Grant Agreement No.: 731677
Call: H2020-ICT-2016-2017
Topic: ICT-13-2016
Type of action: RIA

D3.3: FLAME Platform Architecture and Infrastructure Specification V1

In the Grant Agreement, this deliverable is numbered "D3.4"

Dirk Trossen (InterDigital Europe) | 08/09/2017

D3.3 provides the first version of the FLAME platform architecture and infrastructure specification. It outlines therefore the blueprint that forms the basis for further specification work in WP3 but more importantly the implementation work in WP4 towards a deployable FLAME platform within the infrastructures of Bristol-Is-Open and Barcelona. D3.3 covers main concepts of importance to the FLAME platform, driving use cases, requirements derived from those use cases as well as the platform components and their interactions.

Work package	WP 3 Platform Engineering for an FMI Experimentation Platform
Task	Task 3
Due date	07/08/2017
Submission date	V1.0: 08/09/2017; V1.1: 31/01/2018
Deliverable lead	InterDigital Europe
Version	1.2
Authors	Dirk Trossen, Sebastian Robitzsch, Michael Boniface, Carlos Alberto Martin Edo, Gino Carrozzo, Julia Chatain, Tomas Aliaga, Mike Matton, Federico Michele Faca, Steven Poulakos
Reviewers	Simon Crowle, Joan A. Garcia-Espin
Keywords	Interactive Media, NFV, SDN, ETSI MANO, FLIPS, CLMC, Orchestration

Document Revision History

Version	Date	Description of change
V1.0	08/09/2017	First release
V1.1	31/01/2018	Added official document ID to title page; removed duplicate list of figures
V1.2	19/02/2018	Fixed authors names

DISCLAIMER

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731677.

Project co-funded by the European Commission in the H2020 Programme		
Nature of the deliverable:	**R**	
Dissemination Level		
PU	Public, fully open, e.g. web	✓
CL	Classified, information as referred to in Commission Decision 2001/844/EC	
CO	Confidential to FLAME project and Commission Services	

Co-funded by the Horizon 2020 Framework Programme of the European Union

EXECUTIVE SUMMARY

This deliverable D3.3 provides a first version of the specifications for the FLAME platform and infrastructure. It is therefore positioned as a blueprint that drives further specification work in WP3 as well as implementation work in WP4 towards a deployable platform within the test bed infrastructure of Bristol-Is-Open and Barcelona.

The main outcome are those specifications. Those provide insights into the main FLAME platform components, their relation to media service components provided by media service providers on top of the FLAME platform, and the interaction with the infrastructure provided by stakeholders such as BIO or Barcelona city council. D3.3 not only provides high-level interfaces of the various platform components but further decomposes those components into further sub-function interactions, therefore providing the necessary starting point for the realizations in WP4. D3.3 further provides information on relevant specifications, aiding the necessary technology selection process in WP4.

The methodology driving the specification work starts with the main concepts that underlie the FLAME platform, most importantly related to interactive media systems and the emerging infrastructure view for the 5th generation of (mobile) networks. We then formulate concise platform benefits provided to users and media service providers alike, those benefits exemplified through a collection of use cases as well as complex experience scenarios. Following the standard methodology for defining an architecture specification, we utilize those use cases as well as the insights provided by our market analysis performed in D2.1 to derive requirements for the FLAME platform as well as infrastructure. This allows us to define the business as well as technical architecture of FLAME. With this, D3.3 not only provides the specifications as concise outcomes, it also showcases typical usages as well as exemplary experience scenarios we see enabled by FLAME.

Co-funded by the Horizon 2020
Framework Programme of the European Union

TABLE OF CONTENTS

Co-funded by the Horizon 2020
Framework Programme of the European Union

LIST OF FIGURES

Co-funded by the Horizon 2020
Framework Programme of the European Union

Co-funded by the Horizon 2020
Framework Programme of the European Union

LIST OF TABLES

Co-funded by the Horizon 2020
Framework Programme of the European Union

ABBREVIATIONS

AR	Augmented Reality
CLMC	Cross-layer management and control
COTS	Common off the shelf
EaaS	Experimentation-as-a-Service
FLIPS	Flexible IP Services
FMI	Future Media Internet
FQDN	Fully qualified domain name
HTTP	Hypertext transfer protocol
HW	Hardware
IETF	Internet Engineering Task Force
IMS	Interactive Media System
IP	Internet Protocol
MANO	Management and Orchestration
NAP	Network Attachment Point
NFV	Network Functions Virtualization
ODL	Open Daylight
ONOS	Open Network Operating System
OTT	Over-the-top
PCE	Path Computation Element
SDN	Software-defined Networking
SE	Service Encapsulation
SF	Service Function
SFC	Service Function Chaining
SFF	Service Function Forwarder
SLA	Service Level Agreement
SW	Software

Co-funded by the Horizon 2020
Framework Programme of the European Union

TCP	Transmission Control Protocol
tSFF	Transport Derived Service Function Forwarder
VIM	Virtual Instance Management
VNF	Virtual Network Function
VR	Virtual Reality
XOS	Distributed OS

Co-funded by the Horizon 2020
Framework Programme of the European Union

1 INTRODUCTION

As stated in the DoA, this deliverable D3.3 is the "*FLAME Architecture specification used to define the interfaces and high level capabilities of service layer components (WP4)*". Given the time of developing this specification within the project lifecycle, this specification outlines a first understanding of the blueprint that defines the FLAME platform. This understanding will be further developed through work in WP3 as well as through insights into the concrete specification realization work (based on D3.3) throughout WP4. This will ultimately lead to a refined understanding, which will manifest itself in a second version of D3.3, to be released in M18.

Section 2 starts with cornerstones that underlie the development of the FLAME platform. These cornerstones not only include a concise understanding of what constitutes the FMI ecosystem and an interactive media system (IMS) but also the infrastructure view emerging with the development of network platforms such as based on Network Functions Virtualization (NFV) [NFV] and Software-defined Networking (SDN) [SDN]. It is this emerging view that leads to the positioning of FLAME as a service offering on top of those emerging infrastructure platforms. We then continue our FLAME platform development, based on the cornerstones outlined in Section 2, with a recap of the FLAME platform benefits, first outlined in D3.1, before outlining the business architecture with its IMS as well as infrastructure domain models in Section 3. This business architecture introduces the various stakeholders and actors within the FLAME ecosystem.

Section 4 exemplifies our platform benefits from Section 2.5 through a set of small and concise use cases, serving as design patterns of usage to build more complex experience scenarios. The latter are exemplified in Section 4 through four concrete examples stemming from our user partners in the project in collaboration with major platform partners, ensuring an enriched set of experiences that utilizes the unique benefits provided by the FLAME platform. We see those experiences scenarios as a progression from those outlined in D3.1.

The formulation of the aforementioned use cases and scenarios as well as the market analysis in D2.1 leads to the definition of requirements in Section 5 for the FLAME platform and infrastructure, also addressing specific security requirements. Those requirements are formulated in the usual MUST/SHOULD [MOSCOW] terminology, allowing for a later testing of the platform realization against these requirements.

The core of D3.3 is formed by Section 6, starting with a high-level architecture overview in Section 6.1, followed with a definition of the main abstractions utilized in the FLAME platform specification in Section 6.2. We specifically address the area of management and control in Section 6.3 before addressing the security and privacy aspects of the platform in Section 6.4. Sections 6.5 and 6.6 provide a description of the FLAME platform components, their interfaces (with relevant specifications as a starting point for the technology selection process in WP4) and their interactions.

As suggested in the title, we position D3.3 as a first version of this specification with an intended update in M18, incorporating the insights stemming from further WP3 and WP4 work.

Co-funded by the Horizon 2020
Framework Programme of the European Union

2 CORNERSTONES FOR THE FLAME PLATFORM

The vision of FLAME is to *"…establish an FMI ecosystem based on the Experimentation-as-a-Service (EaaS) paradigm that supports large-scale experimentation of novel FMI products and services using real-life adaptive experimental infrastructures encompassing not only the compute and storage facilities but also the underlying software-enabled communication infrastructure*". In this section, we will tease apart the main concepts associated to this vision before outlining the main platform benefits enabled through these concepts and realized through our architecture in Section 7.

If we break out vision statement down further, we need to consider:

1. The systems being designed, developed and tested are "FMI products and services that use adaptive real-life adaptive experimental infrastructures"
2. The activities are conducted using an **Experiment-as-a-Service** paradigm
3. The outcomes are delivered within an **FMI ecosystem**
4. The **adaptive software-enabled infrastructures** enabling the flexible experimentation
5. The main **concepts** realizing the platform provided to media service providers
6. The **realization** of the platform in the emerging adaptive infrastructures

These concepts and aspects are jargon and not widely understood. Therefore, we further refine each in the following sections before outlining our platform vision and aligning our efforts with ongoing industry efforts in the platform space.

2.1 FMI ECOSYSTEM

We are proposing that "an FMI ecosystem will be established using on an Experiment-as-a-Service (EaaS) paradigm." The term Internet Ecosystem was born from the work conducted within the Future Internet Socio-Economics working group [COURCOUBETIS09] which itself build on definitions from the Internet Society [InternetSociety10] and further elaborated as part of the SESERV [SESERV12] and EINS projects [EINS]. An ecosystem is the term used to describe the organizations and communities that guide the operation and development of the technologies and infrastructure that comprise the Media Internet. We are not using the term *ecosystem* in its more traditional social-ecological meaning.

The approach needs to be defined in a way that is independent of the specific stakeholder roles, as we do not want to constrain the approach to specific ecosystem structures. The purpose is to allow the exploration of different stakeholder roles considering possible distribution of responsibilities, information and control for elements of an overall communication system. Therefore, we do not define a specific ecosystem in our approach, we provide ways to explore how it might be structured and how it might behave through experimentation.

Irrespective of the delivery model, our approach needs to support the creation of knowledge about the structure and behaviour of interactive media systems. The multi-stakeholder nature of the system is our priority and therefore the knowledge created must play a key role in supporting the development of possible B2B and B2C relationship between stakeholders. To be as disruptive as possible and to avoid being constrained by current business models, we are proposing that knowledge is created in an

Co-funded by the Horizon 2020
Framework Programme of the European Union

open way with the goal of driving best practice and standardisation efforts for ecosystem structures. Policies related to access and rights to open knowledge will be fundamental to the development of the ecosystems.

2.2 INTERACTIVE MEDIA SYSTEMS

FLAME is researching and developing "FMI products and services that use real-life adaptive experimental infrastructures". In other words, we are building Interactive Media Systems (IMS) and actually we assume we are dealing with distributed IMS. EXPERIMEDIA [Boniface15] provides a short discussion on Networked Multimedia Systems outlining the changes to user experience and the relationship with advanced infrastructures responsible for storage, processing and transmission of multimedia content. Other general definitions exist [Wikipedia, Britannica] but the common aspect is that the user shifts from the role of observer to a participant where their input (e.g., on-demand navigation and streaming) determines the output.

An important scoping factor is that we are exploring the relationship between media and the infrastructure resources used to store, process and transmit the content. Traditionally, interactive media systems are developed by assuming an infrastructure and without the involvement of the network operator. This is the meaning of over-the-top (OTT) service provision. We are now exploring how to bring the media service providers and operators together in a way that allows the infrastructure to be dynamically optimised based on the demand for interactive multimedia content. The optimisation of the quality of interactive media content vs available infrastructure resources is the primary purpose of the platform. Content itself refers to the information provided through the medium and is what is being expressed. The same information can be expressed through different media.

A definition of Interactive Media is given below [InteractiveMedia]:

"*Interactive media is the integration of digital media including combinations of electronic text, graphics, moving images, and sound, into a structured digital computerised environment that allows people to interact with the content for appropriate purposes. The digital environment can include the Internet, telecoms and interactive digital television.*"

We must model and analyse factors influencing the relationship between content and the communication system used to deliver it. Our initial analysis highlighted personalisation, interactivity, mobility and localisation as our initial set of demand factors that can influence adaptation in the infrastructure [Boniface2016]. These factors will be further analysed considering the expected changes in interactive media and emerging capabilities of the platform. We must also note that mobility was highlighted as a first-class factor considering the requirements of the call (ICT-13-2016)[1] but in refinement of our demand characterisation this may be reformulated.

[1] "iii) large-scale experimentation on Future Multimedia Internet (FMI) services fully integrated with broadcasting, with a focus on high mobility scenarios and its impact on communication and storage infrastructures"

Co-funded by the Horizon 2020
Framework Programme of the European Union

2.3 THE EMERGING INFRASTRUCTURE VIEW

We base our platform on an emerging view of the underlying infrastructure that is aligned with that of future 5G systems. Hence, we assume support for isolation of resources through virtualization techniques at the compute, storage and communication level. More specifically, we assume an adoption of the emerging platform technologies being used within FLAME, specifically those surrounding NFV [NFV] as well as SDN [SDN].

With this in mind, we will formulate our solutions in the emerging view outlined by those technologies, as shown in Figure 1. Hence, we assume **HW resources** being provided by the infrastructure provider at the lowest level. Virtualization technologies will expose **virtual resources** to **SW instances** of functions and services of the platform as well as at the application level, labelled as Virtual Network Functions (VNF) in the figure. The deployment of these virtual instances of functions and services, including their connectivity with each other, is captured through **logical abstractions** that describe the overall end-to-end platform as a service function chain (SFC) [SFC]. In Section 6.2, we will elaborate on the concept of SFC as the platform internal abstraction being used for the FLAME platform, while Section 6.6 will outline the various SFCs of the main internal platform services. Note that we do not suggest here, although do not prohibit either, to use service function chains as the logical abstraction for an end-to-end media service provided to end users.

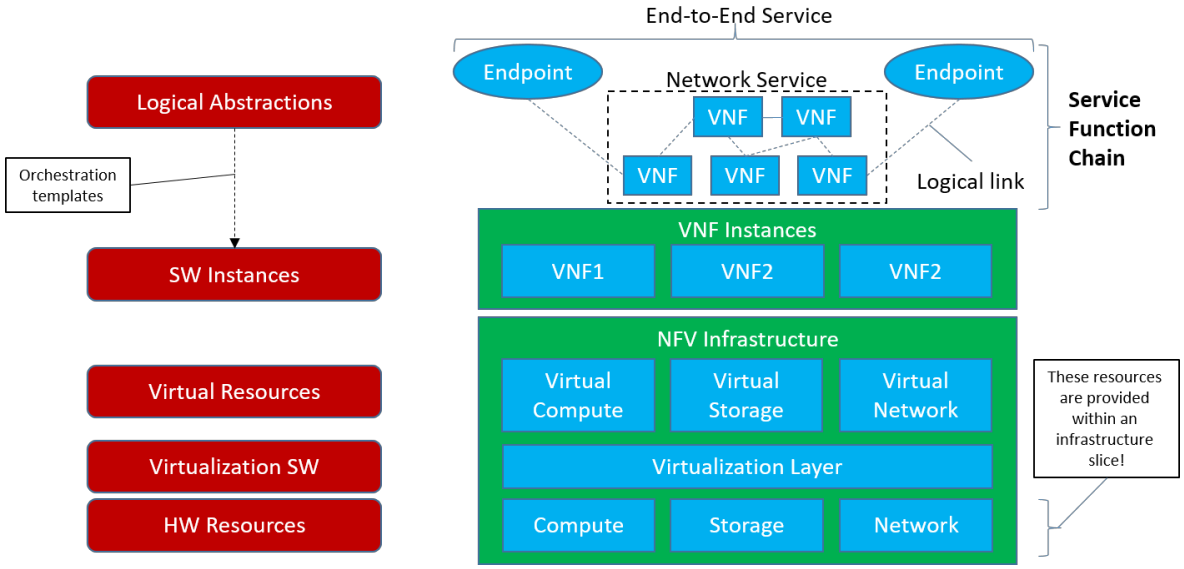

Figure 1: Emerging Infrastructure Resource Management View

In the above figure, we refer to HW resources being provided within so-called **slice** of the infrastructure. This brings us to an important aspect of the emerging resource management view of future infrastructures, namely to provide the management of resources and services provided in Figure 1 with the perception of perfect resource isolation to other deployments happening within the infrastructure. Slicing techniques are being used to establish this perception. The result is resource isolation along all dimensions of resources (compute, storage, network), while the resource owner within said slice can establish its own resource management scheme (shared and/or sliced) to the service provider. We see the notion of not supporting slicing similar to that of offering an exclusive slice over the entire resources of the infrastructure, leading us back to the view in Figure 1.

Once operating within said resource isolation of an infrastructure slice, FLAME will establish its own resource management scheme, following the view in Figure 1. It is important to understand though

Co-funded by the Horizon 2020 Framework Programme of the European Union

that the deep rooting into the underlying infrastructure resources brings about a situation when it comes to the concept of virtualization and the instrument of slicing, namely that **underlying HW resources will remain finite and cannot be sliced infinitely once coupled with performance constraints.** This seemingly obvious statement is particularly relevant when considering shared infrastructure in physically constrained environments. Examples for the latter are street furniture deployments or generally what we consider 'edge deployments of resources'. Such resources very often comprise of a limited number of processors, memory and network capability. Slicing not only introduces often non-negligible overhead but also increases the demand on such limited resources. Platform functions, as introduced in Section 6.6, often need to take priority in resource allocations over non-essential functions (such as media services), therefore leading to non-availability of resources in certain locations of the network. As an example, the realization of the novel routing solution that FLAME will utilize (introduced in Section 6.6.4) demands exclusive usage of several current generation processors in access points of the network. In the currently run experiments in Bristol, this situation already consumes most available resources at the far edge, i.e., the access point, and leaves only limited resources to media service providers to place their media functions, unless capacity increases beyond this minimal demand. Platform-wide resource management must consider such constraints and expose resources to media service providers in an appropriate manner that ensures execution of the platform functions underneath. Suitable orchestration templates (shown in Figure 1) ensure such management to happen.

An important consideration for the resource management at the infrastructure level is that of the **point-of-presence** (POP) of resource pools. Many infrastructure deployments nowadays assume a single POP, e.g., in a local data centre, from which resources will be provided to the service provider (here the FLAME platform as well as the media services utilizing the FLAME platform). Following our vision of FLAME though, we must assume a **multi-POP** approach, i.e. many pools of resources that are distributed over the municipal infrastructure. This is particularly true for resources based in street furniture where it is of utmost importance for the service provider to not only claim resources in general but be specific of its geo-spatial location (e.g., near a building or a landmark). Hence, we assume geo-spatial constraints to govern the reservation of resources within FLAME and its underlying infrastructure.

In addition to the representation of resources at a single or multiple presences, the relation to the infrastructure(s) is crucial. In FLAME, we assume that one FLAME platform instance is provided by a single infrastructure provider, such as those provided by municipalities like Bristol or Barcelona. However, infrastructure brokers can be used to expose a single view over multiple physical infrastructures, e.g., in different municipalities. In that case, the (distributed) infrastructure would appear as a single one towards FLAME albeit provided by two different infrastructure providers. Section 6.2 outlines these relations through the service abstractions utilized for the FLAME platform.

2.4 FLAME AS AN EXPERIMENTATION-AS-A-SERVICE (EAAS) OFFERING

The term EaaS was originally invented by the FIRE initiative and adopted by the EC in the call [ICT-13-2016] as *"…including the development of relevant federation tools and concepts like Experimentation-as-a-Service (EaaS)*". This was following the trend of as-a-Service models which refers to something, in FIRE's case Internet experiments, being made available over the Internet to a customer "*as a service*" [Wikipedia2]. Although it is officially defined, EaaS aims to offer highly programmable, highly observable and Internet accessible ICT resources for the exploration of complex Internet products and services in ways that are managed by self-service and automated business processes. Recent industry announcements suggest that the term is gaining more momentum [IBM16].

Co-funded by the Horizon 2020
Framework Programme of the European Union

Following this concept of '…as a service', we will realise the FLAME Platform on top of technologies that form the overall infrastructure platform, as illustrated in Figure 1. Hence, from an operator and infrastructure provider perspective, the FLAME platform will be implemented as a composition of services within a well-defined pool of resources that is being allocated to the FLAME platform from the infrastructure (provider) side. The FLAME specifications will define APIs that together provide the **FLAME-as-a-Service offering** to media service providers.

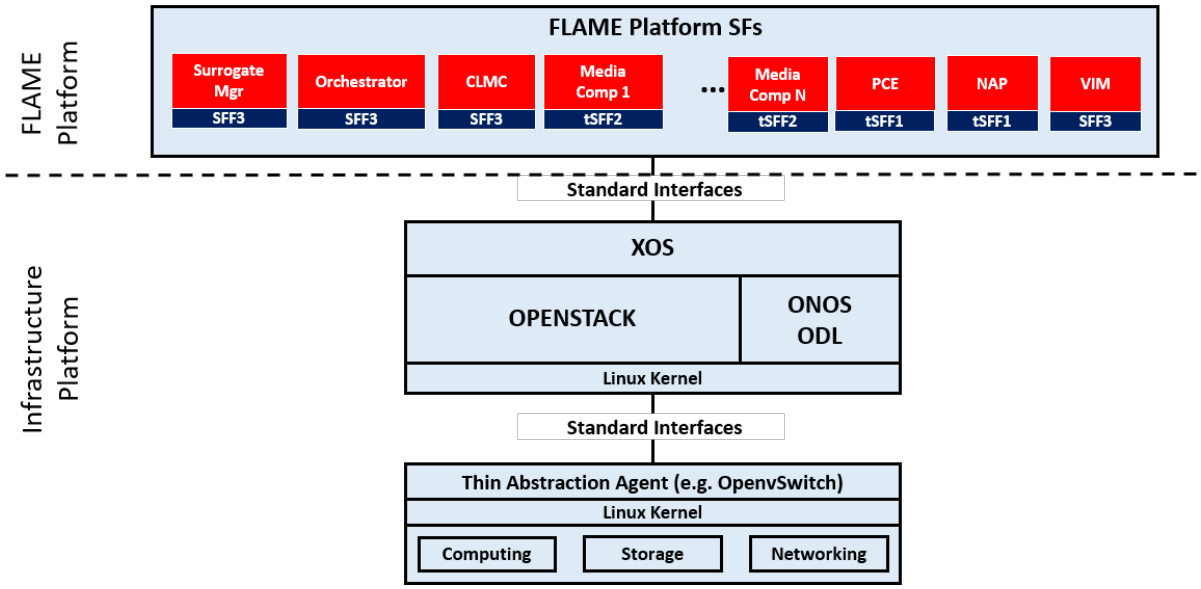

Figure 2: FLAME-as-a-Service Realization over Emerging Network Platforms

The infrastructure platform is accessed via standard interfaces, exposing a network operating system and utilizing virtualization management technologies such as Openstack [Openstack] as well as ONOS [ONOS] or OpenDayLight (ODL) [ODL] on a standard operating system (here Linux). We also support thin abstraction agents towards switching hardware, which allows the Platform to be deployed either over Common-off-the-shelf (COTS) hardware under Linux or over HW-based SDN switches. With this, the Platform can be deployed entirely in the cloud (as a SW instance over managed virtualized compute resources) or over SDN-enabled transport networks. In either case, the resource management to reserve, instantiate, and manage the virtual resources, as shown in Figure 1, will be provided by the underlying infrastructure.

2.5 PLATFORM BENEFITS

The Platform benefits are described in detail in Section 7 of D3.1 "FMI Vision, Use Cases and Scenarios". In general, the goal is to improve performance of interactive media systems whilst managing costs associated with infrastructure resources. Figure 3 provides a summary of four key performance requirements for FMI and generally the 5G space along discussion of associated benefits:

- **Reduce latency**: latency has long been recognized as a major impact on user experience, leading not only to the deployment of content delivery networks but many past and ongoing protocol improvements (e.g., introduction of QUIC [QUIC] aiming at browsing latency improvements). Reducing the service path length is an important target for FLAME through utilizing an intelligent *service endpoint management* and *flexible routing solutions*.

Co-funded by the Horizon 2020 Framework Programme of the European Union

- **Stem unicast proliferation**: the emergence of HTTP as the de-facto streaming protocol in the Internet, infrastructure providers are currently incapable of utilizing in-network multicast capabilities to stem the linear cost explosion that the unicast delivery model of HTTP creates. Through its capability to deliver *HTTP response through in-network native multicast*, FLAME provides a unique capability that significantly reduces costs for multi-viewer scenarios.

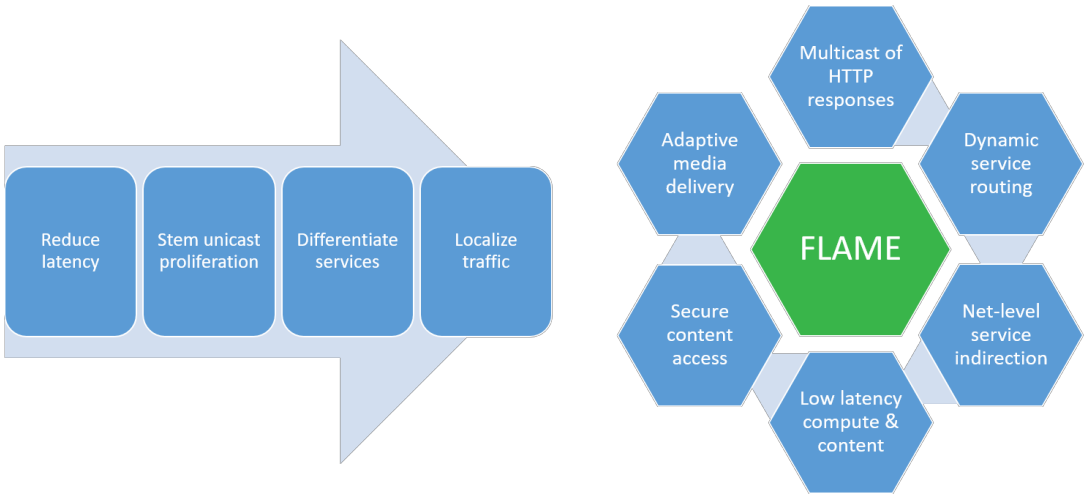

Figure 3: FLAME Platform Benefits Addressing FMI and 5G Requirements

- **Differentiate services**: virtualization opens up the capability to differentiate services by placing service endpoints throughout the network with localized and personalized behaviour. This, however, requires the network to provide a *dynamic service routing* capability that directs traffic to the most appropriate local service instance. Also, a failover mechanism is required to *indirect service requests* if a local instance is unable to provide sufficient service response. Furthermore, *adaptive media delivery* is crucial for differentiation of services, allowing for adapting services, for instance, to different user device requirements by adding transcoding capabilities to the service path for specific users. FLAME provides exactly these capabilities.

- **Localize traffic**: reduction of network traffic is often realized through localizing traffic wherever possible, also addressing the aforementioned latency reduction. Capitalizing on FLAME capabilities to *dynamically route requests* to the most appropriate service instance achieves a likely significant reduction of traffic being sent over longer paths. It also allows for keeping data local in terms of information security as well as possibly exposing the traffic to fewer parties involved. This ability to localize traffic needs to be balanced in a real-life deployment with the possibly higher operational costs for the distributed servers in comparison to centralized data centres. FLAME provides this ability to trade off these aspects towards a commercially viable offering.

- **Remove insecure content access**: the FLAME capability to elevate content delivery from intermediary Content Delivery Networks (CDNs) to fully secured surrogate service endpoints provides further security (e.g., secure content delegation will remove the need for triangular routing to origin servers). Insecure content references would be removed by allowing content to be hosted at surrogate service endpoints with minimal computational authorization functionality. This ensures that content is not exposed to unauthorised parties.

Co-funded by the Horizon 2020
Framework Programme of the European Union

3 BUSINESS ARCHITECTURE

3.1 STAKEHOLDERS

Deliverable D2.1 (Market analysis for the FLAME FMI ecosystem) contains an insight into the innovative ecosystem of FMI (Future Media Internet), which is the FLAME game board. This deliverable describes the market and identifies its stakeholders, defined as participants in a particular sub-market that have a noticeable impact on the functioning of the market, i.e. of the value network of said market. Here sub-market means a sectoral part inside the wide FMI market. The FLAME market is also a part of the global FMI market, as stated in the FLAME terminology.

After the analysis of some key sub-markets, D2.1 identifies the FLAME stakeholders and classifies them in categories and subcategories. Among them, the stakeholders that interact with the platform are:

- **Media service providers**. This category includes a variety of profiles involved in the creation and distribution of media contents and services. The media service providers (both FLAME partners and FLAME future customers) will take advantage of the FLAME platform to deploy and test media services. This category also includes media technology vendors, which provide capabilities that can be required by the media services.

- **Application developers**. This category includes some agents that could be interested in the use of FLAME to test applications with new FMI contents. This is the case, for instance, of developers of new application to depict contents on second screens (such as smart phones) or developers of presentation frontends for new media, such as Virtual or Augmented Reality.

- **Network function providers**. This category contains the network operators. On one hand, FLAME takes advantage of network functions to deploy its virtualised platform over operators' infrastructure. On the other hand, network operators and network functions providers may be FLAME customers interested in the validation of the FLAME approach for the future deployment of FMI services on their respective network resources.

- **Infrastructure providers**, facility providers and equipment managers. These stakeholders are the owners of the infrastructure on which the FLAME platform is built and deployed. These stakeholders are also possible FLAME customers since they may want to replicate the FLAME platform on their own infrastructure in the future.

- **Technology providers** for city services (e.g., outlier behaviour / vandalism detection; tourist attractions, accident prevention). FLAME technological approaches are not only profitable for media service deployments but also for city services that use the city physical infrastructure. This category contains the stakeholders that could provide these kind of services.

