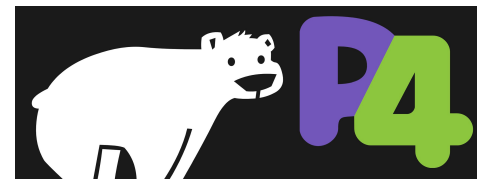


# P4 API WG

## Progress Report and Future Plans

5th P4 Workshop, June 5, 2018

Antonin Bas, Lorenzo Vicisano



# Outline

- The P4 API Working Group
- P4Runtime Primer
- Status Update
- Future Plans

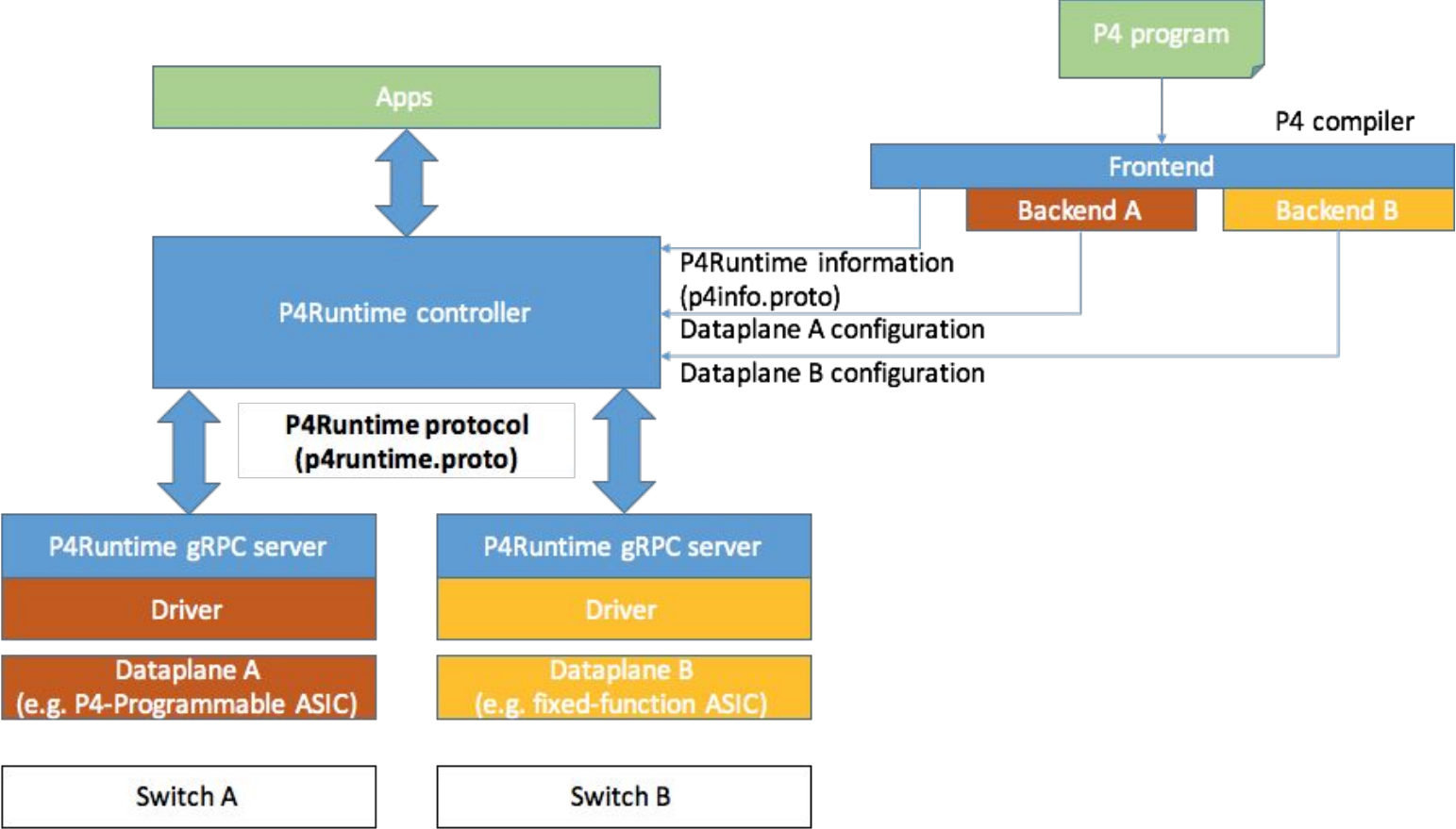
# The P4 API Working Group

- From the WG [charter](#):
  - *“Since the runtime management of P4 objects is out of the scope of the P4 language itself, this Working Group is chartered to define protocol-independent runtime APIs for P4 device management.”*
- The WG was chartered to design and standardize vendor-independent runtime APIs for P4-defined dataplanes
- A P4 runtime API should provide the following functionality:
  - Managing the behavior of the dataplane objects described in a P4 pipeline and retrieving information from these objects when needed (e.g. adding entries to a match-action table, removing them and retrieving the table content).
  - (Re)configuring the forwarding pipeline in the device, by enabling the application to provide a new compatible P4 pipeline.

