Rok Arzenšek
rarzensek@nil.com

Management and Orchestration

in Complex and Dynamic Environment

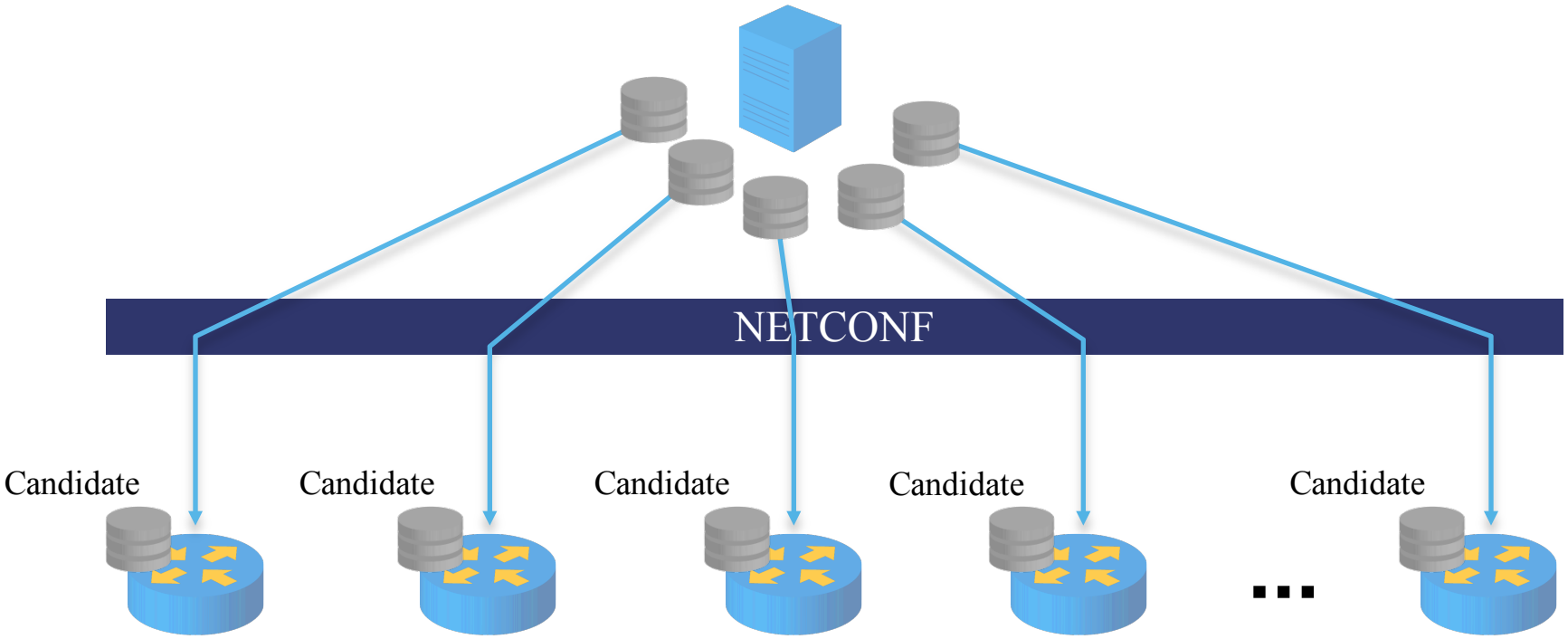
Management and Orchestration Challenges

in Complex and Dynamic Environment

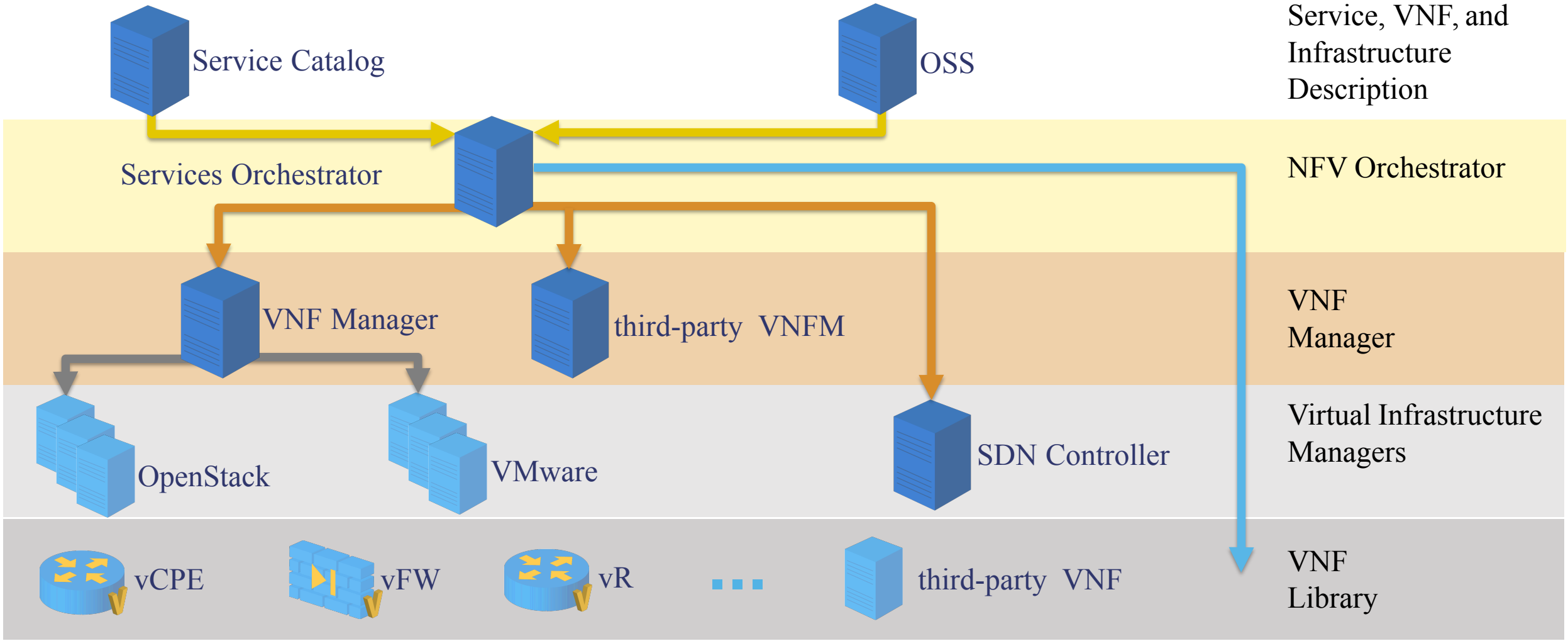
- Long-Lived Transaction
- Scalability and Performance

Network-Wide Transaction in Complex and Dynamic Environment

Network-Wide Transaction



NFV Management and Orchestration



We don‘t know how much time is needed to accomplish a task!

