TOSCA
Topology and Orchestration Specification for Cloud Applications (TOSCA) Standard

How TOSCA Adds Value in the NFV world

OASIS TOSCA Presentation for the ETSI NFV Information Modeling Workshop
Overview and Proposal for a way forward
Agenda - *How TOSCA Adds Value in the NFV world*

I. TOSCA Overview
   - What is TOSCA?
   - TOSCA Key Modeling Concepts
     - Topology, Composition, Lifecycle (management), Portability
   - Interesting Features
     - Containers, Portability, Network modeling

II. A Way Forward
   - TOSCA Modeling Applied to NFV
     - Topology & Composition Concepts applied to NFV (NSD, VNF, VNFFG, NFP)
   - Open Source Implementations
     - Openstack (*Heat-Translator*, *Tacker*, *Senlin*), alien4cloud, Cloudify, etc.
   - Backup slides
     - Layering, Lifecycle sequencing, Policy model
TOSCA Overview
Key Modeling Concepts & Features
Agenda - *How TOSCA Adds Value in the NFV world*

I. TOSCA Overview
   - What is TOSCA?
   - TOSCA Key Modeling Concepts
     - Topology, Composition, Lifecycle (management), Portability
   - Interesting Features
     - Containers, Portability, Network modeling

II. A Way Forward
   - TOSCA Modeling Applied to NFV
     - Topology & Composition Concepts applied to NFV (NSD, VNF, VNFFG, NFP)
   - Open Source Implementations
     - Openstack (*Heat-Translator*, *Tacker*, *Senlin*), alien4cloud, Cloudify, etc.
   - Backup slides
     - Layering, Lifecycle sequencing, Policy model
What is TOSCA?

TOSCA is an important new open cloud standard, that is enabling a unique ecosystem, supported by a large and growing number of international industry leaders...

- TOSCA Version 1.0 Specification approved as an OASIS Standard (published Nov 2013)
- TOSCA Simple Profile v1.0 Specification (YAML) final public review draft, Aug 2014
- Government and Corporate Awareness:
  - OASIS: 600+ participant organizations. 5000+ participants spanning 65+ countries
  - TOSCA Committee: 170+ people 45+ companies/orgs
  - International Standards & Research: ETSI NFV liaison, EU FP7, etc.
  - Industry Analysts: Forrester names TOSCA as a top four cloud open standard (Mar 2014)
- Multi-company Interoperability Demonstrated:
  - OSCON 2015, Open Data Center Alliance 2014, EuroCloud 2013
Agenda - How TOSCA Adds Value in the NFV world

I. TOSCA Overview
   - What is TOSCA?
   - **TOSCA Key Modeling Concepts**
     - Topology, Composition, Lifecycle (management), Portability
   - **Interesting Features**
     - Containers, Portability, Network modeling

II. A Way Forward
   - **TOSCA Modeling Applied to NFV**
     - Topology & Composition Concepts applied to NFV (NSD, VNF, VNFFG, NFP)
   - **Open Source Implementations**
     - Openstack *(Heat-Translator, Tacker, Senlin)*, alien4cloud, Cloudify, etc.
   - **Backup slides**
     - Layering, Lifecycle sequencing, Policy model
Primarily, TOSCA is mainly used to describe the topology of the deployment view for cloud applications

- Defining **Node templates** to describe components in the topology structure
- Defining **Relationship templates** to describe connections, dependencies, deployment ordering

TOSCA can be used to describe the topology of a Network service or VNF as defined by ETSI NFV.

VNF, VL can be defined as node templates in TOSCA. A new virtualLinksTo relationship type can be defined to connect VNF and VL.
Using the TOSCA substitution feature, NFV information model can be described by using multiple TOSCA service templates.
Lifecycle

**TOSCA models have a consistent view of state-based lifecycle**

- have **Operations** (implementations) that can be sequenced against state of any dependent resources
- fits into any **Management Framework** or **Access Control System**

**Standardize Resource Lifecycle**

- `my_resource_name`
- `My_Resource_Type`
- Lifecycle.Standard
  - create
  - configure
  - start
  - stop
  - delete

**Standardize Relationship Lifecycle**

- `source_resource`
  - `Type_A`
  - A
- `target_resource`
  - `Type_B`
  - B
- my_relationship
  - ConnectsTo
- Lifecycle.Configure
  - pre_config_source
  - post_config_source
  - add_source
  - remove_source
  - pre_config_target
  - post_config_target
  - add_target
  - remove_target

**Lifecycle Customization**

- Lifecycle.Configure.NFV
  - `nfv_pre_config`
  - `nfv_pre_delete`
- Lifecycle.Standard
  - create
  - configure
  - start
  - stop
  - delete

Create new NFV Lifecycles or Augment existing (via subclassing)

- Parameters and Policies can be supplied to operations to affect resource behavior (state)
- Workflow - TOSCA is developing workflow to allow handling complex state changes, configurations, etc.

