Future Service Oriented Networks
www.fusion-project.eu

Deliverable D2.3

Cross-layer system models, optimisation algorithm design and simulation results

Public report, Version 3.0, October 2016

Authors

UCL	David Griffin, Miguel Rio, Khoa Phan
ALUB	Frederik Vandeputte
ORANGE	Dariusz Bursztynowski, Paweł Piotrowski
SPINOR	Folker Schamel
IMINDS	Pieter Simoens

Reviewers

Abstract This document specifies the components of the FUSION architecture for service-centric networking and the interfaces defined between the components. It models the cross-layer interactions between service and network layers of the FUSION architecture and simulated the effect of exposing cost information between the layers to improve overall QoS and reduce total costs.

Keywords Service-centric networking, architecture, interface, API, cross-layer, optimisation.

Project funded by the European Union under the
Information and Communication Technologies FP7 Cooperation Programme
Grant Agreement number 318205

TABLE OF CONTENTS

LIST OF ACRONYMS

Acronym	Expansion
ALTO	Application Layer Traffic Optimisation
API	Application Programming Interface
CDF	Cumulative Distribution Function
CPE	Customer Premises Equipment
CPLEX	IBM ILOG CPLEX Optimization Studio
CPU	Central Processing Unit
DC	Data Centre
DCA	Data Centre Adaptor
DCTCP	Data Centre TCP
EC2	Amazon Elastic Compute Cloud
EZ	Execution Zone
FUSION	Future Service Oriented Networks
GA	Genetic Algorithm
GDP	Gross Domestic Product
GHz	GigaHertz
GPU	Graphical Processing Unit
HTTP	Hypertext Transfer Protocol
ILP	Integer Linear Programming
IP	Internet Protocol
ISP	Internet Service Provider
LP	Linear Programming
PoP	Point of Presence
POSIX	Portable Operating-System Interface
PPC	Performance-Predictability Curve
PPP	Purchasing Power Parity
QoE	Quality of Experience
QoS	Quality of Service
RAM	Random-Access Memory
RD	Resolution Domain
RTT	Round Trip Time
SLA	Service Level Agreement
TCP	Transmission Control Protocol
TOSCA	OASIS Topology and Orchestration Specification for Cloud Applications
VM	Virtual Machine

1. EXECUTIVE SUMMARY

This document describes the final architecture for service-centric networking together with the interfaces defined between the components. The work in the final year of the project regarding architectural issues focussed on the modelling of cross-layer interactions between service and network layers of the FUSION architecture.

Interactions between the service and network layers of the FUSION architecture are based around two essential concepts. Firstly *Utility functions* are defined per service by the service provider where utility is described as a weighted combination of metrics relevant for the service performance: infrastructure cost, network latency between clients and service instances, etc. Secondly the *session slot* is a lightweight abstract application metric for measuring the resource utilization and availability for a particular service instance. Execution zones announce the current number of available session slots to the service resolvers to help drive the instance selection algorithms. In addition we explored whether additional information exchanged between the service provider/orchestrator and the underlying ISPs undertaking service resolution can be of benefit by increasing QoS or reducing costs in the deployment and operations of FUSION services.

A centralised algorithm with full knowledge of service placement costs and network transit costs can optimise the placement of services to balance costs and service utility. However, in practice this is unrealistic as the placement algorithm, run by the service orchestrator on behalf of the service provider, would need full knowledge of the transit costs of the underlying ISPs. We have shown that if the ISPs inform the service orchestrator of a proportion of transit costs then overall costs can be reduced and QoS can be increased. The results show that the proportion of transit costs that need to be exposed from ISPs to service providers depends on the degree of mismatch between the transit and placement costs for selecting service instances in specific EZs. In the case of complete alignment of costs there is no benefit in exposing any cost information. In artificial cases where costs are completely reversed (the highest cost EZs are available over the lowest cost transit links and vice versa), then up to 95% of transit costs would need to be exposed. In intermediate cases the top 5% of transit costs need to be identified to the service provider to improve QoS and reduce transit costs for the ISP.

Interactions between the service placement and resolution layers can be extended beyond utility functions and session slots to include a proportion of transit costs being exposed to the service orchestrator to increase overall QoS to the users and to reduce ISP costs.

2. FUSION ARCHITECTURE OVERVIEW

The FUSION framework can be seen in Figure 1. Functionality is divided into 3 layers. At the lower layer IP routing forwards packets using traditional end-to-end protocols. At the upper layer the execution plane consists of all the execution zones where service instances will run. In the middle the service resolution layer forwards request from clients to service instances.

The basic operation of the FUSION system is that orchestration domains – consisting of a potentially large number of geographically distributed execution zones – deploy services on behalf of service providers in one or more execution zones according to the expected demand by service users. This is depicted in the upper layer of Figure 1. Service resolution domains are responsible for matching service requests to execution zones containing running instances of the requested service. This is depicted in the middle layer of Figure 1. Service resolution is anycast in nature – the user simply requests a service and it is the responsibility of the service resolution plane to find the "best" available instance for that request. Once a specific service instance in a specific execution zone has been selected for the user request, data plane communications take place in the data forwarding plane depicted by "IP Routing" in the lower layer of Figure 1.

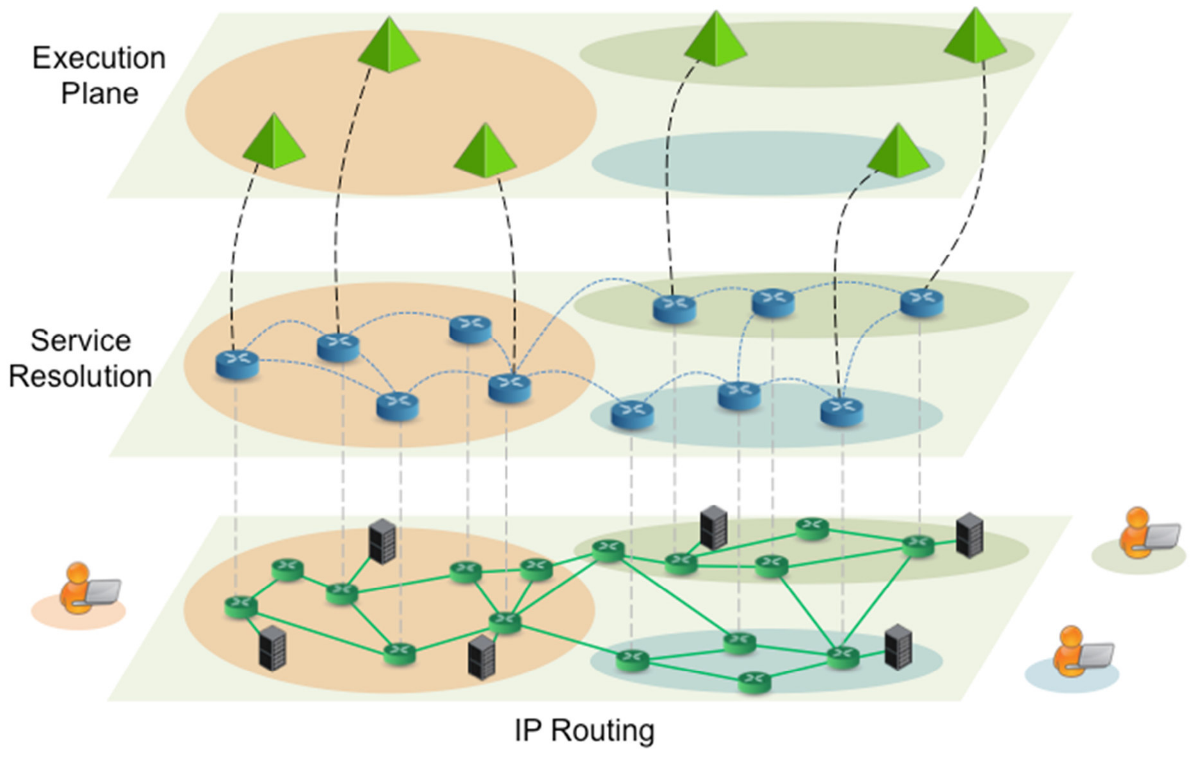

Figure 1: FUSION framework

The functional entities in the FUSION architecture are depicted in Figure 2. The three main entities are the orchestrator, execution zone and service resolver.

The *orchestrator* manages its orchestration domain resources including execution zones and services, which it manages on behalf service providers. The orchestrator is responsible for service management functions including service registration, server placement (selecting appropriate execution zones to execute service instances), service lifecycle management and monitoring.

The *execution zone* is the logical representation of the collection of physical computational resources in a specific location, such as a data centre, which is managed by an orchestrator. The orchestrator has an abstract view of an execution zone and the detailed internals are managed by a zone manager. The zone manager is responsible for managing service instances within its zone but under the instruction of the orchestrator.

The *service resolver* is responsible for maintaining and managing service resolution information to create forwarding paths for queries to be resolved to execution zones containing available running instances of the specified service.

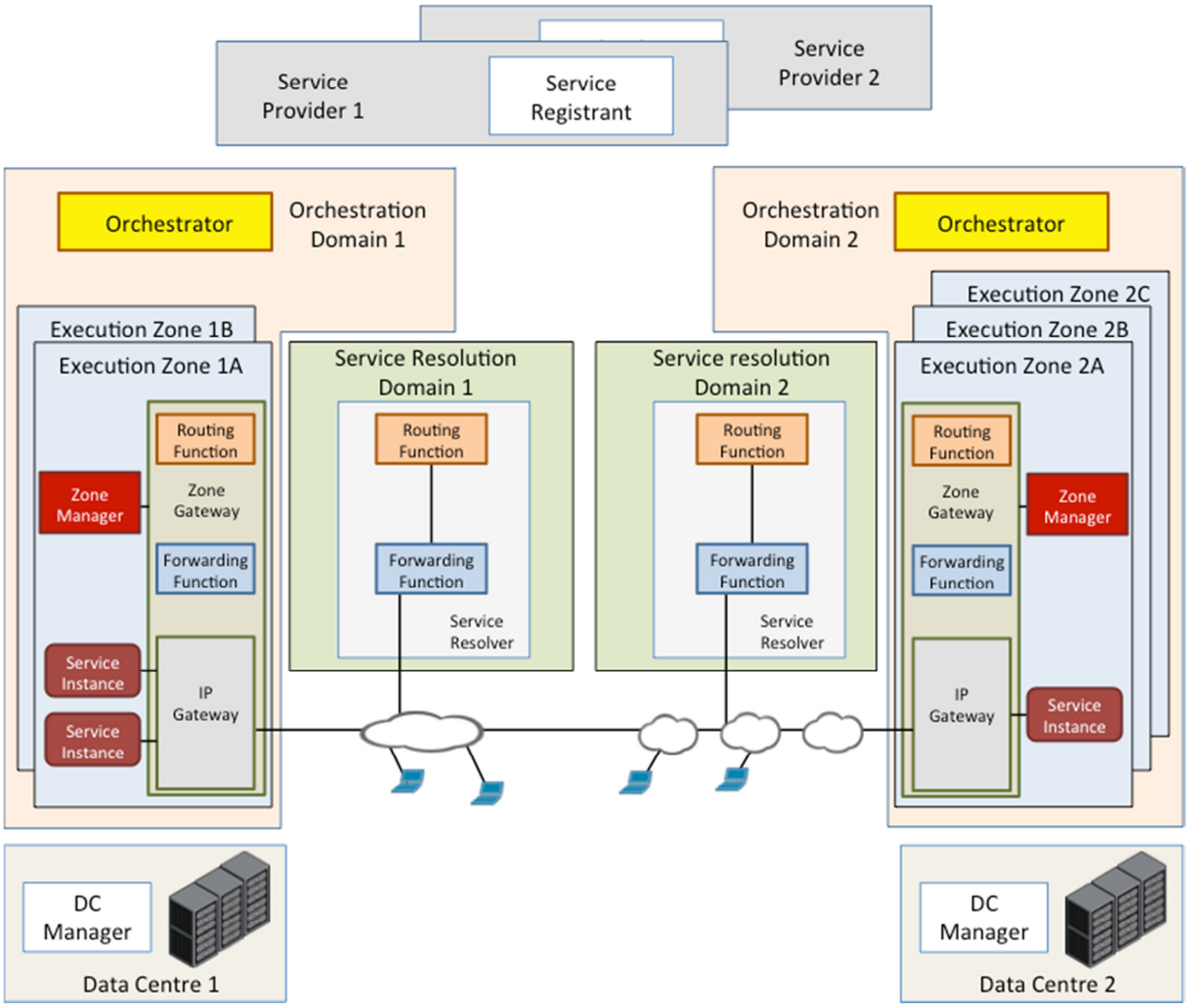

Figure 2: FUSION functional architecture

The specification of all interfaces in the FUSION architecture is contained in Appendix A.

There are essentially two layers to the FUSION system: the service layer comprised of the orchestration domain, including the orchestrator and execution zones; and the network layer comprised of service resolution domains. In this deliverable we consider the interactions required between the two layers to deliver FUSION services to users.

2.1 Interaction scenarios

2.1.1 Interaction overview

The overall service lifecycle across multiple cooperating, but loosely coupled entities in FUSION is depicted in Figure 3.

