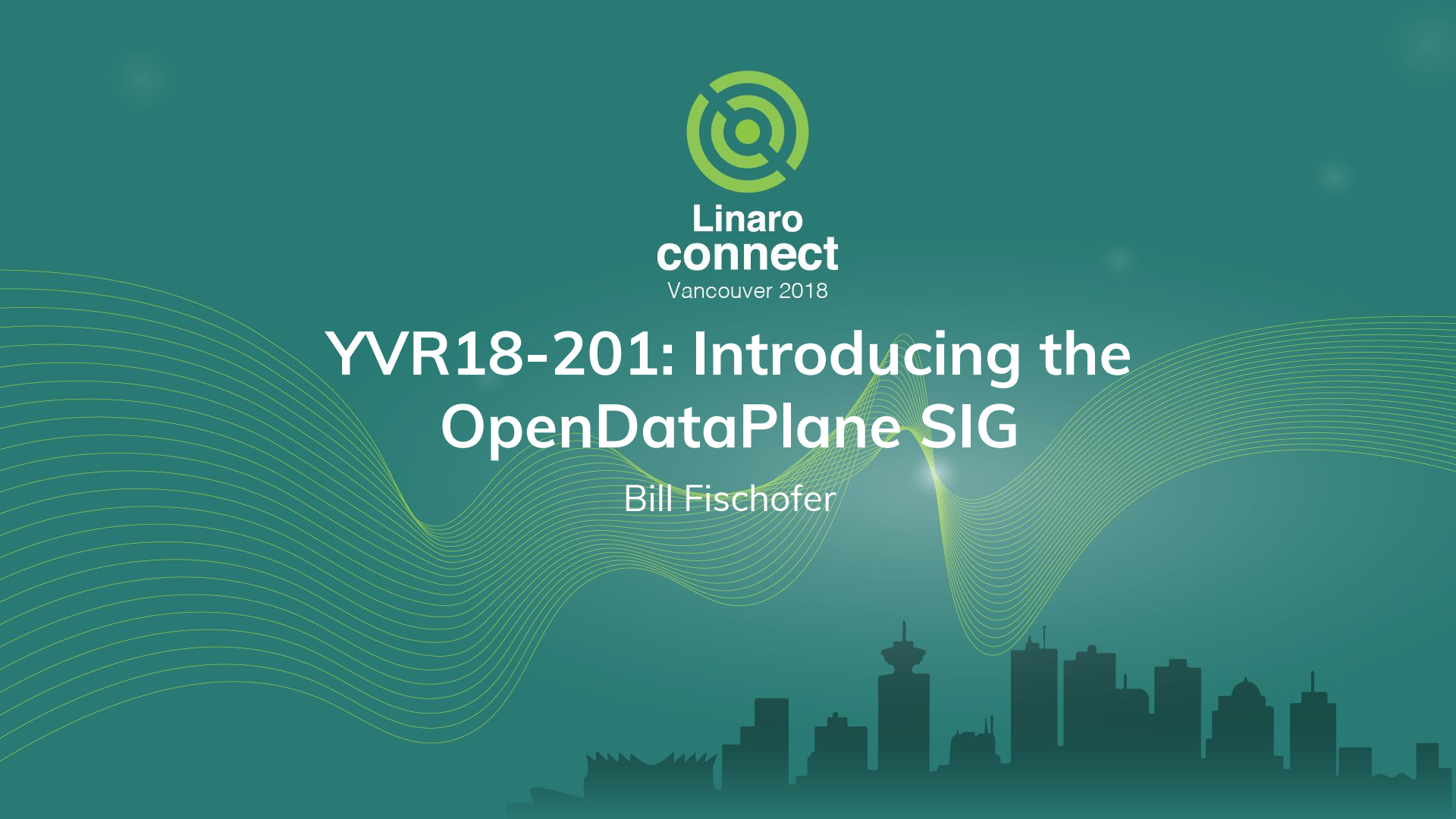


YVR18-201: Introducing the OpenDataPlane SIG

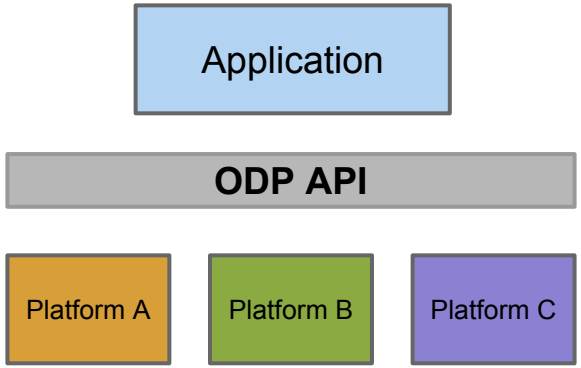
Bill Fischofer

Topics for Today

- Brief background on ODP motivation/rationale
- Where we've come from
- SIG Structure and Advantages
- Where we're going
- Q&A

