CHINA
OpenInfra Days

CHINA
OpenInfra Days | IT大咖说 知识共享平台

Cloud Accelerated

Cyborg Project Year 1 Review

Why Cyborg As A New Project

WHY

CYBORG AS A NEW OPROJECT

- Acceleration has become a necessity rather than an interesting option

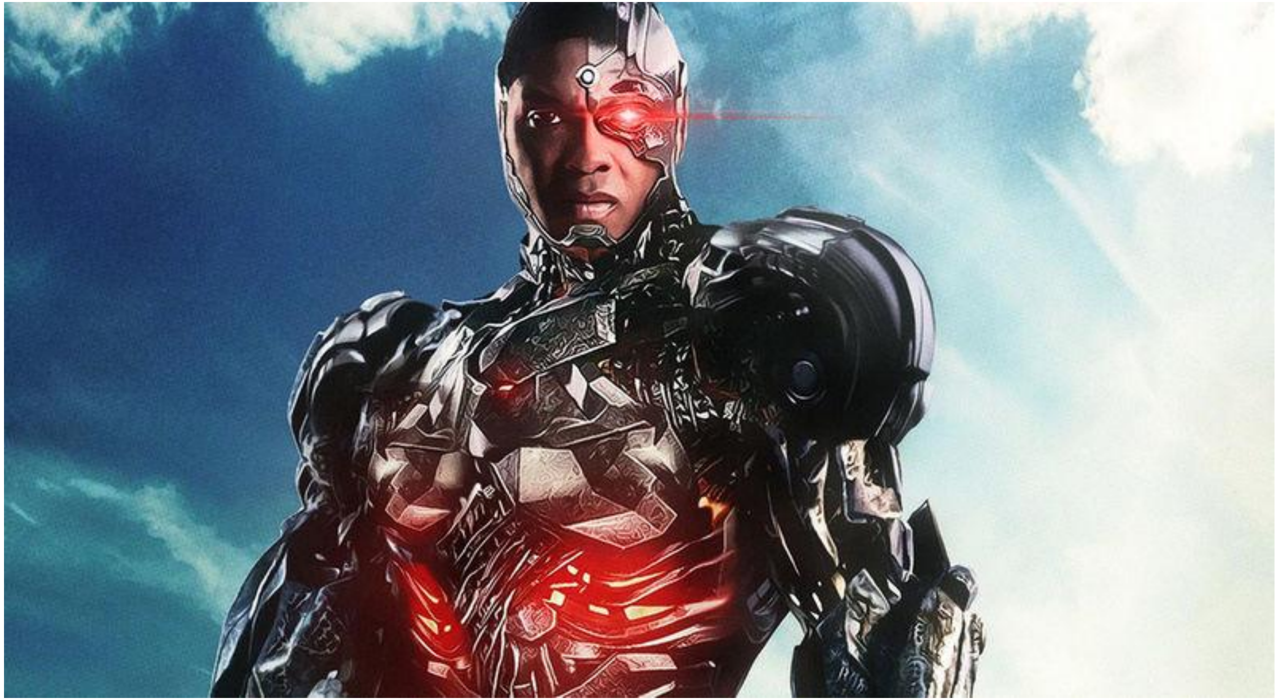
BACKGROUND: HISTORY

- OpenStack Acceleration Discussion Started from Telco Requirements
  - High level requirements first drafted in the standard organization ETSI NFV ISG
  - High level requirements transformed into detailed requirements in OPNFV DPACC project.
  - New project called Nomad established to address the requirements.
  - BoF discussions back in OpenStack Austin Summit.

BACKGROUND: HISTORY

- Transition to Cyborg Project
  - After a long period of discussions within the OpenStack community, we discovered that the initial goal of Nomad project to address acceleration management in Telco was too limited. Developers from Scientific WG help us understand the need for acceleration management in HPC cloud at the Barcelona Design Summit which also led to a lot of discussion on the Public Cloud support of accelerated instances.
  - The aforementioned discussions led us to formally establish a project that will work on the management framework for dedicated devices in OpenStack called the Cyborg Project.

BACKGROUND: HISTORY



- http://fandom.wikia.com/articles/martian-manhunter-replaced-cyborg-justice-league-founder

BACKGROUND: OPENSTACK SCIENTIFIC WG FEEDBACK ON GPU

- GPUs *can* be used in openstack.
  - GPU specific flavors: pci_passthrough alias need to be populated in the properties field of extra_specs for the specific instance
  - KVM tuning is required to achieve acceptable performance
- Two options:
  - Heterogeneous hosts: GPU and CPU-only hosts mixed
    - Good - CPU resources are available for all workloads
    - Bad - scheduler does not prioritize GPU workloads.
  - GPU only Host-Aggregate: GPU hosts are segregated
    - Good - GPU hosts are only used for GPU workloads
    - Bad - CPU-only workloads unable to use underutilized GPU hosts

USE CASES FOR CYBORG OPERATORS

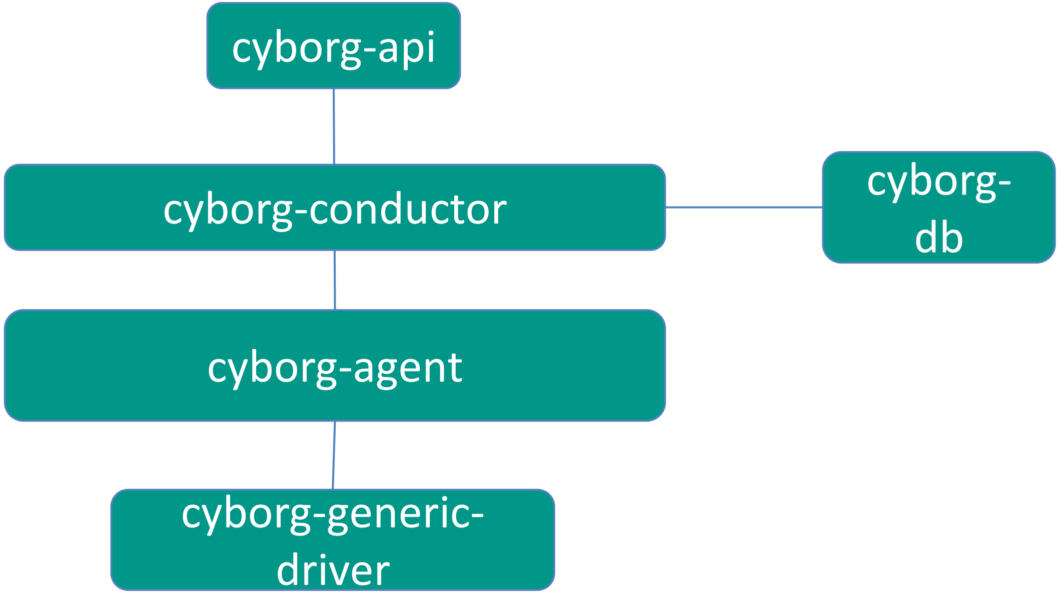
What we actually want from a project like Cyborg:

- List accelerators
  - (cyborg accelerator list --feature-tag *DEEP_LEARNING)*
- Identify and discover attached accelerators
  - (cyborg accelerator discover)
- Attach and detach accelerators to an instance
  - (cyborg accelerator attach --instance-id *FPGA_VF_1)*
- Install and uninstall a driver
  - (cyborg accelerator install --driver-id *SPDK_Driver)*

