project's use cases, is considered an ad-hoc activity that is not implemented as a core service of the platform. This is due to the fact that such endeavour is of a manual nature, and requires human intervention. It is supported by a set of off-the-shelf devices, methods, and algorithms that do not need to be incorporated in the platform. With that in mind, the manner by which these spaces are virtualized should correlate with the platform's technical requirements, with respect to 3D models and segmented 3D spaces.

4.3 MindSpaces machine based deployment

In order to maintain flexibility in the architecture design, a machine-based deployment has been provided. Services that acquire data from online sources, and services that only acquire data from the data storage and interface with the knowledge base, are hosted on the "**cloud machine**". EEG signal acquisition and analysis, together with the VR tool are hosted on the "**VR-EEG machine**", which is consequently deployed in EEG labs to conduct experiments on subjects. Services that support the creation, simulation, and monitoring of the user experience are hosted on the "**design machine**". Finally, services related to monitoring physical spaces through cameras, and behavioural analysis components are hosted on the "**behavioural machine**".

The following figure represents the four machines defined as part of the architecture exercise:

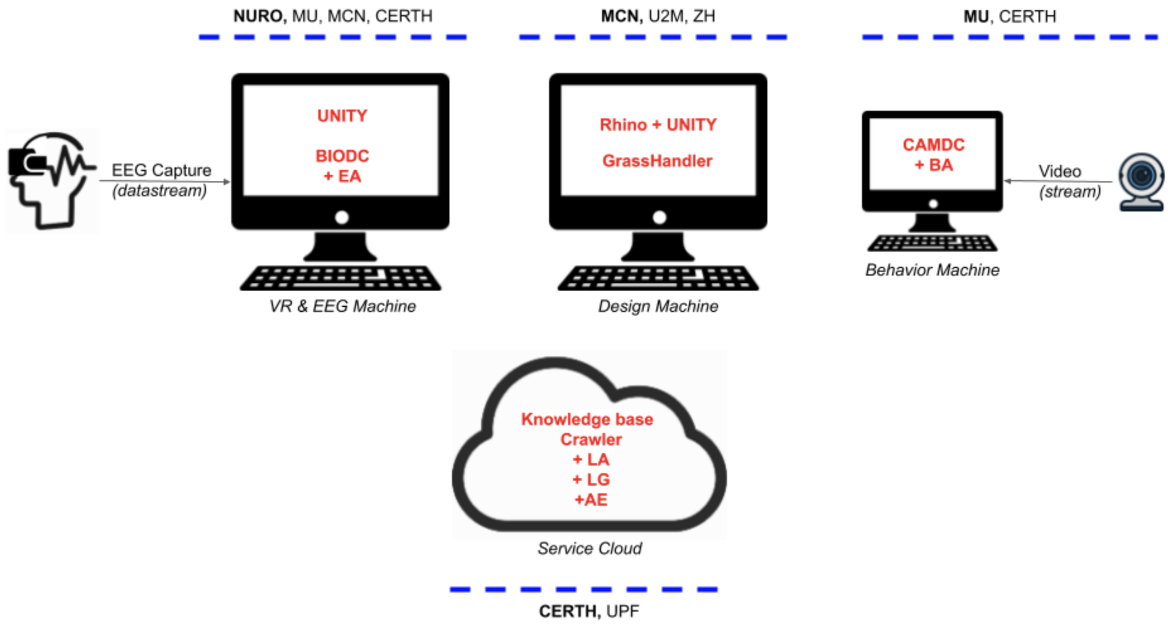

Figure 3 MindSpaces Machines

This separation of services is intended to facilitate the consolidation of each of these four applications within the Mindspaces platform, with a certain degree of independence. Interoperability between machines is tackled in a manner that centres on defining the data and request interfacing mechanisms. The machines are not necessarily physical machines, and the manner by which components and services are deployed in-situ, in the cloud, and on remote servers is subject to technical requirements having to do with integration, overall performance, pipeline design, and other aspects of the architecture.

4.3.1 The behavioural machine

Arguably, the simplest machine among the four is the behavioural machine since it groups a small number of components: the camera footage acquisition service, and the behavioural analysis service. The machine connects to one or more cameras simultaneously (if needed) and acquires footage in both live mode and download mode (from the cameras' internal storage). Video footage is then run through an anonymization algorithm, which will involve removal of identifying information, depending on the data in each use case. Ultimately, the raw footage will be transmitted in a protected way to the server to be processed internally, while only non-identifying metadata will be produced (e.g. tags indicating "new event" or "high level of human interaction in this area of the recorded scene"). The footage is stored on a local storage inside the machine where the behavioural analysis is installed, and is then analysed by the Behavioural Analysis service, which sends behavioural tags over the network to other machines.

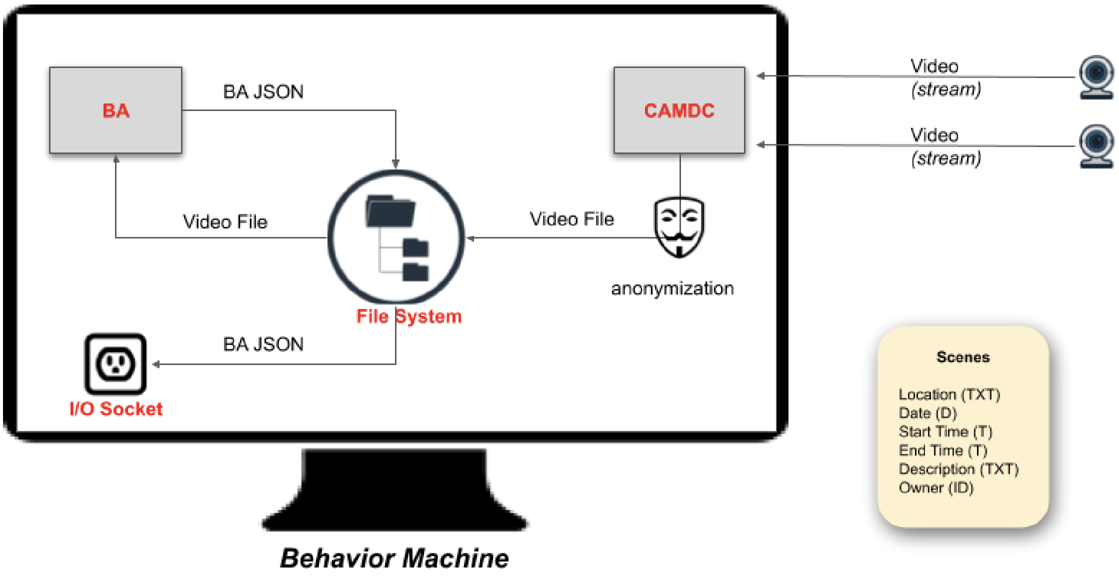

Figure 4 Behaviour Machine

For the first version of the platform, the behavioural machine is not expected to function in live mode, although decisive steps will be taken to consolidate this functionality for future versions. In addition, for the first version, the behavioural analysis data will be retrieved for a specific scene (recording session in a given space for a determined and continuous period of time), instead of sending behavioural tags as they are generated.

The behavioural machine is intended to be deployed in situ on a physical machine, and will not transmit video footage over the network for privacy reasons.

4.3.2 The VR-EEG machine

This machine is dedicated to supporting EEG experimentation in virtual reality (VR). Two hardware components will be simultaneously worn by subjects: one EEG device that will capture brain signals, and one VR device that will show a virtual scene. The brain bio signals, detected by the EEG device will be processed in near-real-time by the emotions analysis service, and sent over the network to other machines. In parallel, the machine will receive change orders from the Knowledge Base (KB) and adjusts the virtual environment accordingly.

By doing so, the machine executes parametrized changes in the virtual environment in a manner that befits the requirements and experimental settings of EEG. In addition, the machine has a local storage that stores the emotional tags generated in different experiments and can be accessed remotely and retrieved as sets. The following diagram shows how the VR-EEG machine is conceptually structured.

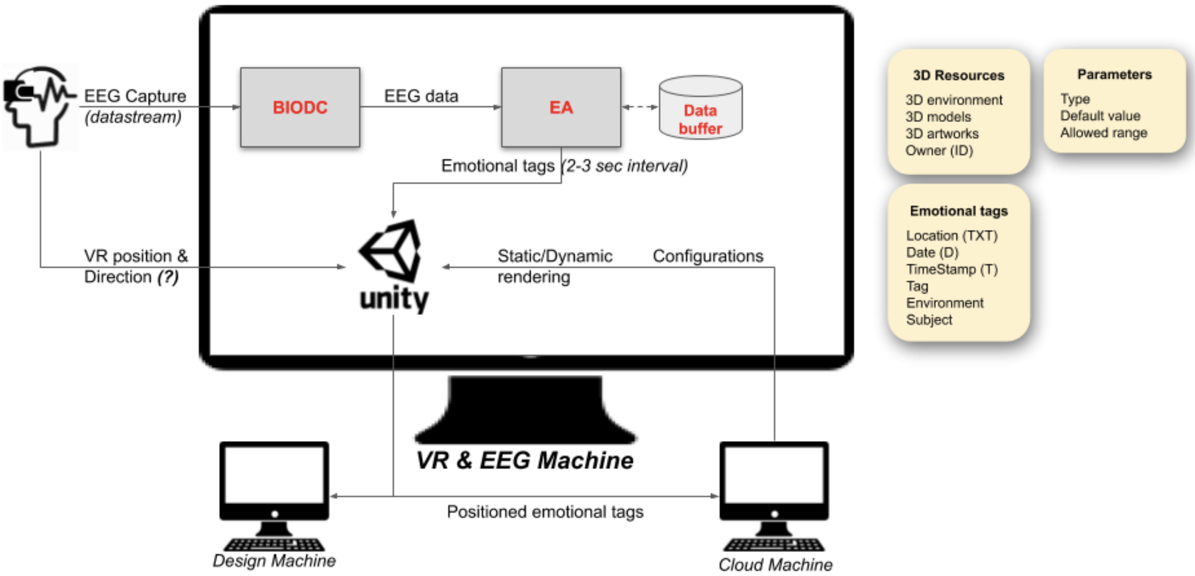

Figure 5 VR-EEG Machine

The machine is expected to be fully operational in the first version of the platform (M18). More robust functionalities, and additional EEG analysis capacities will be contemplated for future versions.

The EEG-VR machine can be deployed locally at any EEG lab associated with the project. The number of physical machines on which the EEG-VR machine that will be deployed during the project will be subject to performance constraints and compatibility requirements with lab settings, and could change from one lab to another.

4.3.3 The design machine

The design machine groups the design tools and components available for the artist or architect. It is mainly composed of three components: the Mindspaces Rhino plugin, the GrassHandler (GH) service, and the ABPS Generative Design and Behavioural Simulation Tool. The design machine can run multiple instances of the Mindspace Rhino, all connecting to the same GrassHandler service hosted in the cloud. This service can also provide connectivity to other design tools and physical assets, and acts as a transformer that compiles analysis data into visual effects and visual transformations.

The design machine is conceived for a flexible use in different aspects of the design process. At its most basic use, it supports the construction of virtual environments in CAD. Above that, the design tool allows to define change parameters in each environment that actuate upon triggers from the knowledge base (KB). The user can simulate this parametric design with data in real time or from previous settings. The user can simulate how changes in the environment can affect the behaviour of the dwellers in the physical space. The user can also monitor how the design parameters are influencing the environment by visualizing emotional and behavioural data as heat maps.

The GrassHandler implements specific and generic types of CAD transformations that allow the designer to determine how aspects change in the environment by combining and customizing these elementary transformations. GrassHandler smoothes out how aspects are changed in a manner that reduces stimulus in virtual environments, in order to preserve the quality of the acquired EEG signal.

The following diagram shows a conceptual design of the design machine:

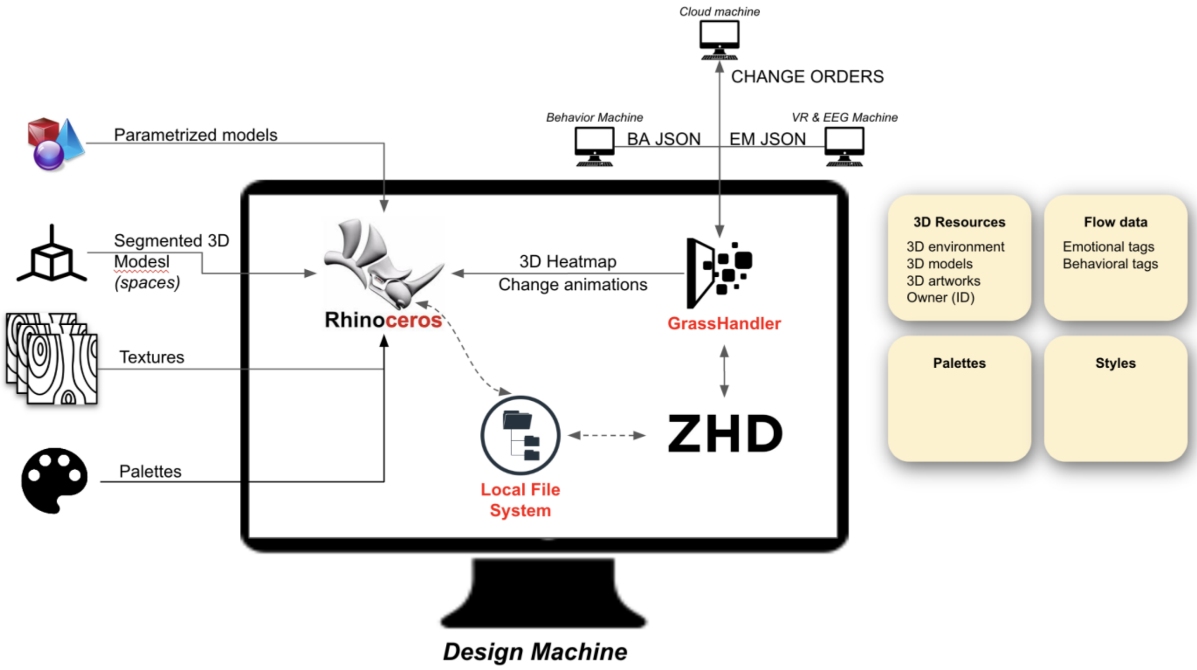

Figure 6 Design Machine

4.3.4 The cloud machine

The Cloud machine was originally conceived to group modules shared among different machines that act as background services. Some of these services require interfacing with the platform users. This is supported through a web-based interface.

