



PrismArch

PrismArch

Deliverable No D1.3

Cross-disciplinary perspective and multi-simulations embedded in VR

Project Title:	PrismArch - Virtual reality aided design blending cross-disciplinary aspects of architecture in a multi-simulation environment
Contract No:	952002 - PrismArch
Instrument:	Innovation Action
Thematic Priority:	H2020 ICT-55-2020
Start of project:	1 November 2020
Duration:	24 months



Project funded by the European Community under the H2020 Programme for Research and Innovation.



Deliverable title	Cross-disciplinary perspective and multi-simulations embedded in VR
Deliverable number	D1.3
Deliverable version	Beta
Contractual date of delivery	31 July 2022
Actual date of delivery	1 September 2022
Deliverable filename	PrismArch_D1.3_v5
Type of deliverable	Report
Dissemination level	PU
Number of pages	122
Work Package	WP1
Task(s)	T1.3
Partner responsible	Zaha Hadid Virtual Reality Group (ZHVR)
Author(s)	Helmut Kinzler (ZHVR), Daria Zolotareva (ZHVR), Risa Tadauchi (ZHVR), Aleksandra Mnich-Spraiter (ZHVR), Jeg Dudley (AKT-II)
Editor	Helmut Kinzler, Daria Zolotareva
Reviewer(s)	Spiros Nikolopoulos (CERTH) , Martin Brösamle (ETH), Christoph Hölscher (ETH)

Abstract	Report on Congregated Multidisciplinary design and simulation, conceptualising the functional and operational requirements for the PrismArch Platform.
Keywords	Cross-disciplinary platform, multidisciplinary platform, unified data space, operational scenarios, architecture, structural engineering, MEP engineering, user interface, operational mock-ups

Copyright

© Copyright 2020 PrismArch Consortium consisting of:

1. ETHNIKO KENTRO EREVNAS KAI TECHNOLOGIKIS ANAPTYXIS (CERTH)
2. UNIVERSITA TA MALTA (UOM)
3. ZAHA HADID LIMITED (ZAHA HADID)
4. MINDESK SOCIETA A RESPONSABILITA LIMITATA (Mindesk)
5. EIDGENOESSISCHE TECHNISCHE HOCHSCHULE ZUERICH (ETH Zürich)
6. AKT II LIMITED (AKT II Limited)
7. SWECO UK LIMITED (SWECO UK LTD)

This document may not be copied, reproduced, or modified in whole or in part for any purpose without written permission from the PrismArch Consortium. In addition to such written permission to copy, reproduce, or modify this document in whole or part, an acknowledgement of the authors of the document and all applicable portions of the copyright notice must be clearly referenced.

All rights reserved.

Deliverable history

Version	Date	Reason	Revised by
1	01/03/2022	Table of Contents	Risa Tadauchi,Aleksandra Mnich-Spraiter
2	25/03/2022	Beta version	Daria Zolotareva
3	28/05/2022	Chapters 1-3 revision	Daria Zolotareva, Helmut Kinzler
4	16/08/2022	Revision to accommodate new mock-ups	Risa Tadauchi, Aleksandra Mnich-Spraiter
5	26/08/2022	Final version	Helmut Kinzler, Daria Zolotareva

List of abbreviations and Acronyms

Abbreviation	Meaning
AEC	Architecture, Engineering and Construction
AI	Artificial Intelligence
AR	Augmented Reality
BIM	Building Information Modelling
CA	Client Approved
CAD/CAM	Computer-Aided Design & Computer-Aided Manufacturing
CD	Cross-disciplinary
CWS	Collective Work Sphere
CWS_AF	Collective Work Sphere_Agreement Form
D_A	Discipline Architecture
D_SE	Discipline Structural Engineering
DO	Design Object
IDQS	Information Display and Query System
IDS	Internal Discipline-specific
IH	Immersed Human
LD	Lead Designer
LD_A	Lead Designer - Architecture
LD_MEP	BIM Manager/Project Coordinator
LD_S	Lead Designer - Structural
MEP	Mechanical, Electrical & Public Health
PA	PrismArch
PAS	PrismArch Sphere
PD	Project Director
PD_A	Project Director - Architecture
PD_MEP	Project Director - MEP
PD_S	Project Director - Structural
PM	Project Manager

POV	Point of View
PWS	Personal Work Sphere
QS	Quantity Surveyor
QUI	Query User Interface
QUICM	Query User Interface Central Model
Recognisable	includes cultural and intellectual knowledge
SD	Senior Designer
SD_A	Senior Designer - Architecture
SE_MEP	Senior Engineer - MEP
SE_S	Senior Engineer - Structural
SL	Sphereing Level
UI	User Interface
ZHVR	Zaha Hadid Virtual Reality

Executive Summary

Deliverable D1.3 builds upon the cross-disciplinary perspectives of the AEC industry discussed in D1.1 and D1.2, by providing a more detailed description of the prerequisites for multidisciplinary information exchange and collaborative design platform functionalities that must be implemented to achieve PrismArch.

The topics cover the general usability guidelines (Section 1), technical requirements for how multidisciplinary data should be managed inside PrismArch (Sections 2), and user-oriented requirements for data should be managed, visualised and interacted with inside PrismArch (Sections 3-6).

Section 2 begins by outlining the core data management requirements for multidisciplinary design and simulation. Various PrismArch-specific requirements that both enable and regulate the unified design space are then described in detail.

Section 3 describes the core functionality of the data management system inside PrismArch and offers an analysis of information display methodologies aimed at arriving at the most efficient means of representing metadata inside PrismArch.

Section 4 looks holistically at the user scenarios established in D6.1 and the requirements for the VR experience set out in Section 3 of D1.2, to arrive at the core functionality necessary to prove the PrismArch concept. The fundamental tasks and actions described in this section informs the PrismArch VR interface proposals in Section 5 and operational mock-ups in Section 6.

Table of Contents

Introduction

- 1. Usability and Performance Guidelines**
- 2. Congregated multidisciplinary design and simulation**
 - 2.1. PrismArch DNA
 - 2.2. Tagging
 - 2.3. Sphereing, Sphereing Levels, and Managing IP
 - 2.3.1. Meeting Sphere
 - 2.4. Sphereing Process
 - 2.5. (Data) Security and (User) Authentication [at onboarding]
 - 2.5.1. PrismArch Licensing
- 3. Information, Display and Query System (IDQS)**
 - 3.1. Information
 - 3.1.1. User Comments
 - 3.2. Information Display Methodologies
 - 3.2.2. Metagraph Concept
 - 3.3. Query System
- 4. User Interface Operational Scenarios**
 - 4.1. Initiating a project
 - 4.1.1. Onboarding: first-time user / organisation
 - 4.1.1.1. Introductory tutorial
 - 4.1.2. Creating a new project; adding team; adding data
 - 4.1.2.1. Hosting the project
 - 4.1.2.2. Onboarding other contractual members
 - 4.1.3. Frequent user experience
 - 4.2. Introductory meeting; distributing tasks
 - 4.3. Discipline-specific production
 - 4.4. Meeting for review and sign-off on production
- 5. Proposals for the VR User Interface: classification and layout**
 - 5.1. Primary gesture-based Interaction
 - 5.2. Tool-based UI
 - 5.2.1. In-app guidance and annotation
 - 5.2.2. Wrist UI
 - 5.2.3. Preview Monitor and Keyboard (personal dashboard)
 - 5.2.4. Metagraph UI
 - 5.2.4.1. Metagraph visual aids: orbits and variables axis
 - 5.2.4.2. Nodes
 - 5.2.4.3. Labels
 - 5.2.4.4. Connectors
 - 5.2.5. Multi-tool
 - 5.2.5.1. Content Creation

- 5.2.6. Clipping tool
- 5.3. PrismArch User Experience Sequence
- 5.4. Environmental UI
 - 5.4.1. Dynamic scaling and transitions of environments
 - 5.4.2. Avatar-related environmental indicators in co-presence environments
 - 5.4.2.1. Primary avatar-based environmental functions
- 6. User Interface Operational Mock-Ups**
 - 6.1. Selecting objects
 - 6.2. IDQS - Metagraph operation: query
 - 6.3. Multi-tool operation: transform of an object (in production mode)
 - 6.4. Environmental operation
- 7. Summary and conclusion**
- 8. Bibliography**
- 9. Images**

Introduction

This deliverable is the final version of the report on the definition of principles, rules, constraints and interfaces for cross-disciplinary and multi-simulation perspectives in VR. As the final deliverable of WP1, it aims to bring together the knowledge synthesised in previously submitted D1.1, D1.2, informed by the use cases established in D6.1 and the ongoing consortium-wide discussions.

In order for multiple disciplines to congregate in a single design space, it is necessary to regulate three key aspects of this unified design space: 1) documented information exchange, involving all informational transactions; 2) ad hoc and flexible groupings of information forming semantic structures and entities needed by the designers to create and formulate project information; and 3) identifiable data to demarcate its ownership and its associated party's IP.

In this document, we revisit the essential sphereing mechanism for project data, first discussed in the case study discussion of D6.1 and elaborated in D1.2, and show how it applies to all three above-mentioned key aspects of a single multidisciplinary VR environment. Through the vantage of sphereing, we look at how users from various disciplines are authorised to access the PrismArch database. We also use sphereing to address issues of information management and data security for ensuring the professional liability for authored data inside the multi-author database.

The D1.3 deliverable is based on the revised version of D1.2 after the first evaluation of PrismArch platform leading the 2nd phase developments of WP2-5. In this report, we use mock-ups to examine how the VR experience should be organised to facilitate the needs of each exemplary design discipline (Architecture, MEP, and Structure), to enable the minimum requirements for professional collaboration across these disciplines inside VR. We define how project information and design assets are recorded, queried, retrieved, organised by various variables, and visualised in structured formats based on the users' needs.

By bringing merged interdisciplinary information into immersive VR, we are bringing project data into a novel environment for users accustomed to working in a highly siloed and segmented 2D projection space. From a user perspective, we recognise the need for intuitive interactivity with the data. The UI must respond to the various immersed and co-present disciplines and serve each of their particular foci and workflow aspects. The multidisciplinary merged dataset presents an opportunity to develop a holistic approach to data access and procedures as well as a centralised UI with a unified aesthetic.

The PrismArch development workflow thus involves learning from what is currently defined as software peripheral to the PrismArch system, migrating necessary functionalities into the PrismArch platform, and developing a novel interface to enable stereoscopic immersive interaction, while still allowing connectivity to outside softwares. This development will greatly aid architectural projects by eliminating current compartmentalisation and information thresholds.

In this document we further develop mock-ups for the functionality of such a holistic UI to enable a consistent and intuitive way of interacting with the Prism Arch system and the totality of the contained user generated project information (including dynamic transitions, etc); lending a unique and functional brand identity to the system's UI and other user-readable aspects, and helping to distinguish PrismArch from other professional software platforms.

1. Usability and Performance Guidelines

In this section, we provide a top-level qualitative description of the requirements for the PrismArch user experience.

Looking toward general usability criteria, we understand that a controlled cognitive load for the user, task efficiency, ease of learning, and subjective satisfaction are essential benchmarks for the development of PrismArch. To this, we must add the general compatibility of the software to the continuous maintainable professional workflow, as defined by the capacity of the software to meet the workflow needs of its user base.

User comfort is key to the successful uptake of the software, and forms the basis of performance criteria. The platform must meet the unique requirements for real-time stereoscopic VR simulation: it must be optimised for low latency, high frame rate, and high resolution. The user should not feel motion sickness at any point, be able to orient themselves inside the immersive environment, and feel safe inside the virtual environment (as earlier outlined in D3.1 Section 2.2: Navigation in Virtual Environments).

General requirements for accessibility and inclusiveness call for recognising the variants of the users and accommodating the extremes, such as including a mode of interaction for the visually impaired. Additionally, it is important to acknowledge linguistic preferences for the global audience. While this has not been the focus of the research to-date, the diversity of the users would need to be addressed in future developments to make it as inclusive as possible.

Customization of the PrismArch environment and user interface is essential from the perspective of multidisciplinary and user comfort. Structural engineering, MEP, and Architecture all have different workflows that need to be accommodated. This could be achieved through system-level customization, aided by APIs that would allow users to inject their additional required tools for professional usability.

Due to the wide projected range of user profiles and disciplines using the final PrismArch platform, the interface has to be modifiable to meet the individual user preferences and industry-based workflows. Ergonomics guidelines¹ must apply to the virtual workspace, with adjustable tool layouts to meet the individuals' physical needs, such as presets for left- or right-handed users, and an adjustable height of the personal avatar to facilitate the most accurate perspective of the immersive space.

Integrity and clarity of the visual information inside PrismArch is a core requirement, and discussed further in the next section of this document, Section 2, Congregated multi-disciplinary design and simulation. In brief, the visual representation of all data (including metadata), clarity about who authored each asset, tag, etc., and visually differentiated core PrismArch classes (design objects, UI, metadata, and the immersed human) are essential to help users make sense of and utilise the immersive PrismArch environment. Visual clarity, aesthetic uniformity, cohesion of the visual cues

¹ We here refer to existing guidelines for the ergonomic design of physical spaces [see Architectural Graphic Standards (Pressman, Andy. Architectural Graphic Standards. Hoboken, N.J: John Wiley & Sons, 2007.); Neufert (Neufert, Ernst, Peter Neufert, Bousmaha Baiche, and Nicholas Walliman. *Architects' Data*. Oxford: Blackwell Science, 2000.); and British Standard (BSI Standards Catalogue. London: The Institution, 1987.)].

and interactions, and uniform quality of the performance characteristics across all tools and spaces are essential, driving the development of the Unreal engine mock-ups by ZHVR.

Lastly, we must address interoperability between technical platforms as well as compatibility with a variety of physical devices that will be used to access PrismArch. Multi-platform output is a common standard in platform development, enabling ease of access for a wider audience and user-base. Although this falls outside the scope of the MVP, the PrismArch should ideally be able to run on different operating systems. It is also important to provide access to PrismArch on 2D screens (phone, laptop, tablet) while on-site or in transit without immersive gear. Meeting attendees should be able to call in from their phones or tablets via telephone line or tablet. While they would not be fully immersed as virtual avatars, it would be important for the participating callers to have a generic indicator of telepresence.

Industry leaders have realised that in order to be successful, their metaverse efforts will need to be built on open interoperability standards. The Multiverse Standards Forum was launched earlier this year (Figure 1a) to foster the development of interoperability standards for an open and inclusive metaverse, and accelerate their development and deployment through pragmatic, action-based projects. Its mission is “to encourage and enable the timely development of open interoperability standards essential to an open and inclusive Metaverse.”² It would be important for PrismArch development to keep abreast of this effort and adhere to any metaverse interoperability standards as they take form.



Figure 1a. Metaverse Standards Forum and some of its members. For a full list of members, please visit the website <https://metaverse-standards.org/members/>.

² *Metaverse Standards Forum Frequently Asked Questions*, The Khronos Group Inc., 2022, <https://metaverse-standards.org/faq/>

2. Congregated multidisciplinary design and simulation

The users of PrismArch are a wide range of individuals operating contractually under the auspices of various companies in the AEC industry. The diverse set of users also includes corporate or individual clients, licensed quantity surveyors, contactors, appointed members of local governing bodies, and members of the public. As a multidisciplinary design and review space, PrismArch is designed to facilitate collaboration between parties and maintain a traceable thread of project information, called the Golden Thread record in the UK. For more information on the Golden Thread record requirement, see the text box below.

UK law sets out that the dutyholder must maintain and manage a golden thread record of building safety information and ensure compliance with building regulations. The golden thread will be a source of evidence to show how building safety risks are understood and how they are being managed on an ongoing basis. The golden thread must be accurate and trusted so that relevant people use it. The information produced will therefore have to be accurate, structured, and verified, requiring a clear change control process that sets out how and when information is updated and who should update and check the information.

The golden thread information needs to be formatted in a way that can be easily handed over and maintained over the entire lifetime of a building. In practical terms, this is likely to mean that it needs to align with the rules around open data and the principles of interoperability – so that information can be handed over in the future and still be accessed. Information should be able to be shared and accessed by contractors who use different software and if the building is sold the golden thread information must be accessible to the new owner.

Preserving the golden thread does not mean everything about a building and its history needs to be kept and updated from inception to disposal. The objective of the golden thread is building safety and therefore if information is no longer relevant to building safety it does not need to be kept. The golden thread, the changes to it and processes related to it must be reviewed periodically to ensure that the information comprising it remains relevant and useful.

Excerpt from *Golden Thread Report*, Published by the UK Government 21 July 2021³

Collaboration requires bringing together different sources of IP. For the purpose of the two-year research activity, we have focused on a sample of users comprising architectural professionals, structural engineers, and MEP engineers. Combined, the three respective datasets and industry-based workflows begin to suggest the level of complexity in a typical architectural project.

Interoperability is at the heart of the PrismArch platform. In order for disciplines to communicate and exchange information about their specific domains, a common framework must be established involving a shared format for the exchange and schematisation of the associated knowledge

³ GOV.UK *Golden Thread Report*, 21 July 2021

<https://www.gov.uk/government/publications/building-regulations-advisory-committee-golden-thread-report/building-regulations-advisory-committee-golden-thread-report#golden-thread-definition>

domains. Stemming out of the D1.2 deliverable and ensuing discussions, the extendable Speckle database was chosen as the underlying framework for the PrismArch platform in Unreal VR, to store and manage the object-level inputs from the three participating AEC partners.

Inside the shared domain, information must be clearly identifiable both to the users and to the system in order to enable a wide range of essential semantic definitions and distinctions that ensure the usability of the platform in a professional environment. This involves managing the read and access rights to portions of the data, and making sure that all information remains usable: searchable, traceable and visually recognisable inside the VR experience. The specifics of the semantic distinctions will be discussed in detail later in the document (see Section 2.2 [Tagging](#)).

It is essential to identify the author and owner of the professional content. Because this information is being stored in a shared domain, we need to demarcate the author groups and manage access to the authored information. One aspect to the identification requirement is persistent and must be imprinted on the authored objects (see Section 2.1 [PrismArch DNA](#)). The other system of identification is more malleable, allowing the assignment of notes and creation of families of objects (or sets of nested items) and creating links and connections between data elements (see Sections 2.2 [Tagging](#); 2.3 [Sphering](#); 2.4 [Sphering Process](#)). We are not only considering design objects, but also events, user groups, points, texts, and other types of data, as well as composites of all the preceding elements combined in different proportions. How this information is retrieved, displayed, and interacted with in VR will be developed in Section 3, [IDQS \(Information Display and Query System\)](#).

The goal of PrismArch is to enable multidisciplinary collaboration by facilitating the amalgamations of multidisciplinary inputs, ensuring that when information is brought together through sphering, it remains identifiable and protected throughout the project's development. The PrismArch software must allow users to amalgamate and make composites of multidisciplinary information without losing the authorship DNA signatures. Design stage submissions are a good example of such multi-authored information packages, which are prevalent in the AEC industry. The ownership rights of the various parts of the composite cannot be overridden by any party, even the party that ultimately legally "submits" the information to the client.

In summary, the ownership of authored information inside PrismArch requires clearly identifiable data, requiring all authored data to contain a unique identifier, or DNA, on all objects and entities to ensure that data and IP is protected for legal and financial reasons; and delineated data protocols, which we call sphering, to ensure that all information can be specified and communicated inside the multi-participant discourse of a architectural project. Sphering and tagging of information enables the formation of emerging requirements for ad hoc data hierarchies; the possibility of multilateral queries, giving users the ability to traverse the entire dataspace and search the contributions of all partners; and the ability to view query results in different topological organisations, ensuring the usability of the unified design and review space.

The following is an outline of the general requirements for a congregated multi-author professional environment:

TECHNICAL REQUIREMENTS (see Figure 2a):

- Unified database
- Data interoperability

1. ability to congregate data coming from proprietary discipline-specific professional softwares;
 2. ability to export into existing professional software formats
- Data demarcation and management of IP
 1. authorship
 2. ownership
 3. access

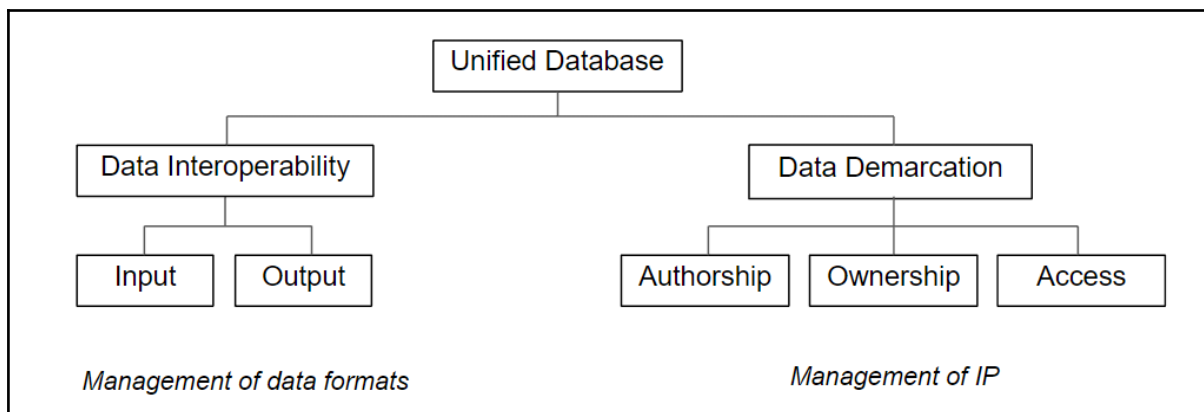


Figure 2a. Technical requirements for the PrismArch platform

The above leads to a traceable project record, mitigates loss of information, and allows potential problems to be addressed and resolved in a timely manner, saving costs.

In addition to these technical requirements, we set out user-centred requirements for how the data should be visualised and interacted with inside the VR environment, which are further developed in Sections 3-6 of this document.

USER-CENTRED REQUIREMENTS:

I. *Holistic approach to Multidisciplinary Data (back-end functionality)* (see Figure 2b)

- Interactive functionality: information management system (IDQS)
- Visual demarcations for differentiating information in VR
 1. distinguishing the core PrismArch classes;
 2. distinguishing project data based on discipline, authorship, ownership, etc.

II. *Multidisciplinary Immersive Functionality (front-end architecture)* (see Figure 2c)

- Virtual environment
 1. Establishing the requirements for the purpose-built PrismArch environments
 2. Designing the immersive virtual environments (Mock-ups and prototypes)
- User Interface: UI elements for multidisciplinary interface
- Avatar creation and customization
- Communication and interaction
- Interdisciplinary functionality
 1. sphereing
 2. interdisciplinary administration, workflow analytics & synchronisation
 3. Project-level customisation (of virtual environment, UI, avatar, and communication)

- Discipline-specific and personal-level functionality
 1. sphereing
 2. disciplinary admin
 3. disciplinary and individual customisation (of virtual environment, UI, Avatar, and communication)
 4. Discipline-specific methodologies, workflows, visualisation styles, and data types (variations of inputs and outputs)

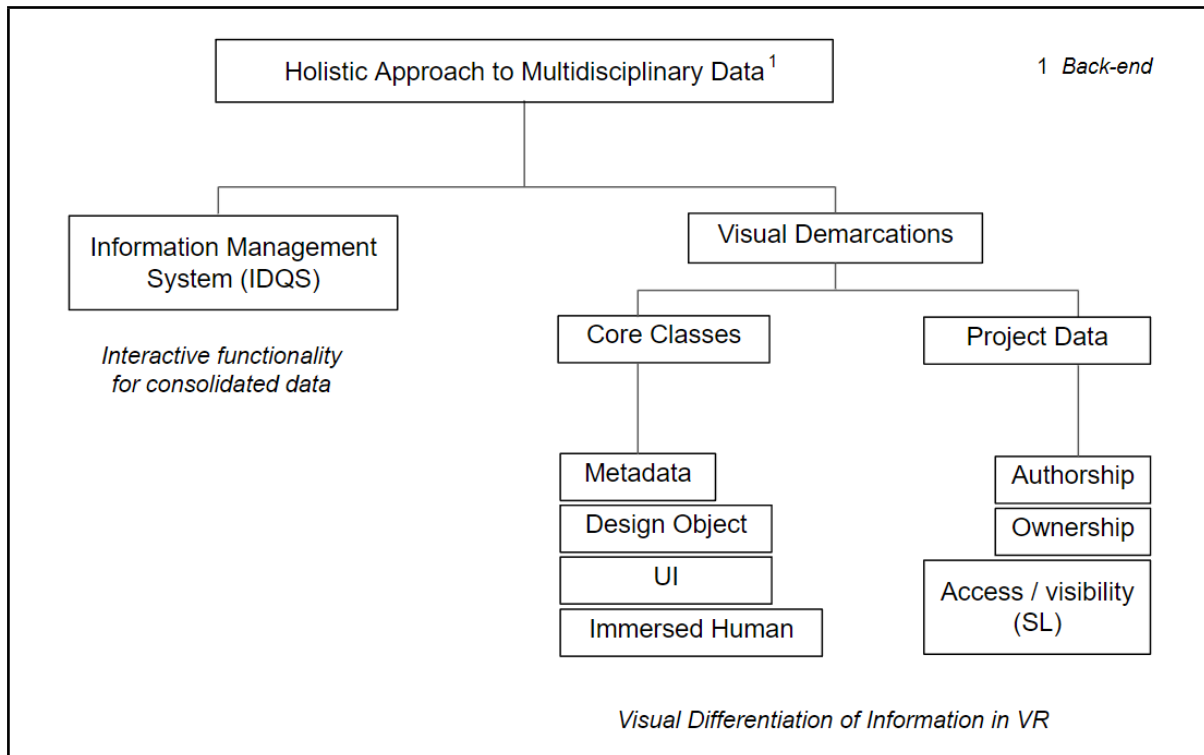


Figure 2b. User-centred requirements for a holistic immersive multidisciplinary project data environment, back-end

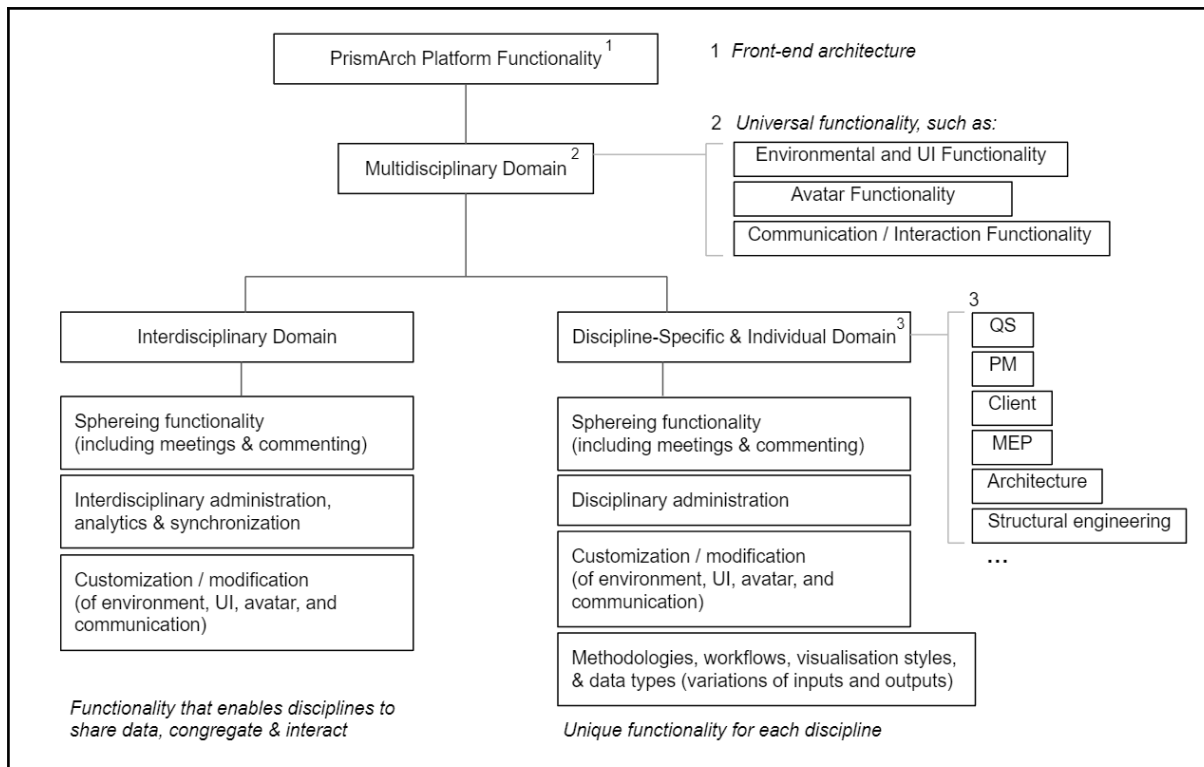


Figure 2c. Front-end architecture for PrismArch Platform Functionality

2.1 PrismArch DNA

We have established the need for persistent and immutable identification of all project data inside PrismArch in D1.1 and D 1.2 and in Section 2. [Cross-disciplinary perspective on multi-simulation](#) of the current document. To be traceable, all data must have an immutable DNA assigned to it at the moment of its instantiation in the database. Each data element (each object/asset) brought into PrismArch needs a unique identifier containing a unique code, PrismArch Project ID, the source of the data (original software), timestamp, and the person and company who authored it. The author, timestamp, and any immutable asset-related metadata from the originating file, taken together, makes up the PrismArch DNA (Figure 2.1a), also referred to as *ID* and *signature* in previous deliverables.

This requirement applies to every design object, user, and every new sphere instance (e.g. any newly created object group from loaded assets or objects immersively modelled using PrismArch’s design tools; any set of images projected onto whiteboards [saved as a single entry ‘whiteboard layout’], virtual 3d model arrangement, or recorded walking path [in the form of user transformation data (location, rotation, scale)]; a set of immersively-created comments; or any combination of the above-listed into one searchable entity). These new immersively-created connections are all examples of PrismArch spheres, discussed in Section 2.3 of this document. For example, a lead designer working on a project may need to combine a set of objects into a folder for submission to the client. That person will group all relevant assets, creating a new sphere and accordingly, a new node in the metagraph, tagging it “Design_Stage_Submission”. This entry will have its own author (the lead designer) and a unique DNA associated with it. It is important to note that the original objects referenced in “Design_Stage_Submission” will need to maintain their own original DNA signatures; inside any sphere, the contained object DNA *must not* be overwritten.

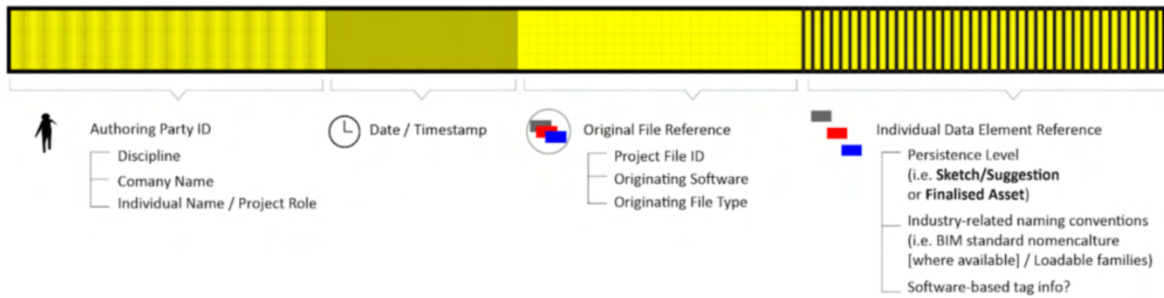


Figure 2.1a: Early PrismArch asset DNA composition idea (as described in D.1.1).

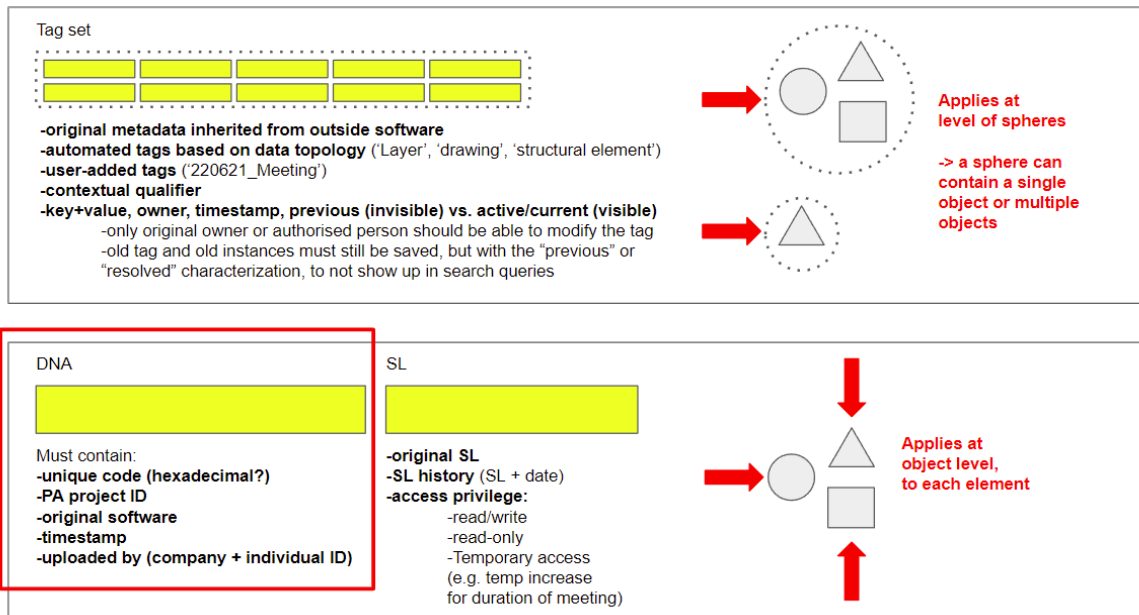


Figure 2.1b. Updated understanding of the identifiers required for PrismArch's unilaterally searchable database.

Figure 2.1b shows our updated understanding of the PrismArch identifiers – the DNA, sphering level, and tag set – along with what information should be contained in each of the identifier categories and where these identifiers apply topologically. The DNA and sphering level are contained on the object level, while tagging applies to the superimposed semantic layer where meaning is mapped onto the PrismArch-created spheres (discussed in the next Section 2.2 *Tagging*)⁴. The method for integrating the DNA Signature was previously described in *D1.2 Section 2.4, Cross Disciplinary Foundation for Ontology*, and is included in the box callout below for reference.

The structure of the Prism_Signature could be kept intentionally basic: it could be a lightweight entity that excludes complex data types, and produces only small increases to the file size, thus ensuring that the creation of Signatures will not slow the flow of information to the Prism Database.

This small entity 'footprint' would also mean that there is the potential for these Signatures to be pushed to a wide array of other sources, beyond the Database. For instance, automated alert systems could be set up to notify users - via email, text message, Slack or other messaging service - when new updates have been received by the Database, and could display the entirety

⁴ In the context of Tagging, *contextual qualifiers* refer to existing associations (contextual relationships) between objects within the PrismArch database. It would be desirable to automate this.

of each Signature, providing users with a compressed history of the commits being made by different parties. These Signatures will also be used to provide a thorough log of work created and changes made during the life of a project, for accountability and 'Golden Thread' requirements (outlined in *D1.1 Section 2.1*).

Schema	Attribute	Example Value
Prism_Signature	CreationDate	2020/06/01
	SphereLevel	2
	CreatorName	Aleksandra Mnich_Spralter
	CreatorRole	Lead Designer
	CompanyName	Zaha Hadid Architects
	Discipline	Architecture
	Sub-Discipline	e.g. Furniture Design and Option
	Project Typology	Residential
	Stage	Concept
	CreatedBySoftware	Rhino 7
	CreatedFromFile	Master model 3D
	DeviceName	PC NAME
	DeviceID	PC NUMBER
Prism_Tag	Name	e.g. Comments, @Helmut.K: can you review the material choice? To Review
Prism_TagConnection	Object ID	e.g. Asset number Element ID (to search within Revit, auto generated number, passed to DataSmith/ Asset Metadata) or GUID (16 digit numbers)
	Tag ID	e.g. Tag number
	Status	e.g. Busy
Prism_InstanceDefinition	User Data	e.g. versions and history

Figure 2.1c. Exemplary chart by AKT-II showing top-level specifications for core PrismArch schemas

Property	Type	Description
AssetId	GUID OR string OR Checksum	Unique GUID or HASH string for this asset. Still TBD.
CreationDate	DateTime	
SphereLevel	int	1 - 5.
PersistenceLevel	enum	Selected from a pre-existing PrismPersistenceLevel type.
CreatorName	enum	
CreatorRole	enum	Selected from a pre-existing PrismArchRole type.
CompanyName	string	Must be empty if user is within SL1. Must be assigned for all other SLs.
Discipline	enum	Selected from a pre-existing PrismDiscipline type. e.g. <i>Architecture</i>
Sub-Discipline	enum	Selected from a pre-existing PrismSubDiscipline type. e.g. <i>Facade Studies</i>
Stage	string	Project Stage.
CreatedBySoftware	enum	Selected from a pre-existing PrismSoftware type.
CreatedFromFile	string	Text string of original file <i>Path</i> - thus includes original folder location, original file name and file extension.
DeviceName	string	Automatically read from the device: PC, laptop, etc.
DeviceId	string	Automatically read from the device.

Figure 2.1d. 'Prism_Signature' Specifications by AKT-II, from *PrismArch_D1.2_v1.0*, p.86

2.2 Tagging

In addition to the immutable PrismArch object DNA, there is a demonstrable need for a flexible system for associating semantic meaning to various subsets and groupings of the data (figure 2.2a). Similar to the way that Web 3.0 evolves the current Web2.0 by aspiring to the idea of a semantic web in a cybernetic evolution of data, we are introducing a unilaterally searchable semantic layer to facilitate design discussion and information exchange inside PrismArch, to enable the transactional aspects of the architectural workflow.

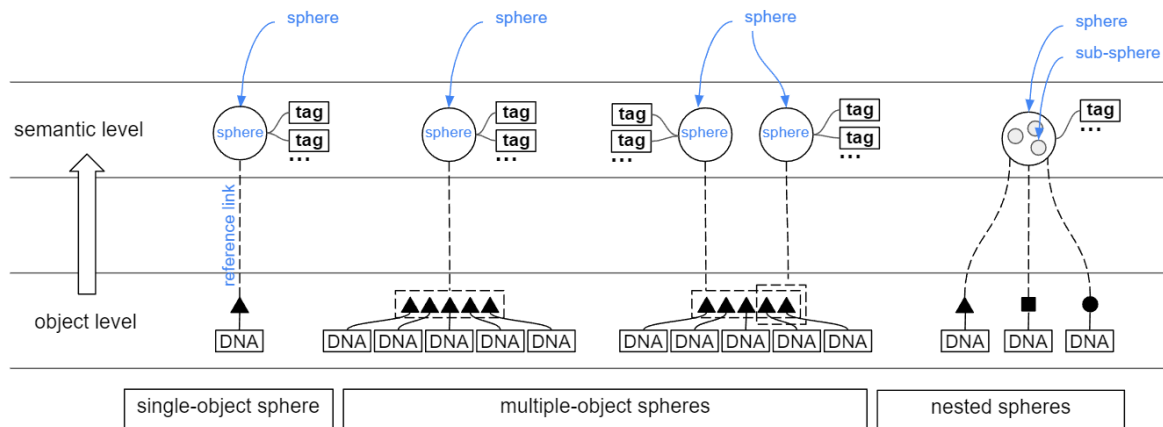


Figure 2.2a. Semantic level tags are applied to functional groupings (spheres) of PrismArch objects, each with their unique object level DNA.

All architectural work falls under either *production* or *transaction*. It is not unusual for architectural design to engage with the object level (root level) during production, while the object is being generated. Any object must first be created or instantiated in 3D before it can be operated on (named, grouped with other objects [sphered], etc.). Root level access is also necessary for maintenance and information purposes. However, the majority of the architectural development discourse and information exchange is transactional – it takes place on the semantic level.⁵

Generally, we can say that design is the process of developing ideas and converging possibilities through dialogue. The process involves: 1. setting out the arguments (initial parameters and design criteria), 2. exemplifying them using simulations, and 3. holding a dialogue for expressing preferences. The architectural dialogue consists of a flow of transactions involving feedback in the form of approval, suggestion for /need of an alternative, or the synthesis of expert input into the design. Key moments of this dialogue are benchmarks for payment and transfer of IP. In summary, the communication of any information – be it an asset or a comment – constitutes a transaction. Even the act of depositing information in a database is a transaction, as this information is stored for future communication.

The tagging requirement was previously discussed in multiple previous Deliverables (most notably within D1.1 Section 4.2 [Taxonomies], D1.1 Section 4.4 [Item 2] and D3.1 Section 4.2 [Scenarios Developed as Part of Project Deliverables] and summarised in D1.2 [2.3.2 Ontology Requirements Outlined in Previous Deliverables]).

In D1.2, AKT II proposed the creation of two tag-related objects that together produce the necessary tagging functionality while working within the existing Speckle 'immutable objectPush-only' paradigm. These self-contained entities that exist entirely *outside* the object(s) that they reference - thus they can be deleted or superseded without deleting or otherwise affecting the object(s) that they refer to. This approach therefore obviates the undesirable requirement to re-upload existing geometric entities to the PrismArch database simply to make minor changes to their attached tags or comments - which could be onerous for a number of

⁵ This suggests that proportionally, there will be more semantic-level descriptions, or *tags*, applied to objects than the sum of the objects themselves.

reasons: increasing data transfers and slowing the system, as well as introducing a point of weakness in data synchronicity.

In D1.2 these two entities were defined as a “*Tag*” and a “*TagConnection*”. Since that submission, further research and development work has been carried out - most notably significant progress on the subject of tagging was made during the May 2022 London plenary - and as a result the entities have been redefined as “*Tag*” and “*TagSet*”.

The two objects would be:

- **Prism_Tag** > an entity comprised primarily of a unique name. Cannot be duplicated. e.g. the “For Review” Tag.
- **Prism_TagSet** > a container entity that stores a list of Tags applied to one or more Prism objects (geometric or otherwise). For each Tag applied, the Tag set also records Who assigned the Tag, and When - i.e. *CreatedByUser* and *CreatedWhen*.

Object is **Immutable**, and Tags are **Externalised outside** the Object.

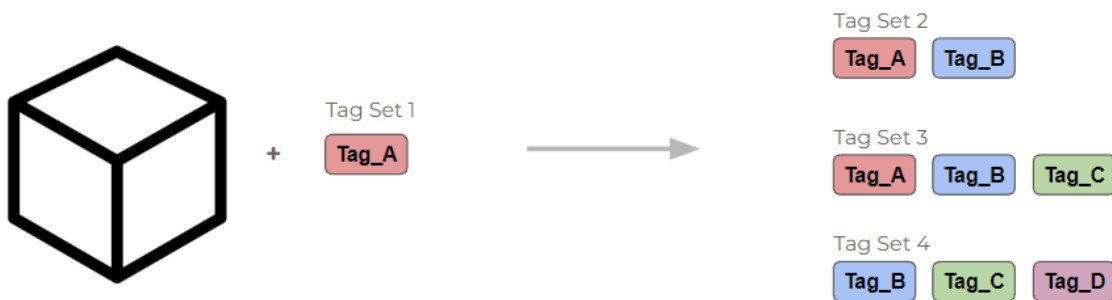


Figure 2.2b. Diagram by AKT-II showing how Speckle Tags could be externalised outside the object.

