Interoperable Federated Cloud Networking

Eduardo Huedo, Rubén S. Montero, Rafael Moreno and Ignacio M. Llorente · Complutense University
Anna Levin · IBM Research - Haifa
Philippe Massonet · Centre of Excellence in Information and Communication Technologies

The BEACON framework enables the provision of federated cloud infrastructures, with special emphasis on inter-cloud networking and security issues, to support the automated deployment of applications and services across different clouds and datacenters. BEACON is distributed as open source (see http://github.com/BeaconFramework) and some enhancements are being contributed to the OpenNebula and OpenStack cloud management platforms.

With the growing number of infrastructure cloud services becoming available there are many benefits to interconnecting several cloud services in the form of a federation. There is a strong industry demand for automated solutions to federate cloud network resources, and to derive the integrated management cloud layer that enables an efficient and secure deployment of resources and services independent of their location across distributed infrastructures. From big companies and large cloud providers interested in unifying and consolidating multiple data centers or cloud sites to SMEs building hybrid cloud configurations, federated cloud networking is needed to support the automated deployment of applications across different clouds and data centers.

Different cloud federation types such as cloud bursting, cloud brokering or cloud peering have been proposed to provide the necessary mechanisms for sharing compute, storage and networking resources. In hybrid clouds, for example, a private cloud is connected with one or more public clouds so that resources from those clouds can be used for deploying and running Virtual Machines (VMs).

Creating and managing cloud federations means that the main components of cloud platforms must interact with each other: the Service, Cloud and Network Managers. The Service Manager is responsible for the instantiation of the service application by requesting the creation and configuration of VMs for each service component included in the service definition, using the cloud interfaces exposed by the Cloud Manager. The Cloud Manager is responsible for the placement of VMs into hosts. The Network Manager is responsible for allocating network resources to manage the federated cloud virtual network and overlay networks across geographically dispersed sites.

This paper describes the BEACON framework for federated cloud networking, and the associated reference architecture for the federated cloud networking. BEACON provides many advanced features for the federated cloud networks such as automated high availability, datacenter location-aware elasticity, automated Service Function Chaining (SFC), and security across clouds.

The BEACON framework is being applied to real applications for airline scheduling[1]. Because of the nature of their business functional requirements, their system services are split into write and read intensive modules which can be located in different geographical locations. For read modules, the latency is critical, so these are set to the closest datacenter to minimize the network latency. For write modules, high availability is required to ensure data integrity, so they are replicated on multiple datacenters. The main benefit for the application is the improvement in user experience by minimizing latency and deploying VMs close to the customer.

Architecture for Federated Cloud Networking

The proposed federation network model addresses the challenge of federating clouds based on different cloud management platforms and network technologies (i.e. Software Defined Networking, SDN)[2]. Moreover, it can be used in different cloud federation architectures such as peer, hybrid, and brokered cloud federation. Figure 1 depicts a high level view of the components necessary to carry out the network federation, and their interactions.