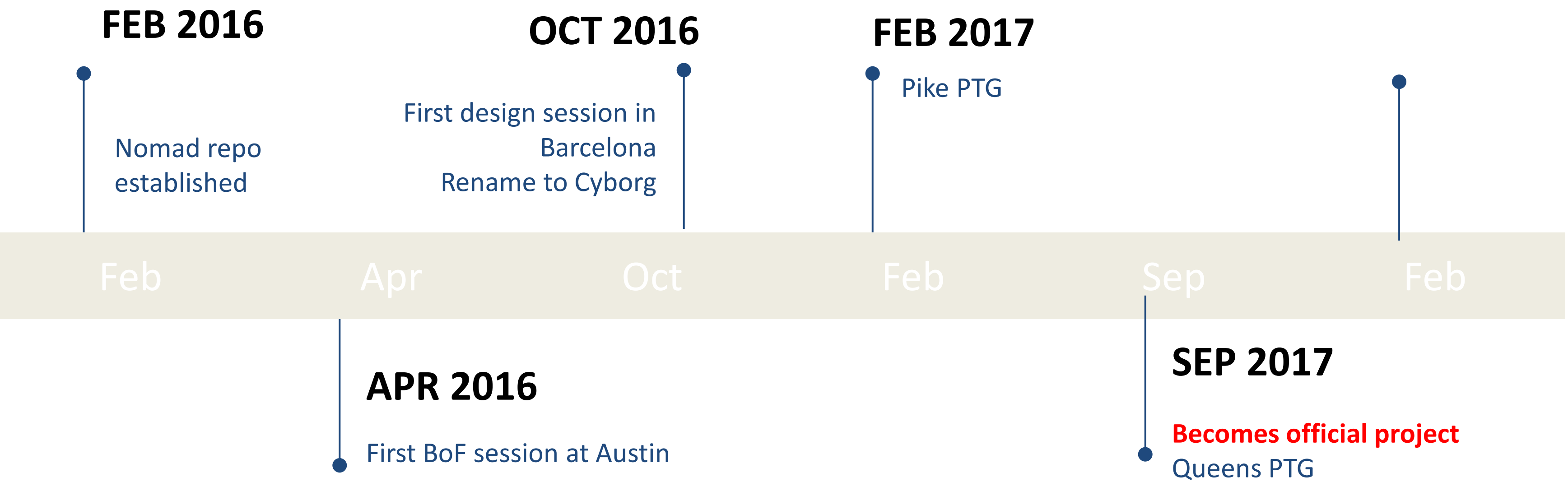
WHAT (CYBORG OVERVIEW)

- Cyborg is a general management framework for accelerators
  - We have the LONGEST team meetings

ARCHITECTURE



TIMELINE



FEB 2016

Nomad repo established

APR 2016

First BoF session at Austin

OCT 2016

First design session in Barcelona
Rename to Cyborg

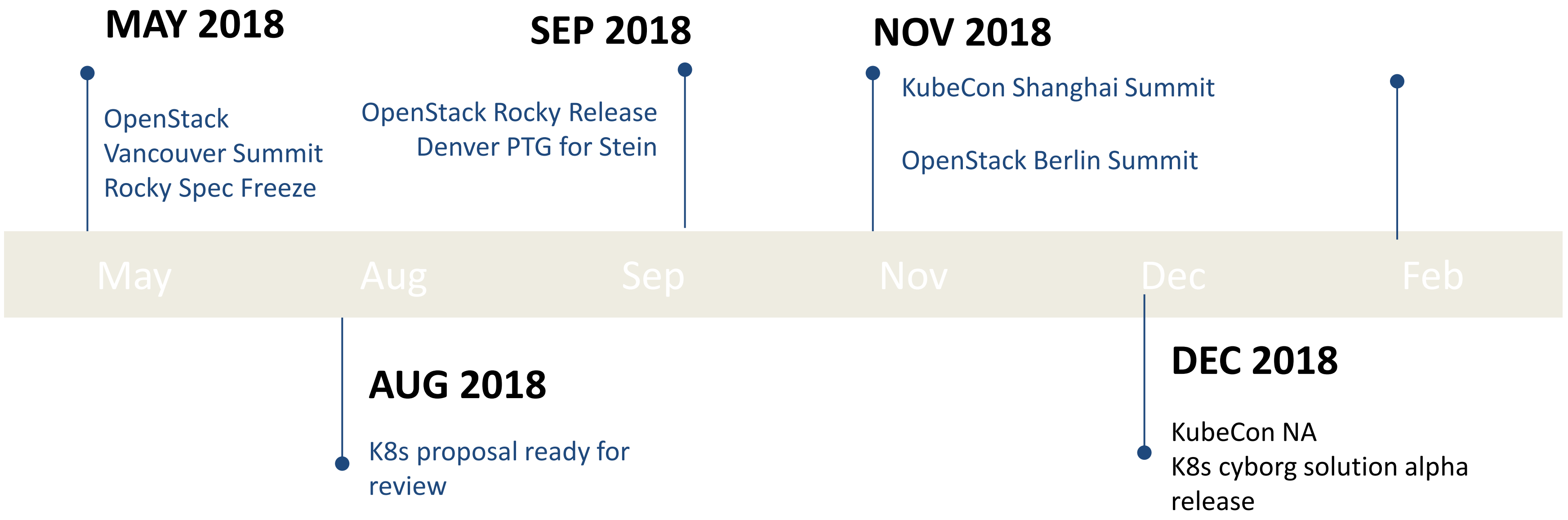
FEB 2017

Pike PTG

SEP 2017

Becomes official project
Queens PTG

TIMELINE (PLANNED)



MAY 2018

OpenStack
Vancouver Summit
Rocky Spec Freeze

AUG 2018

K8s proposal ready for review

SEP 2018

OpenStack Rocky Release
Denver PTG for Stein

NOV 2018

KubeCon Shanghai Summit

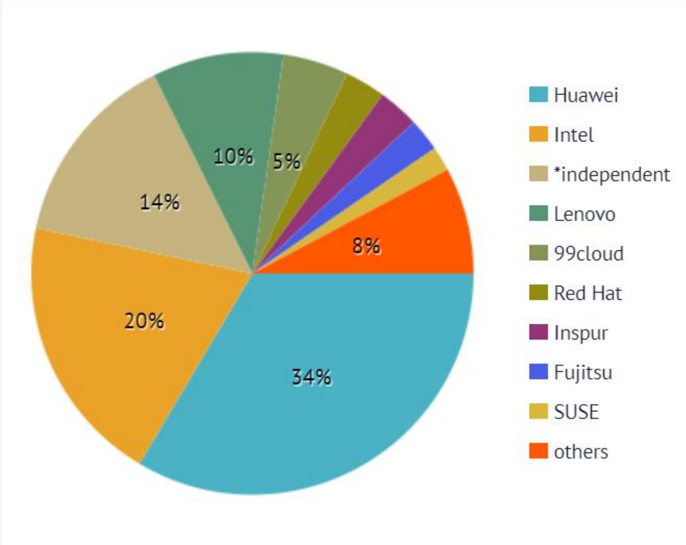
OpenStack Berlin Summit

DEC 2018

KubeCon NA
K8s cyborg solution alpha release

OPEN COMMUNITY

- Development: https://review.openstack.org/#/q/status:open+project:openstack/cyborg
- Use openstack-dev mailing list with [acceleration] or [Cyborg]
- Wiki at https://wiki.openstack.org/wiki/Cyborg
- Weekly irc meeting at #openstack-cyborg
- Stats
- Looking for more resources
- Give a shout out at #openstack-cyborg