**TOSCA Lifecycle can be customized for NFV Resources and Relationships**
By expressing application *Requirements* independently from cloud *Capabilities* and implementation, TOSCA provides:

- Robust set of **Normative Types** for different domains (including NFV)
- **NFV Profile and Types applied to TOSCA enable our Way Forward.**
- Multi VIM Support
- Portability of services across clouds
- Declarative model spanning infrastructure and service
- Manipulate the orchestration declaratively instead of dealing with disparate cloud APIs *(leave to the TOSCA Orchestrator)*

**TOSCA enables NFV applications flexible movement between different cloud infrastructures.**
Agenda - *How TOSCA Adds Value in the NFV world*

I. TOSCA Overview
   - What is TOSCA?
   - TOSCA Key Modeling Concepts
     - Topology, Composition, Lifecycle (management), Portability
   - Interesting Features
     - Containers, Portability, Network modeling

II. A Way Forward
   - TOSCA Modeling Applied to NFV
     - Topology & Composition Concepts applied to NFV (NSD, VNF, VNFFG, NFP)
   - Open Source Implementations
     - Openstack (*Heat-Translator, Tacker, Senlin*), alien4cloud, Cloudify, etc.
   - Backup slides
     - Layering, Lifecycle sequencing, Policy model
TOSCA Model for Containers leveraging Repositories

PaaS Modeling

- Provider chooses to expose or hide underlying runtime topology & implementation

Container Application Modeling

- Agnostic of PaaS Cloud Provider
  - PaaS on OpenStack, Cloud Foundry, Azure, etc.

PaaS Layer exposes “runtimes” as TOSCA Capabilities

- Docker, Nodejs, JSP, J2EE, etc.

Orchestrators could automatically retrieve and deploy a Docker image from a declared Repository

- TOSCA Templates can model repositories
- Orchestrators could dynamically “pull” from multiple repositories
TOSCA Model for Logical Public & Private Cloud Networks

- Application Model separate from Network Model

Layer separation allows developers to model JUST the application & bind later to existing tenant networks (Layer 4)
TOSCA Direction to model Policies

TOSCA defines policies using an Event-Condition-Action model

- Operational policy focus: Placement (Affinity), Scaling and Performance
  - with Rules that are evaluated to execute Automatic and Imperative Triggers

Policies modeled as Requirements using Capability Types that can be attached to
1. Interfaces for specific Operations
2. Nodes and
3. Groups of Nodes
End Part 1

TOSCA Overview
TOSCA
The Way forward
TOSCA Concepts Applied to NFV
Agenda - *How TOSCA Adds Value in the NFV world*

I. TOSCA Overview

- What is TOSCA?
- TOSCA Key Modeling Concepts
  - Topology, Composition, Lifecycle (management), Portability
- Interesting Features
  - Containers, Portability, Network modeling

II. A Way Forward

- TOSCA Modeling Applied to NFV
  - Topology & Composition Concepts applied to NFV (NSD, VNF, VNFFG, NFP)
- Open Source Implementations
  - Openstack (*Heat-Translator, Tacker, Senlin*), alien4cloud, Cloudify, etc.
- Backup slides
  - Layering, Lifecycle sequencing, Policy model
Agenda - *How TOSCA Adds Value in the NFV world*

I. TOSCA Overview
   - What is TOSCA?
   - TOSCA Key Modeling Concepts
     - Topology, Composition, Lifecycle (management), Portability
   - Interesting Features
     - Containers, Portability, Network modeling

II. A Way Forward
   - TOSCA Modeling Applied to NFV
     - Topology & Composition Concepts applied to NFV (NSD, VNF, VNFFG, NFP)
   - Open Source Implementations
     - Openstack (*Heat-Translator, Tacker, Senlin*), alien4cloud, Cloudify, etc.
   - Backup slides
     - Layering, Lifecycle sequencing, Policy model
Topology

Primarily, TOSCA is mainly used to describe the topology of the deployment view for cloud applications

- Defining **Node templates** to describe components in the topology structure
- Defining **Relationship templates** to describe connections, dependencies, deployment ordering