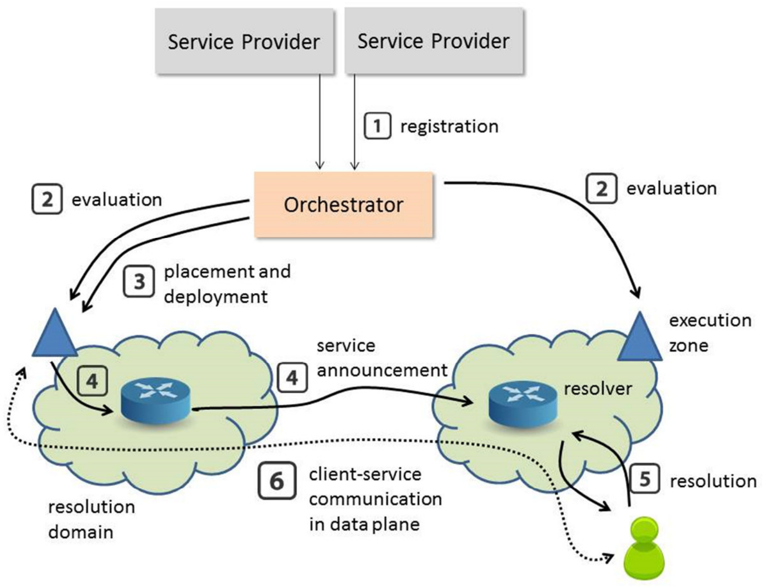

Figure 3: Service lifecycle in service-centric networking

1. Service providers register their service with an orchestrator via an (extended) TOSCA service manifest, containing information such as the service graph identifying service components and their relationship with one another, performance requirements and constraints, and deployment policies.
2. The orchestrator carries out a detailed evaluation of the performance and runtime conditions of a large set of candidate execution locations, named execution zones (EZ). The computational resources may be a dedicated DC of a cloud infrastructure provider, or similar resources co-located with PoP, base stations, etc. provided by an ISP.
3. The evaluation results are used to deploy service replicas in a subset of the EZs, taking into account the service requirements and policies listed in the service manifest.
4. EZs report on their service availability to the service resolution subsystem, which is responsible for creating dynamic forwarding paths for end-user queries. Multiple domain resolvers exchange information on service availability, and each domain has a logically centralized resolver that answers queries from the domain's clients.
5. The resolver returns a locator of the service replica to the client. These locators can contain IPv4/IPv6 address, TCP ports, protocol numbers and/or tunnel identifiers.
6. The client accesses the service replica over a standard IP connection, which is out-of-band of the FUSION framework.

2.1.2 Interaction scenarios

The following figures illustrate the interactions between the architectural components for the main scenarios involved in service deployment, advertisement and querying.

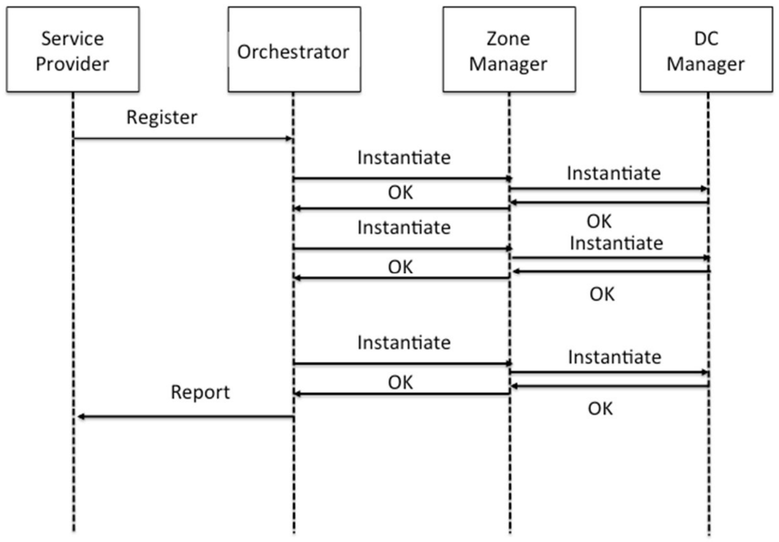

Figure 4: Service registration and deployment

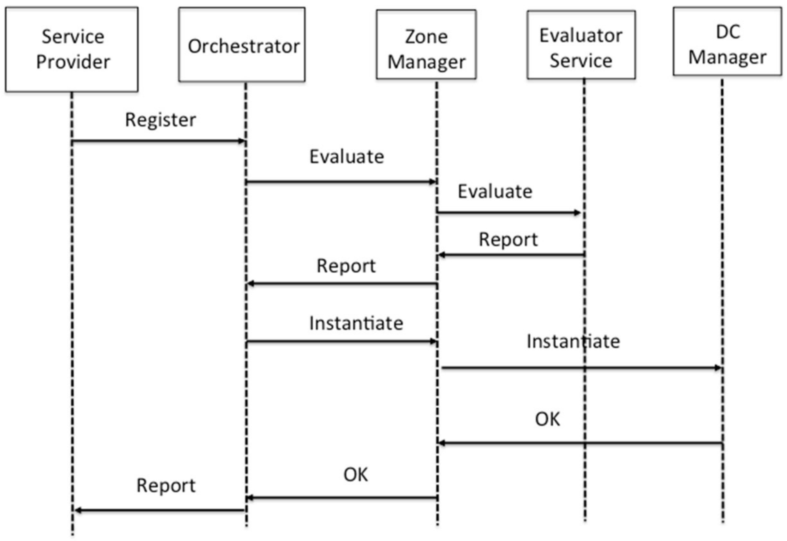

Figure 5: Service registration and deployment following evaluation

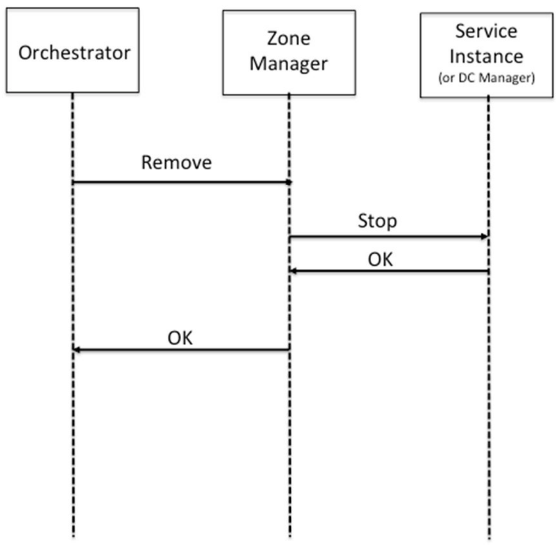

Figure 6: Service instance termination

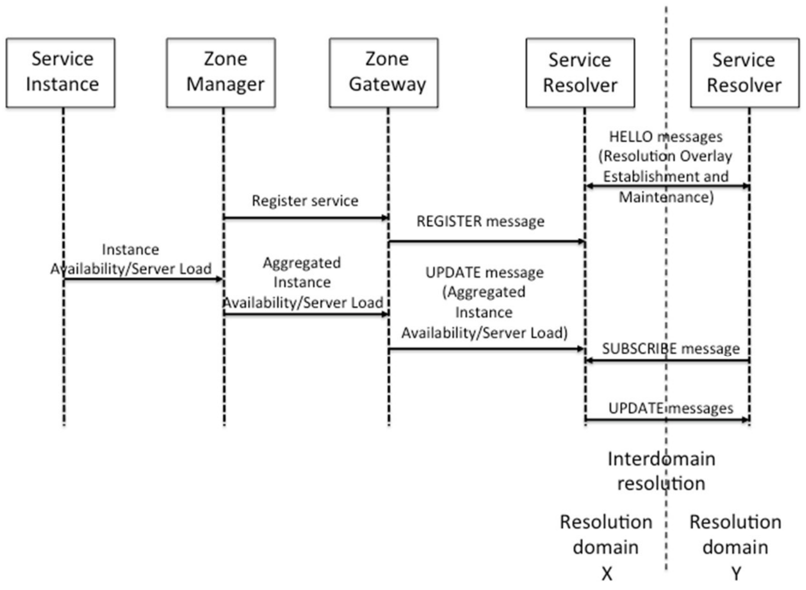
Service Instance
Zone Manager
Zone Gateway
Service Resolver
Service Resolver
HELLO messages (Resolution Overlay Establishment and Maintenance)
Register service
REGISTER message
Instance Availability/Server Load
Aggregated Instance Availability/Server Load
UPDATE message (Aggregated Instance Availability/Server Load)
SUBSCRIBE message
UPDATE messages
Interdomain resolution
Resolution domain X
Resolution domain Y

Figure 7: Service advertisement propagation

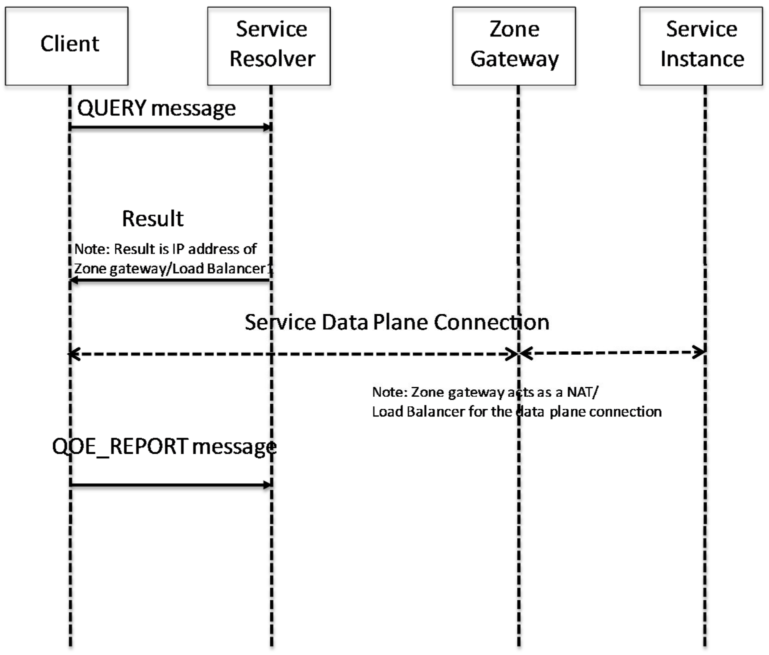
Client
Service Resolver
Zone Gateway
Service Instance
QUERY message
Result
Note: Result is IP address of Zone gateway/Load Balancer
Service Data Plane Connection
Note: Zone gateway acts as a NAT/ Load Balancer for the data plane connection
QOE_REPORT message

Figure 8: Service resolution

The following sequence chart shows service instantiation on demand. Rather than instances being pre-deployed in execution zones and available session slots being announced to the service resolution plane as in the previous cases, the *orchestrator* registers with a service resolver and announces session slots – with the semantic that the slots are potentially available on demand. In this way the service resolver is able to forward queries from clients and the fact that the service has not been instantiated is transparent to the resolution plane.

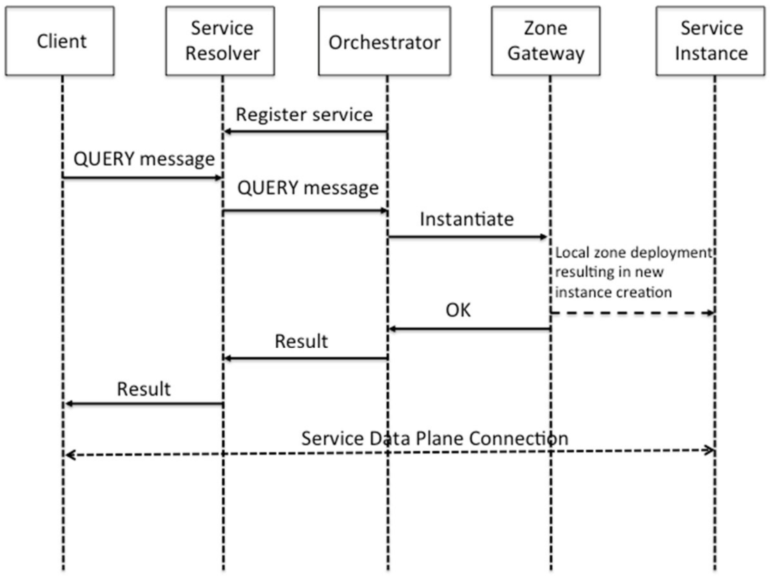

Figure 9: Service instantiation on demand

3. FUSION DEPLOYMENT ISSUES

A heterogeneous, distributed cloud platform is required to be able to cost-efficiently and predictably manage the typical FUSION application services: demanding interactive real-time (media) services running in a cloud environment. To achieve this several aspects are needed: a lightweight composite application service mode; monitoring, profiling and visualization techniques for dealing with heterogeneity, both from an application as well as runtime perspective; inter-service communication mechanisms.

Our main vision is that in the future, more and more demanding workloads and bearer plane services will also be deployed in the cloud. However, as these demanding and sensitive real-time applications are currently typically deployed on dedicated specialized hardware, deploying them as a cloudified service on an unknown general-purpose (third-party) cloud environment can result in substantial performance and QoS degradation, with the net effect that the operational benefits of deploying on a general-purpose cloud with standard cloud nodes are counterfeited by the decrease in efficiency, resulting in the end in higher costs.

As such, the FUSION vision is that for such applications, network functions and workloads to be cost-efficient in a cloud environment, dynamic and adaptive heterogeneous cloud environments are necessary, taking into account the application requirements and the hardware and software platform characteristics, and automatically tuning the software platform and hardware infrastructure based on these requirements and the underlying capabilities and limitations. Apart from being distributed, these heterogeneous cloud environments will consist of novel hardware infrastructures such as micro-servers that are designed for energy efficiency, as well as hardware accelerators for regaining some of the performance efficiency losses from using general purpose hardware only.

This already starts to become visible in the industry, where large players like Google and Facebook and Amazon, owning an impressive array of powerful data centres, are taking various initiatives for increasing the overall efficiency of their data centres. For example, Facebook recently announced to be working in next generation open hardware specifications as part of their Open Compute Project where servers will consist of a heterogeneous mix of specialised hardware to accelerate their infrastructure for better dealing with the new and more demanding applications, including artificial intelligence and virtual reality. Additionally, there are working together with Intel for designing a new processor and corresponding server infrastructure that is optimized for these new workloads.

