



ONOS Intents and Northbound

Yi Tseng
ONOS Build 2017

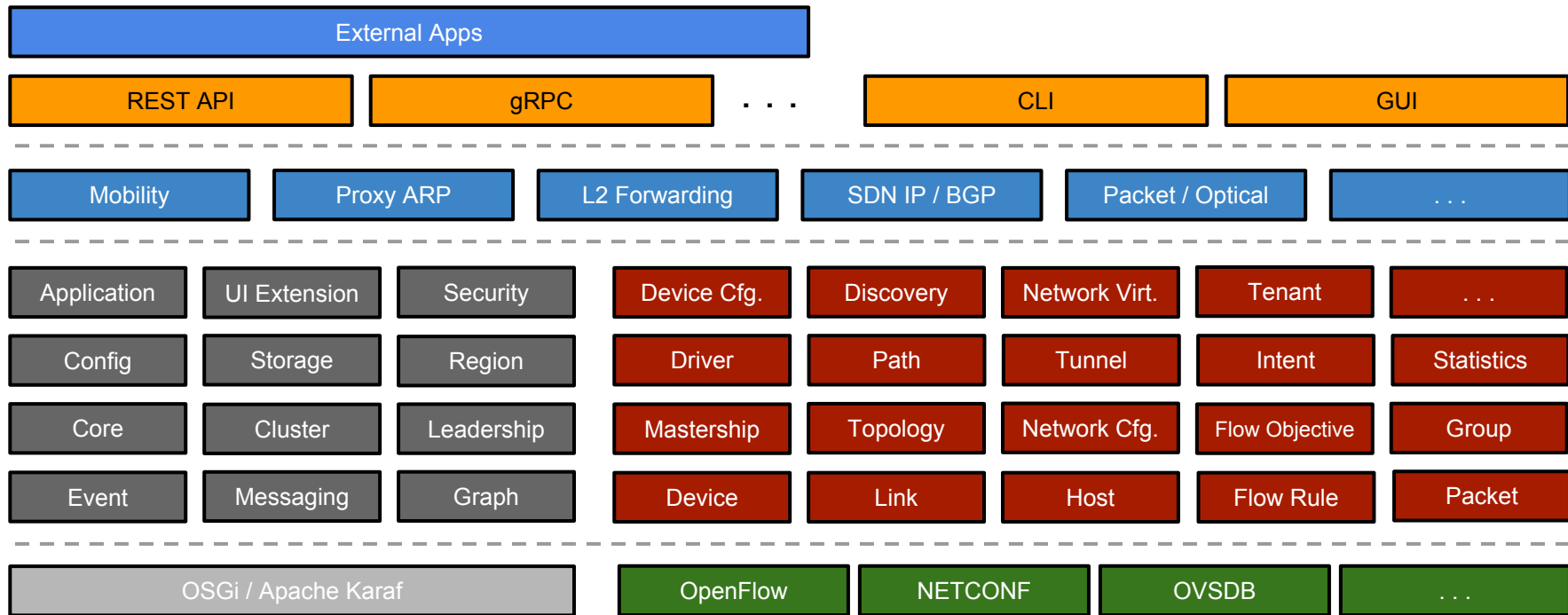
Northbound



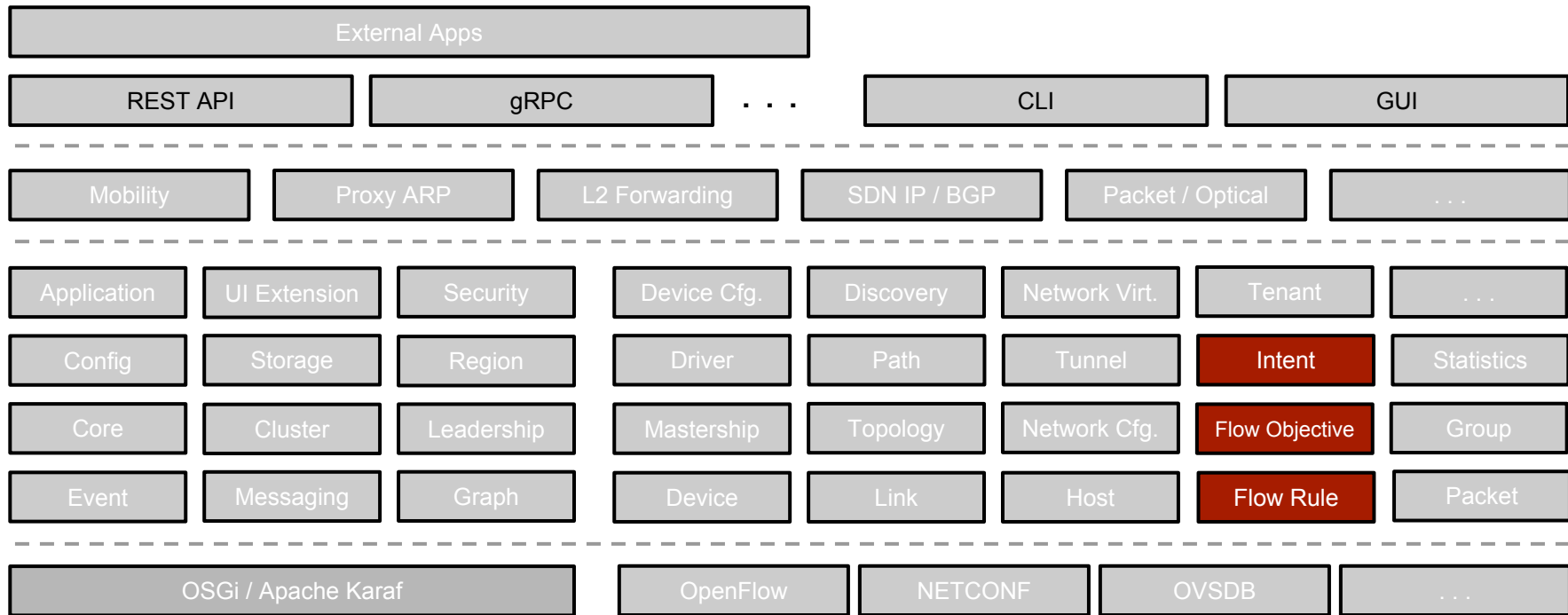
Communicate between the SDN Controller and the services and applications running over the network.

Enable efficient orchestration and automation of the network to align with the needs of different applications via SDN network programmability.