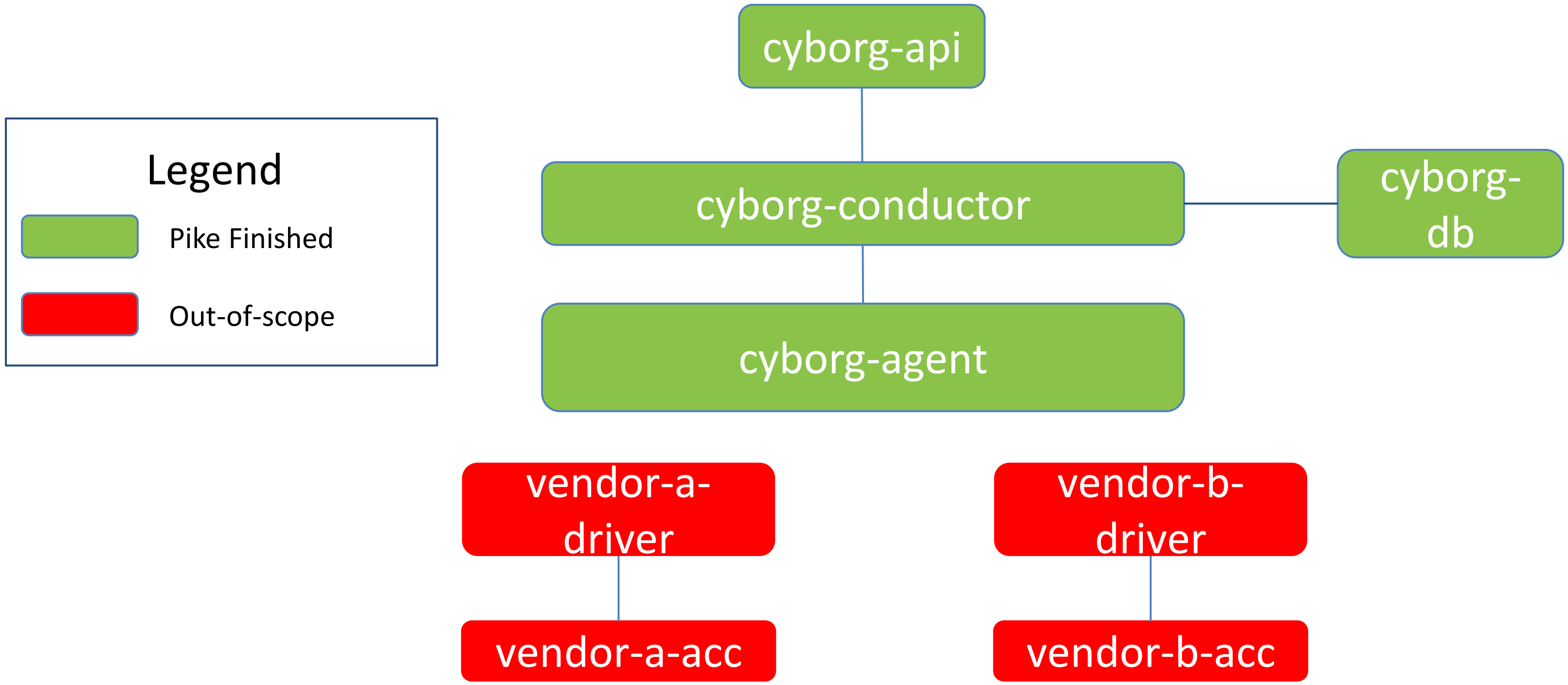
WHAT (CYBORG PIKE RELEASE)

WHAT

(CYBORG PIKE RELEASE)

Cyborg Pike release with its basic framework ready

CYBORG PIKE RELEASE



Legend

🟩 Pike Finished

🟥 Out-of-scope

cyborg-api

cyborg-conductor

cyborg-db

cyborg-agent

vendor-a-driver

vendor-a-acc

vendor-b-driver

vendor-b-acc

PIKE RELEASE

- Self Release
  - Basic framework
    - REST API
    - Conductor & Agent
    - Generic Stub Driver
    - Devstack Plugin
  - Initial docs and testing materials

WHAT (CYBORG QUEENS RELEASE)

WHAT

(CYBORG QUEENS RELEASE)

Cyborg’s first official release with resource provider data model available and some initial drivers

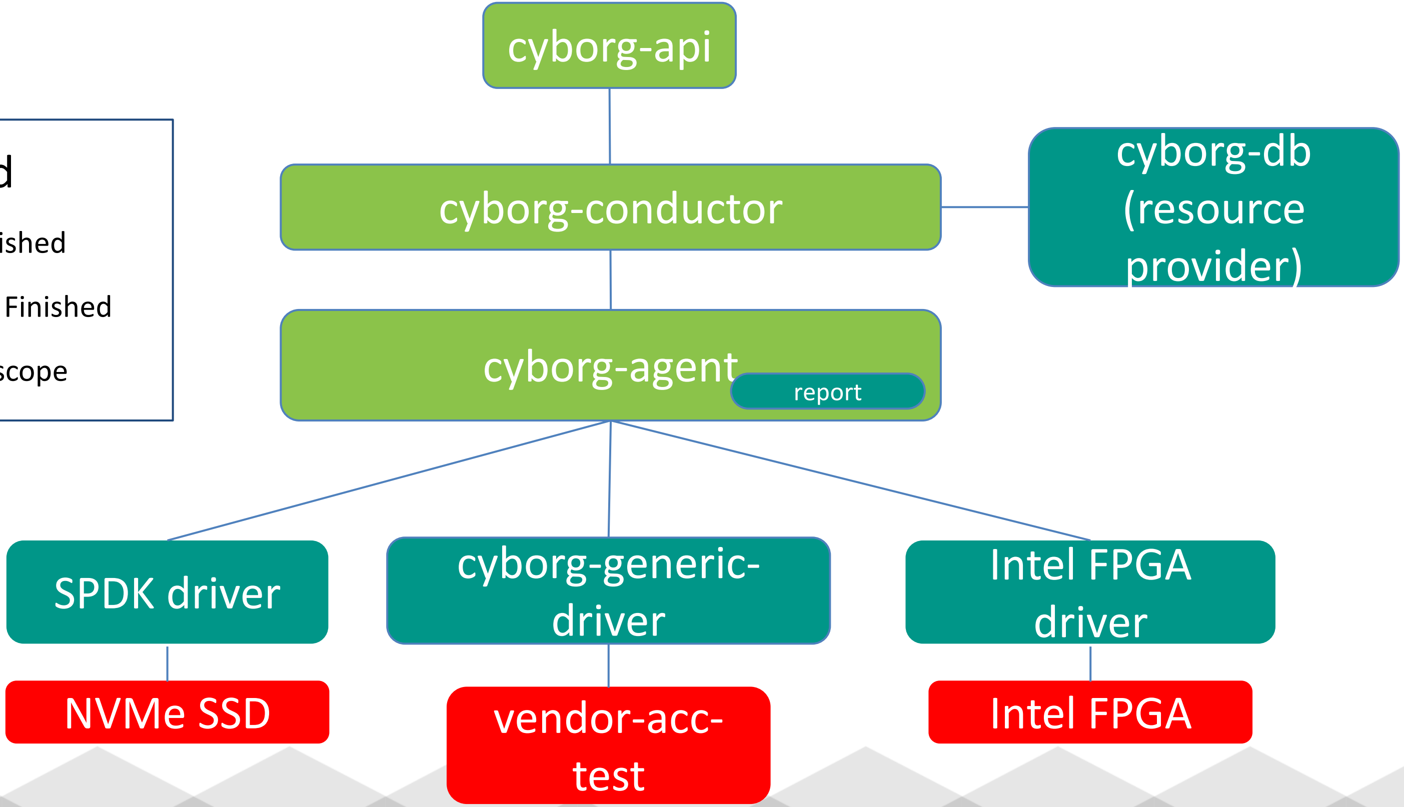
CYBORG QUEENS RELEASE

Legend

🟩 Pike Finished

🟫 Queens Finished

🟥 Out-of-scope



cyborg-api

cyborg-conductor

cyborg-db (resource provider)