Similarly, Google recently joined the same Open Compute Project, contributing their new 48V rack specification and a new form factor to allow OCP racks to fit into existing data centres. These new specifications allow for better energy efficiency, and more importantly, also are more cost effective in supporting new higher-performance systems, incorporating high-performance CPUs and GPUs.

On the more distributed nature of clouds, one can clearly observe the massive new investments that Google and Amazon are making and have recently been making in deploying additional huge cloud data centres around the globe. Amazon currently already has 12 data centres across the globe (of which two in Europe now), and Google, who currently only has 4 for its public Google Cloud Engine (GCE), recently announced to be investing in installing 12 additional data centres across the globe.

Detailed considerations on the deployment of FUSION on heterogeneous cloud infrastructure, both within single execution zones as well as across data centres are covered in Deliverable D3.3 [D3.3]. We motivate a lightweight composite application service model and present a number of profiling and monitoring mechanisms for more optimally deploying these sensitive applications on the various heterogeneous runtime environments. We present a visualization mechanism for representing the trade-off between performance and predictability and study various inter-service communication mechanisms for efficiently implementing this lightweight composite service model. All this is summarized in a description of a heterogeneous cloud platform that automates the entire deployment and configuration optimization process.

4. CROSS LAYER MODELLING AND OPTIMISATION ALGORITHMS

4.1 Cross layer interactions

Interactions between the service and network layers of the FUSION architecture are based around the following concepts:

- **Execution zones (EZ):** These are the entities running the computational aspect of the distributed application. Typically they will be cloud/data centres but they can be smaller micro-data centres. They can be run by the ISPs themselves (current trends point to this being an importance new revenue stream for them).
- **Application (service) providers**: These represent the organizations that wish to run services in execution zones. They may instantiate their service instances directly or use a third-party orchestrator (e.g. cloud broker). They will trade-off the placement costs with the quality of service for their users.
- **Internet Service Providers (ISP)**: These will implement resolution algorithms to resolve users queries, mapping service names to locators. They will also trade-off the QoS of their users with the traffic costs imputed to them by these choices.A **session slot** is a lightweight abstract application metric for measuring the resource utilization and availability for a particular service instance. We define routing epoch as the interval between the resolution system making resolution decisions. Regarding to session slots announcement, if sessions are long compared to the routing epoch then the EZs simply announce a snapshot of what is available. However if session durations are short then an announcement of instantaneous availability is more-or-less meaningless. For example: assume a routing epoch of 10 seconds and a service S1 with an average duration of 100 seconds and S2 with an average duration of 1 second, and a single service instance for each service can each handle 2 sessions simultaneously. The EZ would announce 2 available session slots for S1 as the current session is likely to last much longer than the routing epoch. However for S2, if 2 available session slots are announced, it would mean that only two requests should be forwarded to that EZ, even though the currently active session (as well as those arriving in the near future) is very likely to end during the epoch. Therefore the number of session slot would be announced up to 20 for S2, depending on the service arrival time.
- **Placement cost** is the cost of deploying service instances in EZs. This cost is based on how much resources a service instance requires to use. **Transit cost** is the network cost that an ISP has to pay for forwarding data traffic to other ISPs. The transit cost will be computed based on the volume of data traffic as well as which is the ISP to forward that amount of traffic.
- **Utility functions** are defined per service by the service provider in its manifest. In the utility framework, application providers determine utility function by the two latency thresholds: T_{min} and T_{max}. Note that the utility is not restricted to only latency. In future work, we will extend the utility to be a combination of any QoS metrics such as latency, bandwidth, loss, etc. As shown in Figure 10, we use a non-increasing piecewise linear utility function that is characterized by:
  - We use $k <= 1$ to set priority for users. As shown in Figure 10, there are two users of the same service. The user with lower k ($1 <= k_1 < k_2$) would get a higher utility for the same latency value. As a result, the algorithm gives higher priority for users with low k to connect to closer EZs in order to maximize total utility across all users.
  - If $t <= T_{min}$: depending on the service type, an appropriate value of T_{min} is selected meaning that even if the latency reduces below this value, the improvement is not perceived by the users of that service, thus the utility is unchanged ($U_{max} = 1/k$).
  - If $T_{min} < t <= T_{max}$: QoS is within an acceptable range ($0 <= U < 1/k$). User satisfaction reduces as the latency increases. We also define an optional parameter $T_{fair} \subseteq [T_{min}, T_{max}]$ as the point from which users start to feel disappointed as QoS is getting poor (but still in an acceptable range). Note that T_{fair} is used to simply qualify where is the point that has a

fair QoS, and it does not change the slope of the utility graph which is only affected by T_{min}, T_{max} and k.

- If $T_{max} < t$: the request is blocked (no service) because the latency is beyond the acceptable range.

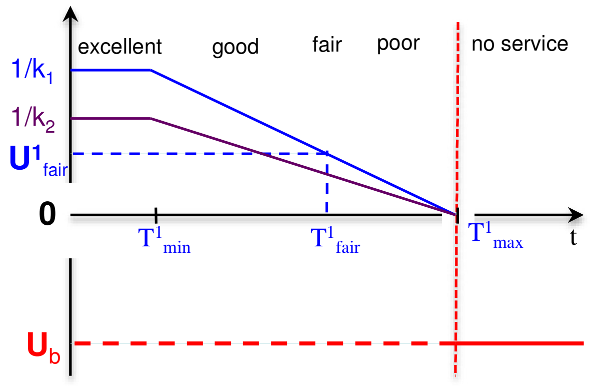

Figure 10 Utility function

In the following we explore whether additional information exchanged between the service provider/orchestrator and the underlying ISPs undertaking service resolution can be of benefit by increasing QoS or reducing costs in the placement and operations of FUSION services.

4.2 Problem statement

As the Internet becomes the enabler for more types of services with a wider spectrum of requirements, pressure is being put onto the Internet ecosystem to facilitate service placement and to select the best replica for each user request. This replication always involves multi-stakeholder trade-offs between costs and quality of service (QoS). In this work, we use our novel utility function (also introduced in deliverable D4.3) to evaluate QoS.

There are many drivers for service placement, including server resilience, network diversity, proximity of servers to users and placement cost. Deploying services closer to the users allows the application providers to improve on QoS metrics like latency and/or throughput for all users. However, the service provider also considers a trade-off between placement cost and user QoS.

Service selection or resolution involves converting a service name to a specific network locator for the selected replica. Our work assumes that the user's ISP is in the best position to make this selection. The ISP has accurate information regarding the user's position in the network, the current network status and, furthermore, it allows the ISP to apply traffic network policies in the selection process to reduce traffic costs.

In this work, we consider the service provider and the ISP as two independent stakeholders. Each of them will act to minimize its cost as well as to maximize the QoS (utility). However, as each stakeholder does its job independently, this can lead to a mismatch which in the end results in poor utility for users. That is the service provider tries to put service instances in cheap execution zone to reduce the placement cost. When the ISP performs resolution, it tends to redirect its users to cheap transit links to minimize the network cost. However, as the service provider has no information about the network cost, this can make the ISP to pay a lot for transit cost if the ISP would like to guarantee good QoS for the users. In this section, we propose optimisation methods for the service provider and the ISP to work in collaboration (e.g. exchange some information between ISP and service provider, ISP pays some extra money for service provider or vice versa) so that the mismatch effect is minimised.

4.3 Optimization formulation

We first assume that there is a centralised model which has all information about the placement and the network cost. This model will minimise the total cost (placement and network cost) as well as maximise user utility. The results of the model will be used as a standard solution for comparing between the solutions when the two stakeholders work independently. Given the utility function introduced in the deliverable D4.3 (and in our paper [PGE+16]), we design the centralised model as well as the service placement and the service selection models using linear programming.

4.3.1 Centralised optimisation model

b_{ij}	bandwidth required by user i to get service j
c^t_{iz}	unit transit cost between user i and EZ z
c^d_z	unit placement cost at EZ z
MIN_COST	the budget cost
$\mathcal{D}$	set of user requests $\mathcal{D} = \{(i,j), \forall i \in \mathcal{I}, j \in \mathcal{J}\}$
d_{ij}	# session slot requested by user i to service j
(i,j,z)	user group i, service j and EZ z
$\mathcal{I}$	set of user groups $\mathcal{I} = \{i\}$
$\mathcal{J}$	set of services $\mathcal{J} = \{j\}$
k_{ij}	priority parameter of user i and service j
l^j_{iz}	latency between user i and EZ z for service j
S^j_z	available session slot of service j at EZ z
t_{ij}	average latency of user i to get service j
U_b	utility value of a blocked user
u_{ij}	utility of user i when getting service j
x^j_{iz}	fraction of user i connects to EZ z to get j
y_{ij}	variable used to compute the utility
$\mathcal{Z}$	set of execution zones (EZ) $\mathcal{Z} = \{z\}$

Table 1 Key notations (in alphabetical order)

We set the transit cost and the placement cost proportionally to the GDP per Capita (PPP) of the countries where execution zones are located (more detail in section 5). The parameter k_{ij} can be used to set different priorities for users and services. For instance when online game and web services share resources, the online game users should have higher priority over the web users to get the service with lower response time. In our simulations (section 5), we consider only one kind of service, and we consider all users are equal, thus we set $k_{ij} = 1$ which means that all the users have the same priority.

4.3.1.1 Step 1: minimizing cost

$$\min \quad \textit{transit\_cost} + \textit{placement\_cost} \tag{1}$$

s.t.

$$\sum_{z \in \mathcal{Z}} x_{iz}^{j} = 1 \qquad \forall (i,j) \in \mathcal{D} \tag{2}$$

$$t_{ij} = \sum_{z \in \mathcal{Z}} l_{iz}^{j} x_{iz}^{j} \qquad \forall (i,j) \in \mathcal{D} \tag{3}$$

$$y_{ij} \geq 0 \qquad \forall (i,j) \in \mathcal{D} \tag{4}$$

$$y_{ij} \geq t_{ij} - T_{min}^{ij} \qquad \forall (i,j) \in \mathcal{D} \tag{5}$$

$$u_{ij} = \frac{T_{max}^{ij} - T_{min}^{ij} - y_{ij}}{k_{ij}(T_{max}^{ij} - T_{min}^{ij})} \qquad \forall (i,j) \in \mathcal{D} \tag{6}$$

$$\textit{transit\_cost} = \sum_{z \in \mathcal{Z}} \sum_{(i,j) \in \mathcal{D}} c_{iz}^{t} b_{ij} x_{iz}^{j} \tag{7}$$

$$\textit{placement\_cost} = \sum_{z \in \mathcal{Z}} \sum_{(i,j) \in \mathcal{D}} c_{z}^{d} d_{ij} x_{iz}^{j} \tag{8}$$

$$x_{iz}^{j} \in [0,1], u_{ij} \leq 1 \forall (i,j) \in \mathcal{D}, z \in \mathcal{Z} \tag{9}$$

Explanation:

- The objective function (1) minimizes the total cost including the transit and the placement cost computed in equations (7) – (8).
- Constraint (2): all the requests of user group i for a specific service j have to be served.
- Equation (3) is used to compute the average latency for the user group i to get the service j. We model connectivity as a full mesh between user groups and EZs. However, for the input of the LP, we remove all pairs of (user group i and EZ z) if the latency $l^{j}_{iz} > T^{ij}_{max}$. This step guarantees that, even without adding explicit constraints in the LP model, the latency for any user group i to connect to any EZ z to get service j is less or equal to T^{ij}_{max}.
- Constraint (4) – (5) ensure that $y_{ij} >= 0$ if $t_{ij} <= T^{ij}_{min}$, otherwise $y_{ij} >= t_{ij} - T^{ij}_{min} > 0$.
- Equation (6) is used to model the utility function defined. More details about the utility function can be found in the deliverable D4.3 or in [PGE+16].
- Constraints (7) and (8) compute the transit cost and the placement cost which are then used in the objective function (1) to minimize the total cost.

4.3.1.2 Step 2: maximizing utility

Given the above formulation we can find the minimum cost (MIN_COST). Then in step 2, we change the objective function to maximise the utility while taking the MIN_COST as a constraint as follows:

$$\max \sum_{(i,j)\in D} u_{ij}$$

s.t.

$$\textit{transit\_cost} + \textit{placement\_cost} \leq MIN\_COST \qquad (10)$$

By using the two-step optimization formulation we can find a solution that maximize the utility while minimizing the transit and placement cost. This solution is considered as an ideal solution as we assume both the network and the placement cost are known in advance. However, in the real world scenario, the service provider and the ISP can work independently. We present next the formulations that the service provider and the ISP use to optimize their own objectives.

4.3.2 Service placement model

The service placement model is similar to the centralised optimization model except there is no information about the transit cost. We simply delete the equation (7) and set the *transit_cost* to zero. In the real world scenario, the service placement model needs to run first to determine where and capacity of service instances to be deployed in execution zones that minimizes the placement cost as well as maximizing the user utility.

4.3.3 Service selection model

Similar to the service placement model, the service selection model can be derived from the centralised model by removing the equation (8) (placement cost) and set to *deploy_cost* to zero. In addition, after placement step, we know the location and the capacity of the deployed service instances. In the selection model, we need to add a constraint on service capacity to guarantee not making the execution zones overloaded.

The optimization formulations above are (pure) linear programming models; therefore they can be solved efficiently in polynomial time. The number of variables x^j_{iz} in the LP problem is |I|* |Z|* |J| where |I| is the number of user groups, |Z| is the number of EZs and |J| is the number of service types. Since |Z| and |J| are usually much smaller than |I|, the worst case complexity of the LP problem will be $O(|I|^{3.5})$.

5. SIMULATION RESULTS OF CROSS-LAYER MODELS

We solve the linear program model using IBM CPLEX solver [CPLEX]. All computations were carried out on a computer equipped with a 3 GHz CPU and 8 GB RAM. We use a dataset with data centres (execution zones) distributed in 216 cities in Western Europe [DCMAP]. The user demand is modelled as Poisson process and is proportional to the population of each city [DATASET]. The latency between users and execution zones are collected based on Haversine distance, the shortest distance between two points around the planet's surface [BRUM13].

