

ONOS+P4 Tutorial Hands-on Activity

ONK18

Link to these slides:

<http://bit.ly/onk18-hands-on>

Recap of ONOS and P4Runtime

- Slides from yesterday

<http://bit.ly/onos-p4-onk18>

Tutorial VM

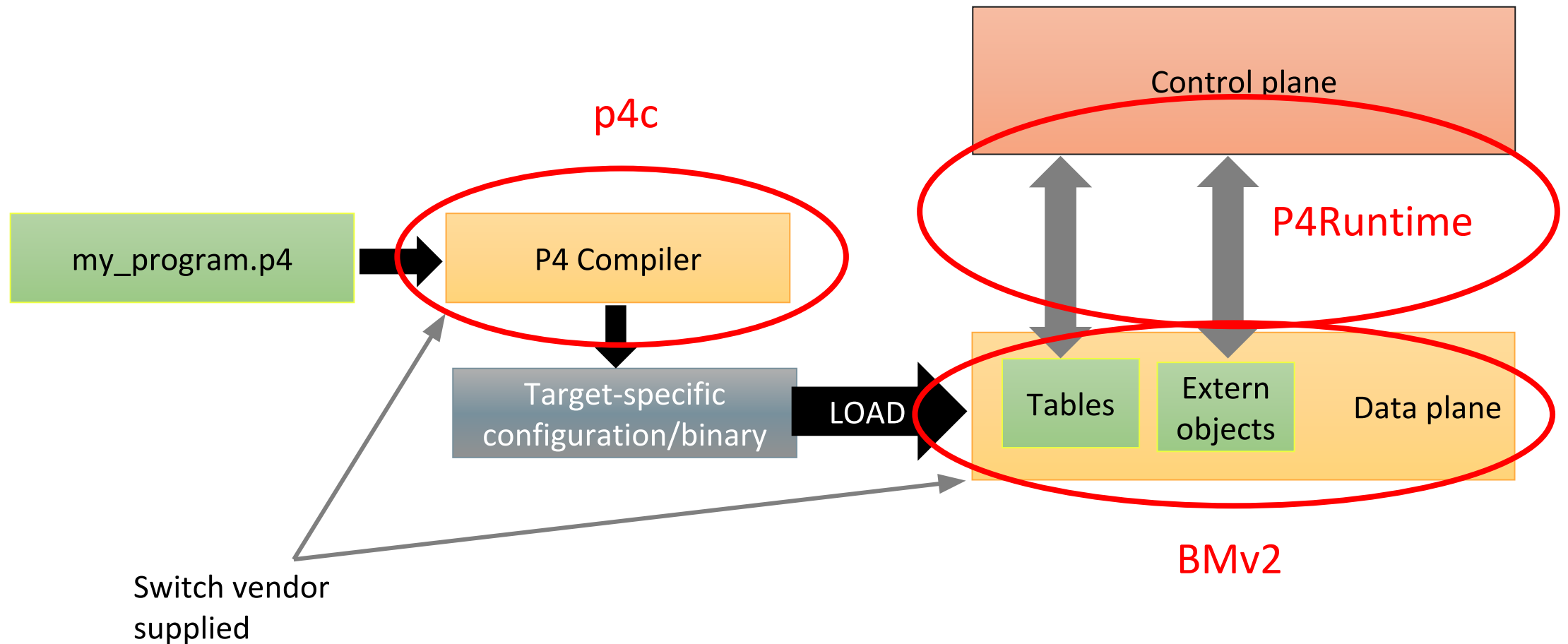
- Instructions to download and import the VM:

<http://bit.ly/onk18-p4-vm>

- To complete the tutorial exercises you will need to download, build and run the latest development version of ONOS.

```
cd $ONOS_ROOT  
git pull origin master  
buck build onos
```

