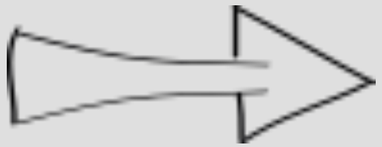


SmartNIC Programming Models

Johann Tönsing

2016-11-09

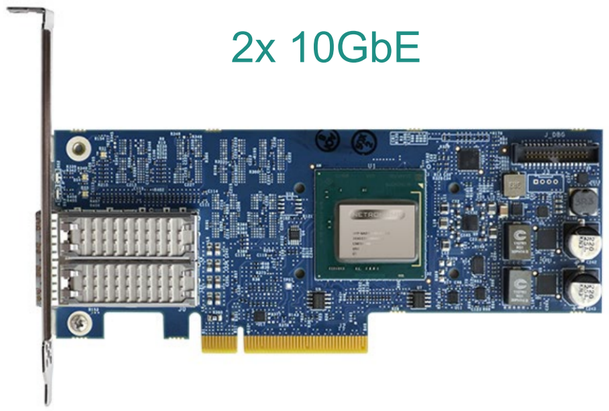
Agenda

- SmartNIC hardware
- Pre-programmed vs. custom (C and/or P4) firmware
- Programming models / offload models
- Switching on NIC, with SR-IOV / virtio data delivery
- SmartNIC performance + TCO
- Silicon and datapath software architectures
- Example code

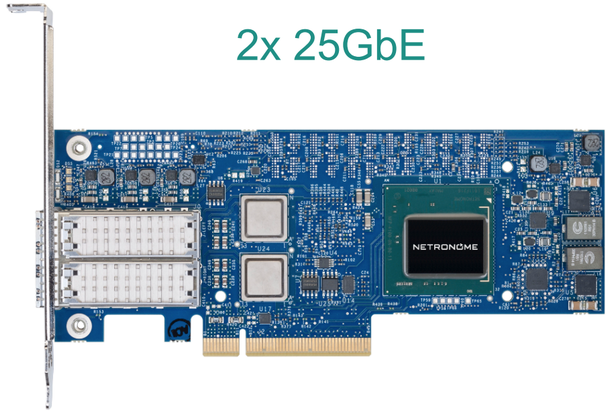
Agilio™ CX SmartNIC Hardware

- Optimized for standard server based cloud data centers
- Low Profile Half Length PCIe form factor, power < 25W
- Based on Netronome's NFP-4xxx silicon (72 C programmable cores, 8 threads each)
- 2GB DRAM for lookup tables / state tables (millions of entries)
- *Dataplane fully implemented in software*

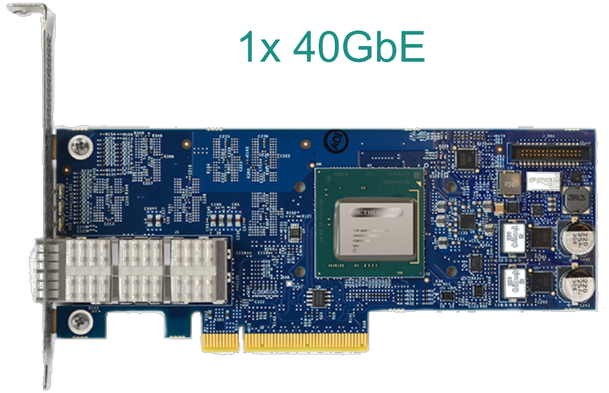
2x 10GbE



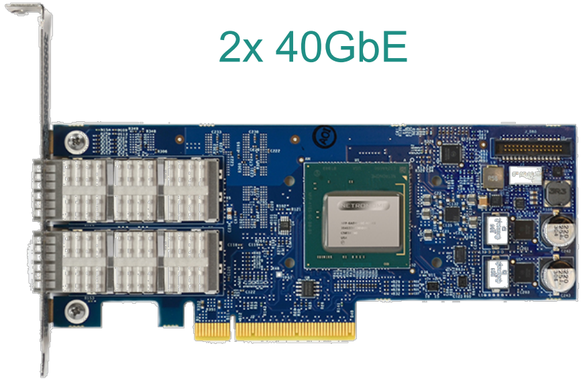
2x 25GbE



1x 40GbE

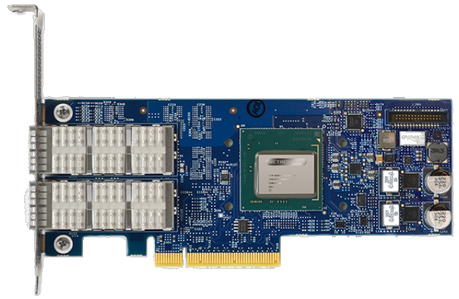


2x 40GbE



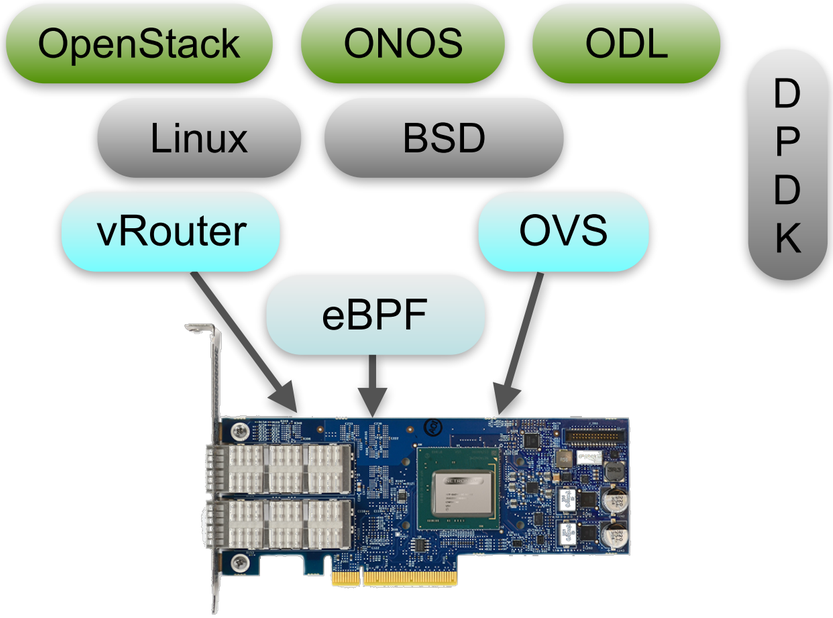
Also available: Agilio™ LX 2x40G / 1x100G with dual PCIe interfaces, 120 cores, 8GB DRAM…

SmartNIC Firmware: Pre-programmed or Custom



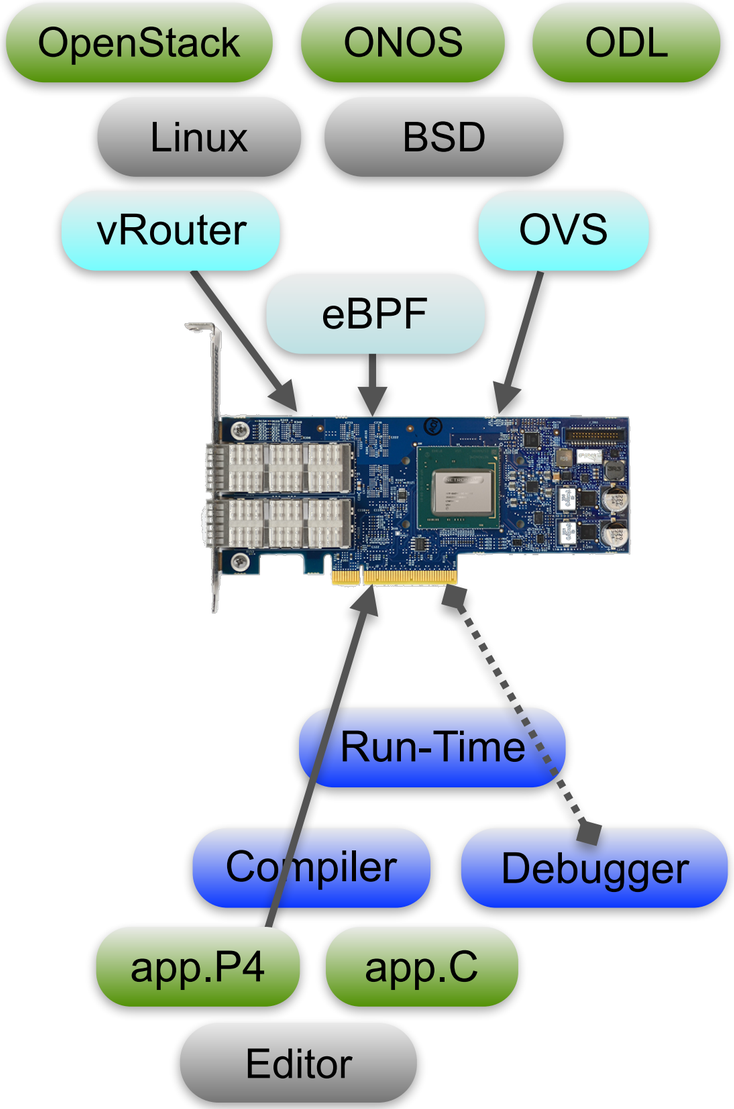
- **SmartNIC with dynamically downloadable firmware**

SmartNIC Firmware: Pre-programmed or Custom



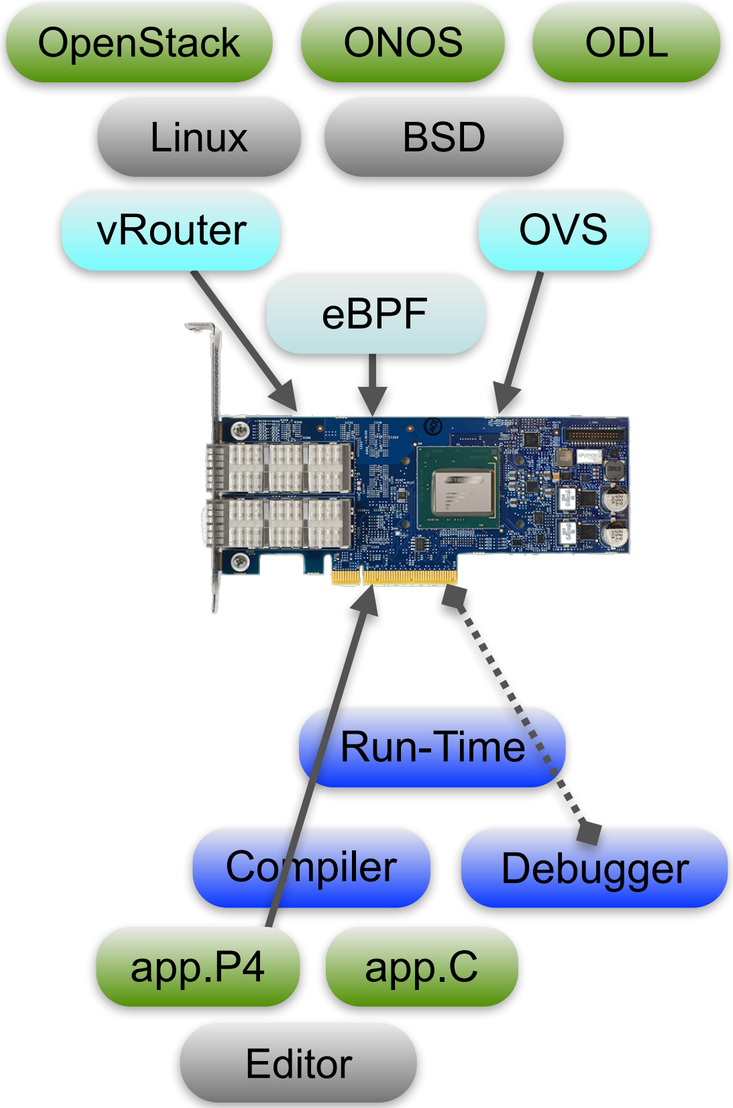
- **OVS / vRouter / eBPF+XDP datapath on host can be accelerated by SmartNIC**
  - ***Firmware / drivers supplied by Netronome***
- **SmartNIC with dynamically downloadable firmware**

SmartNIC Firmware: Pre-programmed or Custom



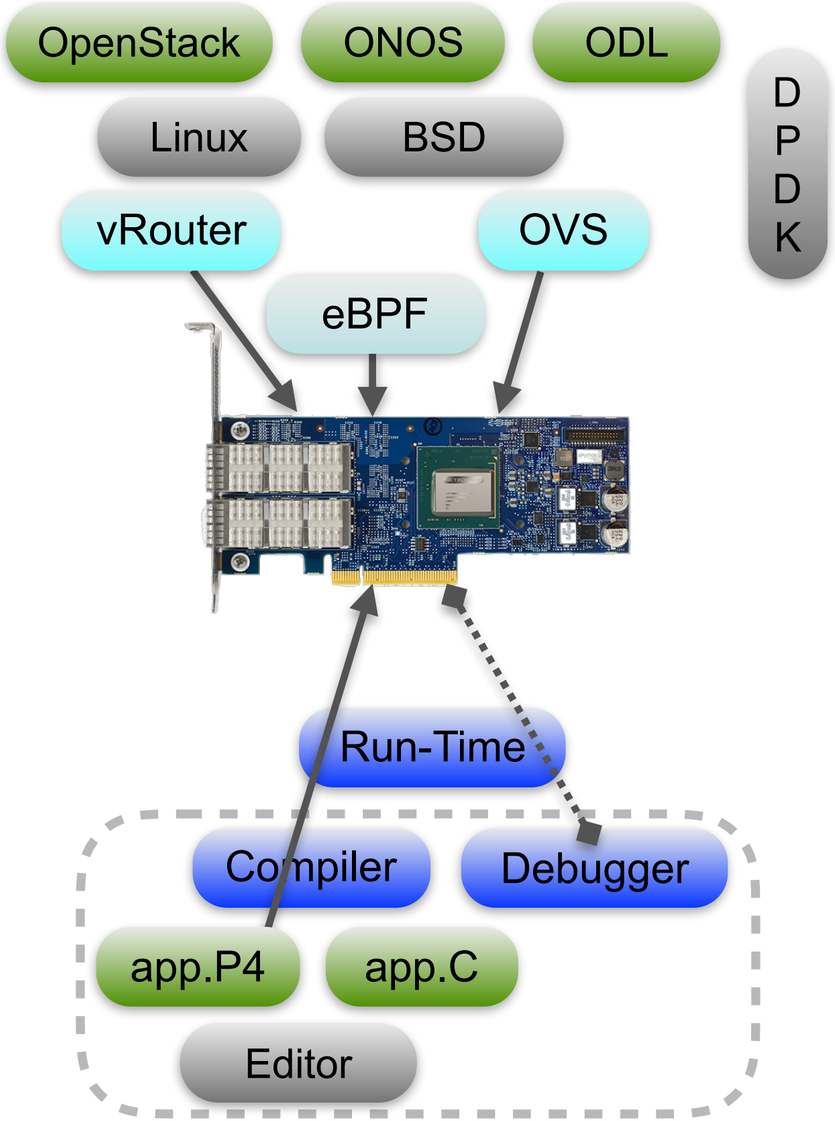
- **OVS / vRouter / eBPF+XDP datapath on host can be accelerated by SmartNIC**
  - ***Firmware / drivers supplied by Netronome***
- **SmartNIC with dynamically downloadable firmware**
- **Firmware can be developed in P4 and/or C**
  - ***Custom dataplane***

SmartNIC Firmware: Pre-programmed or Custom



- **OVS / vRouter / eBPF+XDP datapath on host can be accelerated by SmartNIC**
  - *Firmware / drivers supplied by Netronome*
- **SmartNIC with dynamically downloadable firmware**
- **Firmware can be developed in P4 and/or C**
  - *Custom dataplane*
- **Hybrid - “sandbox / plugin” concept**
  - *Example: C plugin embedded in P4*

SmartNIC Firmware: Pre-programmed or Custom

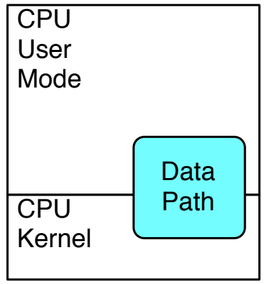


- **OVS / vRouter / eBPF+XDP datapath on host can be accelerated by SmartNIC**
  - *Firmware / drivers supplied by Netronome*
- **SmartNIC with dynamically downloadable firmware**
- **Firmware can be developed in P4 and/or C**
  - *Custom dataplane*
- **Hybrid - “sandbox / plugin” concept**
  - *Example: C plugin embedded in P4*

Programming Models

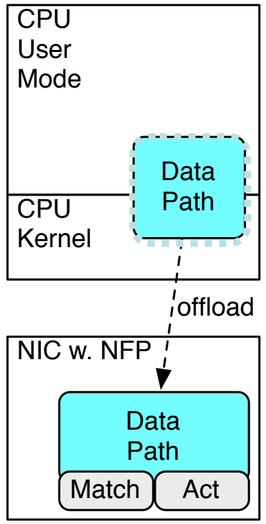
Programming Models

Offload / Acceleration
(transparent -
drop in replacement)

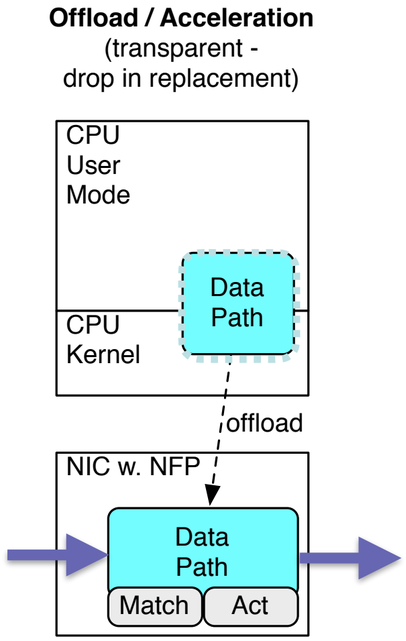


Programming Models

Offload / Acceleration
(transparent -
drop in replacement)

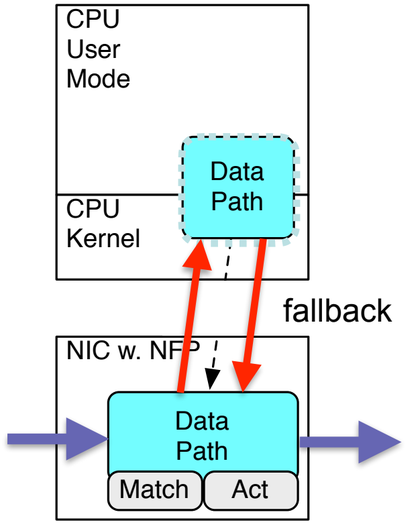


Programming Models



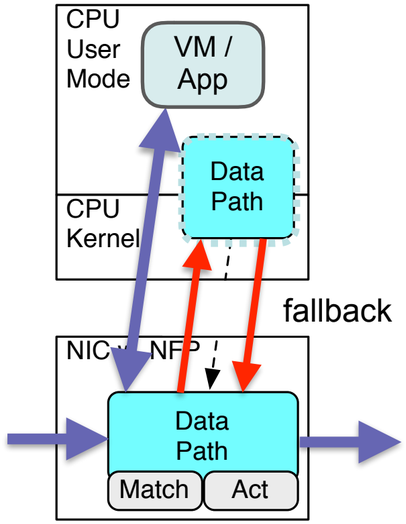
Programming Models

Offload / Acceleration
(transparent -
drop in replacement)

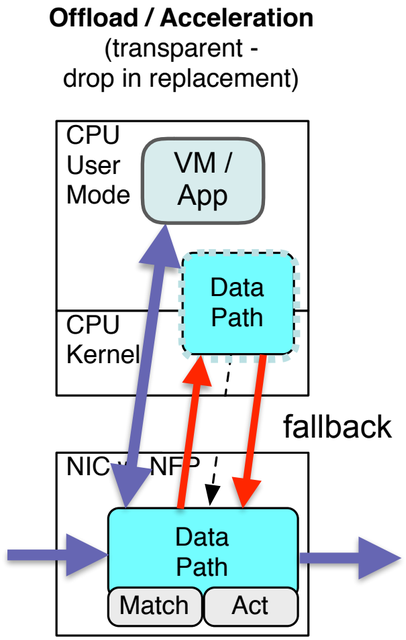


Programming Models

Offload / Acceleration
(transparent -
drop in replacement)



Programming Models



t

OVS/OpenFlow
Match, Action,
Tunnels

Contrail vRouter
Match, Action,
Tunnels

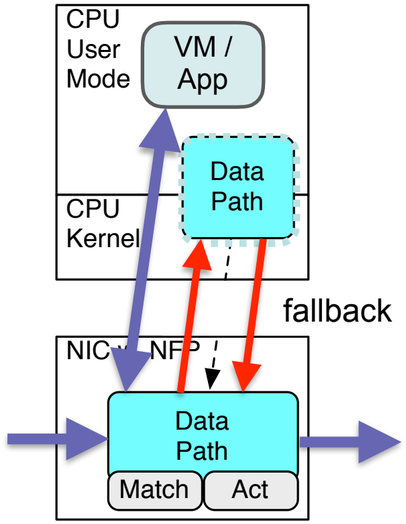
Conntrack
(Firewall)

eBPF / XDP

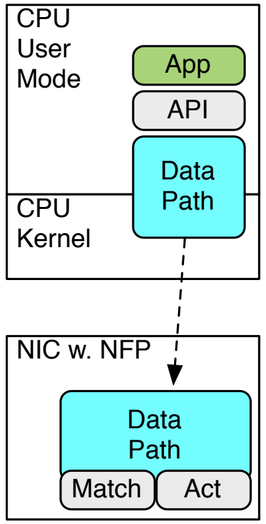
...

Programming Models

Offload / Acceleration
(transparent -
drop in replacement)



CPU Apps Calling APIs
Compatible / Open Sourced
vs. Vendor Extension



t

OVS/OpenFlow
Match, Action,
Tunnels

Contrail vRouter
Match, Action,
Tunnels

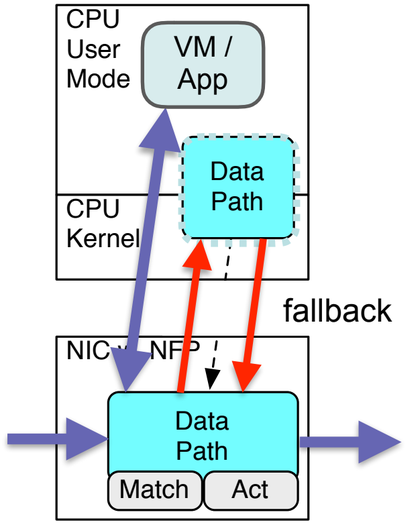
Conntrack
(Firewall)

eBPF / XDP

...

Programming Models

Offload / Acceleration
(transparent -
drop in replacement)



t

OVS/OpenFlow
Match, Action,
Tunnels

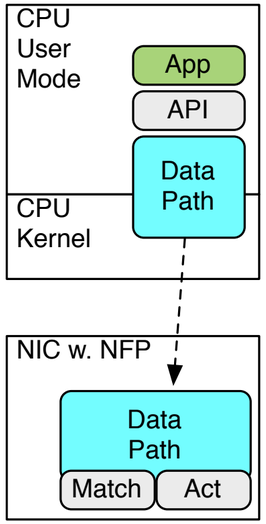
Contrail vRouter
Match, Action,
Tunnels

Conntrack
(Firewall)

eBPF / XDP

...

CPU Apps Calling APIs
Compatible / Open Sourced
vs. Vendor Extension



t

DPDK
Poll Mode Driver

eBPF / XDP APIs

Flow APIs -
Match / Act / Tunnel

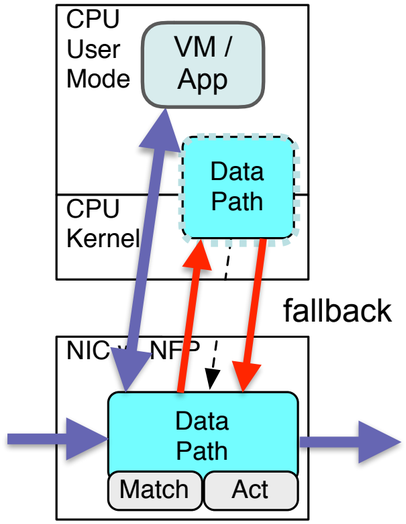
Load Balancing APIs

Crypto APIs

...

