Introduction Recap

- **Quizzes on Canvas** – not graded, but will check that credit bearing participants participate, please login first, can retake as often as you like
- **Discussion** – credit bearing course participants need to login to CANVAS, all open learners please post your thoughts in Slack

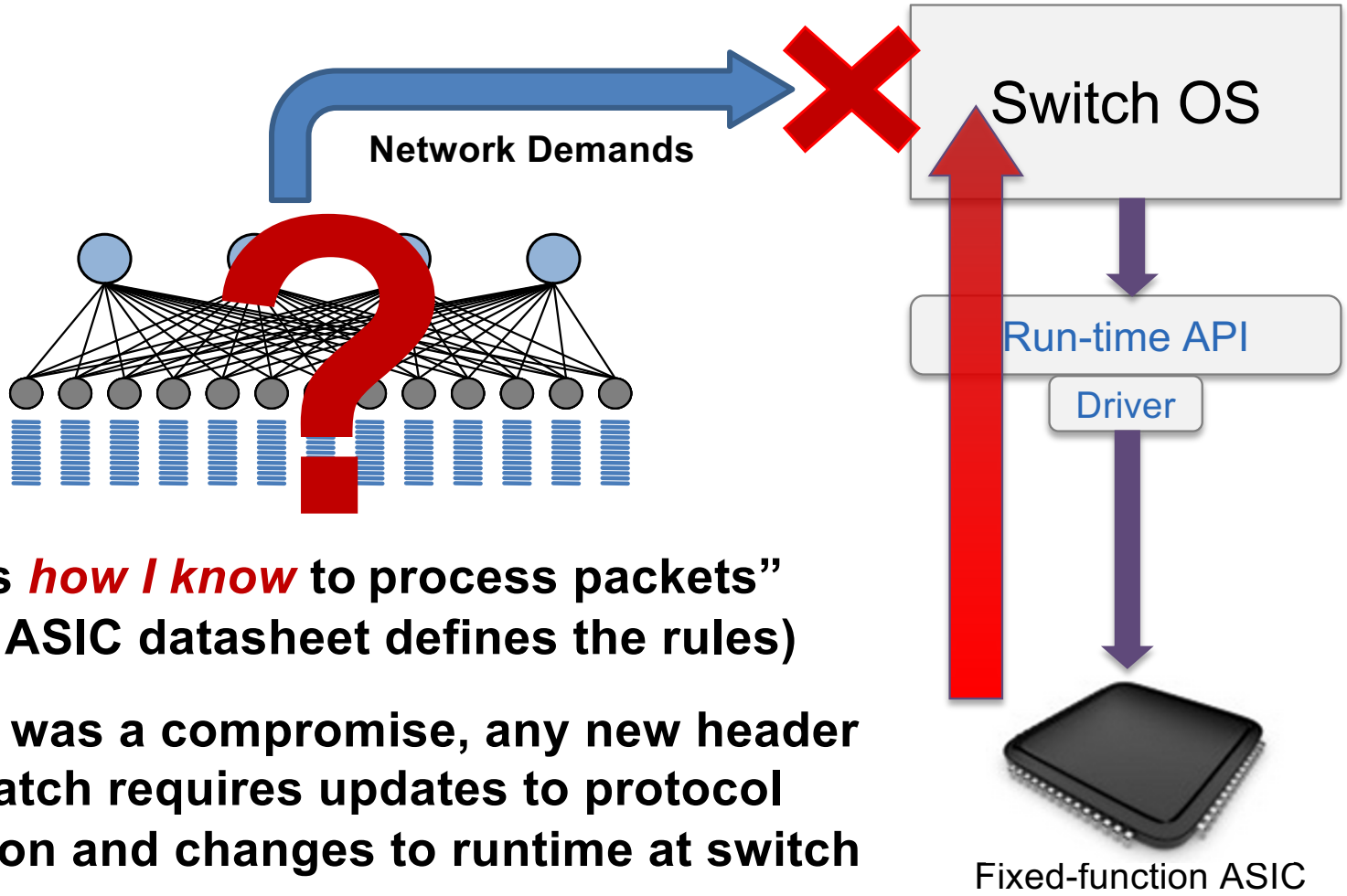
DVAD41 - Introduction to Data Plane Programming