As new objects are instantiated, their DNA signatures are recorded (at the object level) in each Speckle commit⁶. Next, we must enable PrismArch users to merge various Speckle streams into semantic entities. Unfortunately, the current individual Speckle data entries cannot yet address the way that the information must be semantically grouped and distributed to facilitate these transactions. Therefore, a flexible matrix needs to sit on top of, or map onto, this object-oriented information level, as shown in figures 2.2c-2.2h.

⁶ Speckle Docs: Core Concepts, 2022, <https://speckle.guide/user/concepts.html>, accessed 2 July.

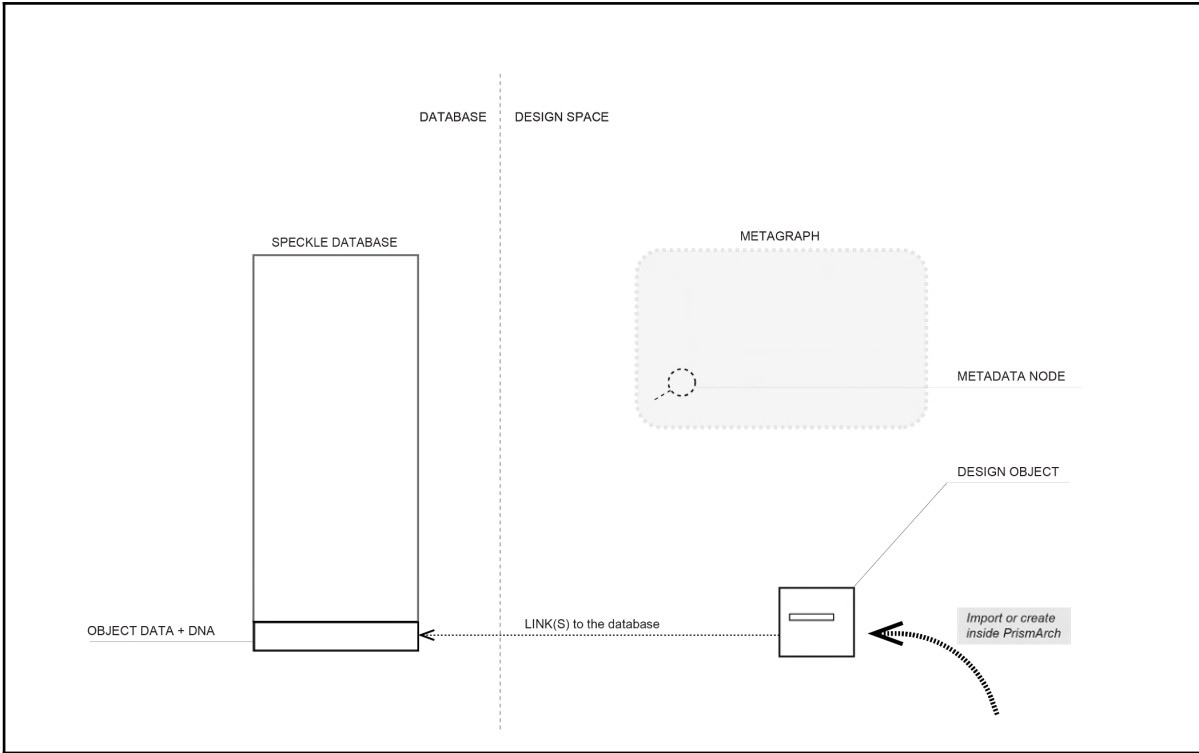


Figure 2.2c. Object level information inside PrismArch

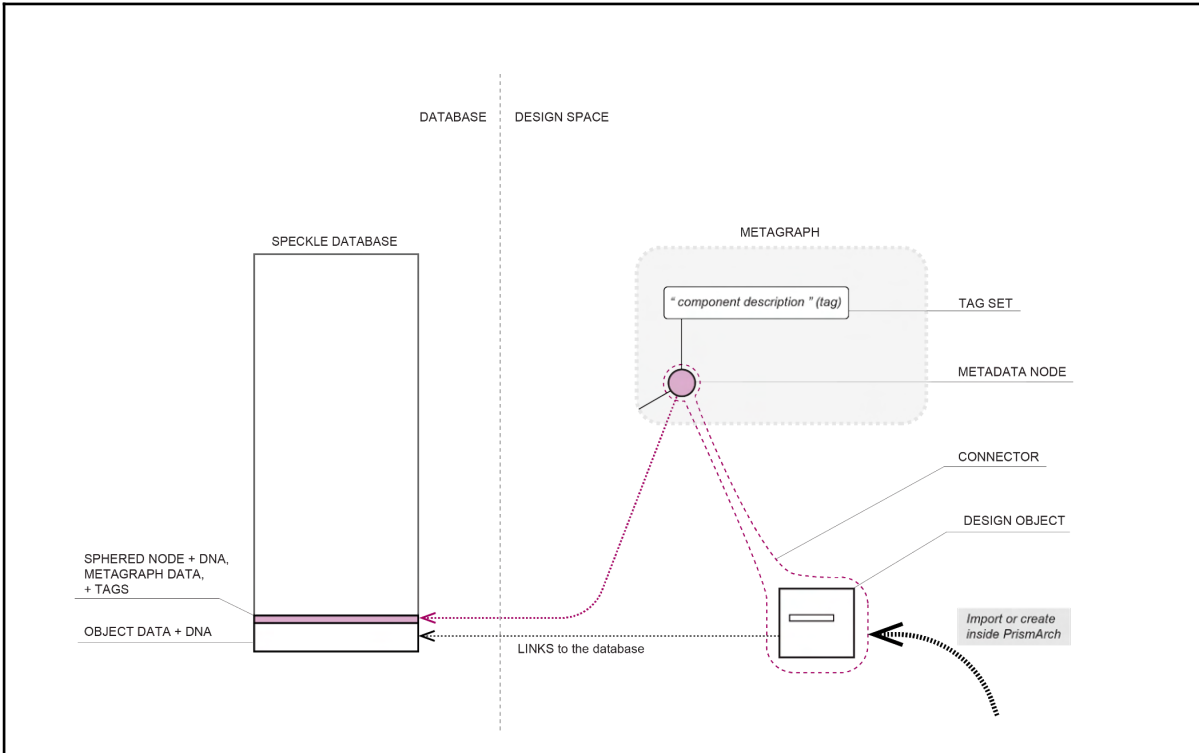


Figure 2.2d. Semantic layer for enabling the architectural workflow

The semantic layer, represented in pink, contains the functionality needed to create meaning from, or bring relevance to, the data entries (e.g. submission folders from various authors, meetings and minutes). The relationships between these higher-level semantic descriptions are not intrinsic to the discrete information packages (i.e. design objects) loaded into the system; the semantic descriptions must sit one level above the object information in order to function/perform.

Importantly, tags are assigned to spheres and not to individual objects. A sphere, represented visually as a node in the metagraph, can contain a single object or multiple objects. Whenever a user spheres something, they automatically have to tag it. In that way, a tag functions the same way as a house name and house number. This concept is shown clearly in the diagrams 2.2e-f below.

In Figure 2.2e, a user creates a new sphere from existing objects, for the purpose of collecting information for submission to the client. New items can be added to this sphere (figure 2.2f) or removed from the sphere without changing its purpose, with the changes tracked in the tag set. This sphere exists on the interdisciplinary sphereing level (SL3), where all partners can input their IP.

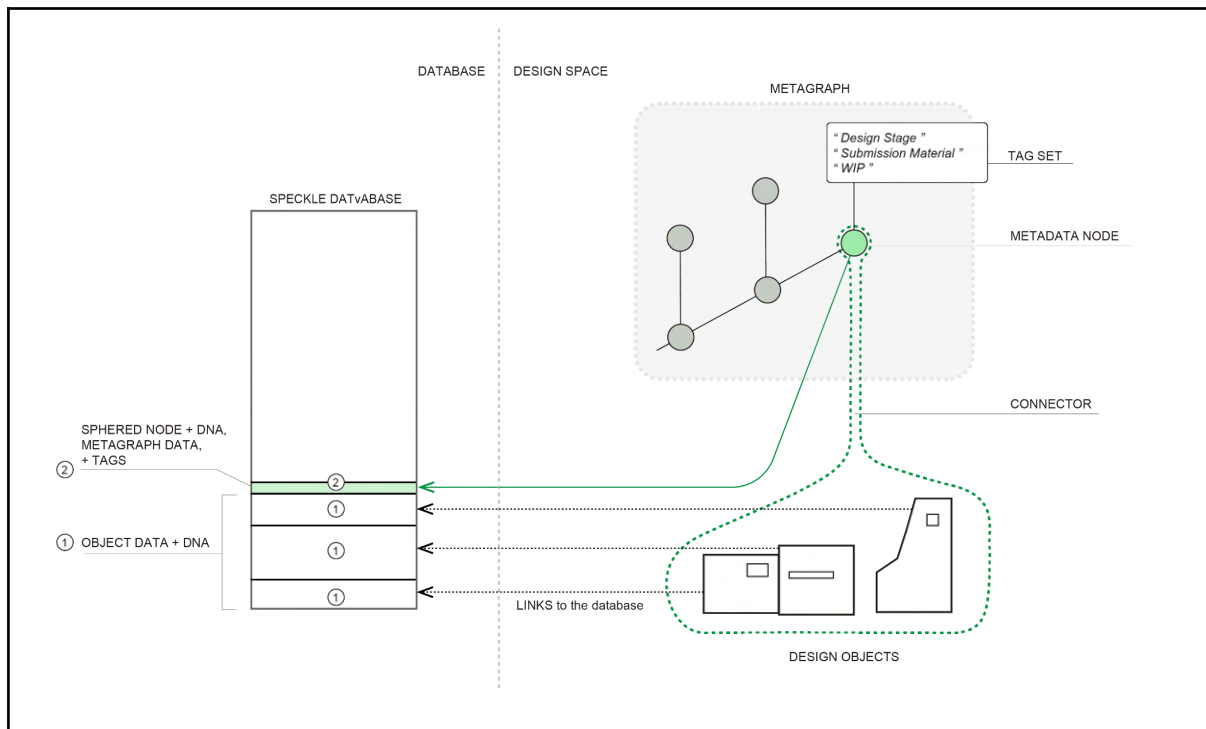


Figure 2.2e. User-created sphere to collate information for submission to client

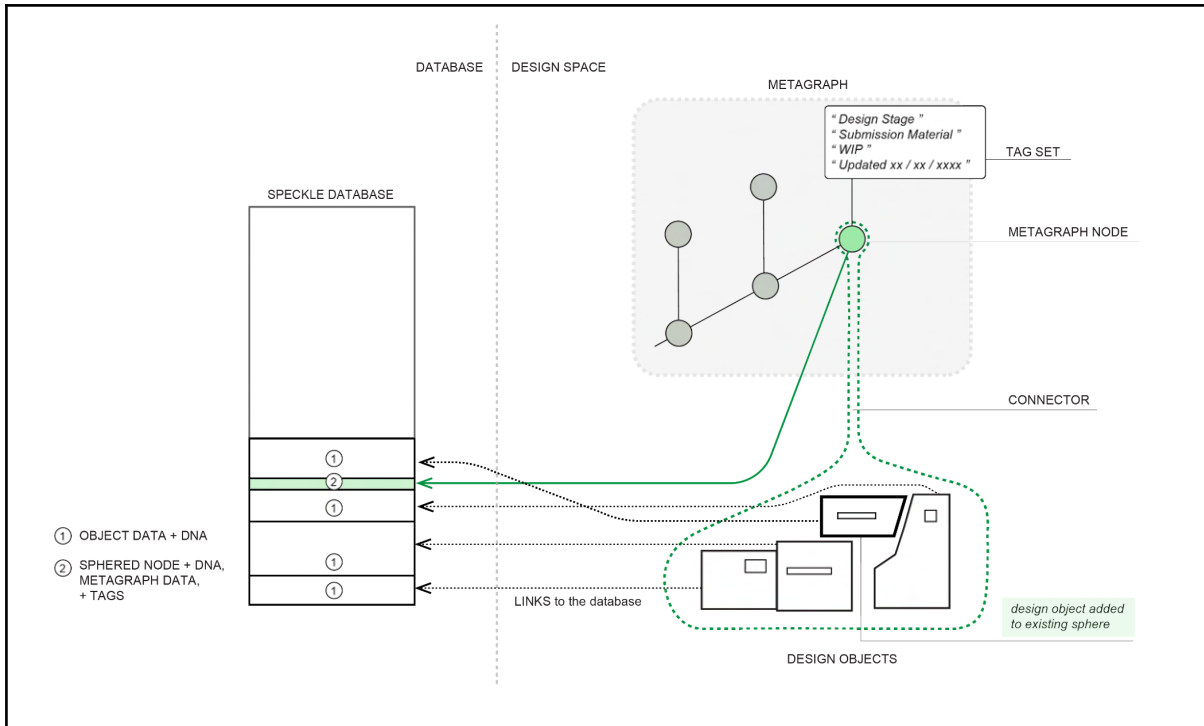


Figure 2.2f. New object added. Throughout the accumulation of data for submission to the client, the purpose of the sphere remains the same.

Once the team is satisfied with the assembled data, a new sphere is created with the same data on SL4 to submit it to the client, as shown in figure 2.2g. This sphere is locked for editing, and forms a record on the golden thread. The tags associated with this new sphere clearly show the design stage, when the data was submitted to the client, by whom this was done, and any other relevant information.

After the client has reviewed the data and is satisfied with the level of the work, they issue written approval that becomes part of the new client-approved sphere, tagged with relevant data including the date approval was received. The email of acceptance and approval constitutes an agreement and marks the completion of a design stage (figure 2.2h). This indicates the transferral of IP to the client, and triggers the payment procedure or protocol.

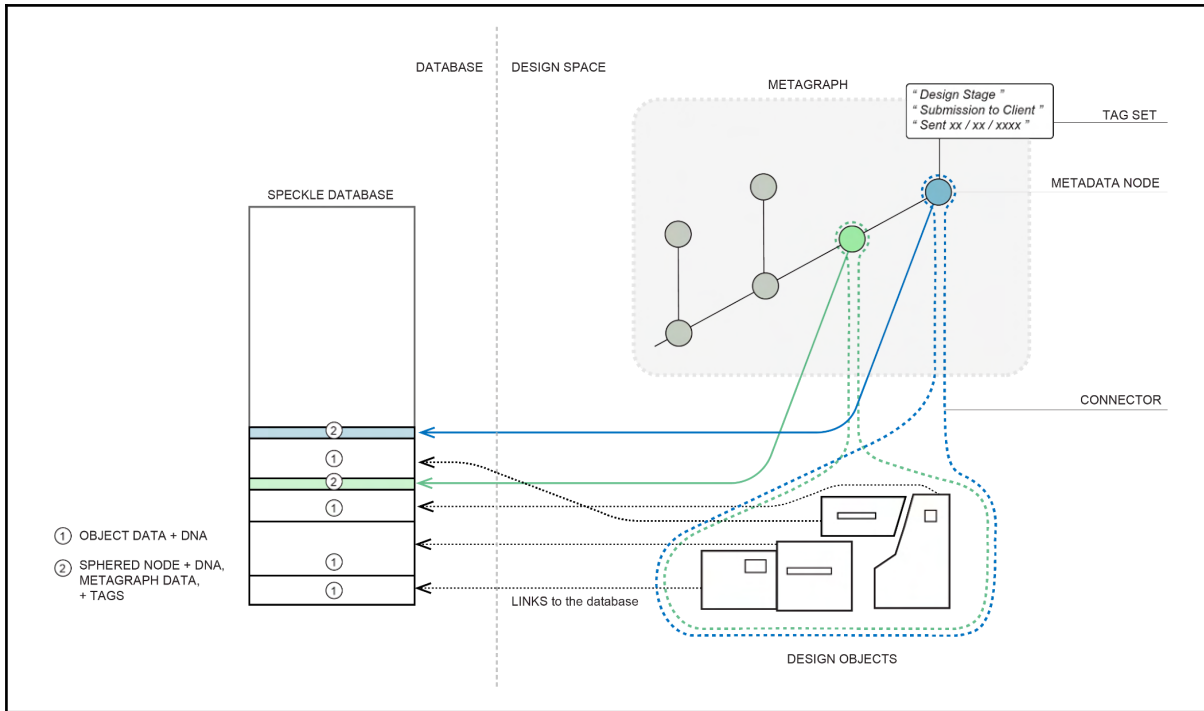


Figure 2.2g. New sphere is created on SL4 visible to the client; the content is locked for editing.

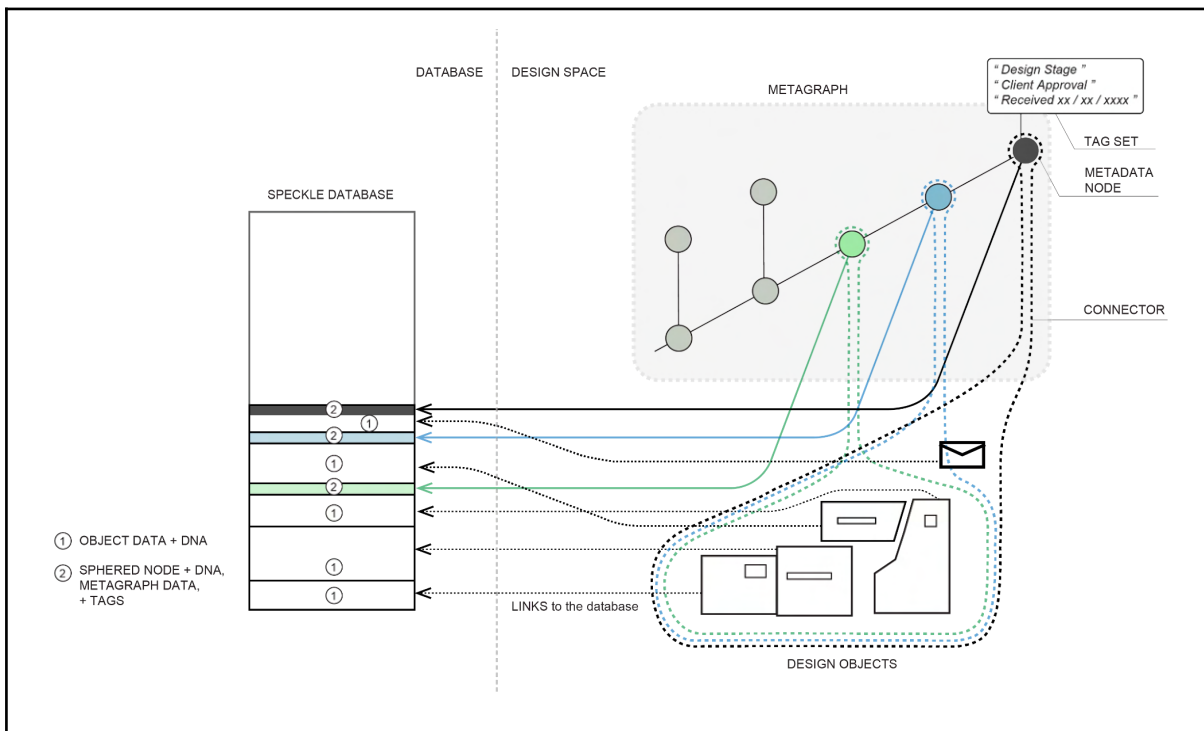


Figure 2.2h. Client approves submission. The sphered data is now owned by the client.

In summary, when communicating design ideas, the user is not dealing with the object-level DNA, but is mainly operating with the data stored on a semantic level. Tagging is essential for working dynamically with loaded discipline-specific data, allowing users to create further layers of searchable information in addition to the data coming from the various softwares. It is this functionality that weaves discrete information sets together into a coherent searchable whole.

We can distinguish between *human-readable* and *discreet* tags. Human-readable tags are human-generated or added with automated tagging support where the system recognizes what the

user is doing and adds or suggests tags. The human-readable tags require a principal tag to be selected automatically, to appear as the title or label of the node on the metagraph. The remaining tags will be presented when the full tag list is opened for review. In the future, it may be possible to curate the title of the metagraph node according to the user-generated query, prioritising one main title over others based on the subject of the search. We can also foresee a need for discreet tags containing back-end information, resulting from the requirement to have machine-readable information that is not necessarily human-readable (i.e. file compatibility information, data formats, etc.).

We see the need for being able to append an unlimited number of tags to the data objects or groups of objects, both manually and automatically with the help of AI. We imagine that automated agents could help human users apply tags by studying and learning from their behaviours. Based on the tags, agents could also potentially introduce new spheres, and make suggestions, such as results of reverse searches (i.e. “find me a 3D model of a chair”).

Additional requirements for the tagging functionality have evolved through workshops with the development partners:

- Tags must be recorded and stay in “memory” even after they are modified.
- They contain “key+value”: owner, timestamp, status (previous [invisible] vs. active/current [visible]).
- Only the original owner of the tag or authorised person can modify the tag.
- Old tags and old instances must still be saved, but as “previous” or “resolved” (and unless searched for specifically, the old tags must not show up for the user).

2.3 Sphereing, Sphereing Levels, and Managing IP

Sphereing and sphereing levels help us manage ownership and access to project data inside PrismArch. While *sphereing* is used to record, access, organise and demarcate assets or combinations of assets in the PrismArch environment according to ownership, the *sphereing level (SL)* is an indicator showing the general level of access of any given object or sphere at any given time.

Sphereing forms guided personal, discipline-specific (internal), and cross-disciplinary constraints on project data and information, and applies to all data existing inside the PrismArch database. By default, the PrismArch platform and the operating system produces a single type of sphere, the PrismArch Sphere (PAS). The PAS is a unique immersed sphere that contains all the visualised project information. This is equivalent to the immersed, experiential, and four dimensional version of the PrismArch Unified Database that exists in the server. Further intrinsic sphereing at the root level occurs as soon as the immersed human (IH) enters or joins the information space with an agreement as to the scale, location, and PrismArch time of that particular immersive experience. The presence of IH creates a persistent individual sphere (on SL1), and leads to the possibility of creation of other spheres automatically or manually by the IH.

Sphereing allows users to group multiple objects in the database and assign meaning to them. In the previous section (2.2 Tagging), we have established that the presence of design objects is

accompanied by the presence of an associated metadata level – a sphere instance. A sphere is an entity equivalent to a “folder” or a “layer”. As topological data divisions, spheres exist as envelopes for collections of data in three dimensional PrismArch coordinates and with the PrismArch time aspect. Visually, they are represented as nodes in the metagraph (see Section 3.2 Display Methodologies).

Applied to objects or groups of objects, sphereing recognises their authors, the authors’ credentials, and any contractually-defined data access agreements. Applied to immersed humans and their actions, individual and collective, sphereing creates the necessary conditions and agreements for their co-presence. Importantly, it is not necessary to duplicate the design objects (or immersed humans for that matter) when creating these interpretative transactional definitions; it is sufficient to reference them. In this way, an IH or design object can be in multiple SL2 or SL3 relationships that are all unique.

The back end, which is Speckle, contains information units that we want to sphere. We need to have safeguards in place that define access privileges and we need to ensure irremovable units (previously authored information of various file types or a single file type).

The Sphering Level (SL) of a given sphere indicates its visibility to all parties in the design process, and therefore has significant implications regarding data security, IP and discipline-related workflows (see figure 2.3a). To ensure these control systems are observed and to safeguard against the SL being changed or updated accidentally by users, the initial Sphering Level of an object is permanently ‘baked’ into its Prism_Arch Signature and packaged with the object when it is uploaded to the PrismArch Database.

Sphere Level	SL 1: Personal Work Sphere	SL 2: In-House Project Team	SL 3: Inter-Disciplinary Project Team	SL 4: Client	SL 5: Public
Data rights	Only the owner of the changes can see the changes	Changes are visible only to In-house team members	Other AEC disciplines can see the changes	Client can access the information	Everyone can access the information
Editing rights	Discipline specific	Discipline specific	Discipline specific	No editing	No editing

Figure 2.3a. Table 4 from D5.1 (reworded), outlining the 5 distinct Sphereing Levels of PrismArch

Any adjustments to the SL must be done by authorised persons in accordance with contractually agreed-upon workflows. The current Sphereing Level of any given sphere is also “baked” into the sphere’s DNA to safeguard against the levels being changed or updated by users accidentally. Because the Sphereing Levels have data security and discipline-related workflow and IP implications, upgrading or downgrading SL must be done by authorised persons in accordance with the agreed-upon contractual workflow.

Because ownership of information in the AEC industry is contractually defined, IP is owned through agreements that are unique to any given project. In most cases, the information initially belongs to the authoring party. Following the submission of the authored information to the client and receipt of payment, the ownership is transferred to the client/developer⁷. There may be subsequent licensing or transfers of information to the building operator; and lastly, the information may be licensed for the public domain (i.e. for virtual presence in the metaverse or other publicity, or for research purposes). As already shown in the tagging diagrams in chapter 2.2 [Tagging](#), Figures 2.2c-2.2i, the sphereing of data thus creates a **unique ownership record**.

Throughout all the stages, the data will need to be optimised for its purpose, shedding any semantic level information that is no longer relevant or legally necessary to retain (eg. streamlining the file size prior to release into the digital public domain, to make it readily available to the largest number of users). For this reason too it is important for the semantic information layer to remain distinct from the object-level DNA.

2.3.1 Meeting Sphere (experiential immersive environment for information exchange)

The meeting sphere is a unique subset of the general sphereing functionality. It contains a spatial design object – an environmental asset for virtual space, which gives users a platform for exchanging information based on the people that they encounter there. Sphereing here involves a multi-user or multi-disciplinary exchange; it forms a collective platform (or temporal stage) for users to share their data on any level above SL1.

The meeting sphere automatically contains the meeting environment and any project information that was pre-selected for the meeting. Here, a user can temporarily show a part of the data to their cross-disciplinary collaborators, possibly by toggling the SL relative to this portion of the project information.

If there is a need to subsequently share a copy of the data shown, it can be selected during the meeting (either by the user or by the collaborator) and copied (a transference of ownership conditions) into the collaborator's Query User Interface (QUI). A new DNA instance may then need to be imprinted on this data, encapsulating the original DNA and also the transference of information, including the terms of the transfer.

There is a need to share project information that we own under strict conditions/parameters which we call licensing of the data. This licensing is restricted by time, or by the specifics of the co-presence location (such as the duration of the meeting). The transference of data will need to be flagged by PrismArch, and waiver forms will need to be issued (for instance, to specify that the transferred data is not to be sold on, but only used for internal reference). As such, any licence should be revocable.

2.4 Sphereing: Process

As discussed in the preceding section, spheres are composites of information; used for creating meaning from data and managing IP. These spheres are not necessarily tied to a single type of PrismArch object, nor to a single discipline. The user-created spheres serve a dedicated purpose for

⁷ The exception to this rule is where the authoring party is a design-build organisation, holding legal responsibility for the construction as well as the design.

the creator and/or for the recipient of the sphere, each unique and having its own unique DNA. If a new sphere is created on the Personal Work Sphere level (SL1), it will still have the user as the author and the timestamp recorded in the DNA (and also in the tags automatically).

Data that is created by the PrismArch user on SL1 under the auspices of their employer or company (as the commercial environment) will eventually need to be shared, either temporarily or persistently, with either a. other members from the same company (SL2); b. other AEC professionals (SL3); c. the client (SL4); or d. with members of the public, such as community stakeholders or the press (SL5). This will either take place in a meeting sphere, where the data will be made visible for only the duration of the meeting, or through a new sphere on another sphereing level (SL2-5), thus making the information accessible to a wider group of collaborators.

When the architectural team creates a submission folder (new sphere) for collating information to share with the client, they do so on SL3 with read and write access for all disciplines collaborating on this project. Once the information is deemed complete, and it is upgraded to SL4 for sharing with the client, a new sphere is created on SL4 with references to the original objects. The data remains saved on the same server, with the only difference being that it is now locked for editing and is visible to the client.

When the architectural team shares a master model with the structural engineering team, it is shared via a sphere on SL3 that is shared with this team-II only; it does not include any other participating members. Any sphere that contains multiple-author content creates a situation where the ownership of the sphere is shared. The easiest way of dealing with this is to establish a read-only limitation on the shared information. However, we also understand that there may be instances where certain elements of the master model may need to be modified or manipulated by another party that did not author them.

For instance, if a member of the structural engineering team proceeds to use this data, a new sphere will be created on SL1 on their Metagraph. A warning sign will pop up, informing the individual that they are in danger of modifying or manipulating content that they haven't authored (IP breach risk). Acknowledging this pop-up would also be a legal step of acknowledging the IP rights of the authoring party.

When a temporary sphereing level increase is created, as in the instance of certain content being shown during a meeting between ZHA and AKT-II, the occurrence must be recorded together with the information about who was present in the meeting. Even when a collaborator leaves the meeting room without having kept a copy of the assets, there still needs to be an automated record stating that the asset was shown to the participants on the occasion of that meeting. Additional assets, such as meeting minutes, can be sphered to the existing meeting sphere later on, creating a record of the shared minutes (when they were shared, with whom, etc.) with an association to the existing meeting data.

2.5 (Data) Security [ongoing] and (User) Authentication [at onboarding]

Whatever is done inside the PrismArch environment has a legal basis in the real world and direct consequences for the project's development based on agreed contractual obligations. Because PrismArch is a professional environment, it is essential for every human individual in the PrismArch system to have a secure and authentic presence inside the virtual space. An extensive list of

providers offering identity authentication services can be found in the Appendix under [Section 2.5 Resources](#).

As previously specified in D1.1 Section 4.2 (B - Constraints), and elaborated on in D1.2 (2.6.1 Security and Authentication), the requirement is for the PrismArch platform know which organisation each user belongs to, as well as what Sphering Levels they are currently operating within for any given project, in order to ensure that they can only share (or have shared with them) data appropriate to their Role, Organisation, Sphering Level, and so on. As stated before, the development must look beyond the 'default' implementation of Speckle to facilitate this.

We have already addressed the importance of the Sphering levels in the previous section. Here, we will further elaborate the meaning of the Sphering adjacencies (as shown in the diagram in Figure 2.5a) that relate to the way that temporal agreements and resulting additional functions can be managed. By looking at sphere adjacencies, we can determine which functions and rules apply to each relationship, keeping in mind that in a commercial world, these adjacencies are regulated by contacts.

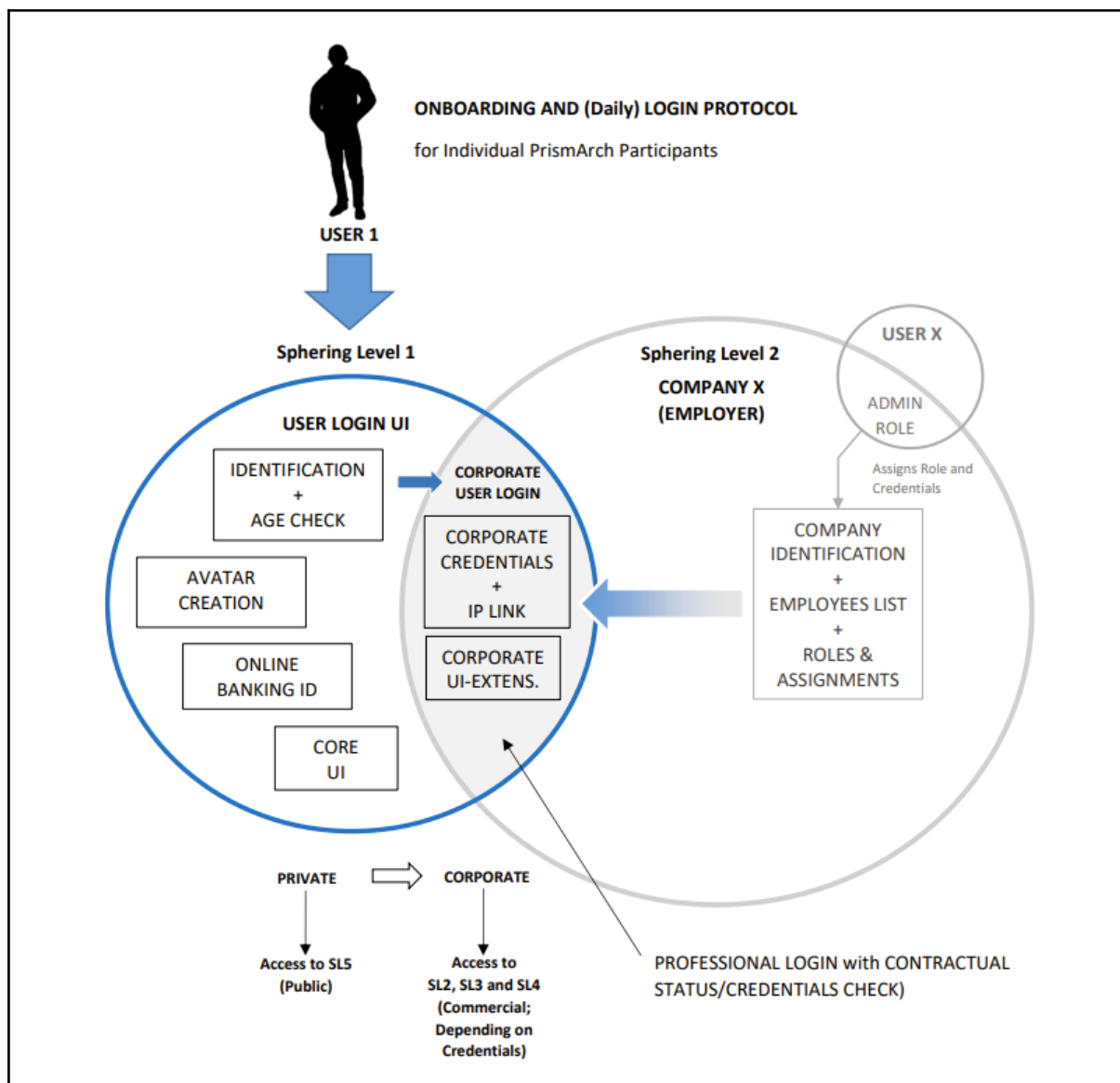


Figure 2.5a. Diagram by ZHVR showing the intersection (adjacency) of the personal SL1 and corporate SL2.

Figure 2.5a shows the principles underlying the individual onboarding routine with the required ID and Age checks. It also depicts the individual user's interface with the professional / employee-employer level relationship. In this case, the company (containing multiple individuals and roles) is a Sphering Level 2 entity. By mutual agreement, the individual user accepts (an) additional interface(s) with the company's credentials, and receives specific work-related UI and IP-creation rules.

Importantly, the user's private SL1 Level primary identity and functionality must be autonomous and protected to all corporate add-ins and additional functions/rules. It can only be exposed to those the individual has agreement with; where it is held protected within a contractual framework.

When entering the SL1 corporate login, the user may have to comply with the agreed corporation's rules such as the workspace or dress code; these may be professional add-ons to their online presence to aid the distinction of the individual's discipline within the work context.

The active corporate SL1 user will have access to all of the company's dedicated software plug-ins, information and services. Any production and information related to the SL2 domain owned by the domain as per contractual agreement shall be labelled as such and stored in a non-private repository to preserve IP rights and access for this information by the SL2 corporate domain.

In line with general data protection no information exchange or collection can be allowed between the employer domain and the private domain. Any AI data collection and assistance will need to be compartmentalised and only connect on open and agreed terms between the two domains.

The application of the mentioned protocols and architecture equally applies to the higher sphering levels in equal form. Differentiated by the type of entities and information contained in the relevant

sphering levels, the principles apply in self-similar form.

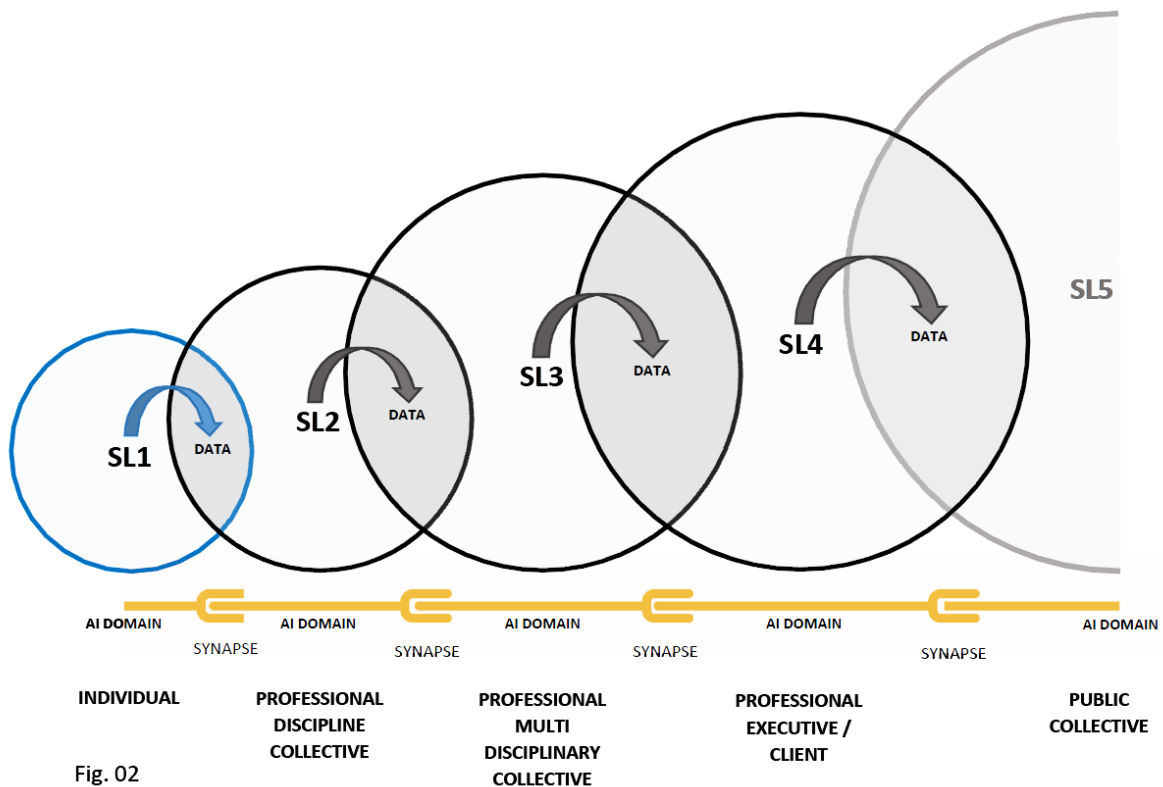


Fig. 02

Figure 2.5b. Diagram by ZHVR showing the connectivity between Sphering Levels, from personal to public.

All Sphering (each sphere) has meaning and purpose. By extension, whenever there is connectivity with other spheres, each relationship, each overlap will also have a unique purpose.

Figure 2.5b focuses on the shared, contractually-regulated data transmission and approval space. It makes visible the demarcation and compartmentalization of the data production and its transmission between entities along the different Sphering Levels. The highlighted synaptic couplings within the data stream relate to the agreed contractual relationships and the containment of sensitive personal/intellectual property related information. Important to note is that each Sphering Entity by default should have autonomy in order to connect to multiple project spheres and parties. What is shown here in singular form, in reality would mean that Company X (SL2) would have many projects to connect into as well as multiple individuals making up its workforce (SL1). For additional information and resources on GDPR, please refer to [Section 2.5 Resources](#) in the Appendix.

Anecdotal example for further reading:

If an immersed human 'Nellie Thomas' is overlapping with an organisation called 'Zaha Hadid Architects', Nellie will have access to the software licenced to Zaha Hadid Architects and have the authority to enact certain relationships, such as authoring data. There will also be other associated benefits, such as receiving a salary from ZHA. Relational values generally represent temporal states. When Nellie, as an employee of ZHA, makes information spheres that she gets paid for, she does so in her SL1 with a relation to ZHA. Essentially, she does her sketches in her personal 'Nellie'

sphere while connecting with the general ZHA project information, stored over project spheres. Nellie's payroll number or employee ID becomes part of the boolean.

It is important to remember that Nellie is a human being, with a physical existence. Her avatar is more than her representation - it is also her identity inside the metaverse (networked, immersive digital collective spaces). Therefore, while the clothing can be dictated by the company, they cannot change Nellie's avatar. No one except the immersed user can decide what to do with their avatar, which is their personal vessel for immersive presence. We are advocating for user privacy and the end of commodification of human identity - mirroring the efforts relating to Web3.0. User information and certain aspects of user actions cannot be disenfranchised and taken from the Personal Sphere; users must be protected from any misuse of the personal information and user data.

There is a list of existential functionalities necessary in order for Nellie to have a virtual presence (SL1) inside PrismArch, involving avatar creation, core UI, identification and age check, and a link into the financial system⁸. The set of functionalities is replicated on the other sphereing levels. The company Zaha Hadid Architects Limited must certainly have a corporate account, online banking capability, and a managed list of employees with their projects and permissions. Zaha Hadid Architects will perhaps also have an extended corporate UI and a custom meeting room design, a predefined employee dress code, and other presets. Similarly, the client must also have online banking capability on SL4, in order to purchase the authored IP.

Each sphere is defined by the existential information contained therein. Moving from 1) personal spheres, 2)in-discipline spheres (collective), 3)interdisciplinary spheres (collective), 4) pre-public (client owned) spheres (collective), and 5) public spheres - each of these sphere levels have objective requirements that make up their self-definition and allow the spheres to enter temporal relationships with other spheres (as shown in figure 2.5a).

If Spheres are the 'presence' or 'entity' values, the sphere adjacencies are relational values. These temporal relationships may evolve, as in the case of Nellie's promotion at ZHA, or by the client's hiring of a different local architect for the collaboration. While a single sphere can be defined, perhaps even at the start of the project, the relational information is accumulated over time through the interactions that take place. Therefore, we anticipate the need for intelligent, AI-assisted management of the growing libraries of this synoptical (relational) content. While this type of AI involvement falls outside the scope of the MVP, it will need to be addressed in the case of future PrismArch development beyond the 2-year research period.

The synoptical operations and functions (relational values) must be understood and developed alongside the discreet Sphereing Levels.

It should be clear from the above discussion that there is a critical requirement for the PrismArch to ensure the security and manage the privacy of all data shared between parties on the platform.

⁸ Link to personal banking falls outside the scope of the MVP, but is mentioned in order to provide a comprehensive outline of the PrismArch concept.

These issues were previously demonstrated in D1.1 Section 4.2 (Data Authority / Ownership of Data), D3.1 Section 3.2 (Data and Document Management); and recapped in D1.2 in Section 2.6 Non-Ontological Requirements for PrimArch, and 2.6.1 Security and Authentication (see text box below).

As described in D4.1 Section 3.2.1 (Speckle system and developments towards integration), and below, the existing Speckle framework contains most of the requirements needed for PrismArch. It is a highly secure data distribution platform, the developer state that:

“Enterprise Speckle Servers (as well as our hosted offering, speckle.xyz) are deployed with end-to-end security:

- The Speckle Server uses https (TLS) to encrypt all incoming data from all clients.
- All data is stored in a managed PostgreSQL database cluster.
- The DB is only accessible from the Kubernetes cluster that runs your server and its other components.
- DB credentials are securely stored in a Kubernetes secret.
- SSL is always used to communicate with the DB.
- Data in the DB is encrypted at rest with LUKS.
- The DB will have a standby failover node, & PITR (point in time recovery).”