In essence, the cloud machine supports the crawling and analysis of discourse and emotions from online sources. For this task, through a web interface, the user sets the concepts and/or sources to crawl, and inspects the discourse and emotions graphs. The cloud machine also houses the Style Transfer service, which is called by the design machine to create new styled objects in CAD, as part of the environmental transformation. Similarly, the cloud machine houses the Knowledge Base that integrates and analyses the different data acquired by the platform (namely emotional tags, behavioural tags, and discourses). The Knowledge Base uses this information to help interactive components decide when and how to change configuration, by selecting change parameters and setting new values for them (e.g. increase illuminance to high, and reduce visible green spaces).

The following diagram shows a conceptual design of the cloud machine:

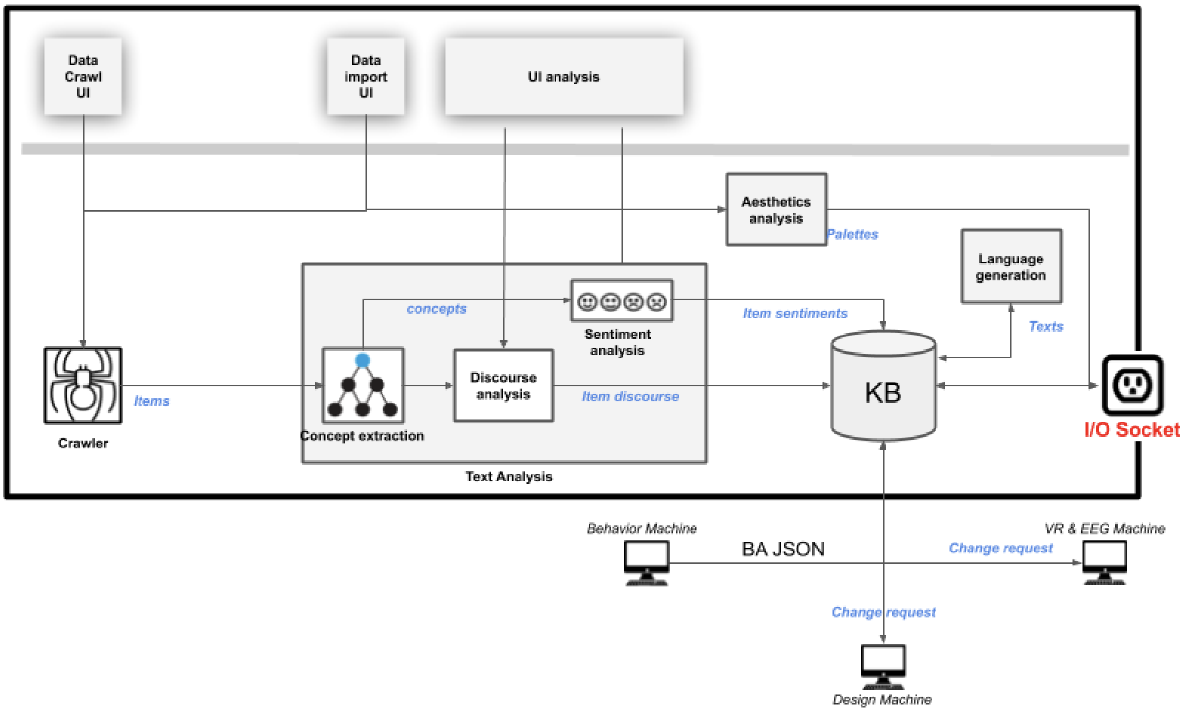

Figure 7 Cloud Machine

The design of the architecture is centred upon the implementation of the core processes supported by the platform, in particular the processes that span several services and modules. The machine design exercise helped to consolidate several concerns in the architecture, and defined a preliminary chaining among the services. Also, a preliminary order between the services has been defined.

Therefore, in order to consolidate the results obtained by the machine design exercise, the deployment model of the platform was analysed. This consists of studying what aspects of the platform come into play under which usage scenario. For this purpose, the following four sub-scenarios or configurations have been identified based on discussions with artists and architects:

- **Project conceptualization and research**: the user defines the project's objectives, location, and approach. Images are uploaded for aesthetic analysis and extraction of palettes and textures; the physical environment targeted by the project is virtualized, and additional 3D assets imported; concepts and concerns are defined, and the crawling process is started to analyse the online discourse and emotions associated with them.
- **Design and production**: the user starts designing the project and acquiring analytical data about the selected space, namely behavioural analysis data; the design produces parameterized 3D objects and the scanned space is also segmented into meaningful components; the user defines design parameters of specific types (e.g. mobility, illumination, decoration, etc.) that will govern how the design of the space changes dynamically in search for an optimal configuration; in order to guide this process, the user can visualize the data from behavioural analysis and emotional analysis as heat maps that reveal chokepoints, hotspots, and trends.

- **Experimentation - indoors/offline**: when the design exercise is completed, it is evaluated in virtual reality by measuring the emotional status of subjects while they experience the virtual environment; in virtual reality, the environment changes automatically according to the emotional status and the intended objectives of the design, for instance, if the objectives were to facilitate mobility and the parameter was moving furniture away from specific areas, the environment will gradually execute this change while the subject's emotional status is monitored, in an effort to identify the most optimal solution; the artist or architect can monitor the experiment in the design machine, and introduce improvements between subjects.
- **Physical deployment and experimentation - outdoors and indoors**: when artists and architects would like to deploy physical interventions in the project's physical space, the platform can provide support for specific processes; behavioural analysis can run in real time and provide monitoring and analytical capacities for the design; the artist or architect can also use the platform's logical components that interpret the results of the behavioural analysis in real time, to automate reactions of physical objects and physical components of the environment (through IoT interfacing).

These usage configurations are shown in the following figure:

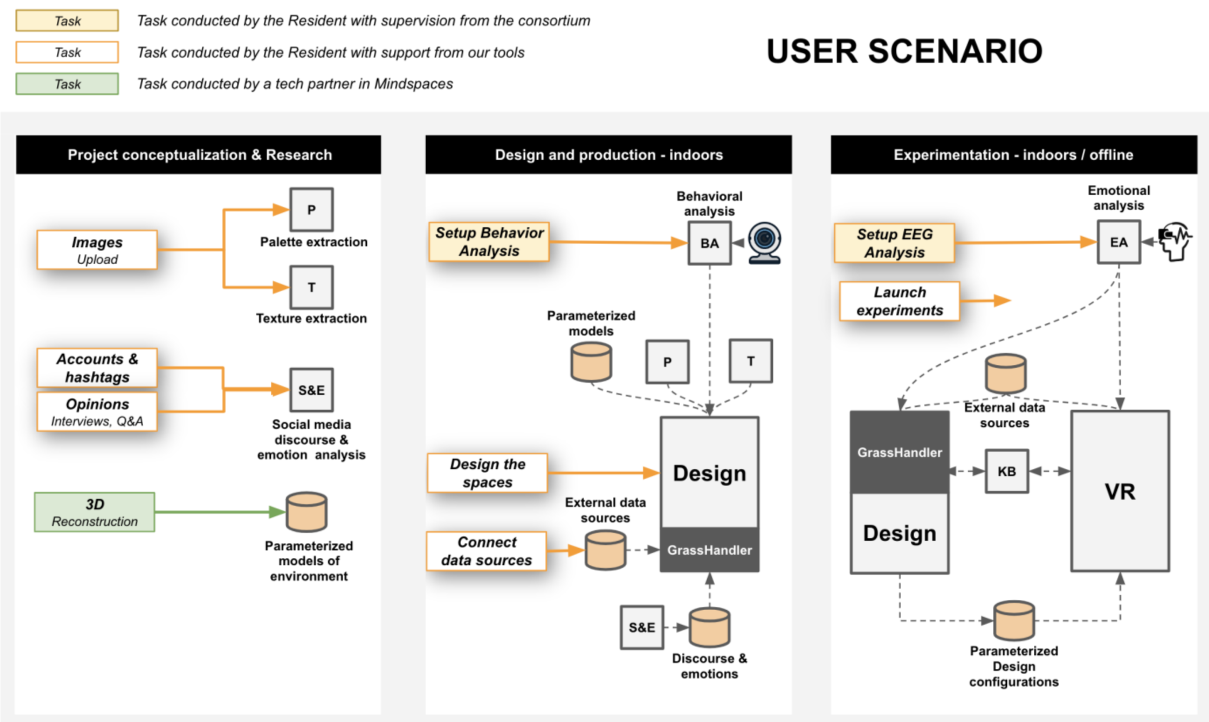

Figure 8 User Scenario

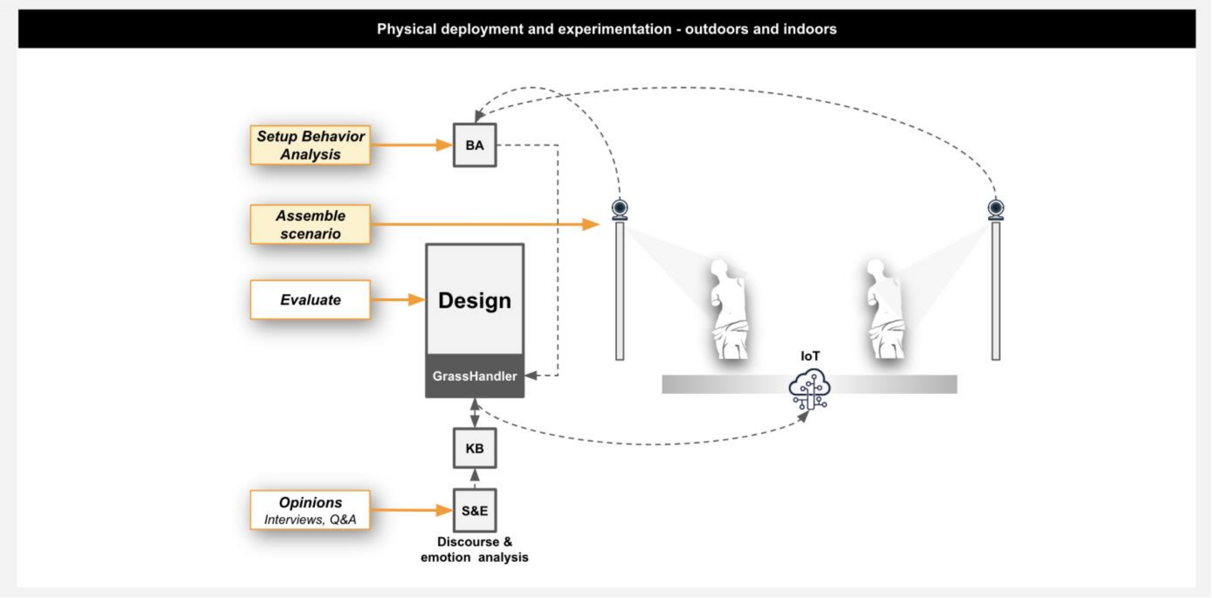

Figure 9 Deployment Scenario

4.4 Architecture and communication design of the platform's first version

The analysis of the deployment model has consolidated the definition of relations between services and consequently helped to design the pipelines for the platform's first version. The resulting architecture design is shown in the following diagram.

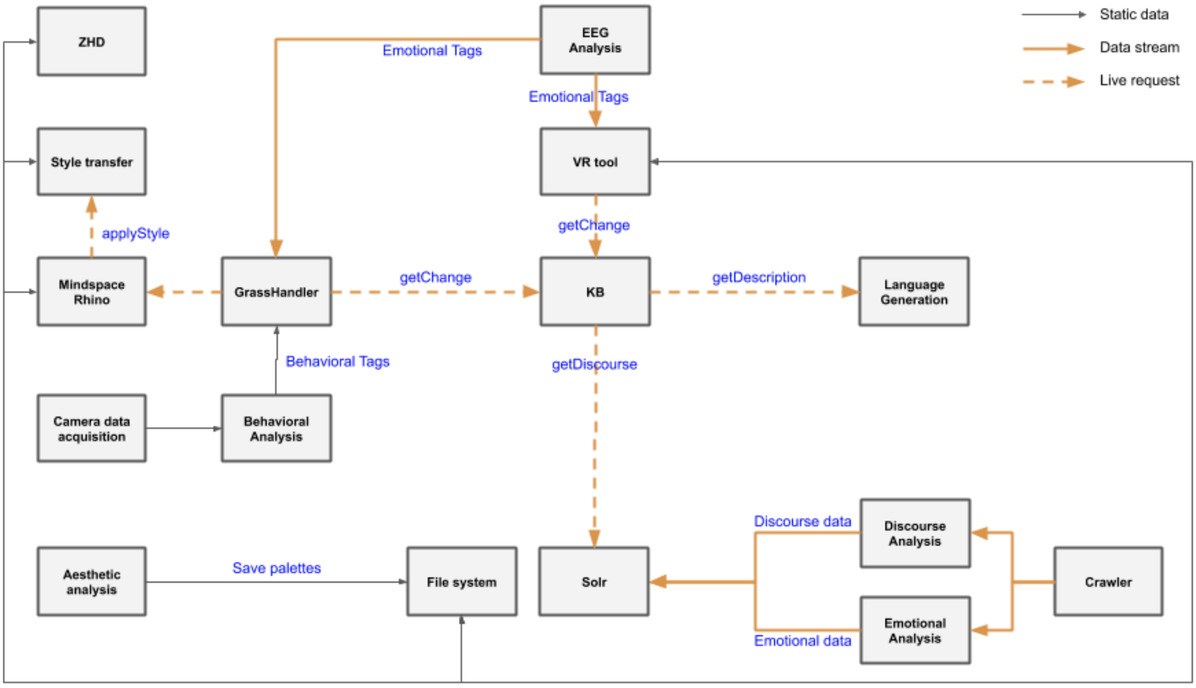

Figure 10 Architecture Design

The following pipelines have been consolidated as part of the platform's first version. They represent the main functionalities intended to be supported by the platform:

- **The semantic pipeline:** In order to semantically analyse data acquired by the crawler from social media, a mechanism is created to extract concepts and structure from processed textual content. These concepts are then structured in a Solr instance hosted in the data storage in order to create the discourse and emotions graphs[10]. Other components can access these graphs by querying the Solr instance's API. This includes web-components that visualize the graphs in an interactive manner that allows users to explore and visually analyse them. In addition, these graphs can be used in external tools and applications developed by the artists and architects.