cyborg-agent

report

SPDK driver

cyborg-generic-driver

Intel FPGA driver

NVMe SSD

vendor-acc-test

Intel FPGA

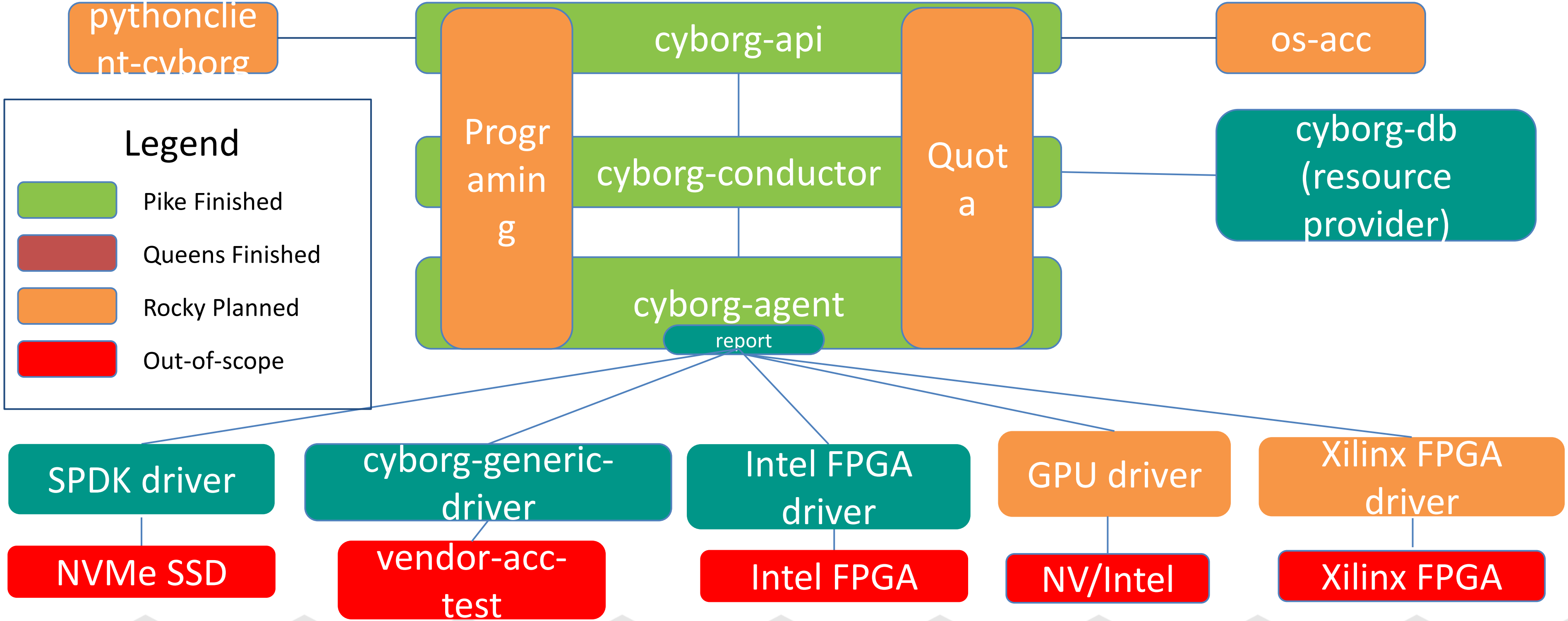
QUEENS RELEASE

- First Official Release
  - Denver PTG discussion: https://etherpad.openstack.org/p/cyborg-queens-ptg
  - Key Features:
    - Resource Provider data model in cyborg DB
    - Interaction with Placement API and resource report
    - Intel FPGA driver
    - SPDK driver

WHAT (CYBORG ROCKY RELEASE)

Many items planned for Rocky release

CYBORG ROCKY PLANNING (OPENSTACK)



pythonclient-cyborg

cyborg-api

cyborg-conductor

cyborg-agent

report

Programming

Quota

os-acc

cyborg-db (resource provider)

Legend

🟩 Pike Finished

🟫 Queens Finished

🟧 Rocky Planned

🟥 Out-of-scope

SPDK driver

NVMe SSD

cyborg-generic-driver

vendor-acc-test

Intel FPGA driver

Intel FPGA

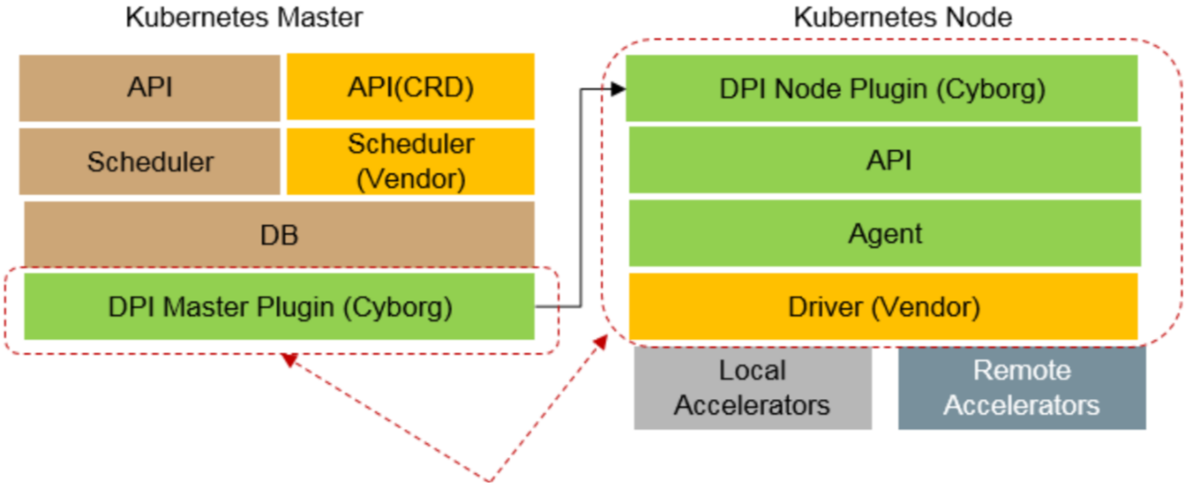
GPU driver

NV/Intel

Xilinx FPGA driver

Xilinx FPGA

CYBORG ROCKY PLANNING (KUBERNETES)



containerized

- Align Cyborg data model with DPI before 1.13 release
- Cyborg DPI Plugin ready when DPI GA
- Consider the possibility of a CRD Acc controller

