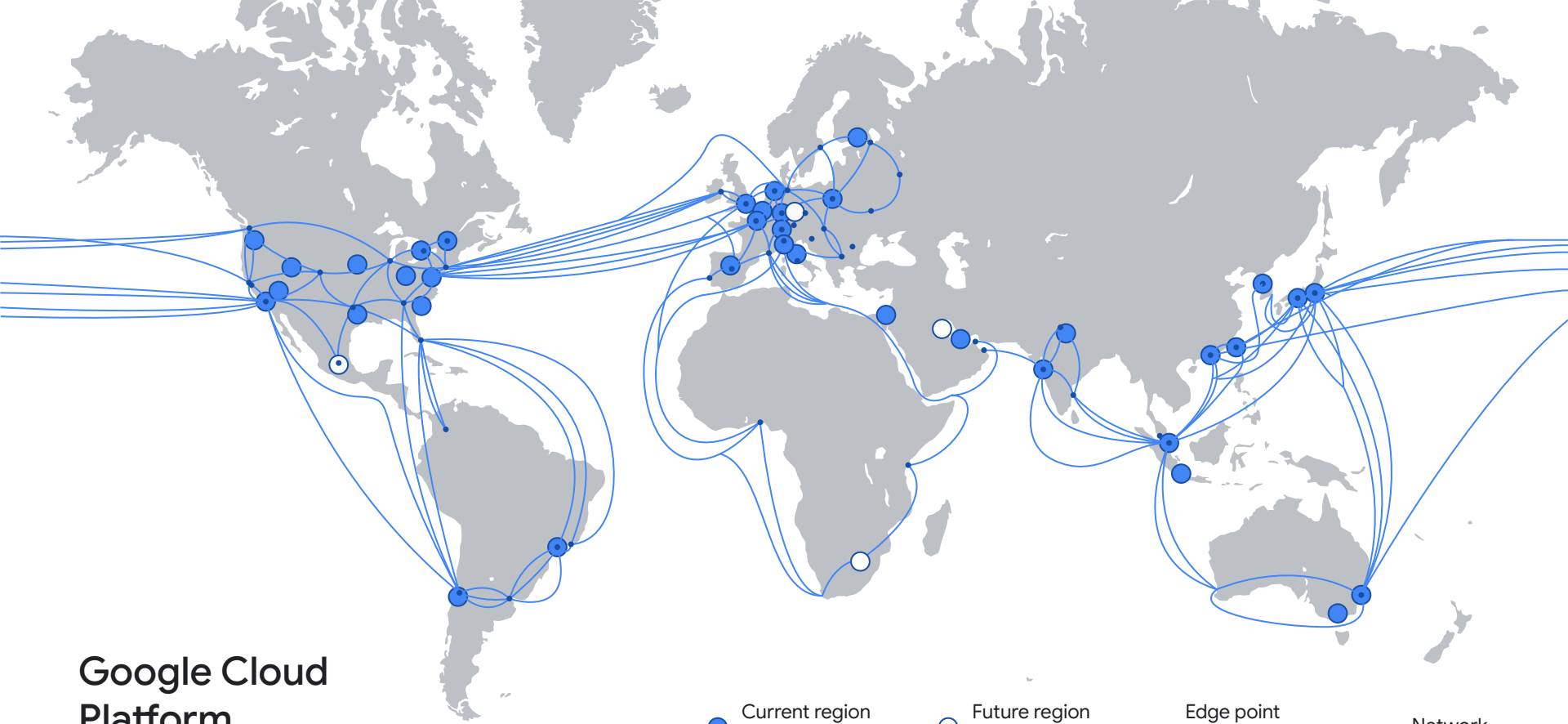


Change Management in Physical Network Lifecycle Automation

Mohammad Al-Fares*, Virginia Beauregard*, Kevin Grant*, Angus Griffith*, Jahangir Hasan*, Chen Huang*, Quan Leng*, Jiayao Li*, Alexander Lin*, Zhuotao Liu‡, Ahmed Mansy*, Bill Martinusen†, Nikil Mehta*, Jeffrey C. Mogul*, Andrew Narver*, Anshul Nigam*, Melanie Obenberger†, Sean Smith§, Kurt Steinkraus*, Sheng Sun*, Edward Thiele*, Amin Vahdat*

**Google †Formerly at Google ‡Tsinghua University §Databricks*



Google Cloud Platform

- Current region with 3 zones
- Future region with 3 zones
- Edge point of presence
- Network

Regions, PoPs, and network



Google Datacenter in Hamina, Finland

Biggest Challenges

Scale

How do you manage a vast, global network?

