D2.1.3

# First Blueprint Architecture for Social and Networked Media Testbeds

v1.02 / 2012-10-23

David Salama (ATOS), Elena Garrido Ostermann (ATOS), Peter Ljungstrand (Interactive Institute), Selver Softic (Infonova), Stefan Prettenhofer (Infonova) , Michael Boniface (IT Innovation), Simon Crowle (IT Innovation), Stephen C. Phillips (IT Innovation), Wolfgang Halb (JRS), Kleopatra Konstanteli (NTUA), Angelos Yannopoulos (NTUA), Manolis Wallace (FHW)

This first blueprint architecture for social and networked media testbeds document provides the architectural foundation for the EXPERIMEDIA facility (connectivity phase; Year 1). A range of scenarios exploring media application services (including social network integration, UGC management/delivery, high quality content management/delivery, 3D internet tools, AR tools and cloud integration), along with the results of earlier EXPERIMEDIA deliverables (D2.1.1 First Methodology; D2.1.2 Scenarios and Requirements; D3.1.1 Infrastructure and Software Assets Inventory) serve to inform its design. The architecture presented here specifies a framework that integrates the FMI experimental facilities provided by venues such as Schladming, CAR and FHW with a technical ecosystem offering FMI capabilities to support test-bed design and implementation. Within the framework, these elements are specified using a FMI based capability abstraction linked to an extensible technology model. Blueprints for the initial EXPERIMEDIA driving experiments are set out using this architecture.

SEVENTH FRAMEWORK PROGRAMME

www.experimedia.eu

| | |
|---|---|
| Project acronym | EXPERIMEDIA |
| Full title | Experiments in live social and networked media experiences |
| Grant agreement number | 287966 |
| Funding scheme | Large-scale Integrating Project (IP) |
| Work programme topic | Objective ICT-2011.1.6 Future Internet Research and Experimentation (FIRE) |
| Project start date | 2011-10-01 |
| Project duration | 36 months |
| Activity 2 | Construction |
| Workpackage 2.1 | Architecture blueprint |
| Deliverable lead organisation | IT Innovation |
| Authors | David Salama (ATOS), Elena Garrido Ostermann (ATOS)<br>Peter Ljungstrand (Interactive Institute)<br>Selver Softic (Infonova), Stefan Prettenhofer (Infonova)<br>Michael Boniface (IT Innovation), Simon Crowle (IT Innovation), Stephen C. Phillips (IT Innovation)<br>Wolfgang Halb (JRS)<br>Kleopatra Konstanteli (NTUA) |
| Reviewers | Angelos Yannopoulos (NTUA)<br>Manolis Wallace (FHW) |
| Version | 1.02 |
| Status | Final |
| Dissemination level | PU: Public |
| Due date | PM6 (2012-03-31) |
| Delivery date | v1.0 2012-04-11; v1.01 2012-04-13; v1.02 2012-10-23 |

| **Version** | **Changes** |
|---|---|
| 1.0 | Initial release |
| 1.01 | Added missing figure |
| 1.02 | Updated front page and metadata table; minor corrections including updated conclusion |

# Table of Contents

# 1. Executive Summary

This first blueprint architecture for social and networked media testbeds document provides the architectural foundation for the EXPERIMEDIA facility (connectivity phase; Year 1). A range of scenarios exploring media application services (including social network integration, UGC management/delivery, high quality content management/delivery, 3D internet tools, AR tools and cloud integration), along with the results of earlier EXPERIMEDIA deliverables (D2.1.1 First Methodology; D2.1.2 Scenarios and Requirements; D3.1.1 Infrastructure and Software Assets Inventory) serve to inform its design. The architecture presented here specifies a framework that integrates the FMI experimental facilities provided by venues such as Schladming, CAR and FHW with a technical ecosystem offering FMI capabilities to support test-bed design and implementation. Within the framework, these elements are specified using a FMI based capability abstraction linked to an extensible technology model. Blueprints for the initial EXPERIMEDIA driving experiments are set out using this architecture.

Experimenters and technologists approaching the EXPERIMEDIA architecture are introduced to venue infrastructure components, baseline and experimental technologies situated within the contexts of content and experiment lifecycles. A series of indicative examples taken from augmented and mixed reality scenarios; technically enhanced, high performance sport training programmes; and novel, cultural education experiences are used to illustrate the potential content lifecycle demands. A FMI based capability model and analytical technique is then introduced as a means of capturing aspects of both content and experimental lifecycles in scenarios such as these. An example of this process is provided using a subset of the technical components introduced in one of the driving experiments. This 'top-down' view of EXPERIMEDIA FMI assets is complemented by a 'bottom-up' technology model which provides a technical account of the coupling of hardware, software and services required to deliver content and experimental lifecycles. Four technical super-structures are introduced: social, audio/visual, pervasive and experiment content models. Interfaces between the super-structures and the underlying infrastructure realise the mechanism by which digital content can be acquired, processed, delivered and managed to provide novel FMI experiences and also capture and manage experimental test-bed data.

Three driving experiment blueprints are set out using the EXPERIMEDIA architecture described above. Each provides the functional components required to implement the test-bed. This is an integration of the venue infrastructure; baseline technologies and capabilities provided by the initial EXPERIMEDIA architecture; and bespoke, experiment specific developments. The particular content lifecycle and experimental data flow is addressed in each case, along with anticipated deployment considerations. This report concludes with a summary of the architecture covered for technologists and experimenters wishing to engage in the EXPERIMEDIA environment.

# 2. Introduction

## 2.1. Purpose

The purpose of this document is to provide facility developers and those conducting experiments with a description of generic Future Media Internet capabilities and technology to be offered by EXPERIMEDIA and a technical methodology for assessing how such technology can be incorporated into experiments.

The primary audience are those responsible for developing implementation technologies, integrating and interconnecting related systems, and operating all or parts of the EXPERIMEDIA systems.

## 2.2. Scope

This document is the first blueprint architecture for social and networked media test-beds. The document describes architecture for EXPERIMEDIA facility for the connectivity phase (Year 1) The architecture considers media application services for the scenarios (social network integration, UGC management/delivery, high quality content management, delivery, 3D internet tools, AR tools, cloud integration), test-bed management services supporting the experiment lifecycle and the technical counter measures necessary to ensure security and privacy of personal data. This document builds pervious deliverables D2.1.1 First EXPERIMEDIA methodology[1], D2.1.2 Scenarios and Requirements[2], D3.1.1 Infrastructure and Software Assets Inventory[3].

EXPERIMEDIA needs to describe the capabilities expected within a Future Media Internet (FMI) architecture and not just the EXPERIMEDIA facility or a specific experiment. As such the descriptions needs to consider the generic Architecture model for a FMI experimental facilities such as those being offered by at Schladming, CAR and FHW. As part of the work to produce the Architecture Blueprint we need to reach a consensus on what capabilities are within an FMI system. By providing a capability map for the FMI with baseline components providing basic implementations we offer the possibility for experimenters to understand how to integrate their technology within an EXPERIMEDIA ecosystem and to support multiple implementations of the same capability if necessary. For example, one experiment may want to focus on P2P content delivery whilst another may focus on augmented reality applications. What they need to is to understand where their experimental components fit into the overall FMI architecture and what generic baseline components from EXPERIMEDIA they need to integrate with.

The approach combines top down and bottom up analysis. The objective is to separate capabilities (what needs to be done) from their technology implementations (how stuff is implemented). Top down we create a Capability Map focused on the business/information models and bottom up with start to describe a Technology Model.

[1] http://www.scribd.com/doc/86068612/D2-1-1-First-EXPERIMEDIA-Methodology-v1-01
[2] http://www.scribd.com/doc/79825559/D2-1-2-Requirements-and-Scenarios-v1-01
[3] D3.1.1 is currently restricted to FP7 programme participants but is expected to be published soon or is available on request.

## 2.3. Initial Guiding Architectural Principles

The purpose of architecture is to provide an abstract description of the structure and behaviour of a system, and the desired impact the system is required to have on its environment. Architecture describes the system scope, what outputs a system produces (in response to inputs), the processes for delivering the outputs, and the resources necessary both in terms of people and other assets.

Architecture is fundamentally communication mechanism and a way to help everyone understand a system. A significant challenge in comprehending a system is that most are complex. A primary goal is to deal with complexity through abstraction and decomposition techniques in a way that considers design principles such as of encapsulation, high cohesion, and loose coupling. Many methodologies have emerged in recent years to support the process of architecture definition. The evolution of methods is driven by both advances in technologies and the types of systems under construction. Our objective is to intelligently select techniques that are most useful for the specific architectural characteristics and challenges faced by EXPERIMEDIA rather than to adopt a single methodology universally.

The specific characteristics of EXPERIMEDIA that must be considered throughout the architectural design are included in the following list.

- **Evolving Requirements**: we are describing architecture but cannot know all requirements in advance. We can describe the general capabilities for an FMI architecture and what it means to operate a facility supporting such systems. However, new requirements will emerge from experiments using the facility that cannot be envisaged now.
- **Integration and Adaptation:** each experiment will develop and operate a FMI system that consists of EXPERIMEDIA baseline technology components, EXPERIMEDIA infrastructure components and experimental components. Architecture must be developed in a way that ensures loose coupling between and efficient integration of components in a way that creates a system of systems. Standardised interfaces should be adopted where possible to reduce need for specific adaptations.
- **Experimentation:** experiments typically require components with high degrees of instrumentation and control to attain insight into the behaviour of systems, their relationship with users and to ensure validity by reducing the influence of extraneous factors and providing repeatability.
- **Security and Privacy:** experiments must be legally compliant in accordance with data protection legislation and therefore must be considered a critical attribute of component and systemic capabilities. Security and Privacy must be by design rather than an add-on.
- **Technology Baseline:** EXPERIMEDIA is not architecting a system from scratch but from a set of technologies supporting different capabilities within the Future Media Internet, and targeting known infrastructure environments. The architectural process needs to combine top down analysis of desired capabilities alongside a bottom up assessment of how each baseline technology and infrastructure supports them. Through

this process overlaps, gaps and integration points can be determined which can inform future development tasks

- **Constraints:** Each component delivers a capability but also has technological and operational constraints on use. For example, technically a component may only support specific protocols or operational may be only available at certain times and with limited resources. This is especially relevant for infrastructure components at each venue that are operated, sometimes by 3rd party companies, for "other" purposes (i.e. EXPERIMEDIA does not have exclusive access).
- **Time Limitations:** the system lifecycle is organised into iterative and incremental activities, with each iteration expected to add functionality. The first iteration is the most challenging considering the novelty of the process, the levels of domain knowledge and maturity of collaborative relationships. The scope of the architecture and capability descriptions is likely to far exceed what can be delivered during the first iteration with significant need to prioritise critical components and integrations between them
- **Viewpoints:** architecture can be described from multiple perspectives; we need to consider how the architecture is presented to different stakeholders.

## 2.4. EXPERIMEDIA Capability Landscape

Figure 1 provides a high level view of the EXPERIMEDIA capability landscape from the aspect of the venue infrastructure (green), baseline technologies (blue), experiment technologies (red) and experiment lifecycle (orange). Each of the venues offers an ecosystem and technical infrastructure for experimentation. EXPERIMEDIA extends this infrastructure with 1) Baseline FMI technologies for delivering new experiences to users 2) additional technologies to support robust and efficient experiment. Finally Experimenters introduce Experimental technologies into the venue that are combined with baseline technologies and venue infrastructure to produce an Experimental FMI System that is assessed and evaluated from both quantitative and qualitative perspectives.

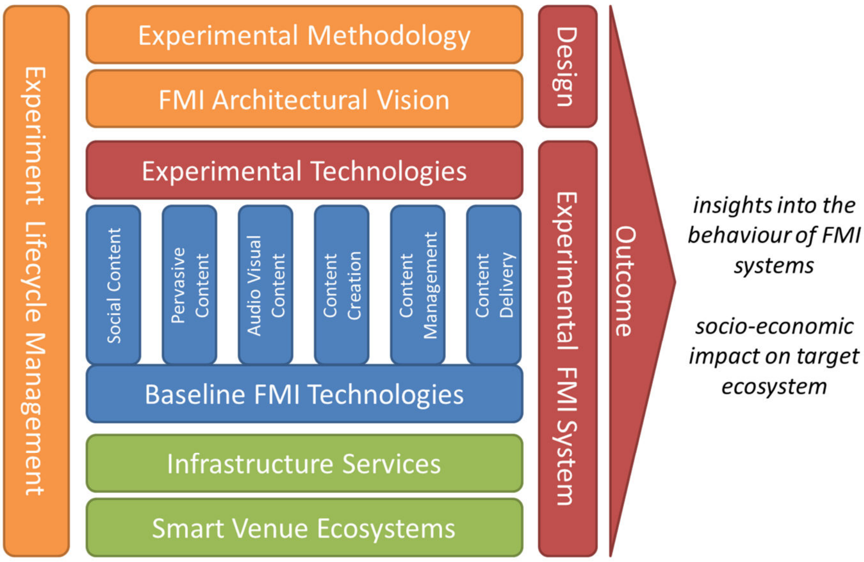


**Figure 1: EXPERIMEDIA capability landscape overview**

## 2.5. Content Lifecycle

The delivery of FMI experiences is closely related to the lifecycle of content. Although there are other lifecycles (e.g. service, experiment and resource) that are relevant and necessary for EXPERIMEDIA, the lifecycle of content is what has the largest impact on Quality of Experience for users and resource management for service providers and network operators. Understanding how the FMI can enhance or disrupt existing content lifecycles will be a major objective for experiments.


**Figure 2: Simple content lifecycle**

The content lifecycle describes activities necessary for the delivery of experiences from a content centric viewpoint. At some stage during the development of the project we may want to consider shifting from "content lifecycle" to "experience lifecycle" as it is the experience that we are interested in assessing rather than the content itself. It may be that this is implicit within some content lifecycles but highlighting the experience may help us focus on the end rather than the means. For the purpose of our analysis at this stage we continue with the content lifecycle and split into three main activities shown in Table 1 below.

**Table 1: Components of the content lifecycle.**

| Phase | Description |
|---|---|
| Content Authoring | Activities that create new content including 3D, 2D, text, metadata. Note that authoring also relates to processes, narratives, and rules. Content Authoring does not naturally lend itself to these process aspects in ways that Experience Authoring might. |
| Content Management | Activities related to the retention and access to content including how content is stored, ingested, described, accessed, searched, navigated, shared, etc. EXPERIMEDIA's primary focus is on social and networked media experiences. The characteristics of these applications are that they require real-time interaction between individuals/communities over the Internet. As such the "Timeline" of events is the core concept for content management in synchronising remote activities. Timeline synchronisation requirements differ between applications. For example, recording a sports training session requires high precision time synchronisation of different data sources if it is to provide useful training insights, however, it is better that the data is correct rather than it appears in real-time. In contrast if the sports training event was broadcast to a live audience then it may be more important to broadcast quickly with a lower level of accuracy on the temporal synchronisation. Now if we consider the FHW scenario where we want to record an interactive education session things are more subjective and accuracy can be dropped, you want to know what events cause questions or changes in direction within the virtual world or additional relationships but not necessarily that these happened at 1h 2.234 secs into the event. |
| Content Delivery | Activities concerned with getting the content to target audiences and participants at the time they need it and in a way that considers their situation and preferences. In traditional, content lifecycle models authoring is offline and can be considered a design time activity. Content creators create web pages and 3D models which are then composed to create applications. Creating an avatar in Second Life is an offline design task. However, increasingly content is dynamically produced based on real-time data. For example, if an avatar is created by automatically capturing human motion and emotion is this part of delivery or authoring? This separation between design-time and real-time authoring is increasingly being blurred through faster networks, faster algorithms and more computation processing power available on demand. A key aspect of content delivery is adaptation to context. Context relates to all of the external elements that need to be considered by an FMI system when delivering content. Increasingly systems are being deployed into situations where either the context is dynamic or even in some cases unknowable. Context can include social, environment, technical, contractual, security and interaction (e.g. HCI). Providing ways to dynamically adapt to context is a key element for us. Delivery places large demands on infrastructure as it is concerned with providing sufficient capacity to scale to larger communities and/or content quality, and performance to ensure a satisfactory quality of experience is provided. Delivery requires all the services necessary to get the content to the users and includes hosting, provisioning, and scaling services, distributing content over networks and ensuring content is distributed efficiently through high performance storage, along with replication and caching strategies |

Our analysis aims to consider EXPERIMEDIA experiments and technologies in relation to the content lifecycle. Whether we are considering an augmented reality (AR) application, a social networking site or a tele-immersion system each has its own mechanisms for creating, managing and delivering content. By considering how each technology deals with the content lifecycle allows us to understand how content from different sources can be used and aggregated with other content to deliver new types of experiences.

In the following sections we discuss different ways in which experiences are delivered through different types of content. These are representative examples rather than concrete experiments and just provide an exploration of the possible experiment space.

### 2.5.1. Experience Example: Augmented Reality

Figure 3 shows a generalised architecture for a mobile augmented reality application. The purpose of this application is to provide an individual with more information about their current location than they can acquire just by looking themselves. This is done by super imposing previously authored content onto a live video stream based on the user's location, point of view and preferences. Users can also interact with the content as they would with a normal web browser to display more information about specific objects. If we explore this type of application from the perspective of the content lifecycle we see:

- Authoring: to create such an application it is necessary to author 3D models, author associated metadata and bind the content to a location/point of view.
- Management: created content must be stored, indexed and made available to those wanting to access it
- Delivery: a video stream must be produced and rendered that combines live video with other 3D/2D Textual Content in real-time based on a user's location and point of view.

The interesting aspect of this application is that it is assumed that the delivery all happens on a mobile device. The camera feed, location sensor, orientation sensor and the display are provided by the mobile device with some additional content provided by a content management system. It is important to break this application down into its component parts because in EXPERIMEDIA we need to imagine situations where content, its management and its delivery are not performed within a single monolithic system. We could imagine delivering an individual view to another person either with or without the augmented content or sourcing content from other content management systems.

