OPEN
Compute Summit
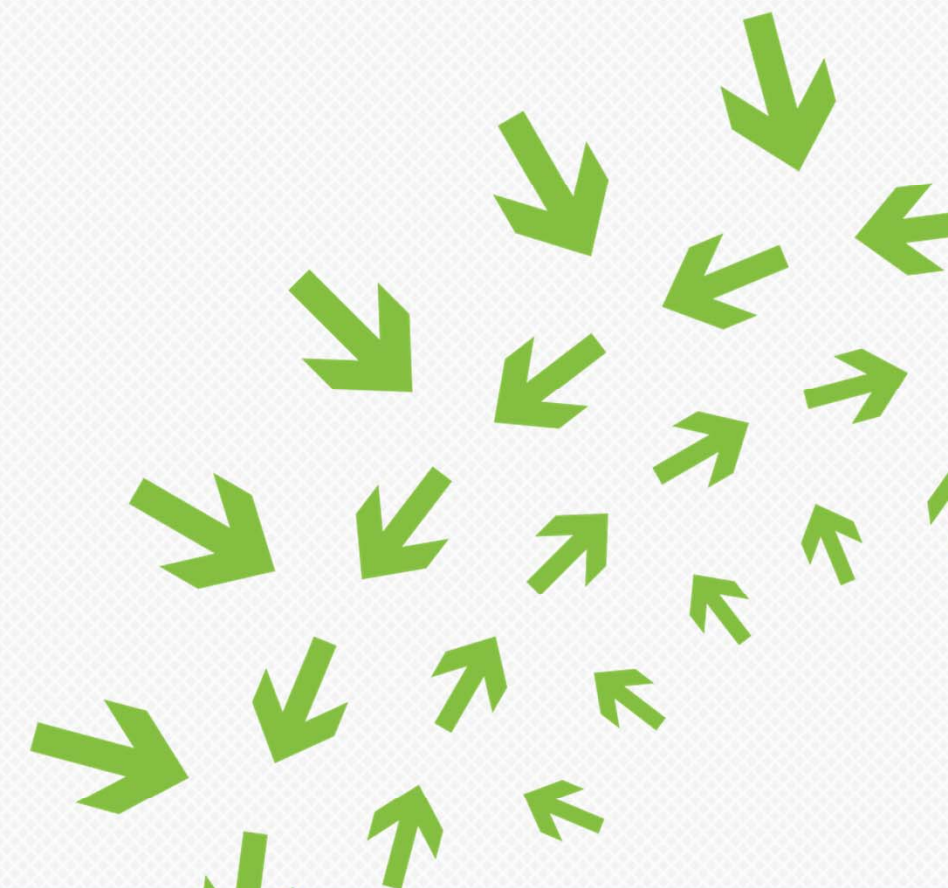### January 28–29, 2014San Jose