- **End users (home and business) / content consumers / customers**. This category includes the users and other stakeholders that extract information from the user consumption and behaviour. In FMI end users become active agents, who interact with the services (as in the interactive media services) and who provide contents as "prosumers" (this is the case in several FLAME scenarios).

Co-funded by the Horizon 2020
Framework Programme of the European Union

3.2 ACTORS

This section narrows the FLAME target down from the list of stakeholder categories to actors. *Actor* does not mean experimenter, but any stakeholder that can play a role in the FLAME platform.

This section relates the FLAME stakeholders to the FLAME platform architecture depicted in Figure 14, detailing the interactions between them and the platform. In this way, the mentioned stakeholders become platform actors.

Media service providers. This is a key category in FLAME since the project is developing an experimentation platform for interactive media services for the FMI. For this reason, media service providers have been identified as main possible FLAME customers. Media service providers are platform users. It must be taken into account that experimentation in FLAME is a multi-actor activity that involves platform users and also FLAME partners. According to the platform approach, media service providers will supply media services (see figure) to be deployed by means of the platform. These media services will include media components, as shown later in Figure 14. The media service provider category includes many different roles and profiles, as compiled in the deliverable D3.1. The following discusses the relationship with the platform for the different kinds of media service providers.

- **Content creators and producers; content publishers; production companies.** This actor category will create contents that will be distributed by means of the platform. FLAME count on this kind of actor in its consortium since the media services imply the content availability. The content creation is linked to the creative industries and include a variety of profiles, like game studios, that are precisely involved in FLAME use cases as well as the specific FLAME partner scenarios.

- **Broadcasters**. Broadcasters are often also content creators and producers. This actor category is also involved in the FLAME use cases and scenarios. Broadcaster main activity is the distribution of media content to final users. In this sense, this actor could be interested in taking advantage of FLAME platform features (stated in Section 2.5) to deliver its contents.

- **Content distributors and aggregators**. This profile puts together contents and services coming from different broadcasters and content creators, as done by PayTV operators or broadcast network owners. In this sense, the interest in the FLAME platform of this actor category is the same than in the previous cases: to test the FLAME platform features for the media service distribution.

- **OTT companies**. As in the case of broadcasters, OTT companies are focused on the distribution of contents to final users, but they use Internet for this purpose without any control or responsibility over the distribution and delivery network. FLAME proposed a new approach that implies the dynamic programmability of the delivery network to optimise the media quality. For this reason, OTT companies can see in FLAME an opportunity to improve their service and take advantage of the new FLAME technological paradigms.

- **Access providers** (cable, satellite, mobile, wired internet…). Unlike the OTT case, this actor category operates the end-user access network. Currently, many telecom operators are becoming also media service providers to integrate more steps in the value chain. In this sense, although playing the role of platform user, the main interest of this actor will focus on the infrastructure (at the bottom of Figure 14) to validate the FLAME technological

Co-funded by the Horizon 2020
Framework Programme of the European Union

approach and to deploy such approach in its own network in the future. The case of mobile network operator is especially relevant due to key character of mobility in FLAME and to the use of geographically extended infrastructures, as mobile networks are.

- **Media technology vendors** (HW and SW implementations): encoding, media servers, media streaming providers; media storage. This is a particular case of media service provider since its focus is not put on the media content but on the technology to enable the media content production, distribution and delivery. In this sense, this actor can act as technology provider for the platform and particularly providers of certain media component functionalities for the media services (see Figure 14), e.g., a transcoding software functionality. This actor can also play the role of platform user to test the performance of a certain product in the FLAME platform.

Application developers. This actor category includes several roles and profiles too. Here, application must be interpreted in the sense of Section 6.1, where an application (or 'app') consists of a number of media components being utilized. One such component usually includes an *application* running on the user device to enable the consumption and enjoyment of the media service by end users. An application developer that wanted to test his/her application at the service endpoint would become a user (according to the blocks of Figure 14) and should additionally provide a media service to feed his/her application. The application developer actor category contains role such as web player suppliers, creator of interfaces for new media (e.g., VR or AR) and creators of second screen applications. These roles are closely related to FLAME. For example, AR is specifically addressed in FLAME use-cases. On the other hand, second screens are required to provide additional media content in mobile phones or to enable the enjoyment of media contents in mobility environments. A game developer is a particular case inside the application developer category although games in FLAME use cases do not only imply the service endpoint.

Network function providers. This category includes the network operators, such as the telecom operators, which we have already explained as media service providers in the access networks. In this sense, this actor can play two different roles in FLAME. On one hand, a media service provider that wants to test the distribution capabilities of the FLAME technical approach. On the other hand, a network infrastructure provider for the access network or for another point of the distribution resources.

Infrastructure providers, facility providers and equipment managers. These actors own (and probably exploit) the physical infrastructures (at the bottom of Figure 14) over which the FLAME platform is deployed. It must be taken into account that these actors are not regular telecom operators, but instead owners or managers of a physical infrastructure deployed in a certain geographical area, such as a city, a campus, a trade fairy or a sport stadium. These actors provide the infrastructure but in the case of FLAME replication, these actors will also integrate the complete FLAME platform, as depicted in Figure 14.

Technology providers for city services. Although FLAME focus is put on media service, the FLAME platform could also support other kind of city services that take advantage of the FLAME benefits and the virtualised, programmable network deployed on the city physical infrastructure. These actors would interact with the FLAME platform in different ways, such as providing services as users or providing technology to deploy services supported on the FLAME platform.

End users (home and business) / content consumers / customers. This category includes the stakeholders that extract information from the user consumption and behaviour. Final users will interact with the FLAME platform in the service endpoints, by means of applications. Concerning

Co-funded by the Horizon 2020
Framework Programme of the European Union

agents that extract information from the user consumption and behaviour, they will interact with the platform by means of the provision of products for monitoring and for the compilation of measurements.

3.3 DOMAIN MODEL

3.3.1 Interactive Media Systems

The high-level domain model is based on the concept of participants interacting in communication processes through exchange of media content. We focus initially on human-to-human (H2H) communication systems in contrast to other types of systems such as those designed for machine communication and control as shown in Figure 4. Nevertheless, some discussion may occur on how we support other forms of communication using the FLAME (e.g. Internet of Things, Security Systems, etc.), especially with the emergence of soft-machines (i.e. online robots and personal intelligent avatars). What's clear is that M2M is beyond the scope of our work but we may consider H2M/M2H in some circumstances where it makes sense for the vision of the FMI.

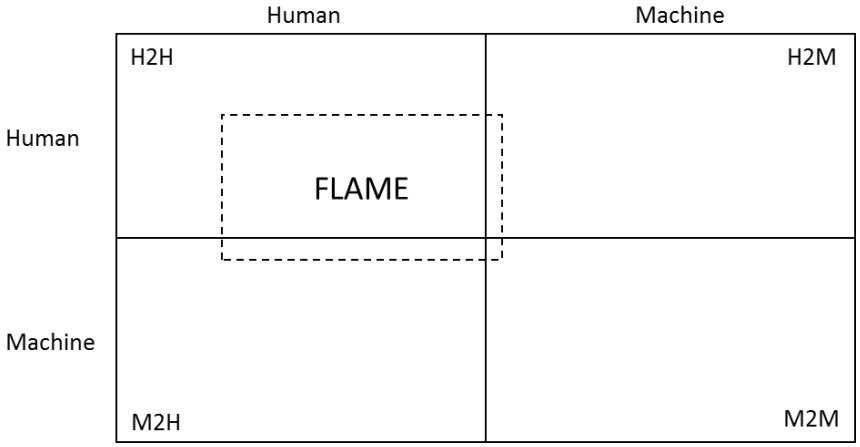

Figure 4: Positioning of FLAME in relation to types of communication system

Figure 5 shows a set of high-level actors in the system that are further described in Table 1. The Participant is the end-user of the system and the primary beneficiary. The Content Producer and Communication Provider are abstract roles that are related to the core assets of Content and Communication Process. Each role could encapsulate further roles or aggregate roles in some business situations as required by the FMI ecosystem. For example, a Communication Provider could be realised through a combination of three organisations Media Service Provider, Platform Provider and Infrastructure Provider but the result of their collaboration is still a communication process supporting exchange of interactive multimedia content. The Content Producer is responsible for managing the production of content, for example, a games provider or broadcaster would be considered Content Providers supporting interaction between Participants. There are cases where Participants create Content as part of an interaction but this is not production as it involves only the exchange of information within a communication process itself.

Co-funded by the Horizon 2020
Framework Programme of the European Union

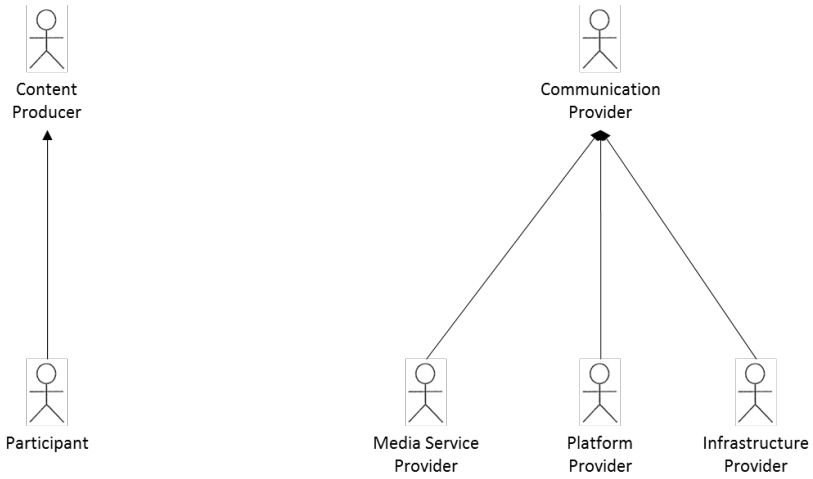

Figure 5: Interactive Media System Actors

Name	**Description**
Participant	A person who uses an interactive media system
Content Producer	A person or organisation responsible for producing and/or managing digital content within a digital communication process
Communication Provider	An organisation responsible for delivering a distributed communication process

Table 1: Description of interactive multimedia system actors

The high-level domain model for demand in interactive media systems is shown in Figure 6. The model focuses on personalisation and interaction that drive adaptation of content and communication processes. Variability in demand increases the cost of delivering interactive media systems [Setiawan16]. Variability in function (including content), space, time and scale requires adaptation of communication processes to deliver expected responses. Modelling and classification of demand and variability will be critical to understanding workload.

Co-funded by the Horizon 2020 Framework Programme of the European Union

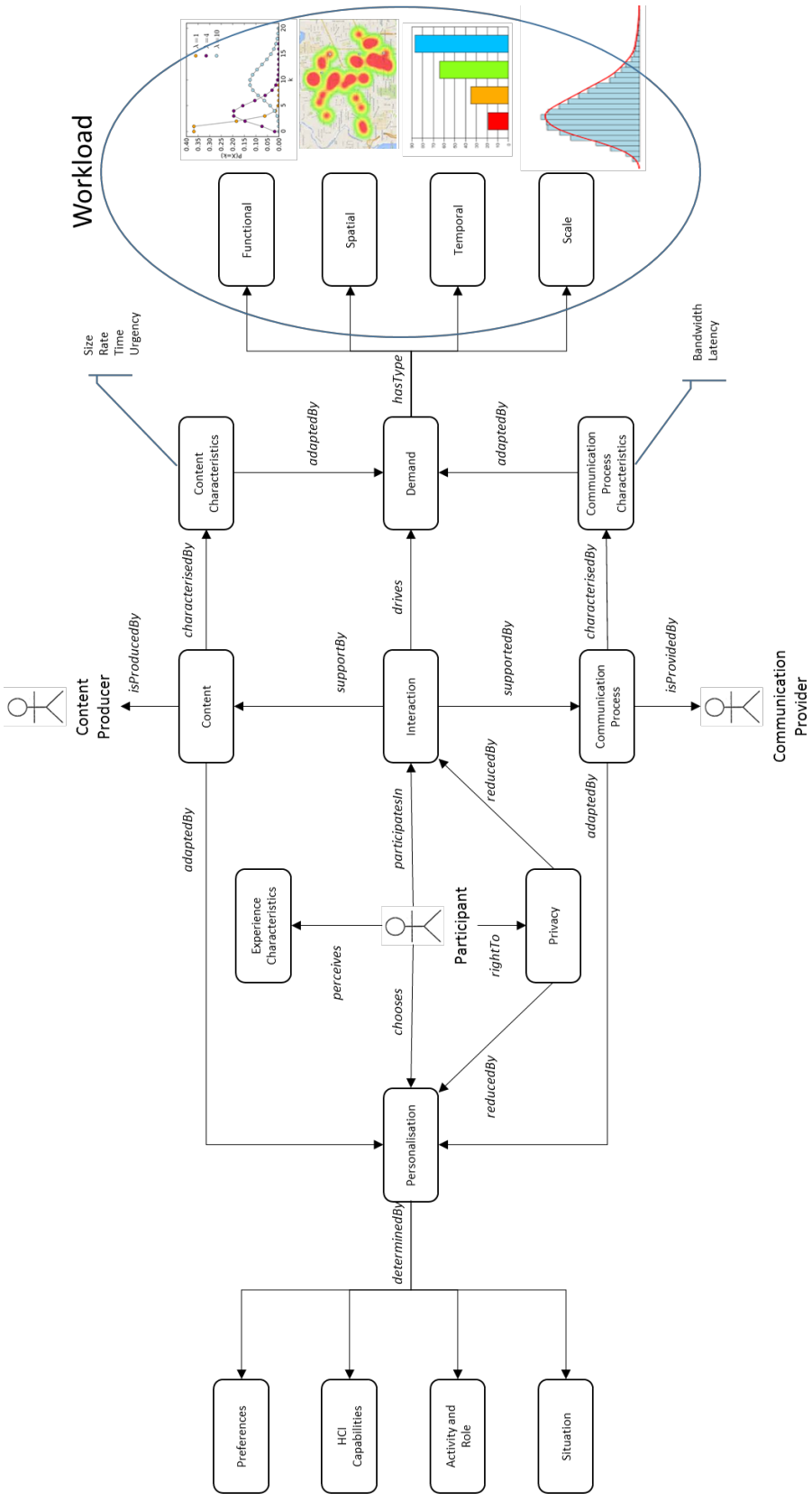

Figure 6: High-level domain model for interactive media systems demand

Co-funded by the Horizon 2020
Framework Programme of the European Union

Concept	Description
Content	The information being expressed within a Communication Process through combinations of media including electronic text, graphics, moving images, and sound. For example, a high-definition video of the Tour De France, an augmented reality guide of Barcelona or a virtual reality sports game.
Content Characteristics	A distinctive characteristic that can be used to measure the degree of excellence of Content. For example, the resolution of a 3D model, the frame rate (FPS) of streamed media, or the PSNR of media compression techniques.
Communication Process	The process of exchanging Content between Participants in response to Interaction through appropriate processing, storage and transmission of media. For example, a Broadcaster publishing a news video to an audience who accesses specific segments on-demand or a Games Provider publishing a location-based game supporting rule-based interaction between players moving around the city of Bristol.
Communication Process Characteristics	A distinctive characteristic that can be used to measure the performance of a Communication Process. For example, the response time of an Interaction under specific conditions, the throughput of Content, or responsiveness to changes in load.
Interaction	The process by which an Interactive Media System responds by presenting Content in response to Participants' actions. For example, presenting a specific video segment in response to a Participant's click navigation, presenting views of a virtual world in response to a Participant's gaze, presenting augmented reality content in response to a Participants location, or animating an avatar in response to a Participant's gesture.
Personalisation	The process of varying Content or Communication Process to meet the Participant preferences, devices, activities and situations. For example, presenting indie-pop music videos to Participant's attending a Kook's gig, reducing or filtering notification update rates from a specific channel, or planning a museum tour based on Participant interests
Privacy	The process of selective disclosure or non-disclosure of personal information within a Communication Process necessary to support Personalisation or Interaction. For example, a Participant permitting a local-based game to track location to geo-located content can be displayed or permitting a music streaming service to access a music library to find similar artists.
Demand	The requested system response from Participants in response to Interaction considering functional, spatial, temporal and scale factors

Co-funded by the Horizon 2020
Framework Programme of the European Union

Functional	How the interaction is being performed. For example, a Participant searching using a Google Query in contrast to search using Voice.
Spatial	Where (space) interaction is happening. For example, watching a video in the street on a mobile device or watching it at home.
Temporal	When (time) Interaction is happening. For example, 10 participants watching a live video stream and five watching it later at various times throughout the evening, synchronising game state between 1000 players interacting in a virtual environment, or eventual consistency of Digital Content published from different sources.
Scale	The size of an Interaction. For example, 10 or 1000 Participants playing a location-based game, or streaming UHD vs SD video content.

Table 2: Description of interactive multimedia media demand concepts

The high-level domain model for an end-to-end Communication Process is shown in Figure 7. The diagram is mirrored to re-enforce the concept of H2H communication. Communication processes are supported by two fundamental layers: Content Layer and Infrastructure Layer. The Content Layer is concerned with interaction, processing and presentation of Content resources to Participants through Applications and Media Services. The Infrastructure Layer is concerned with providing the distributed compute, storage and networking resources necessary to support applications and services performance and availability requirements. The model does not include the Infrastructure Service as this is encapsulated within the concept of a Platform Service.

We should note the constraint that we are not considering control of the device or access network and these must be considered elements of the demand to infrastructure services. However, this should not discourage us from conceptualising how end-to-end communication processes will be managed as it's clear that incorporating device and access networks is expected to be a logical extension of our work conducted by us or others in further projects [5GPPP].

Co-funded by the Horizon 2020
Framework Programme of the European Union

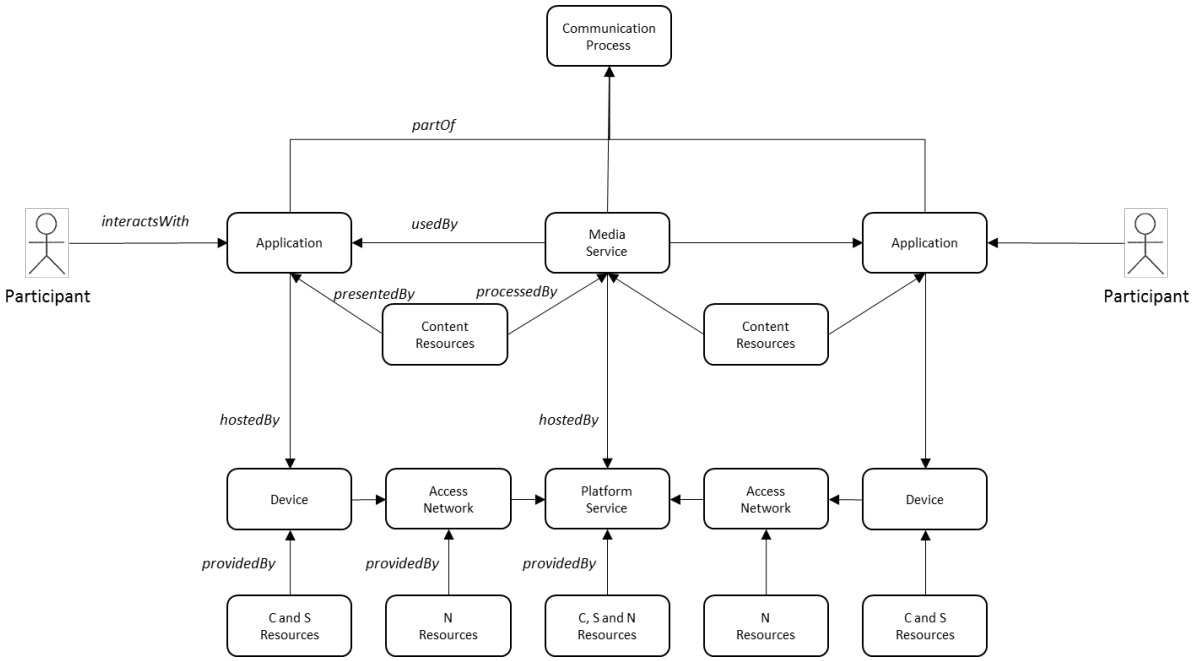

Figure 7: High-level domain model for communication processes

Concept	**Description**
Application	A software program running on an end device that supports human to computer interaction with content resources through processes for visualisation, sound and control.
Media Service	An Internet accessible service supporting processing, storage and retrieval of content resources hosted and managed by the FLAME platform.
Content Resource	An information asset that is exchanged within a Communication Process.
Device	Edge infrastructure used to support human to computer interaction.
Access Network	A telecommunications network that connects end-users to their immediate service provider.
Platform Service	A service supporting the delivery of media services through the management and control of service instances and their compute, storage and network resources.
Compute (C) Resource	A physical or virtual host, host cluster, etc. on which computational processes can be provisioned.
Storage (S) Resource	Physical or virtual infrastructure where content resources and other digital data can be persisted.

Co-funded by the Horizon 2020
Framework Programme of the European Union

Network (N) Resource	Physical or virtual infrastructure where content resources and other digital data can be transported between compute and storage resources.

Table 3: Description of communication system concepts

3.3.2 Platform

The platform domain model is based on the FLAME architecture (see Figure 14) to identify the key platforms concepts and the relationships between them. The domain model also considers other elements that are not included in the architecture diagram but are required for the operation of the platform. For example, this is the case of the templates that describe the media services and that are interpreted by the orchestration layer.

Considering that the Participant is the User shown at the top of the FLAME platform architecture (Figure 14), Figure 8 depicts the high-level view of the domain model for the interaction with the platform and inside the platform.

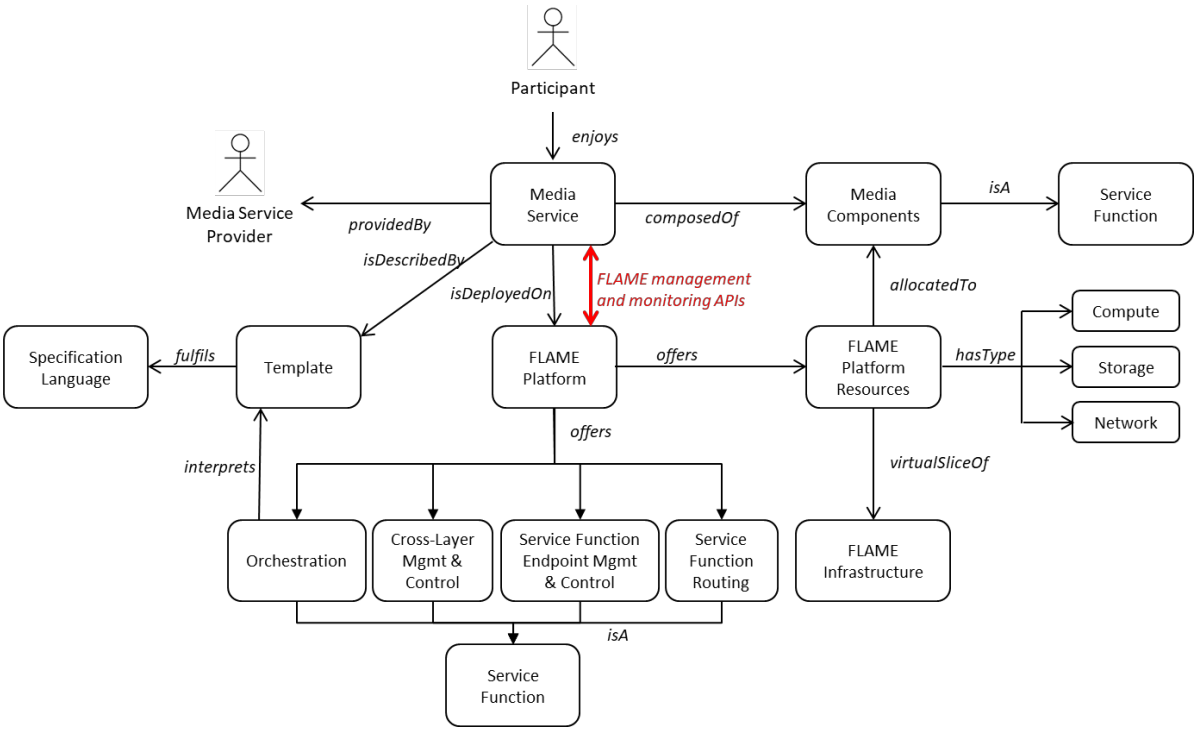

Figure 8: High-level domain model for FLAME Platform

Concept	**Description**
Participant	A person who uses an interactive media system
Media Service	An Internet accessible service supporting processing, storage and retrieval of content resources hosted and managed by the FLAME platform. Media Services

Co-funded by the Horizon 2020 Framework Programme of the European Union

	in FLAME are not a part of the FLAME platform, but they are deployed on it. Media Services are built as a set of Media Components.
Media Service Provider	An organisation responsible for definition and provision of a Media Service. This role may be played by a variety of stakeholders within the FMI ecosystem offering services within an overall communication process.
Media Component	A part of a Media Service. Media Services are decomposed in chains of Media Components, such as media storage, transcoding, adaptive streaming, etc. Media components are implemented utilising Platform resources and other resources outside the FLAME Platform (e.g., end user devices).
FLAME Platform	Distributed programmable resource platform, which is available for the deployment of Media Services with the aim of delivering personalised and interactive experiences to Participants.
FLAME Platform Provider	An organisation that provides or operates a FLAME Platform on infrastructures.
Specification Language	Specification of the descriptors required for the definition, deployment and management of Media Services, including dynamic behaviours that can be explored within experimentation, testing and operations. Specification-Language-compliant Templates will be available for the Media Service Providers to make the definition of Media Service easier. The Specification Language will take into account current orchestration specs for cloud environments, such as TOSCA.
Template	Pattern that is formally expressed by the Specification Language and contains the descriptors for the running of a certain Media Service. Templated will make the description of Media Services easier since Media Service Providers will work on existing documents without defining the Services from scratch. Existing specs that are being considered as reference for the Specification Language are based on the use of templates (e.g., Service Templates in TOSCA).
Orchestration	Process that interprets the description templates coded in the Specification Language and that enables the virtualised resource in the FLAME Infrastructure. FLAME architecture contains an Orchestration layer.
FLAME Infrastructure	Physical equipment deployed across the FLAME cities and that hosts the virtualised FLAME resources.
FLAME Platform Resources	Virtualised capabilities supported by the FLAME Infrastructure and that are made available by the Orchestration layer for the deployment of Media Services on the FLAME Platform. FLAME considers three kinds of resources: compute, storage and network.

Co-funded by the Horizon 2020
Framework Programme of the European Union

Service endpoint management & control	FLAME Platform module in charge of realising the configured service function endpoint policies at the management and control level. For this purpose, it will use the FQDN registration interface to control the registration and deregistration of the service endpoints towards the Service Function Routing component.
Service Function Routing	This component is in charge of suitably configuring the switching fabric of the underlying FLAME infrastructure
Cross-Layer Management and Control	Cross-Layer Management and Control component responsible for improving overall media service management. Control in the FLAME Platform is done in conjunction with the control decision making entities, while the aspect of cross-Layer is a key characteristic in FLAME, since all architecture layers share information to optimise media service and platform performance.
FLAME management and monitoring APIs	Interface between the Media Services and the FLAME Platform. This interface is responsible for the lifecycle of media services including receiving the description templates and delivering them to the Orchestration Layer. FLAME process in charge of supervising Media Service deployment and execution and extracting useful information about the Platform cross-layer performance and processes enabling the exchange of information and the delivery of performance reports and KPIs to the Media Service Providers.

Table 4: Description of FLAME platform concepts

Co-funded by the Horizon 2020
Framework Programme of the European Union

4 USE CASES

This section presents use cases which reflect the advantages of the FLAME platform, how they improve on existing HTTP-based services and outlines the relationship with the stakeholders and the functionality they provide.

Based on the observed pattern of use, we furthermore present four more complex scenarios which present the use of the platform from a media service perspective.

4.1 USE CASES

The core of the service-oriented FLAME architecture is the capability to decouple the relationship between service instances (management plane) from how packets are being sent between them (control and data plane). In order to describe the advantages of the FLAME architecture, the following use cases have been identified which are described in further detail combined with an exemplary scenario.

4.1.1 Fast and Dynamic Service Routing

The use cases which will be presented in this section focus on platform capabilities around fast and dynamic routing of HTTP-based services.

4.1.1.1 Alternative Playout Control

Narrative

This use case is about the cloudification of a media service where application state information of when a user has used the service last is made available at a different time and location. Assuming a streaming application in which the user has stopped the control, this scenario envisages that after a while and in a different condition (e.g., when jumping on a FLAME-empowered public bus) the user can resume the playing from where it was paused. The FLAME platform instantiates local server and caches (e.g., on the bus) to optimize QoS/QoE during a fast move across network access points in the Smart City.

Stakeholders

- *Media service providers* offering a playout service including a playout point (information consumed by end user) and ways to determine the application's state.
- *Platform providers* offering the service routing between decomposed service functions and the migration/replication of service instances in time and space.
- *Infrastructure provider* to provide virtual computing, storage and networking capabilities for the platform to instantiate the playout service at locations within the network.

4.1.1.2 In-session Switching

Narrative

Two playout points of the same media service are available at different locations of the network and the second location has just been connected to the network in order to cope with a growing demand due to an increase in number of users. While some users will continue to be served by Location 1, some will continue consuming the media service from Location 2. As the switching of on-going sessions is

Co-funded by the Horizon 2020
Framework Programme of the European Union

not exposed to the IP endpoints, FLAME ensures that HTTP sessions do not break and – moreover – are transparently carried over to the new playout point.