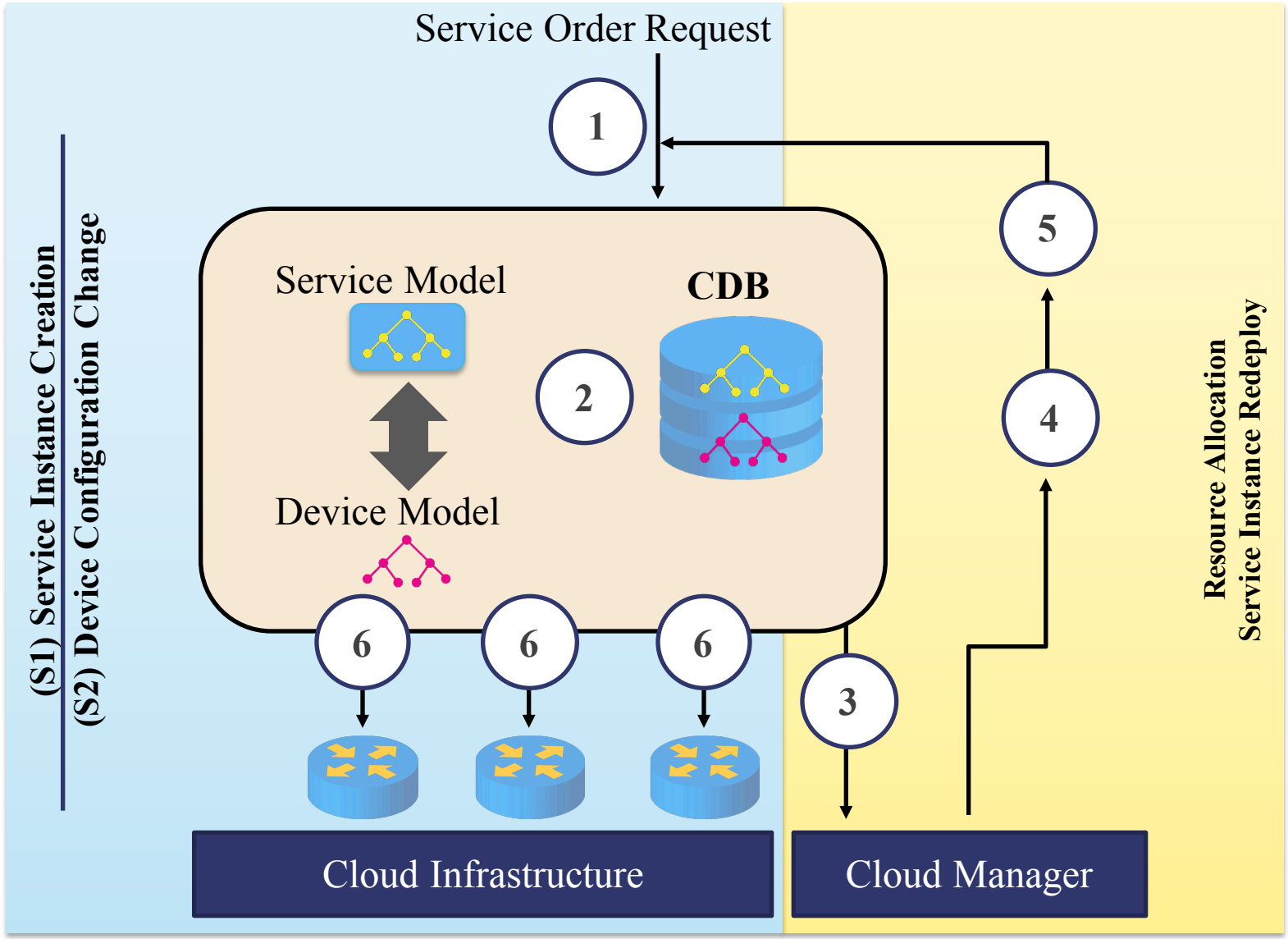
Multi-Stage Transaction

Why Multi-Stage Transaction?

- VIM resource allocation.
- External data resource allocation.
- Devices are not reachable until parts of the service have been set up.
- Respond to changes in the environment and the service parameters.