Open Network Switch Library
OpenNSL for OCP Open Network Switches

Sujal Das, Simon Knee

Broadcom Corporation

OpenNSL Topics
Meeting industry use cases and requirements

- Use cases – end users and OEMs

- API and software package requirements

- OpenNSL concept
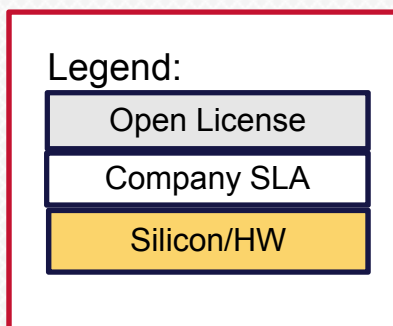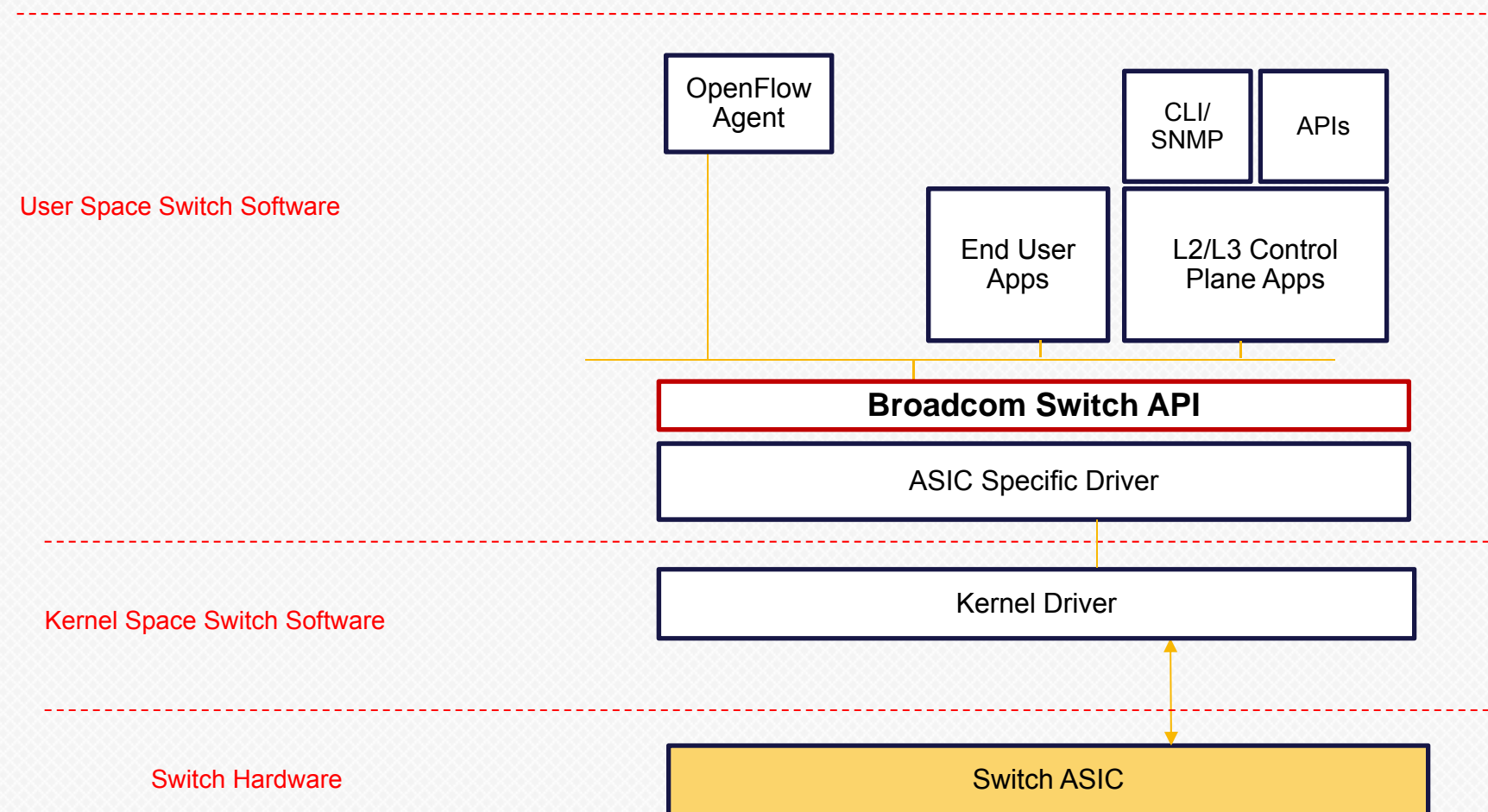
- Sample APIs and package