Stakeholders

- *Media service providers* which can rely on the in-session switching feature to enable/disable playout points at any operational time.
- Platform providers which allow such a transparent in-session switching in order to switch HTTP sessions seamlessly without any extension to existing IP stacks of endpoints.

4.1.1.3 Alternative HTTP-level Streaming Playout

Narrative

This use case is about a group of users separated in space, streaming the same content at roughly the same time using an HTTP-based video on demand application. Based on a reduction in system performance (e.g., bottleneck in the core network path to the streaming service), the service is replicated for a set of users that have the same point of attachment. This results in a recovery of user experience for all affected users transparently to their end devices and the source of the service itself.

Stakeholders

- *Media service provider* allowing to replicate the entire service playout point including the content at a different location.
- *Platform provider* enabling the instantiation of the new surrogate and the transparent switch to the new playout point.
- *Infrastructure provider* that have virtual computing and storage facilities in close proximity to the group of users who have suffered from a service degradation.

4.1.1.4 Multi-source Retrieval

Narrative

A video on demand media service provider offers their service via MPEG DASH which serves segments of audio and video over HTTP. Furthermore, the service provider has several instances of this service located across the city for performance improvement reasons. However, instead of placing all DASH segments on all service instances, they are distributed in a pre-defined manner. Using a multi-source packet transmission approach, the HTTP chunks requested by a particular client are served based on their availability across service instances and other factors such as the network condition towards a particular source.

Stakeholders

- *Media service providers* offering an HTTP service via segments, e.g., MPEG DASH for HTTP level streaming.
- *Platform provider* having the capability to support multi-source content retrieval via HTTP.
- *Infrastructure providers* with compute platforms across their network.

Co-funded by the Horizon 2020
Framework Programme of the European Union

4.1.1.5 Policy-based Routing

Narrative

A particular gaming service requires an end-to-end latency of not more than 5ms and at least 1Mbps throughput in order to ensure no perceptible differences in the application used on the end users' devices. Due to the constraints, several instances of the game service have been placed throughout the network in order to guarantee the performance requirements. When users attach at a new PoA the underlying system selects the service instances which meet the requirements and routes the service flows according to the set policies.

Stakeholders

- *Media service providers* defining the network requirements for their service to work flawlessly.
- *Platform provider* translating the requirements into actions based on real-time performance information from the switching fabric, the service endpoints and about the achieved end to end QoS.
- *Infrastructure providers* to provide virtual compute and storage facilities to instantiate several service instances.

4.1.1.6 Resilience

Narrative

A city conducts improvements to their infrastructure and opens up selected street spots to place new cables. Due to a miscommunication, the fibre connection to a nearby edge computing devices of the local ISPs network has been fully cut by the constructors. As the computing devices were used to serve people in the local area the system immediately realises the situation and repairs the broken service flows by leveraging another computing device a couple of streets away. After several seconds, all affected service flows are repaired.

Stakeholders

- *Platform provider* which has the knowledge about the status of network elements and their reachability. Furthermore, the platform provider needs to have the ability to react to unplanned network changes in a timely manner (order of seconds) by rerouting existing and new service flows away from the affected network element.
- *Infrastructure provider* which has several computing devices across the access and core network made available the platform provider.

4.1.1.7 Service Migration

Narrative

A media service has been defined to provision high quality virtual reality content which is jitter sensitive and throughput intense. Due to unforeseen changes in the network conditions between the user and the instance from where the service is provisioned the service is migrated to another location from where the required media service can be provisioned. The migration includes the allocation of required computational resources, the instantiation of them as well as the actual migration of the service image. After the service has been spun up at the new location, the new endpoint is connected and immediately servers the user.

Co-funded by the Horizon 2020
Framework Programme of the European Union

Stakeholders

- *Media service provider* which provided the boundaries of network conditions required for their service.
- *Platform provider* which allows the continuous measurement of network conditions and the ability to migrate a media service to a new location. Ultimately, the platform provider enables the switching between two service instances at any time without any interaction or knowledge by the client or server.
- *Infrastructure provider* which offers computing, networking and storage resources across their network which can be used to migrate a service.

4.1.2 Low Latency Content

Enabling (extremely) low end-to-end latencies between a client and server is the focus of this section where content is brought closer to the user.

4.1.2.1 One-hop Away Content Retrieval

Narrative

In this use case the media service is a game which is about solving a riddle by recording a short video of the letter found across a city and share it with your team members. Once a team member has received a notification that a new video is available the video file is provisioned directly from the Wi-Fi access points located in lamp posts. Due to the one-hop distance between the client and the server the content is provisioned instantly.

Stakeholders

- *Media service providers* that design their service to be distributed across the network.
- *Platform providers* enabling transparent service routing of HTTP-based communications.
- *Infrastructure providers* providing compute resources across the edge of the access network.

4.1.2.2 Secure Content Delegation

Narrative

A media service provider offers its DRM protected services via HTTPS to its users. In order to satisfy the users by bringing the service closer to the user without comprising on security and rights management delegation servers are located deep within the operator's network and are prepopulated with content. While the platform provides the capability to place secured content on delegation servers without sharing any decryption certificate or key, the platform is given the naming authority to act under the media service provider's domain name. If user requests content the nearest delegation server peered to handle the request which allows a low service creation and service delivery.

Stakeholders

- *Media service providers* that share their TLS certificate to authenticated and therefore trusted network attachment points for HTTP request header peaking purposes.

Co-funded by the Horizon 2020
Framework Programme of the European Union

- *Platform providers* with the ability to allow placement of delegation servers which act under a particular naming authority.

- *Infrastructure providers* that offer compute and storage facilities across their network accessible through standardised APIs.

4.1.3 Secure End-to-end Content

Securing the information exchange between clients and servers is a standard procedure for many popular media services nowadays. This section focuses on use cases around secure end-to-end communication provided by the platform.

4.1.3.1 Authenticated Media Access

Narrative

The outdoor augmented reality mobile application of a service provider authenticates a user over a TLS encrypted connection with an authentication server. For a more responsive user experience there are several authentication servers placed across the infrastructure provider's network based on the demand ensuring fast client authentications. To allow the service routing platform to pick the nearest authentication server the network attachment point of where the user is connected in the city requires to peek into the request header to extract relevant information. As the remaining the communication stays encrypted and all network attachment points are pre-authenticated network entities, the media service provider made its SSL certificate accessible for name authority delegation purposes.

Stakeholders

- Media service providers allowing a trust-relationship instantiation at the service deployment time and the ability to access certificates.

- Platform provider which transparently selects the nearest authentication server using HTTP header packet peeking.

- Infrastructure providers permitting the instantiation of services across their network.

4.1.3.2 Secure Metadata

Narrative

The offered service of a media provider offers users to store their personal information and media preferences (playlists, settings) in the streaming servers and those settings are only accessible by the user and the service provider. As the authentication of the service is encrypted to guarantee the privacy of both parties, the SSL certificate is shared with trusted network attachment points which are the edge network elements of the network allowing the user to gain connectivity. These network elements require certain HTTP head information to deliver the encrypted packet to the nearest streaming server which has the meta data of this particular user to serve the content in the quality the user has previously configured/requested.

Stakeholders

- Media service providers that delegate naming authority to trusted edge network elements.

Co-funded by the Horizon 2020
Framework Programme of the European Union

- Platform provider which provides HTTP header peeking to find the most suitable streaming server.

- Infrastructure provider that offers the instantiation of service instances across their network.

4.1.4 Multicast HTTP

This section focuses on use cases where HTTP responses can be delivered in a multicast fashion enabling network capacity savings over conventional IP.

4.1.4.1 Backend State Synchronisation

Narrative

A media service provider realised an augmented reality game in a smart city environment where gaming instances can be deployed across the entire city due to their rather low-compute requirements. In order to keep certain states in each instance synchronised across the city the service provider uses HTTP to request state updates from the backend. Using a clocked synchronisation interval of three seconds all gaming instances across the city issue an HTTP request at roughly the same time which are served by a single HTTP response to all clients in a multicast fashion.

Stakeholders

- *Media service providers* that chose HTTP as the state synchronisation protocol and a clocked realisation of the service.

- *Platform provider* that allows to serve outstanding HTTP responses in a multicast fashion to all clients.

- *Infrastructure providers* whose networks are built on an SDN switching fabric.

4.1.4.2 Quasi-simultaneous MPEG DASH Delivery

Narrative

A video-on-demand provider has announced the street date of Series 6 their most successful show and delivers its shows over HTTP level streaming. On the evening of the street date the number of viewers using the same broadband operator that happen to watch the same HTTP chunk at roughly the same time increases due to the popularity of the show. Using the abilities of the underlying platform the HTTP requests are served by a single HTTP response in multicast fashion. Furthermore, the platform uses advanced techniques to keep the users quasi-synchronised to ensure a multicast delivery of HTTP responses across all users that happen to watch the new show at roughly the same time.

Stakeholders

- VoD *media service providers* that use HTTP to serve their customers.

- *Platform provider* which can deliver HTTP responses in a multicast fashion.

- *Infrastructure providers* whose networks are built on an SDN switching fabric.

Co-funded by the Horizon 2020
Framework Programme of the European Union

4.1.5 Cross-layer Information and Actions

In order to make the right decisions in media services the FLAME platform offers various information to be collected and retrieved which is the focus of the use cases in this section.

4.1.5.1 Location-based Public Viewing

Narrative

A local broadcaster has public displays installed along the track of a marathon event in the city where the content displayed is based on the preferences of the users (as in which runner they are supporting) in front of the display. Most supporters using the mobile application provided by the broadcaster to track their favourite runner. Using location-based information provided by the platform such as user positions, display positions and the correlation of the number of users in close proximity of the display supporting a particular runner, the service can precisely determine which video snippets of the runners should be presented on the public display.

Stakeholders

- *Media service providers* correlating location-based information about users and their preferences to determine content selection.
- *Platform providers* providing information about user locations.

4.1.5.2 Virtual Device

Narrative

A media service provider released a mobile application which morphs images and videos. As this task is very processing intense, the service collects information about the device from the platform such as power state and access network information. Once the user wants to morph a new image, the service determines if this morphing process should be conducted on the device or in the network based on the information collected. Assuming the power state is high and the network conditions are not satisfactory for this service the notification is given to the application to process the file locally.

Stakeholder

- *Media service providers* using information about devices and network conditions to make a decision on where to process data.
- *Platform provider* that collect the necessary information about the device and the network and make it available.

Co-funded by the Horizon 2020
Framework Programme of the European Union

4.2 SCENARIOS

This section presents an evolved description of the scenarios from D3.1 with a mapping to the presented FLAME architecture and the use cases above. Throughout all scenarios, we will represent each media component by a unique FQDN using the naming convention

`<COMPONENT_ACRONYM>.scenario_name.flame.eu`

Furthermore, each relationship connector between two media components illustrates a direction representing the component that initiates and the component which serves. This information flow is reflected in the naming of the require service level agreement between two media components, i.e.:

`<SERVING_MEDIA_COMPONENT><INITIATING_MEDIA_COMPONENT>`

4.2.1 City Fame

4.2.1.1 Narrative

The city fame scenario envisages a sport event, e.g., marathon, which takes place in a smart city environment where supporters are lining up along the track and take pictures and videos of interesting things happening around them (e.g., favourite runner passes by, a runner has taken over another one in a short sprint). People along the route use a particular application on their end device to show a personalized stream of news of the event created by other people in the city. The content users generate is made available to the local broadcaster for inclusion in their pubic live stream of the event. Furthermore, the city has placed public viewing areas where the live stream of the local broadcaster is shown which has been customised to media content of runners most people in close proximity are interested in.

4.2.1.2 Media Service Composition

This section describes the media service offered by the city fame scenario. The overall media service is composed of a set of service functions (SF) which together make up a Service Function Chain (SFC). Each SF is described in respect to its purpose and relation to the overall scenario.

Description

Media Quality Analysis This media component analyses the media content generated by a user with regards to its quality. Media tags such as content length, file format, size, bit rate or frame rate are first indicators of the suitability of the media file. Based on whether it is audio, video or a picture this component also assesses clarity of speech or blurriness of pictures and videos. The main objective is to provide an immediate feedback to the use about the content quality.
Visual Image Analysis The visual analysis of media content has the main objective to allow the assessment of the recorded information. One of the two main tasks is the identification of runners in pictures and videos. This task is achieved via face recognition and optical character recognition of the start number at the runners' running outfit. The other main task is the assessment of inappropriate and offensive content in the media content uploaded by visually and audibly assessing the content.
Database The database component is the central place to store and provision knowledge about states and historically measured data required to generate user profiles, interest groups and the relationship among them.
Transcoding The transcoding component allows to de- and encode uploaded content into the formats required for the various content consumption devices.

Co-funded by the Horizon 2020
Framework Programme of the European Union

Content Provisioning This media component is essentially the content store where content is located. This content encompasses text, pictures and videos.
Static Template Content This media component simply offers modular web templates for interactors to present media content to the end users.

Relationships of Media Components

The media component relationship chains for the follow me scenario are illustrated in Figure 9. At the bottom of Figure 9, two media service initiators are illustrated, i.e. an application on the mobile end device of a user and a public display on the main square of the city (public viewing). All six media components which serve a particular purpose, as outlined above, are drawn in rectangular boxes and are interconnected by arrow lines illustrating the initiating and serving media component (arrow points towards serving media component).

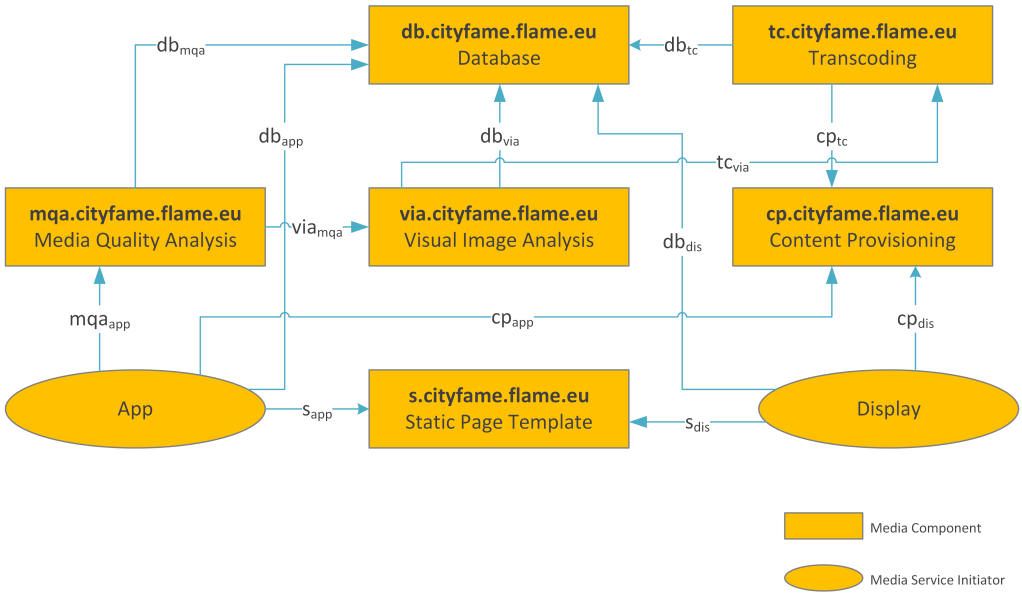

Figure 9: Media component relationships for the City Fame scenario

Service Level Agreement Requirements

The individual functions of the decomposed media service may have very different service requirements for their interactions which must be given to the FLAME platform to allow the appropriate placement and configurations.

Table 5: SLA requirement for the City Fame components

Service Function	SLA Requirements
db	High computational and medium storage demand, stateful
cp	Low computational but high storage demand, stateless
mqa	Low computational and storage demand, system breathing demand based on the number of requests, stateless
s	Low computational and storage demand, stateless
tc	High computational and storage demand, stateless
via	High computational and storage demand, stateless

Co-funded by the Horizon 2020
Framework Programme of the European Union

Table 6 presents the SLA requirement indicators for the relationship between two media components of the Follow Me scenario following the depicted decomposed service in Figure 9.

Table 6: SLA requirement for interfaces of City Fame components

Chain	SLA Requirements
mqa_{app}	Low service creation time, high burst rates, low end-to-end latency and reasonably short response time
cp_{app} cp_{dis}	High network bandwidth demand, low delay variation, low end-to-end latency
db_{app} db_{dis} db_{mqa} db_{tc} db_{via}	Low network bandwidth demand, medium end-to-end latency
s_{app} s_{dis}	Low network bandwidth demand, low end-to-end latency
via_{mqa}	High network bandwidth demand, low delay variation
db_{via}	Low network bandwidth demand, medium delay variation
tc_{via}	High network bandwidth demand, low delay variation
cp_{tc}	High network bandwidth demand, medium delay variation

4.2.2 Follow Me

4.2.2.1 Narrative

The Follow Me scenario is a user-centric service where media content is consumed by a user across multiple devices, e.g., application on mobile device or fixed smart audio device. The service allows the user to consume content anytime and anywhere with all the state about where the user paused and which personalised settings were chosen being made seamlessly available. Furthermore, the media service also envisages third parties to anonymously retrieve preferences of users nearby for targeted advertisement purposes.

4.2.2.2 Media Service Composition

A media service is composed of several media components which form the media service described in the narrative above. This section describes the decomposed media service offered by the follow me scenario. Each decomposed media component is described with regards to its purpose and relation to the overall scenario.

Description of Media Components

Service Authentication This media component allows all service endpoints (application, smart fixed audio device and display) to authenticate themselves against the “Follow Me” service.
Knowledge Database This media component keeps state about the preferences of a user and the outcome of the analysis of various data points (e.g., last played contents, visited locations, etc.) to build the knowledge required to make decisions towards a personalised media experience. The data stored in this function is a mix of user and service monitoring data retrieved from the media application and from the FLAME platform. A correlation and aggregation mechanism can be part of the function and could be embedded in the instantiated function entity: this functionality could be named user preference analysis.

Co-funded by the Horizon 2020
Framework Programme of the European Union

Swipe Me Over This media component allows the user to change the recipient of a media source seamlessly from a fixed smart audio device to the user's mobile device (and vice versa).
Content Provisioning The provisioning of content is realised in by this media component which holds the resources an initiator has requested.

Relationships of Media Component

The media component relationship chains for the follow me scenario are illustrated in Figure 10. At the bottom of the diagram three media service initiators are illustrated and envisaged in the scenario, i.e. an application on a mobile device, a fixed smart audio device and a display (as in a smart TV type of device). While the app and the audio device are part of the user's end device service ecosystem on which the media streaming service is consumed, the display media service initiator belongs to the aforementioned third-party for the purpose of targeted advertisement. The four media components described in the previous section are drawn in rectangular boxes with their FQDN in bold and a description underneath. The relationship between the media initiator and a media component and between two media components is illustrated with an arrow.

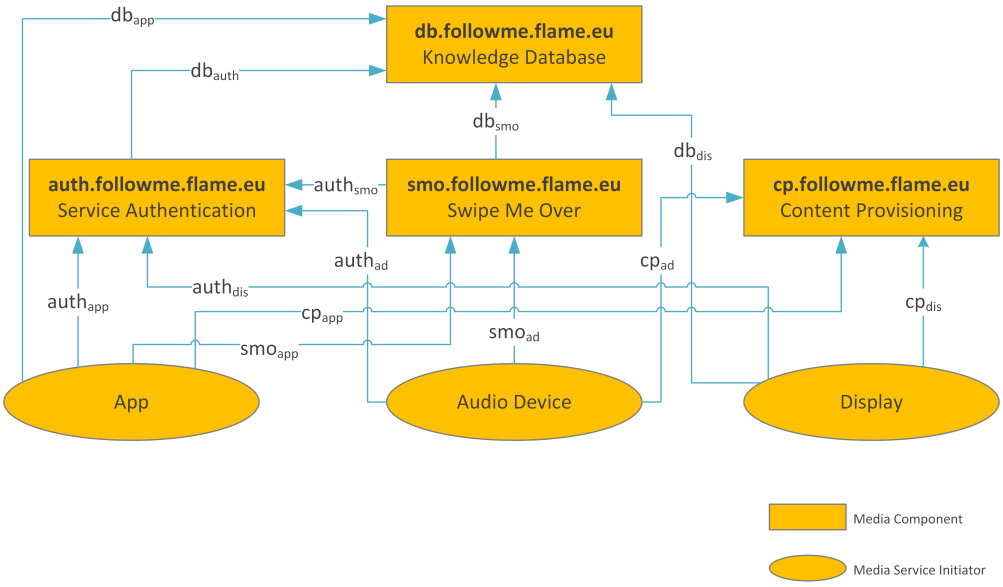

Figure 10: Media component relationships for the Follow Me scenario

Service Level Agreement Requirements

The individual components of the decomposed media service may have severely different service requirements for their interactions which must be communicated to the FLAME platform in order to allow an appropriate computational resource provisioning and ensuring their interactions fall within pre-defined QoS thresholds. Table 7 presents the SLA requirement indicators for the Follow Me scenario following the depicted decomposed service in Figure 10. For simplicity reasons the requirements are categorised as low, medium and high only.

Table 7: SLA requirement for the Follow Me components

Service Function	SLA Requirements
auth	Low computational and low storage demand, stateful
cp	Medium computational and high storage demand, stateless

Co-funded by the Horizon 2020
Framework Programme of the European Union

db	medium computational and storage demand, stateful
smo	Low computational and storage demand, stateless
upa	High computational and storage demand, stateless

Table 8 presents the SLA requirement indicators for the relationship between two media components of the Follow Me scenario following the depicted decomposed service in Figure 10.

Table 8: SLA requirement for interfaces between Follow Me components

Chain	SLA Requirements
$auth_{ad}$ $auth_{app}$ $auth_{dis}$ $auth_{smo}$	Low auth service creation time, medium burst rates, low end-to-end latency and short response time
cp_{app} cp_{ad} cp_{dis}	High throughput demand, low packet delay variation (to prevent streaming degradations and unexpected buffering), medium end-to-end latency
db_{app} db_{auth} db_{dis} db_{smo}	Low throughput demand, medium end-to-end latency
smo_{app} smo_{ad}	Low service creation time, low network bandwidth demand, low end-to-end latency

4.2.3 Gnome Trader

4.2.3.1 Narrative

The Gnome Trader scenario is an augmented reality mobile game where the camera of the device captures the environment and the application seamlessly overlays digital objects on top of it. In this particular application, there are two kinds of overlaid objects. First, **gnome-owned shops** where the player can buy and trade seeds. In these shops, the price of the seeds depends on economic models based on real life offer-and-demand mechanisms. Second, **plants**, that grow from the traded seeds and age according to parameters of the local virtual environment. When buying different seeds, a player can combine them: this will create a new seed generated by mixing the species of the different seeds involved. The player can then plant the resulting seed, thus creating a digital plant overlaid on top of the real world. In order to discover the different seeds and buy at the best prices, the player must scan the physical environment with the application.

4.2.3.2 Media Service Composition

A media service is composed of several media components which form the media service described in the narrative above. This section describes the decomposed media service offered by the gnome trader scenario. Each decomposed media component is described with regards to its purpose and relation to the overall scenario.

Description of Media Components

Database This media component keeps all states related to players and plants such as plant trading activities and locations, liquidity of users and the trading history of a particular player.

Plant Provisioning The provisioning of computed plants is the key task for this media component.

Co-funded by the Horizon 2020
Framework Programme of the European Union

Seed Templates All plants are derived from seed templates which are used when a player virtually plants a new plant somewhere in the city. The templates provide the input to the plant compute component. Each seed corresponds to a plant template: when the player combines seeds, the templates of the different seeds are merged to create a new template.

User Authentication This media component allows a user to authenticate against the media service using pre-defined authentication schemes such as OAuth.

Relationships of Media Component

The media component relationship chains for the follow me scenario are illustrated in Figure 11. At the bottom of the media component relationship diagram (see Figure 11) the media service initiator "App" is depicted representing a user and its Gnome Trader application installed on the user's mobile device. The application communicates with the authentication service to enter the game and uses the plant compute and the database component to display and trade plants on the user's display using the augmented reality app. Once the user plants a new plant the plant compute component obtains the requested seed template(s) from the seed template component, depicted on the top left in Figure 11. Upon a successful computation of a new plant the plant provisioning media component makes it accessible to players.

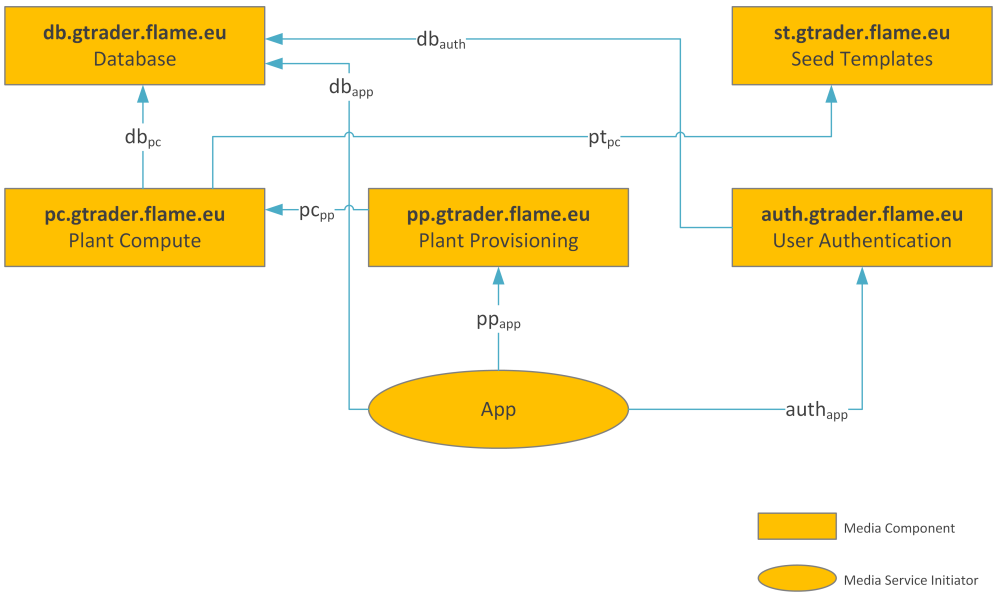

Figure 11: Media component relationships for the Gnome Trader scenario

Service Level Agreement Requirements

The individual components of the decomposed media service may have severely different service requirements for their interactions which must be communicated to the FLAME platform to allow an appropriate computational resource provisioning and ensuring their interactions fall within pre-defined QoS thresholds. Table 9 presents the SLA requirement indicators for the Gnome Trader scenario following the depicted decomposed service in Figure 11. For simplicity reasons the requirements are categorised as low, medium and high only.

Co-funded by the Horizon 2020
Framework Programme of the European Union

Table 9: SLA requirement for the Gnome Trader components

Service Function	SLA Requirements
auth	Low computational and low storage demand, stateful
db	Medium computational and medium storage demand, stateless
pc	Medium with high bursts computational and medium storage demand, stateful
pp	Low computational and medium storage demand, stateful
st	Low computational and medium storage demand, stateless

Table 10 presents the SLA requirement indicators for the relationship between two media components of the Gnome Trader scenario following the depicted decomposed service in Figure 11.

Table 10: SLA requirement for interfaces between Gnome Trader components

Chain	SLA Requirements
$\mathbf{auth}_{\mathbf{app}}$	Low auth service creation time, medium burst rates, low end-to-end latency and short response time
$\mathbf{db}_{\mathbf{app}}$ $\mathbf{db}_{\mathbf{auth}}$ $\mathbf{db}_{\mathbf{pc}}$	Low throughput demand, medium end-to-end latency
$\mathbf{pc}_{\mathbf{pp}}$	medium throughput demand, medium end-to-end latency
$\mathbf{pc}_{\mathbf{app}}$	medium throughput demand, low packet delay variation (to prevent streaming degradations and unexpected buffering) and end-to-end latency
$\mathbf{st}_{\mathbf{pc}}$	Medium throughput demand, medium end-to-end latency

4.2.4 Interactive Storytelling

4.2.4.1 Narrative

The Interactive storytelling scenario provides an immersive story experience in which human users become active participants in an unfolding narrative. The scenario investigates collaborative interactive transmedia storytelling embedded in real-world locations. The demonstration scenario is a story-based city-wide quest.

Interactive Story Experience

An end user (Alice) experiences a story. She installs the story-based city-wide quest application. The application begins a story. The transmedia content is used to present the story (text, audio, images, or video). The story presents decision points enabling the user to influence the outcome. Alice may influence the story by going to physical locations. For example, she may choose for a character in the story to drink by walking to a specific bar, or she may visit a virtual priest character in a story by visiting a specific physical church. AR markers (e.g. predefined signs, city maps or physical objects) may be incorporated in some of the physical locations to support augmented reality presentation of multimedia content. The narrative path is recorded, resulting in a personalized story. Alice could choose to share a video of her experience on social media. The content would include the media content she experienced, decisions made, and the greater story.

Co-funded by the Horizon 2020 Framework Programme of the European Union

4.2.4.2 Media Service Composition

A media service is composed of several media components which form the media service described in the narrative above. This section describes that decomposed media service. Each decomposed media component is described with regards to its purpose and relation to the overall scenario.

Description of Media Components

Database This media component keeps all of the information about the multimedia content, the user, and the user story experience history.

User Authentication This media component allows a player to authenticate against the media service using pre-defined authentication schemes such as OAuth. User Authentication will distinguish between the experience provider and the end user.

Story Compute This media component serves as the storytelling engine. It considers the physical location of the user and other parameters to determine which multimedia information to display, obtaining suitable information from the FLAME platform. The activity of the user is recorded in the database. This media component will inform the caching manager about the predicted location of the user for asset placement in the network.

