# FaaS Proposal


- Name
- Repo Name
- Description
- Scope
- Resources Committed (developers committed to working)
- Initial Committers
- Vendor Neutral
- Meets Board Policy (including IPR)


## Name

Fabric As A Service

## Repo Name

faas

## Description

Currently network applications and network administrators mostly rely on lower level interfaces such as CLI, SNMP, OVSDB, NetConf or Openflow to directly configure individual device for network service provisioning. In general, those interfaces are

```
*  Technology oriented, not application orientied.
*  Vendor specific
*  Individual device oriented, not network oriented.
*  Not declarative, complicated and Procedure oriented.
```

To address the gap between application needs and network interface, there are a few application centric language proposed in ODL such as GBP, NIC, NEMO etc... trying to replace traditional southbound interface to application. Those langugages are top-down abstraction and modeling the application requirements in a more application oritend way. After being involved with GBP development for a while, we feel the top down model still has a quite gap between the model and the underneath network since the existing interfaces to network devices are

```
*  Lack of abstraction of network which make it very hard to map high level abstractions such as GBP, Intent or
any application centric north bound APIs
   to physical network. Often the applications built with these low level interfaces are coupled tightly with
underneath technology and make the application's architecture
   monolithic, error prone and hard to maintain.
```

We think a bottom-up abstraction of network can simplify reduce the gap and easy to implement the application centric model. Moreover in some uses cases, an interface with network service oriented are still desired for example from network monitoring/troubleshooting perspective. That's where the Fabric as a Service comes. To further clarify the confusion between the FaaS and hight level modeling langugage such as GBP, here are some points deserved to be empahsized.

- FaaS and GBP are at different abstraction level and Faas is NOT intend to replace GBP or any other high level modeling languages, they are not conflicting each other. As stated previously, they complement each other GBP is top down and FaaS is bottom up . With those two, GBP and others application centric model can be much more easier rendered .
- GBP has potential we all agree. But in reality users have the final say who is the right language and they may want to have alternatives.
- FaaS can be a common layer underneath GBP/NIC/NEMO or any application centric API
- There are use cases FaaS could be used directly outside of ODL.

Fabric As A Service tries to provide fabric abstraction of the physical network and via these abstraction, FaaS encapsulates device/vendor/network details such as control and optimization into a bigger building block - fabric as illustrated in the diagram below. each fabric provides common network services which includes L2/L3 connectivity, QoS as well as ACL control. Instead of configuring each individual devices, applications will configure fabrics. Those services provides fabric oriented and technology and vendor agnostic APIs to make new application building simpler, easier, faster and less error prone.

blocked URL

A simple network could just be a fabric such as a VXLAN and but in reality, as in below diagram, a network usually consists of multiple fabrics such as VXLAN fabric, VLAN, Even TRILL/SPB etc... Hence, FaaS not only defines the services within a fabric, but also provides services across multiple heterogeneous fabrics.

blocked URL

blocked URL

SDN (Software defined Networking) allows users to define a logical network according to their own needs without knowing too much about the physical detail. The network they want is an infrastructure which provides the network services regarding connectivity, QOS and policy to their applications. The language to define the networking infrastructure should be simple, networking focus and declares their networking needs only and nothing else.

FaaS defines the following top level basic abstracted network primitives and operations on them to the north.

```
* logical switch
* gateway
* logical router
* Tunnel between logical switch and logical router
* ACL
```

With those primitives, a user can easily define a logical network as below.

blocked URL

Through logical network abstraction, FaaS decouples the applications from the underneath physical network technology and device specific interface, so applications can focus on its business logic requirements on network infrastructure needs only that include connectivity, policy and QOS instead of those network implementation details and trivial configuration commands.

blocked URL

Use GBP (Group based Policy) as an example, with FaaS the GBP model could be mapped into physical configuration in a more clear/simple/visible way to physical network.

blocked URL

## Scope

As in the architecture figure below, the FaaS deliverable consist of a Fabric Manager module which provides fabric provisioning services, an unified fabric service interface as well as a list of fabric objects supported and managed by the fabric manager.

blocked URL

**FaaS Interface**

- A YANG model which defines Fabric and Fabric manager and their provided services.

**Fabric Manager**

```
* provides fabric abstraction CRUD operations.
* Create virtual fabrics on Fabric
* L2 and L3 Connectivity set up across fabrics
```

**Fabric abstraction**

```
* VLAN and VXLAN based fabric abstraction
* Fabric OAM functionality including visibility of the mapping between logical and physical configuration as well
as trouble shooting tools
* Each fabric object provides following network service.
   logical switch CRUD operations. For example, create Logical Switch and Update ports on Logical Switch
           a logical switch is a layer 2 network primitive which provides L2 connectivity between physical
devices distributed on a fabric.
   gateway CRUD operations
   logical router CRUD operations. for example, create Logical Router based on a set of Logical Switches, update
logical Switches on logical Router and update logical router’s routing table
            a logical router is a logical l3 network primitive which forwards traffic among multiple logical
switches.
   ACL control and traffic forwarding between logical switches or routers.
   Diagnostics and statistics for logical switches and routers by making the mapping available between logical
network elements and physical network.
   Support both non-openflow (netconf) switches and openflow switches.
```

**Also FaaS will contribute a GBP render as part of the GBP project to showcase the FaaS capability.**

Fabric As A Service depends on, integrates with or may augment the following components of ODL existing components.

- topology manager
- inventory manager
- Service Function Chaining
- statistic manager
- USC plugins <br/?
- openflow plugins
- OVSDB plugins
- NetConf plugins
- MD SAL/clustering service

Notes that

- FaaS will cooperate with ODL SFC project to provide advanced service function chaining functionality. Fabric does not do SFC.
- Fabric will focus on L2/L3/ACL abstraction only.
- Although FaaS intend to support all high level northbound APIs in the future and to support broader range of devices. but for beryllium release, FaaS will support GBP only as northbound and OVSDB/OPENFLOW/Openvswitch as southbound

## Resources Committed (developers committed to working)

Xingjun Chu, xingjun.chu@huawei.com

Yapeng Wu, yapeng.wu@huawei.com

Henry Yu, henry.yu1@huawei.com

Khaldoon Al Zoubi, khaldoon.alzoubi@huawei.com

Guoli Yin, YinGuoli@huawei.com

leslin.dongfeng@huawei.com

songwei80@huawei.com

Alex Zhang <alexzhang@chinamobile.com>

## Initial Committers

Xingjun Chu, xingjun.chu@huawei.com, gerrit id - chuxingjun

Yapeng Wu, yapeng.wu@huawei.com, gerrit id - yapengwu

Henry Yu, henry.yu1@huawei.com, gerrit id - hyu2010

Khaldoon Al Zoubi, khaldoon.alzoubi@huawei.com, gerrit id - khal

Alex Zhang, alexzhang@chinamobile.com - alexzhang

## Vendor Neutral

Have all proprietary trademarks been removed? yes

Have all proprietary logos been removed? yes

Have all proprietary product names been removed? yes

## Meets Board Policy (including IPR)

- Existing code must be provided to Phil Robb (probb@linuxfoundation.org) for Inbound Code Review (ICR) prior to the project gaining resources to move code to OpenDaylight repositories.