

APRIL 3-6, 2017  
SANTA CLARA



ons

**OPEN NETWORKING //**  
HARMONIZE, HARNESS, CONSUME



ons

OPEN NETWORKING //  
HARMONIZE, HARNESS, CONSUME

APRIL 3-6, 2017  
SANTA CLARA

# The Modeling interplay between Standards & Open Source NFV Orchestrator

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



ons

OPEN NETWORKING //  
HARMONIZE, HARNESS, CONSUME

# 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



ons

OPEN NETWORKING //  
HARMONIZE, HARNESS, CONSUME

# Model Terminology

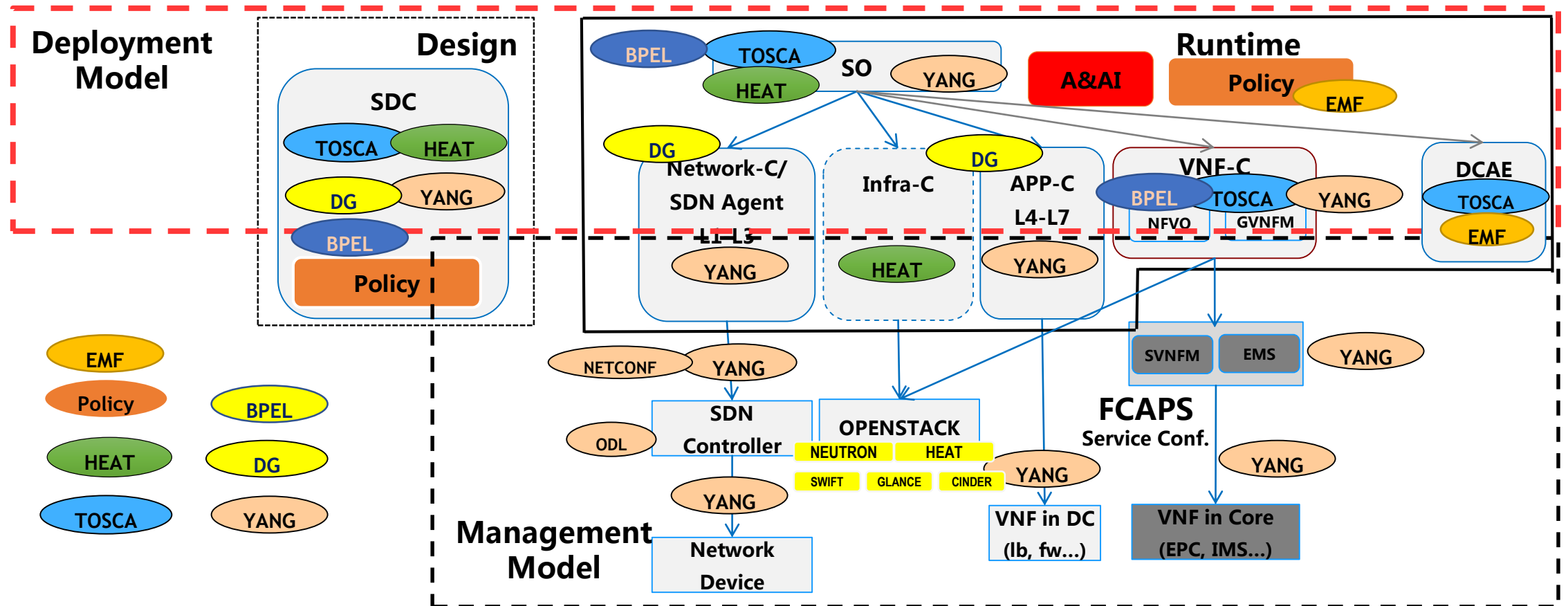
- Model Driven
- Information Model v.s. Data Model
- Design Time Model v.s. Run Time Model
  - Template & instantiation parameters & running status/context
- Various Models involved
  - Deployment/Creation & Management
  - Object template (Resource, Service , Product, etc.), Workflow, Policy, Analytics