Caching Manager According to the predicted future locations of the user, this media component securely stores multimedia content in asset storage components. We assume the caching manager to operate under the assumption that assets are generally retrievable due to the network-level indirection capability of FLAME (see Section 2.5). Hence, predictions are speculative to reduce retrieval latency, while still relying on those possible indirections in case the 'closest' asset storage does not hold the requested asset.

Asset storage This component is distributed in the FLAME network based on some geo-cost planning by the media service (e.g., focussing on certain locations in the city). It acts as a secure content delegation server, i.e., it has name authority (acting under as.story.flame.eu) while not holding content authority. With this, content is retrieved securely without the asset storage knowing the name and/or content of the particular asset, increasing security and privacy for the end user.

Relationships of Media Components

The media component relationship chains for the follow me scenario are illustrated in Figure 12. At the bottom of the media component relationship diagram the media service initiator "App" is depicted representing a user and its Interactive Storytelling application installed on the user's mobile device. The application communicates with the authentication service to enter the service and uses the story compute and database component to initiate the generation of suitable story assets. The story compute component instructs the caching manager to place assets into suitable asset storage components throughout the network, while providing the app with links to the assets, being used for retrieval directly from the asset storage components that host the particular asset.

Co-funded by the Horizon 2020
Framework Programme of the European Union

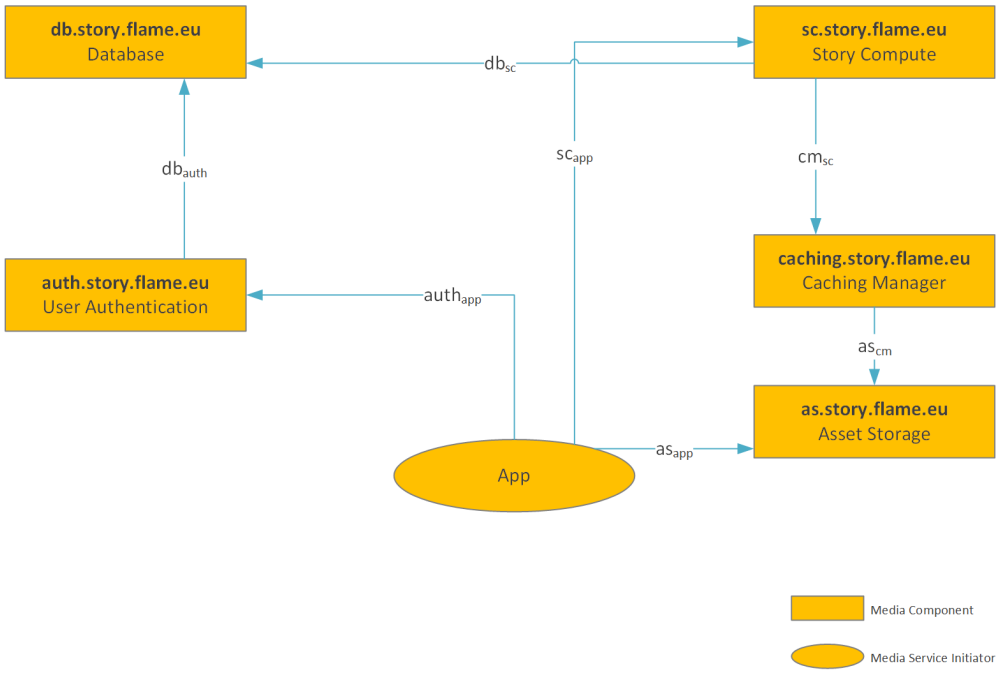

Figure 12 Media component relationships for the Interactive Storytelling scenario

Service Level Agreement Requirements

The individual components of the decomposed media service may have very different service requirements for their interactions which must be communicated to the FLAME platform to allow an appropriate computational resource provisioning and ensuring their interactions fall within pre-defined QoS thresholds. Table 11 presents the SLA requirement indicators for the Interactive Storytelling scenario following the depicted decomposed service in Figure 12. For simplicity reasons the requirements are categorised as low, medium and high only.

Table 11: SLA requirement for the Interactive Storytelling components

Service Function	SLA Requirements
auth	Low computational and low storage demand, stateful
db	Medium computational and medium storage demand, stateless
sc	medium average with high bursts computational and medium storage demand, stateful
as	medium average with high bursts computational and high storage demand, stateless

Table 12 presents the SLA requirement indicators for the relationship between two media components of the Interactive Storytelling scenario following the depicted decomposed service in Figure 12.

Table 12: SLA requirement for interfaces between Interactive Storytelling components

Chain	SLA Requirements
$\mathbf{auth_{app}}$	Low auth service creation time, medium burst rates, low end-to-end latency and short response time
$\mathbf{db_{auth}}$ $\mathbf{db_{sc}}$	Low throughput demand, medium end-to-end latency

Co-funded by the Horizon 2020
Framework Programme of the European Union

sc_{app}	low throughput demand, low packet delay variation (to prevent story experience degradations and unexpected buffering) and end-to-end latency
as_{app}	medium throughput demand, low packet delay variation (to prevent streaming degradations and unexpected buffering) and end-to-end latency
as_{cm}	Medium throughput demand, medium end-to-end latency
cm_{sc}	low throughput demand, medium end-to-end latency

Co-funded by the Horizon 2020
Framework Programme of the European Union

5 REQUIREMENTS

Figure 13 shows an overview of requirements derived from our use cases in Section 4 and the market analysis in D2.1. We categorize those into platform, infrastructure and market related ones.

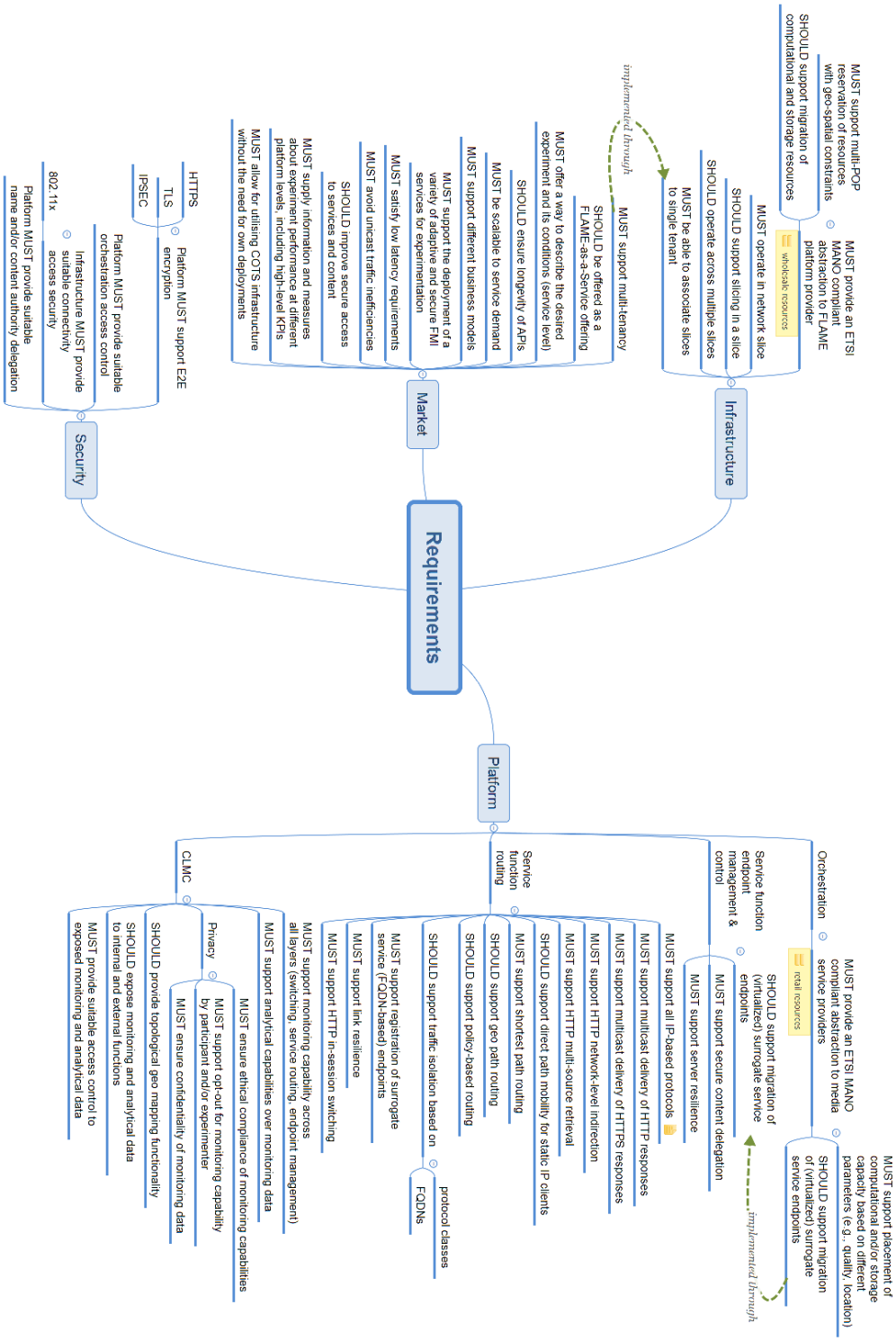

Figure 13: FLAME Requirements

Co-funded by the Horizon 2020 Framework Programme of the European Union

5.1 PLATFORM

Our platform requirements are presented below, divided into the high-level platform components presented later in Section 6.1. For each requirement, we provide links to use cases in Section 4.1, which rely on realizing the requirement.

5.1.1 Orchestration

- **Req.O1**: *MUST provide an ETSI MANO compliant abstraction to media service providers*. This aligns the provisioning of media services with emerging platform concepts and standards in the relevant infrastructure segment, i.e. the ETSI work on NFV. ***(linked to all use cases in Section 4)***

- **Req.O2**: *MUST support placement of computational and/or storage capacity based on different parameters (e.g., quality, location).* This requirement addresses the multi-POP nature of the infrastructure, i.e. the exposure of resources in several/many locations of the infrastructure, not just in a centralized single location. In such case, placement is linked to specific locality and capability of the POP. ***(linked to all use cases in Section 4.1)***

- **Req.O3**: *SHOULD support migration of (virtualized) surrogate service endpoints*. Important for use cases in which service endpoints are virtualized and migrated for capacity planning purposes. ***(linked to use cases in Section 4.1.1.1, 4.1.1.3, 4.1.1.6, 4.1.1.7)***

5.1.2 Service function endpoint management & control

- **Req.SEM1***: SHOULD support migration of (virtualized) surrogate service endpoints*. Realizes Req.O3 at the suitable component. ***(linked to use cases in Section 4.1.1.1, 4.1.1.3, 4.1.1.6, 4.1.1.7)***

- **Req.SEM2**: *MUST support secure content delegation*. It must be possible to delegate the access to content to delegation servers, while such delegation must be secure if needed, in order to reduce latency in content access towards more suitable geographically closer service function endpoints. The FLAME platform will provide suitable interfaces for the registration of authoritative FQDN-based service endpoints that can host secure content ***(linked to use case in Section 4.1.2.2)***

- **Req.SEM4**: *MUST support server resilience*. This is supported through suitable control policies being provided to the FLAME platform as part of the media service template used for the service deployment. ***(linked to use case in Section 4.1.1.6 and 4.1.1.3)***

5.1.3 Service function routing

- **Req.SR1**: *MUST support all IP-based protocols*. Given the proliferation of IP-based protocols, particularly HTTP, FLAME must support those, although the support for other non-IP data plane protocols should be considered through other data plane solutions. ***(linked to all use cases in Section 4.1)***

- **Req.SR2**: *MUST support multicast delivery of HTTP responses*. This realizes a key benefit of our platform, as outlined in Section 3, particularly for multi-user scenarios. ***(linked to use cases in Section 4.1.4)***

Co-funded by the Horizon 2020 Framework Programme of the European Union

- **Req.SR3**: *MUST support multicast delivery of HTTPS responses*. With many content services being end-to-end encrypted, the multicast capability must not get lost in the presence of such encryption. ***(linked to use cases in Section 4.1.4)***

- **Req.SR4**: *MUST support HTTP network-level indirection*. This realizes another key benefit of our platform, outlined in Section 3, by providing the capability to place partial content at service function endpoints while relying on the realization of this requirement to find missing content from alternative service function endpoints. ***(linked to use case in Section 4.1.2.1)***

- **Req.SR5**: *MUST support HTTP multi-source retrieval*. This allows for content being provided by more than one server for load balancing and resilience purposes. ***(linked to use case in Section 4.1.1.4)***

- **Req.SR6**: *SHOULD support direct path mobility for static IP clients*. This removes inefficiencies in routing (i.e. cost reduction) as well as latency for improved service experience. ***(linked to all use cases in Section 4.1 for cases of mobile users)***

- **Req.SR7**: *MUST support shortest path routing*. This defines the standard policy for path computation. ***(linked to all use cases in Section 4.1)***

- **Req.SR8**: *SHOULD support geo path routing*. This defines an alternative policy for path computation. ***(linked to use case in Section 4.1.5.1 and 4.1.5.2)***

- **Req.SR9**: *SHOULD support policy-based routing*. This defines the capability to define any policy constraint for path selection at the service function routing level. ***(linked to use case in Section 4.1.1.5, 4.1.5.1 and 4.1.5.2 and possibly other use cases where such policy is required for socio-economic reasons)***

- **Req.SR10**: *SHOULD support traffic isolation based on protocol classes & FQDNs*. This refers to the capability to provide network-level delivery assurance at fine grain level, e.g., per URL, addressing a key benefit of our platform, namely the service differentiation. ***(linked to all use cases in Section 4.1 when QoS becomes important)***

- **Req.SR11**: *MUST support registration of surrogate service (FQDN-based) endpoints*. The FLAME platform must allow for registering more than just one service endpoint under the same name, i.e. surrogate service endpoints are supported by default. ***(linked to use case in Section 4.1.1.1, 4.1.1.2, 4.1.1.3, 4.1.1.4, 4.1.1.6, 4.1.1.7, 4.1.2.1, 4.1.2.2, 4.1.4.1, 4.1.5.2)***

- **Req.SR12**: *MUST support link resilience*. In case of link failures, the FLAME platform must offer alternative routing alternatives, if available, without any disruption to the service. ***(linked to all use cases in Section 4.1)***

- **Req.SR13**: *MUST support HTTP in-session switching*. When operating in a surrogate service endpoint situation, switching from one service endpoint to another mid-session shall not lead to loss of data. Common HTTP would re-issue the request and discard any previously received data. FLAME will instead utilize already received data and continue the session with the alternative surrogate without re-requesting the full content. This saves costs and reduces latency for flow completion. ***(linked to all use cases in Section 4.1 when surrogate switching is involved but specifically use case in Section 4.1.1.2)***

Co-funded by the Horizon 2020
Framework Programme of the European Union

5.1.4 Cross-Layer Management & Control (CLMC)

- **Req.C1**: *MUST support monitoring capability across all layers (switching, service routing, endpoint management)*. Data for control decision and management planning must utilize data from all layers of the FLAME platform; this is a crucial proposition of our platform from the outset. ***(linked to use case in Section 4.1.1.1, 4.1.1.3, 4.1.1.5, 4.1.1.6, 4.1.5.1, 4.1.5.2)***

- **Req.C2**: *MUST support analytical capabilities over monitoring data*. In order to perform long-term capacity planning, analytical capabilities are crucial, as elaborated later in Section 6.3. ***(linked to all use cases in Section 4.1)***

- **Req.C3**: *MUST ensure ethical compliance of monitoring capabilities*. Such requirement reflects ethical compliance of any monitoring capability that FLAME utilizes and exposes to media services. ***(linked to all use cases in Section 5.1 that involve monitoring)***

- **Req.C4**: *MUST support opt-out for monitoring capability by participant and/or experimenter*. Part of the aforementioned ethical compliance is the ability to fully opt out of any monitoring capability of the platform at the level of user and usage data. ***(linked to all use cases in Section 4.1 that involve monitoring)***

- **Req.C5**: *MUST ensure confidentiality of monitoring data*. Part of the aforementioned ethical compliance is the confidentiality of any gathered data, requiring data protection methods being in place to ensure that monitoring data stays confidential during and after any experiments. ***(linked to all use cases in Section 4.1 that involve monitoring)***

- **Req.C6**: *SHOULD provide topological geo mapping functionality*. In order to allow for geo-based routing of service requests, as expressed by Req.SR8, a mapping needs to exist from geo-based coordinates to topological network information. ***(linked to all use case in Section 4.1.5.1 and 4.1.5.2)***

- **Req.C7**: *SHOULD expose monitoring and analytical data to internal and external functions*. In order to allow for control capabilities at the service function endpoint management and control level, monitoring as well as analytical data is key. The CLMC's objective is to monitor such data and analyse it suitably in order to for it to made available to platform-internal functions for improved operations. ***(linked to all use case in Section 4.1.1.1, 4.1.1.3, 4.1.1.4, 4.1.1.5, 4.1.1.6, 4.1.1.7, 4.1.5.1, 4.1.5.2)***

- **Req.C8**: *MUST provide suitable access control to exposed monitoring and analytical data*. One aspect of realizing confidentiality of monitoring data, the realization of suitable access control solution is key. ***(linked to all use cases in Section 4.1 that involve monitoring)***

5.2 SECURITY

- **Req.S1**: *Platform MUST support E2E encryption, i.e. HTTPS, TLS, IPSEC*. Encryption has become a standard practise for many (content and other) services. Hence it is imperative for the FLAME platform to support such methods of data protection, while retaining the FLAME benefit wherever possible. ***(linked to all use cases in Section 4.1 that would possibly use HTTPS instead of plain HTTP)***

- **Req.S2**: *Platform MUST provide suitable orchestration access control*. The FLAME platform allows for an orchestrated deployment of resources it has allocated at a wholesale level

Co-funded by the Horizon 2020 Framework Programme of the European Union

from the infrastructure. It is therefore imperative that the deployment of such orchestrated services is governed by suitable access control mechanisms, embedded into the orchestration interfaces provided to media service providers. ***(linked to all use cases in Section 4.1)***

- **Req.S3***: Platform MUST provide suitable name and/or content authority delegation:* In order to acts as a surrogate service function endpoint and/or serve content as a delegation server, proper name and/or content authority is imperative to avoid breaches in security and privacy. ***(linked to all use cases in Section 4.1 that involve surrogate and content delegation functionality)***

- **Req.S4**: *Infrastructure MUST provide suitable connectivity access security*. The access to the infrastructure in general as well as the FLAME service specifically is assumed to be governed by SLAs that include the secure access to them. For that purpose, suitable secure access methods at the connectivity level need to be provided, such as realized through 802.11x link level authentication. ***(linked to all use cases in Section 4.1)***

5.3 INFRASTRUCTURE

- **Req.I1**: *MUST provide an ETSI MANO compliant abstraction to FLAME platform provider*. This requirement realizes our preferred infrastructure abstraction, as outlined in Section 2.3 and Section 6.1.

- **Req.I2**: *MUST support multi-POP allocation of resources with geo-spatial constraints.* This requirement is needed to realize any solution for Req.O2.

- **Req.I3**: *SHOULD support migration of computational and storage resources*. This requirement is needed to realize any solution for Req.O3.

- **Req.I4**: *MUST operate in network slice*. This requirement ensures isolation of any resources provided to the FLAME platform. Although dynamic slicing itself might not be supported, exclusive resource access is key to realize this requirement

- **Req.I5**: *SHOULD support slicing in a slice*. Orchestration of a media service within the FLAME platform might want to realize the resource composition for said media service in a single slice, which is in turn provided within the wholesale slice provided by the infrastructure to the overall FLAME platform. For this, slicing in a slice needs support at infrastructure level.

- **Req.I6**: *SHOULD operate across multiple slices*. Another option for providing resource isolation for one or more media services is for the FLAME platform to allocate a wholesale slice for the set of media services, while maintaining a single business relationship for all slices with the infrastructure provider. In contrast to Req.I5, this would provide resource isolation purely through business relations rather than specific slicing solutions.

- **Req.I7**: *MUST be able to associate slices to single tenant*. As part of the realization of Req.I6, the infrastructure must be able to associate the various slices to the FLAME platform tenant.

Co-funded by the Horizon 2020
Framework Programme of the European Union

5.4 MARKET

As shown in the mind map of Figure 13, FLAME is also considering requirements from the market perspective. These requirements identify the needs and expectations of the possible users and customers of FLAME, as explained here:

- **Req.M1**: *MUST support multi-tenancy*. In this way, a group of users shares the same software instance in the server instead of creating multiple instances. This requirement is related to the slice-association for single tenants (infrastructure requirement).

- **Req.M2**: *SHOULD be offered as a FLAME-as-a-Service offering*. The ability to provide FLAME as a pure SW-based product on top of the emerging network platforms outlined in Section 2.3 is likely crucial for market success.

- **Req.M3**: *MUST offer a way to describe the desired experiment and its conditions (service level)*. This requirement is stated as SLA-enabled. FLAME experimenters will design the experiments according to their interests and expectations and FLAME must provide a way to capture this experiment description.

- **Req.M4**: *SHOULD ensure longevity of APIs*. This requirement ensures the long term evolution between media services and FLAME platform by relying on long-lived and well-defined APIs to access the latter.

- **Req.M5**: *MUST be scalable to service demand*. Initial media service deployments are usually planned against expected demand, while such demand can likely change throughout the lifetime of the service. Re-deployment based on new service demand figures is costly; the inherent scalability of the FLAME platform to the evolving demands of the service is therefore a crucial proposition without requiring a full teardown and redeployment of the entire service.

- **Req.M6**: *MUST support different business models*. FLAME operates based on a simple assumption of retailing resources it has agreed to use at a wholesale level from an infrastructure provider. Such simple assumption is meant to open up the support for many business models to prevent lock-in at the business level.

- **Req.M7**: *MUST support the deployment of a variety of adaptive and secure FMI services for experimentation*. Adaptation is key in the FMI services in order to optimise the QoE while consuming media contents.

- **Req.M8**: *MUST satisfy low latency requirements*. Low latency demand is increasing its importance in networked media services due to the large end-to-end delay introduced by current approaches. This delay does not allow the deployment of certain interactive media services. FLAME must take advantage of the new network paradigms capabilities to offer low-latency services.

- **Req.M9**: *MUST avoid unicast traffic inefficiencies*. HTTP unicast transmission causes an inefficient use of network resources since it implies the traffic replication. This requirement is linked to service-routing platform requirements.

- **Req.M10**: *SHOULD improve secure access to services and content*. Security and privacy are important market requirements in FMI services. The multiple versions of media contents in

Co-funded by the Horizon 2020
Framework Programme of the European Union

current intermediary CDNs cause security concerns that FLAME should face. The platform category also contains requirements concerning security.

- **Req.M11**: *MUST supply information and measures about experiment performance at different platform levels, including high-level KPIs*. This requirement ensures the extraction and delivery of the experiment results. This requirement is linked to CLMC requirements in the platform category.

- **Req.M12**: *MUST allow for utilising COTS infrastructure without the need for own deployments*. As mentioned in Req.M6, FLAME wholesales resources of the infrastructure provider, therefore not relying on own deployments from the media service provider. While the latter is possible to complement the resources available at the infrastructure level, e.g., in data centres exposed to the FLAME platform as a media service endpoint (using local load balancing for utilizing the resources transparent to the FLAME platform), FLAME does not rely on such deployments. Following the trend outlined in Section 2.3, we assume that the infrastructure resources are increasingly based on COTS hardware.

Co-funded by the Horizon 2020
Framework Programme of the European Union

6 PLATFORM ARCHITECTURE

6.1 OVERVIEW

Figure 14 shows the FLAME platform architecture, operating on top of an infrastructure such as the one provided in our deployments in Bristol and Barcelona. In this figure, we focus on the main layers of the overall architecture, including the FLAME platform, while showing crucial inter-layer interfaces for explanation of the overall workings. Detailed descriptions on the internal functions of the various layers are deferred here to Section 6.6.

At the very bottom, we assume the existence of the **infrastructure (provider)**, exposing an ETSI MANO compliant interface [NFV] to the FLAME platform for resource management at the *wholesale level*, i.e. the FLAME platform is reserving platform resources in the compute, storage and networking domain. We assume such infrastructure resource exposure to follow our observed emerging infrastructure view, as discussed in Section 2.3, with the ability to reserve resources for the FLAME platform as part of a longer-lived relationship between FLAME platform provider and infrastructure provider, realized with an infrastructure slice.

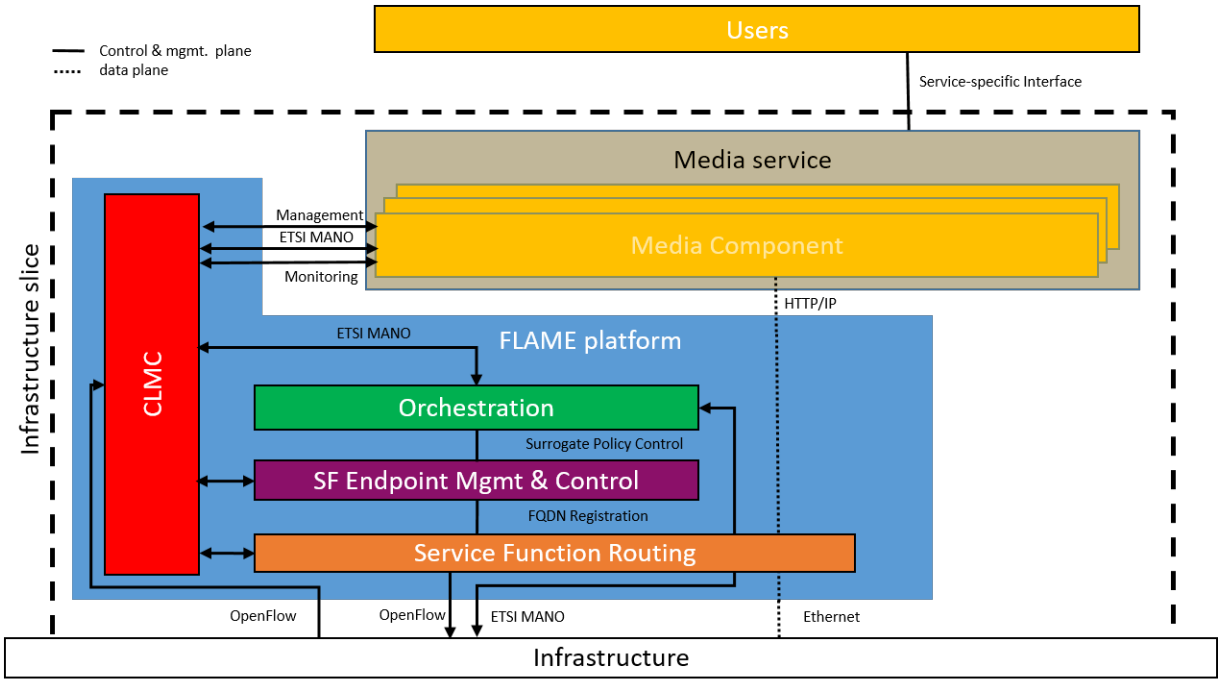

Figure 14: FLAME Platform Architecture

Resources of the infrastructure provided to the FLAME platform are in turn provided as *retail resources* to the media services at the top of the platform through management interfaces exposed to media services. ***In other words, the FLAME platform orchestrates the deployment of media components as well as internal service functions.*** In addition, a monitoring interface allows for information exchange between media services and FLAME platform, which in turn will drive the decisions taken for the management input via the former interface. This combination of management and monitoring interfaces effectively provide the **experiment API** towards media services, allowing for defining and placing the compute, storage and network resources for the specific tests in the experiment. The

Co-funded by the Horizon 2020
Framework Programme of the European Union

experiment API is complemented at the data plane through standard HTTP/IP data plane interfaces of the existing Internet through which media services realize their functionality.

The management interfaces allow for initiating the orchestration of (retail) resources from the media service provider side. We see these interfaces and therefore the dialogue between media services and the FLAME platform evolving over time. In an initial realization, we see media services heavily relying on the ETSI MANO compliant interface, providing details on compute, storage and network resources being utilized in their specific experiment deployment. However, insights obtained from the first experiments, as planned through the FLAME media user partners, will drive the evolution of the FLAME-specific (second) management interface towards one that provides a selection of orchestration templates for specific use cases and therefore removing the need for explicitly defining each orchestration template from scratch, while utilizing the ETSI MANO compliant interface to utilize this evolving template knowledge towards the FLAME platform. This evolution is based on the evolving knowledge through our initial use cases on how to best accommodate demand and supply for specific media service use cases. Through our open calls, this richness of knowledge is meant to expand, e.g., through rich control policies that allows for matching runtime-level demand and supply measurements and feeding them into the control elements of the FLAME platform. Again, instead of needing the media service provider to define all those details through own templates and setups, we see an enriched repository of configurations that is being utilized by the FLAME-specific management interface between media services and FLAME platform rather than utilizing the ETSI MANO compliant variant. *Ultimately, we see the relationship between media services and platform as being realized by high-level KPI-driven service-level agreements (SLAs), expressed through the FLAME-specific management interface and realized through an evolving CLMC on the platform side.*

As indicated in Figure 14, we consider the realization of the **media services** outside the scope of the FLAME platform itself. We do assume, however, that a media service is realized through a set of media components, each communicating with each other through an HTTP/IP-compliant data plane interface, while utilizing the management and monitoring interfaces to the FLAME platform (see below) to facilitate and enable the deployment of those media components through the FLAME platform, i.e. media components are (computing and storage) resources from a FLAME platform perspective. Towards the end user, we see media components utilizing service-specific interfaces and interaction methods. Media components are implemented utilizing platform resources, such as servers, connectivity components (e.g., switches) and others, while also utilizing resources outside of the scope of FLAME, such as end user devices and Internet-of-thing components. *With that in mind, we position FLAME as a distributed programmable resource platform, which can be used by media services for the fulfilment of a desired experience towards users.* It is up to the policy of the FLAME platform provider if resources are provided exclusively to a media service provider or from a shared resource pool. In the latter case, experiments conducted by the media service providers could run concurrently in the system, while the former case provides an exclusive access to the resources for a single media service provider. The governance and specificity of the experiments as well as deployment will likely drive these policies. For instance, exclusive usage of geographically defined resources might make certain media service deployments mutually exclusive since appropriate resources for another deployment are simply not available (see also discussion in Section 2.3).