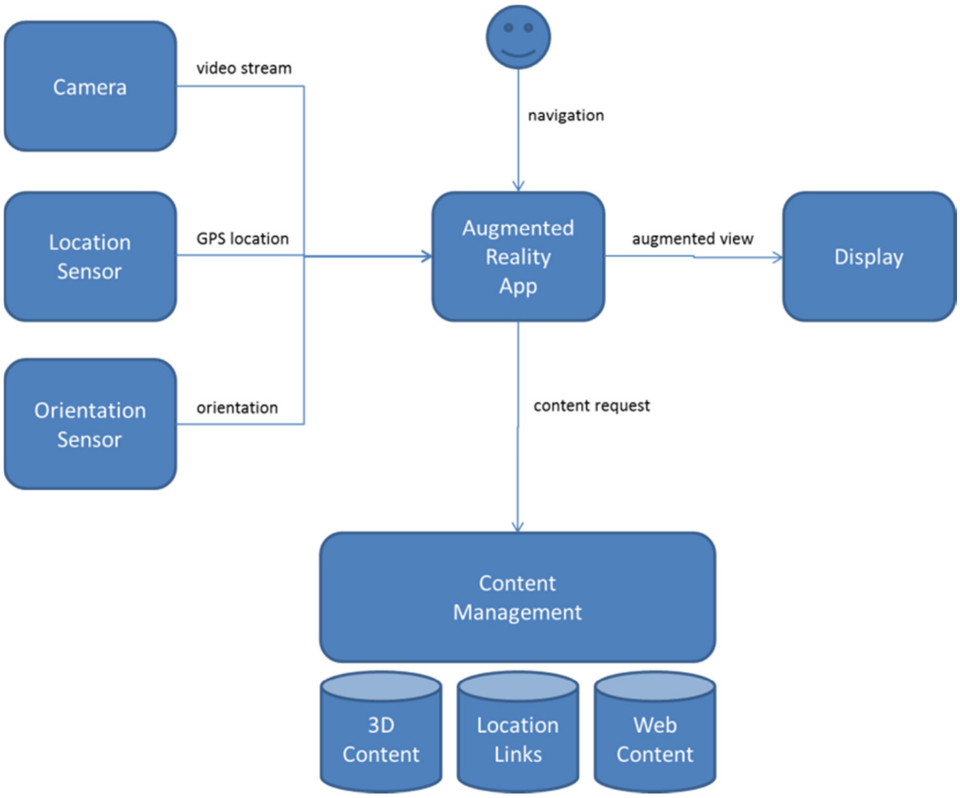


**Figure 3: Example augmented reality application**

### 2.5.2. Experience Example: Mixed Reality Content

The model of authoring, management and delivery can be applied to other situations. The augmented reality scenario described above is about delivering to a single user. Imagine a more interactive scenario where users are distributed in different locations but need to interact in a mixed/augmented reality world. What would such a scenario look like? If we take a recent development from the project Rev-TV[4] we can understand the challenges.

The purpose of Rev-TV is to create a new type of game show that allows participants to join a game as their avatar. The main challenge is that the avatars are constructed in real-time from motion and emotion detection within the home and then combined with studio video feeds for broadcast to the wide audience. The avatars are virtual, the set is virtual but the presenter is real. A basic description of the relationship to the content lifecycle is given below

- Content Authoring: the content for the game show set and base avatar content is created in advance. This includes possible emotions that can be associated with each of the participants. We assume that the rules of the game are also part of content authoring process, for example, if a player makes a gesture (nods head/raises hand) this has meaning in terms of the game rules
- Content Management: the content required for the game is pretty minimal but it is necessary to store the 3D models, feeds and game play rules. There will also be a requirement for archive but this would be beyond the scope of this pilot application

[4] http://www.rev-tv.eu/

- Content Delivery: the content is delivered by mixing 3D content with live video recorded within the studio. The key innovation here is that the avatars are augmented with real-time content from motion and emotion sensors in the home. When a player participates they can move arms and head to play the game and this is displayed to the audience. In addition emotion detection is used to render each participant's facial expressions.

Now the process of augmenting 3D content with live video seems very similar to the AR application described in the first case, however, the way by which 3D content is created and delivered to an audience is different. 3D humans are created on the fly and the sensors, cameras and displays are distributed over a wide area network, including the broadcast network rather than on a single device.

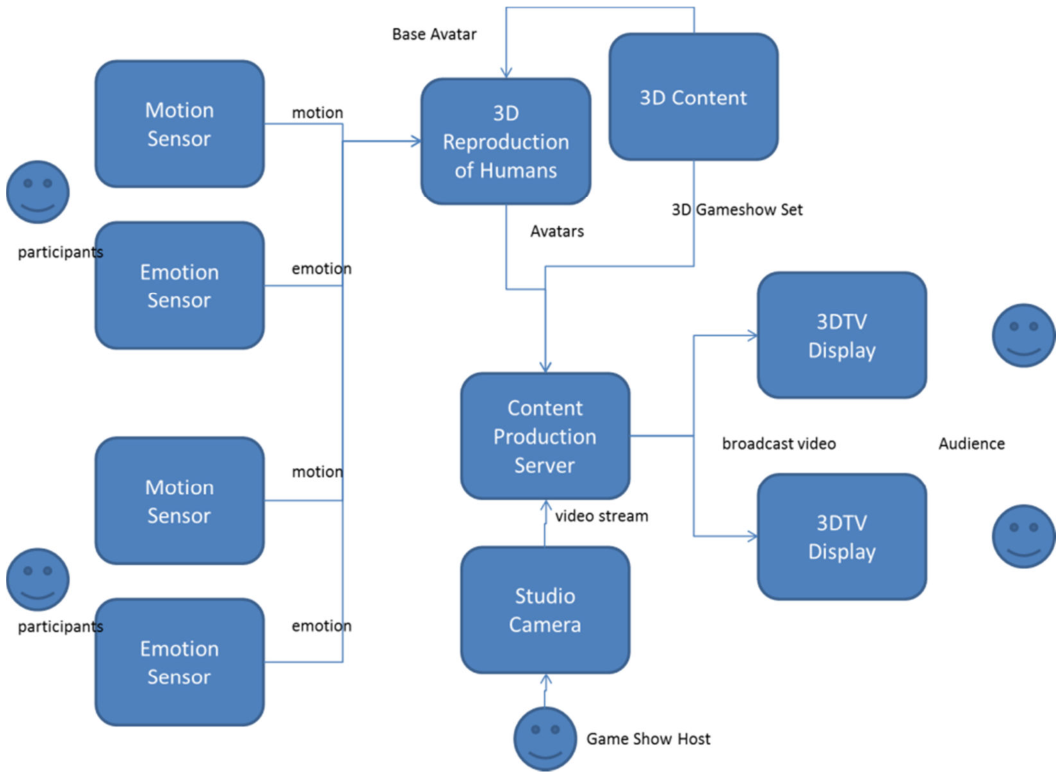


**Figure 4: Example mixed reality application**

What is interesting about using the content lifecycle is that it highlights the potential for interactions between different technologies. If we adopt an approach that decomposed focusing on augmented or 3D experiences we would end up decomposing in technology silos rather than highlighting the generic capabilities.

### 2.5.3. Experience Example: High Performance Sports Training

The CAR scenario focuses on using FMI technologies to monitor and measure sports performance for high performance athletes. The expectation is that using technology will increase the perception of performance and provide more efficient mechanisms to direct training plans. High performance sports training is an extremely instrumented endeavour with lots of quantitative data collected about athletes' physiology (e.g. heart rate, strength, etc) and sport outcomes (e.g. times). The idea is that using advanced video analytics and even 3D capture technologies additional performance insights can be automatically generated and replace manual observations of the live event or video recordings. In effect, using FMI will capture a richer memory of the training experience that can be analysed both in real-time at the track/pool side or later. Here's the relationship to the content lifecycle

- Content Authoring: the content for this scenario relates to the athletes training and competitive event performance. Coaches are responsible for developing training plans and performance objectives. There are various sources of information from physiological sensors, human observations (e.g. how do I feel today), timing devices, and metadata automatically generated from video analysis algorithms. It is the use of video analysis for generation of metadata or even 3D reconstruction of humans that advances the SOTA for CAR. Moving from video data to an understanding of what is happening is important. This can be an automated (e.g. video processing algorithm) or manual process (e.g. key frame annotation) There will clearly be some link to healthcare data (e.g. if the athlete is injured) and dietary information, although that has not been elaborate
- Content Management: maintain a record of an athletes' performance is expected to be maintained in some training system (for CAR we are not sure) that provides coaches, physiotherapists, and other specialists with the necessary tools to record, search and access information about an individual. In addition, we expect video content to be recording in some digital asset management system that records audio/visual material and associated metadata about the video (e.g. how/when recorded). The challenge is to associate the "memory" of a training event as encoded in the video with "understanding" of what happened as recorded in the training system. The integration of different content management systems will be a key challenge. Real-time feedback and high precision timing is a key characteristic of this scenario. We have multiple sources of data that needs to be synchronised accurately over the training event timeline. In a localised situation (e.g. at a swimming pool) this synchronisation is possible as the system context is limited to a single location and will not be subject to networking effects or multiple contexts which we will see in remote interactive scenarios (e.g. REV-TV).
- Content Delivery: the consumers of content are coaches (local and remote) and athletes participating in a training event. During the live event there's not much interaction with the digital media. For example, the athletes are not looking at their screens but are focused on their event. The coaches will be making observations, will be redirecting or even terminating a session but the point is at the track/pool side the digital representation is a rich memory of the experience not the experience itself. If a coach is at a remote location then the situation is different and their experience/perception of a training session will be entirely represented by digital media. However, a coach (remote or local) will only be providing remote feedback rather than participating in the actual sport. The issue for delivery is therefore presenting a memory of the live event for review by all interested parties.

### 2.5.4. Experience Example: Amateur Sports Training

The current scenario described in D2.1.2 focuses organising a day out with a family at a resort with specified rendezvous points, opportunistic events and rescheduling. Using social networks, groups would be able to share their experience (e.g. text messages) and make recommendations to each other in real-time. This scenario focuses on the après-ski activities. An alternative would be to focus on sports performance which is of interest whether you are amateur, expert or professional. People who go skiing want to improve their performance and training/coaching is an essential element of most skiing holidays. We can imagine a new service being developed that provides amateur skiers with performance training while on holiday.

In our scenario, we envisage that a skier captured their route down the mountain with various body sensors and cameras (attached to helmet). The information was streamed in real-time to a performance analytics system that was responsible to assessing the run down the slope and producing rich media content that could be viewed by the skier on their smartphone after the run on the lift or at the bottom of the slope. We imagined that the skier could synchronise the content at a Wi-Fi point associated with the ski lifts and even connect to expert coaches online who could assess the data produced and provide redirection. You could even compare your run with a professional skier in terms of posture, positioning, forces, speed, etc.

Examining the content lifecycle

- Content Authoring: here it's all about capturing the skier's performance on the slope through physiological sensors and video cameras. This data would then need to analysed to create metadata about performance and actionable recommendations about what the skier should do on the next run
- Content Management: in a similar way to CAR, the content management must capture a skiers training objectives and their progress towards them. Digital Asset Management would be necessary to store source sensor and video data.
- Content Delivery: this all depends on how the final results are presented to the users. You could imagine storing the source sensor and video data and producing an annotated video in real-time or producing a single video including annotations as a "post production activity" and then forgetting the source material. Of course you could then publish this stuff online to your friends within social groups if you wanted but this "dissemination" aspect of the scenario does not add much to the capture of the richness of experience.

### 2.5.5. Experience Example: Cultural education

The FHW scenario has much potential which is not currently described in the D2.1.2 deliverable. The scenario has a group of people attending the Theatre at FHW to learn about ancient Greece. There's a local host who is responsible for conducting the session, presenting, answering questions and navigating through the virtual world. The audience can interact with the local host or remote experts throughout the session causing changes in direction of the route through the virtual world.

What is important for this event is capturing and understanding the learning process. Here's where the timeline is important. If the event is 1 hour in during what we need to do is capture all the things that have happened and to try and understand why. For example, what routes are we taking and why? What questions are being asked, why are they important and what impact does this have on the group? What relationships are being established and in response to what information? This sort of information is useful for the event organisers so they can improve the experience but also to those in the audience. Why not have this on an iPAD so that members of the audience can join conversations and see what's going on?

Looking at the lifecycle again:

- Content Authoring: here we have the virtual world and associated metadata about the cultural assets which will have been created in advance. During the session user generated content will be created such as questions/answers between audience and experts. Not sure how the local discussion between local host and audience will be captured, maybe video
- Content Management: we have the content management systems to store the virtual world and whatever systems are used to store cultural metadata. In addition there's the social networking sites used to discuss the event. The important thing is to associate the real-time discussion with the cultural content in some way.
- Content Delivery: here we need to present the virtual world within the Theatre which no doubt is some sort of bespoke system but we need details. In addition services must be established to allow the audience to interact online through some social application between themselves and remote experts.

# 3. Technical Methodology: Capability Analysis

The objectives of this section are to present the EXPERIMEDIA capability analysis process and model to the reader with a view to a) reaching an understanding of the first EXPERIMEDIA FMI capabilities framework and b) prescribing a method to support the analysis of future assets that will render a compatible capability model for use within EXPERIMEDIA.

In the First EXPERIMEDIA methodology v1.0[5] (D2.1.1) the activities related to designing and executing an experiment are outlined (see Figure 5). This document focused on applying Value Impact Assessment from the dimension of User, Business and Technology. A key aspect of the methodology is assessing the "Technical Impact" in relation to the Blueprint Architecture and technologies within the EXPERIMEDIA toolbox. Within the context of this evaluation, we describe the methodology for assessing technical impact of experimental technology.

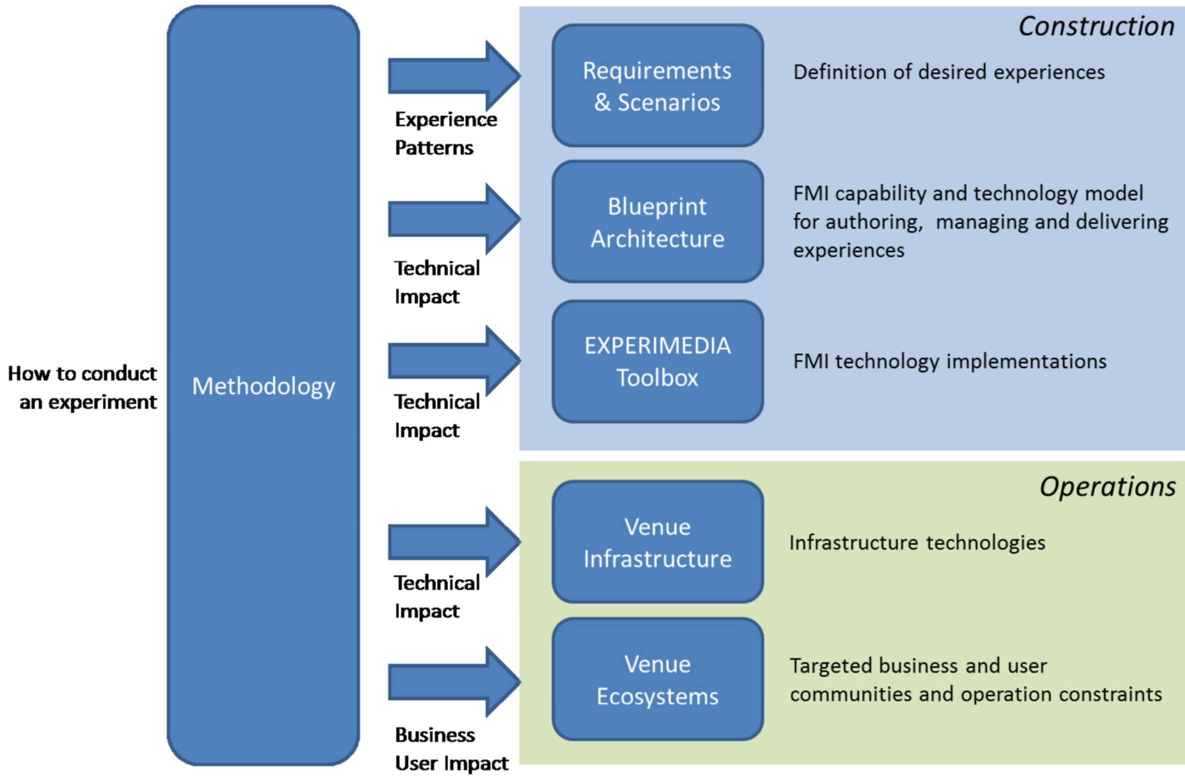


**Figure 5: Relationship of methodology to EXPERIMEDIA activities**

In this architecture document, Sections 3.1.1-3.1.2.5 introduce the reader to the concepts and methods of the FMI based capability analysis and modelling; the high-level, non-technical nature of this method is emphasized here. In Section 3.1.3, the 'top-down' picture of EXPERIMEDIA's capability map is presented and the format in which capabilities are specified is discussed.

[5] http://www.scribd.com/doc/86068612/D2-1-1-First-EXPERIMEDIA-Methodology-v1-01

## 3.1. The Capability Model and Analysis Process

In Section 2.5, we explored a variety of indicative FMI scenarios to understand some of the essential characteristics of how new technologies could deliver innovative social and networked media experiences. Technologists wishing to evaluate their technical assets in an EXPERIMEDIA test-bed will need to be able to describe their capabilities to other stakeholders in the project. Clearly a detailed, technical specification of each asset will be required during the later phases of integration during test-bed development. However, some positioning of the technology within a broader framework of FMI capabilities is also required so that other stakeholders (including venue providers and experimenters) are able to understand where the technology 'fits' in relation to other components, and how it adds value to EXPERIMEDIA framework as a whole. The capability model and analysis process described below addresses this requirement by offering an analytic and descriptive means by which an asset's capabilities can be specified.

Our approach to this representation has deliberately de-coupled the description of the capabilities required by the FMI scenario from the detailed technical specification of the components that ultimately implement them. This separation is essential for the development of the EXPERIMEDIA's FMI capability framework or 'landscape'. De-coupling in this fashion generates high-level abstractions that can be used directly in the analysis and design of FMI content and experimental lifecycles without the burden of having to immediately select and satisfy the technical constraints. Using this high-level capability view also offers the opportunity to evaluate capability fitness in two contexts:

- Formative: the extent to which current FMI capabilities meet the content/experimental lifecycle requirements for a proposed experiment
- Summative: the extent to which a new FMI capability set brought to EXPERIMEDIA enhances an existing capability model

Both formative and summative contexts require a common (and extensible) framework for comparison. The ultimate goal of the capability modelling method described here is to deliver an extensive framework of capability descriptions that provides wide ranging coverage of FMI related test-bed capabilities. At this early stage of the EXPERIMEDIA project, it is realistic to expect that only a limited sub-set of the final framework will be captured; over the life-time of EXPERIMEDIA we expect this capability landscape to evolve and grow.

To contribute to the development of the capability framework, an analysis process must be executed on an asset, this is outlined below.

- Phase 1: Asset deconstruction through the content lifecycle
- Phase 2: Asset deconstruction through the experiment lifecycle
- Phase 3: Capability specification

We will explore these phases using as an example some of the components described in the Schladming driving experiment (see Section 5.1). For this purpose, the essential elements that combine to provide the capabilities of presenting 2D and 3D overlays of related content based on geo-spatial location, over a live video stream are examined. The following sections outline each of the phases used to arrive at a resultant FMI capability abstraction.

### 3.1.1. Phase 1: Deconstruction through the content lifecycle

The first phase of this process is a deconstruction of a FMI asset within the frame of the content lifecycle. This takes place along two dimensions: entities and behaviours see Figure 6. In practice, this process is likely to execute iteratively; however an outline of each sub-domain is presented serially below for the purposes of explanation.