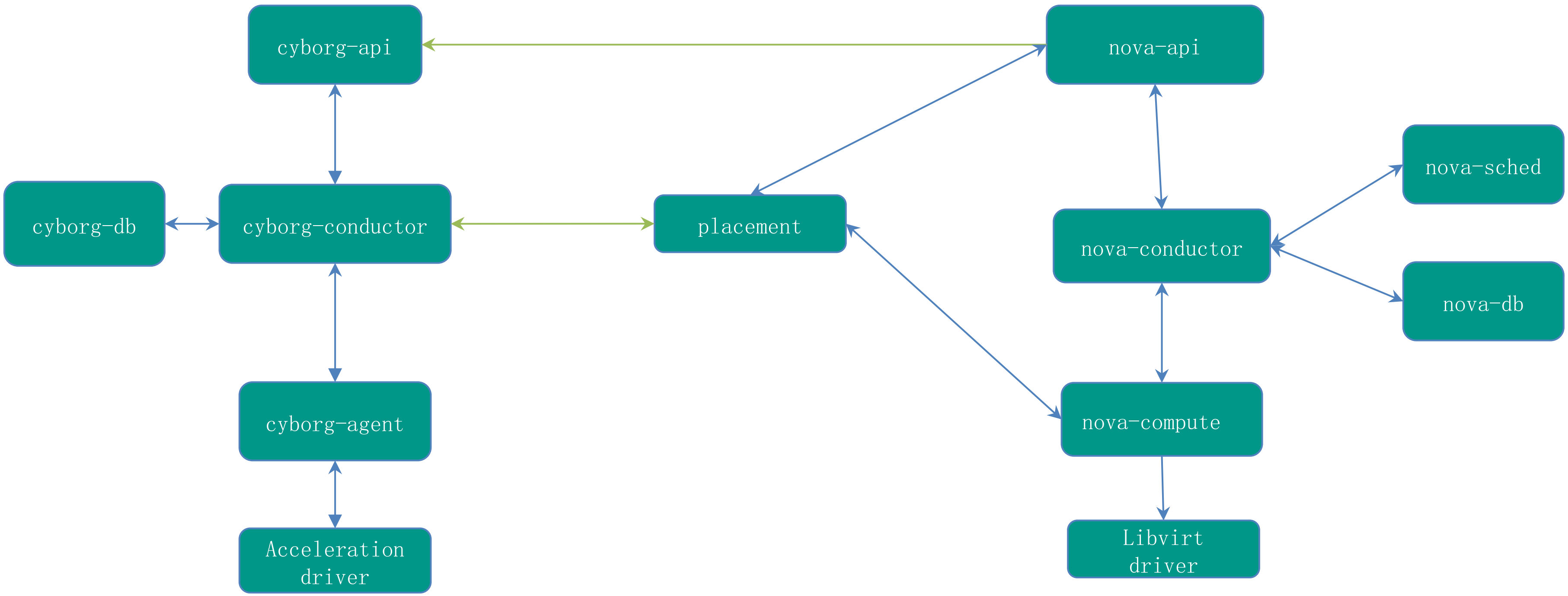
OTHER FUTURE PLANS FOR CYBORG

- Cyborg could be used together with Nova or standalone for bare metal
- Rocky Release Planning with additional ARM collaboration
- Consider the possibility of a CRD Acc controller

HOW

Cyborg could be used together with Nova or standalone for bare metal

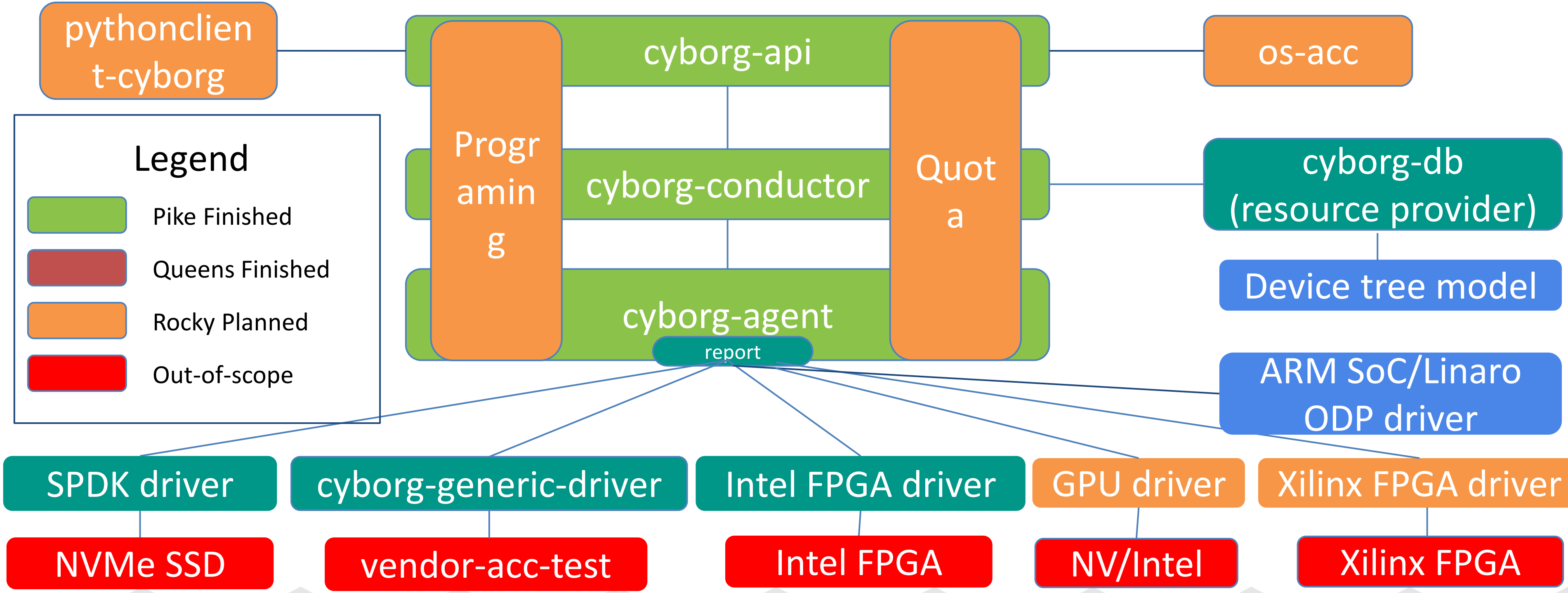
NOVA INTERACTION EXAMPLE



WHERE

Possible ideas for Cyborg ARM collaboration

CYBORG ROCKY RELEASE PLANNING WI ADDITIONAL ARM COLLABORATION



pythonclient-cyborg

cyborg-api

os-acc

Programming

Quota

cyborg-conductor

cyborg-db (resource provider)