ONOS Core Subsystems



ONOS Core Subsystems



Key Northbound Abstractions



Flow Rules

Control protocol independent, Pipeline specific

Flow Objective

Pipeline independent and agnostic flow rule

Intent

Topology independent, network-centric abstraction

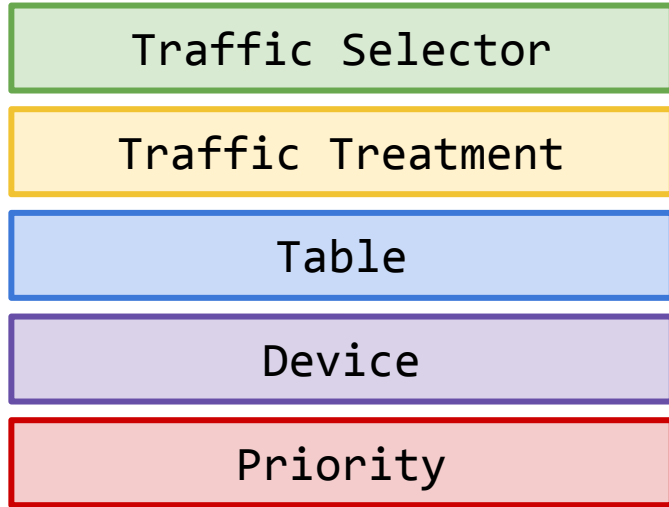


Flow Rule

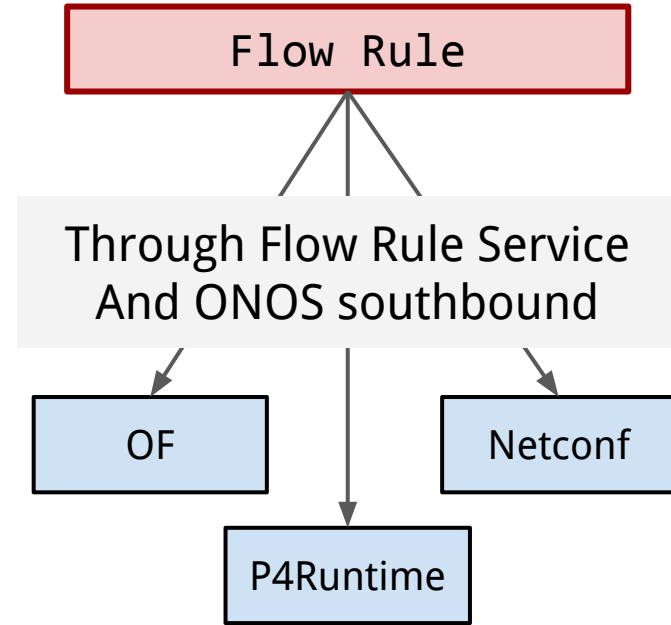
Flow Rule



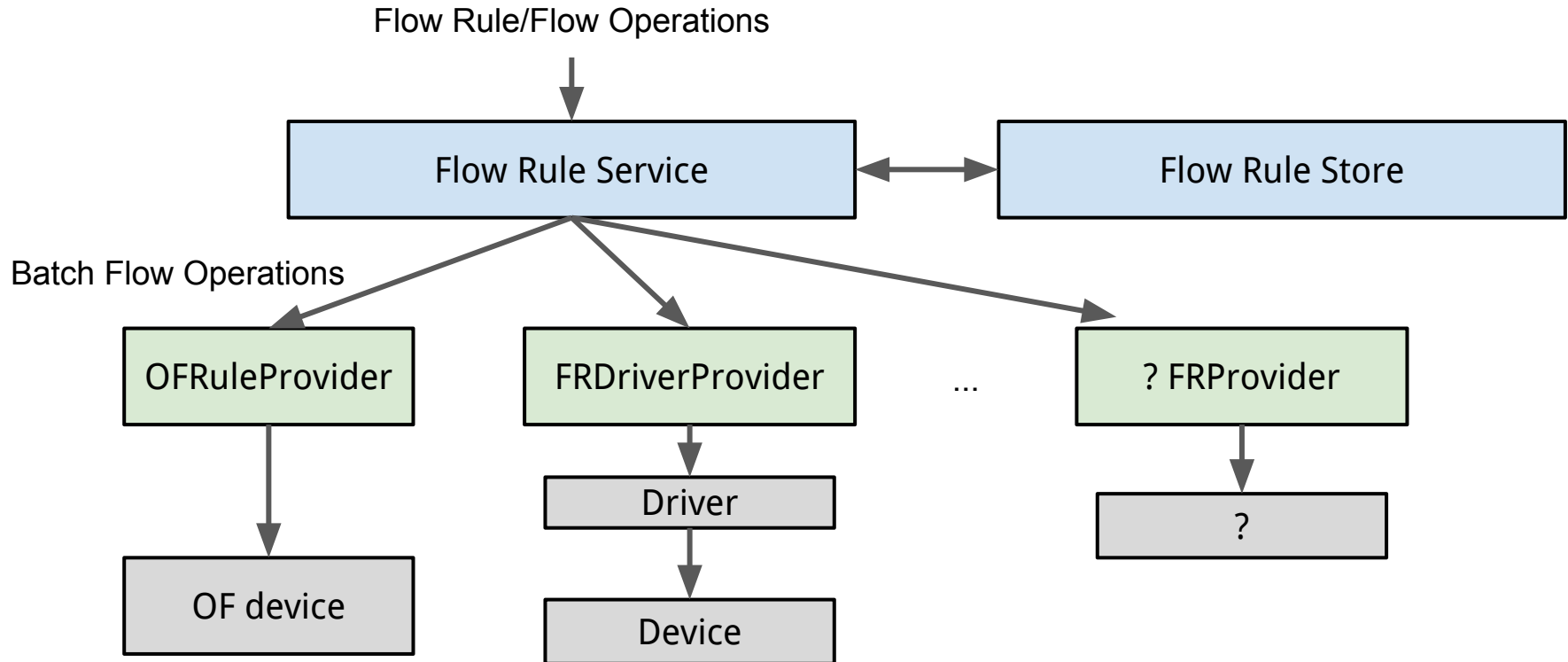
Protocol independent, Pipeline specific



·
·
·



Flow Rule Service architecture





Flow Objective

Flow Objective



- **Abstraction** for applications to be **pipeline unaware** while **benefiting** from multi-table architectures
- **Enable app portability**
 - **Interoperability** between different type of pipelines coexisting in heterogenous networks.
 - Support for a **new pipeline** is achieved through a **new pipeliner behaviour** in a driver.

Types of Flow Objective



Filtering Objective

Permit or deny packet go into the pipeline.

Forwarding Objective

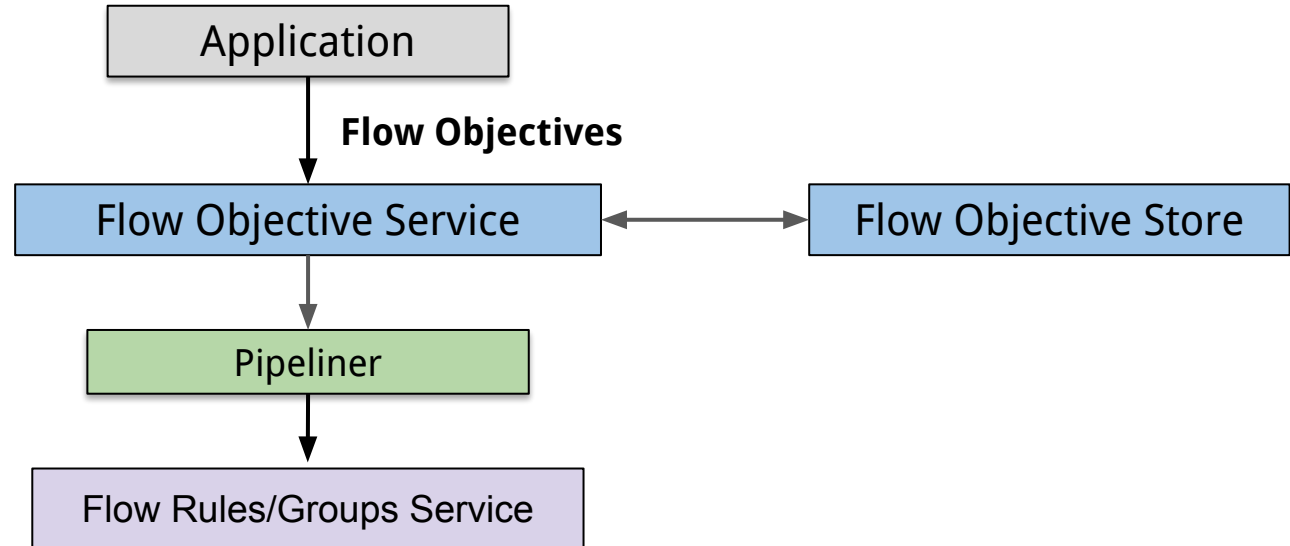
Represents a description of which types of traffic need to be forwarded through the device.