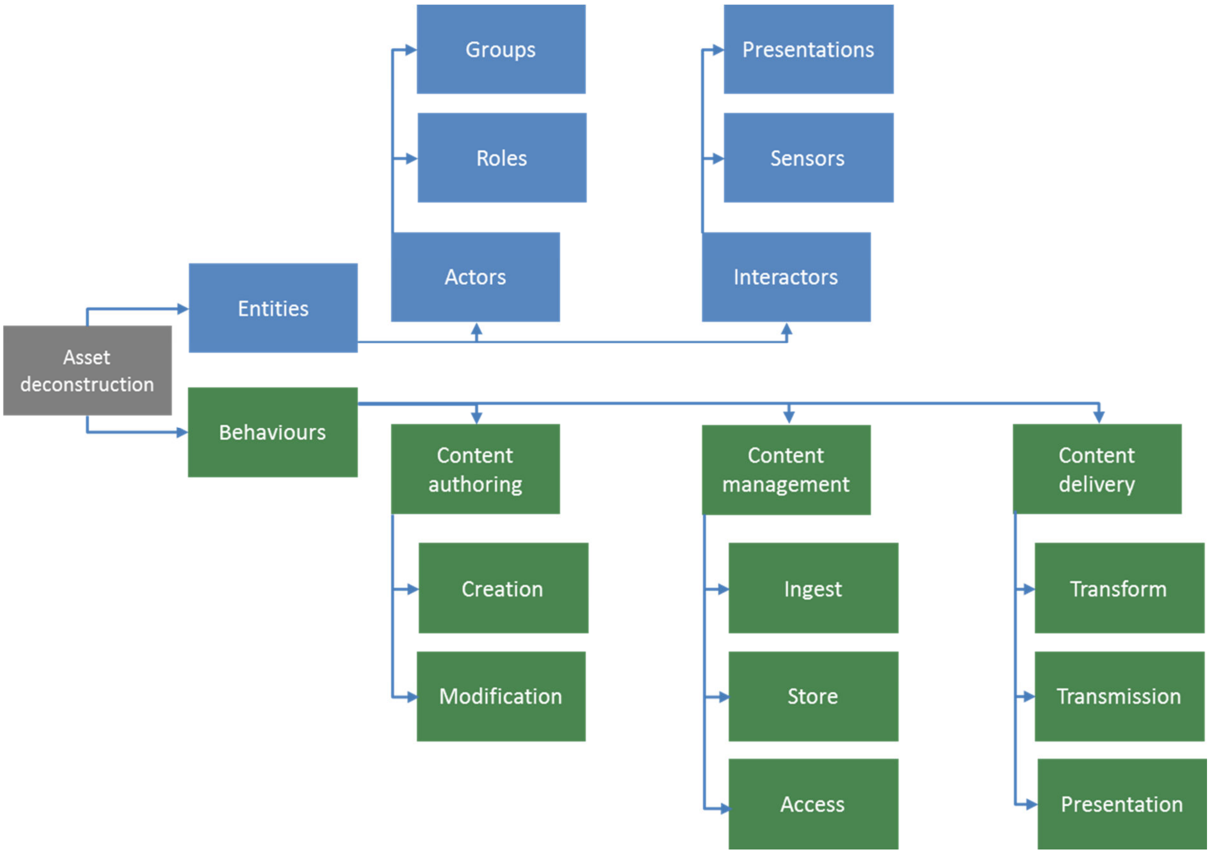


**Figure 6: Asset deconstruction (content lifecycle)**

#### *3.1.1.1. Entities: Actors (Roles and Groups)*

A useful point to start asset deconstruction is the identification of the (usually human) roles and groups that are associated with the asset; this description often leads to further information about other components. In a first pass decomposition, three main roles have been identified in Table 2.

**Table 2: Roles**

| Role | Description |
|---|---|
| AR content viewer | Request and view 2D/3D content |
| AR content manager | Control the ingest and delivery of 2D/3D content |
| AR content author | Create and submit 2D/3D content for ingest |

Potential activities associated with each of these roles suggest data for later parts of the analysis. For example, a content viewer will require some sort of viewing medium within which to perceive the AR content (see Section 3.1.1.2). Further groupings of users can also be considered, where a user adopts more than one role, see Table 3 for examples.

**Table 3: Groups**

| Group | Description | Role(s) |
|---|---|---|
| Passive end users | Users that only consume AR data | AR content viewer |
| Reactive end users | Users that both consume AR data and generate 2D content for ingest | AR content viewer & AR content author |
| Professional content authors | Users that only produce 2D/3D content | AR content author |
| Professional AR managers | Users that control the ingest and delivery of one or more AR content sources | AR content manager & AR content viewer |

In the grouping example shown above, individuals have been grouped according to the roles they could adopt within the envisioned AR scenario. Drawing up this relationship sets a foundation for further investigation into the content lifecycle analysis later.

#### 3.1.1.2. Entities: Interactors (Sensors and Presentations)

Within the structures related to the asset there exist a number of interactive artefacts (or 'interactors') that have an active role to play in a FMI scenario. Interactors are regarded as objects (or a system of related objects) that actively provide one or more sensing or presentation abilities. An obvious and concrete example of such an interactor is a mobile computing device - however it is essential that abstract capabilities of the asset are identified here rather than technology that encapsulates it. Sensing and presentation features for the AR scenario are provided in the tables below.

**Table 4: Sensors**

| Sensing capabilities | Description |
|---|---|
| Geo-location | Identifies a physical location using global positioning data. |
| Orientation | Identifies the orientation of an entity using polar data. |
| 2D Image capture | Captures an array of optical data (luminosity and chrominance) based on a viewing frame in physical space. |
| Alpha-numeric capture | Captures an alpha-numeric input event |

**Table 5: Presentations**

| Presentation modality | Description |
|---|---|

| capabilities | |
|---|---|
| 2D display | Displays an array of optical data (luminosity and chrominance) based on a discrete viewport in physical space |
| Audio rendition | Outputs an audio signal in one or more channels |

The deconstruction in Table 5 seems sparse, but it is important to note that this part of the analysis will link with the results of the analysis of content authoring (see Section 3.1.1.2) and delivery (see Section 3.1.1.10) to provide an overall description of end-user interaction with content.

#### 3.1.1.3. Behaviours: Content authoring (Creation)

Up to this point, the decomposition has focused on the structural components that contextualize the FMI scenario from an interactive, experiential point of view: i.e., a description of users and their environment and interfaces. Now an analysis of an orthogonal aspect of the same FMI scenario is considered: the dynamic or behavioural components of content generation, management and delivery. It may be useful for some analysts to approach this behavioural deconstruction by examining what users might perceive (via available sensors) and what actions they might carry out (via available effectors) in the FMI scenario. This is the approach adopted here; however, other points of view could be just as good as starting point.

Re-visiting the human roles described in Section 3.1.1.1 provides us with some good starting points for unpacking what AR content is generated and by whom.

**Table 6: Content creation**

| Creation capabilities | Content types | Created by | Presentations |
|---|---|---|---|
| 3D object | 3D models and textures | Professional content author | 3D rendering |
| 2D image | Images | Reactive end user; professional content author | 2D rendering |
| Alpha-numeric data | Free text, URL | Reactive end user; professional content author | 2D rendering |
| Environmental data | Geo-spatial data | | 2D/3D rendering |
| Meta-data | AR meta-data | | 2D rendering |

By enumerating the principal content associated with the AR asset and linking its generation with the groups identified earlier (see Table 6), we can start to reveal other characteristics that potentially tie the content lifecycle and the FMI capability landscape together. Here, this includes the capability to digitally re-create and present 3D models and textures (here, notionally assigned to a professional person who this skill) or the ability to add images and text to a virtual component. In the former case, the generative role taken on by the professional content author suggests content generation activities most likely to take place at the beginning of an EXPERIMEDIA content lifecycle. Reactive end users, on the other hand, may engage in a different kind of content generation during their interactive experiences with the AR asset: the type and method by which they generate content will be different from the former case.

#### 3.1.1.4. Behaviours: Content authoring (Modification)

Over time, an asset's content could be modified using methods that vary considerably depending on its situation within its lifecycle. It is important to make a distinction here between *what* forms of manipulation are possible for a particular content component and *how* that manipulation is carried out (which may or may not include some form of human interaction). At this stage of the deconstruction analysis, the type and nature of content manipulation is the primary focus.

**Table 7: Content modification**

| Modification capabilities | Content types | Modified by | Presentations |
|---|---|---|---|
| Spatial transform | 3D models | Professional content author | 3D rendering |
| Spatial transform; Colour space transform; Projection transform | 3D textures, Images | Professional content author | 2D/3D rendering |
| Serial modification, deletion | Geo-spatial data | Professional content author | 2D rendering |
| Serial modification, deletion | Free text, URL data | Professional content author; reactive end user | 2D rendering |
| Serial modification, deletion | AR meta-data | Professional content author | 2D rendering |

Table 7 outlines some common operations associated with the AR content which focus on the 3D model, geo-spatial, overlay and metadata components. It should be noted that these modification methods have been collected from all phases of the content lifecycle in the AR scenario; however each operation itself may (or may not) be limited to a specific phase. For example, spatial transformation on 3D model data or texture colour correction is only like be carried out by the professional content author in a generative phase that preludes the deployment of the AR system proper. In contrast to this, the serial modification (or editing) of text that is associated with a particular geo-spatial location is a type of content modification that could by executed by either a reactive end user or the professional content author.

#### 3.1.1.5. Behaviours: Content management (Ingest)

Many EXPERIMEDIA test-bed scenarios are likely include technologies that either require or directly implement some form of content management system (CMS). The AR scenario described above is a simple example of such a system: all of the content created either professionally or by end users will be stored in various databases (web content will be referred to via URLs) and accessed via a CMS interface. Requests for this content will be made by users who either submit their geo-spatial data to the CMS to retrieve 3D content or send new metadata (perhaps a short commentary on a building) to update the CMS.

Ingest capabilities of a particular asset (or the *dependency* on an ingest capability of that asset) represents a significant part of the interface that facilitates the processes supported by a CMS. Table 8 presents the content ingest capabilities of the CMS depicted in the AR scenario; each content type has one (or potentially more) data formats associated with it.

**Table 8: Content ingest summary**

| Ingest content capabilities | Data format |
|---|---|
| 3D content | OBJ/MD2 file format |
| 2D content | JPG file format |
| Geo-spatial content | WGS84 |
| Alpha-numeric content | UFT-8 |

Data formats associated with content ingest provide an important clue to the potential delivery mechanisms of the content (as well as the lifecycle phases during which ingest occurs). We have already established that the 3D model data will be created by the professional content author. These 3D models are likely to be developed using 3rd party software (such as 3D Studio Max) and digitally stored in a professional format within a *file*: in consequence, a file-based delivery of this content must be considered (see Section 3.1.1.9) as well as any necessary transformation of this data later on during the delivery process (see Section 3.1.1.8).

#### 3.1.1.6. Behaviours: Content management (Store)

The examination of content authoring and ingest described above have revealed several insights that directly reflect the available or required storage capabilities of the asset during the content lifecycle. Whilst actual storage of digital content may be provided an FMI asset, it is not uncommon for storage to be distributed among multiple sources that persist in various container types (file based; web based etc.).

**Table 9: Storage types and related content**

| Storage capabilities | Content |
|---|---|
| File system | 3D models |
| File system | 3D textures |
| Web universal resource | Images |
| Web universal resource | Text |
| Local database | Geo-spatial data |

| Storage capabilities | Content |
|---|---|
| Local database | URL data |
| Local database | AR meta-data |

In consequence, an asset may offer multiple storage container types that are linked together referentially via a central storage facility (typically implemented by a CMS and represented in Table 9 by the AR meta-data storage). The storage systems associated with each of the content types suggest further capability requirements that relate to access (see Section 3.1.1.7) and delivery (both to and from; see Sections 3.1.1.8 and 3.1.1.9).

#### 3.1.1.7. Behaviours: Content management (Access)

Many content management systems offer (or indeed mandate) qualified access to stored content based on some kind of user profiling. The rules that govern this access may apply in a number of respects, such as which content types can be operated upon, the scope within which content can accessed, and read/write privileges. To unpack capabilities of the asset, we consider *user authentication*, *access methods*, and *access qualifier* aspects.

**Table 10: User authentication**

| Authentication capabilities | Role |
|---|---|
| None | AR content viewer |
| User name & password | AR content manager |
| User name & password | AR content author |

In Table 10, the roles discussed in Section 3.1.1.1 have been mapped to a particular method for recognising and authenticating a user's access to the AR content via the CMS. Here, we illustrate this process with examples of general, unauthenticated access as a viewer of AR content and authenticated access via a username and password.

**Table 11: Access methods**

| Access capabilities | Role |
|---|---|
| Search, Read | AR content viewer |
| Search, Read, Write, Update, Delete, Archive | AR content manager |
| Write, Update | AR content author |

Access methods that are made available by the asset on the content that is stored are described in Table 11 - these have also been mapped to the roles associated with access to the content.

#### 3.1.1.8. Behaviours: Content delivery (Transform)

Some aspect relating to content delivery is likely to arise in many of the technical assets brought into the EXPERIMEDIA project. The term *delivery* in this deconstructive process is used to explore three important dimensions (*transformation*, *transmission* and *presentation*) through which an understanding of an asset's capability to integrate with other content related components is achieved.

Some description of content type and format was initially considered in Section 3.1.1.5 in the context of ingest. Over the course of its lifetime, data delivered to or from an asset may undergo changes to its structure in order to meet various processing demands (such as storage, physical transmission or presentation). An example of this includes (but is not limited to) the CMS depicted in the AR scenario described here. Both during ingest and in the course of access, content data may be modified to suit its immediate, next destination.

**Table 12: Delivery transformation**

| Transform content capability | Source format | Destination format |
|---|---|---|
| 3D model | OBJ/MD2 | File/URL reference |
| 3D model | OBJ/MD2 | ZIP of (OBJ/MD2) |
| 2D Image | JPEG | URL |

In the case of the 3D model data, during ingest the data is referred to by a local file or URL path (to reduce the storage footprint on the AR content store), whilst in the initial phase of transmitting the model to a remote viewer, this data is repackaged in a format suitable for transmission and rendering (see Table 12 for an overview).

#### 3.1.1.9. Behaviours: Content delivery (Transmission)

The process of actually moving content data from one asset to another is itself a type of capability that is commonly depended upon by many technical components but also sometimes a behaviour directly supported by some assets, particularly those relating to infrastructure, such as a networking technology.

**Table 13: Transmission**

| Transmission medium capability | Content types | Performance frame |
|---|---|---|
| Local disk storage transfer | Text, 2D/3D, URL data | Real-time |
| GPS carrier signal | Geo-spatial data | Real-time |
| IP network | Text, 2D/3D, URL data | Interactive |

Three delivery transmission scenarios are indicated in Table 13 that relate to the content lifecycle of the AR system as a whole. Performance frame categories here are informally defined as a complete transmission process that:

- Real-time: has little or no perceptible delay
- Interactive: has a discernible delay but does not significantly impact on task performance
- Background/off-line: has significant delay, impacts task performance and must be deferred

The first example illustrates the movement of data during the course of managing material contained within the AR CMS provided directly on the AR content server[6]. Here, the author works on the various materials (perhaps editing some of it using third party software) during which time data moves to and from the CMS via a disk controller in real-time. A modulated data signal from overhead satellites allow the AR display client to locate itself geographically in real-time; this second transmission example is a clear capability delivery dependency on an infrastructure. Finally, the primary content used by the AR system is interactively delivered across an IP network during the processes of creation and presentation.

#### 3.1.1.10. Behaviours: Content delivery (Presentation)

Tightly bound to those stages of the content lifecycle that require some form of human-computer interaction is the process by which the content is presented to the user. The presentation of this content will vary according to the context of its use.

**Table 14: Content presentation pairings**

| Presentation capabilities | Content type | Performance frame |
|---|---|---|
| 3D Rendering | 3D models and textures | Real-time/interactive |
| 2D Rendering | Images, free text, URL data, geo-spatial data, meta-data | Real-time/interactive |

For example, an AR content manager may wish to work with a logical view of the AR asset's contents (perhaps files and folders or distributed databases) with the purpose of selectively updating certain digital assets. This may not require any graphical rendering or the content itself but would be classed as 2D rendering. Conversely, the end users of the AR asset will require a direct (and real-time) rendering of the content data as graphics and text that aligns with the video imagery captured by them. Table 14 illustrates a number of such pairings - without binding these techniques to a specific technology.

### 3.1.2. Phase 2: Deconstruction through the experiment lifecycle

It is expected that some EXPERIMEDIA partners will bring technical assets that are capable of delivering insight through the application of experimental means. Enabling assets in this domain could include software to measure network performance or capture user experience; data

[6] Presented here for the sake of example, a remote connection to the AR CMS is quite likely.

monitoring services; data management services and numeric analysis tooling. In the second phases of the capability deconstruction, an asset is appraised (where it is appropriate to do so) along two dimensions in the experimental lifecycle: data types and methods. This deconstruction is presented in Figure 7, where it is immediately apparent that there are significant areas of cross-over with the content-based disassembly (see Figure 6), particularly with respect to data management and delivery. This is not surprising, since during the course of an experiment, many of the behaviours required to effectively manage content are also applicable to experimental data. Indeed, it may often be the case that some of the content authored and delivered during an experiment may also act as experimental data (or be very closely related to it). For example, in a scenario where social network content is generated, this data may act as a source for a social analytics process, the metrics of which must be ingested, managed and presented to the experimenter.

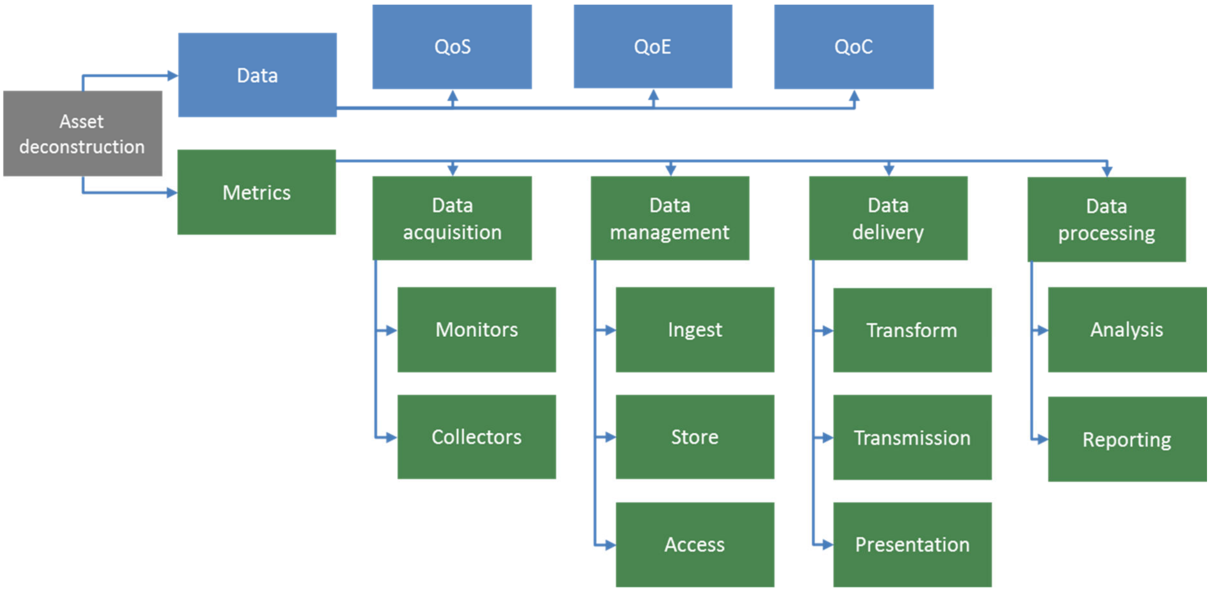


**Figure 7: Asset deconstruction (experiment lifecycle)**

Using a similar approach to that used in Section 3.1.1, we will explore a sub-set of the assets described in the Schladming driving experiment (Section 5.1) to explore their capabilities from an experimental point of view.

