



Overview of DMTF Information and Data Models

ETSI NFV m-SDO Workshop, January 13, 2015

Presenter:

Hemal Shah, Broadcom Corporation
VP of Technology and Senior VP, DMTF

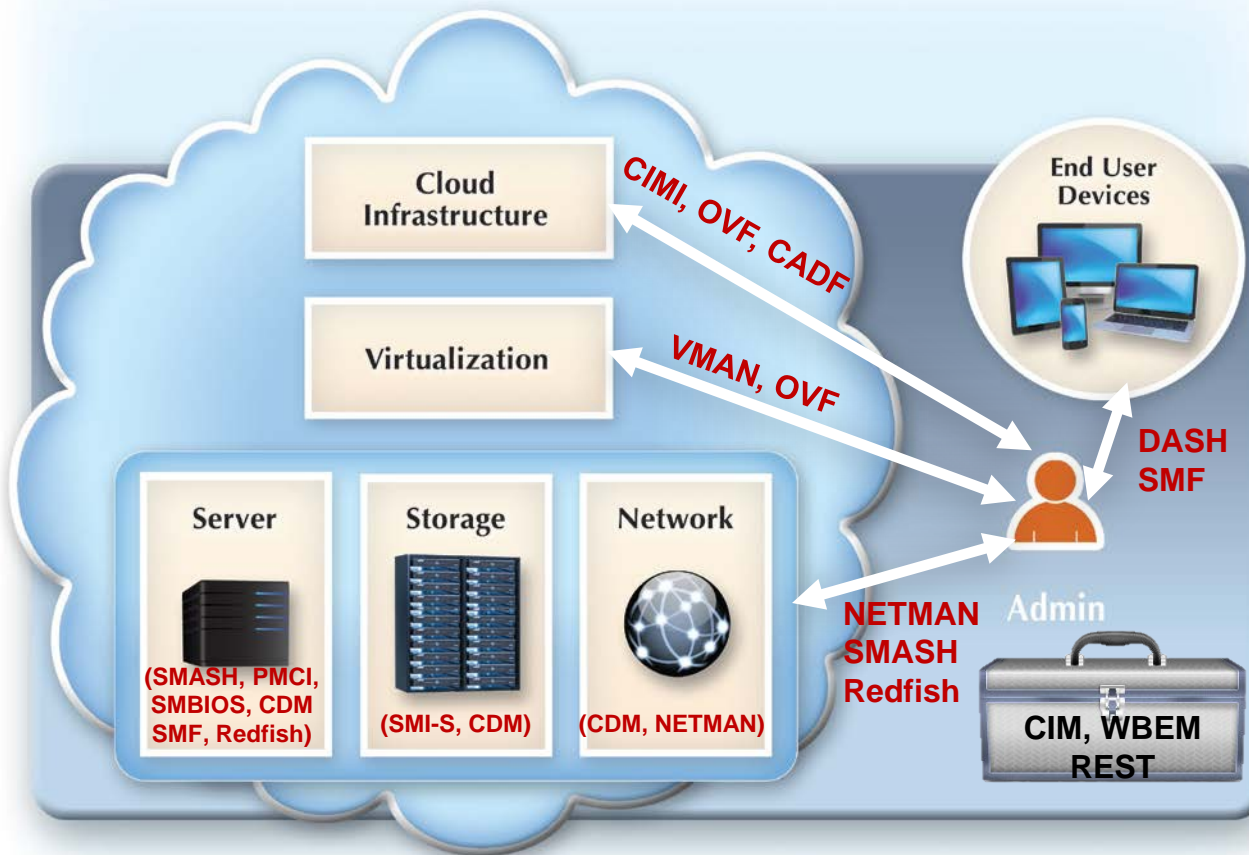
Other Contributors:

Enrico, John Leung, John Parchem, Bhumip, George, Peter, et al.

Agenda

- DMTF Technologies
- DMTF Common Information Model (CIM) Overview
- CIM based models/interfaces: NETMAN, VMAN, SMASH
- Open Virtualization Format (OVF)
- REST based Interfaces/Models: CIMI, CADF, Redfish
- ETSI-NFV Infrastructure Management Mapping
- Proposal to Move Forward

DMTF Management Technologies



- ### Infrastructure Management
- Cloud
 - Virtualization
 - Data Center

- ### Platform Management
- Server & Network
 - Storage (SNIA)
 - Desktop & Mobile

- ### Services Management
- Network services
 - Software Entitlement
 - Security & audit

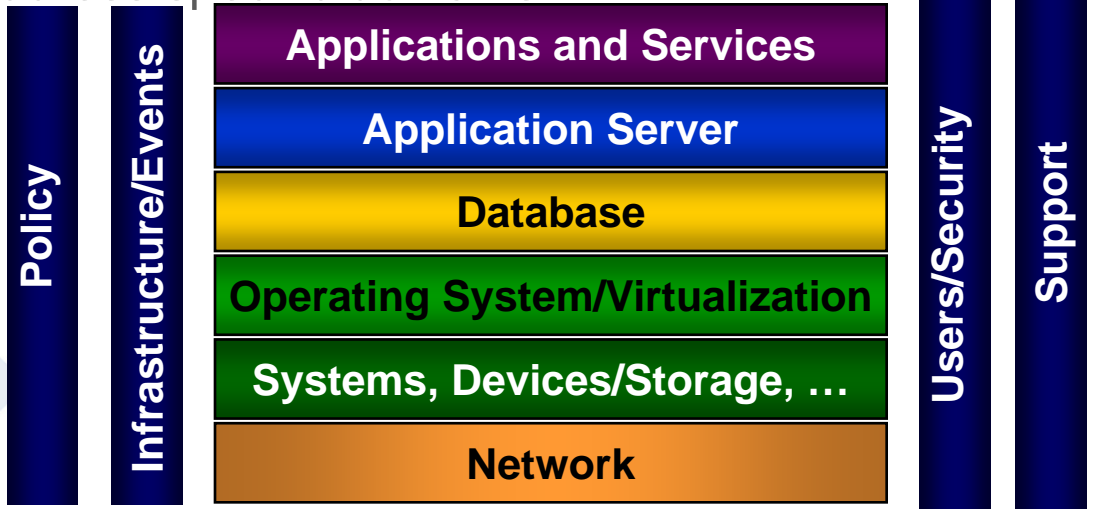
- ### Protocols & Data Models
- WS-Man/CIM-XML
 - REST/JSON/OData
 - CIM & Diagnostics
 - PLDM/MCTP



