Update from the P4 Language Design Working Group

Gordon Brebner

P4.org

P4 Workshop, May 2016

Working Group activity

- **Remit:** Work on P4 language design and specification
  - Focus on real work (technical consensus, open source contributions), and avoid busy work
  - Liaise with P4 Language Advisory Group (of five academic/research lab language experts)
  - Co-chairs: Gordon Brebner and Changhoon Kim

- 6 July 2015 to 2 February 2016: 13 in-person meetings at Stanford
  - Companies: AT&T, Barefoot, Broadcom, Brocade, Cisco, Dell, Huawei, Netronome, PLUMgrid, VMware, Xilinx
  - Universities: MIT, Princeton, Stanford, UMass Lowell
  - Maximum attendance = 24 … at the meeting making P4 v1.1 in-or-out decisions

- Since 22 February 2016: WebEx conference call every two weeks
  - Broaden participation at meetings
  - P4-design mailing list
  - GitHub, including issue tracker: https://github.com/p4lang/p4-spec

“$P4_{14}$” (P4, 2014 to date)

- Birth: the original P4 paper, CCR July 2014
- First public P4 specification published in September 2014
- Current “widely-supported” P4 specification published in March 2015
  - Known as v1.0.2
- Current “pre-release” P4 specification published in January 2016
  - Known as v1.1.0