Introduction to P4

Data Plane Programming

- **Motivation**

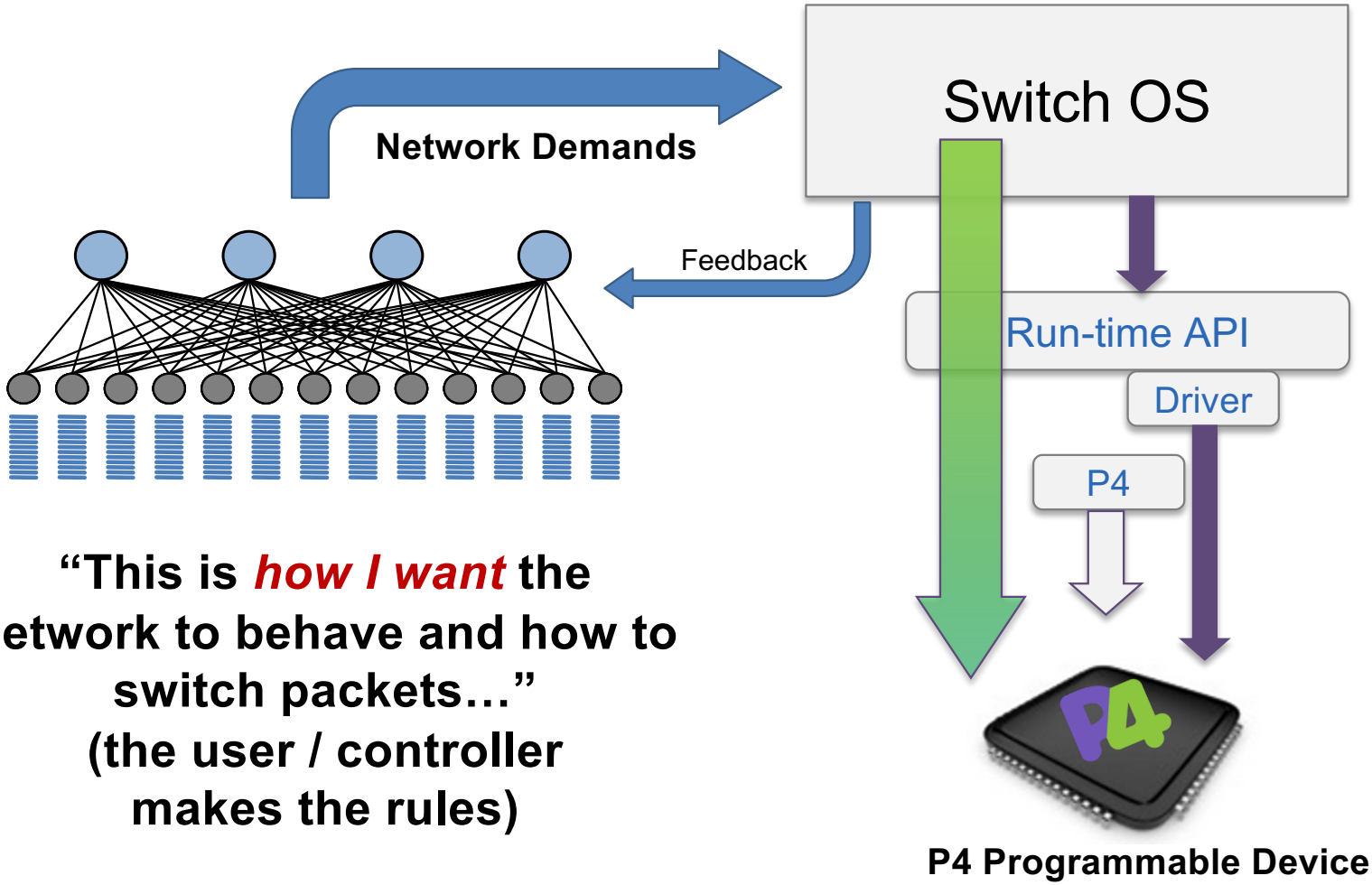
Status Quo: Bottom-up design



"This is *how I know* to process packets"
(i.e. the ASIC datasheet defines the rules)