- Common Information Model
 - Used by multiple orgs (e.g. SNIA SMI-S models for storage)
- Core Specification
 - “Meta”-model, high level concepts and language definitions
- “Core” and “Common” Models
 - Core Model contains info applicable to all mgmt domains
 - Common Models address specific domains

- Application
- Database
- Device
- Event
- Interop
- Metrics
- Network
- Policy
- Systems
- User

CIM



- Subclass from the Core Model



- **Model in terms of objects**
 - An object is an abstraction, consisting of related data/behaviors
 - An object is a named entity that has a set of characteristics (properties and methods), behavior, and a unique identity (Keys)
- **Hierarchical Model**
 - Single inheritance model for objects
 - Inheritance hierarchies refine and specialize the attributes and behavior of a group of objects
 - Association hierarchies relate objects to each other
- **Associations**
 - Classes/Objects that can have properties and methods
 - Inherit semantics, properties and behavior from super classes
 - Can generate events/indications
- **CIM Representation**
 - MOF - Managed Object Format (ASCII or Unicode)
 - VISIO for UML (Unified Modeling Language)
- **A CIM profile is a specification that defines the CIM model and associated behavior for a management domain**



DMTF Managed Object Format (MOF)

- A text based schema language capable of representing UML features, including:
 - Classes
 - Associations (both as references and as association classes)
 - Structures
 - Instances
 - Ability to fully annotate the schema with both information and constraints
- A representation of the CIM Schema is published in MOF

```

[Abstract, Description (
    "An abstraction or emulation of a hardware entity, that may "
    "or may not be Realized in physical hardware. . . . ") ]
class CIM_LogicalDevice : CIM_LogicalElement
{
    . . . .
    [Key, MaxLen (64), Description (
        "An address or other identifying information to uniquely "
        "name the LogicalDevice.") ]
    string DeviceID;
    [Description (
        "Boolean indicating that the Device can be power "
        "managed. . . .") ]
    boolean PowerManagementSupported;
    [Description (
        "Requests that the LogicalDevice be enabled (\\"Enabled\\" "
        "input parameter = TRUE) or disabled (= FALSE). . . .") ]
    uint32 EnableDevice([IN] boolean Enabled);
    . . . .
};
    
```

Qualifiers (points to [Abstract, Description])

Class Name and Inheritance (points to class CIM_LogicalDevice : CIM_LogicalElement)

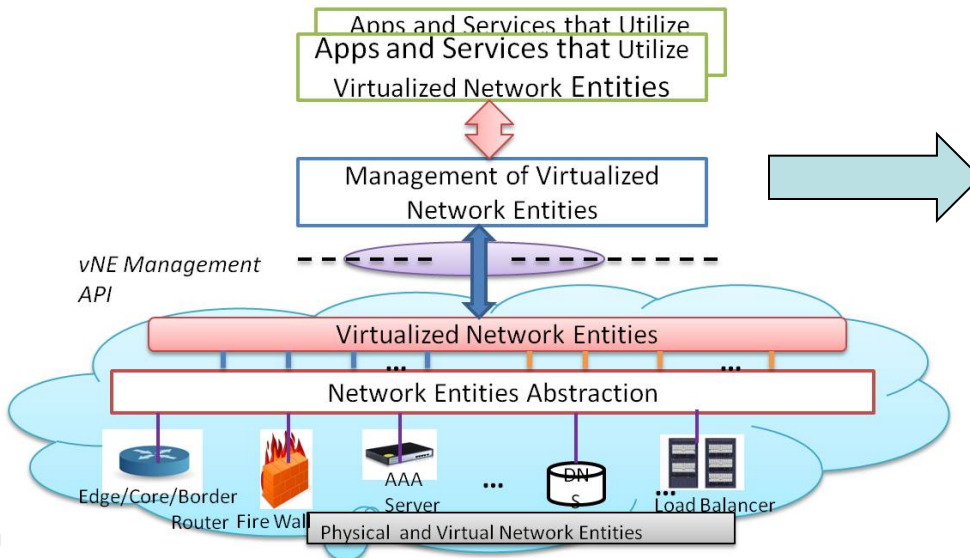
Properties (points to string DeviceID; and boolean PowerManagementSupported;)

Methods (points to uint32 EnableDevice([IN] boolean Enabled);)

- A set of specifications published by DMTF
- Defines how resources modeled using CIM can be discovered, accessed and manipulated.
- Provides the ability for the industry to deliver a well-integrated set of standard-based management tools
- Facilitates the exchange of data across otherwise disparate technologies and platforms
- Protocols
 - CIM-XML – CIM Operations over HTTP(S)
 - WS-Management - SOAP/XML over HTTP(S)
 - CIM-RS – Restful protocol (JSON/XML) over HTTP(S)



Network Management Profiles: Provide abstraction and management of Network entities (VNF resources and services)



