From Zero to Orchestrated

Integrating OSS and BSS

Agenda

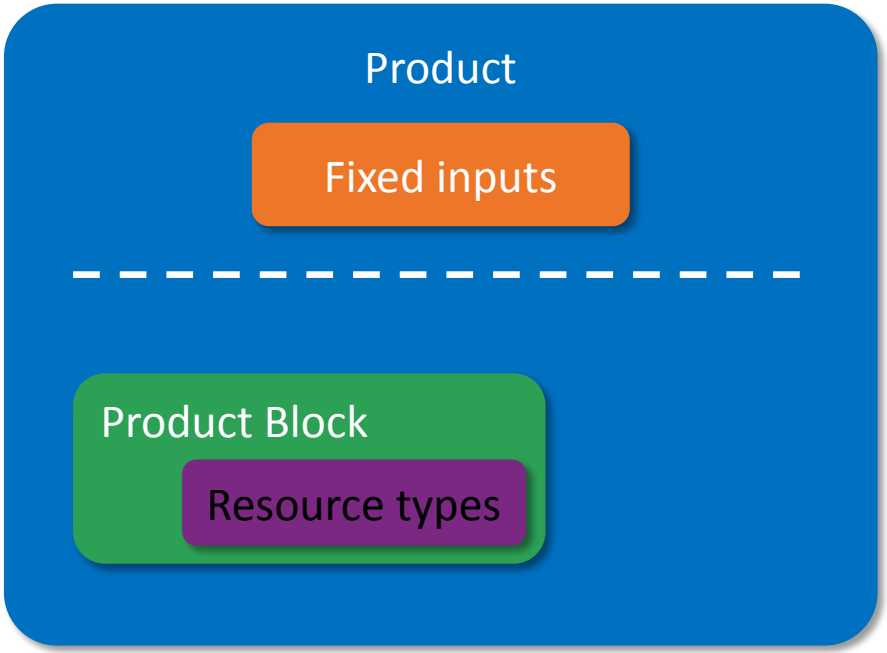
- 8:00 – 8:30 Introduction
- 8:30 – 9:00 Bootstrapping of the development environment
- 9:00 – 9:30 Details of the Orchestrator domain models
- 9:30 – 10:00 Break
- 10:00 – 11:30 Development of your first Orchestrator workflow
- 11:30 – 13:00 Lunch
- **13:00 – 14:30 Integration of OSS and BSS to your workflow**
- 14:30 – 16:00 Tailoring the Orchestrator to your needs (Discussion)

A Quick Recap

- Products describe the logical set of resources that together form a service
- Products contain product blocks and resource types that are populated during workflows, they are different per subscription
- Fixed inputs are like resource types, but are attributes that cannot change during the lifecycle of a subscription
- Workflows are run to create, update, delete and validate subscriptions.
- Subscriptions are a logical set of resources that together describe a valid service for a specific customer
- Domain models describe what a subscription to a product must contain for each lifecycle state and how it can be validated.