The **orchestration component** of the FLAME platform interfaces with the infrastructure resource management, through the aforementioned ETSI MANO compliant interface, to manage the compute/storage/network resources at the retail level, i.e. towards the media service provider, while it will utilize the surrogate policy control interface towards the **service function (SF) endpoint management and control component** to realize the orchestration-level management policies as well as to set suitable shorter-term control policies for (surrogate) service function endpoints. For the realization of the configured service function endpoint policies, the service function endpoint

Co-funded by the Horizon 2020
Framework Programme of the European Union

management and control layer will utilize the FQDN registration interface to control the registration and deregistration of the service endpoints towards the **service function routing component**. With this, the 'visibility' of said FQDN towards service requests being routed by said layer can be controlled. The service routing layer, in turn, will use the OpenFlow interface (e.g., via suitable platforms such as ODL [ODL]) to suitably configure the switching fabric of the underlying infrastructure.

Particular consideration is given in our platform to the gathering of information across various layers, realized by the **cross-layer management and control** (CLMC) component. While such data is useful and needed for control-level decisions, such as the activation of service endpoints, it also provides a rich pool of data for the experiments, either as insights towards the creation of further experimentation or insights that reflect directly on the ongoing experiments, e.g., adjusting crucial longer-term strategies such as those for content placement or media adaptation. FLAME will tap into existing monitoring frameworks provided by the individual sub-systems (such as those provided directly by switching platforms such as ODL or ONOS, obtained through the aforementioned OpenFlow interface towards the underlying infrastructure), while also developing solutions for analytics and knowledge management beyond those existing ones.

In Section 6.5, we will elaborate in more detail on the component-level interfaces outlined in Figure 14, including to present an UML-based version. Before that, we will first introduce the service abstraction being used for the platform-internal interactions as well as discuss the management and control aspects relevant to the FLAME platform.

6.2 PLATFORM-INTERNAL SERVICE ABSTRACTION

Before we detail the platform overview in Figure 14 in terms of platform-internal service interactions, we present the abstractions being used for doing so, visualized in Figure 15 as a relationship diagram. It is important to note that this abstraction only applies to the platform-internal interactions, while any interaction to external media services is exposed via well-defined interfaces at the control and data plane level, as shown in Figure 14. *With that, FLAME does not impose any specific abstractions being used for media service composition beyond the mapping of distinct media components onto service functions that in turn are being deployed through the orchestration process of the FLAME platform.*

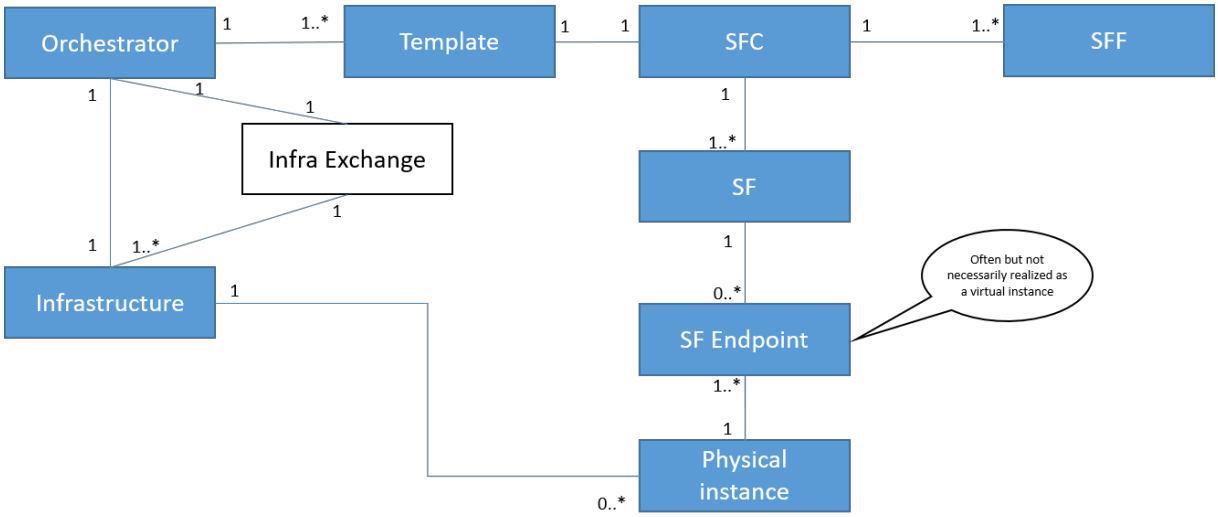

Figure 15: FLAME Platform Internal Service Abstraction

Co-funded by the Horizon 2020 Framework Programme of the European Union

For the interactions within the FLAME platform, we apply the concept of **service function chaining**, as introduced in [SFC7665], which captures service interactions as the serialized execution of **service functions** (SFs), said service functions exchanging messages within defined **service encapsulations** (SEs) through **service function forwarders** (SFFs). Service functions implement the desired platform function in this view, while the SFF provides the interaction mechanism (e.g., inter-process communication, link-local Ethernet based communication, IP-based communication over a routed network) utilizing the specific SE that defines the message format together with the function-specific payload. SFs are usually realized through one or many **SF endpoint**, distributed over a transport network. One or more SF endpoints (of one or more SFs) are executed on a **physical instance** for a compute/storage resource. Often but not necessarily, SF endpoints are realized as *virtual instances* (over a physical instance as a host).

SFCs are usually captured through **templates**, which in turn are utilized by an **orchestrator** to instantiate and control the various SFs as well as the message exchange SFFs over their lifetime (see Section 6.6.2 for the interaction between orchestrator and FLAME-internal SFs). The orchestrator has a one-to-one relationship with the underlying **infrastructure** in case of a single infrastructure operated FLAME platform (see Figure 14 for the utilized southbound interface to the orchestrator of the infrastructure). In the case of multiple infrastructures over which the FLAME platform is operating, we foresee a one-to-one with an **infrastructure exchange**, which in turn manages the one-to-many relationships to the enabling infrastructures over which the FLAME platform operates. This infrastructure exchange fulfils the role of a virtual operator in relation to the FLAME platform and therefore preserves the one-to-one relationship between FLAME orchestrator and the orchestrator of the enabling (virtual) operator of the infrastructure.

6.3 MANAGEMENT & CONTROL IN FLAME

The FLAME platform focuses on flexible management and control of media services and infrastructure resources that are used in ways that deliver enhanced user experience and system performance. The fundamental capabilities supporting this goal are new flexible routing solutions, surrogate management as well as a cross-layer management and control (see Figure 14). We now discuss two aspects of management and control in FLAME. We first start with the distinction of between management and control across an experiment with the FLAME platform, while secondly focussing on the specific aspects of management interactive media systems, as provided by FLAME.

6.3.1 Management Lifecycle

We now elaborate the management lifecycle from the viewpoint of **managing** as well as **controlling** programmable and highly observable resources. We define management and control as follows:

- Management determines infrastructure configuration, resource allocations and variability constraints.

- Control determines how the system responds to requests considering the constraints of management policy and information about current system state (e.g. topology, reachability, etc.).

Management and control can be defined through policies where a policy is a machine-readable document used by the Platform to implement management or control decisions.

It should be noted that “experiment” is a term that describes the context of use and that the lifecycle is equally applicable in all DevOps and business intelligence processes. The fact that we are conducting

Co-funded by the Horizon 2020
Framework Programme of the European Union

an experiment does not change the capabilities needed to manage and control the system. Figure 16 provides an outline of the lifecycle showing the relationship between management motivation (business intelligence, verification and validation, and experimentation down to control within the infrastructure. At the topmost level, the motivation is established setting out an objective with desired outcomes. For an experiment, this could be to test a hypothesis or for business intelligence, this could be to investigate performance of a media service within a specific geographic region. In each case, the decision maker explores service management knowledge to understand how to establish better management and control policies in relation to performance criteria.

Each level has different temporal characteristics. An experiment as a construct that lives over a prolonged period of time (hours or days), reserving the necessary resources for the realization of the experiment through suitable interfaces from the (FLAME) experimentation platform, see Section 6.1. Although we consider the agreement on resource allocation as a technical operation at this level, the overall process is positioned to a large extent at the business level of the system since we see this interaction as being largely intertwined with the governance model of the platform and infrastructure provider.

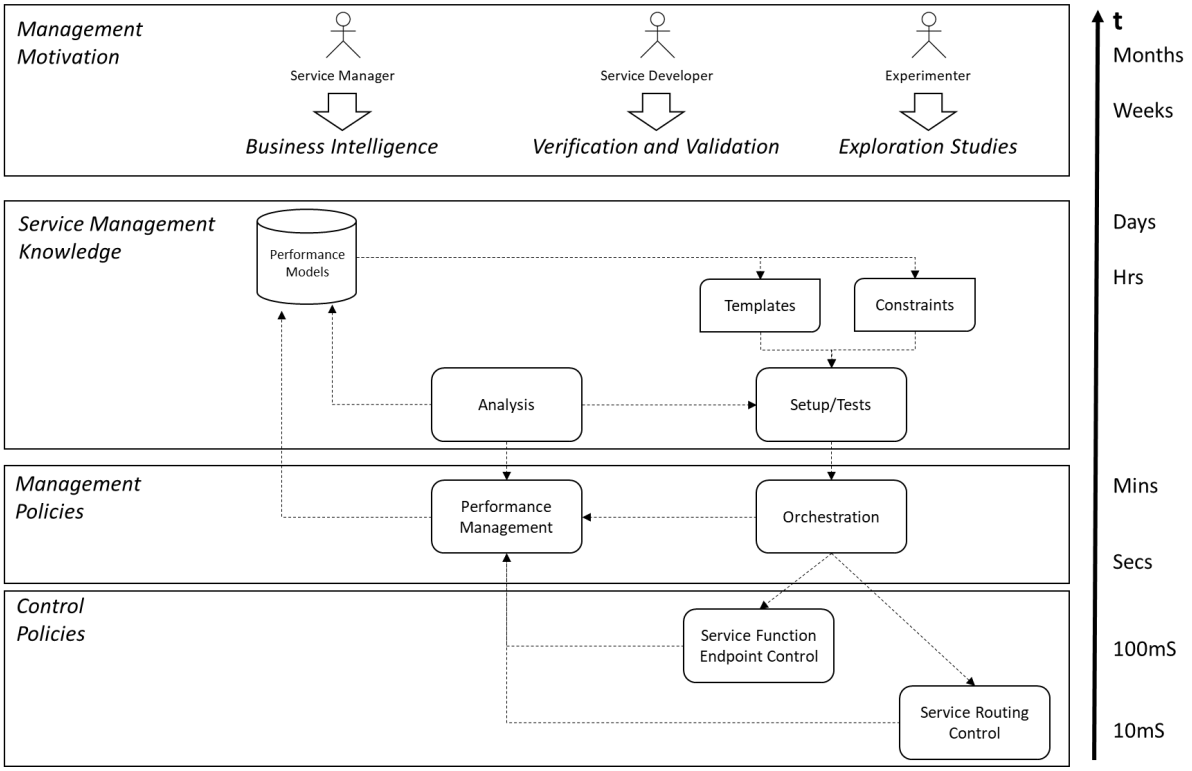

Figure 16: Management & Control in FLAME

Within an experiment, we see several tests being conducted, e.g., in the form of specific interactions with system components, applications being offered (which encode these specific interactions) etc. We position this part of the lifecycle at the management level, realized through **templates** for resource orchestration, together with the **constraints** for the realization of the specific tests. We foresee such templates and constraint models to be either generated specifically for the tests or being utilized from a pool of models that is part of the wider FLAME platform offering (see D2.1 on the current FLAME offering).

Co-funded by the Horizon 2020 Framework Programme of the European Union

The templates utilized for the test are fed into the Management Policy Level using the orchestration capabilities of the platform (see Section 6.1 for the high-level interfaces and layers of the FLAME platform). Orchestration occurs is responsible for the appropriate reservation and instantiation of the compute, storage and network resources for the specified setup. Orchestration also provides lower level control policies to the Platform's service endpoint control and service routing control (see Section 6.1 for a description of those two layers and Section 2.5 for the unique benefits of the FLAME platform, many of which are realized by those two layers). Orchestration occurs within minutes' timescales. At the control policy level, the management of surrogate service endpoints and the routing of service requests between such dynamically controlled surrogate service endpoints is realized within short timespans of (sub-)seconds.

In our lifecycle, we separate management and control through a shift in lifetimes, as indicated on the right hand side of Figure 16. This allows for constructing tests with rich control policies for the service function endpoint control (see Figure 14 for the specific control policy interface being utilized for this in our architecture) being specified. Utilizing the programmable infrastructure being provided by the infrastructure provider, these policies are realized through appropriate southbound interfaces from the platform to the infrastructure.

Alongside this shift from the longer-lived management to the shorter-lived control domain, and the resulting shift in possible tests that utilize these capabilities, we place the Performance Management capabilities of the Platform. Through its monitoring capabilities at the lower control level and its longer-lived analytics and knowledge creation methods, Performance Management spans both control and management domains of the system. Hence, its data and knowledge is utilized for short-lived control decisions (such as instant delay dependent routing decision at the lower levels) as well as longer-lived management decisions (such as changing the placement of resources in certain geographic regions through an analysis of demand and supply of content in the specific test). For the former, a highly responsive data provisioning system (and suitable notification-based APIs) need to be provided by the performance management to control elements at the service function endpoint control and service function routing level.

This understanding of the management and control lifecycle of the Platform is important for creators of test templates as much as for the designers of the Platform since the former drives the proper utilization of Platform capabilities while the latter drives its realization at the architecture and technology level.

6.3.2 Management of Interactive Media Systems

Interactive media systems are complex systems with a large number of shared resources subject to unpredictable requests and affected by external events that cannot be controlled. The systems are typically provided by multiple stakeholders cooperating through service offerings and provisioning of different types of resources to support desired communication process responses.

Figure 17 shows a high-level example ecosystem for streaming content to end users from multiple stakeholders. There are three media service providers (MSP A, MSP B and MSP C) and a platform provider (PP) cooperating to deliver the overall service offering. MSP A offers on-demand videos to end users and uses a 3rd party transcoding service from MSP B. MSP C offers a cataloguing service independently of other MSPs. All Media Service Providers use infrastructure resources offered by the PP. The management of this scenario is achieved through the orchestration of services and resources. The client application orchestrates across the Catalogue and Streaming Services, the Streaming Service orchestrates the Transcoding service whilst the Platform orchestrates the service functions and associated resources required by the media services. These orchestrations are management processes

Co-funded by the Horizon 2020
Framework Programme of the European Union

that are governed by the different independent stakeholders within such a scenario. The MSPs are optimising services in relation to content whilst the PP is optimising services in relation to resources.

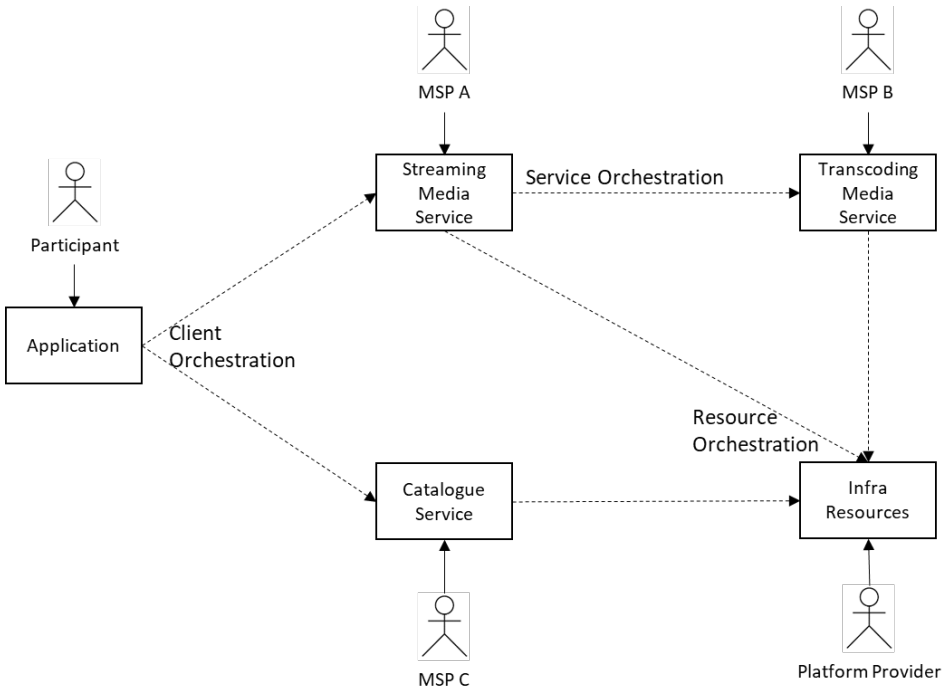

Figure 17: Interactive Media Systems Management

The scenario in Figure 17 shows that management of services and resources in such systems is challenging. The distribution and complexity makes it difficult to have accurate and timely global state information needed for multi-objective optimisation [Mattern89], whilst unpredictable and high frequency interactions within communication processes produces non-deterministic behaviour [Addis12]. As such, we believe that centralised control will not provide adequate solutions to service availability and performance guarantees, considering the complexity, the uncertainty and the multi-stakeholder nature of the systems we are studying.

Generalised processes have been defined (e.g., ISO/IEC 20000, ITIL, MOF, etc.) providing best practice guidance for service and resource management. ITSM products are available that implement the concepts and many service organisations and trained professionals have adopted them [Capterra17]. These guidelines can provide individual stakeholders with descriptions of the types of processes that may to be considered in service management activities. However, the ideas are over 10 years old and subject to an ongoing debate about their value and the changing theories [Betz17]. The problem is that the service lifecycles and associated business processes centred on configuration management tend to be sequential, plan centric and deterministic with minimal consideration of experimentation, agility, DevOps approaches and end-to-end dependability. In addition, as general guidelines, they do not explicitly consider the specific characteristics of the services and resources we are managing or their utility. For example, services based on infrastructure resources have different monitoring and control requirements, in comparison to services based on content resources. If we are considering sharing finer grained resource state then our model and analysis needs to explore the inherent characteristics of resources and the value that sharing specific characteristics would make to service management.

We aim to create knowledge about interactive media systems that allows decision makers within service management processes to understand how services respond to changes in workload and resourcing. Knowledge management is an essential part of a service management underpinning

Co-funded by the Horizon 2020 Framework Programme of the European Union

business operations responsible for delivery of interactive media systems are efficient in terms resource utilisation and effective in terms of meeting customer requirements.

Service knowledge can be structured into the data-information-knowledge-wisdom structure as shown in Figure 18. Data represents discrete facts and numbers, and by themselves have little meaning. Information is generated when data is viewed in context typically involving the use of statistics such as averages or peak and minimum values. Knowledge combines information with experience and is used as a basis for decision-making or taking an action. Wisdom can be created by taking advantage of all the knowledge available, such as recognizing that a recent deterioration of service performance coincided with the adoption of a new resource allocation policy.

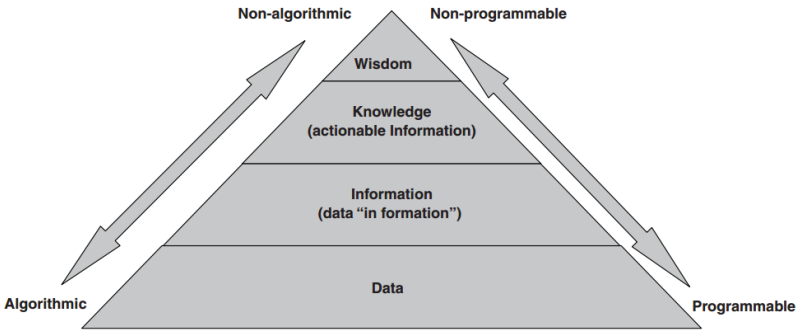

Figure 18: DIKW Pyramid

Service knowledge management can be considered a big data challenge that requires integration of large structured, semi-structured and unstructured data sets within time sensitive business processes. The data and information can be sourced from service request management (demand and response), configuration management, software repositories, and asset management (physical and virtual). Information can be integrated through service definitions, process definitions and integration schema definitions, along with information integration processes for reconciliation and spatial/temporal synchronisation. Integrated information can then be used to support activities such as capacity management, performance and availability management and event (e.g., security) management.

What is clear is that this situation is complex and we need to scope our work accordingly to ensure we maximise the impact of our contribution and that we don't repeat what the market is doing anyway. Much of the industry best practice for service management and associated knowledge management focuses on a single stakeholder view. This is part of the story but insufficient to explore the end-to-end requirements of communication processes within an ecosystem. We are not proposing to replace the single stakeholder approaches but exploring through experimentation end-to-end communication systems and focusing on the interactions between individual stakeholders involved. This is the focus of Performance Management within the FLAME Platform.

6.3.2.1 Working Assumptions and Challenges

This discussion in the previous section leads to our grounding assumptions for modelling and analysing interactive media systems:

- An interactive media system is delivered by multi-stakeholders offering combinations of services and underlying resources to support desired end-to-end communication process responses.

- Each stakeholder owns and controls resources required by the services they offer.

Co-funded by the Horizon 2020
Framework Programme of the European Union

- Each stakeholder may access other resources through services offered by other stakeholders.
- Each stakeholder uses existing service management tools and processes to manage resources and commitments to customers according to their own defined multi-criteria key performance indicators (KPIs).
- The current situation we are dealing with is OTT content delivered over the Internet by Media Service Providers without the involvement of a Network Operator in the control or distribution of the content. Each stakeholder independently optimises resource utilisation and service quality through processes to monitor and predict sufficient system state information, e.g., provided through MPEG-DASH [MPEGDASH], needed for autonomic and human decision-making.
- The desired situation we are targeting is for platform capabilities providing mechanisms for Media Service Providers and Network Operators to work together in the management of resources required for control and distribution of content. Improving the quality of state information shared between stakeholders and control mechanisms has the potential to improve the quality of decision making in service management processes.

These assumptions scope our approach:

- We should position our approach in the context of service management processes but should not necessarily adopt a specific set of guidelines.
- We should not replace existing service management mechanisms used by each stakeholder with a new global management mechanism that optimises across all resources, as this would be futile.
- We should explore ways to improve the efficiently and effectiveness of existing service management mechanisms contributing to end-to-end delivery of a communication process through appropriate sharing of information and control.

The challenges we must therefore address include:

- How to capture end-to-end content flows between multiple stakeholders and relate these to allocated resources?
- How to improve the dependability (i.e. accuracy and consistency) of shared state in communication processes considering the distributed nature of interactive media systems?
- How to increase completeness and availability of state in communication processes related to resources under the control of other stakeholders whilst considering the need to constrain complexity of optimisation?
- How to define conditions under which delegation of direct control of resources between stakeholders is acceptable?

6.3.3 Cross-Layer Management and Control

Cross-Layer Management and Control (CLMC) is responsible monitoring, measurement and assessment of the performance of Media Services delivered by the Platform. The goal is to create knowledge about interactive media systems that allows multi-stakeholder decision makers within

Co-funded by the Horizon 2020
Framework Programme of the European Union

service management processes to understand how services respond to changes in workload and resourcing. The knowledge gained will be used to design and adapt management and control policies associated with Service Request Management, Fault Management and Configuration Management processes[2]. There are many possible management and control decisions and it is the purpose of the CLMC to provide decision makers with empirical knowledge to design and implement better policies.

CLMC uses Key Performance Indicators (KPIs) to measure performance. KPIs are defined for different aspects of the system and associated with service management processes responsible for ensuring that the desired performance is met. KPIs define what success looks like for an interactive media communication process. Table 13 provides examples of media service KPIs based on the use cases defined in Section 4. An SF could be specified with a requirement for an average response time of <100mS for 95% of requests. It is the responsibility of Service Request Management processes to fulfil this KPI through implementation of management and control policies.

Under typical conditions, platform control policies could be established to route requests on the shortest path to an SF or horizontally-scale a compute resource allocated to SFs if demand reaches a specific threshold. If the KPI is at risk of being breached (e.g. RT is <100ms for 96% requests) the platform management policies may redistribute SFs to hosts in ways that are more optimally aligned with the workload. Management policies may be used to breach the KPI if the liability is deemed acceptable or may even inform the Media Service that resources are constrained which results in a Media Service deciding to adapt content in ways that maintains response time KPI whilst reducing content quality. This last case is an example of cross-layer management where the Platform Provider shares resource state information with a Media Service Provider.

KPI	Description	Service Management Process
Service Response Time	Percentage of service requests fulfilled within an acceptable time. E.g. response time <100mS for 95% of requests	Service Request Management
Availability	Percentage of actual uptime of services relative to the total numbers of planned uptime. E.g. 95% uptime over 24hrs	Fault Management
Quality of Experience	Percentage of service requests fulfilled with sufficient content quality e.g. 80% of service requests fulfilled with requested target quality or better	Service Request Management

Table 13: Example Key Performance Indicators (KPIs)

6.3.4 Cross-Layer Knowledge Model

The knowledge model supports multidimensional analysis of KPIs. Each KPI is an important fact about a process that needs to be analysed. The context for KPI measurements are defined as dimensions such as space, time, resources. By recording and predicting context based on historical access information and additional insights from media services themselves demand patterns can be established and resources planned including scaling rules/constraints for situations where actual usage deviates from planned usage.

[2] These are a subset of all possible management process, selected due to their strong relationship to the adaptive service delivery capabilities of the Platform.

Co-funded by the Horizon 2020 Framework Programme of the European Union

The knowledge model is implemented using the principles of OLAP[3]. In OLAP, information is structured to support analysis of KPIs in relation to a set of dimensions. OLAP directly supports the exploration of the relationship between KPIs and dimensions through processes for pivoting, slicing, dicing and drilling the data. The design of KPIs, dimensions and measurement procedures is a critical for CLMC.

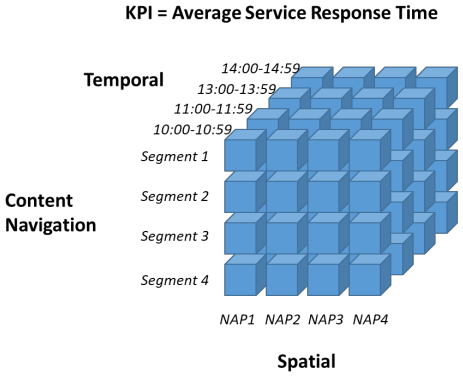

Figure 19: Example multidimensional analysis of KPI - Average Service Response Time.

Figure 19 shows an OLAP cube for Average Service Response time by content navigation, space and time. In this example, a query can be created to establish the response time for "all", "NAPs between 10:00 and 14:59" or just for "segment 4 at NAP1, between 14:00 and 14:59". The dimensions of data will initially be based on expert knowledge but it is expected that within the lifetime of FLAME that empirical data created through experimentation will allow for dimension classes and hierarchies to be derived using semi-supervised machine learning techniques.

Figure 20 shows an initial set of dimensions supporting multidimensional analysis of interactive media systems. These are further elaborated in Table 14. The dimensions allow performance measures by queried through contextual filters, for example:

- Query for Service Response Time by content navigation, by content representation, by user profile, by space, by time
- Query for Service Response Time by resource configuration, by service configuration, by space, by time

The performance of the knowledge model itself is important concern considering the lifecycle requirements of management (Sec-Min) and control decisions (10mS -100mS). The model must consider when precomputed data is needed to optimise frequent low-latency queries forming part of fast control loops. Here the design must consider factors such as pre-processing costs, pre-defined query subsets at low latency and query flexibility. For exploratory analysis, typically conducted by humans it may not be necessary to precompute data, but rather access the data from the underlying storage systems by querying on user requests. The response time may be relatively slow, but this reduces pre-processing and supports a wider range of queries. Due to the experimental nature of the platform usage, it is also important to be able to flexibility define dimensions and queries around KPIs.

[3] http://olap.com/olap-definition/

Co-funded by the Horizon 2020
Framework Programme of the European Union

Although some more general KPIs may be defined in advance, we cannot know a priori all KPIs relevant to all media service providers as these will emerge through experiment design.

