# Evolving towards Software Defined

Eliminating barriers to adoption of IT Innovation

HUAWEI

# Agenda

- **3 Keys to enabling innovation and transformation**
  - Programmable data plane
  - Vertical disaggregation of network solutions
  - Implementation independent integration language
- **Intent based networking**
- **Potential applicability of IBN to Platform Lab's BCP project**
- **Unsolved Problems we need to solve: System architecture, scaling,**

# Something new? Too Scary!

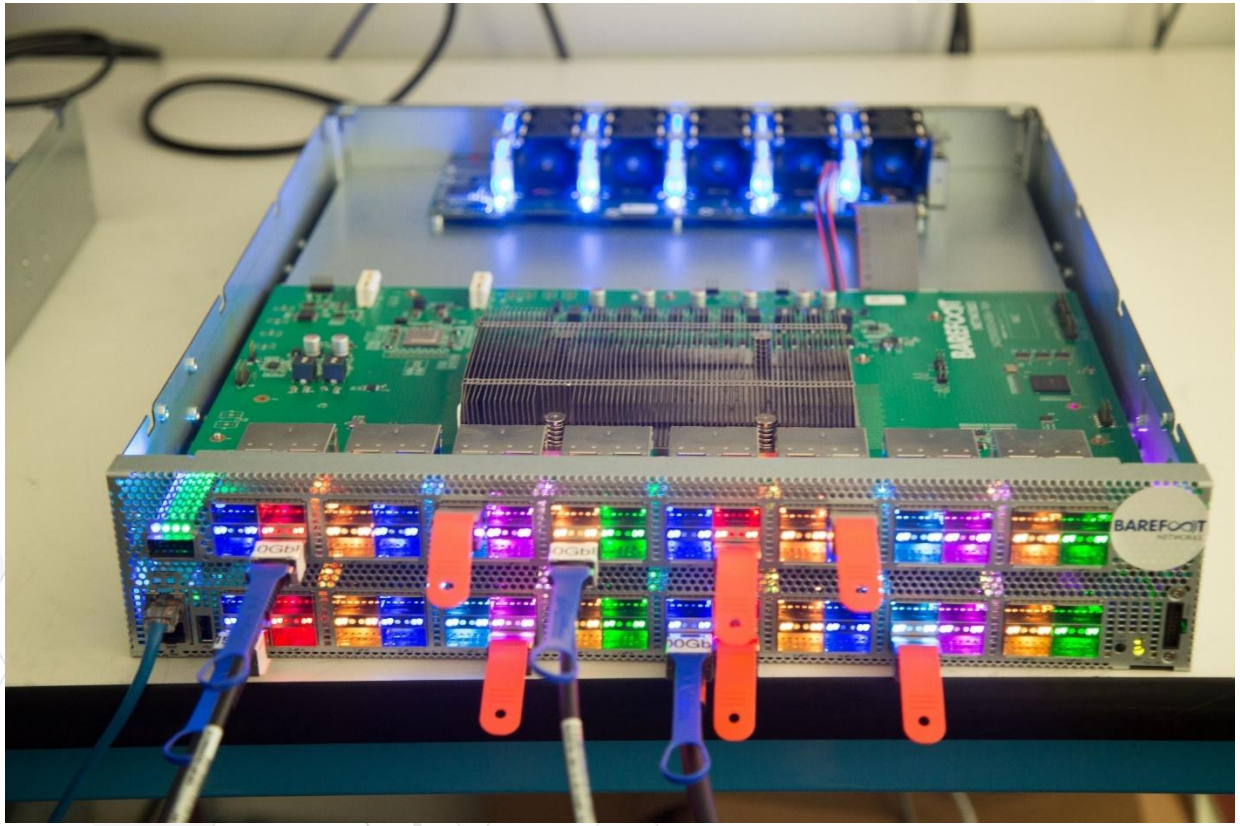
- **Millions of dollars of integration and capital equipment cannot be undone if it’s a disaster.**
- **Must change both hardware platform and operating software simultaneously.**
- **Changes to operations, training, processes too disruptive**
- **Can't succeed until risk is reduced and clear benefit is identified**

Dangerous Road. Can’t Backup!

# The Big Three Barriers to network innovation

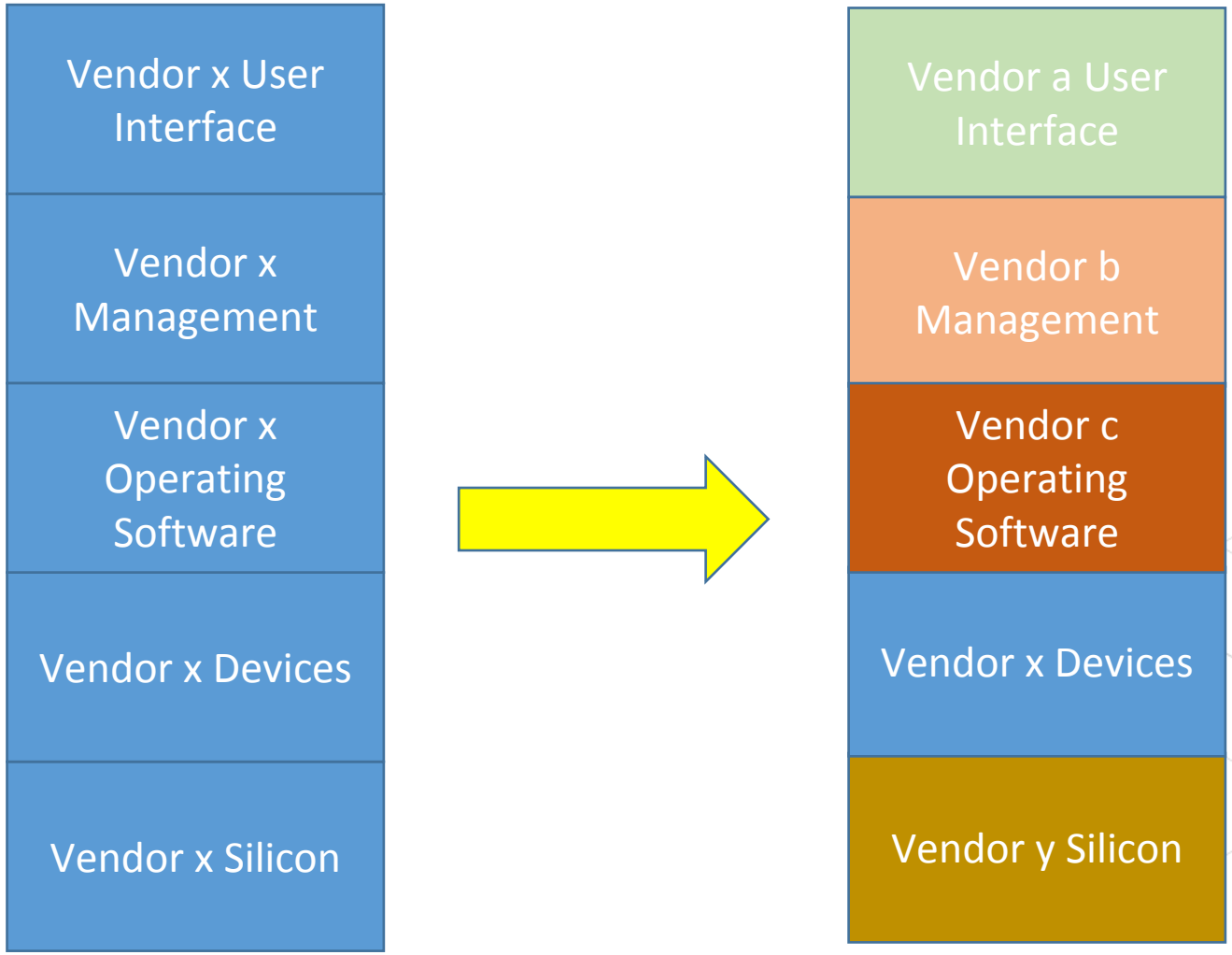
1. Data Plane is Not Programmable
2. Integration work is Entirely Implementation Specific
3. Solutions are vertically integrated

# Data Plane Programmability is coming




- Many startups and established vendors working on this
- New instruction sets and languages (e.g. P4) allow downloading of new “wire-speed” features to deployed device.
- Designers can choose to deploy new logic in device or controller

# Disaggregation can enable risk-free changes



# Intent: Model the application, not the network

- User creates implementation independent description of what applications need from the network.
- Users describe what they need in terms they understand
- Automation and Experts help guide the translation to terms the provider can fulfill
- Cost and risk of trying or changing solution components becomes minimal.