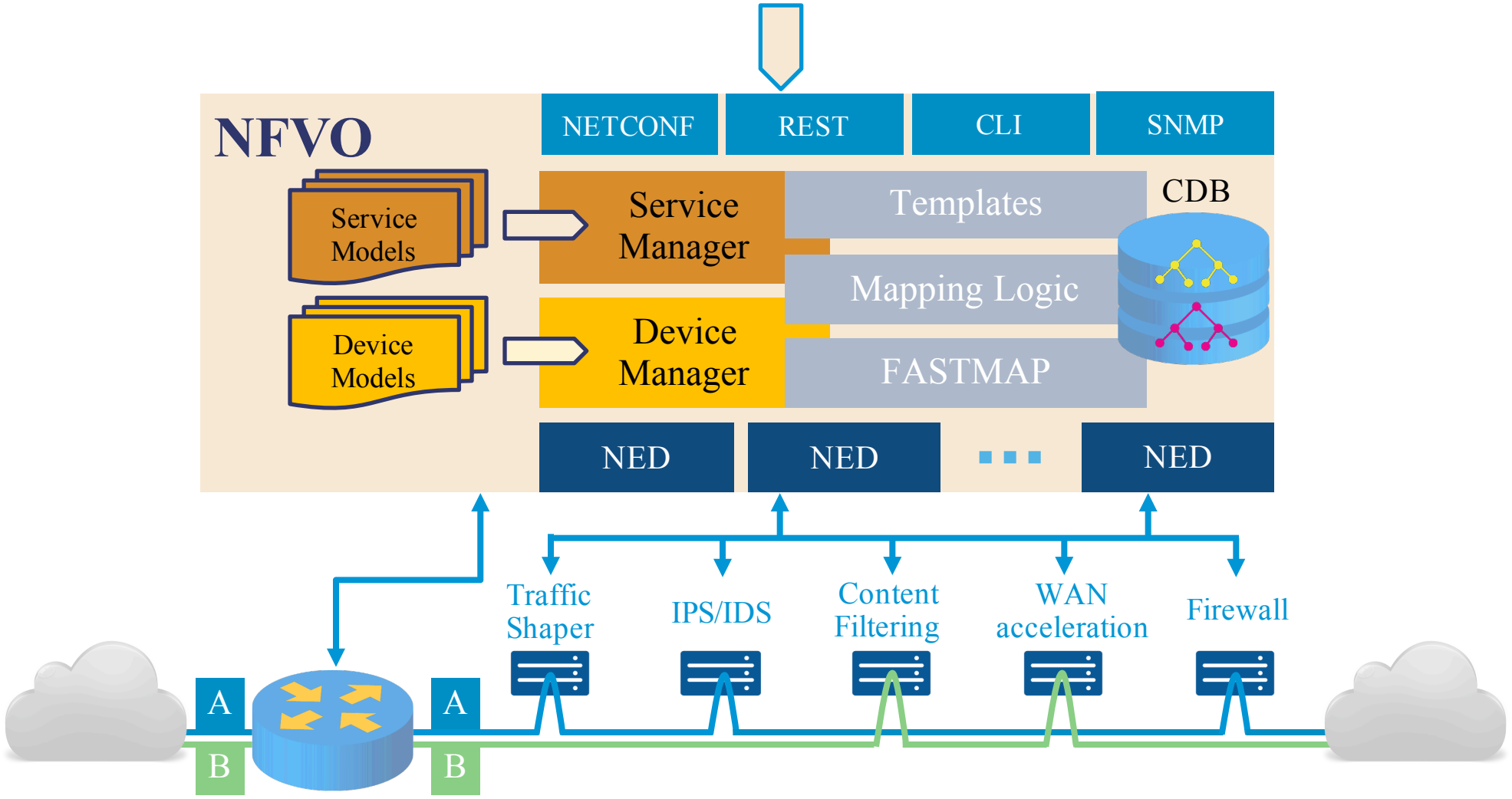
Leave the transaction as soon as possible!

Why Multi-Stage Transaction?