ons

OPEN NETWORKING //  
HARMONIZE, HARNESS, CONSUME

# Data Model templates distribution in SDN/NFV





ons

OPEN NETWORKING //  
HARMONIZE, HARNESS, CONSUME

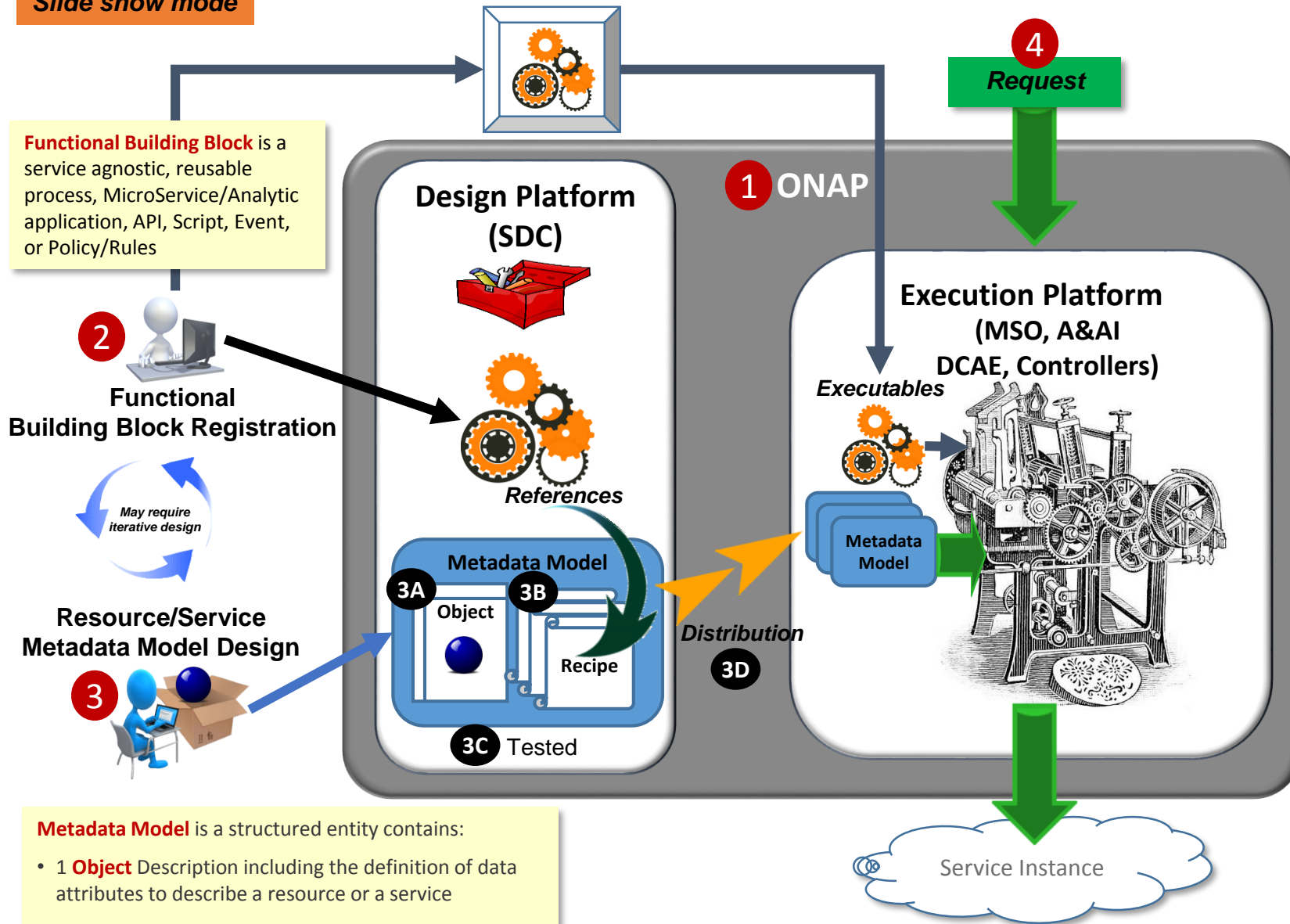
# Outline

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



# ONAP Target Implementation - Design "Metadata Model Driven" Execution

Slide show mode



**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

**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)

**3. Resource & Service Metadata Model Design – Release Independent**

Use Design Platform (SDC) to:

- **3A: Create Object** data model & define attributes for the resource or service object
- **3B: Compose Recipes** (eg., 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