Safety

How do you reliably grow a serving network *safely*?

Deployment

How to you deploy networks *efficiently* and *quickly*?

Interoperability

How do you go from $O(N^2)$ to $O(1)$?

Context: What is MALT?

What is MALT?

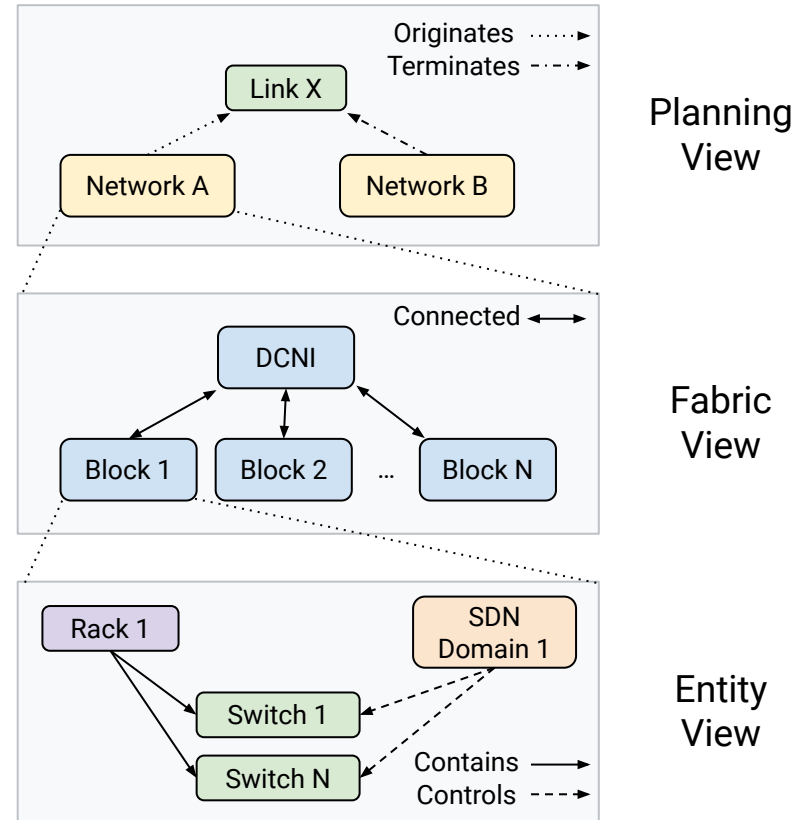
MALT*

- A language to **model** network topology, at multiple levels of abstraction
- Entity/relationship graph
 - Topology, taxonomy, state
- Versioned, immutable, sharded

