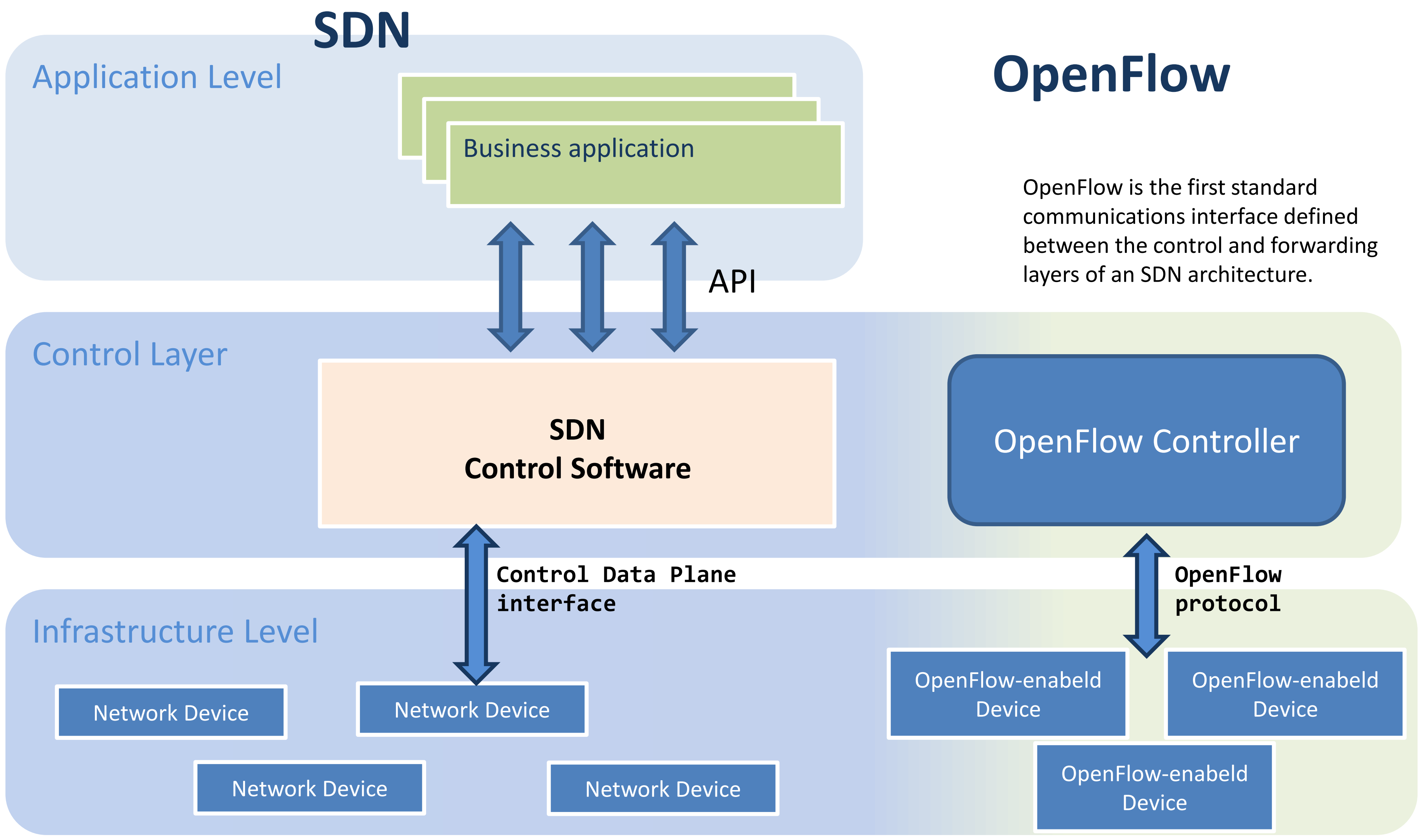
OpenFlow - the key standard of Software-Defined Networks