# Automated Service Lifecycle In ONAP

**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

**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

**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

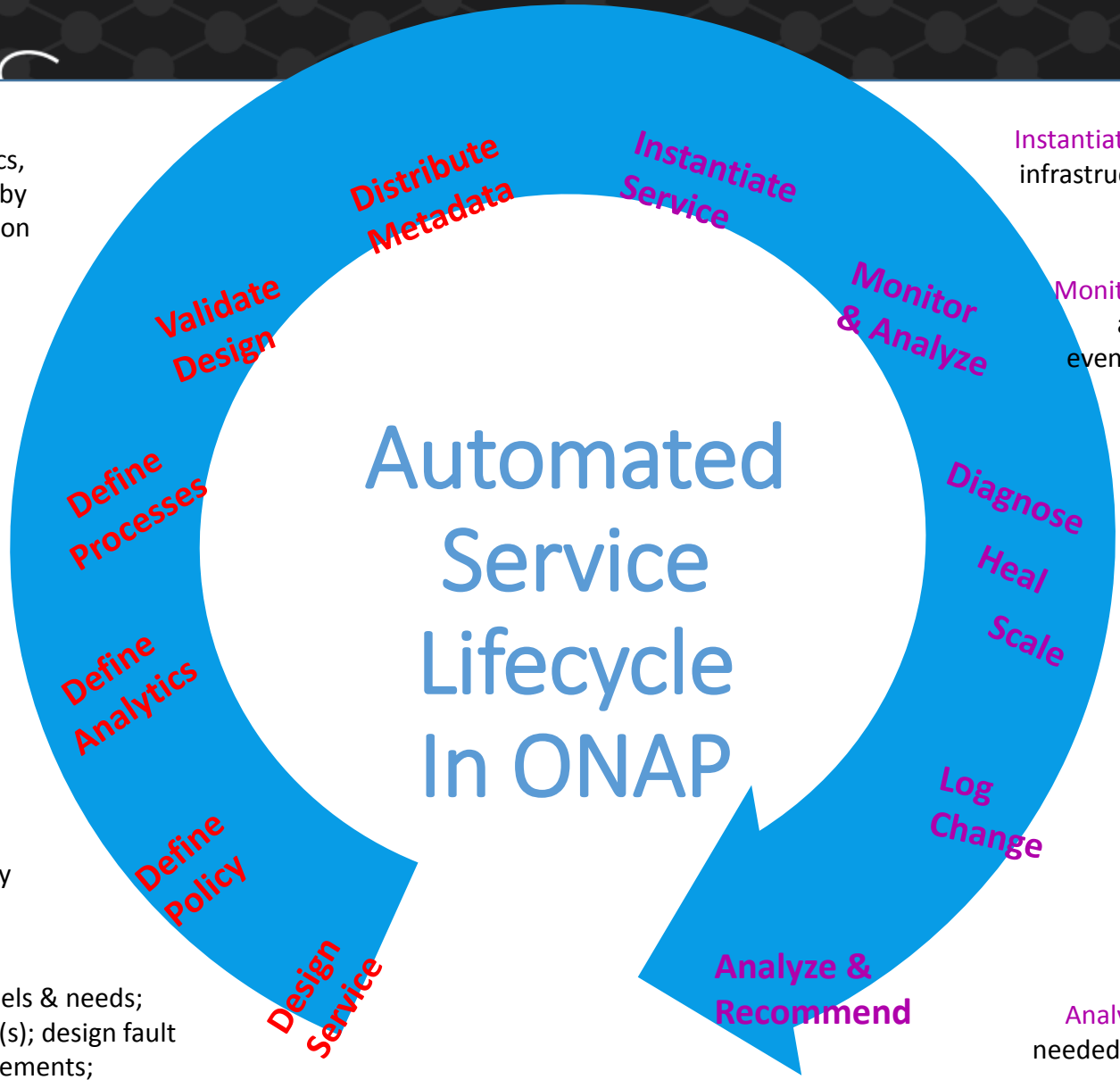
**Instantiate Service** based on customer request or infrastructure needs; set up monitoring controls; test and turn-up service

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

**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

Actors **log/publish** events to record changes made for the required conditions.