Device tree model

cyborg-agent

report

ARM SoC/Linaro ODP driver

SPDK driver

cyborg-generic-driver

Intel FPGA driver

GPU driver

Xilinx FPGA driver

NVMe SSD

vendor-acc-test

Intel FPGA

NV/Intel

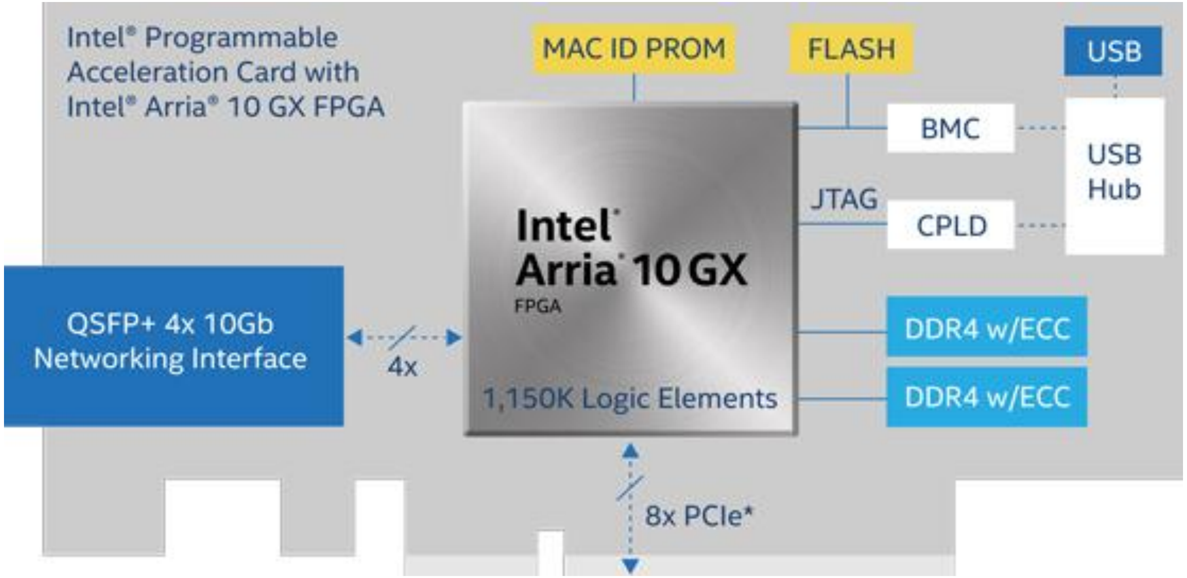
Xilinx FPGA

Legend

- 🟩 Pike Finished
- 🟥 Queens Finished
- 🟧 Rocky Planned
- 🟥 Out-of-scope

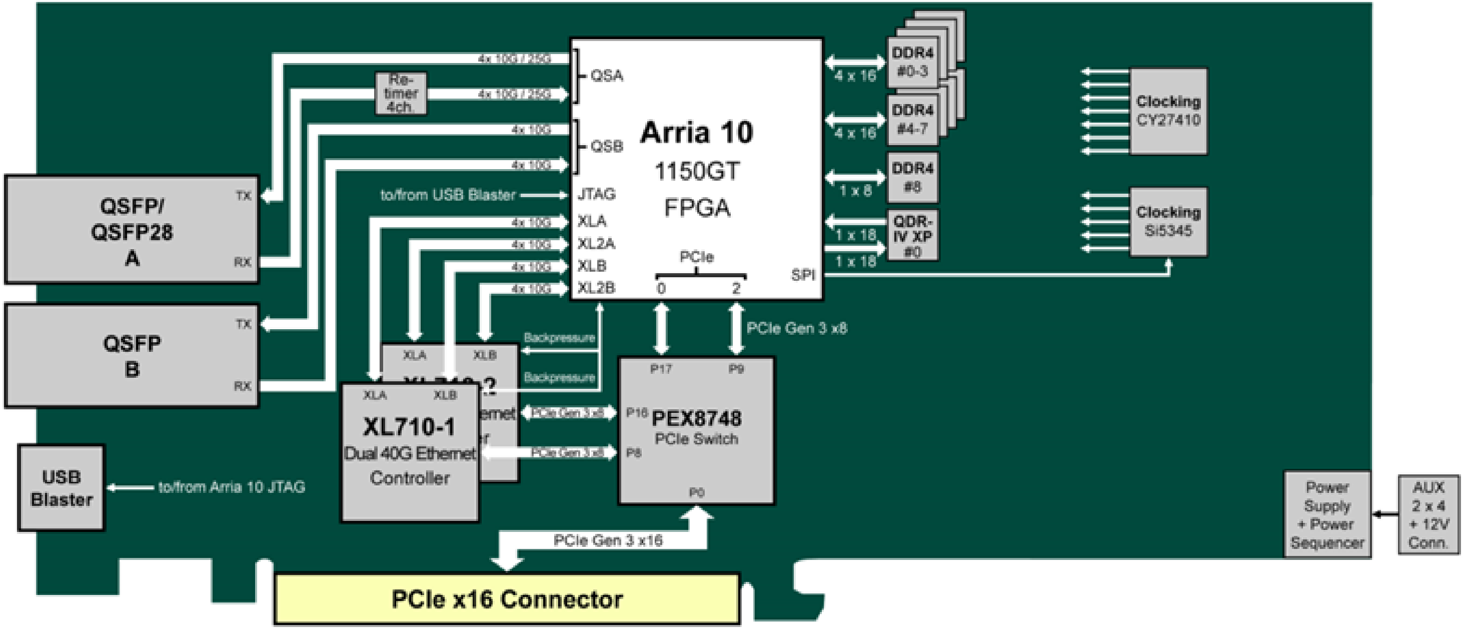
ACCELERATORS

PAC



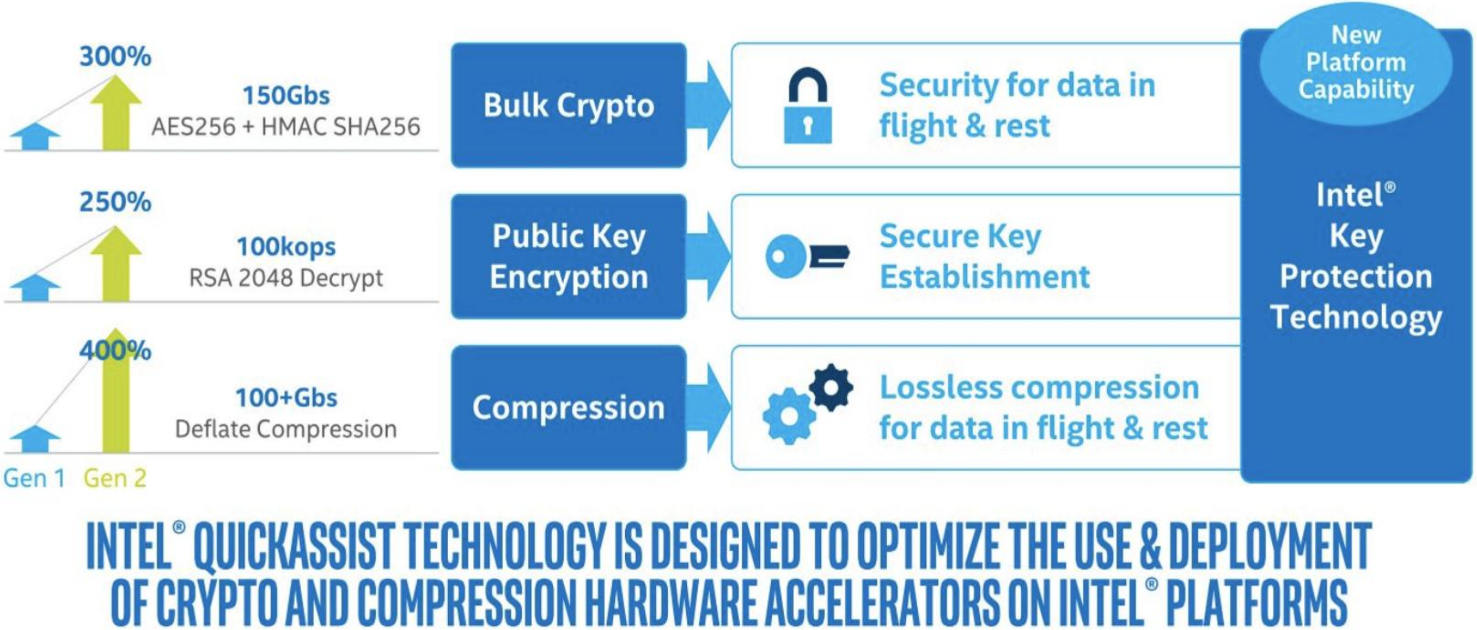