Intent: Take me from A to B

Review of IBN concepts and development status

# INTENT BASED NETWORKING

# Why Intent?

Eliminate “Test Drive” cost and risk

Eliminate Vendor Lock-In

Make Solution Components Fungible

Enable “programming the network” for Non-Experts

Allow Write-once, Run-anywhere Infrastructure Integration

Support Dynamic Behaviors of Network Applications and Resources

# Intent Based Model Versus Traditional Model

## Intent-based Operating Model

- Describe the problem
- Model the Workload requirements
- Tell me what you need
- Make my headache stop
- I need a virtual network (logical isolation) for VMs 1, 2 & 3
- 99%of network “users” only have to understand their business and workloads

## Traditional Network Operating Model

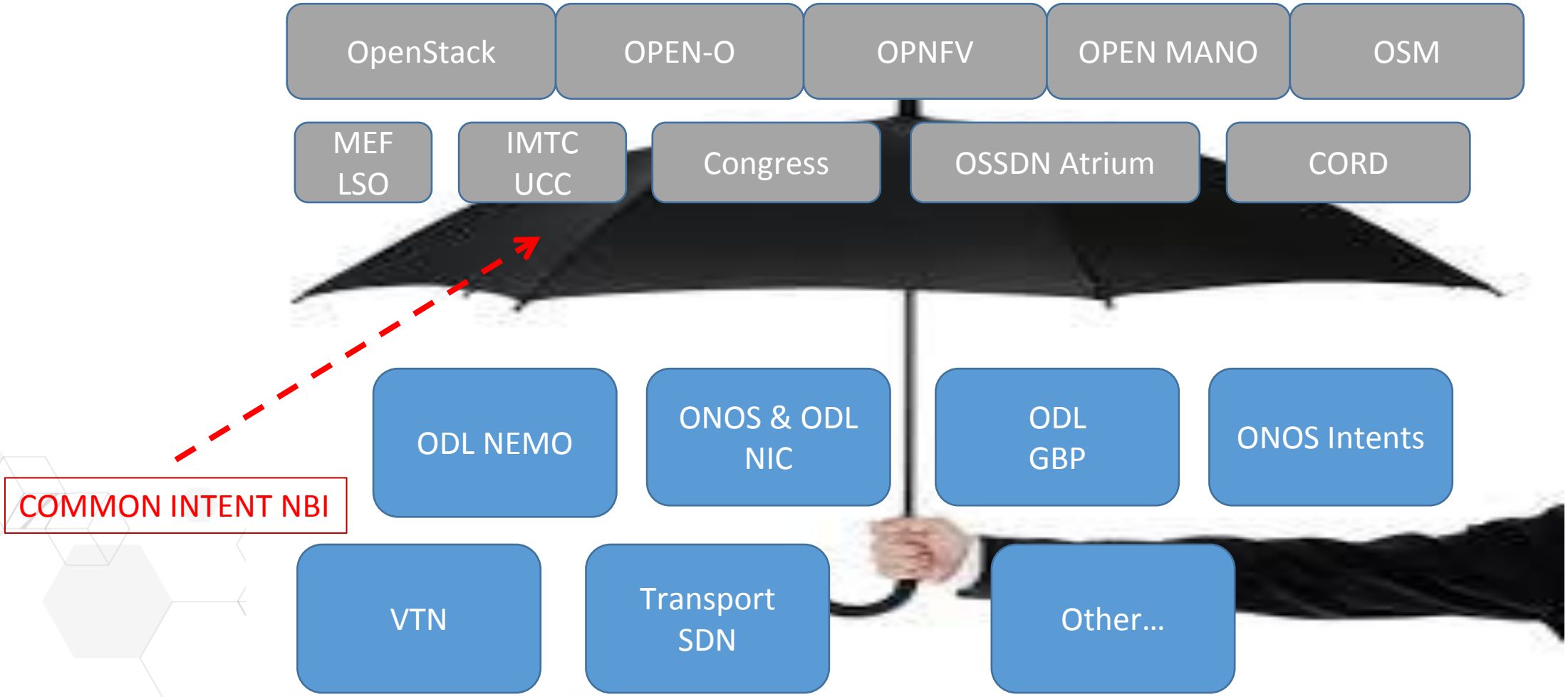
- Describe the solution
- Model the Network
- Tell me what to do
- Give me an aspirin
- I need, e.g., VXLan tunnels, full L2 mesh between VMs 1, 2 & 3,
- 100% of network “users” need to be experts in networking as well as their business and workload verticals.