Programming Models

Offload / Acceleration
(transparent -
drop in replacement)



t

OVS/OpenFlow
Match, Action,
Tunnels

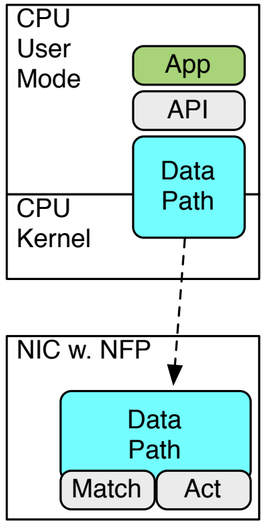
Contrail vRouter
Match, Action,
Tunnels

Conntrack
(Firewall)

eBPF / XDP

...

CPU Apps Calling APIs
Compatible / Open Sourced
vs. Vendor Extension



t

DPDK
Poll Mode Driver

eBPF / XDP APIs

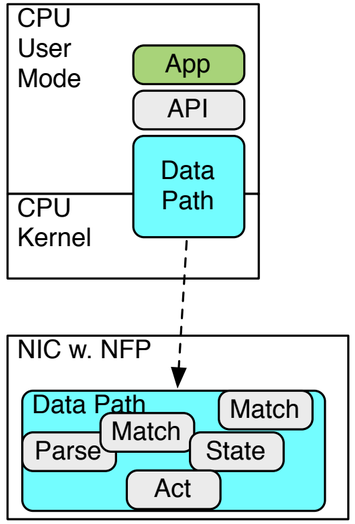
Flow APIs -
Match / Act / Tunnel

Load Balancing APIs

Crypto APIs

...

Flexible Datapath Abstraction
OpenFlow 2.x, P4, PIF, eBPF...



t

Protocol agnostic
flexible parsing

Arbitrary arrangement
of matching tables

Matching without
tables

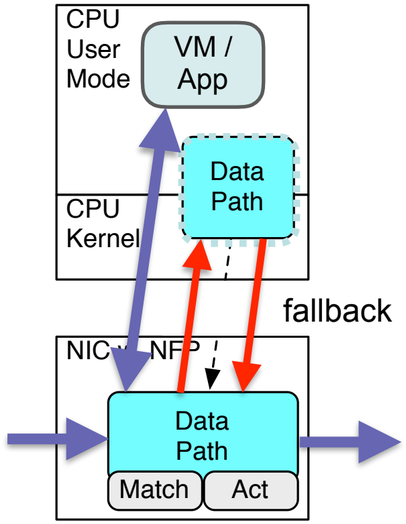
State storage / retrieval

Complex actions

Event handling

Programming Models

Offload / Acceleration
(transparent -
drop in replacement)



t

OVS/OpenFlow
Match, Action,
Tunnels

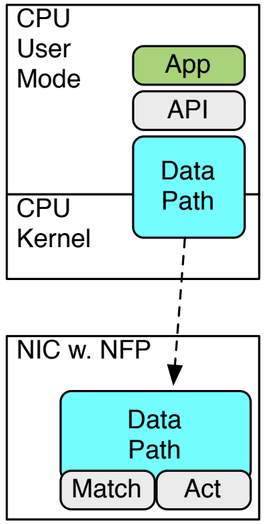
Contrail vRouter
Match, Action,
Tunnels

Conntrack
(Firewall)

eBPF / XDP

...

CPU Apps Calling APIs
Compatible / Open Sourced
vs. Vendor Extension



t

DPDK
Poll Mode Driver

eBPF / XDP APIs

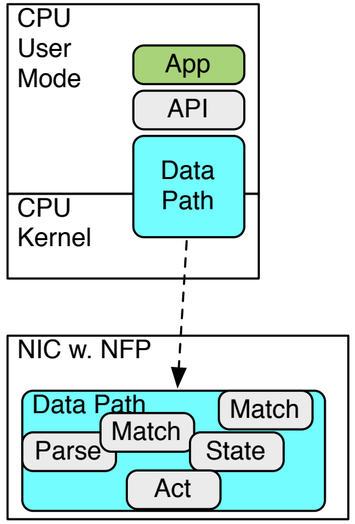
Flow APIs -
Match / Act / Tunnel

Load Balancing APIs

Crypto APIs

...

Flexible Datapath Abstraction
OpenFlow 2.x, P4, PIF, eBPF...



t

Protocol agnostic
flexible parsing

Arbitrary arrangement
of matching tables

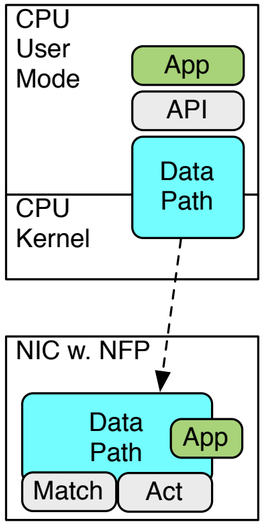
Matching without
tables

State storage / retrieval

Complex actions

Event handling

Hybrid: Datapath
Extensions in CPU / NFP
In C, P4 / PIF, ...



t

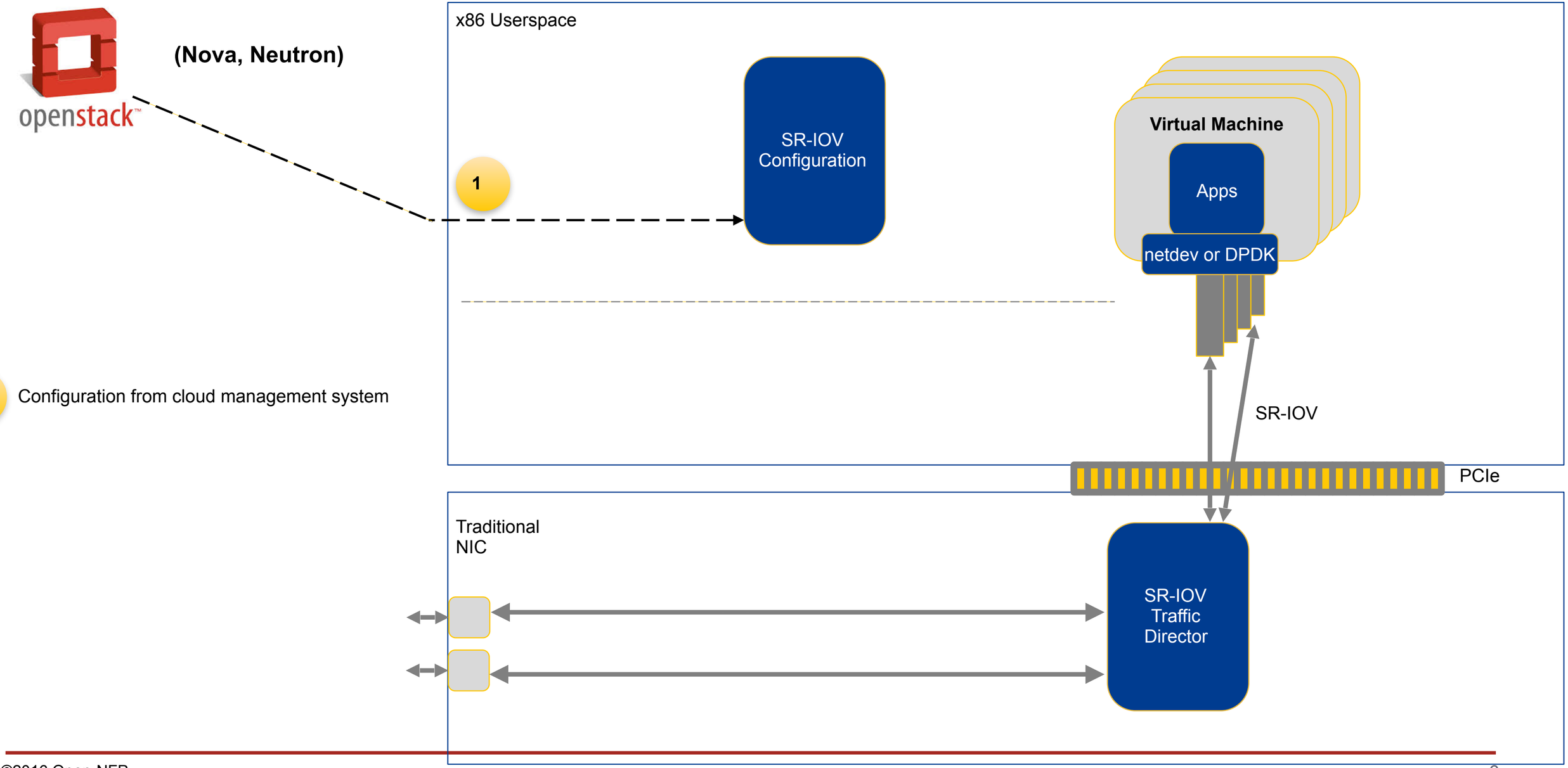
Custom tunnel

Custom action

Custom matching

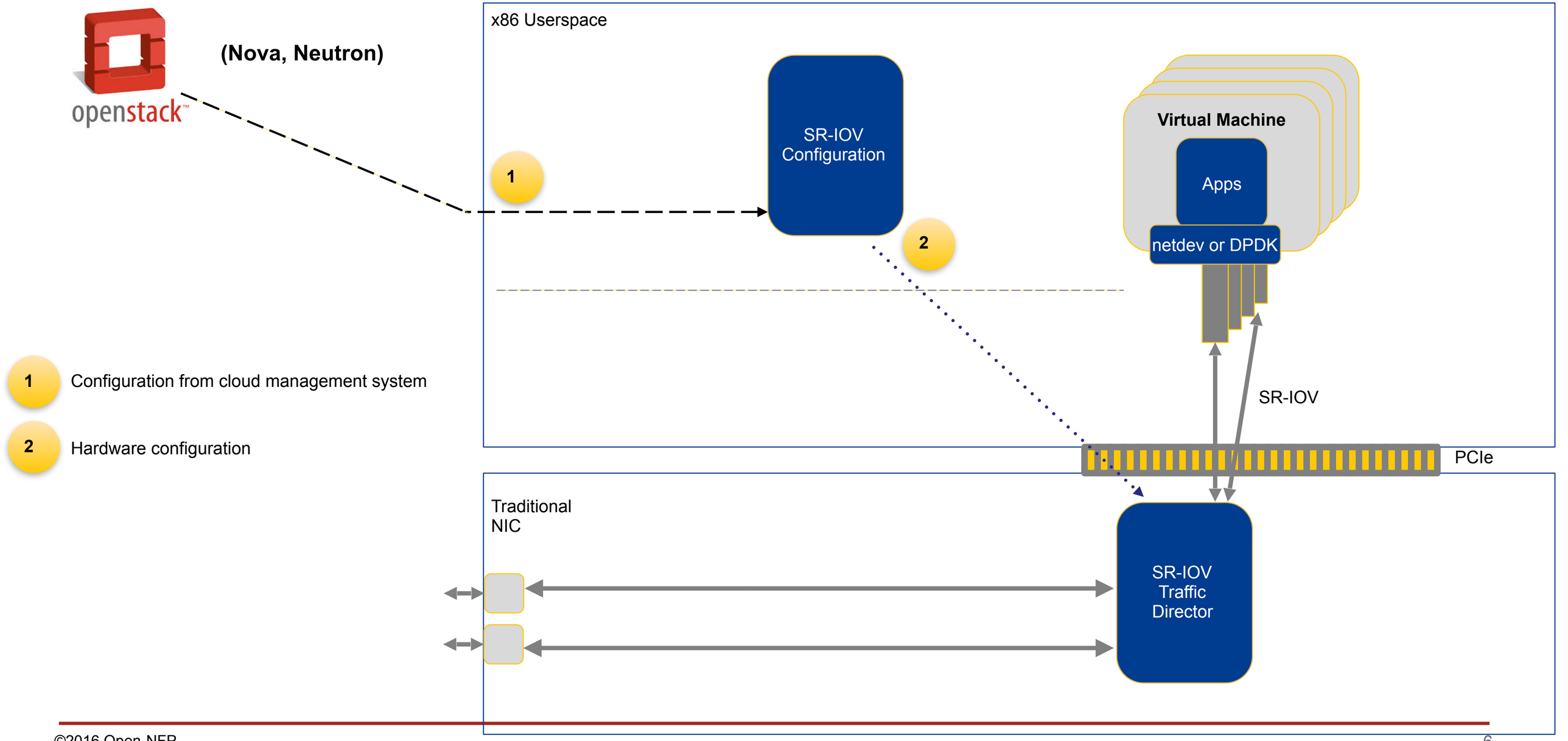
...

Traditional Model: SR-IOV

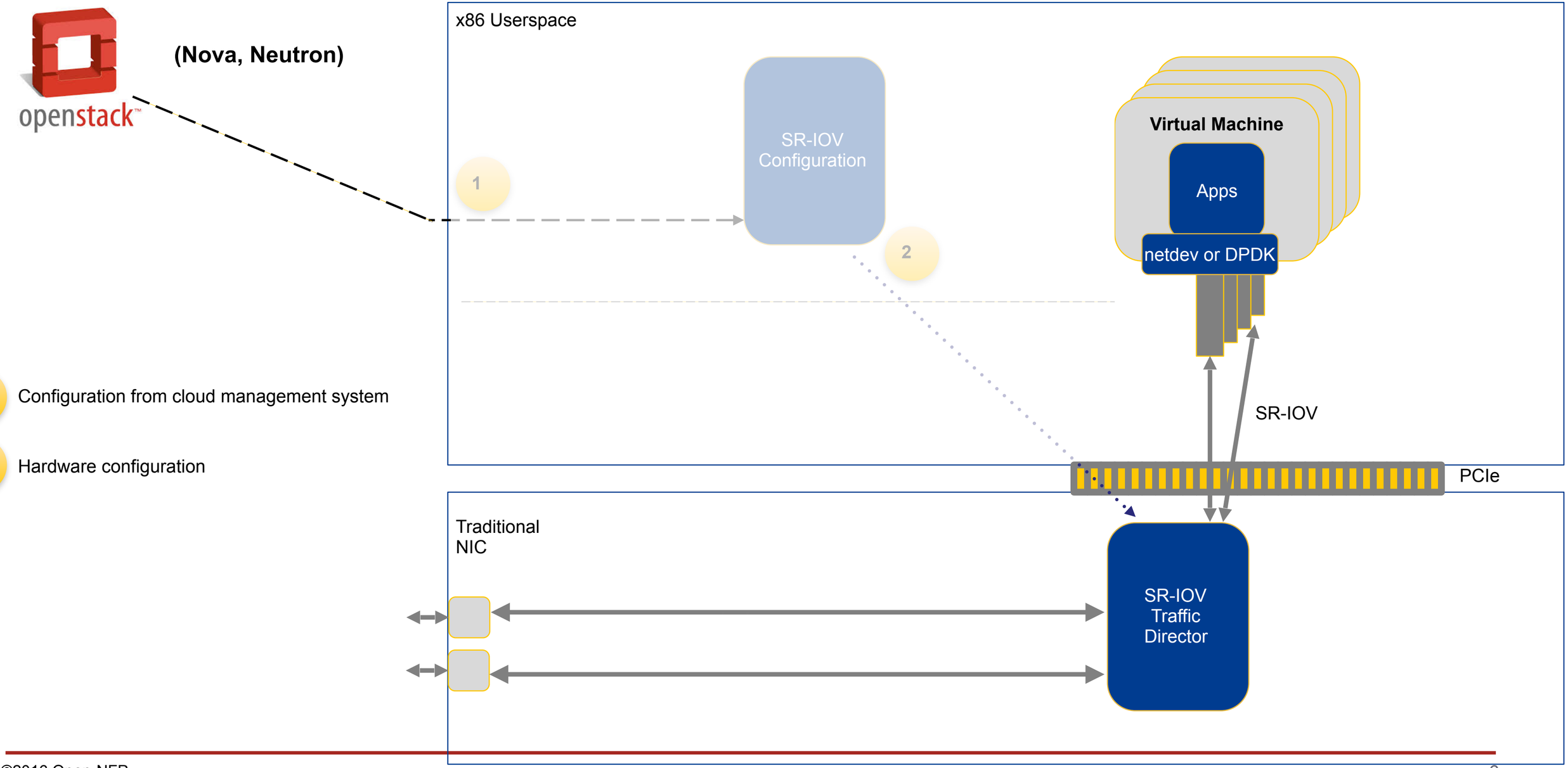


1 Configuration from cloud management system

Traditional Model: SR-IOV



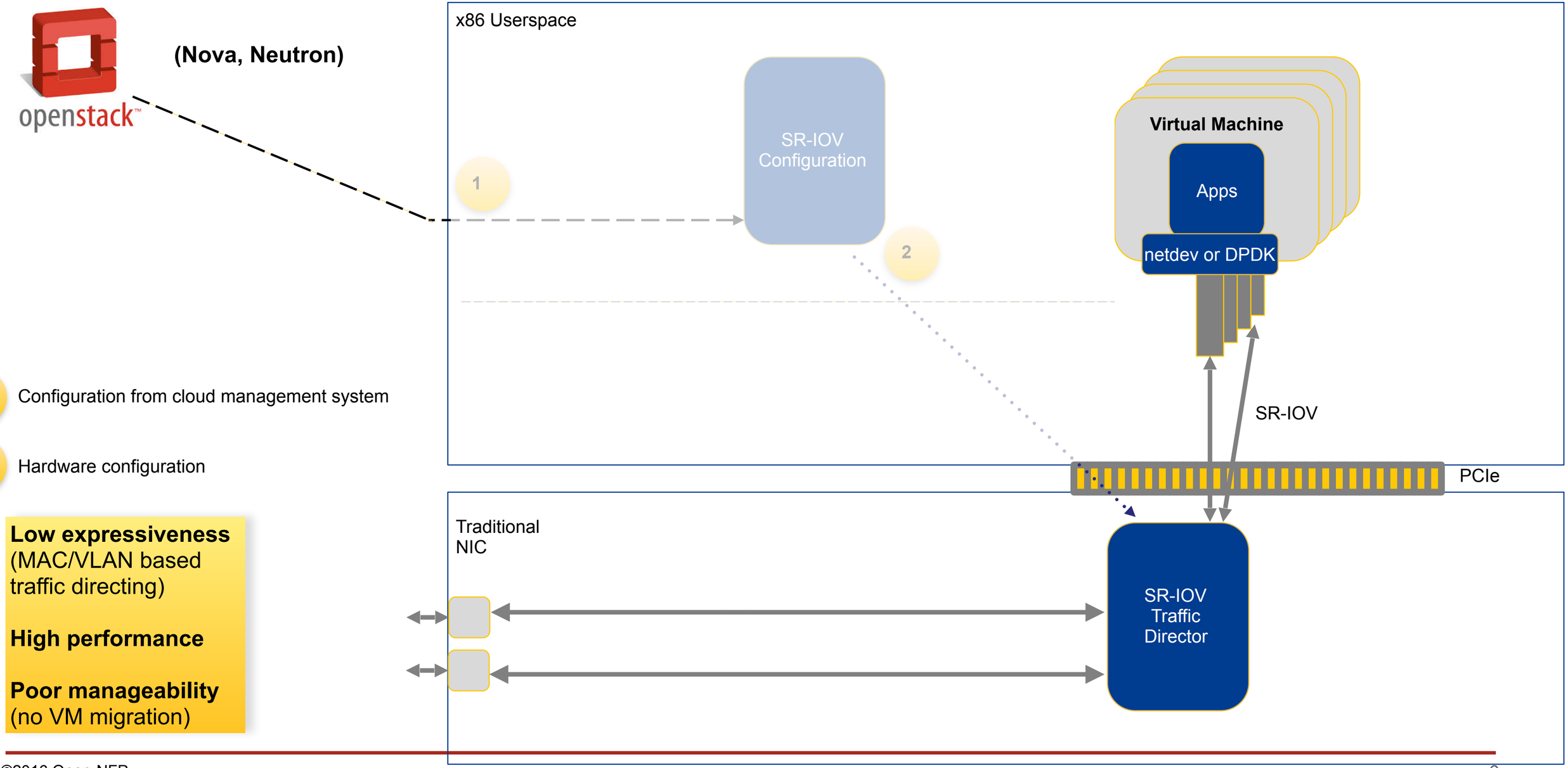
Traditional Model: SR-IOV



1 Configuration from cloud management system

2 Hardware configuration

Traditional Model: SR-IOV



1 Configuration from cloud management system

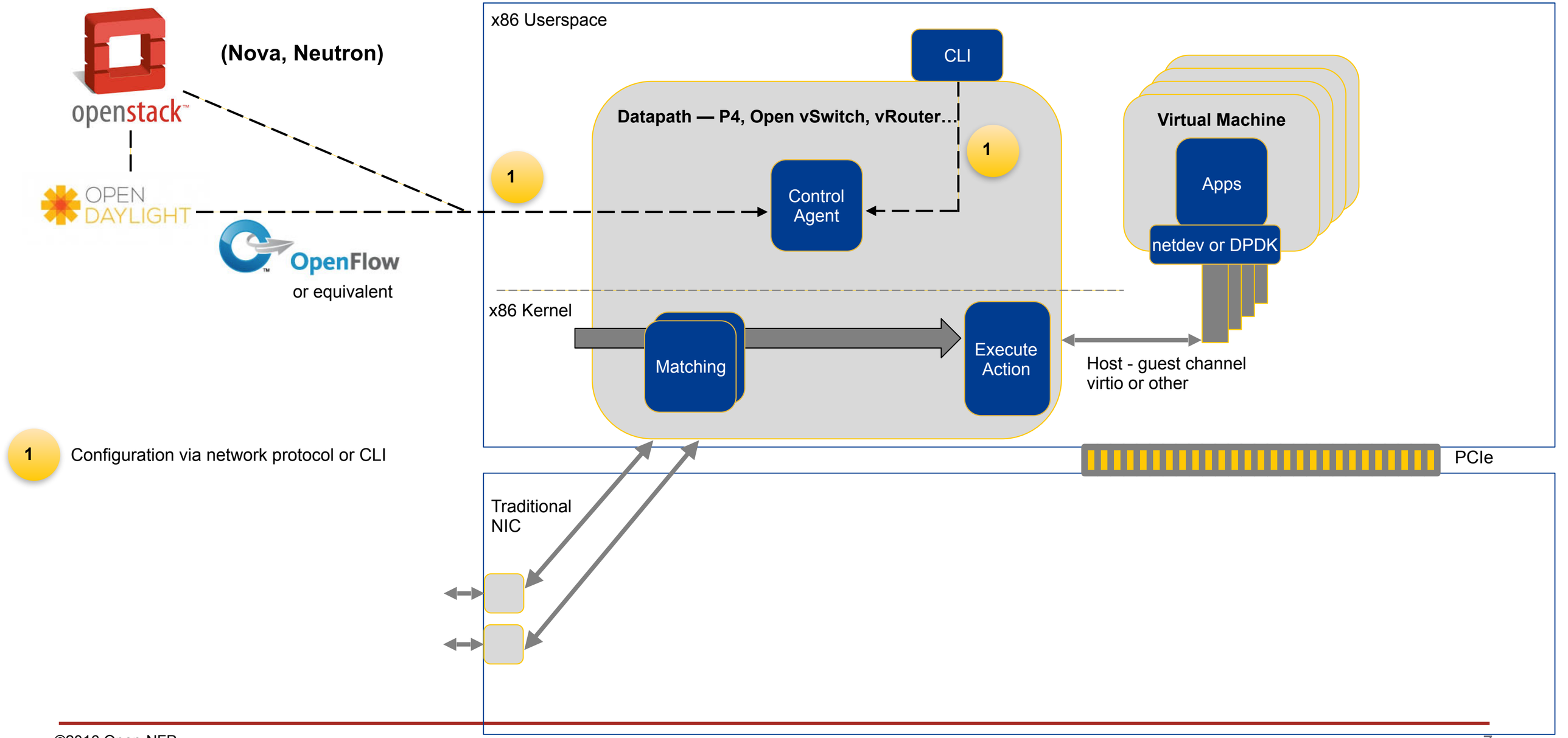
2 Hardware configuration

Low expressiveness (MAC/VLAN based traffic directing)