ACCELERATORS

SMARTNIC



ACCELERATORS

QAT



Bulk Crypto → Security for data in flight & rest

Public Key Encryption → Secure Key Establishment

Compression → Lossless compression for data in flight & rest

New Platform Capability

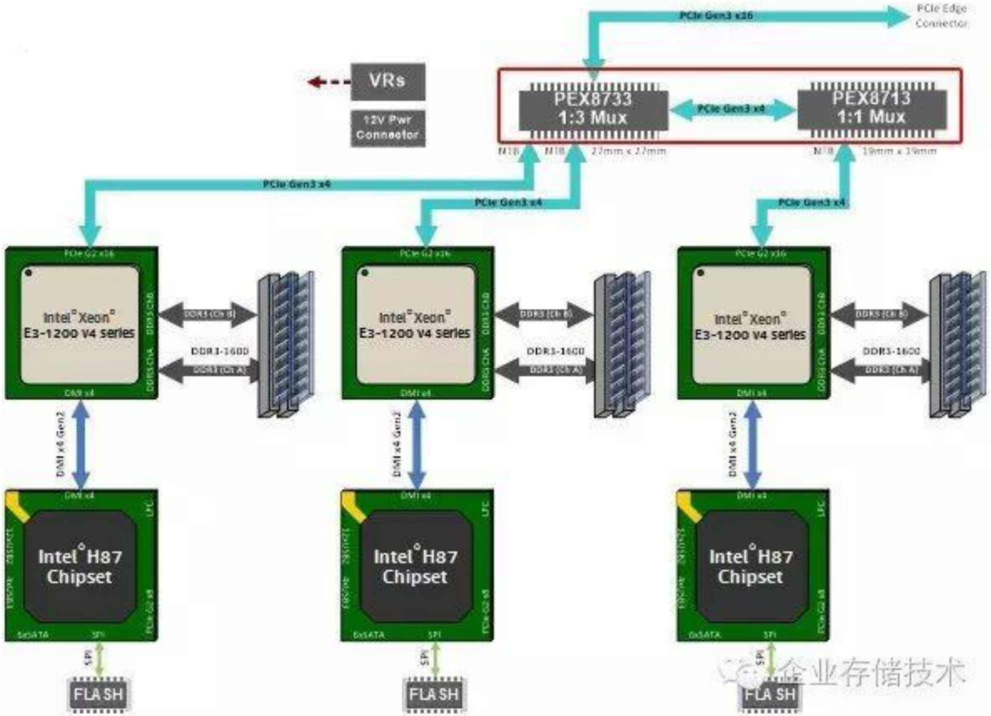
Intel® Key Protection Technology

INTEL® QUICKASSIST TECHNOLOGY IS DESIGNED TO OPTIMIZE THE USE & DEPLOYMENT OF CRYPTO AND COMPRESSION HARDWARE ACCELERATORS ON INTEL® PLATFORMS