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



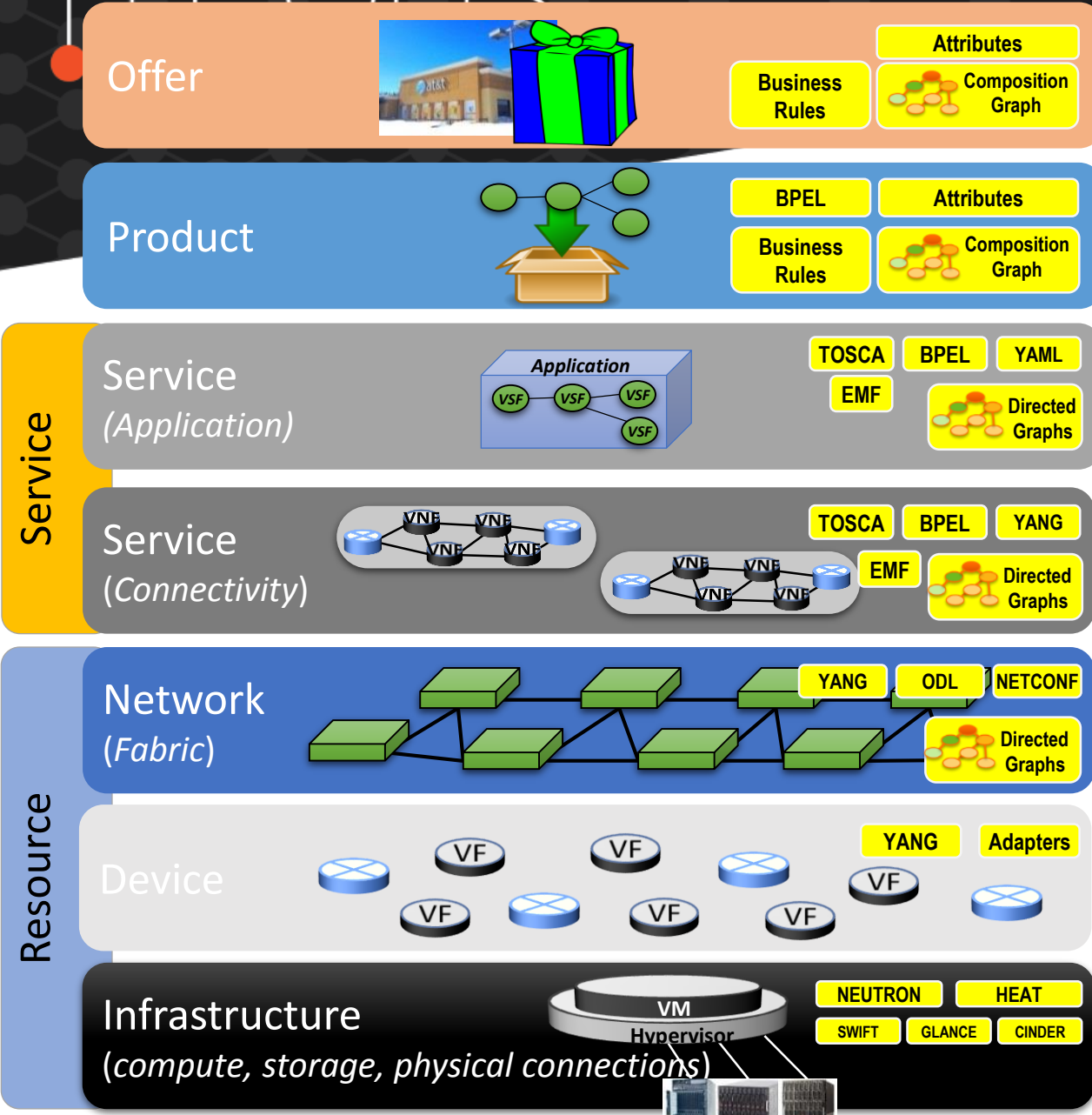
DESIGN PHASE

EXECUTION PHASE



# 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