The main issue causing mismatch between the placement and the selection models is the different in the network and the placement cost. Our evaluation is through simulation and consists of 3 main scenarios on the cost setup. We first consider the placement cost of each execution zones proportioning to the GDP per Capita (PPP) of the country where the execution zone are located. Then we consider the network cost in three scenario: ***(1) is proportional to the placement cost***; **(2) is allocated random in range of the placement cost** and **(3) is inverse with the placement cost**. For each scenario, we will see different levels of mismatch affecting the QoS and then we propose methods to overcome the problem.

For each scenario, we first run the centralised model in section 4.3.1 as an ideal solution to compare with other cases when we run the placement process first and based on this, we find selection solutions. As the selection solutions are based on where we deploy service instances, our solution to reduce the negative effect of mismatch is that the placement process will take into account the top **X% cost** (X is the parameter for the evaluation) of the most expensive transit links. This method is done by running the centralised model in section 4.3.1 but we only consider the transit cost for X% of the links, other links will have zero transit cost. This helps the placement process to avoid expensive transit links which causes more cost for the ISP when doing selection. The centralised model is equivalent to the case X = 100% cost.

5.1.1 Scenario 1: totally aligned cost

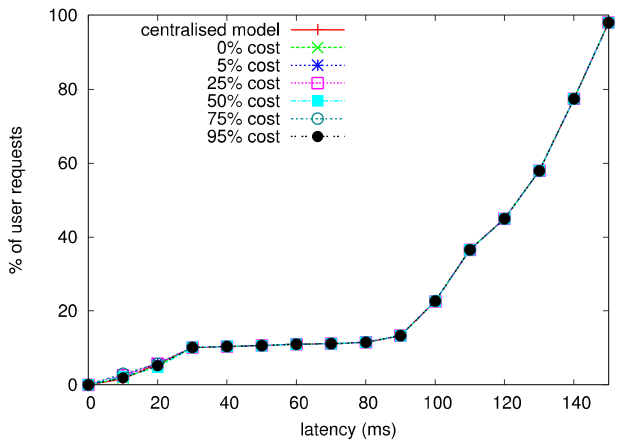

Figure 11 CDF user latency

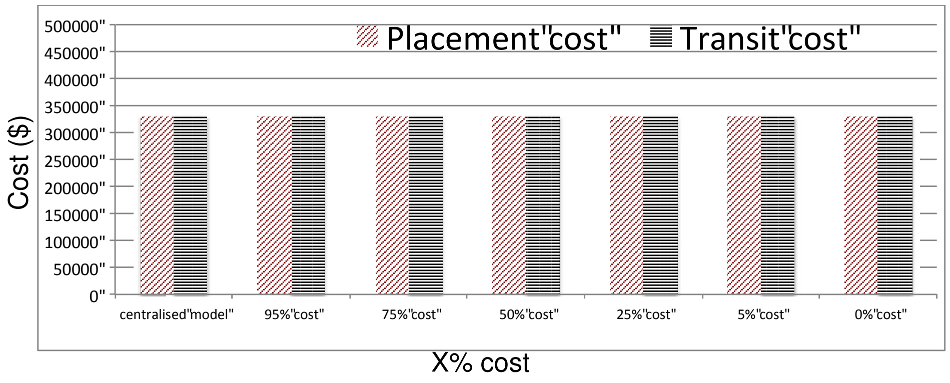

Figure 12 Placement cost vs. transit cost

In this scenario, as the transit cost and the placement cost are totally aligned, there is no effect of mismatch between the placement and the selection model. This has been shown in Figure 11 and Figure 12 where we do not see any differences between the placement and the selection solutions in term of QoS (latency) and cost. This is the ideal scenario when optimizing the placement will also result in optimal solution for the selection and vice versa.

5.1.2 Scenario 2: random cost

5.1.2.1 Without cost constraint in the selection model

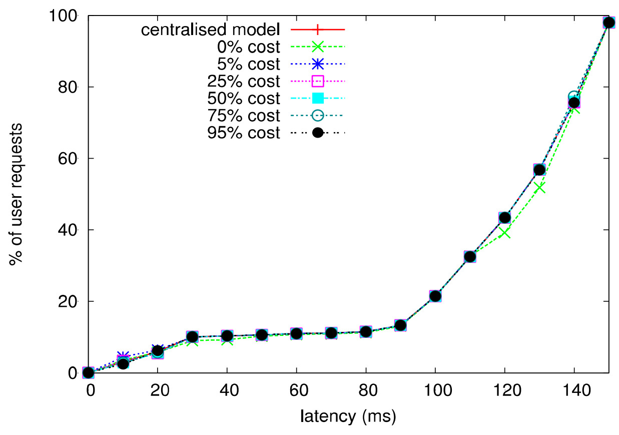

Figure 13 CDF user latency

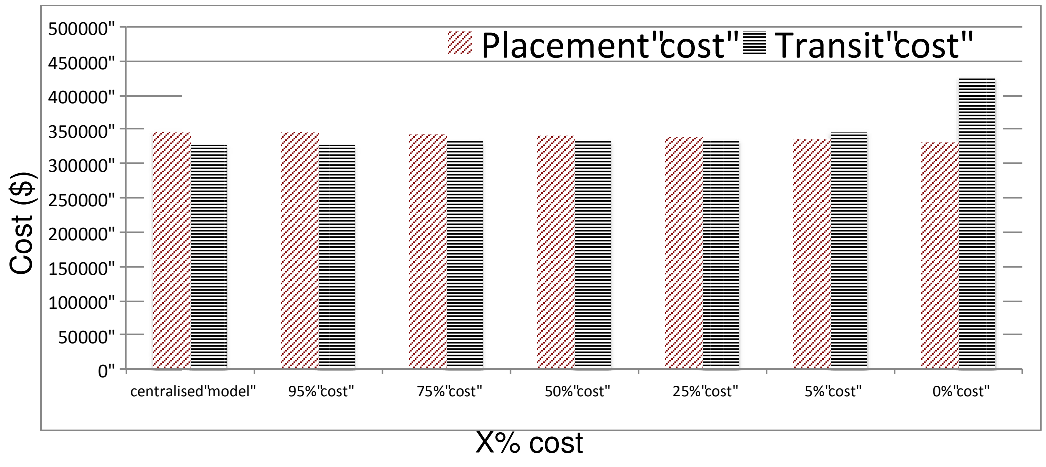

Figure 14 Placement cost vs. transit cost

In this scenario, we first run the placement model with different values of X% cost. Then based on placement solutions, we run the selection model to decide which user requests go to which execution zones. In the selection process, we do not care about the transit cost, just simply try to maximize the utility. For that reason, we can see in Figure 13 that QoS for all different X% values are similar. This is because in all cases, we try to maximize the utility therefore the solutions we get are the ones with the best QoS. The only small difference is in case X = 0%, the selection solution is slightly worse in terms of QoS. Looking at Figure 14 we see that the transit cost that the ISP must pay in case X = 0% is significantly high (to achieve the QoS as in Figure 13). However, if the service provider takes into account X = 5% of the transit links in the optimization process, we can see a win-win game: the ISP pay much less in transit cost (Figure 14) while QoS increases (Figure 13). However, in this case, the service provider must pay a higher cost compared to the case of X = 0%. *To deal with this problem, we can propose that the ISP will pay a compensated cost for the service provider which is still much lower than which the ISP has to pay for the transit cost in case X = 0%.*

5.1.2.2 With cost constraints in the selection model

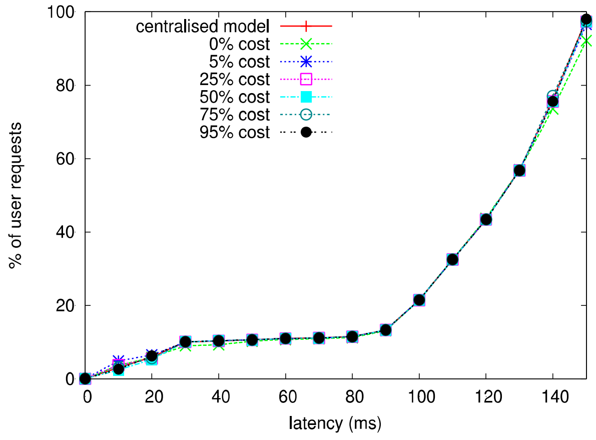

Figure 15 CDF user latency

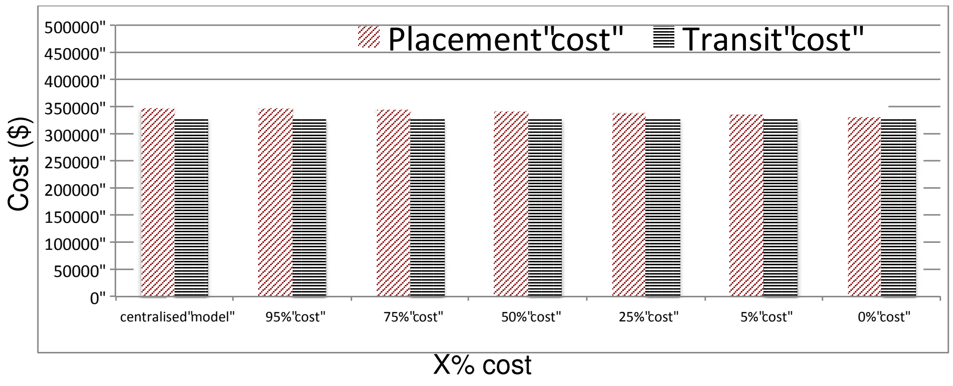

Figure 16 Placement cost vs. transit cost

In this scenario, we assume that the ISP has limited budget and cannot pay over that amount of money. We run the centralised model first to see in the ideal case how much the transit cost is and we use this as the limited budget of the ISP. As shown in Figure 15, we see some blocked user requests if X = 0% meaning that the mismatch between service provider and ISP will result in blocking user requests as the ISP cannot find enough available execution zones to forward the requests. However, given X = 5% we can see good QoS (Figure 15) but the service provider has to pay a slightly higher cost (Figure 16). In this case, as the ISP cannot pay a compensated cost to the service provider, they have to find another way to negotiate or have to agree in blocking some user requests as it is in the case X = 0%.

5.1.3 Scenario 3: inverse cost

5.1.3.1 Without cost constraints in the selection model

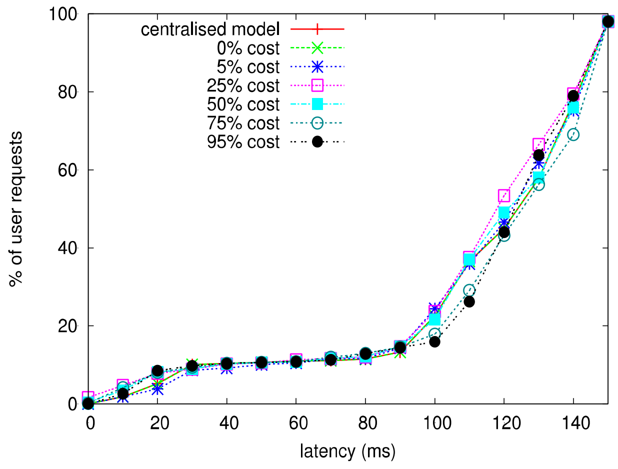

Figure 17 CDF user latency

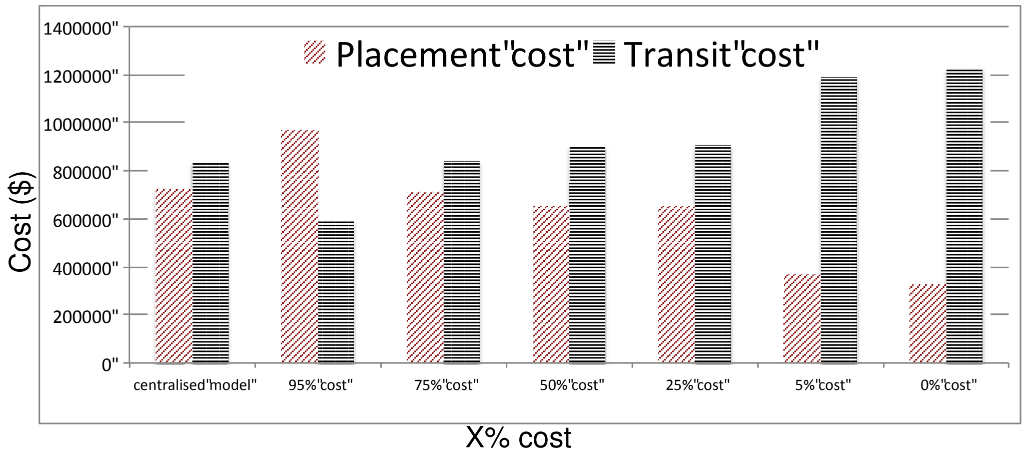

Figure 18 Placement cost vs. transit cost

We consider in this scenario the worst case as the placement cost and the transit cost are inverse. Similar to the scenario in 5.1.2.1, a possible solution is that the ISP will pay a compensated cost for the service provider so that the service provider will consider some transit link cost in the optimization. However, as the mismatch level is the highest, to achieve a relatively good QoS (Figure 17), the service provider need to take into account X = 25% transit link costs (Figure 18).