Use Cases

User survey and feedback driving requirements

- Enable open source innovation on switch ASICs

  - Through OCP, GitHub

- Open Switch API must not break compatibility

  - Broadcom switch SDK and API used widely

  - OEM, OSV/ISV, Operator network operating systems

- Enable operator innovation on OEM/ODM switches

  - Access to open switch APIs for operator-developed apps

  - Operates side-by-side with OEM, OSV/ISV control plane

The Broadcom Switch API Landscape
Many OEM, OSV & End User Apps on Switch API
Apps above Switch API are closed source

User Space Switch Software

OpenFlow Agent

CLI/ SNMP

APIs

End User Apps

L2/L3 Control Plane Apps

Broadcom Switch API

ASIC Specific Driver

Kernel Space Switch Software

Kernel Driver

Switch Hardware

Switch ASIC

Legend:

| Open License |
| Company SLA |
| Silicon/HW |

The OpenNSL API Landscape

Enable Open Source Apps on Switch API Co-exist with Proprietary Apps

User Space Switch Software

| OF 1.3+ Agent | OF 1.3+ Agent | | CLI/ SNMP | APIs | CLI | APIs |

| OpenFlow 1.3+ Abstraction | End User Apps | End User Apps | L2/L3 Control Plane Apps | L2/L3 Control Plane Apps |

One-to-one mapping to Broadcom SDK API to maintain application compatibility

OpenNSL - Open Switch API

ASIC Specific Driver

Kernel Space Switch Software

Kernel Driver

Switch Hardware

Switch ASIC

Legend:
- Open License
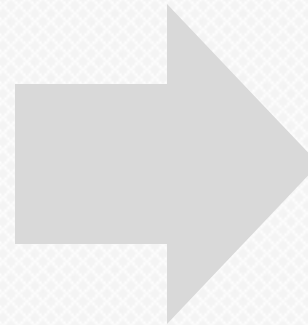- Company SLA
- Silicon/HW