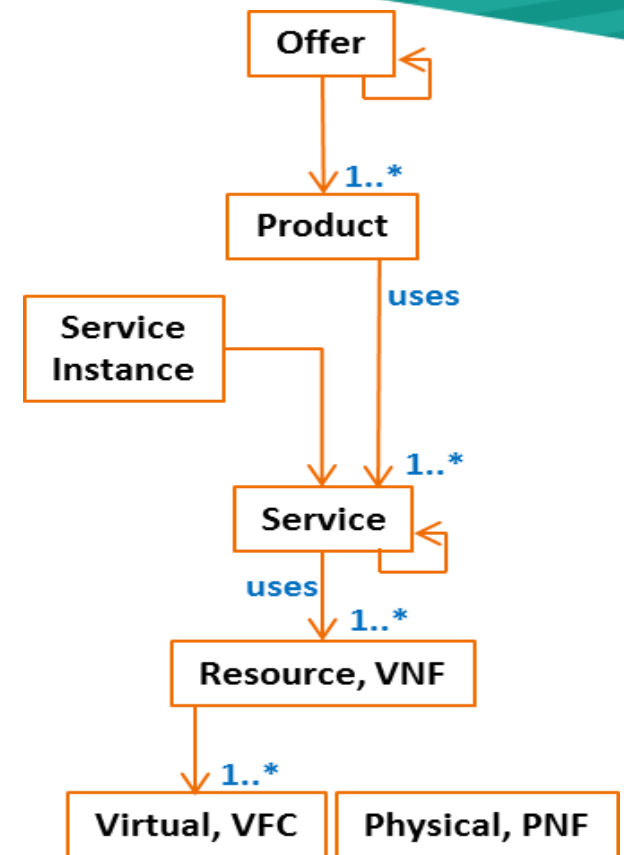
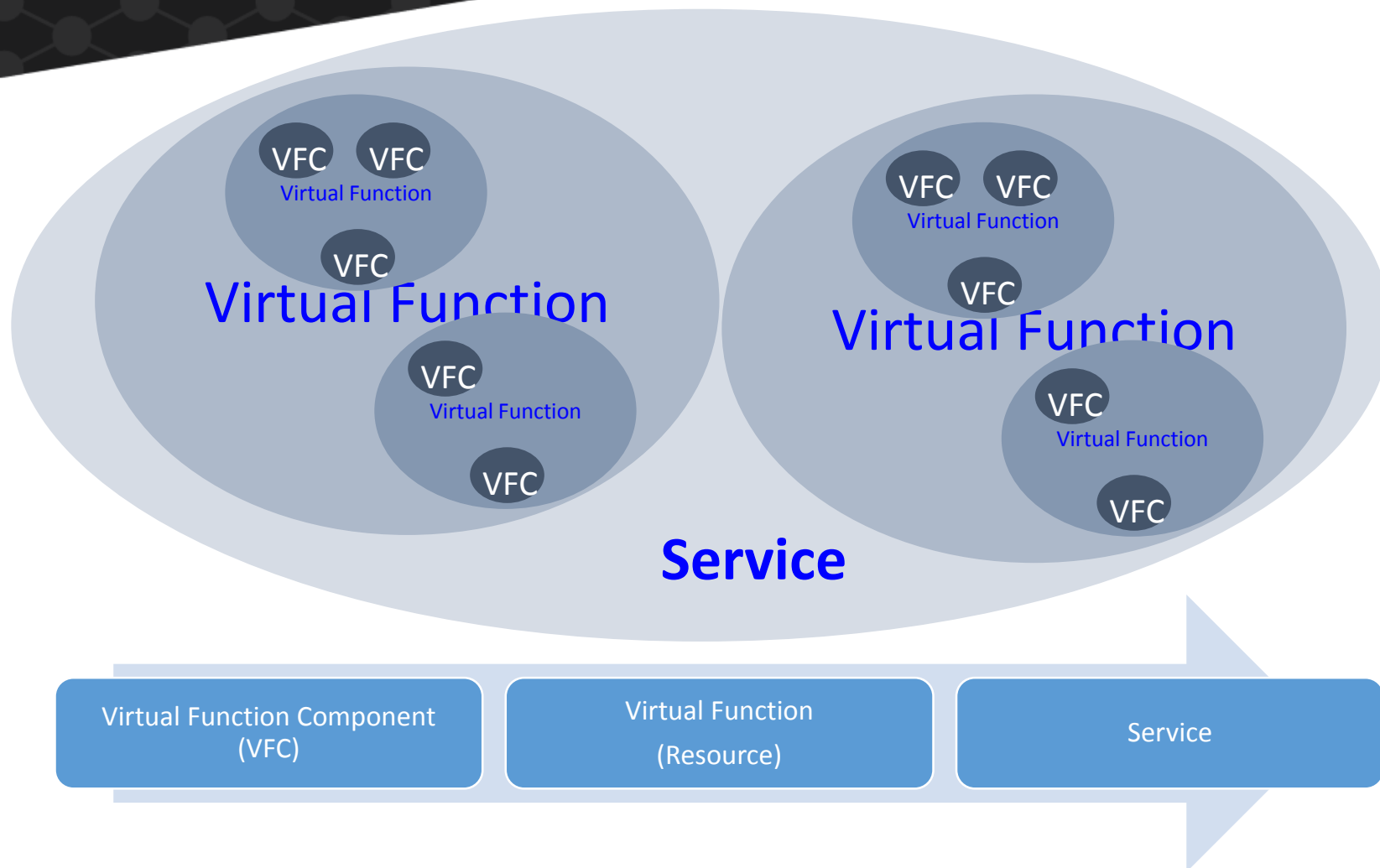
- Compute and storage
- Physical connectivity between nodes (all types & locations)