Leverage protocols, data models, infrastructure created for managing

- Server
- Storage (SNIA)
- Desktop & Mobile
- Virtualization

to perform

- Centralized or distributed Network Management
- And Network Policy Management

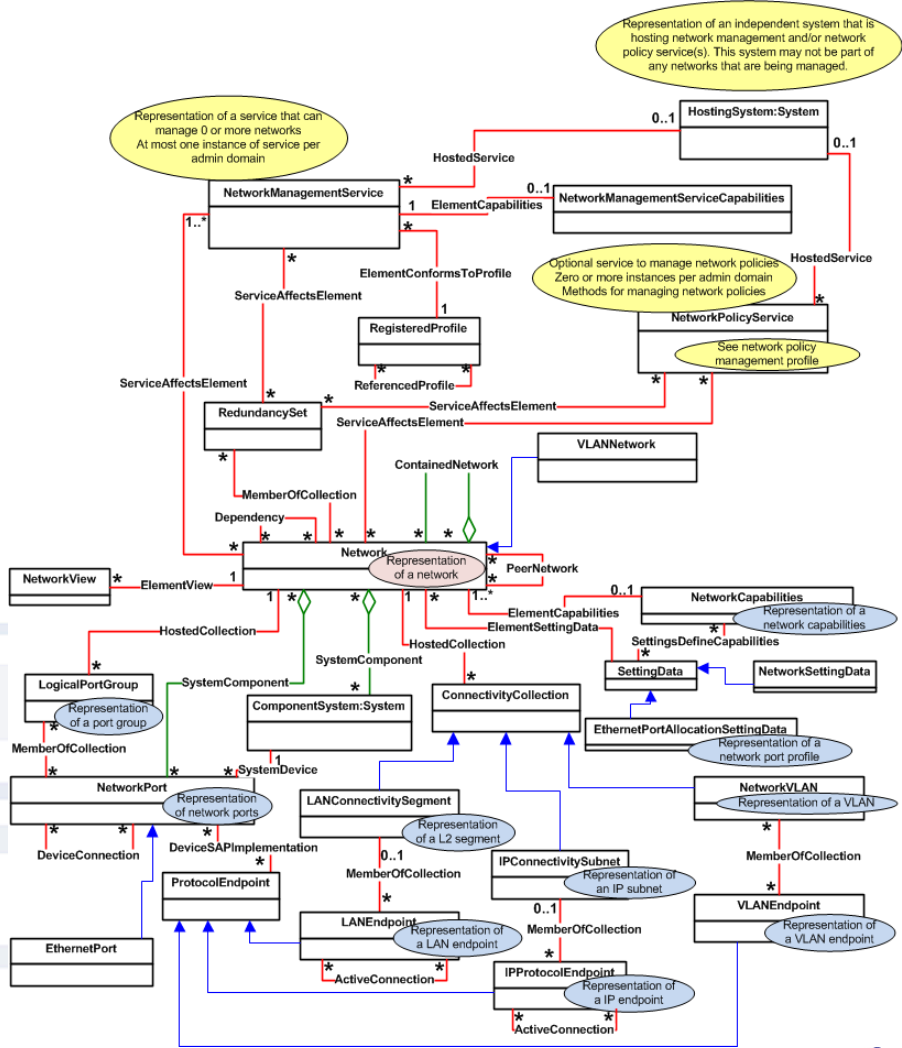
Thus

Unify compute, storage, and network management domains

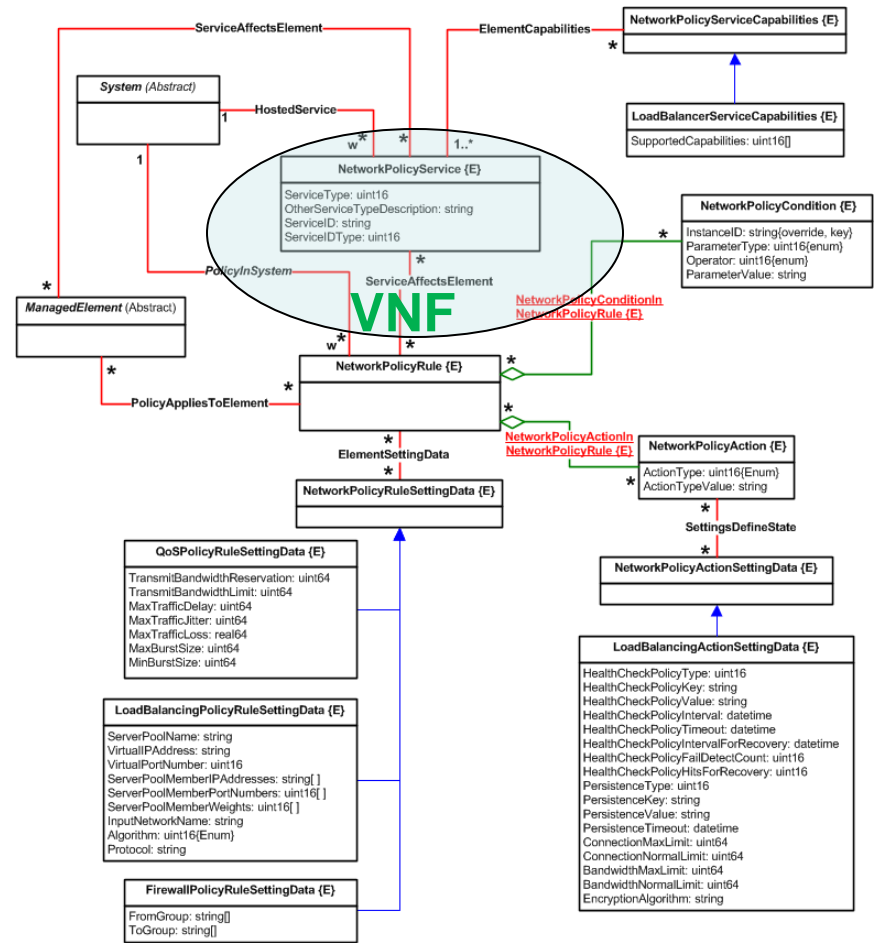
Network topology discovery
 Network capabilities discovery
 Network monitoring and statistics collection
 Network configuration and control
 Network view (a snapshot of network)
 Network resources (ports, endpoints, etc.) inventory
 Network resources configuration and control
 Network services management



Network Management Model (VNF management resources)



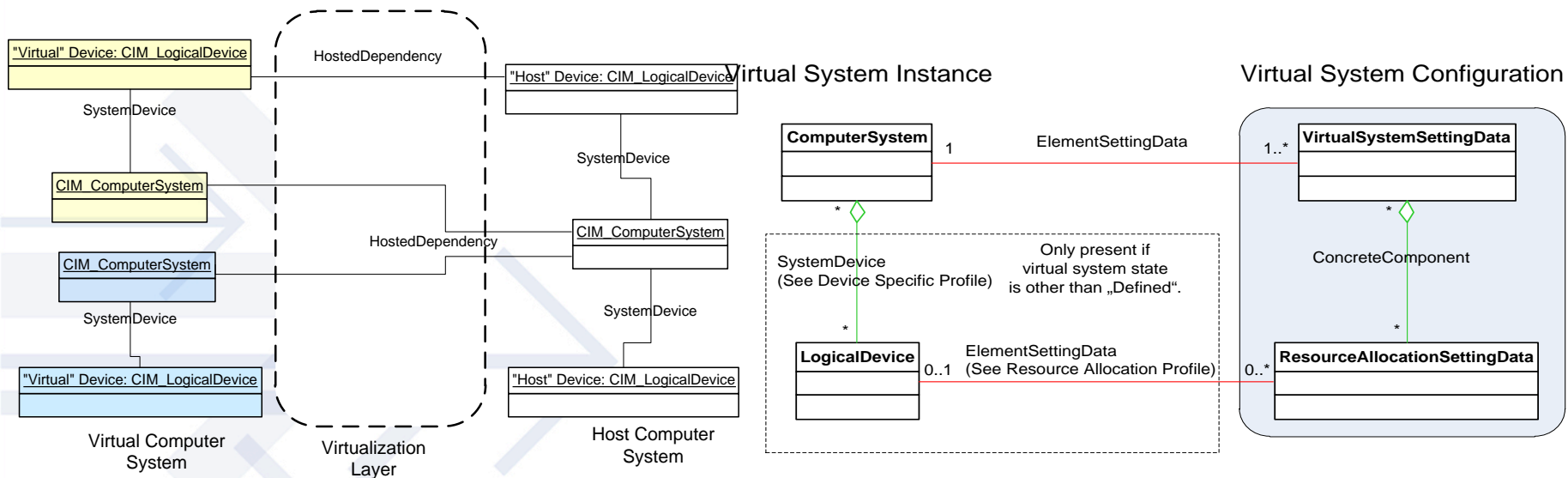
Network Policy Service Model (based on generic policy profile)