# Intent is a Virtual World

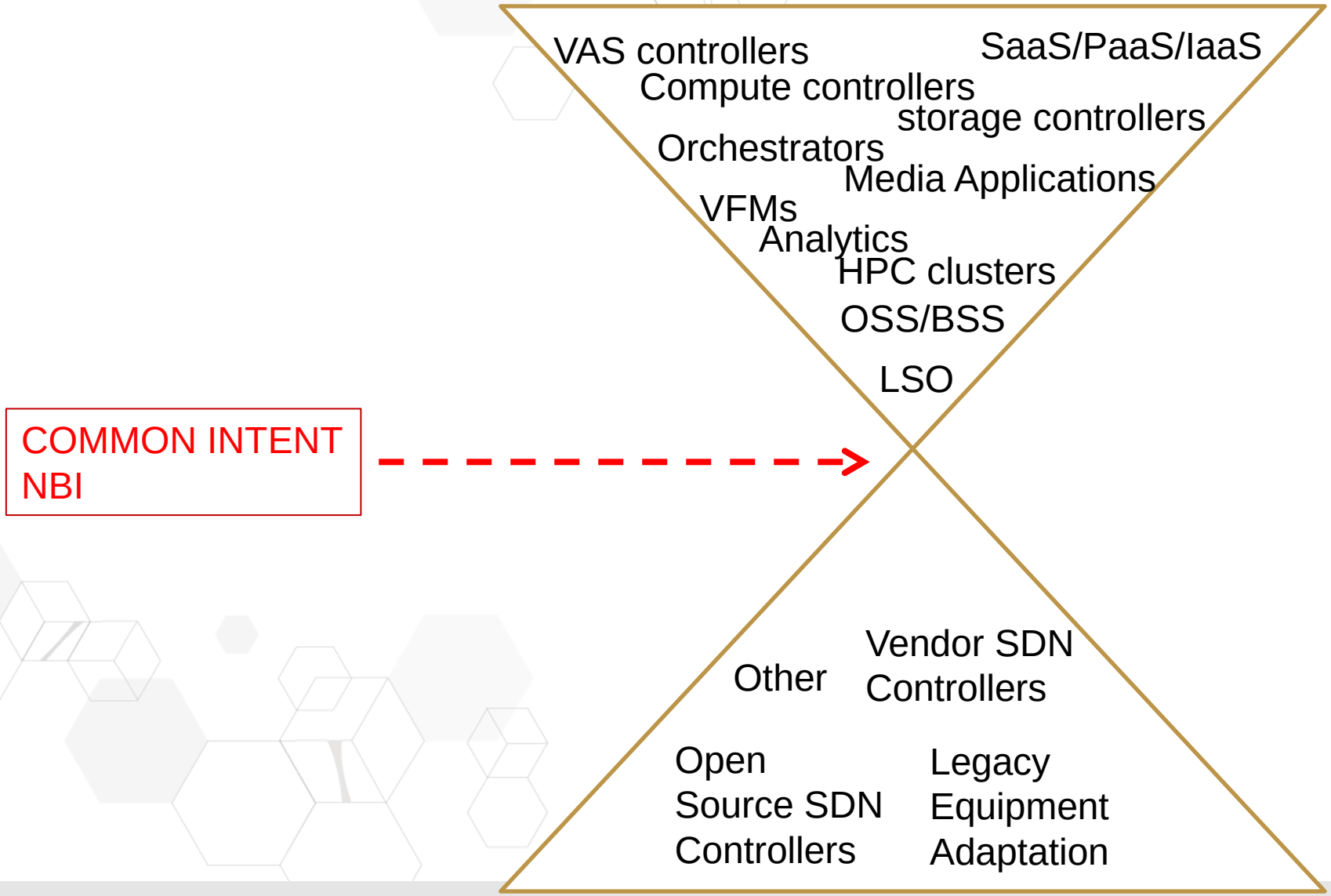
## Label-mapping Makes It Real

- Elastic, infinite, extensible, reliable, available, simple. No corner cases ☺
- Intent relationships can be described between virtual Objects and Object groups
- Intent statements apply run-time extensible set of modifiers and predicates to relationships between objects/groups
- You don’t get to specify or touch underlying resource pool
- Extensible Framework: Add one use case at a time

# Goal: Unifying Common NBI  Shim



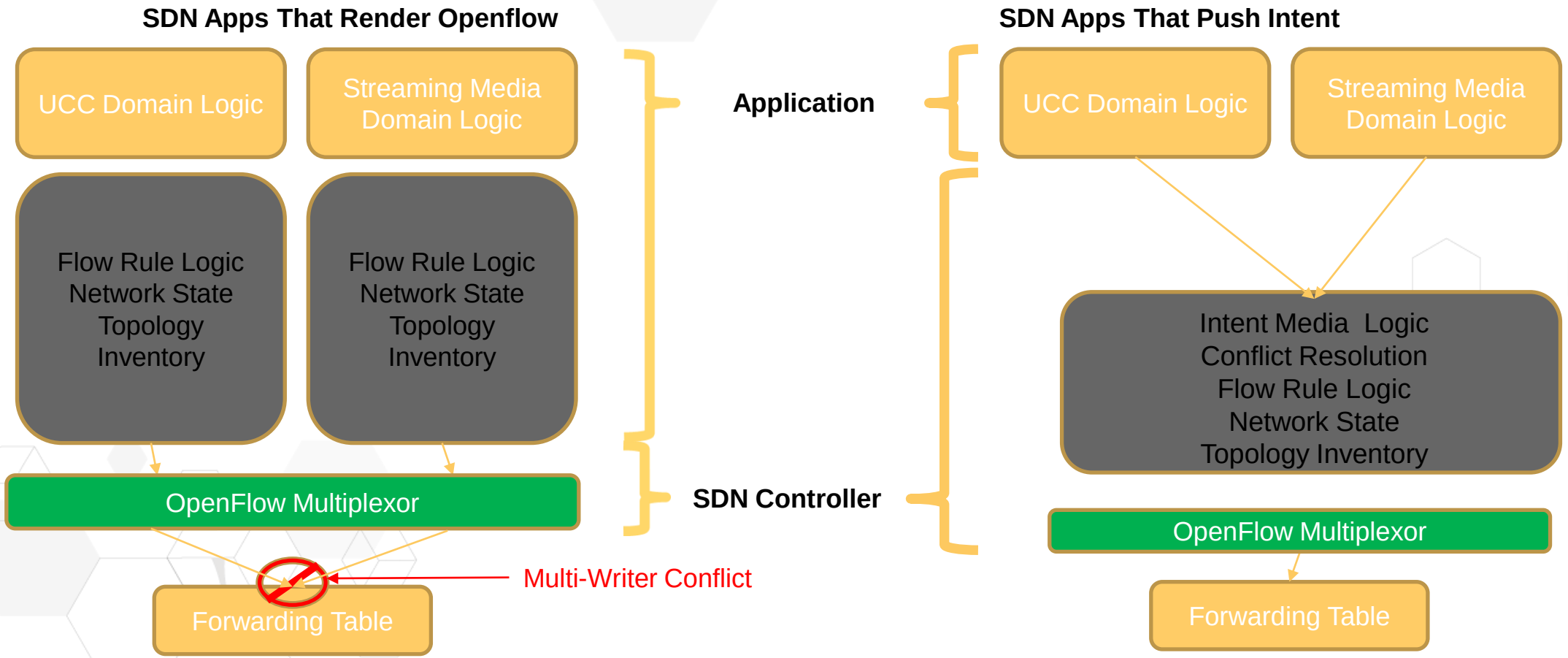
# Narrow Waist Interoperability  Demarcation



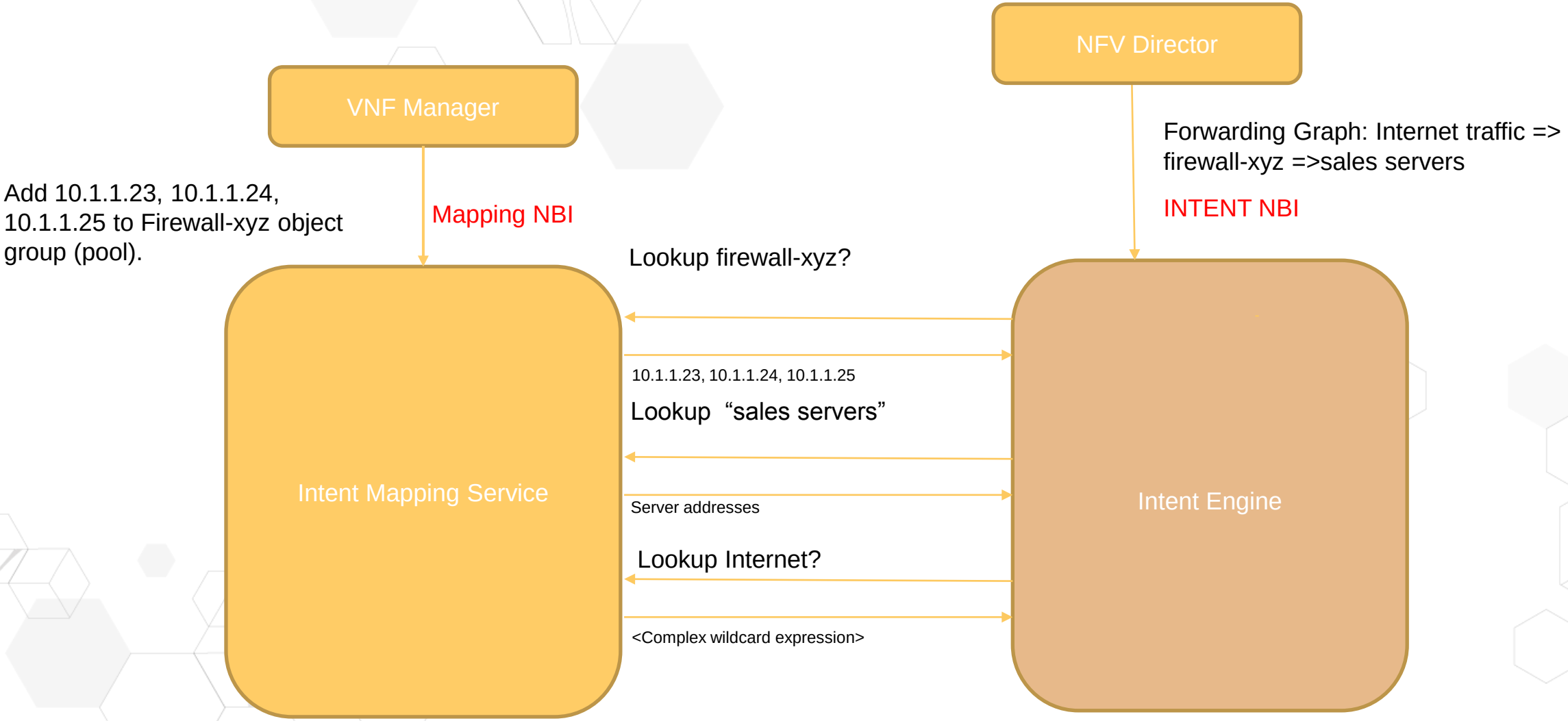
# Any Use Case That Can Be Described Can Be Split Into Intent+ Mappings

