Service Design using SDC - Overlaying Services

Roy Ben Hai, AT&T

Introduction to SDC – Service Design and Creation

- SDC is an ONAP IDE for service designers
- An umbrella for all design-time tools
- Manages a catalog of Resources (VNFs, PNFs, VLs) and Services
- Provides the processes for onboarding new Resources and creating Services
- Creates internal metadata for describing assets which is used by all ONAP components at both design time and run time
- Completely defines the lifecycle of VNFs in the target environment

SDC Major Components

SDC major components:

- **Catalog** - is the repository for Resources (VNFs, PNFs, VLs) and Services
- **Design framework** is used to create and modify Resource and Service definitions in the Catalog
- **Distribution and monitoring** is used to deploy certified assets

Overlaying Services

Common industry design concept:

- Overlay Service using Underlay Services
- Underlay Services - Services that have already been instantiated / deployed
- Overlay Services - Services that use underlay services

Overlaying Services – Framework of thinking

- “Service within a Service”
- Exposing parts of the Underlay services to be used by the Overlay Service
- Configuration to allow the ‘stitching’ between the Overlay and the Underlay

- One example of using Overlaying services is *Port Mirroring*.

Port Mirroring

- Allows copying network packets seen on a switch port to a network monitoring connection on another switch port
- Commonly used for network appliances that require monitoring of network traffic such as an intrusion detection systems, passive probes and more
- NEW – to be introduced in ONAP in the Beijing release

Implementing Port Mirroring – new node types

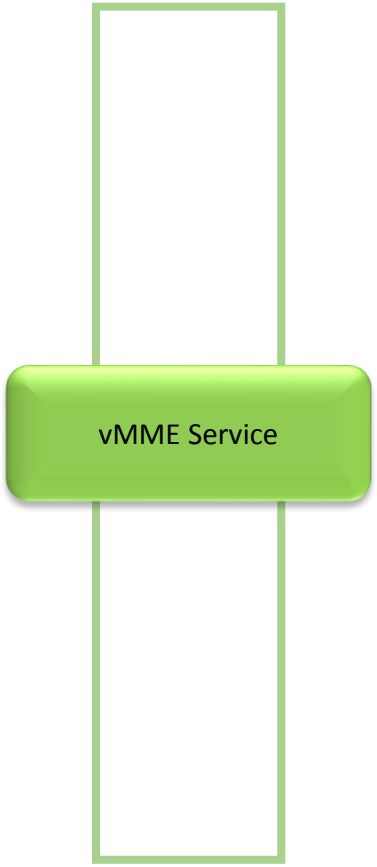
- Service Proxy
  - Represents an existing service in the context of a new service
  - Represents a service which has already been instantiated and exposes the source service's unfulfilled requirements & capabilities
  - Can be generated from an existing service model by dragging the service into the canvas of a new service model
  - Implemented using TOSCA substitution

- PortMirroringConfiguration Resource type
  - a 'PortMirroringConfiguration' resource allows the designer to assign which interfaces (ports) on the Source Service should be assigned to which interfaces (ports) on the Collector Service side.

Implementing Port Mirroring – new Capability type

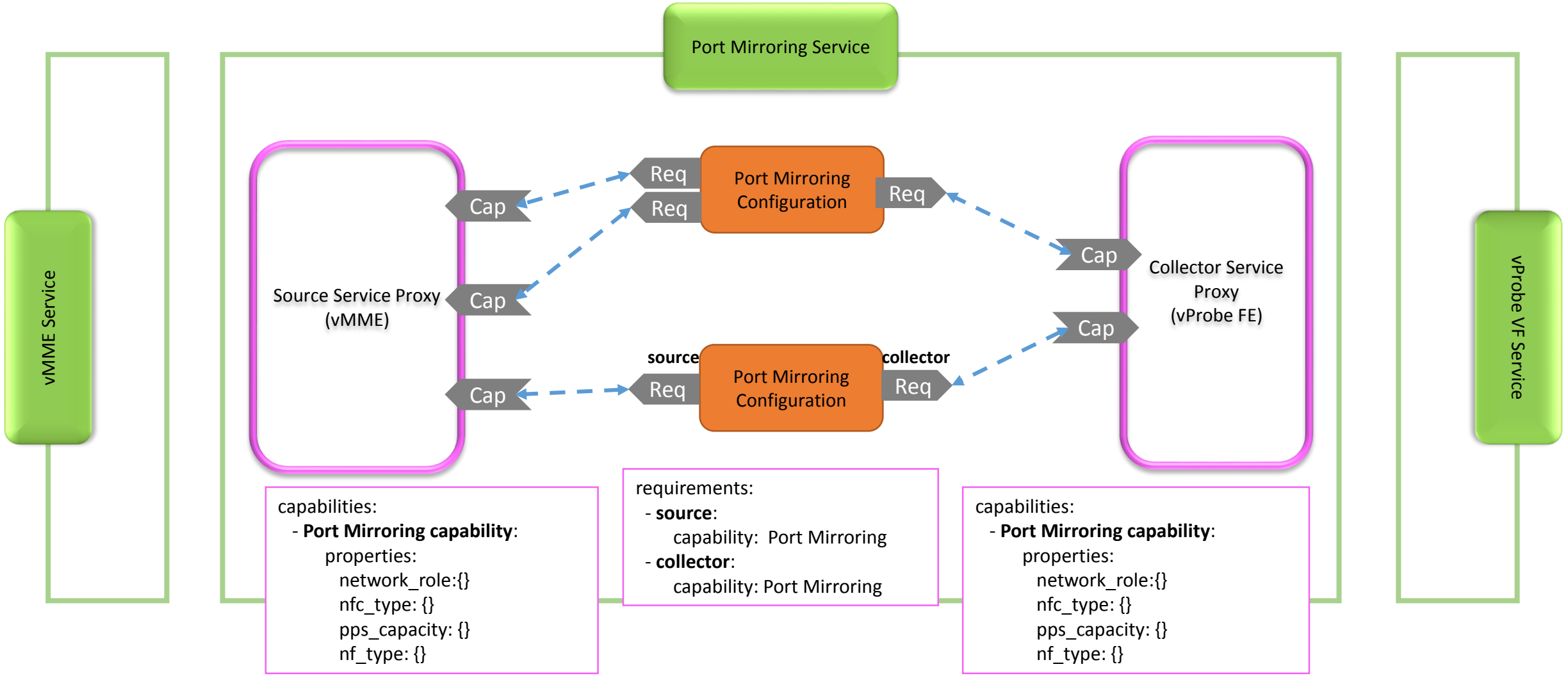
- ‘PortMirroring’ capability
  - When assigned to a port (interface), indicates that the port can take part in port mirroring either as source or as a collector.
  - By default, added to all ports

Port Mirroring – How it works



Port Mirroring – How it works

Port Mirroring – How it works



Demo