The Modeling interplay between Standards & Open Source NFV Orchestrator

Lingli Deng, China Mobile
Rittwik Jana, AT&T
Hui Deng, Huawei
Bruce Thompson, Cisco
Agenda

1, Model Driven Architecture
2, Individual presentations
   2.1, Model Driven Platform
   2.2, YANG/TOSCA issues with Common NFV Modelling
   2.3, Coordination between SDOs and Open Source
3, Panel discussion
4, Open Floor
Model Terminology

• Model Driven
• Information Model v.s. Data Model
• Design Time Model v.s. Run Time Model
  • Template & instantiation parameters & running status/context
• Various Models involved
  • Deployment/Creation & Management
  • Object template (Resource, Service, Product, etc.), Workflow, Policy, Analytics
Data Model templates distribution in SDN/NFV

Deployment Model
- Deployment Model
  - SDC
  - TOSCA
  - HEAT
  - DG
  - BPEL
  - Policy

Design Model
- Design Model
  - BPEL
  - TOSCA
  - HEAT
  - DG
  - YANG
  - SO
  - A&AI
  - VNF-C
  - Runtime
  - Policy
  - EMS
  - EMF
  - VNF in DC (lb, fw...)
  - VNF in Core (EPC, IMS...)

Management Model
- Management Model
  - Policy
  - BPEL
  - HEAT
  - DG
  - TOSCA
  - YANG
  - NETCONF
  - ODL
  - OPENSTACK
  - NEUTRON
  - HEAT
  - SWIFT
  - GLANCE
  - CINDER
  - FCAPS
  - Service Conf.
  - YANG
Outline

- Model Driven Architecture
- **Model Driven Platform: Design & Run Time Modeling**
- YANG/TOSCA issues with Common NFV Modelling
- Coordination between SDOs and Open Source
1. **ONAP Base Platform Framework – Release**
   - Provide Design & Execution Platform capabilities

2. **Service Agnostic Functional Building Block Development & Registration - Release**
   - Develop/test/build/release service agnostic functional building blocks
   - Register building block definitions into the Design Platform (SDC)

   Use Design Platform (SDC) to:
   - **3A: Create Object** data model & define attributes for the resource or service object
   - **3B: Compose Recipes** (e.g., Instantiate, configure, monitor, control loop, change management, etc.) with references to building block functions for lifecycle management of the object
   - **3C: Test, Certify** the designed metadata model & store in the Design Catalog
   - **3D: Distribute** the Design Catalog content to Run-time Catalog

4. **Execution – Continuously**
   - ECOMP component use of the distributed metadata to execute desired behavior

---

Functional Building Block is a service agnostic, reusable process, MicroService/Analytic application, API, Script, Event, or Policy/Rules

**Metadata Model** is a structured entity contains:
- **1 Object** Description including the definition of data attributes to describe a resource or a service
- **N Recipes** as instructions to ECOMP execution engine for Lifecycle Management of the Object
**Design Service** based on resource models & needs; includes design of underlying resource(s); design fault & other data needed from managed elements; incorporate changes based on service lifecycle feedback

Distribute artifacts including design templates, Processes, Recipes, Analytics, Policies, etc. to Design Catalog for use by the execution environment in production

Distribute design templates / artifacts (Processes, Recipes, Analytics, Policies, etc.) to Design Test Catalog; iterate over the design & test; validate design in a Certification & Testing phase

Specify policies governing service/resource behavior; adapt policy changes based on service lifecycle feedback

Define processes for closed loop monitoring & diagnosis; adapt processes with experience

Define analytics governing service/resource behavior; ensure analytic applications are in place to manage behavior; adapt analytics

Analyze & Recommend
Analyze behavior over time to identify changes needed in design, policies, analytics or thresholds governing response; could include new signatures, new analytics etc.

**Apply policy** changes based on service lifecycle feedback

Actors perform the required Actions using defined processes for various closed loop scenarios; some cases may trigger processes for diagnosis that further trigger closed-loop response; verify changes restore the service to needed levels; conditions not addressed by closed-loop response referred to Operations for further analysis

**Monitor service** by listening to events; compute analytics based on data collection; publish events that require healing or scaling based on defined policies

**Distribute artifacts** including design templates, Processes, Recipes, Analytics, Policies, etc. to Design Catalog for use by the execution environment in production

Automated Service Lifecycle In ONAP

**Instantiate Service** based on customer request or infrastructure needs; set up monitoring controls; test and turn-up service

Define analytics governing service/resource behavior; ensure analytic applications are in place to manage behavior; adapt analytics

**Define processes** for closed loop monitoring & diagnosis; adapt processes with experience

Define analytics governing service/resource behavior; ensure analytic applications are in place to manage behavior; adapt analytics

Specify policies governing service/resource behavior; adapt policy changes based on service lifecycle feedback

Actors log/publish events to record changes made for the required conditions.
Artifacts associated with each type of the models in SDC Catalog

**Language & Intended Use of Terms**

**Offer**
- A specific type or configuration of a given product, or a bundle of products
- Defines Market segment, availability, options, contract agreement, price/tax, promotion, discount, charging/billing options, payment options

**Product**
- A brand with single or combined services for customer
- Includes technical composition of the services & internal financial data for costs & revenue accounting

**Layer 4+ Services**
- Applications to provide network based services to customers
- Can be service components or E2E services

**Layer 1-3 Services**
- Service capabilities provided to customers
- Service components used to provide connections for application services

**LAN & WAN Connections**
- Connectivity configurations to provide a core backbone
- Provides connection paths across infrastructure

**Device**
- Physical or virtual component that provides Service connectivity and Network fabric (Includes managed or unmanaged CPE, UCPE)?
- Components supporting the data plane