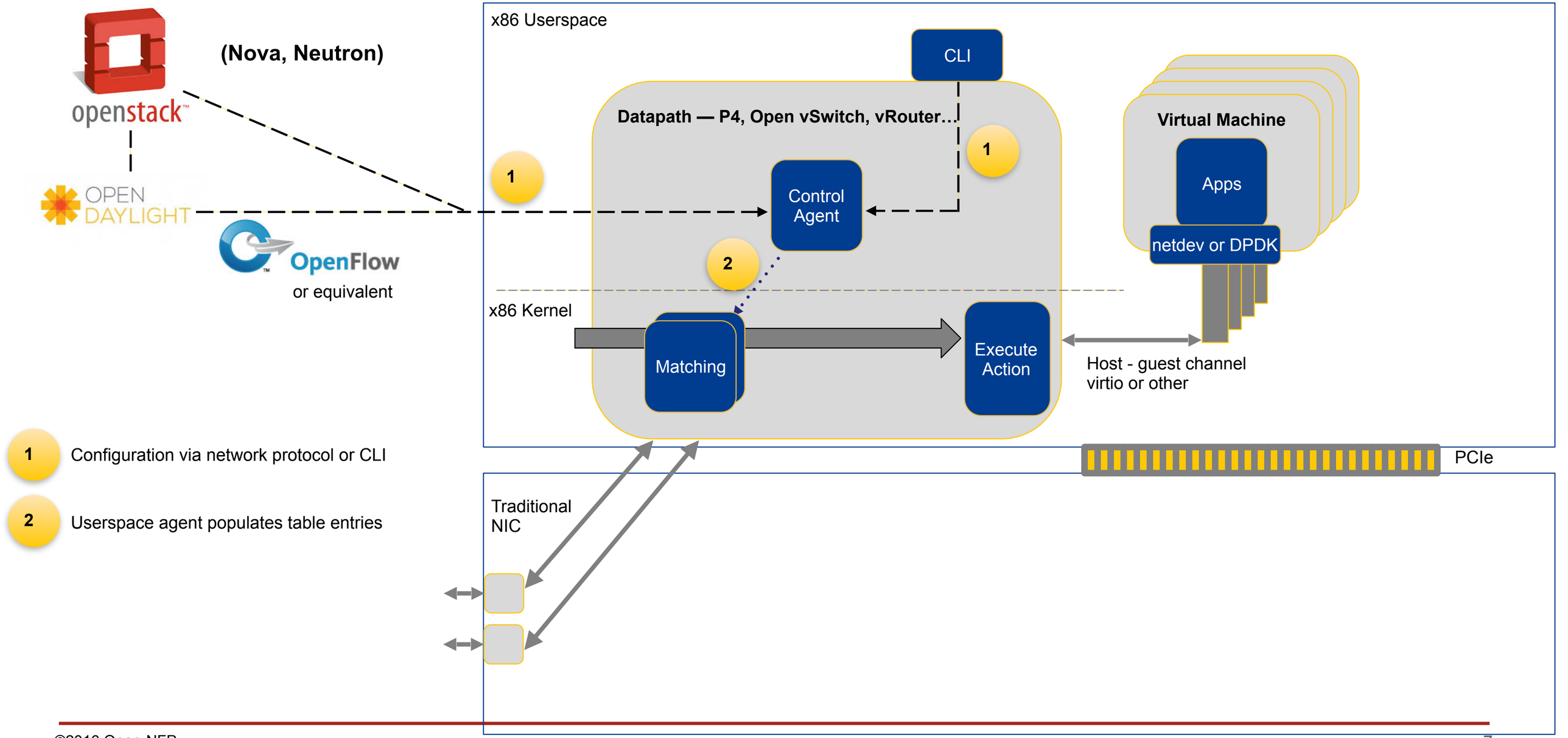
High performance

Poor manageability (no VM migration)

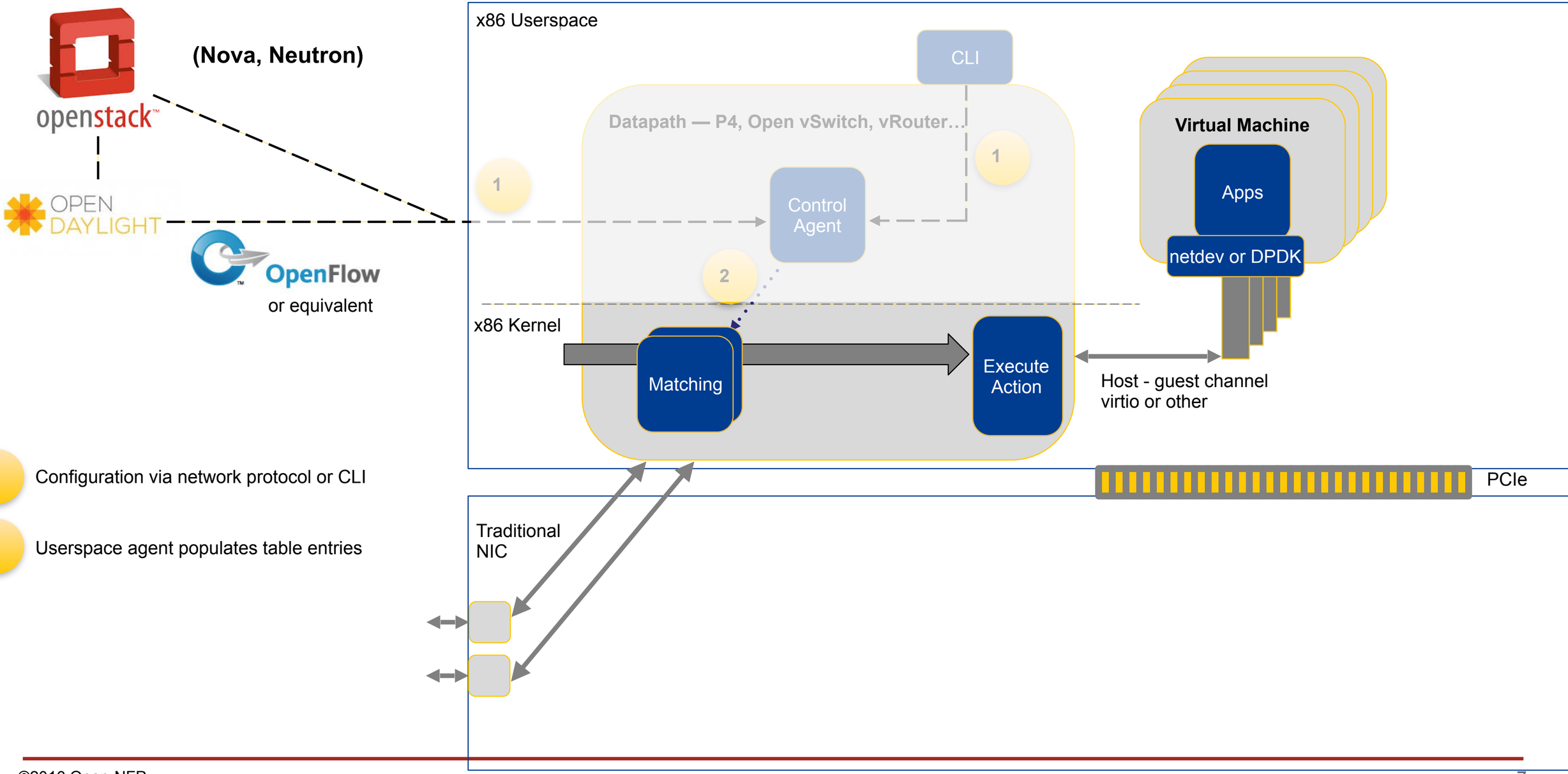
Traditional Model: Host Runs Datapath



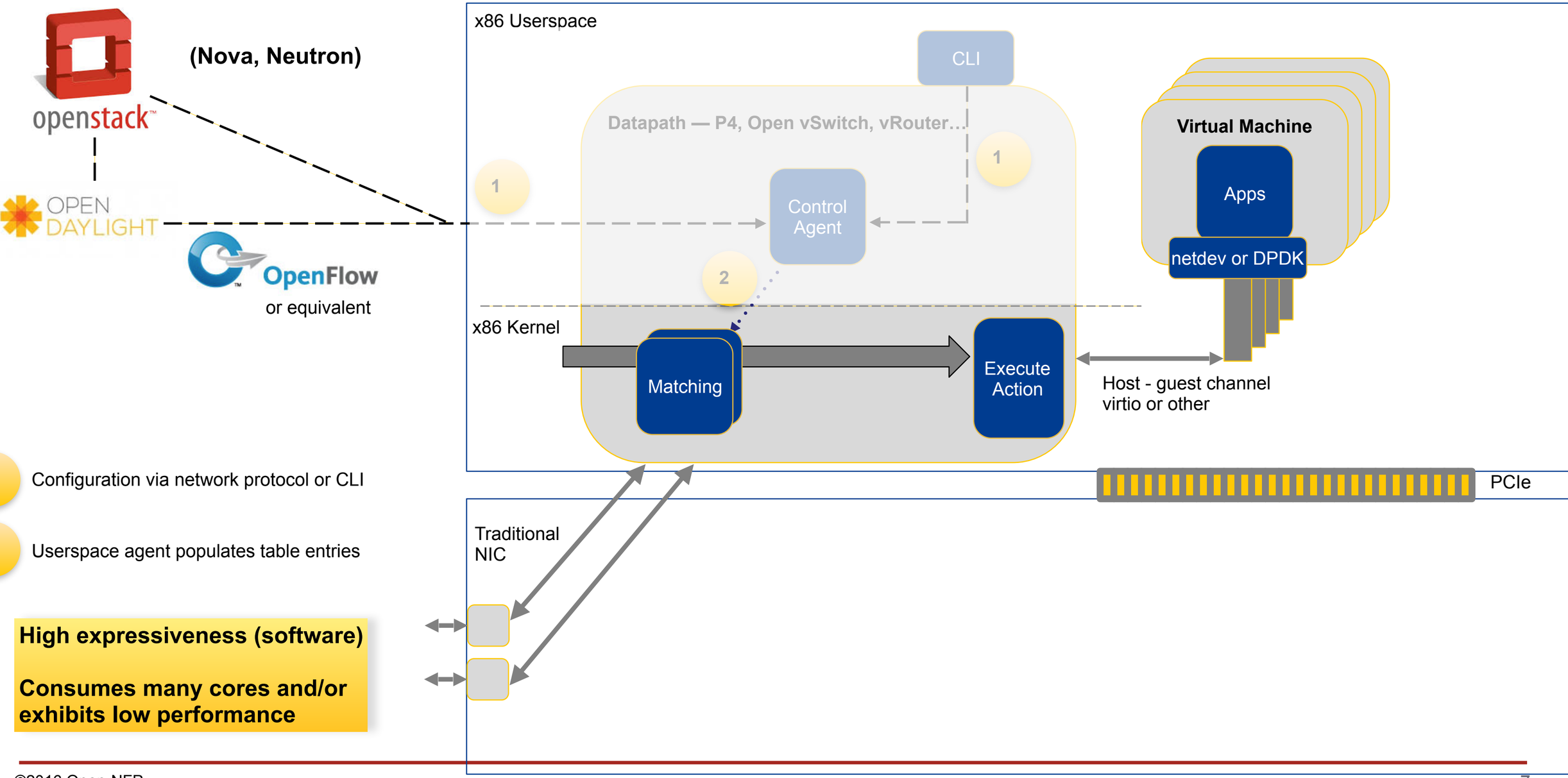
Traditional Model: Host Runs Datapath



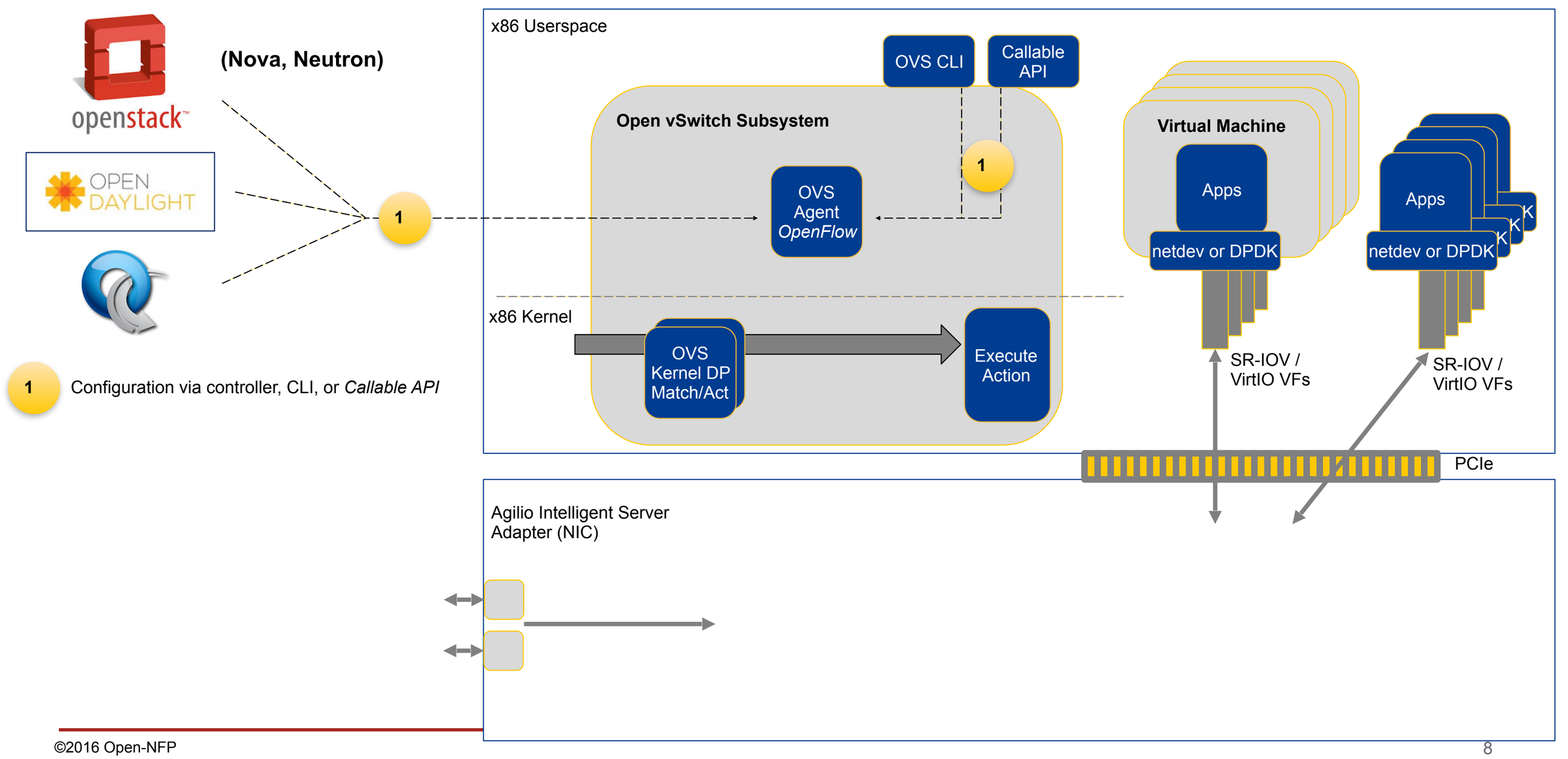
Traditional Model: Host Runs Datapath



Traditional Model: Host Runs Datapath

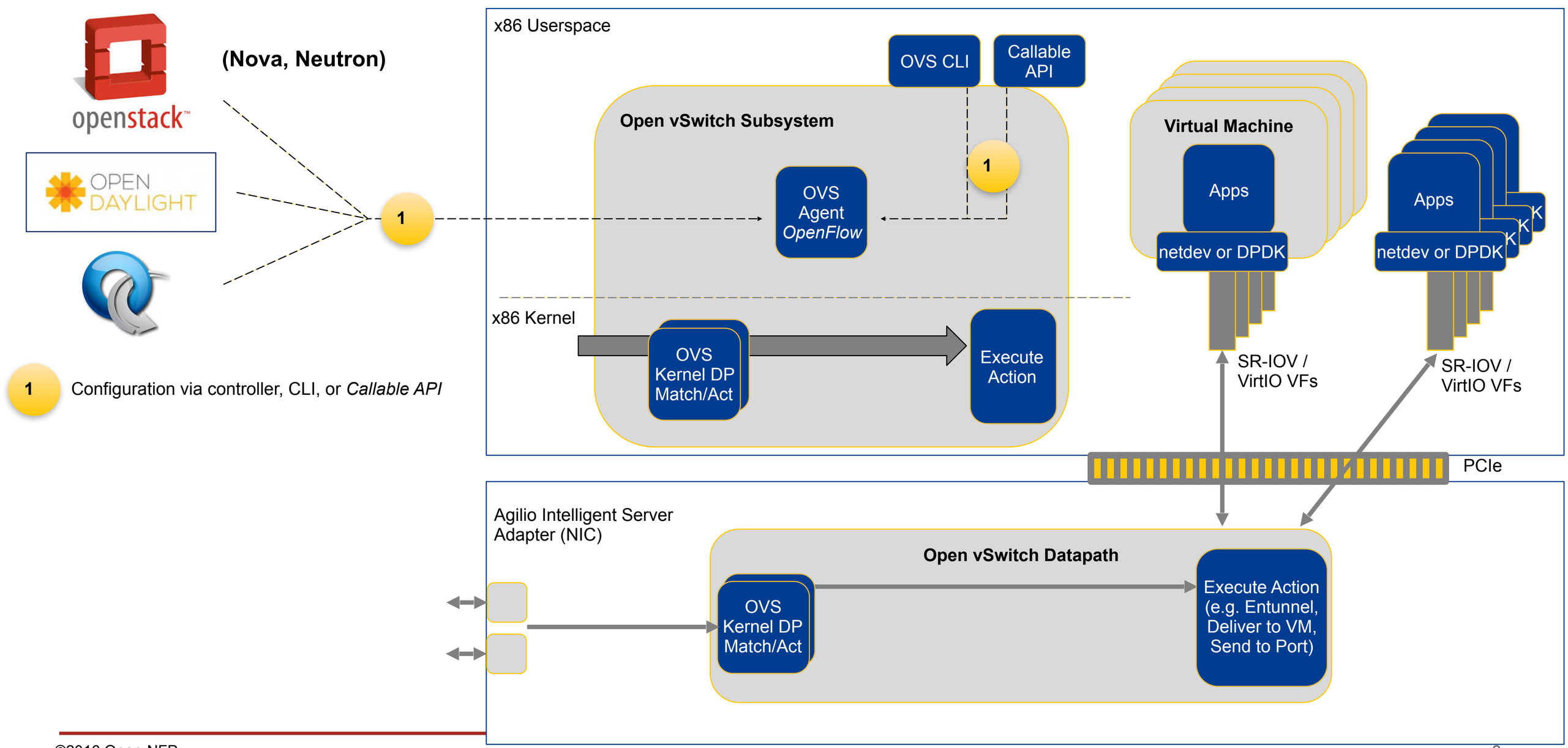


Offload Model: Agilio™ OVS Acceleration



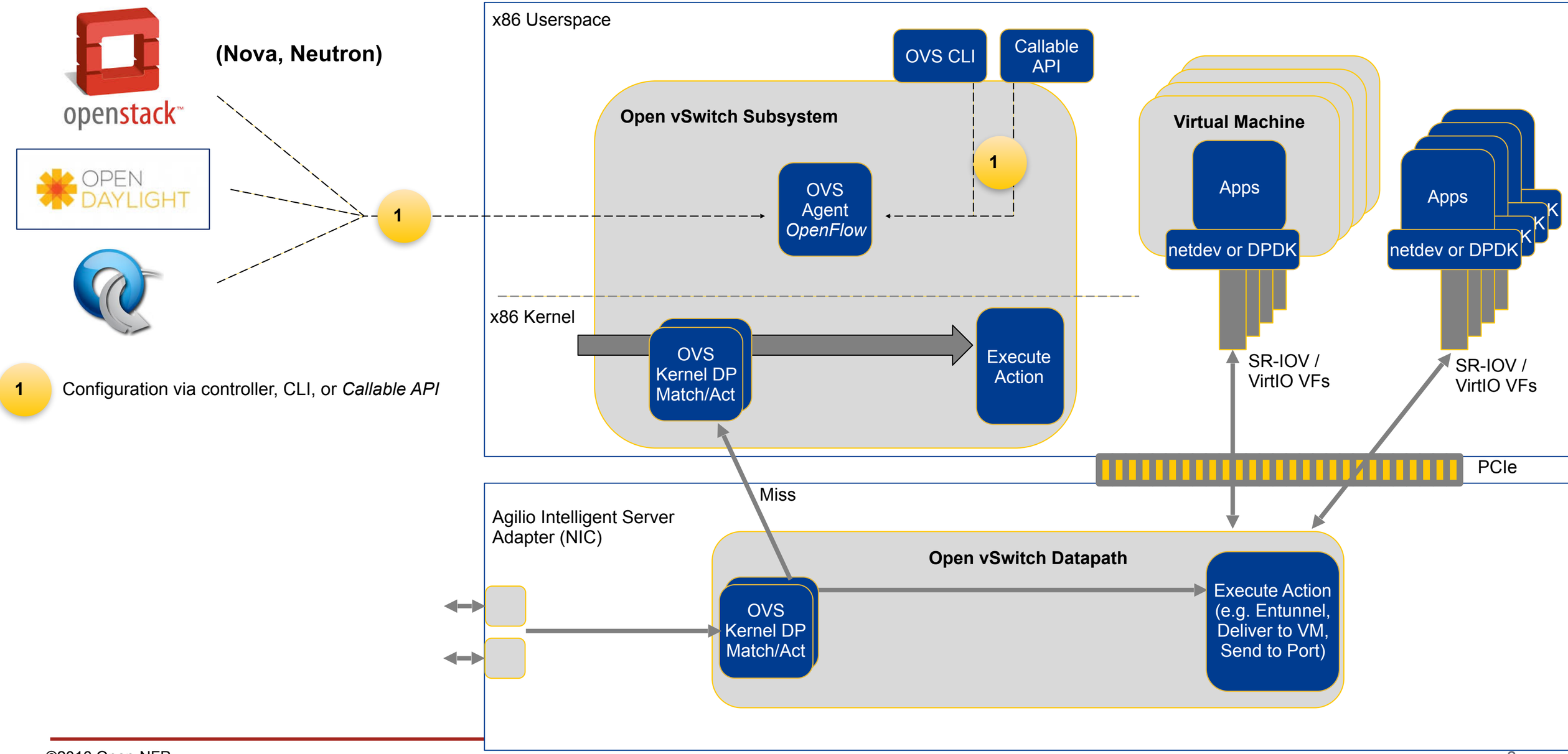
1 Configuration via controller, CLI, or *Callable API*