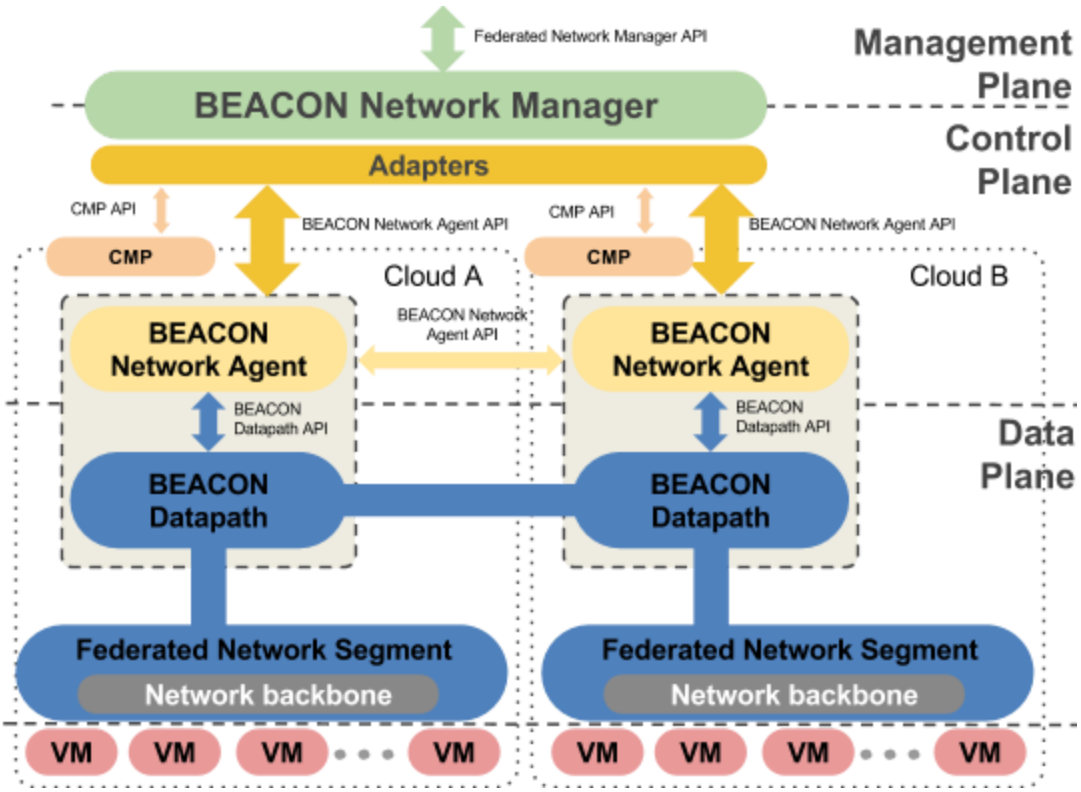

Figure 1. BEACON architecture for cloud network federation.

The BEACON Network Manager is the software component that allows to build a federated cloud network by aggregating two or more Federated Network Segments, which are virtual networks within a cloud infrastructure, each sustained by a physical network backbone. The BEACON Network Manager provides a uniform interface for users in order to set up a virtual federated network in a transparent way, independently from the underlying clouds. In order to do this, it features an API to allow for federated network definitions, and uses adaptors to talk to the Cloud Management Platforms.

The BEACON Network Agent drives the control plane of a federated network. It informs other Network Agents about the known network segments of its domain, and instructs the BEACON Datapath. The Network Agent is present with a well-known endpoint in the cloud infrastructure. The Network Agent provides a REST API to communicate with the BEACON Network Manager and other peer Network Agents. The communication with the BEACON Datapath is based on OpenFlow and the communication with the local SDN controller depends on the chosen SDN technology (e.g. Open Virtual Network or Open vSwitch, see http://openvswitch.org).

The BEACON Datapath defines the data plane as instructed by the Network Agent. The Datapath encapsulates traffic between the different Federated Network Segments and provides the needed mapping to interconnect Federated Network Segments. It can be an instance of different implementations, depending on the type of federated network (L2 or L3) to be built, the nature of the tunnel (GRE, Geneve or any other tunnelling protocol) and any needed adaptation. The configuration for the data plane relies on a full mesh, where all the Datapaths communicate with each other when they belong to the same federated network.

Service Definition and Orchestration

A service is typically a multi-tiered application composed of interconnected VMs with deployment dependencies between them. The BEACON framework provides the cloud user with extended service definition capabilities. For example, service definition is being extended in order to specify required automated high availability across datacenters, datacenter location-aware elasticity (see the related sidebar), SFC (see the related sidebar), and security.

Service definition is based on existing languages, like Heat Orchestration Templates, for OpenStack-based clouds, or OneFlow Service Templates, for OpenNebula-based clouds.

OpenNebula uses cloud bursting drivers to model a remote cloud as any other host (albeit of potentially a much bigger capacity)[3]. Therefore, the scheduler can place VMs in the external cloud as it would do in any other local host. The scheduler transparently chooses if the VM is executed locally or remotely, depending on the requirement and rank expressions defined for the VM. This is also transparent to the Service Manager, OneFlow, which implements policies for location-aware elasticity or automated high availability.

For cloud platforms like OpenStack, that do not support the hybrid model, the BEACON framework includes a multi-domain orchestrator to federate several clouds. It provides:

- Service placement policies that include cloud location constraints.
- Location-aware elasticity rules.
- Federated cloud monitoring.

Cloud location constraints can be specified in the service placement policies to indicate in which cloud each VM must be deployed. Location-aware elasticity policies work in conjunction with the above feature. So if a multitude of people from east Europe access the application suddenly, their requests will not be proxied to the VMs in west Europe because more VMs will be automatically allocated where they are needed. Thus minimizing latency. The orchestrator retrieves monitoring information of the federated cloud network in order to calculate aggregate metrics for managing the federation.