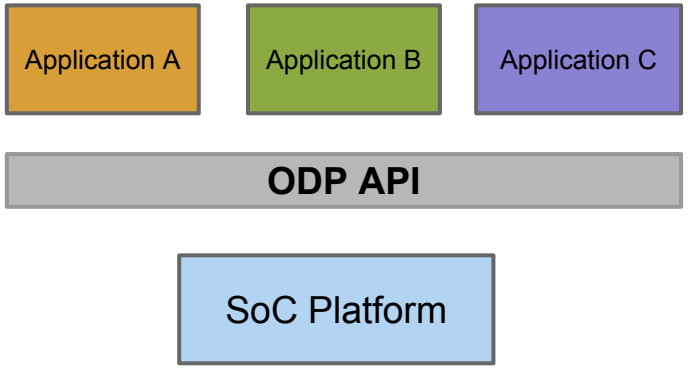
Why OpenDataPlane?

Application Goals



- Write once, accelerate anywhere
- Take full advantage of available HW acceleration w/o effort
- Focus on application, not platform

Platform Vendor Goals



- Compete for any Socket
- Fully exploit unique IP without requiring application changes

OpenDataPlane - A Brief History

2013

- Linaro Networking Group founded, ODP project launched

2014

- First preview release(s)

2015

- OpenDataPlane v1.0 released for evaluation

2016

- First Long Term Support (LTS) release - ODP “Monarch”

2017

- Development releases

2018

- Second LTS release - ODP “Tiger Moth”, ODP SIG Launched - ODP 5th Anniversary

ODP Application Areas Today and Tomorrow

OpenDataPlane is finding traction in many different markets - for example:

- Telecommunications
  - EPC - Evolved Packet Core
  - Cellular Base Stations - pico to macro
  - Cellular Base Station Backhaul - with large encrypted “FAT” pipes
  - vCPE and uCPE - Customer Premise Equipment
  - SD-WAN - Software Defined Wide Area Networks
  - ETSI NFV based solutions - Application does not change with or without acceleration
- Enterprise and Cloud
  - Advanced Firewall Appliances and vFirewall in the Cloud
  - Security and VPN Gateways
    - vIPS/IPS (Intrusion Prevention Systems)
    - Malware detection
  - Routers, Application Load Balancers - any packet moving devices
- Smart NICs
- Edge devices

Platforms of Interest

General Server Platforms

- SW-centric: Cores must touch every packet
- Standardized acceleration (Special CPU instructions, *e.g.*, AES, and/or plug-in cards with PCIe latencies)
- Standardized I/O (mainly PCIe)
- Standard form-factors (mainly rack chassis)
- Leading architectures: x86, Arm (AArch64)
- Mainly cloud environment

Network SoC-based Platforms

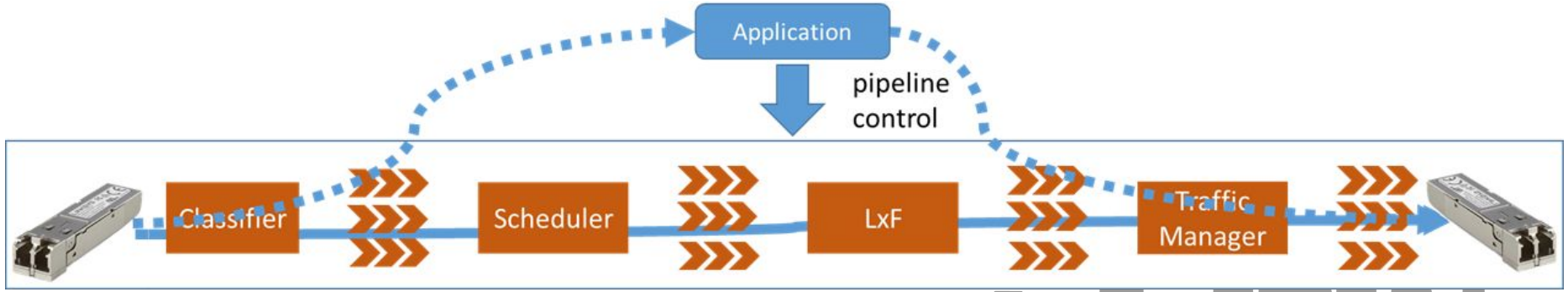
- HW-centric: Packets may bypass cores partially or completely
- Innovative and non-standard approaches to acceleration (value-add IP)
- Varying I/O (*e.g.*, integrated)
- Varying form factors (*e.g.*, Smart NIC, embedded devices)
- Leading architectures: Arm (AArch64), custom
- Mainly edge/embedded environments

Software Implemented vs. Software Defined

Software Implemented: DPDK, Netmap, PacketDirect...