Figure 20: Multiple Dimensions of KPIs

Dimension	Description	Relevant Standards and Tools
Spatial	Used to determine the location of system resources and users in physical space. Physical spaces use geospatial coordinates (longitude, latitude, altitude).	GPS
Temporal	Used to determine change system state over a period.	MPEG-7, ISO 8601
Content Navigation	Used to classify the retrieval of content objects, partitions and decompositions. This includes temporal and spatial navigation within content objects themselves.	MPEG-7
Content Representation	Used to classify encoding alternatives of content	MPEG-DASH, MPEG-4, MPEG-7
User	Used to classify profiles of users	
Service Function	Used to group compositions of service functions and their characteristics	TOSCA, RSPEC
Resource Configuration	Used to classify the resources allocated and configured to deliver a media service	TOSCA, RSPEC
Service Function Configuration	Used to classify service function configurations that define how content is delivered in response to service requests	Puppet, Chef, JuJu, Ansible

Table 14: Dimensions of analysis of interactive media systems

Figure 21 shows example spatial dimension hierarchies used to cluster related system components (e.g. hosts, SFs) and external users within physical space. Location is measured using the geographic coordinate system. Location can be asserted directly in configuration for fixed entities (e.g. hosts in data centres or street cabinets), or dynamically for mobile entities (e.g. user devices, hosts in transport). The primary purpose of spatial dimensions is to understand proximity for decisions related to mobility and localisation. For example, the Platform may want to know the proximity of user populations to NAPs and SF instances. We might define a KPI “Number of Connected Clients” and want to predict how many potential connected clients exist in a given space, adapting resources based on that prediction. If a client has connected to a NAP then the Platform already knows the user’s location

Co-funded by the Horizon 2020
Framework Programme of the European Union

to an accuracy based on the range of the NAP. The Platform also know about the user's previous movements based on where they have connected over time.

In addition, Media Services can provide further spatial information to the Platform about the expected location of users over time, subject of course to consent by users themselves. For example, if a user is accessing a smart city guide, the personalised route gives some indication of expected future locations. The same is true for live events where large populations of users are expected to converge to and diverge from a specific location. Geo-located content is another source of spatial context that can be planned, for example virtual objects delivered to AR applications at locations within a city. It should be noted that the spatial consistency is important for AR applications where correlations between physical location and virtual objects is important. However, spatial consistency of such content is the responsibly of media services rather than the platform itself

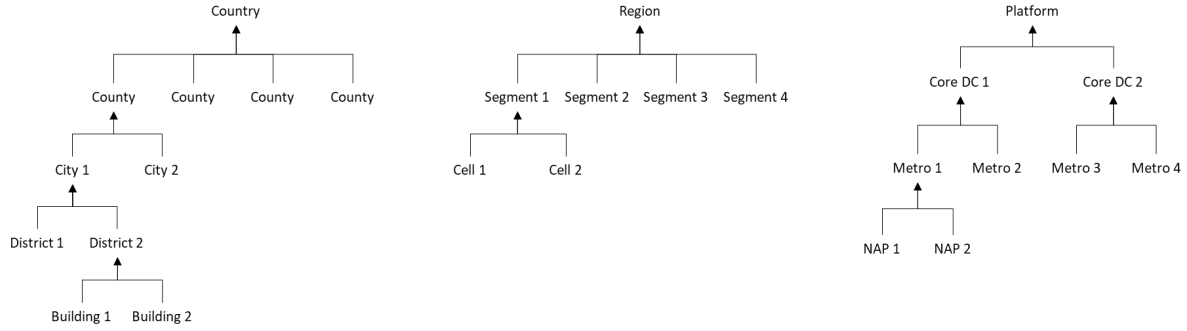

Figure 21: Spatial dimension hierarchies

Figure 22 shows example temporal dimensions used to determine change in system state over a period. The period selected depends on the frequency of decision making, for example, a service manager exploring the sales KPI from a specific service may be concerned with annualised temporal dimensions whilst a Platform orchestrator deciding how much resource to allocate to a service function may be concerned with the usage over the last 30 seconds.

Time synchronisation for monitoring and measurement of performance, usage and other key events is important to correlation of measurements between different distributed computers. Time synchronisation for data correlation and visualisation is important and accuracy should be 20mS for management and 1mS for control. A Timing Service is provided, which offers Platform Time to all service functions (media service and platform). The Timing Service is hosted on a machine that calibrated with a reliable timing source using Network Timing Protocol (NTP). The Timing Service operates in master/slave mode allowing clients to synchronise their time to the server using HTTP. Service functions are synchronised either periodically or at well-defined events within the lifecycle. Clock drift needs to be considered for long running service functions.

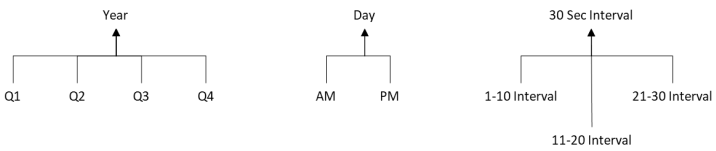

Figure 22: Temporal dimension hierarchies

Figure 23 shows the content navigation dimension that is used to classify the retrieval of content objects, partitions and decompositions. This includes temporal and spatial navigation within content

Co-funded by the Horizon 2020
Framework Programme of the European Union

objects themselves. Content navigation classifications is complex and are described through existing content metadata schemes such as MPD in MPEG-DASH, MPEG-7 or MXF. For each user interaction, the service request must be processed to record the specific navigation action within the content. For example, for MPEG-DASH requests the video segment requested would be logged whilst a file-based post-production workflow or ingest service might refer to an individual file and folders.

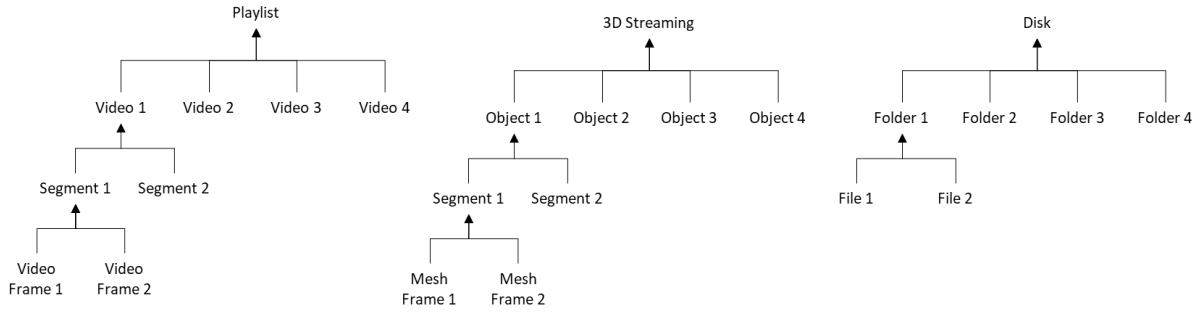

Figure 23: Content navigation dimension hierarchies

Figure 24 shows examples of content representation dimensions for video and audio content. Content representation describes the encoding format (e.g., fidelity, language, etc.) using references to existing standards such as MPEG and H.264. We separate content navigation from content representation to capture situations where the same content is delivered in different formats. For example, an MPEG-DASH service can deliver multiple content representations for the same video at different levels of quality whilst in MPEG-7 the concept of "Variation" is used to indicate different types of representation.

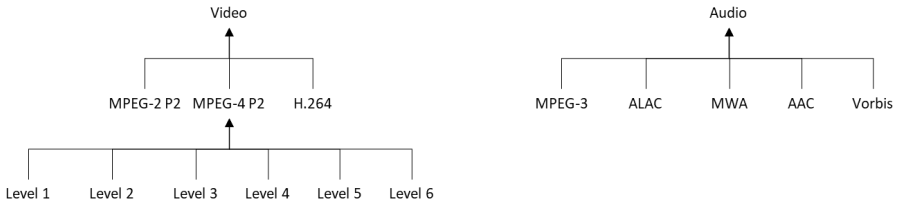

Figure 24: Content representation dimension hierarchies

Figure 25 provides an example of a user dimension hierarchy used to classify and profile users within a population. We are not prescriptive for the user dimension.

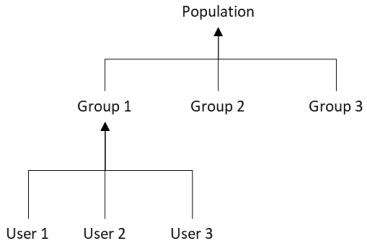

Figure 25: User dimension hierarchy

We are also unaware at the time of writing of available standards for user profiling. We recognise that there are many ways to profile users based on features such as personality, character, attitude, preference and intentions, and that factors most influential to management and control will be highly

Co-funded by the Horizon 2020
Framework Programme of the European Union

variable across populations. The identification of user groups as intermediate abstractions of users and relation to demand will be an important aspect of experimental analytics.

Figure 26 shows examples of service function dimensions. The first case is a hierarchical view of all service functions on the platform using decomposing the platform into media services containing 1-n SFs which themselves have 1-n instances during runtime. Constructing this sort of hierarchy allows the individual performance of a SF instance (e.g. Response Time) to be aggregated across instances of the same type, within the SF of a media service, or for the overall platform. We may also classify SFs in terms of their processing characteristics as shown in the second example where SF's are classified according to an intermediate abstraction for application benchmarks called the Dwarf Taxonomy. Other lower level benchmarks exist such as LINPACK, SPECint and SPECfp. Using such classifications, applications can be modelled in such a way that its performance can be predicted on a range of infrastructure resource specifications.

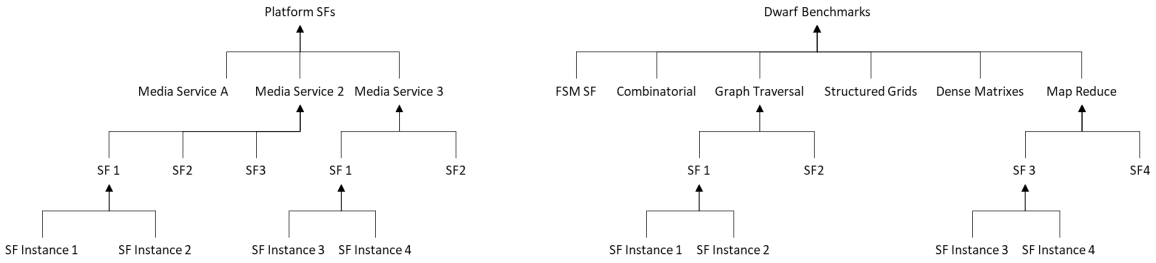

Figure 26: Service function dimension hierarchies

Figure 27 shows an example dimension hierarchy for Resource Configuration used to classify the resources allocated and configured for media service delivery. Resource Configuration is defined through labels that identify meaningful collections of resources and their characteristics. This is similar to the way instance sizes are defined in Cloud computing resource provision. It is important to be able to differentiate between different resource configurations when assessing KPIs and for test suites aiming to optimise performance by variation in resource configuration.

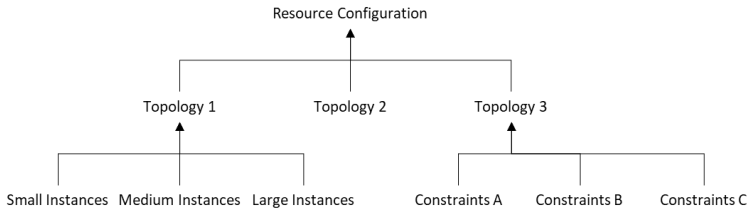

Figure 27: Resource configuration dimension hierarchy

Figure 28 shows an example hierarchy for service function configuration used the way SFs are configured to respond to service requests. SF configurations are highly specific to the SF itself and generalisations of classification across such heterogeneity is difficult. As with Resource Configuration, what is important is to be able to identify different SF Configurations and their influence on KPIs. The representation of SF configurations is also heterogeneous and this depends on the deployment technology selected, for example, Chef, Puppet, Ansible, Juju or other was of automatically scripting the deployment and configuration of SFs.

Co-funded by the Horizon 2020
Framework Programme of the European Union

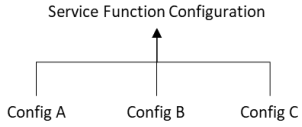

Figure 28: Service configuration dimension hierarchy

6.3.5 Cross-Layer Knowledge Model and PIML

In Section 3.3.1, we describe a high-level domain model for interactive media systems. We describe demand in terms of personalisation and interaction, which is a simplified and refined definition of the PIML characterisation. We now discuss how KPIs and dimensions can be associated with this demand characterisation, as shown in Figure 29. Please note that this figure is not an OLAP dimension hierarchy like those in the previous section as the classes at each level are not of the same type. What we show is that KPIs can be defined for interaction, interaction requests and interaction responses, and that each KPI has context in terms of dimensions.

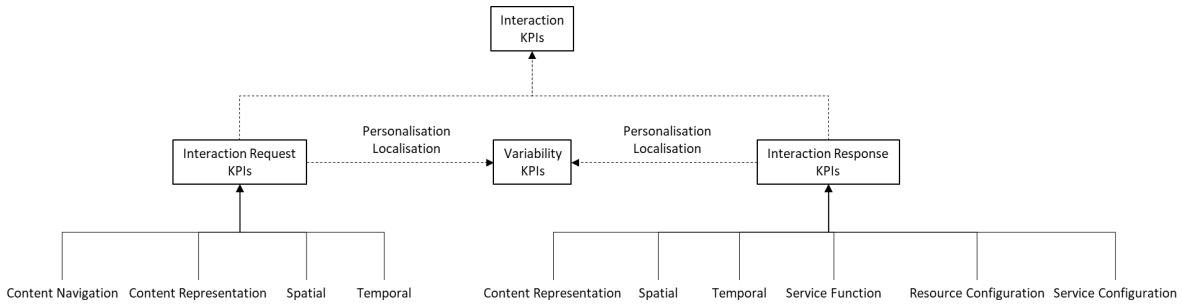

Figure 29: KPIs and Dimensions associated with demand characterisation in the domain model

Interaction KPIs can be calculated through measurement functions applied to request and response measures (e.g. Average Service Response Time). Most service quality KPIs are at the interaction level as it relates to the service quality delivered to users. All dimensions described above can provide context for interaction KPIs. Interaction Request KPIs (e.g. Total Request Rate, Number of Requests) relate purely to demand. The dimensions of Content Navigation and Content Representation are the "Functional" aspect of demand, the "what", whilst the Spatial and Temporal dimensions are self-explanatory. The Interaction Response KPIs (e.g. Total Response Rate) are related to the Content Representation delivered and the aspects of the communication system (Service Function, Resource Configuration and Service Configuration).

Personalisation is the process of responding to requirements from individual users. ***Localisation*** is the process of responding to requirements from groups of users. Both can result in processes for adaptation of content resources and adaptation of infrastructure resources. From the platform perspective, these can be measured as simply the degree of variance in a set of Interaction Requests or Interaction Responses. For example, if 10 users request the same video at the same time from the same access point then the level of personalisation required is low. However, if 10 users request different videos on demand at different times of the day from locations throughout the city then the level of personalisation is much higher. If all users are in the same street, then the service functions could be localised at a single NAP, and if they all speak French then the films available could be localised to only those in French or with subtitles. The key point is that high levels of variance due to personalisation and localisation increase costs and are relevant to Service Efficiency KPIs.

Co-funded by the Horizon 2020
Framework Programme of the European Union

6.3.6 Data Quality

Data quality is an important factor in CLMC to ensure that KPIs and dimensions are fit for decision making in service management processes. Data quality includes aspects such as accuracy, completeness, timeliness and consistency. Data quality is affected by the all processes for observation and measurement such as sensing, integration and aggregation of data within the knowledge model. The CLMC aims to improve the quality of system state in management decisions by sourcing new cross-layer sources of information previously not available.

The CLMC brings together information producers observing and measuring state of different system components with information consumers responsible for management and control decisions. Information producers and consumers are at both service and platform levels. Information producers provide data that is processed to create KPIs and dimensions. The measurement processes used to produce the data must be defined in terms of data quality characteristics and variabilities. Information consumers express data quality requirements for KPIs and dimension data on which they depend on with the available variabilities. Table 15 provides a description of data quality metrics in relation to cross-layer management and control.

Data Quality	Description	CLMC
Accuracy	Data should be sufficiently accurate for the intended use and should be captured only once, although it may have multiple uses.	Dimension granularity (e.g. spatial-temporal resolution, class boundaries); Representation consistency (e.g. time format, class labels)
Completeness	Data collection processes should match clearly specified information needs. The difference between the data needed and the data available.	Content Navigation, Content Format, Spatial and Temporal must be available.
Consistency	Data should be synchronised with different spatial, temporal and relational data within a given time period	Content Navigation must be consistent with the Service Function Configuration and Resource Configuration used to support the navigation; Measurement of state must be temporally synchronised to 100mS.
Timeliness	Data should be captured as quickly as possible after the event or activity and must be available for the intended use within an expected time period. The time between when data is expected and when it is available for use.	The data is available for a decision 95% of the time
Currency	Data must be available quickly and frequently enough to support information needs and to influence service or management decisions. The time between the latest available data time and the current time.	The data is sampled at a frequency of 10Hz
Relevance	Data captured should be relevant to the purposes for which it is to be used. The closeness between data and the desired outputs. The filtering of available data sets to those needed for decision making	Service Response Time by Service Function is needed to control the placement of surrogate services.
Transparency	Data should be traceable back to the origin of data. The access to audit trail and provenance data.	A record of the process used to generate a measurement is available.
Accessibility	Data should be accessible to those that are authorised to do so	A user consents to allow a service provider to share data with a platform provider to improve quality of a specific media service; A platform provider authorises a service provider to access performance data for a specific resource configuration

Table 15: Data quality in relation to CLMC

The CLMC is configured to meet the goals of information consumers considering data quality constraints. The configuration of the CLMC will be driven through experimentation as experiments

Co-funded by the Horizon 2020 Framework Programme of the European Union

define management and control decisions, relevant KPIs, relevant dimensions of data and components within the system-under-test being monitored and controlled.

6.4 INFORMATION SECURITY AND PRIVACY

Information security is concerned with ensuring that only data is protected against unauthorised use. When such information is personally identifiable or sensitive, privacy concerns must be addressed in respect to collection, storage, usage and retention. Security must be considered for processes on the management and control plane where outcomes of such processes result in permissions to access data plane. Cross-layer management and control potentially includes commercially sensitive (e.g. number of users accessing a content resource), personally identifiable (e.g. Alice is interested in comedy films) and sensitive information (e.g. John visited location X, Y and Z between 10:00 and 11:00).

A formal procedure for ethical oversight has been defined in D1.1 NEC - Requirement No. 1 and D1.2 D1.2 NEC - Requirement No. 3. From a technical point of view the impact of the ethics on the architecture deals with data management and process. FLAME is a multi-domain environment bring together Media Service Providers, Platform Providers, Infrastructure Providers and Participants through an Internet accessible hosted Platform. Each of these organisations will operate a distinct security domain and will have responsibilities for security policy within their respective domains. These responsibilities will include:

- Ensure compliance with current laws, regulations and guidelines.
- Comply with requirements for confidentiality, integrity and availability for an organisations employees, students and other users.
- Establish controls for protecting the organisations information and information systems against theft, abuse and other forms of harm and loss.
- Motivate administrators and employees to maintain the responsibility for, ownership of and knowledge about information security, in order to minimize the risk of security incidents.
- Ensure the protection of personal data (privacy).
- Ensure the availability and reliability of the network infrastructure and the services supplied and operated by the organisation
- Comply with methods from standards for information security, e.g. ISO/IEC 27001.
- Ensure that external service providers comply with the organisation’s information security needs and requirements.
- Restrict access to external web sites and services where necessary to mitigate child safety threats

The platform itself must offer various security counter measures to ensure data confidentiality and privacy whilst protecting the services offered by the platform from malicious or accidental threats.

Privacy should also be enabled in the platform. In particular, it is necessary to:

Co-funded by the Horizon 2020
Framework Programme of the European Union

- support data anonymization, so that the association between the identity records and other information cannot be inferred if data were to get stolen or accidentally disclosed in some way outside the platform;

- provide opt out options that allow specific observation acquisition features (like video or audio) to be disabled on request;

- distinguish between the data accessed by different stakeholders hosted on the platform, the reasons for accessing the data and transparency in processes that use personal data;

- enable user profile browsing only to authorized stakeholders.

The security architecture brings together people, processes, and tools to protect data, service and infrastructure assets owned by platform stakeholders. The architecture is driven by regulatory requirements for data protection and privacy, and is structured to efficiently and effectively support platform business processes.

6.4.1 Security Domains

The security architecture defines a set of distinct security domains that delimitate physical and logical units where a single and homogeneous security policy is valid and applied. Each domain has security policy that controls the behaviour of the security services being provided. Secure interaction between domains must be in accordance with agreed security policy governing this interaction. Agreed policies must be implemented within each individual domain. FLAME operates in a multi-domain environment where logical domains span multiple physical domains. The physical security domains (i.e. sites, platform and network) and their respective trust relationships. Each domain must implement security policy to protect their respective assets in accordance with the trusted relationships:

- **Infrastructure domain**: A site that owns physical infrastructure that is used to deliver media services to participants, typically a building or a venue but also includes public spaces offered within cities. The infrastructure provider is the data controller for subjects within the domain and responsible for ensuring data protection including ensuring processing of data conducted by 3rd parties such as the Platform is performed in accordance with relevant regulations (e.g. EU Directive 95/46 and nation state implementations). The Infrastructure Provider must operate an IT infrastructure to manage and deliver media services including servers, desktops, laptops, and other devices connected to FLAME network entry points. The infrastructure must be protected in accordance with standards for information security (e.g., ISO/IEC 27001).

- **Platform Provider Domain**: A physical or logical subnetwork that hosts all platform services and data assets in a distributed infrastructure environment. The Platform Provider is a Data Controller responsible for processing personal information for improving service quality and efficiency. The Platform Provider must comply with all relevant data protection regulation and apply all information security standards for clouds (e.g., ISO/IEC 27017).

- **Media Service Provider Domain**: A logical domain offering media services typically in a commercial context to participants. The MSP may be a virtual organisation where developers are collaborating across multiple sites. Early service development must use anonymised test data sets to verify and validate function and performance. User testing in trials must be apply all relevant security policy for production use as required by Data Controllers.

- **Internet Domain**: An untrusted network operated by a ISPs and Network Operators to provide global connectivity. Stakeholders within this domain have no responsibility for policies in

Co-funded by the Horizon 2020
Framework Programme of the European Union

relation to over-the-top applications running on their network and as such the Internet Domain is untrusted.

6.4.2 Identity Management

Identify management is concerned with controlling information that is used to identify platform users and services for allowing access control permissions to be assigned and evaluated against this identify. The platform provides single sign through one credential for all offered platform services. The platform aims to support OAuth 2.0 offering flexibility for users to be able to use accounts on different providers and to selectively provide access to platform services.

6.4.3 Access Control

The platform restricts access to resources to users that are authenticated and authorised in accordance with well-defined business processes and security policy consequences. Access control policies define rules that specify access privileges to protected resources. The Platform possesses resources and services need to be protected, managed, and monitored. Access control policies define the permissions and usage of these resources by outlining when and how a user can perform an action on a given resource. A policy consists of rules, subjects, and conditions that must be satisfied for access to be permitted. Access control policies are created and updated because of platform business processes. When a user takes an action on a service or resource the action results in policy updates. Each platform component defines business processes that access platform resources and security consequences.

6.4.4 Border Control

All domains must operate a firewall to restrict incoming and outgoing connections according to their security policy. Some Media Service Provider and Infrastructure domains will be restricted to outgoing initiated connections on Port 80 (HTTP) or Port 443 (HTTPS). The FLAME data plane API (see Section 6.5.4) is based on the HTTPS protocol, i.e., utilizing TLS (transport layer security).

6.4.5 Entry Point Security

The Platform Provider operates a security gateway like functionality at the entry point to the network, providing policy enforcement point for access to all platform services. All external access to the platform enters through said entry point. The gateway provides translation from FQDN to internal service functions hosted within Platform Provider's internal trusted domain. This functionality is realized at the service routing function (see Section 6.5.4) together with infrastructure-level access control at link level, e.g., utilizing 802.11x standard based solutions.

6.4.6 Transport Layer Security

Interaction between devices on the Internet domain and Media Service Provider Domains is over an untrusted network, the Internet. To secure interaction over this channel, FLAME uses protocols based on HTTPS using strong encryption.

6.4.7 Authority Delegation

Name as well as content authority usually lies with the owner of a fully qualified domain name. Delegating such authority is crucial, e.g., for content delivery where delegation servers provide authorized content on behalf of the owner. The FLAME platform provides suitable interfaces for such delegation, e.g., in the registration of FQDN-based media service endpoints at specific NAPs (see Section 6.5.4 for the specific interface).

Co-funded by the Horizon 2020
Framework Programme of the European Union

6.5 COMPONENT INTERFACES

We now present the main component interfaces of our architecture in Figure 14. We will reference these interfaces throughout the SFCs presented in Section 6.6. For this, we transform the component architecture in Figure 14 into an UML version, presented in Figure 30.

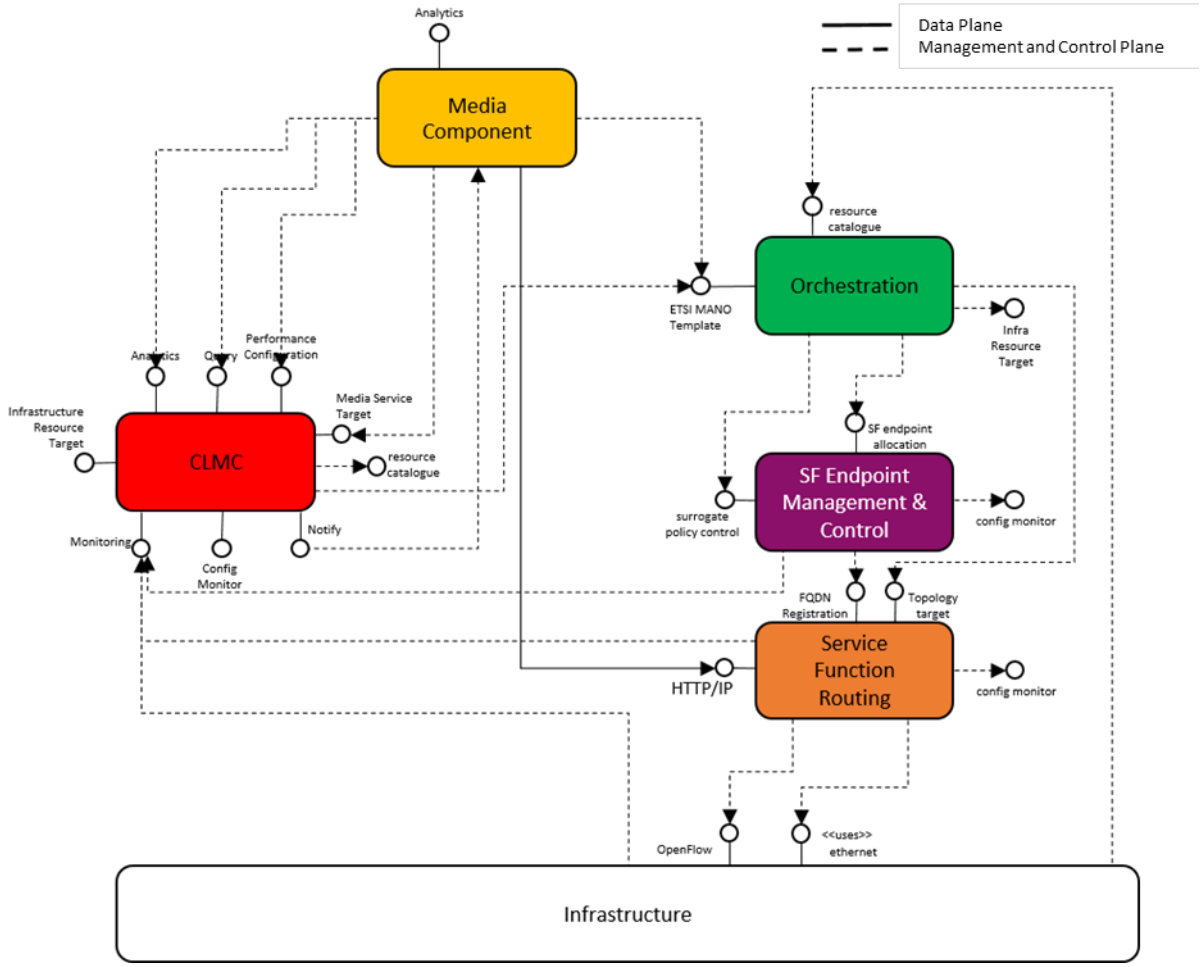

Figure 30: FLAME Platform Architecture – UML version

In the following sub-sections, we outline the interfaces of each of the components in more detail.

6.5.1 Media Component

As outlined in Section 6.1, a media service is comprised of one or more media components, implementing specific sub-functions of the overall service. Although FLAME does not prescribe the specific abstractions or methods being used for the decomposition nor any specific capabilities being realized in the media component, we do assume data plane traffic of the media component to utilize HTTP or other IP-based protocols, see Figure 30. In addition, we expect media components to expose an optional ***Analytics*** interface, providing access to media-related analytics, such as frame rates, drop rates for video content, usage statistics, and alike. For this, there are a number of widely accepted standard KPI definitions, while we expect an HTTP/REST API realization for this interface.

Co-funded by the Horizon 2020
Framework Programme of the European Union

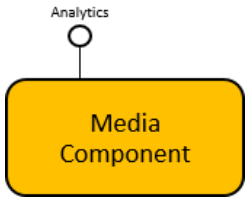

Figure 31: Media Component

No.	Interface	Purpose	Outcome	Relevant Specifications
M1	Analytics	Retrieve media-related statistics and analytics performed by the media component	Persistent set of media statistics and analytics data	

Table 16: Media component Interfaces

6.5.2 Orchestration

Orchestration is concerned with mapping a desired resource configuration onto the resource constraints as defined by the availability of resources at the infrastructure level. The general capabilities of the interfaces are shown in Figure 32 and further elaborated in Table 17. The further decomposition of this component is presented in Sections 6.6.1 and 6.6.2.