**TOSCA can be used to describe the topology of a Network service or VNF as defined by ETSI NFV.**

VNF, VL can be defined as node templates in TOSCA. A new virtualLinksTo relationship type can be defined to connect VNF and VL.
Using the TOSCA substitution feature, NFV information model can be described by using multiple TOSCA service templates.
VNFD example

node_templates:
  VDU1:
    type: toscas.nodes.nfv.VDU
    properties:
      # omitted here for brevity
    requirements:
      - host:
        node_filter:
          capabilities:
            # Constraints for selecting "Host" (Container Capability):
            - host:
              properties:
                num_cpus: { in_range: [1, 4] }
                mem_size: { greater_or_equal: 2 GB }
            # Constraints for selecting "os" (OperatingSystem Capability):
            - os:
              properties:
                architecture: { equal: x86_64 }
                type: linux
                distribution: ubuntu
      interfaces:
        VM_image: vdu1.image # the VM image of VDU1
    implementation: vdu1_config.sh
  VDU2:
    type: toscas.nodes.nfv.VDU
    properties:
      # omitted here for brevity
  VDU3:
    type: toscas.nodes.nfv.VDU
    properties:
      # omitted here for brevity

CP21:
  type: toscas.nodes.nfv.CP
  properties:
  requirements:
    virtualbinding: VDU1
  virtuallink:
    internal_VL

CP22:
  type: toscas.nodes.nfv.CP
  properties:
  requirements:
    virtualbinding: VDU1
    virtuallink: internal_VL

CP23:
  type: toscas.nodes.nfv.CP
  properties:
  requirements:
    virtualbinding: VDU2
    virtuallink: internal_VL

CP24:
  type: toscas.nodes.nfv.CP
  properties:
  requirements:
    virtualbinding: VDU3
    virtuallink: internal_VL internal_VL

virtuallinkable: [CP21, virtuallinkable]
Network forwarding path as defined by **ETSI NFV** is an order list of connection points forming a chain of network functions (VNFs or PNFs). A new “Forwarder” requirement is defined in this specification to model the network forwarding path by using ordered list of multiple “Forwarder” requirements. Each “Forwarder” requirement points to a single connection point.
Using TOSCA **Group** element to describe forwarding graphs

```
Groups:

VNFFG1:
  type: tosca.groups.nfv.vnffg
  description: forwarding graph 1
  properties:
    vendor:
    version:
    vl: [VL1,VL2,VL4]
    vnf: [VNF1,VNF2,VNF3]
  targets: [Forwarding path1, Forwarding path2]

VNFFG2:
  type: tosca.groups.nfv.vnffg
  description: forwarding graph 2
  properties:
    vendor:
    version:
    vl: [VL1,VL3,VL4]
    vnf: [VNF1,VNF2]
  targets: [Forwarding path3]
```

Source: xxx
Agenda - *How TOSCA Adds Value in the NFV world*

I. TOSCA Overview
   - What is TOSCA?
   - TOSCA Key Modeling Concepts
     - Topology, Composition, Lifecycle (management), Portability
   - Interesting Features
     - Containers, Portability, Network modeling

II. A Way Forward
   - TOSCA Modeling Applied to NFV
     - Topology & Composition Concepts applied to NFV (NSD, VNF, VNFFG, NFP)
   - **Open Source Implementations**
     - Openstack (*Heat-Translator*, *Tacker*, *Senlin*), alien4cloud, Cloudify, etc.
   - Backup slides
     - Layering, Lifecycle sequencing, Policy model
TOSCA in Open Source

**Heat-Translator**
TOSCA Template Translation to other Domain Specific Languages (DSLs)

**Senlin**
Clustering + Placement & Scaling Policies

**Tacker**
NFV MANO
[https://wiki.openstack.org/wiki/Tacker](https://wiki.openstack.org/wiki/Tacker)

**Cloudify**
Service Orchestration & Management

**alien4cloud**
Topology, Type & LCM Design

**Seaclouds**
Open, Multi-Cloud Managemnt
[www.seaclouds-project.eu/media.html](http://www.seaclouds-project.eu/media.html)

**CERN Indigo-DataCloud**
Data/computing platform targeted at scientific communities
[http://information-technology.web.cern.ch/about/projects/eu/indigo-datacloud](http://information-technology.web.cern.ch/about/projects/eu/indigo-datacloud)

**OPNFV Parser**
Deployment Template Translation
Automated TOSCA-based Orchestration Now Part of OpenStack

TOSCA-Parse and Heat-Translator are part of the OpenStack Heat orchestration project
- Latest TOSCA features integrated: Networking, Block & Object Storage...
- Availability to use on command line & user input param support

New features in Liberty release and plans for Mitaka
- **Tacker** NFV MANO integration using TOSCA NFV Profile
- **Murano** (Application catalog integration) with OpenStack client
- TOSCA parser available as independent **Python library** (pypi)
- TOSCA **Policy** schema and **Group** schema
- **Plug-ins**: HOT Generator now supports additional plug-ins to allow translation to other DSLs besides HOT, such as **Kubernetes**
OpenSource related to ETSI NFV and OASIS TOSCA