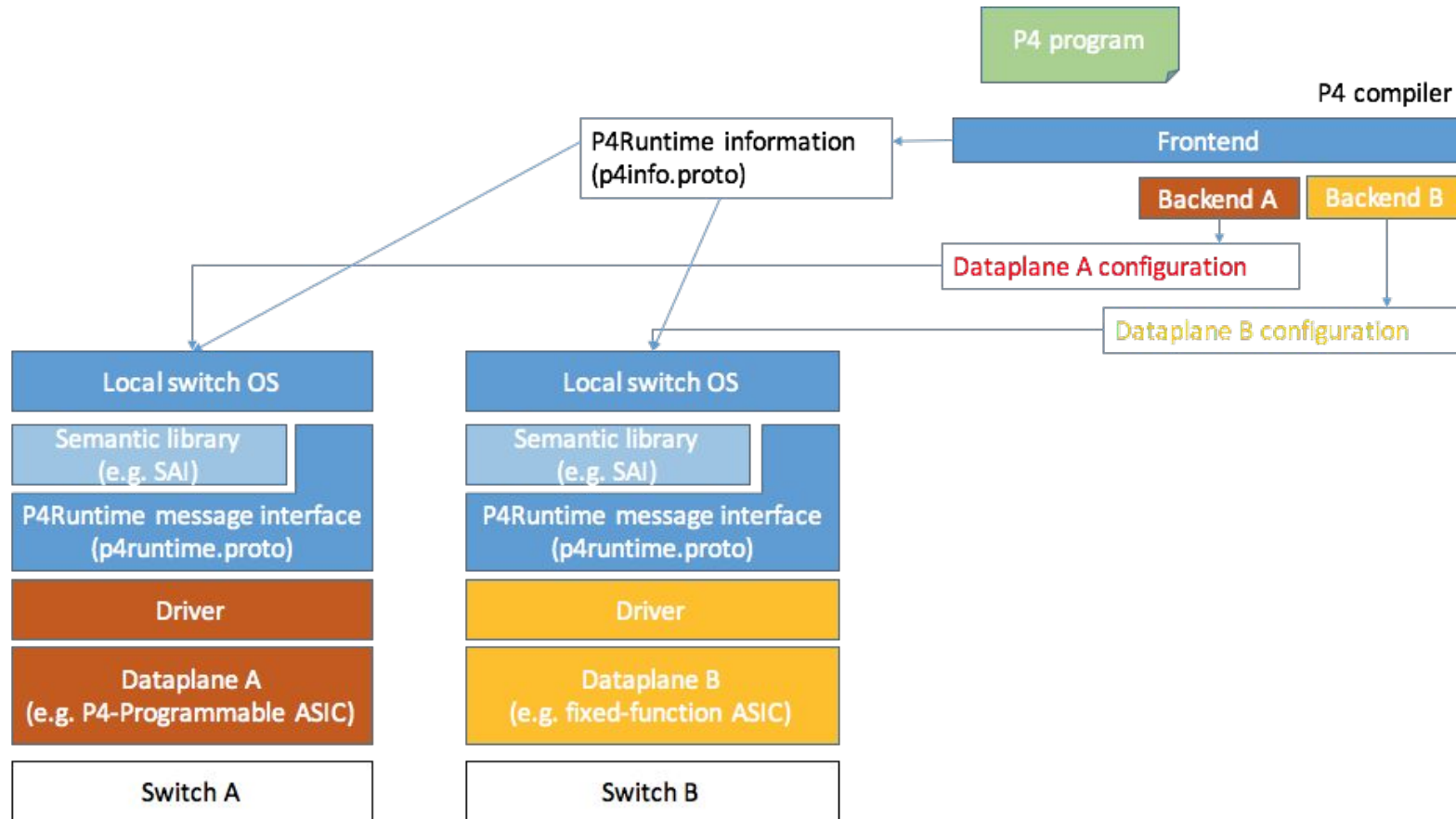
# P4Runtime Primer

- Provides runtime control of P4 entities
- A silicon-independent API which does not require the target device to be fully P4-programmable:
  - P4 as a descriptive language for fixed-function pipelines
  - P4Info as an interface “contract” between control-plane and data-plane
- Can be used for local control (NOS integration) and remote control (SDN) as a generalization of OpenFlow targeted at P4-defined pipelines
- Wire format: Protobuf + gRPC

# P4Runtime with SDN and heterogeneous targets



# P4Runtime as a local API (switch southbound interface)



# P4Runtime scope

- P4 built-in objects (tables, value sets) and PSA externs (counters, meters, action profiles, ...)
- Extensions for vendor-specific externs (non-PSA architectures)
- Basic session management for SDN:
  - Controller replication, with P4 pipeline partition
- Packet I/O
- Batching support, with different atomicity guarantees including “rollback-on-error” and “dataplane-atomic”
- Target reconfiguration with a new P4 pipeline

# Working group deliverables

According to the charter:

- ✓ 1. A RPC interface definition (serialization format and RPC protocol) implementing a P4 runtime API
- ✓ 2. A specification document defining the semantics of P4Runtime
- 3. A specification of the configuration bindings necessary for P4 runtime control expressed as an OpenConfig schema in YANG format
- 4. A working open-source implementation of 1. and 3., including a client and server libraries and bindings for the p4lang software switch.



# P4Runtime Status

- P4Runtime v1.0-RC1 published in June 2018: includes RPC interface definition and specification document defining the semantics of the API
  - <https://p4.org/p4-spec/docs/P4Runtime-v1.0.0.pdf>
- P4Runtime v1.0 to be released in Q3 2018
- P4Runtime v1.0 goals:
  - Designed for P4<sub>16</sub> and PSA v1.0
  - Covers 95% of use-cases the WG was aware of
  - Some features were deferred for simplicity

# Retrospective of 12 months of work

- Started with a draft outline of the protobuf
- Bi-weekly meetings, attended by 10-15 engineers from different member companies