### 3.1.2.1. Data: QoS, QoE and QoC

From the perspective of the experimental lifecycle, three main dimensions of data are considered in the FMI capability framework offered by EXPERIMEIDA, these are: *quality of service* (QoS), *quality of experience* (QoE) and *quality of community* (QoC). Each of these dimensions host a wide variety of potentially useful data types which themselves may be generated by a number of sources (dependent on the nature and setting of the experiment itself). Quality of service data typically reflects direct, objective measurements of physical characteristics of the environment (such as the geo-spatial location or temperature of a device) or the performance characteristics of software or hardware components (for example, the number of video frames presented per second). To understand the experiential aspect of an EXPERIMEDIA test-bed, typical QoE data includes quantitative measures (such as interaction logging or task monitoring) and qualitative measures (often self-report data, such as a psychometric scale response, elicited from the user via a human-computer interface). Finally, QoC measurements may also reflect both quantitative and qualitative aspects of a community over a specific time frame (such as the relative growth of a community size or the composure of a community based on sentiment analysis).

**Table 15: Example QoS, QoE and QoC metrics**

| Metric | Dimension | Description |
|---|---|---|
| Virtual object requests per second | QoS | The number of content requests made by the AR viewers on the AR content provider per second |
| Virtual scene render FPS | QoS | The number of rendered virtual scenes presented to the user per second achieved by the rendering view |
| Number of content request cancellations per user session | QoE | The number of cancel actions made by each user whilst requesting AR content data |
| 'Relevancy' rating of content | QoE | A subjective rating of the degree to which users felt the AR content presented to them was relevant. |
| Number of new users posting UGC content per day | QoC | A count of the number of users, who have never before posted content to the AR content server, do so for the first time |

Above, Table 15 illustrates some of the potential measurement types that could be collected to shed light on the performance, usage and user experiences during the execution of an experiment in the Schladming scenario. Note that the actual data source for these measures is not bound to the type here - this coupling is made after data acquisition capabilities are understood.

#### 3.1.2.2. Data acquisition: Monitors and collectors

The purpose of experimental monitors and collectors is to gather data from a variety of sources - some of these will be extracted directly from experiment specific data generating components (such as network traffic monitor) whilst others will be collected indirectly from data generated within the content lifecycle. In either case, a metric should be associated with the data gathered for experimental analysis purposes.

The encapsulation of raw data sampling and metric generation is considered here as a 'monitor', see Table 16 for examples.

**Table 16: Possible AR monitors**

| Metric monitoring capability | Metric output | Content source | Format |
|---|---|---|---|
| QoS | Virtual object requests per second | Content access | Integer value |
| QoS | Virtual scene rendering FPS | Content delivery | Integer value |

Some EXPERIMEDIA assets will not perform all the functions of a monitor but nevertheless provides the means by which (raw) experimental data can be gathered (collected) for subsequent processing elsewhere. Examples of these include QoE sampling and SNS interfaces (see Table 17): in both cases, content creation capabilities form the basis of a data sample. Each logged response (see Section 3.1.2.3 for further discussion of data management) is not in itself sufficient to produce a metric, however, subsequent access and processing of the data will render such a measure.

**Table 17: Potential AR collectors**

| Metric collecting capability | Input for metric(s) | Content source | Format |
|---|---|---|---|
| QoE | Average 'relevancy' rating of content | Content creation | Key-value pair |
| QoC | Number of new users posting UGC content per day | Content creation | XML data |

### 3.1.2.3. Data management: Ingest, store and access

Many of the components relating experimental processes and data management are shared by capabilities described in the content lifecycle deconstruction (see Sections 3.1.1.5-3.1.1.7); these aspects should also be considered as they relate to data, monitors and collectors, but this exercise will not be repeated here. Overlaps in this area of analysis may result in cross-cutting concerns between the content and experimental lifecycles where the experiment may wish to collect raw content data from a CMS store. A result such as this may then provoke a refinement of the content based capability deconstruction, to extend access descriptions for example.

### 3.1.2.4. Data delivery: Transform, transmission and presentation

A similar, iterative approach may also be required for transformation, transmission and presentation analysis, examples of which can be found in Sections 3.1.1.8-3.1.1.10. Again, this exercise will not be repeated here; however, further presentational capabilities may emerge as a result of considering what, if any, reporting behaviour (see Section 3.1.2.5) an asset offers the experimenter. Table 18 presents a few examples of well-known visualisation methods that could be applied to the metrics examples discussed in the context of this analysis.

**Table 18: Report presentation examples**

| Presentation capabilities | Analysis method support |
|---|---|
| 2D bar-chart | Frequency |
| Box plot | Variance |
| Heat map | Correlation |

#### 3.1.2.5. Data processing: Analysis and reporting

Data that has been delivered to a data management system during the course of an experiment may require subsequent processing before it is useful to the experimenter. In the case of raw data delivered by a collector, some processing will be required to derive suitable metrics: the QoE data collector described in Section 3.1.2.2 is one such example where sample set of bi-polar responses must be processed (for each user) to generate a point along an attitudinal dimension.

**Table 19: Analysis examples**

| Analysis capabilities | Input data types | Output metric |
|---|---|---|
| Frequency | Numeric/Category | Numeric value/category |
| Variance | Numeric value | Average; standard deviation |
| PCA | Numeric value | Correlates (matrix) |

Data processing activities consider here relates to analytical methods required to generate some insight into the nature of the raw data sets; Table 19 presents just a few examples. Finally, an asset's reporting capabilities can be considered as an aggregation of an analysis type and a specific QoS, QoE or QoC related metric generated within a particular time frame.

**Table 20: Example metric reporting**

| Report capability | Metric dimensions | Analysis | Presentation | Performance frame |
|---|---|---|---|---|
| Time-based frequency analysis; (New posts/day) | QoC | Frequency | 2D graph | Interactive /background |
| Spread of average rendering speed; (Mean & STD FPS) | QoS | Variance | Box plot | Interactive |
| Driving experiences during content interaction; (Pearson R matrix) | QoE | PCA | Heat map | Background |

The performance frame may vary depending upon data processing factors (including data volume, computation and delivery efficiency) and sampling issues (for example, some QoE or QoC data sets may take minutes, hours or days to gather).

### 3.1.3. Phase 3: Capability specification

An analytical process that addressed the components of both content and experimental lifecycles was provided in the previous section using a subset of the assets from the Schladming driving experiment as an example. Analysis of assets using this technique yields a collection of capabilities (and related entities) that can be added to the FMI capability landscape as a whole.

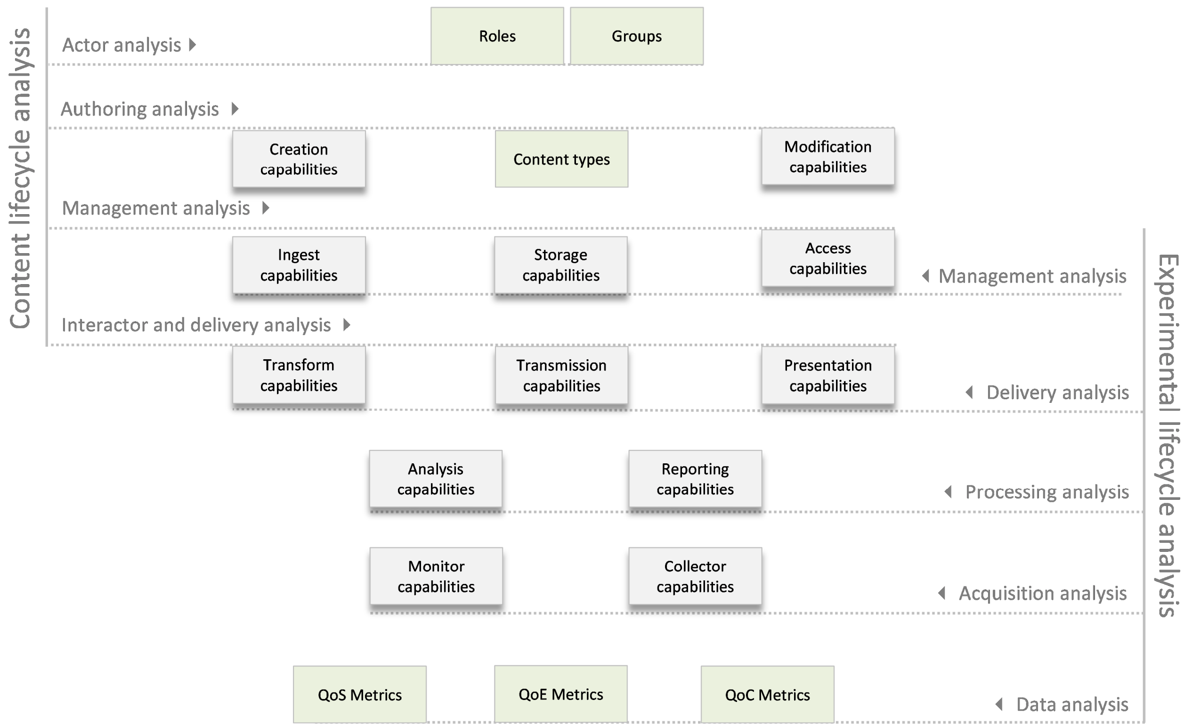


**Figure 8: Capability analysis result**

Figure 8 presents the outputs of each phase of the analysis in terms of the capabilities and other primary information entities (role, groups, content types and metrics) it may generate. It is expected that for any given asset, only a relatively small subset of all the capabilities represented in the entire FMI capability landscape will be directly implemented. For every one of the asset's implemented capabilities, each is likely to be dependent on other capabilities found elsewhere in the landscape. Some of the information contained in the tables presented in Sections 3.1.1-3.1.2 refers directly to capabilities and entities described in other parts of the analysis; this forms a dependency graph, depicted in Figure 9.

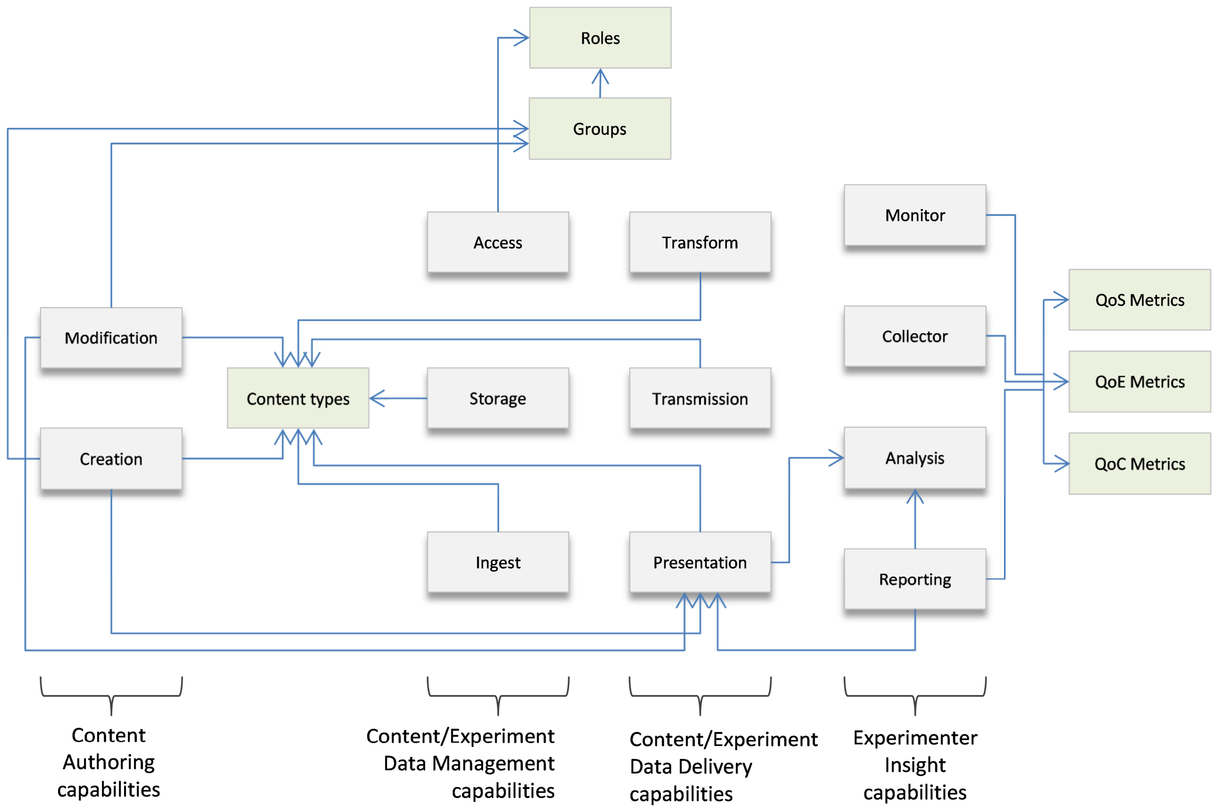


**Figure 9: Capability dependency graph**

The outcome of the each analysis exercise is therefore a set of capabilities that an asset directly provides (with attributes particular to its type) and dependencies on other capabilities that it may or may not also offer.

#### 3.1.3.1. *Capability specification and positioning*

At the end of the capability analysis process of the asset brought to EXPERIMEDIA will be a set of specifications that have been derived from one or more of its technical components. The proposed format for these specifications is set out as follows:

 

**Table 21: Capability specification format**

| **Asset name** | |
|---|---|
| Capability ID | |
| Capability type | |
| Capability sub-type | |
| Capability description | |
| Capability attributes | <table><tr><th>Attribute</th><th>Value</th></tr><tr><td></td><td></td></tr></table> |
| Capability dependencies | <table><tr><th>Type</th><th>Sub-type</th><th>ID</th></tr><tr><td></td><td></td><td></td></tr></table> |
| Technical component | |

Each capability identified should be assigned a unique identification label (based in part on the asset name) - this identifier may be used by other capability specifications in describing their dependencies. Two capability types are also defined in the specification; the first positions the capability within the overall EXPERIMEDIA capability landscape at level 2 (see Figure 10), whilst the second type provides specialisation (see Section 7 for example specialisations). A constrained set of example specifications, based on the AR client device from the analysis example described above, is presented in Section 8. A brief description within each capability specification should outline the nature of the capability whilst specific attributes (such as content format handling or performance frames) will qualify the specification further.

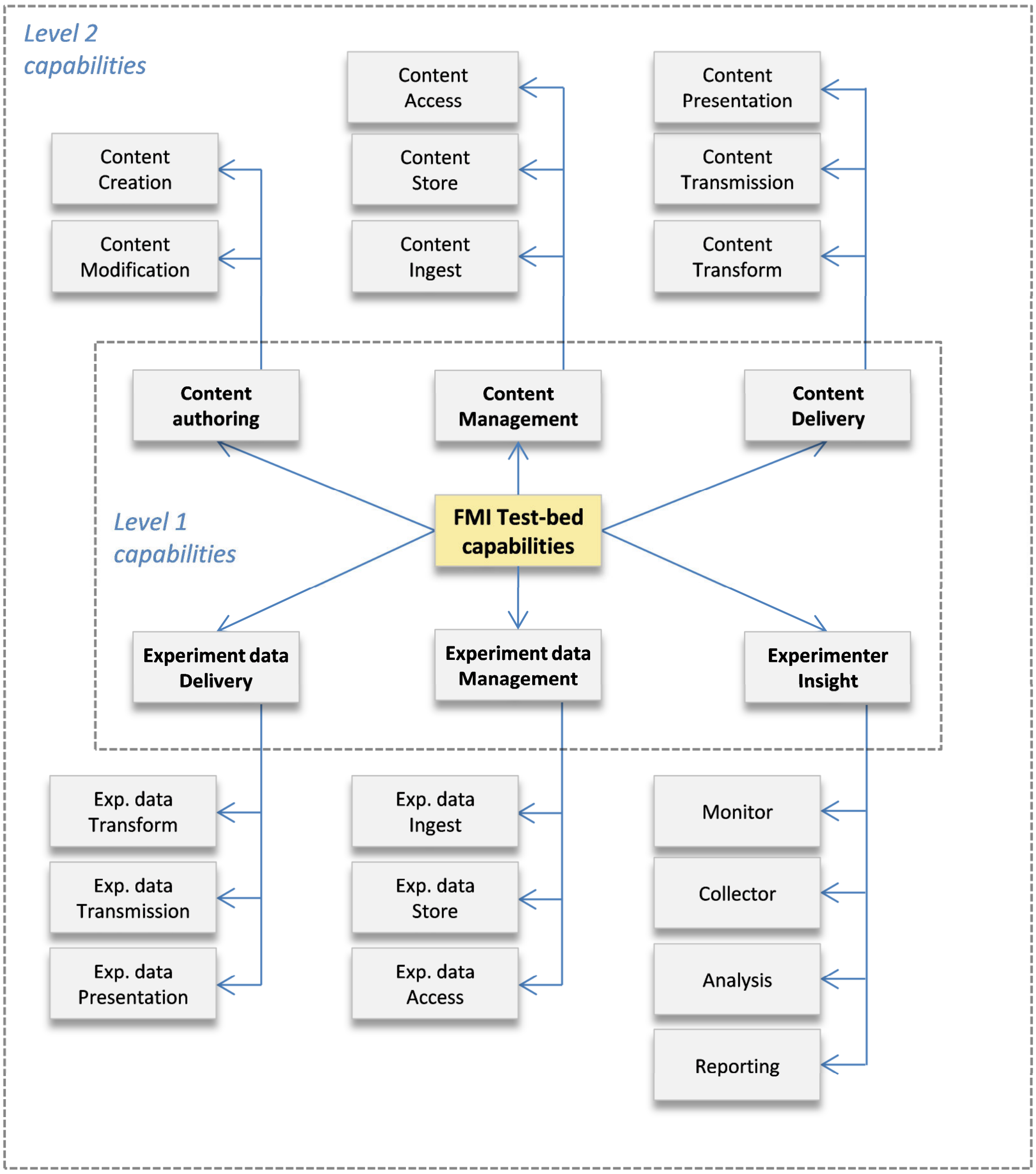


**Figure 10: Level 1 and Level 2 FMI capability landscape**

Many of the specifications will refer to other capabilities in their definition (content creation instances may refer to delivery presentations, for example). These may or may not be provided by the asset - if a dependency is provided by the asset, its ID should be provided. Finally, the technical component (or sub-component) that each specification derives from is provided: this provides a mapping from the capability abstraction to the realised technical implementation.