The Service Manifest is decomposed and stored in an internal database. From this internal representation of the Service Manifest, the database can be queried for fragments related to a specific cloud of the federation and a cloud specific service manifest can be created. This cloud-specific Service Manifest can be submitted to the Service Manager of the target cloud platform.

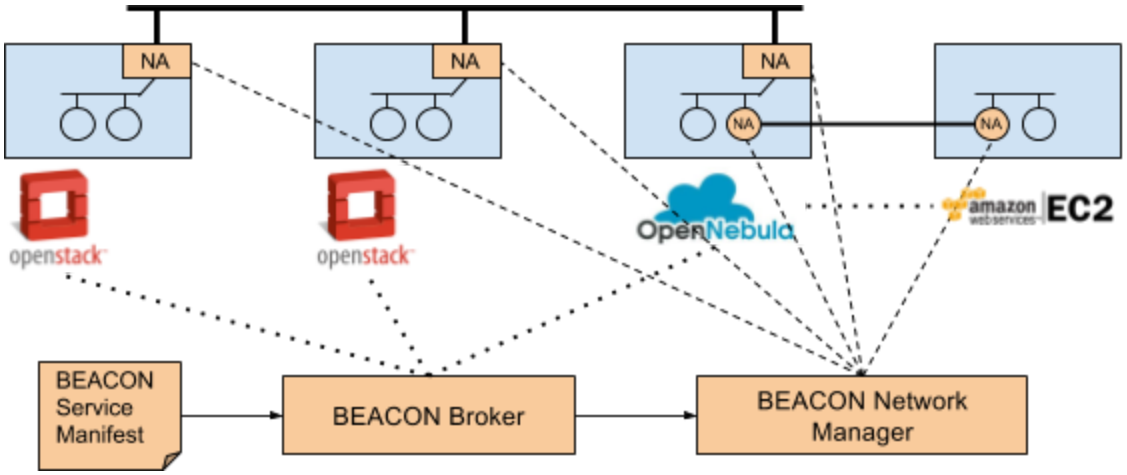

Figure 2. BEACON deployment.

Figure 2 shows a possible deployment of the BEACON framework. In this case, the BEACON Broker orchestrates (dotted lines) three private clouds based on different cloud management technologies. To access public clouds, like Amazon EC2, the OpenNebula cloud bursting drivers are used (also, dotted lines). To create the federated network (thick solid lines), private clouds provide a BEACON Network Agent, which are managed (dashed lines) by the BEACON Network Manager. However, the public cloud does not provide it, so it is deployed as a VM, extending the virtual overlay network to the remote cloud and connecting it to the private cloud, which is already connected to the federated network.

Multi-cloud Security

Federated cloud networks enable VMs to communicate across different clouds as if they were on the same network. Communicating across different clouds means that levels of security can vary due to different local cloud security policies. Securing communications by combining the Layer 2 Tunneling Protocol (L2TP) with IPsec (L2TP/IPsec), for example, is not sufficient because it does not address all security concerns. Also, if only specific application level protocols are authorized, such as HTTPS, then complementary Deep Packet Inspection (DPI) analysis must be performed. Or if there is a risk of intrusion, then complementary intrusion detection analysis must be performed. Furthermore, this approach secures all overlay networks in the same way, and does not allow to have security policies that are specific to an overlay network.

Our approach to improving security relies on enforcing a global security policy on the federated cloud network using SFC across the different clouds[4]. With this approach, if the overlay network is application specific, then the global security policy will also be application specific. In our approach the global network security policy is defined in a Service Manifest. It is then parsed by the BEACON Network Manager component and is transformed into VNF and SFC

configurations, e.g. first DPI then intrusion detection, for each of the clouds in the federation. Once the VNF and SFC are configured in all federation clouds inbound and outbound flows are routed through an SFC of VNF on each side of the federated network. Monitoring data from the security VNF can be aggregated in the BEACON Network Manager component for further analysis.

The main benefit of this approach is that the deployment, configuration and chaining of the security VNF may be automated. Having network administrators perform this manually across the different clouds would take time and be error prone if the federated cloud network is large and/or the network topology or the security policy change often.

BEACON addresses the challenge of designing and developing an interoperable framework that can be integrated into different cloud middleware and yet provide support for virtual networking and security on different federation approaches.

Acknowledgements
This work has been supported by the BEACON project, grant agreement number 644048, funded by the European Union Horizon 2020 Programme under topic ICT-07-2014.