5.1.3.2 *With cost constraints in the selection model*

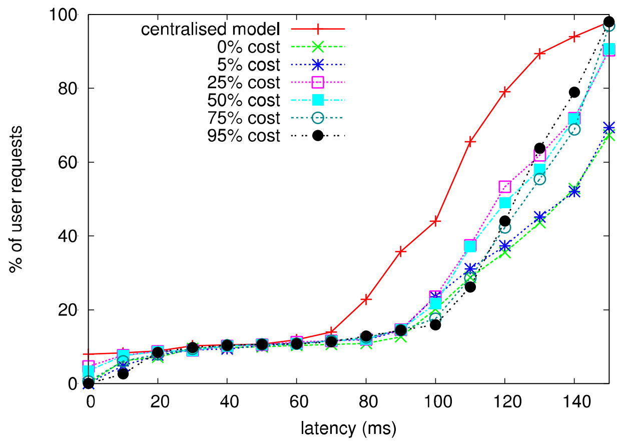

Figure 19 CDF user latency

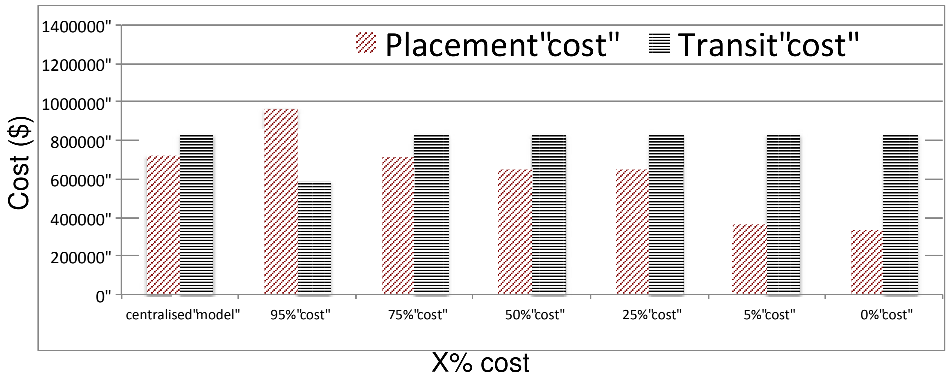

Figure 20 Placement cost vs. transit cost

This scenario is similar to the one in 5.1.2.2: we limit the budget of the ISP. As the mismatch level is the highest, we can see very different ranges of QoS for different X% (Figure 19). We found that to avoid blocking requests, the service provider needs to take into account 95% of transit links which cost him a significant amount of cost (Figure 20). Therefore, depending on a negotiation between the service provider and the ISP, we can find a suitable operating point based on QoS and cost.

6. CONCLUSIONS

A centralised algorithm with full knowledge of service placement costs and network transit costs can optimise the placement of services to balance costs and service utility. In practice this is unrealistic as the placement algorithm, run by the service orchestrator on behalf of the service provider, would need full knowledge of the transit costs of the underlying ISPs. ISPs are unlikely to expose full transit cost information that has been negotiated between it and its transit providers to third parties due to business confidentiality concerns. The modelling work described in this document show that if the ISPs inform the service orchestrator of a proportion of transit costs then transit costs can be reduced and QoS can be increased. However, taking these transit costs into account can increase the overall cost of the service orchestrator as avoiding high cost transit links can mean that higher cost execution zones need to be used to meet QoS constraints. In this case there is a misalignment of incentives: the orchestrator is being asked to increase its placement costs in order to reduce the transit costs of the ISPs. In some cases QoS suffers when the placement algorithm has no knowledge of transit costs when the resolution algorithm is cost constrained, e.g. there is increased latency in Figure 13, or increased blocked requests in Figure 15. In these cases it is in the interests of the placement algorithm to take into account the exposed transit costs in its placement decision as it will increase QoS for its users. If the ISPs wish to reduce costs further then they can compensate the service provider for using more expensive execution zones. For example, Figure 14 shows that if the ISP exposes the top 5% of transit costs to the service orchestrator and requests that the placement algorithm avoids placing services in EZs that use those transit links, then part of the cost saving made by the ISP could be used to compensate the service provider for increased placement costs and both parties will benefit.

The results of the studies reported in this deliverable show that the proportion of transit costs that need to be exposed from ISPs to service providers depends on the degree of mismatch between the transit and placement costs for selecting service instances in specific EZs. In the case of complete alignment of costs there is no benefit in exposing any cost information. In an artificial case, see section 5.1.3, where costs are completely reversed (the highest cost EZs are available over the lowest cost transit links and vice versa), then up to 95% of transit costs would need to be exposed. In intermediate cases, such as those examined in section 5.1.2, the top 5% of transit costs need to be identified to the service provider to improve QoS and reduce transit costs for the ISP.

In conclusion, interactions between the service placement and resolution layers can be extended beyond utility functions and session slots to include a proportion of transit costs being exposed to the service orchestrator to increase overall QoS to the users and to reduce ISP costs. In addition if compensation payments are made from the ISP to the service provider then overall costs can be reduced. ALTO can be used as the mechanism for the exchange of cost information between the ISPs and the service provider/orchestrator [ALTO].

7. REFERENCES

[ALTO] RFC7285 Application-Layer Traffic Optimization Protocol

[BRUM13] Glen Van Brummelen. “Heavenly Mathematics: The Forgotten Art of Spherical Trigonometry” in Princeton University Press, 2013.

[CPLEX] www-01.ibm.com/software/commerce/optimization/cplex-optimizer

[D3.3] FUSION D3.3, Final node design, algorithms and protocols for service-oriented network management and simulation results, June 2016.

[DATASET] https://github.com/richardclegg/multiuservideostream

[DCMAP] http://www.datacentermap.com

[ZHZ+10] Z. Zhang, Y. Charlie Hu, M.Zhang, R.Mahajan, A. Greenberg and B. Christian. “Optimizing Cost and Performance Online Service Provider Networks” in NSDI 2010.

[PGE+16] T. K. Phan, D. Griffin, E. Maini and M. Rio. “Utility-maximizing Server Selection” in IFIP NETWORKING 2016.

8. APPENDIX A: FINAL SPECIFICATION OF FUSION INTERFACES

This appendix is formed by internal report I2.1, “Final Specification of FUSION Interfaces”. The report provides the full details on the FUSION protocol specifications, detailing the various APIs, Objects as well as parameters and expected values.

TABLE OF CONTENTS

1. Application Service Interface

1.1 Object Definitions

Object 'State'

ServiceState or InstanceState is an enumerator, representing the various states (as well as transition states) an application service can be in:

- REGISTERING/REGISTERED
- PROVISIONING/PROVISIONED
- DEPLOYING/DEPLOYED
- STARTING/STARTED/RUNNING
- STOPPING/STOPPED
- TERMINATING/TERMINATED
- etc.

In the FUSION prototype, we implemented a subset of these states.

Parameter name	Type	Min Occurence	Max Occurence	Description
state	Enum	1	1	Service and/or instance state

Object 'ServiceManifest'

ServiceManifest represents a complex structure that defines the complete or partial description of a service, its dependencies, policies, image locations, instantiation parameters, etc. This can be provided in a TOSCA format, though the prototype implementation supports a more lightweight JSON-based variant. Below a subset of the various key parameters.

The instantiation parameters are a collection of service-specific parameters for specializing the instantiation of a particular application service, and are opaque w.r.t. FUSION orchestration. These parameters allow to specialize generic application service components for different service types.

This contrasts with deployparameters, which are FUSION-specific configuration parameters during deployment. These may include CPU, memory, hardware, real-time, or other requirements FUSION should or could take into account while deploying the service.

Parameter name	Type	Min Occurence	Max Occurence	Description
serviceid	String	1	1	ID of the service
evaluators	List of evaluatorids	0	-	List of type 1 and type 2 evaluator service IDs
autoscale	Dictionary	0	1	Autoscaling policies
ports	List of Ports	0	-	List of explicit application ports.
instantiationparams	Dictionary	0	1	List of service instantiation parameters
deployparams	Dictionary	0	1	List of FUSION deployment parameters

Object 'SessionSlots'

SessionSlots summarizes the total and available session slots of a particular service and/or an individual service instance. Each entity should provide at least two of the parameter values; the others can be infered automatically

Parameter name	Type	Min Occurence	Max Occurence	Description
total	Integer	0	1	Total session slots
used	Integer	0	1	Number of session slots that are currently in use
free	Integer	0	1	Number of immediately available session slots

Object 'ServiceConfig'

ServiceConfig represents a specific service configuration that a particular generic service/resource instance can be turned in to.
This is related to the multi-confguration services, or service aliasing feature, where particular service/resource instances can host multiple service types at once, each with their own custom configuration.

Parameter name	Type	Min Occurence	Max Occurence	Description
configid	String	1	1	Configuration ID
serviceid	String	1	1	Service ID
index	Integer	1	1	Service instance configuration index
instantiationparams	Dictionary	0	1	Service instantiation parameters

1.2 General Management Interface

GET /1.0/ping

Check FUSION Application Service availability

Response Output Data

None

Notes

- Services are not required to implement this interface

1.3 Service Instance Management Interface

GET /1.0/configs

Returns a list of all active service configurations associated with this service instance.

Response Output Data

List of ServiceConfigs

Notes

- Services are not required to implement this interface

POST /1.0/configs

Add a new service configuration to this instance, with proper instantiation parameters.

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of the service
instantiationparams	Dictionary	no	none	Service instantiation parameters

Response Output Data

None

Notes

- Services are not required to implement this interface

DELETE /1.0/configs

Remove all service configurations from this service instance (possibly terminating all active sessions for these configurations first).

Response Output Data

None

Notes

- Services are not required to implement this interface

GET /1.0/state

Fetch current internal state information of this service instance.

Response Output Data

InstanceState

Notes

- Services are not required to implement this interface

PUT /1.0/state

Change internal service lifecycle state (e.g., start, pause, serialize, terminate, etc.) of this instance. This can be used for gracefully shutting down particular instances, (re)starting it, etc.

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
state	String	yes	none	New instance state

Response Output Data

None

Notes

- Services are not required to implement this interface

DELETE /1.0/state

Gracefully terminate this instance.

Response Output Data

None

Notes

- Services are not required to implement this interface

GET /1.0/configs/{configid}

Fetch detailed information of a particular service configuration.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
configid	String	yes	none	config ID

Response Output Data

ServiceConfig

Notes

- This service API is optional

DELETE /1.0/configs/{configid}

Remove this service configuration from this instance (possibly terminating all active sessions from this configuration first).

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
configid	String	yes	none	config ID

Response Output Data

None

Notes

- This service API is optional

GET /1.0/slots

Fetch session slot information from this service instance. Note that a service instance can also push changes to its session slot information to the zone manager.

Response Output Data

SessionSlots

Notes

- Services are not required to implement this interface

PUT /1.0/slots

Change the number of available session slots of this service instance.

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
slots	SessionSlots	yes	none	New session slots

Response Output Data

None

Notes

- Services are not required to implement this interface

DELETE /1.0/slots

Remove all available slots from this service instance (i.e., terminate service instance).

Response Output Data

None

Notes

- Services are not required to implement this interface

2. Data Centre Adaptor Interface

2.1 Object Definitions

Object 'Port'

Port represents a mapping of an particular public service port (type) onto the actual addressible endpoint.

Parameter name	Type	Min Occurence	Max Occurence	Description
type	String	1	1	Port type
port	Int	1	1	Port
endpoint	URL	1	1	Public endpoint

2.2 Environment Management Interface

GET /1.0/environments

Returns a (filtered) list of all registered runtime environments in this DCA.

Request Query Parameters

Parameter name	Type	Mandatory	Default value	Description
envids	List of Strings	no	none	List of runtime environment IDs

Response Output Data

List of Environments

Notes

- Additional filter parameters such as feature and/or regional in type could be added as well

GET /1.0/environments/{envid}

Fetch detailed information of this zone (i.e., services, state, etc.).

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
envid	String	yes	none	ID of runtime environment

Response Output Data

Environment

2.3 General Management Interface

GET /1.0/ping

Check FUSION DC Adaptor availability

Response Output Data

None

2.4 Service Instance Management Interface

GET /1.0/instances/{instanceid}/configs/{configid}

Fetch detailed information a particular service configuration associated with this instance.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
instanceid	String	yes	none	instance ID
configid	String	yes	none	config ID

Response Output Data

ServiceConfig

DELETE /1.0/instances/{instanceid}/configs/{configid}

Remove this service configuration from the instance, possibly terminating all active sessions first.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
instanceid	String	yes	none	instance ID
configid	String	yes	none	config ID

Response Output Data

None

GET /1.0/instances

Returns a (filtered) list of all service component instances in this DCA, along with their current state and session slot information.

Request Query Parameters

Parameter name	Type	Mandatory	Default value	Description
zoneids	List of Strings	no	none	List of zone IDs
serviceids	List of Strings	no	none	List of service IDs
instanceids	List of Strings	no	none	List of instance IDs

Response Output Data

List of Instances

POST /1.0/instances

Create a new service component instance in this DCA. Typically, this also immediately starts the instance, but this could also simply be a provisioning or creation request.

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
zoneid	String	yes	none	ID of the zone
manifest	ServiceManifest	yes	none	partial service manifest description
instantiationparams	Dictionary	no	none	Service instantiation parameters
deployparams	Dictionary	no	none	Service deployment parameters

Response Output Data

None

DELETE /1.0/instances

Remove all or a subset of all service instances from this zone (e.g., in the context of a service provider).

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
zoneids	List of Strings	no	none	List of zone IDs
serviceids	List of Strings	no	none	List of service IDs
instanceids	List of Strings	no	none	List of instance IDs

Response Output Data

None

GET /1.0/instances/{instanceid}/ports/{port}

Returns the port mapping information of the specific service port type for this instance

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
instanceid	String	yes	none	ID of instance
port	String	yes	none	service port type

Response Output Data

ServicePort

GET /1.0/instances/{instanceid}

Fetch detailed information of this high-level service instance (i.e., session slots, runtime state, etc.).