| Intent | Mapping |
| --- | --- |
| • Changes in management-plane time, human time, and minutes/hours<br>• Does not change based on state of network, endpoints, users.<br>• Independent of protocol, media, vendor, etc.<br>• Easily understood and authored by non-experts<br>• Simple test to determine whether desired state is portable enough to be intent | • Changes in control-plane time, real-time, and sub-second<br>• Changes whenever the state of the network or resources changes.<br>• Specific to resolving abstract intent to protocol, media, vendor, etc.<br>• Requires deep understanding of technology, networks, etc. |

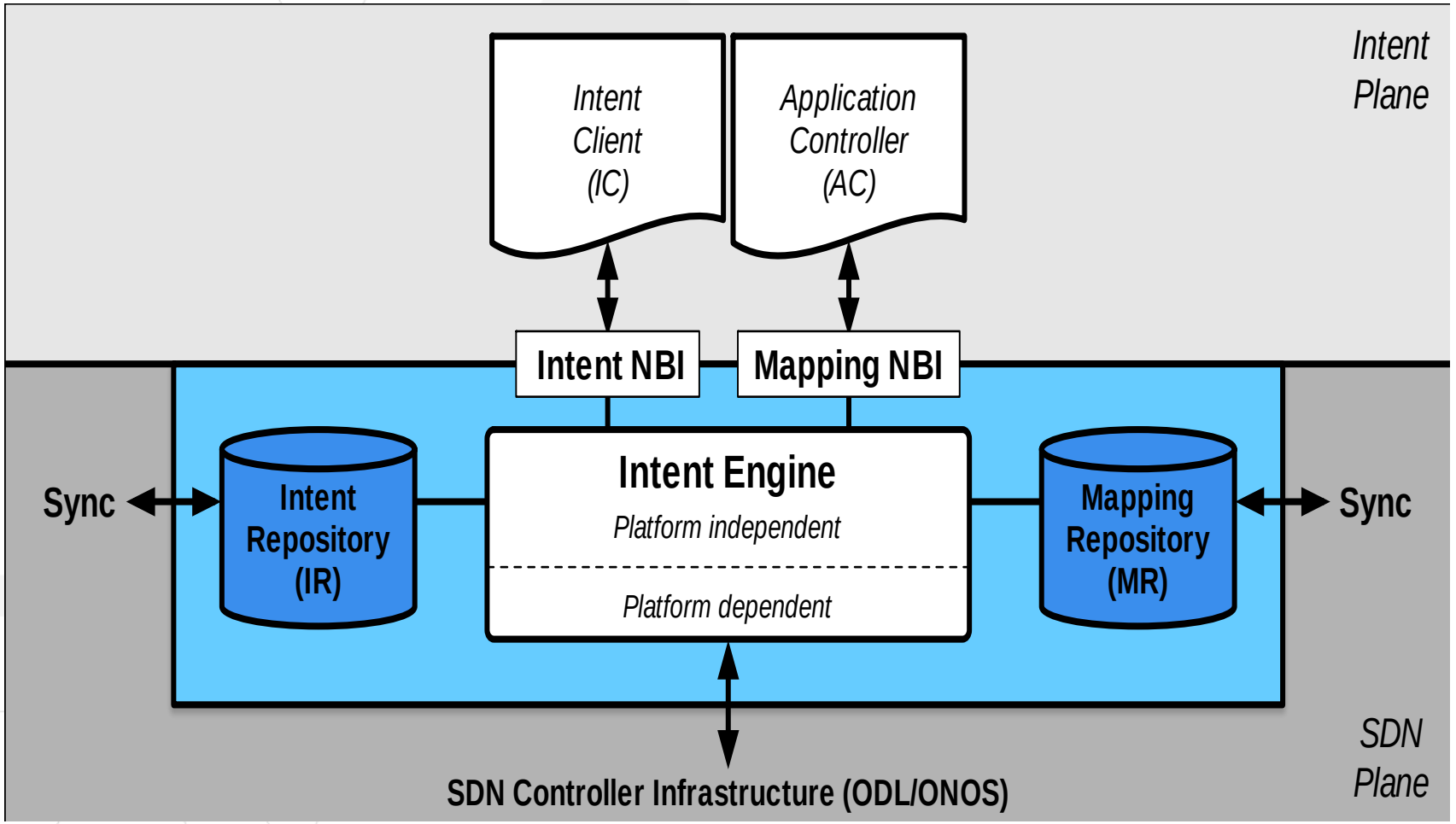
# The Intent Killer App – Solving the Multi-Writer Problem



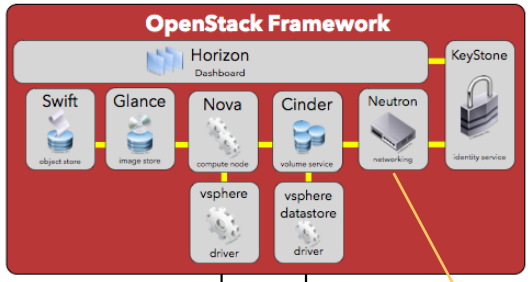
# Intent Based SFC/NFV

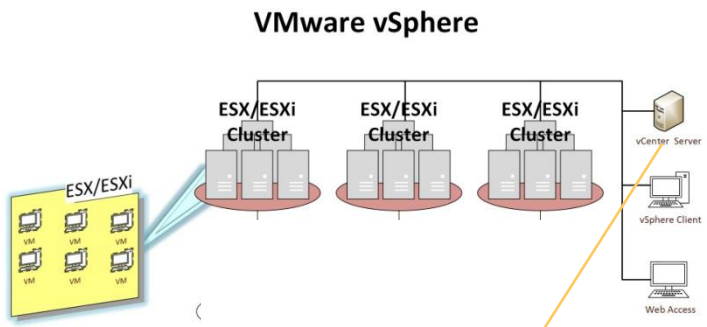


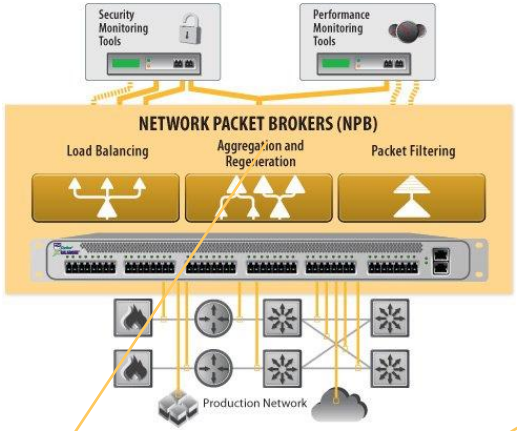
# Intent Based System Architecture



# External sources of truth feed real-time mapping




Skype for Business

Mapping Repository

Network Experts and Engineers

# NBI Specifics - Intent NBI Atoms

- Object
- Object Group
- Modifier

RESTCONF CRUD operations on above items

YANG model based

- Intent objects and their relationships form graph
- Graph theory can be applied for resolving aggregate requirements, config, minimal update, multi-path routing, etc.