From <https://speckle.systems/security/>

The same online document also confirms that Speckle complies with the EU’s GDPR policy.

See <https://speckle.systems/privacy/> for additional information.

The security arrangements above are for Speckle’s Enterprise solution - however it confirms that their platform can achieve the level of data security necessary for PrismArch, and the above should be treated as our standard requirement for PrismArch.

Data Visibility and Confidentiality

However, while Speckle generates highly secure data connections, it does not provide the functionality necessary to isolate incoming data from different parties, or provide granular permission levels that control which users can see or operate upon different data sets. Speckle currently only provides differentiation between Reviewer, Contributor and Owner.

2.5.1. PrismArch Licensing

This section presents a short discussion of licensing options to help clarify the Sphering Levels from the standpoint of PrismArch commercialisation.

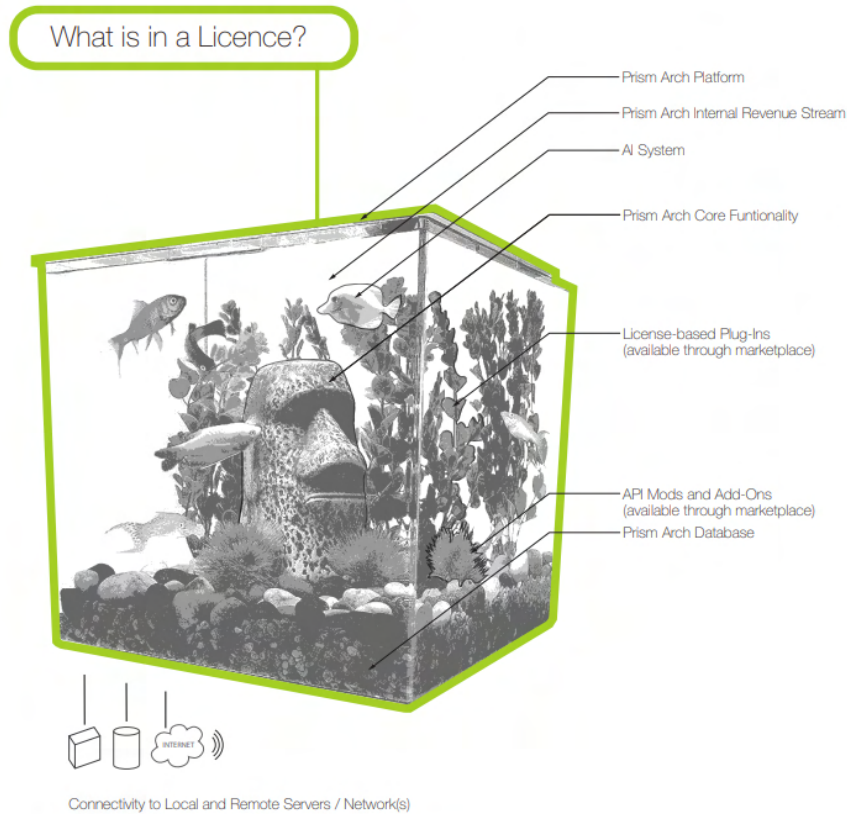


Figure 2.5.1a. PrismArch commercial licence ecosystem

The individual sphere licence SL1 should be free for non-commercial use and SL5 access to public metaverses. Subsequently, any licence related to production with SL2-4 capability will be a paid suite. These commercial licences could potentially be scaled and paid according to the number of SL1 employees. Ideally the connectivity of PrismArch would expand to more disciplines, even those disciplines that do not generate 3D models, such as Quantity Surveyors (QS) or Project Managers (PM). Although these disciplines do not generate geometry, they will be able to author metadata into the models created by the other disciplines, expanding the project knowledge base.

We will first establish the parameters of the most essential licence type - the Private (Individual / Light) PrismArch licence (Figure 2.5.1a). Offered for free, it offers private individuals core SL1 capabilities and read-only access to the public SL5 domain. Trial access to extended capabilities could be offered as part of the marketing strategy, allowing the user to author content into SL5 using Mindesk or other software through API add-ons from the PrismArch marketplace.

The key to successful public engagement is interoperability. If someone has access to SL5, they should be able to use the same metaverse standardisation so they don't have to change their avatar, allowing them to bring themselves into different worlds fluidly and ensuring their data does not need to be re-entered and their identity re-created from scratch.

Allowing private individuals access to the PrismArch platform creates an immersive public domain, providing the feedback necessary to advance the architectural practice. This also creates the possibility of generating publicity for the individual projects by hosting them on the Metaverse. The functionality of SL5 is outside the scope of the MVP, but an important component of any future developments.

We feel it is in the commercial interests of future clients to start hosting networked immersive virtual clones of their projects even prior to completion. It is therefore our expressed goal for PrismArch SL5 to be only a button press away. While the cloud hosting and delivery would necessarily be an additional service and an additional financial commitment, it should be straightforward to bring forth the actual content coming from the lower Sphering Levels in PrismArch. Finally, although SL5 is a public domain, the client may wish to restrict access in part or as a whole.



Figure 2.5.1b. SL1 immersed presence allows access to SL5 public domain

The next type of licence is focused on the AEC industry, providing company (full) paid licences to corporate clients for managing access to SL2-SL4 (see Figure 2.5.1c). This licence continues to provide the SL1 core capabilities, offers read/write SL5 access, and enables expanded company-based presets and UI for SL2, SL3, and SL4 environments. It streamlines and simplifies the onboarding of staff and distribution of team roles. It also includes unlimited access to the PrismArch productivity software suite, including Mindesk.

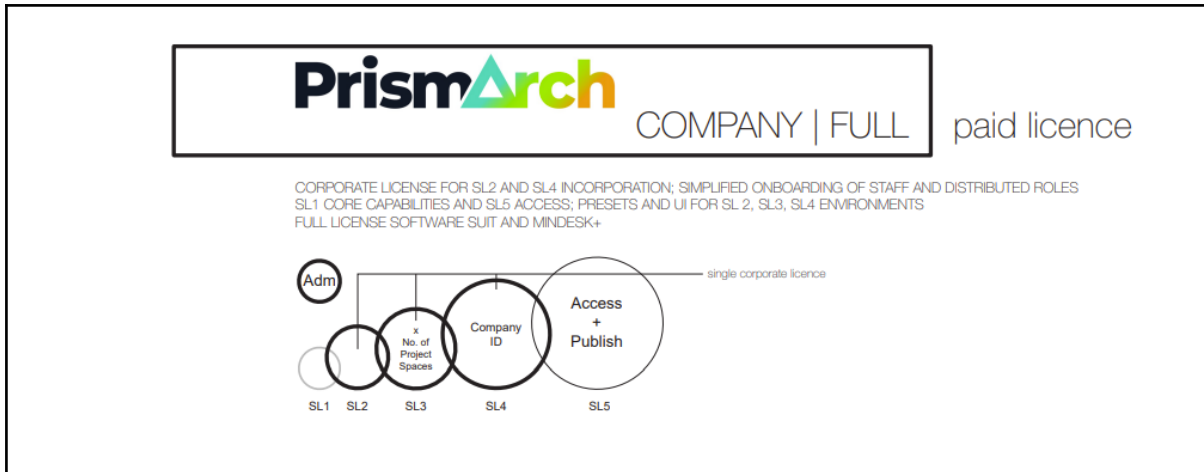


Figure 2.5.1c. Commercial / Corporate licence

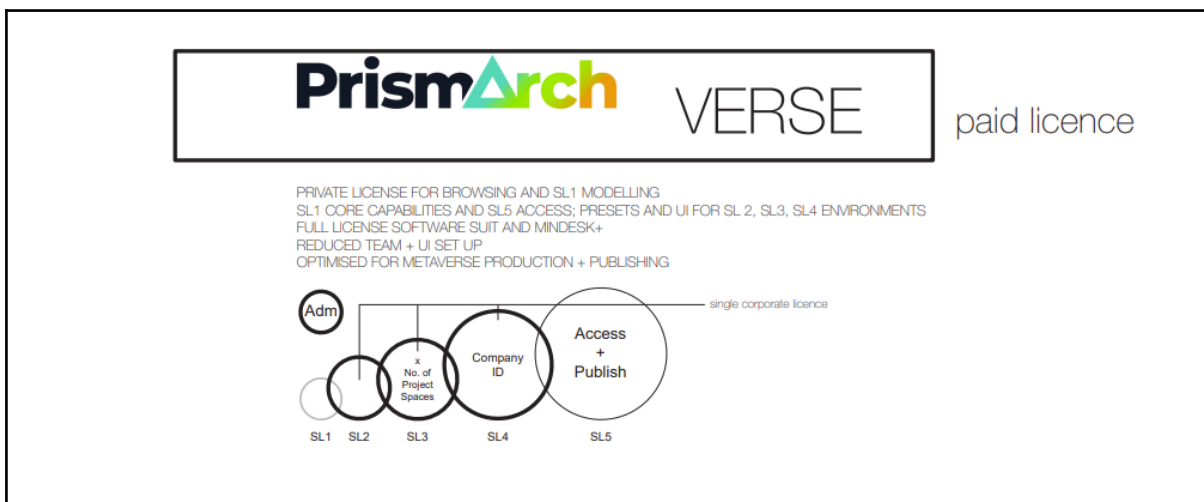


Figure 2.5.1d. Verse licence, allowing users to simulate and build for the virtual domain

The final licence type is intended for creating simulations for the virtual domain. Unlike physical architecture, the virtual domain will not have the same constraints. Therefore this would be a lighter version of the commercial licence. It may have other forms of optimization for size and culling of data for speed.

3. Information Display and Query System (IDQS)

PrismArch aims to deliver a VR solution for capturing and manipulating the entirety of the multidisciplinary project data space. Crucially, in order for this project data to have meaning, it must be identifiable and interactable inside the immersive VR space. In this section, we provide a high-level summary of the proposed data structure and the UX requirements for collapsing multidisciplinary project information into unified immersive data space in a way that generates knowledge for the users and stakeholders.

To address the complexity of the embodied access to multidimensional information space, we set out to describe and provide mock-ups of the Information Display and Query System (IDQS) necessary to generate useful knowledge from the stored project information.

The notion of Tagging and Querying information was introduced in D1.1, Section 4.2 Taxonomies, Rules, and Constraints, under “Remarks”⁹:

Data cannot exist inside PrismArch World Engine unless the asset has the DNA and tag, thus every asset that exists will have a unique label/tag that works as an identifier. Bringing data into the PrismArch World Space (AEC Design and Collaboration Space) requires an identification system that allows the system to keep track of the component, know its file origin, author, timestamp, version, and any other metadata relevant to the functioning of the PrismArch World Space as outlined in this document. Using the filtering tool proposed for the PrismArch Platform, disciplines should be able to access, search, and interact with the data.

In D6.1, we set out the four root classes of the information system: 1. immersed human beings; 2. design objects (also referred to as assets); 3. UI objects; and 4. metadata. We further defined Sphering to ensure that all information contained is identifiable with the relevant project-appropriate level of access at any given time in the project’s development. Crucially, the sphering level for any given portion of project information can and will be adjusted as often as necessary throughout the development of the project; any time a change is made, the level change information also becomes part of the searchable metadata, telling the story of the project’s development.

The IDQS sits at the root of PrismArch. It consists of the totality of the information space contained inside PrismArch, taken together with the functionality to view, query, and interact with various subsets of this information in a practical way. It comprises: 1. a pool of unilaterally searchable project information, and 2. a responsive *metagraph*, a functional UI element, with metadata nodes representing the information.

The holistic project information space is discussed in more detail in the next section, 3.1 [Information](#). The organisation and visualisation of the information is described in 3.2 [Display Methodologies](#) and 3.3 [Query System](#). Further visual aids for demarcating information are presented in 3.4 [Information Demarcation](#). Section 3.5 IDQS [Operation and Navigation](#) brings these concepts together in a mock-up, describing how (a) user(s) interact(s) with the IDQS system.

3.1 Information

Information inside PrismArch refers to the entirety of the structured multidisciplinary project data – including workflow, production, IP ownership, and user data – that is presented in a way that is meaningful to the user. The PrismArch platform concept is unique in its (so far theoretical) aim to grasp and contain the entirety of the project data and to make it accessible, viewable, and interactable as information in VR. While it is necessary to limit the scope of the prototype to fit the development into the available timeline, it is also necessary for us to communicate the considerations that must be addressed for this platform to be functionally usable outside of the testing environment.

The real-world commercial and legal context must inform the visibility and interactability of the information contained inside the database: the purpose of the information in some cases requires

⁹ This was touched upon in D4.2, Appendix II by CERTH. Examples of Tag and Query content were provided (described as ‘SPARQL queries executable by the semantic retrieval service to support various scenarios’), including the commit history, author history, room information, and object history.

that it be kept confidential, and in other cases needs to be shared with a small subset of the users. This creates a requirement for a robust data management system that ensures the required information is shared only with the necessary spheres and not anywhere else.

The unique DNA signature assigned to all PrismArch objects forms the basis of the flexible data class taxonomy. As previously discussed, the DNA contains all relevant data received from the source file¹⁰. This DNA **must** be searchable in the same way as are tags, comments, contact cards, and sphereing instances (such as grouped objects, meetings, etc). The associated tags, comments, and sphered instances are also crucial sources of project-related information that must be accessible and searchable via the IDQS.

A central requirement is that inputs from all disciplines must be brought into a unified language and given the same level of description. We are relying on flexible ontologies and semantic data integration for easily retrievable data, accessed via user-directed query. Conceptually, we are moving away from relying on software-specific ontologies / ways of classifying information (such as Rhino layers, BIM COBie or IFC classes from Revit, or any other software). We propose to create a flexible data class taxonomy on top of the file-specific structures in PrismArch so that no matter where objects come from, they are accessible and searchable via the IDQS. This is done without losing the connectivity to the original software or the original file structure ontologies; we are embedding this initial information and keeping it retrievable and searchable. The superimposed semantic data class taxonomy is not a predefined exhaustive set of key values, but a network of meaning that develops over time, based on user actions and interactions (transactions) and the progression of project file data stored inside PrismArch.

Every object must be readable (retrievable) based on the credentials of every user. Beyond the object DNA, the retrievable description is something that implies meaning and purpose, implicitly linking the object, the sender, and the receiver. These transactional aspects, as already discussed in Section 2.2 [Tagging](#), will be registered on the semantic level via the sphereing and tagging interface. We must enable the possibility of universal descriptions, because in principle they could apply to every object. However, there is an inherent economy behind the semantic descriptions, because they are going to only be applicable and added where the workflow demands them: how data is interpreted and imbued with meaning is ultimately up to the users.

A domain-specific ontology has been designed by CERTH for the scope of PrismArch that integrates information from various VR and architectural design tools. The ontology is expressed in Web Ontology Language (OWL) format, allowing the definition of a unified structure that contains complex relations between objects. The structure is based on existing ontologies and standards, namely ifcOWL and EXPRESS. To ensure data integration, information is semantically represented in RDF (Resource Description Framework) triples format and stored in the Knowledge Base (KB). The ontological schema focuses on two main aspects: the structural characteristics of objects (i.e. Walls, Rooms, etc) and the commit-related information from SpeckleDB (e.g. commit id, stream id, author, etc). The reason behind this decision is offering the appropriate knowledge structures to provide, in the end, the opportunity to access and search for various information across different commits. The supported scenarios include, but are not limited to, detecting objects that have specific tags, the

¹⁰ There is some ambiguity about what, technically speaking, needs to be embedded on the object level, in the DNA, to enable the objects to be re-exported out into their native softwares. Conceptually, we believe the information stored in the DNA on the object level should be minimal; with the semantic layer carrying the majority of the project-related definitions.

meshes of a layer, object and mesh history etc. To support each of the scenarios, the appropriate semantic queries are executed on top of the Knowledge Base.

For a detailed discussion of the requirements for the PrismArch ontology, including a summary of requirements outlined in previous deliverable, please refer to D1.2, Section 2.3. As stated in the previous deliverable, architectural naming conventions require flexibility. A closed, bottom-up topology for data fails to meet the open-ended and innovative way that architectural designers work as they define and redefine their spaces in ever-greater resolution across the project design stages. What we propose, accordingly, is an open-ended information ontology, akin to a semantic web, that grows and matures together with the project.

3.1.1 User Comments

Comments and redmarking suggestions are part of the transactional communications used by users to structure and navigate their collaborative workflow. Project-related workflow comments must necessarily sit on the semantic layer. On the object level, we could have several meshes that, when taken together, represent a “chair”. The comment “chair to be updated”, which is transactional by nature, is yet another semantic level of representation for the communication of an idea. This means that if a comment (as part of a design evaluation procedure) relates to 200 objects, it will not have to be replicated over those 200 object instances on the Speckle object level.

Therefore, comments have to be lifted out of the object-level classification, and applied to groups of the higher sphere / semantic level. A “Comment” tag will be given to the sphere instance. Whereas the comment object, such as the 3D data for the written comment “move the extinguisher two metres to the east”, will be stored as a design object inside the sphere instance (among all other related comments made in a commenting session).

There must also be notifications about the comments that are created, edited, replied to, or resolved, and these communications about comments also sit on the semantic level of functionality. Once a sphere is created and tagged with the value “comment”, there may be a pop-up that says, “do you want to share this with your colleague(s)?” This operational function sits within the UI.

The comment sphere will need to contain the reference to the contained design objects, the users involved, the visibility or sphereing level of the comment, the relationship between the comment (as a design object) and the object(s) being commented on, and any communications or notifications related to the comment itself. Related to this, a comment object might have a “WIP” button and some workflow-related procedures, such as adding the colleagues’ e-mails and notifying them.

When an object is brought into PrismArch (e.g. a phone), and given a name or set of names (“phone”, “mobile”, “iPhone” etc.), it becomes associated with an object’s unique DNA reference in the Speckle database. We also have situations where the location of the “Living Room - Ground Floor” only exists as a concept or idea - where the geometry or object data has not yet been created. In this case, a user may choose to create a stand-in geometry reference, such as a small 3D sphere or a surface plane, and associate the comment(s) with it.

3.2 Information Display Methodologies

Considering the visual nature of the immersive 3D experience, we believe it is paramount to ensure that all obtained project data is visible and interactable inside the PrismArch system.

The immersive 3D environment provides a sense of depth that opens a wider capacity to display information than by any other means, including 2D or non-immersive 3D visualisations. As part of our process, we investigated existing formal methods for representing information. We selected those that could apply to the totality of undifferentiated data experienced inside PrismArch's VR environment for further development, focusing especially on visualisations of data organised according to multiple parameters.

Multi-Parameter Query

One such example is the interface for data optimisation according to multiple fitness variables in the Octopus plugin for Grasshopper 3D (Figure 3.2a), where solutions are distributed in 3D space (interactable in desktop mode). It is unclear, however, if this visualisation style can sensibly accommodate more than three variables.

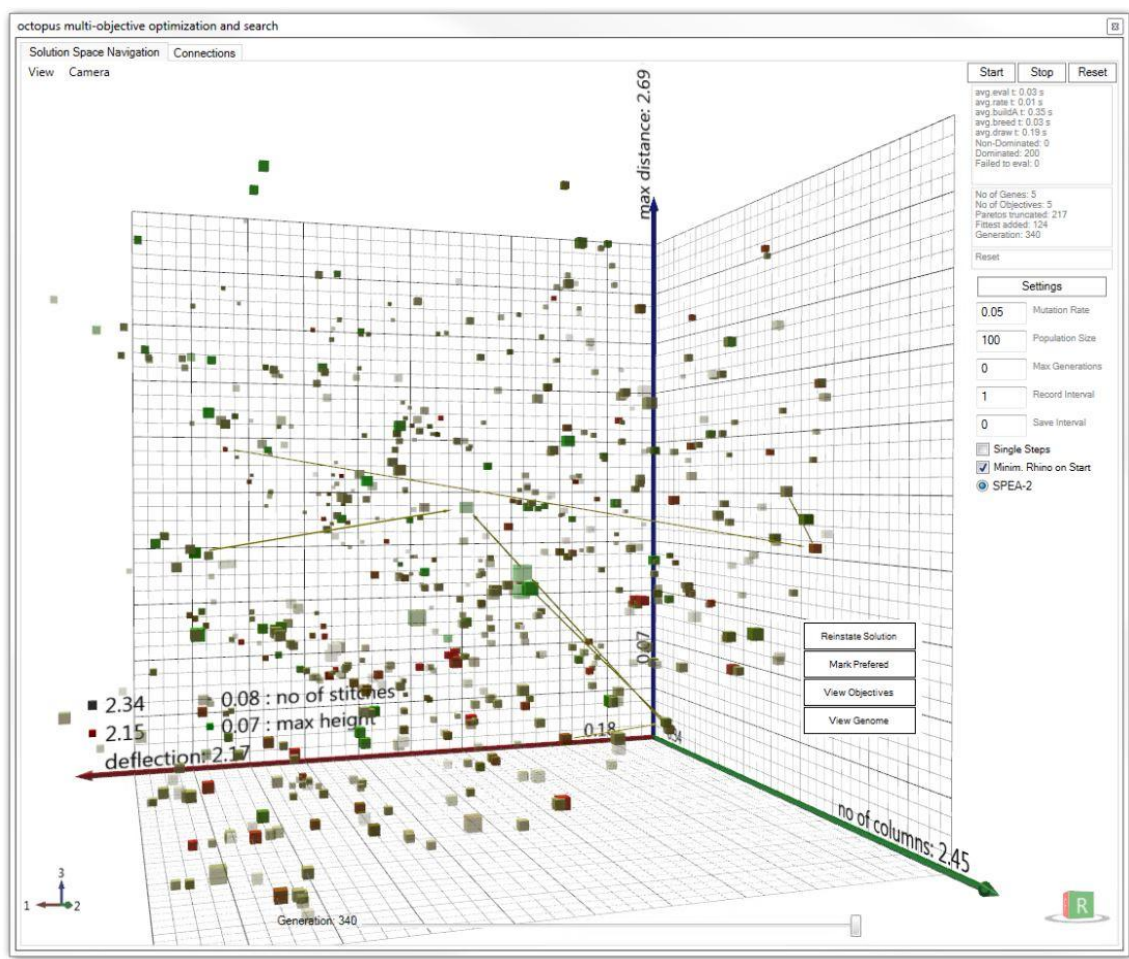


Figure 3.2a. Octopus multi-objective optimization and search, developed at University of Applied Arts Vienna in collaboration with Bollinger+Grohmann Engineers, <https://www.grasshopper3d.com/group/octopus?overrideMobileRedirect=1>

Force-Directed Graph

Softspace AR developed Prototype01 as an interactable spatializer for Dropbox content based on the force-directed graph to test 3D force-directed graphs for immersive reality and 3D folder structure explorers (Figure 3.2b). To read more about the prototype, access the web page [here](#).



Figure 3.2b. Softspace Prototype01, force-directed graph AR interface actuated by controller-less hand-tracked UI https://substack.soft.space/p/softspace-ar-prototype01?s=r&utm_campaign=post&utm_medium=web

In immersive 3D environments, the force-directed graph evenly distributes nodes and organises them in a way that minimises overlaps. Research apps Roam and Obsidian have both introduced this visualisation style to their list. Aside from the pleasing aesthetic, the graph generates insights about how well the system of knowledge is integrated, helps easily spot missing links, and can be used to navigate through the dataspace¹¹.

¹¹ Eleanor Konik, *It's Not Just a Pretty Gimmick: In Defense of Obsidian's Graph View*, 2021, <https://www.obsidianroundup.org/its-not-just-a-pretty-gimmick-in-defense-of-obsidians-graph-view/>



Figure 3.2c. Obsidian graph view of data by Eleanor Konik, *It's Not Just a Pretty Gimmick: In Defense of Obsidian's Graph View*, 2021

<https://www.obsidianroundup.org/its-not-just-a-pretty-gimmick-in-defense-of-obsidians-graph-view/>

An open-source VR version of the 3D force-directed graph created by UX designer Vasco Asturiano (found [here](#)) was shared with the consortium last year, and forms the basis of the Metagraph development in PrismArch.

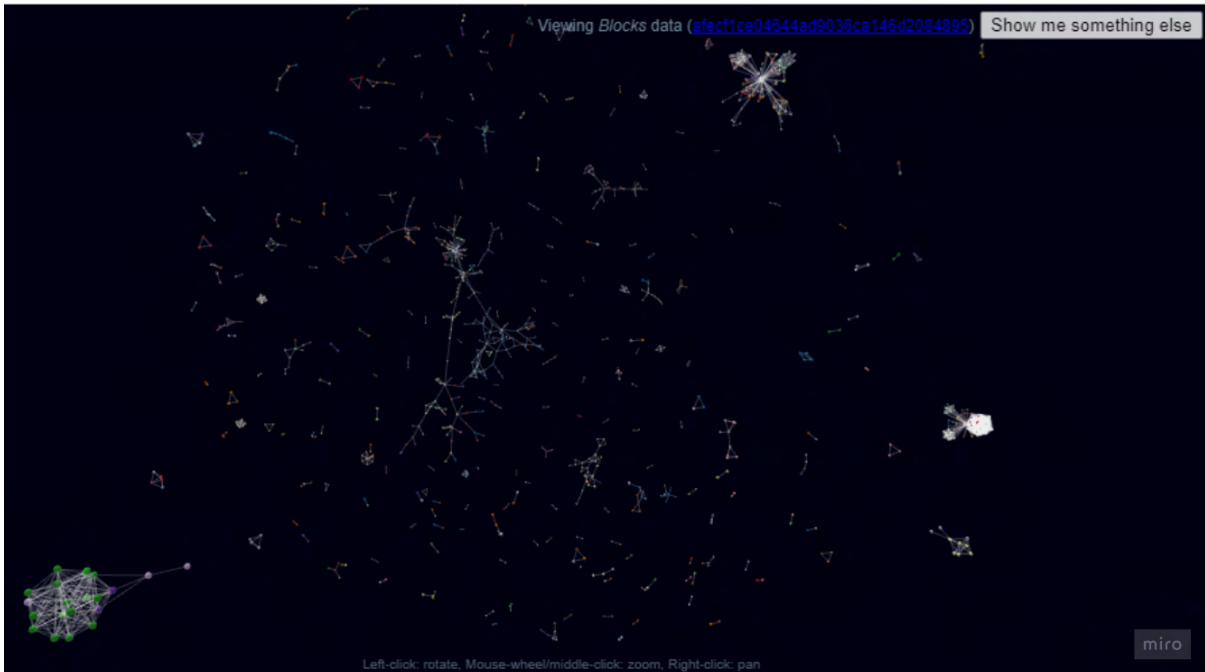


Figure 3.2d. 3D Force-Directed Graph (ThreeJS) created by Vasco Asturiano, 2021

<https://bl.ocks.org/vasturiano/02affe306ce445e423f992faeea13521>

3.2.1 Metagraph Concept

We conceived of the metagraph as an immersive visualisation method for the information contained inside PrismArch. In the metagraph, the display responds to the search query in a way that ensures a user is only presented with the relevant information, rather than being overwhelmed by the totality of the available project information.

Before a search is instantiated, all available information can be viewed as an undifferentiated foam of nodes filling the space of the metagraph according to default parameters. Aside from the default preset, there is no logic for viewing this project foam. The logic for the structure and display of the nodes is introduced by the user's query. Because not all nodes necessarily have connections to other nodes (in other words, not all nodes are related semantically), the metadata nodes need to be able to sit side by side. The metadata foam is thus the aggregate of the undifferentiated data seen in totality prior to the enactment of the search query. The first step of the user interaction with the metagraph involves entering (typing or speaking) in a keyword, providing a means for the foam to self-organise. This first externally-introduced parameter is equivalent to instantiating a search.

We see a conceptual need to present information in different ways based on the user-directed search query, and we are taking this as the foundation for the UI mockup development. In a project space where the magnitude of available information is n , with an unlimited number of semantic links inside the project-related design space and potentially an unlimited number of external references, it is important to provide a flexible and direct means of accessing any of the required data. The interface should allow the user to configure the layout and organisation of the metagraph depending on the nature of the query. It should also allow the user to access all information - from the very top-level groupings down to the granular detail - underlying the chosen organisational method.

What we present in our mock-ups is a concentric radially growing information display system. However, we could also imagine other ways of presenting or organising the data: linear or inward-growing, for instance. Another way of displaying the information could be with leaders. There would be a special mode of representation, collapsing the metagraph so that the nodes disappear and the tags are linked to the relevant design objects and act as leaders for the design content.

Taking the radial force-directed graph as the baseline, we have laid out several organisational templates for viewing and interacting with data. It should be possible to display the results in various ways, perhaps enabled by toggling between several pre-sets such as: 1. authoring party, 2. time, 3. size, 4. file type. Figure 3.2.1a is a two-dimensional study of ways that the metagraph would reorganise relative to the user's query. Relative to alphabetically-organised search results, we were able to see that alphabetised results do not organically fit in with radial organisation; rather, a 2D scrollable search result "box" would be most helpful for browsing data indexed alphabetically.

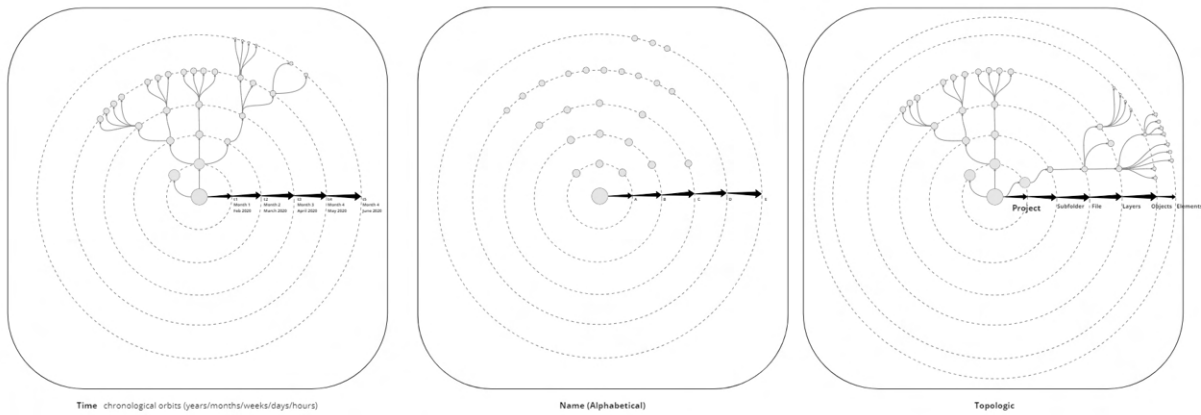


Figure 3.2.1a. Information reorganised relative to the search query

The spherical orbits of the metagraph along which nodes are organised, shown as concentric dashed circles in Figure 3.2.1a, are a flexible scaffolding for the organisation of the metadata nodes. For example, when searching along the time parameter, these would represent days, months, or years - depending on the timescale of the query. Alternatively, the spheres could represent significant dates along the project's development.

The metagraph interface needs to be both flexible and responsive. Examples of metagraph data responding to the user search may involve the cost estimation data presented along a cost estimate timeline. An alternative user search could require all available site photos to be displayed on-location. There may also be a need to collapse the dimensionality of the graph, so it can be interacted with in 2D, similar to the time-based 2D layouts shown in figure 3.2.1b. This shows only some of the numerous ways that the IDQS would ideally be able to respond to the inputs and inquiries of the immersed users, serving as the single port of call for all project information.

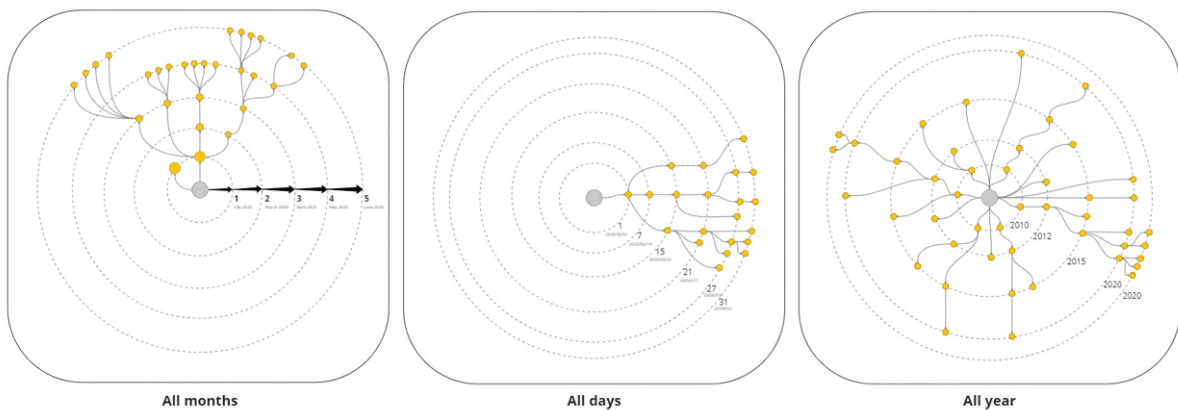


Figure 3.2.1b. PrismArch 2D radial metagraph organisational diagram, proposing time-based templates for search results, organised along chronological orbits representing different time intervals (years, months/weeks/days/hours).

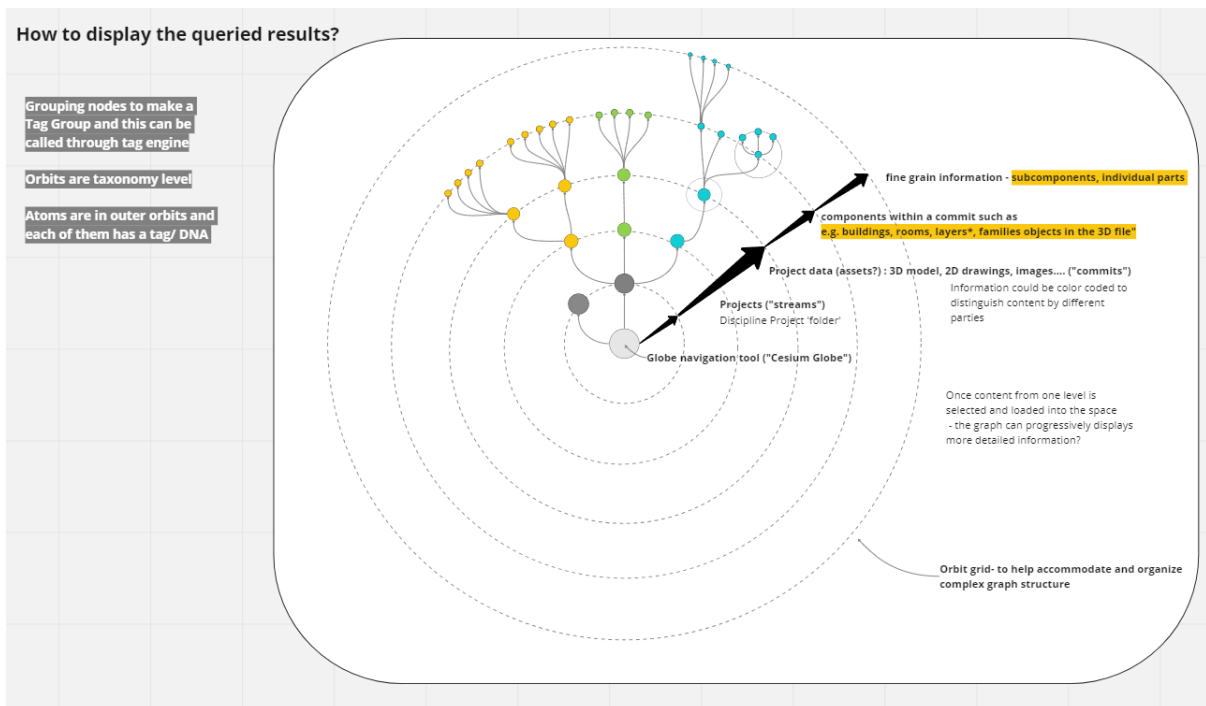


Figure 3.2.1c. PrismArch 2D metagraph structure diagram, organised according to present-day Speckle integration

To align our research with present-day developments, we diagrammed how the data is currently structured in the MVP prototype (see Figure 3.2.1c). Here, the spherical orbits reflect the increasing granularity of data, progressing from entire projects in PrismArch to object subcomponents. Starting at the central node of the earth globe, where all projects are shown, we next progress to the orbit of 1. all PrismArch project nodes, then 2. nodes representing project streams in Speckle, next 3. nodes representing commits containing project data (3D models, 2D drawings, images, text, etc.), the 4. components of the committed data (such as buildings, rooms, layers, beams, ducts, and other families of objects taken from the 3D or 2D file), and finally, the 5. subcomponents or individual parts of the families - individual objects or meshes.

This type of organisation is reminiscent of the familiar nested folder structure. However, unlike the Microsoft folder structure combined with proprietary software file structures (where access to the granular data is only achieved by opening each of the the relevant specialist softwares), PrismArch users can access information at all different levels (from: "folder" - "subfolder" - "file" - "layer" - "element") via the uniform interface of the IDQS.

The 3D force-directed graph also presents an opportunity to examine the overall project data. We may need to see the proportional relative values of data generated by the various disciplines, or we may wish to discover the node locations of any unresolved comments and mark-ups. We can see the benefit of facilitating a means to explore this dataspace at a larger scale, like a skydome where the outer node results are projected onto an invisible sphere, similar to the screenshots of the immersive exploration mode in VRNetzer (see Figure 3.2.1d)¹².

¹² Pirch, Sebastian & Mueller, Felix & Iofinova, Eugenia & Pazmandi, Julia & Hütter, Christiane & Chietini, Martin & Sin, Celine & Boztug, Kaan & Podkosova, Iana & Kaufmann, Hannes & Menche, Jörg. (2021). The VRNetzer platform enables interactive network analysis in Virtual Reality. Nature Communications. 12. 10.1038/s41467-021-22570-w.

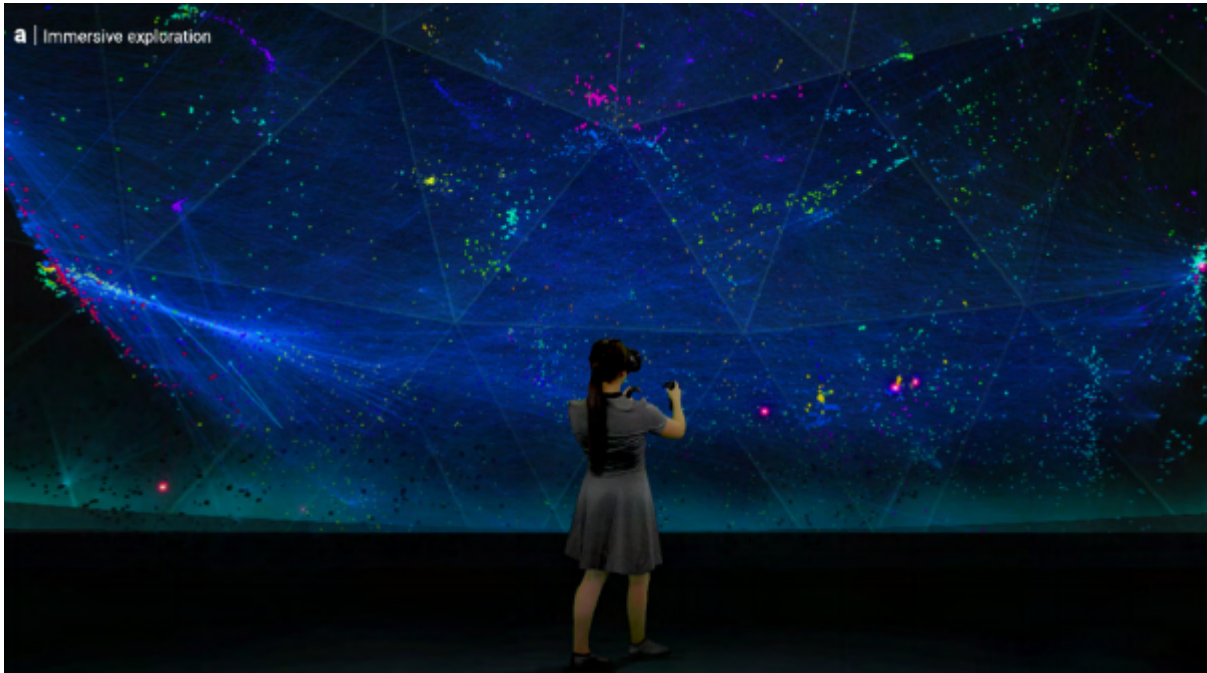


Figure 3.2.1d. Immersive exploration of forge-directed graph in VRNetzer platform

https://www.researchgate.net/publication/351080869_The_VRNetzer_platform_enables_interactive_network_analysis_in_Virtual_Reality. Accessed 25 July, 2022.

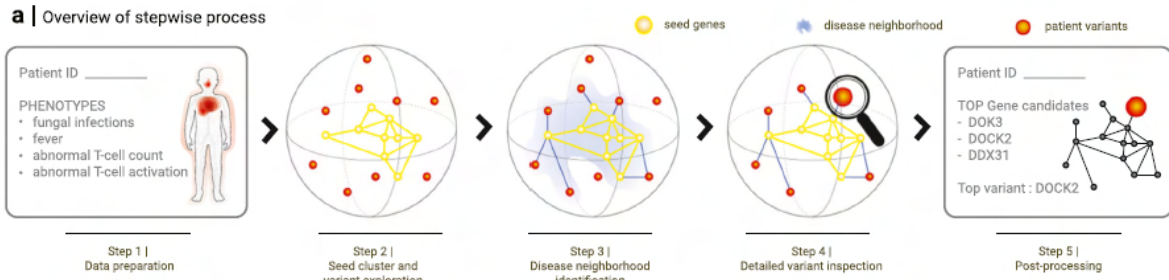


Figure 3.2.1e. VRNetzer: overview of the different steps of the data analysis procedure, from collecting relevant data in the web module to performing interactive network analyses in VR and finally post-processing the results.

In summary, the metagraph is the ultimate access gateway to metadata inside PrismArch and for retrieving and connecting out to design objects.

3.3 Query System

The information display and query system is an information-management system. In the context of PrismArch, the evolving holistic project information space and the resulting possibility of responsive information display is predicated on the existence of a flexible tagging system on the semantic level (user-actuated and automatic) that allows users to make sense of the project information in a dynamic way. However, this form of knowledge management will only be useful / serve a purpose if users also have the ability to query the project data through multiple-parameter or boolean searches (e.g. 'MEP' + NOT 'ductwork' + 'added by user 1' + 'NOT' 'first floor').

The query system interaction therefore involves the following operational actions and sub-actions:

1. Creating new nodes (new entries into the database)

- importing assets
- selecting assets
- sphering assets: naming them and creating the metadata node. This is what we refer to as tagging. This process will automatically assign "the user" as the author.

2. Query

Viewing the project information tailored to a specific interest (using the IDQS to hone in on a particular object). The automated response is where the system presents the user with a new version of the metagraph based on the query

- selecting a node or set of nodes
- selecting the referenced design object, and previewing it or downloading it directly
- refining a search, and having the metagraph redrawn
- selecting a few nodes, and having the MetaGraph focus and redraw (to narrow the search result). This could be used to sphere the selection and create a new node, or to go back to the previous step.
- saving a MetaGraph selection; the graph will be used to generate new sphered nodes throughout the project.
- Loading and unloading design objects through the graph.