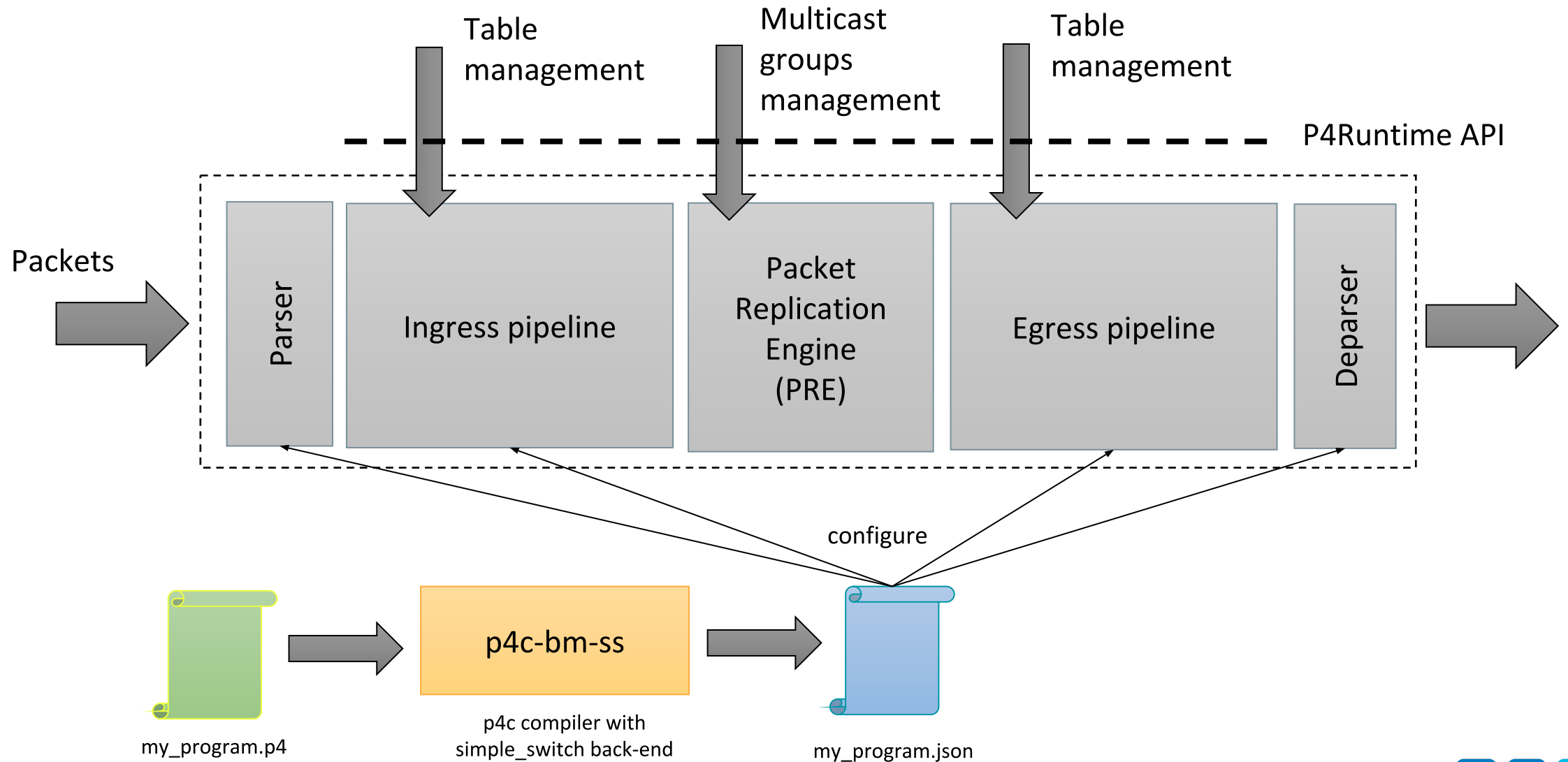
P4 tools overview



BMv2

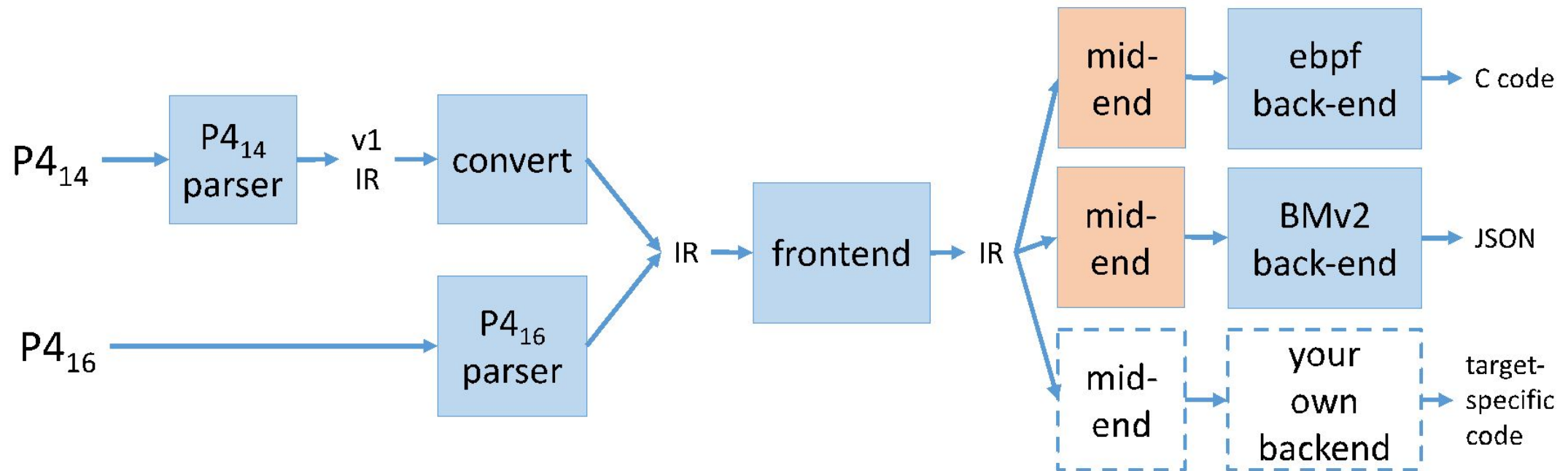
- <https://github.com/p4lang/behavioral-model>
- BMv2 = Behavioral-Model version 2
- User-space software switch to emulate a P4 data plane
- Aimed at being 100% conformant to the P4 specification
 - Performance is non-goal, i.e. low throughput
- Architecture-independent
 - Mostly generic code which can be used to implement any P4 target
 - Some targets provided as part of the repo, e.g. `simple_router`, `simple_switch`

BMv2's simple_switch_grpc target



p4c

- <https://github.com/p4lang/p4c>
- Open-source front-end compiler
 - Generates P4Info
- Support multiple back-ends (vendor-supplied)
 - Generate code for ASICs, NICs, FPGAs, software switches and other targets
 - Some of them are open-source (BMv2, eBPF)