# Intent and BCP

- **BCP will control a superset of systems that includes network and cloud computing infrastructure (in order to further support MEC applications for drones, robots, autonomous vehicles, etc. )**
- **It makes sense to build this next generation automation/control system using an intent based interface**
  - ONF intent NBI work is becoming de facto standard way to interface with network controllers supported by work in ONF, ODL, ONOS, etc.
  - Architectural benefits including modularity, composability, portability, future-proofing, migration enabling, multi-vendor, controller agnostic, protocol agnostic, etc., etc.
- **CRI would like to explore working with platform lab to solve some of the problems that we know stand between our current prototyping, and a deployable hyper-scale control system.**

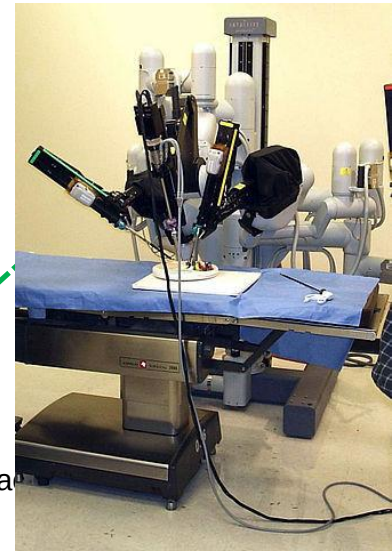
# Sample Design and Research Problems

- **How do we balance the centralized/global state sharing versus the distributed/local state.**
  - Fully autonomous won't work. Fully Centralized won't work.
- **We have stated that Intent is global in nature, and changes relatively slowly (e.g. human/policy timescale)**
  - We can replicate this slow changing, low volume data at massive scale
- **Much of the rendering logic will be pushed out to small, shared-nothing intent domains each with a smaller number of objects and devices to control. The system scales-out linearly to the extent we are able to live with shared-nothing**
- **We need a way to efficiently, coherently distribute the bare minimum of shared state information.**

# Transit Path Advertisement and Scheduling

- End-to-end deployment of intent can and will cross multiple disjoint intent domains.
- Some higher level (logically centralized cooperative intelligence) logic within the intent stack must understand the available ingress/egress paths available for stitching together end-to-end service behaviors across multiple otherwise autonomous, shared-nothing domains.
- Exactly how does an intent domain advertise any/all interconnecting network interfaces with adjacent intent domains. What resources capabilities need advertising and how are they interpreted by central logic.
- Do we need one or more additional controller-of-controllers layer to stitch end-to-end across these meta-domains for max scale?
- Looks like BGP with constraint routing problem space, but needs new solution? Fast-reconvergence based on global view “memory”?

# E2E Path Computation Across Intent Domains



# Mobile Edge Service scheduling and State Handoff

# Mapping Service Replication, Compression, Synchronization

- First cut can make great progress with OTS distributed key-value stores and dense state exchange
- Ultimate scale will require optimized, multi-path aware transactional systems and sparse/summarized state exchange.
- Need to invent, model, simulate, prove techniques to achieve global telco and web scale.

# Minimal update to global rendering

- When a change occurs to the state of infrastructure, intent or mappings, the intent engine has to compute and push new rules to adapt the network to the new combination of inputs.
- The naïve implementation recomputes everything from scratch, possibly resulting in massive thrashing of traffic in-flight with resulting dropped sessions, etc.
- The problem that needs to be solved is to build a rendering engine that can generate assembly-language (e.g. openflow rules) for many network devices at scale in response to state changes that minimize the disruption to the existing state of rules satisfying the aggregate end-to-end requirements.

# Power of implementation-agnostic Model

## Intent Data

- Portable. Implementation/state independent
- Scale-able. Compact,  global meta-data
- Compose-able. Common, general model
- Understandable. No army of experts
- Secure-able. No flow-tables, topo, inventory
- Write-once, run anywhere
- Future-proof. No more integration expenses here

## Mapping Data

- Changes with platform, infra, state
- Fast changing, locally meaningful
- Segmented, per domain
- Requires implementation expertise
- Exposes more powerful abstractions
- Maintenance per-implementation
- Remaining  subset that changes as you operate or move platforms.

# Intent Levels The Playing Field

## Vendors Who Can Compete on Price/Performance/Innovation Win

Operators are primary beneficiaries

Network Effect drives ecosystem

**Virtuous cycle of vendors supporting IBN and operators asking for IBN**

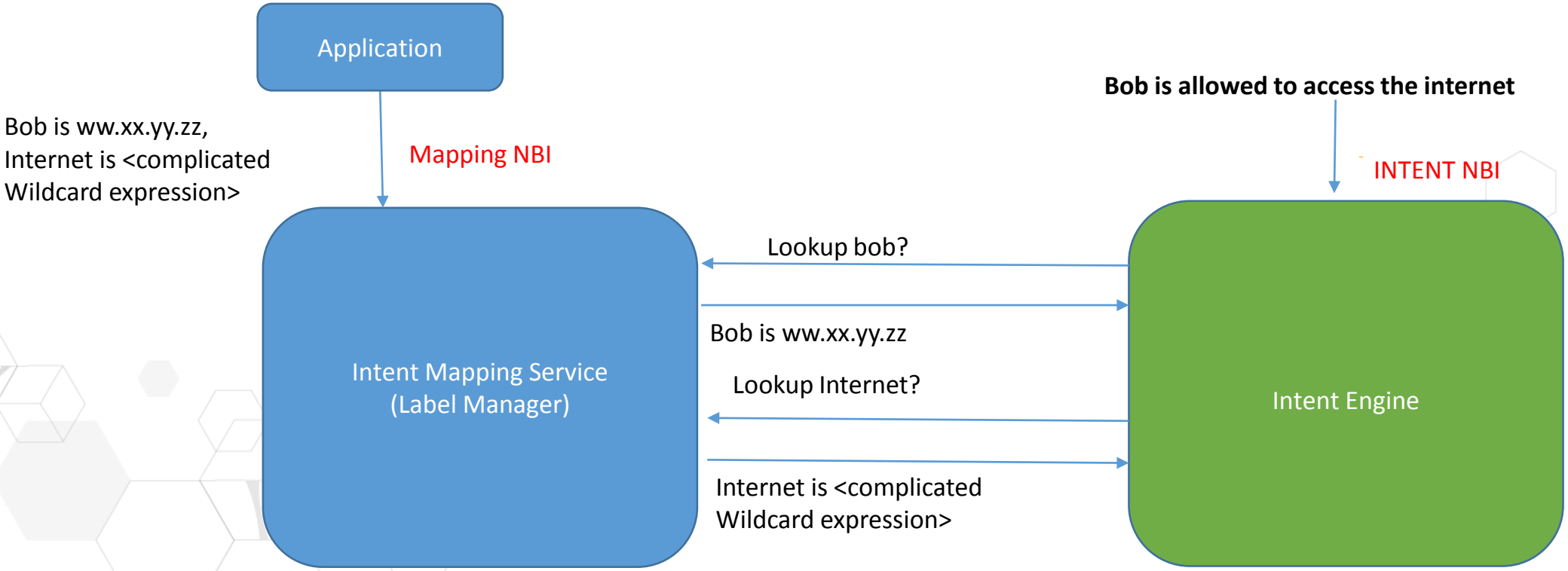
# Thank You

- Next Question:

How can we collaborate and contribute to solving these problems in BCP context?