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
instanceid	String	yes	none	ID of instance

Response Output Data

Instance

DELETE /1.0/instances/{instanceid}

Terminate and remove this service instance from this DCA.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
instanceid	String	yes	none	ID of instance

Response Output Data

None

GET /1.0/instances/{instanceid}/ports

Returns a (filtered) list of all port/endpoint mappings with respect to this instance. This mapping provides a translation of the private IP address and ports to the public IP address and ports.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
instanceid	String	yes	none	ID of instance

Response Output Data

List of ServicePorts

GET /1.0/instances/{instanceid}/state

Fetch state information of this service component instance in this DCA.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
instanceid	String	yes	none	ID of instance

Response Output Data

InstanceState

PUT /1.0/instances/{instanceid}/state

Change lifecycle state of the instance (e.g., provision, deploy, terminate).

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
instanceid	String	yes	none	ID of instance

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
state	String	yes	none	New instance state

Response Output Data

None

DELETE /1.0/instances/{instanceid}/state

Remove this instance from the DCA (possibly terminating the instance first).

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
instanceid	String	yes	none	ID of instance

Response Output Data

None

GET /1.0/instances/{instanceid}/configs

Returns a list of all service configurations associated with this instance.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
instanceid	String	yes	none	instance ID

Response Output Data

List of ServiceConfigs

POST /1.0/instances/{instanceid}/configs

Add a new service configuration to this instance, with proper service instantiation parameters.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
instanceid	String	yes	none	instance ID

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of the service
instantiationparams	Dictionary	no	none	Service instantiation parameters

Response Output Data

None

DELETE /1.0/instances/{instanceid}/configs

Remove all service configurations from this instance.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
instanceid	String	yes	none	instance ID

Response Output Data

None

2.5 User Management Interface

GET /1.0/users/{userid}

Fetch detailed information of this user.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
userid	String	yes	none	ID of user

Response Output Data

User

DELETE /1.0/users/{userid}

Remove this user from this DCA. Users can only be removed when all their state is removed first.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
userid	String	yes	none	ID of user

Response Output Data

None

GET /1.0/users

Returns a (filtered) list of all registered users in this DCA.

Request Query Parameters

Parameter name	Type	Mandatory	Default value	Description
userids	List of Strings	no	none	List of user IDs

Response Output Data

List of Users

POST /1.0/users

Register a new user in this DCA.

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
userid	String	yes	none	User ID
user	String	yes	none	User name
roles	List of Strings	no	none	User roles
password	String	yes	none	User password

Response Output Data

None

DELETE /1.0/users

Remove all (or a subset of) users from this DCA.

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
userids	List of Strings	no	none	List of user IDs

Response Output Data

None

2.6 Zone Management Interface

GET /1.0/zones

Returns a (filtered) list of all registered zones in this DCA.

Request Query Parameters

Parameter name	Type	Mandatory	Default value	Description
zoneids	List of Strings	no	none	List of zone IDs

Response Output Data

List of ZoneEndpoints

Notes

- Additional filter parameters such as feature and/or regional in type could be added as well

POST /1.0/zones

Add a new zone to this DCA.

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
zoneid	String	yes	none	zone ID
endpoint	String	yes	none	zone Endpoint

Response Output Data

None

Notes

- Additional filter parameters such as feature and/or regional in type could be added as well

DELETE /1.0/zones

Remove all registered zones from this DCA.

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
zoneids	List of Strings	no	none	List of zone IDs

Response Output Data

None

Notes

- Additional filter parameters such as feature and/or regional in type could be added as well

GET /1.0/zones/{zoneid}

Fetch detailed information of this zone (i.e., services, state, etc.).

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
zoneid	String	yes	none	ID of execution zone

Response Output Data

Zone

DELETE /1.0/zones/{zoneid}

Remove this zone from this DCA (terminating all running instances first).

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
zoneid	String	yes	none	ID of execution zone

Response Output Data

None

3. Domain Orchestrator Interface

3.1 General Management Interface

GET /1.0/ping

Check FUSION domain orchestrator availability

Response Output Data

None

3.2 Resolver Management Interface

GET /1.0/resolvers

Returns a (filtered) list of all registered service resolvers in this domain

Request Query Parameters

Parameter name	Type	Mandatory	Default value	Description
resolverids	List of Strings	no	none	List of resolver IDs

Response Output Data

List of Resolvers

Notes

- Additional filter parameters such as feature and/or regional in type could be added as well

POST /1.0/resolvers

Register a new resolver in this domain (only can be done by domain admin and authorized resolver admins).

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
resolverid	String	yes	none	resolver ID
endpoint	String	yes	none	zone Endpoint
userid	String	yes	none	resolver user ID

Response Output Data

None

Notes

- Additional filter parameters such as feature and/or regional in type could be added as well

DELETE /1.0/resolvers

Remove all or a subset of resolvers from this domain (e.g., in the context of a resolver admin).

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
resolverids	List of Strings	no	none	List of resolver IDs

Response Output Data

None

Notes

- Additional filter parameters such as feature and/or regional in type could be added as well

GET /1.0/resolvers/{resolverid}

Fetch detailed information of this resolver.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
resolverid	String	yes	none	ID of FUSION resolver

Response Output Data

Resolver

DELETE /1.0/resolvers/{resolverid}

Remove this resolver from this domain.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
resolverid	String	yes	none	ID of FUSION resolver

Response Output Data

None

3.3 Service Management Interface

GET /1.0/services/{serviceid}

Fetch detailed information of this service (i.e., manifest, runtime state, etc.).

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service

Response Output Data

Service

DELETE /1.0/services/{serviceid}

Remove this service from this domain (possibly terminating all running instances first).

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service

Response Output Data

None

GET /1.0/services/{serviceid}/state

Fetch global state information of this service in this domain.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service

Response Output Data

ServiceState

PUT /1.0/services/{serviceid}/state

Change global domain service lifecycle state (e.g., register, provision, deploy, terminate) in the entire domain.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
state	String	yes	none	New service state
deployparams	Dictionary	no	none	Service deployment parameters
instantiationparams	Dictionary	no	none	Service instantiation parameters
terminateparams	Dictionary	no	none	Service termination parameters

Response Output Data

None

DELETE /1.0/services/{serviceid}/state

Remove this service from this domain (possibly terminating all running instances first).

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
zoneids	List of Strings	no	none	Optional list of zoneids to terminate and unregister this service from

Response Output Data

None

GET /1.0/services/{serviceid}/zones

Returns a (filtered) list of all zones in this domain where this service is registered.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service

Request Query Parameters

Parameter name	Type	Mandatory	Default value	Description
zoneids	List of Strings	no	none	List of zone IDs

Response Output Data

List of ZoneEndpoints

Notes

- Additional filter parameters such as feature and/or regional in type could be added as well

POST /1.0/services/{serviceid}/zones

Register this service in a new zone.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
zoneid	String	yes	none	zone ID

Response Output Data

None

Notes

- Additional filter parameters such as feature and/or regional in type could be added as well

DELETE /1.0/services/{serviceid}/zones

Remove this service from all or a subset of zones in this domain (terminating all instances and unregistering the service first in all respective zones).

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
zoneids	List of Strings	no	none	List of zone IDs

Response Output Data

None

Notes

- Additional filter parameters such as feature and/or regional in type could be added as well

GET /1.0/services

Returns a (filtered) list of all registered services in this orchestration domain, along with their current state and session slot information. Filtering is done based on ACL as well as an optional filtering request parameter

Request Query Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceids	List of Strings	no	none	List of service IDs

Response Output Data

List of Services

POST /1.0/services

Register a new service in this domain.

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
manifest	ServiceManifest	yes	none	full service manifest in some format (JSON, XML, YAML, ...)

Response Output Data

None

DELETE /1.0/services

Remove all (or a subset of all) services from this domain (e.g., in the context of a service provider).

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceids	List of Strings	no	none	List of service IDs

Response Output Data

None

GET /1.0/services/{serviceid}/slots

Fetch detailed information of the number of available and used session slots of this service in the entire domain.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service

Response Output Data

SessionSlots

PUT /1.0/services/{serviceid}/slots

Change the number of available session slots of this service in the entire domain.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
slots	SessionSlots	yes	none	New Session slots state
zoneid	String	no	none	Zone ID to change slots

Response Output Data

None

DELETE /1.0/services/{serviceid}/slots

Remove all active slots from this service in this domain (or optional zoneids)(i.e., terminate service instances).

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
zoneids	List of Strings	no	none	Optional list of zoneids to terminate this service from

Response Output Data

None

3.4 Service Zone Management Interface

GET /1.0/zones/{zoneid}/services/{serviceid}

Fetch detailed information of this service in this specific zone.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
zoneid	String	yes	none	ID of zone
serviceid	String	yes	none	ID of service

Response Output Data

Service

DELETE /1.0/zones/{zoneid}/services/{serviceid}

Remove this service entirely from this zone in this domain (possibly terminating all running instances first).

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
zoneid	String	yes	none	ID of zone
serviceid	String	yes	none	ID of service

Response Output Data

None

GET /1.0/zones/{zoneid}/services/{serviceid}/state

Fetch global state information of this service in this zone.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
zoneid	String	yes	none	ID of zone
serviceid	String	yes	none	ID of service

Response Output Data

ServiceZoneState

PUT /1.0/zones/{zoneid}/services/{serviceid}/state

Change service lifecycle state of this service in this specific zone (e.g., register, provision, deploy, terminate).

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
zoneid	String	yes	none	ID of zone
serviceid	String	yes	none	ID of service

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
state	String	yes	none	New service state
deployparams	Dictionary	no	none	Service deployment parameters
instantiationparams	Dictionary	no	none	Service instantiation parameters
terminateparams	Dictionary	no	none	Service termination parameters

Response Output Data

None

DELETE /1.0/zones/{zoneid}/services/{serviceid}/state

Remove this service from this zone (possibly terminating all running instances first).

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
zoneid	String	yes	none	ID of zone
serviceid	String	yes	none	ID of service

Response Output Data

None

GET /1.0/services/{serviceid}/zones/{zoneid}/slots

Fetch detailed information of the number of available and used session slots of this service in this specific zone.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service
zoneid	String	yes	none	ID of zone

Response Output Data

SessionSlots

PUT /1.0/services/{serviceid}/zones/{zoneid}/slots

Change the number of available session slots of this service in this zone.
Note that in case of a zone manager, this will simply update the state information. In all other cases though, it will deploy or terminate session slots.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service
zoneid	String	yes	none	ID of zone

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
slots	SessionSlots	yes	none	New Session slots state

Response Output Data

None

DELETE /1.0/services/{serviceid}/zones/{zoneid}/slots

Remove all active slots from this service in this zone (i.e., terminate service instances in this zone).

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service
zoneid	String	yes	none	ID of zone

Response Output Data

None

GET /1.0/zones/{zoneid}/services/{serviceid}/slots

Fetch detailed information of the number of available and used session slots of this service in this specific zone.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
zoneid	String	yes	none	ID of zone
serviceid	String	yes	none	ID of service

Response Output Data

SessionSlots

PUT /1.0/zones/{zoneid}/services/{serviceid}/slots

Change the number of available session slots of this service in this zone.
Note that in case of a zone manager, this will simply update the state information. In all other cases though, it will deploy or terminate session slots.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
zoneid	String	yes	none	ID of zone
serviceid	String	yes	none	ID of service

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
slots	SessionSlots	yes	none	New Session slots state

Response Output Data

None

DELETE /1.0/zones/{zoneid}/services/{serviceid}/slots

Remove all active slots from this service in this zone (i.e., terminate service instances in this zone).

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
zoneid	String	yes	none	ID of zone
serviceid	String	yes	none	ID of service

Response Output Data

None

GET /1.0/services/{serviceid}/zones/{zoneid}/state

Fetch global state information of this service in this zone.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service
zoneid	String	yes	none	ID of zone

Response Output Data

ServiceZoneState

PUT /1.0/services/{serviceid}/zones/{zoneid}/state

Change service lifecycle state of this service in this specific zone (e.g., register, provision, deploy, terminate).

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service
zoneid	String	yes	none	ID of zone

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
state	String	yes	none	New service state
deployparams	Dictionary	no	none	Service deployment parameters
instantiationparams	Dictionary	no	none	Service instantiation parameters
terminateparams	Dictionary	no	none	Service termination parameters

Response Output Data

None

DELETE /1.0/services/{serviceid}/zones/{zoneid}/state

Remove this service from this zone (possibly terminating all running instances first).

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service
zoneid	String	yes	none	ID of zone

Response Output Data

None

GET /1.0/services/{serviceid}/zones/{zoneid}

Fetch detailed information of this service in this specific zone.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service
zoneid	String	yes	none	ID of zone

Response Output Data

Service

DELETE /1.0/services/{serviceid}/zones/{zoneid}

Remove this service entirely from this zone in this domain (possibly terminating all running instances first).

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service
zoneid	String	yes	none	ID of zone

Response Output Data

None

3.5 User Management Interface

GET /1.0/users

Returns a (filtered) list of all registered users in this domain.

Request Query Parameters

Parameter name	Type	Mandatory	Default value	Description
userids	List of Strings	no	none	List of user IDs

Response Output Data

List of Users

POST /1.0/users

Register a new user in this domain.

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
userid	String	yes	none	User ID
user	String	yes	none	User name
roles	List of Strings	no	none	User roles
password	String	yes	none	User password

Response Output Data

None

DELETE /1.0/users

Remove all (or a subset of) users from this domain.

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
userids	List of Strings	no	none	List of user IDs

Response Output Data

None

GET /1.0/users/{userid}

Fetch detailed information of this user.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
userid	String	yes	none	ID of user

Response Output Data

User

DELETE /1.0/users/{userid}

Remove this user from this domain (possibly terminating all running instances first).

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
userid	String	yes	none	ID of user

Response Output Data

None

3.6 Zone Management Interface

GET /1.0/zones/{zoneid}/services

Returns a (filtered) list of all registered services in a specific zone in this orchestration domain.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
zoneid	String	yes	none	ID of execution zone

Response Output Data

List of Services (registered in zone)

POST /1.0/zones/{zoneid}/services

Register a new service in this zone.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
zoneid	String	yes	none	ID of execution zone

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service to register in selected execution zone

Response Output Data

None

DELETE /1.0/zones/{zoneid}/services

Remove all services from this zone (terminating all instances and unregistering them first).