Next Objective

Represents a next hop which will be translated by a driver into the appropriate group or actions needed to implement the egress function.

(An objective may result in multiple rules at the device.)

Flow Objective Architecture

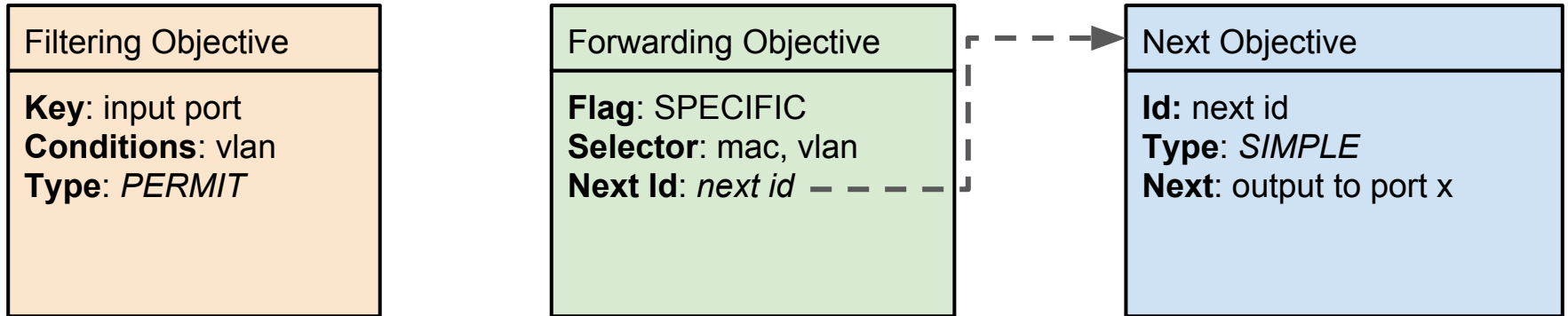


Pipeliner is used to translate flow objectives to the specific flow rules and groups for a given device and pipeline.