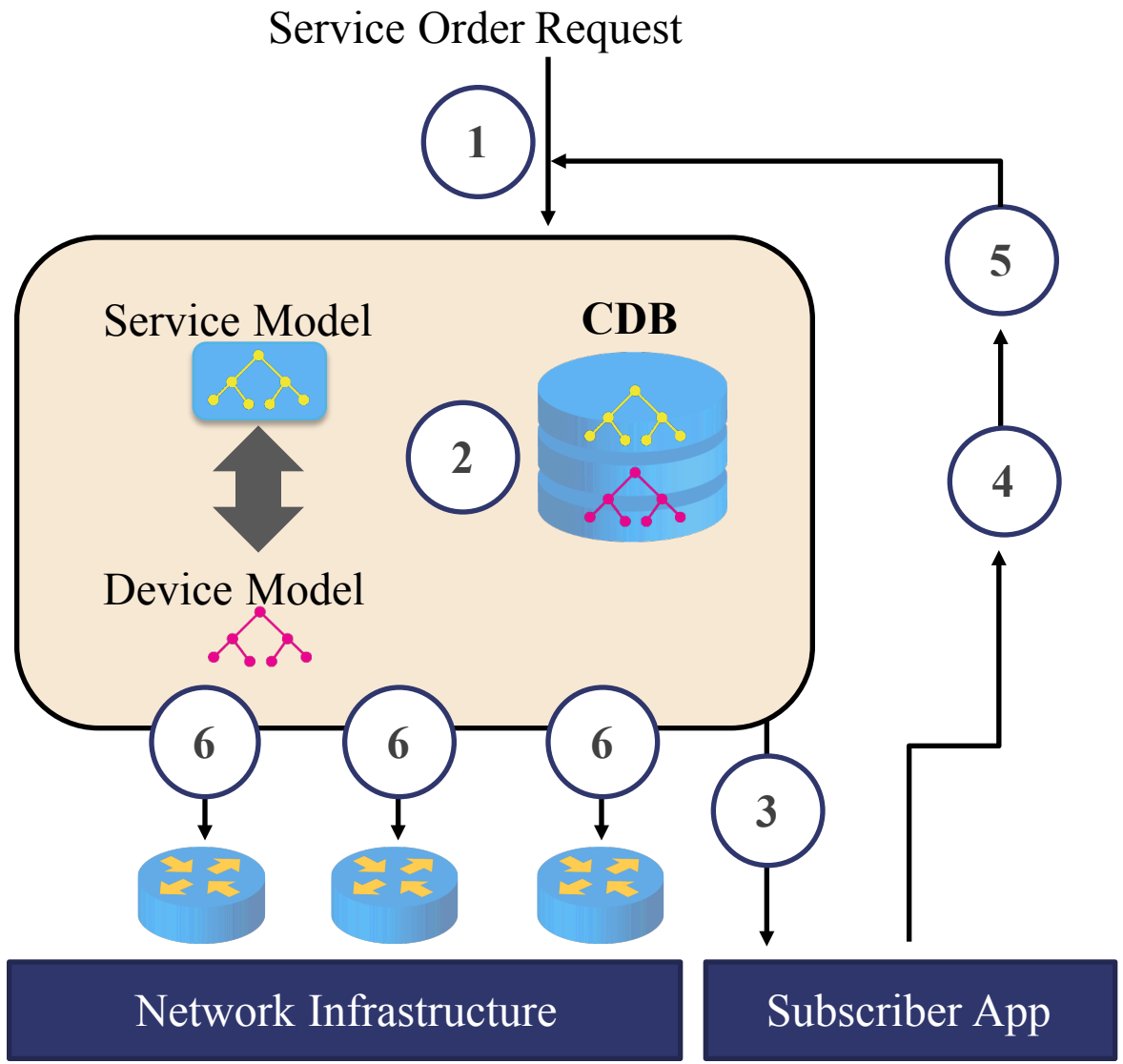


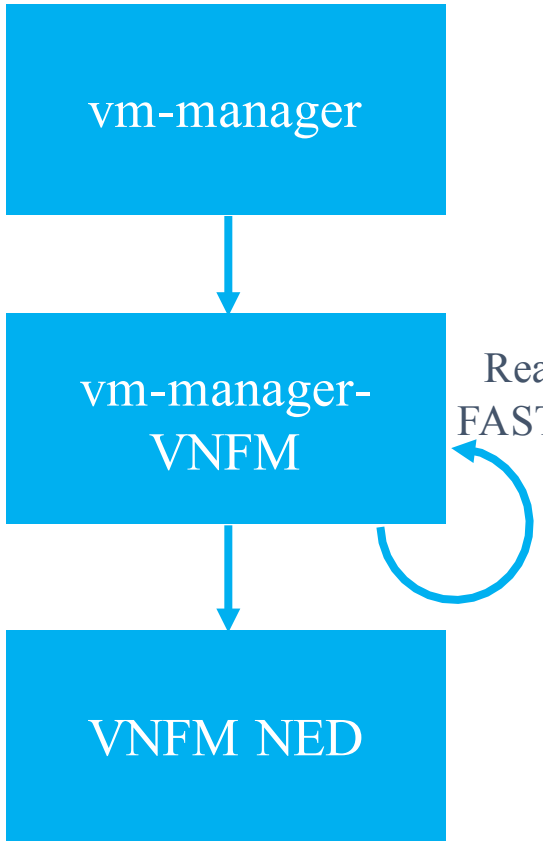
NFVO Architecture

Simple Service Management Interface for Dynamic End-to-End Service Management of Complex Services