Dmitry Orekhov, Epam Systems

Software-defined network

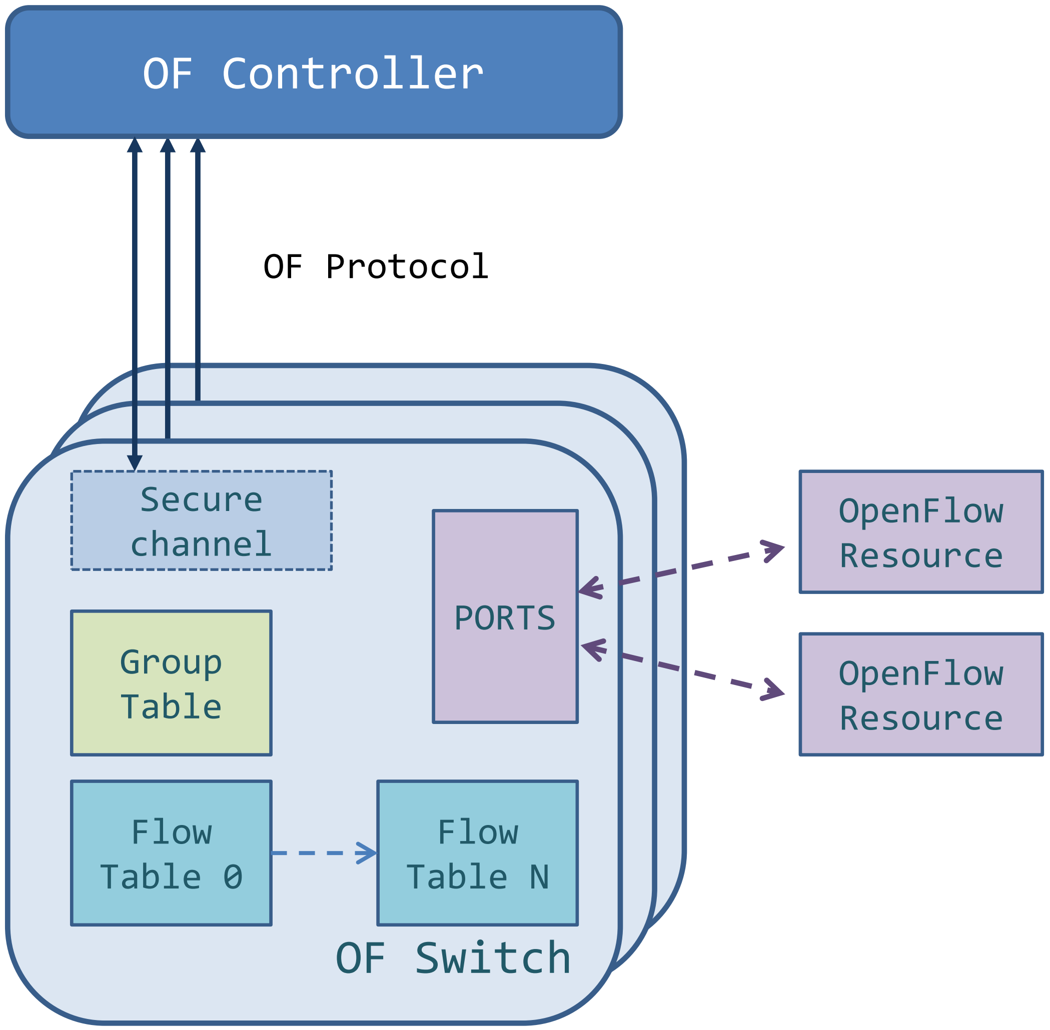
The Need for a New Network Architecture	Limitations of Current Networking Technologies
• Changing traffic patterns • The rise of cloud services • “Big data” means more bandwidth • The “consumerization of IT”	• Complexity that leads to stasis • Inconsistent policies • Inability to scale • Vendor dependenc
The key idea of SDN	
Network control is decoupled from forwarding and is directly programmable.	