Pipeline example



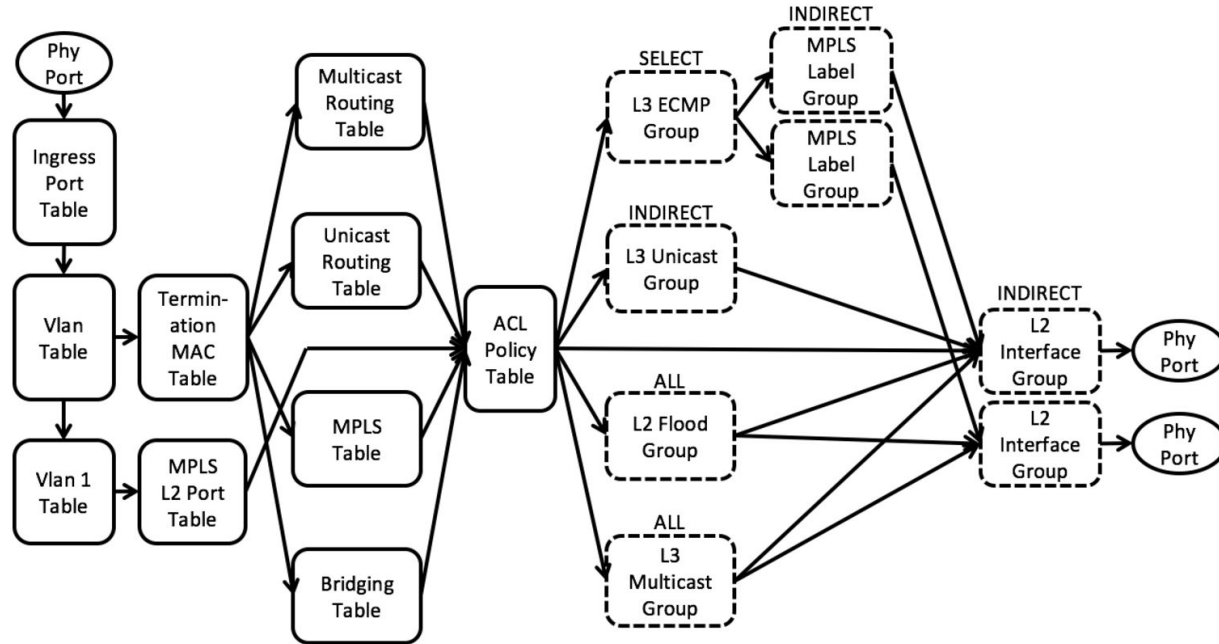
OF-DPA Pipeliner, L2 unicast



Pipeliner example



OF-DPA Pipeliner, L2 unicast



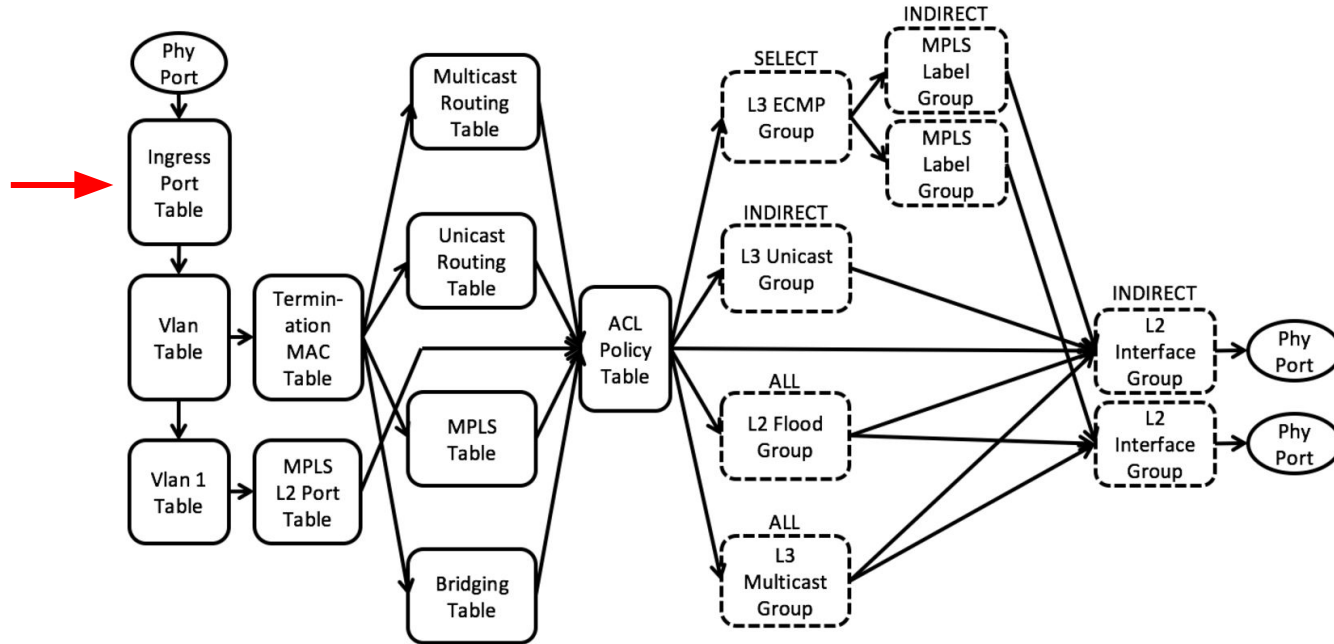
GROUP

TABLE



Pipeliner example

OF-DPA Pipeliner, L2 unicast



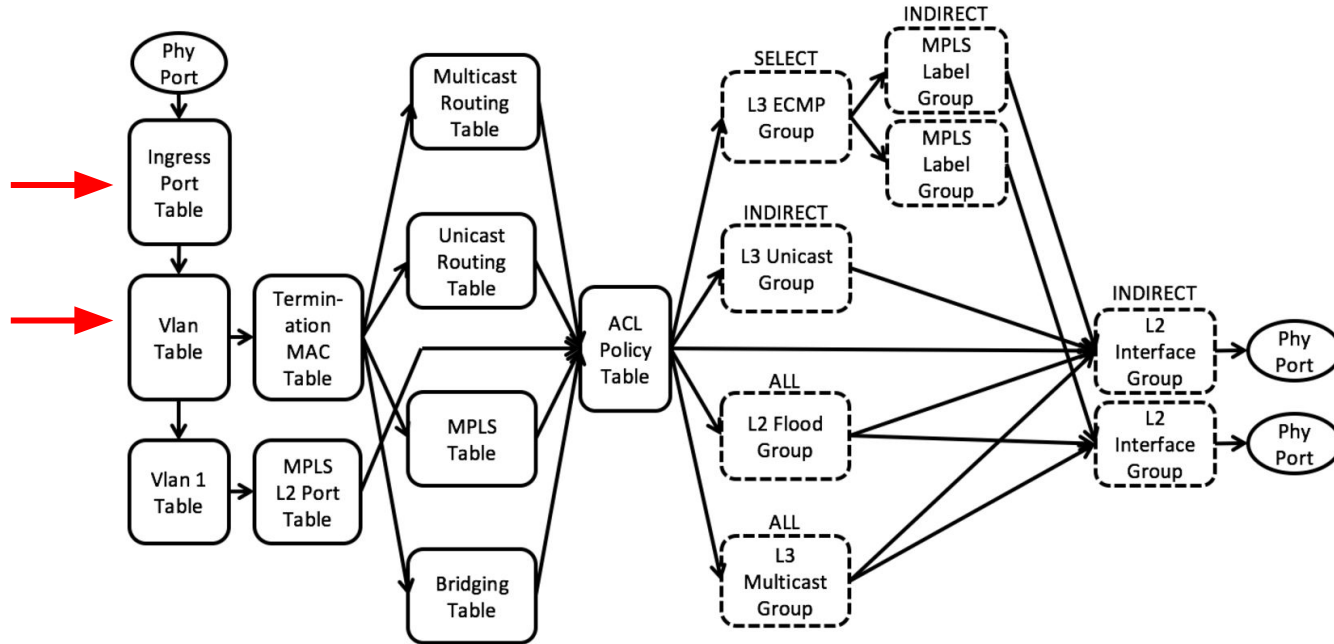
GROUP

TABLE

Pipeliner example



OF-DPA Pipeliner, L2 unicast



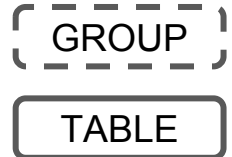
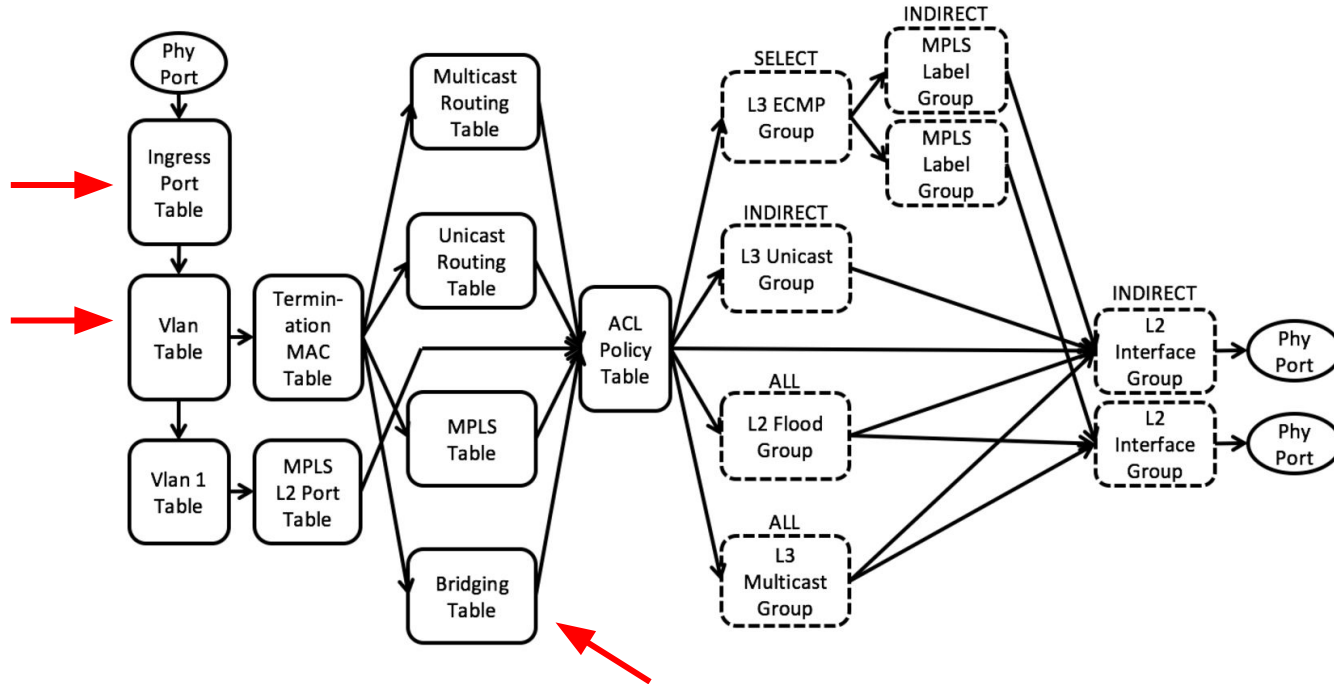
GROUP