Virtualization Management (VMAN)



- Addresses the management lifecycle of a virtual environment
- VMAN's CIM profiles standardize many aspects of the operational management of a heterogeneous virtualized environment
 - Supports creation, modification, deletion and inventory of virtual resources
 - Enable mapping of virtual resources to underlying resources
- VMAN has been adopted and published by the American National Standard Institute (ANSI) International Committee for Information Technology Standards (INCITS) as INCITS 483-2012.



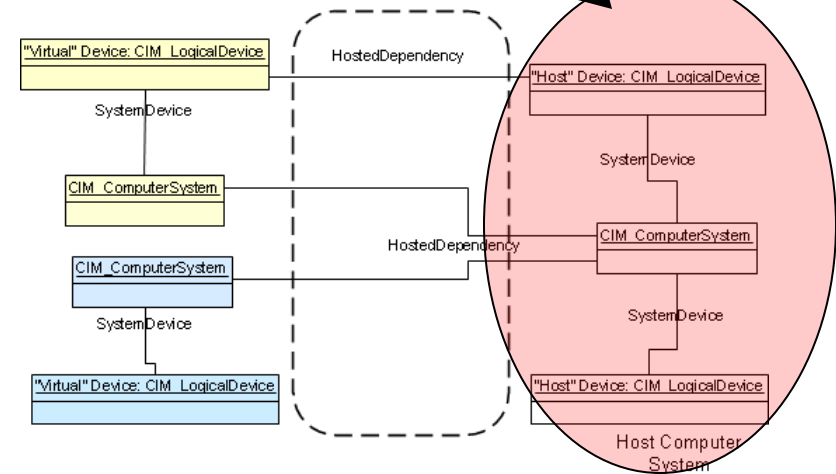
What is SMASH?



- Systems Management Architecture for Server Hardware
 - Allows to manage computer systems used to host virtualized environments
 - Add a diagram

- A suite of specifications

- Industry standard protocols/profiles
- Unifies the management of servers
- Vendor independent
- Platform neutral
- Independent of machine state



- SMASH specifications utilize the CIM data model and industry standard transports/security mechanisms

- Align out-of-service with in-service manageability.
- Align in-band with out-of-band manageability.

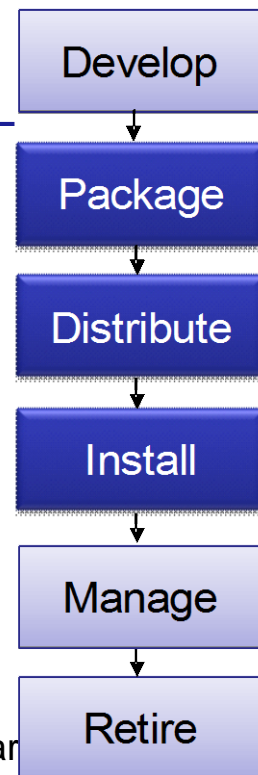
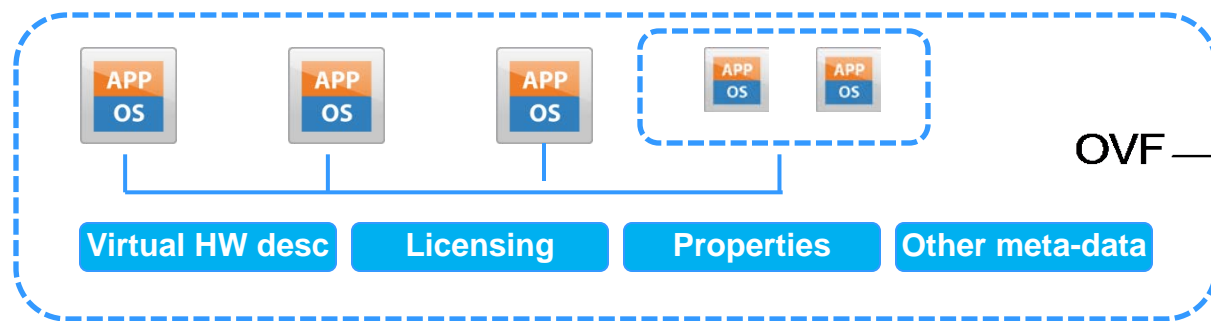
- 1.0: Dec 2006, 2.0: Sep 2007, 2.1: Dec 2014

- Features categories

- Inventory, Monitoring, Control, Diagnostics, Repair, Alerts



- A **packaging and distribution format** for virtual appliances
- Provides a complete description of a system of VM/multi-VMs in XML



- Vendor and platform independent: Interoperable across platforms
- Extensible: Facilitate value-added features by DMTF or 3rd parties
- DMTF specifications
 - Version 1.1 - Jan 2010 (Virtual machines, Product Information, Licensing, Virtual hardware requirements, Deployment Options, Resource Requirements...)
 - Version 2.0 – Dec 2012 (Network Port Profiles, Scaling, placement policies, Encryption, Disk sharing, device boot order, advanced data transfer to guest...)
 - Version 2.1 – Dec 2013 (Activation Process, Meta data, Network Policy Service...)
 - ANSI/INCITS 469-2010 - Sept 2010 (national standard)
 - **Version 3.0 – In development (includes VNF packaging)**
 - ISO/IEC 17203 - August 2011 (international standard)

Cloud Infrastructure Management Interface (CIMI)

What is it?

