SDN and NFV Strategy with Open Source

Daniel Veillard
Standards Principal Manager
Red Hat
Apr 23, 2015

http://veillard.com/Talks/SDNNFV-Beijing-Apr-2015_v2.pdf

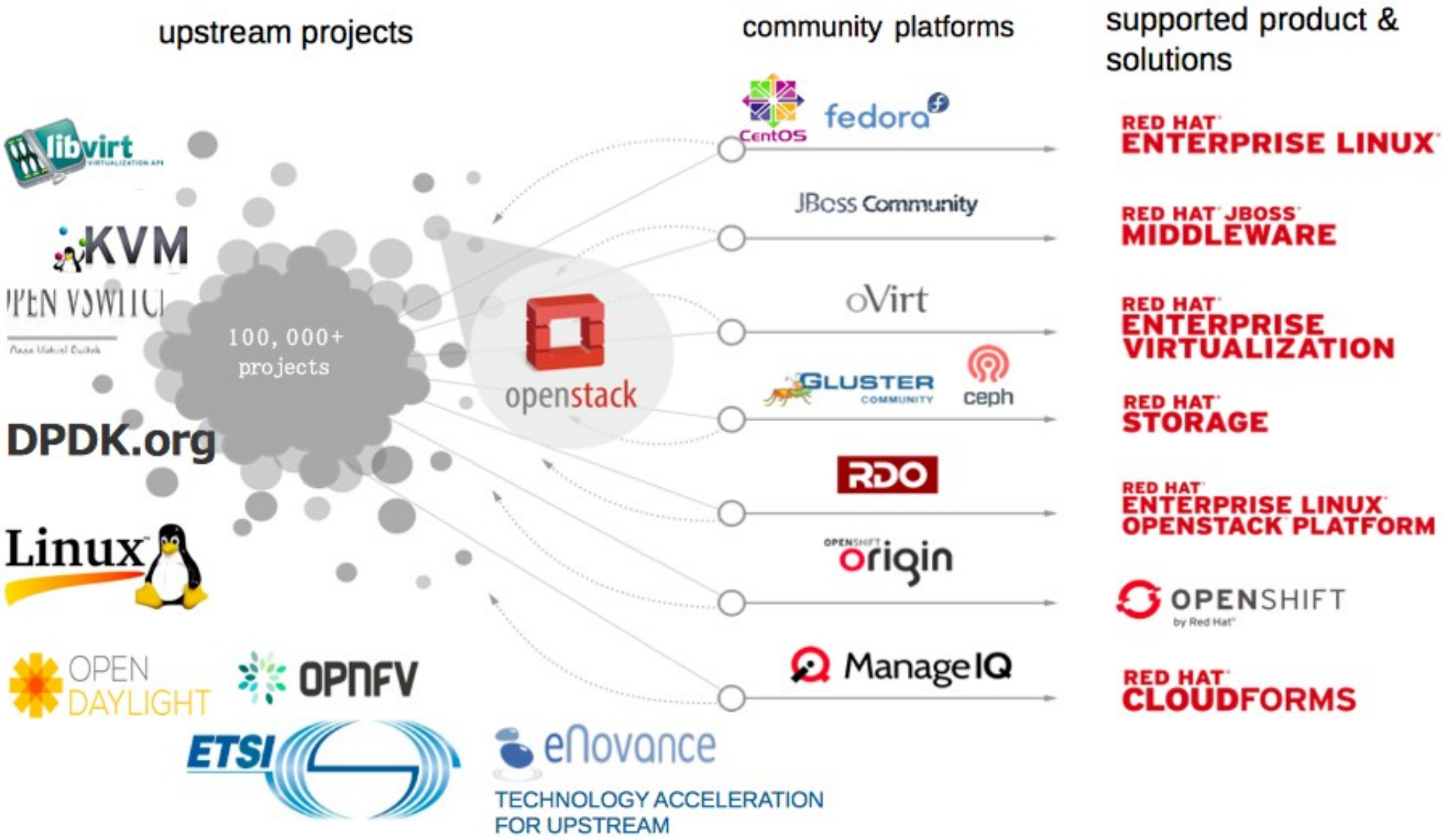
Agenda

- Introduction
- Open Source NFV
- ETSI
- OPNFV
- OpenDaylight
- OVS+DPDK
- Red Hat and NFV

Who am I ?