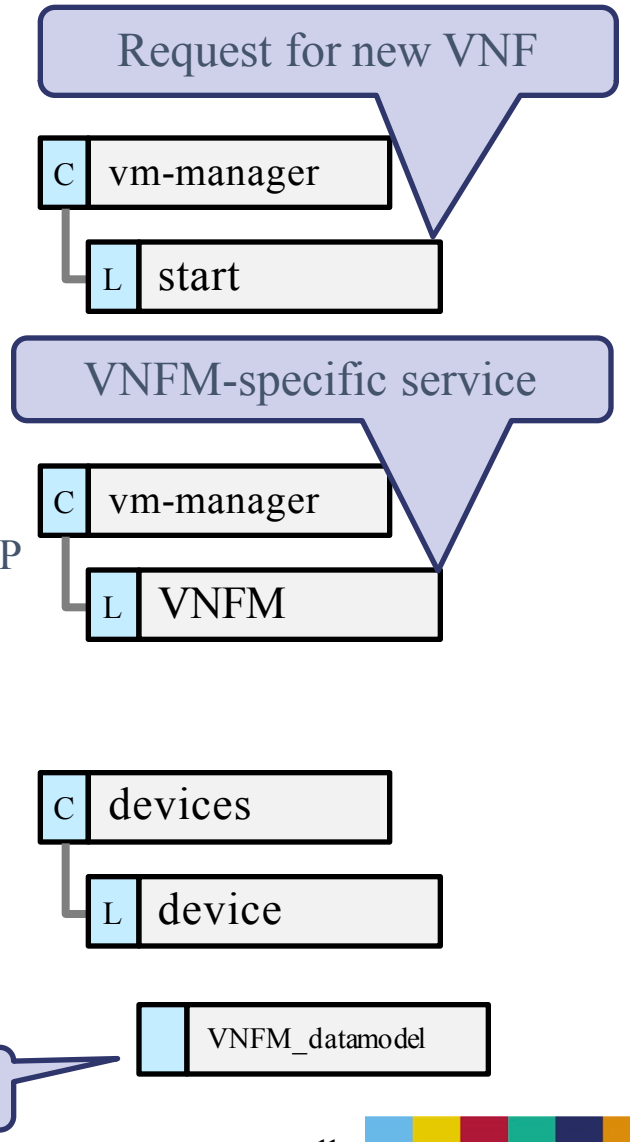


NFVO and VNFM Implementation





Reactive FASTMAP



Multi-Stage Transaction Guidelines

- Spinning up VMs first and configuring them when available.
- When some devices are not reachable until parts of the service have been set up.
- Service automatically responds to changes in the environment (e.g. VM mobility, network connectivity).
- Service automatically responds to changes in service parameters (e.g. service catalog, policy).
- Service takes a long time to activate and you do not want to lock the database.

Scalability and Performance

Addressing Performance Limitations

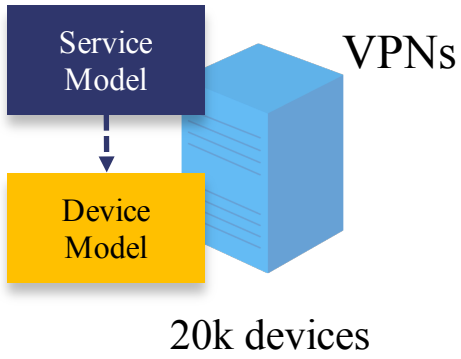
Vertical Scaling

- **Increase capacity:** reached the hard limits
- **Optimize code:** ran out of options
- **Change NFVO behavior:** nothing more to be done

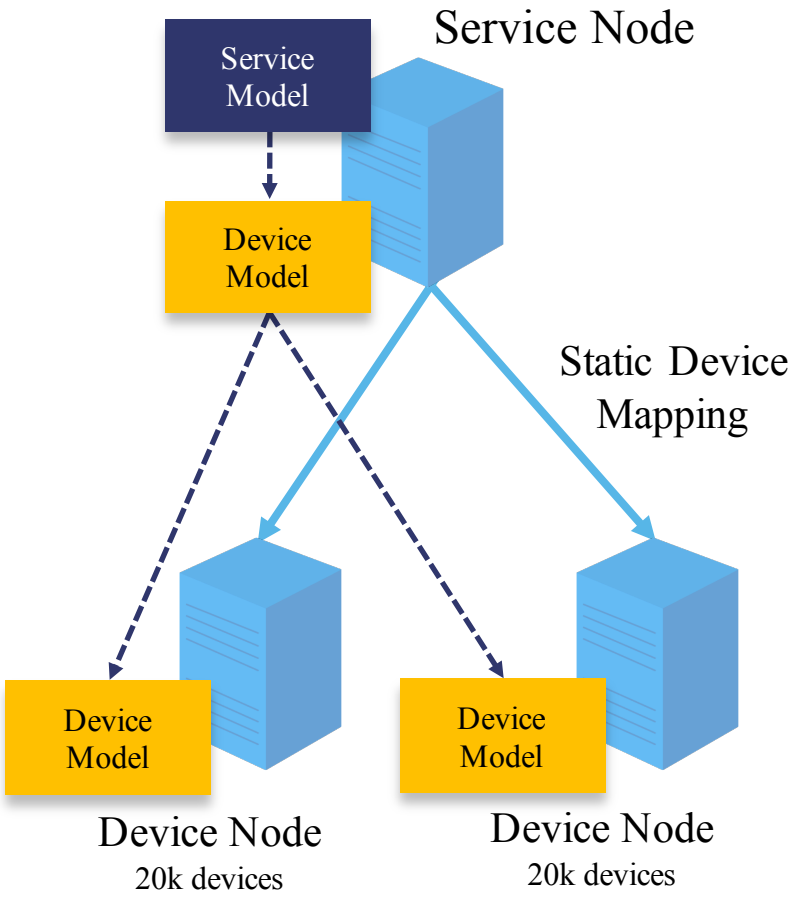
There is only so much you can do!