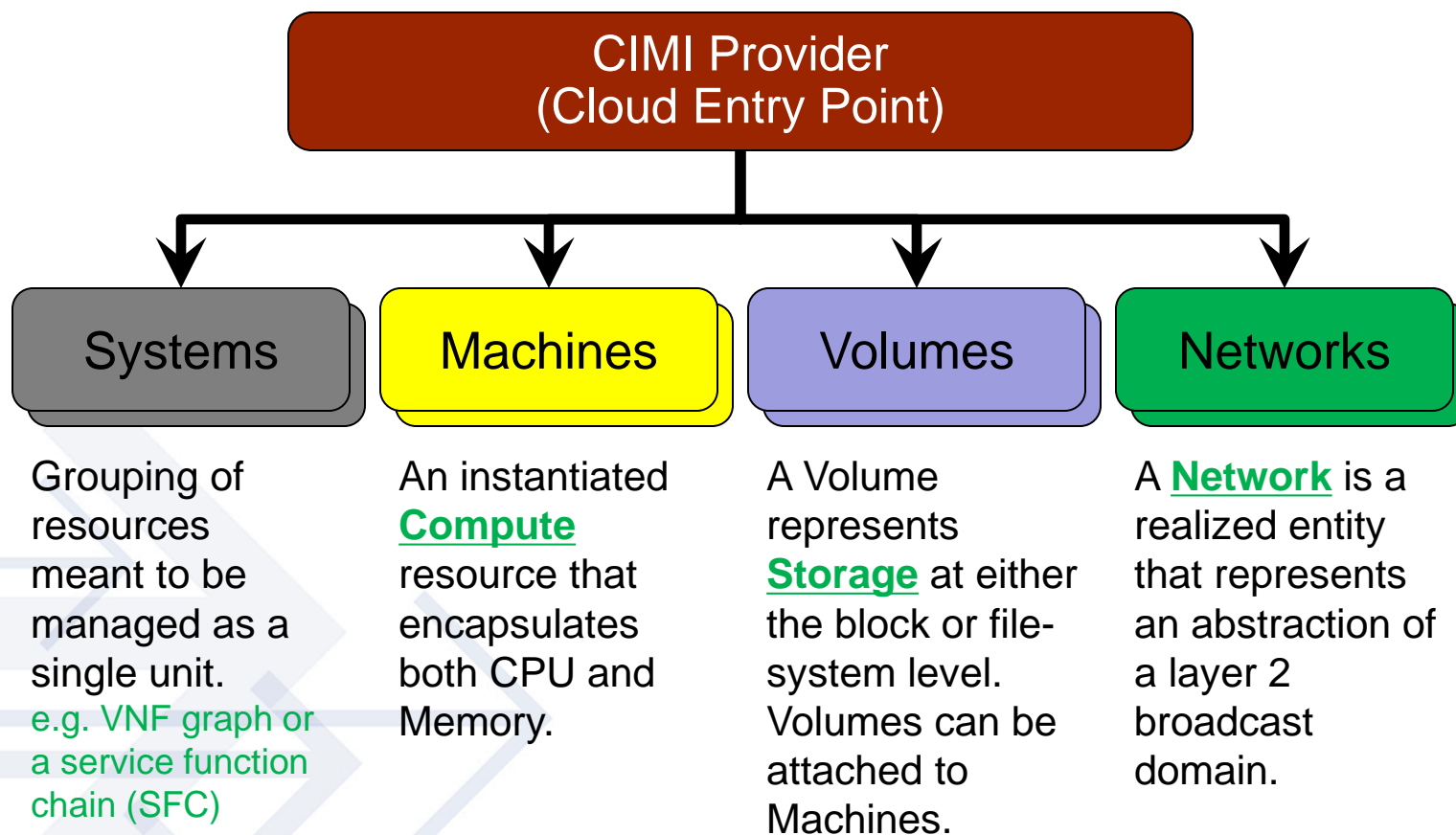
- A Management interface between the cloud service consumer / provider
- Includes a cloud resource model and a REST/HTTP binding to the model
- CIMI is mappable to underlying cloud infrastructures (e.g. OpenStack)

What problems does CIMI solve?

- Cloud customers are using **various interfaces** to manage clouds:
 - EC2, OpenStack Nova, Cloud Stack, Open Nebula, vendor specific
- Each API **involves work to develop, test and maintain**
 - Little to no stability, versioning support, or backward compatibility
- APIs are **under control of specific vendors**, not open standards
- Open Source projects (CloudStack, OpenStack, Eucalyptus) only interoperate if everybody is **using the same code** – no winners here
- Customers need multiple clouds to balance risk and so they must either use only clouds with the same code, or **write multiple adapters** to each cloud

CIMI Orchestration Model

Core Resources



CIMI also supports importing/exporting of workloads using OVF

Cloud Audit Data Federation (CADF)



- Standard for the Federation of Cloud Audit Data
- Data Model with a Normative, Prescriptive Audit Event Data Format
- REST based Interface definitions and a compatible Interaction Model