References

1. A. Debski, B. Szczepanik, M. Malawski, S. Spahr and D. Muthig, *"In Search for a Scalable & Reactive Architecture of a Cloud Application: CQRS and Event Sourcing Case Study"*, IEEE Software, in press.
2. A. Levin, K. Barabash, Y. Ben-Itzhak, S. Guenender, and L. Schour, *"Networking architecture for seamless cloud interoperability"*, *IEEE International Conference on Cloud Computing*, 2015.
3. B. Sotomayor, R. S. Montero, I. M. Llorente, and I. Foster, *"Virtual infrastructure management in private and hybrid clouds"*, *IEEE Internet Computing* 13 (5), 14-22, 2009.
4. P. Massonet, A. Levin, M. Villari, S. Dupont, and A. Michot, *"Enforcement of Global Security Policies in Federated Cloud Networks with Virtual Network Functions"*, *15th International Symposium on Network Computing and Applications*, 2016.

Datacenter Location-Aware Elasticity

The possibility to allocate resources in multiple clouds opens up avenues to a new type of elasticity based on where the new capacity is allocated, that is, the location-aware elasticity. In this scenario, applications are able to allocate functional components close to the client for improving performance, or due to security or jurisdictional constraints. In general, these components should be able to communicate through public networks, and so being able to tolerate long latencies.

As a paradigmatic example, we can consider a content distribution system able to allocate access servers on specific geographic locations in order to address increasing amount of client requests in that area (e.g. country specific marketing campaigns or release of new media).

Figure A shows a block diagram of the distribution system. Initially the system is deployed in a private cloud, where the original content is developed and stored. At a given time the load of the application is expected to increase in a specific geographical area. This process fires the allocation of a new component in a public cloud located closer to the clients, so effectively reducing the latency of the application and improving the user experience.

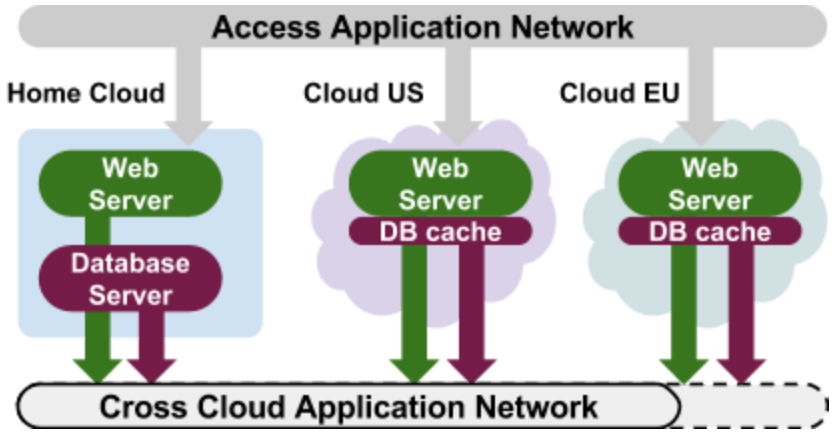

Figure A. Content distribution system across Clouds.

The contents of the application are basically static and distributed through standard web mechanisms; so new web servers can be allocated in any location provided they have access to the content. Typically, to improve performance in the access to the content database a distributed memory object caching system should be considered. Note that the redirection to the best server can be implemented through custom DNS records or with the GeoDNS extension.

Web servers, DB caches and DB servers communicate through a secure virtual LAN. This LAN dynamically grows to multiple locations and additionally control and monitor traffic for the application components.

Automated Service Function Chaining across Clouds

The major benefit of federated cloud networks is the flexibility they provide to customize and manage the federated network according to the specific application. An important part of application deployment is the specification of the application policy, which includes network service requirements. The specified requirements must be independent of the underlying infrastructure and define, for the virtualized environment, Virtual Network Functions (VNF) to be used as well as their topology, in terms of Service Function Chains (SFC).

Different VNFs may list load balancers, deep packet inspection, encryption, decryption, intrusion detection or firewalls. The deployment and configuration of SFCs composed from the required VNFs across different clouds is managed by a BEACON Network Manager. This component is responsible for enforcing a coherent global network security policy across the network federation. Figure B shows a federated cloud network distributed across two clouds.

The global federated network policy is defined in a service manifest, then the VNF and SFC configurations are derived from the global policy and specified for each cloud. Finally, the SFCs are deployed across clouds.