MALTShop

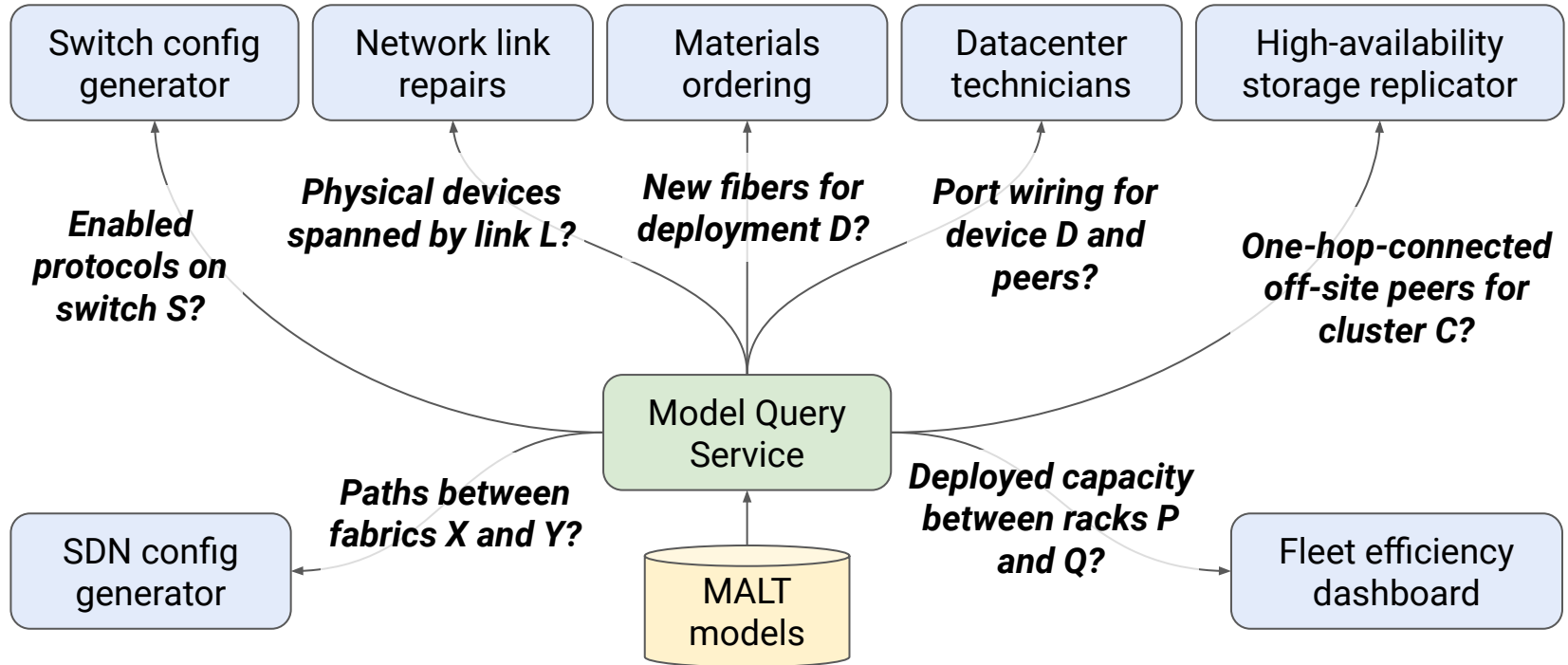
- Model storage system
- Replicated

*Mogul et al. "Experiences with Modeling Network Topologies at Multiple Levels of Abstraction," NSDI 2020



Example models available at [github](#)