# ons

OPEN NETWORKING //  
HARMONIZE, HARNESS, CONSUME

## Virtual Function Entity Relationship - Illustration





ons

OPEN NETWORKING //  
HARMONIZE, HARNESS, CONSUME

## Outline

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



ons

OPEN NETWORKING //  
HARMONIZE, HARNESS, CONSUME

# 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**

	Structured Schemas	Relationships	Schema Extensions	Life Cycle Primitives	Execution Environment
<b>TOSCA</b>	Node, Capabilities, etc.	Requirements / Capabilities	Derived From	Defined in TOSCA Simple Profile	get_input, get_property, get_attribute
<b>Yang</b>	Containers	LeafRef, Identity / IdentityRef	Augment	Defined in Yang Schema	Netconf HTTP Rest



ons

OPEN NETWORKING //  
HARMONIZE, HARNESS, CONSUME

# TOSCA and Yang Models for NFV

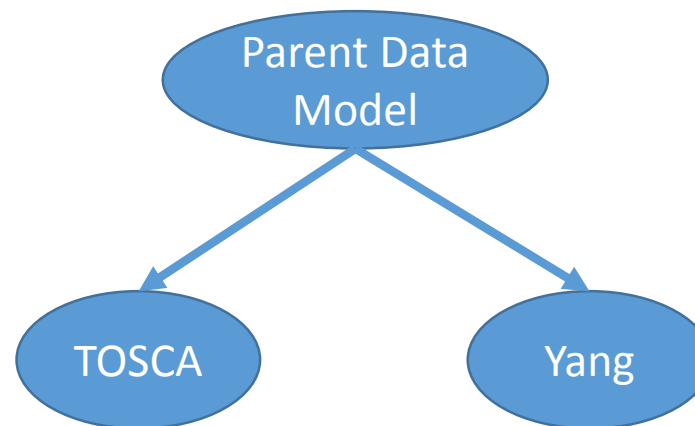
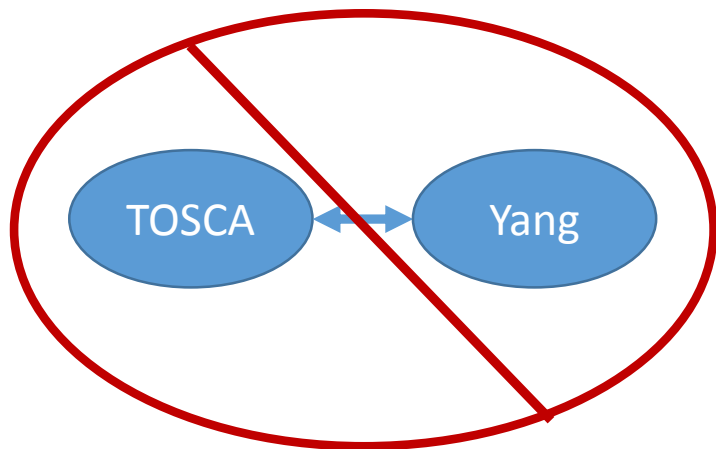
Difficult to Translate Directly Between Yang and TOSCA