- **The aesthetic pipeline**: the aesthetic analysis service aims to extract colour palettes with emotional overtones from artworks. It utilizes a rich and varied collection of artworks as a basic input, and generates a set of emotionally-tagged palettes as a result. The palettes are stored in the platform's centralized file system, and are then retrieved by the design tool, which uses them as resources to help users create and transform virtual environments.

- **The emotion analysis pipeline**: arguably the most complex pipeline in the platform, its objectives centre on capturing the emotions that envisioned artistic and/or architectural interventions provoke in the people dwelling in the targeted physical

[10] https://lucene.apache.org/solr/

space. EEG devices are conjointly worn with VR headsets by subjects in a lab settings. While the VR headset shows the envisioned environment in a given configuration to the subject, the EEG device continuously capture brain signals, which are read through the devices' API by the emotional analysis service. The service processes the signals in near-real-time, relying on accumulated signals (around 30 sec of signals) to qualify the emotional state of each subject. The emotional state is also located in the virtual space in order to identify where and under which circumstances the subject experienced specific emotions. This information is relayed to the Knowledge Base, which decides when the emotional state has stabilized sufficiently (and therefore was accurately captured), and then orders the VR to change the environment's configuration according to a specific parameter. The VR executes this change in a manner that avoids stimulating the subject so as to conserve the quality of the EEG data. In parallel, the design tool is also reading the emotional data and visualizing the changes in the environment simultaneously as they happen in VR, allowing the artist or architect to introduce changes in the environment between subjects.

- **The design pipeline**: The main component of this pipeline is the Mindspace design tool, which is an integrated CAD environment that allows users to create 3D spaces and objects, and design how they evolve and change according to specific parameters. The design tool shares the created spaces and objects with other components through the data storage's file system, and also shares the designed change rules with the knowledge base. In order to empower the design further, the design tool reveals the behavioural and emotional data gathered from the space as heatmaps. It also imports a series of emotion-tagged colour palettes from the file storage, and connect to the style transfer service that allows user to change the style of specific objects and surfaces by selecting from a set of available styles. After creating the virtual environment and its parametric design, the design tool can simulate how this environment changes. In addition, it monitors live emotional experiments conducted in VR and behavioural experiments conducted in physical environments. This is done in coordination with the knowledge base. Furthermore, the user can utilise the ABPS Generative Design and Behavioural Simulation Tool to generate virtual workplace environments and conduct elaborate simulations to study how design changes could affect the behaviour of its dwellers.
- **The behavioural analysis pipeline**: Several cameras are deployed in a physical space to analyse the behaviour of people in it. The camera footage is passed to a set of behavioural analysis algorithms tailored to identifying the paths that people take while moving in the space, their interactions with the space and among themselves, among other concerns. The analysed data is stored as session-based datasets, each referring to a specific time and date segment of the analysis of a given space. The data can be aggregated temporally to reveal behavioural patterns that influence the design. This aggregation is handled by the design tool, which visualizes this data is a meaningful way (e.g. heat map). In future versions, behavioural analysis can be conducted in real time in certain conditions.

Other pipelines may be consolidated during the course of the project to streamline how the platform supports artists and architects in the conceptualization, design, evaluation, and implementation of interventions in private, semi-public, and public spaces.

4.4.1 GRPC Communication

GRPC is a modern, lightweight communication designed by Google. It's a high-performance, open-source universal Remote Procedure Call (RPC)[11] that works across a dozen languages running in any OS. In a GRPC communication model a client application can directly call a method on a server on a different machine similarly as it would call a local object making it easier to create a distributed application and services[12]. The communication model starts by defining a service, the specifying the methods that can be called with the parameters and the return types.

GRPC uses protocol buffers[13], a mature open source mechanism to help in serializing structured data, although GRPC can be used with data formats such as JSON[14]. A sample proto message looks like the following:

```
message Person {
  string name = 1;
  int32 id = 2;
  bool has_ponycopter = 3;
}
```

4.4.1.1 GRPC vs REST APIs

REST API communication was the most popular alternate method to handle the communication between the system components. Although due to many restraints of the RESTful communication, some of them are the following[15]:

- Need to do is update client library whenever there is a change in API contracts
- Streaming is difficult
- Duplex streaming is not possible
- Not Language agnostic
- Semantic versioning is required whenever API contract needs to be changed.

GRPC helps in creating a more robust communication model for the platform, as well as gives the developers more flexibility in the development and deployment of the services.

4.4.1.2 Communication of the MindSpaces platform

Figure 11 also shows the communication between the various components of the system based on the technical requirements. Based on this, we developed a methodology to gather all the messages that each service will be able to receive and send using an excel sheet. The excel sheet gathers the following information for each service:

IP		Message	Message Type	Message Content	Content Type	Message info

Table 8 GRPC Message definitions

Although, each service is in the process of defining the GRPC messages to connect to other services there are initial messages that have been defined. The messages are listed in Appendix A.

[11] https://en.wikipedia.org/wiki/Remote_procedure_call
[12] https://grpc.io/docs/guides/concepts/
[13] https://developers.google.com/protocol-buffers/docs/overview
[14] https://cloud.google.com/endpoints/docs/grpc/transcoding?hl=en
[15] https://restfulapi.net/rest-architectural-constraints/

4.5 Data Model

The data model of the platform describes all the data objects created in or imported to the platform. An ownership protocol is used to structure the platform's data model: instances of data objects are grouped by projects, each of which is associated to a single owner or user. This allows each user to control his/her assets and also manage several projects in the platform.

In each project, a set of **3D models** are made available, constructed from scanning the physical space in which the project is executed. Their raw images are preserved and later utilized in the style-transfer process, which changes the styles of 3D objects in the design space. Similarly, **3D resources** are assets imported by the user to be used in the design. A **design configuration** is a virtual environment designed by the user to represent a specific state of the physical environment. Configurations are chained by **Change Parameters**, each specifying how the virtual environment changes from one configuration to another. The Change Parameters are described by human-.readable **Text Arrays**, each discussing how and why a change parameter was executed in a specific experiment. **Colour palettes** and **textures** are elemental assets, each associated to a set of emotions, and provided to the user in the design tool. **Behavioural tags** are grouped under **BA Scenes**, each representing an interval in time at a given data associated with a particular location. Similarly, **Emotional Tags** are grouped under **EEG Experiments**, each associated with a particular subject and setting. For each project, a **discourse set** and an **emotions set** are constructed from the analysed information crawled from social media.

The following diagram shows the platform's conceptual data model, from a DS perspective:

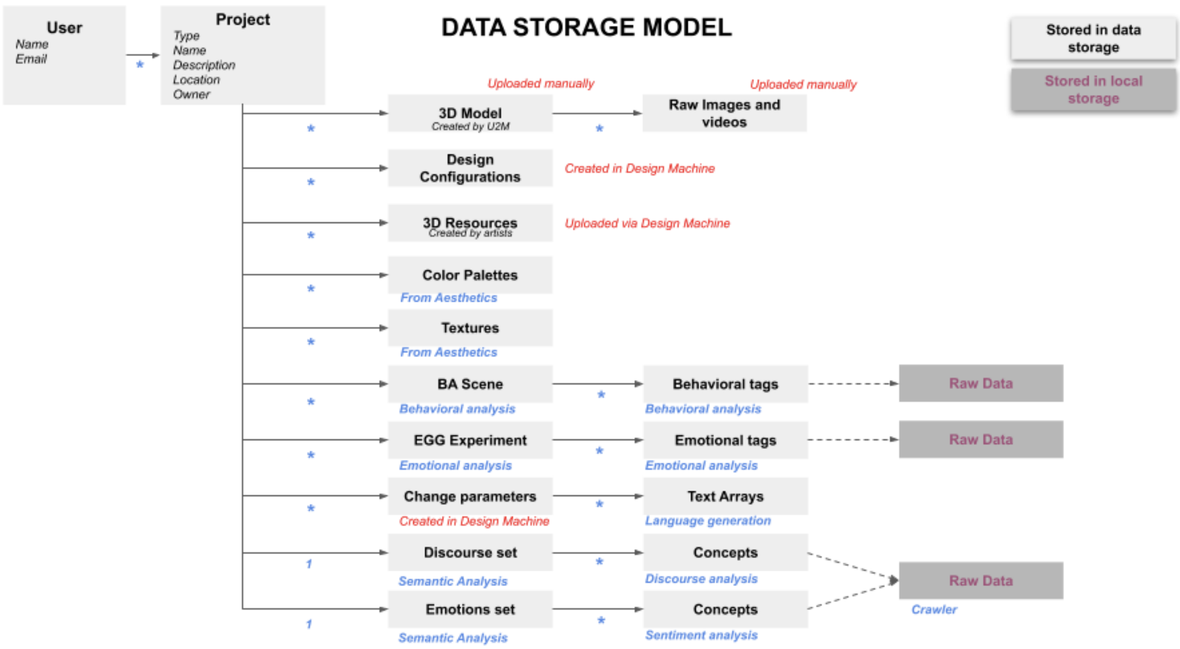

Figure 11 Data Storage Model

5 REQUIREMENTS, SPECIFICATIONS AND FUNCTIONALITIES OF THE MINDSPACES COMPONENTS

5.1 Requirements, Specifications and functionalities of the MindSpaces services

A "service" is defined as a standalone component of the platform architecture16. It communicates with other services as a single entity or point. Internally, a service may integrate different components, each with a specific role of function, but externally the service acts as an integrated application. A service can be hosted on its own independent server, and is managed by a service owner that is responsible for the health of the service.

5.1.1 Aesthetics extraction and Texture proposal

Service owner: CERTH

This Service includes two components:

a) The Aesthetic Extraction (AE). AE aims to extract and categorize the aesthetics of paintings based on their style (i.e. impressionism, cubism and expressionism), creators, genre and emotion. Moreover, the AE will support the extraction of colour palettes from paintings and the creation of colour palettes based on colours suggested by users. This service will analyse visual content from paintings aiming to extract aesthetics aspects as metadata to the MindSpaces platform that could be used as a basis for inspiring architectural and artistic 3D-models.
b) Texture Proposal (TP). TP creates novel textures which will be provided to architects and designers inspiring them to create novel spaces. This service receives a 3D modelID which is connected to a set of images of the 3D model and a "style" image and returns a new set of images which is the result from the fusion of the input images with the selected style from painting or artwork.

5.1.1.1 Functionalities

Function Name	Description	Data input *(expected from other services)*	Data output *(to other services)*
Aesthetics extraction (AE)	Extracts the style, genre, creator, emotion and colour palette from a painting	i) A digital image of a painting ii) A set of colours suggested by the user	i) The aesthetics i.e its style, creator, genre, emotion and colour palette. ii) A colour palette
Texture proposal (TP)	Transfers the texture-style from a selected painting or artwork into the input image.	A model ID connected to a set of images of the 3D model and the selected style from painting or artwork.	A new set of images related to the 3D model which is the result from the fusion of the input images with the selected

[16] https://en.wikipedia.org/wiki/Service-oriented_architecture

			style from painting or artwork.

For each of these functions, specify the following:

Function Name	**Trigger** *(what makes it run)*	**Expected response time** *(how long it takes to run)*	**Capacity** *(nb requests handled)*
Aesthetics extraction (AE)	The user will be able to upload a collection of paintings and receive the aesthetics as metadata. Moreover, the user could receive a set of colour palettes.	5s per image	1 request at a time
Texture proposal (TP)	The user will be able to change among different aesthetic styles in order to change the texture of the interior and exterior spaces offline	More than 20s to apply the style image to the content image.	1 request at a time

5.1.1.2 Requirements

Operating System: Linux for TP and Windows for the AE.
CPU: i5-9600k
RAM: 16 GB
Disk Space: 30 GB
Graphic card: 8 GB
Others:

- Expected capacity in processing requests: 1 request at a time.
- Expected availability and reliability of the service: 1 request at a time to handle. The use of a queue is under consideration.
- Interoperability requirements (*if relevant*)

Scalability:
There is a linear dependency between the number of requests and the processing time. If there is a need for multiple requests the service has to scale horizontally.

Data Security:
There is not any additional security policy. The service relies on the security policy of the middleware.