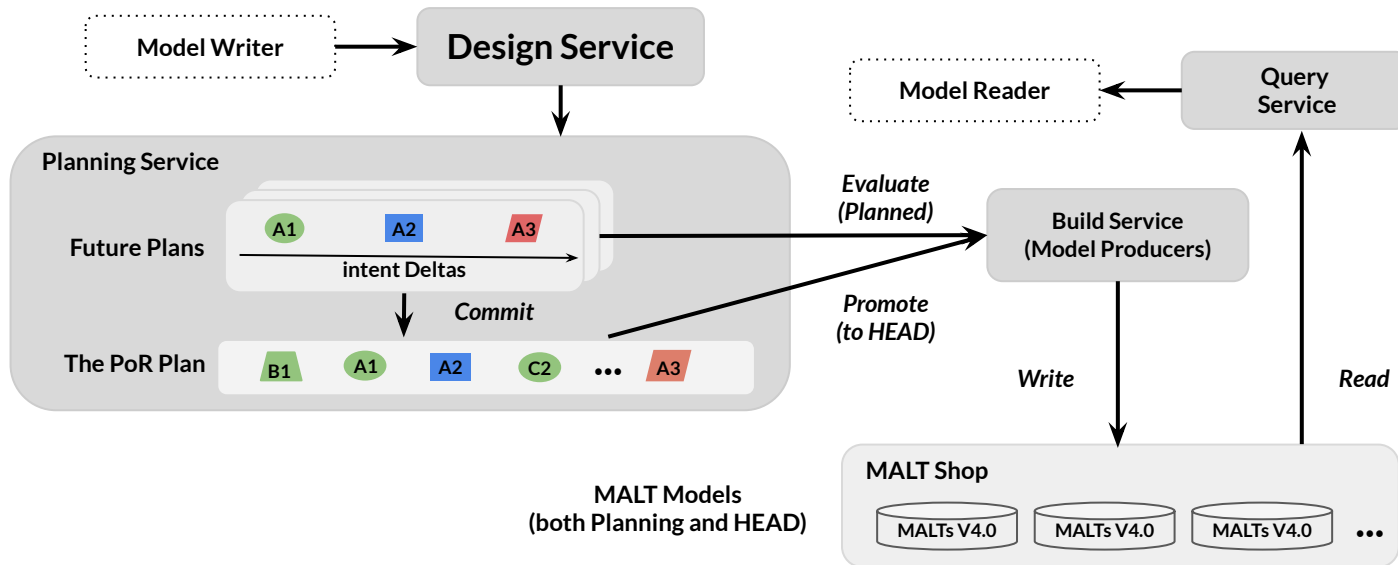
What is MALT used for?



And many, many more ...

Model-Generation and Management

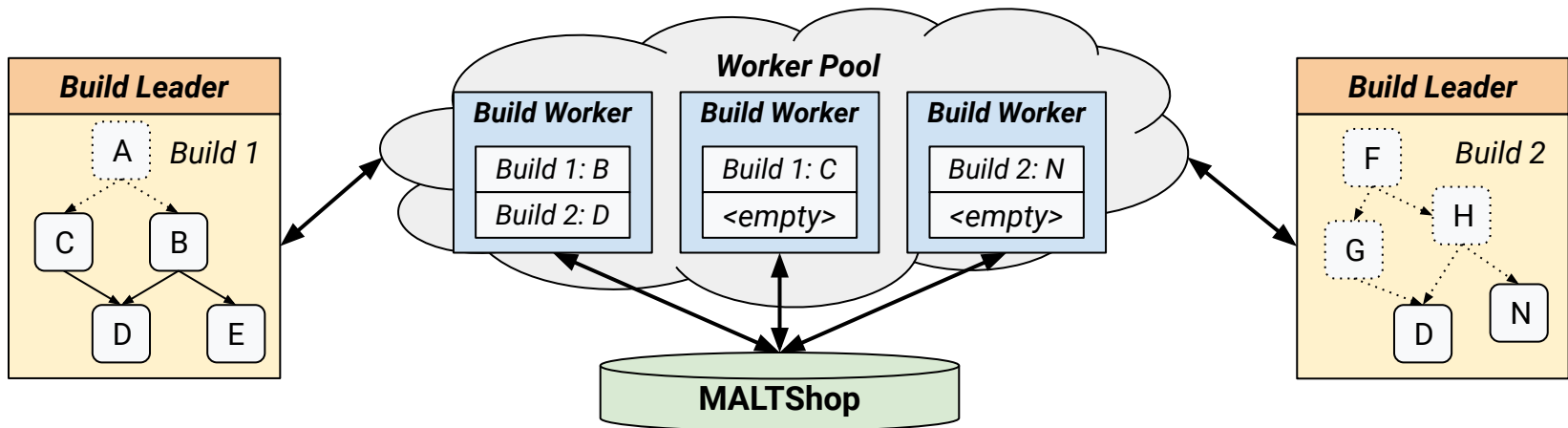
Model Design Service



Front-end for workflows to mutate current and future models

- Maintains a Unified Intent Model (UIM)
- Generates intent-delta for each change
- Synchronization point

Model Build Service



Distributed data-flow graph execution engine

- “Compiles” intent into concrete models
- Operates on: (1) Data (mostly MALT), (2) Rules
- Input-dependent, stateful
- Caching