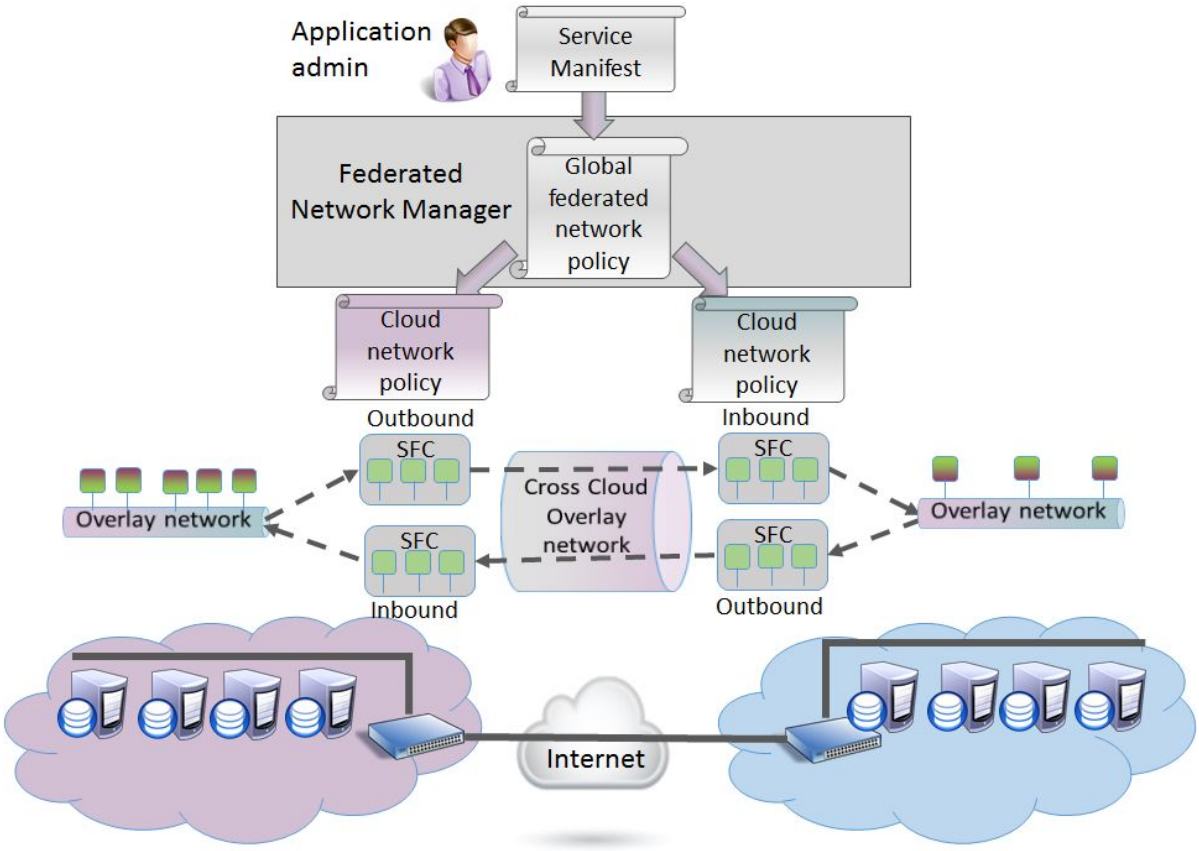

Figure B. Federated cloud network.

The deployment and configuration of the SFCs in federated networks are extremely complex and error prone tasks that require coordinating SFCs across heterogeneous networks empowered by different virtualization technologies. Therefore, SFC composition and deployment must be automated.

There are standardization efforts on SFC automation. IETF SFC working group[1] developed a framework for SFC operation and administration. The Open Network Foundation proposed an SFC architecture[2], which includes an SFC network controller responsible for setting up the service chaining; traffic flow Classifier and service function Forwarder that forwards packets through the VNFs defined in the policy table.

The proposed general architecture can be applied to any network. However, in order to implement this architecture in federated heterogeneous networks, there is still a need to coordinate SFC network controllers and classifiers, so they will speak the same language and deploy and control NFVs in an efficient way. This additional orchestration is done by the BEACON Network Manager.

When these concepts are applied to a real application, the application administrator defines in a Service Manifest the requirements for load balancing and firewall use. Then, the BEACON Network Manager automatically translates these requirements to the specific cloud network policies, which trigger deployment and configuration of the inbound and outbound SFCs in each cloud separately. The application traffic is then forwarded automatically through the deployed SFCs.

References

1. Service Function Chaining (sfc) working group, http://datatracker.ietf.org/wg/sfc.
2. *L4-L7 Service Function Chaining Solution Architecture*. ONF TS-027. Version 1.0, 2015.