Sample 1-1 Mapping
Rx/Tx & Event Reporting APIs

Broadcom SDK APIs

```
int bcm_rx_start(…)
    Initialize the Rx subsystem (start a Rx thread)
int bcm_rx_register(…)
    Register an upper Rx handler
int bcm_tx_init(…)
    Initialize the Tx (start a Tx thread)
int bcm_tx(…)
    Send a packet
int bcm_linkscan_register(…)
    Register an upper linkscan handler
    …
```
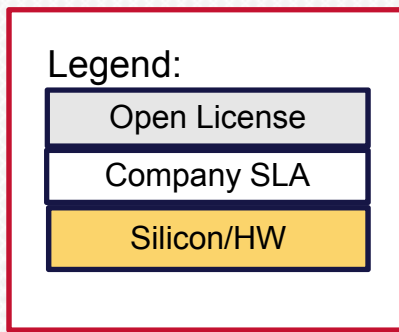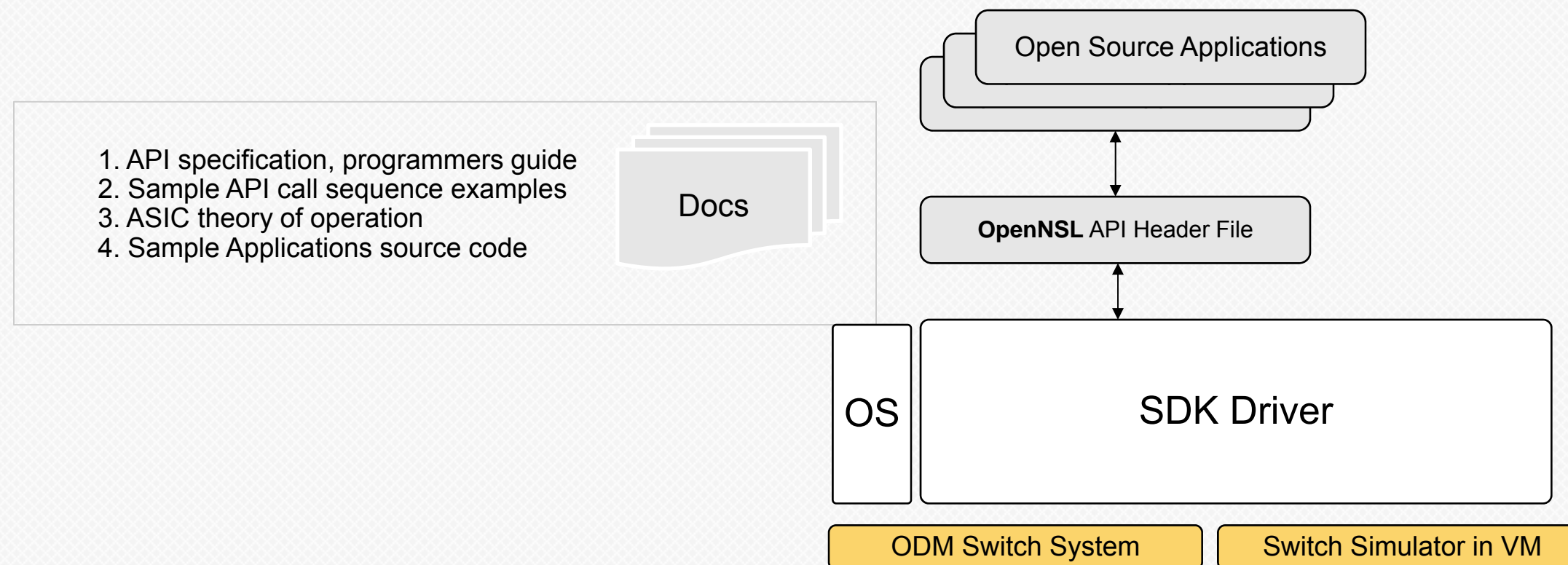
OpenNSL APIs

```
int opennsl_rx_start(…)
    Initialize the Rx subsystem (start a Rx thread)
int opennsl_rx_register(…)
    Register an upper Rx handler
int opennsl_tx_init(…)
    Initialize the Tx (start a Tx thread)
int opennsl_tx(…)
    Send a packet
int opennsl_linkscan_register(…)
    Register an upper linkscan handler
    …
```

Open Source App Use Case
Enable operators, academia, researchers, start-ups

1. API specification, programmers guide
2. Sample API call sequence examples
3. ASIC theory of operation
4. Sample Applications source code