5.1.1.3 Logical Design

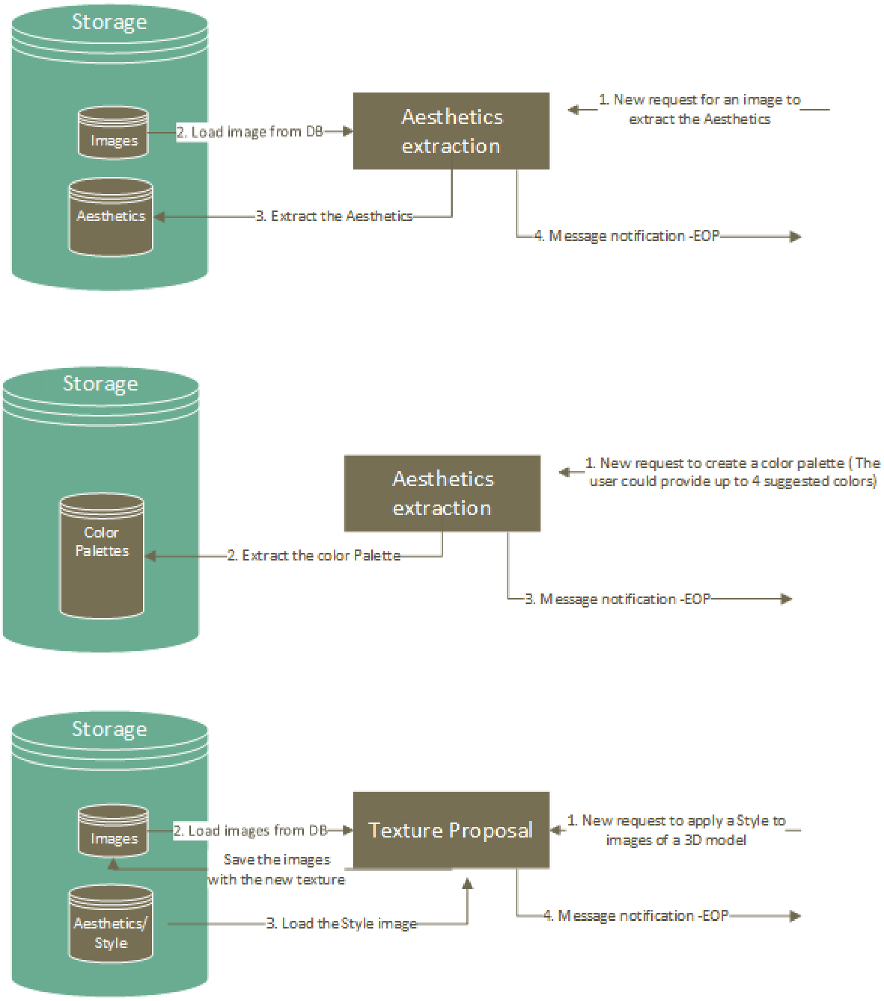

Figure 12 AE&TP Logical Design

Components:

Component name	Data input	Data output	Function(s) performed
AE	A request to extract the aesthetics (style, creator, genre, emotion, colour palette) from an image	The aesthetics (genre, creator, style, emotion and colour palette)	Load the image from the DB, Extract the aesthetic, Save aesthetic for the input image, Send an EOP message
TP	A request to transfer a style from a painting into a set of images of a 3D model	The set of images with the new texture	Load images, load painting, apply style transfer, save the new images, send an EOP message.

5.1.1.4 Input and Output data

Inputs:

Input source	Data Type (live stream, packaged, or static)	Description of the data
AE- An image (The id of the image) or a set of suggested	Offline data	Images of paintings for the case of the extraction of aesthetics and a set of suggested colours (up to 4) for

colours (up to 4)		the case of the creation of a colour palette
TP- style image and model (IDs in order to load them)	Offline data	A set of assets (styles/paintings) and 3D models

Outputs:

Output data	**Data Type (live stream, packaged, or static)**	**Description of the data**
TP-New 2D images	Offline data	A set of new 2D images produced by the fusion between input images of a 3D model and style image
AE-Saves the metadata of an image i.e. creator, genre, style, emotion, colour palette	Offline data	A set of extracted attributes used as metadata.

5.1.2 Bio signal data collection

Data collection of bio signals falls into WP3, Task 3.1 and is responsible for collecting users' physiological data while experiencing the VR installations of the MindSpaces project. Physiological signals (e.g., EEG, GSR) will be collected in order to be used for the analysis and development of emotion recognition algorithms, used in WP5 task 5.1. Subsequently, biosignal data collection constitutes a submodule of the Emotional State Recognition service described in section 5.1.3.

EEG Devices - Enobio

Enobio is a mobile and wireless device that captures electroencephalographic (EEG) signals with high precision[17]. It provides high dynamic resolution (24 bits, 0.05 μV) and offers a bandwidth of 0 to 125 Hz. It connects to the computer through Bluetooth and allows the setup of eight, 20 or 32 electrodes in 39 different head positions. For MindSpaces project, the selected EEG channels are: FP1, FP2, AF3, AF4, F3, F4, F7, F8.

Enobio allows the set-up of the EEG channel recordings with dry electrodes, solidgel or gel-based electrodes. The flexible headcap design allows adapting the headset montage to the project applying disposable electrodes for the experiments and dry electrodes to facilitate out-of-the-lab recordings. This flexibility regarding the number and type of electrodes makes it easy to use and tolerable for the subjects to wear.

Enobio provides its own SDK (NIC2) in order to be able to connect to the device and manage it through the computer. It consists of a rechargeable battery with a life of 6.5 hours with wireless data transmission (range of 10 meters from the computer running the software NIC2).

[17] https://www.neuroelectrics.com/solutions/enobio/8/

It also offers basic and advanced modes in order to design and monitor experiments. In addition, it can integrate data from external sources and share data with external systems, like Lab Streaming Layer (LSL) library.

Enobio provides flexibility between the number of channels and head position, it also offers different choices regarding the type of electrodes (dry, solidgel, gel based) which make it easy to use, it is wireless, mobile, precise and compatible with other external systems and sources. Enobio outperforms other available EEG recording devices, like Emotiv Epoc, Emotiv Insight and Muse, which do not cover all the above features, and for this reason, it is selected to be used on the project.

GSR Device

Regarding the data collection of Galvanic Skin Response (GSR) signals, Shimmer3 GSR + UNIT is used for MindSpaces project. Main function of Shimmer3 is to measure the Galvanic Skin Response (or electrodermal activity) of the subject's skin[18]. In response to internal and external stimuli, sweat glands become more active, increasing moisture content on the skin and allowing electrical current to flow by changing the balance of positive and negative ions in the secreted fluid. Shimmer3 also provides its own SDK that offers the ability to connect and manage the device and record the data. It is wireless, connects to the computer through Bluetooth and portable. In addition, it does not cause any discomfort to the users since it uses only two GSR sensors. Moreover, it is compatible with external sources and can share data with external systems, like Lab Streaming Layer (LSL) library.

Synchronization

Different types of biosignal data will be collected for MindSpaces project. Signal acquisition and synchronization between the two different kinds of bio signals, EEG and GSR, is achieved with either the use of Lab Streaming Layer (LSL) library, or the use of LabRecorder software[19].

LSL library offers the ability to stream bio signals and provide them as input to the Emotional State Recognition service that is described in the following section. LSL is a system for the unified collection of measurement time series in research experiments that handles both the networking, time-synchronization and real-time access. LSL has many language interfaces, including Python and Matlab.

LabRecorder is a software which offers synchronization and records all data into a single file[20]. The LabRecorder is the default recording program that comes with LSL. It allows recording all streams on the lab network (or a subset) into a single file, with time synchronization between streams.

5.1.3 Emotional State Recognition

This service is responsible for emotion extraction from physiological signals. Physiological signals will be acquired from lightweight recording devices, such as Enobio. This component will analyse the above signals and specific features will be extracted from these signals in

[18] https://www.shimmersensing.com/products/shimmer3-wireless-gsr-sensor

[19] https://www.neurobs.com/pres_docs/html/03_presentation/06_hardware_interfacing/02_lab_streaming_layer.htm

[20] https://github.com/sccn/lsl_archived/wiki/LabRecorder.wiki

order to assist to map the signals to the corresponding emotional states in the two dimensional valence arousal space.
The two dimensional valence arousal space considers the emotions, not as discrete states, but as continuous ones and maps them in a two dimensional space described by valence and arousal.
In Mindspaces ecosystem the algorithms developed in this WP will measure the emotional responses of the end users while experiencing the various MindSpaces art installations in VR to assess their impact and effectiveness.

5.1.3.1 Functionalities

Function Name	**Description**	**Data input** *(expected from other services)*	**Data output** *(to other services)*
Biosignal data acquisition	This function will be responsible for acquiring the signals from the recording device in real time. More information are described in the *"Bio signal data collection"* section.	Input LSL stream	Raw bio signals
Data preprocessing and prediction	This function will be used for processing of bio signals and prediction of the emotional tag	Raw bio signals	Emotional tag, valence tag, arousal tag, timestamp
Emotional tags transfer	This function will be used to send the results (tags and the timestamps) to Grasshopper and Unity	Emotional tag, valence tag, arousal tag, timestamp	Emotional tag, valence tag, arousal tag, timestamp

For each of these functions, specify the following:

Function Name	**Trigger** *(what makes it run)*	**Expected response time** *(how long it takes to run)*	**Capacity** *(nb requests handled)*
Biosignal data acquisition	Manual activation	Seconds(sec)	Linear, handles one request every time
Data preprocessing and prediction	Function call	The response time may vary from 2 - 10 sec	Linear, handles one request every time
Emotional tags transfer	Function call	Milliseconds(ms)	Linear, handles one request every time

5.1.3.2 Service Requirements

Operating System: Windows
CPU: intel core i5
RAM: >4 GB
Disk Space: >250 GB
Graphic card: not needed

Others: No

- Expected capacity in processing requests: Linear. 1 request per time
- Expected availability and reliability of the service: This service is connected to the recording devices and supports the experiment running every time.
- Interoperability requirements (*if relevant*): The service has to be compatible to LSL library

5.1.3.3 Logical Design

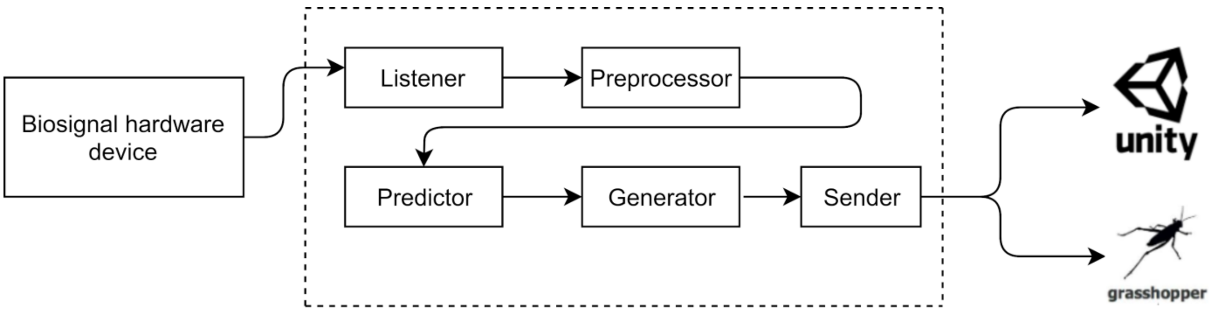

Figure 13 Emotional State Recognition Logical Design

Components

Function Name	Data input	Data output	Function(s) performed
Listener - Data Getter	Input LSL stream with data	Biosignals	This module connects to the device and receives data through LSL library
Preprocessor	Biosignals	Preprocessed signals	This module preprocesses the raw signals
Predictor	Preprocessed signals	Emotional tag	This module predicts the emotional tag from the preprocessed signals
Output Generator	Emotional tag, valence tag, arousal tag, time	Dictionary object (results)	This module wraps up the tags and the timestamp into a dictionary object
Sender	Dictionary object (results)	-	This module implements the gRPC clients and sends the results to Grasshopper and Unity through gRPC communication

Related Technical Requirements

TR	Title	Description			Related URs

TR 6.	Acquire biometric data from subjects through light weight biometric devices	Light weight recording devices will be used in order to record biometric data from subjects, while experiencing VR environments			UR_4
TR 7.	Create an emotional state recognition service in order to recognize the emotional states of users in real time	An emotional state recognition algorithm will be developed and trained in order to predict the emotional state of the users, while experiencing a VR environment in real-time.			UR_8, UR_9, UR_10, UR_11, UR_12, UR_13, UR_14, UR_15, UR_16, UR_30

5.1.3.4 Input and Output Data

Input Data Sources:

Input source	**Data Type (live stream, packaged, or static)**	**Description of the data**
Bio signals	Live stream	Bio signals obtained from recording devices will be used as input in this service

Output Data:

output data	Data Type (live stream, packaged, or static)	Description of the data
Emotional tags, valence tags, arousal tags, timestamps	Live stream	The output of this service will be Emotional tags representing the dimensional state on the two dimensional valence arousal space Valence tags representing predicted valence metric Arousal tags representing predicted arousal metric Timestamps

5.1.4 Visual Data Collection Service for Behaviour Analysis

Data collection from visual sensors falls into WP3, Task 3.2. A collection of video recordings from indoors and outdoors will take place in order to apply visual behaviour analysis and get new insights from the areas under study. Video recordings will be collected from RGB cameras and stored locally on the same machine where the visual behaviour analysis component is installed. This component includes: a) the interface which supports the connection and the extraction of videos and b) the component for the anonymization of video via the use of a face blurring technique.

Interface for video cameras: This is a windows application targeting .NET Framework 4.6.1. It is written in C#. It includes the OpencvSharp v4.0.30319 package. The user will be able to run the application in the visual behaviour machine to collect and store visual data for further processing.

Face blurring: This is a module for the anonymization of videos. After the collection of video frames a face blurring technique will take place in order to blur the detected faces from video frames. This is a module written in Python and it will be installed on the same machine.

5.1.5 Social media and web data crawling/scrapping service

Service owner: CERTH

A web app with GUI for the users (artists/architects) that includes all the crawling and scraping functionalities envisioned in the project, and defined under T3.5 in the project document. It encapsulates web crawling tools designed to extract freely available textual and visual content from open public web resources, including from Twitter. Regarding the external competition, there is not, to this moment, a standard tool to extract open data from online sources due to the heterogeneity in web technologies and content. Thus, it is of great interest to experiment with the parameters of the query formulation.

5.1.5.1 Functionalities

Function Name	**Description**	**Data input**	**Data output**
Twitter Crawling	Executes the data aggregating procedure from twitter public posts based on keywords and/or geolocation. Intended to be run offline.	Queries	Content from web resources, in json format, posted via web to the Textual Analysis Service.
Scrapping From Selected Websites	Executes the data aggregating procedure from pre-selected websites provided by the users based on website domain and its user comments.	Web Entry Points	Content from websites, in json format, posted via web to the Textual Analysis Service

For each of these functions:

Function Name	**Trigger**	**Expected response time**	**Capacity**

Twitter Crawling	A user commences the function by providing the high-level query terms then periodical searches follow	Depends on data volume to be crawled due to fixed limit of Twitter API serving per API Key	Capability of satisfying multiple queries simultaneously with a single twitter API Key
Scrapping From Selected Websites	A user commences the function by providing the high-level query terms	Magnitude of minutes for average sized websites	Linear, handles one request every time

5.1.5.2 Service Requirements

Operating System: Linux
CPU: -
RAM: >8GB
Disk Space: >30GB
Graphic Card: Not Needed
Others:

- Expected capacity in processing requests: Difficult to evaluate because of the dependence on external factors, will be calculated empirically
- Expected availability and reliability of the service: 100%
- Interoperability requirements: The data to be posted to the textual analysis service need to follow the json format (.json).