OpenFlow and Software-Defined Network

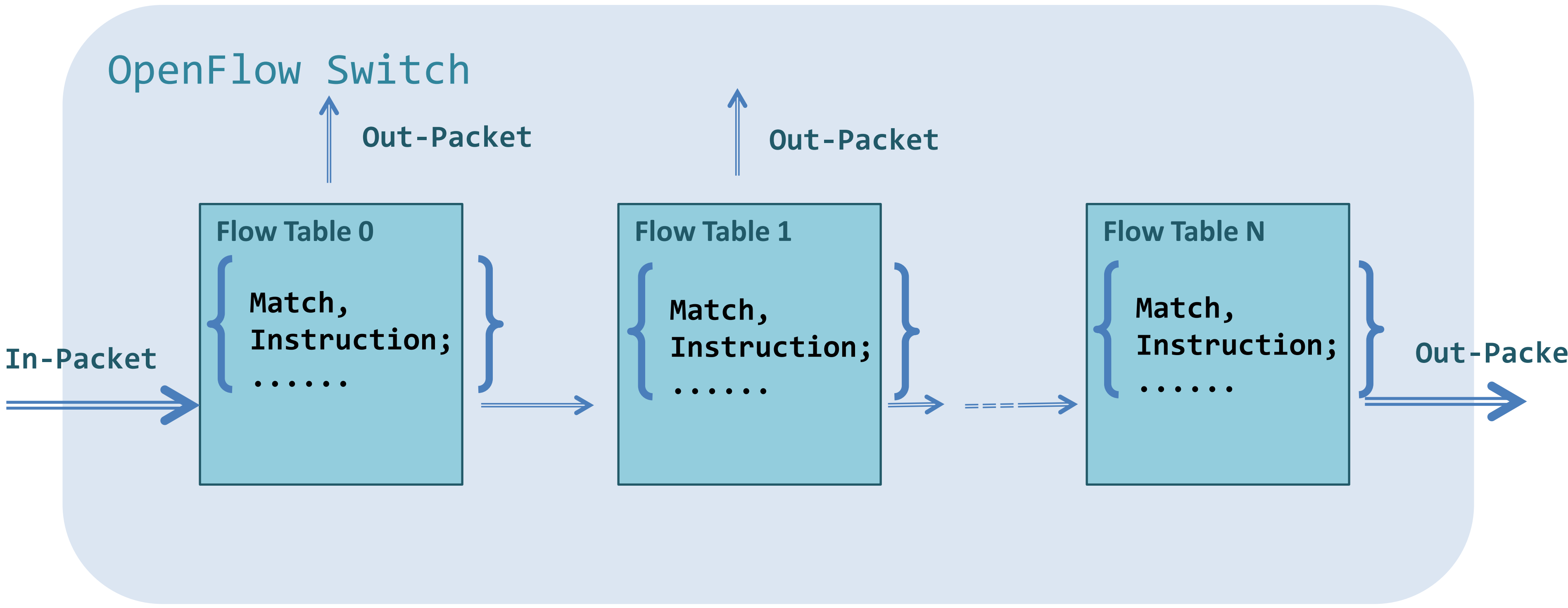


OpenFlow is the first standard communications interface defined between the control and forwarding layers of an SDN architecture.

OpenFlow switch and Controller



Packet forwarding inside OpenFlow switch



OpenFlow Switch
Out-Packet
Out-Packet
Flow Table 0
{ Match, Instruction; ...... }
Flow Table 1
{ Match, Instruction; ...... }
Flow Table N
{ Match, Instruction; ...... }
In-Packet
Out-Packe

- Packet may transferred to other table
- Packet header may be modified
- Packet may be forwarded to given port or just dropped
- Packet may be applied to given QoS

OpenFlow Switch: key elements

OpenFlow tables

Pipeline

Ports

OpenFlow Channel

Flow table entry: key elements

Match Fields	Priority	Counters	Timeout	Cookies	**Instruction set**

Match criteria:

- Ingress-port
- Ethernet MAC
- ARP
- IPv4 and IPv6
- TCP ports
- VLAN, MPLS etc.

Instruction:

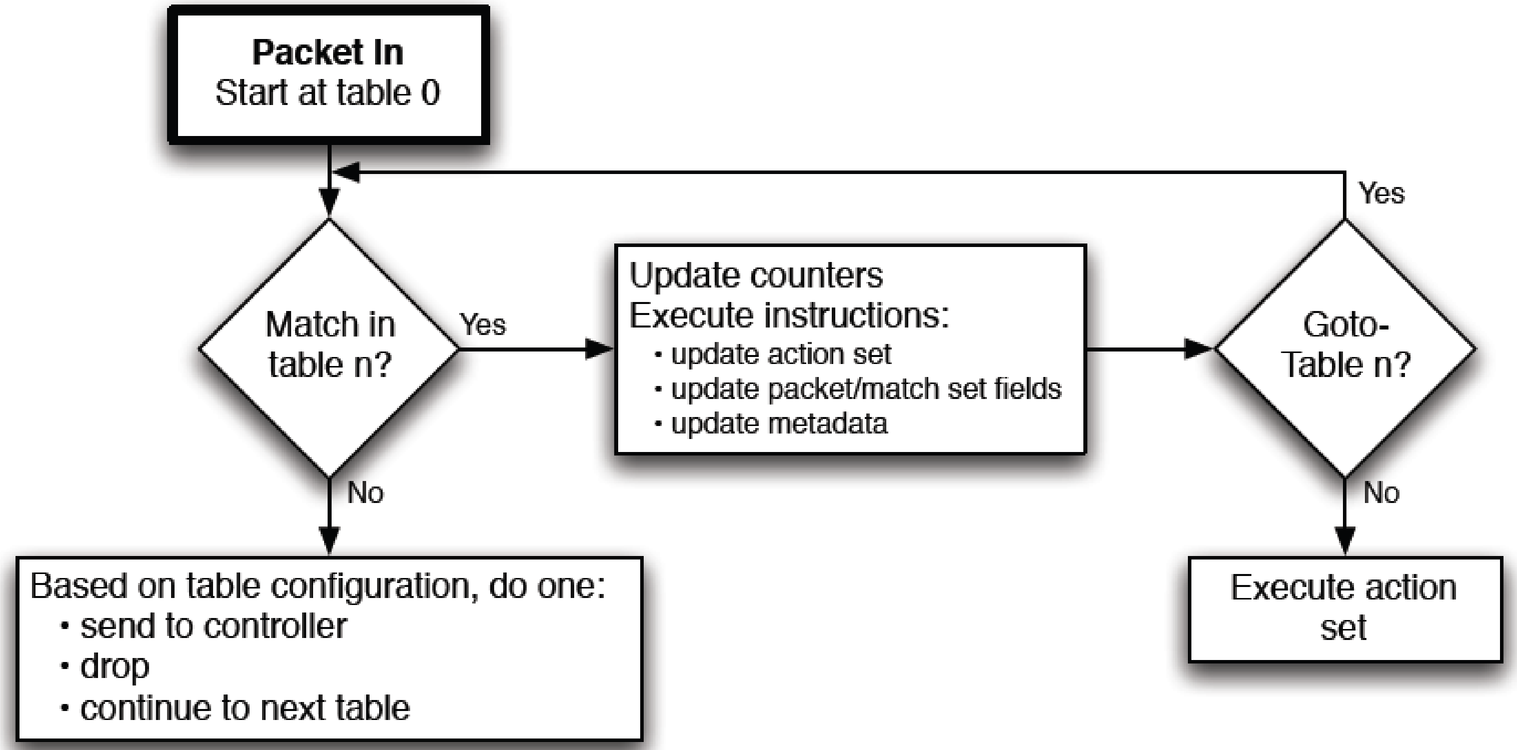
- Go-To Table
- Modify Metadata
- Action Set {forward, apply QoS, drop, Apply to Group}

OpenFlow examples

	Switch port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Prot	TCP sport	TCP dport	Action
Switching	*	*	00:1f :..	*	*	*	*	*	*	Port6
Flow switching	Port3	00:20..	00:1f ..	0800	Vlan1	1.2.3.4	5.6.7.8	4	17264	Port6
Firewall	*	*	*	*	*	*	*	*	22	Drop
Routing	*	*	*	*	*	*	5.6.7.8	*	*	Port6
VLAN switching	*	*	00:1f ..	*	Vlan1	*	*	*	*	Port6, port7, port8

OpenFlow can be compared to the instruction set of a CPU. It specifies basic primitives that can be used by an external software application to program the forwarding plane of network devices, just like the instruction set of a CPU would program a computer system.

Matching


Packet In
Start at table 0
Match in table n?
Yes
No
Update counters
Execute instructions:
• update action set
• update packet/match set fields
• update metadata
Goto-Table n?
Yes
No
Based on table configuration, do one:
• send to controller
• drop
• continue to next table
Execute action set

OpenFlow Protocol: key messages

- **Handshake**
- **Configuration**
- **Modify**
- **Statistics**
- **Error**

- **Asynchronous messages: Packet-In**

- **Symmetric messages: Echo Request-Responce**

OF Controller – Switch: Feedback

- **Packet-In – Packet-Out: Controller learns Switch based on information about incoming packets sent by Switch**
- **Error messages: Switch sends to controller messages about malformed or inappropriate packets.**