Offload Model: Agilio™ OVS Acceleration



1 Configuration via controller, CLI, or *Callable API*

Offload Model: Agilio™ OVS Acceleration

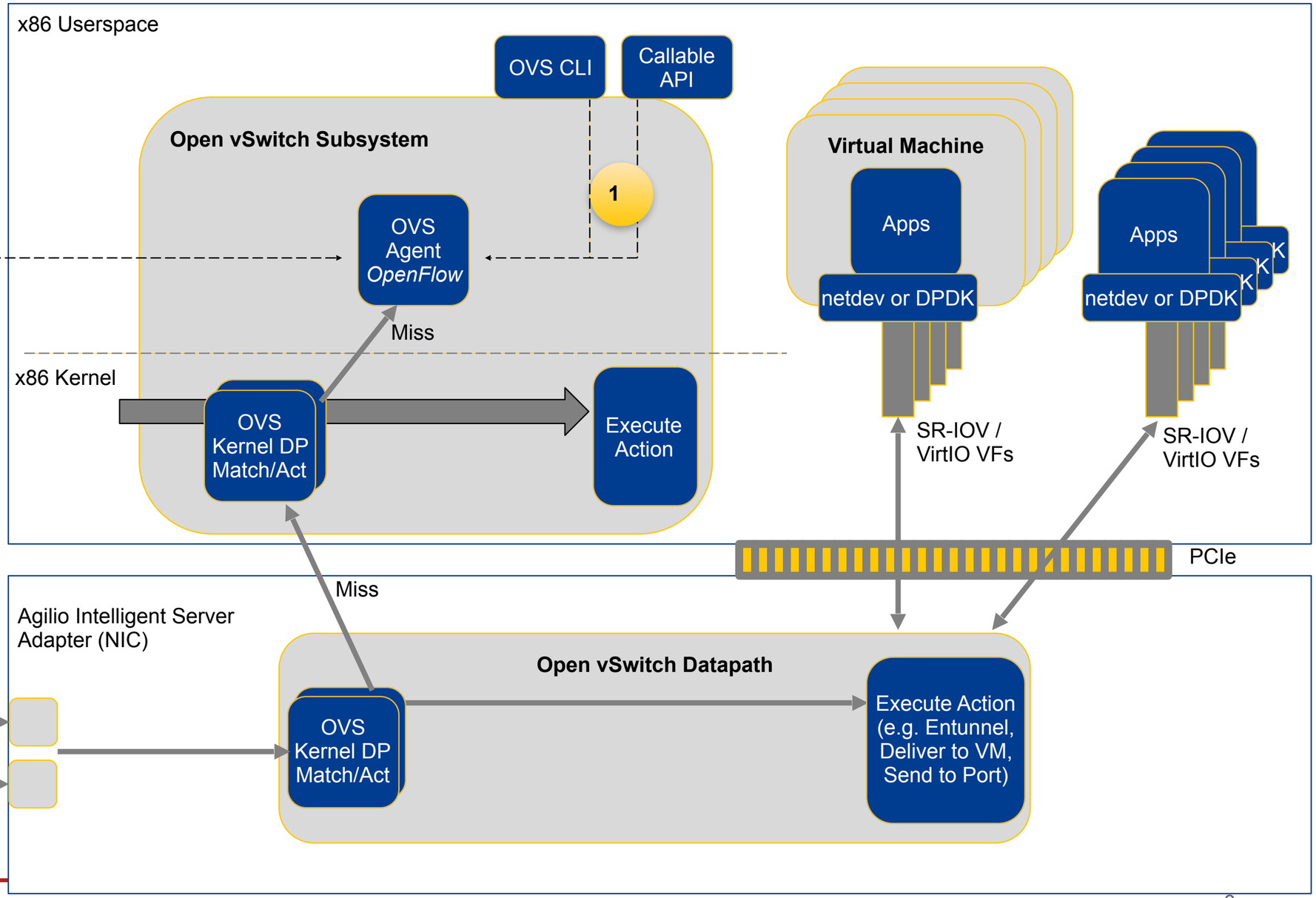


1 Configuration via controller, CLI, or *Callable API*

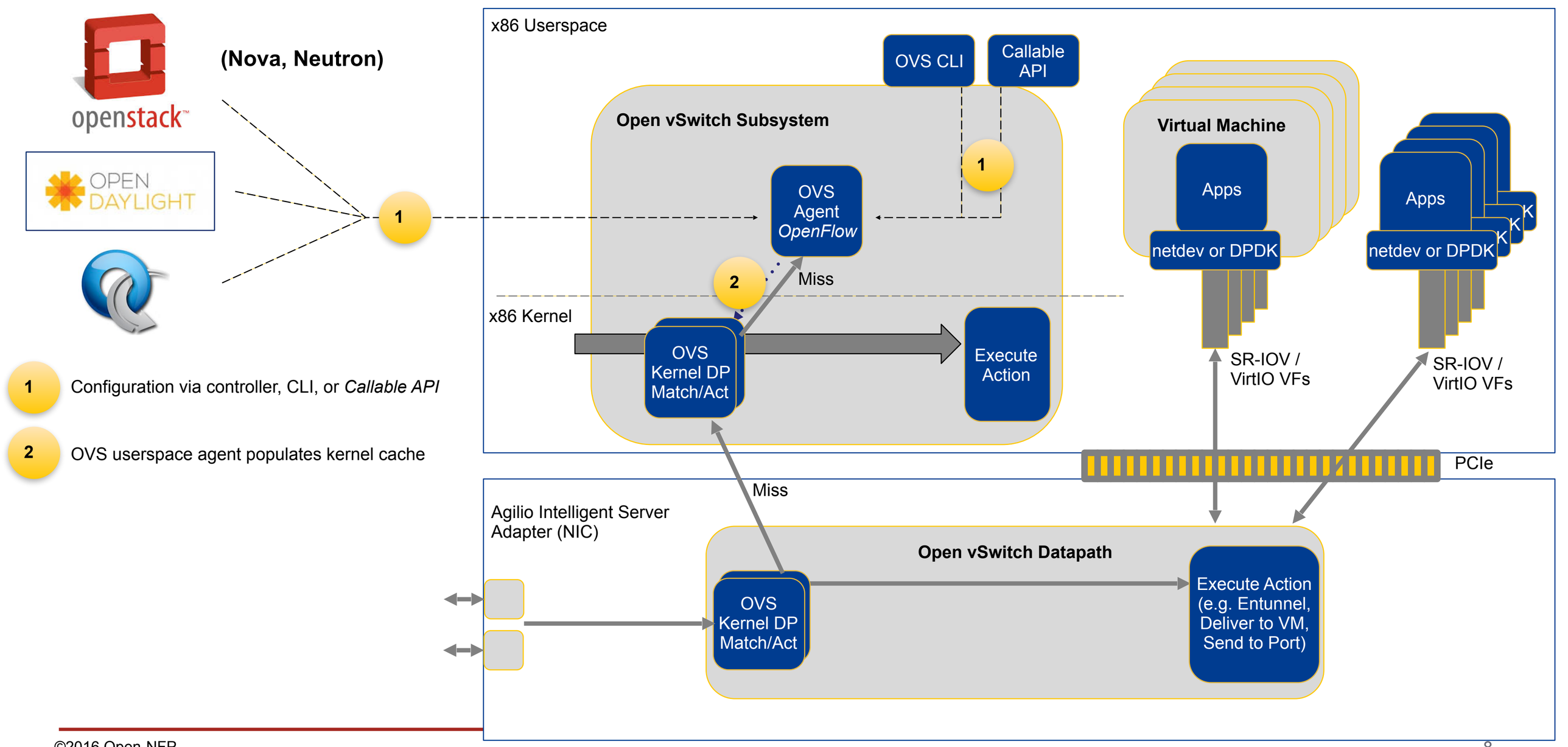
Offload Model: Agilio™ OVS Acceleration

(Nova, Neutron)

1 Configuration via controller, CLI, or *Callable API*



Offload Model: Agilio™ OVS Acceleration



1 Configuration via controller, CLI, or *Callable API*

2 OVS userspace agent populates kernel cache

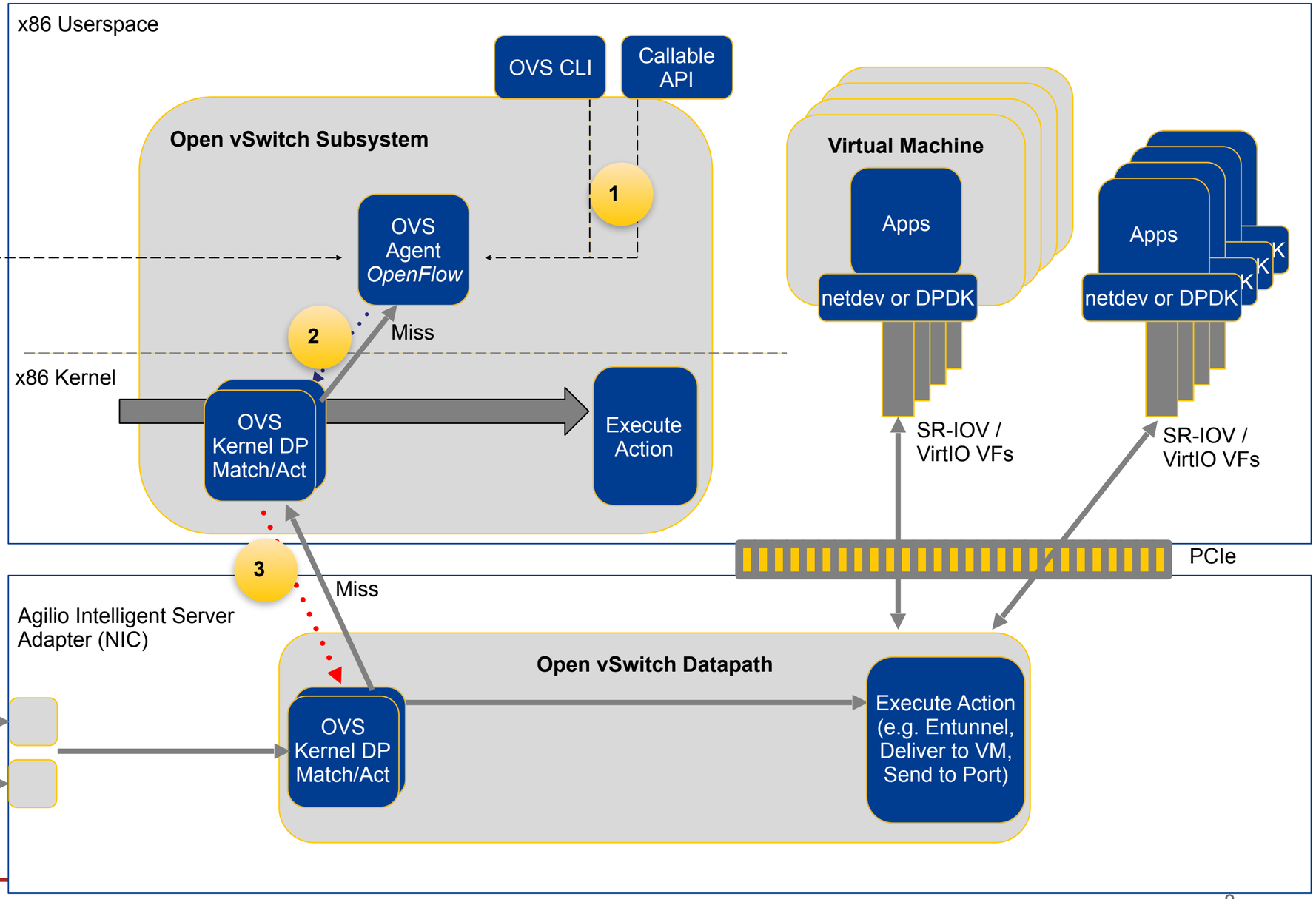
Offload Model: Agilio™ OVS Acceleration

(Nova, Neutron)

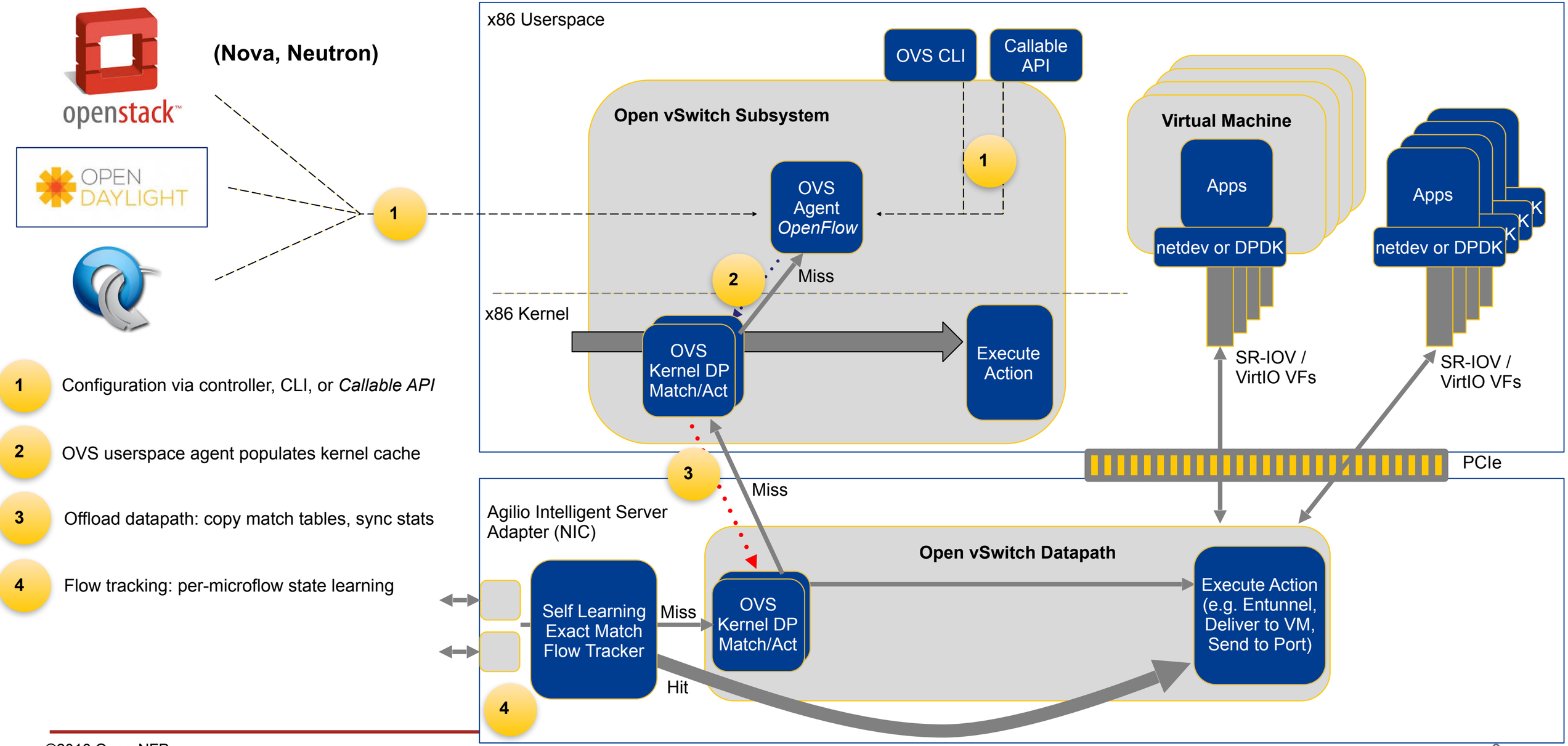
1 Configuration via controller, CLI, or *Callable API*

2 OVS userspace agent populates kernel cache

3 Offload datapath: copy match tables, sync stats

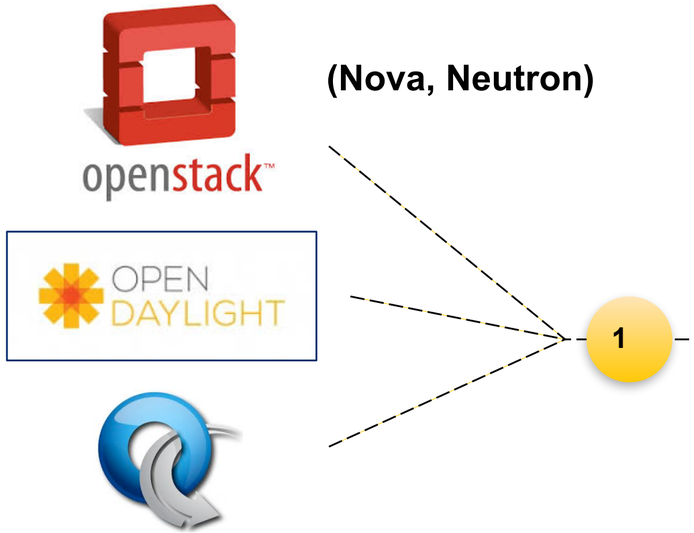


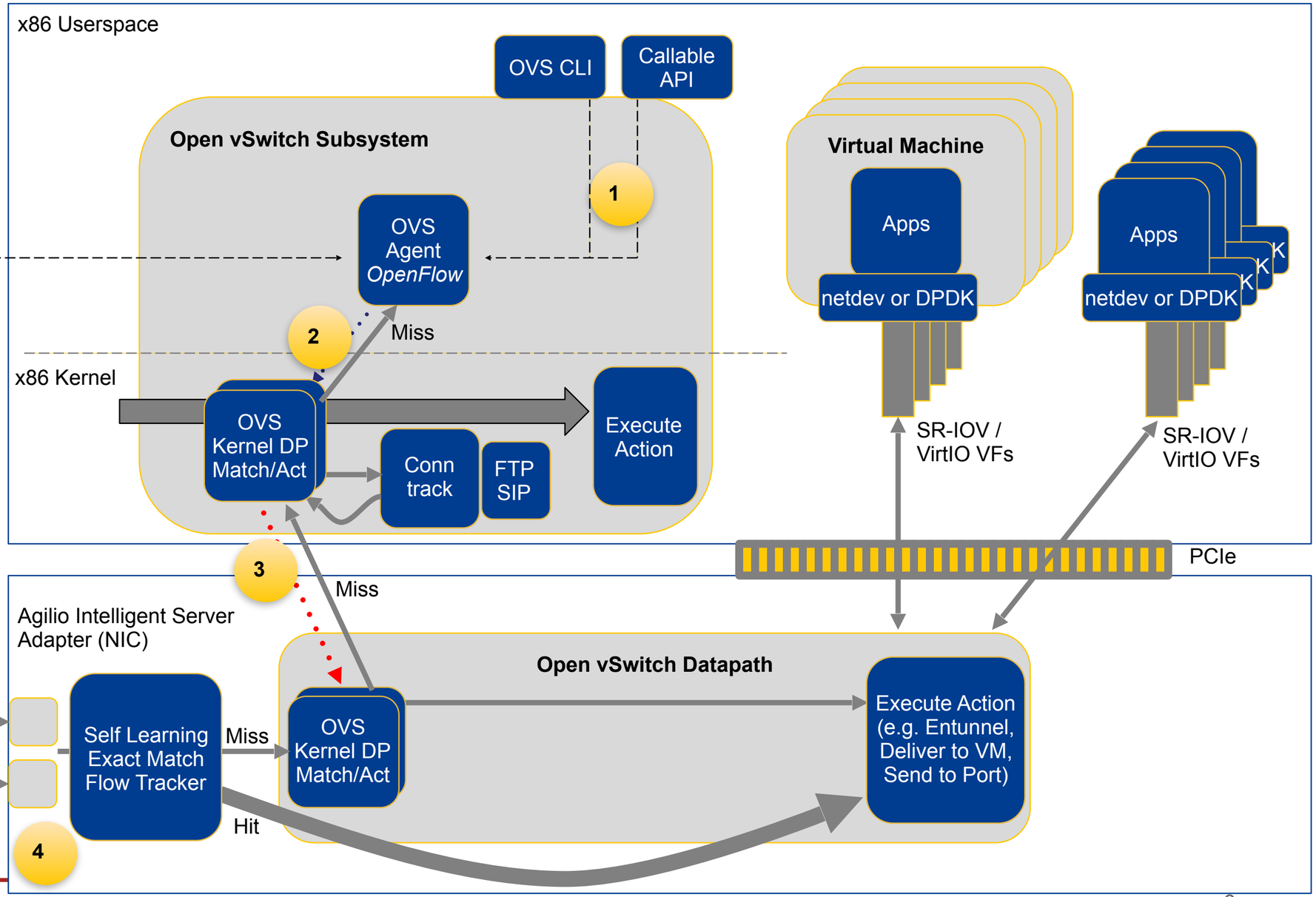
Offload Model: Agilio™ OVS Acceleration



1. Configuration via controller, CLI, or *Callable API*
2. OVS userspace agent populates kernel cache
3. Offload datapath: copy match tables, sync stats
4. Flow tracking: per-microflow state learning

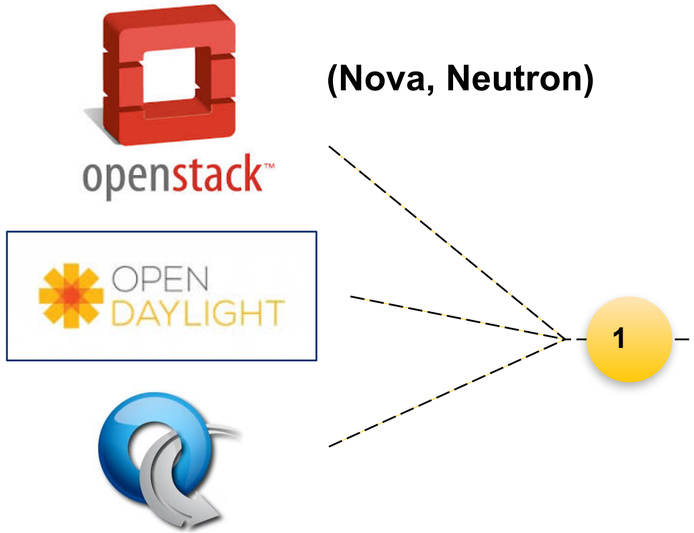
Offload Model: Agilio™ OVS Acceleration



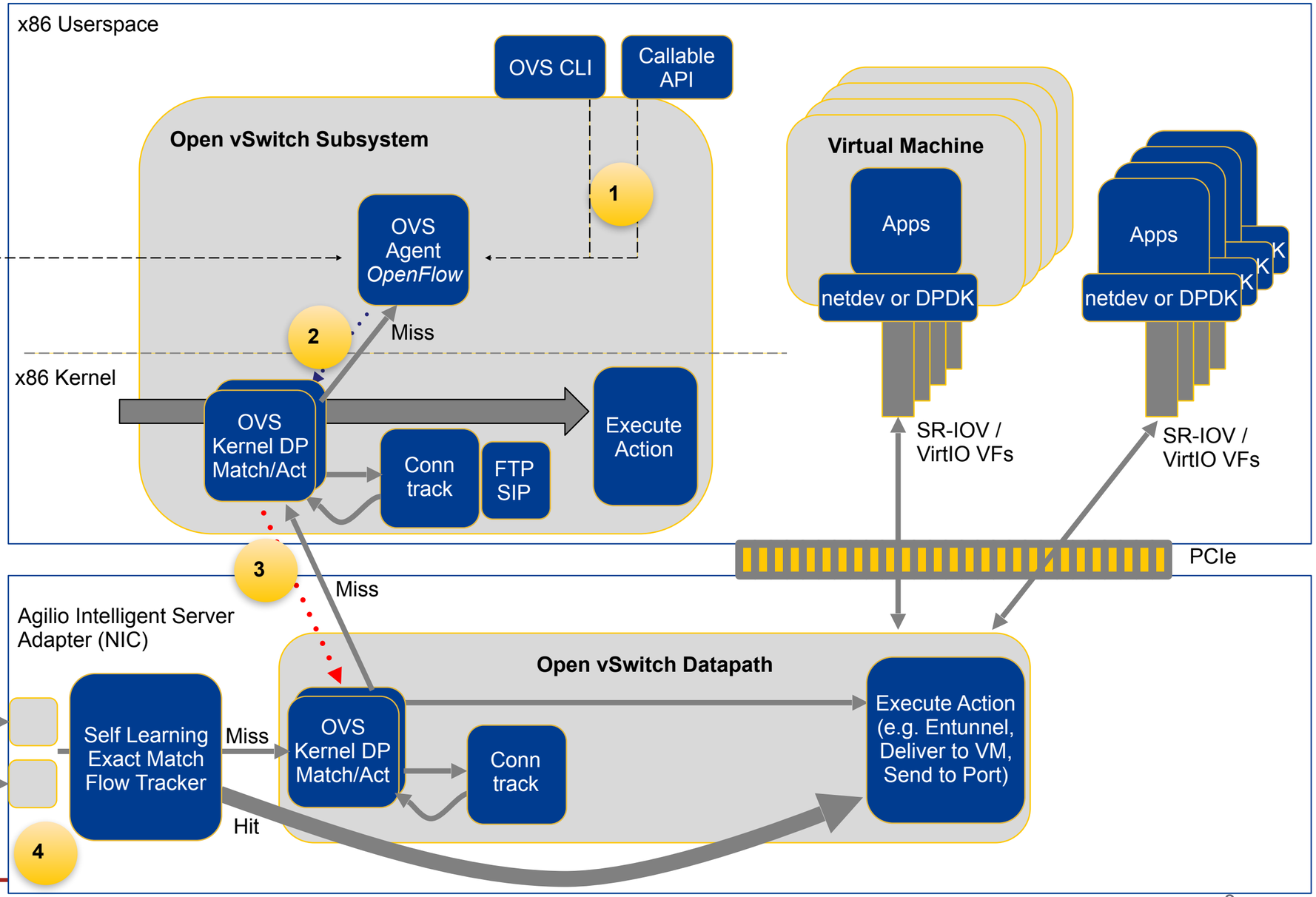


1. Configuration via controller, CLI, or *Callable API*
2. OVS userspace agent populates kernel cache
3. Offload datapath: copy match tables, sync stats
4. Flow tracking: per-microflow state learning

