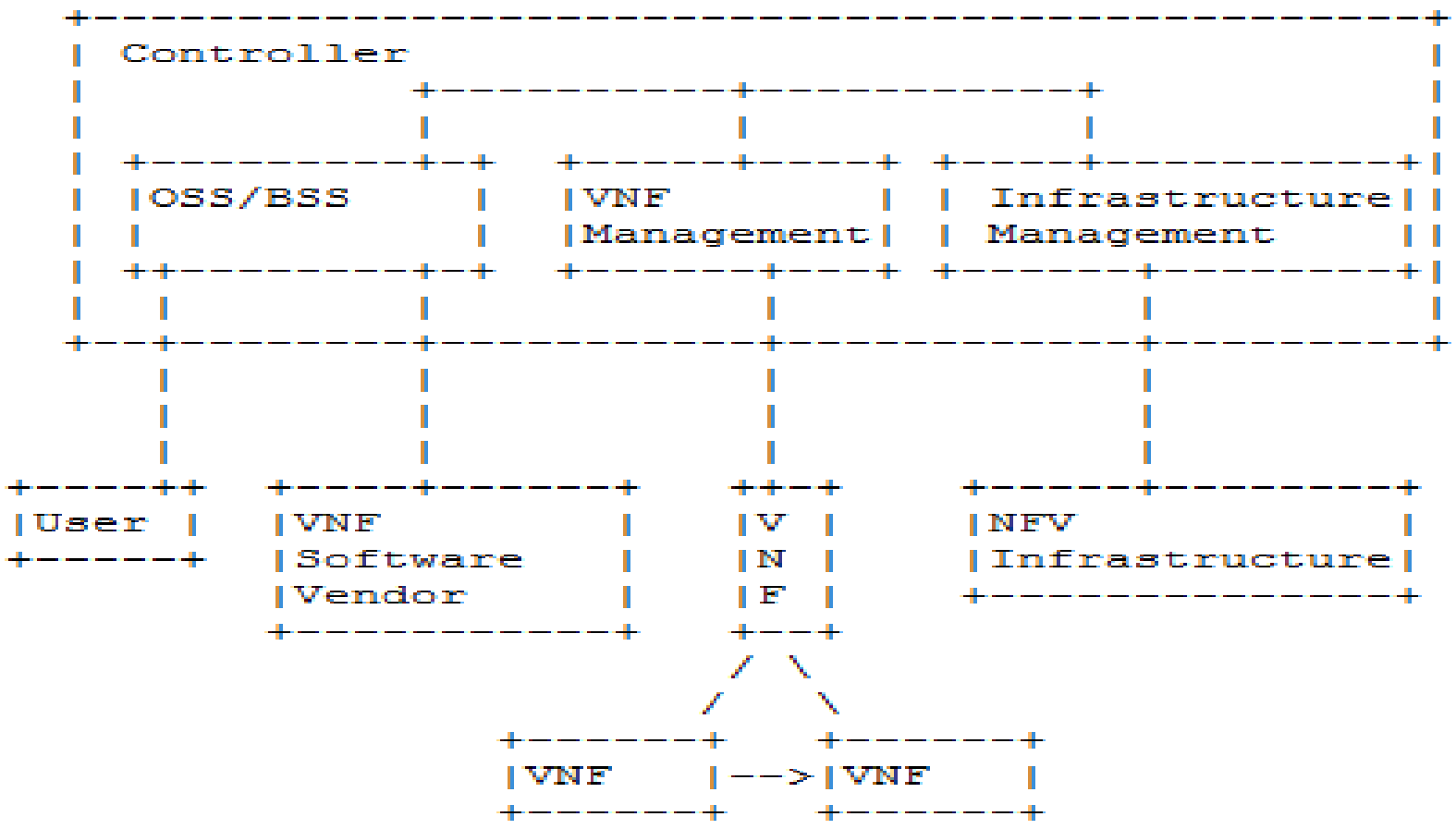
Virtual Network Function Configuration Architecture

Hong Zhou zhouhong@huawei.com
Haibin Song haibin.song@huawei.com
Qiao Fu fuqiao@chinamobile.com