TABLE

Pipeliner example



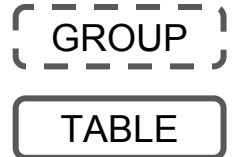
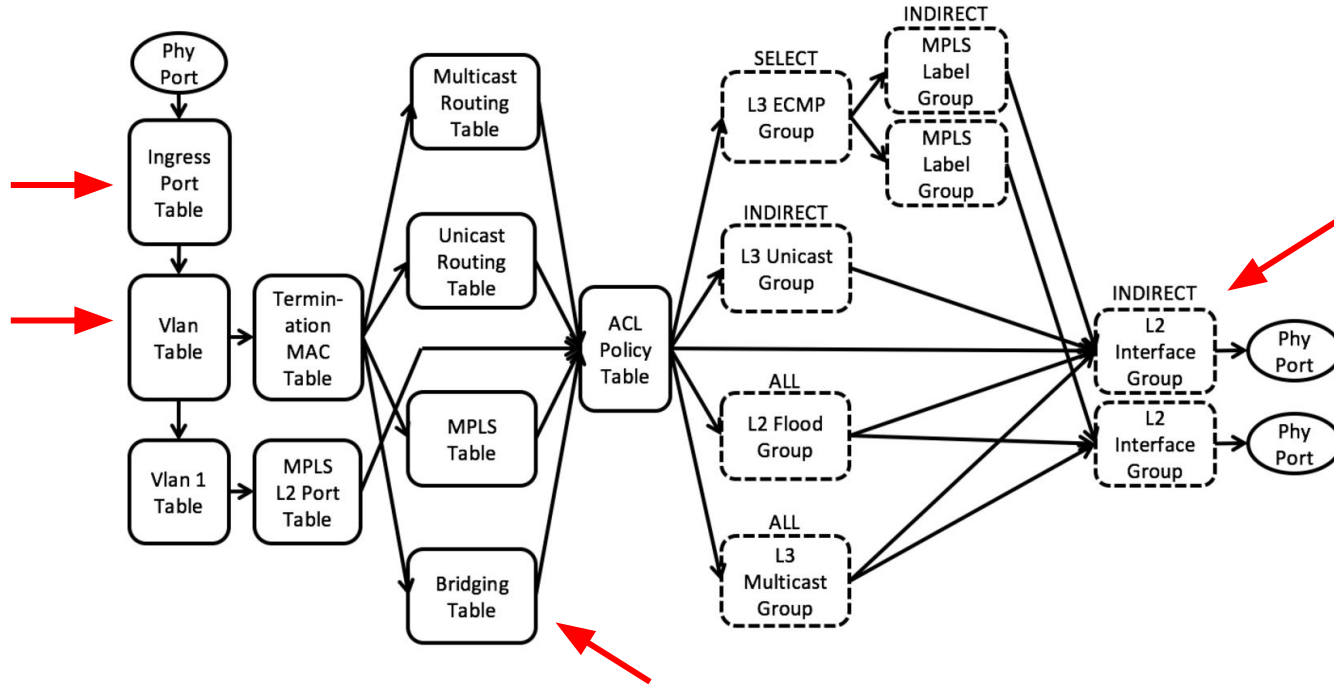
OF-DPA Pipeliner, L2 unicast



Pipeliner example



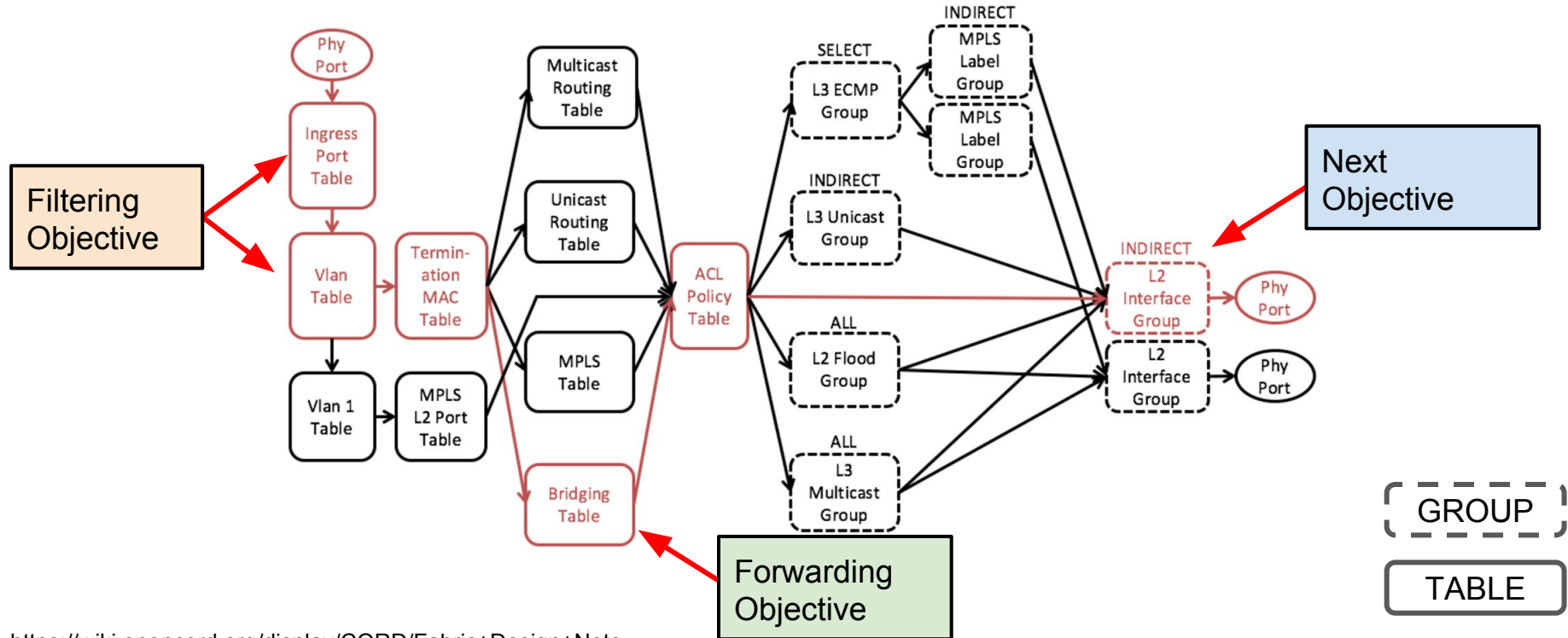
OF-DPA Pipeliner, L2 unicast



Pipeline example



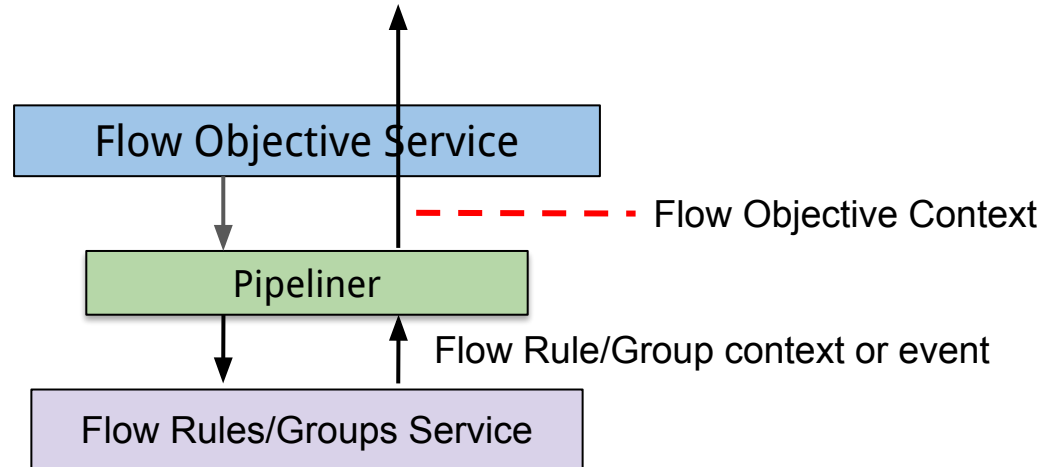
OF-DPA Pipeliner, L2 unicast



After applied to the Pipeliner



- Generates specific **Flow Rules** and **Groups** and installed by *Flow Rule Service* or *Group Service*
- Reports the status of Objectives by using **Flow Objective Context** (if exists).





