Automated Resource Control
in Virtualized Network Environments

Pedro Martinez-Julia
Network Science and Convergence Device Technology Laboratory, Network System Research Institute National Institute of Information and Communications Technology (NICT)
pedro@nict.go.jp

Network Function Virtualization Research Group, Internet Research Task Force
IETF 97 Meeting, Seoul, South Korea
15 November 2016
Automated Resource [De]Allocation (I)

Virtual Network Operator (VNO)

Virtual Resource Controller
{Computing & Network}

InP ↔ VNO Interfaces

Allocation / Deallocation Requests

State Notifications

Controller
{Network}

Controller
{Computing}

Monitor
{Network}

Monitor
{Computing}

Infrastructure Provider (InP)

Application:
- Requirements,
- Events,
- Incidents,
- ...

State & Others

Resources

OPEN DAYLIGHT
OpenNebula
EUCALYPTUS
openstack
© National Institute of Information and Communications Technology
Automated Resource [De]Allocation (II)

• Justification:
  – Enable **Virtual Network Operators (VNOs)** to **adapt elastically** their allocated resources to tackle dynamic requirements, events, and incidents, such as in emergency situations.
  – However, it is difficult for VNOs to “talk” to different **Infrastructure Providers (InPs)** and get the same functions, as their interfaces are heterogeneous and **not idempotent**.

• Challenges:
  – Research and agree common methods for VNOs to interact with InPs to communicate their basic **de/allocations or adaptations** requests, for example:
    • OpenDayLight, ONOS, Ryu, etc. (REST):
      – POST /resources/network/links ; PUT /resources/network/links/XXXXX
    • OpenStack, OpenNebula, Eucalyptus, etc. (REST):
      – POST /resources/computing/nodes ; PUT /resources/computing/nodes/YYYYY
    • Both including a JSON (or preferably RDF/Turtle) body with the description of the resources.
  – Specify common methods for InPs to **notify** the state of allocated resources to VNOs, including resource usage statistics:
    • Polling the REST interface is not a good approach, a “PUSH” approach should be used.
  – Regardless of the encoding of requests and responses, resource **descriptions** should follow a **common ontology**.
Support for M2M/IoT (I)

IoT Objects & Devices

“Ad-Hoc” Network

- Location-dependent virtual resources are consumed by “things”.
- Connect Ad-Hoc systems with other infrastructure resources.

VNO

Virtual Resources

VRC

InP ↔ VNO Interfaces

Monitor / Controller {Network & Computing}

InP

Distributed Premises

FOG NODES

“Close to Consumers”

Application:
- Requirements,
- Events,
- Incidents,
- ...

E/W Interfaces
Support for M2M/IoT (II)

• Justification:
  – **M2M/IoT** systems demand elastic and location-based network and computing services,
  – and need to extend “**Ad-Hoc**” systems with network and computing functions provisioned by Cloud/NFV providers.

• Challenges:
  – VNFs should be able to be instantiated within **FOG** infrastructures, which are close to their consumers.
  – Control (REST) interfaces must offer location-based (and FOG) resources:
    - POST /resources/computing/nodes
      [BODY] ... Qualities: 1CPU+5RAM+500HD; **Location**: 5BLDG+2F; ... 
      [RESPONSE] ... ID: FEDCBA98 ...
    - POST /resources/network/links
      [BODY] ... EpA: **MY-AP**; EpB: FEDCBA98; ...
  – This requires:
    - InP interfaces to expose Cloud and FOG resources with little hassle for consumers:
      – Stabilizing interfaces among InPs or among their premises (east/west).
    - Mechanisms to interconnect data-planes of “**things**” and Cloud/NFV providers.
Thanks for your attention