# Simple Connectivity Use Case: Bob's Internet

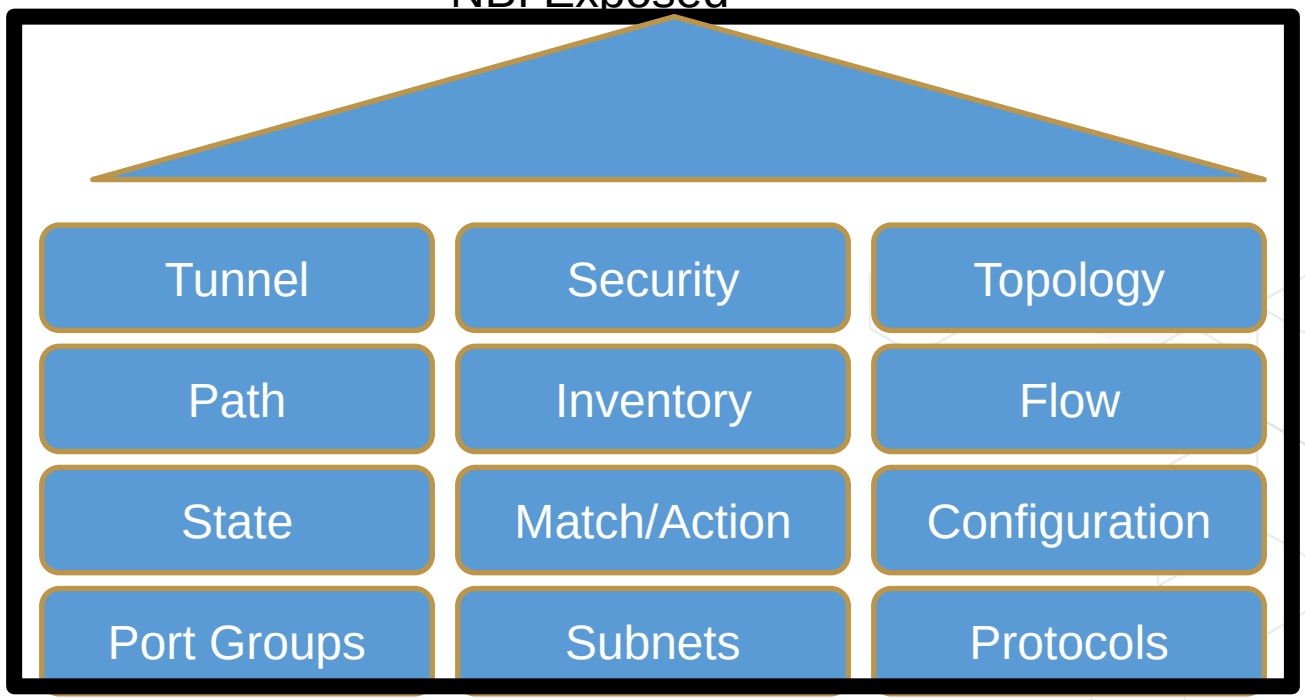
Labels not understood by intent syntax resolved by mapping service



# IBN reduces SDN Attack Surfaces

*What I need*

Common Intent NBI Exposed




"Fine Grained" NBIs Exposed

Tunnel | Security | Topology
Path | Inventory | Flow
State | Match/Action | Configuration
Port Groups | Subnets | Protocols