The query system is a means of filtering the totality of the information contained inside the PA database. To arrive at the necessary subset of information, It should be possible to query assets by

date, author, option, info type, as well as any other characteristic that has been introduced to the data on the semantic layer via the tags. The results of a search query ought to be able to be further refined by the introduction of additional search criteria or removal of a part of the selection. It may be necessary to combine the results from two or more saved searches.

AI and neural network assistance could help streamline the user interactions with the query tool by suggesting search values based on user activity. From the UI Perspective, narrowing down the project-related information will require multi-selection, and a filter search UI.

4. User Interface Operational Scenarios

To establish the operational scenarios for UI development, we revisited the multidisciplinary and discipline-specific user scenarios set out in Chapters 2 and 3 of D6.1. We must preface the discussion by reiterating that the two user scenarios established in D6.1 are exemplary, and many more such scenarios will be needed to flesh out the requirements of the system. It is also important to repeat that in order to accommodate the needs of the numerous AEC disciplines, the platform will need to maintain a degree of openness, similar to the API approach, allowing it to take on new functionality.

In addition to the user scenarios set out in D6.1, the operational scenarios for UI development are informed by the feedback collected from the testing of the first PrismArch prototype. The testers experienced a lack of clarity regarding the navigation sequence and difficulty with the usability of particular tools, leading to the recognition of the need for a unifying top-level vision for the development of the platform. As the technical partners continue to focus on solving individual technical issues and building discrete tools and functionalities, we feel it is essential to simultaneously consider the development of PrismArch on a holistic level.

The need for a holistic approach initiated a consortium-wide discussion about the PrismArch users' journey (figure 4a). This diagram was shared with the PrismArch team to agree on the goals for the MVP. For ZHVR, this diagram guided the focused studies of the PrismArch environments and scene transitions, and shaped the understanding of the requirements for the IDQS and other proposed tools.

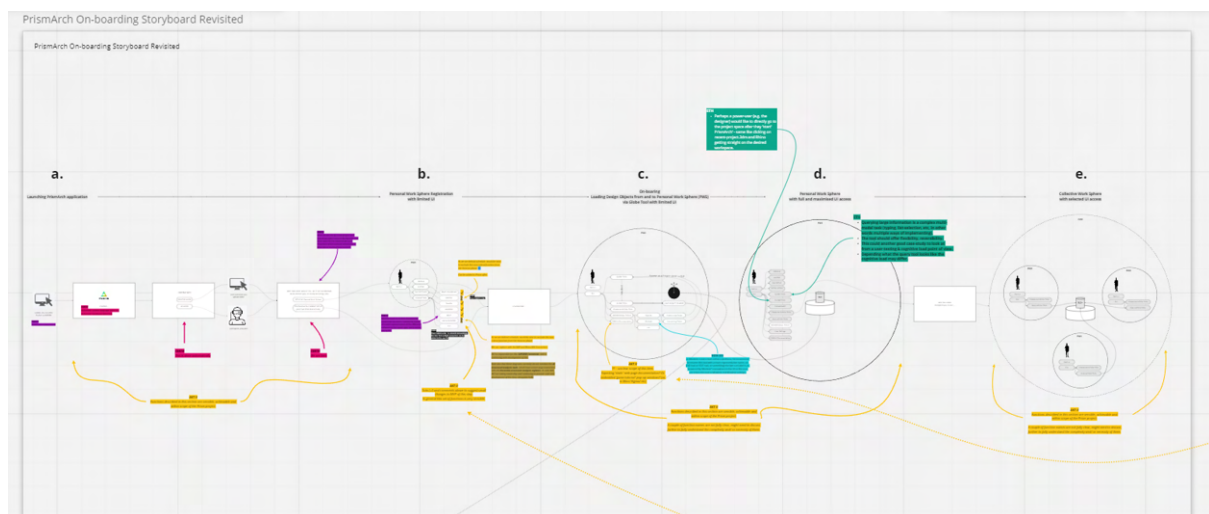


Figure 4a. ZHVR diagram in Miro, illustrating the journey from starting the application to the exit. (See full-resolution image in the Appendix)

The table in figure 4b shows our process for extracting essential functionality from the user scenarios. We analysed the demonstration narratives in terms of: the AEC tasks at various stages of the project's development; the users involved; the sphereing level of the data; the actions and sub-actions necessary to complete the task in PrismArch; and the required tools and functionality. We also noted the types of data formats used in the examples; the accumulation of project information over time; and the accumulation of applicable query criteria for the metagraph interface as the dataset evolves. This enabled us to identify the core requirements (MVP) and helped us visualise how the proposed tools and functionalities work together to enable an exemplary AEC workflow.

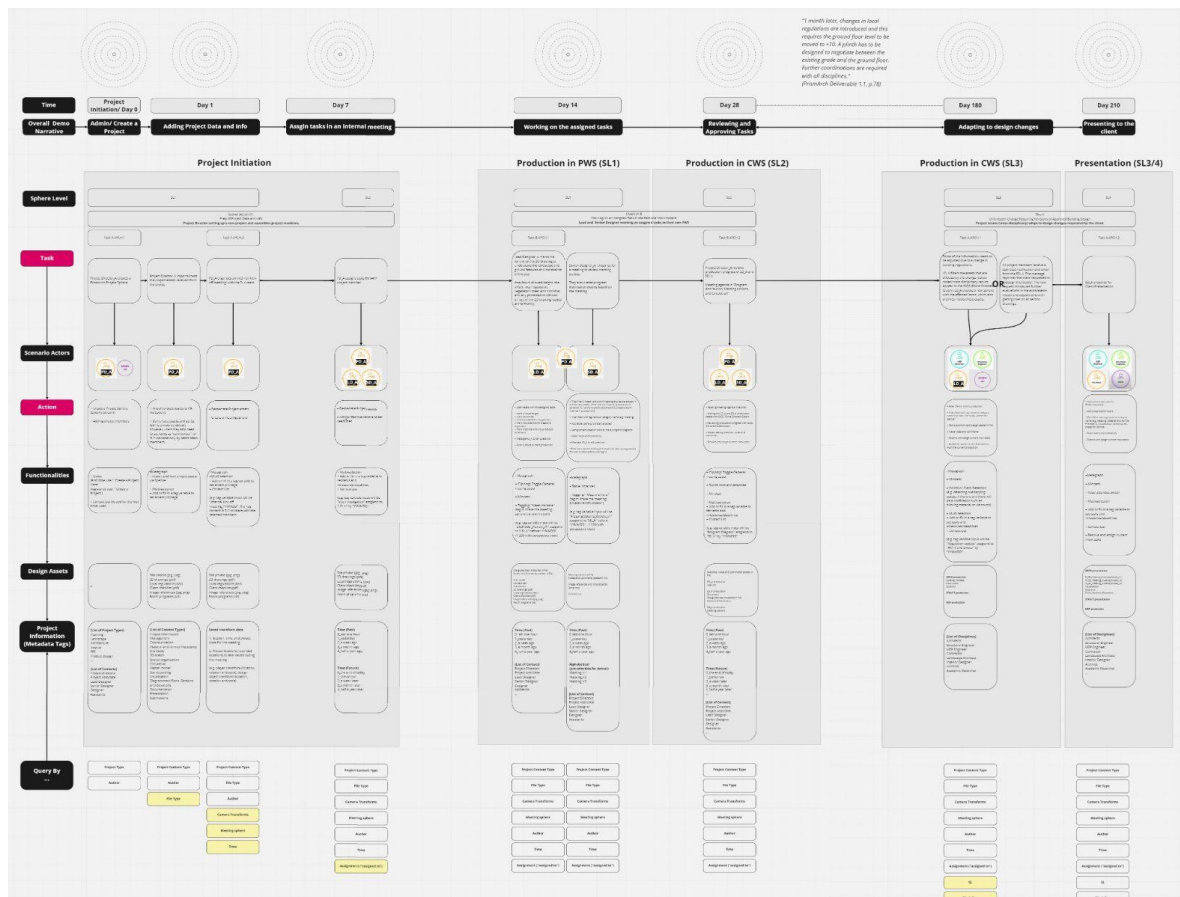


Figure 4b. Table of Operational Scenarios based on D6.1 user scenarios. (See full-resolution image in the Appendix)

Through the analysis of the aforementioned multidisciplinary and discipline-specific use case scenarios described by all disciplines in D6.1 and formalised in the D6.1 operational scenario diagram (figure 4b), we observed the repetition and self similarity of actions undertaken by the AEC teams on their projects. Examining the case studies, we identified a set of fundamental tasks that occur across all disciplines, which include (but are not limited to):

- **initiating a project:**
 - 1) interdisciplinary (by architect or client);
 - 2) in-discipline (by project lead of every participating discipline)
- **loading externally-created content;**

- creating content in PA;**

- meeting organisation:**

- 1) in-discipline meeting (SL2);
- 2) inter-disciplinary meeting (SL3);
- 3) client meeting / presentation (SL4)

- selection and curation of content** to be reviewed during a meeting (saving camera locations [1:1 scale]; setting up whiteboards, loading multiple instances of the model, etc):

- 1) in-discipline meeting (SL2);
- 2) inter-disciplinary meeting (SL3);
- 3) client meeting / presentation (SL4)

- sharing content** a) for further development, as part of the architectural design process; or b) the submission of a completed work package:

- 1) within the internal project team (SL2);
- 2) between different disciplines (SL3);
- 3) issuing content to the client – e.g. as a progress update, an RFI (request for information), or submission of completed design stage (SL4)

Furthermore, essential functionalities and tools are necessary to complete these tasks. These include content creation tools, multi-selection of objects, tagging functionality (to add necessary information and give meaning to the content), the metagraph (to interact with the project information), access to contacts (to share content with teammates and assign tasks), and all discipline-specific UI, which can be accessed through the related menu via the wrist monitor.

We feel it is essential that the PrismArch platform accommodate these tasks and the related tools and functionalities to enable AEC project development process, both for in-discipline and for multidisciplinary collaborative scenarios. Additionally, the platform should be open enough to enable specialists to add their applications. Thus, with the help of APIs and further software integration, we will have a means of addressing the full spectrum of disciplinary needs, as well as the customizable functionality required for each branded commercial user organisation.

4.1 Initiating a project

The project initiation activities will vary based on whether the user and organisation are already registered with PrismArch, or whether this is their first time using the platform. Users and organisations new to the platform will be required to complete the onboarding procedure. Once onboarding is complete, the authorised user will be able to initialise a new PrismArch project sphere and create a project team. They will be able to load and tag any available project information, setting the sphereing level and any applicable data restrictions in the process. The authorised user who created the project sphere will also need to assign team roles and manage the collective.

4.1.1. Onboarding: first time user / organisation

The onboarding procedure requires users to authenticate their identity, create their personal PrismArch account, and generate an avatar (alternatively, they can load their personal avatar, adhering to the new Metaverse interoperability standards).

Once the identity verification is complete, the user will be able to link to an existing professional organisation. The professional organisation will need to have a professional PrismArch licence that

will be extended to all associated employees. If self-employed, the user will need to provide their licence to operate as a sole trader.

4.1.1.1 Introductory tutorial

An in-app introductory tutorial would be helpful to help familiarise the user with the PrismArch work environments, UI, and project data handling capabilities. As a popular reference, we suggest looking at the in-app tutorial for the Microsoft Virtual Reality Portal,¹³ which familiarises users with the different features and services available through Windows Mixed Reality and immersive VR headsets. The tutorial covers navigating the virtual home environment, using controllers and speech as inputs, and integrating services and applications¹⁴.

4.1.2. Creating a new project; adding team; adding data

Once onboarded, an authorised user will initialise the project by creating a unique PrismArch project sphere with its unique PrismArch project number. This is equivalent to creating a project folder on the server. Even if there is no other data inside the sphere, the sphere will contain information about the user who created it, the time stamp, and the unique project number.

The user will be able to add their internal team members to the project, as well as any relevant external contacts. They will also be able to set the project location. They will import any assets pertaining to the project – received either from the client or any other party – set the Sphereing Level of this data (e.g. SL1, private; SL2, viewable by all internal team members or by select team members; SL3, viewable to all multidisciplinary parties; SL4 - viewable to all, including the client) and any restrictions if applicable (eg. “confidential”, or “read-only”), and assign tags (e.g. “brief”, “site photos”).

4.1.2.1 Hosting the project

Once a project is initialised, the hosting of the project content becomes part of the contractual overhead costs. At ZHVR, we feel that it is in the company’s interest to run and operate the archive to cover the liability, at least for the 5-year minimum record-keeping requirement, after which the cost of server maintenance could be transferred to the client. Alternatively, the project space could be co-funded and costs shared with the client from the onset. Furthermore, it may be possible to co-finance the project space with additional parties for whom long-term visibility in the metaverse is desirable.

4.1.2.3 Onboarding other contractual members

In the AEC industry, contractual relationships evolve over the lifetime of a project. Through the process of contract negotiations, several contractors or subcontractors for a given work scope may be engaged and involved in bidding prior to the contract being assigned to the chosen company.

Whichever party assigns their contractor or subcontractor also sets out the project parameters that define their contractual relationship. The individual in the project manager role of the parent organisation will have the responsibility of assigning roles and managing the collective. The cyclical

¹³ Microsoft Documentation. Learn Mixed Reality, 10 March 2022.
<https://docs.microsoft.com/en-us/windows/mixed-reality/enthusiast-guide/learn-mixed-reality>

¹⁴ Microsoft Documentation. Start your VR journey, 14 April 2022.
<https://docs.microsoft.com/en-us/windows/mixed-reality/enthusiast-guide/vr-journey>

production team will require onboarding of new entities and also termination of active participation after completion of the agreed work scopes. This will entail updating the directories and updating the live contacts as the project progresses.

After a party submits their services and fulfils contractual role - it must be contractually agreed as to whether they continue to have access to their part or the whole of the persistent project data. In principle, access to the project data can be granted to anyone, even after the completion of the project (eg. through hosting the project on SL5, it becomes available to everyone). Access can also be limited to a certain type of information, and can be granted for a certain limited amount of time. It may be necessary to consider a partitioned database set-up for content authored by any single party, to facilitate these data transactions.

4.1.4 Frequent user experience

We feel it should be easy for users to access their work where they left off. For instance, the user should not have to configure their avatar or log into their office e-mail every time they enter PrismArch. Certain log-in settings, as well as environmental and design object tool presets, will need to be preserved (saved) for every individual user. The system may also need to store the user camera locations, files last-accessed, and frequent contacts.

Any in-app aids and tutorials will also need to be considered, so that the user is not shown the same information on every log-in.

4.2 Introductory meeting; distributing tasks

After initialising the Project and loading the available content, a user should be able to schedule a meeting with their internal team to introduce the project and distribute (assign) work. They will select the assets that they wish to share with the team and create a meeting sphere on SL2, tagging it with relevant input (e.g. "internal kick-off-meeting"). Using their contacts list, they will be able to invite the participants to the meeting. The meeting sphere system functionality will automatically register the date and time when the meeting is set to take place, the participants, and the meeting duration; and this data becomes searchable in the PrismArch database.

Either during the meeting or afterwards, the user can assign tasks to the team and set deadlines. They will describe the task and notify the collaborator. The user may also wish to sphere relevant assets to the task. For instance, the site data ("site plan" 2D drawing [.pdf], site photos [.jpg, .png], and brief [.doc] describing the local zoning regulations) could be sphered to the task of the team member assigned to building the 3D site model. The team member assigned the programmatic studies could receive the room programs (.xls), client checklist (.pdf), and the brief (.doc) together with their task assignment.

4.3 Discipline-specific production

After the initial coordination and assignment of tasks, individual production takes place on the SL1 sphereing level, in the personal work sphere or in a collective co-working setting. It is here that the discipline-specific tasks diverge. The requirements for discipline-specific VR interface were set out at length in D1.2, Section 3.5. Since the main author of this work package is a representative of the architectural discipline, all the mock-ups provided in this document are considered necessary and vital to the architectural design workflow. We also see the general tool-based UI elements (described in Section 5.3) as expandable to receive specialised menu items, connecting out to

specialist functionality. This expansion of the basic functionality will be driven in part by the marketplace.

4.3.1 Structural UI Requirements

In D1.2, the AKT-II team outlined their assumption about how structural engineers will incorporate the PrismArch VR toolkit into their existing workflow.

- 1) *Outside or Inside PrismArch* - The structural engineer examines an architectural model and produces an abstract model. If the Rhino modelling tools are used, this process could be undertaken either outside or inside the PrismArch environment (via the Mindesk API). This abstracted model will form the geometric input for the structural model.
- 2) *Outside PrismArch* - The abstracted geometric model is loaded into external Structural Engineering software, and structural engineering-specific properties are assigned, to generate the complete structural model.
- 3) *Outside PrismArch* - This structural model is analysed and run in the same external software.
- 4) *Outside PrismArch* - The structural engineering results generated by this analysis (along with the original properties, or 'inputs' of the model) are exported from this external software and imported into the PrismArch Database.
- 5) *Inside PrismArch* - These results can now be visualised and interacted with using the PrismArch VR Toolkit.

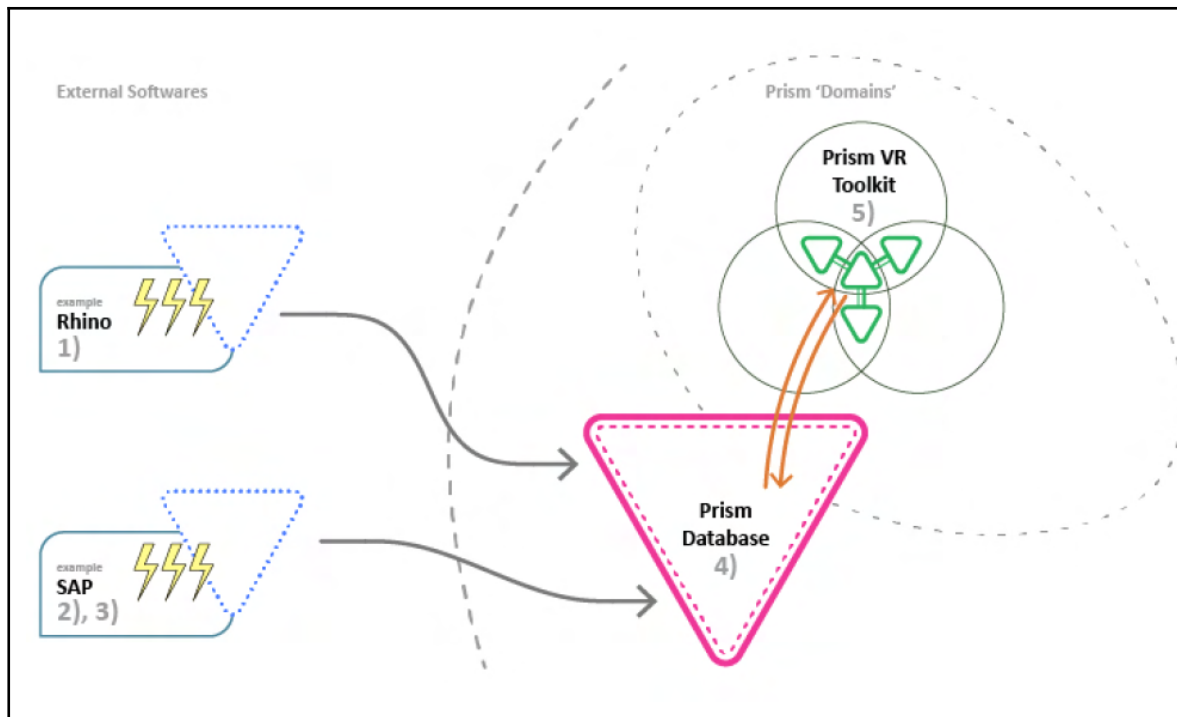


Figure 4.3.1a - D1.2 diagram illustrating how PrismArch could be incorporated into structural engineer workflows

These assumptions were refined by AKT-II during two phases of interviews prior to the submission of D1.2. We feel that the outcomes of Phase 1 are well in alignment with the multidisciplinary AEC operational scenarios presented in this document. Based on the structural engineering use cases, AKT-II generated scenarios where structural results will be shown inside PrismArch VR: 1) briefing

meeting, 2) independent work followed by a design review, 3) presentation of results to external partners. These scenarios are then further broken down into specific actions.

Inside the PrismArch VR, three types of operations will be imperative for structural engineers to engage with structural results: *displaying*, *interacting*, and *exporting*. While *interacting* and *exporting* can be addressed by the Multi-Tool, preview Monitor and/or Whiteboard, and the Metagraph UI, the Multi-tool and Metagraph UI will need to have a “structural preset” to help engineers visualise and query their results inside PrismArch.

The structural discipline requires specific visualisations of forces, visualised as abstract elements “external” to the design object, overlaid onto 3D results: bending moment diagram, shear diagram, axial forces, and torsion (figures 4.3.1b-4.3.1c). This type of visualisation is unique to the structural discipline, and involves the translation of forces into three-dimensional design objects in VR. Another type of visualisation is the representation of analysis results (forces and utilisation) as a colouring mapping, applied to the surface of the examined object (figure 4.3.1d). This type of surface colour visualisation, albeit with different presets, may also be utilised by MEP engineering discipline for the visualisation of environmental analysis.

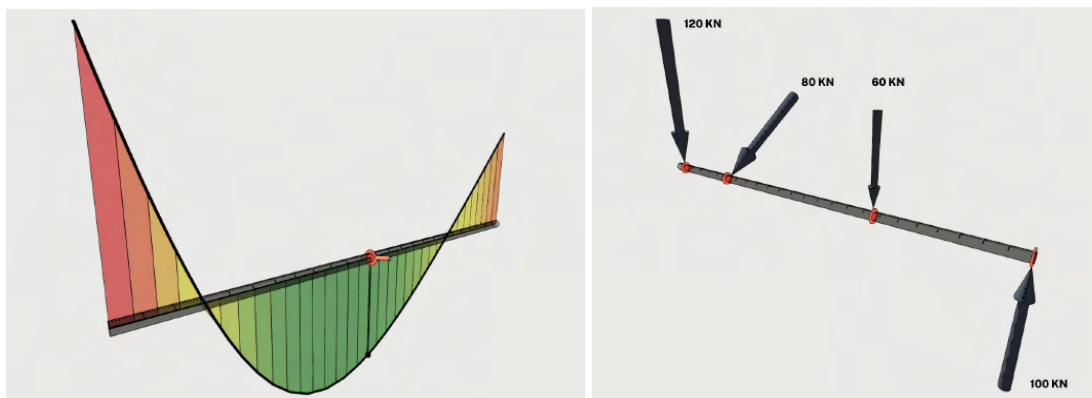


Figure 4.3.1b. D1.2 figures 3.6.3c and d showing the representation of bending moment, and 3D load points.

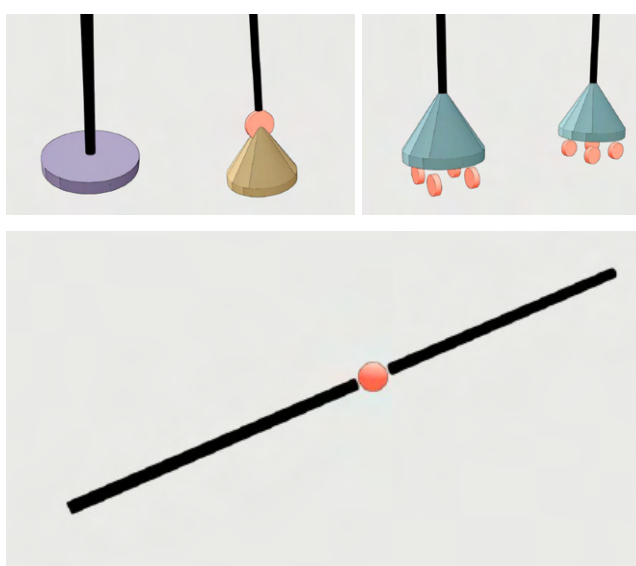


Figure 4.3.1c. D1.2 figures 3.6.3i and j, showing the representations of structural supports

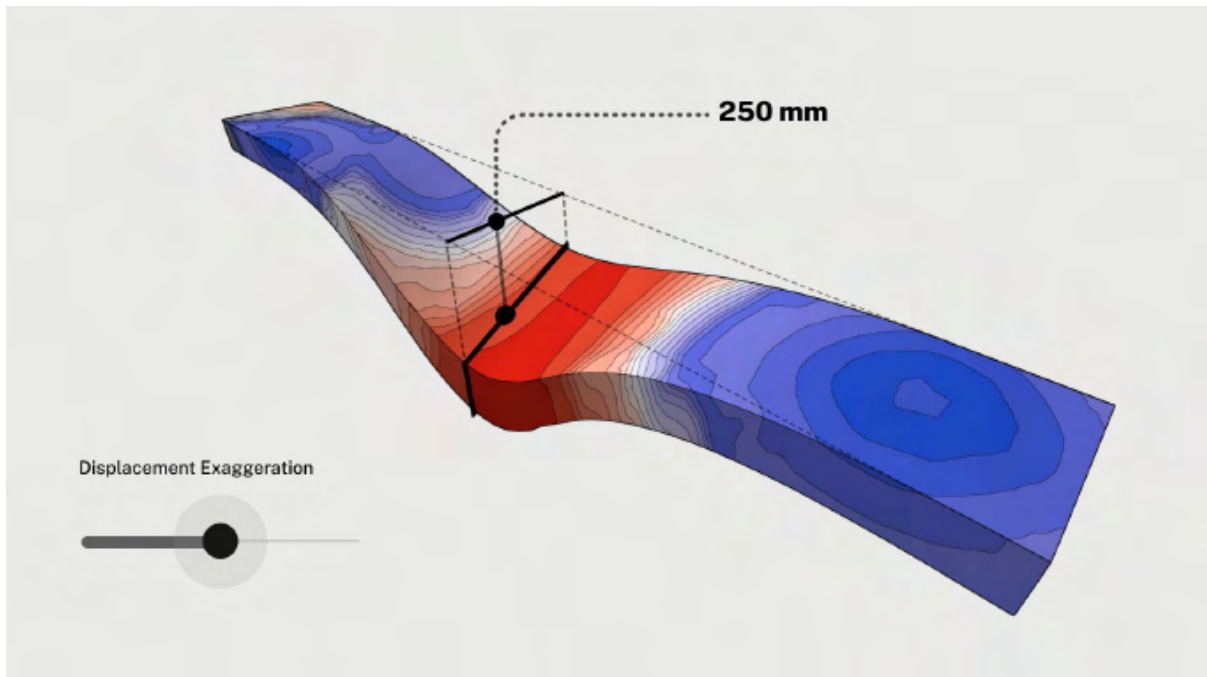


Figure 4.3.1d. D1.2 Figure 3.6.3k. Interactive visualisation of a beam's deformation

The ability to represent text-based values (numeric results from reactions or number values from interaction) alongside the selected object type should be available to all disciplines (see figure 3.1f). Enabled via the IDQS, it will be utilised for the display of various different metadata values (e.g. cost, material finish, or any other type of metadata that may need to be visualised in association with the loaded design object). However, the structural engineering work setup will benefit from a shortcut or preset for the display of numeric values on their results.

Structural engineers perform complex calculations and prepare structural models within specialised desktop software. The results of these calculations are then imported into the PrismArch VR platform to be visualised and interacted with. It would significantly increase work efficiency if the commonly occurring user selections are configured as presets.

Excerpt from D1.2, highlighting the importance of custom structural presets

The Phase 2 interview methodology involved 'open card sorting' of the repetitive actions based on the structural use case scenarios, grouping them into functional categories that made sense to the engineers (figure 4.3.1e).

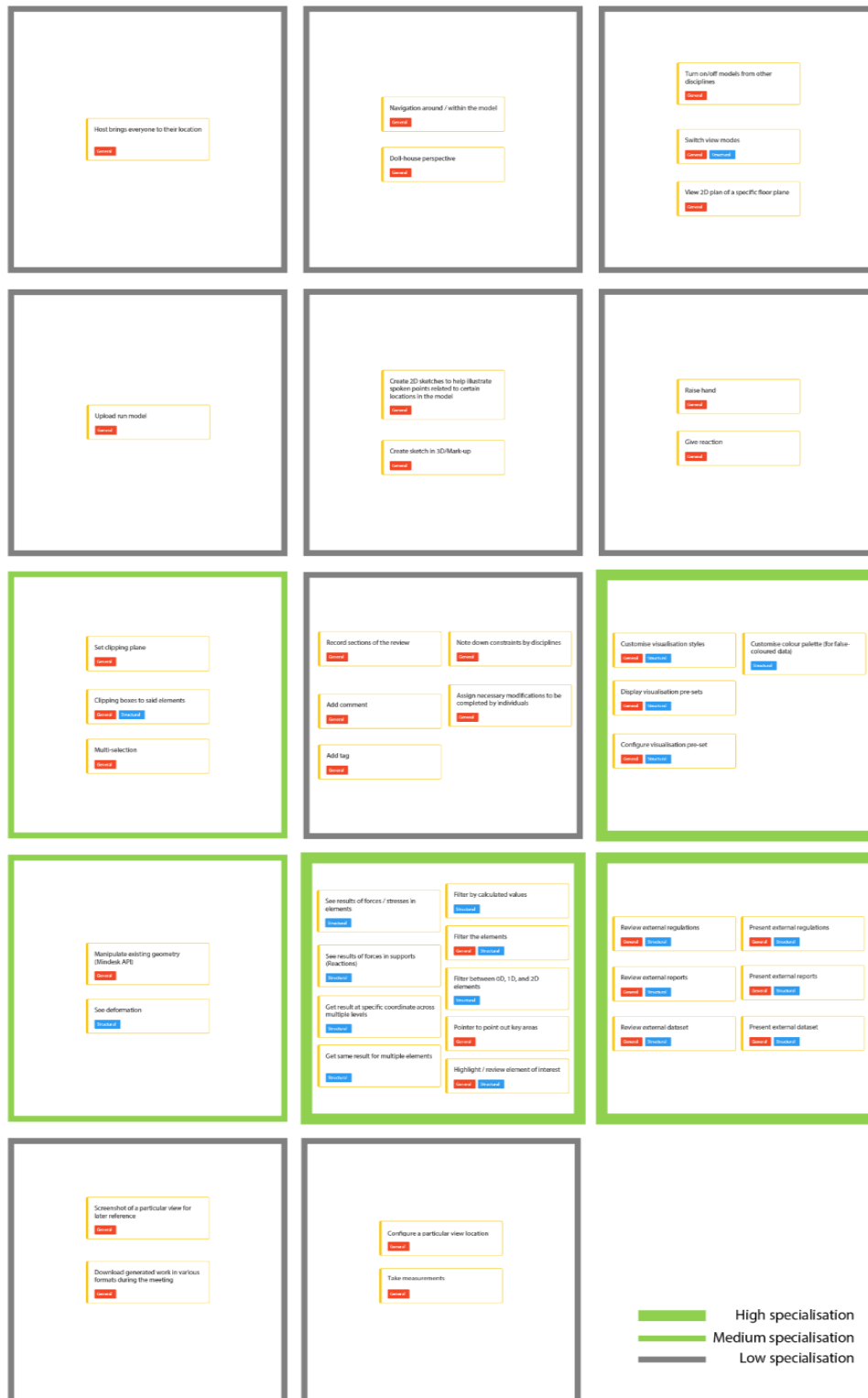


Figure 4.3.1e - Reprint from D1.2: the groupings and priorities after card sorting

Analysing the results, nine out of fourteen repetitive structural engineering tasks are described as “low specialisation” and require the basic productivity tools essential to all disciplines. The two tasks described as “medium specialisation” require multi-selection, clipping planes / clipping volumes, and the ability to manipulate geometry. These requirements are also considered as fundamental to all

disciplines working in 3D, which rely on this core functionality for their 3D production and review workflows.

The third category of “high specialisation” structural UI requirements have three areas of functionality: 1) visualisation presets, 2) filtering of structural results and element preview, and 3) access to external data. In our view, these three functionalities are achievable with the overall UI strategy presented in this document, and will need to be developed specifically for the structural discipline as discipline-specific APIs.

Visualisation presets

The visualisation presets involve customizable visualisation styles, customisable colour palettes, and configurable visualisation presets. The requirement for the filtering of structural results according to specialised parameters is aligned with the functionality of the IDQS and the Metagraph information display system. However, custom API will need to be developed for the specific analytical display modes. For instance, the ability to highlight and preview elements of interest and their neighbours (Figure 4.3.1f), while the remainder of the elements appear as ghosted or in grey is a display requirement specific to the structural engineering discipline, but one that should be achievable through a structural extension to the IDQS and Metagraph UI.

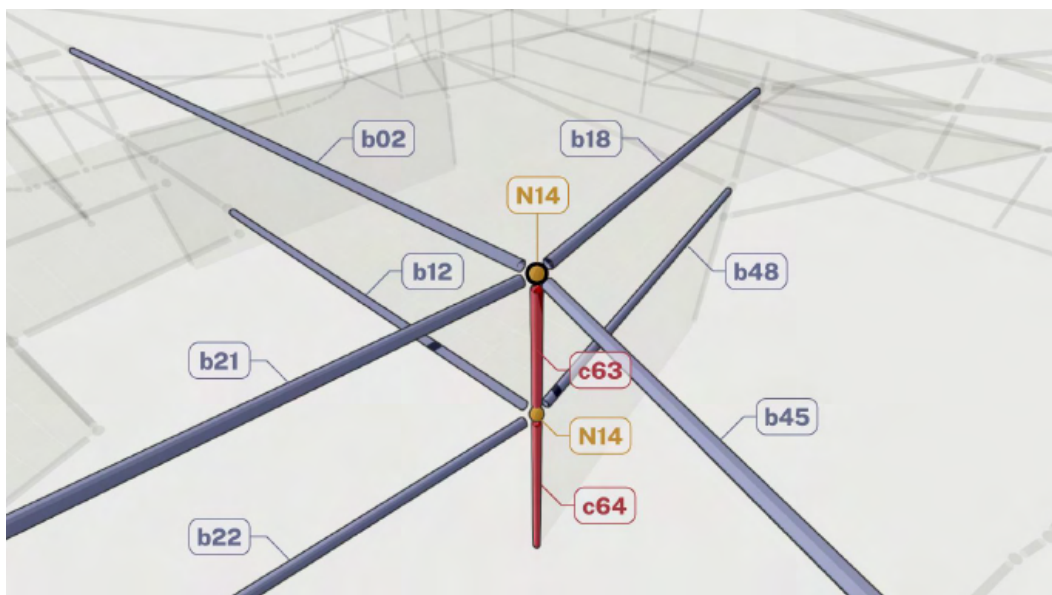


Figure 4.3.1f. D1.2 Figure 3.6.3o, graphically selecting elements, and automatically isolating their topological neighbours

Filtering

Multiple ranges of results can be pre-set and saved for presentation purposes as part of the Multi-tool functionality, described in Section 5.2.4.

External Data

Access to external data – for reviewing and presenting external regulations, datasets, and reports – should be a functionality available across all disciplines. The specific types of structural datasets and the way that they must be interacted with will need to be integrated into the capability of the IDQS.

4.3.2 MEP UI Requirements

The MEP discipline set out their requirements in D1.2, Sections 3.5.3 and proposals for MEP-specific UI in 3.6.4. The key requirements brought to the consortium’s attention are related to: 1) metadata, 2) collision detection, and 3) connectivity to dedicated simulation software.

MEP engineers are tasked with importing, accessing/displaying and reading out/exporting increasing amounts of metadata as a project matures. For example, the information requirement for specified components increases dramatically when a project moves from RIBA stage 3 into stage 4 in the UK (see figures 4.3.2a). To address the stringent requirements and allow for advanced metadata settings inside the VR environment, the MEP discipline will require dedicated specification interfaces.

2 Provide an outline description of the deliverable.

Name	Definition
Description	A description of the type of object to detail any design intent.

3 Provide an outline description of the deliverable.

Name	Definition
Description	A description of the type of object to detail any design intent.

4

Provide enough information to allow the selection of the manufacturer product to meet requirements. This information may also be used to replace the installed product during the operation stage of the building's lifecycle. Information covering the execution of the deliverable should also be provided in the associated specification.

Name	Definition
Manufacturer	The Manufacturer of the High-voltage current transformers.
Standards	An example value being To BS EN 61869-1 and BS EN 61869-2.
Equipment type	An example value being Indoor.
Format	An example value being Bar primary bushing type.
Rated voltage	An example value being 0.72 kV.
Rated short duration power frequency withstand voltage(Ud)	An example value being 3 kV.
Rated lightning impulse withstand voltage (Up)	An example value being 20 kV.
Rated primary current	An example value being 50 A.
Rated secondary current	An example value being 1 A.
Number of secondary windings	An example value being One.
Rated short time current	An example value being 12.5 kA for one second.
Instrument function - Measuring current transformer - Accuracy class	An example value being 0.1.
Instrument function - Measuring current transformer - Instrument security factor	An example value being 5.
Instrument function - Protective current transformer - Accuracy class	An example value being 5P.
Instrument function - Protective current transformer - Accuracy limit factor	An example value being 5.
Rated output	An example value being 2.5 VA.
Rated frequency	An example value being 50 Hz.
Operating temperature range	An example value being -5°C to 40°C.
Creepage distance (minimum)	An example value being 500 mm.
Dimensions	The Dimensions of the High-voltage current transformers.
Winding lead length (minimum)	An example value being 1000 mm.
Label information	The Label information of the High-voltage current transformers.

Figure 4.3.2a. D1.2 Figure 3.5.3d - Different levels of information of a transformer as the project matures.

Source: <https://toolkit.thenbs.com/uniclass/ss>

For collision detection, MEP engineers must be able to load the architectural and structural models and initiate an automated clash-detection process. To review the collisions, users will need custom view modes in PrismArch to view them in real scale inside VR, and the associated clash detection functionality to view the clashes as a list for cross-checking purposes. This list should be visible inside the model, and be able to be exported.

MEP simulation softwares used to evaluate thermal comfort, air flow, air quality, and other relevant factors will need to have connectivity into PrismArch with custom connectors and dedicated menu items in the Wrist UI.

4.4 Meeting for review and sign-off on production

After a period of individual production, it is imperative for collaborators to come together to evaluate the results (SL2). This also occurs when multidisciplinary teams meet to discuss their design progress (SL3). Similarly, the project team will need to present their design development to the client at the end of each design stage (SL4), or more frequently, depending on how involved the client is in the design process.

Although meetings may be less formal on the internal team (SL2) level, the basic functionality will remain the same as for the meetings taking place on SL3 and SL4. It is essential that individuals and teams are able to curate information they wish to review ahead of the meeting time.

Using the presentation work sphere, the user will set up views of the model or several instances of the model; adjust the display styles according to the information that needs to be communicated and the meeting context; load external simulation content and set the value ranges that they wish to display; arrange relevant images on the presentation boards; and adjust the meeting room parameters as needed to match the purpose of the presentation.

The meeting set-up will need to be recorded as part of the meeting sphere, together with the meeting record, in-meeting mark-ups, production, note-taking, and action points. For meetings taking place inside the 1:1 project simulation, it would be helpful to record the participants' camera paths.

In the meeting, users should be able to compare new designs to the previous versions, or compare several versions of a design. This can be done by viewing multiple instances of the model side by side; or this can be done by viewing as a set of superimposed models (instanced in the same location) that are toggled on/off in visibility. Merging information and datasets for review purposes requires maintaining a clear distinction between the authors. This can be achieved through visually differentiating between the sources of information on both the design object and in the metagraph.

It is necessary to be able to multi-select objects and create new spheres, both ahead of and during the meeting. This is useful for commenting, isolating aspects of the project that need revision or further development, and for assigning tasks.

Ahead of a multidisciplinary meeting dedicated to coordination and clash detection after a change to the building regulations, the lead Architectural designer may demarcate all affected areas:

LD_A creates a new sphere: "items affected by building regulation change" (or "Items to be updated"; "building regulations change") and sends a notification to all the authors of these items saying that these will need to be revised. A meeting will be called to discuss the building regulations change and review the sphere created by LD_A before any action is taken by the team. After the meeting, the respective parts of the project will be pulled back into SL1 for revisions.¹⁵

¹⁵ Architectural tasks and actions based on the D6.1 Section 3.1 user scenario narrative *Situation A: Incidents at the Design Development Stage - Unforeseen Change Requiring Revisions of Approved Building Design*

Similarly to preparing a meeting, submissions to the client or to other members of the team require multi-selection of objects, creating a new sphere, (e.g. “DD submission to client”), creating a new node in the MetaGraph with SL4 (read-only). If any modifications are needed to the submission, one would recall the data and make a new submission for the record.

5. Proposals for the VR User Interface: classification and layout

This section describes the composition of the UI, and provides a definition of where they sit relative to the user and the design object. This section is an evolution of the UI interfaces proposed in D1.2 Section 3.6 based on the functionality outlined in D1.1, Section 4.4 and the user scenarios developed in Chapters 2 and 3 of D6.1, informed by the UI operational scenarios outlined in Section 4 of this deliverable¹⁶.

The guiding design principles informing the UI development are: user-centred design; intuitiveness, simplicity; clarity and immediacy; universality (one set of primary tools for all disciplines); consistency (similarity of interaction across applications); uncluttered visual field (nested functionality, where tools minimise or disappear when not needed); and responsiveness and feedback (dynamic level of detail; dynamic functionality).

We aim to keep the primary interaction with the virtual environment as direct and intuitive as possible, and layer complexity as needed to assist users in performing more intricate tasks. The immersed human is at the heart of the development; they are the driving, scaling, and defining factor in the platform. Because the platform is human-focused, we also develop the virtual environment around the immersed human as part of the responsive interface.

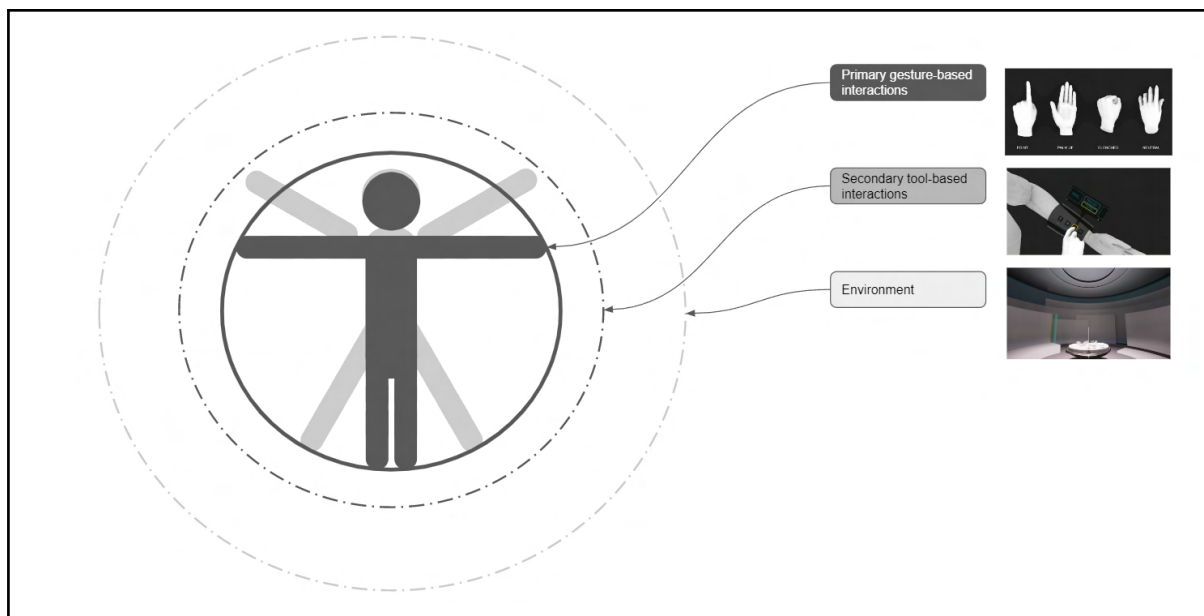


Figure 5a. PrismArch interactions

¹⁶ The proposed UI development stems from our 10 years professional experience as immersive VR designers (and almost 30 years experience as consumers), and the 3D UI reference text by J LaViola et al. *3d User Interfaces : Theory and Practice* (Second Edition), Addison-Wesley, Boston, 2017. However detailed, the majority of these proposals in this section remain theoretical; they will need to be tested inside VR to be evaluated and further refined. We plan to work with ETH in the coming months to test a selection of the proposed functionality.