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
zoneid	String	yes	none	ID of execution zone

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceids	List of Strings	no	none	IDs of services to unregister in selected execution zone

Response Output Data

None

GET /1.0/zones

Returns a (filtered) list of all registered zones in this domain

Request Query Parameters

Parameter name	Type	Mandatory	Default value	Description
zoneids	List of Strings	no	none	List of zone IDs

Response Output Data

List of ZoneEndpoints

Notes

- Additional filter parameters such as feature and/or regional in type could be added as well

POST /1.0/zones

Register a new zone in this domain (only can be done by domain admin and authorized zone admins).

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
zoneid	String	yes	none	zone ID
endpoint	String	yes	none	zone Endpoint

Response Output Data

None

Notes

- Additional filter parameters such as feature and/or regional in type could be added as well

DELETE /1.0/zones

Remove all or a subset of zones from this domain (e.g., in the context of a zone admin).

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
zoneids	List of Strings	no	none	List of zone IDs

Response Output Data

None

Notes

- Additional filter parameters such as feature and/or regional in type could be added as well

GET /1.0/zones/{zoneid}

Fetch detailed information of this zone (i.e., services, state, etc.).

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
zoneid	String	yes	none	ID of execution zone

Response Output Data

Zone

DELETE /1.0/zones/{zoneid}

Remove this zone from this domain (terminating all running instances first).

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
zoneid	String	yes	none	ID of execution zone

Response Output Data

None

4. Evaluator Service Interface

4.1 Object Definitions

Object 'Evaluation'

Evaluation represents the result of an evaluator service upon a particular evaluation request.

Parameter name	Type	Min Occurence	Max Occurence	Description
evaluationid	String	1	1	ID of the evaluation
score	Dictionary	1	1	Evaluation score (could be a simple number, or a more complex structure, see also the D3.x deliverables)

Object 'Analysis'

Analysis represents the result of a state 2 cost-benefit analysis coming from a corresponding evaluator service upon a particular cost-benefit analysis request from the domain orchestrator.

Parameter name	Type	Min Occurence	Max Occurence	Description
analysisid	String	1	1	ID of the analysis
ranking	List of RankedOffers	1	-	Ranking of each evaluation offer for each environment. Each ranked offer is a Dictionary, containing the offerid, envid, and a ranking Score (Dictionary)

4.2 Stage 1 Probe Management Interface

GET /1.0/evaluations

Returns a (filtered) list of all stored evaluations done by this evaluator service. Note that we do not require evaluator services to store all evaluations; some can be with relatively short expiration dates (or even not stored at all).

Request Query Parameters

Parameter name	Type	Mandatory	Default value	Description
evaluationids	List of Strings	no	none	List of evaluation IDs

Response Output Data

List of Evaluations

Notes

- Each evaluator service could either implement the stage 1 API, stage 2 API, or both.
- Additional filter parameters such as feature and/or regional in type could be added as well

POST /1.0/evaluations

Request/create a new evaluation. This will return a (cached) score. This is the main function of the evaluator service for doing the evaluations.

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	service ID
instantiationparams	Dictionary	no	none	Service instantiation parameters
deployparams	Dictionary	no	none	Service deployment parameters

Response Output Data

Evaluation

Notes

- Each evaluator service could either implement the stage 1 API, stage 2 API, or both.
- Additional filter parameters such as feature and/or regional in type could be added as well

DELETE /1.0/evaluations

Remove all stored evaluations from this evaluator service.

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
evaluationids	List of Strings	no	none	List of evaluation IDs

Response Output Data

None

Notes

- Each evaluator service could either implement the stage 1 API, stage 2 API, or both.
- Additional filter parameters such as feature and/or regional in type could be added as well

GET /1.0/evaluations/{evaluationid}

Fetch detailed information of a particular evaluation done by this evaluator service.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
evaluationid	String	no	none	evaluation ID

Response Output Data

Evaluation

Notes

- Each evaluator service could either implement the stage 1 API, stage 2 API, or both.

DELETE /1.0/evaluations/{evaluationid}

Remove this evaluation from the evaluator service (in case the evaluation was stored).

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
evaluationid	String	no	none	evaluation ID

Response Output Data

None

Notes

- Each evaluator service could either implement the stage 1 API, stage 2 API, or both.

4.3 Stage 2 Analysis Management Interface

GET /1.0/analyses/{analysisid}

Fetch detailed information of a particular stage 2 cost-benefit analysis done by this evaluator service.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
analysisid	String	no	none	analysis ID

Response Output Data

Analysis

Notes

- Each evaluator service could either implement the stage 1 API, stage 2 API, or both.

DELETE /1.0/analyses/{analysisid}

Remove this analysis from the evaluator service (in case the analysis was stored).

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
analysisid	String	no	none	analysis ID

Response Output Data

None

Notes

- Each evaluator service could either implement the stage 1 API, stage 2 API, or both.

4.4 Stage 2 Probe Management Interface

GET /1.0/analyses

Returns a (filtered) list of all stored stage 2 analyses done by this evaluator service. Note that we do not require evaluator services to store all analyses; some can be with relatively short expiration dates (or even not stored at all).

Request Query Parameters

Parameter name	Type	Mandatory	Default value	Description
analysisids	List of Strings	no	none	List of analysis IDs

Response Output Data

List of Analyses

Notes

- Each evaluator service could either implement the stage 1 API, stage 2 API, or both.
- Additional filter parameters such as feature and/or regional in type could be added as well

POST /1.0/analyses

Request/create a new stage 2 analysis. This will return a (cached) rank. This is the main function of the stage 2 evaluator service for doing the cost-benfit analyses.

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	service ID
offers	List of Offers	yes	none	Zone offers

Response Output Data

Analysis

Notes

- Each evaluator service could either implement the stage 1 API, stage 2 API, or both.
- Additional filter parameters such as feature and/or regional in type could be added as well

DELETE /1.0/analyses

Remove all stored analyses from this evaluator service.

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
analysisids	List of Strings	no	none	List of analysis IDs

Response Output Data

None

Notes

- Each evaluator service could either implement the stage 1 API, stage 2 API, or both.
- Additional filter parameters such as feature and/or regional in type could be added as well

5. NetworkMap/Cost Interface

5.1 Object Definitions

Object 'NetworkMap'

NetworkMap represents detailed information on a particular network map

Parameter name	Type	Min Occurence	Max Occurence	Description
meta	Dictionary	0	1	List of metadata information, such as vtag info: {"vtag":{"resource-id":"default-map","tag":"151a05a2b45"}}
network-map	Dictionary	1	1	Directionary of networkmap entries: {"PID1":{"ipv4":["126.20.49.1", "126.20.49.2"], "ipv6":["2001:0db8:0123:4567:89ab:cdef:1234:5678"]}

Notes

- This service allows to obtain map definition for a given map. tag element is a timestamp and changes when the map information has been changed.

Object 'CostMap'

CostMap represents detailed information on a particular cost map

Parameter name	Type	Min Occurence	Max Occurence	Description
meta	Dictionary	0	1	List of metadata information, such as vtag info, cost-type and cost-metric: {"dependent-vtags":{"resource-id":"default-map", "tag":"15148a85ebc"}, "cost-type":{"cost-mode":"numerical", "cost-metric":"routingcost"}}
cost-map	Dictionary	1	1	Directionary of costmap entries, containing the cost in some unit from one network map to others: "PID1":{"PID2":5,"PID3":7}

Notes

- This service allows getting the cost map for a given map . tag element is a timestamp and changes when map information has been changed. In a cost-map, the first PID is the source, and the next one is the destination, e.g., cost count from PID1 to PID3 is 7.

5.2 Map Interface

GET /alto/costmap

Obtain details on a network costmap

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
map	String	no	none	The name of the network map
cost-mode	String	no	none	The code mode value
cost-metric	String	no	none	The cost metric value

Response Output Data

CostMap

GET /alto/networkmap

Obtain the list of network maps

Response Output Data

List of NetworkMap Names (example: {'maps':['map1','map2','default-map']})

Notes

- This service allows to obtain all network maps defined in the ALTO Server.

GET /alto/networkmap

Obtain details on a network map

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
map	String	yes	none	The name of the network map

Response Output Data

NetworkMap

Notes

- This service allows to obtain map definition for a given map. tag element is a timestamp and changes when the map information has been changed.

6. Resolver Interface

6.1 Object Definitions

Object 'Endpoint'

Endpoint represents a structure that defines the service's locator, which is resolvable to clients that belongs to network PID defined in Endpoint's EndpointGroup. In process of service name resolution, it is the final result.
An endpoint of a service's endpoint group is defined by IP. It means, that the same IP may be used as locator's address for different endpoint group or even for different service (sid).

Parameter name	Type	Min Occurence	Max Occurence	Description
id	Integer	0	1	ID of an endpoint
address	String	1	1	IP address of a locator
weight	Integer	1	1	Weight of a locator in that network PID
hits	Integer	0	1	Represents counter of hits for this locator. It is used rather to get information, and not to set it.

Object 'NetworkRange'

NetworkRange is a structure which represents network subnet as a subset of a network PID.

Parameter name	Type	Min Occurence	Max Occurence	Description
id	Integer	0	1	ID of a network range
subnet	String	1	1	Text definition of network subnet in CIDR notation

Object 'Service'

Service represents a structure that defines the service that will be resolved into locator.

Parameter name	Type	Min Occurence	Max Occurence	Description
id	Integer	0	1	ID of a service
name	String	1	1	Name of a service (SID)
endpointGroup	Array of EndpointGroup	1	1	List of endpoint groups that belong to the service (may contain many 'EndpointGroup' objects).

Notes

- EndpointGroup list in XML is enclosed in element wrapper named .

Object 'NetworkPID'

NetworkPID represents a structure that defines set of network ranges.

Parameter name	Type	Min Occurence	Max Occurence	Description
id	Integer	0	1	Id of a network PID
name	String	1	1	Name of a network PID
range	Mixed	0	-	List of network ranges that belong to the network PID

Notes

- Range list in XML is enclosed in element wrapper named

Object 'EndpointGroup'

EndpointGroup represents a structure that defines set of service's endpoints which are resolvable to clients that belong to defined network PID.
An EndpointGroup of a service is defined by pair: sid and pid_name. It means, that the same IP may be used as locator's address for different network PID

Parameter name	Type	Min Occurence	Max Occurence	Description
id	Integer	0	1	ID of an endpoint
pidName	String	1	1	Name of a network PID that the locator belongs to
endpoint	Array of Endpoints	1	1	List of Endpoints that belongs to the EndpointGroup (may contain many 'Endpoint' objects)

Notes

- Network PID with the name 'pidName' has to be defined earlier. If not, operation will fail.
- Endpoint list in XML is enclosed in element wrapper named .

Object 'Location'

Location represents a final result of a service name resolution process.

Parameter name	Type	Min Occurence	Max Occurence	Description
request_id	Integer	0	1	ID of a request (if any)
service	String	1	1	Name of a service to resolve
ipv4address	String	0	1	IPv4 address of a locator
ipv6address	String	0	1	IPv6 address of a locator
debuginfo	String	0	1	Debug info from service resolving process. It has to be explicitly requested.

Object 'Response'

Response is a structure which epresents syntax of response for every single request

Parameter name	Type	Min Occurence	Max Occurence	Description
status	Integer	1	1	Status of an operation: 0=OK, non 0 = ERROR
message	String	1	1	Text description of the operation status
data	Mixed	0	1	The real data requested in web-service call

Notes

- Currently 'data' structure always contains array of objects, even if called web-service does operation on single object (then 'data' array contains only one object).

6.2 Client-side Interface

GET /FusionResolver/1.0/location/{sid}

Resolve service name into its locator

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
sid	String	yes	none	Name of the service to be resolved into location

Request Query Parameters

Parameter name	Type	Mandatory	Default value	Description
client	String	no	client IP address	Represents a client address, that should be used into service name resolving process instead of real client (remote) address.
type	String	no	client IP type	Defines types of location IPs to be resolved. Accepted values are: ipv4, ipv6, all. If omitted, or non-accepted value, the type will be calculated from client address type (see client parameter).
debug	Boolean	no	false	If set to true, debug info from resolving process is included into response data.

Response Output Data

Location

6.3 Mapping Interface

GET /FusionResolver/1.0/services

Get the information about all services registered in the resolver

Request Query Parameters

Parameter name	Type	Mandatory	Default value	Description
detail	Boolean	no	false	If set to true, list of service's locators (endpoints) is included into response output data

Response Output Data

List of Services

Notes

- Parameters to filter out the services list will be implemented later.

GET /FusionResolver/1.0/ping

Check FUSION resolver availability

Response Output Data

None

GET /FusionResolver/1.0/networks

Get information about all networks registered in resolver

Request Query Parameters

Parameter name	Type	Mandatory	Default value	Description
detail	Boolean	no	false	If set to true, list of PID's subnets is included into response output data
cidrFilter	Boolean	no	false	If set to true, network list will be filtered to match cidr value
cidr	String	no	client IP	Parameter represents IP address or subnet to filter out network list

Response Output Data

List of NetworkPID

GET /FusionResolver/1.0/service/{sid}

Manage endpoints (locators) of the service (including their weights, policies, etc.)

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
sid	String	yes	none	The name of the service

Response Output Data

Service

Notes

- Names of the service in URI and Service structure must be equal. If not, operation will fail.
- Parameters to filter out the service's endpoing list will be implemented later.

POST /FusionResolver/1.0/service/{sid}

Manage endpoints (locators) of the service (including their weights, policies, etc.)

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
sid	String	yes	none	The name of the service

Response Output Data

Service

Notes

- Names of the service in URI and Service structure must be equal. If not, operation will fail.
- Parameters to filter out the service's endpoing list will be implemented later.