OpenFlow was a compromise, any new header field match requires updates to protocol specification and changes to runtime at switch

A Better Approach: Top-down design



"This is *how I want* the network to behave and how to switch packets…"
(the user / controller makes the rules)

Benefits of Data Plane Programmability

- **New Features** – Add new protocols
- **Reduce complexity** – Remove unused protocols
- **Efficient use of resources** – flexible use of tables
- **Greater visibility** – New diagnostic techniques, telemetry, etc.
- **SW style development** – rapid design cycle, fast innovation, fix data plane bugs in the field
- **You keep your own ideas**

Think programming rather than protocols…

Programmable Network Devices

- **PISA: Flexible Match+Action ASICs**
  - Intel Flexpipe, Cisco Doppler, Cavium (Xpliant), Barefoot Tofino, …
- **NPU**
  - EZchip, Netronome, …
- **CPU**
  - Open Vswitch, eBPF, DPDK, VPP…
- **FPGA**
  - Xilinx, Altera, …

These devices let us tell them how to process packets.

What can you do with P4?

- **Layer 4 Load Balancer – SilkRoad[1]**
- **Low Latency Congestion Control – NDP[2]**
- **In-band Network Telemetry – INT[3]**
- **In-Network caching and coordination – NetCache[4] / NetChain[5]**
- **Aggregation for MapReduce Applications [7]**
- **… and much more**

[1] Miao, Rui, et al. "SilkRoad: Making Stateful Layer-4 Load Balancing Fast and Cheap Using Switching ASICs." SIGCOMM, 2017.
[2] Handley, Mark, et al. "Re-architecting datacenter networks and stacks for low latency and high performance." SIGCOMM, 2017.
[3] Kim, Changhoon, et al. "In-band network telemetry via programmable dataplanes." SIGCOMM. 2015.
[4] Xin Jin et al. "NetCache: Balancing Key-Value Stores with Fast In-Network Caching." To appear at SOSP 2017
[5] Jin, Xin, et al. "NetChain: Scale-Free Sub-RTT Coordination." *NSDI*, 2018.
[6] Dang, Huynh Tu, et al. "NetPaxos: Consensus at network speed." SIGCOMM, 2015.
[7] Sapio, Amedeo, et al. "In-Network Computation is a Dumb Idea Whose Time Has Come." *Hot Topics in Networks*. ACM, 2017.