Model Query Service

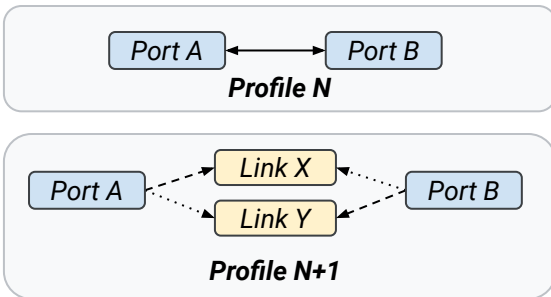
Model Profiles

- Profile Evolution
- “Semantically equivalent” models

Semantic and Canned Queries

- Insulate clients from schema details and schema changes

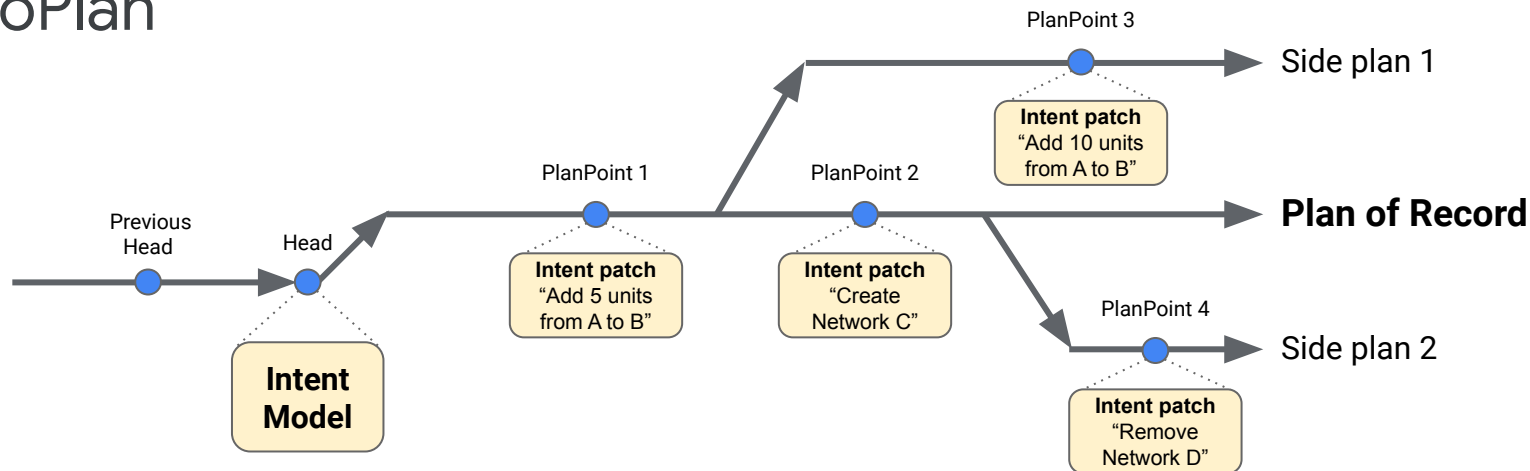
Connected \longleftrightarrow Originates \dashrightarrow Terminates $\cdots\rightarrow$



Example queries:

- “What is the set of enabled interfaces on switch X?”
- “What are the BGP peers of fabric Y at time T?”