There are three types of interactions inside PA: hand gestures, tool-based, and environmental (figure 5a). The first and most immediate circle of interactions are the primary gesture-based interactions. These are required for basic functionality, as well as to interact with the tools. The tool-based interactions are one step removed from the IH, who relies on the primary gesture-based interactions to "talk" to them. Tool-based interactions are the next layer after the primary gesture-based interactions.

After the tool-based interaction, the next layer of interaction is the responsive dynamic environment, which adjusts the size of the work sphere based on the contents, and responds to the activity of the user and the amount of co-present users. For instance, if a user loads a model that is too big for the space, the environment can adjust dynamically based on the bounding box of the loaded object. Besides the automated resizing function, there is a level of functionality that can be manually interacted with by the user. We describe the environmental components in Section 5.3, and later provide a detailed description of the operation of the environments in Section 6.5 of this document.

We have organised the UI development into a taxonomy based on the method of interaction. The chart in Figure 5b shows all the various types of operations that fall under each interaction method, and the PA class domain that the operations apply to.

UI Design					
Principles	<ul style="list-style-type: none"> -intuitiveness; -immediacy; -universality; -repetitiveness (similarity of interaction across applications); -uncluttered visual field (nested functionality, where tools minimise or disappear when not needed); -responsiveness and feedback (dynamic level of detail; dynamic functionality) 			PA class domain	example
Methods	interface / interaction	operation		PA class domain	example
	Hand gestures	Primary operations: grab / hold; select; multi-select; deselect Mobility operations: teleport; step backwards / step forwards		All core classes IH (avatar)	
	Tool-based	System operations & UI options: adjust system settings; adjust local tool / environment settings Project & data management (IDQS) operations: search; preview; load; group / sphere; tag; export; comment; invite; notify Work environment selection: -default; modeling / production (synchronous & asynchronous); presentation / meeting -show perimeter window Presentation elements operations: -panels: create new (extrude from floor marker), adjust height and width (interact with edge); minimise / maximise, move (along perimeter), re-order -radial model shelf: (same as panels, except "adjust height, width, and depth") -model tables; clipping planes Mindesk operations: create new (using Mindesk API)		All core classes IH; DO; Metadata Avatar DO; UI DO	
	-wrist UI; -preview monitor; -keyboard; -metagraph; -whiteboard; -presentation wall panels; -presentation plinths; -content creation				
	-multitool	Transform operations: scale / rotate / translate / transform		DO; UI	
	Environmental	Dynamic space operations: -dynamic change of room dimensions (can also be manipulated manually via the tool-based interactions) -indication of SL -transition from enclosed studio space to no studio space (backdrop or immersive sky dome for immersed / full-scale / 1:1 scale)		Avatar	
	-automated (based on number of people, content, etc.) Or can be interacted with via menu-based manual settings /(override				

Figure 5b. UI typology matrix showing the methods of interaction (or interface), the operations enacted, the PA class domain that the operations affect, and examples (thumbnails of mock-ups).



Figure 5c. Diagram of the PrismArch user interface.

The figure 5c above shows the layout of the UI inside the immersive environment. Following the “layers of interaction” concept, the wrist UI and its submenu are attached to the avatar, and provide access to all other tools through a simple menu based on the four PrismArch classes: IH, DO, UI, and Metadata.

5.1 Primary gesture-based interaction

Following the introduction of our UI approach in the preceding Section 5, this section will describe the primary, immediate actions that will be achieved using hand gestures.

The hand gestures utilised are:

- Pointed index finger
- Palm (palm up; palm face down)
- Clenched fist
- Neutral hand

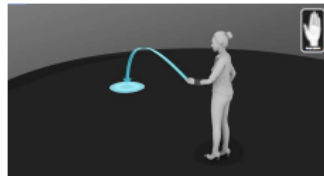
These functions enacted by hand gestures are (but not limited to):

- Teleportation
- Step backwards (stepping forward will be enacted through teleportation)
- Selection and deselection
- Holding & moving objects
- Scaling Objects
- Undo & redo

HAND GESTURES



TELEPORTATION

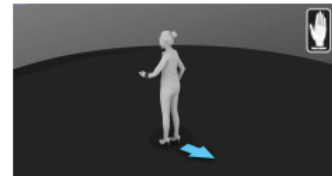


PALM FACE DOWN PRESS TO START TELEPORTATION INDICATOR.

DOUBLE PRESS PALM FACE DOWN TO CONFIRM TELEPORTATION JUMP / HIDE TELEPORTATION INDICATOR

OPERATION TO WORK USING DOMINANT OR NON-DOMINANT HAND

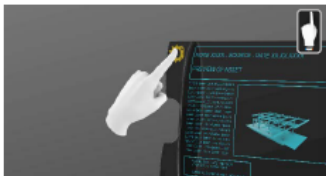
STEP BACKWARDS



POINT PALM FACE HAND FORWARD & DOUBLE PRESS TO STEP BACKWARDS.

OPERATION TO WORK USING DOMINANT OR NON-DOMINANT HAND

SELECTION / CONFIRMATION (VIA BUTTON)

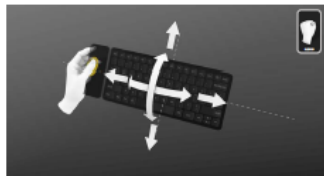


POINT HAND DOUBLE PRESS TO SELECT OBJECT.

POINT HAND DOUBLE PRESS TO CONFIRM/SELECT FUNCTION ON A MENU BUTTON

POINT HAND AND DOUBLE PRESS AWAY FROM OBJECT TO DE-SELECT OBJECT

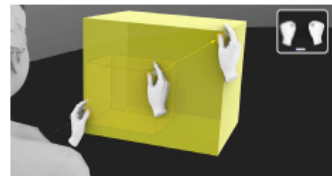
HOLDING & MOVING OBJECTS



CLENCH DOMINANT OR NON-DOMINANT HAND PRESS AND HOLD TO GRAB OBJECT AND TO MOVE OBJECT.

RELEASE CLENCHED HAND TO RELEASE OBJECT. THEN DE-SELECT.

SCALING OBJECTS



CLENCH BOTH HANDS PRESS AND HOLD TO GRAB SELECTED OBJECT TO ESTABLISH UNIFORM SCALE VECTOR. MOVE HANDS TO SCALE OBJECT ALONG THE VECTOR.

RELEASE CLENCHED HAND TO RELEASE OBJECT. DE-SELECT OBJECT.

1D SCALING CAN BE ENABLED THROUGH WRIST-UI BUTTON

Figure 5.1a. Primary Gesture-based interaction / direct interaction (see Appendix for full-resolution mock-up).

5.2 Tool-based UI

The featured UI-Tools will be used in various ways in line with the flow of interactions during or at the start of any operation. The tools will either be spawned by the user or they will appear automatically through prior interactions whenever a user interaction or user input is required with the assistance of such a tool.

Following the above-stated principle of a streamlined and intuitive UI solution for PrismArch, we propose a high level access tool (wrist UI) that allows the user to control the most vital and generic functions in a singular way and in immediate range. Further menus then can be accessed to enter into more specific and non-routine functionalities, or to spawn other tools that offer these more specific functions (i.e. Metagraph UI for data search, or any of the production tools).

5.2.1 In-app guidance and annotation

We propose three types of guidance and annotations inside the PrismArch application: non-interactable indicators; user interaction indicators; and interactive indicators. Each type of indicator works for the following: physical controller and virtual button inputs; tools; project information; and design assets.

All indicators will always face the user avatars to be properly visible. Font and texture sizes can be dynamic depending on the distance from the users to the indicator targets.

5.2.1.1 Non-interactable indicators

In case controllers have to be used instead of gloves or optical hand tracking for the MVP, non-interactable annotation for controller inputs will be necessary. The examples shown in this section are for the purpose of demonstrating the prompt type, and not indicative of the proposed visual style of the desired UI.

Non-interactable annotation for controller input:

This type of annotation indicates a functionality map for controller inputs and lets users know which physical controller button enables what functionality.

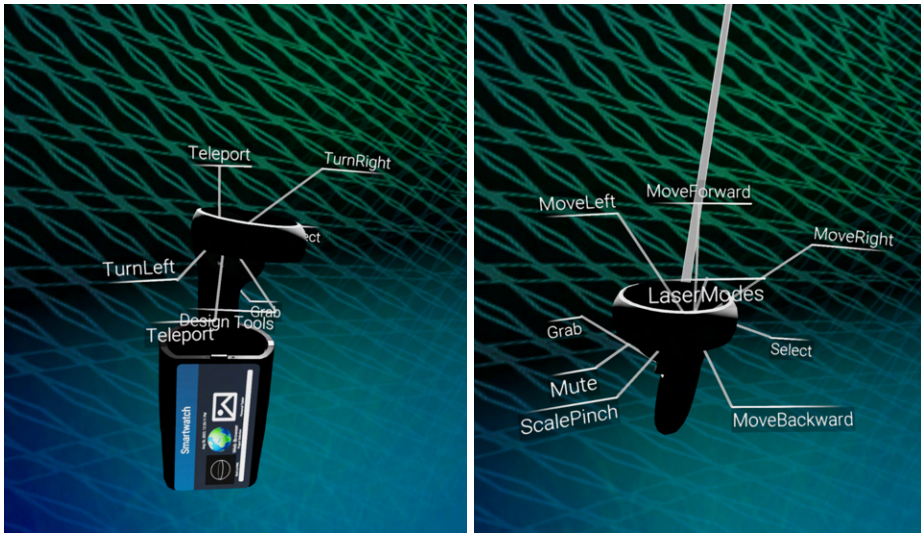


Figure 5.2.1.1a. Examples of leaders for controller inputs

Non-interactable annotation for virtual button inputs:

This type of annotation indicates a functionality map for virtual button inputs and lets users know which virtual button enables what tool. It also works as a description of each button icon.



Figure 5.2.1.1b. Examples of leaders for virtual button inputs

5.2.1.2 User interaction indicators (tutorials and user prompts)

One time user interaction (tutorial):

One time user interaction is a step by step guide or set of instructions designed to familiarise users with a tool.

User Scenario example:

1. Arrow indicator shows up on Multi Tool, together with: “Pull the handle to operate this tool”
2. User pulls the handle in the indicated direction
3. Arrow indicator disappears
4. Next indicator pops up
5. Repeat 1-4 until the instruction is completed.

This type of indicator helps users to learn how to interact with a specific tool, and this can be useful in a tutorial space or in interactive tool documentation.

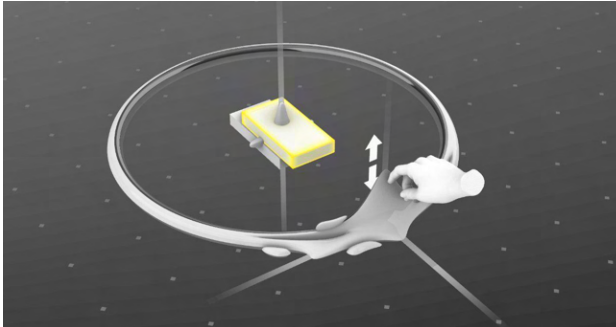


Figure 5.2.1.2a. Examples of interaction guides

5.2.1.3. Real-time UI guidance

Real-time UI guidance is a smart user guide that is visualised when the user is taking specific actions or gestures while using a tool (e.g. Scaling action while using Modelling tool).

These types of indicators help reduce user cognitive load and offer guidance during decision making instances.

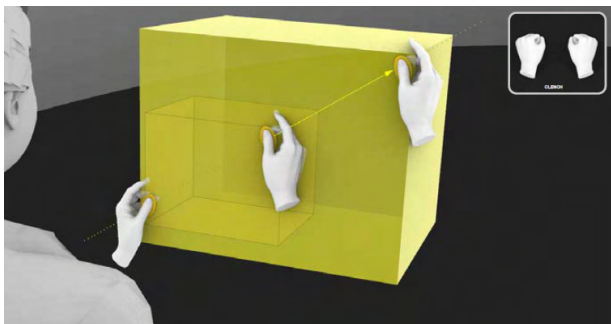


Figure 5.2.1.3a. Examples of interaction guides

5.2.1.4. Interactive indicators

“Look At” Arrow indicator (Marker):

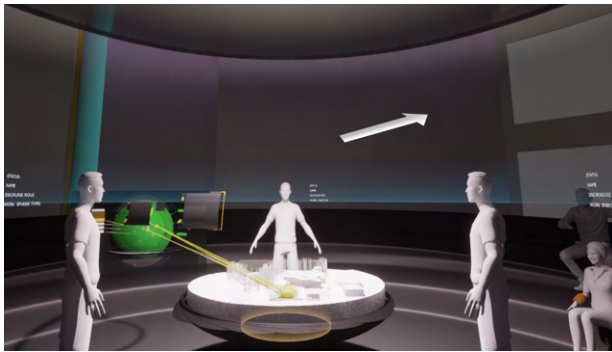


Figure 5.2.1.4a. Example of Look At Arrow indicator user scenario inside Meeting setup

The “Look At” Arrow indicator is an interactive and movable location indicator for selected Project Information and Design Objects. This type of indicator helps users orient collaborators to a specific item that is being referenced (e.g. pointer or laser equivalent during a meeting and presentation). This type of indicator can also work independently as a marker.

5.2.2 Wrist UI

The wrist UI is the default top-level tool inside the PrismArch, providing an entry portal to all functionalities inside the design space, organised according to the four core classes - IH, DO, Metadata, and UI. The wrist tool contains all the menus and functions that one can call in one easy-to-reach location, akin to a smartphone home screen. We also imagine that in the future evolution of the PrismArch platform, disciplines will have additional API functionality in the wrist tool.

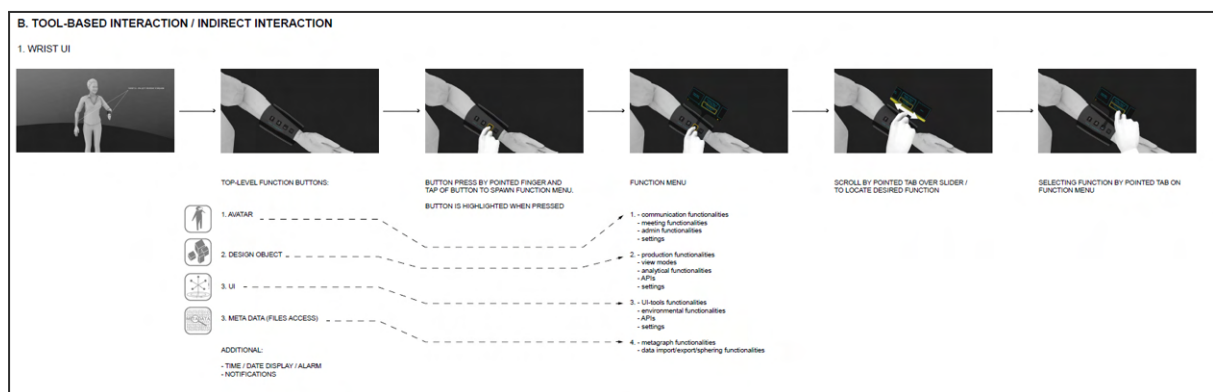


Figure 5.2.2a. Wrist UI mock-up, showing how functionality is accessed via the sub-menu. (Full-resolution image is in the Appendix)

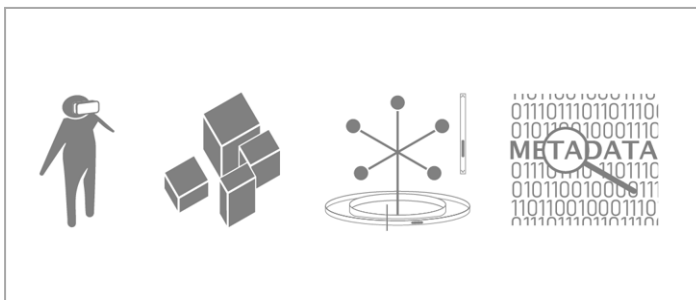


Figure 5.2.2b. Wrist UI including Top-level functionality buttons: 1.Avatar; 2.Design Object; 3. UI; 4. Meta data (files access). With additional time and date display, alarm and notifications.

Function menu

Avatar:

- Communication functionalities
- Meeting functionalities
- Admin functionalities
- Settings

Design Object:

- Production functionalities
- View modes
- Analytical functionalities
- APIs
- Settings

UI:

- UI-tool functionalities
- APIs
- Settings

Meta data (file access)

- Metagraph functionalities
- Data import/export/sphering functionalities

5.2.3. Preview monitor and keyboard

The preview monitor and keyboard constitute the input and output interface. If the wrist UI is comparable to a smartphone home screen, the preview monitor and keyboard are akin to a portable laptop. They are the sub-tool for text- and image-based information, and functionality requiring

numeric or alphabetic input. The preview monitor and keyboard can be accessed from the wrist UI, and will also be made available automatically, when needed for other applications and operations.

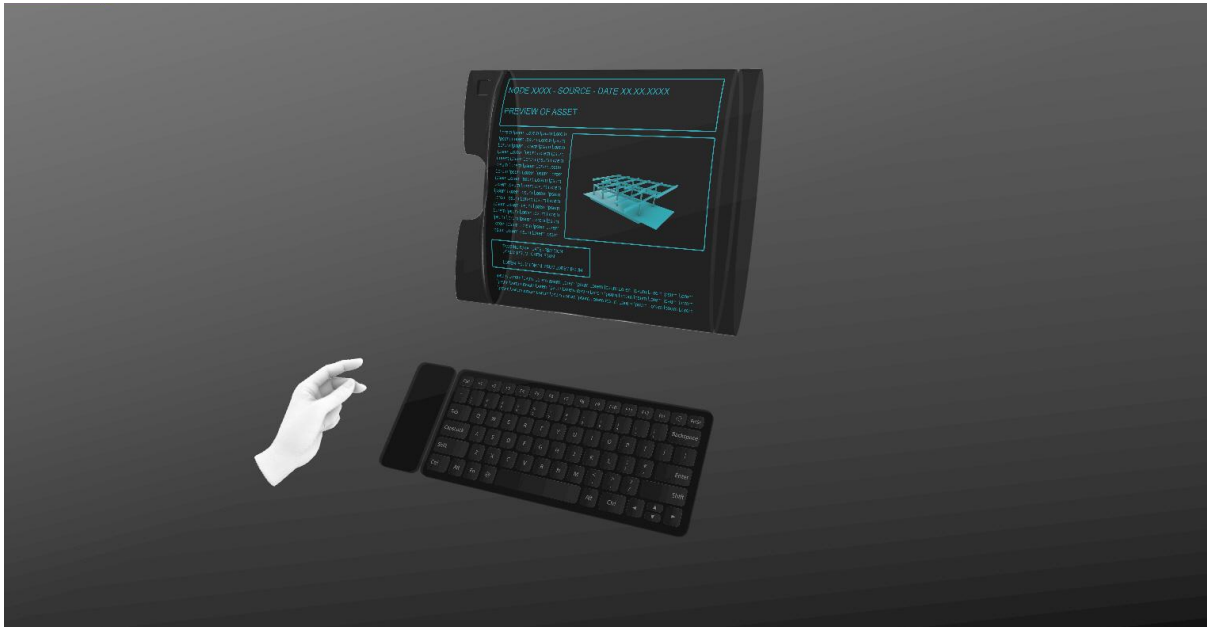


Figure 5.2.3a. Preview monitor and keyboard.



Figure 5.2.3b. Preview monitor and keyboard, right-to-left hand mode.

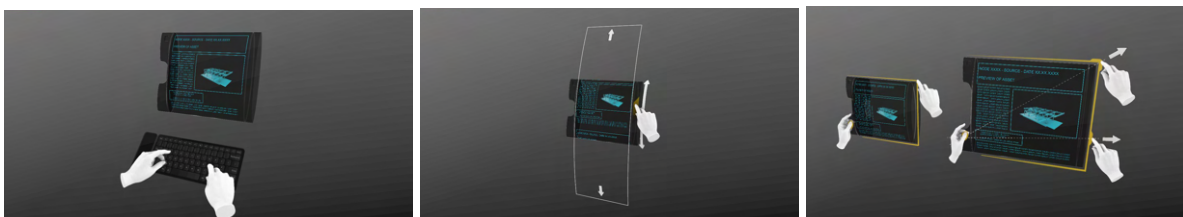


Figure 5.2.3c. Operations: 1. Primary set up for keyboard and monitor; input 2. Scrolling of content on the monitor 3. Adjusting the size of the monitor (full-resolution image is in Appendix)

5.2.4. Metagraph UI

As a functional UI element, the *metagraph tool* (formerly force-directed graph) engages with the metadata class and allows the user to interact with the totality of the project data. It has to be scalable, operable, and must function in a wide array of user scenarios. The operational scenarios were discussed in Section 4, and a number of them will be further developed as operational mock-ups for the Metagraph UI in Section 6.2 of this deliverable.

The user is reliant on the interface of this tool for their day-to-day activity, which is empowered by its functionality. Through testing and further development of the metagraph, we will see how much flexibility and malleability it will need to be usable as a universal tool for engaging with, organising, and managing the information contained in PrismArch.

The metagraph has the following associated *functionality and UI*:

- a) Selection/deselection: multi-selection of nodes, and ability to deselect a subset of the selection
- b) Search: searching by hashtags or keywords and search booleans
- c) Display (of results): organisation by various variables: 1) time (chronologically), 2) alphabetically, 3) topologically (e.g. server folder, subfolder, file, layer, object)
- d) Creation of new Metagraph nodes: 1) ability to create groups, and 2) nesting of information. Users can group multiple Metagraph nodes using the multi-selection tool. When a selection of Metagraph nodes is sphered by the user, it creates a new group node in the metadata, with all the participating nodes nested therein.

The Metagraph also has a set of *visual aids* that demarcate information:

- a) a connector class (a.k.a. “goo”) that highlights the link between classes. In association with the Metagraph, it connects metadata nodes to the relevant loaded design objects
- b) Display methods for coding the relevant information in the Metagraph, such as highlighting content by organisation, or person in a team, or time

The image below (Figure 5.2.4a) shows a meeting scenario, with callouts identifying the core PrismArch classes and the connectivity between them. The immersed humans, represented by their avatars, stand around a design object that is seen at a reduced scale. The metagraph, representing a structured set of organised metadata, appears above the table. There are images on the whiteboards around the perimeter, with a green indicator showing the relationship between the two selected images and their metadata node location.

In the case of the whiteboard image displays, the individual referenced images would be sphered together with their position, size, and orientation data. The layout of the whiteboards would thus be recorded in a separate sphered node containing all the referenced images and layout data, and represented in the metagraph. However, this node is not currently selected (highlighted) in the image.

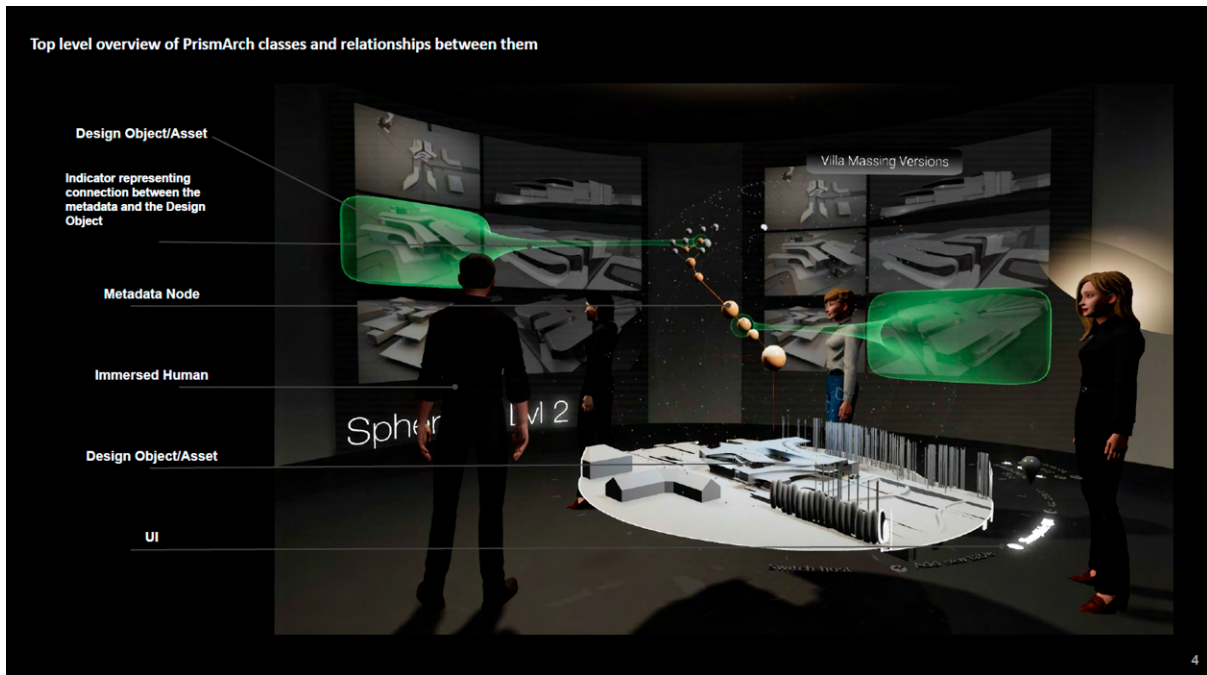


Figure 5.2.4a. ZHVR mock-up showing the PrismArch classes and their relationships.

The metagraph is the interface for the IDQS, which is a part of the PrismArch Content Management System (CMS); the IDQS is always in operation, even when the metagraph is minimised or closed. It is important to note that every recorded action by immersed humans inside PrismArch forms a new metadata record (visualised as a node), as well as every new loaded or created object, every new object group, new sphere, tag, comment, etc. The record, through the information contained in the semantic layer, as well as the DNA of objects and semantic-level tags, links any object or action to the person who instantiated or created it and the time when it was instantiated or created. There are also topological links to previous file versions, and hierarchical nesting structures reminiscent of the windows folder system.

How the data is displayed ought to reflect what data the user needs to (asks to) see, and how they wish to work with it / investigate it. We envision that the metagraph node structure will reorganise itself into different states depending on the parameter (or type of relationship) that is used to investigate the search results. Some possible types of parameters/ relationships are: temporal (chronological order), by discipline, by nested folder structure, by owner/creator of objects, to name but a few. The metagraph remains one and the same, but the results are displayed based on the search query and organisational preference of the user. The same nodes may therefore appear in different locations in the metagraph data visualisation based on the parameter.

5.2.4.1 Metagraph visual aids: orbits and variables axis

Orbits are visual aids that act as a measurement system that is referencing the central point and the gradual increase or decrease of values that are asked for by the user such as time, design stage, etc.

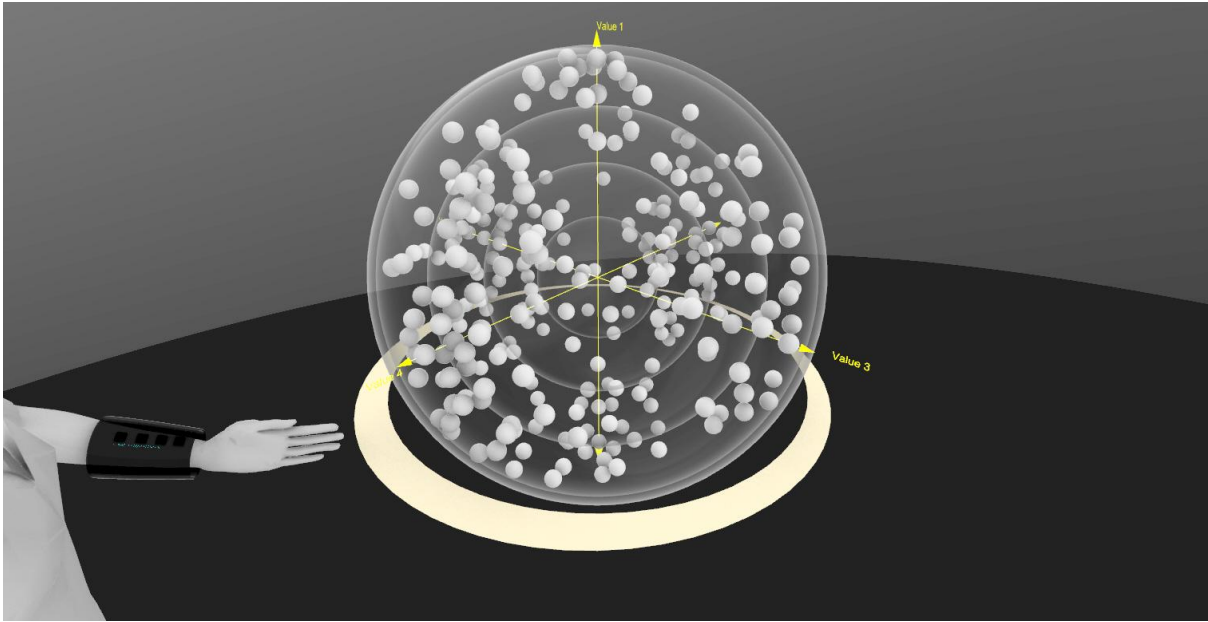


Figure 5.2.4.1a. Metagraph: display of query results based on six search values (we can have n-dimensional search space).

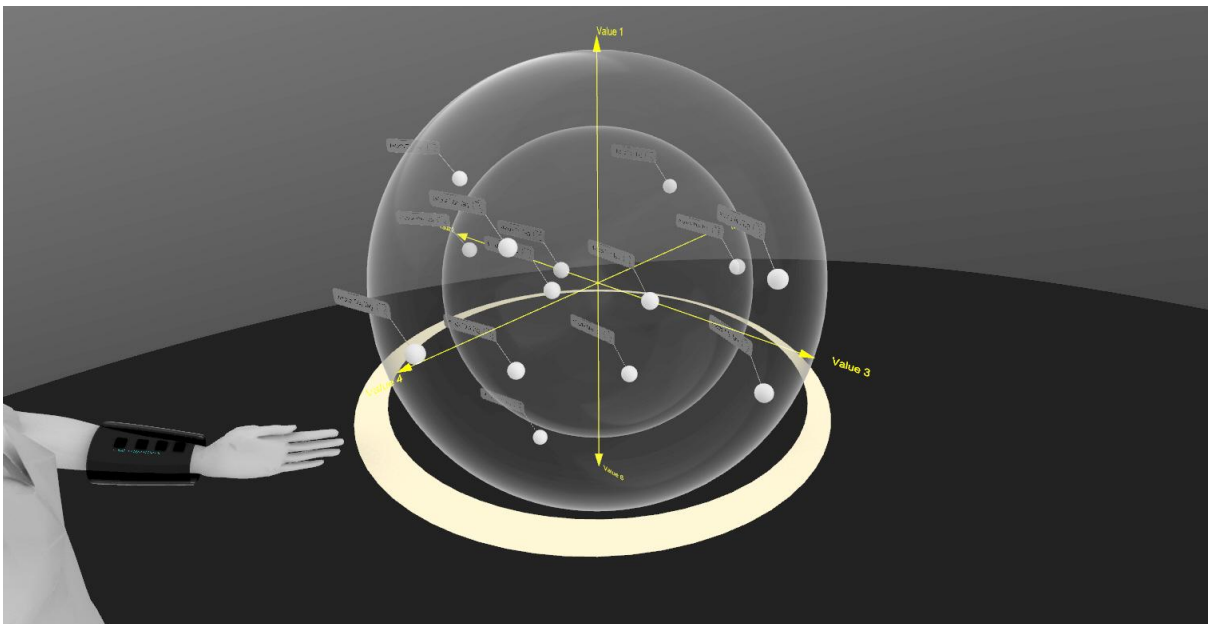


Figure 5.2.4.1b. Metagraph: display of query based on six specified search values. Zoomed in graph results.

5.2.4.2 Nodes

The meta-graph UI component encompasses *nodes* that are representations of metadata. The framing of the search query by the user allows the Metagraph to self-organise and present only the nodes containing information that is relevant to that query; to that situation.

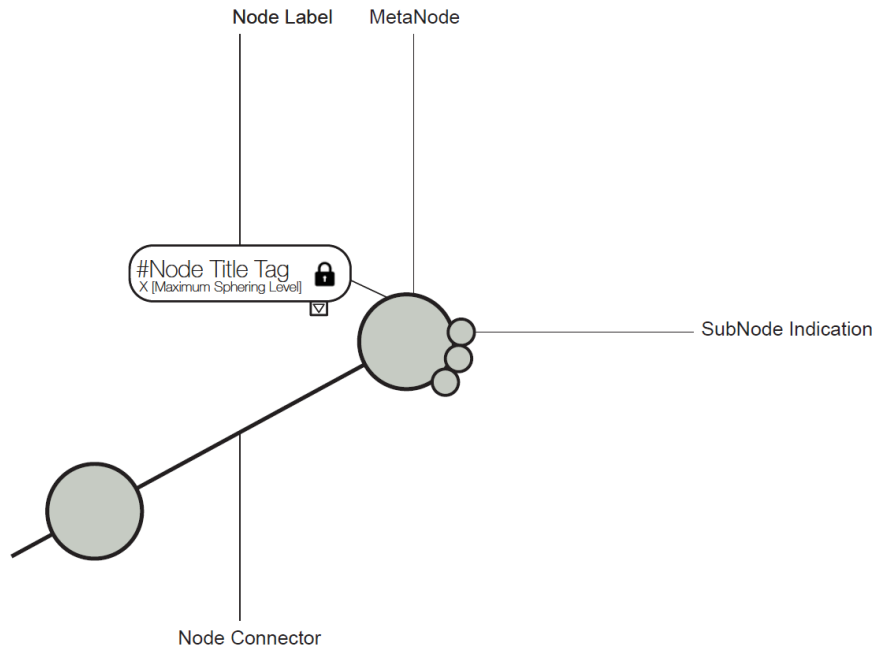


Figure 5.2.4.2a. Metagraph: nodes and nested nodes.

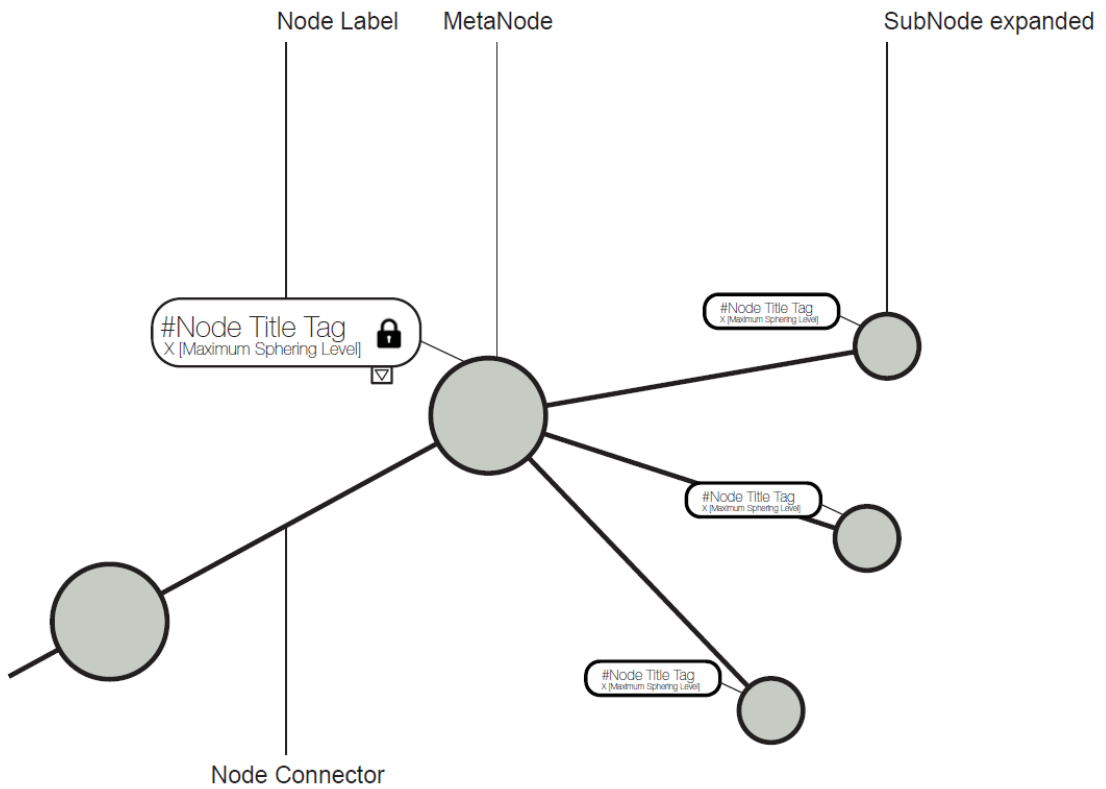


Figure 5.2.4.2b. Metagraph: SubNodes expanded.

5.2.4.3 Labels

The node labels scale together with the dynamic scaling of the metagraph and the increasing or decreasing spacing between the metagraph nodes, moving through a range of label sizes calibrated for the range of distances to the user POV and the number of nodes. The minimised label would only show the title tag in connection to the metagraph node representation. The sphere / node could already be colour-coded to indicate the authoring party.

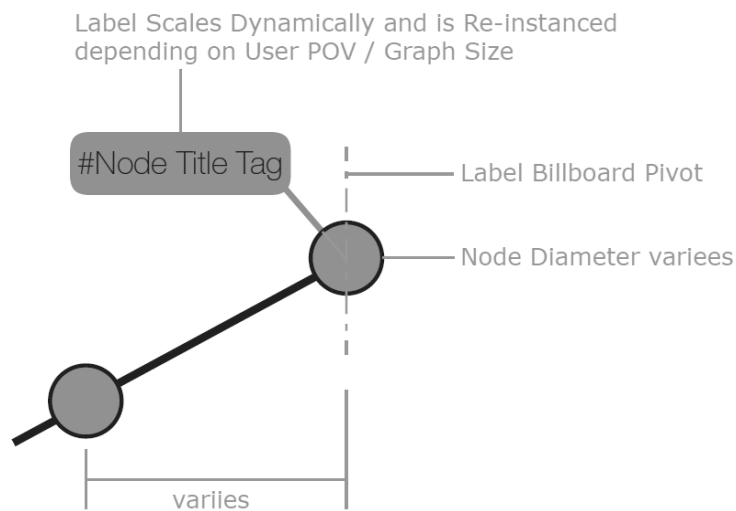


Figure 5.2.4.3a. Minimised Node LOD. All labels billboard and the end of the label connector, attaching to the centre of the connected node. To note: due to visibility issues and anti-aliasing, the label connector / leader line may be lost from view when observed from a far distance or at a small scale.

In the close-range scale, the user can interact with the label to review key aspects of the node. Swiping sideways or upwards reveals the full preview of tag information available.

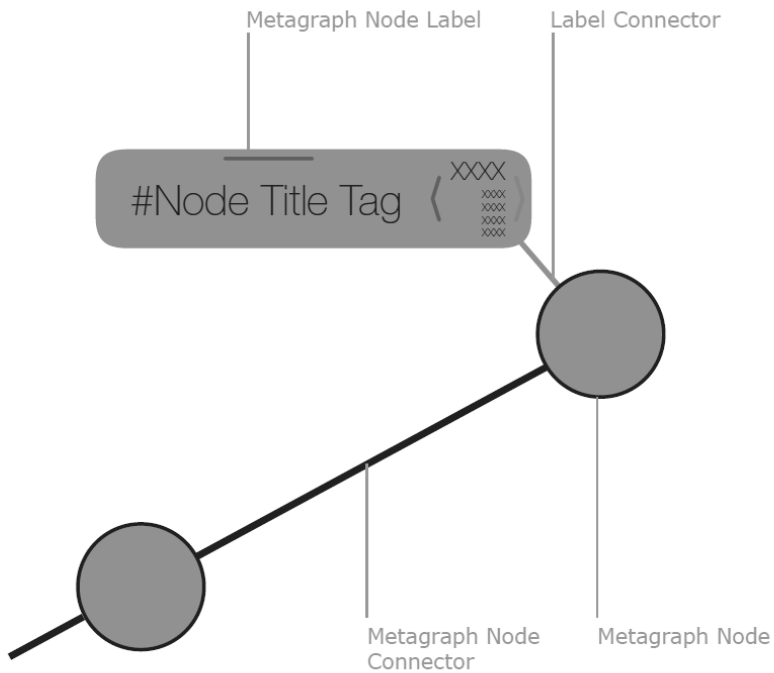


Figure 5.2.4.3b. Node Label - Scaled for Close Range User POV. Arrows on the label allow the user to swipe for additional information.



Figure 5.2.4.3c. Mock-up of metagraph toolkit (accessed by swiping laterally) with functionality relating to the Sphering Level, read/write permissions, visibility, haptic feedback, and auto-reset timer.