### 3.1.4. Summary

In Section 3.1, a FMI capability model has been set out indicating how technical assets brought to the EXPERIMEDIA project can be specified in terms of their capabilities for the purpose of designing a test-bed with content and experimental lifecycles in mind. An analytical process was provided to demonstrate how such capabilities can be derived and a means by which these descriptions can be captured and positioned within an overall framework presented.

# 4. Technology Model

The objective of this section is to present how the first technical assets brought to the EXPERIMEDIA project are to be organised and integrated. Readers are presented with four super-structures that address content and experimental data creation, processing and management (see Sections 4.1.1-4.1.4). Components within each collectively provide the functionality required to deliver a super-structure's external services to clients via interfaces. In some cases, the super structure's overall functionality can also be enhanced through an extensible interface based 'plug-in' mechanism. Integration with infrastructure with respect to these super-structures is discussed in Section 4.1.5, while security and privacy best practice is discussed in Section 4.1.6.

## 4.1. Content Lifecycle Technologies

Three principal technical models have been established to support content authoring, management and delivery offered an EXPERIMEDIA facility for this architecture blueprint include:

- Social Content Component (SCC) model
- Audio Visual Content Component (AVCC) model
- Pervasive Content Component (PCC) model
- Experiment Content Component (ECC) model

Each component model focuses on a different content aspect within the FMI and technologies supporting the lifecycle of the specific content. A key aspect for EXPERIMEDIA is to provide tools and services that support the seamless mixing of different content types in the delivery of user experience where the content lifecycles are implemented within separate systems. Social, Audio Visual and Pervasive content present the content types supported by generic technology assets to which we add an Experiment content model supporting all data related to the setup, execution and analysis of experiments.

Each model encapsulates components drawn from the technical repository developed to support the first driving experiments. The architecture depicted in each model indicates high-level relationships between the internal components; the services the model provides to external clients; and also which services are required by the model to support its content related function. Provision and usage of services is represented using the UML interface nomenclature: a ball ended arc and 'open cup' ended arc representing provision and usage respectively. By convention for these models, infrastructure requirements are always presented at the bottom of the diagram. Some of the usage interfaces will also be stereotyped by an *<extension point>* - this explicitly indicates how additional further technical components can be added to the model to enhance its functionality in the future. All the models are illustrated in Figure 11.

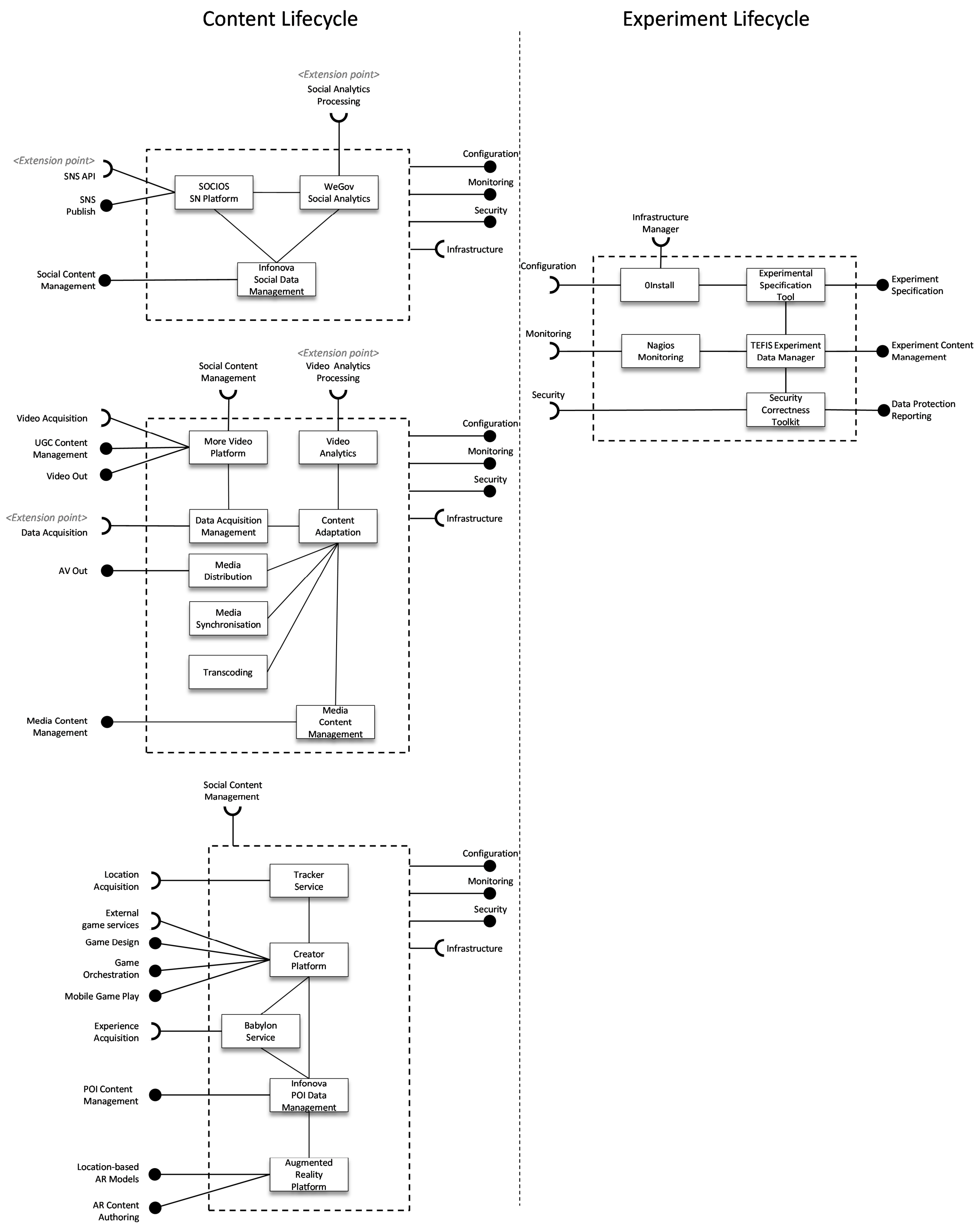


**Figure 11: Overview of the component models**

## 4.1.1. Social Content Component Model

The social technology model addresses the requirement to gather, manage and analyse data that is generated on social networking sites during the course of an experiment, see Figure 12. Internally, the *SocIoS SN* platform provides a gateway to enable access to a wide range of different social networks - providing read access and content publishing capabilities. A framework for generating online community metrics will be enabled by the *WeGov* component.

To do so, the *WeGov* based technology will use the *SocIoS* platform to access data and an extensible range of social analytics processing components to execute the analysis. Various data accessed and generated by both the *SocIoS* and *WeGov* components should be persisted by *Infonova* social data management. The implementation of the Social Content Technology model is known as the Social Content Component (SCC).

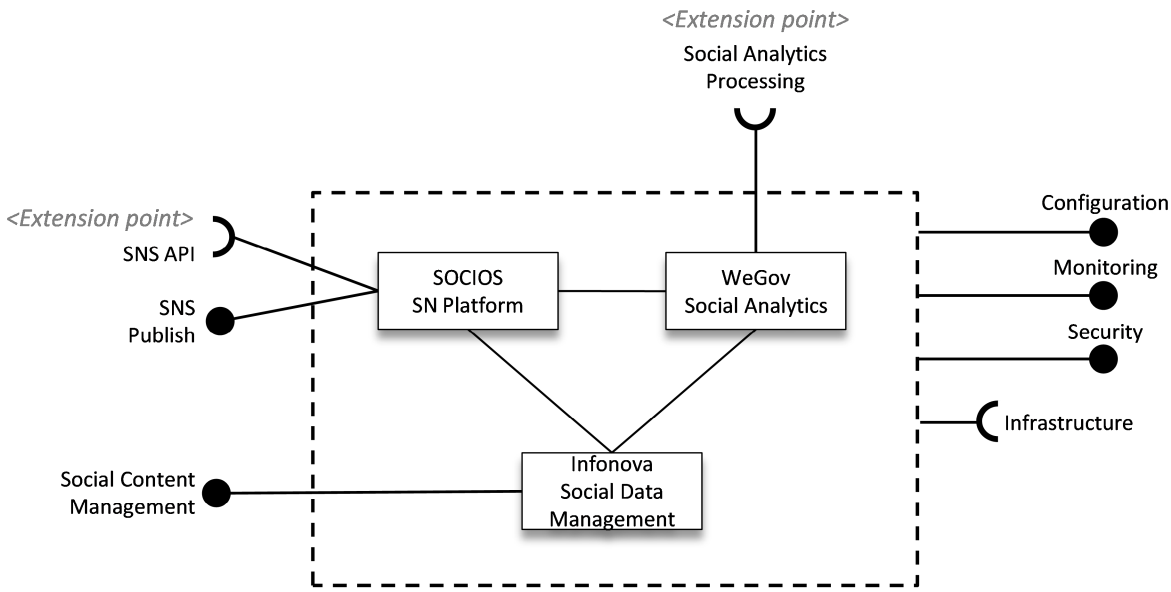


**Figure 12: Social content component model**

The integrated social technology model will realise a number of important capabilities that will be made available for experimenters to use for a complete experimental platform. A social content management interface will offer the means by which social network content can be accessed by other components. A monitoring service, based on the internally managed social analytics functionality, will also be exposed to users of the model to provide quality of community and other related metrics. Finally, a security interface and specification will be offered so that technical clients using the social technology model will only gain access to content via secure means.

### 4.1.2. Audio Visual Component Model

Two of the most common content types in an EXPERIMEDIA test-bed scenario are expected to be audio and video (AV), both of which will be contained in a variety of formats and containers (files and streams being most common). The video and audio technology model is primarily focussed on ingest, transformation and re-distribution of this content; storage and delivery related to this data is not directly supported here but is instead a dependency of the model (see Figure 13).

Initially within the model, two components will be deployed which will deal with in-coming and out-going video and audio functionality. Both the *MoreVideo* platform and *ATOS streaming* platform will depend upon AV acquisition technologies (such as a digital camera and MPEG transport stream delivery infrastructure) to act as an ingest source (depicted as interface dependencies in the model). Content reaching the *MoreVideo* platform can then be managed (through an editing process) or transformed through a transcoding process (via *ATOS streaming*). Internal AV handling components will provide a method for accessing the AV data they have processed (via an exposed *Video Out* interface). Additional AV content management services, provided via an exposed interface will also be visible to clients, where the implementing technology supports it.

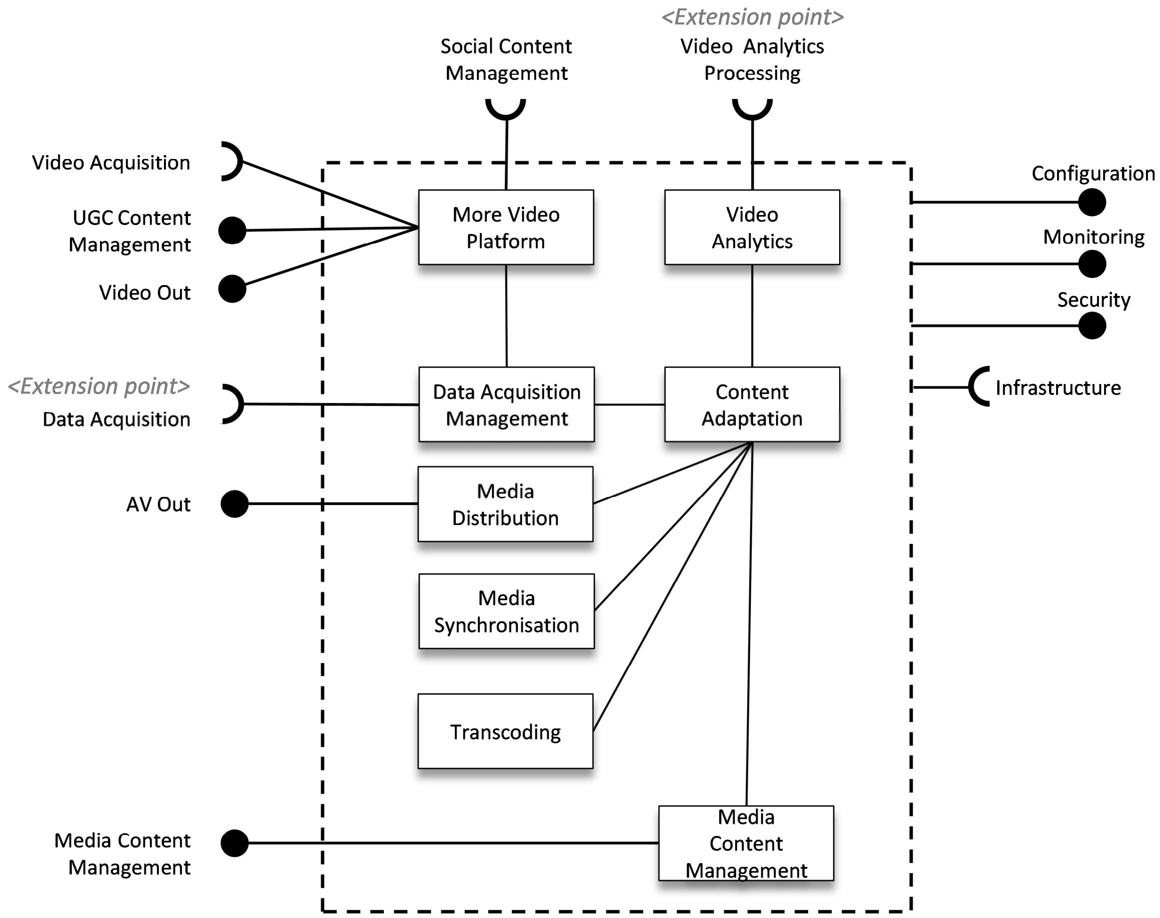


**Figure 13: Audio visual component model**

Enhanced behaviour of the video and audio technology model includes an extensible range of video analytics processing (providing, for example, the ability to track objects in a video stream) and the ability to synchronize meta-data with video data (also extensible). Further enhancements to AV content ingest should also be possible for internal AV management systems (such as the *MoreVideo* platform) where content from a social context (provided by the social content technology) could be used. Monitoring capabilities relating to the performance of the AV technologies in operation will offer the experimenter metrics for evaluation, whilst a security interface will control access to certain media sources (for example, from a social network).

### 4.1.3. Pervasive Content Component Model

A key element of an EXPERIMEDIA test-bed is the ability to locate and monitor users within the experimental environment. In addition to this functionality, there may be some situations where a selected range of elements within the environment need to be controlled during the course of an experiment. Together, these two important requirements give rise to the third model described here: the pervasive content technology model, see Figure 14. The implementation of the model is known as the Pervasive Content Component (PCC).

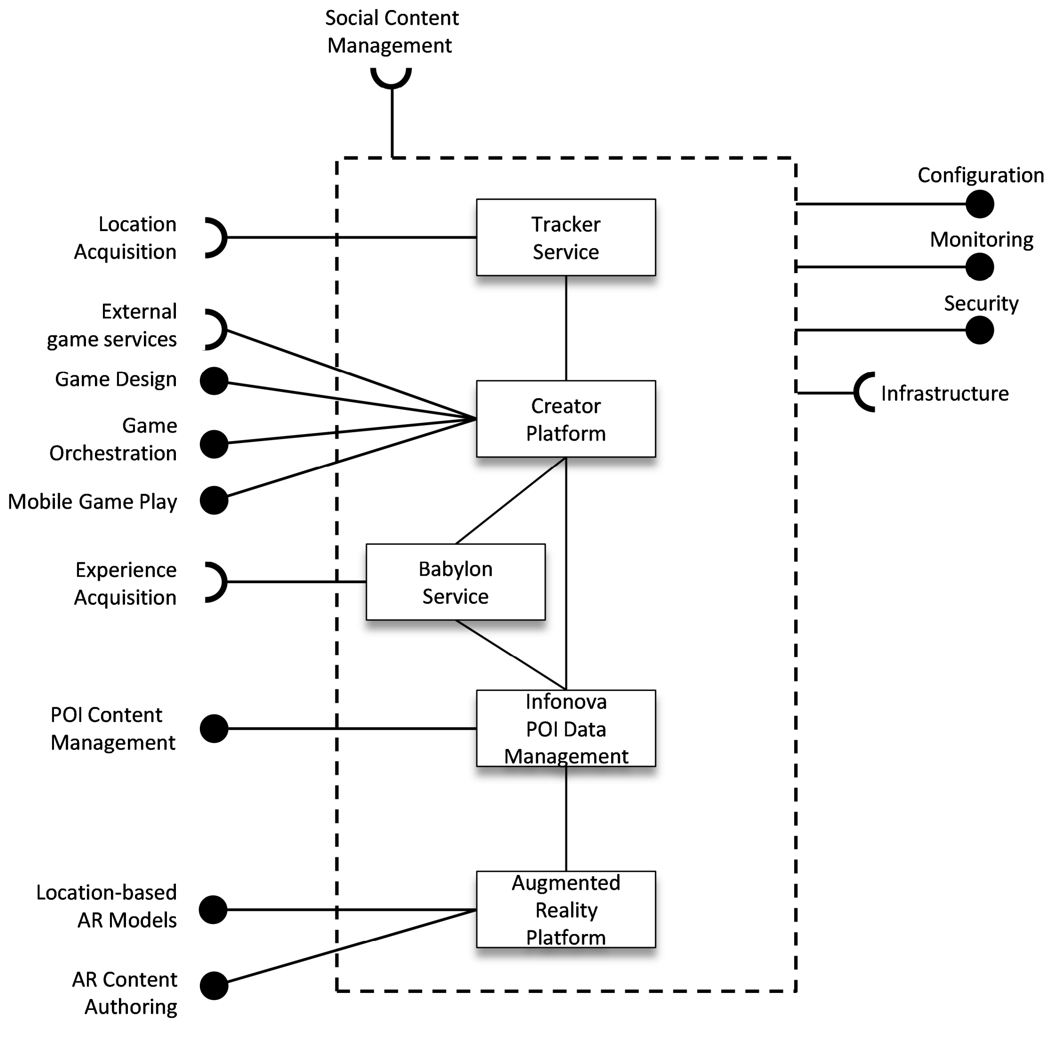


**Figure 14: Pervasive content component model**

The pervasive technology model contains components that collectively gather data about a user's physical location, quality of experience, points of interest and interactions. Physical location is used in both the context of tracking a user's location (*Tracker Service*) and also as a means by which AR-based content can be selected for delivery and user generated data (such as points of interest or social content) can be mapped to a spatial location (*Augmented Reality platform*). Access to all interesting spatial locations and related content is then exposed for clients of the pervasive content technology model using the POI content management interface. User experience is captured by the Babylon service (deployed on mobile devices): the logged data of which (along with other user or location based data) can accessed or managed by other components.