# Intent Levels The Playing Field

## Vendors Who Can Compete on Price/Performance/Innovation Win

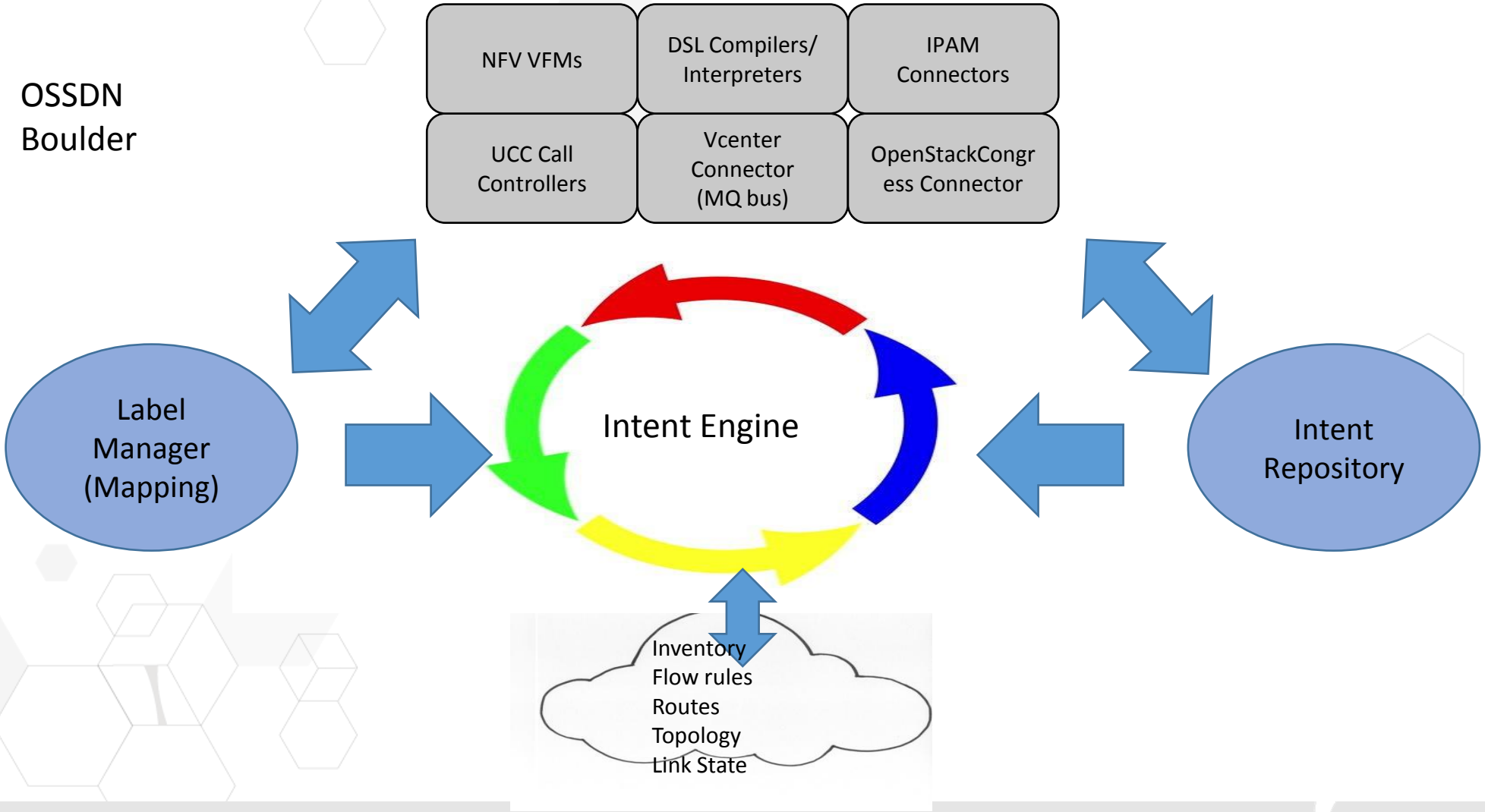
**Operators are primary beneficiaries**

**Network Effect drives ecosystem**

## Virtuous cycle of vendors supporting IBN and operators asking for IBN

# OSSDN Boulder – Intent Demarc

OSSDN Boulder

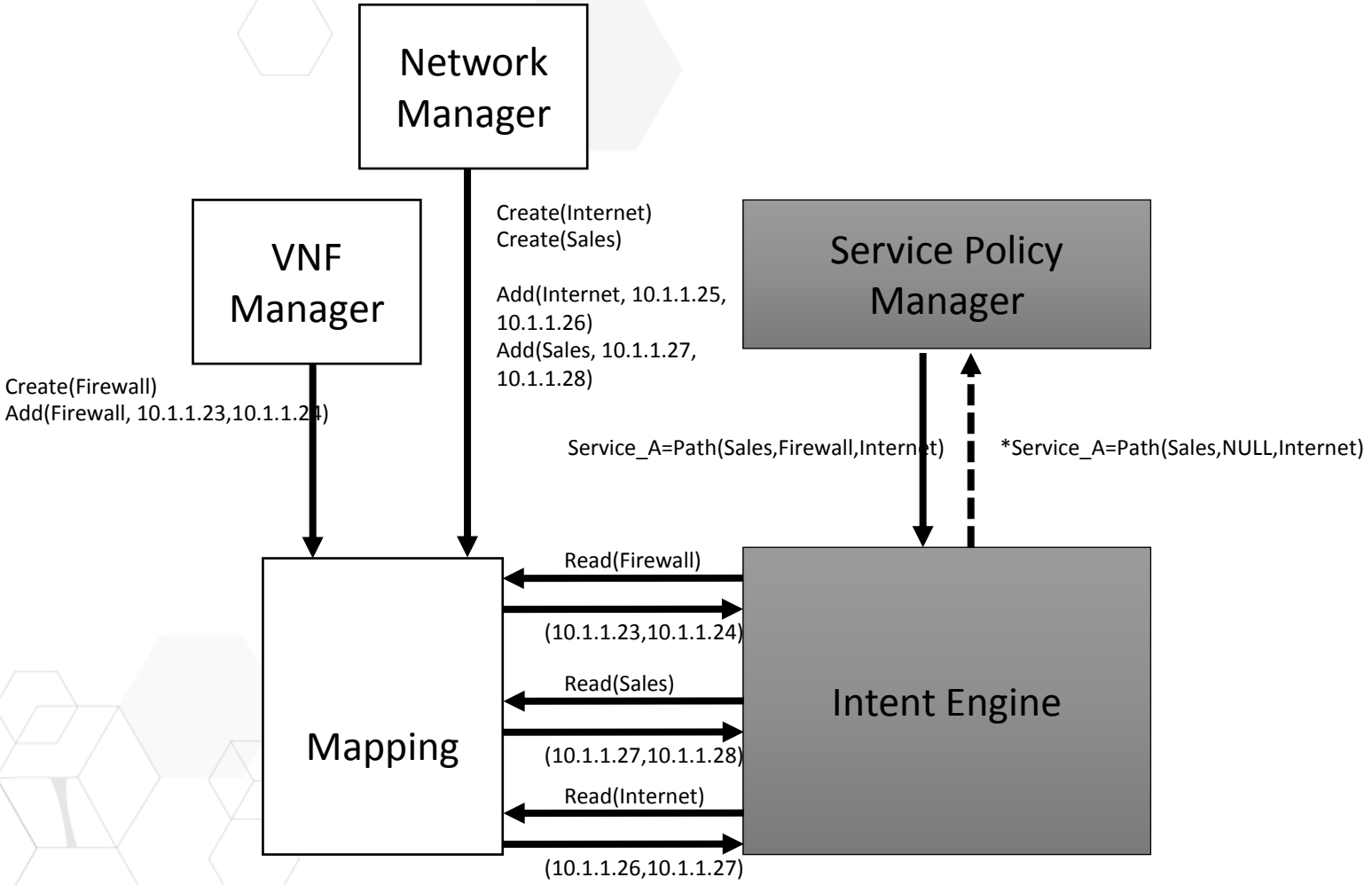


# Over-Prescription Yields Fewer Solution Choices

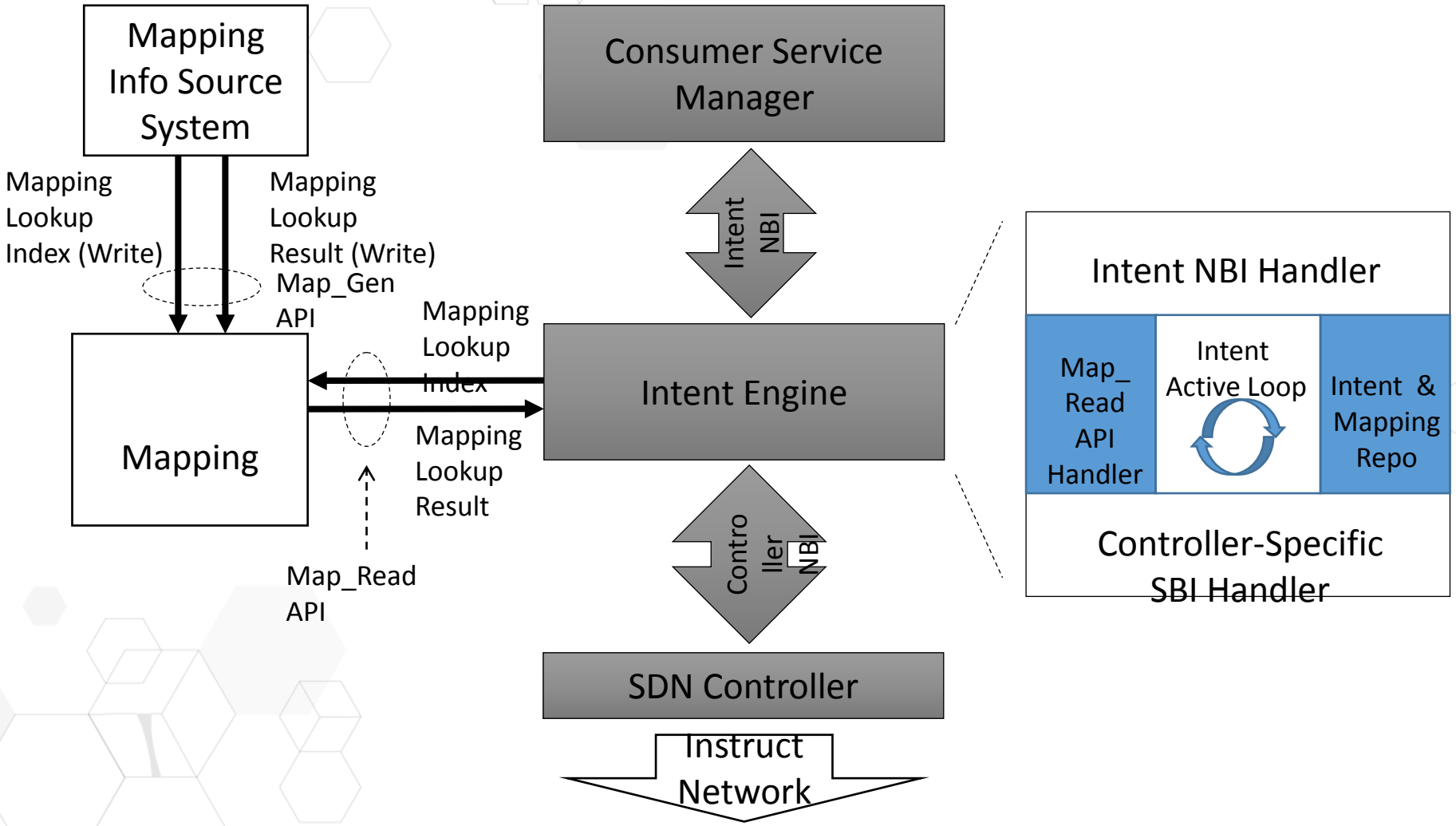
# Any Use Case That Can Be Described Can Be Split Into Intent+ Mappings

| Intent | Mapping |
| --- | --- |
| • Changes in management-plane time, human time, and minutes/hours<br>• Does not change based on state of network, endpoints, users.<br>• Independent of protocol, media, vendor, etc.<br>• Easily understood and authored by non-experts<br>• Simple test to determine whether desired state is portable enough to be intent | • Changes in control-plane time, real-time, and sub-second<br>• Changes whenever the state of the network or resources changes.<br>• Specific to resolving abstract intent to protocol, media, vendor, etc.<br>• Requires deep understanding of technology, networks, etc. |