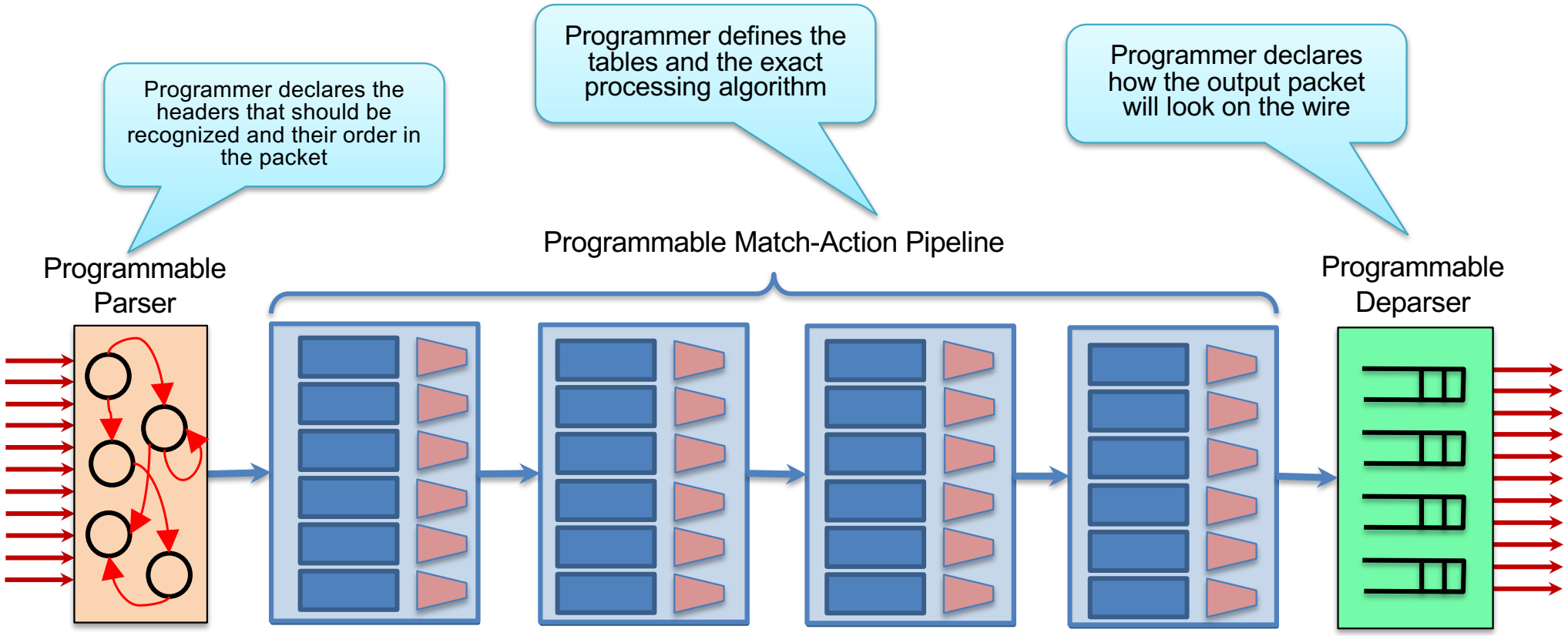
Brief History and Trivia

- May 2013: Initial idea and the name “P4”
- July 2014: First paper (SIGCOMM CCR)
- Aug 2014: First $P4_{14}$ Draft Specification (v0.9.8)
- Sep 2014: $P4_{14}$ Specification released (v1.0.0)
- Jan 2015: $P4_{14}$ v1.0.1
- Mar 2015: $P4_{14}$ v1.0.2
- Nov 2016: $P4_{14}$ v1.0.3
- May 2017: $P4_{14}$ v1.0.4

- Apr 2016: $P4_{16}$ – first commits
- Dec 2016: First $P4_{16}$ Draft Specification
- May 2017: $P4_{16}$ Specification released