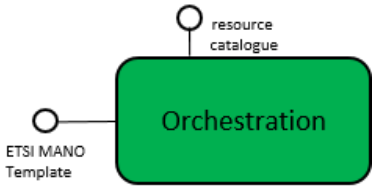

Figure 32: Orchestration Component

No.	Interface	Purpose	Outcome	Relevant Specifications
O1	ETSI MANO	Provide a template-level description of desired resource configurations for a media service to be deployed. Access control to initiate orchestration is included in this interface.	Persistent set of service function endpoints and their topological relationship	ETSI MANO
O2	Resource catalogue	Provide a template-level description of available resources (and their constraints) at the infrastructure level in which the media service template needs to operate. Furthermore, provide suitable infrastructure-level resource handles allowing for the management of the resource instances.	Resource catalogue derived from the wholesale infrastructure (slice) template as well as infra-level resource handles for management	ETSI MANO

Table 17: Orchestration Interfaces

Templates, both at the O1 and O2 interfaces, are based on ongoing specification work in ETSI MANO [NFVMANO]. The infrastructure provides a resource catalogue in the form of such template, referred to as the **slice template**. This template provides the constraining framework, representing the wholesale resources, within which the set of all media service templates need to operate. A specific

Co-funded by the Horizon 2020 Framework Programme of the European Union

media service template is provided at the O1 interface to the orchestration component, either from the CLMC component or the media component directly. Section 6.6.1 will elaborate on the different usage for these two types of access to the orchestration template.

The resource catalogue, provided by the O2 interface is utilized by the CLMC component for capacity management, i.e., for planning the dimensioning of the various media service templates within the constraints set out by the slice template provided by the O2 interface. The infrastructure-level resource handles provided by the O2 interface are utilized by the orchestrator to initiate the suitable service endpoint management and control by providing such information via the SFEMC1, see below.

6.5.3 Service Function Endpoint Management & Control

This component is concerned with the management and control decisions pertaining to SF endpoints. The management decisions concern the placement and instantiation of suitable virtual instances, while control decisions concern the visibility of such SF endpoints in the overall service path, e.g., realizing use cases for alternative playout points as described in Section 4.1.1.3. The general capabilities of the interfaces are shown in Figure 33 and further elaborated in Table 18. The further decomposition of this component is presented in Section 6.6.2.

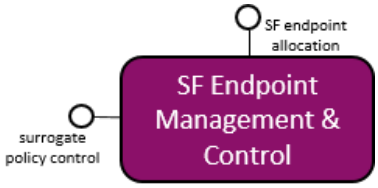

Figure 33: Service Function Endpoint Management & Control Component

No.	Interface	Purpose	Outcome	Relevant Specifications
SFEMC1	SF endpoint allocation	Provides virtual instances of SF endpoints based on received orchestration information, utilizing the received infrastructure-level resource handles.	Persistent set of virtual instance handles to SF endpoints for future management and control decisions	ETSI MANO
SFEMC2	Surrogate policy control	Provides suitable orchestration information for the control of SF endpoint states from PLACED over BOOTED to CONNECTED (see Section 6.6.2 for more information on these states)	Persistent set of SF endpoint specific control policies being monitored and executed by the component	ETSI MANO

Table 18: Service Function Endpoint Management & Control Interfaces

Via the SFEMC1 interface, suitable orchestration information is received based on realizations of the ETSI MANO specifications. The component utilizes the infrastructure-level resource handles received via this interface to realize the functionality of a virtual instance manager (VIM) [NFVMANO], placing suitable SF endpoints and instantiating the necessary virtualized resources. Via the SFEMC2 interface, control policies will be provided, derived either from existing or evolving ETSI MANO service templates, that define the SF endpoint specific control behaviour within the overall media service deployment. While current ETSI MANO compliant templates include little of such control policy, FLAME will investigate other control policy approaches with the final aim to integrate such policies into a coherent overall template description for the media service.

Co-funded by the Horizon 2020 Framework Programme of the European Union

6.5.4 Service Function Routing

The Service Function Routing component provides a number of control plane as well as a data plane interface for the purpose of efficiently routing service requests of any SF endpoint to one or more others. The general capabilities of the interfaces are shown in Figure 34 and further elaborated in Table 19. The further decomposition of this component is presented in Section 6.6.4.

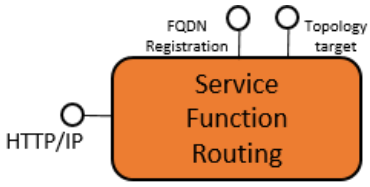

Figure 34: Service Function Routing Component

No.	Interface	Purpose	Outcome	Relevant Specifications
SFR1	FQDN registration	Registration of an SF endpoint with the service routing	Persistent set of internal state information for suitably forwarding any request to the registered FQDN-based SF endpoint	Internal FLIPS specifications with the intention of IETF standardization
SFR2	Topology target	Specification of a suitable network topology for the media service path outlined in the orchestration template	Persistent set of forwarding rules being pushed to the infrastructure	ETSI MANO
SFR3	HTTP/IP	Transmission of IP-based protocols at the HTTP or IP level	Persistent set of internal path information suitable for forwarding the request in the network	IETF specifications for HTTP and IP-based protocols

Table 19: Service Function Routing Interfaces

The data plane interface SRF3 is based on existing protocol specifications standardized by the IETF for protocols such as HTTP, TCP, SCTP or other IP-based protocols. This interface differentiates HTTP as a specific application protocol due to its proliferation in the Internet with about 60% or more traffic being based on HTTP. HTTP-level SF endpoints, i.e. represented by a fully qualified domain name (FQDN), are registered through the SFR1 interface and will become 'visible' in the media service path as a result, i.e. any future service request to said FQDN will possibly be routed to the registered SF endpoint. On the management side, the SRF2 interface will be utilized in the orchestration to establish media service path specific topologies, resulting in suitable forwarding rules to be pushed to the infrastructure (via the OpenFlow or similar interface), this information being provided in an ETSI MANO compliant description of the service path. For the realization of this component, FLAME has selected the FLIPS solution provided by IDE, which comes with FLIPS-specific protocol specifications (see Section 6.6.4.5 for an overview of those protocols). However, the component could be realized via standard IP technologies for which existing DNS (for SFR1) specifications would be utilized.

6.5.5 CLMC

The CLMC component offers a series interfaces supporting the monitoring, measurement and assessment of performance, in addition to configuration of processes supporting the integration and

Co-funded by the Horizon 2020
Framework Programme of the European Union

organisation of data for analytics. The general capabilities of the interfaces are shown in Figure 35 and further elaborated in Table 20.

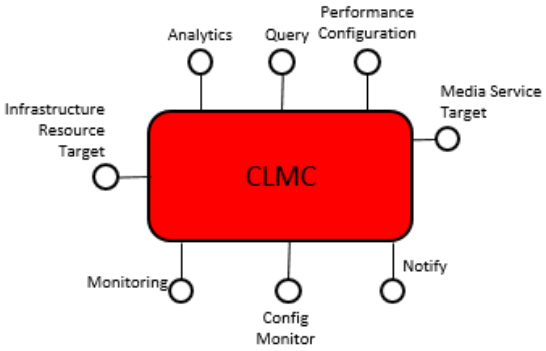

Figure 35: CLMC Component

No.	Interface	Purpose	Outcome	Relevant Specifications
CLMC1	Performance Configuration	Used to configure monitoring and measurement processes for performance assessment and analysis of a Media Service. Configuration will include setting up processes for multidimensional analysis of KPIs from monitoring data including aggregation, classification and stakeholder viewpoints.	Performance monitoring and measurement model for a media service including runtime notification rules.	TM Forum TMF-628, ETSI Mano
CLMC2	Media Service Target	Used to set performance targets for a Media Service through specification of KPIs within a service function chain.	Set of performance requirements for a Media Service	
CLMC3	Infrastructure Resource Target	Used to set performance targets for the Platform through specification of KPIs for Infrastructure Resources. This comes through the SLA knowledge of the orchestrator	Set of performance requirements for the Platform	
CLMC4	Query	Used to retrieve KPI measurements for a given set of criteria	KPI measurement	TM Forum TMF-628, ETSI Mano. Mainly proprietary options for the information model and query language
CLMC5	Notify	Used to notify subscribers of performance measurements through event-trigger-action rules such as current KPI performance	Notifications of configured performance events	Many pub/sub specifications
CLMC6	Analytics	Used by developers of interactive media systems or sub-system to understand performance through exploration or requests, responses and workload	Knowledge on how to best achieve KPIs through correlated data, etc.	Mainly proprietary options for the information model and query language
	Monitoring			
CLMC8a	Monitoring (infrastructure)	Used to monitor infrastructure resources (compute, storage and network) including resource allocation, resource usage and resource performance metrics. Resource monitoring data is high frequency (uSec-mSec) data. These data may be aggregated and resampled at points of publication to reduce the frequency and quantity of monitoring data through the network.	Persistent set of transactional data describing the state of infrastructure resources over time related to a known context within a Resource Configuration	Pub/sub protocol supporting a range of collection and sampling models

Co-funded by the Horizon 2020 Framework Programme of the European Union

CLMC8b	Monitoring (Service Function Configuration)	Used to monitor changes in Service Configuration occurring in response to demand	Persistent set of Service Function configuration change events	TMF640 Activation and Configuration API, ETSI Mano VFN Configuration
CLMC8c	Monitoring (Service Function)	Used to monitor SF including service request, service response and service performance. Service monitoring data is high frequency (mSec) data. These data may be aggregated and resampled at points of publication to reduce the frequency and quantity of monitoring data through the network.	Persistent set of transaction data describing the state of a Service Function.	Pub/sub protocol supporting a range of collection and sampling models
CLMC8d	Media Service Resource Configuration	Used to set a Resource Configuration for a Media Service including target allocations and constraints.	Persistent set of infrastructure resource configuration change events	TMF640 Activation and Configuration API, ETSI Mano VFN Configuration

Table 20: Performance Management Interfaces

There are different standards and specifications relevant to Performance Management. For the ***Performance Management Configuration***, the TM Forum Performance Management API (TMF-628) [TM17] provides a restful API for managing the lifecycle of production and collection of performance measurements. ETSI Mano includes Network Service, VNF and Resources performance management operations to measure performance synchronously "Get Performance Measurement Results" and asynchronously "Notify". Although it should be noted that the ETSI Mano specification is abstract and architectural and not at the level of an API with message formats. ETSI and TM Forum separate processes for measuring performance KPIs (Performance Management) from contractual processes provided by SLA Management with the former underpinning latter. For ***Configuration Monitoring***, TMF640 Activation and Configuration API provides a more general version of the VFN Configuration operations within ETSI Mano. It is likely that the same API can be used to monitor configuration objects and changes associated with both infrastructure resources and service functions. For ***Resource and Service Monitoring*** there are no widely accepted standards but we expect to adopt a pub/sub protocol supporting a range of collection options allowing for resampling, aggregation and buffering of monitoring data.

For the knowledge model, ***KPI Classifications*** have been formally defined by ITU-T, ITIL and many others, covering all aspects of business operations. We expect to focus on KPIs relevant to Service Quality (Response Time, Throughput, Setup, Availability, etc.) and Operational Efficiency, as these are the KPIs most influenced by the Platform's capabilities. A taxonomy of key KPIs will be selected and made available to Platform users when assessing the performance of different interactive media systems and their configurations. For ***Multi-Dimensional Modelling and Queries,*** there are no recent standardisation attempts around OLAP. The MDX query language was established in 1998 to provide a more intuitive model to SQL whilst Microsoft established the XML for Analysis (XMLA) SOAP based specification in 2001, which was widely adopted by vendors as it supported MDX. However, recent developments have focused on products rather than specifications especially integration of multi-dimensional analysis within big data and stream-based processing platforms (e.g., Apache Kylin [KYLIN] and Apache Lens [LENS]).

6.5.6 Infrastructure

Our view on the infrastructure is outlined in more detail in Sections 2.3, 2.4, and 6.1. From an interface point of view, we focus on the ability to configure forwarding capabilities and utilize the data plane of the infrastructure. These capabilities are outlined in Figure 36 and further elaborated in Table 21.

Co-funded by the Horizon 2020 Framework Programme of the European Union

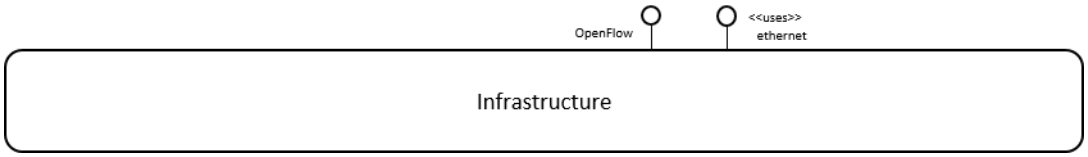

Figure 36: Infrastructure Component

No.	Interface	Purpose	Outcome	Relevant Specifications
I1	Openflow	Provide capability to define switch-specific forwarding rules compliant with OpenFlow specifications and associated with the FLAME-specific infrastructure slice	Persistent set of forwarding rules per SDN switch	OpenFlow
I2	Ethernet	Provide data plane access at the Ethernet framing level		Relevant IEEE Ethernet standards

Table 21: Infrastructure Interfaces

As outlined in previous sections, we assume the infrastructure resources to be provided through the concept of a slice. Within such slice, forwarding rules can be associated to the specific SDN-abstracted forwarding switches associated to the slice. The I1 interface provides such rules to the infrastructure which in turn will suitably configure the switches. Any translation onto legacy components, provided through the SDN abstraction to the FLAME platform, is seen as being part of the infrastructure. At the data plane level, a simple Ethernet abstraction of the data plane is assumed at this stage. Other Layer 2 technologies can be supported by extending the supported framing at this interface. However, this is completely subject to the availability of the source code of the firmware and knowledge on the supported framing protocols. Commonly this is not provided by network device vendors.

6.6 SERVICE FUNCTION CHAINS

The following subsections present various SFCs for the main internal interactions between core components of the platform architecture, as shown in Figure 14. We divide the interactions into various areas. We will start with the *experimentation SFC*, which connects with the *management & control SFC* for driving the management policies of the experiment into the FLAME platform. We then outline the *main (FMI) data plane SFC* for the exchange of media service interactions over the data plane provided by the FLAME platform. Due to the core contributions of the service function routing solution to the benefits of the FLAME platform, as outlined in D3.1, we further decompose the service routing component of our platform into the specific *Service function routing SFC*. As a crucial enabler for the controllability and observability of our platform, we further decompose the CLMC part of our platform as a dedicated SFC.

Throughout the presentation of the various SFCs, we will reference the specific component interfaces being utilized. There will, however, be a number of SFC-internal interfaces and protocols, whose exact detail is not within the scope of D3.3 but will be covered throughout the various realizations of the SFCs in WP4. We will also utilize the colour coding of Figure 14 to indicate which component realizes which service function.

6.6.1 Experimentation SFC

Our architecture overview in Figure 14 outlines two management interfaces that constitute the experiment API for experimentation stakeholders. This will include one or more media service

Co-funded by the Horizon 2020
Framework Programme of the European Union

providers and the platform provider. In the SFC shown in Figure 37, we further elaborate on this interaction between a media component and the FLAME platform service functions involved in this dialogue. Note that the media component realization is not within the scope of the FLAME platform specification, denoted by the different colour.

For the ETSI MANO compliant interface interaction, the media component will directly interact with the **orchestrator** SF of the FLAME platform via the ***O1*** interface. This service function is responsible for receiving an ETSI MANO compliant template of the service to be deployed and executing the necessary steps towards the FLAME management and control elements. This execution is defined in [NFVMANO] and will be extended in the management and control SFC presented in Section 6.6.2. The orchestrator furthermore monitors and observes the fulfilment of the SLA towards the media component, e.g., providing adjustment capabilities by initiating new resource allocations within the constraints laid out in the provided orchestration template.

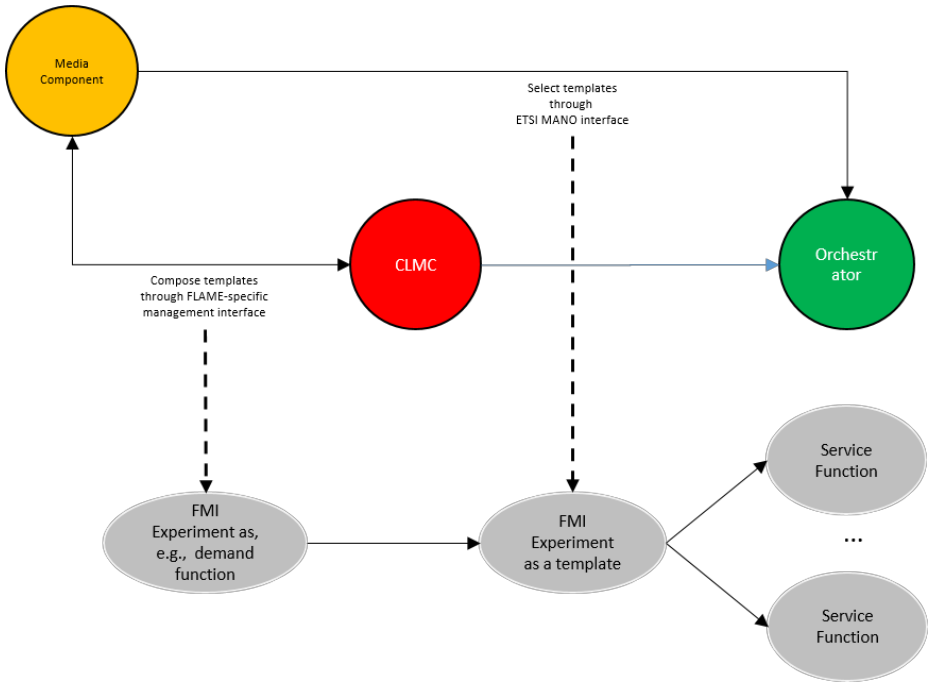

Figure 37: Experimentation SFC

Apart from this template-oriented dialogue, we see an evolution of the interaction between the media service provider and the FLAME platform, as already outlined in Section 6.1, towards a more SLA-oriented dialogue, realized via the second interface. For this, the media component will interact with the **CLMC SF** of the FLAME platform, decomposed later in Section 6.6.5, via the interfaces ***CLMC1***, ***CLMC2***, and ***CLMC3***. The CLMC SF is responsible for gathering information throughout the platform as well as from media components directly (the latter provided through a separate monitoring interaction between media component and CLMC) and reasoning about demand patterns that will allow for making improved judgement about fulfilling constraints of given SLAs. In the envisioned interaction at management level between CLMC and media component, information on possible SLAs is returned to the media component via the ***CLMC4*** and ***CLMC5*** interfaces, either proactively as a result of changes in the system or reactively based on explicit querying by the media component, including analytics information via the CLMC6 interface. Section 6.6.5 will elaborate on the necessary decomposition of the CLMC SF to ensure such evolving dialogue between media component and the CLMC SF as a whole. The result of any new SLA-level agreement between CLMC and media component is then delivered to the orchestrator SF in the form of an ETSI MANO compliant orchestration template.

Co-funded by the Horizon 2020 Framework Programme of the European Union

6.6.2 Management & Control SFC

Management and control in FLAME is initiated through the provisioning of an ETSI MANO compliant orchestration template through the experimentation SFC, as described in the previous subsection, and shown in Figure 38. For the functioning of this SFC, we divide the interactions into management and control related ones, driven by our considerations presented in Section 6.3.1.

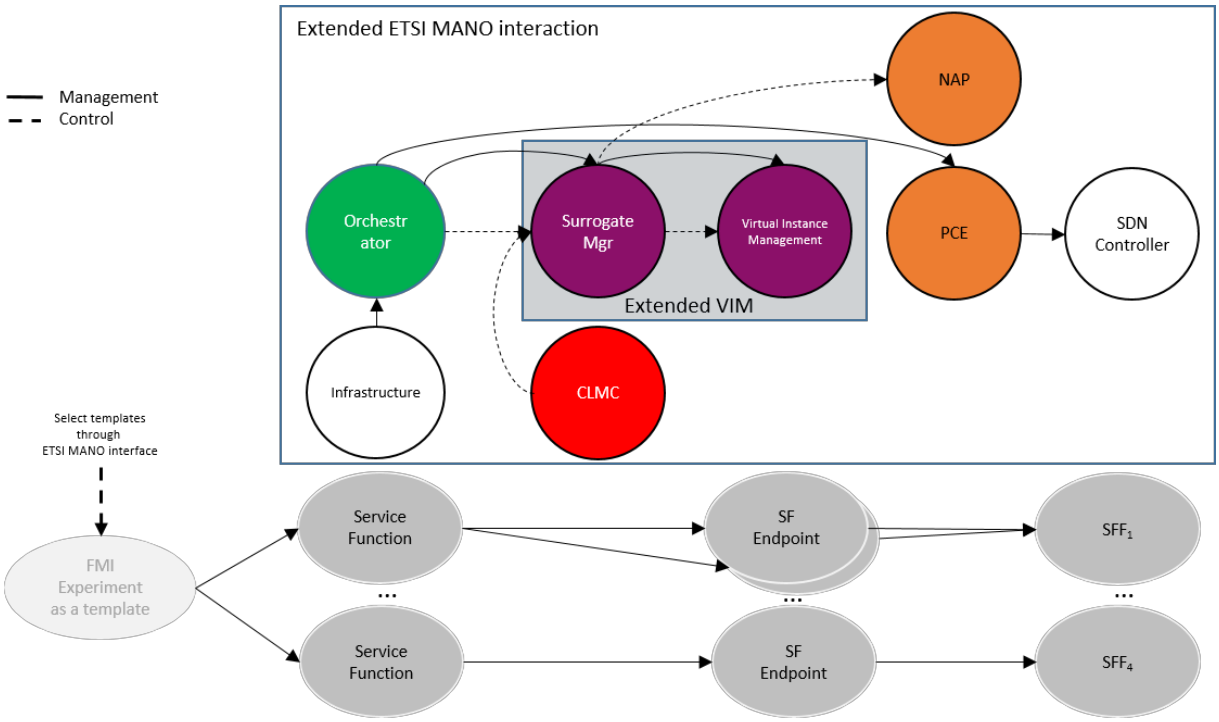

Figure 38: Management & Control SFC

From the **infrastructure**, here shown as a separate SF, the orchestrator will receive an ETSI MANO compliant resource description of the wholesale resources being allocated to the FLAME platform. After receiving the request for orchestration from the media component (or the CLMC, as shown in Figure 37), the orchestrator initiates a number of orchestration steps in interaction with the various SFs, namely:

- At the *management* level, the orchestrator instructs the **surrogate manager SF** via the ***SFEMC1*** interface to initiate the placement (and possible bootup) of computational and storage resources, according to the orchestration template, representing a media component service instance. For each such service instance, the surrogate manager SF will maintain a *SF endpoint control state* (see Figure 39). The initial state for each service instance is determined through the orchestration template and might change during the overall service lifetime, based on the control policies executed by the surrogate manager. For the latter, the orchestrator provides the surrogate manager SF with the control policies included in the orchestration template.

- For the *management* of the virtual instances of (media) service endpoints at the service instantiation time, the surrogate manager interacts with the **virtual instance management (VIM) SF**. The functioning of the VIM SF is aligned with existing solutions for ETSI MANO compliant resource management (WP4 will investigate the suitability of existing VIM solutions for the realization of this SF). The combination of the surrogate manager SF and the VIM SF

Co-funded by the Horizon 2020 Framework Programme of the European Union

could also be positioned as an **extended VIM SF**, particularly when considering standardization of the control policies and therefore embedding a surrogate management functionality into the current VIM realizations available.

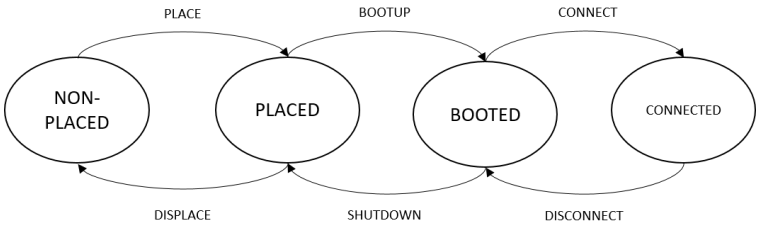

Figure 39: SF Endpoint Control State

- In addition, for the *control* of the virtual instances of (media) service endpoints during the service lifetime, the surrogate manager SF will execute the suitable control policies, provided at orchestration time by the orchestrator via the interface ***SFEMC2***, and instruct the VIM SF to adjust states of the compute/storage resources, i.e. place a previously non-placed virtual instance or boot up a previously virtual instance. For this, it will utilize knowledge (in the form of monitoring data) being obtained via the CLMC SF.

- For the *management* of the communication resources, i.e. the switching fabric of the underlying infrastructure, the orchestrator instructs the **Path Computation Element (PCE) SF** via the ***SFR2*** interface, which in turn will configure the switching fabric through the **SDN controller SF** of the underlying infrastructure, utilizing the OpenFlow compatible southbound interface ***I1***. The configuration of the communication infrastructure is driven by the forwarding solution being utilized at the data plane level – for more information on this, refer to the details of the Service function routing SFC in Section 6.6.4.

- For the *control* of communication resources, the surrogate manager SF communicates with the **NAP SF** of the Service function routing SFC (see Section 6.6.4.1) for any state changes of a virtual instance from (only) BOOTED to CONNECTED and vice versa, see Figure 39, via the ***SFR1*** interface. With this, the routability of media service requests to a particular media service endpoint can be controlled at runtime, depending on the control policies provided as part of the orchestration template.

- The interaction between CLMC SF (see Section 6.6.5) and surrogate manager SF serves two *control* purposes, realized via the ***CLMC8*** interfaces. Firstly, it can lead to virtual instance control state changes based on input data from the CLMC pertaining to specific control policies provided in the orchestration template. For instance, exceeding a specific compute load might lead to activating another virtual instance, i.e. placing the state of said instance into CONNECTED state. Secondly, input data from the CLMC might relate to routability of service requests to virtual service instances. For instance, a virtual service instance might have become disconnected from the network due to a fault, reported to the CLMC SF and provided as monitoring data to the surrogate manager SF, which in turn changes the state accordingly to BOOTED for this instance and possibly instruct a state change for another instance to CONNECTED. *It is important to note here that the control decisions lie with the receiving entity, e.g., the surrogate manager, not the CLMC.*

Co-funded by the Horizon 2020
Framework Programme of the European Union

6.6.3 Main (FMI) Data Plane SFC

The main interactions for a media service at the data plane is shown in Figure 40 as being realized through the service function routing component of the FLAME platform via the ***SFR3*** data plane interface that connects said media components at the data plane level. As indicated in Figure 14, the media components itself are not part of the FLAME itself albeit their chaining at the data plane is being realized by the FLAME platform. By preserving the data plane abstractions of well-known protocols, such as HTTP, RTP or generally any IP-based protocol, the SFC-based decomposition of the FLAME platform is not exposed to the media components and therefore the media service developers, allowing for utilizing any available composition technique for this part of the overall system.

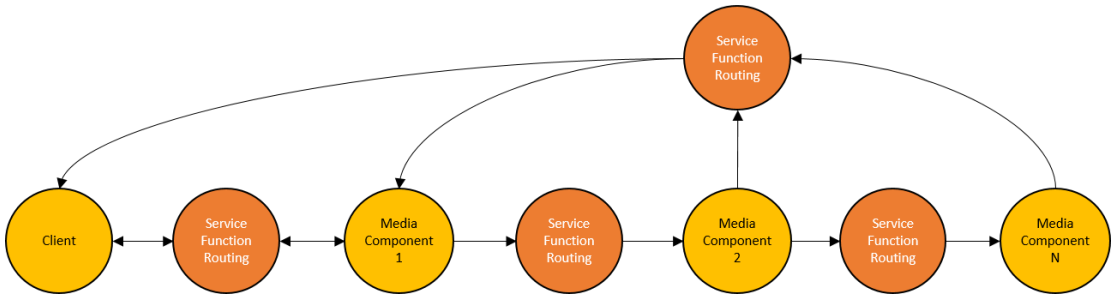

Figure 40: Main (FMI) Data Plane SFC

The routing function is realized by the **service function routing** SF. From an SFC architecture [SFC] perspective, this SF enables flexible and dynamic chaining of the connection service functions, here the media components, including the flexible redirection to virtualized instances. This capability is outlined in [Pur17] and its realization is presented in the following subsection.

6.6.4 Service Function Routing SFC

Figure 41 further decomposes the Service Function routing SF, introduced in Figure 40, realizing the data plane transfer of media service interactions over the underlying infrastructure. The composition is shown as a sub-component architecture in Figure 42. In the following sub-sections, we further describe the operations of these sub-SFs.