Docs

Open Source Applications

OpenNSL API Header File

OS | SDK Driver

ODM Switch System | Switch Simulator in VM

Legend:
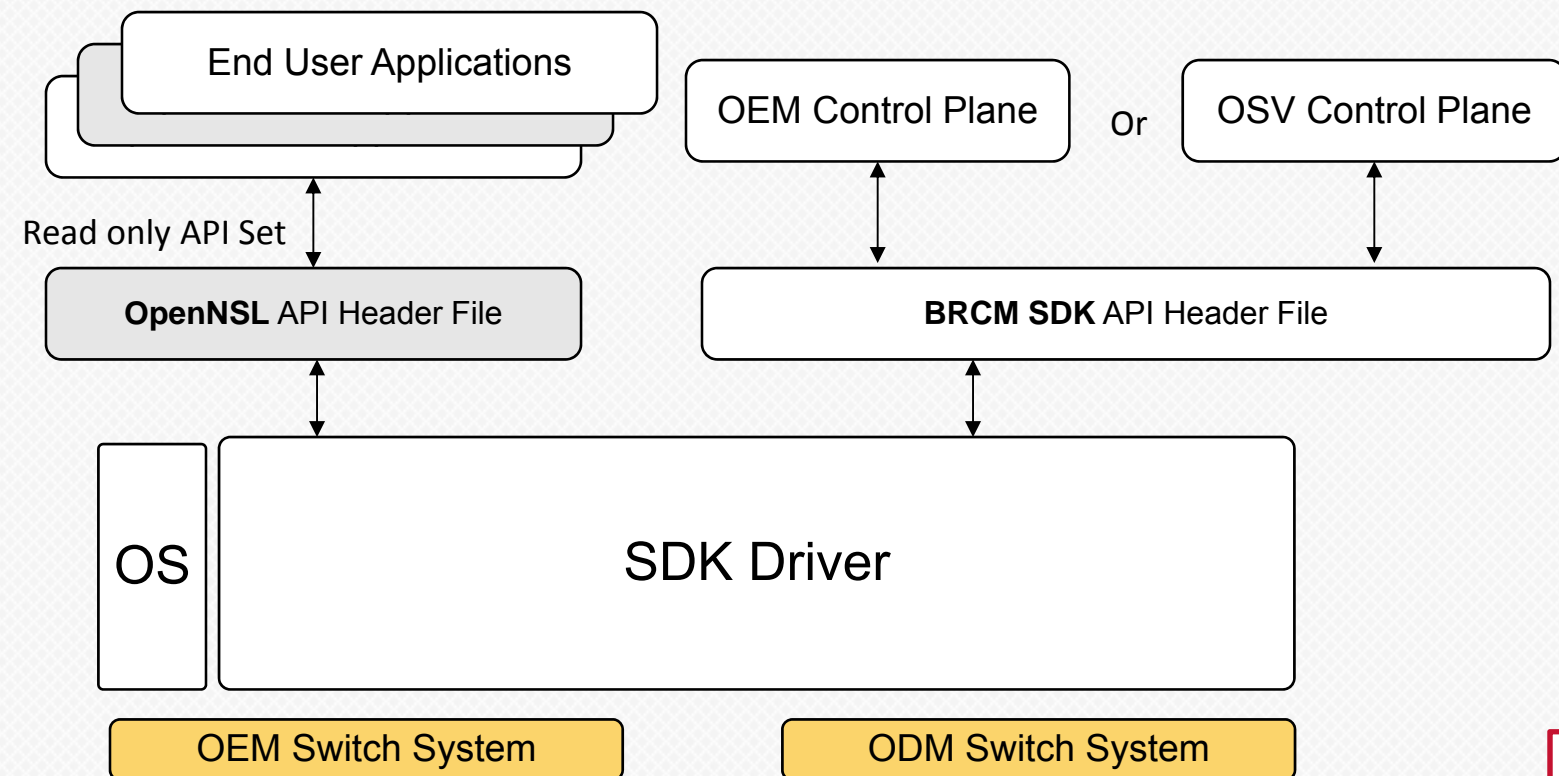Open License
Company SLA
Silicon/HW

OEM Use Case
Enable OEMs to expose SDK APIs to their end users

1. API specification, programmers guide
2. Sample API call sequence examples
3. ASIC theory of operation
4. Sample Applications source code