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**





ons

OPEN NETWORKING //  
HARMONIZE, HARNESS, CONSUME

# 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**





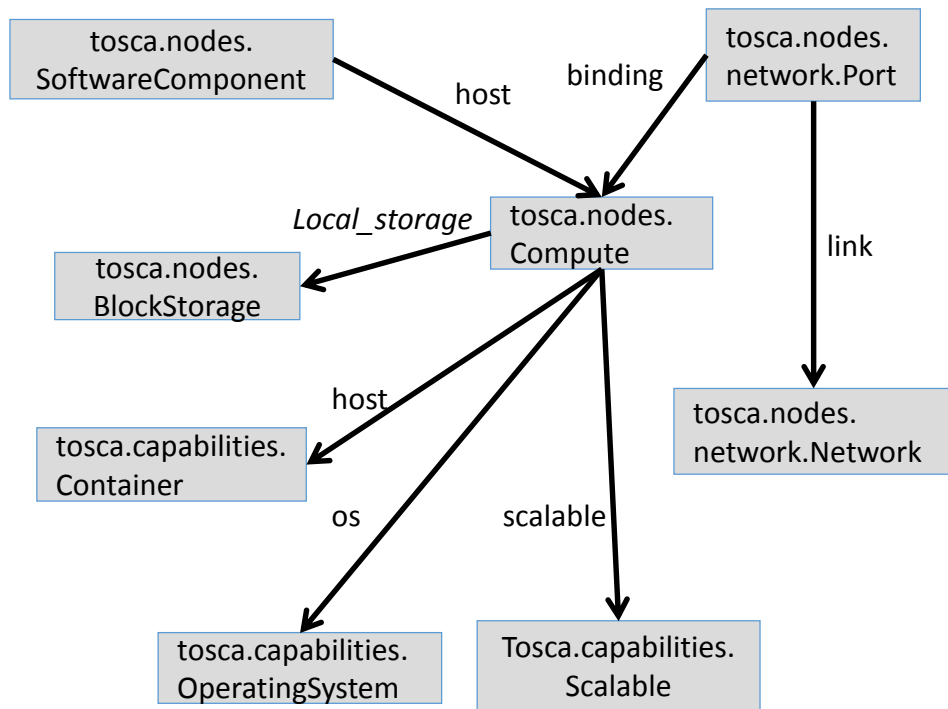
ons

OPEN NETWORKING //  
HARMONIZE, HARNESS, CONSUME

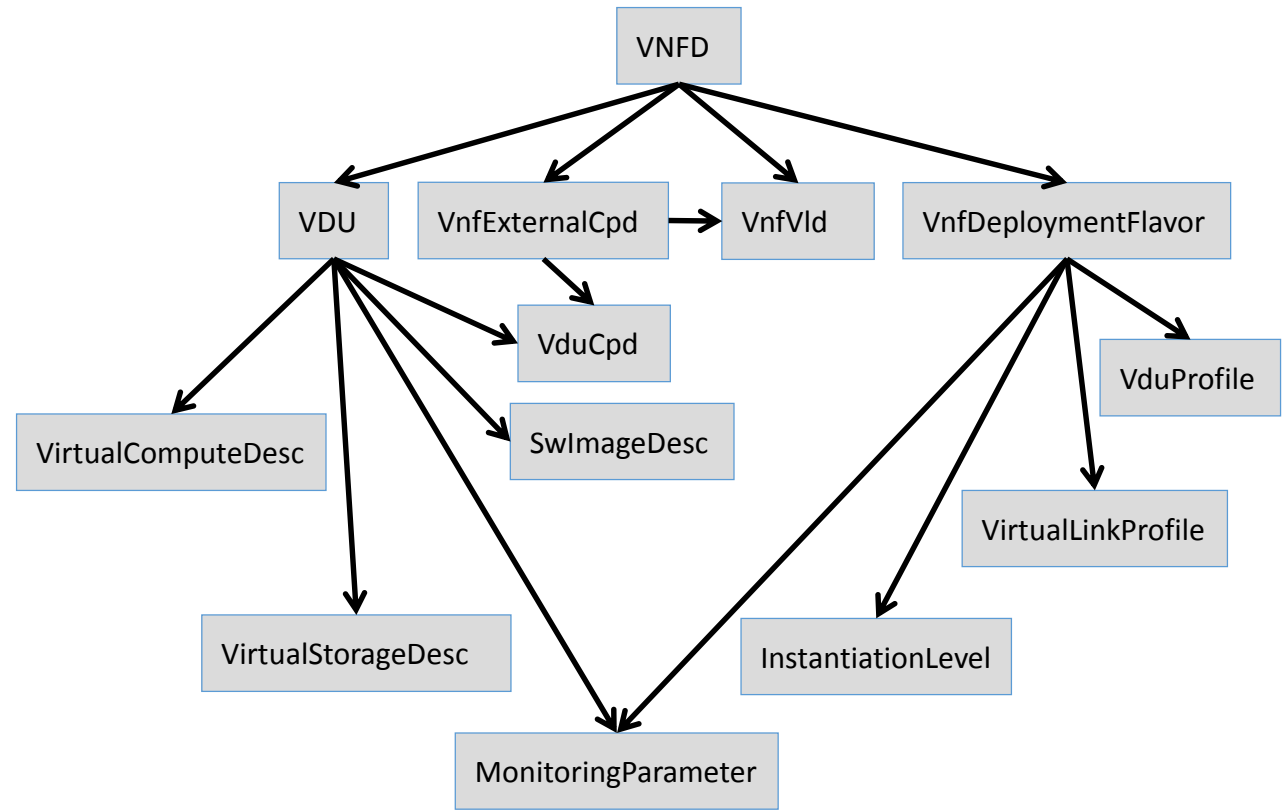
host

# Tosca NFV Profile Structure

## TOSCA Simple Profile



## ETSI NFV IFA011





ons

OPEN NETWORKING //  
HARMONIZE, HARNESS, CONSUME

## Outline

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



ons

OPEN NETWORKING //  
HARMONIZE, HARNESS, CONSUME

# What is the issue?

ETSI				OPEN-O	OSM	Tacker
Working Group	No.	Version	Title			
MAN	GS NFV-MAN 001	v010101	Management and Orchestration	✓	✓	✓
IFA	GS NFV-IFA 007	v020101	Management and Orchestration; Or-Vnfm reference point	✓		✗
IFA	GS NFV-IFA 011	v020101	Management and Orchestration; VNF Packaging	✗	✗	✗