**Infrastructure**
- Compute and storage
- Physical connectivity between nodes (all types & locations)
Outline

• Model Driven Architecture
• Model Driven Platform: Design & Run Time Modeling
• YANG/TOSCA issues with Common NFV Modelling
• Coordination between SDOs and Open Source
**NFV Orchestration with TOSCA and Yang**

Yang is a Data Modeling Language
- Defines Language Constructs for Defining Any Schema
- Yang was Originally Designed for Configuration of Appliances
- Yang Schemas Have Been Defined For NFV Orchestration

OASIS Defines TOSCA Data Modeling Language and Base Schema in a Single Specification
- TOSCA Was Designed Specifically for Cloud Orchestration

**TOSCA and Yang Provide Similar Functionality to Define Models for NFV Orchestration**

<table>
<thead>
<tr>
<th></th>
<th>Structured Schemas</th>
<th>Relationships</th>
<th>Schema Extensions</th>
<th>Life Cycle Primitives</th>
<th>Execution Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TOSCA</strong></td>
<td>Node, Capabilities, etc.</td>
<td>Requirements / Capabilities</td>
<td>Derived From</td>
<td>Defined in TOSCA Simple Profile</td>
<td>get_input, get_property, get_attribute</td>
</tr>
<tr>
<td><strong>Yang</strong></td>
<td>Containers</td>
<td>LeafRef, Identity / IdentityRef</td>
<td>Augment</td>
<td>Defined in Yang Schema</td>
<td>Netconf HTTP Rest</td>
</tr>
</tbody>
</table>
TOSCA and Yang Models for NFV

Difficult to Translate Directly Between Yang and TOSCA

- TOSCA and Yang Have Similar Functionality to Define Data Models
- TOSCA Defines a Data Model and an Execution Environment
  - Yang Separates Data Model from Execution Environment

Yang and TOSCA Data Models Can be Derived from the Same Parent Data Model
Parent Data Model for NFV

Potential Parent Data Models for TOSCA and Yang NFV Models

ETSI NFV IFA011 UML Model – Designed Specifically for NFV
TOSCA Simple Profile in Yaml - Data Center Centric Model Can be Applied to NFV

ETSI NFV and TOSCA Data Models Were Developed Independently
ETSI NFV and TOSCA Define Similar Information Elements, Different Structure
Difficult to Create a Data Model Derived from TOSCA Simple Profile That is Consistent with ETSI NFV

Original TOSCA NFV Profile – TOSCA NFV Profile CSD003
  Derived from TOSCA Simple Profile
  Node Hierarchy Different Than ETSI NFV Data Model

New TOSCA NFV Profile – ETSI NFV Sol Work Item #1
  Node Hierarchy The Same as ETSI NFV Data Model
  Most Nodes, Capabilities, etc. Not Derived from Existing TOSCA Simple Profile

TOSCA Simple Profile and / or ETSI NFV Models Must Change to be Consistent
Outline

• Model Driven Architecture
• Model Driven Platform: Design & Run Time Modeling
• YANG/TOSCA issues with Common NFV Modelling
• Coordination between SDOs and Open Source
## What is the issue?

<table>
<thead>
<tr>
<th>Working Group</th>
<th>No.</th>
<th>Version</th>
<th>Title</th>
<th>OPEN-O</th>
<th>OSM</th>
<th>Tacker</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAN</td>
<td>GS NFV-MAN 001</td>
<td>v010101</td>
<td>Management and Orchestration</td>
<td>✓/</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>IFA</td>
<td>GS NFV-IFA 007</td>
<td>v020101</td>
<td>Management and Orchestration; Or-Vnfm reference point</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>IFA</td>
<td>GS NFV-IFA 011</td>
<td>v020101</td>
<td>Management and Orchestration; VNF Packaging</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>IFA</td>
<td>GS NFV-IFA 014</td>
<td>v020101</td>
<td>Management and Orchestration; Network Service Templates</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>SOL</td>
<td>SCL001</td>
<td></td>
<td>TOSCA based NFV Descriptors (VNFD/NSD)</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>SOL</td>
<td>SCL003</td>
<td></td>
<td>Or-Vnfm Protocols</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>SOL</td>
<td>SCL004</td>
<td></td>
<td>VNF Package</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OASIS</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Document Name</td>
<td>Version</td>
<td>Chapter</td>
<td>Title</td>
<td>OPEN-O</td>
<td>OSM</td>
<td>Tacker</td>
</tr>
<tr>
<td>TOSCA NPV (MANO based)</td>
<td>v1.0 CSD03</td>
<td>8.9.10.11</td>
<td>TOSCA Template for VNFD, VLD, VNFFGD, NSD</td>
<td>✓/</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>TOSCA NPV (IFA based)</td>
<td>v1.1 CSD04</td>
<td>8.9.10.11</td>
<td>TOSCA Template for VNFD, VLD, VNFFGD, NSD</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>TOSCA Simple Profile YAML</td>
<td>v1.1</td>
<td>3.9</td>
<td>Basic TOSCA yaml feature</td>
<td>✓/</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>TOSCA Simple Profile YAML</td>
<td>v1.1</td>
<td>6</td>
<td>CSAR</td>
<td>✓/</td>
<td>×</td>
<td>×</td>
</tr>
</tbody>
</table>
OpenDeployment.org about NFV Modeling between standard and open source

① Give up ETSI NFV ISG IFA/SOL spec, just let OSS run
② Tosca fork because of IFA 1:1 map to Tosca profile
③ OpenDeployment coordinate between SDOs and various OSS projects on NFV descriptors by reverse direction
Panel discussion
Thank You