- Openstack Senlin
- OPNFV Parser
- ETSI NFV
  - NFVO / VNFM / Catalog
  - OpenStack Tacker
- TOSCA parser
- OpenStack Heat Translator
  - heat-translator

- Policy
- Data modeling
- ETSI NFV Descriptor
- NSD VNFD VNFFGD
- OASIS TOSCA
- OPNFV Parser
- OpenStack Tacker
- TOSCA parser
- OpenStack Heat Translator
  - heat-translator

APIs
- OpenStack Dashboard
- Compute
- Networking
- Storage

OpenStack CLOUD OPERATING SYSTEM

OpenStack Shared Services
TOSCA Resources – Learn More

- **TOSCA Technical Committee Public Page** *(latest documents, updates, and more)*

- **OASIS YouTube Channel, TOSCA Playlist**
  - [https://www.youtube.com/user/OASISopen](https://www.youtube.com/user/OASISopen), [http://bit.ly/1BQGGHm](http://bit.ly/1BQGGHm)

- **TOSCA Simple Profile in YAML v1.0** *(latest committee approved draft)*
  - [http://docs.oasis-open.org/tosca/TOSCA-Simple-Profile-YAML/v1.0/TOSCA-Simple-Profile-YAML-v1.0.pdf](http://docs.oasis-open.org/tosca/TOSCA-Simple-Profile-YAML/v1.0/TOSCA-Simple-Profile-YAML-v1.0.pdf)

- **TOSCA Simple Profile for NFV v1.0** *(latest committee approved draft)*
  - [http://docs.oasis-open.org/tosca/tosca-nfv/v1.0/csd02/tosca-nfv-v1.0-csd02.pdf](http://docs.oasis-open.org/tosca/tosca-nfv/v1.0/csd02/tosca-nfv-v1.0-csd02.pdf)

- **Contact the Technical Committee Co-Chairs:**
  - Paul Lipton, paul.lipton@ca.com; Simon Moser, smoser@de.ibm.com

- **Today’s Presenters from the TOSCA TC:**
  - Shitao Li, lishitao@huawei.com
  - Matt Rutkowski, mrutkows@us.ibm.com
  - Chris Lauwers, lauwers@ubicity.com
  - Sridhar Ramaswamy, sramasw@Brocade.com
  - Sivan Barzily, sivan@gigaspaces.com
How TOSCA Adds Value in the NFV world

End Part 2
A Way Forward
TOSCA
Backup Slides
TOSCA Pattern Domains: IaaS, PaaS, NFV, HW and more

- TOSCA’s Simple Profile Specification (YAML) Primary Goal was to
  - Simplify Application-Centric modeling, but also supports modeling of
  - DevOps & Workflow: Groups, Policies, Repositories, Artifacts, Configurations

TOSCA

PaaS, SaaS Patterns

Containers & Domain Patterns

IaaS Patterns

SW-Defined Systems and HW Patterns

Logical

Software Apps / Services

Physical

- Simplify Application-Centric modeling, but also supports modeling of
- DevOps & Workflow: Groups, Policies, Repositories, Artifacts, Configurations
TOSCA Orchestrators - Standardized Lifecycle Sequencing

**Deploy Sequencing**
- TOSCA Orchestrator
- TOSCA Lifecycle Operation
- TOSCA Node State
  - create()
  - configure()
  - start()
  - initial
  - created
  - configured
  - started

**Undeploy Sequencing**
- TOSCA Orchestrator
- TOSCA Lifecycle Operation
- TOSCA Node State
  - stop()
  - delete()
  - initial
  - created
  - configured

**Source-Target Sequencing**

**Lifecycle.Configure**
- TOSCA Orchestrator
- TOSCA Lifecycle Operation
- TOSCA Node State
  - create()
  - configure()
  - start()
  - initial
  - created
  - configured

**Combined Sequencing**

**Node A**
- (source)
- Operations
  - create
  - configure
  - start

**Node B**
- (target)
- Operations
  - add_target
  - remove_target

**Relationship A:B**
- Operations
  - pre_configure_source
  - post_configure_source
  - pre_configure_target
  - post_configure_target
  - target_changed
TOSCA Policies Sample: Event-Condition-Action

Event
- Name of a normative TOSCA Event Type
- That describes an event based upon a Resource “state” change.
- Or a change in one or more of the resources attribute value.

Condition
Identifies:
- the resource (Node) in the TOSCA model to monitor.
- Optionally, identify a Capability of the identified node.
- Describe the attribute (state) of the resource to evaluate (condition)

Action
Describes:
- An Operation (name) to invoke when the condition is met
- within the declared Implementation
- Optionally, pass in Input parameters to the operation along with any well-defined strategy values.

– Allows Triggers to be declared based upon an Event, Condition, Action model