Docs

End User Applications
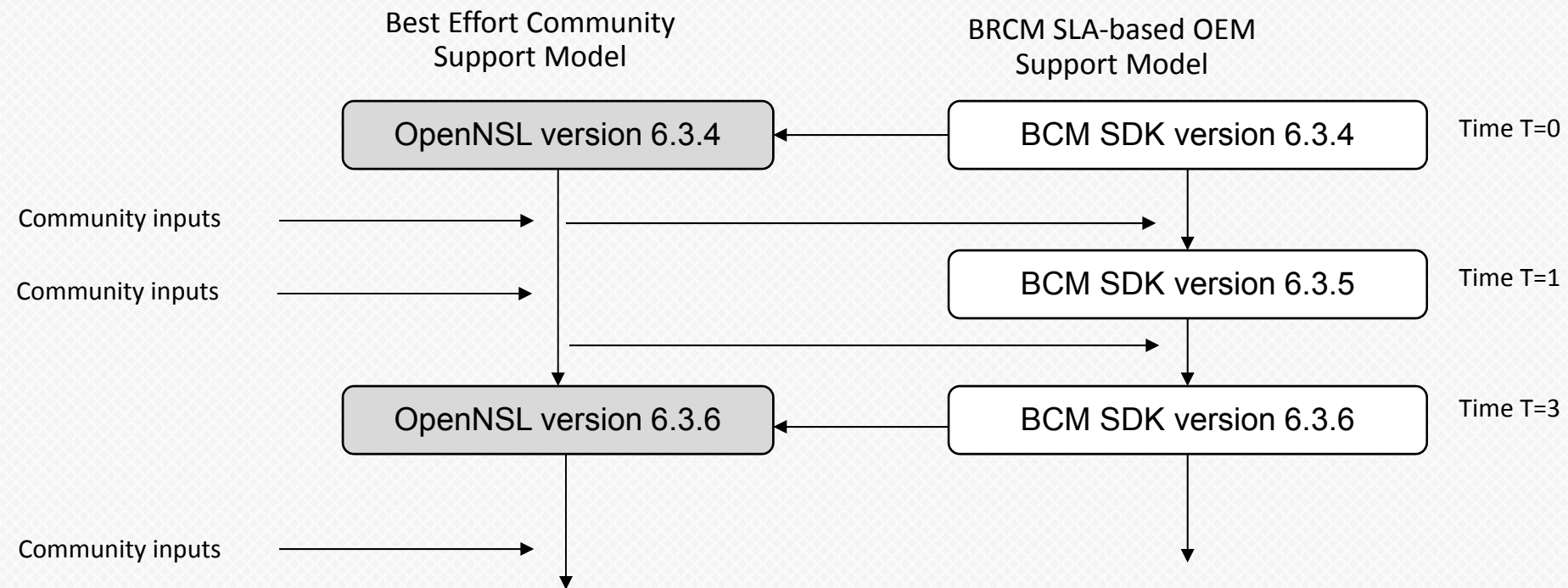
OEM Control Plane   Or   OSV Control Plane

Read only API Set

OpenNSL API Header File

BRCM SDK API Header File

OS   SDK Driver

OEM Switch System

ODM Switch System

Legend:
| Open License |
| Company SLA |
| Silicon/HW |

Keeping Versions in Sync
To Support the OEM Use Case

Best Effort Community
Support Model

BRCM SLA-based OEM
Support Model

| OpenNSL version 6.3.4 | ← | BCM SDK version 6.3.4 | Time T=0 |

Community inputs →

Community inputs →

| BCM SDK version 6.3.5 | Time T=1 |

| OpenNSL version 6.3.6 | ← | BCM SDK version 6.3.6 | Time T=3 |

Community inputs →

OpenNSL releases are tied to Broadcom SDK releases.
    Same versioning nomenclature to be used to avoid confusion
    Not all SDK releases might have a corresponding OpenNSL release.

OpenNSL API Groups Being Considered
Subset releases with proper documentation

- Error Codes
- Initialization
- Port Configuration
- Link Monitoring and Notification
- VLAN Management
- Spanning Tree Groups
- Switch Control
- Class of Service Queue Configuration
- Rate Limiting
- Layer 2 Address Management
- Layer 3 Management