NFV Configuration



Principles

- Controller is the brain
  - A user does not have to communicate directly with its multiple VNFs, but a simple controller
  - Act as a broker to retrieve the existing parameters, and configure VNFs and their connections
- Controller is agnostic of the configured service parameters
  - But is aware of how/when to apply configurations to which VNFs, or related forwarding equipments
- The key is the information and the data model
  - VNF model
  - Resource model
  - Service/forwarding graph model
  - Monitoring/reporting status

User-Controller Interface

- Lifecycle management
  - VNF installation
    - VNF name, quantity, preferred locations(e.g. data center level), components selection,
    - resource requirements, capability requirements,
    - Whether on-demand resouce allocation, and the automatic scale-out/scale-in needs resource policy which will trigger the event from the user or provider
  - VNF update, termination
  - A forwarding graph data model for service flow
- Configuration
  - A service template containing: Identify of VNF, user signature, service parameters
- Report Information from the Controller
  - Status, logging, accounting

Software Vendor-Controller Interface

- VNF descriptor from the software vendor
  - Type (options provided by controller for classification), function description, resource requirements, software environment requirement, capability per instance, pricing and etc.
  - Publish, update, off-the-shelf of a VNF
- Software packages

Controller-VNF Interface

- Lifecycle management
  - Create, delete, update
- Automatic scale-out/scale-in
  - With creating new instances or deleting existing instances
- Monitoring
  - Resource (CPU, link and etc) usage
- Coordinate with the infrastructure management module
  - Splitting traffic for load balancing (change the forwarding rules)

Controller-Infrastructure Interface

- Configure the underlying network and forwarding rules
- Lifecycle management of VMs
- There are some existing tools for it
  - Openstack, Cloudstack…

- *May leave it out of scope*
  - *Too many implementation details*

Security

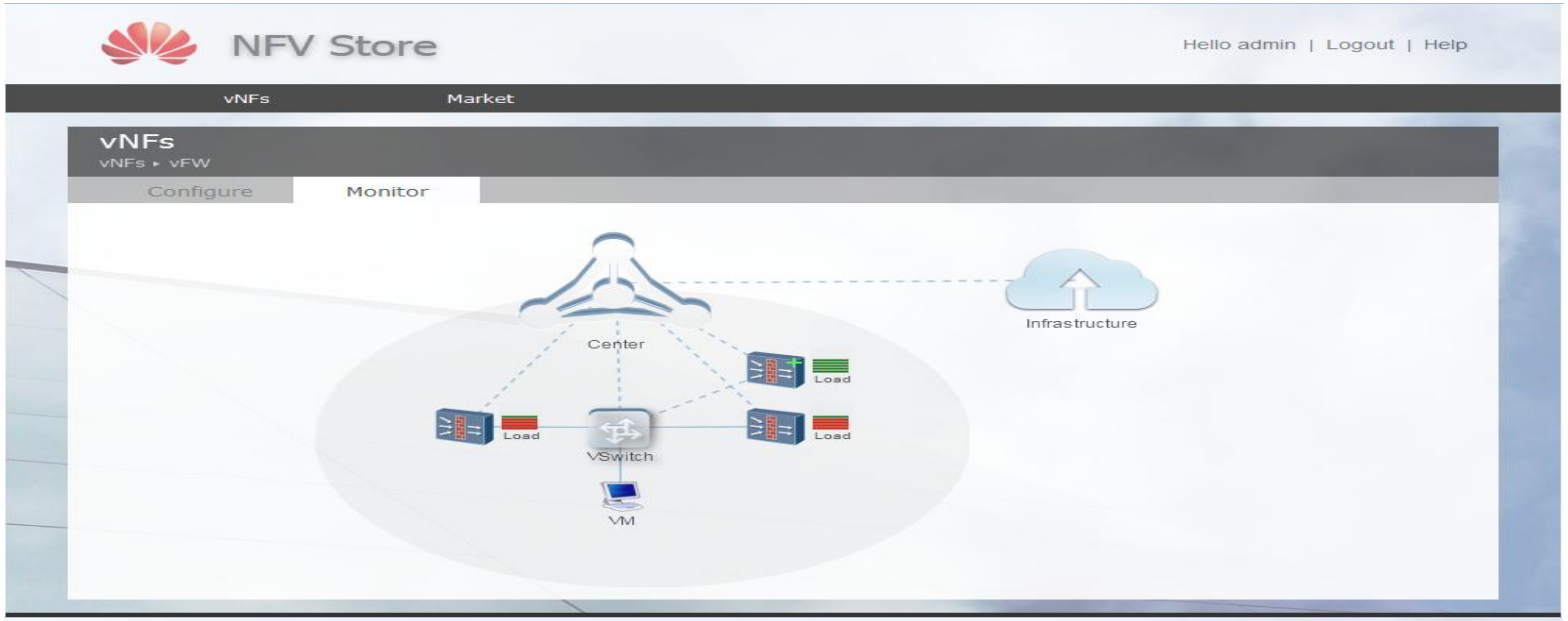
- All user controller interactions MUST be validated bi-directionally
- An encryption of messages is mandatory