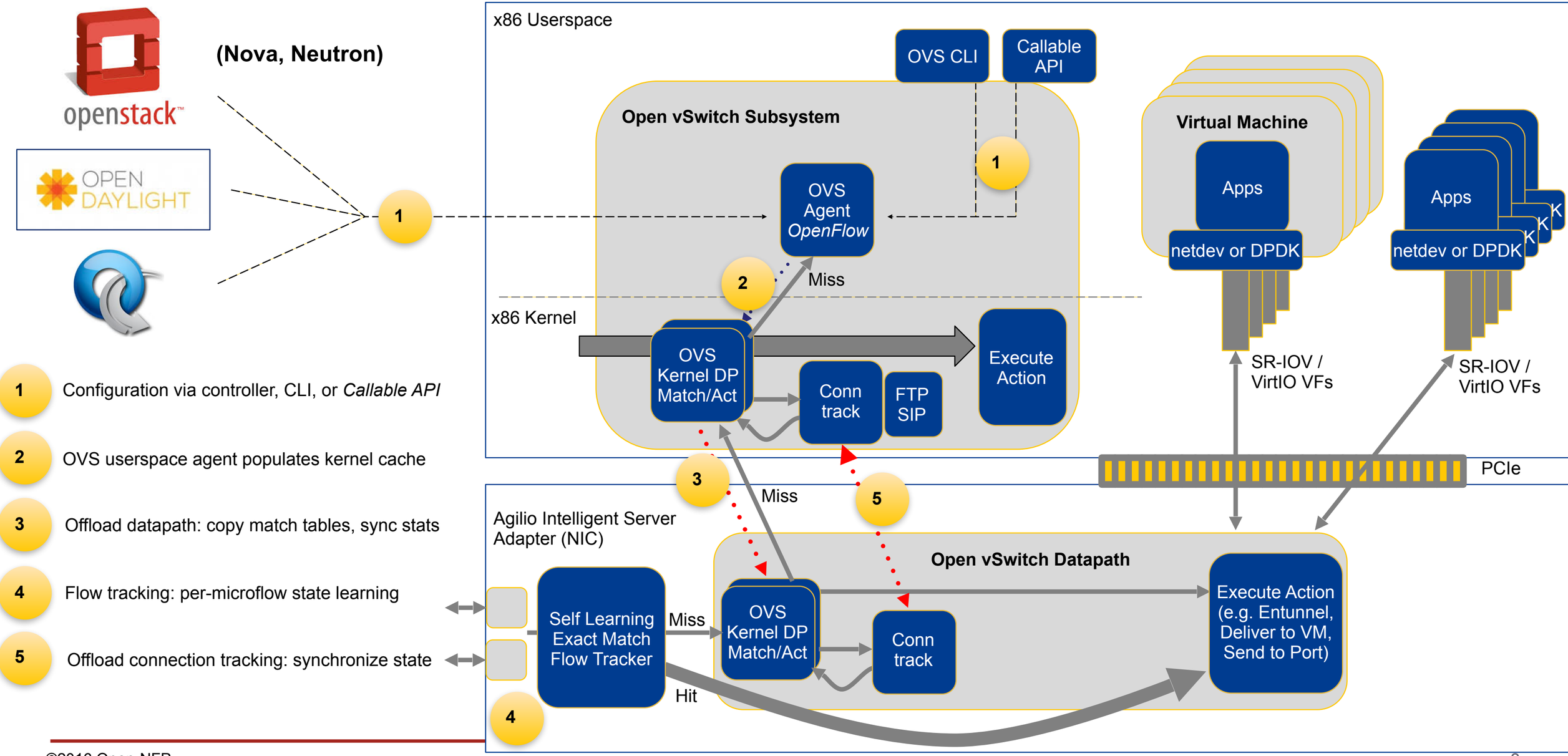
Offload Model: Agilio™ OVS Acceleration



1. Configuration via controller, CLI, or *Callable API*
2. OVS userspace agent populates kernel cache
3. Offload datapath: copy match tables, sync stats
4. Flow tracking: per-microflow state learning

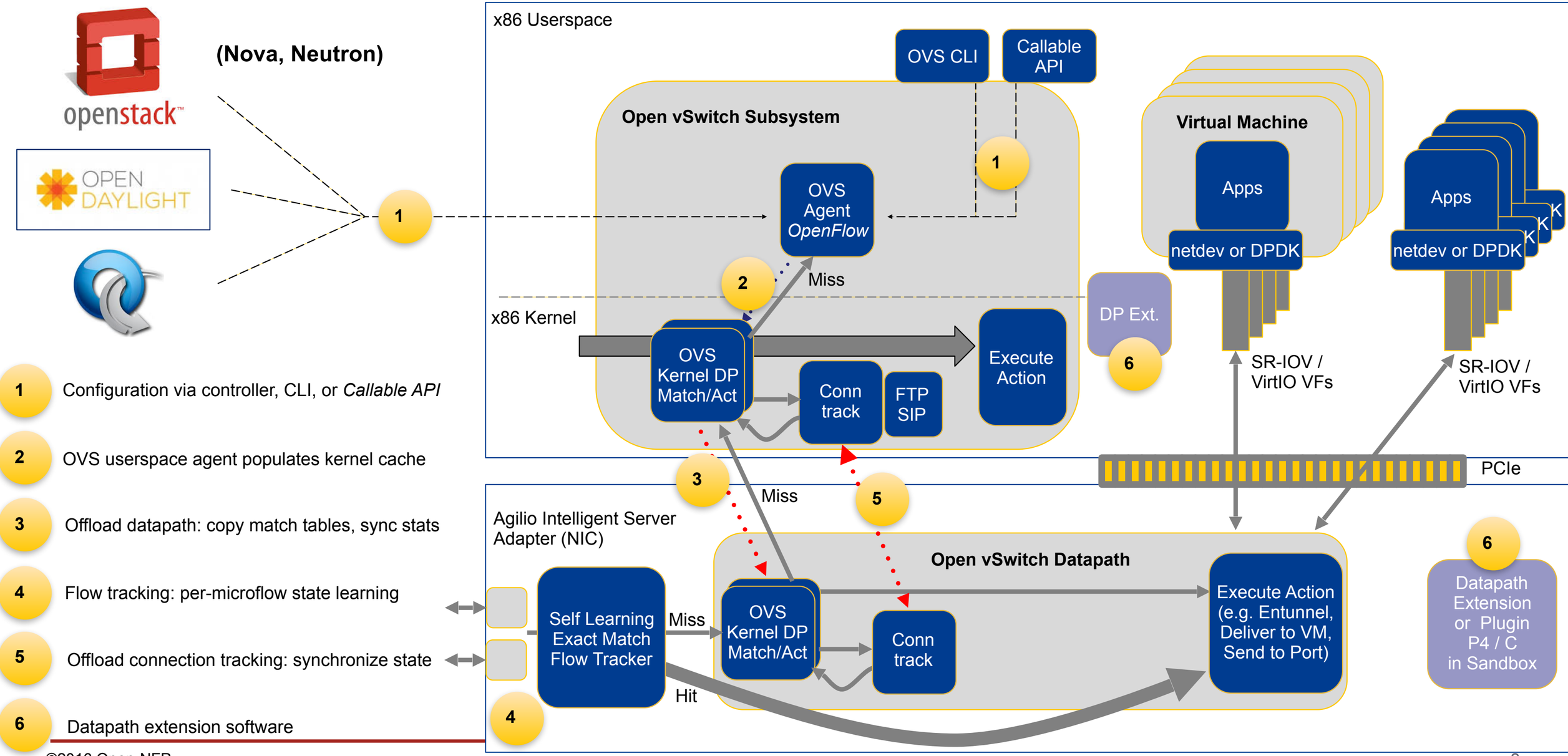


Offload Model: Agilio™ OVS Acceleration



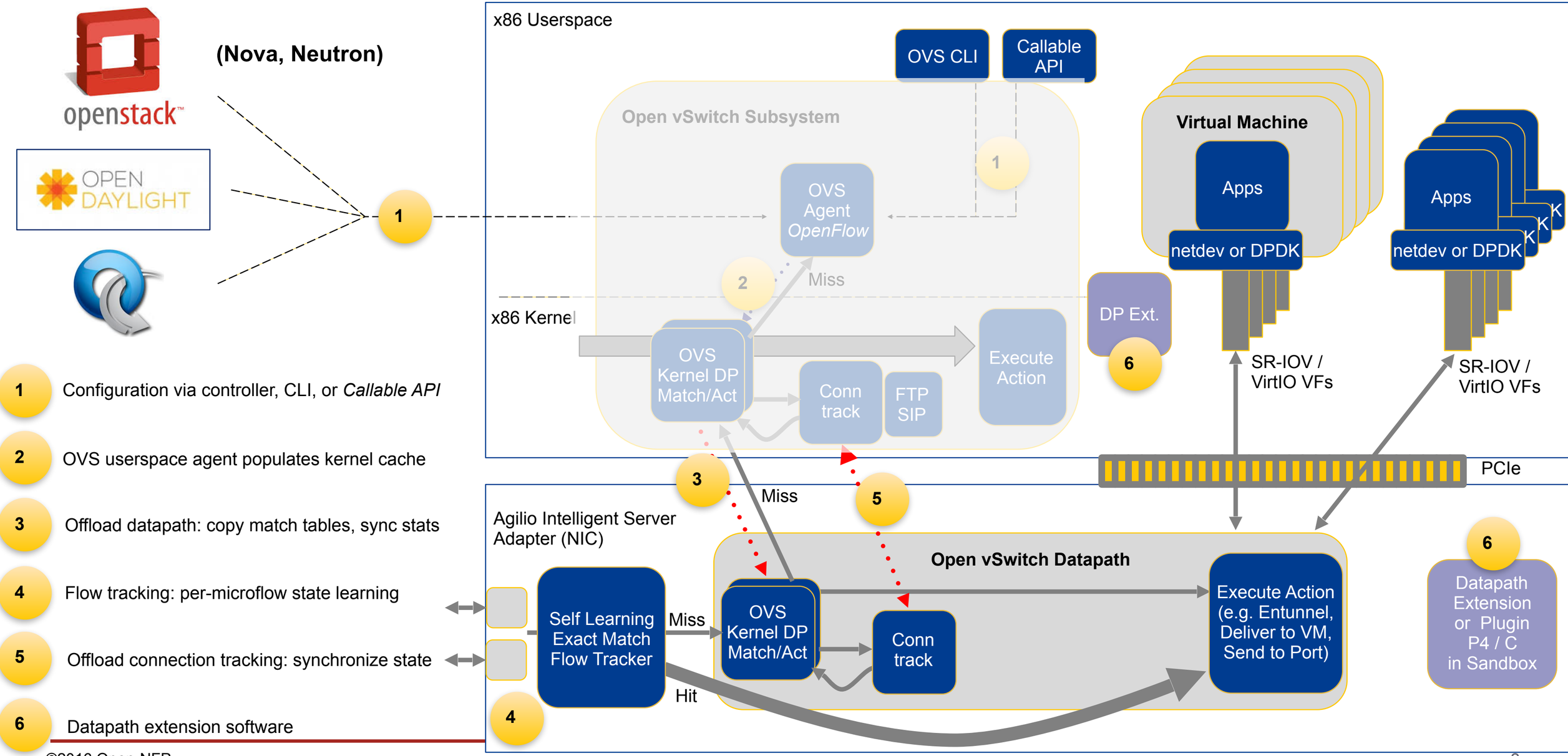
1. Configuration via controller, CLI, or *Callable API*
2. OVS userspace agent populates kernel cache
3. Offload datapath: copy match tables, sync stats
4. Flow tracking: per-microflow state learning
5. Offload connection tracking: synchronize state

Offload Model: Agilio™ OVS Acceleration



1. Configuration via controller, CLI, or *Callable API*
2. OVS userspace agent populates kernel cache
3. Offload datapath: copy match tables, sync stats
4. Flow tracking: per-microflow state learning
5. Offload connection tracking: synchronize state
6. Datapath extension software

Offload Model: Agilio™ OVS Acceleration

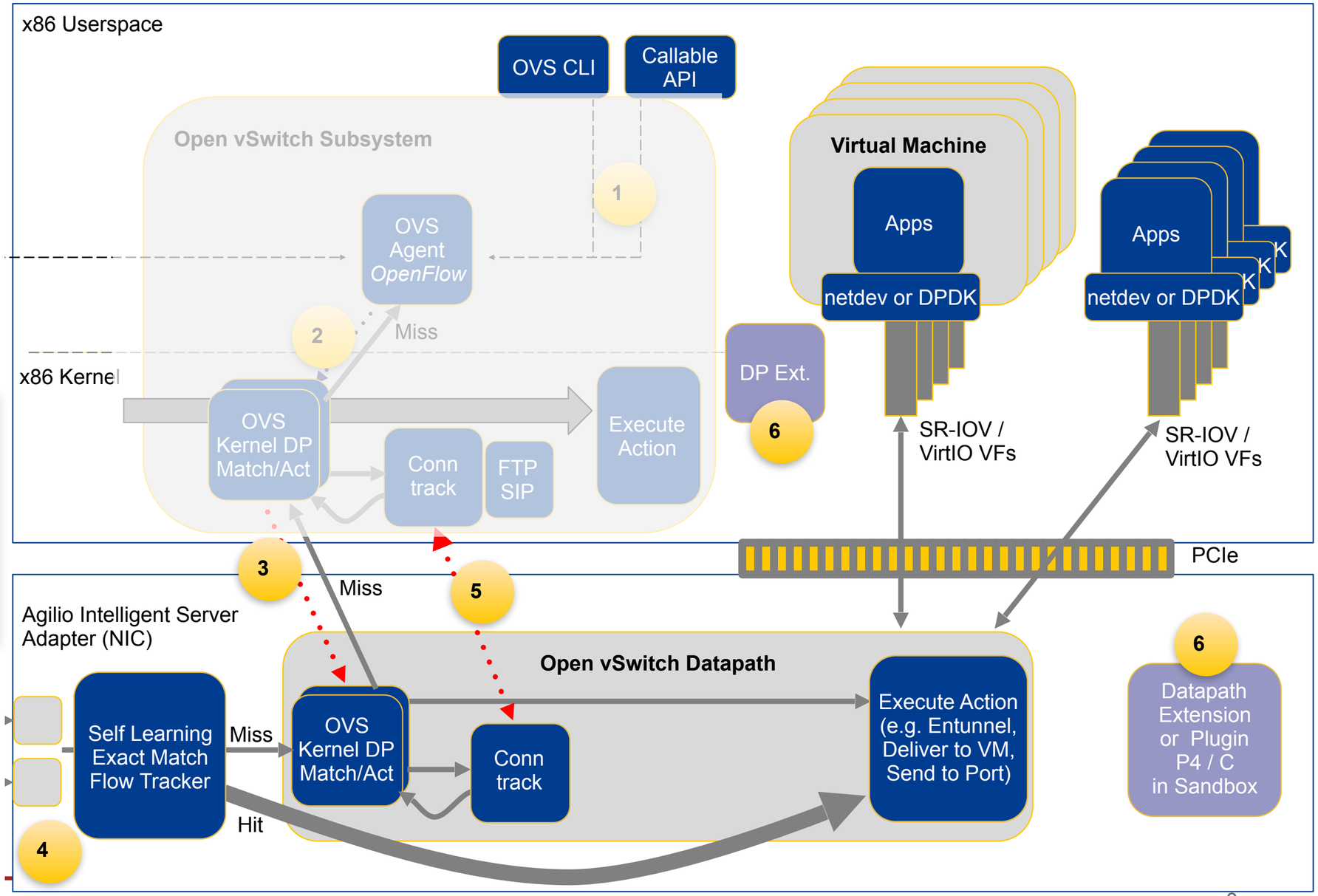


1. Configuration via controller, CLI, or *Callable API*
2. OVS userspace agent populates kernel cache
3. Offload datapath: copy match tables, sync stats
4. Flow tracking: per-microflow state learning
5. Offload connection tracking: synchronize state
6. Datapath extension software

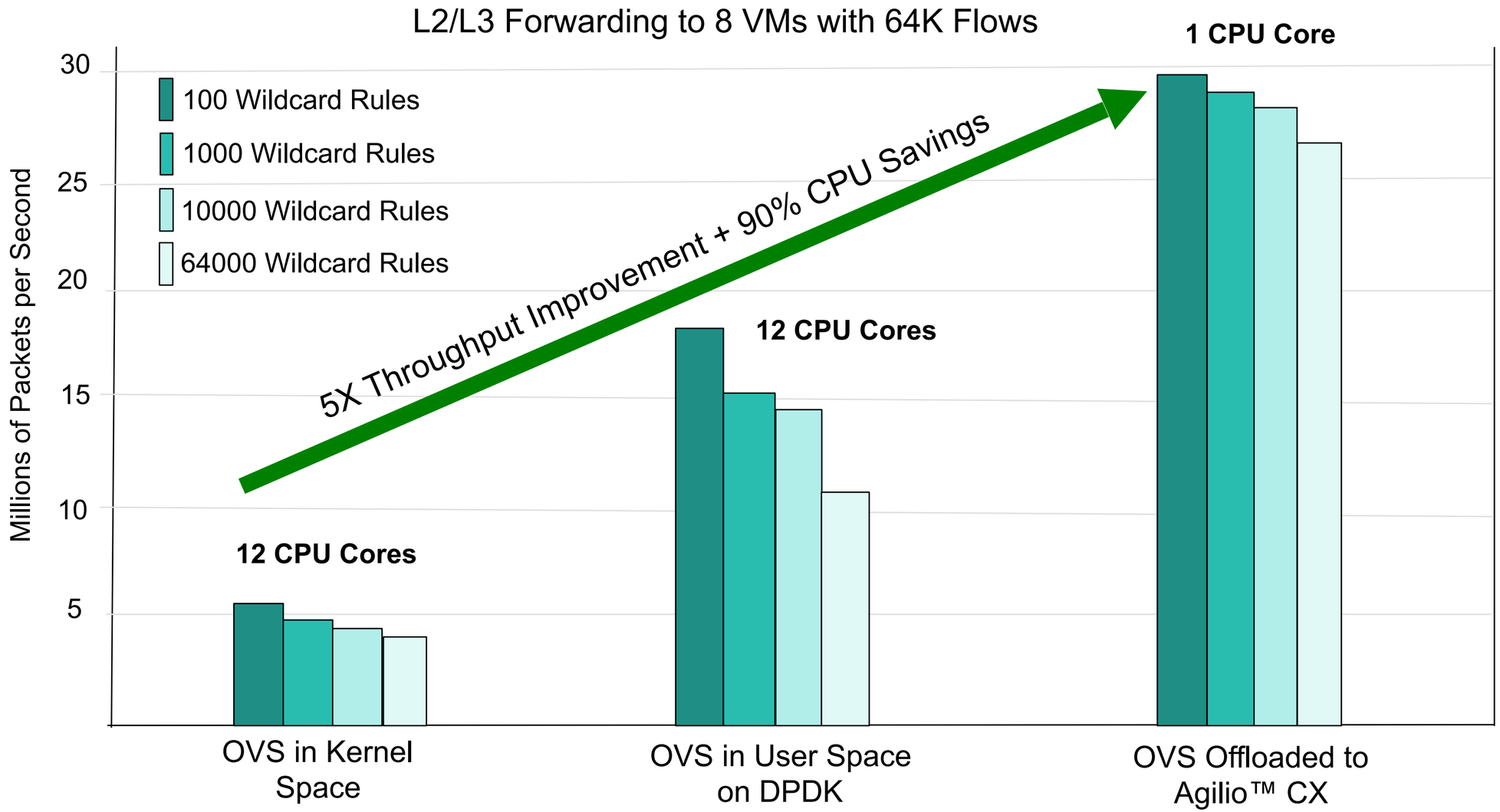
Offload Model: Agilio™ OVS Acceleration

Best of all worlds

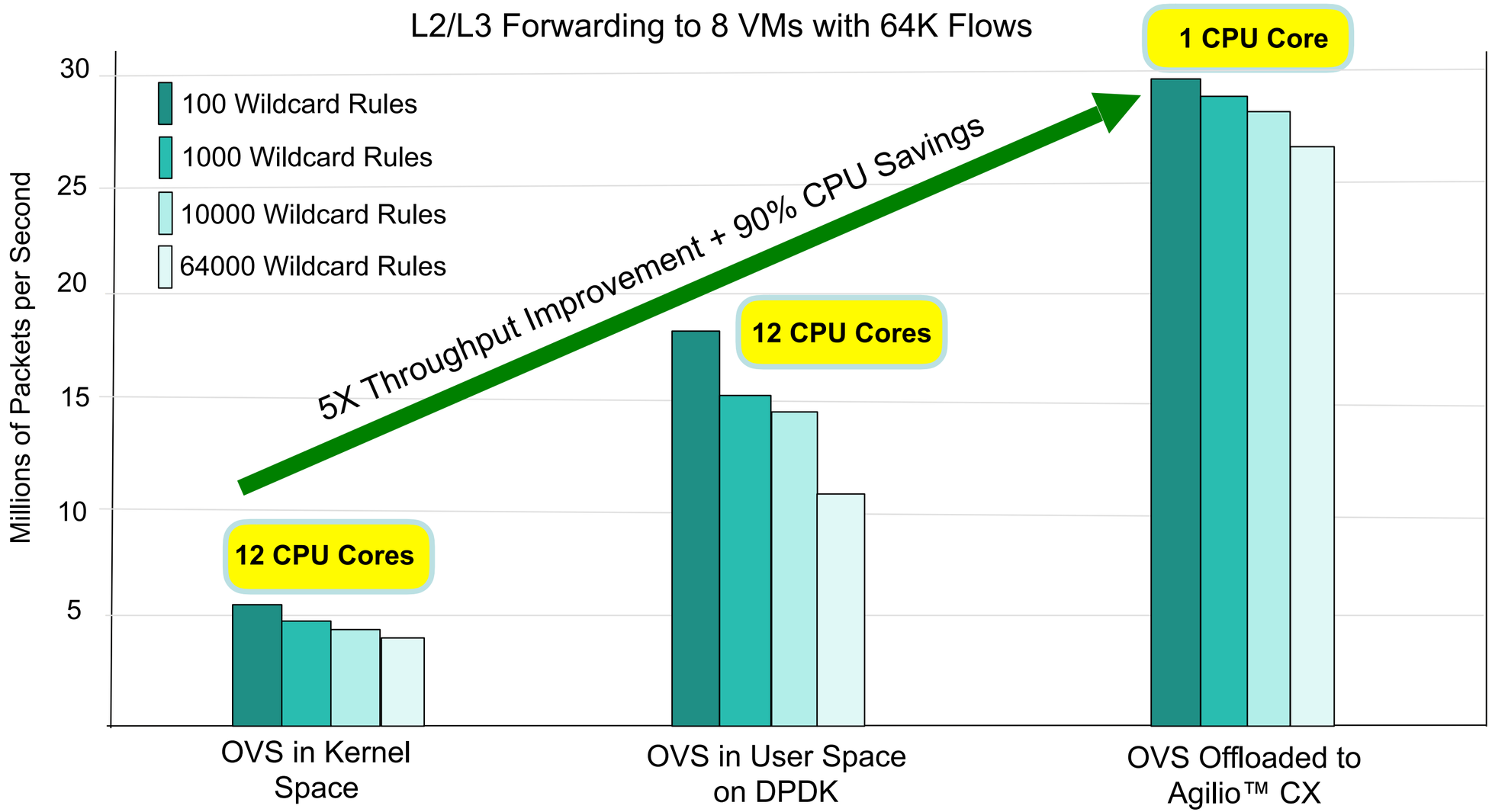
- Performance of SR-IOV
- Flexibility of virtio (VM migration)
- Performance and CPU core saving of switching on SmartNIC