- Packet, Transmit and Receive APIs
- Statistics
- Kernel Network (KNET) Configuration
- Warm Boot
- Diagnostic Shell
- Bidirectional Forwarding Detection
- Mirroring
- Link Aggregation
- Field Processor
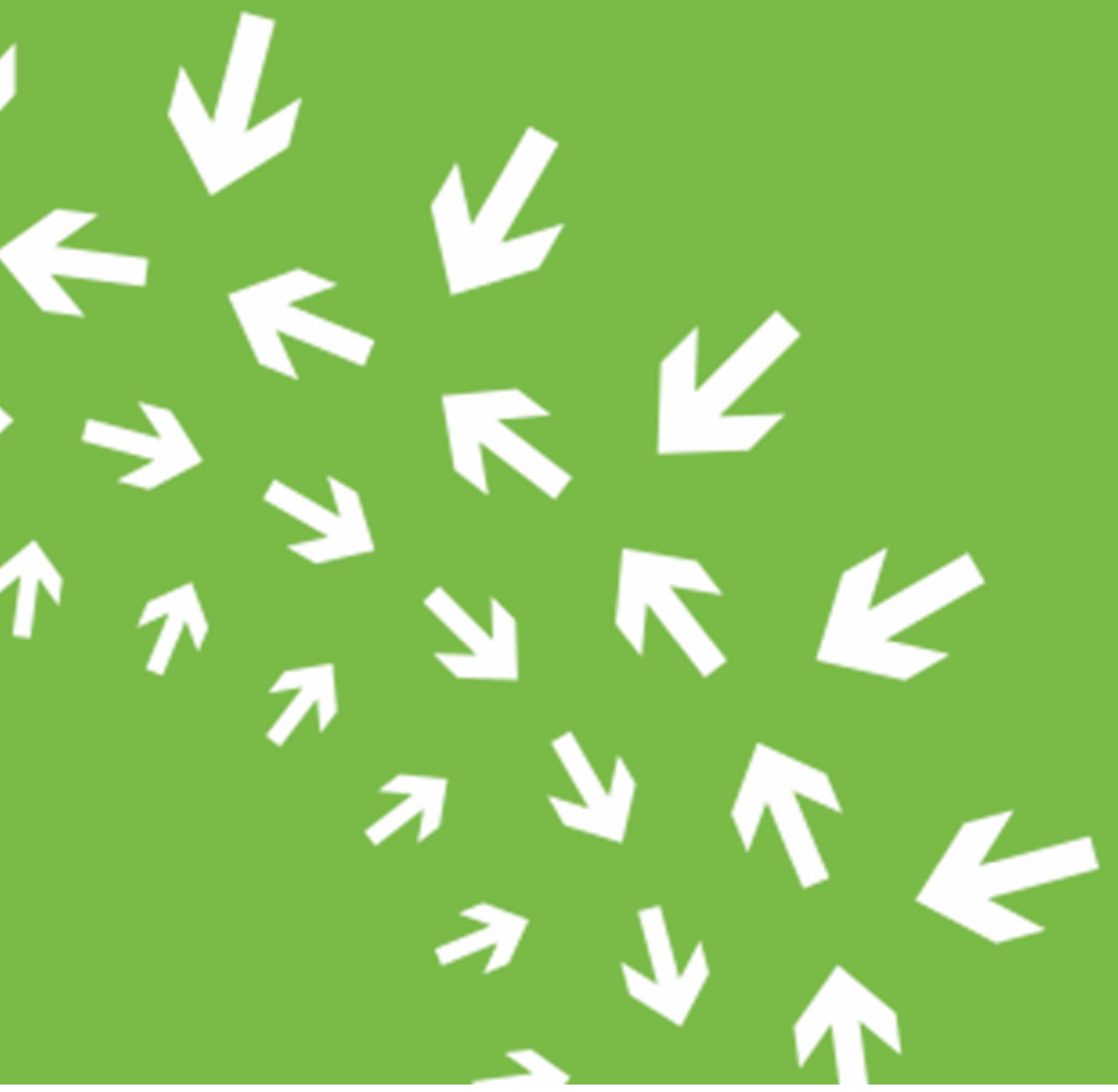- MPLS Management
- VXLAN Management

Next Steps

Stay tuned for OpenNSL release plan

- Releasing the API is easy
  - Proven, comprehensive set available now
  - Just need to map a subset 1-1
- Documentation will take a bit longer
  - Must be suitable for the general open community
- Ensure all use cases are met satisfactorily
  - Let us know if we missed any

Thank You!