Scalability:
There is a linear dependency between the number of requests and the processing time. If there is a need for multiple requests the service has to scale horizontally.

Data Security:
There is not any additional security policy. The service relies on the security policy of the middleware.

5.1.5.3 Logical Design

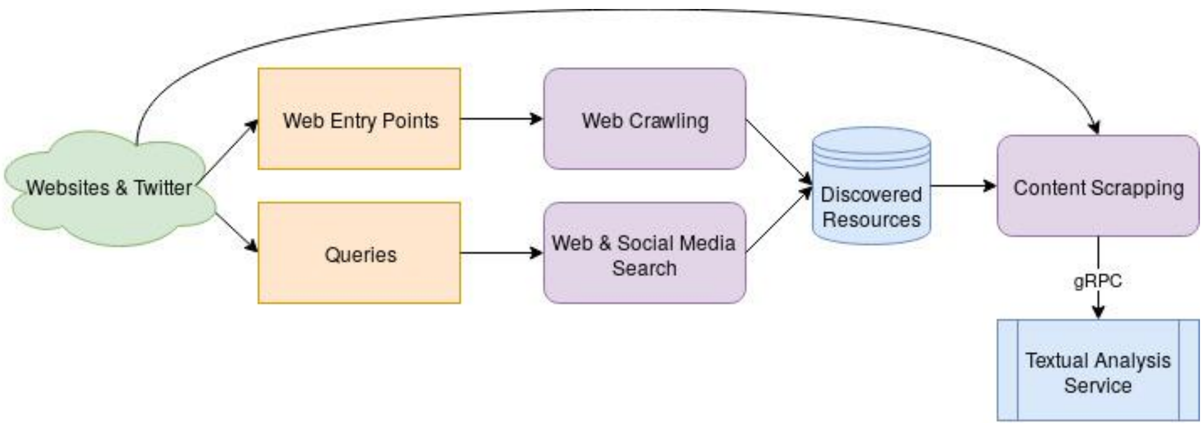

Figure 14 Visual Data Collection Logical Design

Components

Component name	Data input	Data output	Function(s) performed

Web scrapping	Web resource URLs	Web pages content (textual and visual)	Extracts content from web pages
Twitter crawling & scrapping	Queries(e.g. hashtags, user accounts)	Social media posts	Extracts content from social media

Related Technical Requirements

TR	Title	Description	Function	Function performed	Related URs
TR 4.	Create a crawling service to gather online content from social media	Frameworks will be developed that will utilize available APIs from relevant social media platforms to fetch online public user data	Twitter Crawling	Extracts content from social media	UR_6, UR_9, UR_20, UR_65, UR_72
TR 5.	Create a scrapping service to gather online content from websites	A framework will be developed that will fetch online content from websites	Scrapping From Selected Websites	Extracts content from web pages	UR_6, UR_9, UR_20, UR_65, UR_72

5.1.5.4 Input and Output Data

Input Data Sources:

Input source	Data Type (live stream, packaged, or static)	Description of the data
Strings provided by the user (artist/architect)	Live stream	The user (artist/architect) will provide the content of the query in a string format.

Output Data Sources

Output data	Data Type (live stream, packaged, or static)	Description of the data
JSON structures	Live Stream	The output contains json structures about each web entity crawled with the respective textual (and/or visual) data and metadata.

5.1.6 KB Population

Service owner: CERTH

The Knowledge Base (KB) population service concerns the service which maps various pieces of information incoming from multiple MindSpaces modules to the RDF-based representation format, that will depend on custom ontologies derived from existing ontologies and will be implemented during the scope of MindSpaces. The knowledge structures will also supply for all the indispensable semantics required to generate textual descriptions for each component (Text generation module in WP5). The KB population service will support JSON. The service will be also responsible for updating the KB itself when incoming information is available from other components.

5.1.6.1 Functionalities

Function Name	**Description**	**Data input**	**Data output**
Behavioural Analysis Offline Bulk Bundle Results Insertion	Remote method to upload massively the results of the behavioural analysis service to the KB	JSON object	Updated KB (RDF triples that correspond to the incoming data)
Aesthetics Analysis Offline Bulk Bundle Results Insertion	Remote method to upload massively the results of the aesthetics analysis service to the KB	JSON object	Updated KB (RDF triples that correspond to the incoming data)
Textual Analysis Offline/Online Bulk Bundle Results Insertion	Remote method to upload massively the results of the textual analysis service to the KB	JSON object	Updated KB (RDF triples that correspond to the incoming data)
EEG Analysis Results Online Insertion	Remote method to upload the results of the EEG analysis service to the KB	JSON object	Updated KB (RDF triples that correspond to the incoming data)

For each of these functions:

Function Name	**Trigger**	**Expected response time**	**Capacity**
Behavioural Analysis Offline Bulk Bundle Results Insertion	New available data	Unknown	Linear, handles one request every time. Slight concurrency can be supported in later stages
Aesthetics Analysis Offline Bulk Bundle Results Insertion	New available data	Unknown	Linear, handles one request every time. Slight concurrency can be supported in later stages.

Textual Analysis Offline/Online Bulk Bundle Results Insertion	New available data	Unknown	Linear, handles one request every time. Slight concurrency can be supported in later stages
EEG Analysis Results Online Insertion	New available data	Unknown	Linear, handles one request every time. Slight concurrency can be supported in later stages

5.1.6.2 Service Requirements

Operating System: Linux
CPU: Intel® Xeon® Silver 4108
RAM: >8GB
Disk Space: >50GB
Graphic Card: Not Needed
Others:

- Expected capacity in processing requests: At the beginning, the service will not support concurrency, e.g. any request to map data will be processed only when the previous one is finished. If needed, concurrency can be supported.
- Expected availability and reliability of the service: 100%
- Interoperability requirements: JSON-style message format (protobuf[21]) will be the primary structure for semantic data transfer, following the gRPC protocol messages

5.1.6.3 Logical Design

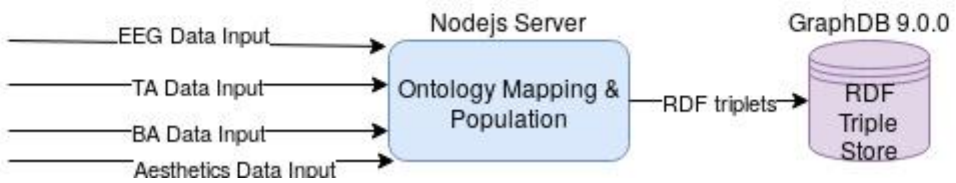

Figure 15 KB Logical Design

Components

Component name	Data input	Data output	Function(s) performed
Ontology mapping & Population	Analysis results from EEG, Textual Analysis (TA), Behavioural Analysis (BA) and Aesthetics Analysis	Updated KB (RDF triples that correspond to the incoming data)	Mapping multiple JSON data objects to RDF data

Related Technical Requirements

TR	Title	Description	Functions	Function performed	Related URs

[21] https://en.wikipedia.org/wiki/Protocol_Buffers

TR 8.	Create a knowledge base that will fuse multimodal data and unify them into one ontological model	The KB will serve as a semantic information database, holding data and controlling the flow about geolocation, and correlating emotional states with objects present at scene.	Behavioural Analysis Offline Bulk Bundle Results Insertion, Aesthetics Analysis Offline Bulk Bundle Results Insertion, Textual Analysis Offline/Online Bulk Bundle Results Insertion, EEG Analysis Results Online Insertion	Mapping multiple JSON data objects to RDF data	UR_5, UR_10, UR_11, UR_12, UR_13, UR_14, UR_15, UR_16, UR_51, UR_61, UR_62

5.1.6.4 Input and Output Data

Input Data Sources:

Input source	**Data Type (live stream, packaged, or static)**	**Description of the data**
JSON objects	live stream	JSON objects are surrounded by curly braces {}. JSON objects are written in key/value pairs. Keys must be strings, and values must be a valid JSON data type (string, number, object, array, boolean or null). Keys and values are separated by a colon. Each key/value pair is separated by a comma.

Output Data Sources

Output data	**Data Type (live stream, packaged, or static)**	**Description of the data**
RDF Triples	static	a RDF triple is a set of three entities that codifies a statement about semantic data in the form of subject–predicate–object expressions

5.1.7 Semantic Integration and Reasoning Service

Service owner: CERTH

The reasoning service will handle the further knowledge analysis depicted in the Knowledge base (KB). In more detail, the module will create a unified representation of the available resources, considering information relevant to texture and aesthetics (Aesthetics module in WP4), named entities, concepts and relations extracted from textual analysis (Linguistic analysis module in WP4), cognitive and emotional data extracted from EEG and physiological sensors (Emotional and cognitive sensing module in WP3), as well as exterior and interior objects and other content-specific attributes (City sensing module in WP3). At this point, the ontologies created by the KB population module will be the basis for the reasoning and information coupling algorithms (WP5), enabling the decision-making assets, respectively to the use case requirements. Generally, the reasoning module will be responsible for acquiring higher-level knowledge that is not easily grasped by humans and will be generated by the output of other modules.

5.1.7.1 Functionalities

Function Name	Description	Data input	Data output
StartReasoning	Remote method to start the reasoning task	-	Updated KB with new knowledge (RDF triples)
StartSearch	Remote method to be called in order to retrieve results from the data storage	Filtering data from the interface	Query results in the form of the RDF model

For each of these functions:

Function Name	Trigger	Expected response time	Capacity
StartReasoning	Always running, starts with the start of PUCs	Not applicable	Linear, handles one request every time. Slight concurrency can be supported in later stages
StartSearch	When there are sufficient new available data in data storage	Unknown	Linear, handles one request every time. Slight concurrency can be supported in later stages.

5.1.7.2 Service Requirements

Operating System: Linux
CPU: Intel® Xeon® Silver 4108
RAM: >8GB
Disk Space: >50GB
Graphic Card: Not Needed
Others:

- Expected capacity in processing requests: At the beginning, the service will not support concurrency. If needed, concurrency can be supported.
- Expected availability and reliability of the service: 100%

5.1.7.3 Logical Design

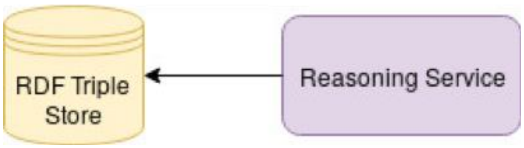

Figure 16 Semantic and Reasoning Logical Design

Components

Component name	Data input	Data output	Function(s) performed
Inference engine	-	Updated KB with new knowledge (RDF triples)	Derives implicit relations
Reasoning Service	Filters	Query results in the form of RDF model	Query formulation / enrichment

Related Technical Requirements

TR	Title	Description	Functions	Function performed	Related URs
TR 9.	Create a Reasoning Service providing the changes that are wanted to occur	The Reasoning Service will provide a set of changes to occur inside the VR based on logic on what the user (artist/architect) want to achieve as the emotional state of the subject	StartSearch, StartReasoning	Query formulation / enrichment	UR_24, UR_41, UR_42, UR_44, UR_50, UR_57, UR_58, UR_64

5.1.7.4 Input and Output Data

Input Data Sources:

Input source	Data Type (live stream, packaged, or static)	Description of the data
RDF Triples	Static	a RDF triple is a set of three entities that codifies a statement about semantic data in the form of subject–predicate–object expressions

Output Data Sources

Output data	Data Type (live stream, packaged, or static)	Description of the data
JSON Object	Live stream	JSON objects are surrounded by curly braces {}. JSON objects are written in key/value pairs. Keys must be strings, and values must be a valid JSON data type (string, number, object, array, boolean or null). Keys and values are separated by a colon. Each key/value pair is separated by a comma.

5.1.8 Human Behaviour Analysis

Service owner: MU-DKE

The Mindspaces Human Behaviour Recognition module will detect patterns of individual, group or crowd behaviors in indoors or outdoors spaces. This will involve static camera-based tracking for activity detection and recognition and eventually activity/behaviour pattern extraction.

5.1.8.1 Functionalities

Function Name	**Description**	**Data input (expected from other services)**	**Data output (to other services)**
Motion detection	Motion detection	None - Video data	json
Motion tracking	Motion tracking	None - Video data	json
Activity Detection (maybe)	Activity Detection	None - Video data	json
Activity Recognition (maybe)	Activity Recognition	None - Video data	json
Event Detection	Event Detection	None - Video data	json
Interaction Detection (maybe)	Interaction Detection	None - Video data	json
Function Name	**Description**	**Data input**	**Data output**
Textual analysis (TA)	Extracts linguistic information including semantics and sentiments from text sources	Plain text	Concepts, predicate-argument relations, sentiment tags and other linguistic and semantic information

For this function:

Function Name	**Trigger (what makes it run)**	**Expected response time (how long it takes to run)**	**Capacity (nb requests handled)**
Motion detection	If motion is visible in frame or sequence of frames	Several seconds per sequence (offline probably)	
Motion tracking	After motion is detected	Several seconds per sequence (offline probably)	

Activity Detection (maybe)	After motion is detected and people are recognized	Several seconds per sequence (offline probably)	
Activity Recognition (maybe)	When activity is detected	Several seconds per sequence (offline probably)	
Event Detection (maybe)	When motion is detected	Several seconds per sequence (offline probably)	
Interaction Detection (maybe)	When an event is recognized between 2 or more people	Several seconds per sequence (offline probably)	

5.1.8.2 Service Requirements

Operating System: Ubuntu
CPU: AMD Threadripper 1920X atleast but dependent on the GPU
RAM: >12GB
Disk Space: 10GB for trained models
Graphic Card: >12GB VRAM for model
Others:

Data Security:
No access protocol is currently envisioned. The service relies on the security policy of the middleware.