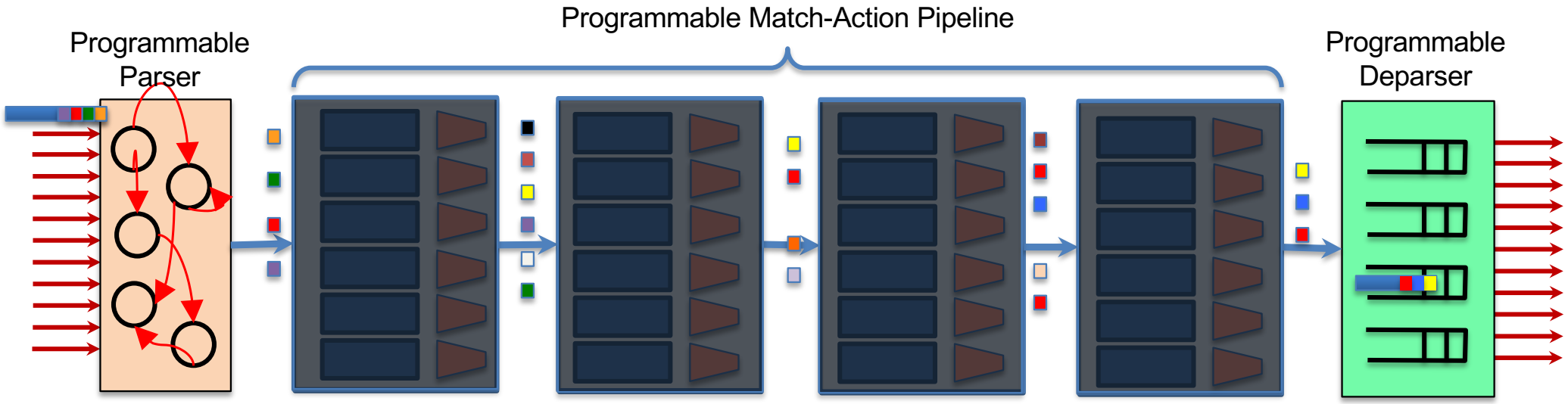
P4_16 Data Plane Model

PISA: Protocol-Independent Switch Architecture



PISA in Action

- Packet is parsed into individual headers (parsed representation)
- Headers and intermediate results can be used for matching and actions
- Headers can be modified, added or removed
- Packet is deparsed (serialized)



P4$_{16}$ Language Elements

Parsers — State machine, bitfield extraction

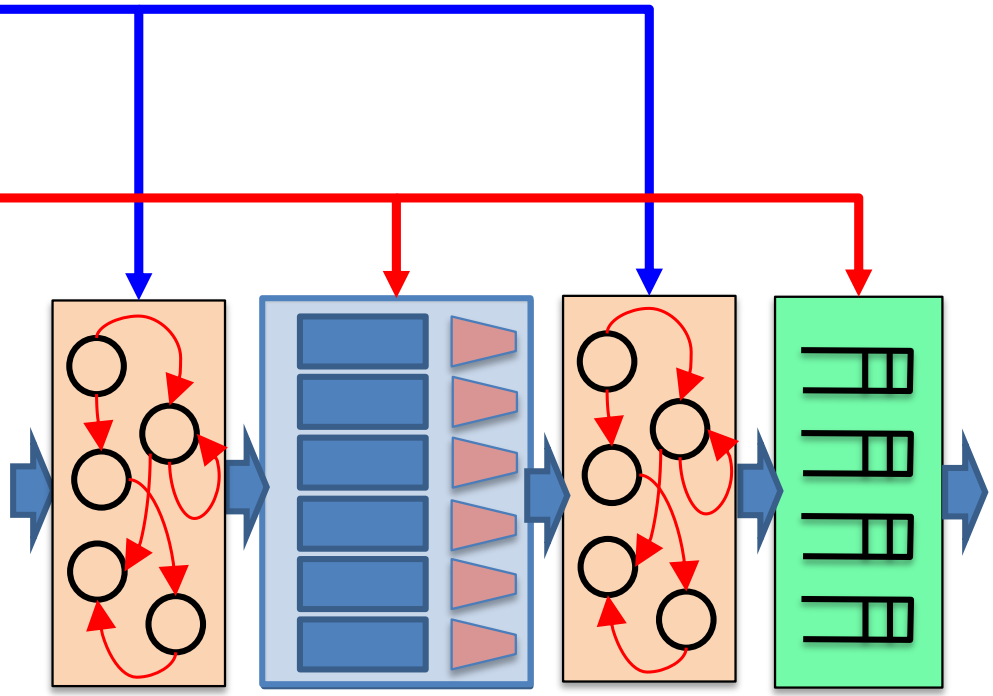
Controls — Tables, Actions, control flow statements

Expressions — Basic operations and operators

Data Types — Bistrings, headers, structures, arrays

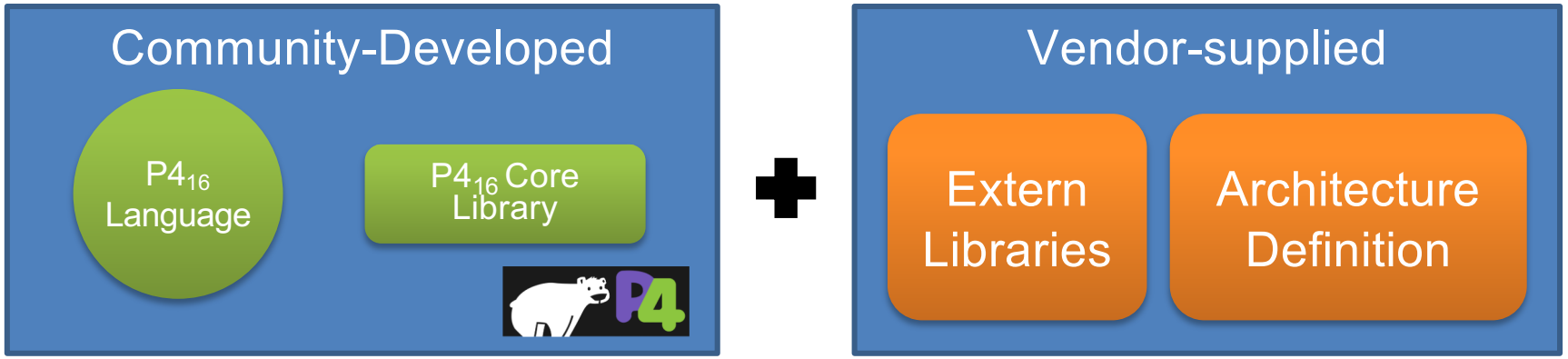
Architecture Description — Programmable blocks and their interfaces