An IPsec application has to adapt to the various types of accelerators: none, crypto, IPsec crypto device, IPsec lookaside, IPsec inline

Software Defined: OpenDataPlane



An IPsec application just uses the IPsec inline API that itself adapts to the underlying accelerators

ODP as an “Innovation Boundary”

For Applications

- Define functional needs without concern for how they are met
- No requirement to be expert in platform architecture or hardware internals
- Permit portability across all conforming implementations with at most a recompile
- Defines an ABI that permits application binary portability within an Instruction Set Architecture (ISA).
- Avoid vendor lock-in without “least common denominator” limited functionality.

For SoC Vendors

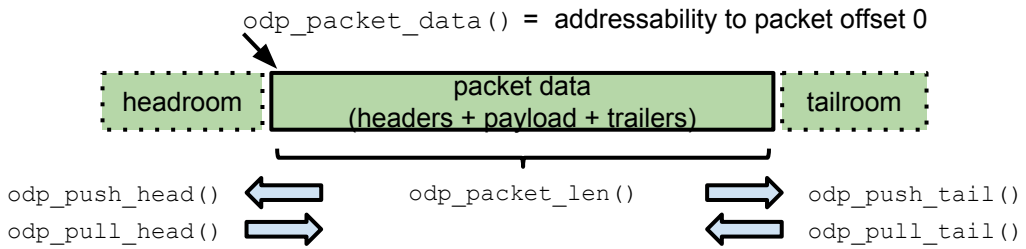
- Implementation internals not exposed across API boundary
- APIs may be realized in HW, SW, or any combination without impacting applications
- Vendors not constrained by a fixed SW-centric implementation model
- Permits unique platform value to be exposed without requiring any application changes
- Universal SDK that avoids internal fragmentation

Advantages of ODP Abstract APIs

Applications focus on the ***what***

Applications deal with functional needs, not implementation details.

- E.g., packet operations performed on **odp_packet_t** objects:

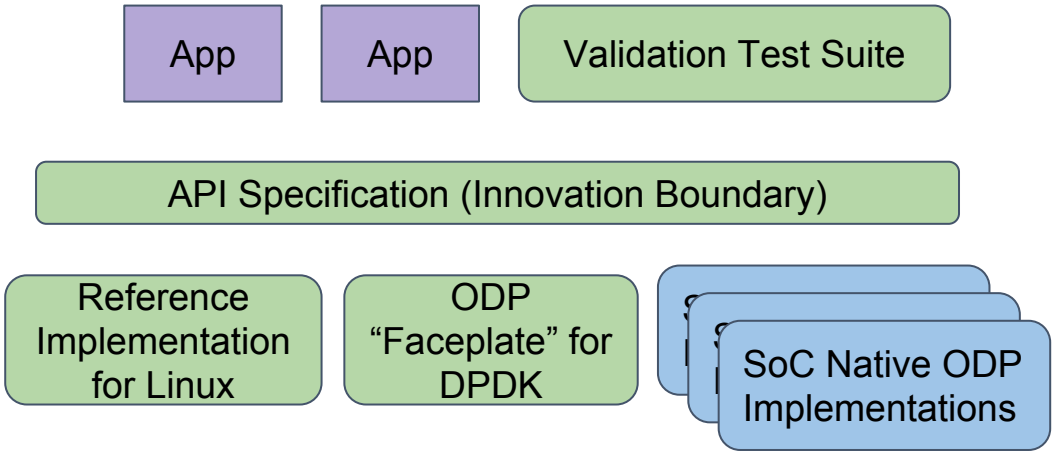


Implementations focus on the ***how***

Implementations not constrained by SW implementation assumptions

- Easily support HW buffer/packet managers, queues, schedulers, etc.
- Key to differentiated value without requiring application changes

ODP Structure and Scope



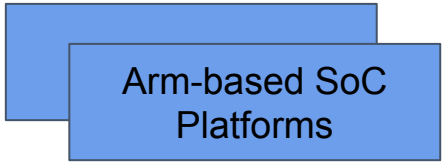
- Third Party App Work Product
- Vendor Work Product
- ODP SIG Work Product

- API Specification defines the ODP "innovation boundary"
- Reference Implementations are pure-SW implementations of ODP, and a starting point for SoC native implementations. Platform-neutral and can run anywhere that has a Linux kernel and/or DPDK.
- Validation Test Suite ensures that all implementations conform to the ODP API specification
- SoC Native Implementations offer best-in-class performance with full HW offload/acceleration

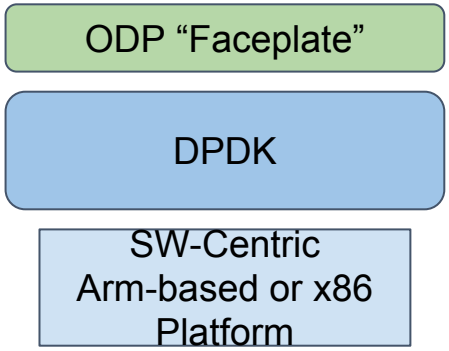
ODP Cross-Platform Support

Write Once - Accelerate Anywhere

ODP API



Optimal Performance



Good Performance

Direct mapping of ODP APIs onto DPDK

Binary compatibility across platforms sharing same Instruction Set Architecture (ISA). Recompile to switch to different ISA.