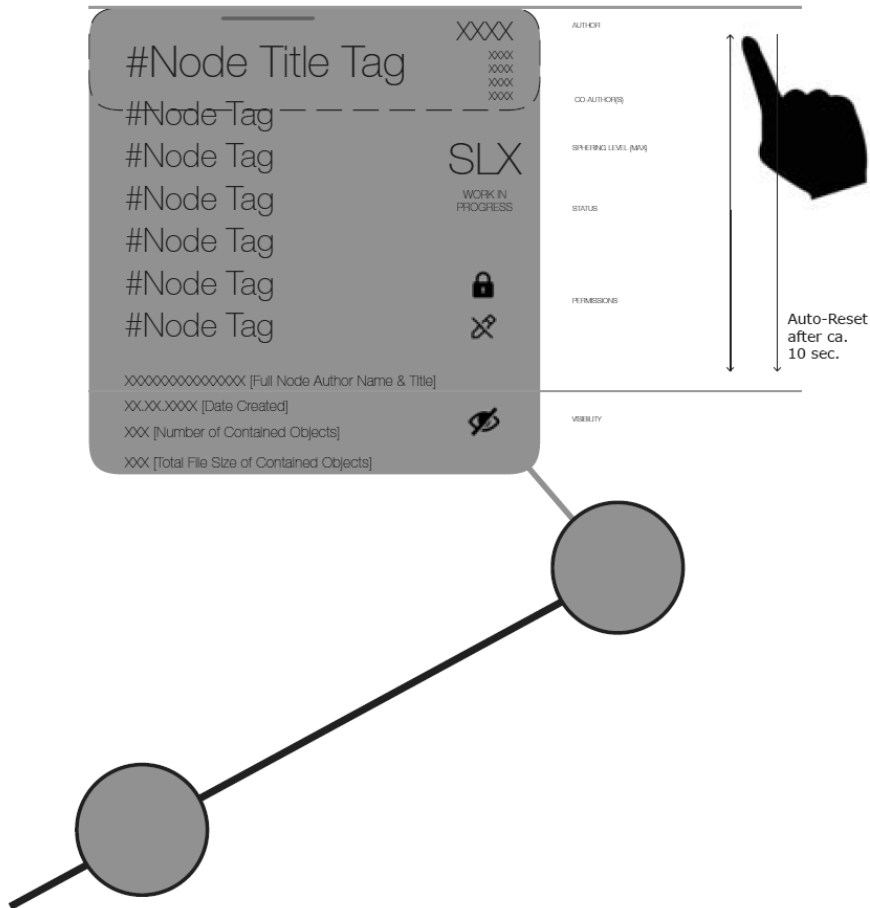


Figure 5.2.4.3d. Mock-up of metagraph label extended information, accessed by scrolling vertically.

To avoid confusion, the swiped off-centre and expanded label views will need to reset or contract after a predefined time period.

5.2.4.4 Connectors

A connector is a visual indicator that works between classes (metadata node and design object). The metagraph “goo” connector is used to represent the relationship between the metadata class and the design object class: a linkage showing the relationship between the index reference / metadata class instance and the design objects. This indicator can be activated from either direction; helping the user to understand the related node in the metagraph, or the related content inside the loaded design object

Because the location and scale requirement for the design object varies, the “goo” connector must be flexible.

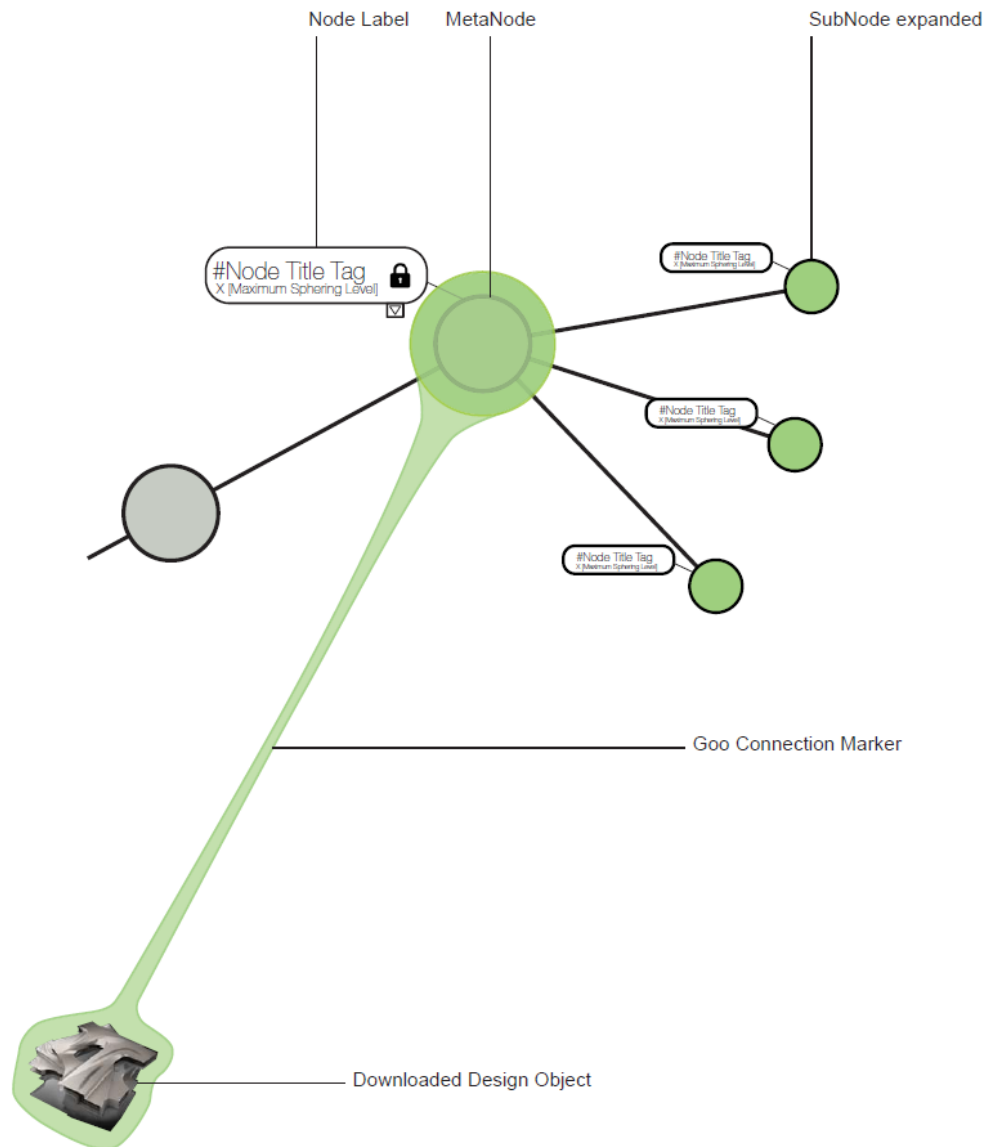


Figure 5.2.4.4a. Mock-up of a metagraph: Goo is a visual indicator of a connection between the metagraph nodes and Design Objects.

5.2.5 Multi-tool

The multi-tool¹⁷ is envisaged as a universal tool for creating and manipulating design objects in PrismArch. It is used in relation to the design object class to create objects, and to manipulate, modify, and adjust their parameters and presets. Visually, it is similar to the gumball UI in various 3D modelling software (e.g. Rhino 7). However, the size, location, and shape of the multi-tool responds dynamically to the use context and the required functionality. Additionally, the multi-tool has expandable functionalities for the different disciplines, which can be extended through APIs that modify the tools' performance while maintaining the familiar methods of interactions.

¹⁷ Previously "Disc UI".

The multi-tool is used throughout the use case scenarios; it is a widely applicable tool that should be easy to learn and enable intuitive interaction with design objects. The tool will physically adapt itself according to the scenario, while maintaining the functionality and consistent operation.

Following the selection of an object, the multi-tool can be called using the Wrist UI. The functionality of the disc can vary depending on the type of object(s) selected (see figure 5.2.5a): design object (3D model), white board, camera object, model table, metagraph, or metagraph node. The functionality of the disc can be locked: 1) to the model table or plinth (see figure 5.2.6b); or 2) to any design object.

E.g.

If DO is selected, the user can use the disc (multi-tool) to adjust (the DO's) location, scale and rotation.

If the Clipping plane tool is selected, the user could use the disc (multi-tool) to adjust its size.

If a camera object is selected it could have camera settings such as ISO, Shutter speed etc.

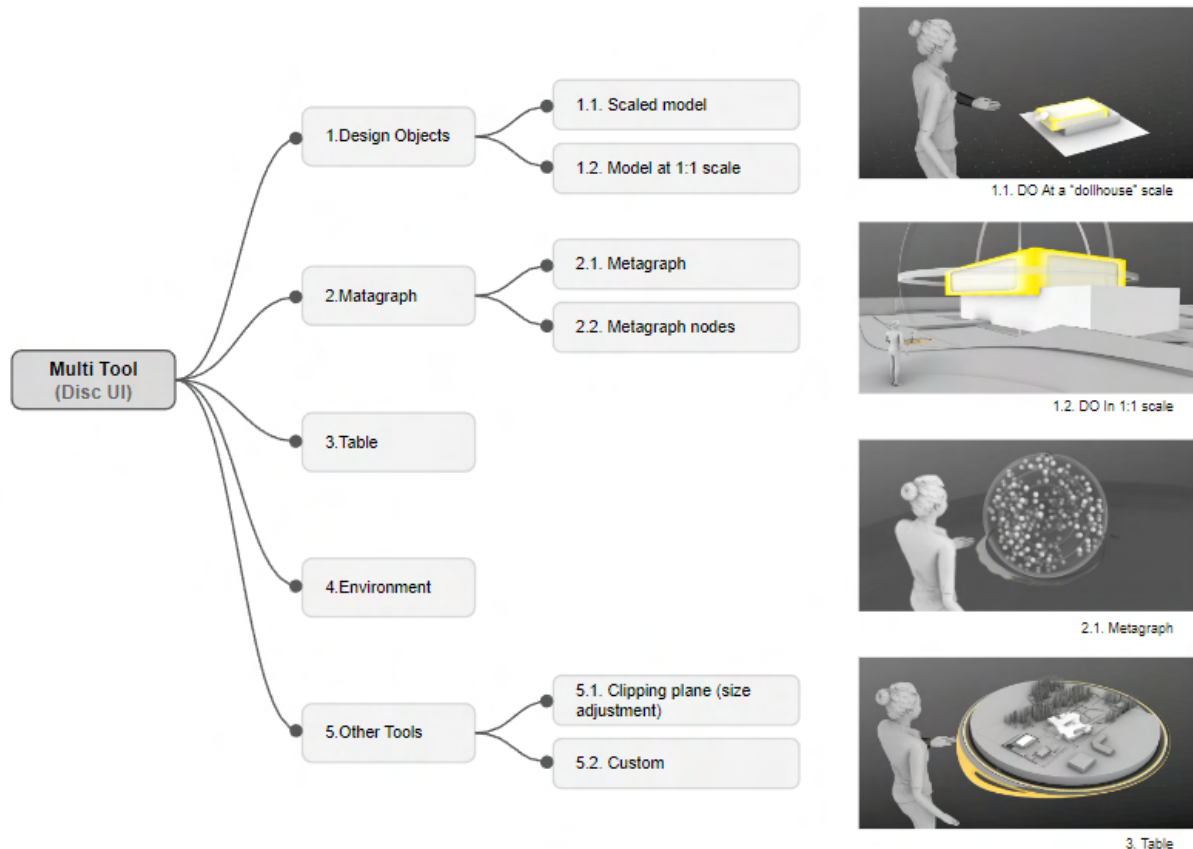


Figure 5.2.5a. Objects that can be manipulated using Multi-tool

The following key use case scenarios were taken into account to develop scalable and adaptable universal solutions:

- 1) *Scale model on presentation table / global model controls*

When a scaled model is on the presentation table, the multi-tool associated with the table controls the entire model globally.

2) *Design object*

The Multi-tool associated with individual Objects in users view and within his/her reach

3) *1:1 (+) scale:*

In this specific case, when objects are large and out of users' physical reach, there is a requirement for the user to adjust objects remotely using the scaled remote version of the Multi-tool.

There are two fundamental modes of operation:

- 1) *Presentation mode:* the tool enables users to preset functionality in order to curate the meeting and discuss various aspects of the project as required.
- 2) *Production mode:* the tool is easily accessible and intuitive in use while working to manipulate and create assets.

The multi-tool allows users to modify and control the following parameters of the selected design object(s):

1. *Transform operations:* rotate, scale, move
2. *Time:* project history (when the asset was created); daylight simulation
3. *Lighting:* artificial lighting presets and controls
4. *Material properties:* colour, transparency, material selection
5. *Display style:* clay, ghosted, rendered;
6. *Camera:* ISO, shutter speed, exposure

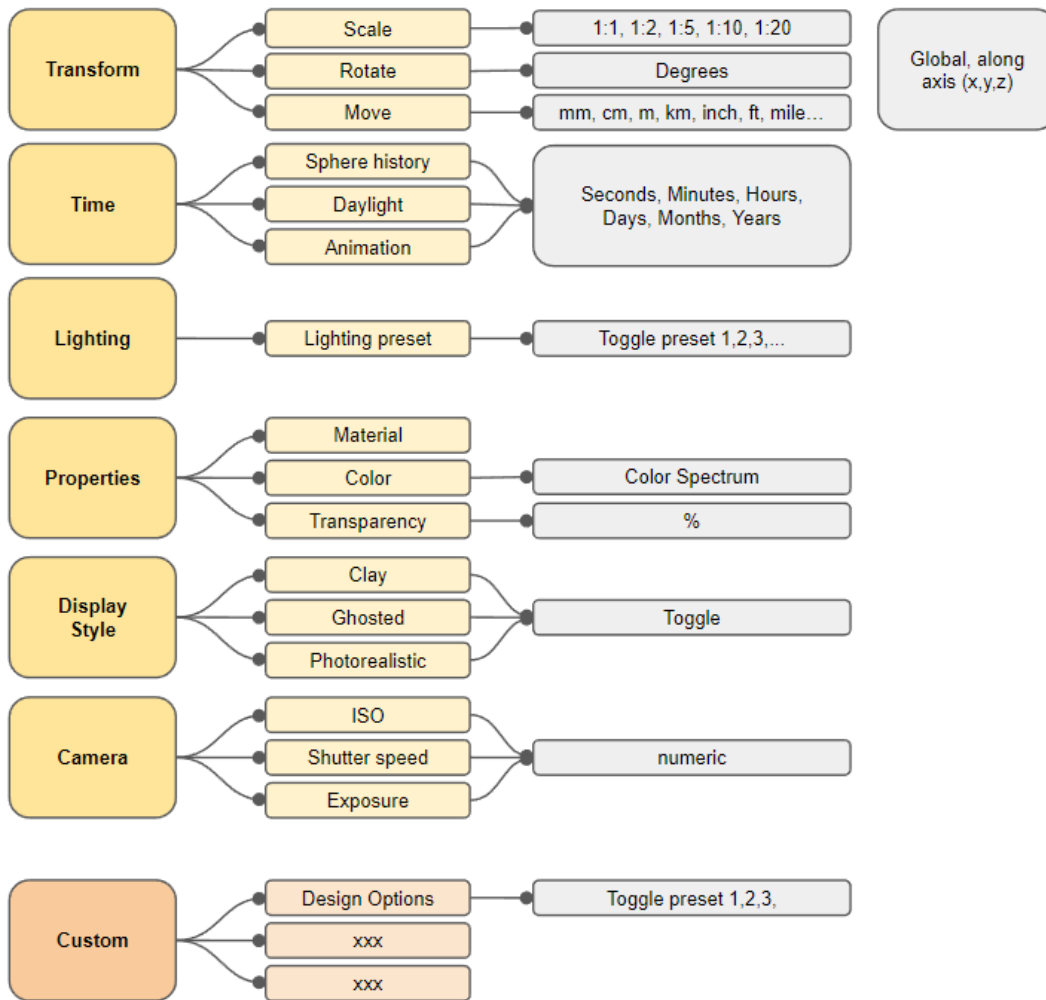


Figure 5.2.5.b. Parameters that can be manipulated using Multi-tool

Additionally, the user can pre-program the tool by assigning custom parameters (e.g.: using the disc to toggle between a number of design options).

The tool consists of the following components:

- *Disc*: Interactive component used to adjust parameters and manipulate objects in PA
- *Guides*: Indication of the direction of user actions, and of the units (visually distinct from the Disc component)
 - *Pivot point*: Component associated with transform operations, Indicating centre from which the object is manipulated from

Depending on the use case scenario the multi-tool transforms and adapts its form to accommodate the user’s activity. We envision the multi-tool to fluidly adapt its form in response to user interaction and required functionality. Therefore the tool shapes vary from a simple disc for slider modifications to a more complex form that enables the user to manipulate object transform parameters.

In addition to the fully activated and functional form, we propose a minimised variant, which is displayed when the tool is not interacted with.

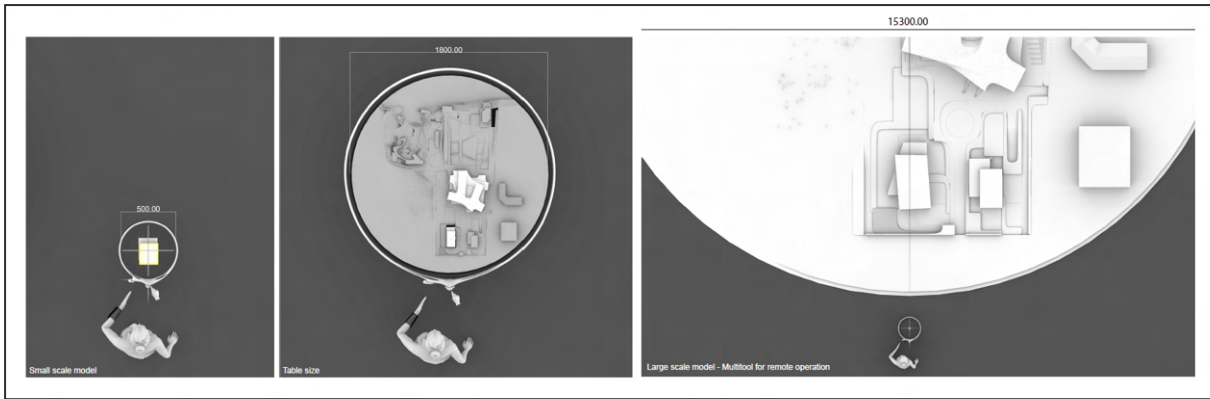


Figure 5.2.5c. The Multi-tool responds to the scale of the selected model as well as distance between user and selected object: 1) Multi-tool inscribing a small size model; 2) Multi-tool inscribing a model on the meeting table, allowing user to make global changes to the model; 3) Remote variant of the Multi-tool allowing user to make modification to large size design objects, in 1:1 or larger scale. (See full-resolution image in the Appendix)

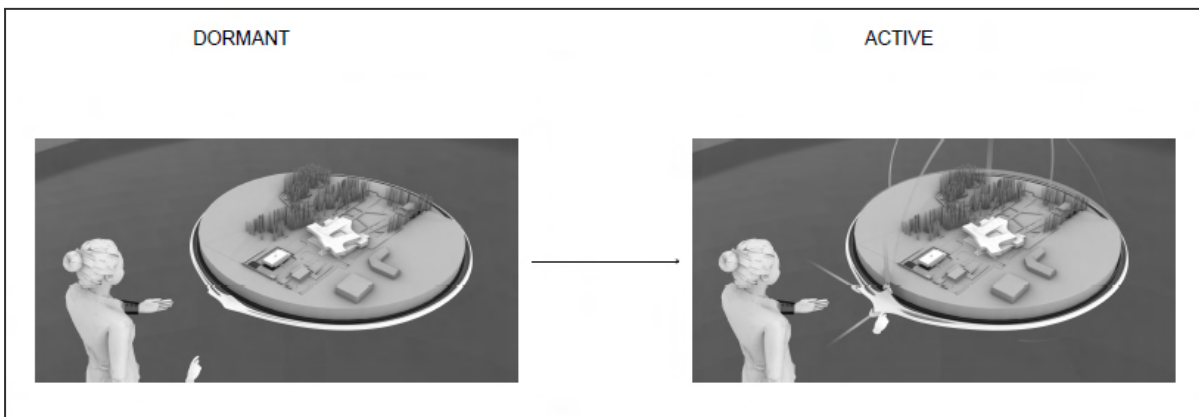


Figure 5.2.5d. Multi Tool adjusting its form depending on the user interaction required functionality. Transform operation from the left: scaling, translation, rotation. (See entire mock-up in full-resolution in the Appendix)

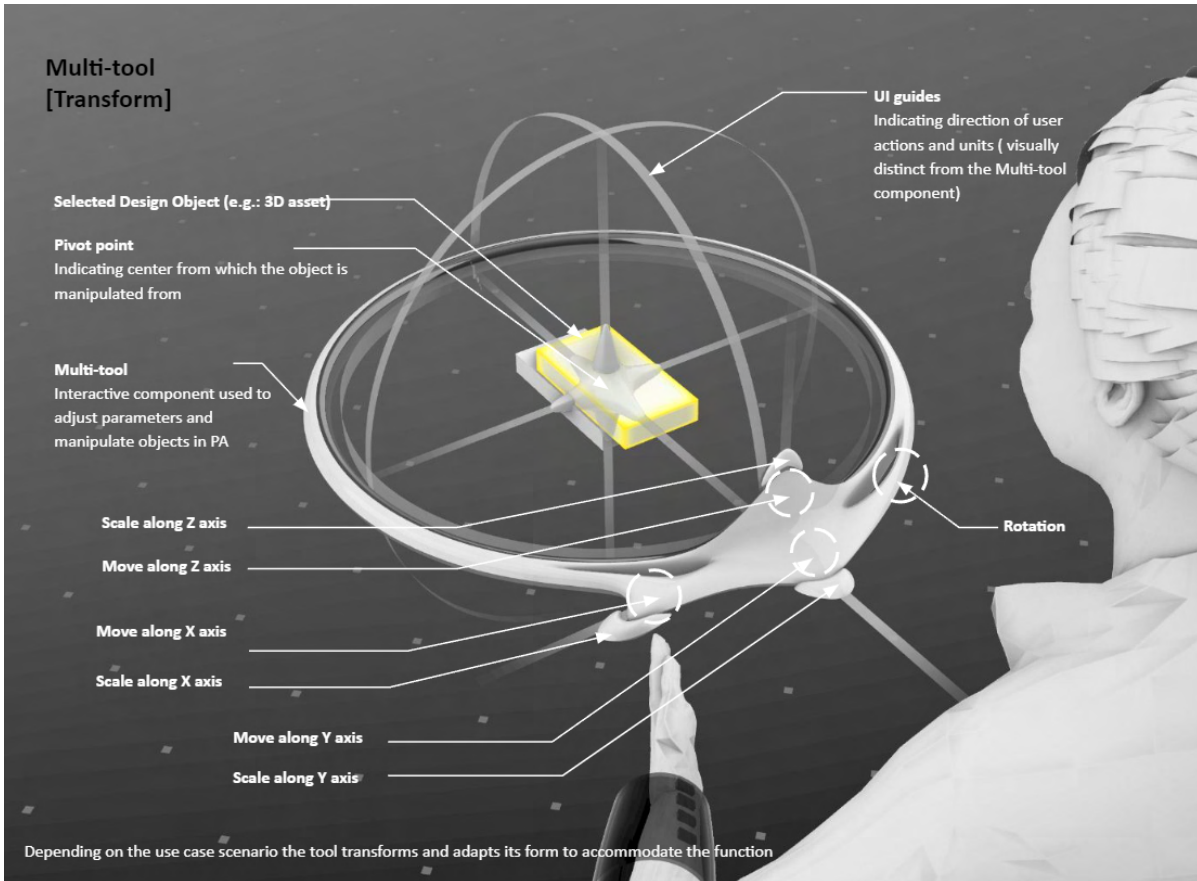


Figure 5.2.5e. Components of the multi-tool variant for manipulating objects transform parameters: adjustable pivot point; multi-tool disc with interactive zones to access required functionality; guides.

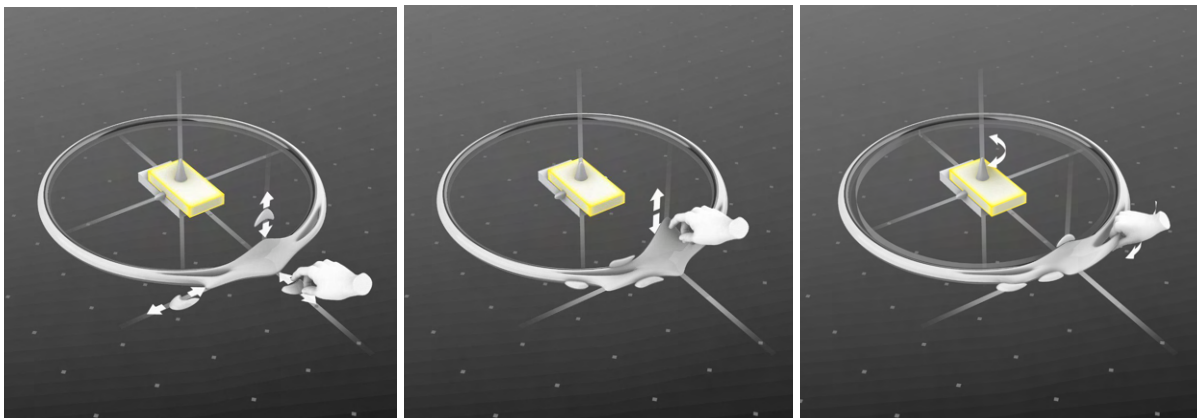


Figure 5.2.5f. Multi-tool adjusting its form depending on the user interaction required functionality. Transform operation from the left: scaling, translation, rotation.

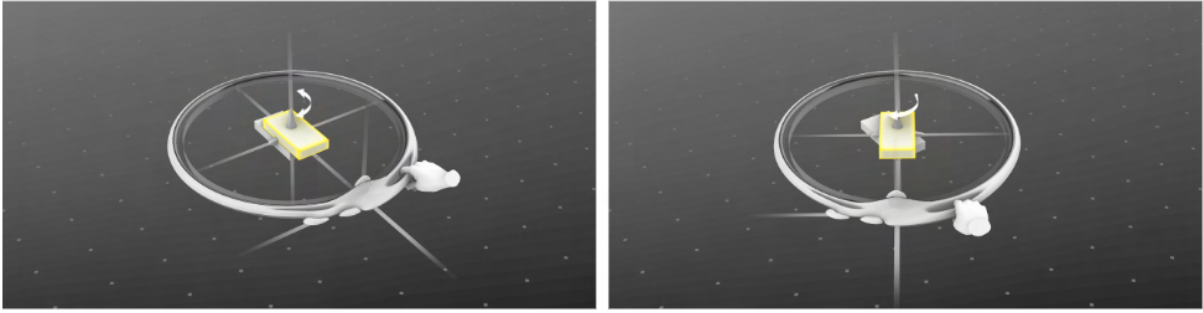


Figure 5.2.5g. Rotation: pivot point at the default object centre location

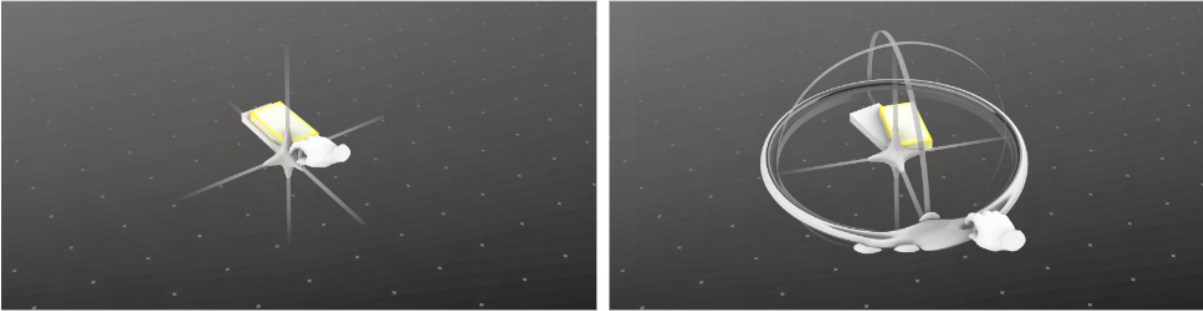
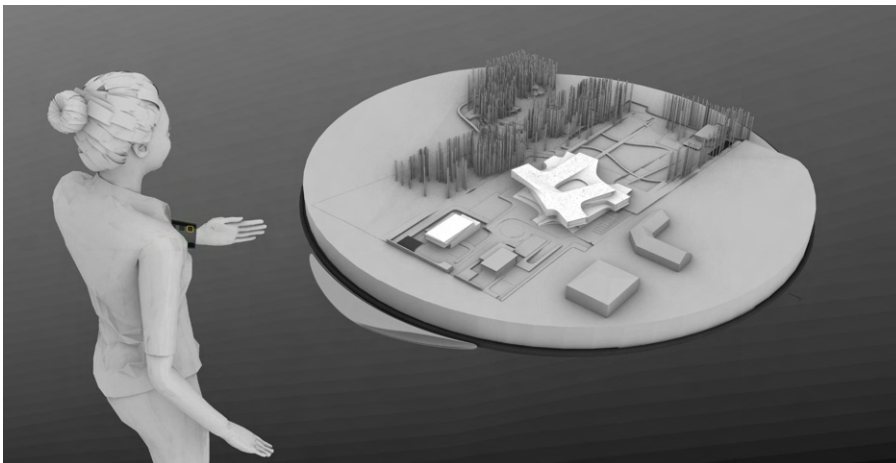


Figure 5.2.5h. Rotation: pivot point adjusted to a custom location

Presentation mode:

Users should be able to create a curated presentation and save various model configurations or states in order to discuss them during a meeting. Saved functions are attached to the meeting table. The Multi-tool is maximisable from a persistent UI tab attached to the meeting table. When maximised, the tab expands presenting all functions.



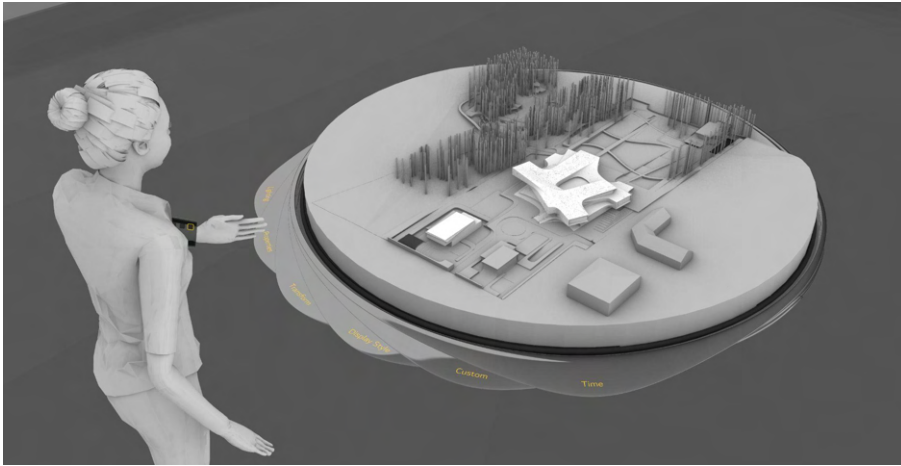


Figure 5.2.5i. Top: Minimised Multi-tool attached to the table. Bottom: Series of tabs with functions pre-programmed by the user for presentation purposes.

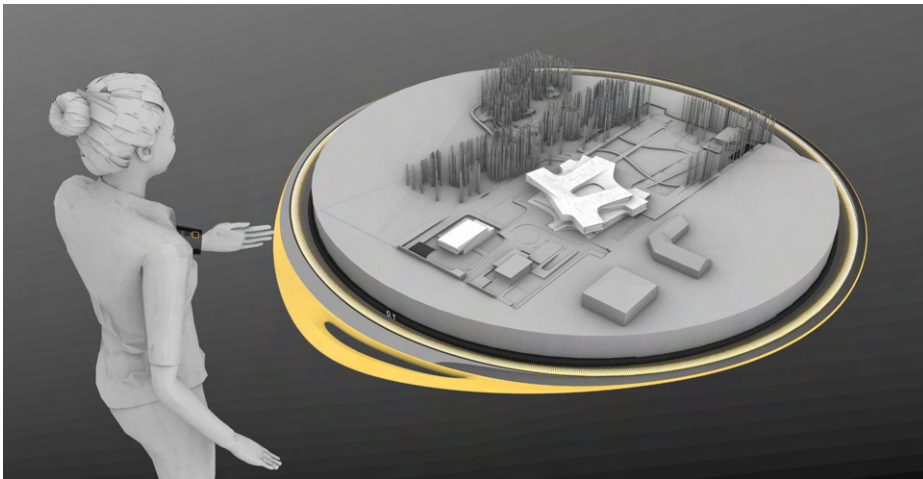


Figure 5.2.5j. Following the selection, the multi-tool is displayed in its operational form.

Work mode:

Depending on the scale in which user operates and whether the assets are within their reach, the tool adapts to maintain its usability (ergonomics):

- At the individual object level
- Miniature tool for remote operations on large objects outside of immediate reach and in 1:1(+) scale scenario

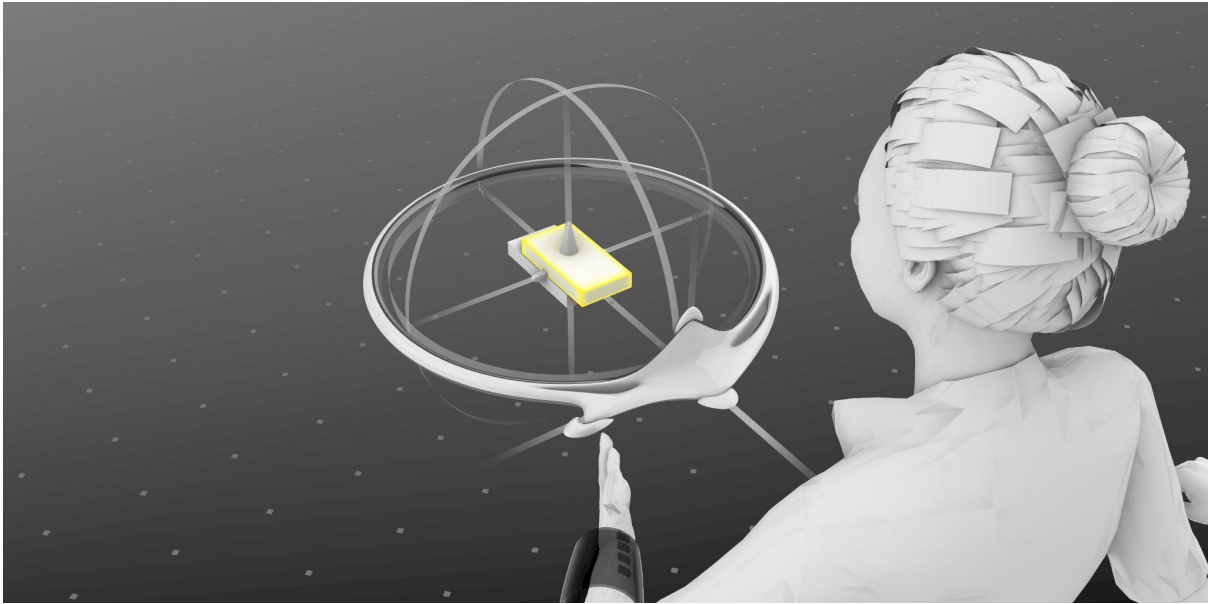


Figure 5.2.5k. Multi-tool in its transform operation mode, enabled at a selected scaled object.

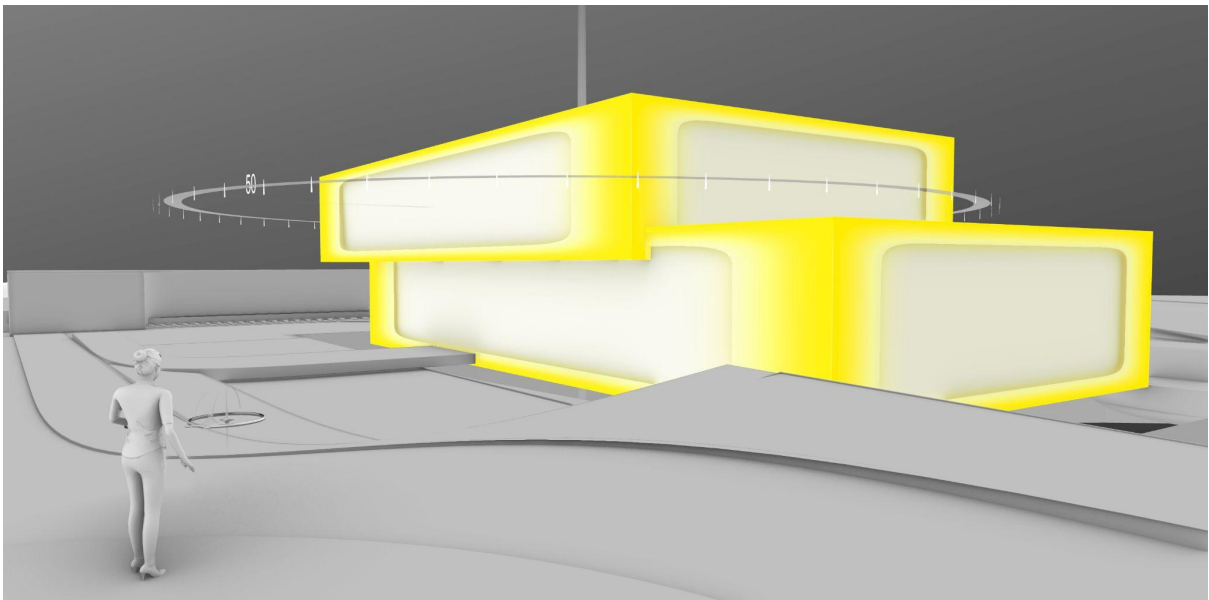


Figure 5.2.5l. In 1:1(+) scale, the user manipulates 3D assets remotely using a scaled multi-tool. In conjunction with the multi-tool spawned in front of the user, “ghosted” guides (visual aids) are displayed at the location of the 3D asset to provide visual feedback during object manipulation.

5.2.5.1 Content creation

The multi-tool would also cover the content creation aspect. We would like to work together with Mindesk to develop the UI and Multi-tool transformation. The content creation would need to be embedded in the entire workflow, from gesture-based to tool-based, to avoid the introduction of new and diverging UI strategies and concepts to the PrismArch UI ecosystem.

5.2.6 Clipping Tool

The proposed solution combines the initially proposed clipping tool, the mini-map navigation tool (see D1.1, Section 4.4) and adds a new function to assist teleportation into the 1:1 scale simulation space. The primary nature of the tool is a three-dimensional clipping volume for a design object – mainly, in this context, for a 3D model. The clipping volume allows the content to be ‘clipped’ ; it trims the 3D model content and reveals the contained interior geometry for a specific focus and purpose for the user.

The shape of the clipping volume can be adjusted by the user; or the user can instantiate a bespoke shape created using the production tool as a clipping shape. The clipping shape can be hidden entirely when the ghosted volume is no longer needed for modification of the clipping shape. This will make the trimmed content more accessible.

The displayed content can be set to different display modes and individual lighting conditions. The clipped edges of the content can be colouredised to show the sectioned parts.

One of the major uses for the clipping tool is to spawn multiple versions, iterations, or different design object content inside the design space; to either aid the production workflow by offering different scale or different disciplinary content as a reference to the user, or to showcase options during design evaluation.

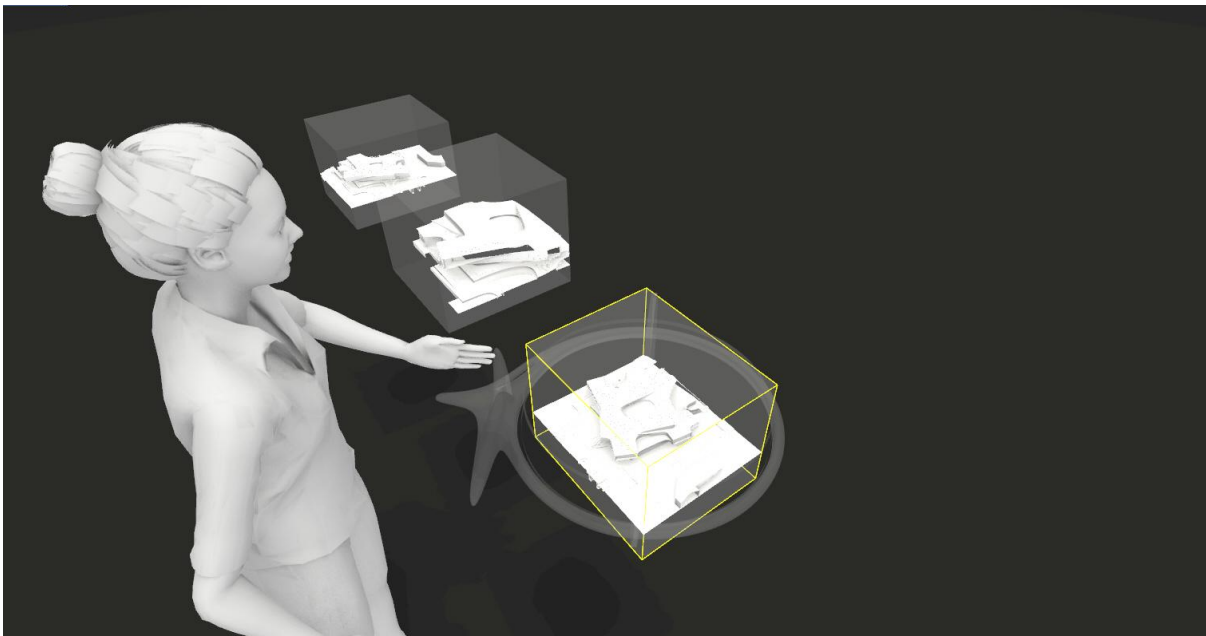


Fig. 5.2.6a showing a user creating a new clipping shape and clipped content. Other existing clipping tools are already in place showing content in different scale and display modes.

A unique function identified by us is the clipping tool in conjunction with the multitool in presentation mode, and a plinth or table. The clipping shape can be snapped and locked onto a plinth asset. This group of assets then can be scaled and moved in order to curate a presentation.

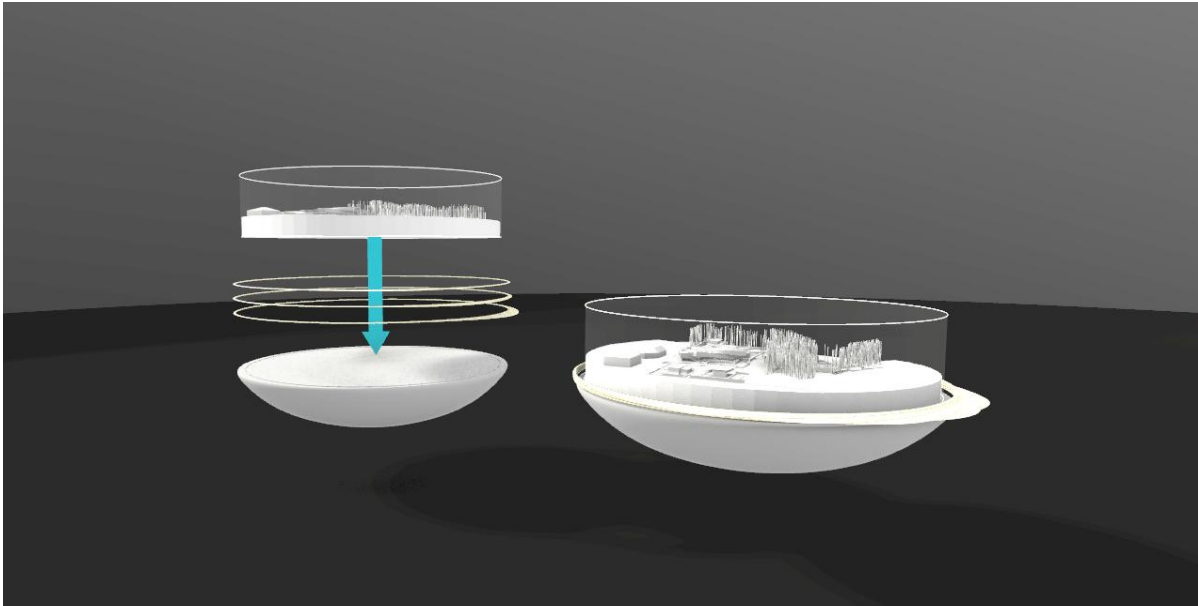


Fig 5.2.6b showing the cylindrical clipping tool being attached to the presentation plinth or table asset and with the added tabbed multitool for presentation purposes.