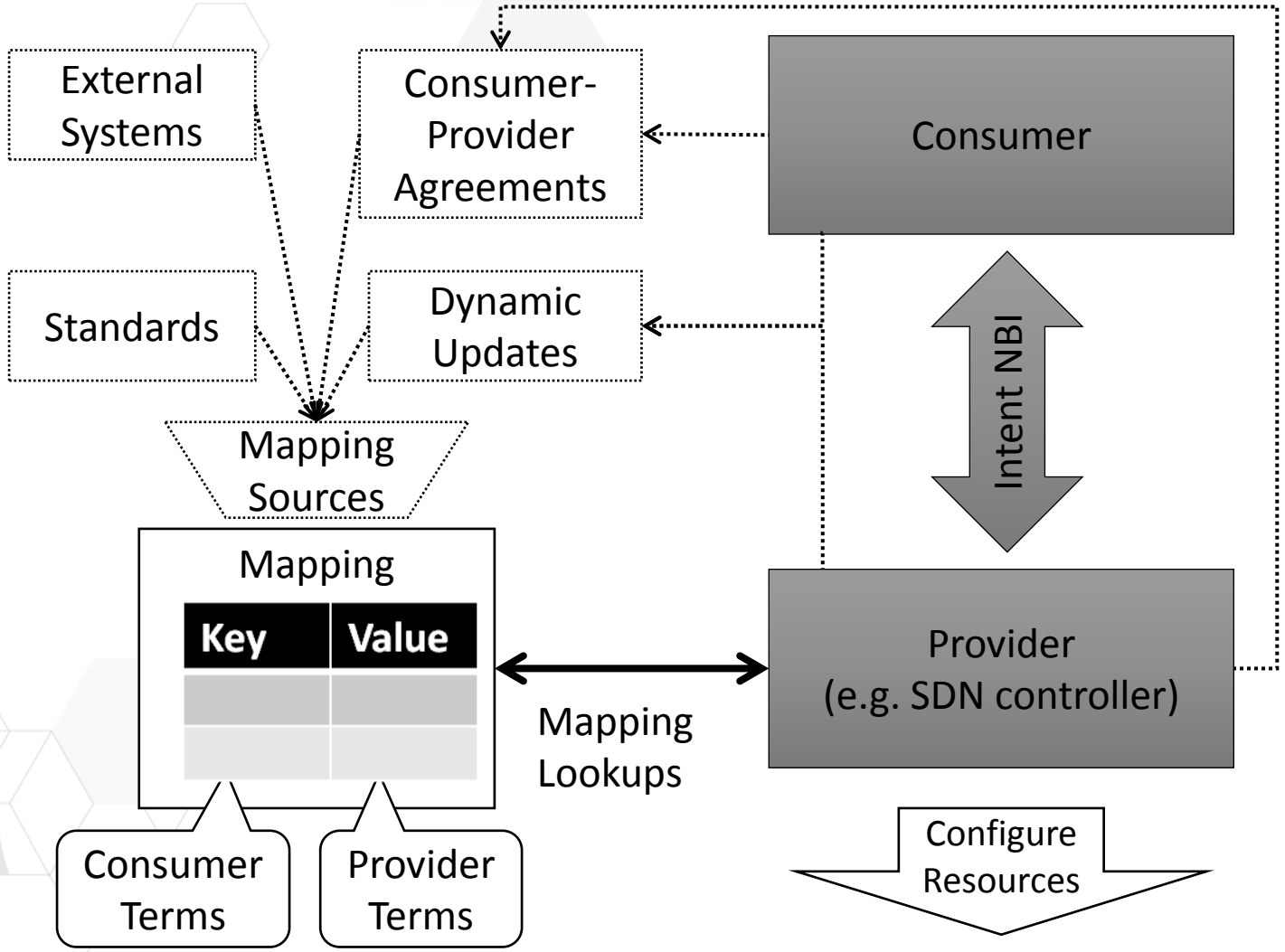
# Intent Based Service  Function Chaining



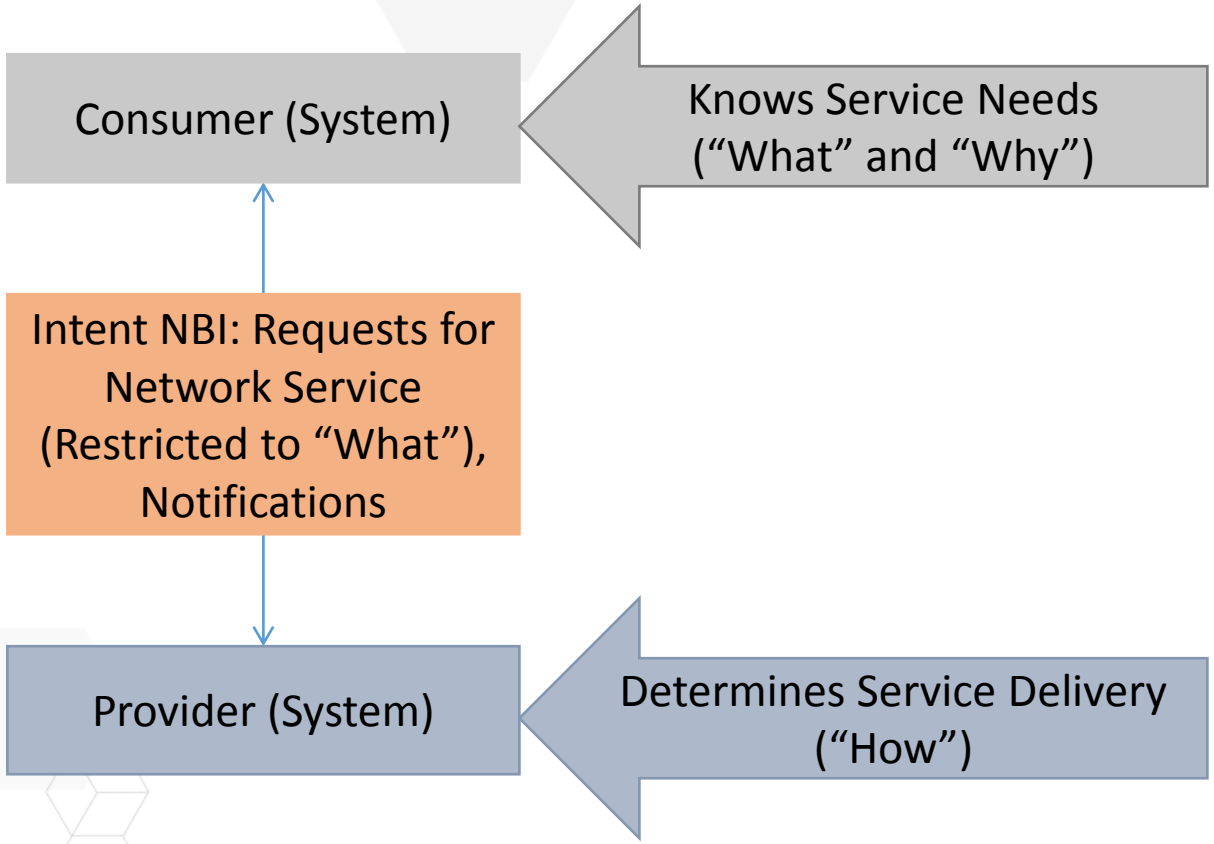
# SDN Controller “Intent Engine”



# Getting From Consumer model to producer model



# Consumer-provider interactions using Intent NBI



# Architectural representation of Intent NBI and mapping