IFA	GS NFV-IFA 014	v020101	Management and Orchestration; Network Service Templates	✗	✗	✗
SOL	SOL001		TOSCA based NFV Descriptors (VNFD/NSD)	✗	✗	✗
SOL	SOL003		Or-Vnfm Protocols	✗	✗	✗
SOL	SOL004		VNF Package	✗	✗	✗
OASIS						
Document Name	Version	Chapter	Title			
TOSCA NFV (MANO based)	v1.0 CSD03	8, 9, 10, 11	TOSCA Template for VNFD, VLD, VNFFGD, NSD	✓	✗	✓
TOSCA NFV (IFA based)	v1.1 CSD04	8, 9, 10, 11	TOSCA Template for VNFD, VLD, VNFFGD, NSD	✗	✗	✗
TOSCA Simple Profile YAML	v1.1	3.9	Basic TOSCA yaml feature	✓	✗	✓
TOSCA Simple Profile YAML	v1.1	6	CSAR	✓	✗	

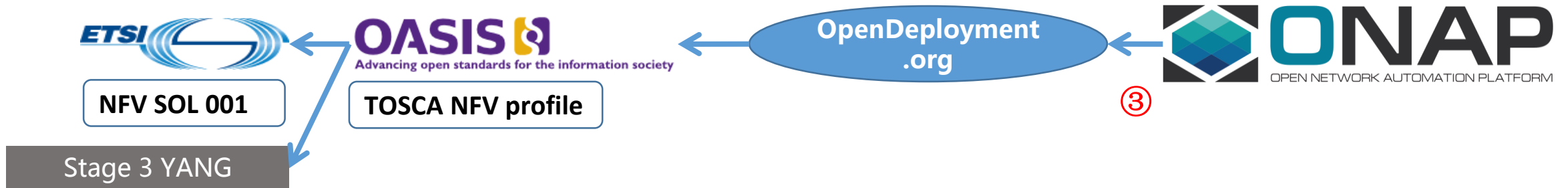
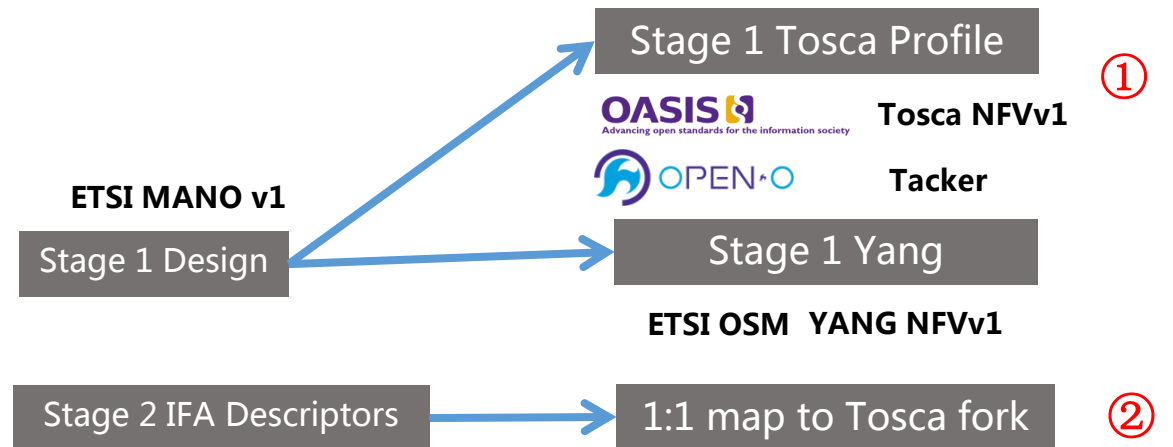


ons

OPEN NETWORKING //  
HARMONIZE, HARNESS, CONSUME

## 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**





ons

OPEN NETWORKING //  
HARMONIZE, HARNESS, CONSUME

# Panel discussion



ons

OPEN NETWORKING //  
HARMONIZE, HARNESS, CONSUME

**Thank You**