- Daniel Veillard
- Red Hat Principal Standard Manager
- Software engineer, original author and maintainer of libvirt, libxml2, libxslt
- Experience with W3C, DMTF, OASIS,...
- Manage a team of community specialists
- More than 20 years working on open source

Community and Red Hat



Open Source NFV

Network Functions Virtualization (NFV)

- Network Functions are trapped in function specific HW
  - Expensive to procure, integrate, deploy and operate
- Virtualize Network Functions (VNFs)
- Distribute VNFs on COTS-based IaaS – a Cloud
  - Bring modern data center architecture to Telco network
- Steer traffic with SDN
  - Traffic must traverse chain of functions in well-defined order

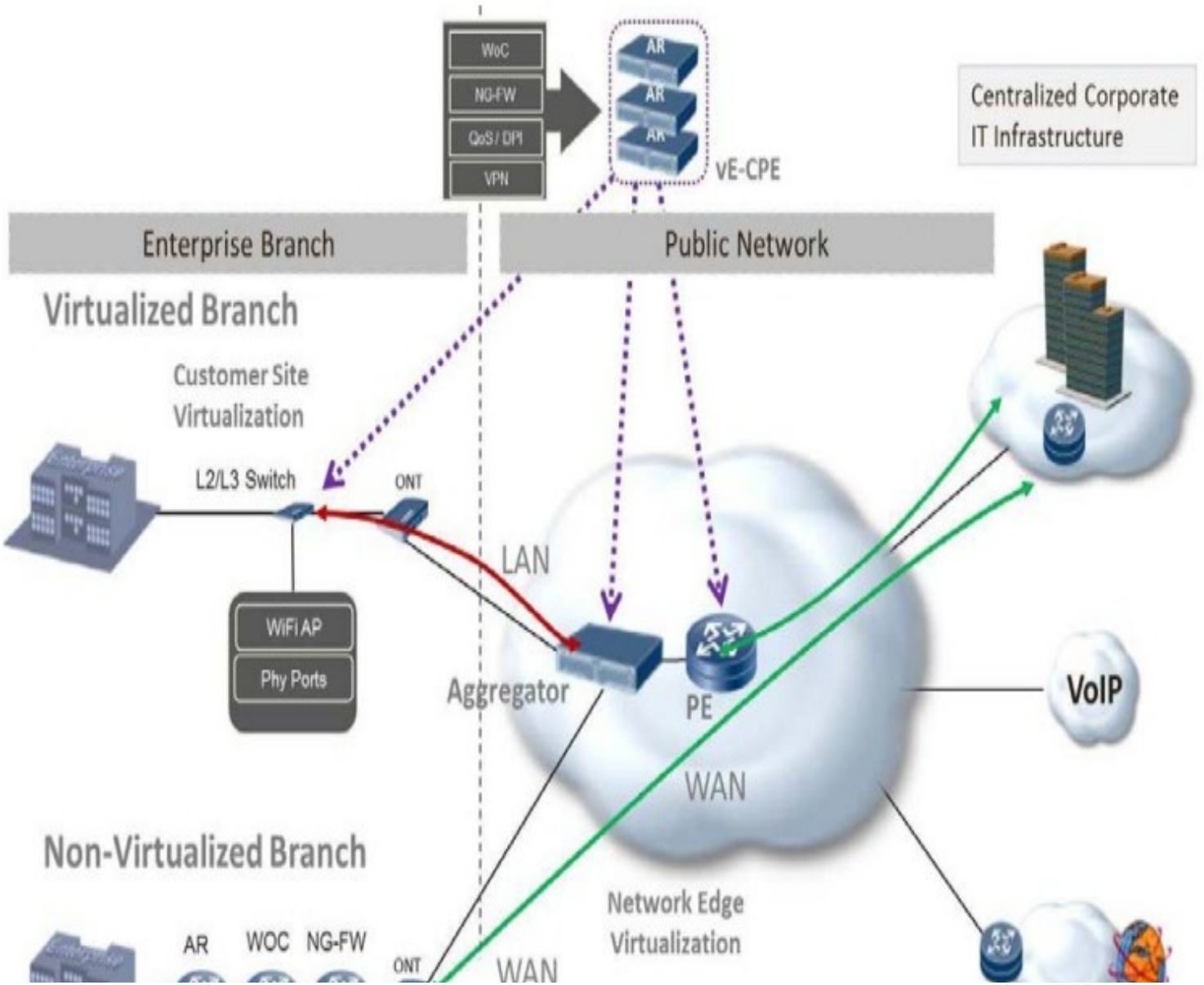
ITU Telecom World 2013 Outcomes Report:

