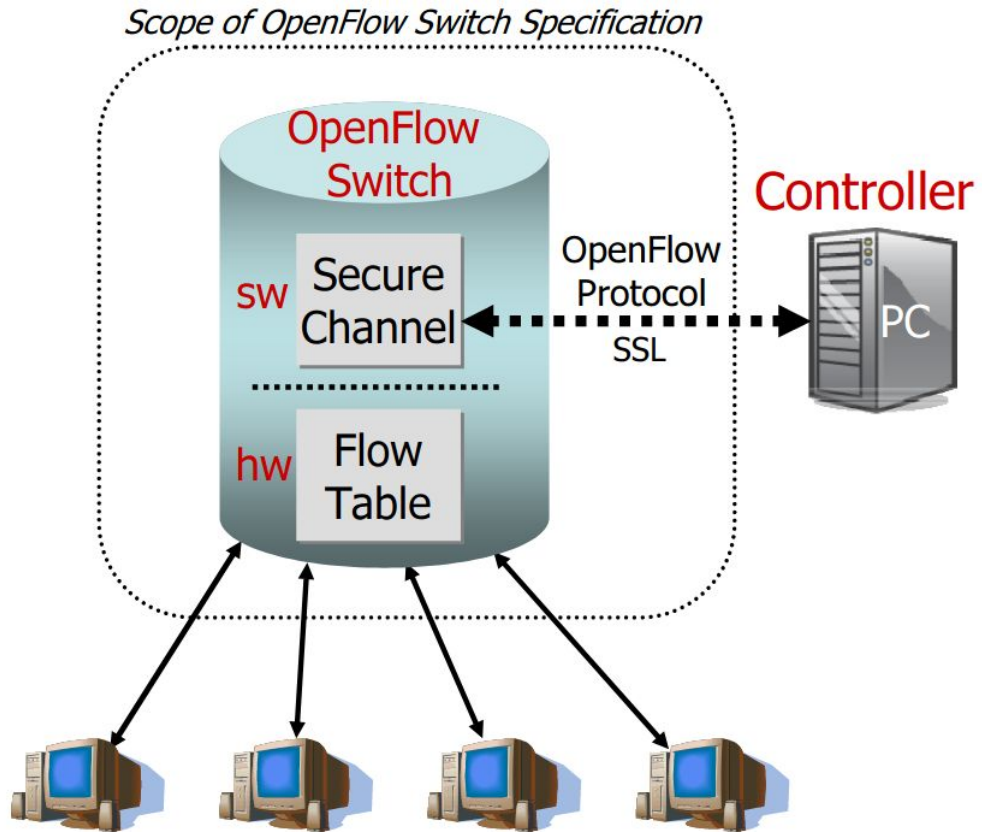# The transience of the physical world

OVS+OVN'21

Frode Nordahl, Senior Engineer, OVN Engineering

CANONICAL · ubuntu

OpenFlow was conceived for use on physical switches

Scope of OpenFlow Switch Specification

OpenFlow Switch

sw — Secure Channel

hw — Flow Table
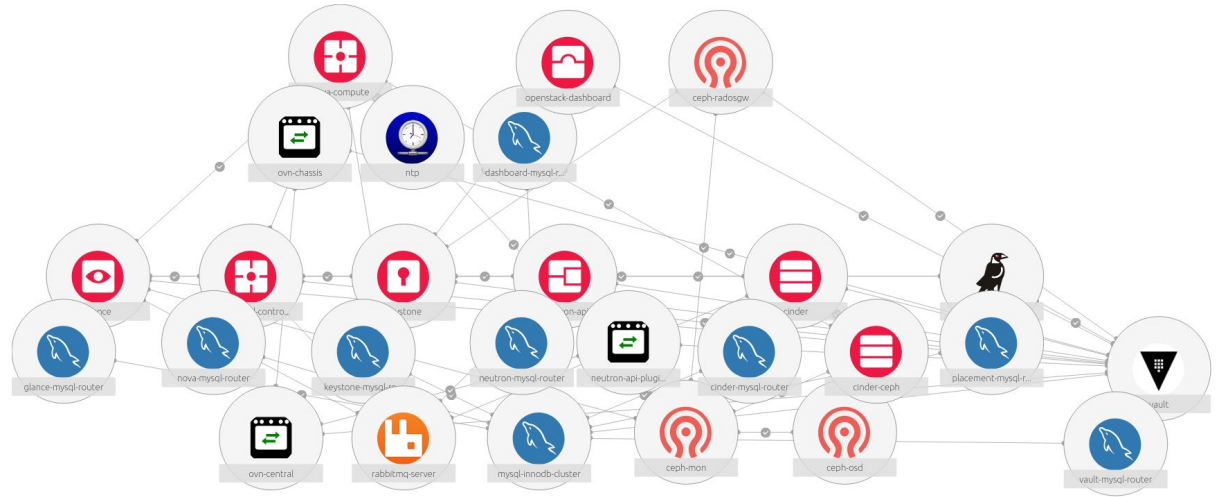
OpenFlow Protocol SSL

Controller

PC

OVS and OVN has been a proving ground for OpenFlow in virtual networks

- Truly distributed architecture

- Secure communication

- Scalable

- Efficient code with small footprint

We've mastered the art of repeatable deployments of complex software

Next frontier is repeatable deployments of networking hardware