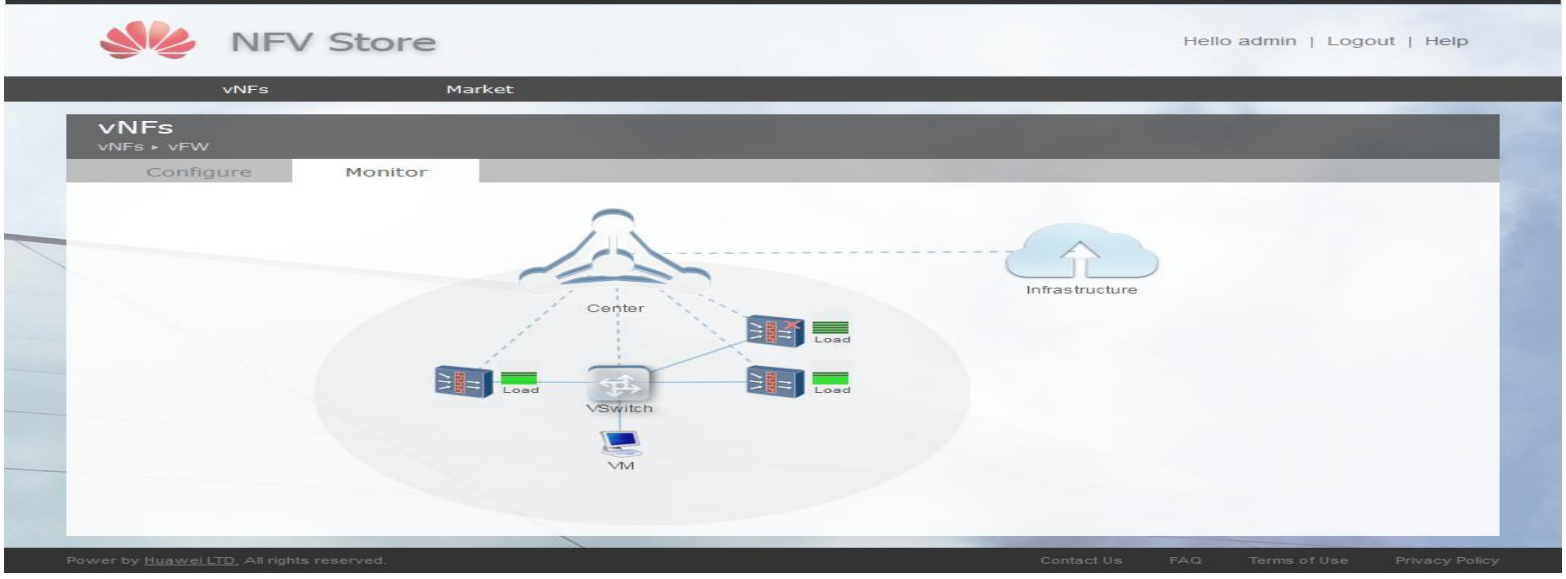
Next Step

- Gap analysis with NetConf and NetMod