As the clipping tool can be snapped to objects, it can also be assigned to the user and be moved around by way of creating an overview model and navigation tool. The user avatar location and the locations of other participating avatars can be indicated inside the scaled navigation model (shown in ghosted display mode). The navigation model can also be used to determine teleportation into or when inside the full scale simulation. For the teleportation, the user can move a small avatar icon into or inside the overview model and press confirm to initiate the teleportation to the chosen location in the simulation.

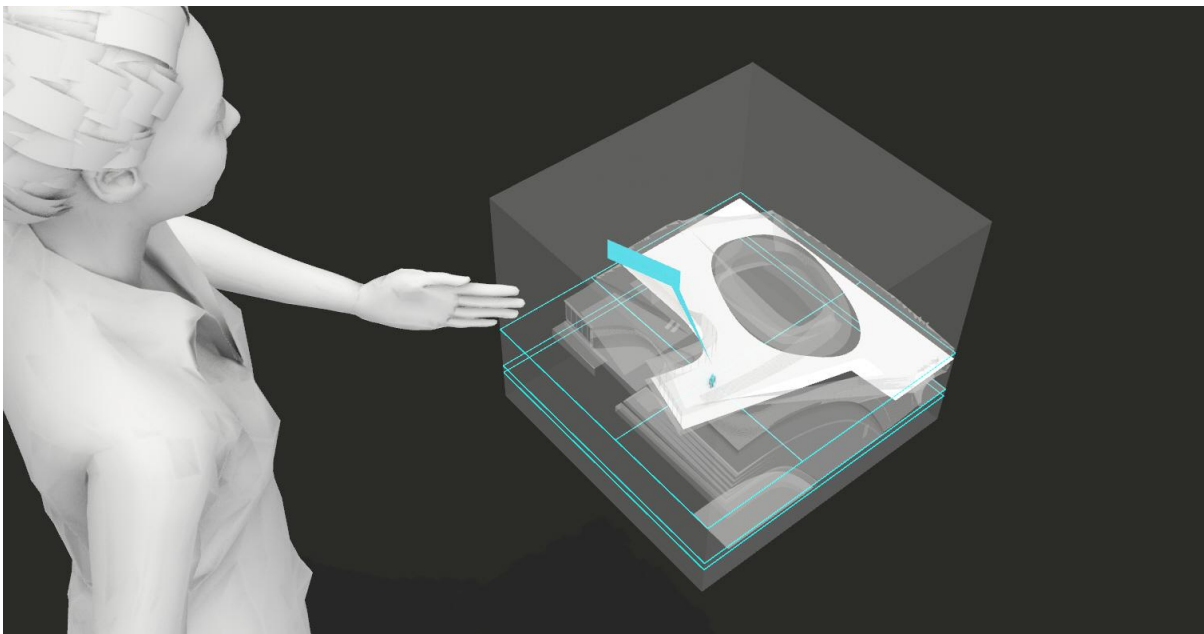


Fig. 5.2.6c showing the use of the clipping tool as minimap and teleportation locator. The content is ghosted and floor areas are highlighted in response to the user interaction, to allow the user to easily select where to drop in.

5.3 Environmental UI

Environment UI is a type of user interface that allows users to modify and load different Display and Work Sphere modes. User notifications, alerts and indicators are also part of the environment UI. These inform users of any changes to the current status of Work Sphere modes and their setups.

Display modes include:

- Rendered (photorealistic with dynamic lighting)
- Modelling (dark/ light theme grids - enabled automatically when Mindesk tool is enabled)
- Clay
- Ghosted
- Colour coded
- Discipline type
- Object type
- Custom
- Element and neighbour
- Mesh Analysis

Work Sphere modes include:

- *Meeting setup*
(preparation, presenting, commenting in meeting style environments with plinths)
- *Production setup*
(simulation, query, modelling in production style environments)

Display modes are top level view modes that can be applied to the entire environment regardless of the Work Sphere setup types. They are equivalent to a display filter.

Work Sphere modes involve user types and user-specific project information, as well as characteristics of the status of each Work Sphere. In all use cases, users can access all UI elements from the wrist UI.



Figure 5.3a: Example of Display mode set to 'Rendered' and Work Sphere mode set to 'Production'

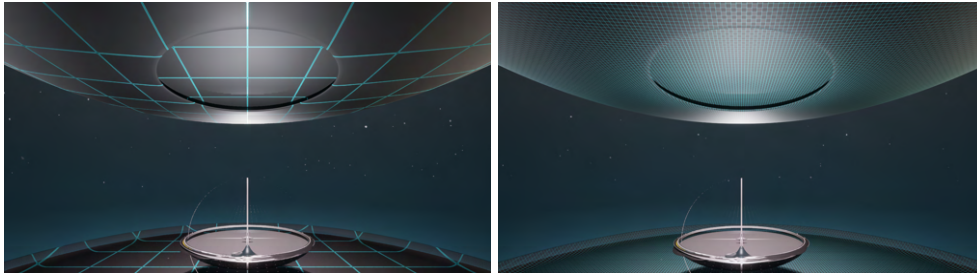


Figure 5.3b: Example of Display mode set to 'Modelling' 'dark theme' and Work Sphere mode set to 'Production'. Metre unit grid (left) Centimetre unit grid (right)

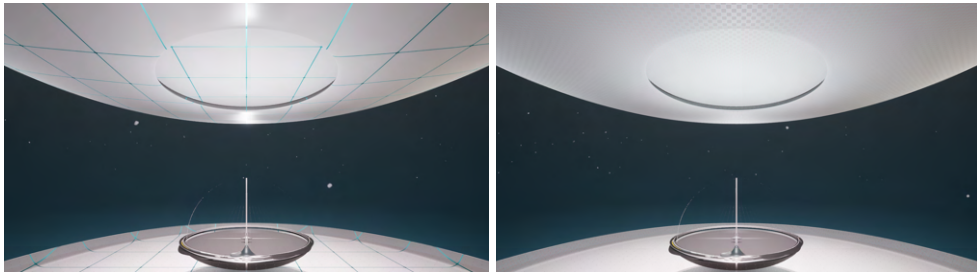


Figure 5.3c: Example of Display mode set to 'Modelling' 'light theme' and Work Sphere mode set to 'Production'. Metre unit grid (left) Centimetre unit grid (right)

PrismArch has two types of Environment UI: Dynamic Work Sphere Environment UI and Avatar Indicator Environment UI.

5.3.1 Dynamic scaling and transitions of environments

The *Dynamic Work Sphere Environment UI* is a resizable interior space that represents the concept of envelope for all loaded PrismArch class assets.

The components constituting the Dynamic Work Sphere Environment UI as follows:

- (a) Non interactable assets (ceiling, floor, furniture)
these elements can be unlocked, modified, or substituted if required
- (b) Interactable panels (curved presentation boards)

5.3.2 Avatar-related environmental indicators in co-presence environments

The *Avatar indicator Environment UI* is user action and role related markers that indicate specific conditions and status of users.

The components of Avatar Environment UI includes;

- (a) Name labels and profiles
- (b) Status (active, busy, in a meeting, be right back, offline)
- (c) Action type (speaking, hosting, presenting, modelling)
- (d) Sphere level
- (e) Work Sphere mode
- (f) Work Sphere setup type

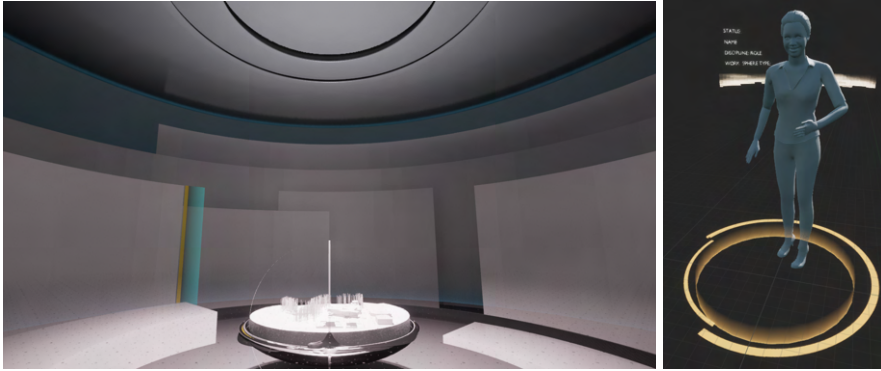


Figure 5.3.2a: Dynamic Work Sphere Environment UI (left) Avatar Indicator Environment UI (right)

Both of the above examples define and visualise specific conditions of the environments that the users are or will be perceiving/ experiencing. It is not only helpful for the users to be spatially aware of their existence inside the virtual environment but also helpful for communicating with others who are co-existing in the same virtual environment. For example, in a Meeting Work Sphere setup, each user can visually confirm if any of the attendees are in a private call or in an asynchronous modelling mode.

Replication scenario:

1. User A and B are in the same Work Sphere setup - Meeting and the Work Sphere mode is set to Default mode (List and Photorealistic)
2. User A is modelling. The user's Work Sphere view mode is set to modelling (no replication)
3. User B is reviewing the images that are on the prepared panels. The user's Work Sphere view mode is set to Default mode (replicated to all other attendees but User A)
4. mode is set to Default mode (replicated to all other attendees but User A)
5. User B can see User A is modelling because User A's Avatar indicates the modelling status.

5.3.2.1 Primary avatar-based environmental functions

- *Spatial sound*: speaker direction sound must have a spatial source inside VR, ideally related to the spatial location of the speaker.
- General notification sounds are universal in space, unless it's an object-specific sound (like a doorbell).
- *Avatar mouth movement* when someone is speaking
- *Eye tracking*, to indicate where the others are looking through pupil rotation (part of the facial expression),
- *Blinking* (can be automated)
- *Body posture*

5.3.3 Overall PrismArch Bookending and Spatial Sequence

From the experiential aspect of orientation, we wish to be able to interact with the design objects at full scale and also see it from a god-like perspective.

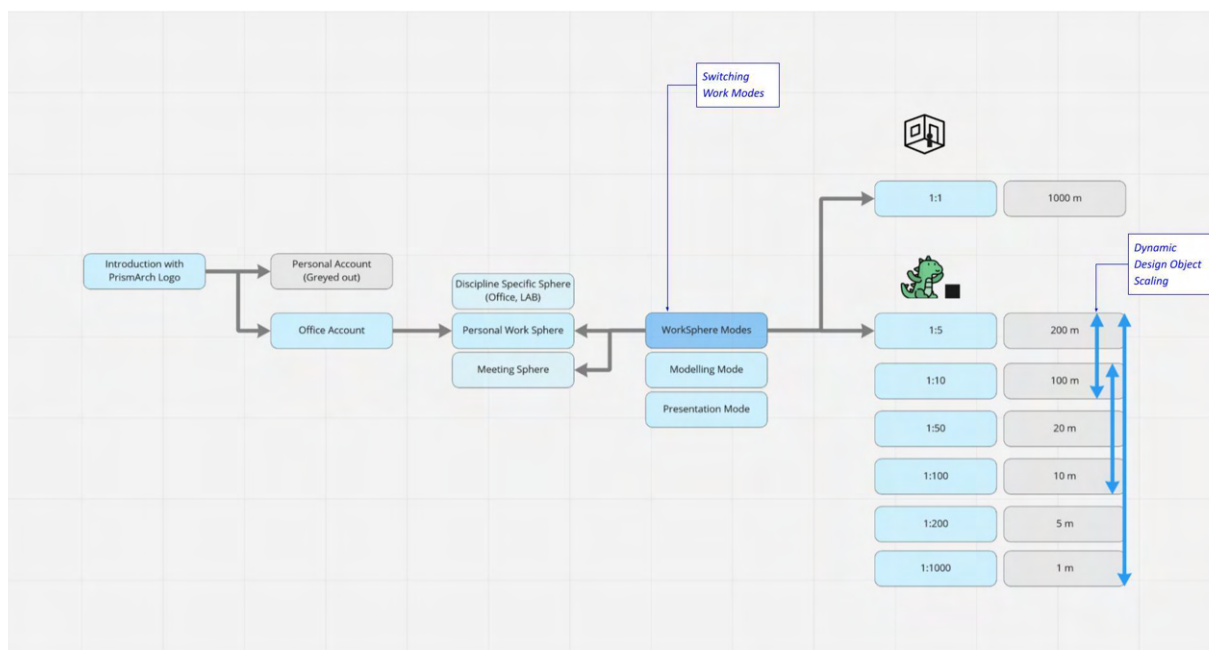


Figure 5.3.3a: PrismArch user operation narrative from platform entry to production

OVERALL EXPERIENCE V1 (FIRST PERSON VIEW)
SEQUENCE AND TRANSITIONS



Figure 5.3.3b: PrismArch user operation sequence and transitions from platform entry to production (See full-resolution images in the Appendix)

Transition effects such as loading/unloading, showing/hiding and animating assets can help stitch together the various stages of the user experience. This helps to create a seamless experience by applying the same design discourse and theme throughout navigation and operation of the application. It also helps users to identify and be aware of the application loading process.

The Top level description of PrismArch overall operation sequence can be found below:

1. User opens PrismArch application from PC (monitor)
2. User selects VR or Desktop mode from PC (monitor)
3. User puts a headset on
4. User sees Intro (lobby) space
5. User selects Private or Office mode (Private mode is greyed out and only office mode can be selected)
6. Transition effects/ animations are played and user is ported to company branded Personal Work Sphere with his/her discipline and title/role
7. (a) User continues to work in Personal Work Space for assigned tasks
(b) User organises or joins a meeting from Personal Work Sphere
8. (if 7(b) is selected) Transition effects/ animations are played and user is ported to prepared Collective Work Sphere with his/her discipline and title/role

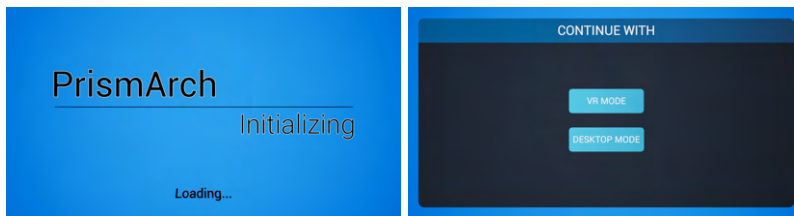


Figure 5.3.3c: User opens PrismArch application on PC and selects a VR or Desktop mode

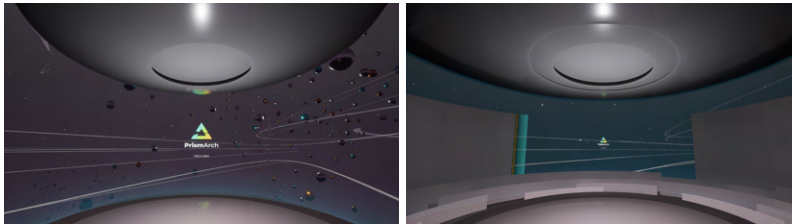


Figure 5.3.3d: User is being ported from Intro Space to Personal Work Sphere

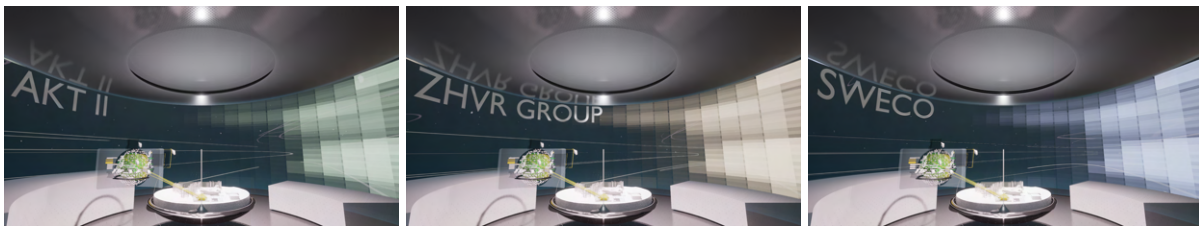


Figure 5.3.3e: User is ported to Office theme Personal Work Sphere

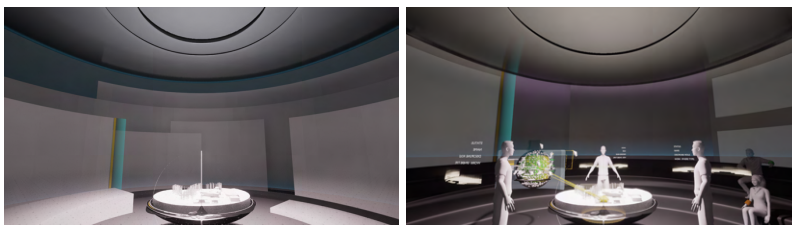


Figure 5.3.3f: User is being ported from Office Theme Personal Work Sphere to Collective Work Sphere

6. User Interface Operational Mock-Ups

ZHVR developed a number of conceptual UI mock-ups to further guide the development process and to illustrate the above developed principles for the UI creation of the Prism Arch platform. These mock ups attempt to synthesise the UI requirements on both, the functional-ergonomic as well as the aesthetical level. Given the status and aim of the PrismArch project these exemplary models were chosen to illustrate the maximum level of representation in regard to the gesture and tool based interaction for all currently envisaged functionalities of PrismArch - for the architectural and all other involved AEC disciplines. ZHVR is aware that in order to achieve the final UI solutions these conceptual designs will have to undergo thorough testing, but also receive critical additional functionalities and multiple alternative interaction 'pathways' to allow for individual user preferences. The described mock-ups are therefore an initial set of guidelines for the immediate development and further future UI design.

6.1 Selecting objects

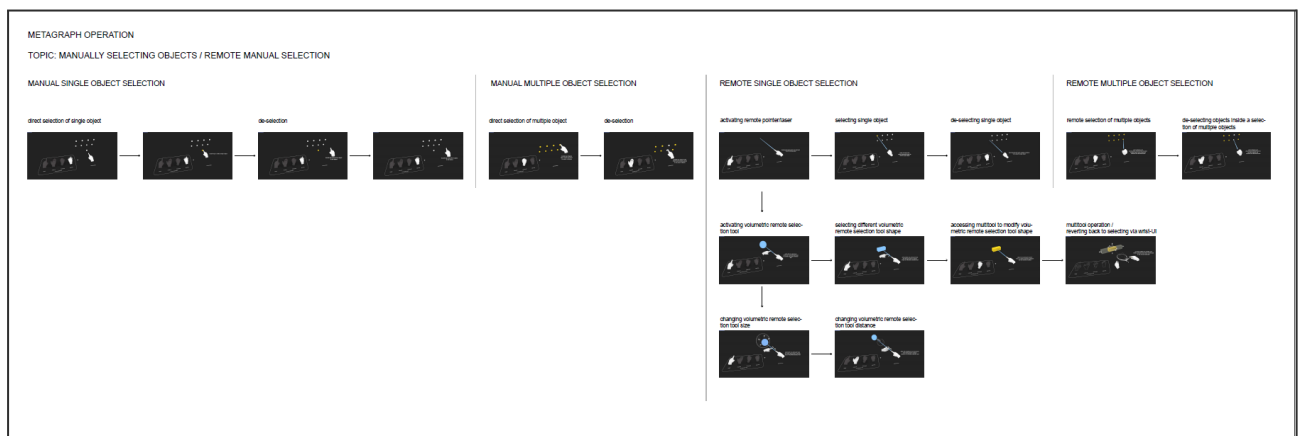


Figure 6.1a. Operational scenario flowchart describing methods to select one or multiple objects in PA. (Full-resolution image is in the Appendix).

6.2 IDQS - Creating a new project

Using the Globe tool Project Director selects a geo location of a new project and creates a new project Sphere. The Project Director inputs the project name: 'Villa Project' Using the preview monitor and the keyboard. A new node is created and displayed on the Globe.

1. IDQS: Starting a new project and creating project data

Using the Globe tool Project Director selects a geo location for a new 'Villa Project'

New project Sphere is created

Using user monitor and the keyboard, Project Director assigns a name: 'Villa Project' to the new project sphere

It is now visible on the Globe



Figure 6.2a. Operational scenario describing steps to create a new project: selecting project location using Globe tool

The directors selects files to be loaded into PA e.g.: an program excel sheet (excel) and site photos (jpeg) received from the client. Once the selection is confirmed, assets are loaded into PA.

The director organises the uploaded information. He/she uses a multi selection tool to select files and assign names. He/she uses a multi selection tool to select individual nodes to create a new "Brief" sphere containing uploaded project information. It is a part of the project data and nodes containing assets brought into the PA environment are displayed on a single orbit representing time value.

1. IDQS: Starting a new project and creating project data

Loading multiple file types from the browser projected on the wrist monitor (t=1)
(e.g. Director adding an program excel sheet (excel) and site photos (jpeg) received from the clients)

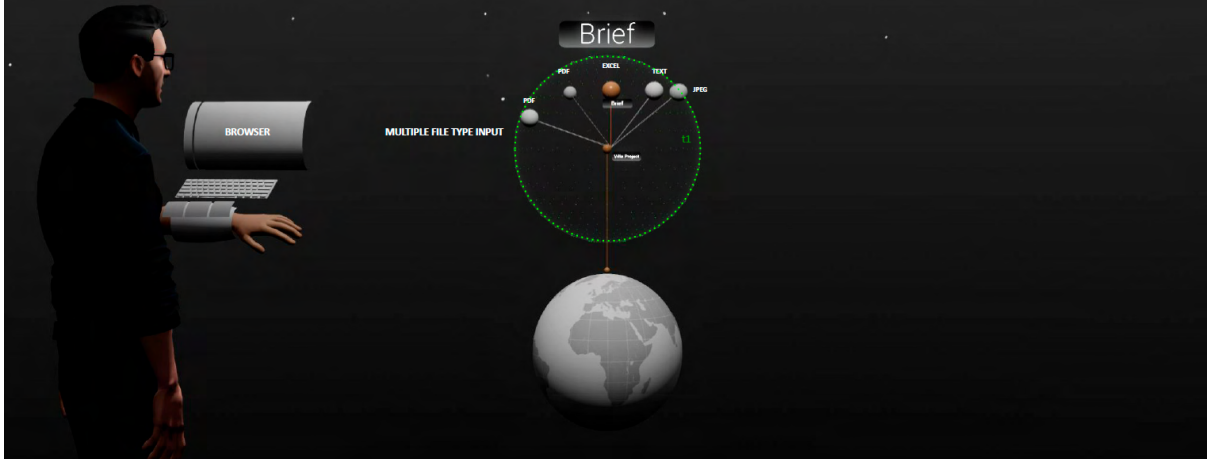


Figure 6.2b. Operational scenario describing steps to create a new project: load an external project data into PA environment and preview a Metagraph. Nodes of the Metagraph contain different types of information in various file formats. We could also imagine having different node shapes to visually encode different data formats (not shown on the image)

As he/she adds more information over time, new orbits appear, displaying nodes added at different times. The graph can be minimised or maximised using the time increments along the matagraph's axis.

5. Sphering/Tagging

Using his wrist monitor and keyboard he inputs a name of a new sphere

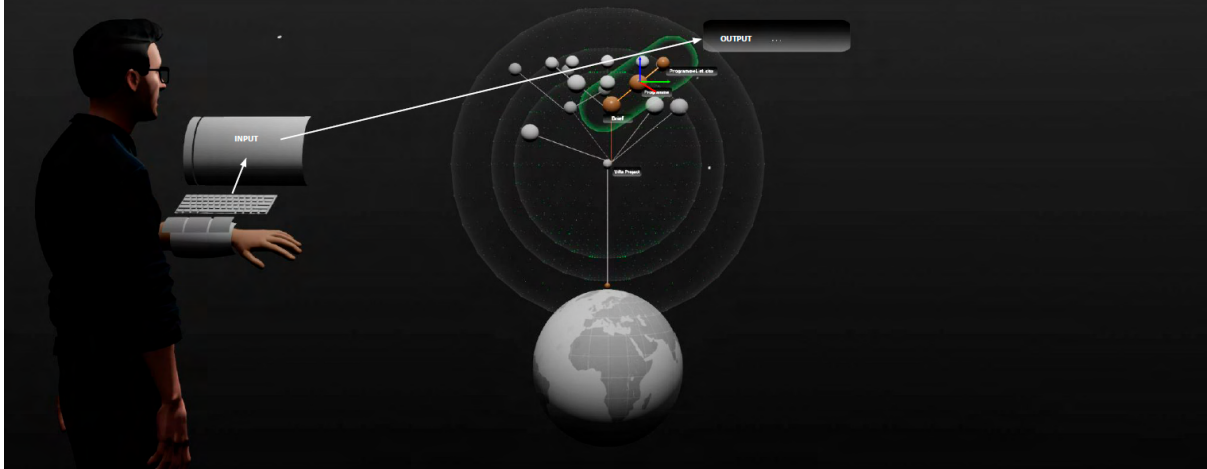


Figure 6.2c. Operational scenario describing interaction with the Metagraph to select multiple nodes, sphere and tag them to create a new node with a new name.

5. Sphering/Tagging

The name tag appears next to the highlight

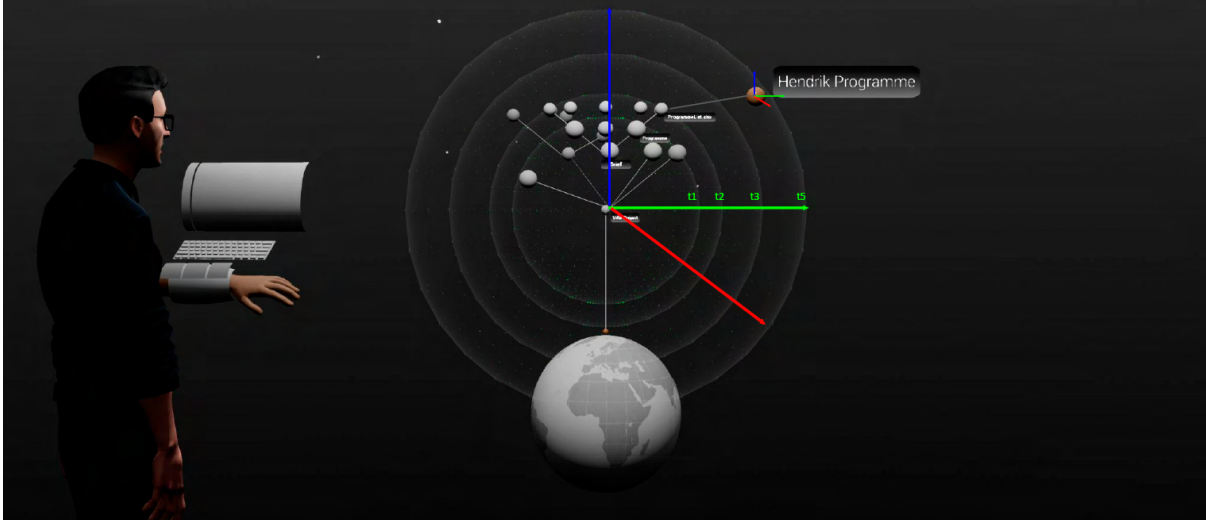


Figure 6.2d. Operational scenario describing interaction with the Metagraph to create a new node with a new name located at the new orbit. Orbits are a visual aid displaying time value.

6.3 IDQS - Metagraph operation mock-ups

The operational mock-ups in this section demonstrate the query process and interaction with the metagraph nodes. What we are showing in this section is exemplary only; full-fledged research will be needed to identify the possibility and functionality of alternative ordering systems and transformation that the metagraph would require to meet every need for interacting with data.

6.3.1 Query process mock-up

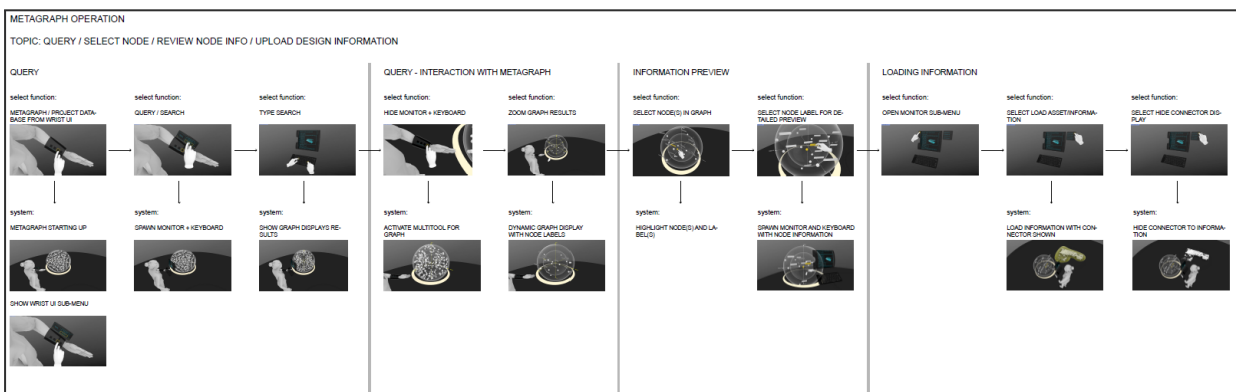


Figure 6.3.1a. Operational scenario flowchart describing steps to query information via IDQS, select nodes of the metagraph, review information about the node and load a design information into the PA environment. (Full-resolution image is in the Appendix)

6.3.2 Interaction with metagraph nodes and nested nodes

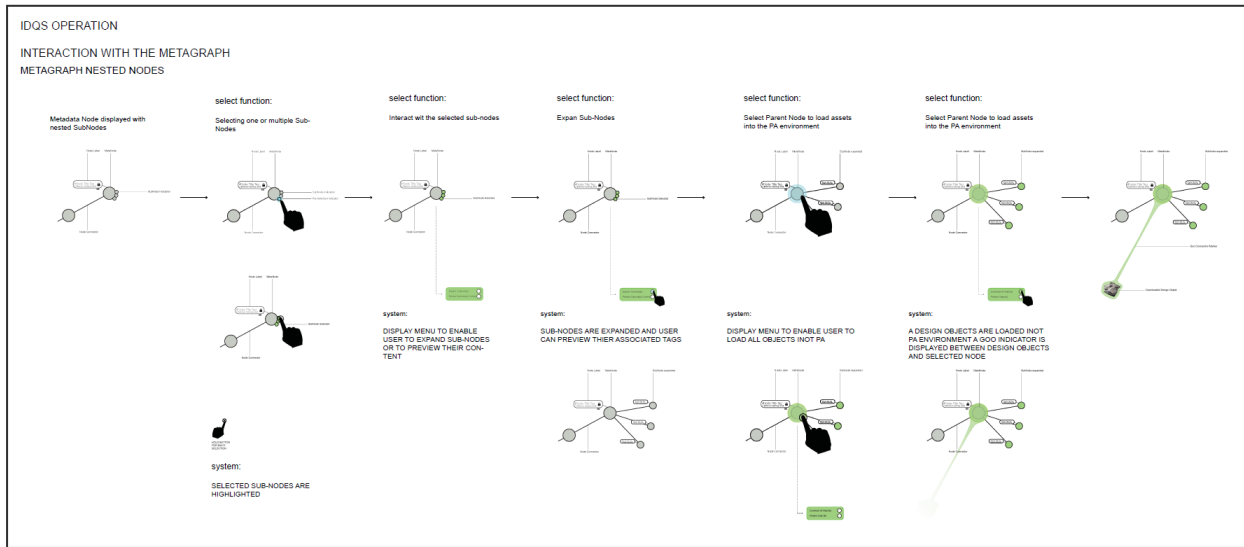


Figure 6.3.2a. Operational scenario flowchart describing steps to interact with Metagraph nodes. (Full-resolution image is in Appendix)

The nested nodes are only visible and interactable when the metagraph is in a specific level of resolution.

6.4 Multi-tool operation: transform of an object (in production mode)

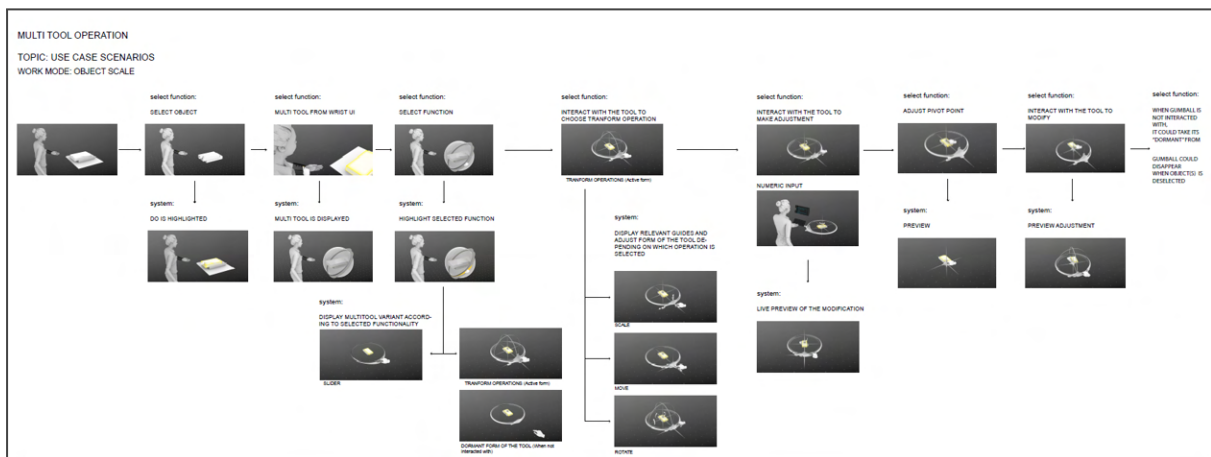


Figure 6.4a. Operational scenario flowchart describing interaction with the Multi-tool to manipulate objects in PA. (Full-resolution image is in the Appendix)

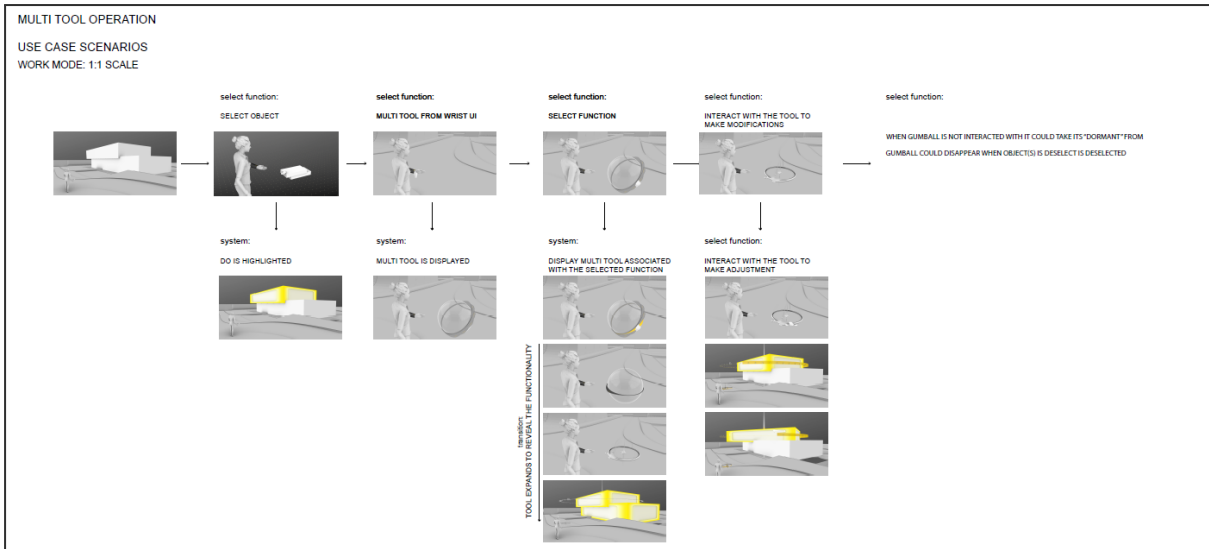


Figure 6.4b. Operational scenario flowchart describing interaction with the Multi-tool to manipulate objects remotely in 1:1 (+) scale. (Full-resolution image is in the Appendix)

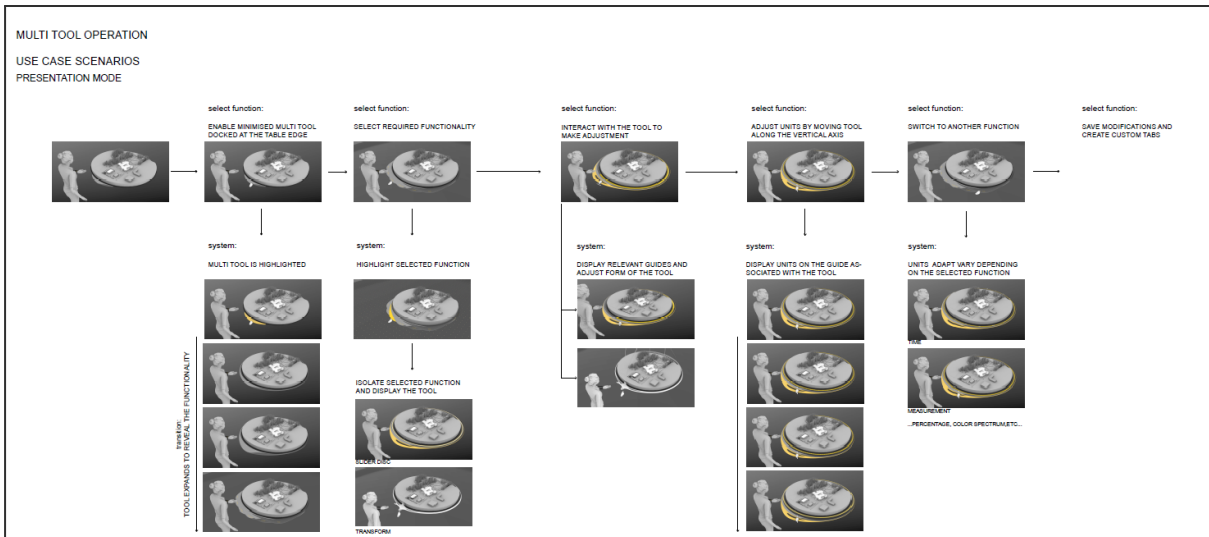


Figure 6.4c. Operational scenario flowchart describing interaction with the Multi-tool to manipulate objects in presentation mode. (Full-resolution image is in the Appendix)

6.5 Environmental Operation (mockup)

Dynamic Work Sphere Environment UI

Dynamic Work Sphere Environment UI executes two types of events; 'Switching display modes' and 'Resizing and modifying Work Sphere modes'.

Event 1: Switching display modes

Situation 1: {Switching display modes}

Operation type 1: user manually selects a display mode option

Interaction Input: wrist UI

Scenario: user selects a display mode from wrist UI monitor.

Examples of display mode options are:

- Rendered (photorealistic + lighting)
- Modelling (dark/ light theme grids - enabled automatically when Mindesk tool is enabled)
- Clay
- Ghosted
- Colour coded (Discipline type, Object type)
- Custom (e.g. Element and neighbour, Mesh Analysis)

Operation type 2: user enables Mindesk modelling tool

Interaction Input: wrist UI

Scenario: user enables Mindesk modelling tool and new Work Sphere environment materials (grid texture) or post-process layer are loaded. The grid scale is in ratio to the loaded and interacting scaled Design Objects.

Event 2: Resizing and Modifying Work Sphere setups

Work Sphere setups can be resized and modified under the situation where users view scaled Design Objects. There is an exception scenario where users select to view real-scale Design Objects.

Situation 1: {Viewing scaled Design Objects}

Operation type 1.a: user manually sets new Work Sphere size value

Interaction Input: wrist UI and keyboard to set the value

Scenario: system runs one time animation starting from current Work Sphere size to the set Work Sphere size value. (play one time animation)

Operation type 1.b: system produces and updates a bounding box for all loaded assets within current user Work Sphere setup. Work Sphere size is resized automatically according to the total volume of the loaded assets, which cannot exceed the size of Work Sphere interiors.

Interaction Input: wrist UI to enable/ disable the automation

Scenario: as the user scales up/down design objects, current user Work Sphere size updates under a specific condition.

Operation type 2: user manually modifies and customises specific Dynamic Work Sphere Environment UI component by directly interacting with the components.

Interaction Input: laser selection, grab selection

Scenario: The user is already happy with the current Work Sphere setup but would like to rearrange the components such as panel locations, panel content size, changing background.

In operation type 1.b, the automatic Work Sphere resizing is based on the total volume of all the assets loaded within current user Personal or Collective Work Sphere, including;

- Scale of Design Objects
- Number of presentation plinth

- Number of meeting attendees

Users should not immediately receive the resizing feedback. Under certain conditions, the auto resize functionality should be locked.

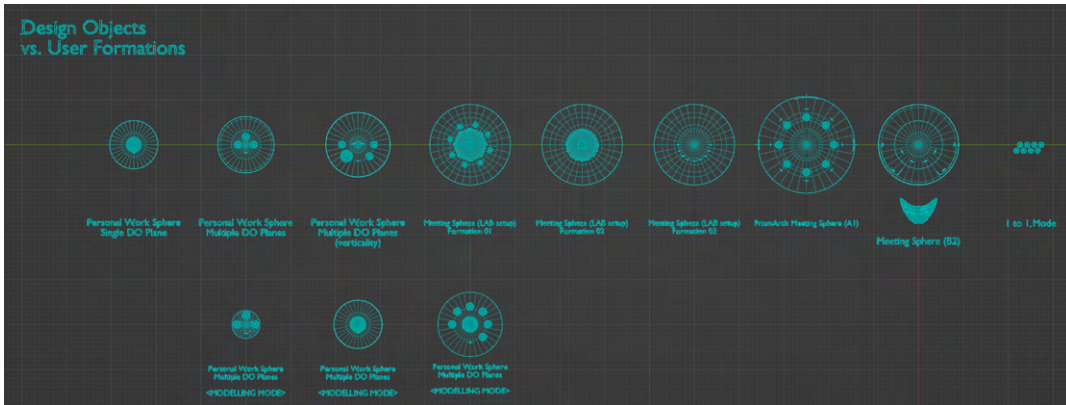


Figure 6.5a: **Example of Operation type 1.b.** Formation variations of users, Design Objects and Design Object plinths (top view). (Larger image is in the Appendix)

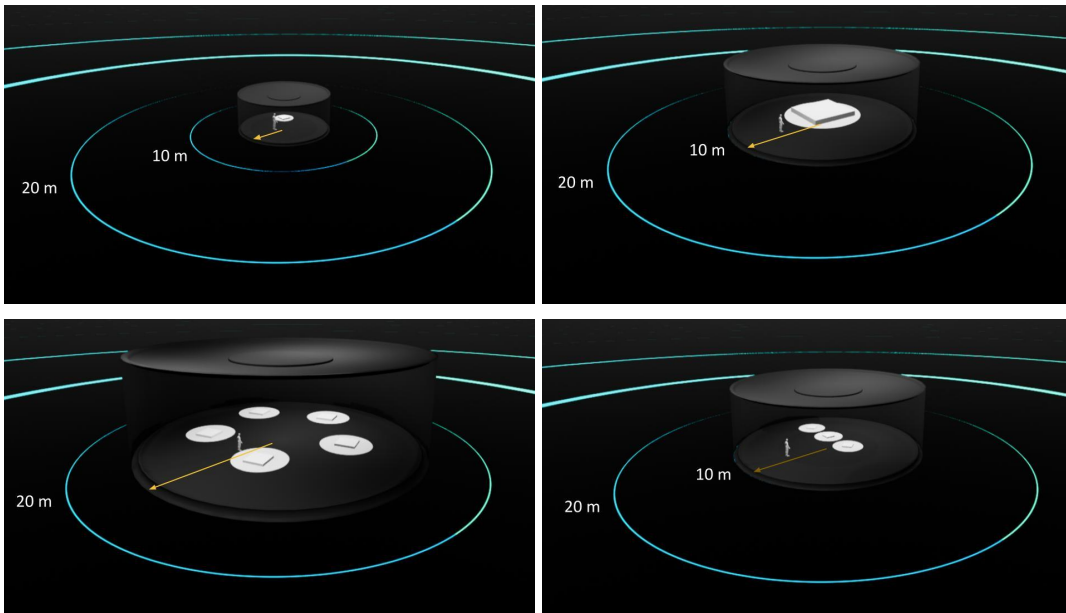


Figure 6.5b: **Example of Operation type 1.b.** Formation variations of users, Design Objects and Design Object plinths (perspective view) and Work Sphere size corresponding to the total volume of loaded assets inside the current Work Sphere interiors.