- Does that mean teaching the deployment tooling about the quirks of every closed hardware/software ecosystem on the planet?

- Does that mean teaching the deployment tooling about current physical network configuration paradigms? (Generating static configuration)

- Increasing need for integration between workload and physical network (anycast load balancing, liveness detection etc.)
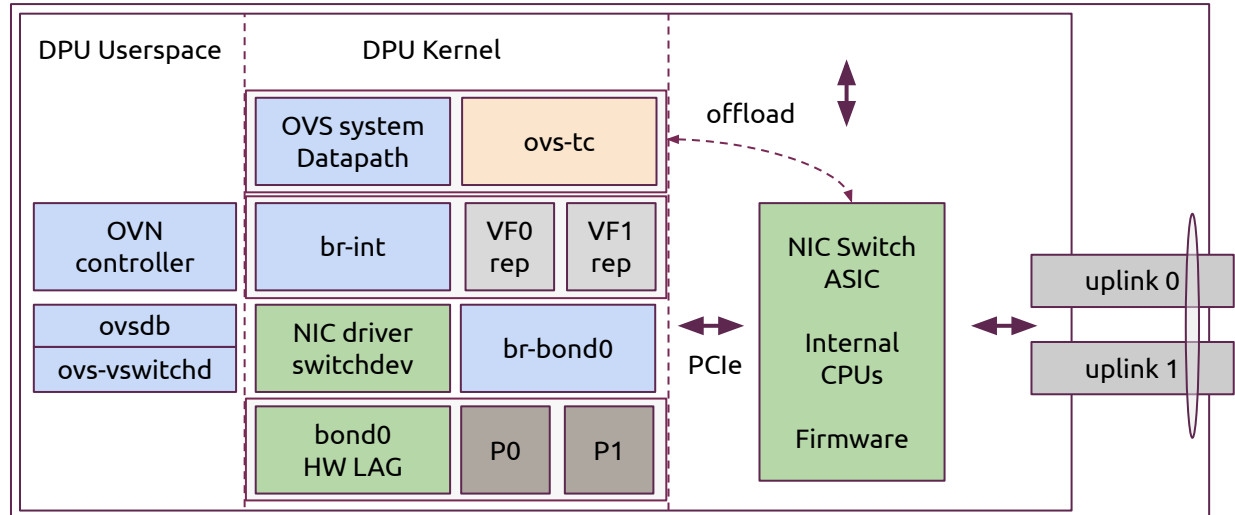
A shift in integration point for inter-vendor compatibility
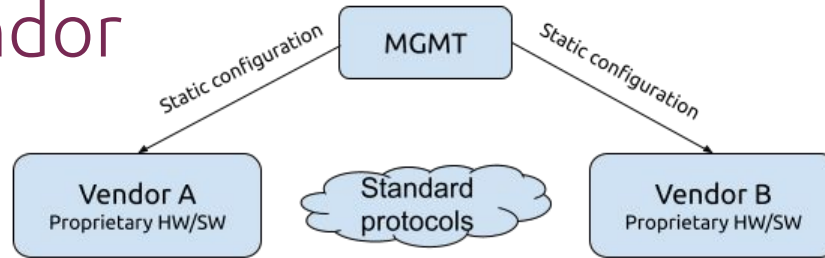
A shift in integration point for inter-vendor compatibility
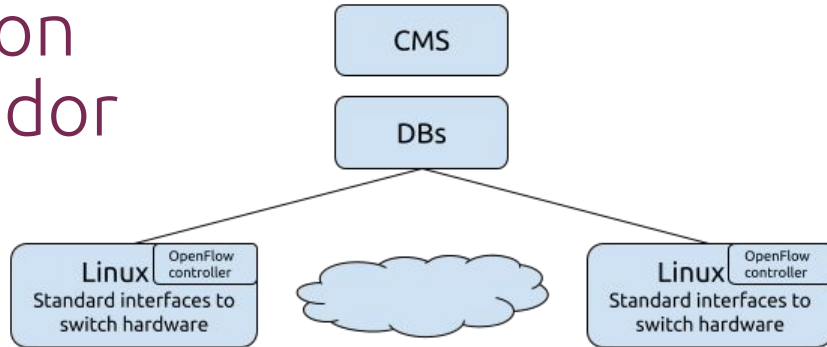
It has already begun