=> Markets: Adapt or die

Why NFV?

- Reduce time-to-market for new services
  - Improve business agility
- Reduce CAPEX and OPEX costs

Virtual Network Function as a Service



Network Functions

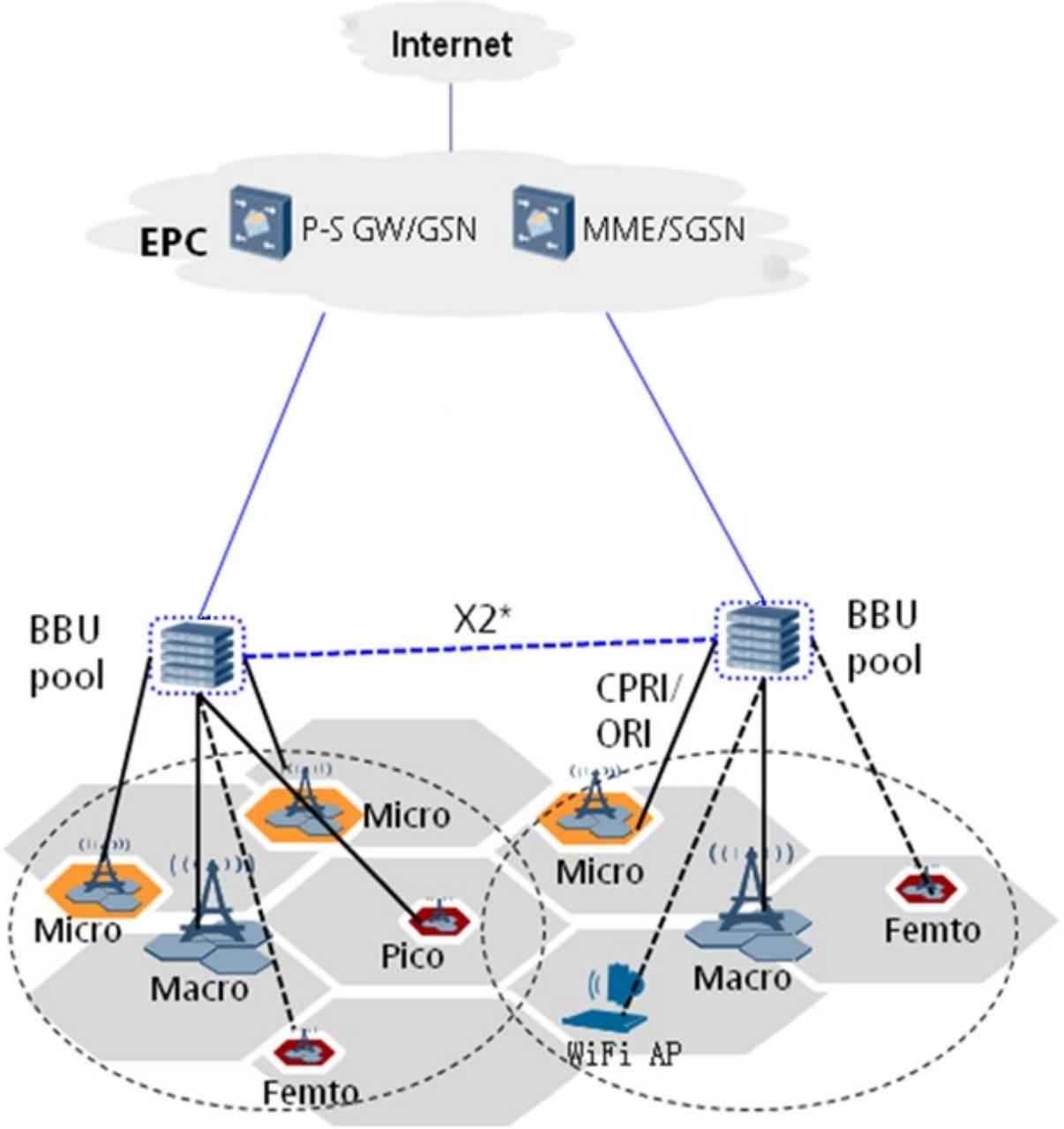
- Access Router
- WAN Optimization
- Firewall
- CG-NAT
- ADC\IPS
- NG-Firewall
- DPI
- Monitoring QoS