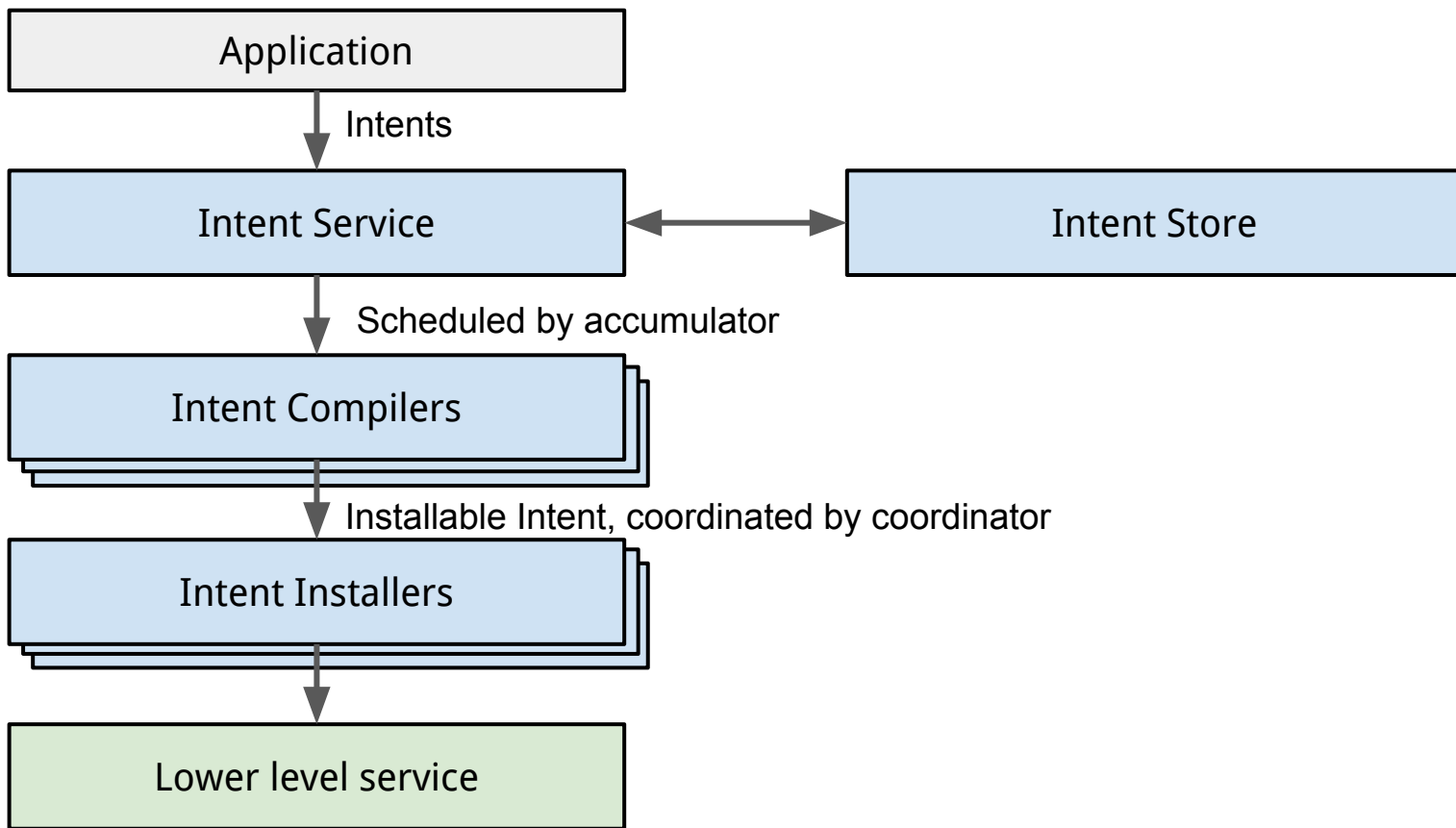
Intent Framework

Intent Framework

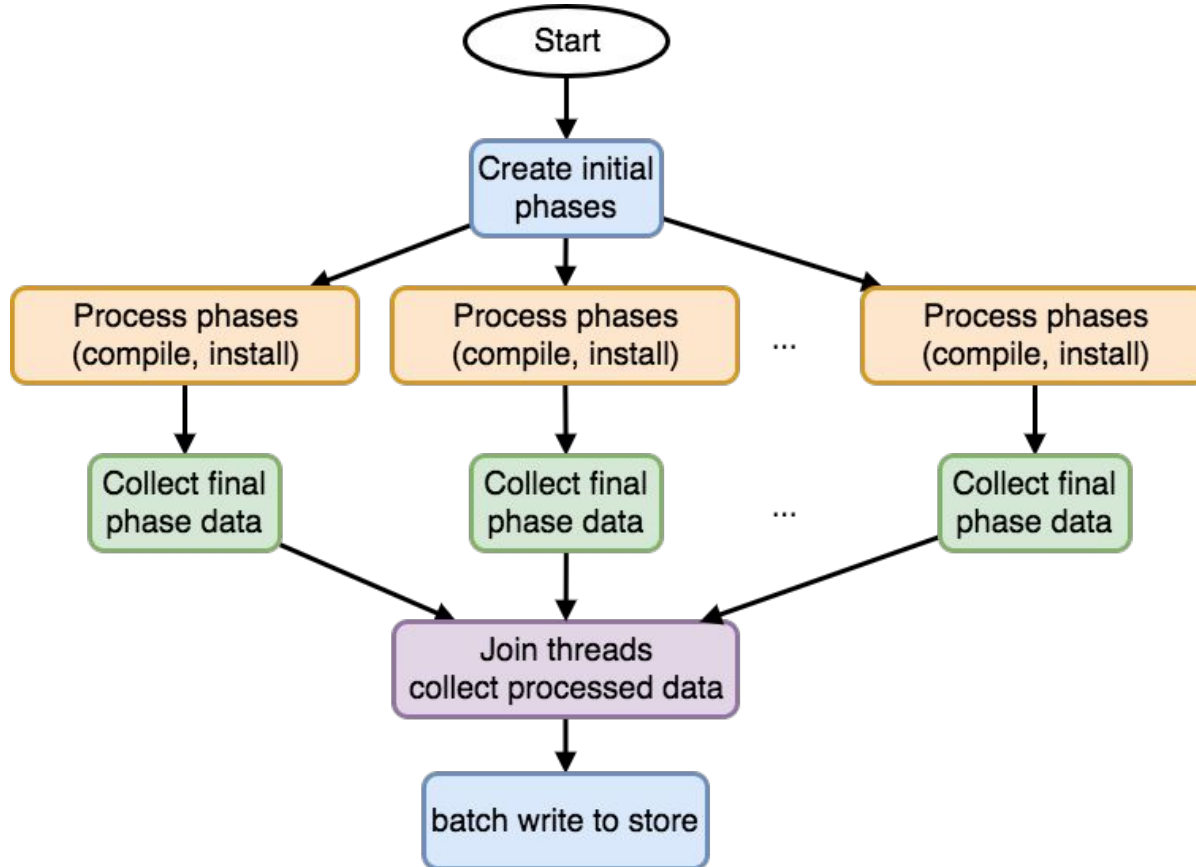


- **Topology independent, network-centric** interfaces
- Focuses on *what* should be done rather than *how* it is specifically programmed
- Abstracts unnecessary network complexity from applications