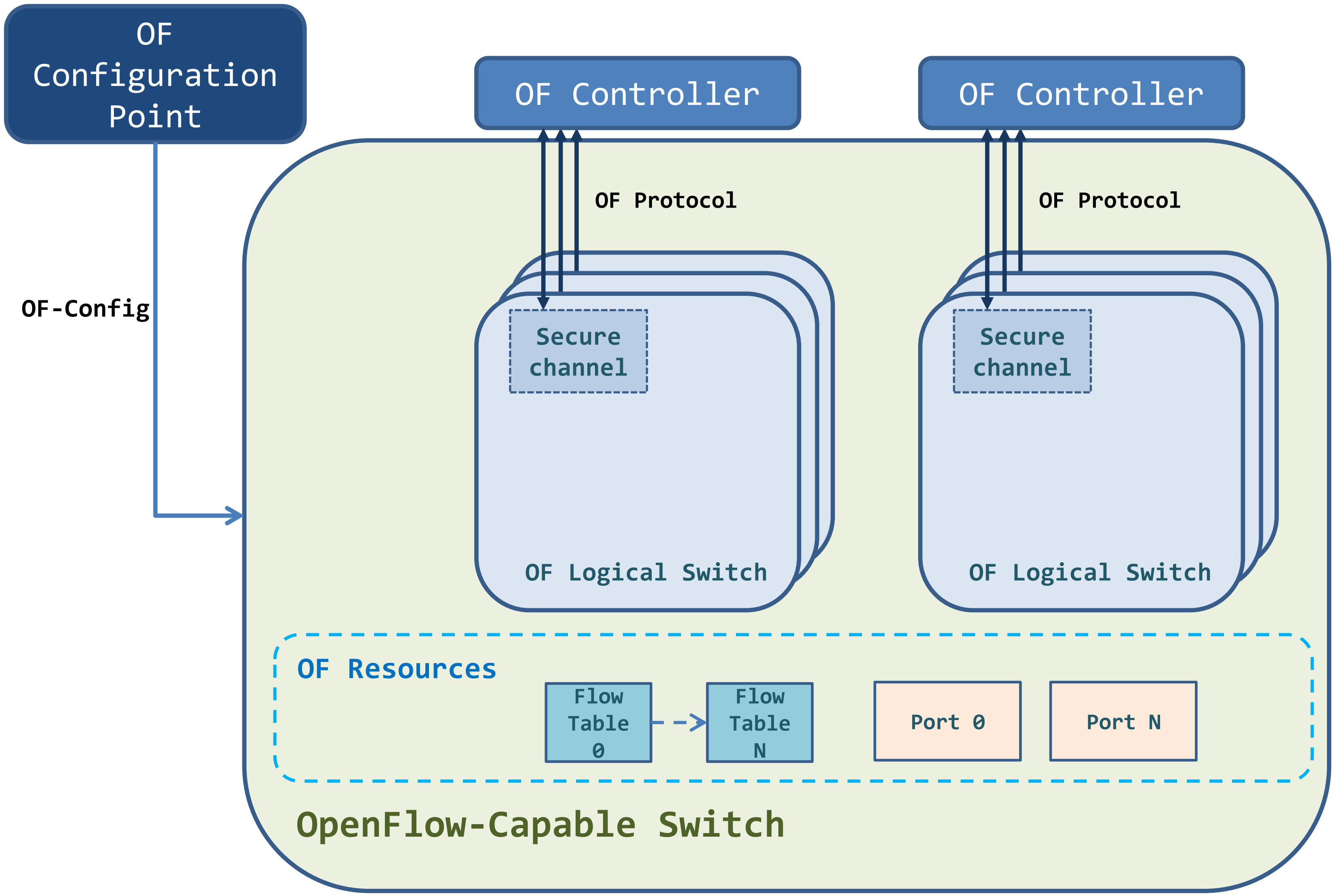
Group Table: “Aspects” of OpenFlow

Group Identifier	Group Type	Counters	**Action bucket**

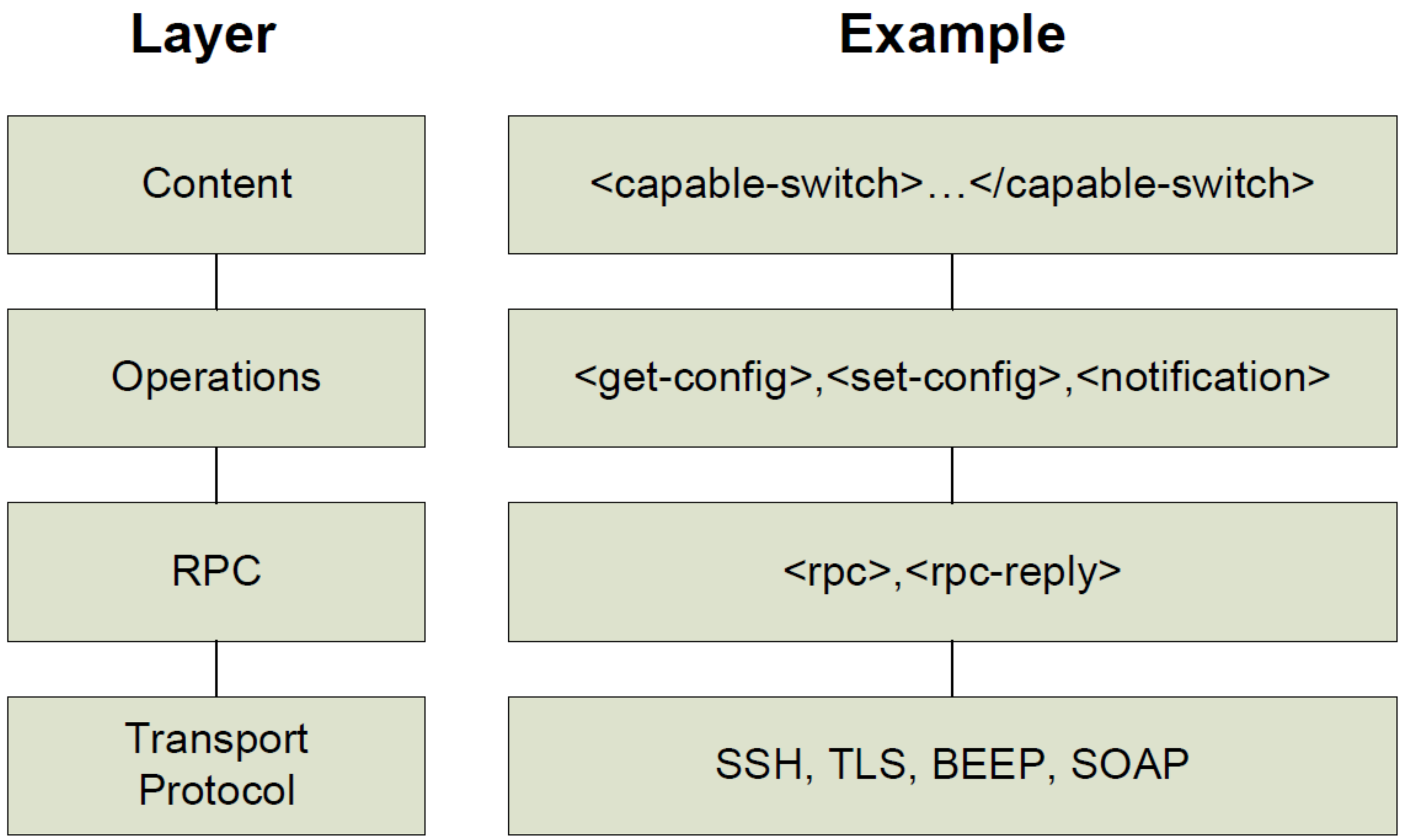
```
All
Select
Indirect
Fast Failover
```

Groups represent sets of actions for flooding, as well as more complex forwarding semantics (e.g. multipath, fast reroute, and link aggregation). As a general layer of indirection, groups also enable multiple flows to forward to a single identifier (e.g. IP forwarding to a common next hop). This abstraction allows common output actions across flows to be changed efficiently.

OF Config


OF Configuration Point
OF Controller
OF Controller
OF-Config
OF Protocol
OF Protocol
Secure channel
Secure channel
OF Logical Switch
OF Logical Switch
OF Resources
Flow Table 0
Flow Table N
Port 0
Port N
OpenFlow-Capable Switch

NETCONF


Layer
Example
Content
<capable-switch>…</capable-switch>
Operations
<get-config>,<set-config>,<notification>
RPC
<rpc>,<rpc-reply>
Transport Protocol
SSH, TLS, BEEP, SOAP