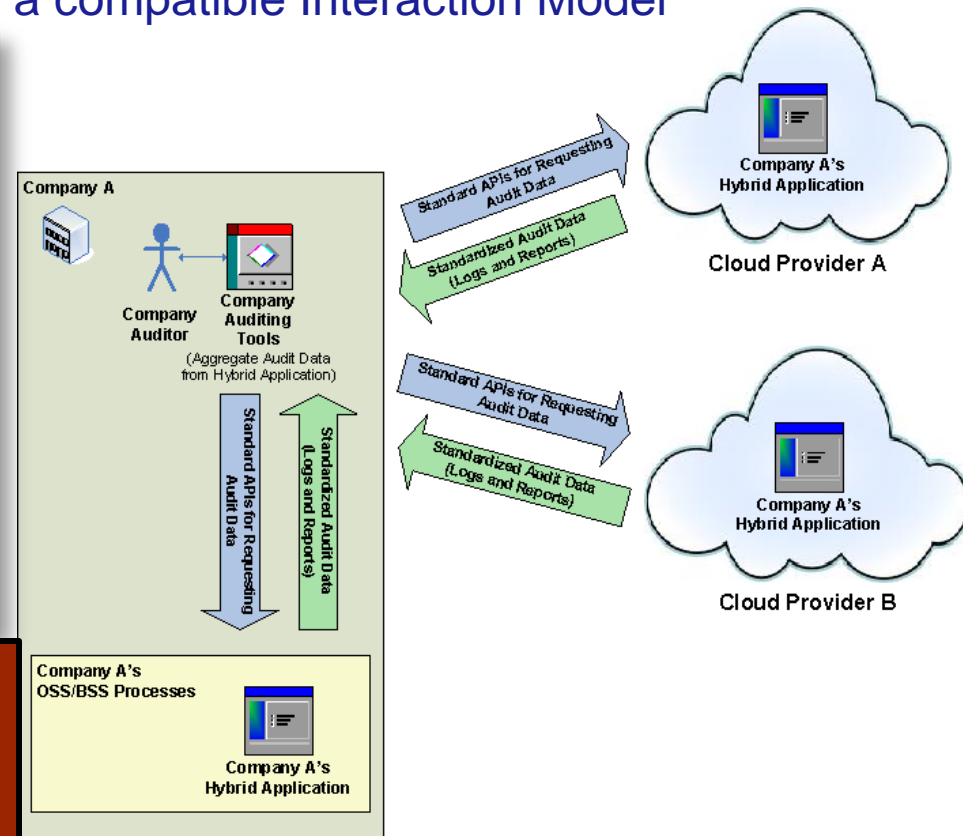
Key Consumers / Audience

Professionals Responsible for Certifying Compliance with IT, Industry, Government, Regional and Corporate Policies

e.g. Auditors, Chief Compliance Officer (CCO), Chief Risk Officer (CRO), Chief Info. Sec. Officer (CISO), CIO, CFO, etc.

Implemented in OpenStack Ceilometer

Demonstrating strength of collaboration between standards and open source.





Redfish Scope and Goals

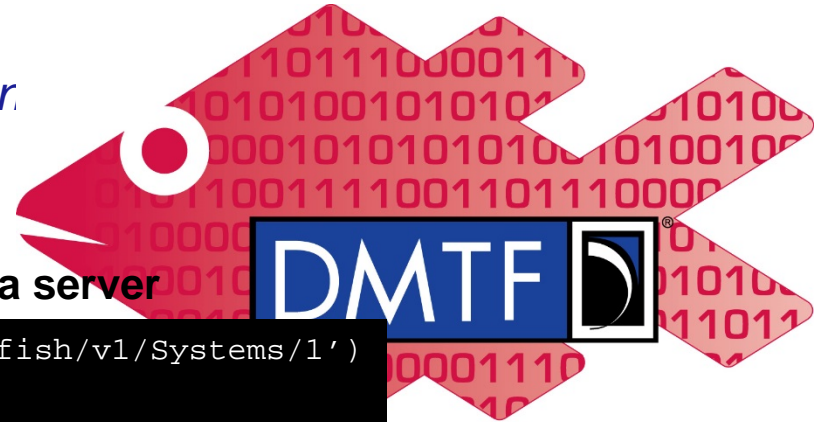
- *An open industry standard specification and schema for simple, modern and secure management of scalable platform HW*

Python code to retrieve serial number from a server

```
rawData = urllib.urlopen('http://<ipaddr>/redfish/v1/Systems/1')  
jsonData = json.loads(rawData)  
print( jsonData['SerialNumber'] )
```

Output:

```
1A87CA442K
```



Redfish

- RESTful interface over HTTPS in JSON format based on OData v4
- Usable by client applications and browser-based GUIs
- A secure, multi-node capable replacement for previous interfaces
- Schema-backed human-readable output
- Covers popular use cases and customer requirements
- Intended to meet OCP Remote Machine Management requirements



Introduction to the Redfish data model

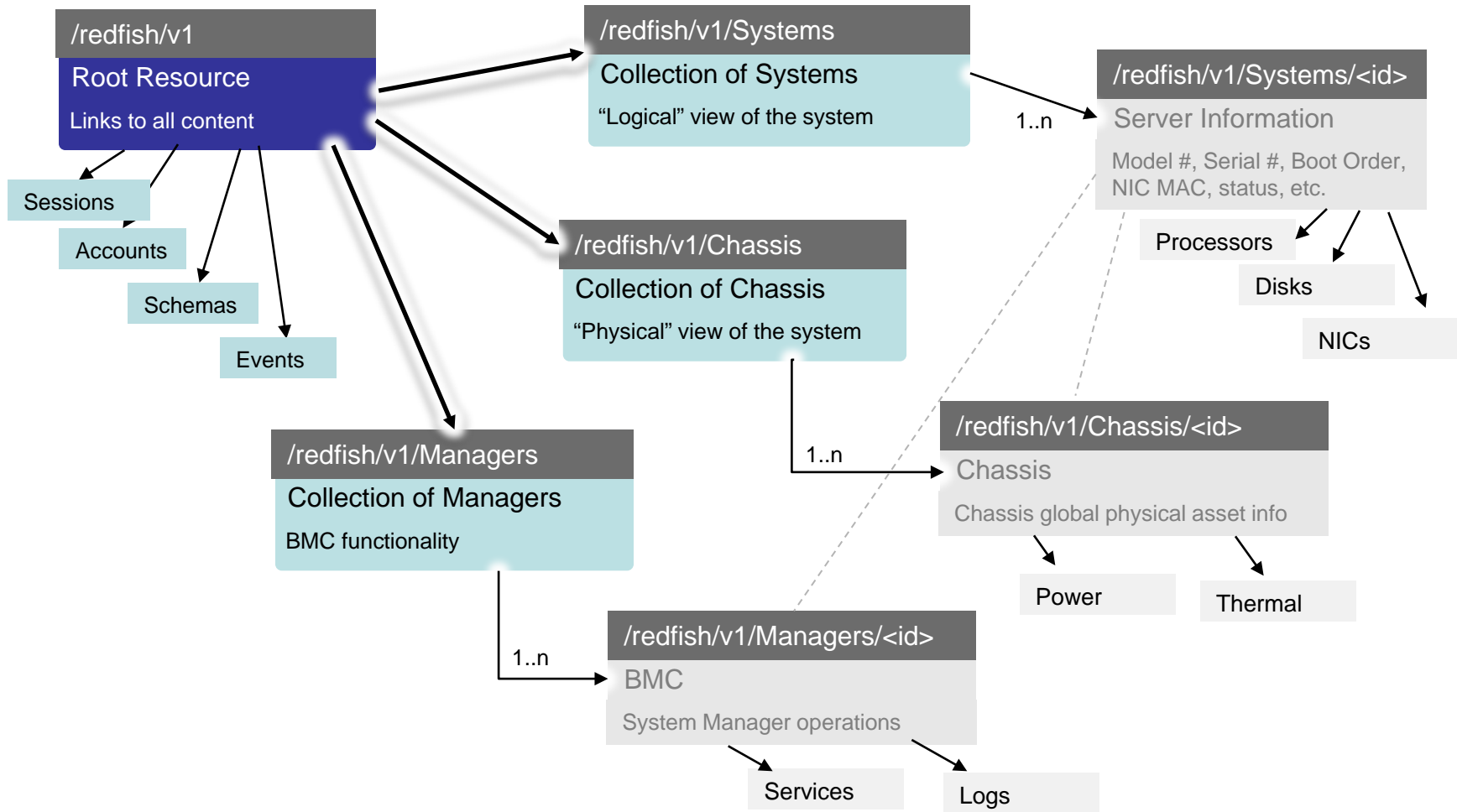
- All resources linked from a Service Entry point (root)
 - Always located at URL: /redfish/v1
- Major resource types structured in 'collections' to allow for standalone, multi-node, or aggregated rack-level systems
 - Additional related resources fan out from members within these collections