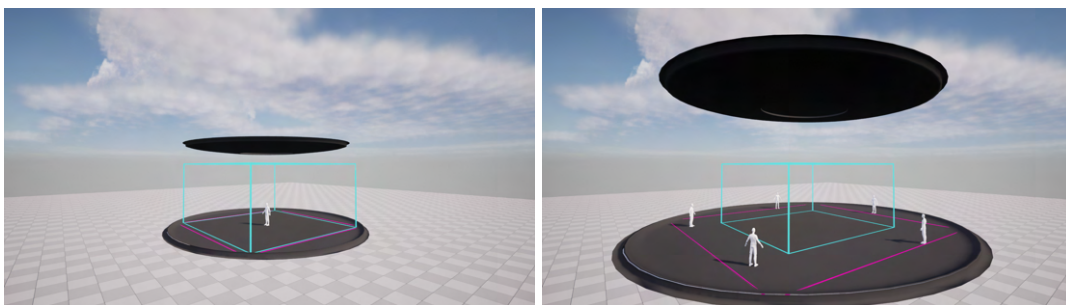


Figure 6.5c: **Example of Operation type 1.b.** Early mock-up of auto-resizing boundary. Blue line shows an initial boundary box size, pink line shows updated boundary box size.

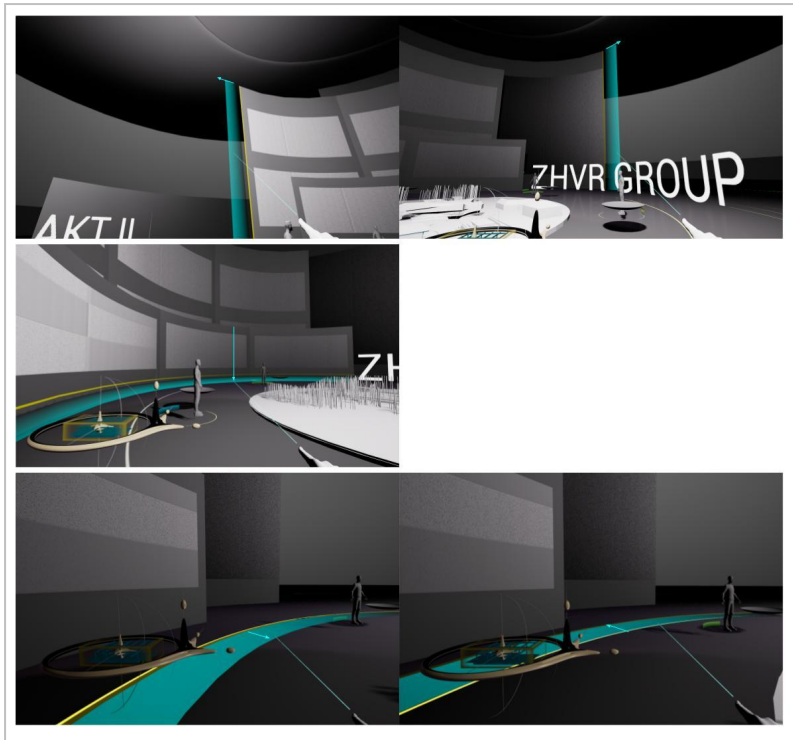


Figure 6.5d: **Example of Operation type 2.**
Manual adjustment of Environment UI component

Situation 2/ Exception: {Viewing real-scale (1:1 scale) Design Objects}

Operation type 3: user loads real scale model and the Work Sphere becomes temporality hidden.

Interaction Input: wrist UI and mini-me pin

Scenario: user spawns a mini-me pin from the wrist UI and places it in the desired location on the target scaled model

1. User selects a mini-me drop pin icon
2. A miniature avatar asset is spawned in front of the user
3. User selects the miniature avatar asset and places it on the scaled design object.
4. User can then adjust the miniature avatar asset location to where they would like to travel
5. Confirm to port the user
6. User camera blinks
7. User sees the real scale model at previously selected location

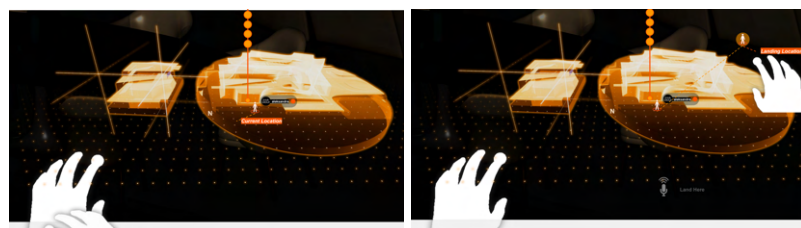


Figure 6.5e: User interacting with a mini-me drop pin icon to select his/her

landing location inside the current Work Sphere setups to view real-scale Design Objects. (Full-resolution image is in the Appendix)

For the Collective scenario of the exceptional operation of viewing real-scale Design Objects, meeting attendees can be treated as a group while preparing for landing. After all levels are loaded, the landing orientation as a group will be released and users can freely move around individually. This avoids users accidentally bumping into each other during the transitions.

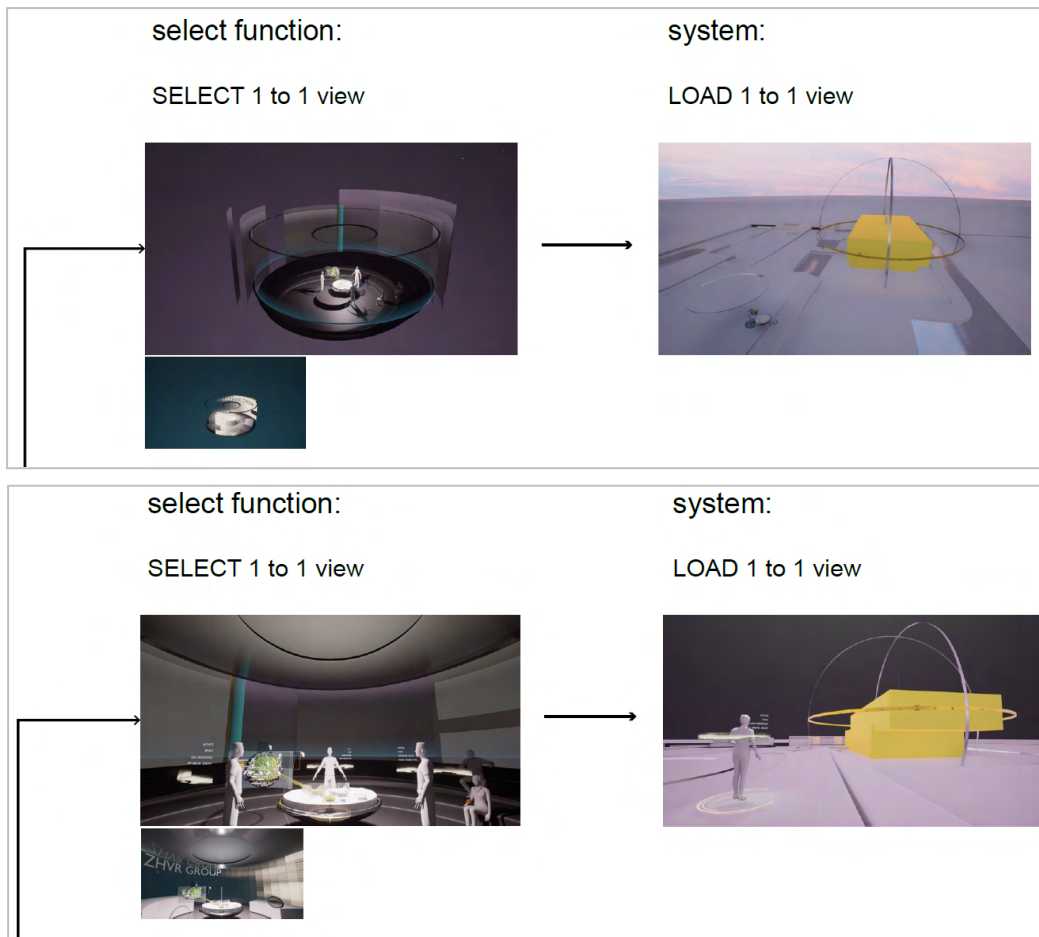
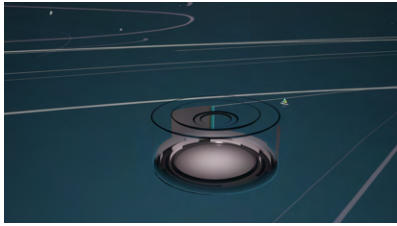
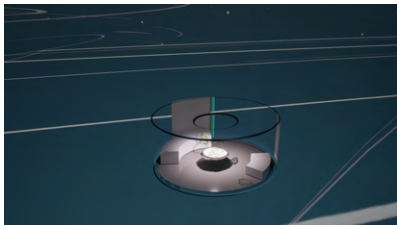


Figure 6.5f: **Example of Operation type 3 - Exception: Viewing real-scale Design Objects.** First person view of user transition from Collective Work Sphere to 1 to 1 scale view (left) Bird's Eye view of user transition from Collective Work Sphere to 1 to 1 scale view (right)



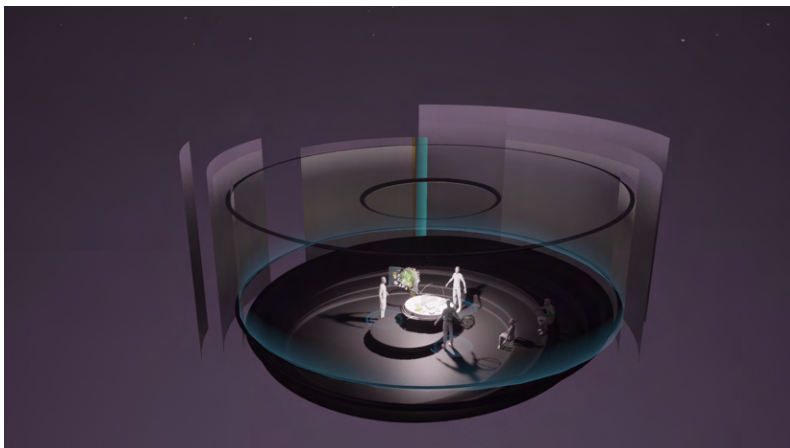
(1) Loading Personal Work Sphere assets.



(2) Personal Work Sphere assets are loaded.

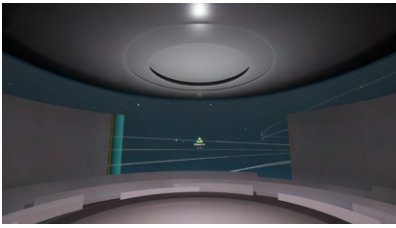


(3) Loading Collective Work Sphere assets.

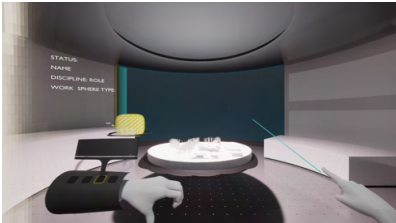


(4) Collective Work Sphere assets are loaded.

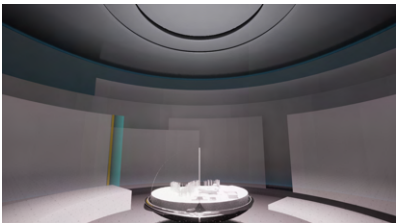
Figure 6.5g: **Example of Operation type 1.b.** Sequence of transition effects from Personal Work Sphere to Collective Work Sphere setups (Bird's eye view)



(1) Loading Personal Work Sphere assets.



(2) Personal Work Sphere assets are loaded.



(3) Loading Collective Work Sphere assets.



(4) Collective Work Sphere assets are loaded.

Figure 6.5h: **Example of Operation type 1.b.** Sequence of transition effects from Personal Work Sphere to Collective Work Sphere setups (First person view)

7. Summary and conclusion

In this deliverable, the requirements for a unified collaborative environment posed in D1.1 and D1.2 were further elaborated and consolidated into a holistic vision of the functionality of PrismArch as a multidisciplinary simulation-based AEC platform. This is the third and final version of the report, and final deliverable of the Work Package.

We have aimed at conceptualising with mockups how the VR experience should be layered to facilitate the needs of each design discipline, and have emphasised the points of intersections across disciplines from a cross-disciplinary view.

While many new VR softwares have emerged for the AEC industry since the start of the research, few handle the legal and contractual aspects of multidisciplinary data (address IP) or help maintain a golden thread record of the project. This document is an academic attempt to describe and create the conditions for such a multidisciplinary space. However, many of the ideas presented here will need to be refined through testing, and further evolved through AEC professional uptake and usage.

In our view, the platform will be based on proprietary core functionality, opening up to a commercial API-based plug-ins for extending this functionality by creating a marketplace for API development. By opening to the API-based development solution and allowing for the injection of additional functionality, we stand to achieve our multidisciplinary aims quicker than by attempting to do the entire development in-house. In this document, we have tried to pinpoint what functionality and tools will be required at the outset, and what can be left to the API development.

As a next step, these requirements will be evaluated by the technical partners to see what aspects can be incorporated in the MVP.

References

Section 1, Usability and Performance Guidelines

Pressman, Andy. *Architectural Graphic Standards*. Hoboken, N.J: John Wiley & Sons, 2007.

Neufert, Ernst, Peter Neufert, Bousmaha Baiche, and Nicholas Walliman. *Architects' Data*. Oxford: Blackwell Science, 2000.

BSI Standards Catalogue. London: The Institution, 1987.

Section 2, Congregated multi-disciplinary design and simulation

“Golden Thread in UK Legislation”, from UK Government Policy Paper *Building Regulations Advisory Committee: golden thread report*, 21 July 2021.

<https://www.gov.uk/government/publications/building-regulations-advisory-committee-golden-thread-report/building-regulations-advisory-committee-golden-thread-report>. Accessed 30 March, 2022.

Section 3, Information Display and Query System

Eleanor Konik, *It's Not Just a Pretty Gimmick: In Defense of Obsidian's Graph View*, 2021.

<https://www.obsidianroundup.org/its-not-just-a-pretty-gimmick-in-defense-of-obsidians-graph-view/>. Accessed 22 July, 2022.

Pirch, Sebastian & Mueller, Felix & Iofinova, Eugenia & Pazmandi, Julia & Hütter, Christiane & Chiettoni, Martin & Sin, Celine & Boztug, Kaan & Podkosova, Iana & Kaufmann, Hannes & Menche, Jörg. (2021). *The VRNetzer platform enables interactive network analysis in Virtual Reality*. Nature Communications. 12. 10.1038/s41467-021-22570-w.

https://www.researchgate.net/publication/351080869_The_VRNetzer_platform_enables_interactive_network_analysis_in_Virtual_Reality. Accessed 25 July, 2022.

Section 5, Proposal for the VR User Interface: classification and layout

LaViola, Joseph J. & Kruijff, Ernst & McMahan, Ryan P. & Bowman, Doug A. & Poupyrev, Ivan. *3d User Interfaces : Theory and Practice*. Second ed. Boston: Addison-Wesley, 2017.

APPENDIX

APPENDIX A: PrismArch Documents

Deliverable No D1.1 (2021, March). "Report on current limitations of AEC software tools, leading to user and functional requirements of PrismArch". <https://prismarch-h2020.eu/download/472>

Deliverable No D3.1 (2021, May). "Report on cognitive issues in VR-aided design environments". <https://prismarch-h2020.eu/download/480>

Deliverable No D5.1 (2021, July) "First version of architectural design and integration protocol". <https://prismarch-h2020.eu/download/482>

Deliverable No D6.1 (2021, July). "Define the architectural projects and usage scenarios for demonstration and evaluation". <https://prismarch-h2020.eu/download/483>

Deliverable No D1.2 (2021, Sept.) "Elaborated report of cross-discipline principles-rules-constraints, and interfaces definition for cross-disciplinary and multi-simulation perspectives in VR". <https://prismarch-h2020.eu/download/485>

APPENDIX B: Images

Figure	Reference
1a	"Members". Metaverse Standards Forum, 2022. https://metaverse-standards.org/members/ Accessed June 2022.
2a	Image by ZHVR Group, Zaha Hadid Architects, 2022.
2b	Image by ZHVR Group, Zaha Hadid Architects, 2022.
2c	Image by ZHVR Group, Zaha Hadid Architects, 2022.
2.1a	Image by ZHVR Group, Zaha Hadid Architects, 2021.
2.1b	Image by ZHVR Group, Zaha Hadid Architects, 2022.
2.1c	PrismArch Schemas. Image by AKT-II, 2022.
2.1d	'Prism_Signature' Specifications, <i>PrismArch_D1.2_v1.0</i> , 2022., p86. Image by AKT-II.
2.2a	Image by ZHVR Group, Zaha Hadid Architects, 2022.
2.2b	PrismArch Speckle Tags, London Plenary Presentation. Image by AKT-II, 2022.
2.2c-h	Images by ZHVR Group, Zaha Hadid Architects, 2022.
2.3a	Project initialization and management, <i>PrismArch_D5.1</i> , 2021, Image by CERTH. p12, 202, revised for D1.3 by ZHVR, 2022
2.5a-b	Image by ZHVR Group, Zaha Hadid Architects, 2022.
2.5.1a-d	Image by ZHVR Group, Zaha Hadid Architects, 2022.
3.2a	Octopus, Image by Robert Vier, 2022. https://www.grasshopper3d.com/group/octopus?overrideMobileRedirect=1
3.2b	Softspace Prototype01, Image by Softspace, 2022. https://substack.soft.space/p/softspace-ar-prototype01?s=r&utm_campaign=post&utm_medi

	um=web
3.2c	Obsidian Graph View, <i>It's Not Just a Pretty Gimmick: In Defense of Obsidian's Graph View</i> , Image by Eleanor Konik, 2021. https://www.obsidianroundup.org/its-not-just-a-pretty-gimmick-in-defense-of-obsidians-graph-view/
3.2d	Force-Directed Graph, Image by Vasco Asturiano, 2021. https://bl.ocks.org/vasturiano/02affe306ce445e423f992faeea13521
3.2.1a-c	Images by ZHVR Group, Zaha Hadid Architects, 2022.
3.2.1d-e	Pirch, Sebastian & Mueller, Felix & Iofinova, Eugenia & Pazmandi, Julia & Hütter, Christiane & Chiettoni, Martin & Sin, Celine & Boztug, Kaan & Podkosova, Iana & Kaufmann, Hannes & Menche, Jörg, 2021. <i>The VRNetzer platform enables interactive network analysis in Virtual Reality</i> , <i>Nature Communications</i> , 12. 10.1038/s41467-021-22570-w. Viewed on Researchgate. https://www.researchgate.net/figure/Immersive-network-exploration-a-Green-screen-composition-showing-a-user-immersed-in-the_fig2_351080869 . Accessed on March 2022.
4a-b	Images by ZHVR Group, Zaha Hadid Architects, 2022. Full resolution images can be found in PrismArch_D1.3_Chapter4_Appendix.pdf. https://drive.google.com/file/d/1y_ymRPsG5q46Mh6kozGbiR3Z-_Keld_Z/view?usp=sharing
4.3.1a	Diagram illustrating how PrismArch could be incorporated into structural engineer workflows, PrismArch_D1.2_v1.0, 2022. Image by AKT-II.
4.3.1b	D1.2 figures 3.6.3c and d showing the representation of bending moment, and 3D load points, PrismArch_D1.2_v1.0, 2022. Image by AKT-II.
4.3.1c	D1.2 figures 3.6.3i and j, showing the representations of structural supports, PrismArch_D1.2_v1.0, 2022. Image by AKT-II.
4.3.1d	Interactive visualisation of a beam's deformation, PrismArch_D1.2_v1.0, 2022. Image by AKT-II.
4.3.1e	Groupings and priorities after card sorting, PrismArch_D1.2_v1.0, 2022. Image by AKT-II.
4.3.1f	Graphically selecting elements, and automatically isolating their topological neighbours, PrismArch_D1.2_v1.0, 2022. Image by AKT-II.
4.3.2a	Different levels of information of a transformer as the project matures, PrismArch_D1.2_v1.0, 2022. https://toolkit.thenbs.com/uniclass/ss
5a	Image by ZHVR Group, Zaha Hadid Architects, 2022.
5b-c	Images by ZHVR Group, Zaha Hadid Architects, 2022. Full resolution image can be found in PrismArch_D1.3_Chapter5_Appendix.pdf. https://drive.google.com/file/d/1KDgYszsJHmZi4XsDjvpHThrNQU0DSQx/view?usp=sharing
5.1a	Image by ZHVR Group, Zaha Hadid Architects, 2022. Full resolution image can be found in PrismArch_D1.3_Chapter5_Appendix.pdf. https://drive.google.com/file/d/1KDgYszsJHmZi4XsDjvpHThrNQU0DSQx/view?usp=sharing
5.2.1.1a-b	Screenshots from current PrismArch application, 2022.
5.2.1.2a	Image by ZHVR Group, Zaha Hadid Architects, 2022.
5.2.1.3a	Image by ZHVR Group, Zaha Hadid Architects, 2022.
5.2.1.4a	Image by ZHVR Group, Zaha Hadid Architects, 2022.
5.2.2a-b	Images by ZHVR Group, Zaha Hadid Architects, 2022.

5.2.3a-c	Images by ZHVR Group, Zaha Hadid Architects, 2022.
5.2.4a	Image by ZHVR Group, Zaha Hadid Architects, 2022.
5.2.4.1a-b	Images by ZHVR Group, Zaha Hadid Architects, 2022.
5.2.4.2a-b	Images by ZHVR Group, Zaha Hadid Architects, 2022.
5.2.4.3a-d	Images by ZHVR Group, Zaha Hadid Architects, 2022.
5.2.4.4a	Image by ZHVR Group, Zaha Hadid Architects, 2022.
5.2.5a-b	Images by ZHVR Group, Zaha Hadid Architects, 2022.
5.2.5c-d	Images by ZHVR Group, Zaha Hadid Architects, 2022. Full resolution image can be found in PrismArch_D1.3_Chapter5_Appendix.pdf. https://drive.google.com/file/d/1KDgYszsJHmZl4XsDjvpHThrNQU0DSQx/view?usp=sharing
5.2.5e-l	Images by ZHVR Group, Zaha Hadid Architects, 2022.
5.2.6a-c	Images by ZHVR Group, Zaha Hadid Architects, 2022.
5.3a-c	Images by ZHVR Group, Zaha Hadid Architects, 2022.
5.3.2a	Images by ZHVR Group, Zaha Hadid Architects, 2022.
5.3.3a	Image by ZHVR Group, Zaha Hadid Architects, 2022.
5.3.3b	Image by ZHVR Group, Zaha Hadid Architects, 2022. Full resolution image can be found in PrismArch_D1.3_Chapter5_Appendix.pdf. https://drive.google.com/file/d/1KDgYszsJHmZl4XsDjvpHThrNQU0DSQx/view?usp=sharing
5.3.3c-f	Images by ZHVR Group, Zaha Hadid Architects, 2022.
6.1a	Image by ZHVR Group, Zaha Hadid Architects, 2022. Full resolution image can be found in PrismArch_D1.3_Chapter6_Appendix.pdf. https://drive.google.com/file/d/1J8PMin_wZtsXPGIrbQPX1t_fr7WGfq-L/view?usp=sharing

APPENDIX C: Resources

C1. Section 2 Resources

The Golden Thread legislation has evolved since the start of this research project, when it was a proposal in the Draft Safety Bill. It has since passed into law, and set out in Legislation across several tiers, as shown in the chart below.

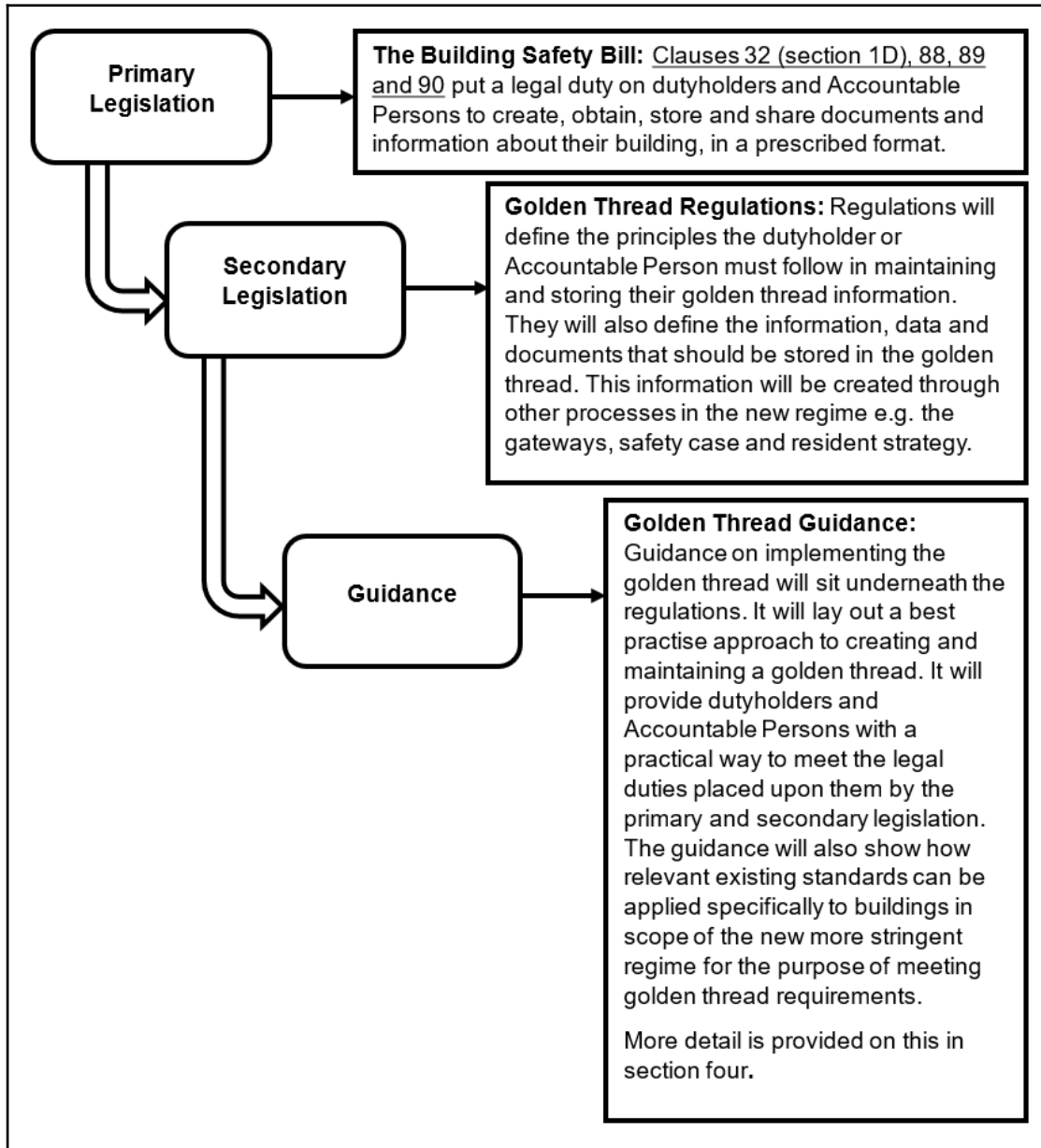


Figure C1.1a. “Golden Thread in UK Legislation”, from UK Government Policy Paper *Building Regulations Advisory Committee: golden thread report*, 21 July 2021

<https://www.gov.uk/government/publications/building-regulations-advisory-committee-golden-thread-report/building-regulations-advisory-committee-golden-thread-report>

Golden Thread Guidance

In the UK, the golden thread is the information that allows one to both understand a building and simultaneously the steps needed to keep both the building and people safe, now and in the future.

1. The golden thread will hold the information that those responsible for the building require to show that the building was compliant with applicable building regulations during its construction and provide evidence of meeting the requirements of the new building control route throughout the design and construction and refurbishment of a building identify, understand, manage and mitigate building safety risks in order to prevent or reduce the severity of the consequences of fire spread or structural collapse throughout the life cycle of a building.
2. The information stored in the golden thread will be reviewed and managed so that the information retained, at all times, achieves these purposes.
3. The golden thread covers both the information and documents and the information management processes (or steps) used to support building safety.
4. The golden thread information should be stored as structured digital information. It will be stored, managed, maintained and retained in line with the golden thread principles (see below). The government will specify digital standards which will provide guidance on how the principles can be met.
5. The golden thread information management approach will apply through design, construction, occupation, refurbishment and ongoing management of buildings. It supports the wider changes in the regime to promote a culture of building safety.
6. Building safety should be taken to include the fire and structural safety of a building and the safety of all the people in or in the vicinity of a building (including emergency responders).
7. Many people will need to access the golden thread to update and share golden thread architects, contractors and many others. Information from the golden thread will also need information throughout a building's lifecycle, including but not limited to building managers, to be shared by the Accountable Person with other relevant people including residents and emergency responder.

C2. Section 2.5 Resources

A. Global Privacy Laws:

The General Data Protection Regulation (GDPR). The harshest of privacy rules. While drafted in the EU it applies to all parties getting in contact therefore making all entities liable – even outside of the EU.

<https://gdpr-info.eu/>

Key-words:

Privacy-By-Design (the system has to respect privacy by its very architecture; it cannot be a negotiable add-on aspect, etc.).

Consent (Processing personal data is generally prohibited, unless it is expressly allowed by law, or the data subject has consented to the processing.)

Personal Data (In addition to general personal data, one must consider above all the special categories of personal data (also known as sensitive personal data) which are highly relevant because they are subject to a higher level of protection. These data include genetic, biometric and health data, as well as personal data revealing racial and ethnic origin, political opinions, religious or ideological convictions or trade union membership.)

B. Age Verification Service Apps:

1. JUMIO:

https://go.jumio.com/age-verification-ppc?utm_source=google&utm_medium=cpc&utm_campaign=2019_%20Age_%20Verification&utm_content=PPC_Age_Verification_Page&bt=379912820116&bk=age%20verification&_bm=e&_bn=g&_bg=76602573024&gclid=Cj0KCQIA2sqOBhCGARIsAPuPK0gJTC2KuUS728XJEacqhaHgybDEwenX2dYeZzby4iuDpw74ub-ACPYaAnRHEALw_wcB

2. YOTI:

https://www.yoti.com/business/age-verification/?utm_medium=adwords&utm_campaign=Yoti_B2B_AV_UK_PPC_Desktop&utm_source=age_verification_exact&utm_content=543755969897&utm_term=e_age%20verification&hsa_acc=3662910724&hsa_cam=11029654691&hsa_grp=108511476616&hsa_ad=543755969897&hsa_src=g&hsa_tgt=kwd-315369290&hsa_kw=age%20verification&hsa_mt=e&hsa_net=adwords&hsa_ver=3&gclid=Cj0KCQIA2sqOBhCGARIsAPuPK0j4tOPinBm95FH9eBR-CJq5JYAM4cuBF2wjFAWU3MS4CuCLFtM0W8aAp_TEALw_wcB

3. VERIFF:

https://www.veriff.com/use-cases/age-verification?keyword=age%20verification%20system&utm_term=age%20verification%20system&utm_campaign=Search+--+Sales+--+Europe&utm_source=adwords&utm_medium=ppc&hsa_acc=1064629533&hsa_cam=11650241724&hsa_grp=108647930050&hsa_ad=480675308155&hsa_src=g&hsa_tgt=kwd-28873706&hsa_kw=age%20verification%20system&hsa_mt=b&hsa_net=adwords&hsa_ver=3&gclid=Cj0KCQIA2sqOBhCGARIsAPuPK0j50vYs5_uMPEWHJWiw91qAWf8DssO9M6o-4raVVz2CeYP9WHbakiQaAuS9EALw_wcB

4. VERIFYMYAGE: <https://www.verifymyage.co.uk/>

C. Identity Verification Service Apps:

1. TRUSTID:

https://www.trustid.co.uk/?https%3A%2F%2Fwww.trustid.co.uk%2Four-serviutm_campaign=Online%20Identity%20Checks

2. IDPAL: <https://www.id-pal.com/home/>

3. TRULIOO:

https://www.trulioo.com/?UTM_Source=Advertising&UTM_Medium=Search&UTM_Content=Brand&ads_cmpid=1695413170&ads_adid=66837987260&ads_matchtype=b&ads_network=g&ads_creative=516212833595&utm_term=trulioo&ads_targetid=kwd-379050819062&utm_campaign=Branding_Competers&utm_source=google&utm_medium=ppc&utm_content=Trulioo&gclid=Cj0KCQIA2sqOBhCGARIsAPuPK0hhNS-A7LzzbV_Rcl0tQxQ2WIANZTJ9yKiBM_Raf5sFi_TvA8DacUaAvOteALw_wcB

4. ELECTRONICIDENTIFICATION: <https://www.electronicid.eu/en/blog/post/identity-verification/en>

5. ONFIDO: <https://onfido.com/>

6.

https://www.trulioo.com/?UTM_Source=Advertising&UTM_Medium=Search&UTM_Content=Brand&ads_cmpid=1695413170&ads_adid=66837987260&ads_matchtype=b&ads_network=g&ads_creative=516212833595&utm_term=trulioo&ads_targetid=kwd-379050819062&utm_campaign=Branding_Competers&utm_source=google&utm_medium=ppc&utm_content=Trulioo&gclid=Cj0KCQiA2sqOBhCGARIsAPuPK0hhNS-A7LzZbV__Rcl0tOxO2WIANZTJ9yKiBM_RAf5sFi_TvA8DAcUaAvOtEALw_wcB

7. GBG:

https://www.gbgplc.com/en/products/id-document-verification/?utm_term=age%20verification&utm_campaign=&utm_source=adwords&utm_medium=ppc&hsa_acc=2994421838&hsa_cam=2017491500&hsa_grp=110089002385&hsa_ad=470928644147&hsa_src=g&hsa_tgt=kwd-315369290&hsa_kw=age%20verification&hsa_mt=e&hsa_net=adwords&hsa_ver=3&utm_term=age%20verification&utm_campaign=IDscan++EU++Scanner&utm_source=adwords&utm_medium=ppc&hsa_acc=2994421838&hsa_cam=2017491500&hsa_grp=110089002385&hsa_ad=470928644147&hsa_src=g&hsa_tgt=kwd-315369290&hsa_kw=age%20verification&hsa_mt=e&hsa_net=adwords&hsa_ver=3&gclid=Cj0KCQiA2sqOBhCGARIsAPuPK0jFOJxd0-ZR11IFzGL6B9eKKr35OzyUyzsGDlee1GfwiKjv6-jDbNlaAnt9EALw_wcB

Please note that some apps offer both age and identity verification.

General Info on Wikipedia: https://en.wikipedia.org/wiki/Age_verification_system

C3. Section 3 Resources

Reprint of D1.1, Section 4.2: PrismArch Taxonomies, Constraints and Rules

The Taxonomies, Constraints and Rules are defined by the user, spatial and time dimensionality of the architectural project. Therefore, we have defined the following standardisation to be followed throughout the project.

A. Taxonomies

World Type, Space Type, Data Type, Process Level, Level of Information, Discipline/User Type, Discipline/User Tools, Discipline/ User Data Type:

Each taxonomy has the following terms:

World Type: Physical Reality, Virtual Reality; World Engine; World Data

Space Type: Project Space, Discipline Space

Data Type: Project Data, Discipline Data

Process Level (UK/USA): Strategic Definition / Briefing, Mobilisation, Preparation of Brief / Feasibility Study, Concept Design/ Schematic Design, Spatial Coordination / Design Development, Technical Design / Construction Documentation, Construction, Post-Construction I / Handover and Close-out, Post-Construction II / Use /

+ Research / + Marketing (PR)¹⁸

Level of Information: Design Reviews, Management, Diagrammatic, Documentation, Presentation, 2D/3D sketch, Orientation, Analysis, Aesthetic, Functional ...

¹⁸ Research & Marketing (PR) was not included in previous version

Discipline/User Type:

Customer: Private Clients, Developer Clients
Admin: Project Administrator
Management: Project Manager
Architecture: Project Directors, Project Associates, Project Architects, Project Contributors, Local Architects
Interior: Project Associates, Project Contributors
Structure: Project Directors, Project Associates, Senior and Design Engineers, Senior and Junior BIM technicians
MEP: Project Directors, Project Manager, Senior and Design Engineers, Senior and Junior BIM technicians
Civil: Civil Engineers
Landscape: Landscape Designer
Consultants: Facade Consultants, Waterproofing Consultants, Fire Protection Consultants, Lighting Consultants
Contractor : Main Contractor, Facade Contractor, Glazing Contractor, MEP Contractor
Press: Senior Associate, Digital Media Manager
Event Curator: Event Curator, Venue Owner
Public Audience: Public Engagement

Discipline/User Tools:

Immersive: GravitySketch, Medium, Mindesk, Twinmotion,
Game Engines: Unreal Engine 4, Unity, Omniverse
Specialist: IES Virtual Environment (IES VE), One Click LCA, Elmhurst Design SAP, Pyrosim, Pathfinder, Specialist AV Software, Reakt (AKT's in-house Interoperability platform), Project-specific C# and Python scripts, Plaxis 2D /3D
3D: 3ds Max, Maya, Microstation, AutoCAD, Revit, Navisworks, MicroStation, BIM360, Rhinoceros 3D, Grasshopper, SketchUP, SolidWorks, Trimble ProDesign, DIALux, WinDEBUT, GASWorkS, SAP, Sofistik, Robot, FBeam, S-Concrete, TEDDS, RAPT, IDEAstatica
2D: Sweco Internal Software, Bluebeam Revu, Sticky Notes, MS OneNote, MS Planner, Microsoft Office, After Effects, Media Encoder, Indesign, Photoshop, Illustrator, Google Maps, Web Browser, PDF Xchange, Snipping Tool, Photos
Communication: Zoom, MS Teams, Skype, Outlook
System: Operating System, Graphics Card
Physical: Meetings, Reviews, Phone Calls, Physical Model, Hand Sketches, Hand Calculation

Discipline/ User Data Type:

Architecture:

- Contractual setup
- Local and global building regulations
- Site info (models, photos, point cloud data, weather, geographical data)
- Precedent projects (formal and program analysis, images, 3D models)
- Cultural and aesthetic references (concept research, images)
- Solar / Shading studies

- Materials and textures
- Construction method
- Computational tools and software (design process and results [e.g. grasshopper parameters to explore aesthetically-pleasing and energy-efficient facade design])
- Project brief (client's requirements)
- Program, Circulation, Zoning (private vs public), Project Area Calculations, Spatial Organisation
- Physical models and mockups
- 2D orthographics drawings (plans, sections, elevations)
- Hand sketches

Structure:

- Historical site information and database
- Local and global building regulations
- Safety and durability
- Structural performance (load, stress and strengths)
- Material performance and characteristics
- Beams, columns and foundations
- Building sustainability
- Architectural design, programme quality, adaptability, buildability
- Health and safety
- Test results
- Documentation

MEP:

- Building height ratio, environmental data
- Basement space planning
- Traffic and logistic planners
- Landscaping and public realm ventilation openings to spaces below ground
- Rainwater attenuation and storage
- Sub-slab drainage
- Riser and distribution zoning
- Low and zero carbon systems (LZC) integration
- 3D model / building information modelling (BIM)
- Construction programme and cost plan
- Electrical distribution
- Lighting requirement

B. Constraints:

Access Privilege

Individual/ User Data:

- "Cannot proceed / Work in progress" (personal)
- "Can View" (within the same discipline space)
- "Can Comment" (within the same discipline space)

- “Can Edit” (within the same discipline space)
- “Can Share” (within the same discipline space)
- “Can Submit” (to external disciplines not to client)

Discipline Project Data (Team-based Information):

- “Cannot proceed / Work in progress” (only open to the same discipline members)
- “Can View” (within the same discipline space)
- “Can Comment” (within the same discipline space)
- “Can Edit” (within the same discipline space)
- “Can Share” (to external disciplines not to client)
- “Can Submit” (to client)

Data Authority / Ownership of Data:

PrismArch based General IP: Data created within the experiential space by each author/user from any discipline

Discipline brand IP: Data loaded from the Physical Reality space, or modified data based on the loaded data.

Public IP: This is equivalent to putting information out into the internet - any information put out into the public realm must be ironclad, and filtered significantly to protect our immediate commercial interests.

Client Owned IP (proprietary): Discipline-based IP and liability management (proprietary) - distinct specialisation and distinct liability that is part of the overall skill-set needed to resolve the project.

C. Rules

Licensing of Commercial IP - IP is owned by author / designer / manufacturer (ownership of design expertise and brand always stays with the design practice or author).

Project-related Information that is singular in its use, and is needed to resolve the project, is released by the respective authoring entity into the shared project space. This information is temporal, licenced to the client for a certain use, and is hosted inside the client’s domain. There could be some exclusivity to this information (use limited to a certain market).

Public Realm and exclusivity: the project could, potentially, only be visible to a certain part of the public, dependent on royalties for distribution and rights.

Examples: “Project Contributors can not close their assigned tasks unless the Architectural Project Director approves them” or “Project Directors from all disciplines can not submit their production until getting approvals from all other Project Directors. Until this is done, the project cannot proceed to next process level/phase (however, all disciplines can go back to all submitted and approved project-related events (incidents) from the past” or “Client can visit and approve the discipline approved content inside VR” or “AI agent can not edit Structural design but can propose the correct structural loading criteria based on the applicable local structural regulations (e.g. hurricane zone wind loads)”

C4. Section 4 Resources

Microsoft Documentation, Learn Mixed Reality, 10 March 2022.

<https://docs.microsoft.com/en-us/windows/mixed-reality/enthusiast-guide/learn-mixed-reality>

Accessed 24 August 2022.

Microsoft Documentation, Start your VR journey, 14 April 2022.

<https://docs.microsoft.com/en-us/windows/mixed-reality/enthusiast-guide/vr-journey> Accessed 24

August, 2022.