Example: Throughput vs. Number of Rules

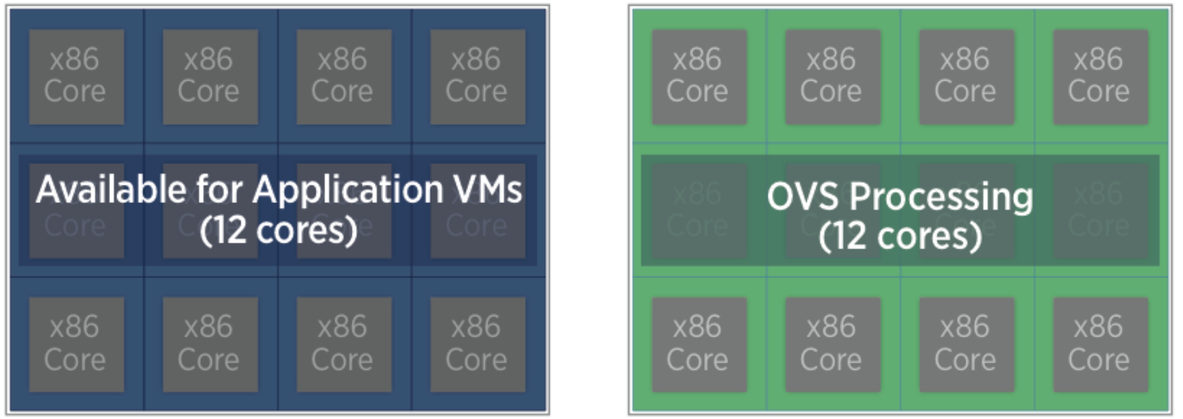


Example: Throughput vs. Number of Rules

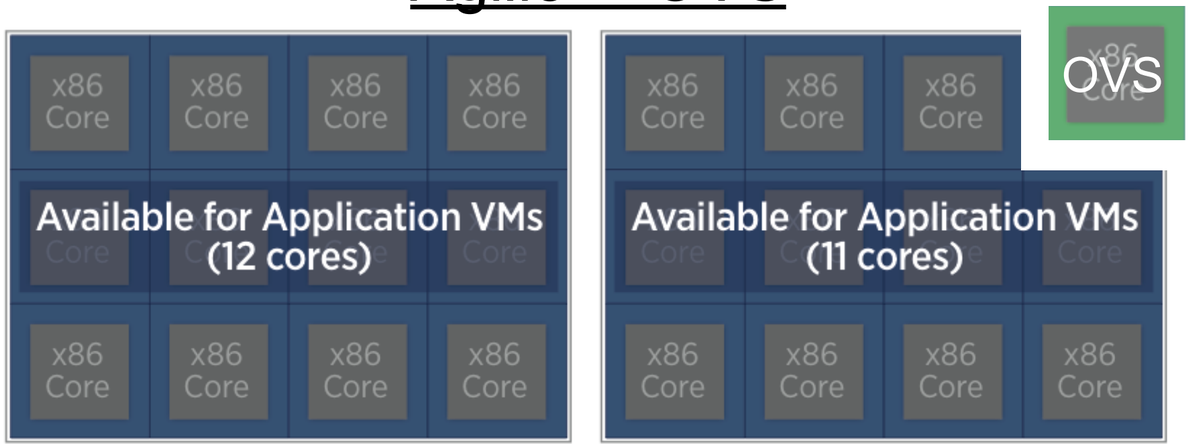


Tested Scenario: Server CPU Core Allocation

Unaccelerated OVS (Kernel / User Mode)

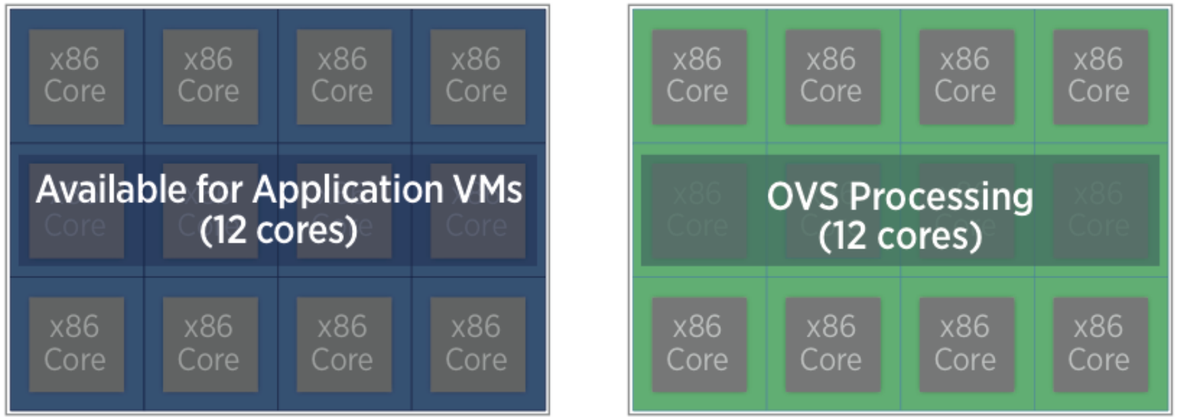


Agilio™ OVS

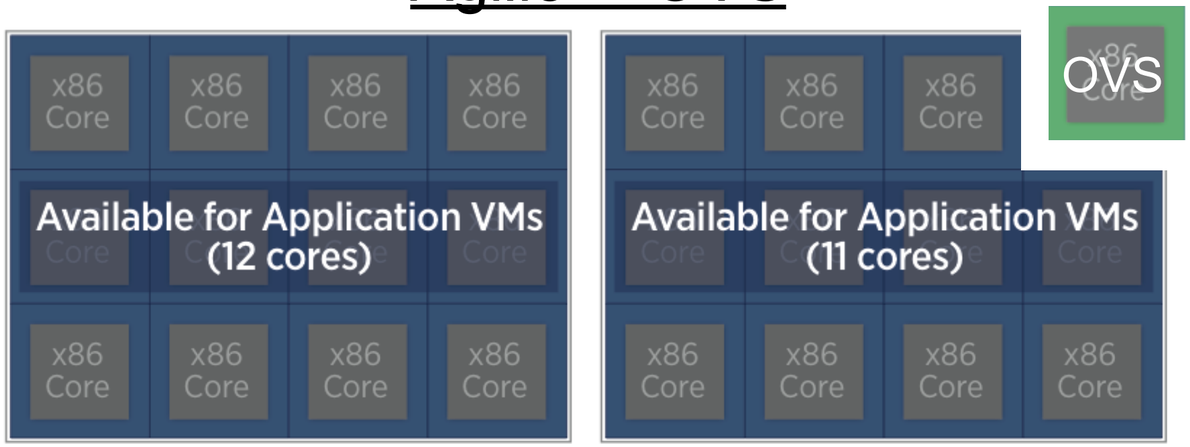


Tested Scenario: Server CPU Core Allocation

Unaccelerated OVS (Kernel / User Mode)



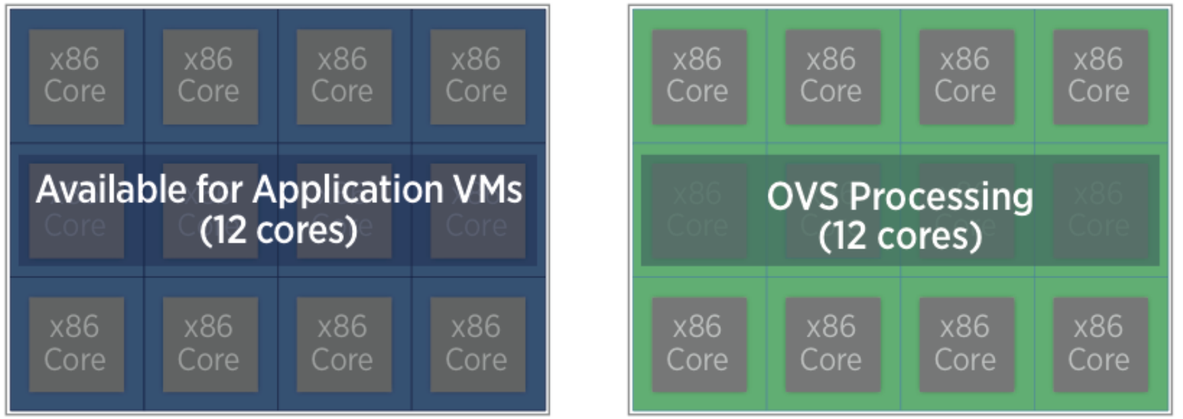
Agilio™ OVS



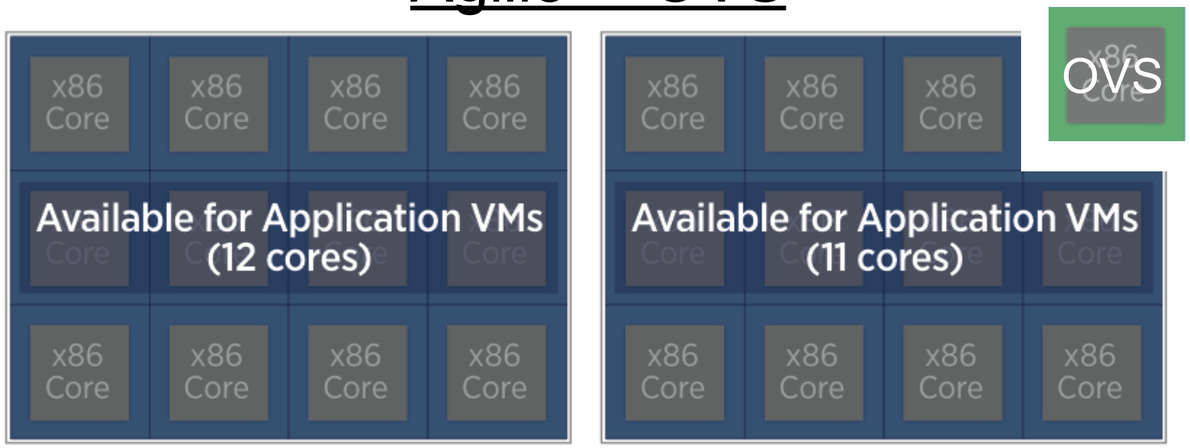
Typical result: replace 3-6 racks with 1 rack!

Tested Scenario: Server CPU Core Allocation

Unaccelerated OVS (Kernel / User Mode)



Agilio™ OVS

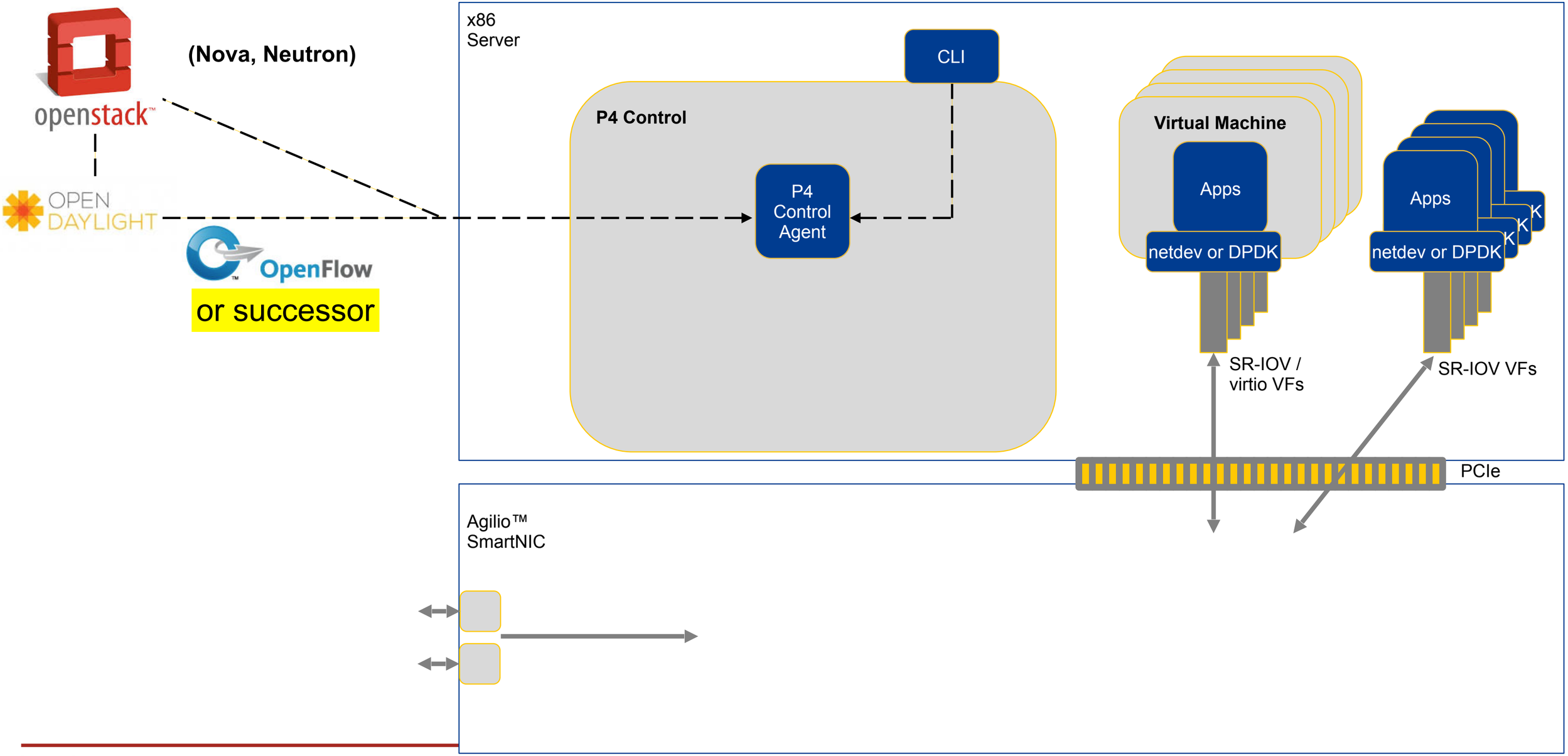


Benefits for your use case:

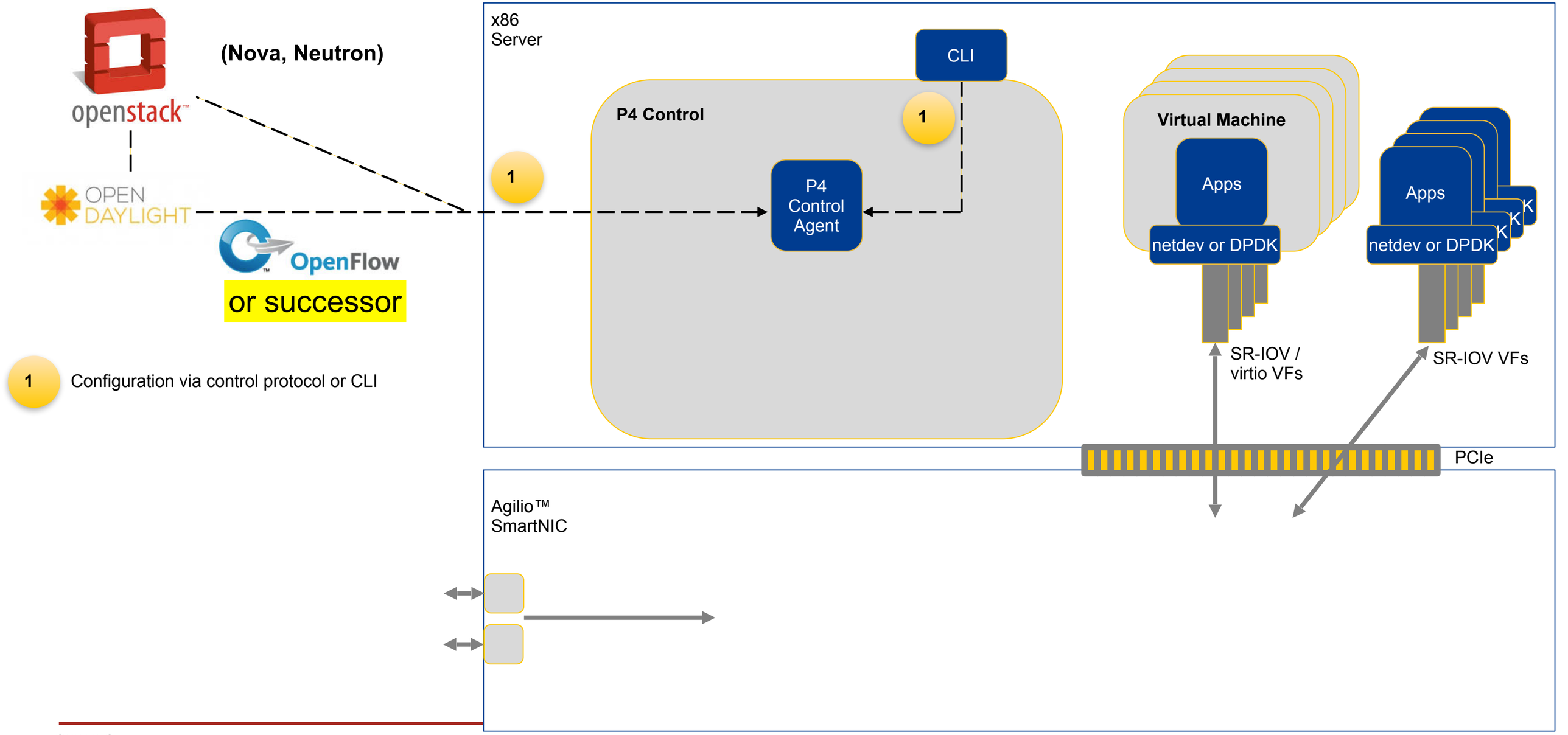
https://www.netronome.com/products/ovs/roi-calculator/

Typical result: replace 3-6 racks with 1 rack!

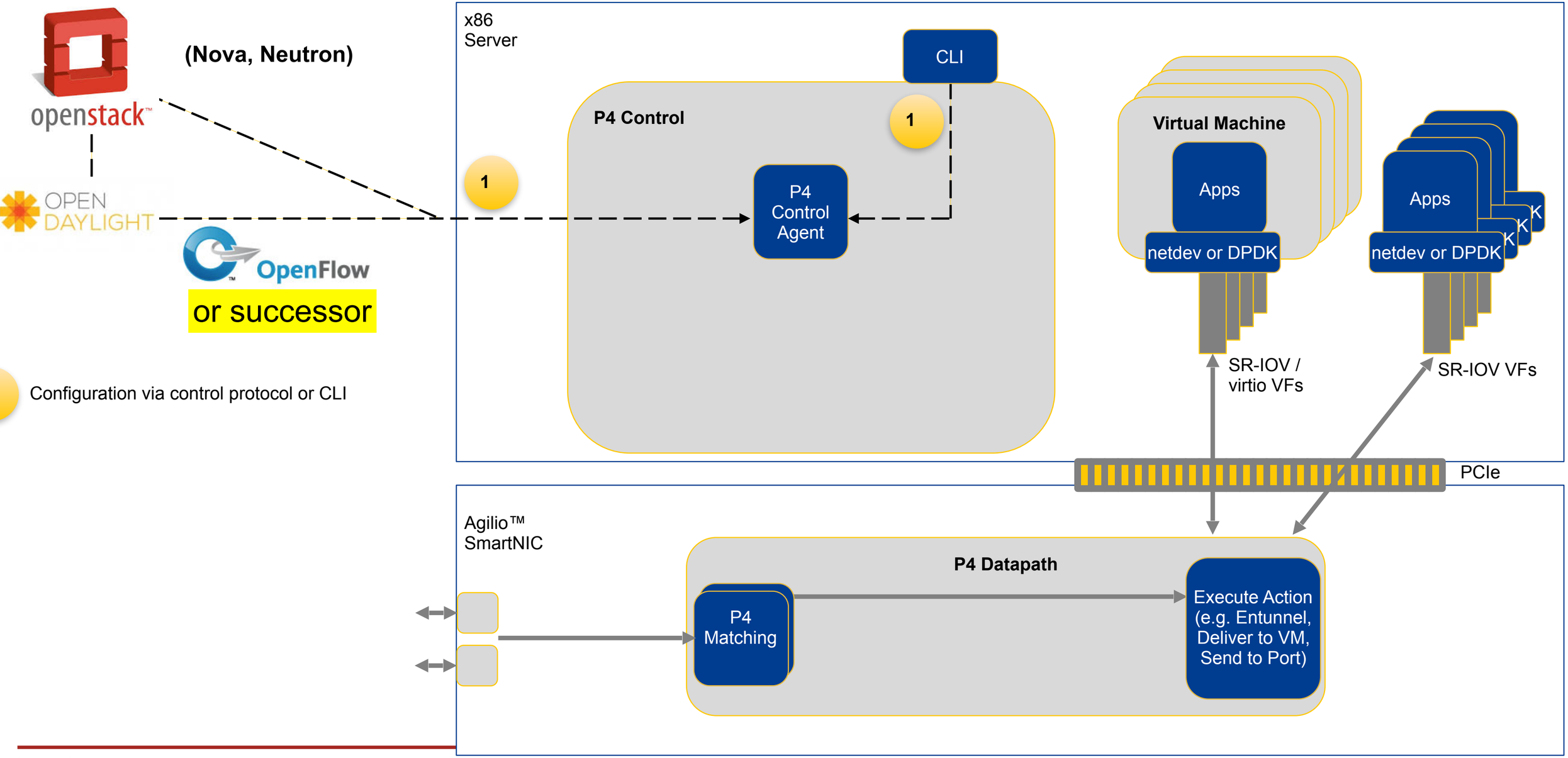
P4 Datapath on SmartNIC



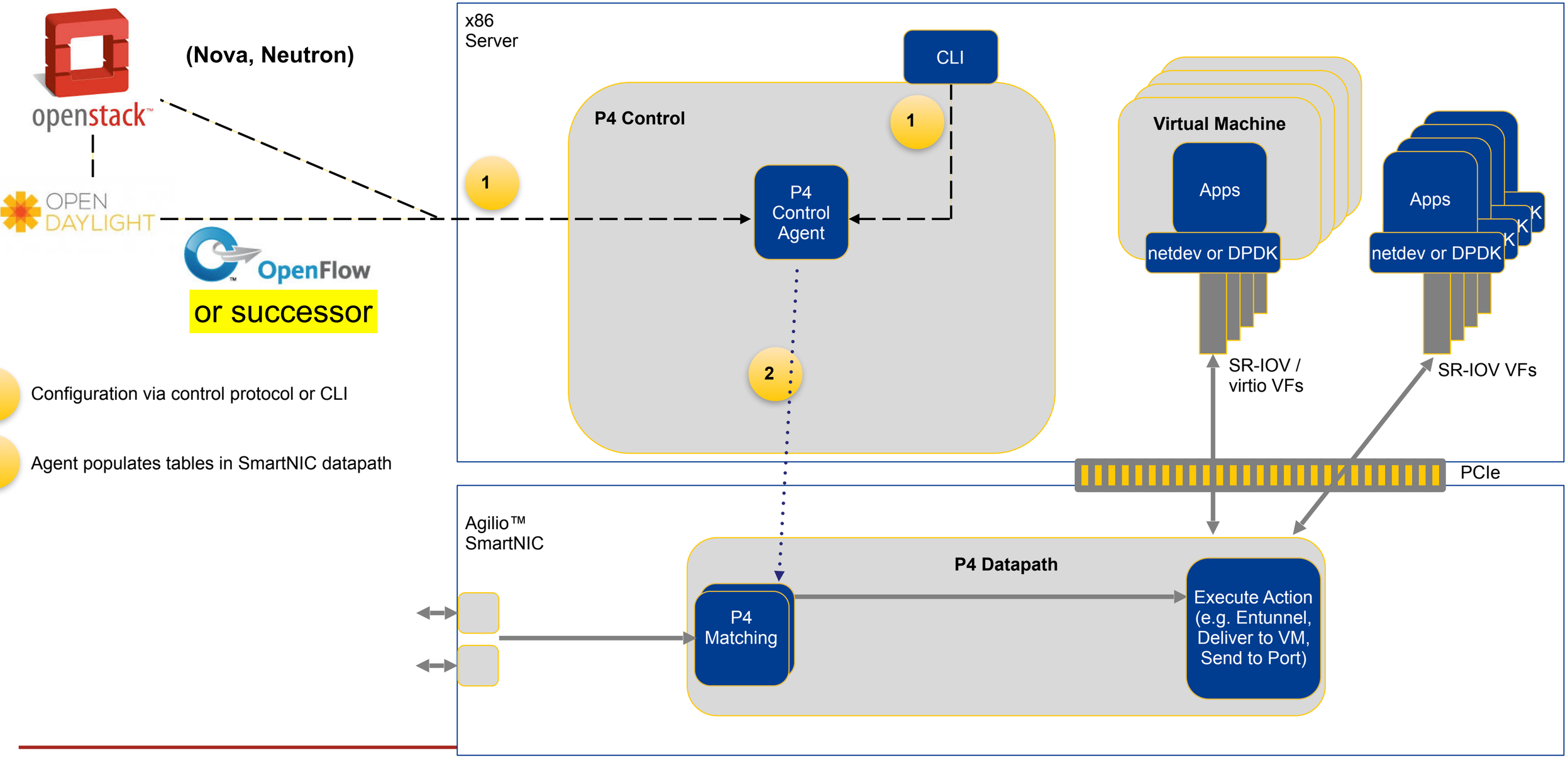
P4 Datapath on SmartNIC



P4 Datapath on SmartNIC



P4 Datapath on SmartNIC



1 Configuration via control protocol or CLI

2 Agent populates tables in SmartNIC datapath

Traditional vs. SmartNIC Accelerated Networking

Forwarding / Virtual Switching Technology
SR-IOV
Intelligent Datapath *P4 / vRouter / Open vSwitch…*