A degree of control over the environment in which users interact is provided in this model by the Creator platform. Creator offers support for pervasive game design and orchestration through an extensible mechanism by which connected computing devices in the environment can be manipulated centrally by a games master. This toolkit will be as the basis for provisioning more advanced levels of control over an extensible range of interactive devices in the experimental space. As with the previous content technology models, monitoring and security interfaces will allow clients to activity monitor and qualify access to pervasive content related metrics (such as QoS and QoE measures).

### 4.1.4. Experiment Content Component Model

Experimentally driven research and system testing is an essential element of a FIRE facility to ensure efficient operations and robust/accurate results. EXPERIMEDIA adopts an information centric view on the experiment lifecycle focusing on highly instrumented FMI technology components that deliver the necessary QoS and QoE data to experimenters so they can analyse the behaviour of technical systems in relation to user experience. Experiment content model offers three main interfaces to Experimenters: experiment specification, experiment content management and data protection reporting. An Experimenter uses an Experiment Specification Tool to describe the system requirements. We expect to consider infrastructure and FMI application specification separately in the definition to leverage previous investments in experiment descriptors (e.g. BonFIRE).

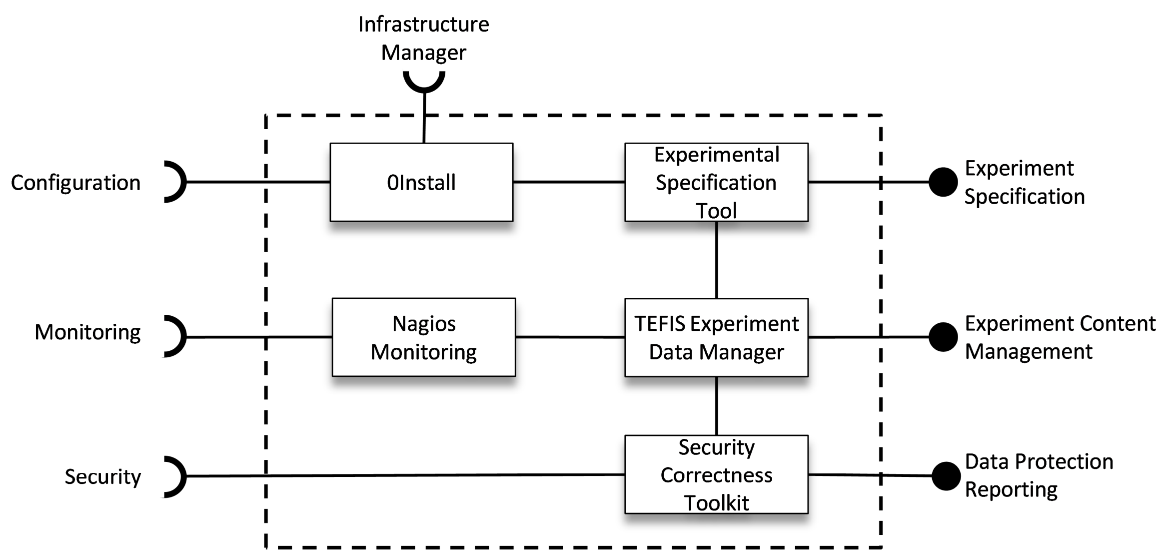


**Figure 15: Experiment content component model**

Experiment definition drives software packing, distribution and installation which are supported through 0install. 0install is a decentralised cross-distribution software installation system available under the LGPL that allows software developers to publish programs directly from their own web-sites, while supporting features familiar from centralised distribution repositories such as shared libraries, automatic updates and digital signatures. 0install does not define a new packaging format; unmodified tarballs or zip archives can be used. Instead, it defines an XML metadata format to describe these packages and the dependencies between them. A single metadata file can be used on multiple platforms (e.g. Ubuntu, Debian, Fedora, OpenSUSE, Mac OS X and Windows), assuming binary or source archives are available that work on those systems. Data management is supported by the TEFIS Experiment Data Manager (EDM) which builds on the iRODs distributed file system and extends this to provide a metadata schema and repository for FIRE. The EDM supports the storage, access, navigation and search of experiment content and integrates directly with monitoring systems such as Zabbix and Nagios.

Finally, an important element of EXPERIMEDIA is demonstration of legal compliance of experiments. As such we expect to adapt a security correctness tool (e.g. SAM) to allow for the assessment of security countermeasures configured to protect different types of content included in an experiment. The content centric view of EXPERIMEDIA ensures that data is treated as a first class asset allowing for the explicit development of policies and controls for data confidentiality and access. The Experiment content model uses three interfaces from other content models: Configuration, Monitoring and Security.

### 4.1.5. Cloud Infrastructure Integration

Infrastructure relates to the compute, storage, networking, sensor and actuator technology necessary to run FMI applications and services described above. Each of the Content components depends on infrastructure for its operation and an experimenter must be able to setup and control infrastructure required by their experiment.

Each of the EXPERIMEDIA venues offers infrastructure for use in experiments which is described in deliverable D3.1.1 First Infrastructure and Software Assets Inventory. In some cases the infrastructure is currently bespoke and developed for a targeted application (e.g. Tholos). EXPERIMEDIA will explore how such infrastructure can be "opened" through standardised interfaces so that the capabilities are available for a wider variety of applications. In other cases the available infrastructure is more general purpose (e.g. virtualised hardware, 3G/4G networks) and here EXPERIMEDIA will use standard API specifications offered by such technology.

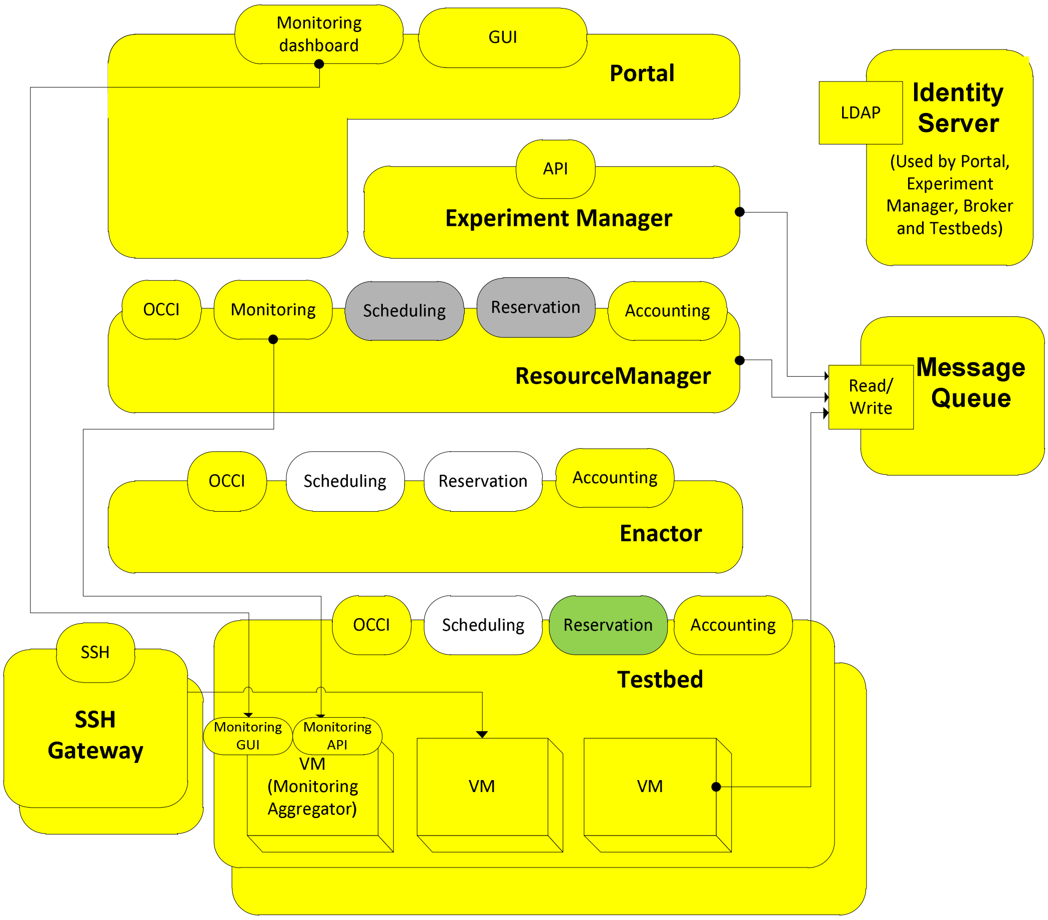


**Figure 16: BonFIRE Architecture © BonFIRE Consortium**

Special attention will be made towards the integration of cloud infrastructures including controlled networking. For venues such as Schladming the use of cloud infrastructure is important to support the up-scaling and down-scaling of infrastructures for dynamic communities attending live events. Here's EXPERIMEDIA expects to work closely with the EC BonFIRE project that is specifically developing a FIRE facility for multi-site clouds and controlled networking. The BonFIRE architecture and technology model is expected to be integrated directly with EXPERIMEDIA throughout the experiment lifecycle. BonFIRE offers EXPERIMEDIA many benefits including:

1) Infrastructure abstractions and experiment methodology: the Open Cloud Computing Interface (OCCI) abstractions and experiment descriptors for infrastructure can be used to describe and provide access to infrastructure necessary for FMI experiments. EXPERIMEDIA should use rather than replicate this work.
2) Cloud management technology: the Open Nebula cloud management software provides standardised access to cloud resources. The software is available on Apache 2 license and could be deployed directly at EXPERIMEDIA venues for hybrid cloud and cloud bursting scenarios
3) Performance experiments for infrastructure aspects: the BonFIRE facility services could be used to ensure that the FMI system will scale and perform appropriately prior to deployment at the venues. This could be achieved using the Virtual Wall or two BonFIRE sites. This is

an example of a longer experiment lifecycle where one lab based FIRE facility is used before rolling out the system to users in pilots at EXPERIMEDIA venues.

4) Up-scaling and downscaling for Schladming Venue: BonFIRE is not a production cloud but if an experiment wanted to explore horizontal/vertical scaling as part of the experiment then BonFIRE could be a suitable cloud infrastructure. If however it's just computational resources that are necessary then a commercial cloud such as Amazon would be more appropriate.
5) Controlled GEANT networking for CAR Venue: BonFIRE is developing architecture for dynamic allocation of Bandwidth over GEANT using the AutoBAHN software. The development of this capability as part of the experimental process and integration of controlled networking with cloud management software such as Open Nebula could be used directly by EXPERIMEDIA

### 4.1.6. Security and Privacy

EXPERIMEDIA must adopt security and privacy best practice associated with experiments using the facility for legal compliance. The detailed regulatory requirements are outlined in the EXPERIMEDIA deliverable D5.1.2 Ethical, legal and social framework and each experiment will be required to complete the Privacy Impact Assessment described in D5.1.1 First EXPERIMEDIA methodology. In general the principles can be summarised below:

- minimise the collection and processing of personal data and apply anonymisation techniques to remove the ability to identify individuals where possible, acknowledging that EXPERIMEDIA cannot just use fully anonymous data as some applications involve billable, personalised services or incentivised service contributions
- protect two types of data: 1) service data including user queries which are often trackable (correlatable with an individual) and even traceable (to a specific individual) and (2) sensor data collected from users to provide services (e.g. images/voices of people).
- only store user profiles with consent and only for the purpose and lifetime of experiments with no commercial exploitation of user profiles within the lifetime of the project

EXPERIMEDIA must adopt technical measures and processes necessary to minimise these privacy risks. From an architectural perspective EXPERIMEDIA will create FMI systems that consist of different component technologies. Each of these technologies will have different trust models and technical solutions/protocols for secure data storage, encrypted transfer, controlled and auditable access for different classes of data distributed over communication channels.

EXPERIMEDIA does expect every technology to adopt a single security solution but does expect each component to describe a supported security model for different types of content. Taking Social Content for example, SocIoS preserves the level of security offered by SNS API (e.g. the authentication process is controlled by the SNS themselves. Encrypted connections are always used for data transmission, while no data is currently stored by SocIoS. For Experimental Content, TEFIS uses iRODS (https://www.irods.org/) to store and manage data. Part of the data management is access control, and this is supported by basic authentication mechanism for users is username / password authentication, and each data object is protected by permissions, which determine who can do what to which data object. There is a hierarchy of permission for each data object: "read only", "read/write" and "own". "Read" and "read/write" are obvious, but "own" means read/write permissions plus the authority to give permissions to other people.

The challenge for the detailed design phases is to develop security models for each Component (SCC, AVCC, PCC, and ECC) that are:

- consistent with the capabilities of underpinning technologies, and where necessary extend and adapt
- able to be aggregated with other Components whilst preserving the security properties
- describable in terms of a formal model that can be assessed to ensure that the security properties of an experiment are acceptable

EXPERIMEDIA will use the developed component security models to verify security properties about the FMI systems. This will be achieved using the SAM Tool[7]. The tool allows experimenters to assess what behaviours must be assured in the components they own and what behaviours the experimenter requires of other parties. Without the security model we are relying on the programmers' and administrators' intuitions. In such situations an experimenter would never know whether it is safe to grant access to anything or anyone, or if to do so would contravene legal requirements. Let alone be able to report back to auditors that the correct countermeasures had been put in place.

[7] http://serscis.eu/sam/introduction.html

# 5. Example Architecture Scenarios from Driving Experiments

In this section we provide initial architecture scenarios for the driving experiments demonstrating how each both uses and drives the development of generic EXPERIMEDIA components. For each experiment we describe an experiment summary, the functional components (experiment and facility), the relationship to the content lifecycle, experiment data, technology considerations, and deployment considerations. At this stage of the project lifecycle, the planning and design of the driving experiments has only just started and therefore statements on experiment data (e.g. QoS, QoE), technical development and deployment remains at a high level. Detailed architecture for each experiment will be developed and distributed in further EXPERIMEDIA deliverables

## 5.1. Schladming

Schladming as a tourist place depends on visitors and their satisfaction. The driving experiment in this venue will therefore aim at providing technology that improves the visitor experience. As part of the experiment a mobile application will be created which allows visitors to experience the region and its activities in a modern and innovative way. The app will provide information to visitors on their mobile phone when they need it and also make use of augmented reality (AR) features as of the possibility to improve the quality of service using the social media reporting, rating and emotional feedback interfaces. The driving experiment comprises the mobile client software as well as the data provisioning backend including content lifecycle management and analytical usage tracking along with the interactive location and service based evaluation of delivered data.

### 5.1.1. Functional Components

For the experiment several building blocks need to be integrated as shown in Figure 17. The experiment drives the facility development by bringing together social and pervasive augmented reality content through mobile applications targeting the Schladming visitor experience. The experiment aims to validate and verify the integration of generic components supporting Social Content (SCC) and Pervasive Content (PICC) and how this enhanced contextual information can provide personalised delivery and collective experiences.

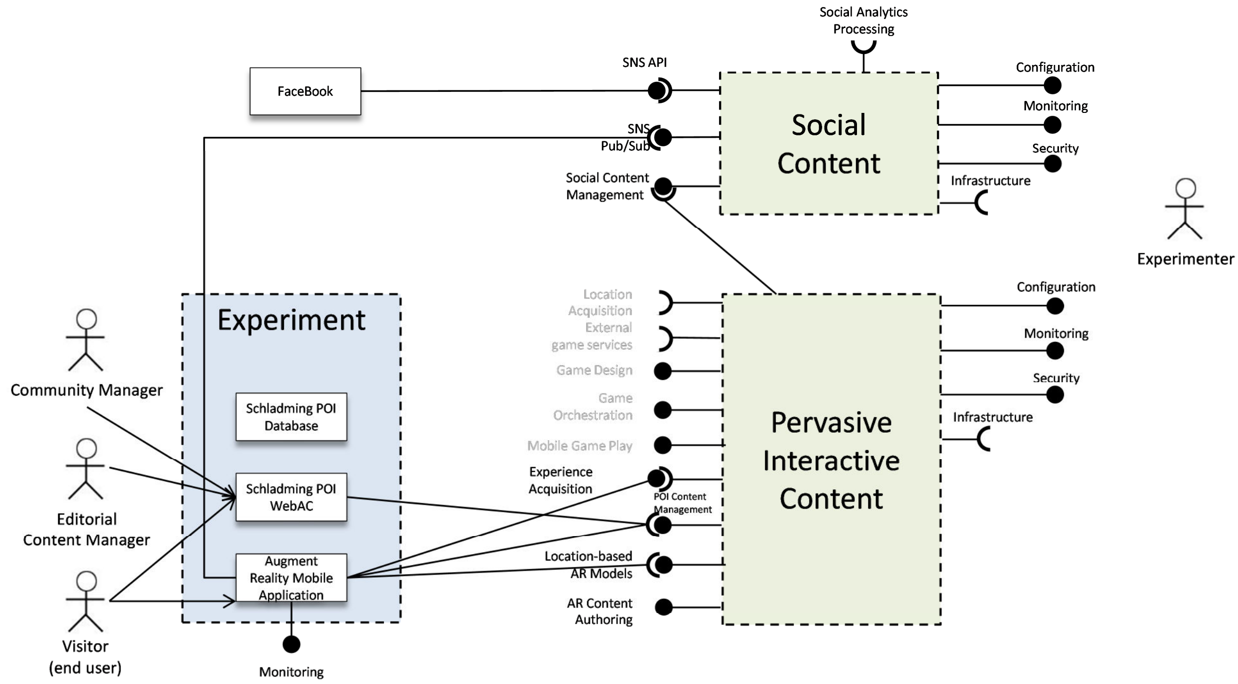


**Figure 17: Schladming experiment components**

Editorial content from the tourism board on Points of Interest (POIs) within the Schladming region is stored within Pervasive & Interactive Content Component (PICC) and accessed through the POI Content Management interface. Information about the POI (e.g. opening hours, tours and prices) is published by the Editorial Content Manager who is in effect the owner of the POI. POIs are locations such as a cultural landmark or places where visitors dynamically discover interests related to their groups/communities such as shopping outlets and social events. POI content is stored by Infonova's R6 adapted to provide POI maintenance and asset management. A Editorial Content Manager can create, delete, update, (de)activate POI content, manage POI content providers and manage POI content enhanced with social/ evaluation data using the Web Administration Centre (WebAC) of R6.

Visitors to Schladming (end users) access and contribute to POI Content via a mobile client application. Visitors will have read access to published POI content but are also able to create and update UGC associated with POIs. UGC attached to a POI can include social network data such as photos, videos, sentiment and emotion sensing. The linking of POI to social content will be supported by Infonova's R6 through integration between the PICC and SCC via the Social Content Management API. In addition, through the SCC visitors can interact with social networks of their choice most relevant to their community.