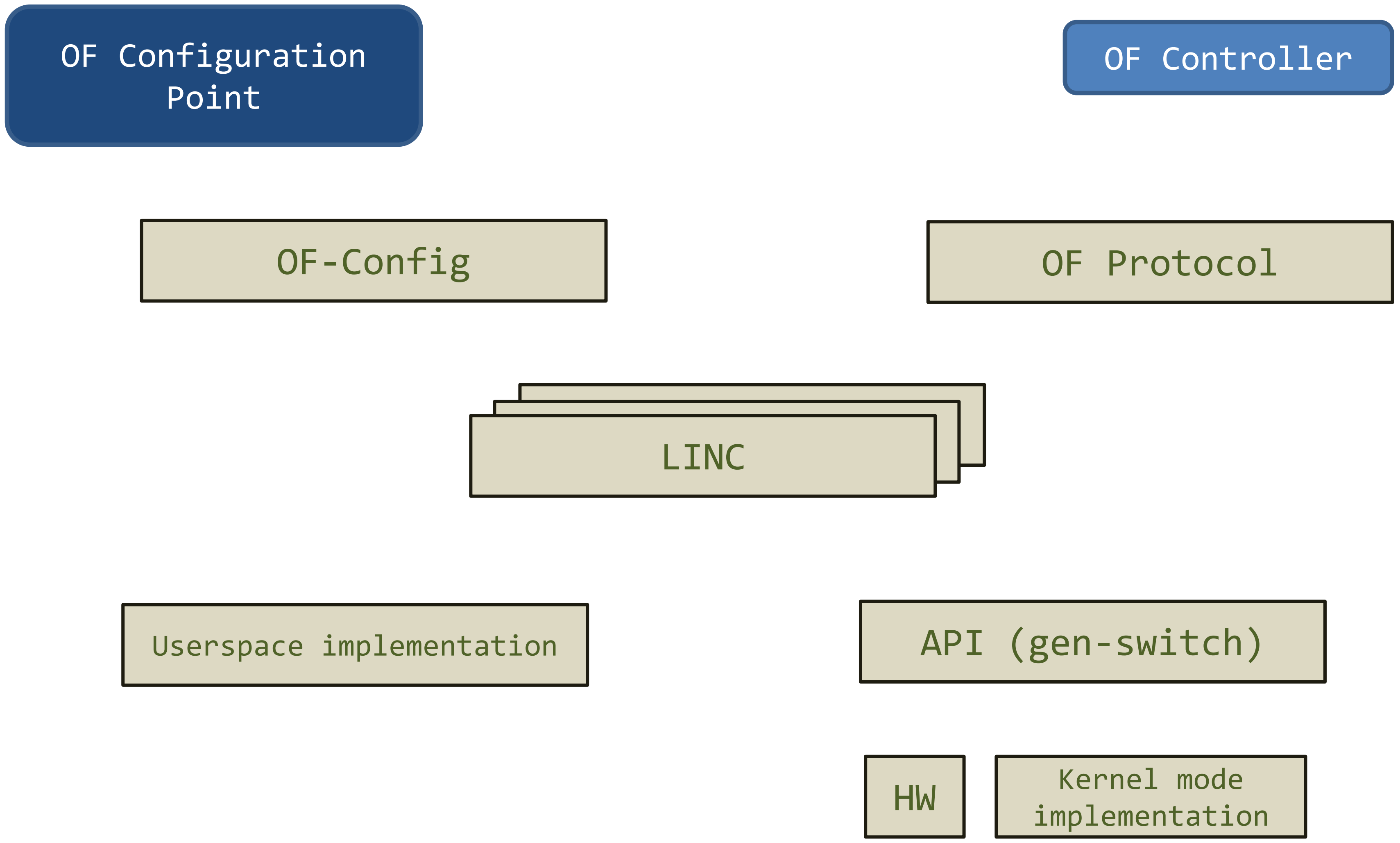
```
<capable-switch>
   <id>CapableSwitch0</id>

   <configuration-points>
   ...
   </configuration-points>

   <resources>
   ...
   </resources>

   <logical-switches>
   ...
   </logical-switches>
</capable-switch>
```

LINC switch


OF Configuration Point
OF Controller
OF-Config
OF Protocol
LINC
Userspace implementation
API (gen-switch)
HW
Kernel mode implementation

Reference

- OpenNetworking Foundation (OpenFlow documents)

  https://www.opennetworking.org/about/onf-documents
- FlowForwarding (LINC Switch)

  http://www.flowforwarding.org/
- Floodlight OpenFlow controller

  http://floodlight.openflowhub.org/
- Apache Avro

  http://avro.apache.org
- And me, Dmitry Orekhov (Dmitry_Orekhov@epam.com)