DELETE /FusionResolver/1.0/service/{sid}

Manage endpoints (locators) of the service (including their weights, policies, etc.)

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
sid	String	yes	none	The name of the service

Response Output Data

None

Notes

- Names of the service in URI and Service structure must be equal. If not, operation will fail.
- Parameters to filter out the service's endpoing list will be implemented later.

GET /FusionResolver/1.0/network/{pid_name}

Manage data of a requested network PID

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
pid_name	String	yes	none	The name of the network PID

Response Output Data

NetworkPID

Notes

- Names of PIDs in Uri and NetworkPID structure must be equal. If not, operation will fail.

POST /FusionResolver/1.0/network/{pid_name}

Manage data of a requested network PID

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
pid_name	String	yes	none	The name of the network PID

Response Output Data

NetworkPID

Notes

- Names of PIDs in Uri and NetworkPID structure must be equal. If not, operation will fail.

DELETE /FusionResolver/1.0/network/{pid_name}

Manage data of a requested network PID

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
pid_name	String	yes	none	The name of the network PID

Response Output Data

None

Notes

- Names of PIDs in Uri and NetworkPID structure must be equal. If not, operation will fail.

7. Zone Manager Interface

7.1 Object Definitions

Object 'Offer'

Offer represents a the result returned by the zone manager, wrapping the various service deployment evaluations into a particular business offer.
Note that the deployment parameters represent parameters for the various orchestration layers, whereas the instantiationparams represent service-specific configuration/customization parameters (which are opaque to FUSION)

Parameter name	Type	Min Occurence	Max Occurence	Description
offerid	String	1	1	ID of the offer
serviceid	String	1	1	ID of the service
expires	Date	0	1	Expiration date of the offer
scores	Evaluations	1	-	Evaluation results
deployparams	String	0	1	Service deployment parameters
instantiationparams	String	0	1	Service instantiation parameters

7.2 General Management Interface

GET /1.0/dca

Fetch detailed information on the DCA that is currently registered to this zone.

Response Output Data

DCA

PUT /1.0/dca

Register a DCA to this zone. In the current model, we only support a zone to be deployed on top of a single DCA layer.

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
dcaid	String	yes	none	DCA ID
endpoint	String	yes	none	DCA Endpoint
userid	String	yes	none	DCA admin user ID

Response Output Data

None

DELETE /1.0/dca

Detach and remove this DCA from this zone. This should only be possible if no instances are active anymore from this DCA.

Response Output Data

None

GET /1.0/ping

Check FUSION zone manager availability

Response Output Data

None

POST /1.0/ping

Trigger the zone to do a ping to the supplied client IP address

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
ip	IP	yes	none	client IP address

Response Output Data

Ping

GET /1.0/domain

Fetch detailed information on the domain that is currently registered to this zone.

Response Output Data

Domain

PUT /1.0/domain

Register a domain to this zone. In the current model, we only support a zone to be part of one domain.

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
domainid	String	yes	none	domain ID
endpoint	String	yes	none	domain Endpoint
userid	String	yes	none	domain orchestrator user ID

Response Output Data

None

DELETE /1.0/domain

Detach and remove this domain from this zone. This should only be possible if no services are active anymore from this domain.

Response Output Data

None

7.3 Resolver Management Interface

GET /1.0/resolvers

Returns a (filtered) list of all registered service resolvers in this zone

Request Query Parameters

Parameter name	Type	Mandatory	Default value	Description
resolverids	List of Strings	no	none	List of resolver IDs

Response Output Data

List of Resolvers

Notes

- Additional filter parameters such as feature and/or regional in type could be added as well

POST /1.0/resolvers

Register a new resolver in this zone.

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
resolverid	String	yes	none	resolver ID
endpoint	String	yes	none	resolver Endpoint
userid	String	yes	none	resolver user ID

Response Output Data

None

Notes

- Additional filter parameters such as feature and/or regional in type could be added as well

DELETE /1.0/resolvers

Remove all or a subset of resolvers from this zone.

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
resolverids	List of Strings	no	none	List of resolver IDs

Response Output Data

None

Notes

- Additional filter parameters such as feature and/or regional in type could be added as well

GET /1.0/resolvers/{resolverid}

Fetch detailed information of this resolver.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
resolverid	String	yes	none	ID of FUSION resolver

Response Output Data

Resolver

DELETE /1.0/resolvers/{resolverid}

Remove this resolver from this zone.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
resolverid	String	yes	none	ID of FUSION resolver

Response Output Data

None

7.4 Service Evaluator Offer Management Interface

GET /1.0/services/{serviceid}/offers

Returns a (filtered) list of all stored evaluation offers for this specific service in this zone.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service

Response Output Data

List of ServiceOffers

POST /1.0/services/{serviceid}/offers

Create a new service evaluation offer for this service in this zone: this is the service evaluation API for the domain to make an evaluation.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
instantiationparams	Dictionary	no	none	Service instantiation parameters
deployparams	Dictionary	no	none	Service deployment parameters
timeout	Time	no	none	Optional deadline for returning service evaluation offers

Response Output Data

ServiceOffer

DELETE /1.0/services/{serviceid}/offers

Remove all stored evaluation offers for this service in this zone (i.e., flush the cache).

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service

Response Output Data

None

GET /1.0/services/{serviceid}/offers

Returns a (filtered) list of all stored evaluation offers for this specific service in this zone.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service

Response Output Data

List of ServiceOffers

POST /1.0/services/{serviceid}/offers

Create a new service evaluation offer for this service in this zone: this is the service evaluation API for the domain to make an evaluation.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
instantiationparams	Dictionary	no	none	Service instantiation parameters
deployparams	Dictionary	no	none	Service deployment parameters
timeout	Time	no	none	Optional deadline for returning service evaluation offers

Response Output Data

None

DELETE /1.0/services/{serviceid}/offers

Remove all stored evaluation offers for this service in this zone (i.e., flush the cache).

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service

Response Output Data

None

GET /1.0/offers

Returns a (filtered) list of all stored evaluation offers of all services in this zone.

Request Query Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceids	List of Strings	no	none	List of service IDs

Response Output Data

List of ServiceOffers

POST /1.0/offers

Create a new service evaluation offer for a service in this zone. The serviceID must be provided in the body of the request.

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service
instantiationparams	Dictionary	no	none	Service instantiation parameters
deployparams	Dictionary	no	none	Service deployment parameters
timeout	Time	no	none	Optional deadline for returning service evaluation offers

Response Output Data

ServiceOffer

DELETE /1.0/offers

Remove all stored evaluation offers for all of a subset of services in this zone (i.e., flush the cache).

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceids	List of Strings	no	none	List of service IDs

Response Output Data

None

GET /1.0/offers/{offerid}

Fetch detailed information of a particular offer.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
offerid	String	yes	none	offer ID

Response Output Data

ServiceOffer

DELETE /1.0/offers/{offerid}

Remove this offer from this zone.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
offerid	String	yes	none	offer ID

Response Output Data

None

7.5 Service Instance Management Interface

GET /1.0/instances

Returns a (filtered) list of all (composite) service instances in this zone, along with their current state and session slot information.

Request Query Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceids	List of Strings	no	none	List of service IDs
instanceids	List of Strings	no	none	List of instance IDs

Response Output Data

List of Instances

POST /1.0/instances

Explicitly create a new service instance in this zone. Typically, this also immediately starts the instance, but this could also simply be a provisioning or creation request.

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of the service
instantiationparams	Dictionary	no	none	Service instantiation parameters
deployparams	Dictionary	no	none	Service deployment parameters

Response Output Data

List of instanceids

DELETE /1.0/instances

Remove all or a subset of all service instances from this zone (e.g., in the context of a service provider).

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceids	List of Strings	no	none	List of service IDs
instanceids	List of Strings	no	none	List of instance IDs

Response Output Data

None

GET /1.0/instances/{instanceid}/ports

Returns a list of all registered and mapped service instance ports

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
instanceid	String	yes	none	ID of instance

Response Output Data

List of ServicePorts

GET /1.0/instances/{instanceid}

Fetch detailed information of this high-level service instance (i.e., session slots, runtime state, etc.).

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
instanceid	String	yes	none	ID of instance

Response Output Data

Instance

DELETE /1.0/instances/{instanceid}

Terminate and remove this service instance from this zone.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
instanceid	String	yes	none	ID of instance

Response Output Data

None

GET /1.0/instances/{instanceid}/slots

Fetch detailed information of the number of available and used session slots of this service instance.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
instanceid	String	yes	none	ID of instance

Response Output Data

SessionSlots

PUT /1.0/instances/{instanceid}/slots

Change the number of available session slots of this service instance.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
instanceid	String	yes	none	ID of instance

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
slots	SessionSlots	yes	none	New Session slots state

Response Output Data

None

DELETE /1.0/instances/{instanceid}/slots

Remove all active slots from this instance (terminate this instance).

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
instanceid	String	yes	none	ID of instance

Response Output Data

None

GET /1.0/instances/{instanceid}/state

Fetch state information of this service instance in this zone.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
instanceid	String	yes	none	ID of instance

Response Output Data

InstanceState

PUT /1.0/instances/{instanceid}/state

Change lifecycle state of the service instance (e.g., provision, deploy, terminate).

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
instanceid	String	yes	none	ID of instance

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
state	String	yes	none	New instance state

Response Output Data

None

DELETE /1.0/instances/{instanceid}/state

Remove this service instance from the zone (possibly terminating the instance first).

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
instanceid	String	yes	none	ID of instance

Response Output Data

None

GET /1.0/instances/{instanceid}/ports/{port}

Returns the port mapping information of the specific service port type for this instance

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
instanceid	String	yes	none	ID of instance
port	String	yes	none	service port type

Response Output Data

ServicePort

7.6 Service Management Interface

GET /1.0/services/{serviceid}/ports/{port}

Returns the port mapping information of the specific service port type

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service
port	String	yes	none	service port type

Response Output Data

ServicePort

GET /1.0/services/{serviceid}/slots

Fetch detailed information of the number of available and used session slots of this service in this zone.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service

Response Output Data

SessionSlots

PUT /1.0/services/{serviceid}/slots

Change the number of available session slots of this service in this zone.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
slots	SessionSlots	yes	none	New Session slots state

Response Output Data

List of instanceids

DELETE /1.0/services/{serviceid}/slots

Remove all active slots from this service in this zone (i.e., terminate service instances).

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
zoneids	List of Strings	no	none	Optional list of zoneids to terminate this service from

Response Output Data

None

GET /1.0/services/{serviceid}/ports

Returns a list of all registered and mapped service (instance) ports

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service

Response Output Data

List of ServicePorts

GET /1.0/services/{serviceid}/instances

Returns a (filtered) list of all (composite) service instances of this service in this zone, along with their current state and session slot information.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service

Response Output Data

List of ServiceInstances

POST /1.0/services/{serviceid}/instances

Explicitly create a new instance of this specific service in this zone. Typically, this also immediately starts the instance, but this could also simply be a provisioning or creation request.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
instantiationparams	Dictionary	no	none	Service instantiation parameters
deployparams	Dictionary	no	none	Service deployment parameters

Response Output Data

List of instanceids

DELETE /1.0/services/{serviceid}/instances

Remove all instances of this service from this zone.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service

Response Output Data

None

GET /1.0/services

Returns a (filtered) list of all registered services in this zone, along with their current state and session slot information.

Request Query Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceids	List of Strings	no	none	List of service IDs

Response Output Data

List of Services

POST /1.0/services

Register a new service in this zone.

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
manifest	ServiceManifest	yes	none	partial service manifest in some format (JSON, XML, YAML, ...)

Response Output Data

None

DELETE /1.0/services

Remove all (or a subset of) services from this zone (e.g., in the context of a service provider).

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceids	List of Strings	no	none	List of service IDs

Response Output Data

None

GET /1.0/services/{serviceid}/state

Fetch state information of this service in this zone.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service

Response Output Data

ServiceState

PUT /1.0/services/{serviceid}/state

Change service lifecycle state (e.g., register, provision, deploy, terminate) in this zone.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
state	String	yes	none	New service state
deployparams	Dictionary	no	none	Service deployment parameters
instantiationparams	Dictionary	no	none	Service instantiation parameters
terminateparams	Dictionary	no	none	Service termination parameters

Response Output Data

None

DELETE /1.0/services/{serviceid}/state

Remove this service from this zone (possibly terminating all running instances first).

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service

Response Output Data

None

GET /1.0/services/{serviceid}

Fetch detailed information of this service in this zone (i.e., slots, runtime state, etc.).

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service

Response Output Data

Service

DELETE /1.0/services/{serviceid}

Remove this service from this zone (possibly terminating all running instances first).

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
serviceid	String	yes	none	ID of service

Response Output Data

None

7.7 User Management Interface

GET /1.0/users/{userid}

Fetch detailed information of this user.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
userid	String	yes	none	ID of user

Response Output Data

User

DELETE /1.0/users/{userid}

Remove this user from this zone. Users can only be removed when all their state is removed first.

Request Path Parameters

Parameter name	Type	Mandatory	Default value	Description
userid	String	yes	none	ID of user

Response Output Data

None

GET /1.0/users

Returns a (filtered) list of all registered users in this zone.

Request Query Parameters

Parameter name	Type	Mandatory	Default value	Description
userids	List of Strings	no	none	List of user IDs

Response Output Data

List of Users

POST /1.0/users

Register a new user in this zone.

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
userid	String	yes	none	User ID
user	String	yes	none	User name
roles	List of Strings	no	none	User roles
password	String	yes	none	User password

Response Output Data

None

DELETE /1.0/users

Remove all (or a subset of) users from this zone.

Request Body Parameters

Parameter name	Type	Mandatory	Default value	Description
userids	List of Strings	no	none	List of user IDs

Response Output Data

None