(ETSI #2)

Example: VNFaaS

- Requires service chaining support in OpenDaylight, OpenStack Neutron
  - Enable Nova instances as nodes in a Neutron service chain
  - SFC support in OpenDaylight
- Performance
  - DPDK-accelerated Open vSwitch
- Reliability
  - HA instances in Nova

Virtualized Mobile Base Station



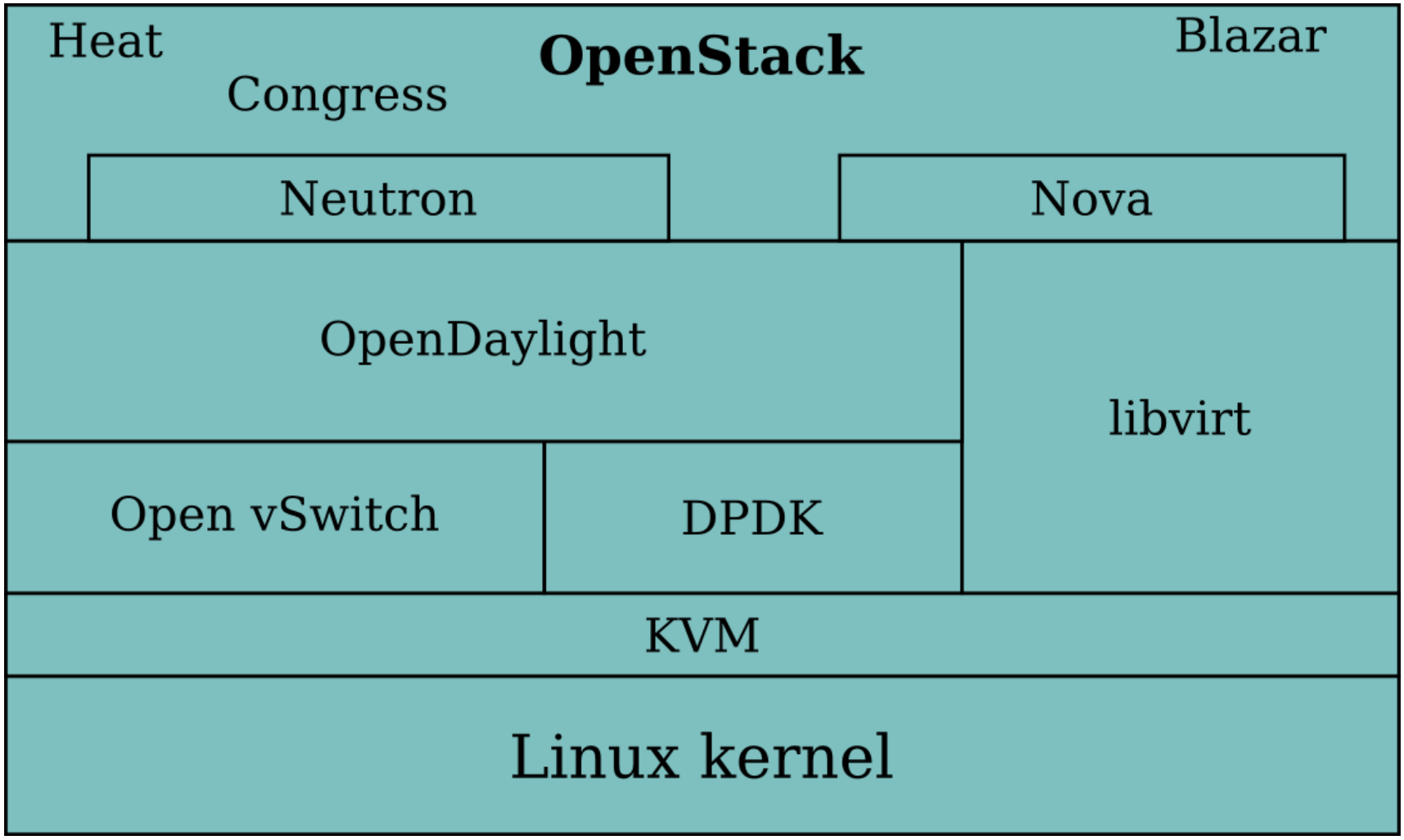
- Multiple RAN technologies from multiple vendors to be consolidated on a single BS to improve utilization
- BS virtualization can share resources among multiple logical RAN nodes from different systems.
- C-RAN for efficient resource utilization among different physical BSs.
- Baseband Unit (BBU) pool with high performance servers and real-time processing for signaling capacity