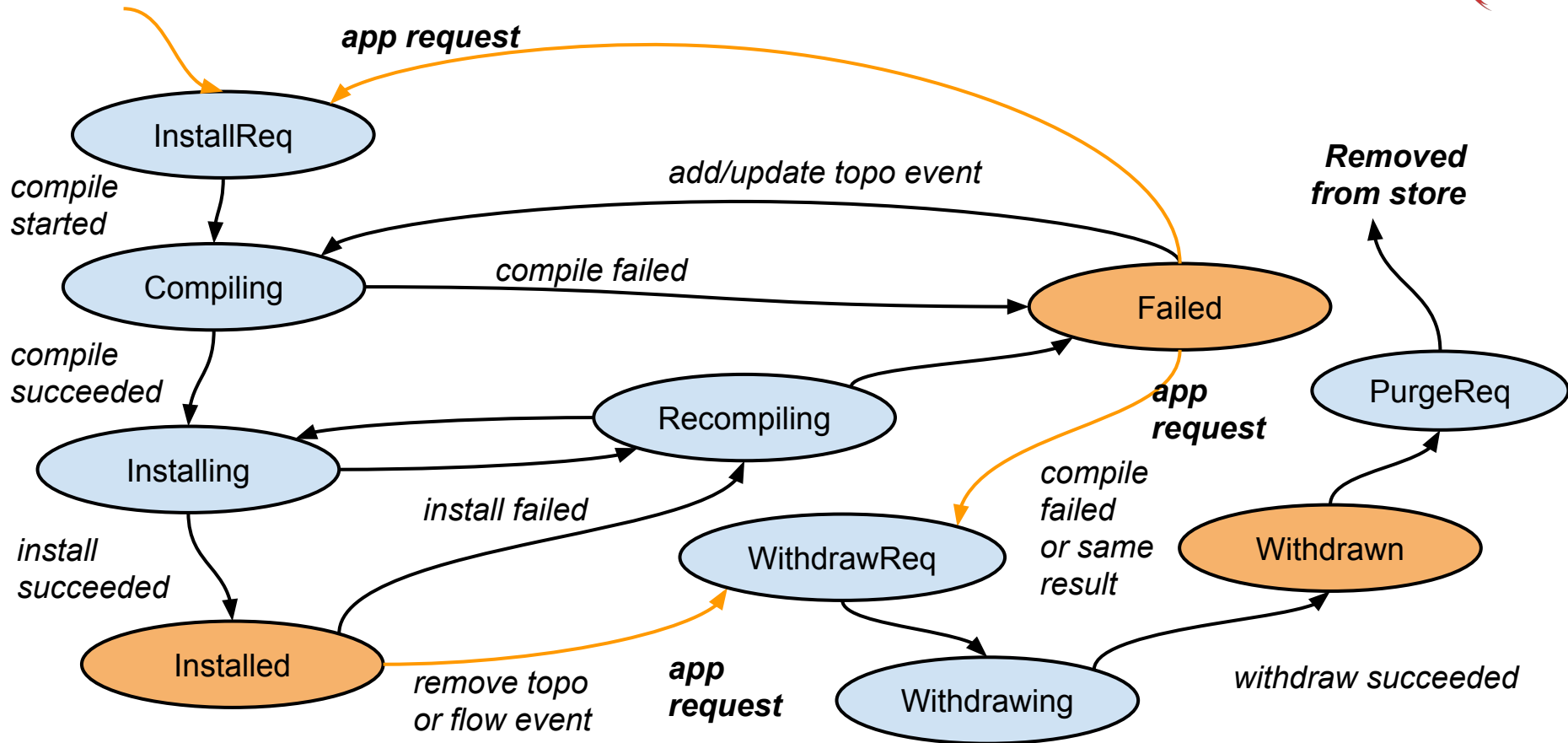
Intent Framework Architecture



Multi-thread Intent processing



Intent State Machine



Extensible

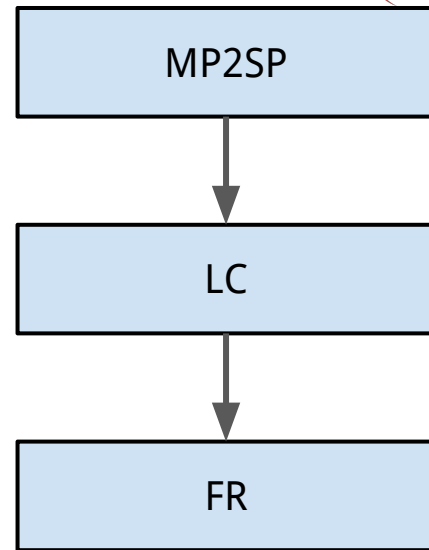


The Intent Framework can be **easily extendable**:

- New Intent compiler
- New Intent installer
- New routing policy, rule or constraint

Intent Compiler

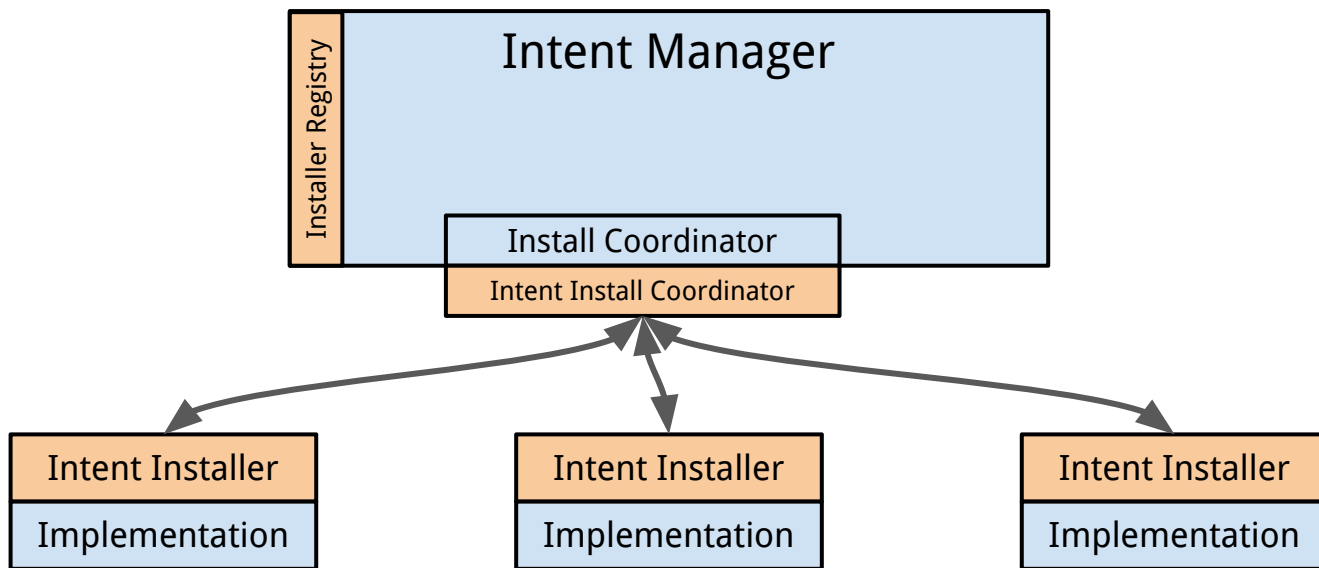
- Compiles an Intent to lower level Intent(s)
- Installable Intent: lowest level Intents
 - Flow Rule Intent
 - Flow Objective Intent
 - Domain Intent
 - Protection Endpoint Intent