Traditional vs. SmartNIC Accelerated Networking

Forwarding / Virtual Switching Technology	Traditional Approach (Unaccelerated)
SR-IOV	**Limited expressiveness** to direct traffic to VMs (no support for general match/action rules, tunnel termination, stateful firewalling) **High throughput** **No VM migration support** (difficult to manage)
Intelligent Datapath ***P4 / vRouter / Open vSwitch…***	**High expressiveness -** match/action, tunnels, stateless/stateful firewalling etc. **Limited throughput** **High CPU utilization** (e.g. 50% of cores)

Traditional vs. SmartNIC Accelerated Networking

Forwarding / Virtual Switching Technology	Traditional Approach (Unaccelerated)	Fully Programmable SmartNIC Accelerated Approach
SR-IOV	**Limited expressiveness** to direct traffic to VMs (no support for general match/action rules, tunnel termination, stateful firewalling) **High throughput** **No VM migration support** (difficult to manage)	**High expressiveness** - match/action, tunnels, stateless/stateful firewalling etc. ***and* SR-IOV based data delivery to VMs** **High throughput** **Virtio** supporting VM migration (facilitating cloud optimization and upgrading)
Intelligent Datapath *P4 / vRouter / Open vSwitch…*	**High expressiveness** - match/action, tunnels, stateless/stateful firewalling etc. **Limited throughput** **High CPU utilization** (e.g. 50% of cores)	**Higher throughput** (~5x higher) **Lower CPU utilization** (~10x lower)

Flow Processor Silicon Architecture

Network Flow Processor 4xxx (used on Agilio-CX SmartNICs)

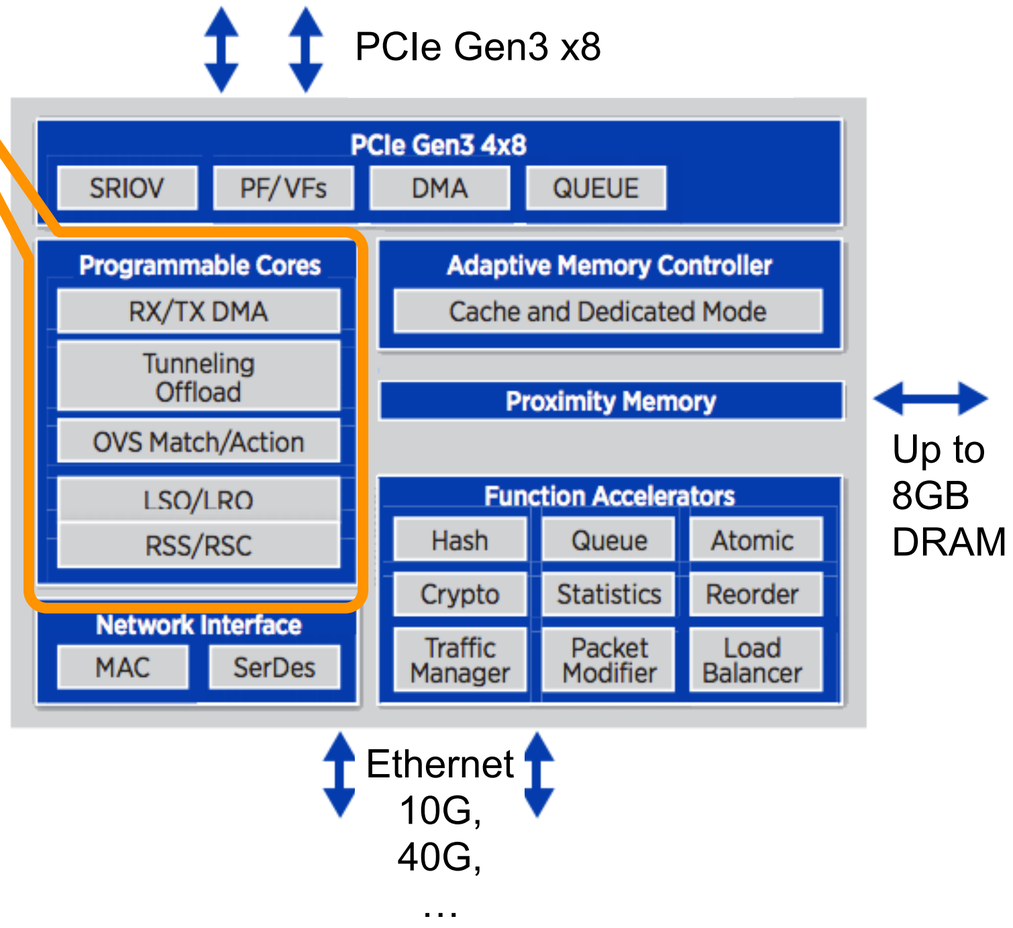
- Highly parallel multithreaded architecture (8 threads / core) for high throughput
- Purpose built Flow Processing Cores (72) maximize flexibility
- H/W accelerators further maximize efficiency (throughput/watt)

Fully software defined feature set - examples:

- Network and PCIe SR-IOV / VirtIO RX/TX with stateless offloads
- Flexible tunneling support (e.g. VXLAN, GRE, VLAN, MPLS, NSH)
- Flexible Match/Action processing - many packet fields / protocols
- Highly scalable and fine grained security policies

External DDR3 accommodates millions of flows / rules

Convenient programmability using P4 and C



Flow Processor Silicon Architecture

Network Flow Processor 4xxx (used on Agilio-CX SmartNICs)

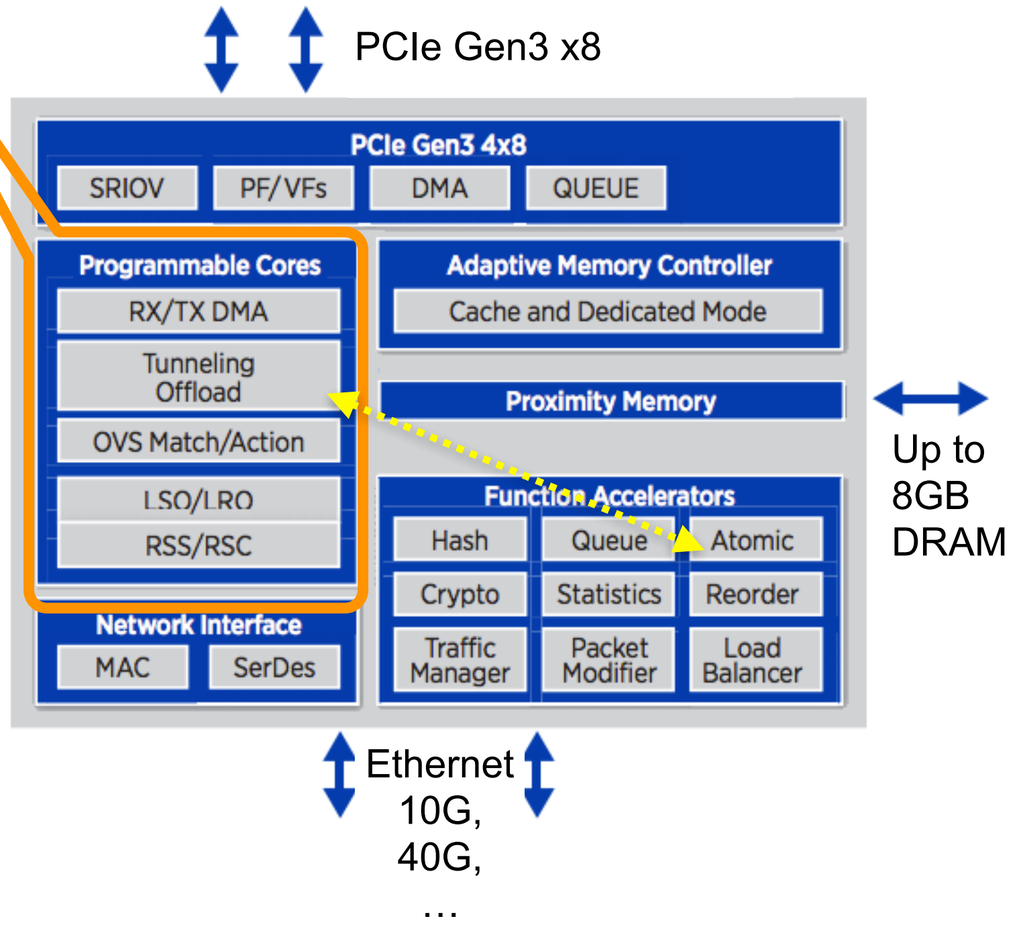
- Highly parallel multithreaded architecture (8 threads / core) for high throughput
- Purpose built Flow Processing Cores (72) maximize flexibility
- H/W accelerators further maximize efficiency (throughput/watt)

Fully software defined feature set - examples:

- Network and PCIe SR-IOV / VirtIO RX/TX with stateless offloads
- Flexible tunneling support (e.g. VXLAN, GRE, VLAN, MPLS, NSH)
- Flexible Match/Action processing - many packet fields / protocols
- Highly scalable and fine grained security policies

External DDR3 accommodates millions of flows / rules

Convenient programmability using P4 and C



Flow Processor Silicon Architecture

Network Flow Processor 4xxx (used on Agilio-CX SmartNICs)

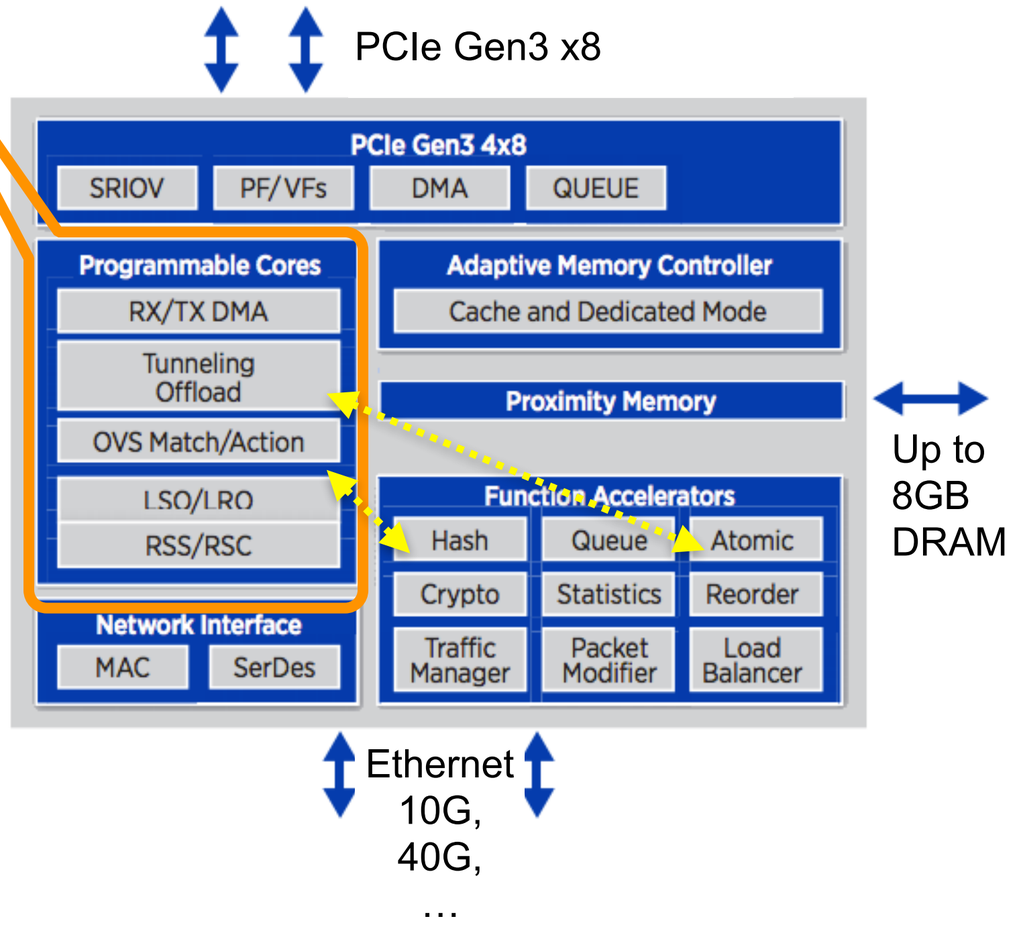
- Highly parallel multithreaded architecture (8 threads / core) for high throughput
- Purpose built Flow Processing Cores (72) maximize flexibility
- H/W accelerators further maximize efficiency (throughput/watt)

Fully software defined feature set - examples:

- Network and PCIe SR-IOV / VirtIO RX/TX with stateless offloads
- Flexible tunneling support (e.g. VXLAN, GRE, VLAN, MPLS, NSH)
- Flexible Match/Action processing - many packet fields / protocols
- Highly scalable and fine grained security policies

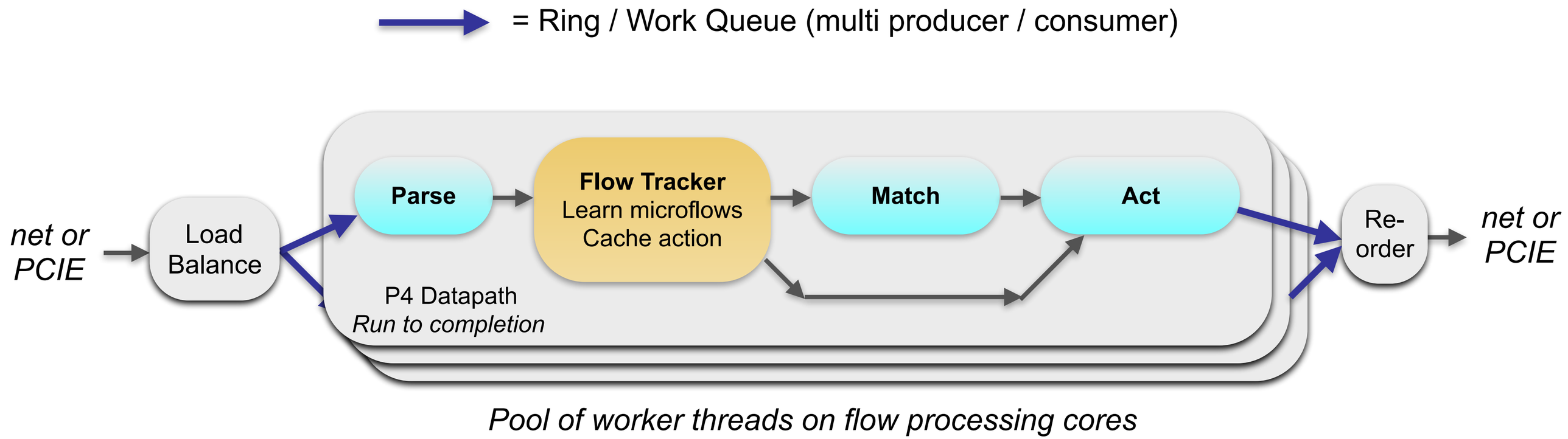
External DDR3 accommodates millions of flows / rules

Convenient programmability using P4 and C



SmartNIC Datapath “Worker” Software Architecture

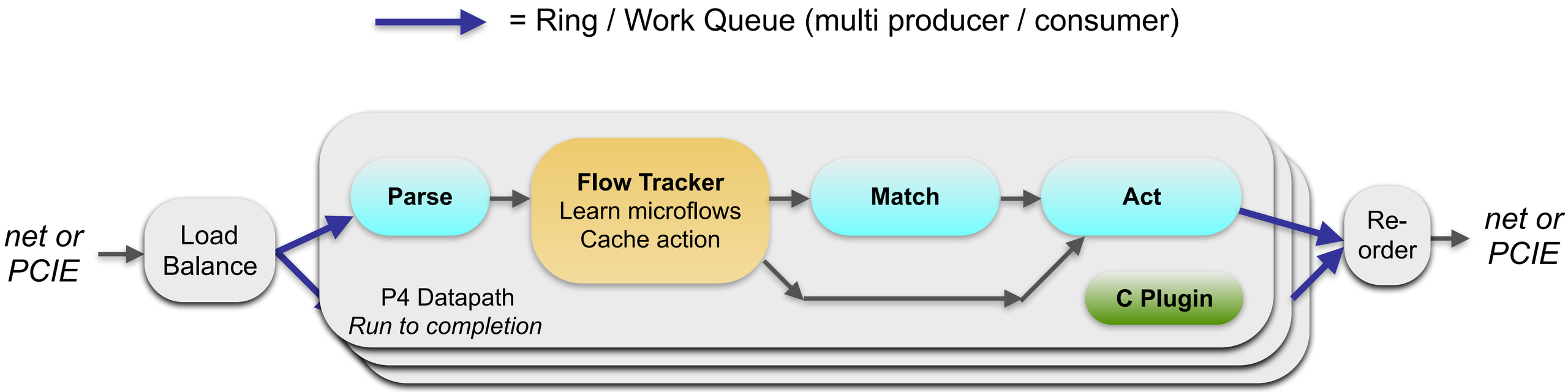
- Load balancer distributes each packet to next available thread for optimum throughput
- Hardware assisted reordering ensures packet order is maintained
- Flow tracker statefully learns / tracks millions of sessions
- Matching performed using DRAM-backed tables - capacity millions of entries
- Actions efficiently performed in on-chip memory



Pool of worker threads on flow processing cores

SmartNIC Datapath “Worker” Software Architecture

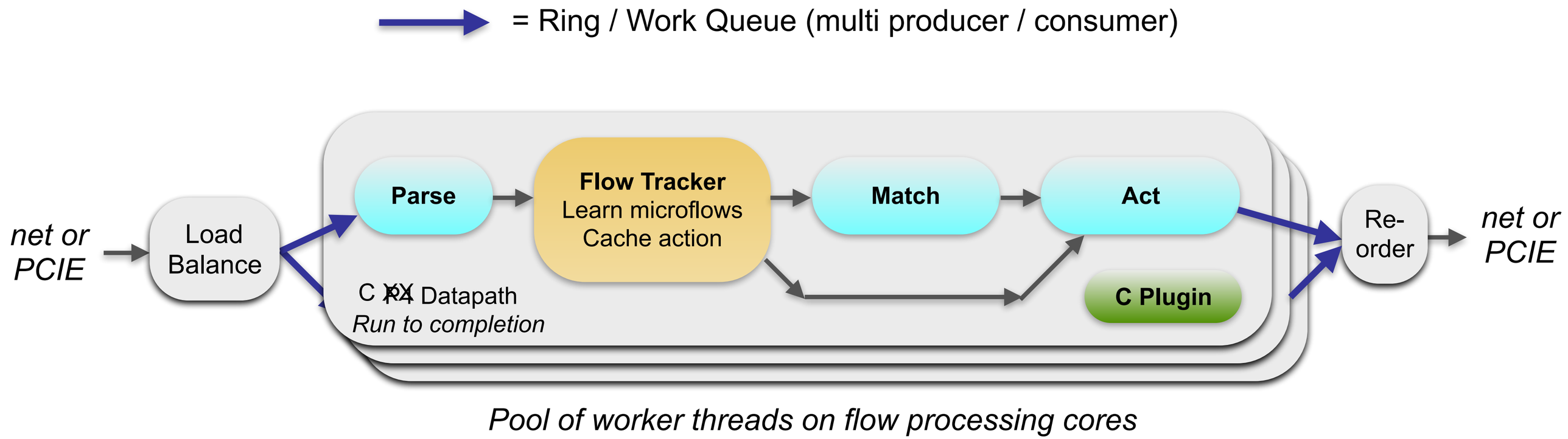
- Load balancer distributes each packet to next available thread for optimum throughput
- Hardware assisted reordering ensures packet order is maintained
- Flow tracker statefully learns / tracks millions of sessions
- Matching performed using DRAM-backed tables - capacity millions of entries
- Actions efficiently performed in on-chip memory



Pool of worker threads on flow processing cores

SmartNIC Datapath “Worker” Software Architecture

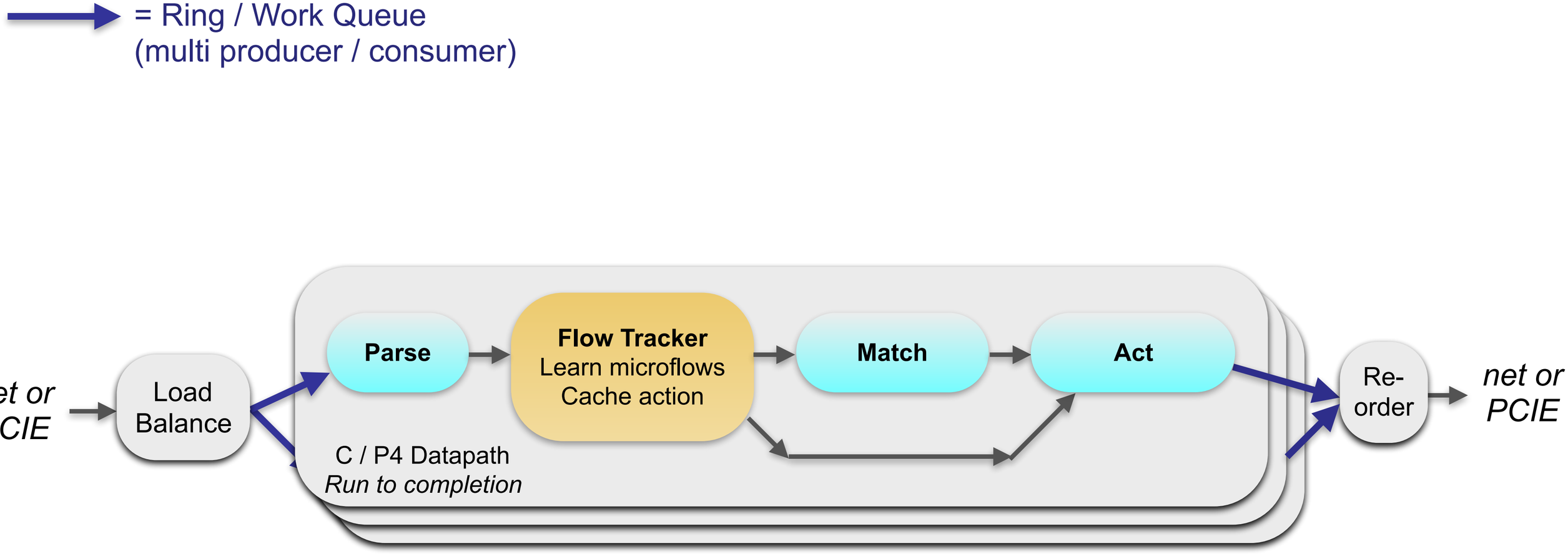
- Load balancer distributes each packet to next available thread for optimum throughput
- Hardware assisted reordering ensures packet order is maintained
- Flow tracker statefully learns / tracks millions of sessions
- Matching performed using DRAM-backed tables - capacity millions of entries
- Actions efficiently performed in on-chip memory