Example of p4c command

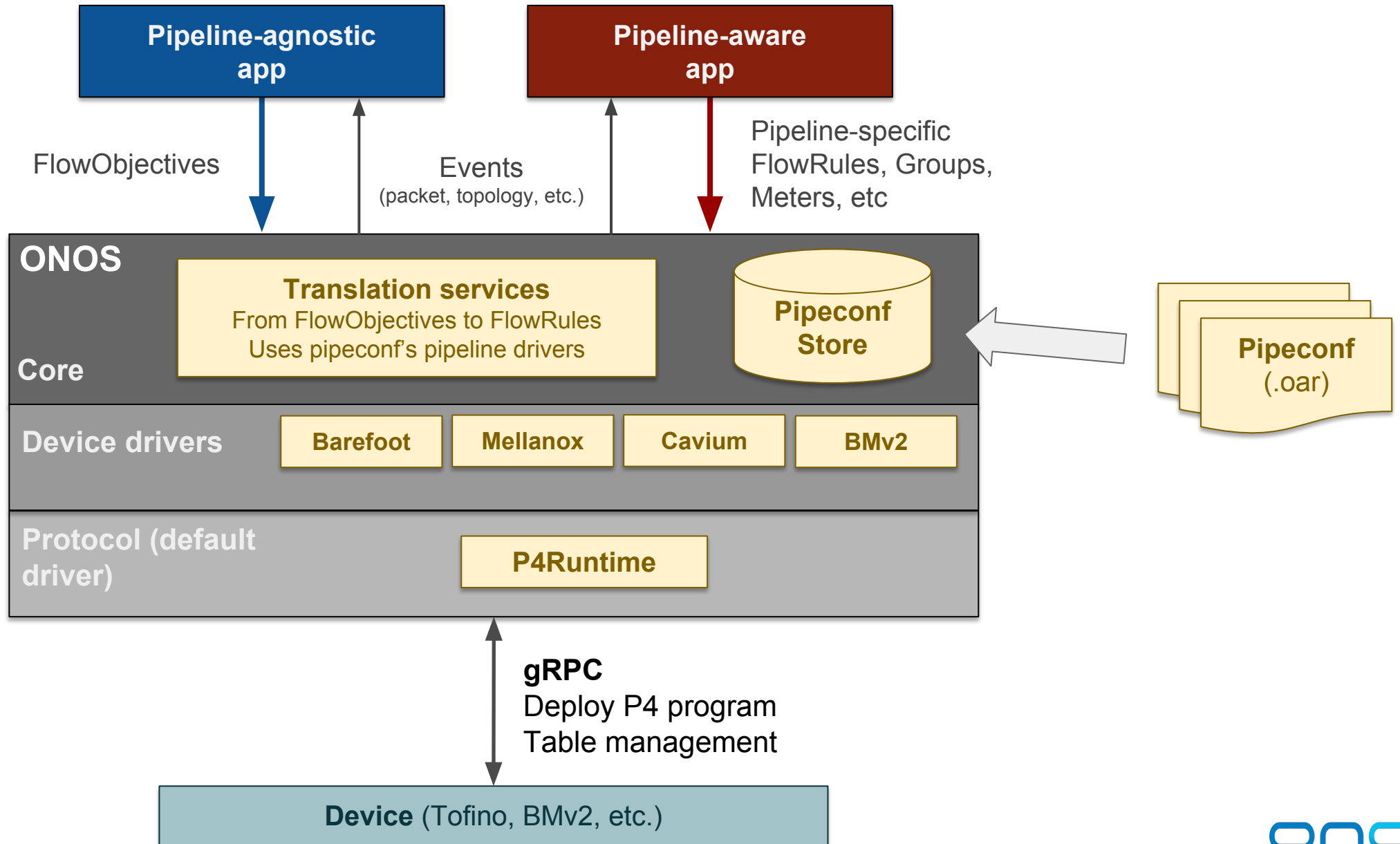
```
p4c-bm2-ss -o mytunnel.json \  
  --p4runtime-file mytunnel.p4info \  
  --p4runtime-format text \  
  mytunnel.p4
```

Or check Makefile under:

```
$ONOS_ROOT/apps/p4-tutorial/pipeconf/src/main/resources/Makefile
```