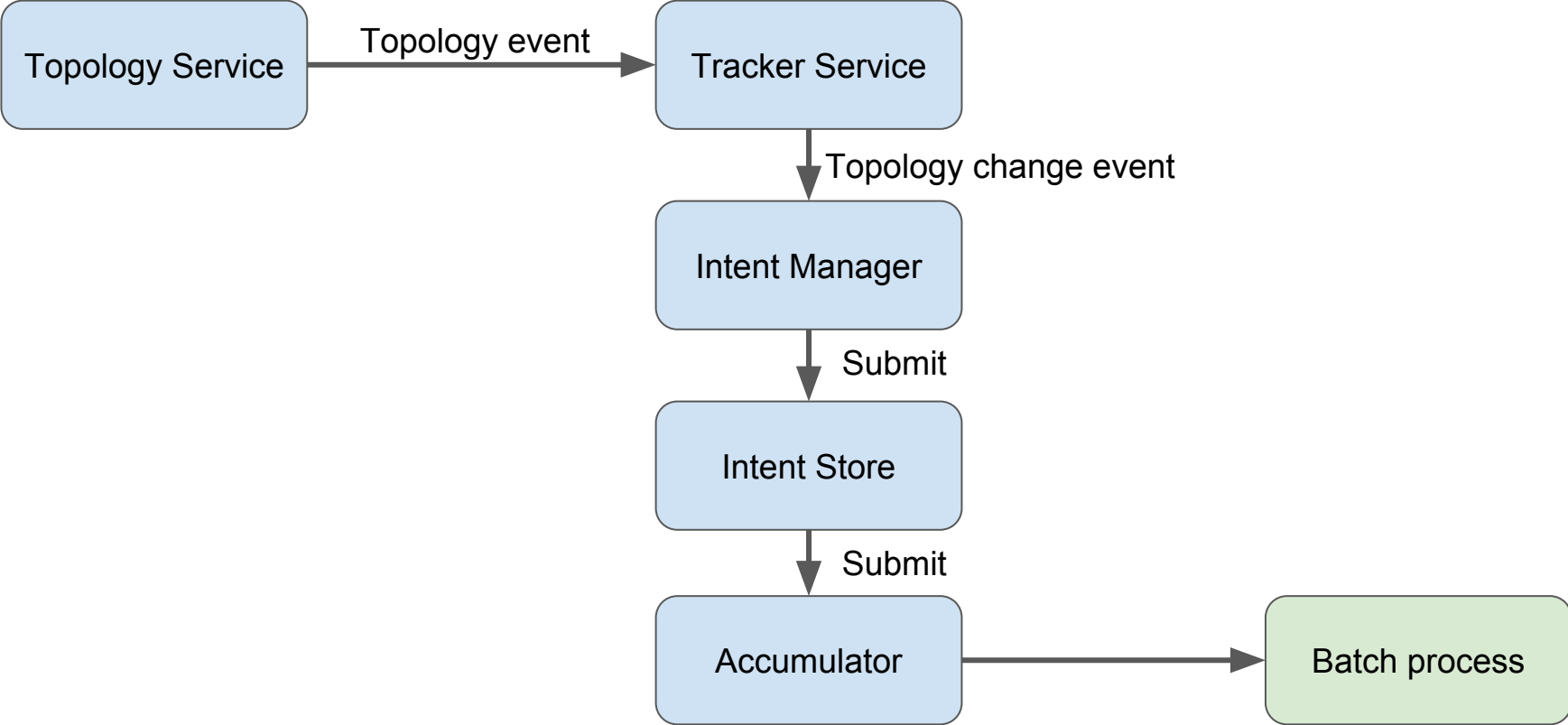
Intent Installer



- Install/Uninstall installable Intents
- May optimize install/uninstall/reinstall process
- Extendable(1.10 Kingfisher~)



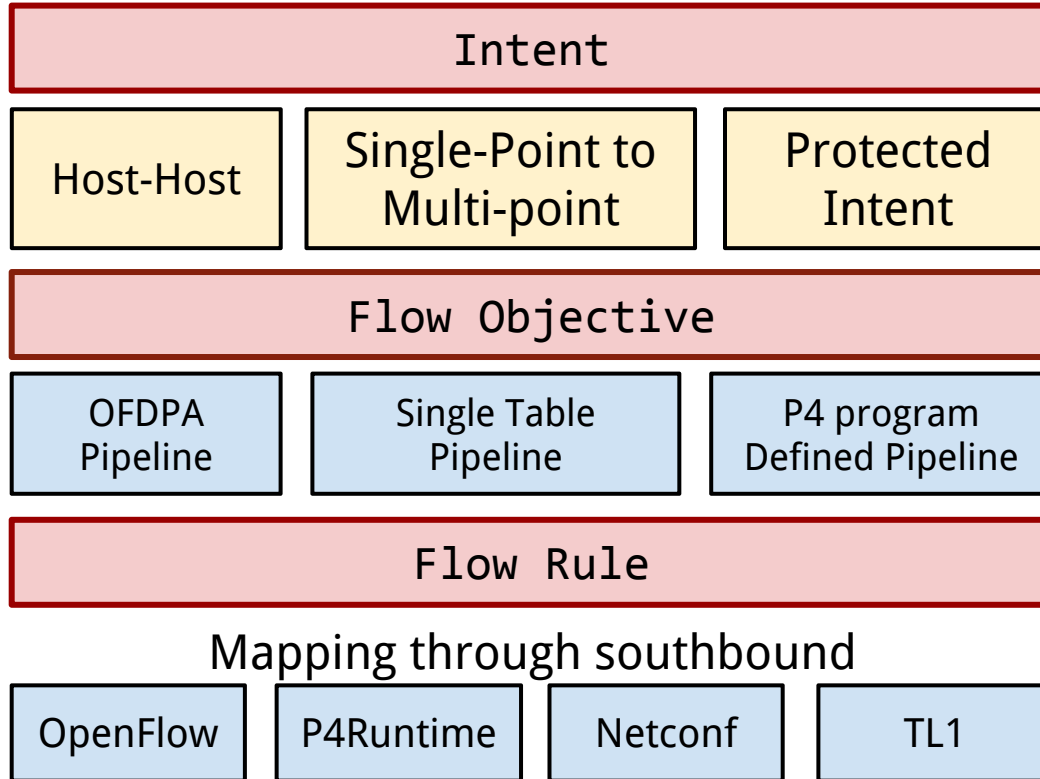
Link/Device failure handling for Intent



Network Programming



Abstract
to
concrete



Summary



- Reduced application complexity
 - **Reusability**
 - **Portability**
- **Easiness** of network programmability
- **Heterogeneous networks**
 - **Device-agnostic** behavior
 - **Protocol-agnostic** behavior
 - **Pipeline-agnostic** behavior
- **High availability, scalability and performance**



Software Defined Transformation of Service Provider Networks

Join the journey @ onosproject.org

Backup slides

Intent Information



- **IntentData:** (internal data structure)
 - Metadata for the top-level Intent
 - Has reference to installable Intents, etc.
- **IntentKey:**
 - Unique identifier for a network "Intention"
 - IntentKey will not change when you update route, etc.
- **IntentId:**
 - Unique identifier for an Intent object
 - IntentId will change when you update route, etc.

Reaction to failure for Intents



- Each Intent has the **network resources** registered
- All the Intents that are involved in the failure will be re-compiled.
- The recompilation process is a **multi-threading** process that:
 - Resubmit Intents
 - Finds a new entire path from source to destination
 - Generates new Installable Intents and reinstall (may optimized by Intent installer)