# Comparison of pertinent OF controller characteristics

	Language	Database / Message engine	DB Clustering algorithm	DB Scale Out	CMS API	Distributed by design (OF controller on each switch/chassis)
OVN	C	OVSDB	RAFT	OVSDB Relay	OVSDB	✅
ODL	Java	MD-SAL	RAFT + Gossip	RPC	REST	❌
ONOS	Java	Atomix	RAFT	?	REST	❌
Faucet	Python	YAML	N/A	N/A	N/A	❌

Why is **distributing** the OF controller so important?

`actions=controller` ← makes the OF controller a run time critical data plane component

Switch operation depends on having flow tables (re-)programmed, local OF controller widens the options of how to reinstate the configuration in the event of a partial outage

Bare metal provisioning systems look alot like the CMSs we're used to work with
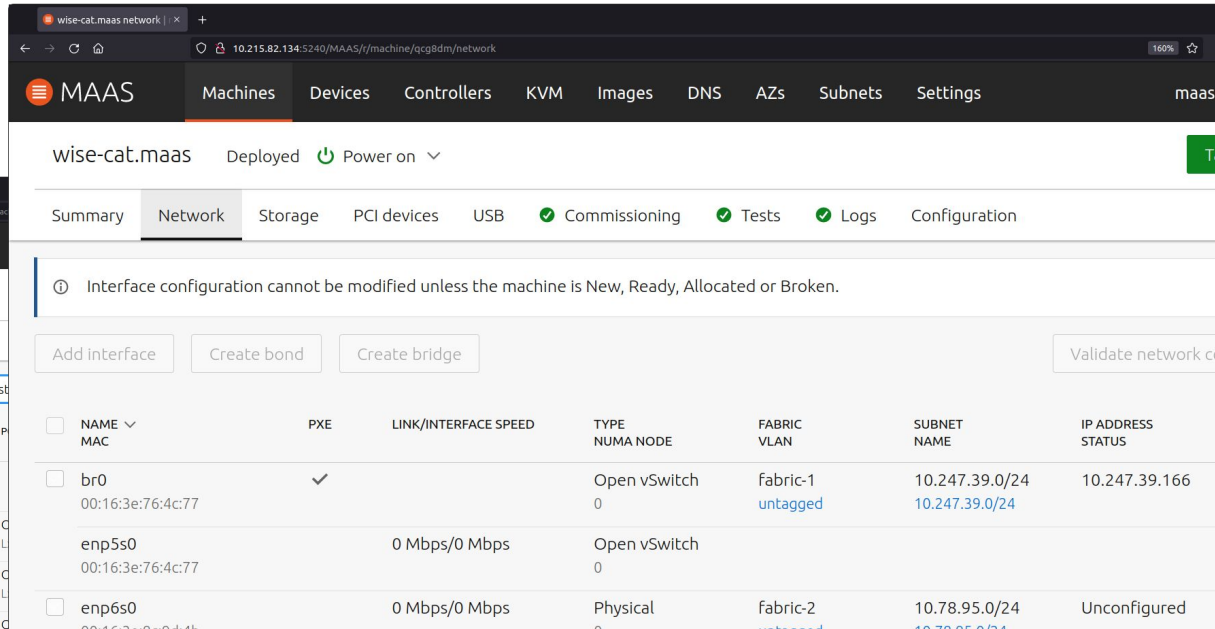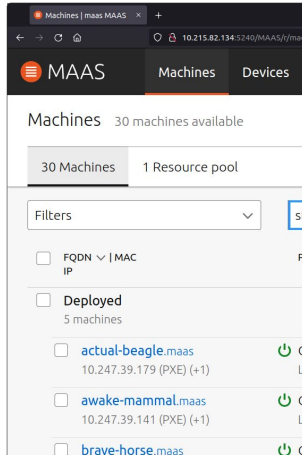
Thank you. Questions?

Let's discuss!
    To: ovs-discuss@openvswitch.org
    Cc: frode.nordahl@canonical.com