P4 support in ONOS

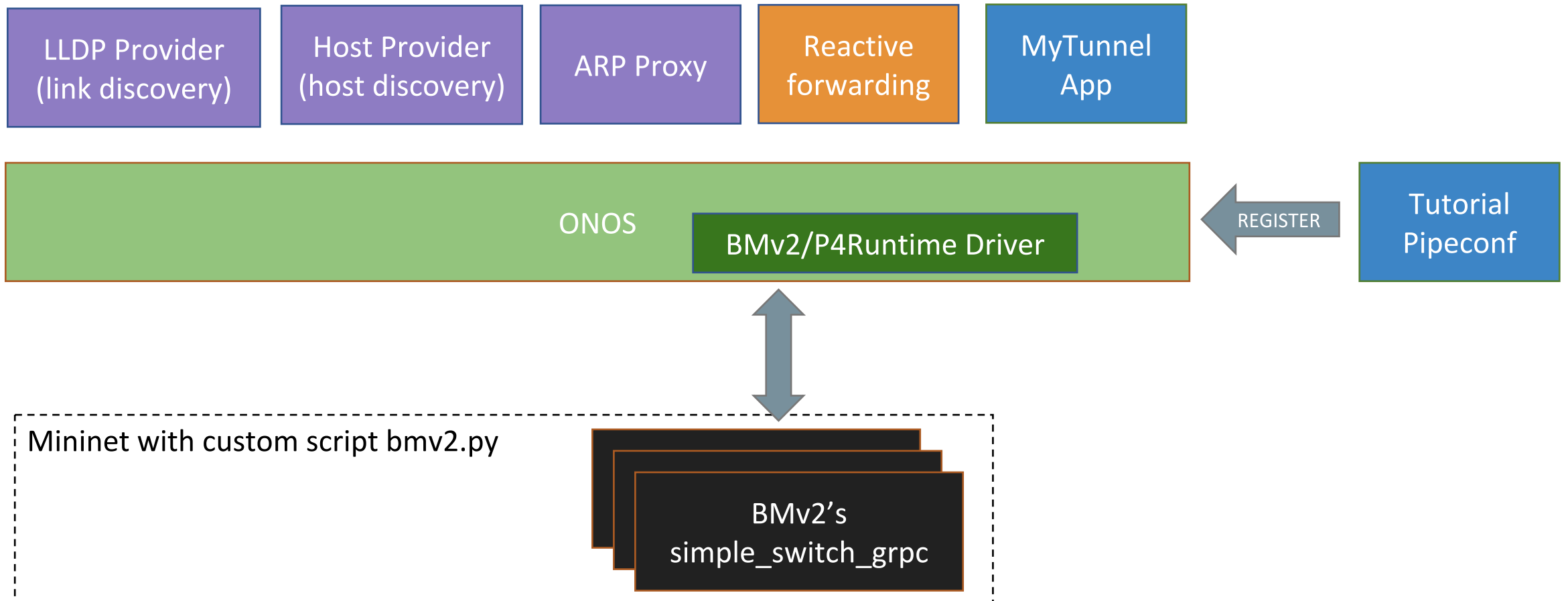
9



Exercises overview

- **Run ONOS with BMv2/P4Runtime support**
- **Load pipeconf**
- **Run Mininet with BMv2**
- **Run applications to provide connectivity between hosts**
 - Exercise 1: pipeline-agnostic application (Reactive Forwarding)
 - Exercise 2: pipeline-aware application (MyTunnelApp)
 - Requires to write few lines of Java code

ONOS Environment



LLDP Provider App

- Provides means to discover network links by injecting LLDP packets in the network
 - Install packet request (flow rule) on each device
 - Match: `ETH_TYPE = LLDP`
 - Instructions: `OUTPUT(CONTROLLER)`
 - Periodically issues LLDP packets via packet-out for each switch port

ARP Proxy App

- Intercepts ARP requests sent by hosts via packet-in, generates ARP replies. I.e. ARP packets are not forwarded on the network, but are handled exclusively by the controller
 - Match: `ETH_TYPE = ARP`
 - Instructions: `OUTPUT(CONTROLLER)`

Host Location Provider

- Learns location of hosts by passively intercepting ARP packets
 - Mapping: ETH/IP addr \leftrightarrow Switch-ID:Port-ID
- No rules installed, listens for packet requests installed by other applications (e.g. ARP proxy)

Reactive Forwarding App

- Intercepts IP packets for which there are no matching flow rules on the switch
- If the location of the destination host is known, installs the necessary flow rule to forward subsequent packets
- Otherwise, packet is flooded on all ports of the switch where it was received via packet-out
- Flow rules
 - IP packet intercept
 - Match: `ETH_TYPE = IPv4`, Instruction: `OUTPUT(CONTROLLER)`
 - Forwarding
 - Match: `IN_PORT, ETH_SRC, ETH_DST`, Instruction: `OUTPUT(port)`