Why an ODP SIG?

Recognize Project Maturity

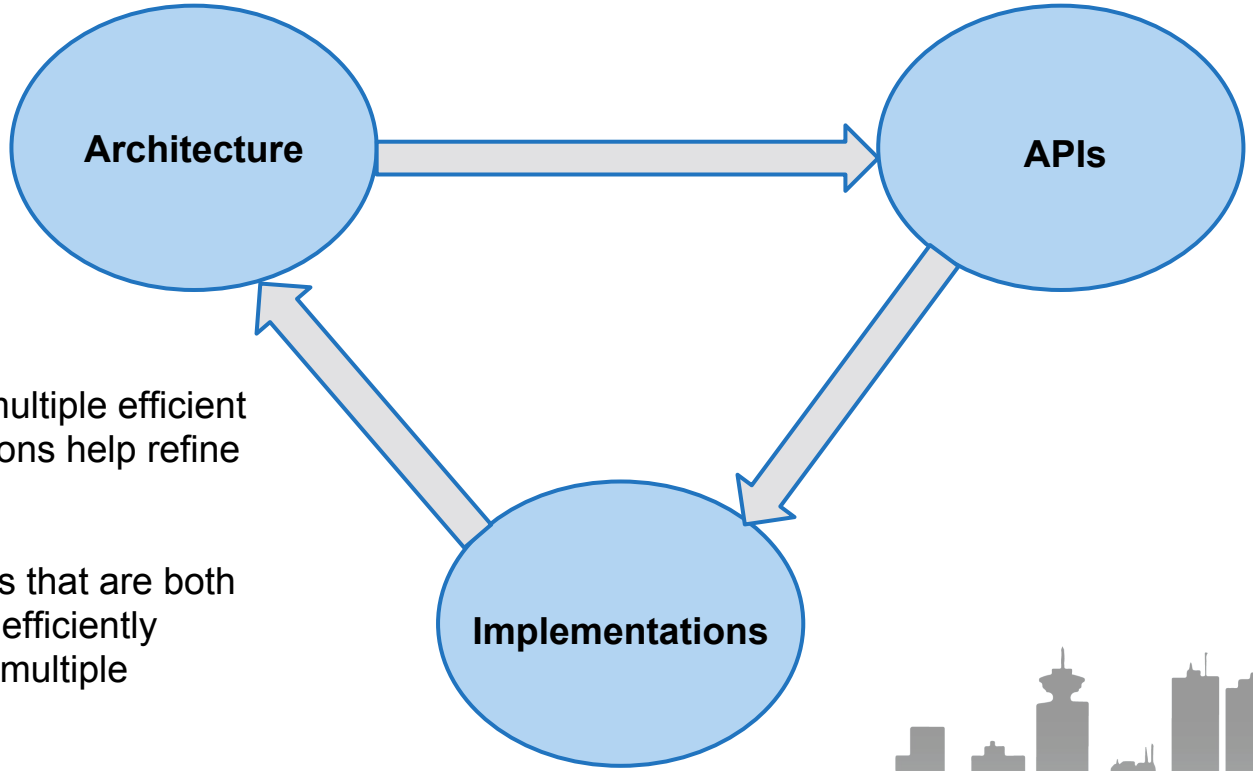
Autonomy and Control by Independent Steering Committee

More Flexible Membership Options

- Cost
- Resource commitments
- Smaller companies
- Universities

Enable more Responsive and Focused attention to Member and community needs

ODP Development Workflow



Iteration as multiple efficient implementations help refine architecture

Result is APIs that are both portable and efficiently mappable to multiple platforms

Where we're going...

New APIs Under Development

- Compression
- 3GPP Crypto Algorithms
- Lightweight Flows
  - See Session YVR18-217 this afternoon
- Tunneling

Scalability and Performance

- ODP now tested on large core-count systems (>200)
- Continued streamlining of SW scheduler--now showing performance within 10% of poll mode operation.

Investigations (Joint with LTN SIG)

- DPDK bug fixes and tuning for performance/scaling on Arm
- High-efficiency East-West messaging for virtualized/containerized apps

Q&A

Questions?


Linaro
connect
Vancouver 2018
Thank you!