Extern Libraries — Support for specialized components

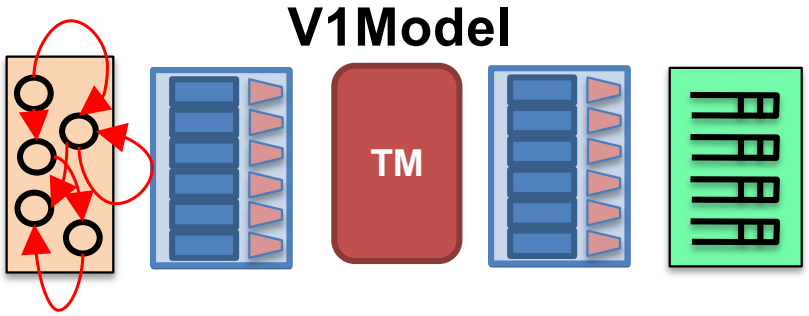


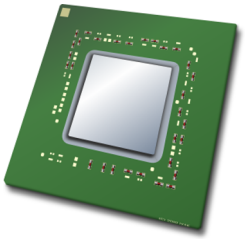
P4_16 Approach

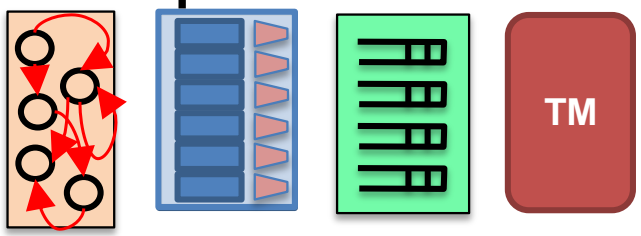
Term	Explanation
P4 Target	An embodiment of a specific hardware implementation
P4 Architecture	Provides an interface to program a target via some set of P4-programmable components, externs, fixed components

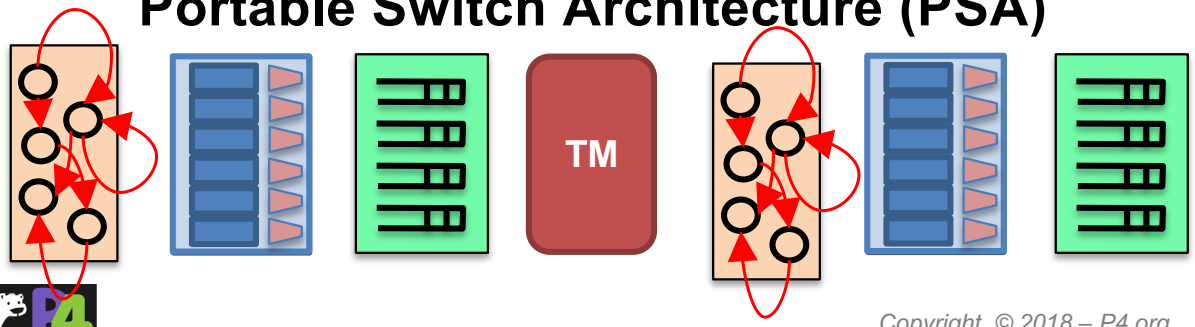


Example Architectures and Targets









Anything

Programming a P4 Target