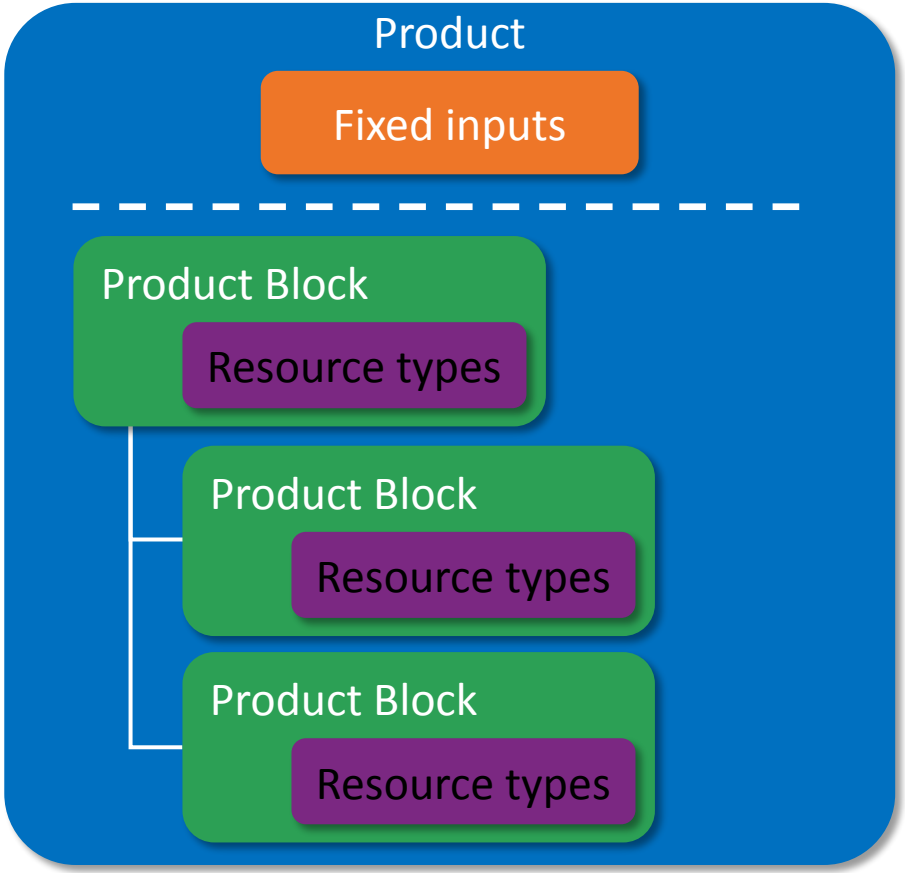
Product Model

- Fixed inputs describes immutable parameters of the product
- Product blocks forms building block of resource types (key value pairs)

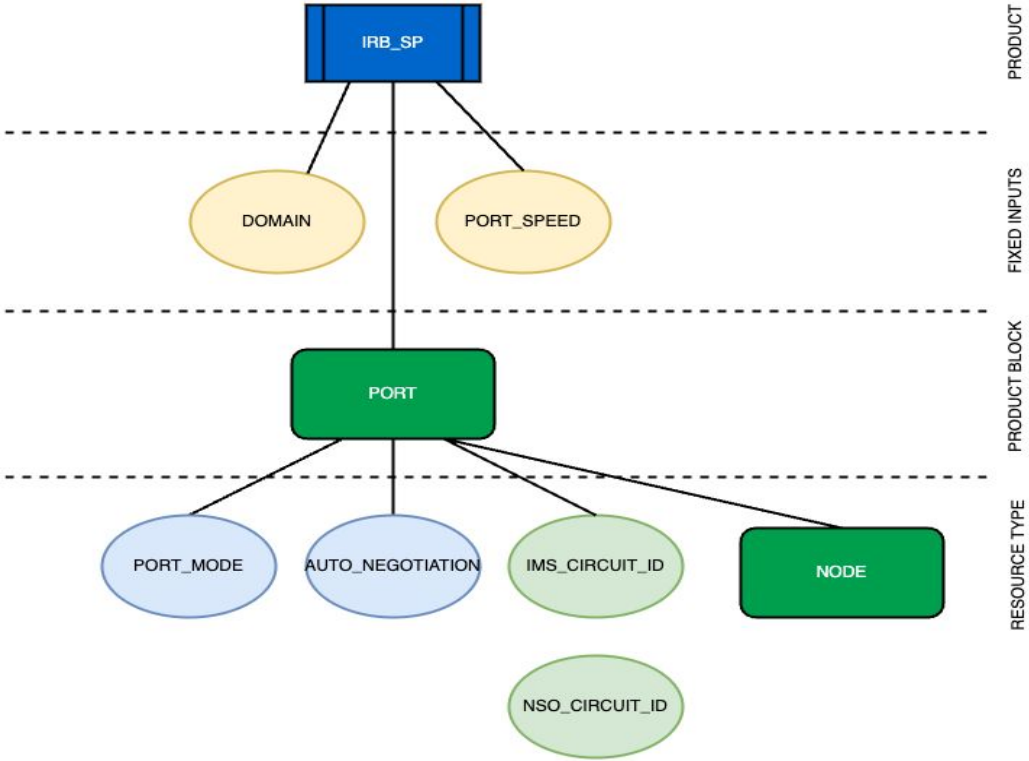


Product Model – little bit more complex

- Fixed inputs describes immutable parameters of the product
- Product blocks are containers for resource types (key/value pairs) and (optional) other product blocks