Three Main Collections:

- **Systems:** properties expected from an OS console
 - Items needed to run the “computer”
 - Roughly a logical view of a computer system as seen from the OS
- **Chassis:** properties needed to locate the unit with your hands
 - Items needed to identify, install or service the “computer”
 - Roughly a physical view of a computer system as seen by a human
- **Managers:** properties needed to perform administrative functions
 - aka: the systems management subsystem (BMC)



Redfish Resource Map (highlights)





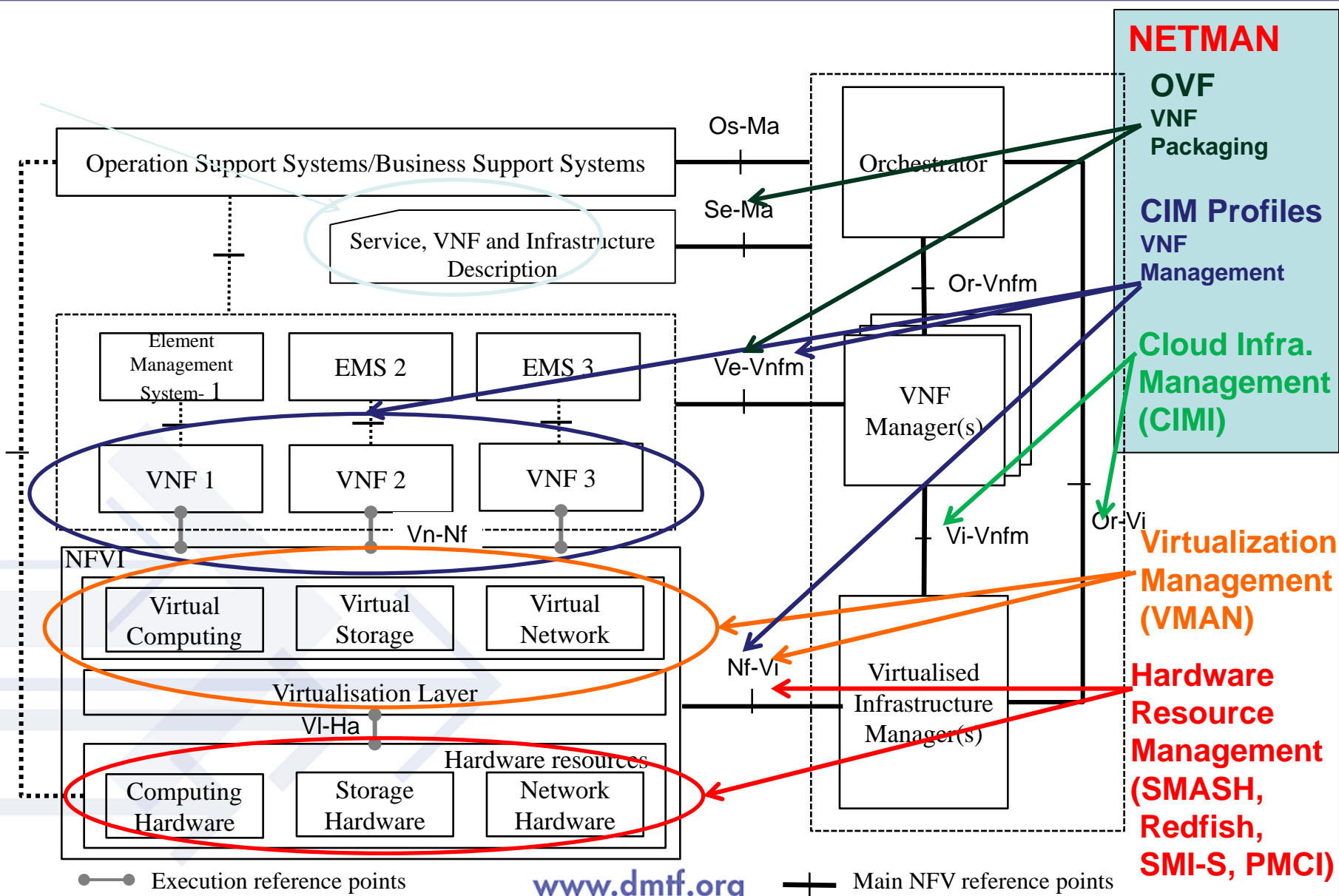
Redfish and CIM

	Redfish (2014-todate)	CIM (1997-todate)
Summary	REST-based semantics for high-level manageability of the platform	Object-based modeling semantics for low-level manageability of the platform & components
Platforms	Multi-node compute	Compute, storage ¹ , network, virtualized environments
Design goals	<ul style="list-style-type: none"> • Simple and intuitive interface • End-user can use without additional knowledge 	<ul style="list-style-type: none"> • Robust object model which can model complex platforms. • Objects manipulated by a client application
Technologies	Protocol: HTTP Encapsulation: JSON Schema: json-schema & OData	<ul style="list-style-type: none"> • Protocols (CIM-XML, web services, RESTful, etc.) • Encapsulation: XML • Schema: CIM (meta-model, MOF)