Pool of worker threads on flow processing cores

Datapath Distributed over Microengines

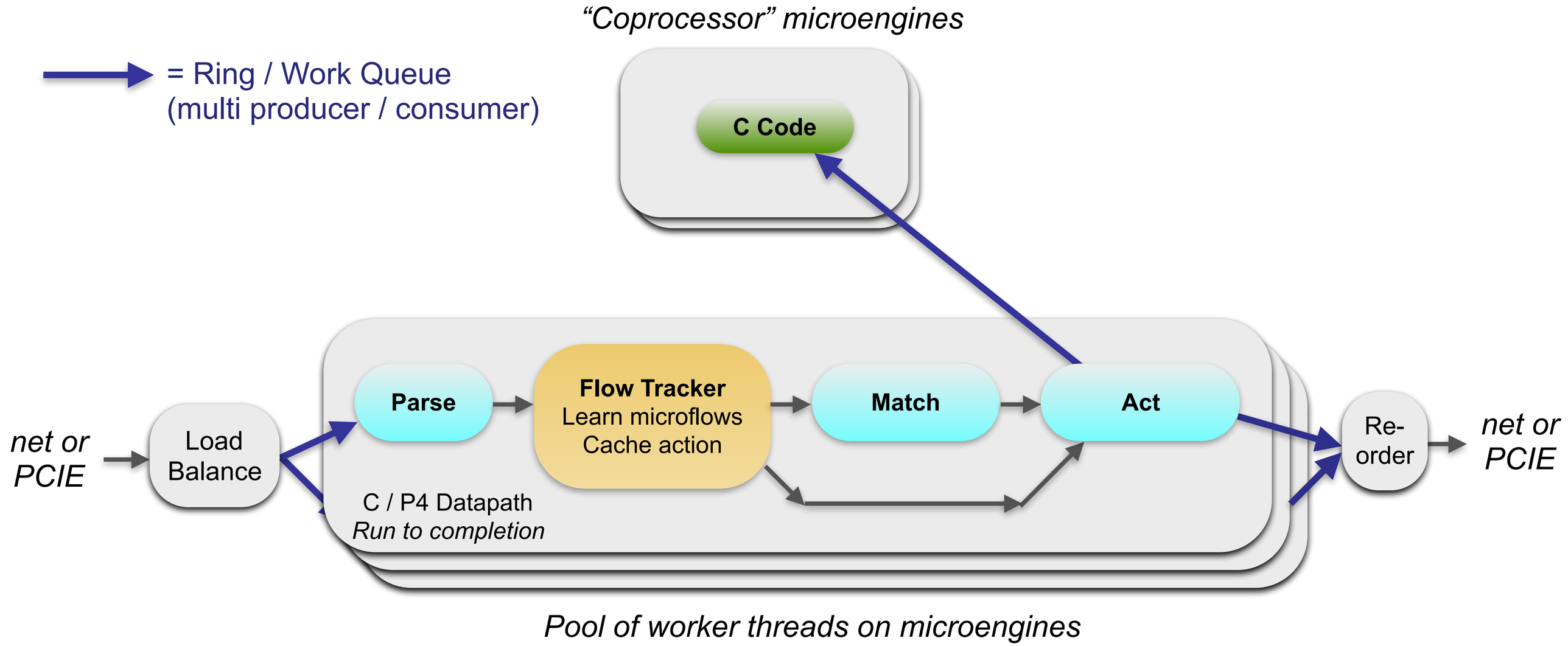
- Worker uses ring to forward packets to other microengine(s)



Pool of worker threads on microengines

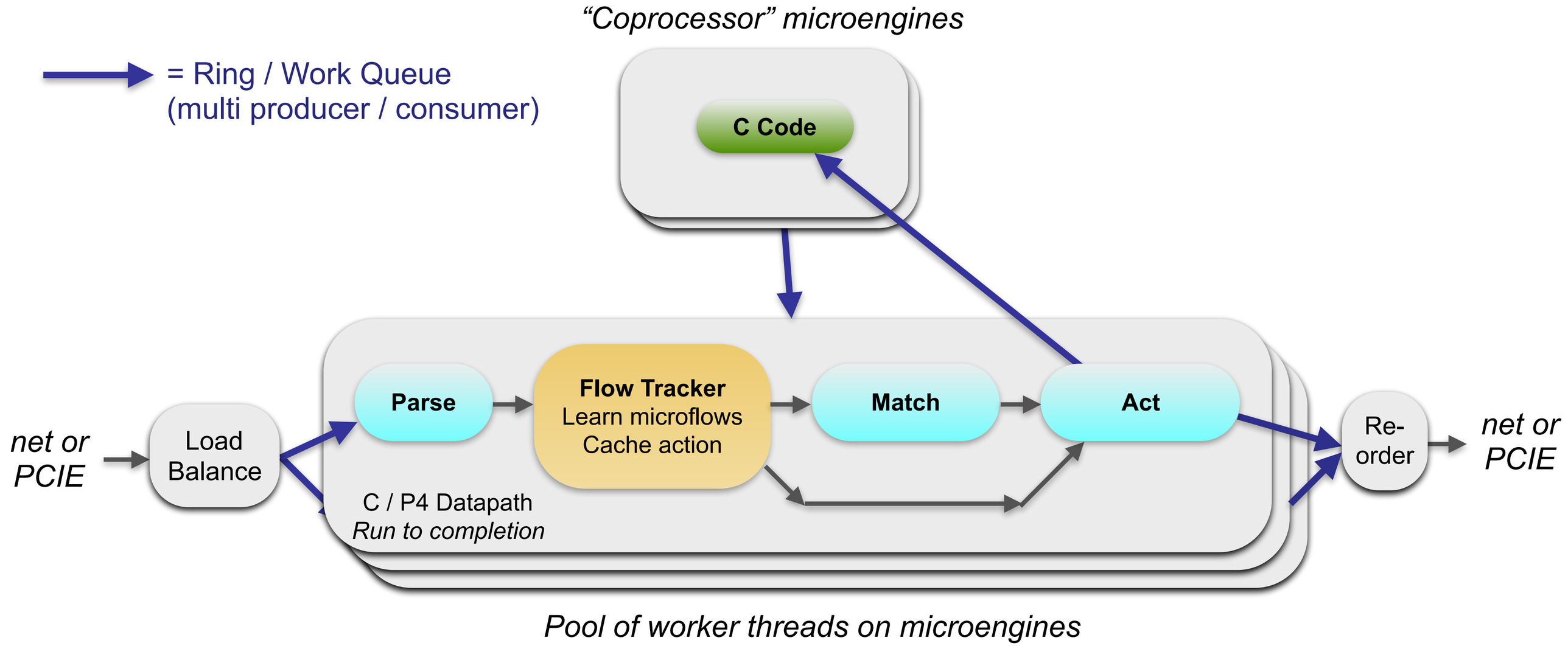
Datapath Distributed over Microengines

- Worker uses ring to forward packets to other microengine(s)



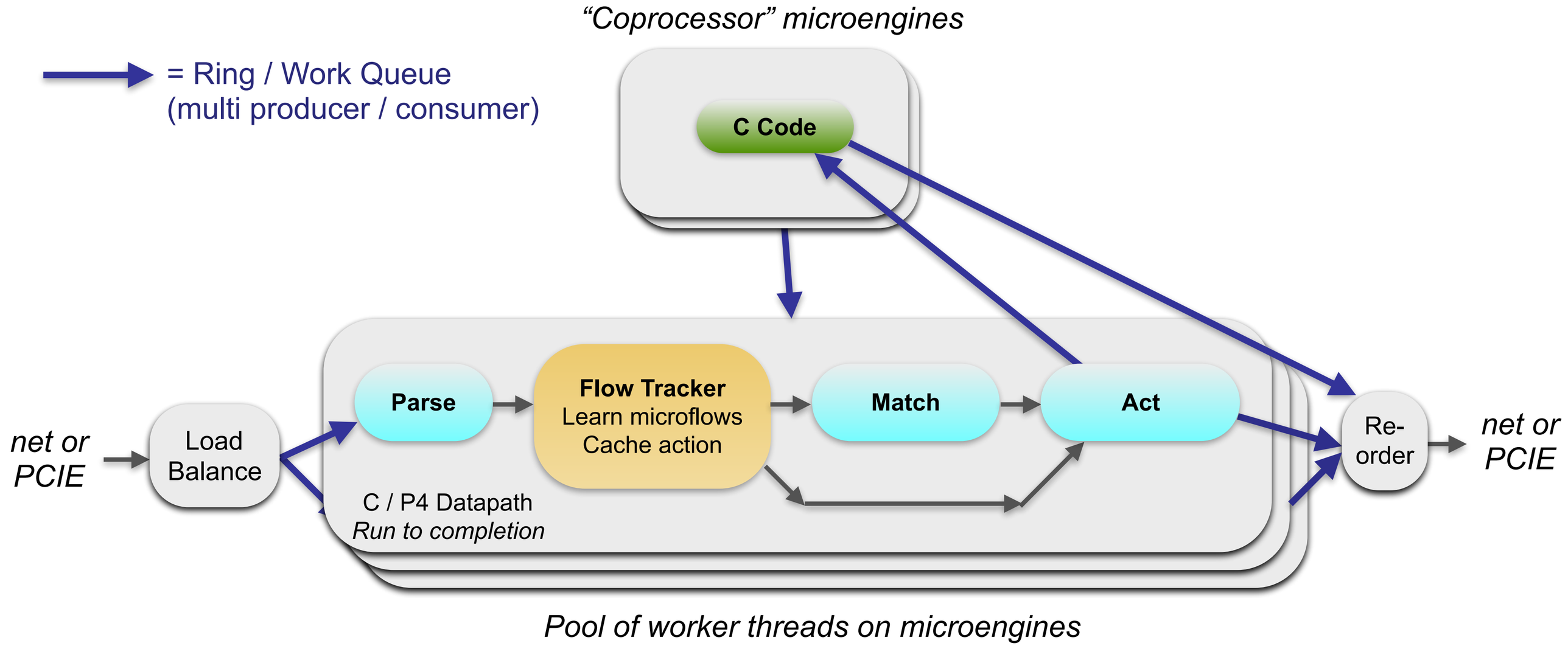
Datapath Distributed over Microengines

- Worker uses ring to forward packets to other microengine(s)



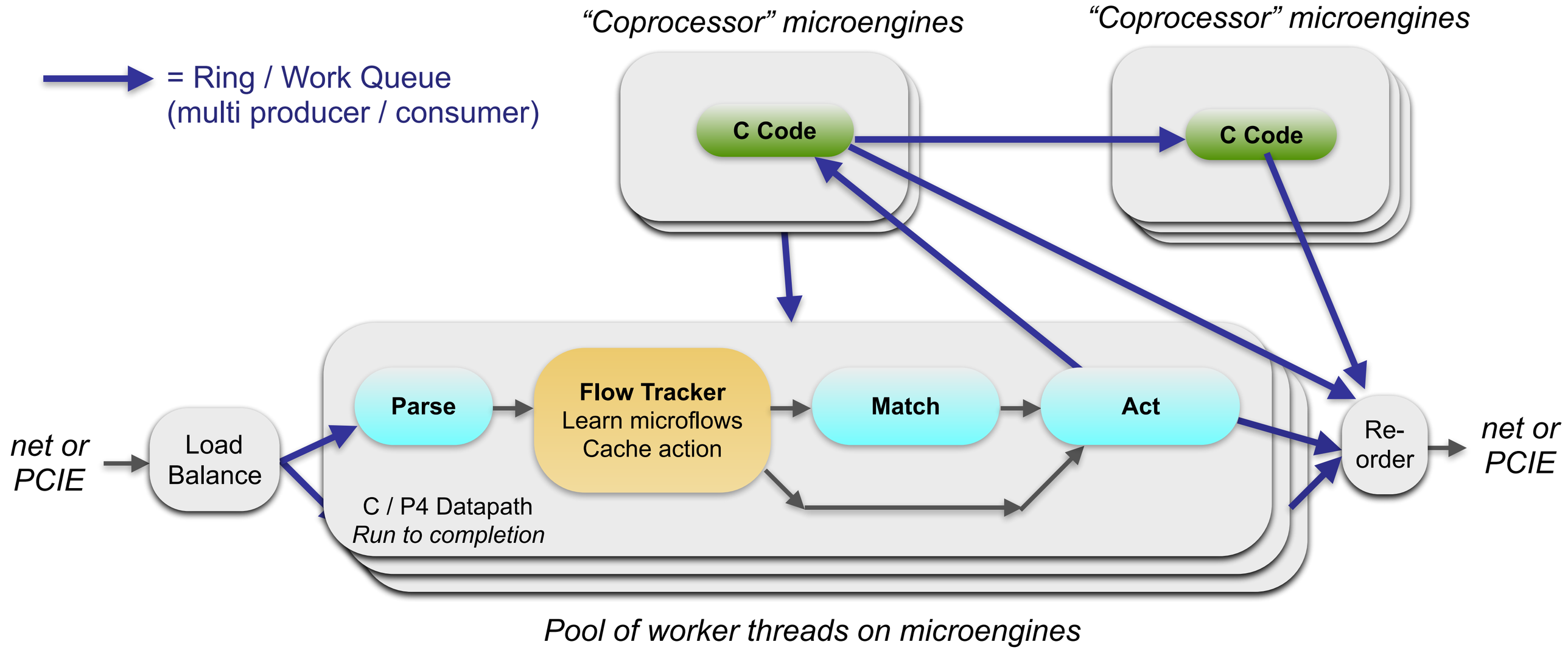
Datapath Distributed over Microengines

- Worker uses ring to forward packets to other microengine(s)



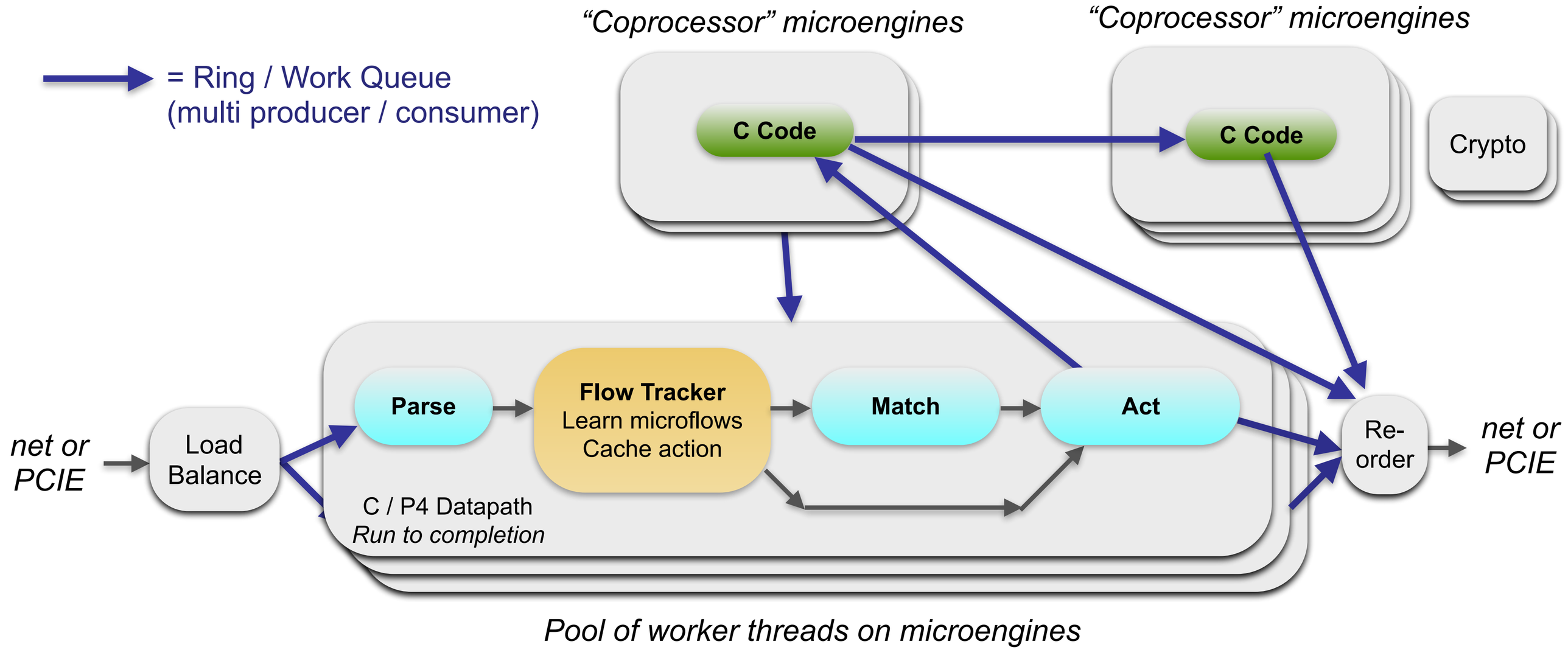
Datapath Distributed over Microengines

- Worker uses ring to forward packets to other microengine(s)



Datapath Distributed over Microengines

- Worker uses ring to forward packets to other microengine(s)



Example: P4 “main” implementing a simple NIC

```
header_type eth_hdr {
  fields {
    dst : 48;
    src : 48;
    etype : 16;
  }
}
header eth_hdr eth;

parser start {
  return eth_parse;
}

parser eth_parse {
  extract(eth);
  return ingress;
}
```

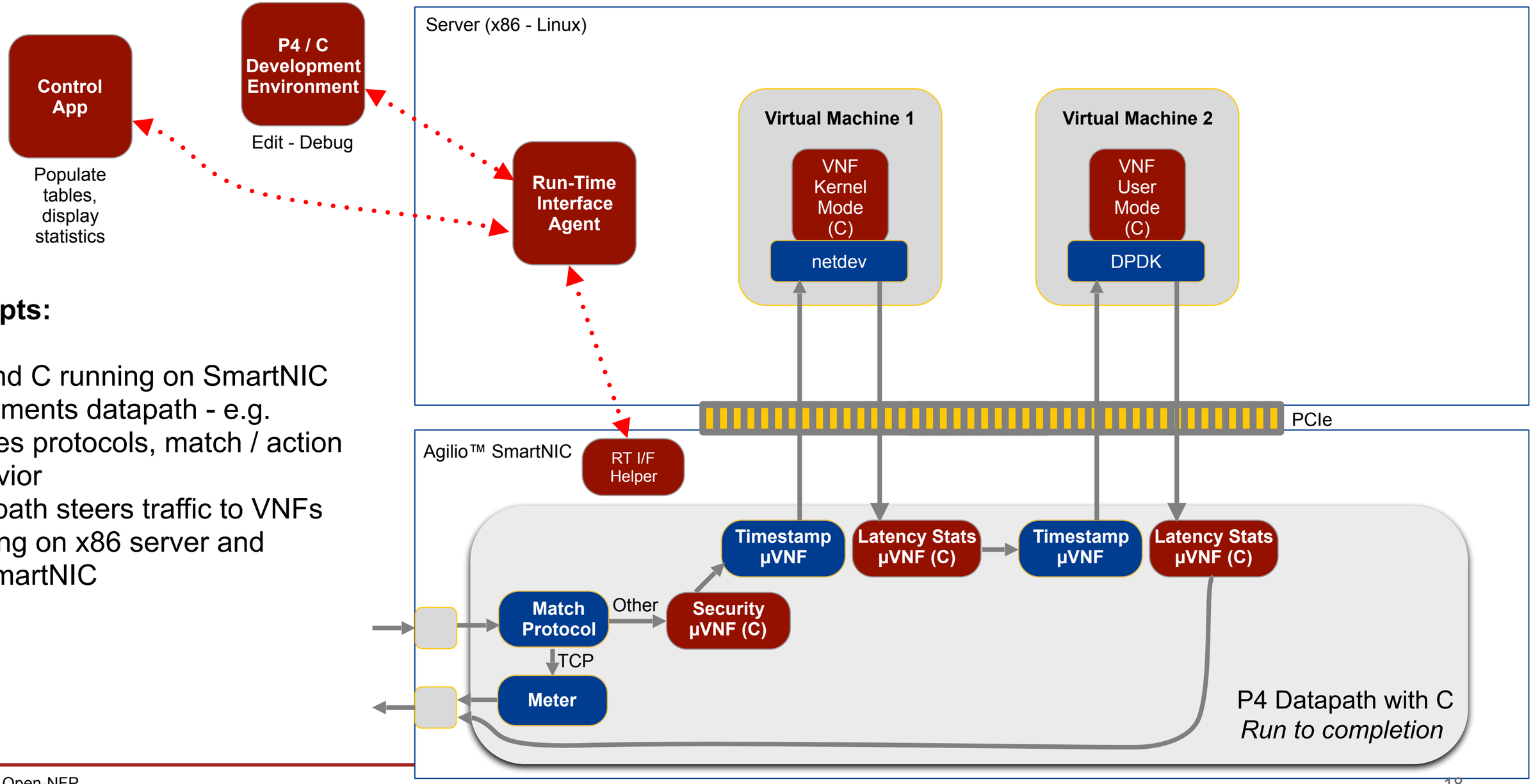
```
action drop_act() {
  drop();
}

action fwd_act(port) {
  modify_field(standard_metadata.egress_spec,
  port);
}

table in_tbl {
  reads {
    standard_metadata.ingress_port : exact;
  }
  actions {
    fwd_act;
    drop_act;
  }
}
```

```
control ingress {
  apply(in_tbl);
}
```

Example: C Code

```
static void
init_me()
{
    uint32_t cmask;
    cmask = disable_ctxs();
    pkt_init_rx();
    pkt_init_tx();
    enable_ctxs(cmask);
}


void
main()
{
    /* Configuration and initialization */
    if (ctx() == 0)
        init_me();

    for (;;) {
        /* half the threads receive from the wire and half from the host */
        if ((ctx() & 1) == 0) {
            if (pkt_rx_wire() == -1)
                goto transmit;
        } else {
            if (pkt_rx_host() == -1)
                goto transmit;
        }

        /* INSERT YOUR CODE HERE */
        /* Bump packet to the adjacent port */
        Pkt.p_dst = Pkt.p_src ^ 1;

        /* Attempt to send and drop if we encountered an error */
transmit:
        pkt_tx();
    }
}
```

Example of Fully Customized Datapath (P4 / C)



Concepts:

- P4 and C running on SmartNIC implements datapath - e.g. defines protocols, match / action behavior
- Datapath steers traffic to VNFs running on x86 server and on SmartNIC

Next Steps

- Use Agilio™ SmartNICs with existing dataplanes
  - Use Agilio™ OVS (with / without Conntrack)
  - Use Agilio™ Contrail vRouter
  - Use Agilio™ eBPF/XDP
- Program Agilio™ SmartNICs *(following sessions - SDK, P4, OVS, eBPF/XDP)*
  - Program using P4, C, eBPF/XDP…
- Participate in open source and standards evolution: openstack.org, openvswitch.org, opencontrail.org, p4.org, iovisor.org, open-nfp.org, opennetworking.org, opensourcesdn.org, opnfv.org, linuxfoundation.org
  - Examples: P4 / OpenFlow callable run-time API (cross-body effort starting), acceleration APIs

Increase flexibility, improve performance, free up server resources!

Thank You!

More information: netronome.com
and: open-nfp.org