P4 Architecture Working Group Update

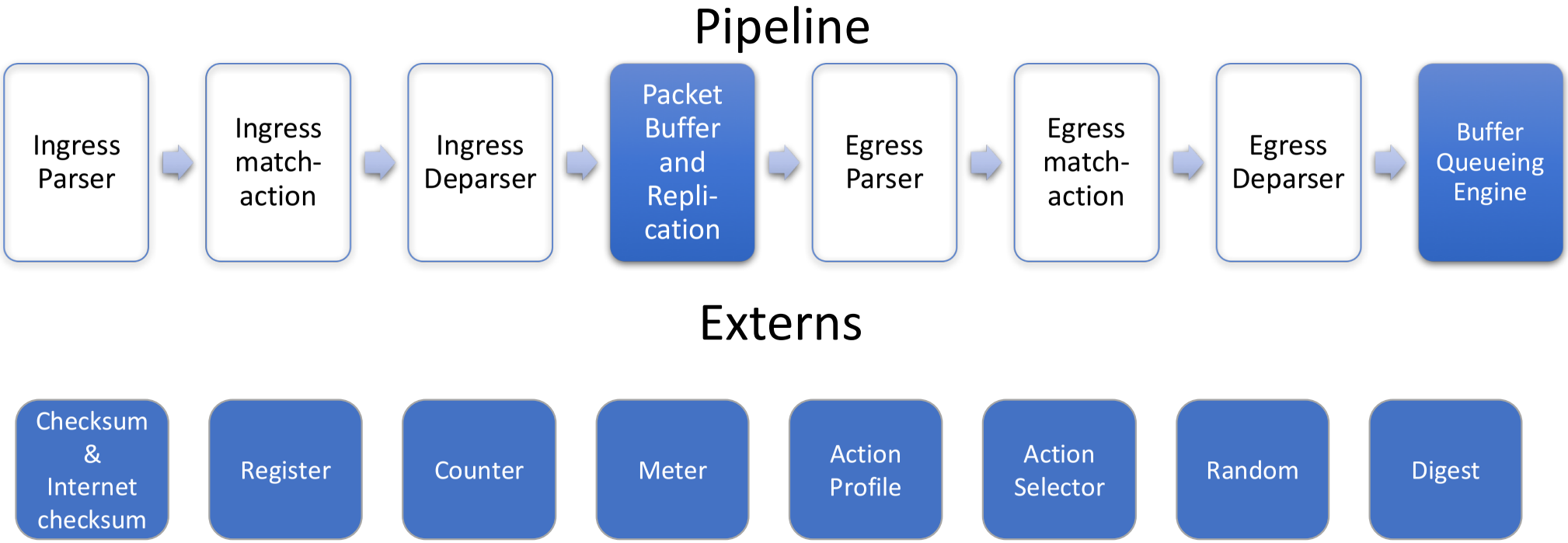
June 5, 2018

Calin Cascaval, Andy Fingerhut – working group co-chairs

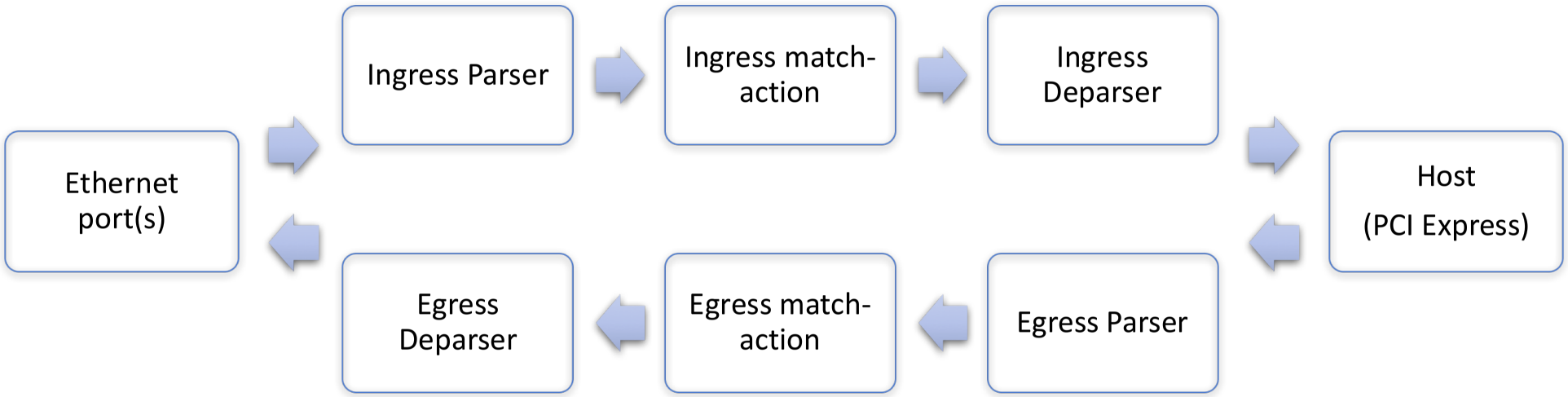
Portable Switch Architecture (PSA) v1.0

- Published March 1, 2018
- Purpose:
  - P4_14 defined the language and PISA switch architecture with ingress processing, packet buffer, then egress processing
  - P4_16 language spec defines the language constructs only, with a provision for defining multiple architectures
    - PSA is the first one published, and is very similar to PISA architecture
    - We expect vendors to implement PSA on their devices
      - Perhaps with custom extensions

PSA Pipeline



NIC Architecture

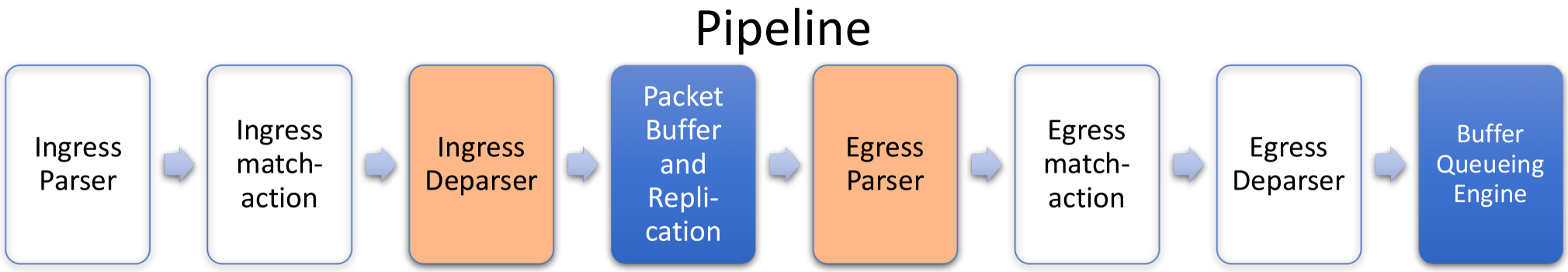


Feature	Comparison of P4_16+PSA to P4_14
Hash, Checksum, Random, Digest, Action Profile, Action Selector	Same, perhaps more precisely specified
Counters, meters	Same. Now updates to direct counters and meters are optional. Enables control plane the choice between actions that do updates, or not.
Registers	P4_16 spec requires explicit @atomic block to guarantee atomic behavior.
InternetChecksum	New subtract method enables incremental update of checksums, e.g. NAT
Timestamps	One when ingress begins, one when egress begins.
Unicast	Default ingress action is drop unless your P4 code changes that. P4_14 was ”send to port 0”
Multicast, Resubmit, Recirculate, Clone	Similar, but now much more precisely specified for better portability (next slide)

More precision for better portability

- Multicast, resubmit, recirculate, and clone operations, including:
  - How and where to invoke these operations
  - What metadata is carried with the packet
    - P4_16 no longer has field lists, so we defined another way to specify this.
  - Intrinsic metadata to distinguish these packets from others in your P4 code
  - Full example PSA programs in Github, excerpts in PSA spec
- PSA standard types for objects like PortId's, ClassOfService, MulticastGroup, etc.
  - Enable vendors to have custom bit widths for these values in their devices, but common widths in P4Runtime control plane API

PSA explicit ingress deparse & egress parse



Future plans

- Status of implementation in open source p4lang projects
- Enabling ECN marking and congestion control features
- Work with P4 Applications group on enabling full INT functionality, e.g. queue length observation in PSA
- Work with P4 API group on table timeout notifications, control plane types
- New P4_16 language features like optional parameters
- Compiler and BMv2 implementation