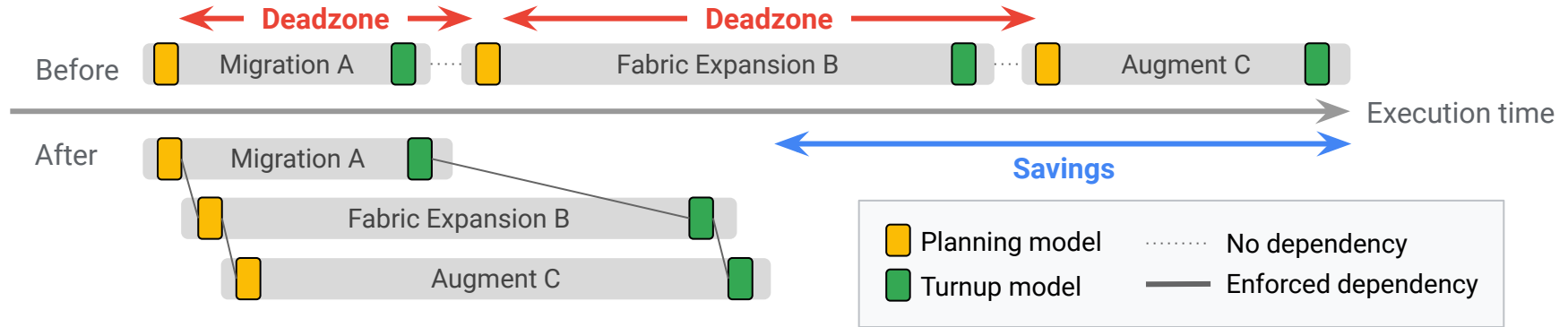
TopoPlan



- Planning Service to manage *Future Network Topology*
 - Analogous to a Version Control System (VCS)
- **Plan:** Similar to a commit branch
 - Maintain “plan of record,” as well as side plans and what-ifs
- **PlanPoint:** Intent Delta, similar to a commit
- Build service can evaluate at any point

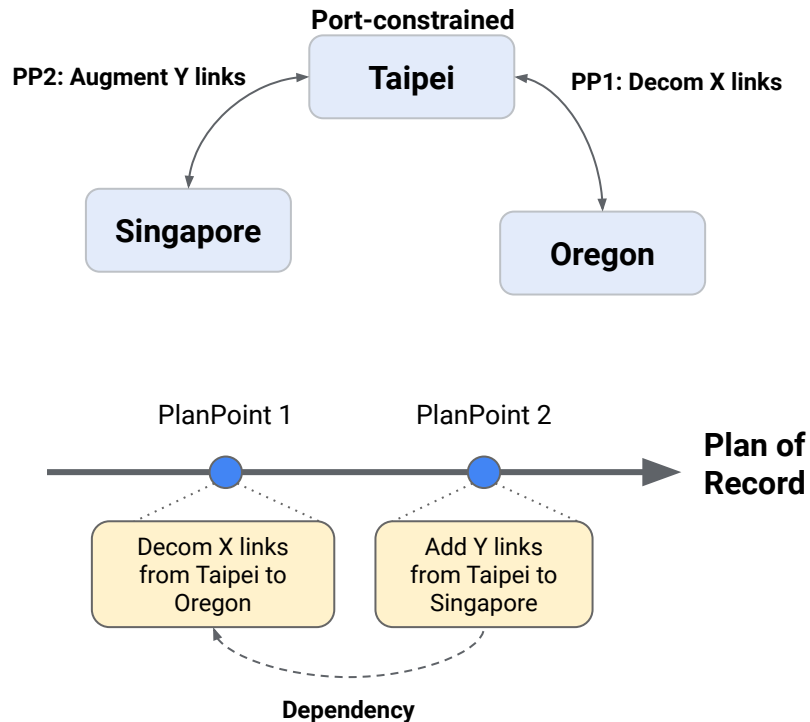
Experiences / Case-Studies

Deadzone Reduction



- Deployment projects are long-lived
- **Deadzones:** Artificial serialization of projects
- Detailed planning support
 - a. Separates future network state from operational models
 - b. Allows execution pipelining
- **Benefit:** streamlined deployment execution

WAN Change Management



Datacenter vs. WAN networks

- DCs (mostly) independent
- WAN is globally interdependent

Project Dependency Management

- Enables project concurrency
- Surface impact of plan-change
- Execution sequence resilience

Summary and Takeaways

Summary

Scale

Automate, automate, automate

Safety

Validate network invariants, as early as possible

Deployment

Detailed planning support, with dependency tracking

Interoperability

Common modeling language, with profile evolution