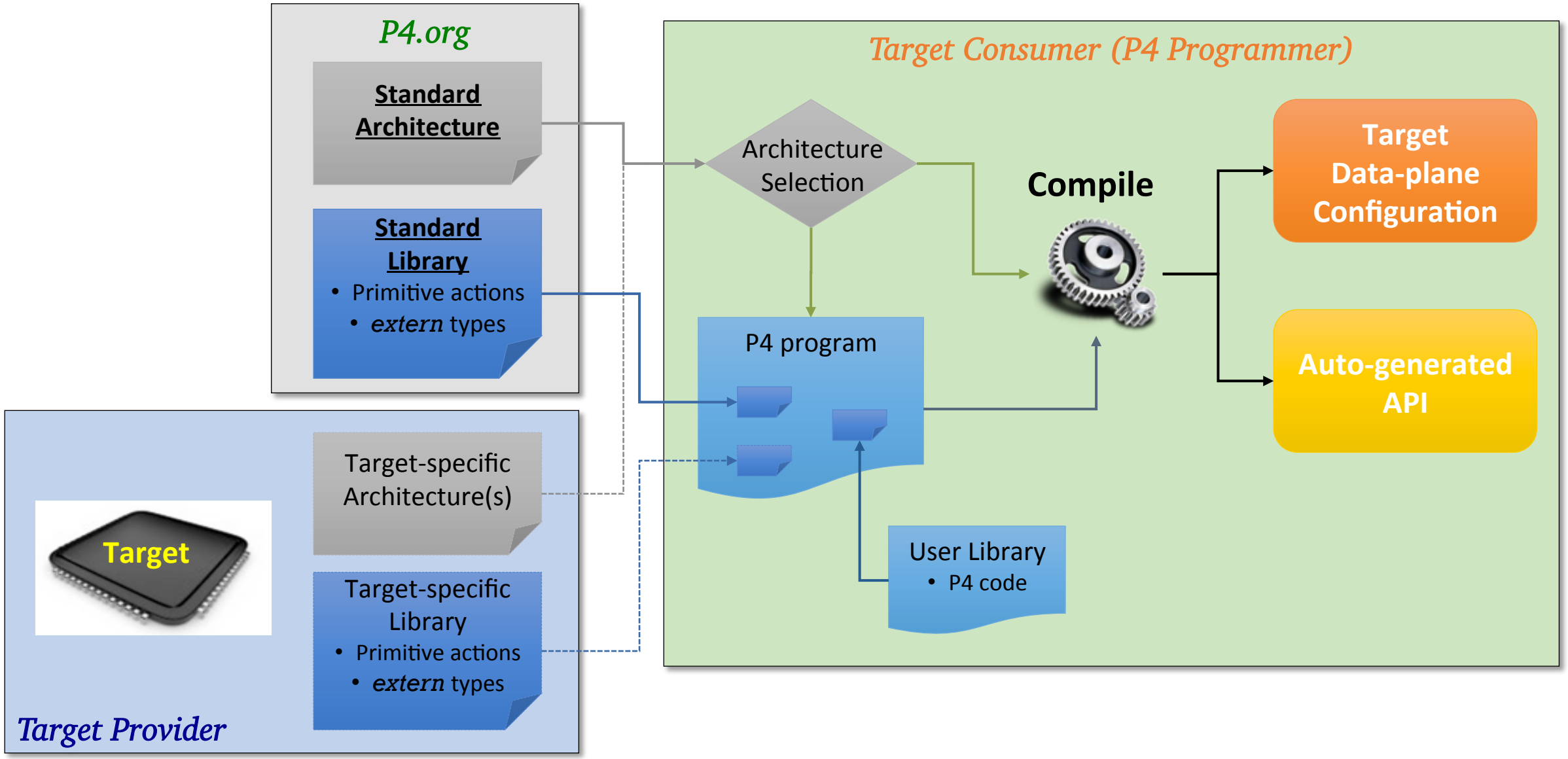
Summary of current “pre-release” spec

- No major changes from the current “widely-supported” spec
  - So translation is easy

- More precise specification of data types used in P4
- Expressions can be used in more contexts (e.g. in modify_field)
- Sequential execution semantics for groups of actions
- Introduction of *extern* types to access functions outside the P4 core

- See Section 17.2.1 of spec for a full list of changes

Goal: language-architecture separation



"$P4_{16}$" (P4 for the future, from 2016 onwards)

- Work in progress within p4-design
  - Prototype implementations coming soon
  - Emerging from careful discussion, review, and consensus in the WG and the LAG

- Main goals:
  - Simplify and clean up P4
  - Enable language-architecture separation
  - Provide a precise specification of the language
  - Provide a reference implementation of the language
  - Address important feedback received on the $P4_{14}$ versions

- Intended to be the stable language base for the future
  - **All further changes to $P4_{16}$ in the future will be backward-compatible**
  - Conversion tools to map $P4_{14}$ to (the working draft of) $P4_{16}$ are already available

$P4_{16}$ retains the strengths of $P4_{14}$

- **Same core abstractions:**
  - Headers
  - Metadata
  - Parsers and deparsers
  - Tables
  - Actions
  - Control flows

- **Declarative language:** specify the “what” rather than the “how”

$P4_{16}$ improvements on $P4_{14}$

- **Simplified core language** (79 keywords => 40 keywords)
  - Many language constructs moved into libraries accessed via the extern mechanism (counters, meters, registers, hashes, packets, checksums, lots of primitive actions, etc.)
  - Core language semantics specified in detail

- **Familiar programming language constructs used where appropriate**
  - Examples: const, struct, enum, typedef, assignment statement, exit and return statements

- **Deparser declarations**

- **Support for many network devices, not just switches**
  - Added language constructs to describe target architectures

Current status

- $P4_{16}$ is the main focus of working group meetings at present
- Draft spec and sample $P4_{16}$ code available in GitHub:
  - https://github.com/p4lang/p4-spec/tree/master/v1.2
- Issues tracker recording active discussion points
- Prototype implementations in progress
- All contributions very welcome - join p4-design!

Additional working groups in formation

- Standard architecture working group
  - Define the standard architecture, and associated standard library
- API generation working group
  - Formally define common run-time API generation framework
  - Program-independent API generation approach deserves more attention
- Logistics
  - Still looking for additional volunteers to lead the WGs
  - Will begin official activities soon (next few months)
    - May initially use some p4-design meetings to kick-start these other WGs
    - … given overlap of people's intrests between p4-design and these other WGs