(ETSI #4)

Example: C-RAN/vRAN

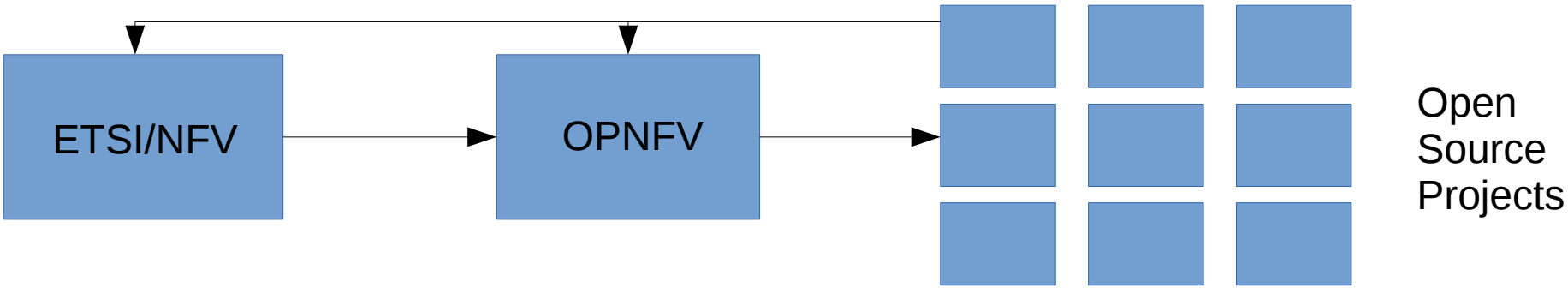
- Determinism
  - Real-time patches to KVM
- Performance
  - DPDK accelerated Open vSwitch
- Reliability at scale
  - HA instances in OpenStack Nova
  - HA OpenStack
  - Fault management and reporting
  - IPv6 support – kernel through Nova, Neutron

Many components on an NFV stack



NFV Communities

ETSI NFV	Standard, driving industry trends, documenting functional requirements
OPNFV	integrating open source components, communicating needs upstream
OpenStack	abstraction/integration of virtual compute, network, storage
libvirt, qemu/KVM, Ceph, Open Daylight, Open vSwitch	Underlying infrastructure management



Upstream first

Upstream first

- Red Hat believes changes should be designed, written and proposed with upstream
- Maintaining significant out-of-tree patches across multiple projects is unsustainable and creates lock-in

Not always easy!

