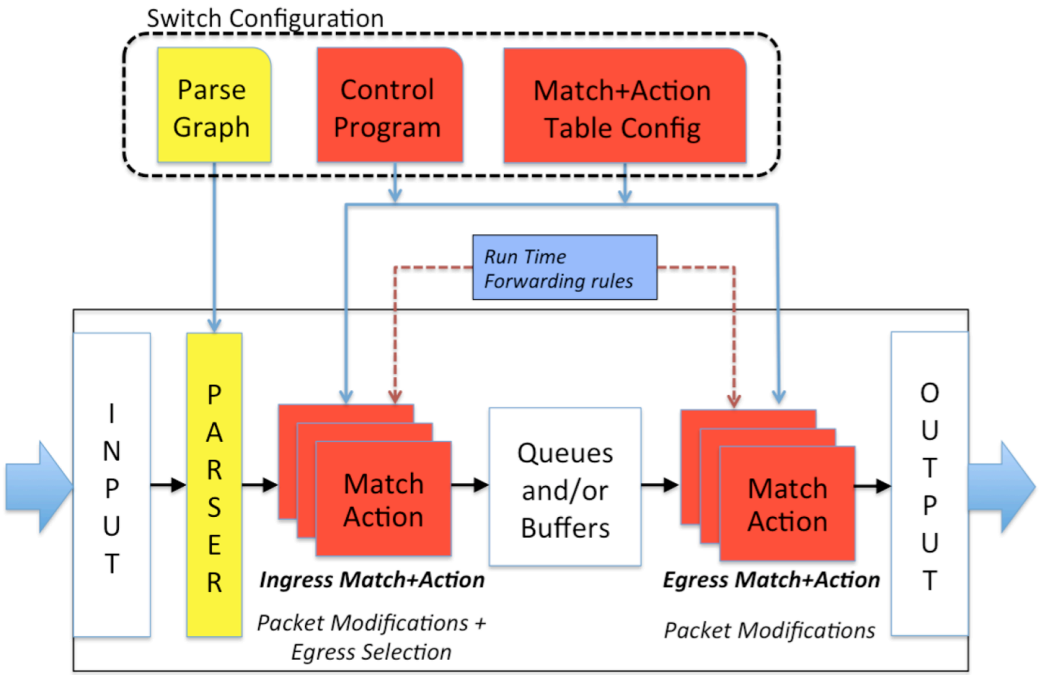
P4 Architectures

Calin Cascaval, Dan Daly

Motivation

- PISA (Protocol Independent Switch Architecture) is a single pipeline forwarding architecture
  - Defined the need for programming the data plane
  - $P4_{14}$ targeted PISA-like devices
- $P4_{16}$ has outgrown PISA
  - Need to target multiple programmable devices with different architectures
  - Separation of concerns



What is a P4 Architecture

- Architectures are the *programming model*:
  - The view of the pipeline targeted by the P4 program
  - How the P4 programmer thinks about the underlying platform (data plane)
  - May be different from the hardware target

Architectures in $P4_{16}$

- Architectures are a new capability in $P4_{16}$ to enable P4 on a diversity of devices:
  - Hardware: switches, routers, NICs
  - Software: OVS
- In general provide a logical view of the processing
- Architectures insulate programmers from the hardware details
  - Providers define architectures and implement compiler backends to map architectures to targets

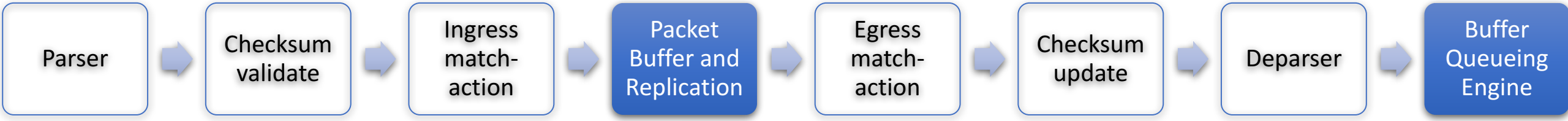
BAREFOOT NETWORKS

Portable Switch Architecture (PSA)

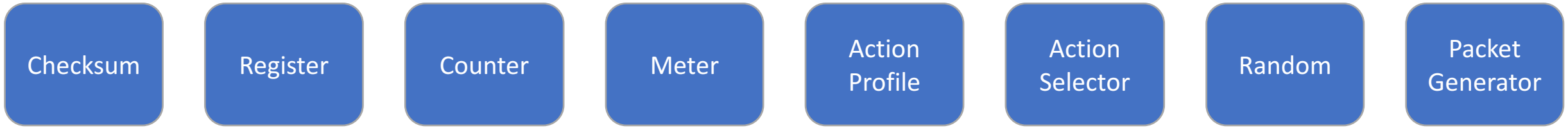
- A switch architecture specification that may be implemented on any switch target
- Goals
  - Composability
    - Multiple networking functions implemented on PSA combine into a program running on a single pipeline
  - Portability
    - Networking functions, e.g., INT, will work consistently across network devices
- Non-goals
  - Performance portability

PSA Pipeline (proposed)

Pipeline



Externs



Architecture Working Group

- Draft available for comments:
  - http://p4lang.github.io/p4-spec/docs/PSA.html

- Please join!
  - p4-arch@p4.org mailing list
  - github discussions

Backup

Outline

- What is a P4 architecture
  - A programming model equivalent
- The role of architectures in P4_16
  - new capability to allow for a diversity of devices
  - decouples P4 programs from targets
- The Portable Switch Architecture (PSA)
  - Goals: Composability, Portability, Comparability
  - Examples: ...