5.1.9 Textual Analysis (TA)

Service owner: UPF

The textual analysis service receives unstructured text data (in any of the supported languages) and extracts linguistic and semantic information, including but not limited to concepts appearing in the text, when possible with links to external knowledge resources such as DBpedia, and relations between them[22]. It also aims at identifying sentiments of citizens' contributions to social media and opinions in online content on websites.

The information extracted from textual collections is fed into a knowledge base where it is aggregated and potentially fused with information acquired from data sources of another type to allow for an understanding of end-users' feelings and needs and further semantic reasoning. The service can also be used to detect concepts and sentiments in a stand-alone text provided by a toolset user for a subsequent gathering of general opinions on various related aspects by inference from the knowledge base.

[22] https://wiki.dbpedia.org

5.1.9.1 Functionalities

Function Name	Description	Data input	Data output
Textual analysis (TA)	Extracts linguistic information including semantics and sentiments from text sources	Plain text	Concepts, predicate-argument relations, sentiment tags and other linguistic and semantic information

For this function:

Function Name	Trigger	Expected response time	Capacity
Textual analysis (TA)	First, Crawler or Scraper that posts textual content from Twitter or websites via web in JSON-style message format (protobuf). Second, a toolset user submitting a plain text.	~ 10 seconds to several minutes	1 request at a time

5.1.9.2 Service Requirements

Operating System: Linux / Docker
CPU: AMD64
RAM: 10GB per language
Disk Space: 30GB shared + ~5GB per language
Graphic Card: Not needed
Others:

- Expected capacity: 1 request at a time. Throughput varies based on pipeline configuration and document length, between ~10 seconds per document and several minutes per document (per computation node). Can handle multiple parallel requests when used as a REST service.
- Expected availability and reliability of the service: The service will be available at any time. The use of a queue is under consideration

Scalability:
The service can scale horizontally by adding computing nodes.

Data Security:
No access protocol is currently envisioned. The service relies on the security policy of the middleware.

5.1.9.3 Logical Design

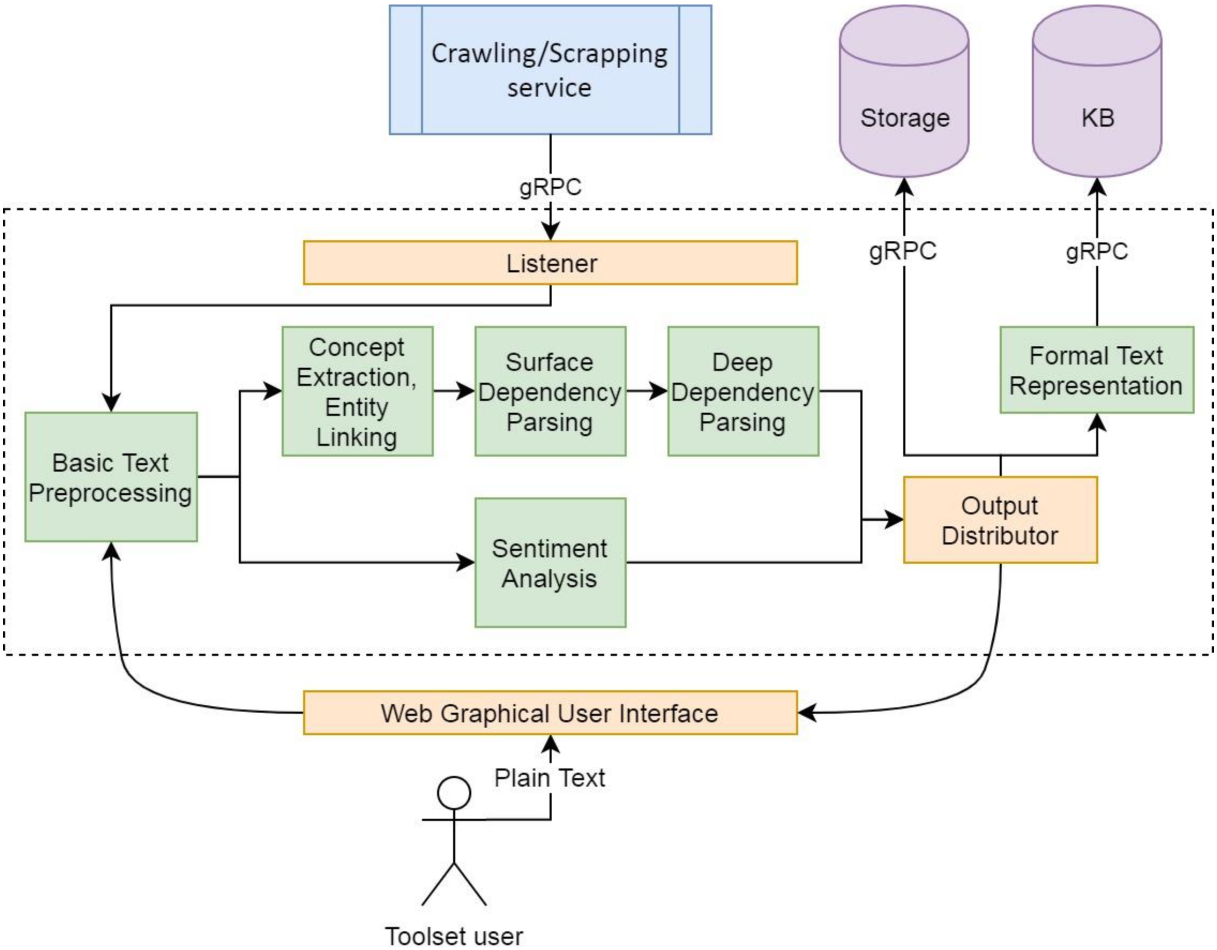

Figure 17 Textual Analysis Logical Design

Components:

Component name	Data input	Data output	Function(s) performed
Listener	JSON-style message (protobuf) with a text	Plain text	This module connects to Crawling/Scraping service and receives textual data
Basic Text Preprocessing	Plain text	Internal structures with lists of tokens, part-of-speech tags, and lemmas	Basic preprocessing of a text, in particular, tokenization, part-of-speech tagging, and lemmatization
Concept Extraction, Entity Linking	Internal structures with lists of tokens, part-of-speech tags, and lemmas	Concepts linked to external resources	This module performs extraction of concepts and linking them to external knowledge resources
Sentiment Analysis	Internal structures with lists of tokens, part-of-speech tags, and lemmas	Textual emotional tags	This module identifies feelings and opinions expressed in a text

Surface Dependency Parsing	Concepts linked to external resources	Syntactic structure	This model identifies surface relations
Deep Dependency Parsing	Syntactic structure	Deep syntactic and semantic structures	This module detects deep syntactic and semantic predicate-argument structures
Output Distributor	Different layers of linguistic annotation	-	This module implements the gRPC client and sends the results to Storage and Knowledge Base (first, passing results through Formal Text Representation module) via gRPC communication or returns results to be visualized within Web GUI
Formal Text Representation	Different layers of linguistic annotation	Formal Text Representation	This module projects obtained linguistic structures onto formal text representation leaving only necessary information for the knowledge base

5.1.9.4 Input and Output Data

Input Data Sources

Input source	**Data Type (live stream, packaged, or static)**	**Description of the data**
Plain text	Static	Plain text obtained from Crawling/Scraping service or through UI will be used as input in this service

Output Data Sources

Output data	**Data Type (live stream, packaged, or static)**	**Description of the data**
Linguistic and semantic information	Static	Discourse and sentiment information with special formal representation for a knowledge base

5.1.10 Text Generation (TGS)

Service owner: UPF

The text generation service takes structured input from the knowledge base, which can originate from textual analysis of one or more input texts or from non-textual sources such as outputs from processing of EEG or visual signals (i.e., emotions and the visual behaviour events/activities), and verbalizes the information as human-readable text in one of the supported languages. The graph-transduction grammars and lexical resources are used to perform all the main stages in generation (text planning, linguistic generation, determination of the structure of the sentence, introduction of grammatical words, resolving the morphological agreements between the words are resolved, ordering the words, introduction of punctuation signs) to provide a projection of complex ontological configurations onto lexicalized semantic structures and their subsequent realization as natural language sentences.

5.1.10.1 Functionalities

Function Name	Description	Data input	Data output
Text Generation (TGS)	Verbalize structured information as human-readable text	Linguistic and semantic structures from the knowledge base	Generated plain text (in one of the supported languages)

For this function:

Function Name	Trigger	Expected response time	Capacity
Text Generation (TGS)	Knowledge base will provide information triples to be verbalized	~3 seconds	1 request at a time

5.1.10.2 Service Requirements

Operating System: Linux / Docker
CPU: AMD64
RAM: 2GB per language
Disk Space: 1GB per language
Graphic Card: Not needed
Others:

- Expected capacity: 1 request at a time. Throughput varies based on number of input structures (e.g. triples). Typically ~3 seconds for 7 triples. Can handle multiple parallel requests when used as a REST service.
- Expected availability and reliability of the service: The service will be available at any time. The use of a queue is under consideration.

Scalability:
The service can scale horizontally by adding computing nodes.

Data Security:
No access protocol is currently envisioned. The service relies on the security policy of the middleware.

5.1.10.3 Logical Design

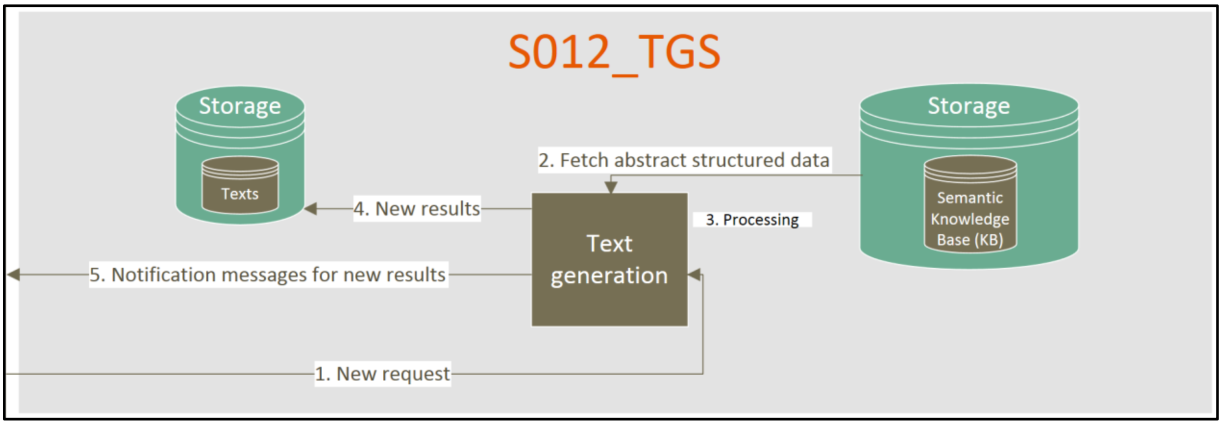

Figure 18 Text Generation Logical Design

Components:

Component name	Data input	Data output	Function(s) performed
TGS	Linguistic and semantic structures from the knowledge base	Plain text	Loads the set of triples from the KB, performs all the stages in generation, saves a text for the input structures, and sends an EOP message

5.1.10.4 Input and Output Data

Input Data Sources

Input source	Data Type (live stream, packaged, or static)	Description of the data
KB (output of semantic integration and reasoning)	Static	Triples established based on linguistic and semantic structures and other information acquired from non-textual data sources and stored the knowledge base

Output Data Sources

Output data	Data Type (live stream, packaged, or static)	Description of the data
Plain text	Static	Generated human-readable text in one of the supported languages)

5.1.11 The GrassHandler service

Service Owner: McNeel

GrassHandler is a middleware module responsible for collecting data from sensors and experiments, and managing the design environments. Its role is complementary to the KB: while the KB interprets data and decides when/how to change the environment, the GrassHandler module executes these changes for the design and monitoring tools, namely the MDT. In addition, the GrassHandler makes the data from sensors and experiments available as a design and monitoring resource, both in real-time and in ad-hoc fashion.

GrassHandler is deployed on a server or virtual machine with (minimum recommended) Windows Server 2016 or 2019 (LTSC) installed. It requires Rhino Compute, an experimental project that allows remote access to the Rhino's geometry library through a stateless REST API. Compute is based on the Rhino Inside™ technology that allows to embed Rhino advanced geometry calculation inside an online web service. The two typical scenarios for using Compute are running as a web server on a remote Window Server operating system and running locally on a user's computer for debugging or providing local services to applications[23].

5.1.11.1 System requirements

Operating System: Windows Server 2016 or 2019
CPU: 2 vCPU
RAM: >4GB
Disk Space: >2GB
Graphic Card: Not Needed
Others: GrassHandler uses TopShelf to make it easy to configure and run it as a service on Windows.
Scalability: The system is designed to support simultaneous for dozens of users, and can be expanded further by increasing the CPU and RAM resources of the server. In a cloud environment, the system can be hosted in a server with auto-scaling capacity.
Data security: The system can run under an SSL security certificate, using a simple ACME client for Windows.

5.1.11.2 Logical design and components

GrassHandler is composed of 6 main components as shown in the following figure. Rhino Compute is a background service use by GrassHandler to run as a cloud service. gRPC I/O is an implemented communication framework that allows it to exchange data with other Mindspace components. On top of these components, a GrassHopper instance runs as a shell and an editor for implemented processes. These include elementary CAD transformations rendered in real-time (grouped under the Transformation Renderer), and data transformations that interpret real-time and ad-hoc data sets (grouped under Data Handler). Finally, the GrassHander integrates an interfacing mechanism with Rhino API, allowing access to more than 2.400 API calls from Rhino3D.