Upstream First

- Requires discipline (small changes, discussed first)
- Relationships (your priority might not be their priority)
- Many communities – impacts multiple parts of the stack

Open Source NFV Summary

- NFV requires a collection of multiple projects
- Use-cases require features which touch many components
- Maintaining forks of multiple projects is unsustainable
- Multiple service providers and NEPs have similar requirements
- Engaging with upstream projects early is required for success

Standards and Open Source

- Traditional “cathedral” way too sequential
  - Requirements, specifications, implementations, interop
- The IETF motto sounds more appropriate

  Rough consensus and running code
- This does require adaptation of process
  - Early public drops of work in progress
  - Feedback loop extended to the given communities

ETSI NFV

- European Telecommunications Standards Institute
- Network Functions Virtualization standard development
  - Analysis of needs toward standardization
    - Gather representative set of telco use cases and requirements
    - ETSI specs only architecture-level (interfaces and information models), not implementation-level (protocols, data models)
  - Sharing experience in development
    - Proof-of-Concept framework, end-to-end testing
  - Telco view of the problem
- Red Hat joined the effort and is now full member

What is OPNFV?

- Open Source NFV Reference Implementation
  - Open, Transparent, Merit-based
  - Upstream first
  - Integrate and validate
- Consortium
  - Promote NFV use-cases upstream
  - Develop and test features to fill gaps
  - Red Hat is Platinum Founding member

Goals of OPNFV

- Help OPNFV members engage relevant upstream communities
- Understand NFV requirements (from ETSI NFV), translate them for upstream developers
- Key communities:
  - OpenStack Telco Working Group
  - Open Daylight, Open vSwitch, DPDK
  - CentOS NFV SIG
  - ETSI NFV PoC and Open Source Work Items

Red Hat OPNFV plans

- Focus on identifying and fixing gaps upstream
- Red Hat's OPNFV default will be Red Hat Enterprise Linux OpenStack Platform
- We expect upstream projects to satisfy NFV use-cases, and will drive change upstream to ensure this
- Desire to avoid “Carrier Grade” forks of OpenStack

What is OpenDaylight?

- Open Source SDN
  - Open
  - Transparent
  - Merit-based
- Consortium
  - Facilitate
  - Advocate
  - Support
  - Red Hat is Platinum Founding member

OpenDaylight SDN Platform

- Modular, extensible, pluggable
- Java/OSGi/Karaf based platform
- Evolving towards model driven using YANG
- Multi-protocol
- Eclipse Public License

Red Hat ODL Focus

- Network Virtualization for OpenStack
  - ML2 ODL driver + extensions (L3, *aaS)
- Overlay networks (including HW)
  - Add OVSDB HW_VTEP schema support
  - Underlay informing (e.g. QoS)
- MD-SAL, AAA, OpFlex, SFC
- Infrastructure (testing and performance)

Open vSwitch (OVS)

- Multi-layer virtual switch
  - Configuration managed w/ OVSDB
  - Flow tables controlled by OpenFlow
- Provides connection between VMs on same host
- Provides uplink to physical network via host NIC
- Data fast path in-kernel
- Challenges
  - kernel networking stack can be bottleneck
  - 64 byte packet processing rates suffers