Mininet script: bmv2.py

- Provided with ONOS
- Executes BMv2's `simple_switch_grpc` inside Mininet
- Automatically generates and push to ONOS the `netcfg.json`
 - With P4Runtime server IP/port and pipeconf ID
 - Files located under `/tmp`
- Example command to use `bmv2.py`

```
sudo -E mn --custom $BMV2_MN_PY \  
  --switch onosbmv2,pipeconf=p4-tutorial-pipeconf  
  --controller remote,ip=127.0.0.1
```

`$BMV2_MN_PY` env variable that points to the location of the `bmv2.py` script

ONOS terminology

- **Criteria**

- Match fields used in a FlowRule

- **Traffic Treatment**

- Actions/instructions of a FlowRule

- **Pi* classes**

- Classes used to describe protocol-independent constructs
- Equivalent of P4Runtime generic entities
- Examples
 - PiTableId: name of a table as in the P4 program
 - PiMatchFieldId: name of a match field in a table
 - PiCriterion: list of match fields each one defined by its name and value
 - PiAction: action defined by its name and list of parameters

Known limitations

- **Cannot restart Mininet while ONOS is running**
- **Need to restart ONOS before restarting Mininet**
 - To kill ONOS, press ``ctrl-c`` in the ONOS log terminal window.
 - To kill Mininet, press ``ctrl-d`` in the Mininet CLI or type ``exit``

Exercise instructions

Open file in VM:

`~/onos/apps/p4-tutorial/README.md`

Or, use nice Markdown preview in GitHub:

<http://bit.ly/onos-p4-readme>

Links

- Instructions to download and import the VM:

<http://bit.ly/onk18-p4-vm>

- To complete the tutorial exercises you will need to download, build and run the latest development version of ONOS.

```
cd $ONOS_ROOT  
git pull origin master  
buck build onos
```

Tutorial exercises: **<http://bit.ly/onos-p4-readme>**

These slides:

<http://bit.ly/onk18-hands-on>

Bonus 3rd exercise: modify P4 program

- Look for `t_tunnel_ingress` and `t_tunnel_fwd`
- We need to add counters to these 2 tables
- Look for the way we have added counters to `t_l2_fwd`
 - First, we instantiate the counter
 - `direct_counter(CounterType.packets_and_bytes) l2_fwd_counter;`
 - Then, we added the counter instance to `t_l2_fwd`
 - `counters = l2_fwd_counter;` (Inside `t_l2_fwd` table definition)
- Do the same for `t_tunnel_ingress` and `t_tunnel_fwd`
 - Using a different name for the counter instance, i.e. not `l2_fwd_counter`
- Re-compile `mytunnel.p4`
 - `cd ~/onos/apps/p4-tutorial/pipeconf/src/main/resources`
 - `make`
- Execute again Exercise 2
 - When executing ok clean ONOS will be rebuilt automatically, and so the pipeconf
- Now you should see flow counters being updated in the ONOS CLI and UI

ONOS-P4 Brigade - Join the effort!

22

Learn more - P4 Brigade Wiki:

<https://wiki.onosproject.org/display/ONOS/P4+brigade>

P4 Brigade mailing list:

brigade-p4@onosproject.org

Weekly meetings:

Every Tuesday @ 09:00 AM or 10:00 PM US Pacific Time
(alternate times)

Contributions from many brigade members. Thank you!

Carmelo Cascone, Andrea Campanella, Jonghwan Hyun, Uyen Chau, Brian O'Connor (**ONF**),
Esin Karaman, Serkant Uluderya, Mehmed Mustafa, Ekber Aziz (**Netsia**), Frank Wang (**Inspur**),
Yi Tseng, Kevin Chuang, Nate Tang, Iver Liu (**NCTU**), Wu Shaoyong, Jian Tian, Ke Zhiyong (**ZTE**),
Tom Tofigh (**AT&T**), Phil Huang, Ahbee Wai (**Edgecore**)