[23] https://www.rhino3d.com/compute

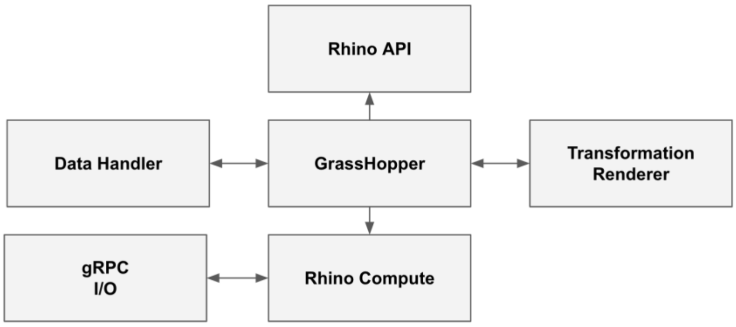

Figure 19 GrassHandler Logical Design

5.1.11.3 Related Technical Requirements:

The technical requirements of GrassHandler are mainly related to the Mindspace Design Tool requirements. This service is currently conceived as a middleware for supporting the design process.

5.1.11.4 Input and Output Data Sources

Input Data Sources:

Input source	**Data Type (live stream, packaged, or static)**	**Description of the data**
Emotional tags	Live stream	Data generated by Emotional Analysis in real-time
Emotional set	Static dataset	Dataset generated by Emotional Analysis during an experiment, involving one or multiple subjects.
Behavioural tags	Live stream	Data generated by Behavioural Analysis in real-time
Behavioural set	Static dataset	Dataset generated by Behavioural Analysis during a monitoring session.
Sensor packets	Live stream	Data collected from any type of sensors.
Change Requests	Live stream - low frequency	Request for transitioning the environment in simulation, issued by the KB.

Output Data Sources:

Output data	**Data Type (live stream, packaged, or static)**	**Description of the data**
getChanges	Live stream - low frequency	Requests sent to the KB to acquire new changes.

5.1.12 Wellbeing analysis service

Studies show that cortisol levels, one of the best transient stress indicators, are correlated with how close a subject walk to the walls in a closed environment. Based on this, a wellbeing analysis service is conceived to interpret the level of "stress", "comfort" or "wellbeing" elicited by a subject in an environment. This will be accomplished by training a machine-learning model based on EEG data collected in different environmental settings. In these settings, the environment is static (no changes are introduced while the subject is in VR), but the camera moves in three variations; a) next to the walls, b) in the middle of the room, c) in between. This would serve to estimate the neural correlates of comfort/"wellbeing" in an environment, and consequently model it so that it can be used as a reference to interpret real-time EEG signals in dynamic simulations.

This service is conceived for the second version of the platform, and currently is under conceptualization, which includes the creation and development of its experimental setting.

5.1.13 Data Storage

Service Owner: Nurogames

The Data Storage component of the platform will be used to store the raw data and the processed data of the user.

5.1.13.1 System Requirements

Operating System: Windows Server 2016 or 2019
CPU: 2 vCPU
RAM: >8GB
Disk Space: >1TB
Graphic Card: Not Needed
Others: NodeJS installed.
Scalability: The system is designed to support simultaneous for multiple requests, and can be expanded further by increasing the Disk Space.
Data security: The system can run under an SSL security certificate.

5.1.13.2 Databases

1. SOLR DB: The SOLR Database will be used to store the concepts from the semantic pipeline and will be used to create the discourse and emotions graphs. Other components can access these graphs by querying the SOLR instance's API.
2. MongoDB: The MongoDB will be used to store the 3D models and other data which can be accessed by the KB and other components of the system through GRPC requests[24].

5.1.13.3 Input and Output Data Sources

Input Data Sources:

Input source	Data Type (live stream, packaged, or static)	Description of the data
storeData	Any type of data	Any type of data can be stored in the MongoDB or SOLR (used for TA only)

Output Data Sources:

[24] https://www.mongodb.com/

Output data	Data Type (live stream, packaged, or static)	Description of the data
getData	On requests	Sends the relevant data to the KB or other components on request

5.1 User tools

This section of the deliverables documents the functionalities and the UI/UX of the tools that will be used by each of the cohort of the user in the MindSpaces platform. There are 3 cohorts of users in the platform

1. The Designers – These are the designers that will be designing the environments that the platform will display and run the experiments with. These would be the main user of the MindSpaces Design Machine.
2. The Architects – These will be architects that will work on creating different parameterised models of various environments. These will be the main user of ABPS Generative Design and Behavioural Tool.
3. The End User – These are the users that will be in the VR environment along with EEG sensor to help perform the experiment. These will include citizens, office workers, senior citizens based on the user case.

5.2.1 The Mindspaces Design Tool

The Mindspaces Design Tool (MDT) is a Rhino3D plugin developed on top of the Rhinoceros 3D framework, and tailored according to the needs of the artists and architects that will use the Mindspaces platform. The MDT allows its users to perform a series of tasks that address core concerns of the Mindspaces processes, including:

- **Designing virtual environments** on the basis of 3D scans taken of the physical space addressed by the user's project. The design includes the utilization of segmented elements in the space (e.g. doors, lighting fixtures, walls, etc.), and 3D objects imported by the user to the environment.
- **Visualizing emotional and behavioural maps** of the environment, using data captured and processed by EEG and Behavioural analyses. Based on this information, the user identifies stress points and specific locations in the space, which represent opportunities for artistic and/or architectural interventions.
- **Introducing and configuring design parameters** that transform the environment from one state to another. This is accomplished by associating a set of transformations to elements of the 3D space, introduced by the user or segmented from the scans.
- **Simulating the evolution of the environment** to review how the introduced design parameters change it, and refine this design before and between EEG experiments and behavioural analysis.

The following wireframes show an envisioned GUI design of the MDT.

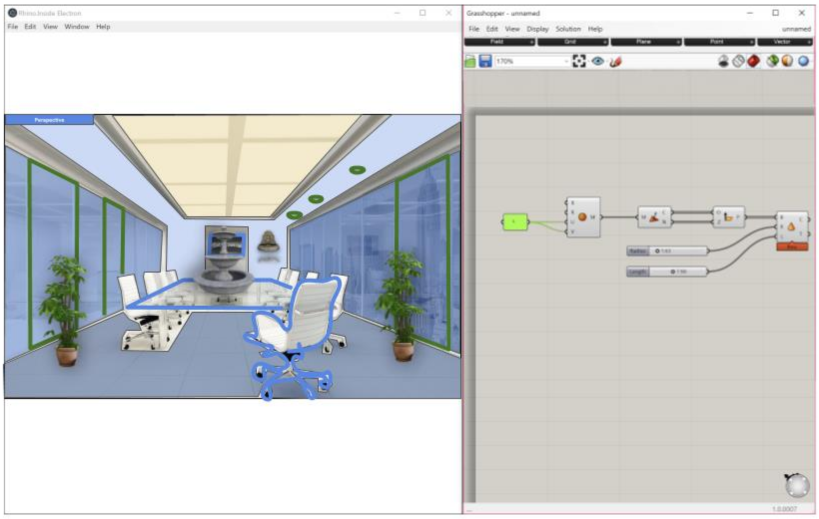

Figure 20 MDT Wireframe 1

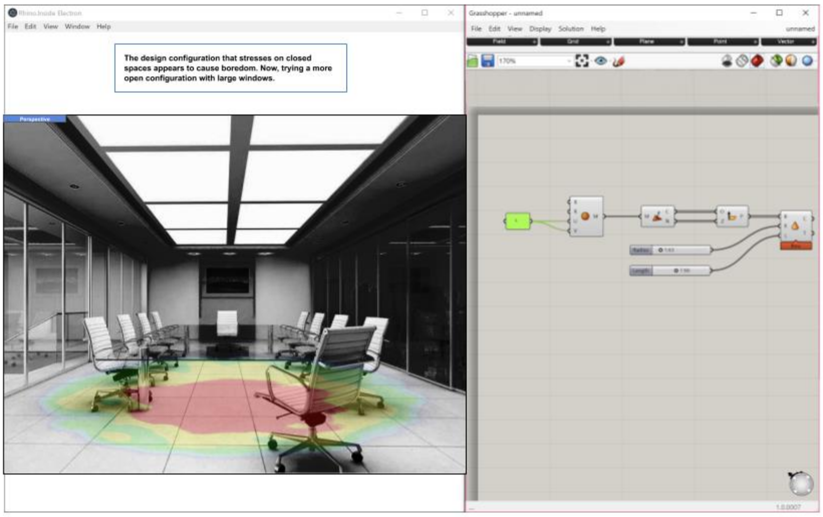

Figure 21 MDT Wireframe 2

Installation requirements:

The MDT functions as a software installed on the user machine, having a fast CPU and at least 8GH of RAM. Graphic cards able to accelerate graphic computations are recommended. In addition, the MDT can be also hosted as a cloud service (Platform-as-a-Service). The user installs the tool by deploying the software, according to instructions that will be provided with the tool.

5.2.2 ABPS Generative Design and Behavioural Simulation Tool

Tool owner: Zaha Hadid Architects

The ABPS Generative Design & Behavioural Simulation Tool (Agent Based Parametric Semiology) is a platform built in Unity for the semi-autonomous design of virtual workplace environments and agent-based simulation of human behaviour in relation with them, yielding a user constrained design model which is "crowd aware." User defined parameters provide constraints that drive the design algorithm to generate configurations of architectural features composing a virtual workplace. Cognitive agent-based behavioural modelling is used for the simulation of people interacting within 3D virtual environments through autonomous decision making of actions such as focused working, social activities, collaborative activities, and other daily activities in relation to each other and the environment. Simulation data is collected to measure environmental, spatial, social, and behavioural performance such as social encounters, working, space and asset utilization, in order to predict the performance and improve the workplace design models. This enables users to predict the social and spatial performance of design options before they are physically constructed. The ABPS tool has two primary components:

ABPS Designer: The generative design components provide for the semi-autonomous design of workplaces controlled through user defined parameters and constraints. Starting from a virtual mesh model of fixed architectural elements, users can define parameters for spatial organization and occupation strategies, types of design features, and other constraints for the design of a workplace environment through a user interface. User input data is then utilised by the tool to constrain and generate design variations which can be tested and evaluated by the user and simulation tools.

ABPS Simulator: The behavioural simulation components provide the user with an interface for defining the characteristics of the "population" of people that would inhabit the virtual workplace environment. From these parameters, a virtual crowd of "agents" is generated to simulate collective human behaviour in the virtual workplace over periods of time. Simulation data surrounding asset utilisation, occupancy, and social behaviour is collected and output as a series of heatmaps over the virtual model as well as numerical data for the evaluation and improvement of design performance.

Component name	Data input	Data output	Function(s) performed
ABPS Designer	3D Mesh Model + User Preferences (UI Interface)	ABPS Unity Scenes	Generates ABPS dynamic unity scenes from 3d mesh models and user inputs
ABPS Simulator	ABPS Unity Scenes + User Preferences (UI Interface)	Simulation data (behaviour / occupancy data, asset data, spatial heat maps mesh data)	Runs simulations over ABPS dynamic unity scenes

The following screen captures illustrate a work-in-progress vision of the ABPS Simulator:

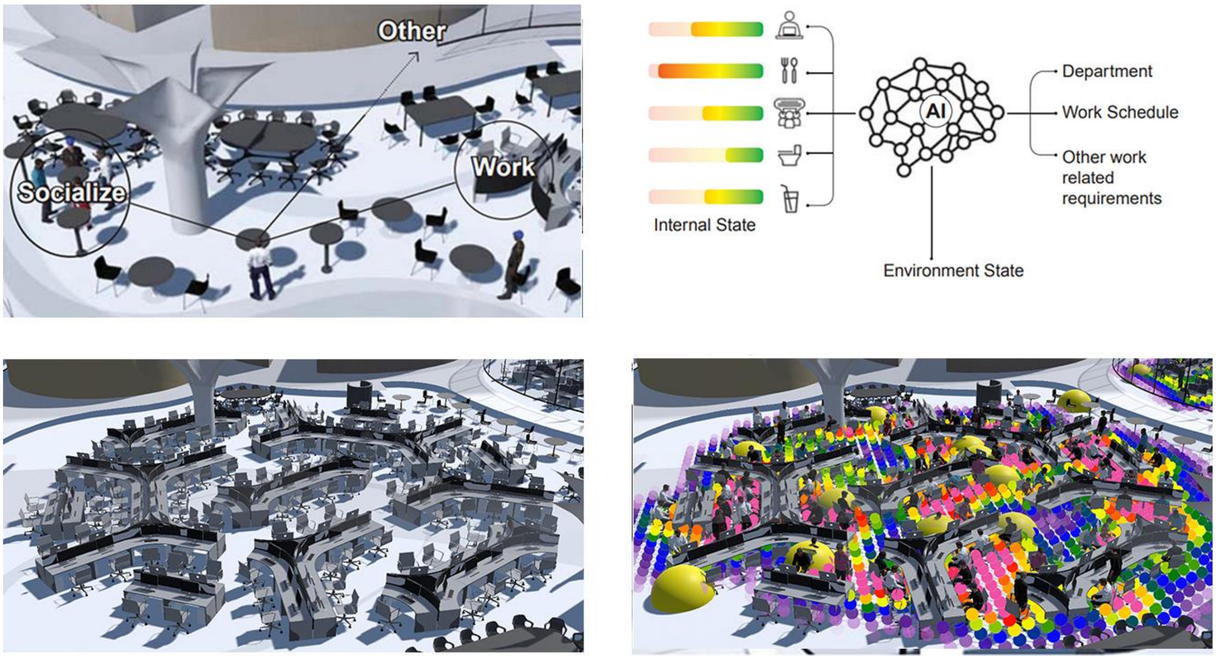

Figure 22 ABPS Screenshots

Logical Design

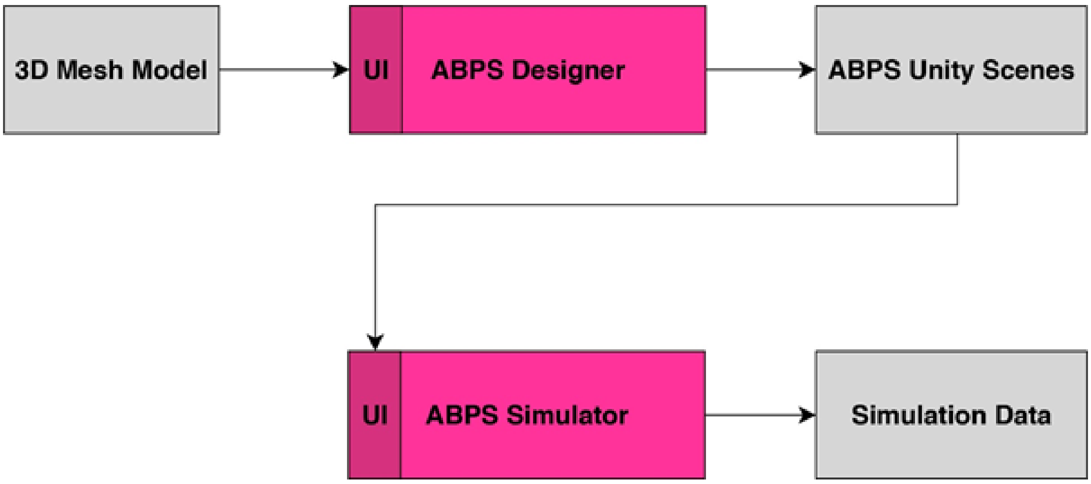

Figure 23 ABPS Logical Design

Installation requirements:

The ABPS Tool functions as a software installed on the user machine, having a fast CPU and at least 8GH of RAM. Graphic cards able to accelerate graphic computations are recommended. In the future, the ABPS Tool is potentially foreseen to be hosted as a cloud service (Platform-as-a-Service). The user installs the tool by deploying the software, according to instructions that will be provided with the tool.