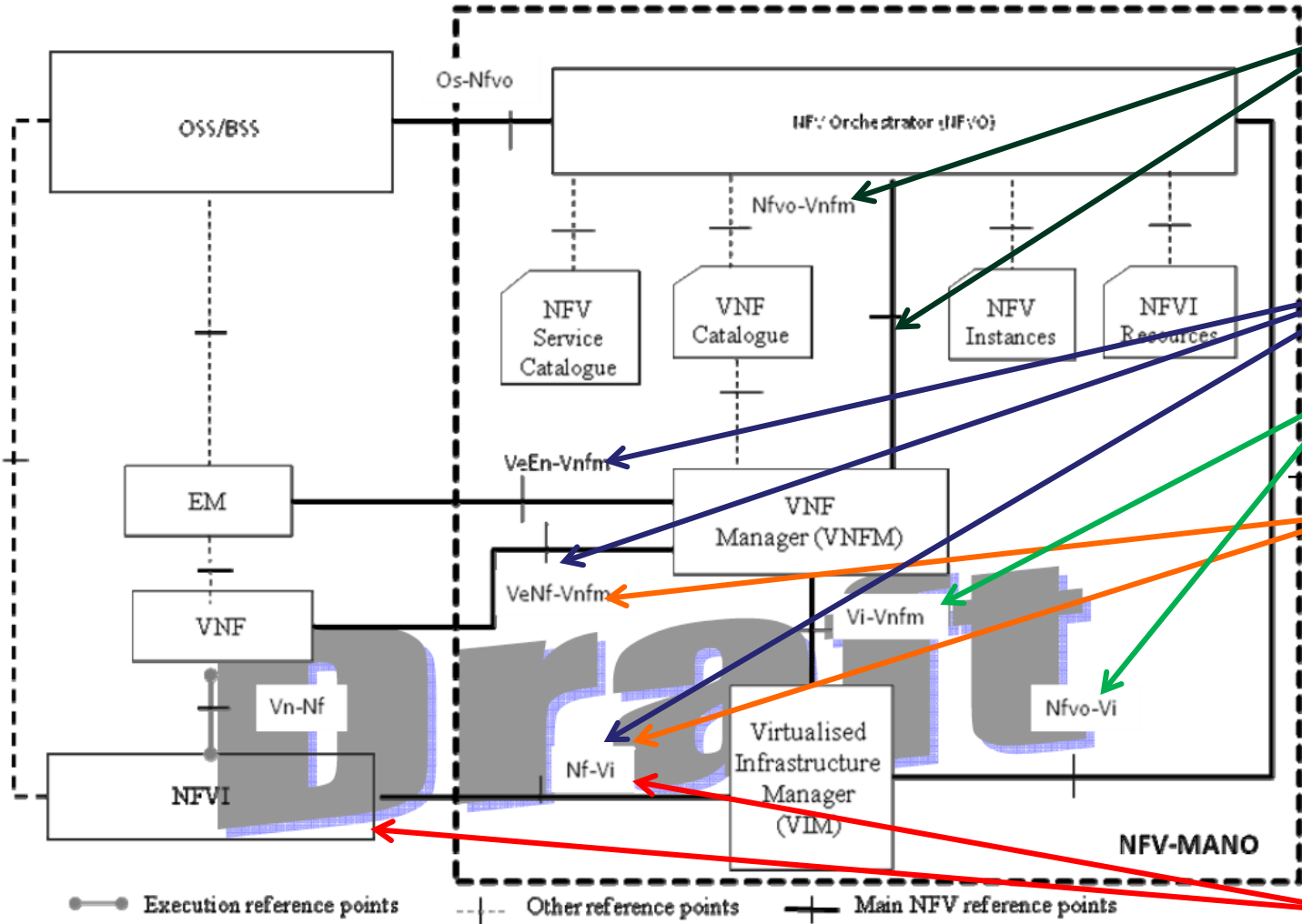
CIM = Common Information Model

¹SNIA (Storage Networking Industry Assoc.)

DMTF Technologies for NFV



NFV MANO Architectural Framework Mapping to DMTF Standards



- NETMAN**
- OVF**
- VNF Packaging**
- CIM Profiles**
- VNF Management**
- Cloud Infra. Management (CIMI)**
- Virtualization Management (VMAN)**
- Hardware Resource Management (SMASH, Redfish, SMI-S, PMCI)**



DMTF Standards are synergistic to NFV MANO Arch

DMTF Standard/Initiative	NFV MANO Functionality	Applicable Ref Points/Artifacts
OVF	Packaging/distribution of VNFs	Nfvo-Vnfm (VNF descriptor usage)
NETMAN	VNF management Network management Network policy/service mgmt	VeEn-Vnfm, VeNf-Vnfm Nf-Vi
CIMI	Virtualized resource capacity/catalog/performance management and orchestration	Vi-Vnfm, Nfvo-Vi
VMAN	Virtualized resource management	Nf-Vi, VeNf-Vnfm
Redfish/SMASH	Physical server infrastructure management	Nf-Vi
SMI-S (SNIA)	Storage management	Nf-Vi
PMCI	Physical platform component intercommunications	

Proposal to move forward

- Leverage DMTF information/data models as basis for NFV infrastructure mgmt
 - Scope of the DMTF is clear: it's all about management
 - Drive specifications through TC and participation
 - Consider bringing work into the DMTF
- Unify and Harmonize multiple views of ETSI-NFV Common Information Model
- Identify gaps (if any) in existing Information/Data Models
- Keep Information/Data Models protocol agnostic
- DMTF recommends Alliance Partners mechanism for work across multi-SDOs
 - DMTF Originated Work
 - Feedback from the DMTF
 - DSP Acquisition
 - Work In Progress Release capability
 - Feedback into the DMTF
 - Alliance Liaison
 - Joint Member (companies that are members of both organizations).
 - The DMTF Technology Adoption Policy
 - The DMTF Feedback Portal
 - Alliance Partner Originated Work
 - Similar mechanisms would speed things along if you wish DMTF input



Thank you