Technical assets used include Infonova R6 Core with its broad range of functionality with its API as nexus for incoming outgoing data transfer, WebAC web application and the RESTful Interface specially adapted for the needs of the project. For the thin mobile client an augmented reality / mobile application framework is used and integrated into the mobile application that will deliver this information to the Visitors. The mobile client connects to PICC to retrieve the respective content and operational parameters bound to this particular information. Those can be e.g. evaluation or rating values or references on social content that was produced in activity logging is primarily conducted on the server and the log files are made available to the ECC for later analysis of the experiment.

**Table 22: Schladming technology assets**

| Scope | Technology Asset |
|---|---|
| Specific (Experiment) | Mobile Application Client<br>Mobile Application Monitoring (Logger)<br>Schladming POI Database |
| Generic (Facility) | Infonova R6 Core Framework + data storage<br>Infonova WebAC<br>Infonova POI Data Manager<br>Augment Reality Platform<br>Babylon Service<br>SocIoS SN Platform<br>Infonova Social Data Management |
| Infrastructure (Venue) | Wi-Fi, 3G and 4G networking |

### 5.1.2. Content Lifecycle

**Table 23: Schladming content lifecycle**

| Phase | Description |
|---|---|
| Authoring | Content that is made available on the mobile client can be both editorial and UGC. Original Editorial content will be used from the Schladming authorities who already have a database with thousands of main POIs in this area. Editorial content is available in distributed data repositories that can be imported using the POI Content Management interface or manually added by the Editorial Content Manager using the WebAC interface. The content about POIs generated by Visitors would also use the POI Content Management interface via the EXPERIMEDIA mobile app. Access to UGC would be either done using the read-only interface operation for data retrieval or directly by logging in on the WebAC platform if the Visitor is about to make some content management task for individual POIs. As long as the data resides on the PICC it can be managed using the WebAC. In this way a Visitor or Editorial Content Manager can create, update or delete this information. Additional information is available through the Social Content Management interface where the social networks supported by the adapters can be tapped to provide enhanced POI data. These social network adapters can serve both in a read and write fashion as they allow consuming data from social networks as well as publishing information to the social networks. |
| Management | Editors at the tourism board are responsible for curating the editorial content. The original data sources will be consolidated and data management supported by the PICC by the Infonova's R6 platform. User generated content (UGC) can also be handled the PICC where it will be possible to store UGC within the scope of EXPERIMEDIA. For UGC a community manager will handle issues of reported inappropriate content and liaise with the community. Federated community content outside the control of EXPERIMEDIA might also be referenced without quality assurance. |
| Delivery | Content Delivery will be provided by the PICC that builds mainly on Infonova modules. |

| Phase | Description |
|---|---|
| | The federated content is made available to the mobile client and will also be accessible from a web interface. Evaluation, rating values and references on social content as well will be attached to POIs and as such stored in PICC. Some content generated within social networks may remain located in corresponding social networks due to preventing privacy issues. |

### 5.1.3. Experiment Content

Log files store information about application use and data modification operations. These log files are made available to the experimenter for analysis through monitoring interfaces via ECC. Log file data includes information about data sources utilized from the mobile client, changes in data sources as well as additional information about the mobile client. The mobile client will also make available analytical information regarding the localization of the mobile device along with user interaction logs. Log data is therefore captured on the client side and submitted to the ECC. In addition to access logs, ratings, social media references and evaluation values (e.g. from emotional sensing) from the ECC are also available to the experimenters.

For the analysis of the experiment analytical information about data utilization is logged. This includes logging of data creation/modification and read access on POI data. In addition, Visitor interaction information from the mobile client is also stored in the log files and as attributes to the POI of concern. In wider range it will be provided to Visitors to create their own POIs in interaction with social media but without the possibility to delete or update then. Those POIs will be revised and in the case of usefulness left activated otherwise they will be deactivated after certain point of time.

### 5.1.4. Deployment Considerations

Components of integrating partners will use the Infonova REST interface that provides the possibility to retrieve the POI description according to their needs enhanced with evaluation data that will be collected through social media activity and rating or emotion feedback interfaces from EXPERIMEDIA mobile app. It has to be noted that POI definitions would reside at one central point and still can be used in different contexts defined trough virtualisation (different views on the same data- a feature that Infonova R6 can provide). In the primary context of Schladming experiment only single virtualisation will be used as a view for editors on the WebAC platform. However this can be easily changed due the flexibility of Infonova R6. Additionally many other POI providers can be easily switched to this infrastructure using these features. In the case that different POI providers also want include also some evaluation data with single POIs this those values can be easily attached to the POI data using the abilities of Infonova R6data model to expand with unlimited amount of key value pair (of course the limitation is given through physical space and the by the largeness of storage). Infonova R6 core functionality supports easy integration of additional features to the existing data model (here POI data) based on key-value pairs.

## 5.2. CAR

CAR is a facility that gives support to athletes and coaches to develop their sporting talent. The Technology implemented in EXPERIMEDIA should serve as a tool to improve the biomechanics, performance psychology, make talent identification, and detect skills from the elite athletes. More concretely, in the experiment we would like to enhance the process to create new synchronized swimming choreography and improve the training sessions done by the team. It offers the opportunity to have a ubiquitous system for the trainers that will help to make remote-work.

For the synchronized swimming team that is training, it is important to be able to make corrections in situ while they're still in the water. The trainer seeks in the just recorded training session to a specific position of the video using the music tags. Some of the coaches might be at home, and they can take advantage of FMI technology and work remotely, having control over the video streaming and having a whiteboard to put marks on the screen.

### 5.2.1. Functional Components

For the experiment several building blocks need to be integrated as shown in Figure 18. The experiment drives the facility by mixing and adapting multiple sources of sensor, metadata and AV content. The experiment will verify and validate the generic component for Audio Visual Content (AVC).

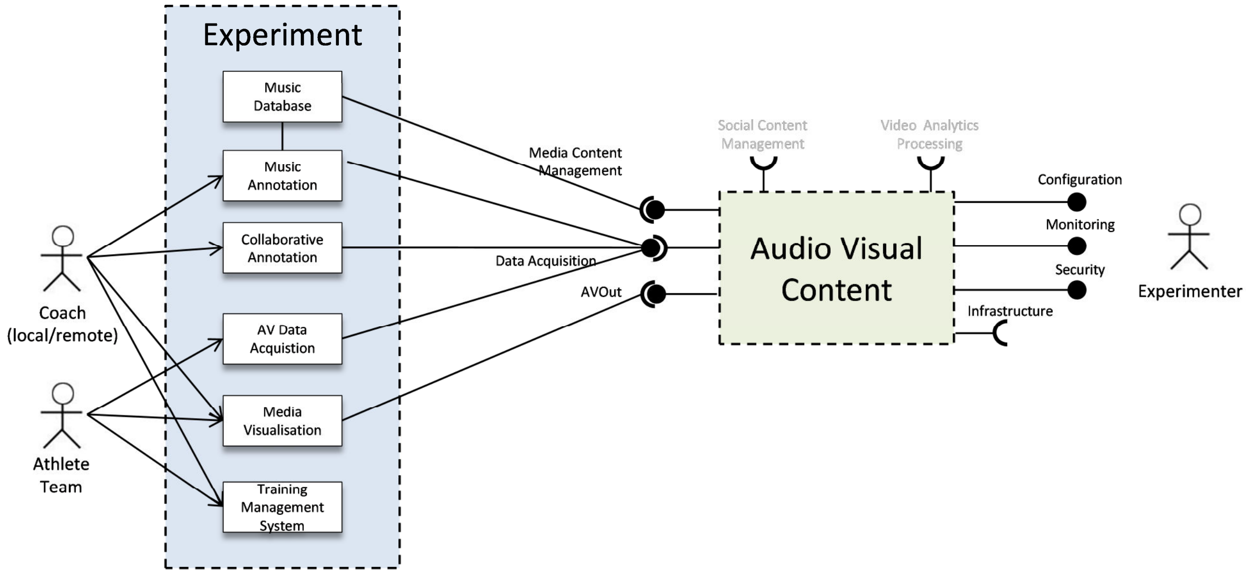


**Figure 18: CAR experiment components**

The input to CAR experiment will come from several sources, while the athletes are training; around four HD and high frame rate video input, the sensors like microphones 'listening' to the music, and annotation of the music done by the coach. In order to support the proper media synchronisation the system will collect information such as latency and generate proper time stamps. Some of the annotation done by the coach has to be received synchronously with the audio-visual content; therefore, metadata with timestamp has to be encapsulated in a proper container and synchronized with the cameras/microphone input.

The video capture will be captured in HD and high frame rates for different specialist, nevertheless the video will be adapted in order to be consumed during the training sessions by consumer devices and for the proper distribution in Internet. Depending on the device, the transport protocol and the parameters of the selected codec H.264 will be required. The coaches will probably also want to replay some part of the training session: the DVR service will allow performing this task.  A coach might want to share his impression with other coaches outside the training area: this will be done through the collaborative annotation service.

The CAR synchronized swimming team is also interested in have a movement database. A media content management will be provided as part of the AVC that builds on a private cloud infrastructure.

**Table 24: CAR technology assets**

| Scope | Technology Asset |
|---|---|
| Specific (Experiment) | Training Session Management<br>Collaborative Annotation |
| Generic (Facility) | Data Acquisition Management<br>Content Adaptation<br>Media Distribution<br>Transcoding<br>Media Synchronisation (music annotation, sensors, coach annotation)<br>Media Content Management (ingest, storage, access)<br>Media Visualisation and Playback |
| Infrastructure (Venue) | Sensors (Cameras, Microphones)<br>Private cloud infrastructure<br>Devices for coaches (PC, large displays, tablets) |

### 5.2.2. Content Lifecycle

The preparation choreography of the Olympic Synchronized Swimming team is a long task and is described in deliverable D212, Section 3.3.1.4. We have detected two different moments where the system will be used:

- Offline or Pre-training: data is introduced into the system by coaches that will used during the training session
- Training session: moment where the swimmers are at the swimming pool and they have a training session. During one of these training sessions the girls might perform several times the same choreography o pieces of it.

**Table 25: CAR content lifecycle**

| Phase | Description |
|---|---|
| Authoring | When the team starts a new choreography from scratch, they decide the music they're going to use, provide ideas for movements and annotate the music. This ideas can be |

| | |
|---|---|
| | some that they already recorded in EXPERIMEDIA from previous choreographies that where discarded. Content is created from a variety of sensor sources (cameras, microphones) and through manual annotations of both music and video. |
| Management | The Coach needs to store planned choreographies (music and annotations) and recorded training sessions (sensors, AV data, annotations). For the music numbers are assigned to different parts of the music and recorded in Media Content Management. During the training session some of the coaches might want to record some of the movement that the girls have done, or just check some specific part of the just performed choreography. The DVR control will allow them to replay some parts of the performance and record whatever they want in the cloud storage. In the recording all the metadata information is also recorded and can be replayed as in a live event. The collaborative annotation aims to help the coaches to share information using their screens as a whiteboard where the images of the team are in the background. |
| Delivery | During a training session, the girls perform the choreography. The coaches that are at the swimming pool are able to visualize one screen the movements of the team with the information or metadata that comes from the annotated music in the offline phase. The experiment will also explore the potential of video analytics calculated from the incoming cameras and consolidated with information from the 3D and the athletes system. Further definition of these capabilities and technology model would be necessary before the technologies could be adopted within this experiment.<br>In order that offer this service to the coaches, the system will need to: acquire the information from cameras and other sensors, transcode it in a useful format (it is device dependant). Data will be retrieved from the video analytics and consolidated with information from the athletes' content management system and 3D system. The consolidated information will be joined with the existing music annotation. The experiment assumes that the size of resulting information is small enough to be embedded and that everything can be streamed to the coaches. If this assumption is incorrect alternative architectures will be used. The coaches are able to use different HMI devices and even a remote coach will be able to get all the information. |

In order to be able to synchronize properly, the video analytics has to tell the video delay and the metadata synchronization system how much time it takes as maximum to generate the information. It will be studied in the scope of the project if the results are better when the data delay is multiple of the GOP size.

### 5.2.3. Experiment Content

Not defined at this stage

### 5.2.4. Deployment Considerations

Here we describe the components to be deployed in the CAR embedded experiment.

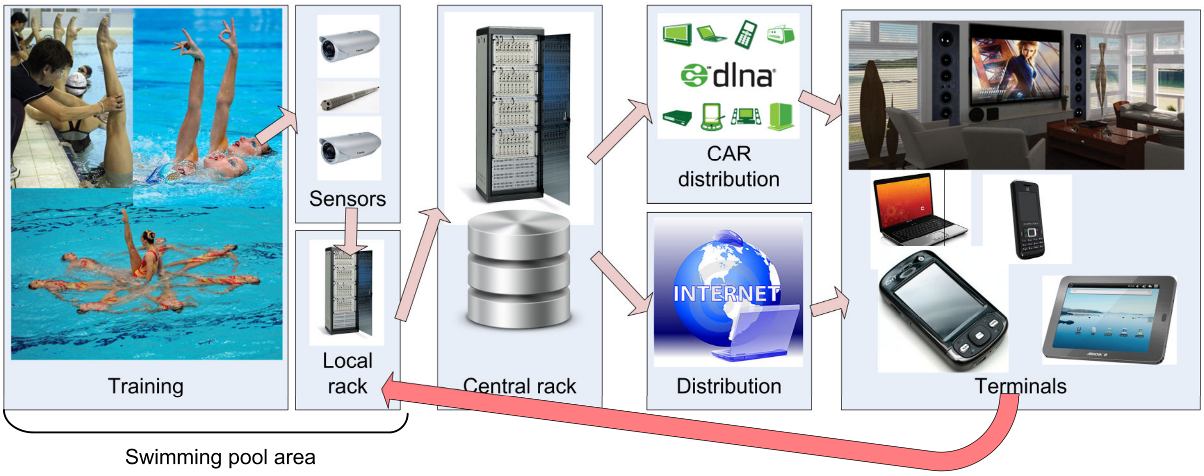


**Figure 19: CAR deployment diagram**

The locations of the various components are listed in the table below.

**Table 26: Locations of CAR components**

| Location | Component |
|---|---|
| Swimming Pool Area | Cameras, Microphones |
| HMI | Playback<br>Collaborative annotation<br>Interaction with remote trainers<br>Devices for trainers in situ (PC, Large screens, Tablets) |
| Local Rack | Ingest<br>Training session management server<br>Metadata synchronisation<br>Live Transcoding |
| Central Rack | Local storage (Long term training session, Media Library: music)<br>On demand transcoding<br>Internal content publication DLNA: publication for local HMI<br>Private cloud |
| Online | Transcoding<br>Streaming<br>DVR<br>Publication of HMI (Live training session, visualization of metadata, collaborative content annotation) |

## 5.3. FHW

The FHW driving experiment, as described in D2.1.2, falls generally within the context of shared, real-time, immersive and interactive cultural and educational experiences. It involves the deployment of an experimental multimedia and social networking platform over the Internet. This platform can be divided into two applications

**Collaborative presentation application:** During the usual operation of Tholos, an FHW museum educator guides the audience through a 3D educational movie. To this direction, the proposed scenario involves the deployment of an FMI application over the EXPERIMEDIA facility that will enable the collaborative presentation of the 3D movie that is projected in Tholos. In more detail, in the proposed scenario a museum educator who is physically located in Tholos, responsible for presenting and for navigating through the 3D movie, is joined by a panel of experts, i.e. given that the movie displayed in Tholos is about some ancient ruins, this panel of experts could be historians and/or archaeologists that were actually involved in the excavation of the ruins. This team of experts may be geographically dispersed and are brought together and in contact with the museum educator and the audience in Tholos by using the EXPERIMEDIA facilities.

**Augmented reality application:** The FHW audience has the ability to visit an exhibition related to the content presented in Tholos. To prepare for this, the presentation system in Tholos will generate metadata, synchronised with the Tholos visual content. The metadata is meant to relate the audience's view of the real world with that of the virtual world. During the Tholos presentation, the audience can tag content of interest using their mobile devices and receive suggestions about visiting related exhibitions hosted in the FHW. Live extra information regarding the specific real item could be offered to the visitor upon visiting the suggested exhibit.

### 5.3.1. Functional Components

The functional components requirement for the FHW experiment is shown in Figure 20. In this example we only explore the collaborative presentation application. The augmented reality application will be elaborated in the experiment design to be published in a future deliverable.

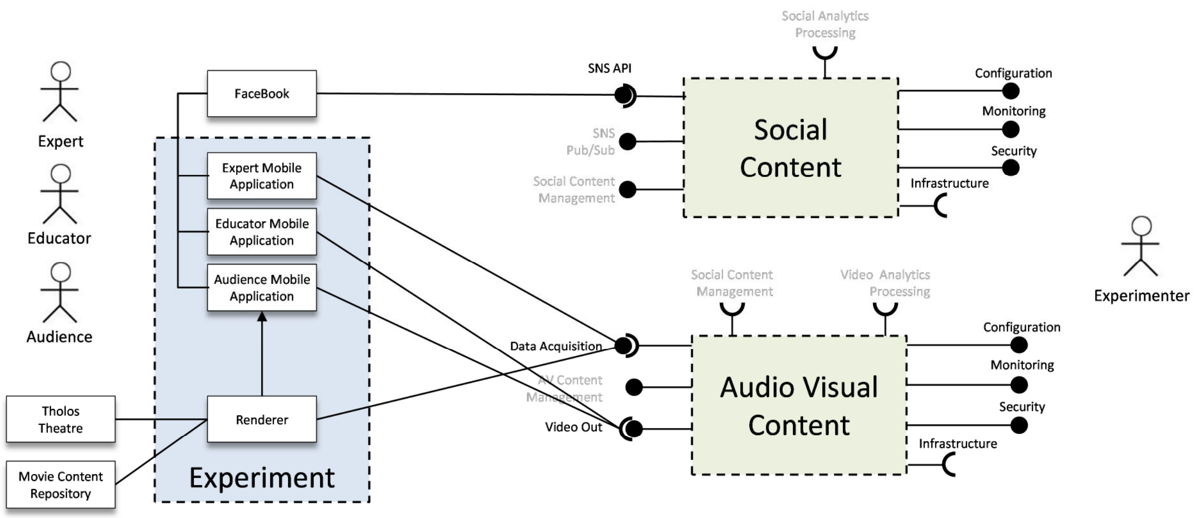


**Figure 20: FHW experiment components for the collaborative presentation application**

The actors involved in this application are: the museum educator, the audience, the panel of experts, and the experimenter. The panel of experts, with the use of an EXPERIMEDIA end-user Application, will be able to remotely view in real time as the presentation is given the 2D version of the 3D movie that is being shown to the audience. The audience, while viewing the 3D movie, will be able to interact with the panel of experts (i.e. ask questions, make comments, express preferences, etc.) through their preferred social networking interface using smart-phones, laptops, or tablet PCs.

During the presentation, the experts can send feedback to the audience and the museum educator either by answering questions in real time, or by texting answers. To facilitate this, the experts' end-user application will be enhanced with a social network plug-in that will automatically collect and present to the archaeologist any relevant comments/questions coming from the remote audience in Tholos.