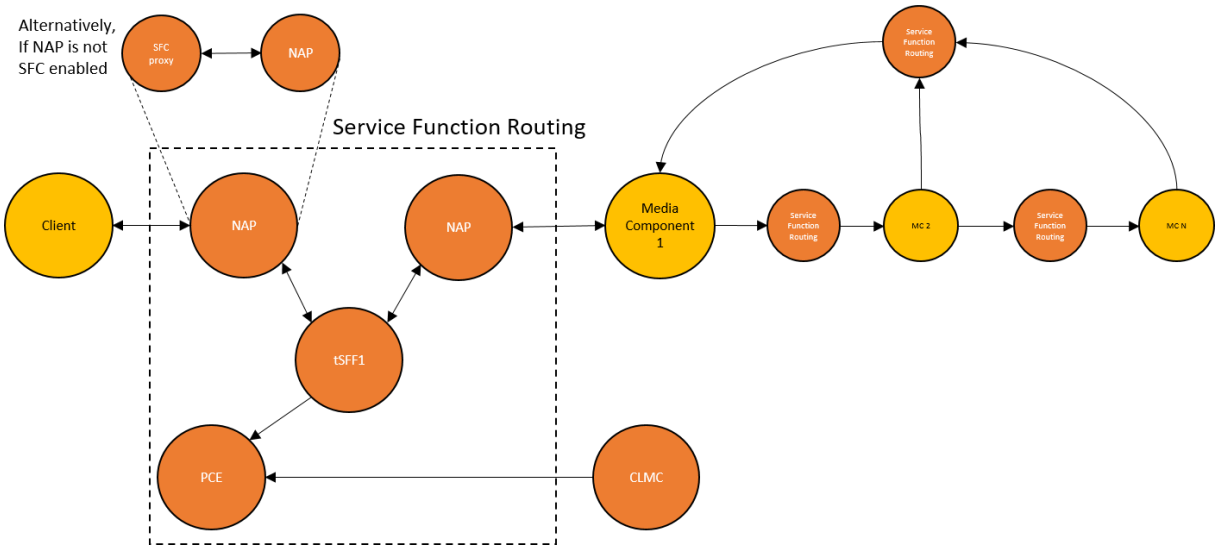

Figure 41: Service Function Routing SF Decomposition

Co-funded by the Horizon 2020 Framework Programme of the European Union

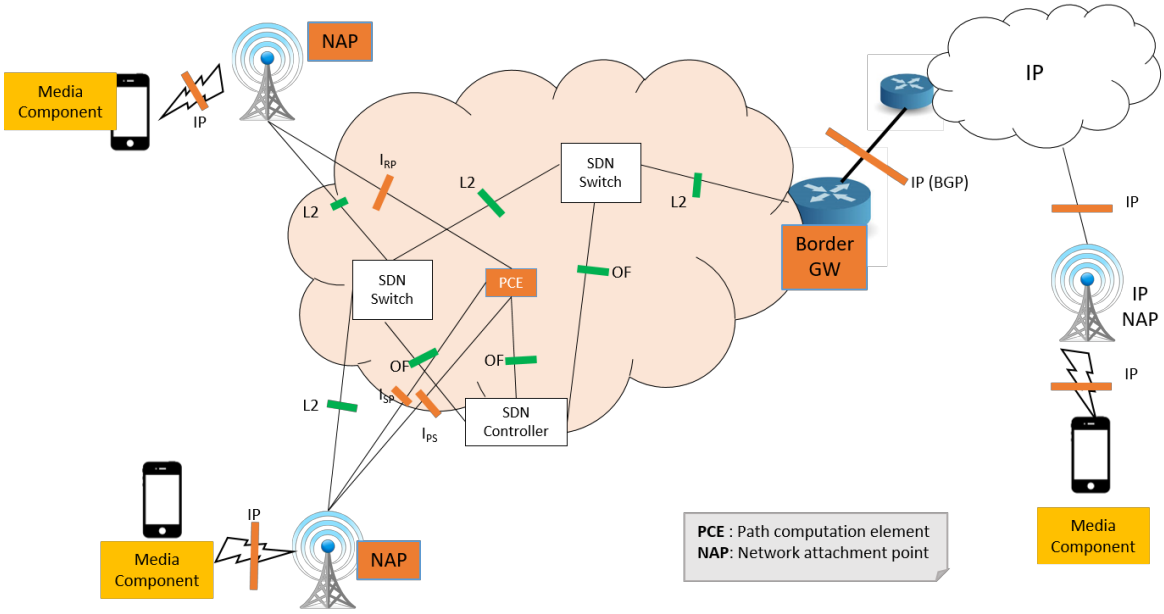

Figure 42: Service Function Routing SF Decomposition as Component Architecture

6.6.4.1 Network Attachment Point (NAP) SF

Each media component (or any other IP-based device) is connected to at least one NAP. This NAP serves as an edge protocol termination point, i.e. terminating application sessions, transport protocol sessions or below. In the current realization of the NAP, HTTP sessions are terminated in a proxy operation, i.e. the TCP sessions is being terminated and the HTTP request is being forwarded towards a suitable other NAP in the network. Any other protocol but HTTP is terminated at the IP level, forwarding the resulting IP packets to a suitable other NAP in the network. At said suitable other NAP in the network, shown on the right side of the decomposed Service function routing SF in Figure 41, the HTTP or IP level packets are then forwarded to a suitable media component (or generally IP-based service endpoint). The Border GW shown in Figure 42 acts as a NAP but connects to a peering autonomous system rather than a media component, for cases that services are being requested from outside the infrastructure domain provided by the FLAME platform. The NAP performs a number of functions relating to operations and management, e.g., for registration of service endpoints at the fully qualified domain name (FQDN) level, handling of path updates, mobility support and more.

6.6.4.2 Path Computation Element (PCE) SF

In order to forward HTTP or IP-level packets from one NAP to another (see above), the PCE SF is being utilized. This PCE works closely with the underlying infrastructure, through suitable OpenFlow-based management interfaces, to configure the forwarding actions of the SDN-enabled switching fabric. For this, the PCE assigns every link of the underlying network topology to a unique bit in a pre-defined bitarray (with the size of the bitarray large enough to store all such unique bit positions, i.e. the size of the array equals the number of network links). A forwarding action from point A in the network to another point B can now be represented a unique bitarray in which all bitpositions of those links belonging to the path from A to B. Figure 43 shows this simple forwarding operation. In this example, the path from S (originating in a peering network) to A is represented as a simple binary OR of the three links from the border GW to the NAP to which media component A connects, i.e. 10010010 is the binary representation of this path. On the other hand, the path from S to B is represented as 10100010 as one link is different here compared to the previous. With that in mind, the transfer of a packet from S to A and B is now easily achieved by a binary OR of the individual path. We refer to this capability as **coincidental multicast**, which is utilized to send a single HTTP-level response to more than

Co-funded by the Horizon 2020 Framework Programme of the European Union

one client at the same time (for cases in which the arrival of the client requests is aligned in the right arrival window). It is the role of the PCE to receive path computation requests in the form of either an IP or FQDN destination as well as a source node identifier (which is assigned to each NAP by the PCE during the bootstrapping process). The result of this operation is the bitarray, as shown in Figure 43, and is returned to the requesting, i.e. originating NAP. In order to improve HTTP (request/response) operations, the PCE will also return the return path from the destination NAP to the requesting NAP in the same operation. This supports requests and responses being sent over different paths in the network.

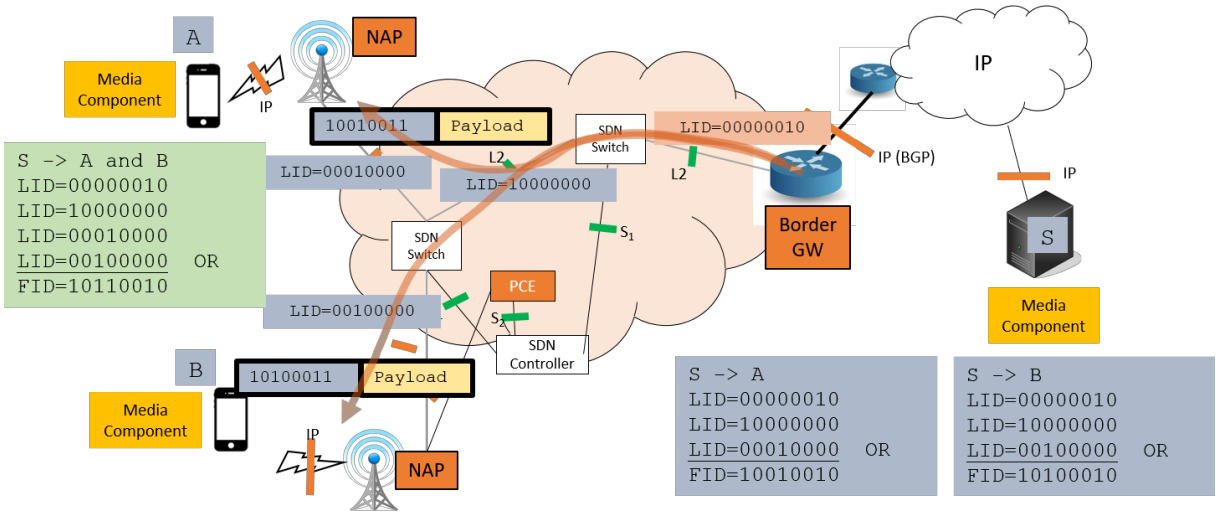

Figure 43: Service Function Routing SF tSFF$_1$ Realization

6.6.4.3 tSFF$_1$ SF

The last sub-SF of the Service function routing SF is that of the **tSFF$_1$** (see Figure 41), which is according to the SFC definitions [SFC] a transport-derived service function forwarder. This function encapsulates the packet received from the NAP into a transport network specific packet format for delivery of this specific transport network. There are two known realizations of this tSFF$_1$, namely:

- **SDN-based**: This solution encapsulates the packets into a standard Ethernet frame format. It utilizes the IPv6 source and destination fields of the Ethernet header for storing the aforementioned bitarray. During the bootstrapping of the network during the orchestration of the infrastructure-facing resources (not the media component facing ones), the PCE will interact with the SDN-based infrastructure to configure suitable forwarding rules in each intermediary SDN switch of the infrastructure. For each SDN switch, a number of rules needs to be defined that is propositional to the number of supported hardware output ports (often 48) at said local switch. The forwarding is realized by checking the unique bitposition for each individual output port and, if set to 1, forwarding the incoming packet over said output port. With that, multicast transmission is easily supported, as shown in Figure 43. The required SDN match operation is that of a wildcard (with the wildcard being the bitarray with only the specific output port bitposition set) and is supported with OpenFlow V1.2 [OpenFlow] upwards, i.e., in most commercially available SDN switches. For more information on this forwarding approach, see [Reed16], including memory costs.

- **BIER-based**: The Bit Indexed Explicit Replication (BIER) working group of the IETF has been defining a bit-based forwarding solution similar to that outlined above for SDN. Instead of labelling links, the specific egress, i.e. outgoing, router is specified with a unique bitposition in

Co-funded by the Horizon 2020 Framework Programme of the European Union

a given bitarray. The principles of path construction as well as ad-hoc multicast delivery (by combining two or more paths into a single multicast one by a simply binary OR) remain the same. Mappings of the BIER forwarding architecture are provided over SDN, MPLS and other transport networks. [BIER17] outlines the use case for an HTTP-specific NAP utilizing such forwarding solution.

6.6.4.4 Realization over link-local Access Networks

The SFC decomposition shown in Figure 41 shows a general realization over an SFC-compliant service chaining architecture. With this, media components could be connected via several Layer 2 links to the Service function routing SF, e.g., within a cable network. In most of our use cases, however, we can assume that media components will be connected to the NAPs of the SF via a link-local, e.g., single Ethernet/WiFi, link. Even in the case of the media component being hosted in a data centre, the latter would appear as the IP-level endpoint to the FLAME platform (with the data centre networking to the actual media component instances being opaque to the FLAME platform). For this reason, we can simplify the SFC to the one shown in Figure 44, where the connection between media components and the NAPs are realized by another **$tSFF_2$**, representing Ethernet-based link-local IP-level communication.

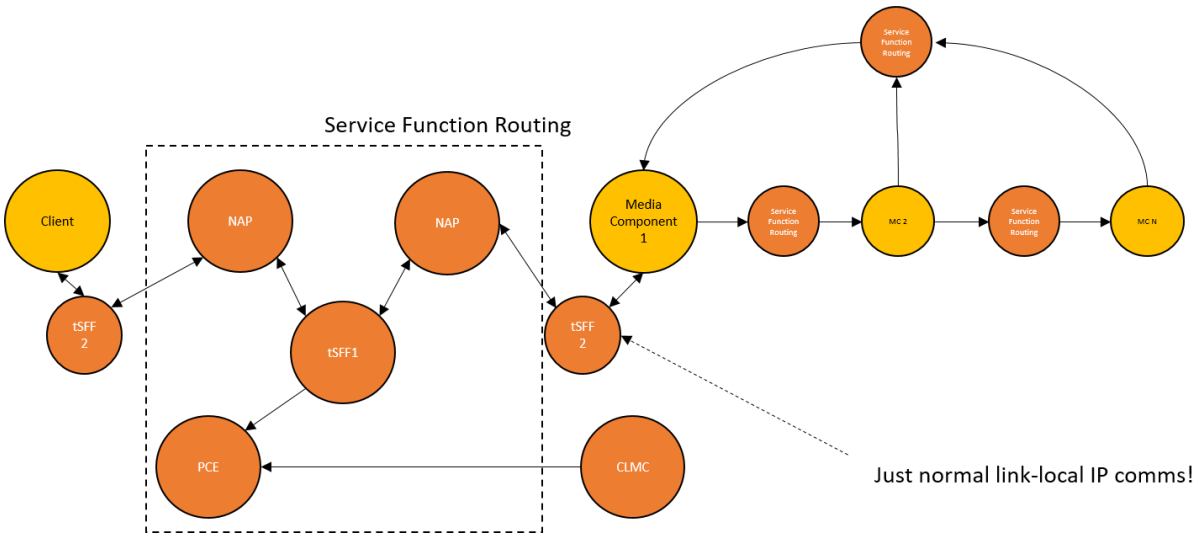

Figure 44: Service Function Routing SFC with link-local clients and media components

6.6.4.5 Protocols for Realizing the Sub-SF Interactions

There are a number of protocols involved in the interactions between the Service function routing sub-SFs, i.e. the NAP, PCE and the $tSFF_1$. Figure 45 shows the various categories of protocols that will form part of the overall Service function routing SFC as part of the platform realization efforts in WP4.

In the **basic HTTP/IP level message exchange** category, protocols for the exchange of HTTP as well as general IP-based protocols are foreseen as well as for IP multicast (used for IPTV). A protocol for conveying path computation requests as well as receiving forwarding path updates is required as a NAP-PCE protocol, while we also provide a protocol for reliable data plane transmission (as a replacement for the traditional TCP and other transport protocols between media components and NAPs). The path-based forwarding itself is captured as a protocol, outlining the southbound interactions to SDN as well as packet format for Layer 2 packets. There are a number of **operations and management** (OAM) protocols, e.g., for distributing routing prefixes, registering FQDN-based (i.e. HTTP-level) service endpoints, or the general monitoring support for connection to the CLMC SF in FLAME. In the **extended protocols** category, we foresee protocols for content certificate distribution,

Co-funded by the Horizon 2020 Framework Programme of the European Union

link-local multicast improvements (for dense user scenarios) and secure content delegation, among others.

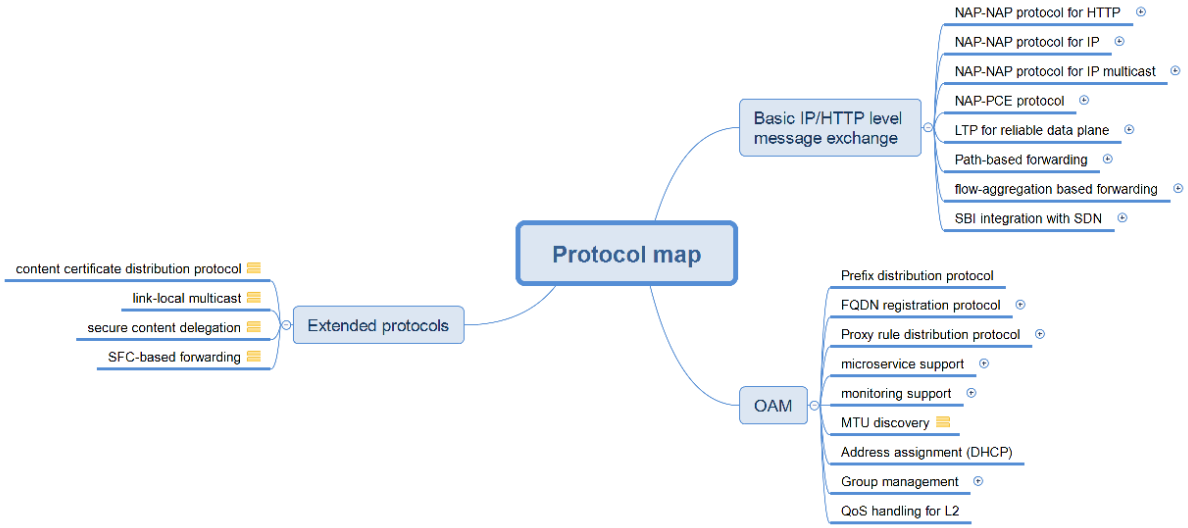

Figure 45: Service Function Routing Protocols

Within the H2020 POINT [POINT] project, the fundamental protocols shown above have been developed in a proof-of-concept prototype, with FLAME driving this work forward in terms of operational and management integration. Protocol specifications exist for those developments but are currently revised for the advanced FLIPS platform, developed internally by IDE, for provisioning to the FLAME project. The protocol map shown in Figure 45 also forms the basis for standardization efforts within the Internet Engineering Task Force (IETF) with the ambition to derive standardized protocols on the basis of these protocols throughout the duration of the project.

6.6.5 CLMC SFC

The SFC decomposition of the CLMC is shown in Figure 46. The SFC describes a stream-processing pipeline that delivers stream-based OLAP cubes for cross-layer knowledge that can be queried and analysed at runtime by media and platform components, or explored by aiming to design templates.

The SFC is designed to support multiple sources of monitoring information from information providers (***CLMC8***) at the platform and media service levels distributed to distributed transactional data storage. Examples of this data include service request events, service response events, resource usage from service functions and hosts. This data is high frequency and high volume data that is delivered using a Pub/Sub SF. The transactional data sources are stored in a various Transactional Data Storage SF sources relevant to the components being monitored. For example, the data stores used to store network monitoring may be different from those used to store user interaction and media service usage due to factors such as technology, ownership and control.

The transaction data storage provides an evidence base for experiments and trials conducted using the platform. The evidence base is persistent with appropriate long-term retention and access policies. The evidence provides quantifiable data supporting hypothesis testing, allows evidence for effectives of the platform to be established, and subject to permission, allows multiple stakeholders to explore openly the consequence of collaborative decision making in interactive media systems. Where appropriate, parts of the evidence base will be made available as open research data.

Co-funded by the Horizon 2020 Framework Programme of the European Union

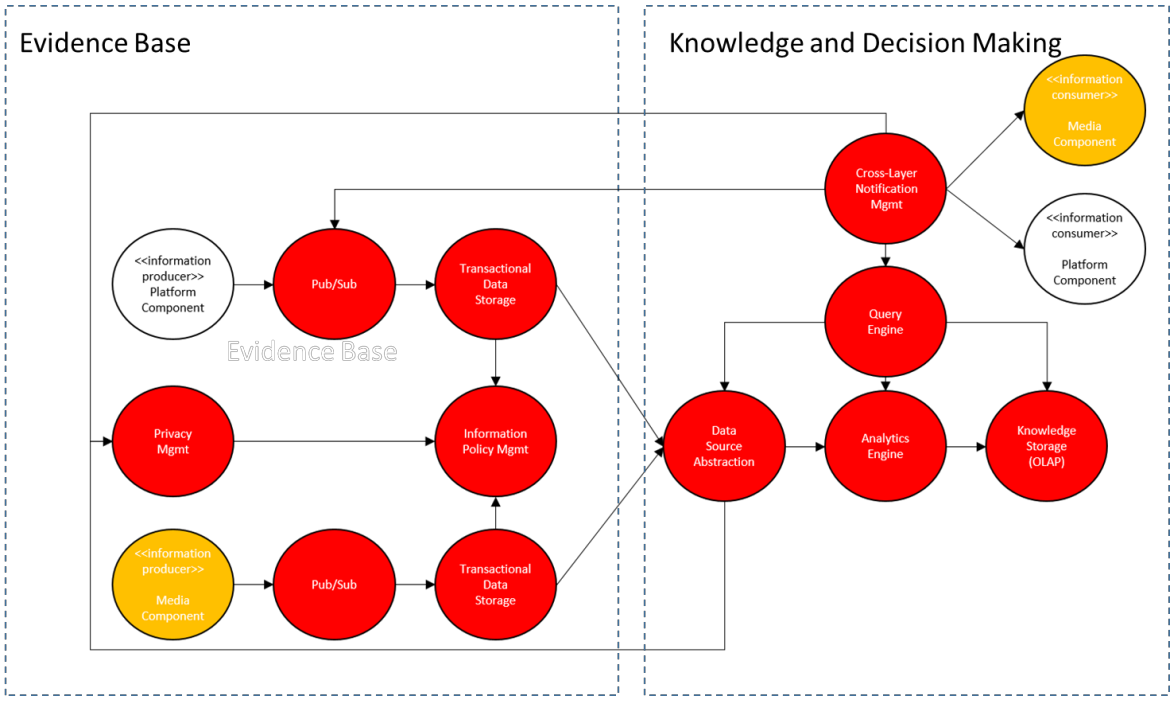

Figure 46: CLMC SFC

The data sources are then processed by a Data Source Abstraction SF to create data abstractions representing the dimensions relevant to specific cross-layer management queries and analytics. The creation of abstractions from different data sources is a computationally intensive activity considering the scale and velocity of monitoring data. Various abstraction algorithms can be considered from simple numerical aggregation to more complex machine learning and graph analytics. The algorithms would be executed in a cluster-computing environment such as Hadoop [HADOOP], Hive [HIVE] or Spark [SPARK]. The interface towards the data abstraction layer is expressed through SQL query language.

The selection of the dimensions is driven by queries posed by Cross-layer Notification Management SF (***CLMC2, CLMC3, and CLMC5***). The queries are executed by a Query Engine SF (***CLMC4, and CLMC6***) that selects the dimensions (data abstractions) needed and accesses OLAP cubes stored within the Knowledge Storage SF. The query language is based on SQL using ODBC, JDBC or RESTful API. The generation of the OLAP cubes is achieved using the Analytics Engine SF, which requests data abstractions and maintains currency of cubes according to the data quality requirements. Different OLAP implementations are available that are optimised according to performance characteristics of queries (i.e. MOLAP, ROLAP and HOLAP). The analytics SFs allow the evidence base to be explored in ways that translate the evidence into knowledge and decision making. This knowledge can then be encoded in templates describing the configuration of services and resources necessary to achieve specific performance outcomes.

Access to the data is governed by Information Policy Management SF and Privacy Policy Management SF. Information policy defines the authorisation rules for access to data sources. Privacy policy management implements processes used to gain consent from users for disclosure of personal information. These processes result in changes to information policies used to enforce consent. Privacy processes must inform users of the specific data collected and the purpose of data processing. The purpose of processing and relevant data abstractions are provided by the Cross-layer Notification Manager SF. The way in which personal information is processed to produce data abstractions is

Co-funded by the Horizon 2020 Framework Programme of the European Union

provided by the Data Source Abstraction SF. The User Managed Access[4] (UMA) specification, which builds on OAuth 2.0[5], is relevant to Information Policy and Privacy Management. UMA protocols are designed to increase the control individuals have over personal information stored at distributed services. Various use-cases and scenarios are provided exploring different privacy and consent implications that can provide the basis for control of personal information within CLMC[6].

[4] https://docs.kantarainitiative.org/uma/rec-uma-core.html

[5] https://oauth.net/2/

[6] https://kantarainitiative.org/confluence/display/uma/UMA+Scenarios+and+Use+Cases

Co-funded by the Horizon 2020
Framework Programme of the European Union

7 CONCLUSIONS

This document provides the specifications for the FLAME platform and infrastructure. Following a traditional architecture development methodology, we also provide an overview of the founding cornerstones in which such specifications will be realized, i.e., the ecosystems of FMI and emerging infrastructure platforms, as well as the use cases that possibly utilize the platform. The latter combined with the market analysis of D2.1 supported the definition of concise requirements against which we can test the realization of the platform at a later stage of the project lifecycle, leading possibly to a refinement of the specifications in D3.3 v2 in M18.

Co-funded by the Horizon 2020
Framework Programme of the European Union

REFERENCES

[5GPPP] 5G Public Private Partnership, https://5g-ppp.eu/

[Addis12] Addis, M., Boniface, M., Papay, J., Servin, A., Zlatev, Z., & Kousiouris, G. (2012). Modelling and Analysing QoS for Real-Time Interactive Applications on the Cloud. In Achieving Real-Time in Distributed Computing: From Grids to Clouds (pp. 1-15). IGI Global.

[BIER17] IETF https://tools.ietf.org/html/draft-ietf-bier-use-cases-05, “Bier Use Cases”, 2017

[Betz17] http://www.bmc.com/blogs/itil-and-devops-lets-not-paper-over-the-differences/

[Boniface15] Boniface, Michael, Stefano Modafferi, Athanasios Voulodimos, David Osborne, and Sandra Murg. "EXPERIMEDIA: Report on the features and opportunities for networked multimedia systems." (2015).

[Boniface16] Boniface, M., Trossen, D. and Calisti, M., 2016. Tackling user-centric media demands through adaptable software defined infrastructures.

[Capterra17] http://www.capterra.com/itsm-software/

[COURCOUBETIS09] COURCOUBETIS, C., Boniface, M., and Man-Sze, L.I., 2009. Future Internet socio-economics–challenges and perspectives. Towards the Future Internet: A European Research Perspective, p.1.

[EINS] European Network of Internet Science, http://www.internet-science.eu/

[HADOOP] https://hadoop.apache.org/

[HIVE] https://hive.apache.org/

[IBM16] IBM, IBM launches experimentation-as-a-service, available at http://www.eweek.com/cloud/ibm-launches-experimentation-as-a-service-offering

[ICT-13-2016] http://ec.europa.eu/research/participants/portal/desktop/en/opportunities/h2020/topics/ict-13-2016.html

[InternetSociety10] Internet Society, Internet Ecosystem, available at https://www.internetsociety.org/sites/default/files/Internet%20Ecosystem.pdf

[InteractiveMedia] Definition for interactive media, available at http://www.atsf.co.uk/atsf/interactive_media.pdf

[Kalogiros11] Kalogiros, C., Courcoubetis, C., Stamoulis, G.D., Boniface, M., Meyer, E.T., Waldburger, M., Field, D. and Stiller, B., 2011, May. An approach to investigating socio-economic tussles arising from building the future internet. In The Future Internet Assembly (pp. 145-159). Springer Berlin Heidelberg.

[KYLIN] http://kylin.apache.org/

[LENS] https://lens.apache.org/

[Mattern89] Mattern, F. (1989). Virtual time and global states of distributed systems. Parallel and Distributed Algorithms, 1(23), 215-226.

[MPEGDASH] http://mpeg.chiariglione.org/standards/mpeg-dash

[MOSCOW] https://en.wikipedia.org/wiki/MoSCoW_method

[NFV] ETSI, Network Functions Virtualisation, available at http://www.etsi.org/technologies-clusters/technologies/nfv

Co-funded by the Horizon 2020 Framework Programme of the European Union

[NFVMANO] ETSI GS NFV-MAN 001, "Network Functions Virtualization (NFV); Management and Orchestration", V1.1.1, available at http://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_NFV-MAN001v010101p.pdf, 2014

[OASIS13] Topology and Orchestration Specification for Cloud Applications (TOSCA). Version 1.0. 23 November 2013. OASIS Standard. available at http://docs.oasis-open.org/tosca/TOSCA/v1.0/os/TOSCA-v1.0-os.pdf

[ODL] OpenDayLight website, available at https://www.opendaylight.org/

[OpenFlow] OpenFlow, available at https://www.opennetworking.org/sdn-resources/openflow

[ONOS] Open Network Operating System (ONOS) website, available at http://onosproject.org/

[Openstack] Openstack website, available at https://www.openstack.org/

[Phillips15] Phillips, S.C., Bashevoy, M., Boniface, M., Crowle, S., Engen, V., Sinkala, Z. and Wiegand, S., 2015, March. Linking quality of service and experience in distributed multimedia systems using PROV semantics. In Service-Oriented System Engineering (SOSE), 2015 IEEE Symposium on (pp. 117-126). IEEE.

[POINT] H2020 POINT project, available at https://www.point-h2020.eu/, 2017

[Pur17] D. Purkayastha, A. Rahman, D. Trossen, "USE CASE FOR HANDLING DYNAMIC CHAINING AND SERVICE INDIRECTION", Internet Draft, available at https://tools.ietf.org/html/draft-purkayastha-sfc-service-indirection-00, 2017

[QUIC] QUIC, a multiplexed stream transport over UDP, available at https://www.chromium.org/quic

[Reed16] Martin J. Reed, Mays Al-Naday, Nikolaos Thomos, Dirk Trossen, George Petropoulos, Spiros Spirou, Stateless multicast switching in software defined networks, In proceedings of ICC 2016, Kuala Lumpur, Maylaysia, 2016

[SDN] Open Networking Foundation, Software-defined Networking (SDN) Definition, available at https://www.opennetworking.org/sdn-resources/sdn-definition

[Setiawan16] Setiawan, N.Y. and Sarno, R., 2016. Multi-Criteria Decision Making for Selecting Semantic Web Service Considering Variability and Complexity Trade-Off. Journal of Theoretical and Applied Information Technology, 86(2), p.316.Vancouver

[SESERV12] SESERV website, available at http://www.seserv.org/getting-started

[SFC] IETF Service Function Chaining Working Group, available at https://datatracker.ietf.org/wg/sfc/about/

[SFC7665] IETF, "Service Function Chaining (SFC) Architecture", RFC 7665, October 2015

[SPARK] http://spark.apache.org/

[TM17] TM Forum, "Performance Management API REST Specification (TMF628) R14.5.1", 2017, available at https://projects.tmforum.org/wiki/display/API/Performance+Management+API+REST+Specification+%28TMF628%29+R14.5.1

[Wikipedia2] As a service concept, available at https://en.wikipedia.org/wiki/As_a_service

Co-funded by the Horizon 2020 Framework Programme of the European Union