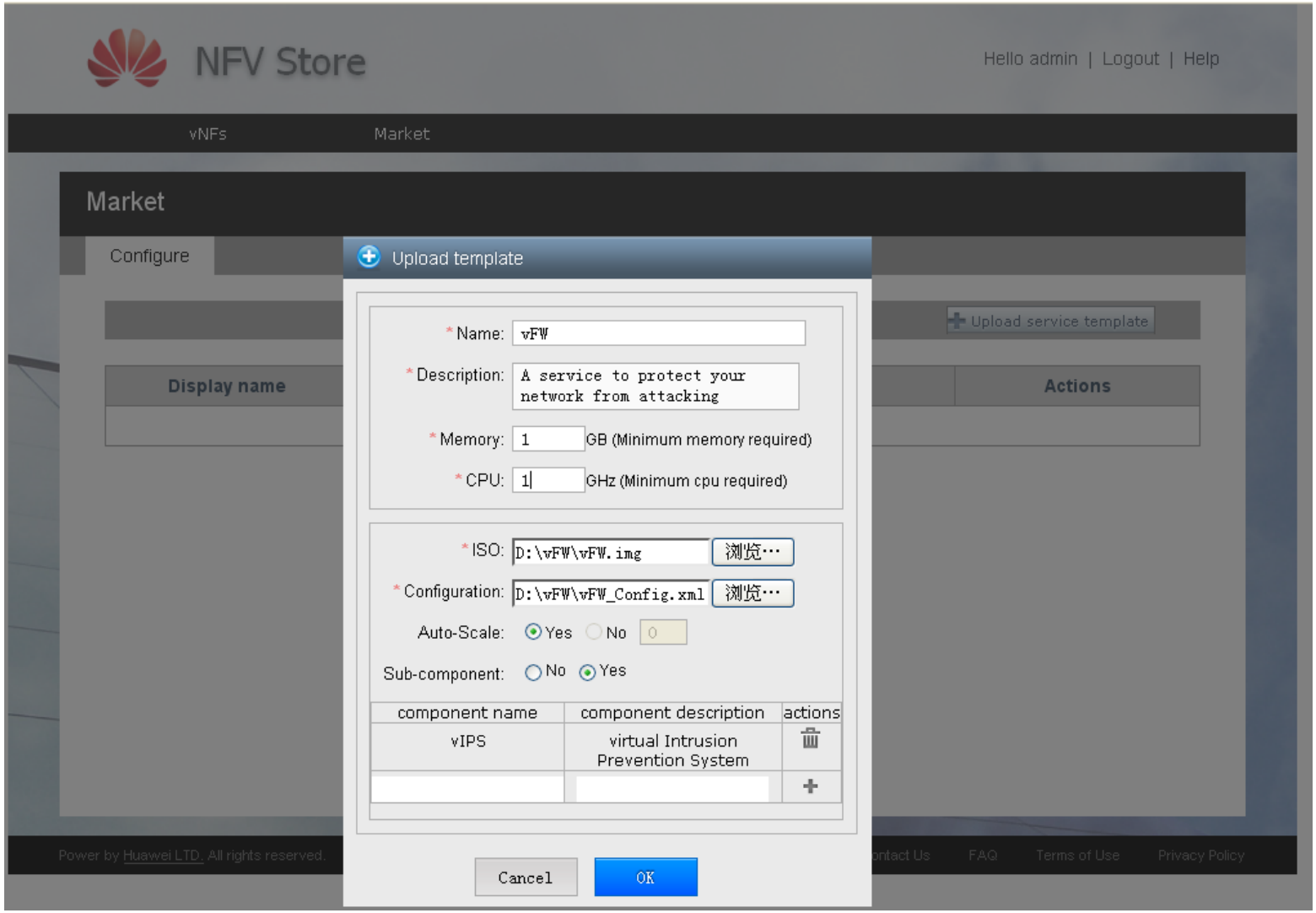
Thanks!

POC Prototype





POC Prototype



POC Prototype