In order to support this experiment, the existing FHW facility will be extended with the EXPERIMEDIA facility, which will be also physically located at the FHW premises. The overall experiment is being complemented by end-user applications for the actors of the experiment (remote experts and audience).

**Table 27: FHW technology assets**

| Scope | Technology Asset |
|---|---|
| Specific (Experiment) | Expert Application<br>Audience Mobile Application |
| Generic (Facility) | Socio-Mobile Visual Toolbox<br>Infonova R6<br>Data Acquisition Management<br>Content Adaptation<br>Media distribution<br>Transcoding |
| Infrastructure (Venue) | Smartphones<br>The "Tholos Theatre" which is a dome-shaped structure where digital content is projected onto a concave screen in its interior. The movies are projections of 3D models and this operation is supported by a powerful cluster.<br>Tholos Cluster<br>FHW 3D Model Repository<br>The communities which power of social networks cannot be overlooked and with the capabilities of tools like Socio-Mobile Visual Toolbox much information can be retrieved from most of them using a common API. |

### 5.3.2. Content Lifecycle

**Table 28: FHW content lifecycle**

| Phase | Description |
|---|---|
| Authoring | The content under this scenario is authored mainly in real time. During a presentation in Tholos, the social activity content exchanged between the educator, the audience and the panel of experts will be collected by an application running in the EXPERIMEDIA facility part of the Socio Mobile Visual Toolbox. This information is further processed by the Testbed Management monitoring service in order to produce QoE information |

| Phase | Description |
|---|---|
| | that is of interest to the Experimenter. 3D content is stored in the FHW repository and the VR show is generated dynamically and projected to Tholos by specialized software of FHW. At the same time the same content is streamed to the experts' application. |
| Management | User created content from social activity related with the venue should be securely and privately stored. A possible way of doing this is by a daemon application using SocIoS API to retrieve data from the networks and storing it through R6 Core Framework using its REST interface. Only authorized experimenters should access it and query certain information from the aggregated data, possibly using the R6 framework as well. In addition, available 3D assets should be associated its corresponding metadata. |
| Delivery | The monitoring and QoE related content is delivered and displayed to the Experimenter through a dedicated Testbed management interface. The 3D content should be displayed in Tholos and to the experts' devices via their EXPERIMEDIA end-user application. It is important that the movie flow is synchronised between the experts and the audience. |

### 5.3.3. Experimental Data

Not defined yet.

### 5.3.4. Deployment Considerations

Not defined yet.

# 6. Conclusions and Recommendations

The architecture presented in this document has set out a framework that integrates experimental facilities provided by EXPERIMEDIA venue partners with a technical ecosystem that offers a range of FMI capabilities and technologies. Experimenters, technologists and venue providers alike can refer to specifications based on this architecture to first understand the potential of their infrastructure or technical assets as active components in future FMI scenarios and second be provided with a technical framework within which they can integrate their technology with other components and derive experimentally driven evaluation results.

Newcomers to the EXPERIMEDIA project have been provided with the guiding principles that have (and continue to) shape the EXPERIMEDIA architecture. A vision of innovative FMI scenarios in highly social, media rich environments is also presented as a background to the development of the architecture. The EXPERIMEDIA capability landscape provides an abstraction of the elements within content and experimental lifecycles that drive the creation, delivery and management of digital content over a variety of infrastructures and technical systems. These capability specifications can then be mapped to the technologies and infrastructures described as part of the EXPERIMEDIA technology model. Within the technology model, four super-structures (social content; audio/visual content; pervasive content and experiment content) provide the essential foundation for the development of specific experimental test-beds. Three driving experiment blueprints using this foundation have been developed to demonstrate the application of this approach.

Readers of this first architecture are reminded that this is a structure and supported set of processes that is in an early phase of development and not intended to be definitive and complete at this stage. The architecture has been designed from the outset to be able to adapt to changing requirements and support integration of FMI assets (through open calls) in the months and years to come. As the project moves forward, it is anticipated that some of the top-level FMI capabilities and supporting technology super-structures will evolve and enhance to meet these demands. This enhancement process is currently being driven by the anticipated first release of the first driving experiments by the EXPERIMEDIA consortium. The architecture will be updated at project-month 18 (March 2013) in document D2.1.6, taking into account the experience of the first baseline software release, driving experiments and initial feedback from the first open call experiments.

# 7. Appendix A: Example FMI types and sub-types

## 7.1. Content Capability Types and Sub-Types

| Level 1 Capabilities | Level 2 Capabilities | Level 3 Capabilities |
|---|---|---|
| Content authoring | Content Creation | Alpha-numeric; 2D Image; 3D Image; 2D object; 3D object; Environmental; Meta-data |
| | Content Modification | Spatial transform; Colour space transform; Projection transform; Serial modification/deletion |
| Content management | Content Ingest | Alpha-numeric; 2D content; 3D content; Geo-spatial content |
| | Content Store | File system; Web universal resource; Local database; Distributed database; Cloud storage |
| | Content Access | No authentication; Username/password authentication; Search; Read; Write; Update; Delete; Archive |
| Content Delivery | Content Transform | Alpha-numeric; 2D Image, 3D Image; 2D object; 3D Object; Compressed binary |
| | Content Transmission | Local disk storage transfer; IP network; Infra-red network; Bluetooth network |
| | Content Presentation | 2D rendering; 3D rendering; stereo audio rendering; surround audio rendering; Haptic rendering |

## 7.2. Experiment Capability Types and Sub-Types

| Level 1 Capabilities | Level 2 Capabilities | Level 3 Capabilities |
|---|---|---|
| Experimenter Insight | Monitor | QoS metric; QoE metric; QoC metric |
| | Collector | QoS metric; QoE metric; QoC metric |
| | Analysis | Frequency; Variance; PCA; Correlation… |
| | Reporting | Extensible, bespoke reports based on particular metrics |
| Experiment Data Management | Exp. Data Ingest | Alpha-numeric; Key-Value alpha-numeric pairs; 2D content; 3D content |
| | Exp. Data Store | File system; Web universal resource; Local database; Distributed database; Cloud storage |
| | Exp. Data Access | No authentication; Username/password authentication; Search; Read; Write; Update; Delete; Archive |
| Experiment Data Delivery | Exp. Data Transform | Alpha-numeric; Compressed binary |
| | Exp. Data Transmission | Local disk storage transfer; IP network; Infra-red network; Bluetooth network |
| | Exp. Data Presentation | 2D graph; 2D box plot; 2D map; 3D graph, etc. |

# 8. Appendix B: Example capability specifications

## 8.1. Examples from the Augmented Reality Sub-Component: Client AR Renderer

| **JRS Augmented Reality Platform** | |
|---|---|
| Capability ID | JRS_AR_CLIENT_CONTENT_CREATE |
| Capability type | Content Creation |
| Capability sub-type | 2D Image |
| Capability description | Live capture of a 2D image from a digital camera source onto which AR content is later overlaid. |
| Capability attributes | **Attribute**: **Value**<br>Linked role: AR content viewer |
| Capability dependencies | **Type** / **Sub-type** / **ID**<br>Content Presentation / 2D rendering /<br>Data format / JPG file format /<br>Performance frame / Interactive / |
| Technical component | JRS Mobile AR Client |

| **JRS Augmented Reality Platform** | |
|---|---|
| Capability ID | JRS_AR_CLIENT_CONTENT_CREATE_GPS |
| Capability type | Content Creation |
| Capability sub-type | Environmental (Geo-location) |
| Capability description | Real-time capture of current geo-spatial location used to retrieve AR content for the viewer's location |
| Capability attributes | **Attribute**: **Value**<br>Linked role: AR content viewer<br>Data format: WGS84<br>Performance frame: Real-time |
| Capability dependencies | **Type** / **Sub-type** / **ID**<br>Content Delivery / IP network / |

| | |
|---|---|
| Technical component | JRS Mobile AR Client |

| **JRS Augmented Reality Platform** | |
|---|---|
| Capability ID | JRS_AR_CLIENT_CONTENT_PRESENT_2D |
| Capability type | Content Delivery |
| Capability sub-type | 2D Rendering (Alpha-numeric) |
| Capability description | Interactive rendering of associated point of interest user commentary related to spatial position over an IP network |
| Capability attributes | **Attribute** / **Value**: Linked role — AR content viewer; Data format — UFT-8; Performance frame — Interactive |
| Capability dependencies | **Type** / **Sub-type** / **ID**: Content Delivery — IP network; Content Access — Username/password; Content Access — Search; Content Access — Read |
| Technical component | JRS Mobile AR Client |

Capability attributes:

| Attribute | Value |
|---|---|
| Linked role | AR content viewer |
| Data format | UFT-8 |
| Performance frame | Interactive |

Capability dependencies:

| Type | Sub-type | ID |
|---|---|---|
| Content Delivery | IP network | |
| Content Access | Username/password | |
| Content Access | Search | |
| Content Access | Read | |

| **JRS Augmented Reality Platform** | |
|---|---|
| Capability ID | JRS_AR_CLIENT_CONTENT_PRESENT_3D |
| Capability type | Content |
| Capability sub-type | 3D Rendering |
| Capability description | Interactive rendering of 3D model content delivered over an IP network |
| Capability attributes | **Attribute** / **Value**<br>Linked role: AR content viewer<br>Data format: MD2 file format<br>Performance frame: Interactive |
| Capability dependencies | **Type** / **Sub-type** / **ID**<br>Content Delivery / IP network /<br>Content Access / Search /<br>Content Access / Read / |
| Technical component | JRS Mobile AR Client |

# 9. Appendix C: Technical Model Components

## 9.1. SocIoS SN Platform

| **SocIoS** | |
|---|---|
| Overview | The SocIoS API is defined as: an aggregation of methods provided by underlying SNS APIs. The SocIoS API will map a standard interface to collections of methods and objects of the SNS APIs. This will allow the developer to initiate multiple instances of the same functionality to various SNSs by a single method call. The underlying SocIoS Object Model allows the unification of the underlying object models of the SNSs, by capturing the most dominant concepts and their characteristics, and provides mechanisms to manage SNS entities in a unified way. |
| Sub-components | SocIoS API and Object Model |
| External reference | http://www.sociosproject.eu/ |
| Additional notes | None |

## 9.2. Social Analytics Platform

| **WeGov** | |
|---|---|
| Overview | WeGov tools will make it possible to detect, track and mine opinions, behaviour, and discussions topics including their origins, bias and evolution from within a social network context. |
| Sub-components | Web page crawler; semantic analysis modules |
| External reference | http://wegov-project.eu/ |
| Additional notes | None |

## 9.3. Infonova Social/POI Data Management

| **Infonova Data Management** | |
|---|---|
| Overview | Infonova R6 is a highly integrated solution stack sitting on top of a SOA-aligned integration platform. Components included for use in EXPERIMEDIA include the Infonova Core, WebAC (Web Administration Centre) and REST API. The Infonova Core (Infonova Integration Framework) is an integration platform that supports business logic and access to an underlying database. WebAC is a web app running in a browser fully connected to Infonova Core via HTTP and Infonova API (Part of the Core itself). WebAC allows administration of data model specified and stored in the underlying database using core functionality. A specially designed REST API is available that uses Infonova API resides inside Infonova Core, which together with the functional modules mentioned, provides RESTful access to data in underlying database systems. |
| Sub-components | Infonova Core, WebAC, REST API |
| External reference | http://infonova.com/7_20_ENG_HTML.htm |

| | |
|---|---|
| Additional notes | None |

## 9.4. MoreVideo Platform

| **MoreVideo Platform** | |
|---|---|
| Overview | MoveVideo is an audio/visual production platform for Mobile Collaborative Live Video Mixing. A first generation of applications, in this genre, make it possible to broad cast live video streams from various types of use contexts over mobile networks such as 3G ( see for example bambuser.com and qik.com). MoreVideo supports a second generation of such applications, where professional techniques for collaborative live video editing are made available on mobile platforms. Using networked camera phones, it is possible to mix live concurrent video streams from multiple users for public display on internet and locally. The design space includes adapting these new possibilities, previously only available to professional TV-production teams, to amateurs in various contexts of use. Such situations might include the broadcast of multiple live images of soccer matches by parent or, as demonstrated by the Instant Broadcasting System, to visitors at night clubs, and to visitors of public exhibitions. |
| Sub-components | N/A |
| External reference | http://www.tii.se/projects/MoreVideo! |
| Additional notes | None |

## 9.5. Video Analytics

| **Video Analytics** | |
|---|---|
| Overview | Algorithms for person detection and tracking in high-resolution panoramic video streams, obtained from a panoramic camera stitching video streams from 6 HD resolution tiles. The tracking algorithm detects and tracks persons over six static and rectified HD image-sequences from the OmniCam. Instead of using the ultra-high definition image, each video tile is separately analysed by different workstations to enable real-time analysis. The AV content analysis uses a CUDA accelerated feature point tracker, a blob detector, a CUDA HOG person detector, which are used for region tracking in each of the tiles before fusing the results for the entire panorama. The results of the person and blob detector for each image of the different image sequences yield the regions of detected persons for further processing. Furthermore, person IDs are linked to the appropriate combined regions with their corresponding feature points. |
| Sub-components | FascinatE |
| External reference | http://www.fascinate-project.eu/index.php/tech-section/ |
| Additional notes | FascinatE supports developments in audio-visual content analysis and multimedia metadata and semantics. |

## 9.6. ATOS Streaming Platform

| **ATOS Streaming Platform** | |
|---|---|
| Overview | Supports video streaming using the following protocols MPEG2-TS, RTP, RTSP, RTMP |
| Sub-components | N/A |
| External reference | N/A |
| Additional notes | None |

## 9.7. Metadata Synchronisation

| **LIVE** | |
|---|---|
| Overview | The capability to synchronise and embed metadata in audio/visual streams from XML |
| Sub-components | Human annotation; automatic annotation and semi-automatic annotation tools. |
| External reference | N/A |
| Additional notes | LIVE's Metadata Generation System consists of three sub-components that support meta-data annotation: Human Annotation Tool (HAT); Automatic Annotation and the Semi-automatic Annotation. |

## 9.8. Transcoding

| **Transcoding** | |
|---|---|
| Overview | Video: MPEG2, MPEG4 Part 2, H.264/AVC<br>Audio: MP3, AAC (LC & HE), MPEG1 Part1/2, Speex, G.711 |
| Sub-components | N/A |
| External reference | N/A |
| Additional notes | None |

## 9.9. Tracker Service

| **Tracker Service** | |
|---|---|
| Overview | With Tracker it is possible to keep track of a group of people in real-time. Tracker has been used for game mastering and typically fits well for someone already hosting outdoor games or playful activities and who is interested in extending these experiences with a more dynamic event flow. Use Tracker to both guide and direct players during a game but also to view what really happened afterwards, supporting an evaluation and debriefing event. |
| Sub-components | N/A |

| | |
|---|---|
| External reference | N/A |
| Additional notes | None |

## 9.10. Creator Platform

| **Creator Platform** | |
|---|---|
| Overview | Creator is a software platform for creating, setting up and running pervasive games. The supported design process is split into four distinct steps: game design, content creation, location adoption and orchestration. The platform is implemented as a web service and the whole application is accessed through our website. This also makes it quite easy to integrate games into other services and devices. Creator supports a module system allowing connecting basically any kind of external service to it, e.g. web service or mobile clients. |
| Sub-components | N/A |
| External reference | http://www.slideshare.net/kallep/pervasive-game-development-with-the-creator |
| Additional notes | None |

## 9.11. Babylon Service

| **Babylon Service** | |
|---|---|
| Overview | Babylon is a tool that supports user-oriented evaluations of location-based services as well as for development of such services. Babylon is a tool that makes it easy to evaluate the opinions of the users while they utilize the game or service. Thus it becomes possible to more easily find out what the users think and experience while using the location-based service, such as playing a pervasive game. |
| Sub-components | N/A |
| External reference | http://www.mobile-life.org/upload/publication/88/original/babylon.final.2.pdf |
| Additional notes | None |

## 9.12. Augmented Reality Platform

| **JRS AR tools** | |
|---|---|
| Overview | The JRS augmented reality tool suite includes technology that manages and delivers 3D content for augmented presentation on mobile devices. Applications include support for path finding and public transport. |
| Sub-components | AR mobile content viewer; AR content server |
| External reference | http://www2.ffg.at/verkehr/projekte.php?id=745&lang=de&browse=programm |
| Additional notes | None |

## 9.13. 0Install

| **0Install** | |
|---|---|
| Overview | Zero Install is a decentralised cross-distribution software installation system. Other features include full support for shared libraries, sharing between users, and integration with native platform package managers. It supports both binary and source packages, and works on Linux, Mac OS X, Unix and Windows systems. It is fully Open Source |
| Sub-components | N/A |
| External reference | http://0install.net |
| Additional notes | None |

## 9.14. Experiment Monitor

| **NAGIOS** | |
|---|---|
| Overview | Nagios is a powerful monitoring system that enables organizations to identify and resolve IT infrastructure problems before they affect critical business processes. Designed with scalability and flexibility in mind, Nagios gives you the peace of mind that comes from knowing your organization's business processes won't be affected by unknown outages. Nagios is a powerful tool that provides you with instant awareness of your organization's mission-critical IT infrastructure. Nagios allows you to detect and repair problems and mitigate future issues before they affect end-users and customers.<br>Nagios is the monitoring infrastructure used by Infonova |
| Sub-components | N/A |
| External reference | http://www.nagios.org/ |
| Additional notes | None |

## 9.15. Experiment Data Manager

| **TEFIS EDM** | |
|---|---|
| Overview | IT Innovation have developed a distributed data management infrastructure, building on iRODS, to support the experiment lifecycle, for the experimenter, the TEFIS platform in support of the experimenter, and the testbeds integrated with the TEFIS platform. A pre-defined folder structure helps manage data objects; the folders and objects are enriched with TEFIS-defined meta-data to support free-form searching, and experiment sharing. |
| Sub-components | N/A |
| External reference | http://www.tefisproject.eu/ |
| Additional notes | None |

## 9.16. Security Correctness Toolkit

| **SAM** | |
|---|---|
| Overview | The SERSCIS Access Modeller (SAM) takes a model of a system (e.g. a set of objects within a computer program or a set of machines on a network) and attempts to verify certain security properties about the system, by exploring all the ways access can propagate through the system. It is designed to handle dynamic systems (e.g. systems containing factories which may create new objects at runtime) and systems where behaviour of some of the objects is unknown or not trusted. It reduces the chance of mistakes. It makes assumptions explicit. For example, if a security property could be enforced by adding a restriction in either one of two components being developed, each component developer might assume it would be added at the other point. Modelling the whole system forces us to make that choice and document it. All the safety properties that are checked when building the initial system can be automatically rechecked when the system changes. When safety properties are checked manually when writing code (or deploying systems), changes to the system later can make the assumptions behind those checks invalid. |
| Sub-components | N/A |
| External reference | http://serscis.eu/sam/ |
| Additional notes | None |