- Opened 33 github issues, closed 16
- Parallel work on protobuf and textual specification
- Set v1.0 inclusion criteria in February
- The hardest design decision:
  - Dealing with ports id representation: how to enable a uniform, target-agnostic port numbering scheme in P4Runtime?
  - Lead to the introduction of the `type` construct in P4\_16: unlike `typedef`, which is just an alias, `type` introduces a completely new type, which can have specific properties such as different data-plane (HW-specific) and control-plane (HW-agnostic) representations.
  - The P4Runtime server is in-charge of translating the control-plane representation to the data-plane one, using a mapping provided through gNMI / OpenConfig

# Next Steps

- Next steps
  - Finalize and release P4Runtime v1.0 (interface definition and specification) by the end of Q3
  - Release a best practice document for users & implementers after getting some experience
  - Work on interaction with OpenConfig (e.g. port translation) and define new YANG model if necessary
  - Work on fixing some of the known-limitations with minor (backwards-compatible) P4Runtime releases
  - Start scoping-out next major (non-backward compatible) release: P4Runtime v2.0
- P4Runtime v1.0 known limitations (listed in specification)
  - No runtime control for hashing units (minor release if PSA support)
  - Indirect match-tables are limited to a no-op default action (minor release if use-case)
  - Support for PSA `Random` and `Timestamp` extern (minor release if use-case)
  - Support for arbitrary expressions in match-keys and action parameters (probably P4Runtime v2.0)

# How to contribute

- Review P4Runtime v1.0-RC1 draft and send comments to one of the public p4.org mailing lists (<http://lists.p4.org/mailman/listinfo>): p4-discuss / p4-dev
  - <https://p4.org/p4-spec/docs/P4Runtime-v1.0.0.pdf>
- Join the P4 API WG if you are a p4.org member by subscribing to the p4-api mailing list (<http://lists.p4.org/mailman/listinfo/p4-api> [lists.p4.org](http://lists.p4.org)) and help us work on future P4Runtime versions!
- Implement P4Runtime v1.0 on your ASIC / NIC / software switch / toaster / ...