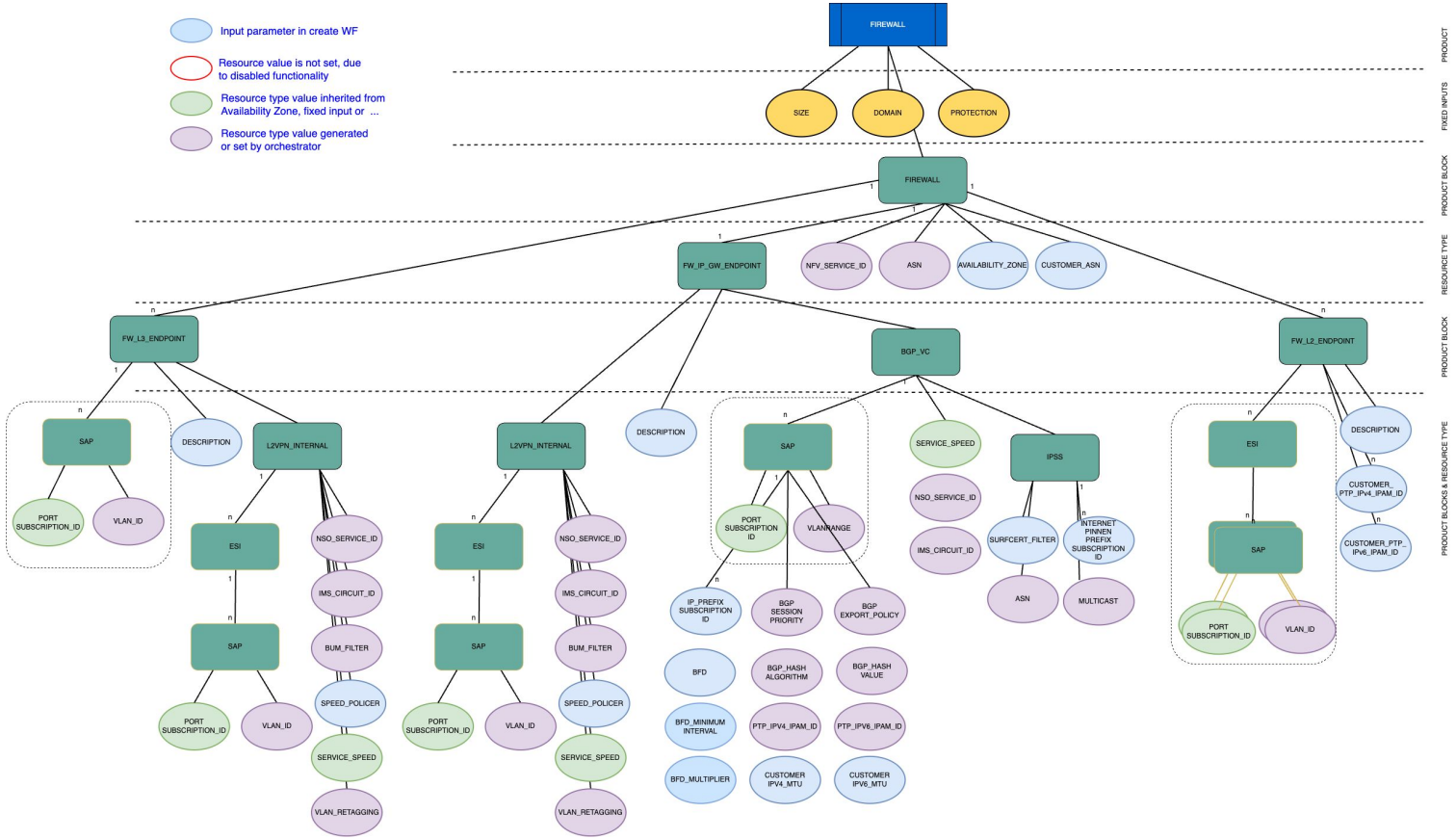
Simple Port Product



Customer facing parameter
- to be filled in by USER

Pointer to OSS/BSS, autogenerated by orchestrator or OSS/BSS

Complex FW Product Model



Questions So Far?

Agenda

- Exercise in creating a process
- Methods of integrating OSS and BSS
- Some more about validator

What methods can you use to integrate sources of truth to achieve automated service delivery?

Are there others in the room who are willing to share some experiences?

- What method(s) have you chosen?
- What success have you had?
- What challenges did you have?
- Where did you define sources of truth?

Two methods: centralised and decentralised

Centralised

- A single point of data entry. A user will enter data at a single point and that will be distributed across all other systems. Most or all validation logic is centralised
- Pros: Very reliable, easy to trust and debug, single pane view
- Cons: Rigid, time consuming to implement, paradigm shift in processes

Decentralised

- Multiple data entry points. A user will enter data in the CMDB, CRM and/or IPAM. Validation logic is handled very close or at the source of truth.
- Pros: Flexible, less of a change in processes, easy to start, less data migration
- Cons: More checks and balances, difficult to control, consistency not guaranteed

Exercise: Create a layer3 circuit for ACME corp. at NREN Wile E. Coyote

In a group of 3 – 4 people, do the following exercise:

- Design an automated process (workflow) through which you could deliver a layer3 circuit for customer ACME corp.
- Include as much of the process as you can: i.e how you can gather requirements for the service, what the automation flow looks like, what resources you need to configure the service (nodes, interfaces, VRF ids etc)?
- Define where your sources of truth live and how they interact to create working network configuration.
- Describe each step in as much detail as possible
- How do you know when the process is finished?
- When do humans interact with the automated process?

Take 20-30 minutes to discuss this and prepare a short pitch about what this would look like.

Centralised: The Orchestrator First Mentality

Integrating OSS and BSS:

- The orchestrator is designed to be in control of all data
- The system works best when all data entry is done whilst running workflows, the benefits include:
- Defining your validations once
- It uses repeatable workflows to translate the data into valid OSS/BSS objects
- It eliminates data entry errors
- This process only allows sane data to enter the system

Integrating with a CMDB

Netbox

- This workshop has chosen netbox to produce an example implementation of an integration with a CMDB
- It models nodes, interfaces, vlans and circuits to make it work.
- When creating a circuit, the user may only select interfaces to create the service if they are valid choice