5.2.3 MindSpaces VR Tool

Tool Owner: Nurogames

The VR Tool for MindSpaces platform will be tool that will be used by the end users of the platform. The end users being the citizens, artists, office workers, senior citizens. The tool will be built to view the 3D Models that will be designed in the Design Machine of the platform.

The tool will be created using Unity[25] and will use GRPC for communication. The tool will be connected to the EA which will be responsible for sending the emotional tags which when added with the location of the user, will be send to the KB for a receiving a new configuration.

5.2.3.1 System Requirements

Operating System: Windows 10 Professional.
CPU: i7
RAM: Minimum 16 GB
Disk Space: 10 GB
Graphic card: Minimum Nvidia GTX 1060 6 GB VRAM
Others:

- Oculus Rift or Oculus Rift S attached
- The system should have at least one HDMI port and one USB 3.0 port.

Scalability:
One VR Tool can be used by one user simultaneously. The system is easily scalable as it will be included in the VR-EEG machine which can be deployed on any system.

Data Security:
There is not any additional security policy. The component relies on the security policy of the middleware.

5.2.3.2 Functionalities

According to initial assessment of the requirements, the VR Tool will perform the following functionalities:

- The tool will take the incoming configuration from the KB and load the 3D Models from the DS
- The tool will ingest the EA tags from the EA component
- The tool will integrate the EA Tags and the virtual location of the user in the 3D environment.
- The tool will send the user location and the EA Tag to the KB to get a new configuration.
- The tool will allow the user to move inside the VR environment using teleportation.

5.2.3.3 Logical Design

Figure 25 shows the design and the communication of the VR Tool with the other components of the platform. The tool is composed on a 3D model loader, a communication module, EA Tags and user location module and a user movement control module.

[25] https://unity.com/

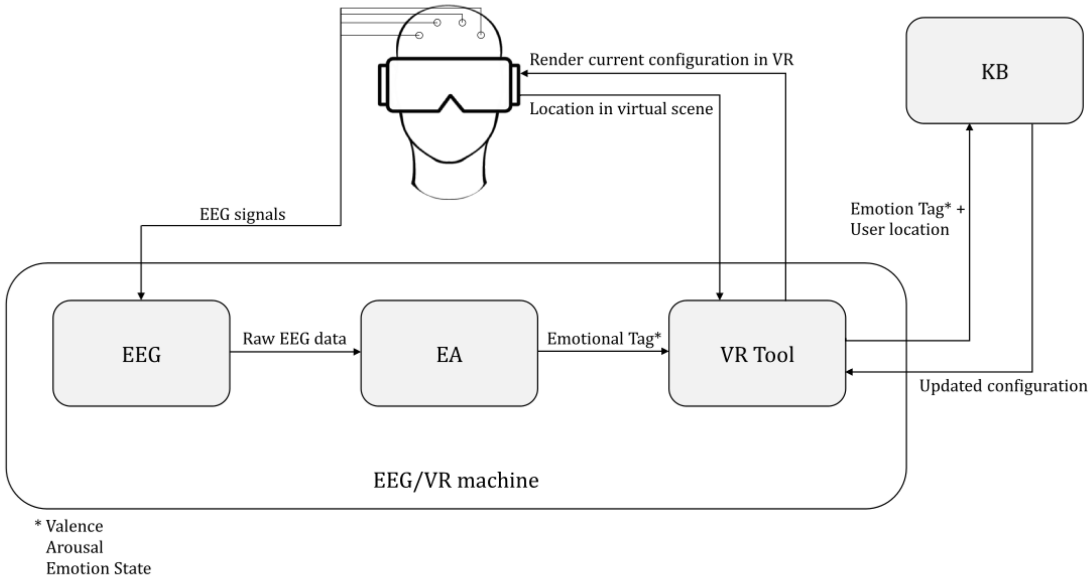

Figure 24 VR Tool communication design

Figure 25 shows the various components of the VR Tool. The main communication of the VR Tool will be with the EA, KB and DS.

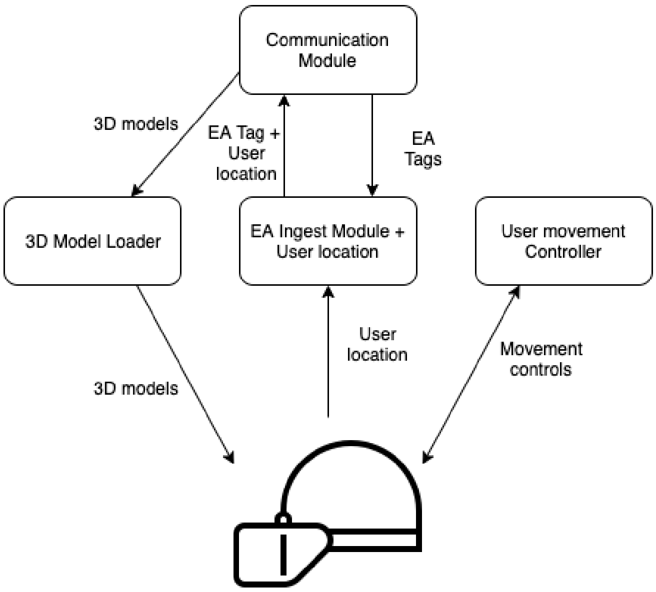

Figure 25 VR Tool logical design

6 CONCLUSIONS

In this deliverable we have presented and discussed the technical specifications of the MindSpaces platform based on early analysis that precedes the implementation of its first operational prototype. After a short introduction on the general practices for requirements collection and analysis, and the delimitation of the scope of such activity in the context of MindSpaces, we discussed the technical specifications, first on a platform level, and then on a module level, addressing services, machines, and user tools. Finally, we conducted an aggregative analysis of requirements by which common concerns related to the architecture design specifications (data management, communication, and platform processing pipeline) were revisited, and extended by relying on the elementary specifications collected.

The platform will have an iterative approach with at least three distinct versions, that will be implemented and evaluated in a consecutive manner, of the platform in order to meet the user requirements defined under the pilot use cases in the project. Therefore, the particular set of user requirements that are addressed under a specific version should be isolated and clearly stated in order to orient the user evaluations. For instance, it is expected that the first "proof-of-concept" version of the platform, otherwise known as V1, will incorporate rudimentary and basic functions. The following version V2 will address pilot use cases more in depth and demonstrate the viability and added-value of the processes described by these.

Each of the service of the platform has been documented in the deliverable which shows that the services will be easily adaptable to the user requirements and will also follow a similar approach in implementation as the platform with three distinct versions. There are still various issues that needs to be addressed in the definition of the services, but these would be solved during the implementation and iteration process.

Finally, the deliverable addresses the end user tools that will be the front end of the system. These tools include the design tool, the ABPS Generative Design and Behavioural tool, and the VR Tool of the system. We also address the UI and UX as well as the first prototypes of these tools. The entire deliverable provides a technical overview of the each of the components of the system such as the services, the machines for deployment, the communication model, the data model, and the end user tools. Each of the service will be developed following an iterative approach for better results by the end of the project. This will insure a more streamlined and result oriented implementation of the system.

BIBLIOGRAPHY

[1] Liao, L. (2002). From Requirements to Architecture: The State of the Art in Software Architecture Design. Department of Computer Science and Engineering, University of Washington , 1-13..

[2] Lehman, Meir M. The programming process. internal IBM report, 1969..

[3] Lehman, M.M.; Belady, L.A. (1985). Program evolution : processes of software change. London: Academic Press Inc. ISBN 0-12-442441-4..

[4] Lehman N, Meir M., et al. Metrics and laws of software evolution-the nineties view. En Software metrics symposium, 1997. proceedings., fourth international. IEEE, 1997. p. 20-32..

APPENDIX A – GRPC Messages

Knowledge Base

IP		Message	Message Type	Message content	Content Type	Useful for	Message info

160.40.52.155:50051		GetChangeRequest	received	message GetChangeRequest{
				string project_id = 1;
				string last_change_id = 2;
				new_EEG_tag tag = 3;
				}
				message new_EEG_tag{
				string subject_id = 1;
				string experiment_id =2;
				string project_id = 3;
				google.protobuf.Timestamp
				timestamp = 4;
				int32 EEG_Tag = 5;
				int32 valence = 6;
				int32 arousal = 7;
				SubjectPositionInformation
				position_info = 8;
				}
				message
				SubjectPositionInformation {
				float pos_x = 1;
				float pos_y = 2;
				float pos_z = 3;
				float rot_x = 4;
				float rot_y = 5;
				float rot_z = 6;
				}
					GRPC	KB (as a serveR)	Contains EEG and user position

160.40.52.155:50051		GetChangeResponse	sent (PoC)	message GetChangeResponse
				{
				string objectid = 1;
				string size = 2;
				string texture =3;
				bool smooth = 4; }
					GRPC	VR Tool, GH	Contains PoC suggestions for environment changes. WIll become more solid in future when Design Parameters are firm and Reasoning will be established

Crawler

IP		Message	Message Type	Message content	Content Type	Useful for	Message info

160.40.52.155:50052		getCrawlerRequest	sent	message getCrawlerRequest {
				string created_at = 1;
				string id_str = 2;
				string full_text = 3;
				Entities entities = 4;
				Place place = 5;
				Retweeted_Status retweeted_status = 6;
				int64 retweet_count = 7;
				int64 favorite_count = 8;
				bool possibly_sensitive = 9;
				string lang = 10;
				}
				
				message Entities {
				google.protobuf.Any hashtags = 1;
				google.protobuf.Any symbols = 2;
				google.protobuf.Any mentions = 3;
				google.protobuf.Any urls = 4;
				google.protobuf.Any media = 5;
				}
				
				message Place {
				string id = 1;
				string place_type = 2;
				string name = 3;
				string country = 4;
				}
				
				message Retweeted_Status {
				string created_at = 1;
				string id_str = 2;
				string full_text = 3;
				Entities entities = 4;
				Place place = 5;
				int64 retweet_count = 6;
				int64 favorite_count = 7;
				bool possibly_sensitive =8 ;
				string lang = 9;
				}
					GRPC	TA (as a server)	data per tweet

160.40.52.155:50052		getCrawlerResponse	received	string query = 1;	GRPC	CR (as a client)	ACK from TA

Emotion Analysis

IP		Message	Message Type	Message content	Content Type	Useful for	Message info

-		EmotionalTag	Sent	message
				EmotionalTag{
				int32
				emotional_state = 1;
				int32 valence = 2;
				int32 arousal = 3;
				int32 time = 4;
				}
					gRPC	GH,
						Unity, KB
							Contains predicted emotional analysis data

		ACK	Received	message ACK {
				int32 ack_tag = 1;
				}
					gRPC	EA	Response from server. (A local server is considered to test the communication. The messages could be changed in the future based on the new requirements.)

VR Tool

IP		Message	Message Type	Message content	Content Type	Useful for	Message info

		new_EEG_tag	Received	message EmotionalTag{
				int32 emotional_state = 1;
				int32 valence = 2;
				int32 arousal = 3;
				int32 time = 4;
				}
					GRPC	EA (as Client)	Contains the new EEG TAG

		new_Config	Sent	message GetChangeRequest{
				string project_id = 1;
				string last_change_id = 2;
				new_EEG_tag tag = 3;
				}
				message new_EEG_tag{
				string subject_id = 1;
				string experiment_id =2;
				string project_id = 3;
				google.protobuf.Timestamp timestamp = 4;
				int32 EEG_Tag = 5;
				int32 valence = 6;
				int32 arousal = 7;
				SubjectPositionInformation position_info = 8;
				}
				message SubjectPositionInformation {
				float pos_x = 1;
				float pos_y = 2;
				float pos_z = 3;
				float rot_x = 4;
				float rot_y = 5;
				float rot_z = 6;
				}
					GRPC	KB (as Server)	Contains EEG and user position

Style Transfer

IP	Component	Message	Message Type	Message content	Content Type	Useful for	Message info	Comments
160.40.52.169:50051		STrequest	Sent	message STrequest{ int32 model_id = 1; int32 styleimage = 2; }	gRPC		Contains 3D model's id needed to get related images and the id of the style image	**Needs DS.getImages( ) in order to get the related to the 3D model images**
160.40.52.169:50051		STresponse	Received	message STresponse{ string output_path = 1; }	gRPC		Contains the path that the stylized image(s) is/are saved	**Needs DS.saveImages( ) in order to save the stylized images**