Design for Scalability and Performance

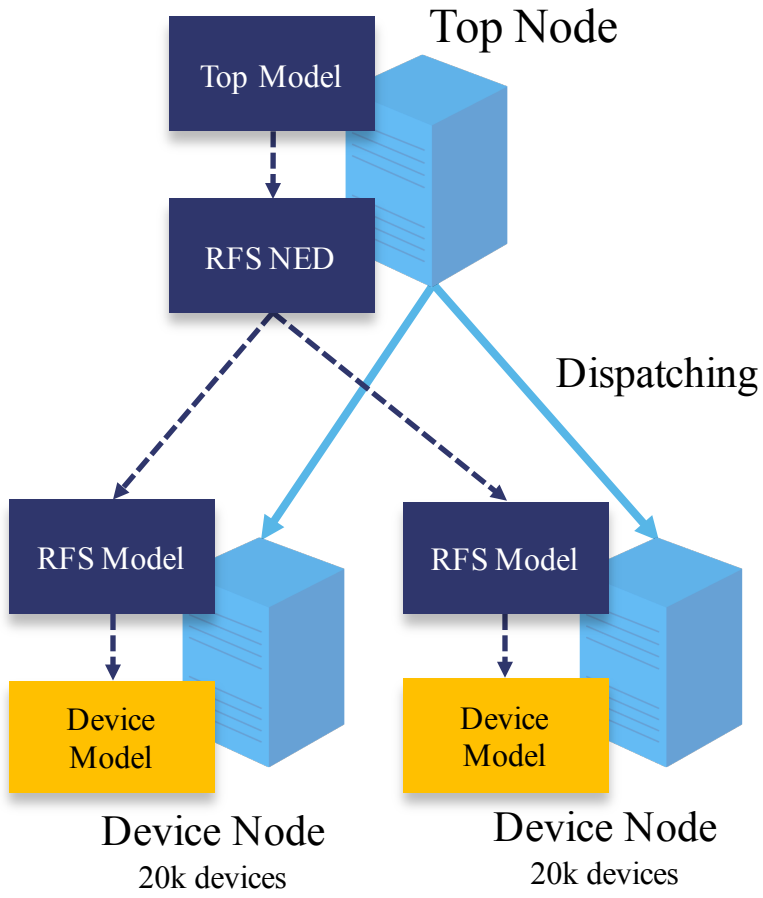
Per-App NFVO



NFVO Cluster

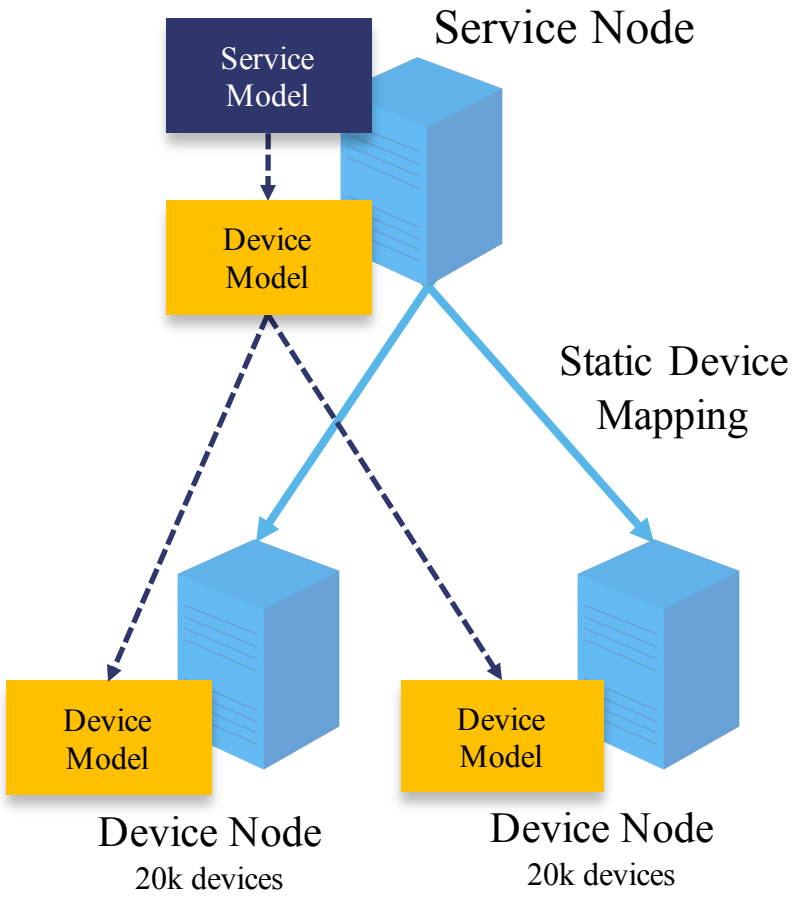


NFVO LSA



Horizontal Scaling

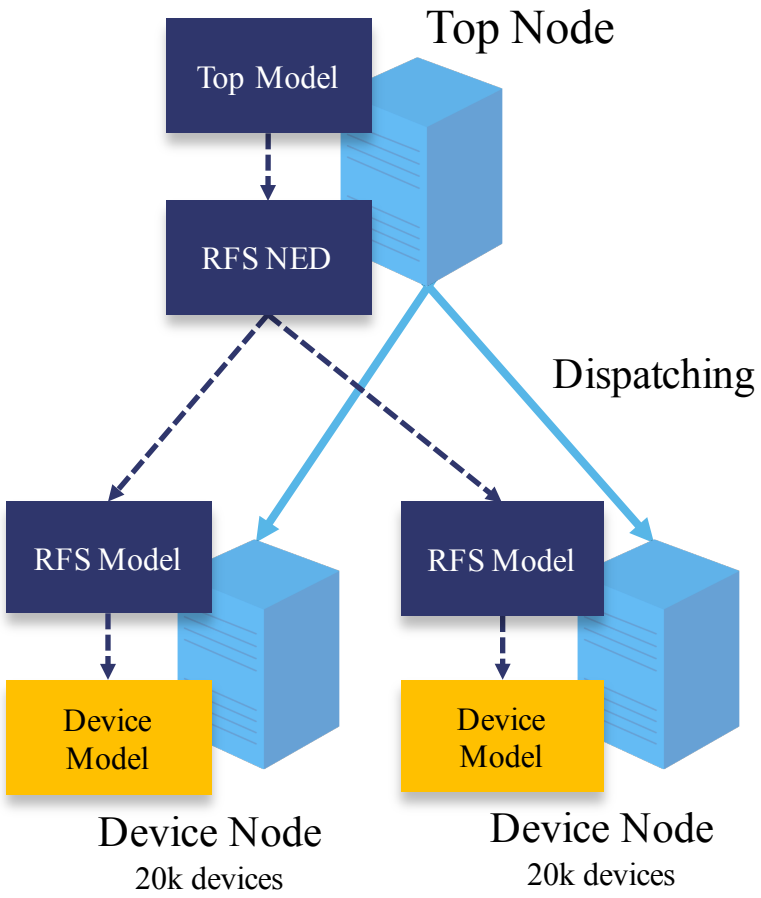
NFVO Cluster



- Device mappings need to be maintained.
- Performance can be severely impacted by cluster functionality:
  - Every call to device data on the service node results in a NETCONF RPC.
  - Cluster caching should be enabled to improve performance but it will increase memory utilization.

Horizontal Scaling

NFVO LSA (Layered Service Architecture)



- The top node only sees a small number of devices (RFS nodes).
- Total number of devices has no impact on performance of the top node.

LSA Design Guidelines

- Use LSA for virtually limitless scalability.
- Make your top-level service model agnostic to device, platform, interface, and technology.
- Devise the simplest dispatch method possible or at least one that is easy to maintain.
- Implement integration with external systems at the top layer.
- Implement resource allocation at the appropriate layer.
- Use LSA-ready design today even if you run just one node.

Q&A