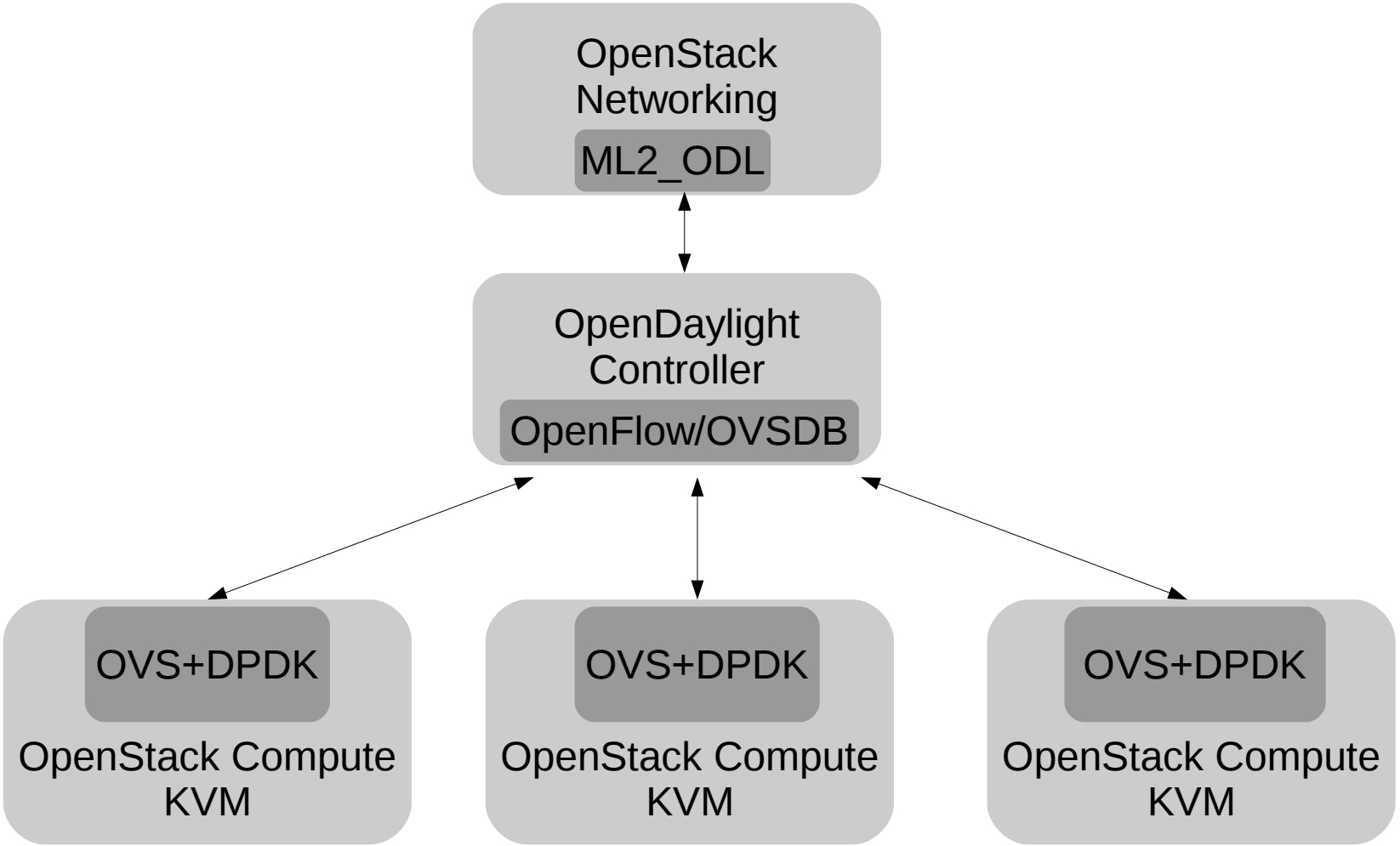
DPDK

- Library for userspace packet processing
- Directly manages NIC with userspace poll mode driver (PMD)
  - Drivers for both physical and virtual NICs
- Polls driver NIC for packets, NIC DMAs directly to application buffers
- Platform specific optimizations
  - Hugepages, NUMA and cacheline aware
  - Batched packet processing
  - CPU instructions (SSE4, AVX, etc)

DPDK Challenges

- API/ABI compatibility, difficult to package in distribution
- Duplicate driver stacks, limited driver support
- Compile time rather than runtime optimizations
- Currently x86-centric
- OVS integration disables kernel features

Putting it all together



Making NFV and OpenStack real

Communities

- Created Sub-Team upstream to focus on NFV
  - https://wiki.openstack.org/wiki/TelcoWorkingGroup
  - Upstreamed NFV related patches in Juno cycle
- OPNFV brings additional resources to focus on NFV requirements in OpenStack
  - http://opnfv.org/

Red Hat and NFV

- Ability to effect change through entire Linux stack
- Active in Open vSwitch, DPDK
- Platinum, founding member of OpenStack Foundation, OpenDaylight Foundation and OPNFV
- Leading developer in Linux kernel, KVM, qemu, libvirt, OpenStack

Partner Ecosystem

- Bring production quality NFV platform to market with rich functionality together with commercial partners
- Open Source to products expertise, long term support

Thank yo