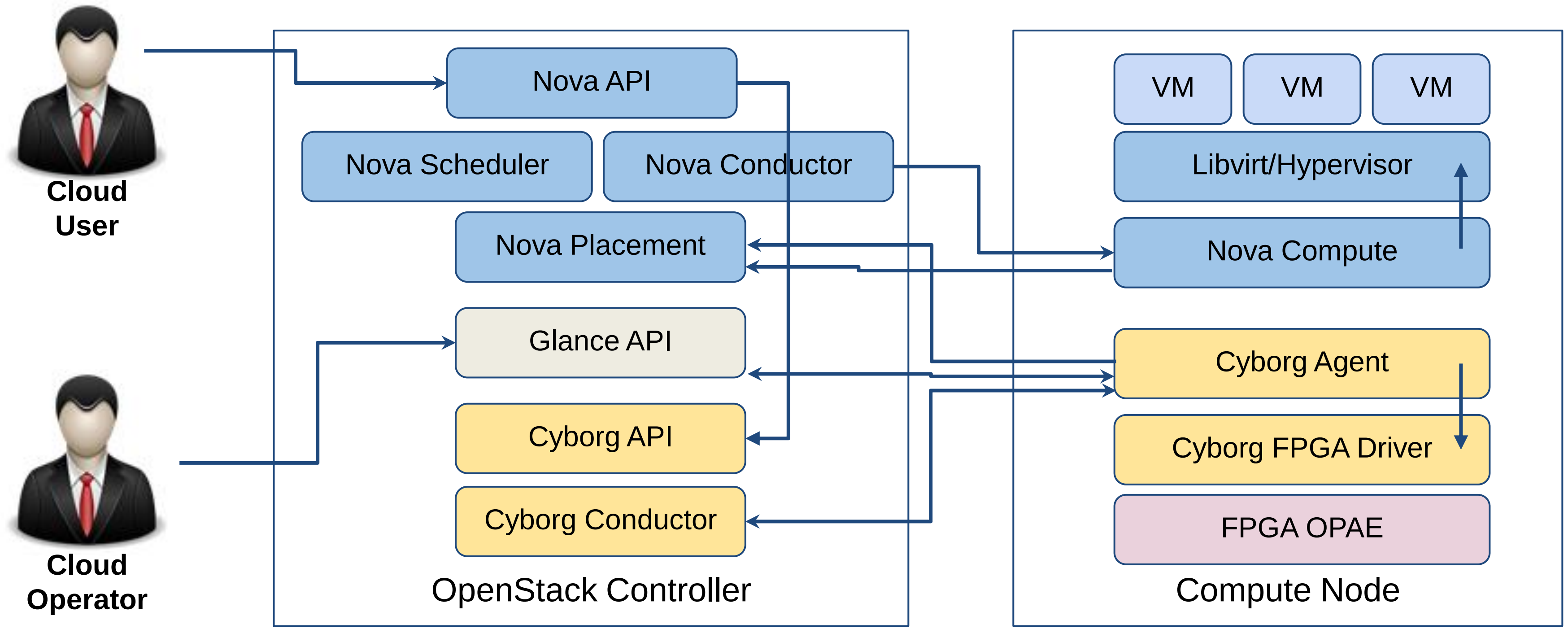
ACCELERATORS

VCA

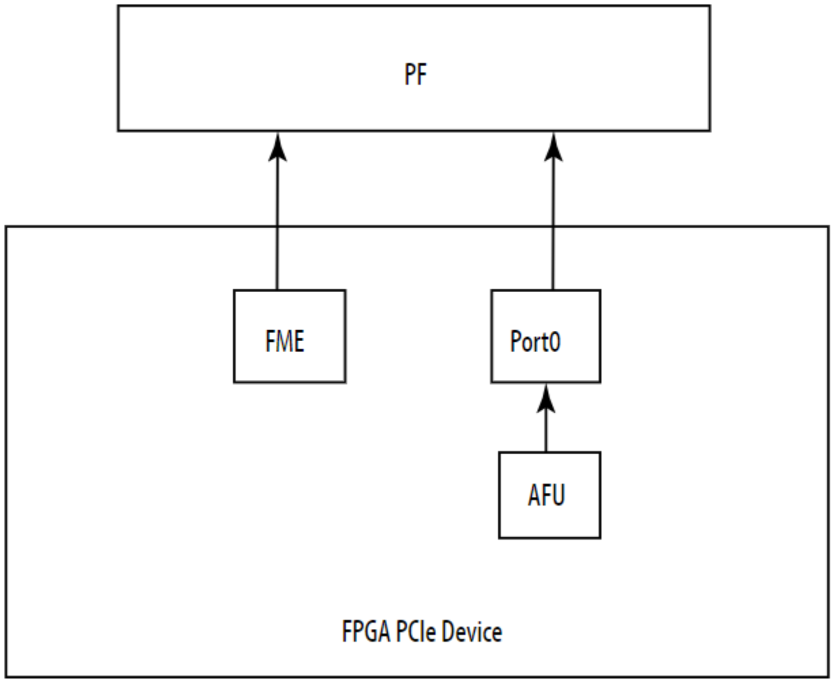
Intel® Visual Compute Accelerator VCA
For Emerging 4K/UHD, H.265 Transcode / Video Delivery Markets
VCA
intel



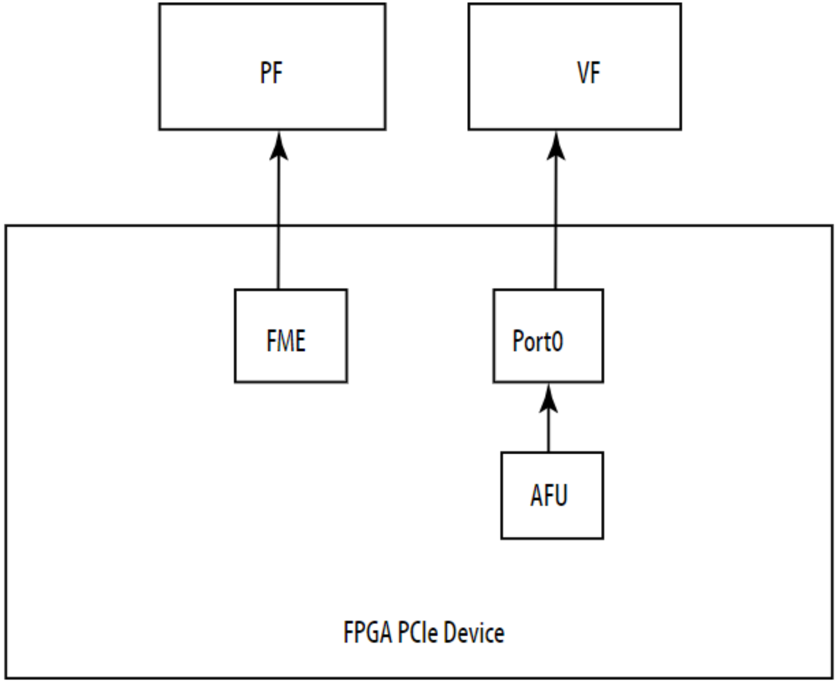
FPGA Orchestration - Architecture Components



Cloud Use Cases



FPGA PCIe Device



Virtualized FPGA PCIe Device

Cloud Use Cases

FPGA as a Service

Give me a region of type X

Programming security is paramount!

- Request-time Programming
  - User request includes bitstream ID
  - Infra programs bitstream
- Runtime Programming
  - VM requests bitstreams at runtime
  - Infra handles the requests

Accelerated Function as a Service

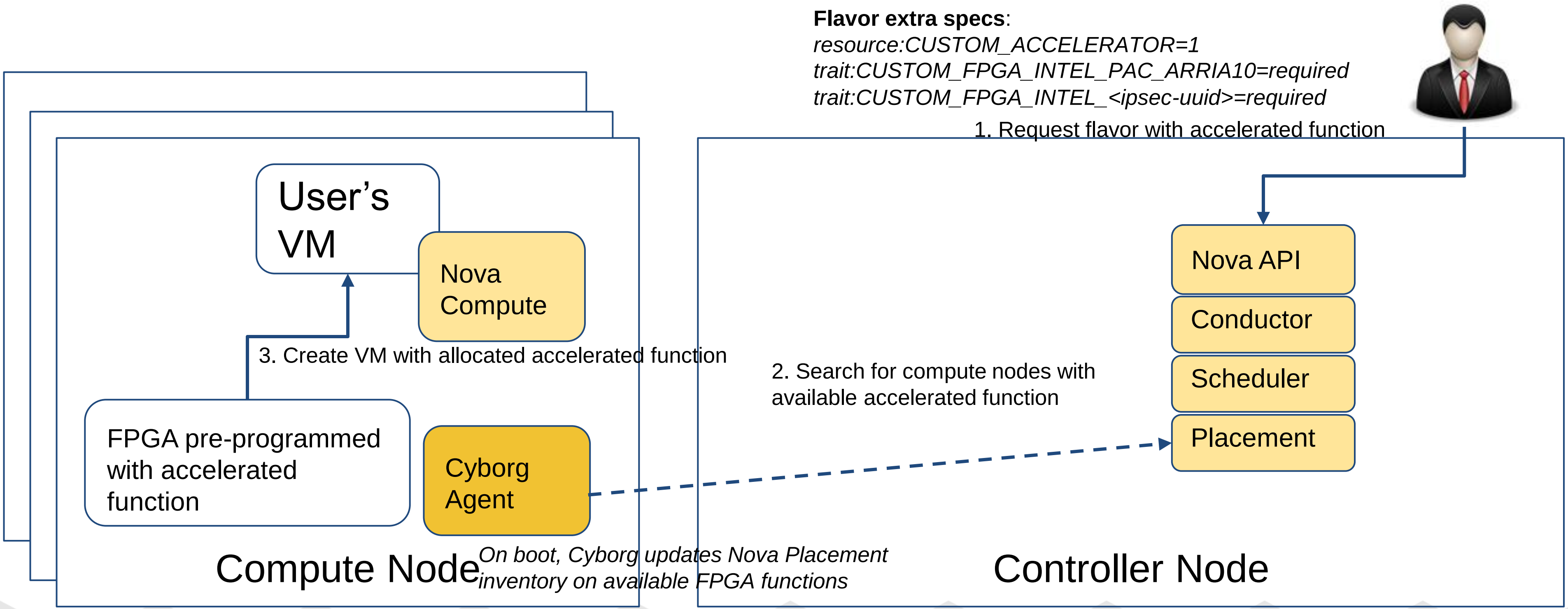
Give me an instance of ipsec

Need to say what device's drivers are in the VM

Operator Model:

- Pre-programmed: For Simplicity, Security, Peak provisioning …
- Orchestrator-programmed: If not available, program an unused region.

AFaaS: Pre-programmed



Flavor extra specs:
resource:CUSTOM_ACCELERATOR=1
trait:CUSTOM_FPGA_INTEL_PAC_ARRIA10=required
trait:CUSTOM_FPGA_INTEL_<ipsec-uuid>=required

1. Request flavor with accelerated function

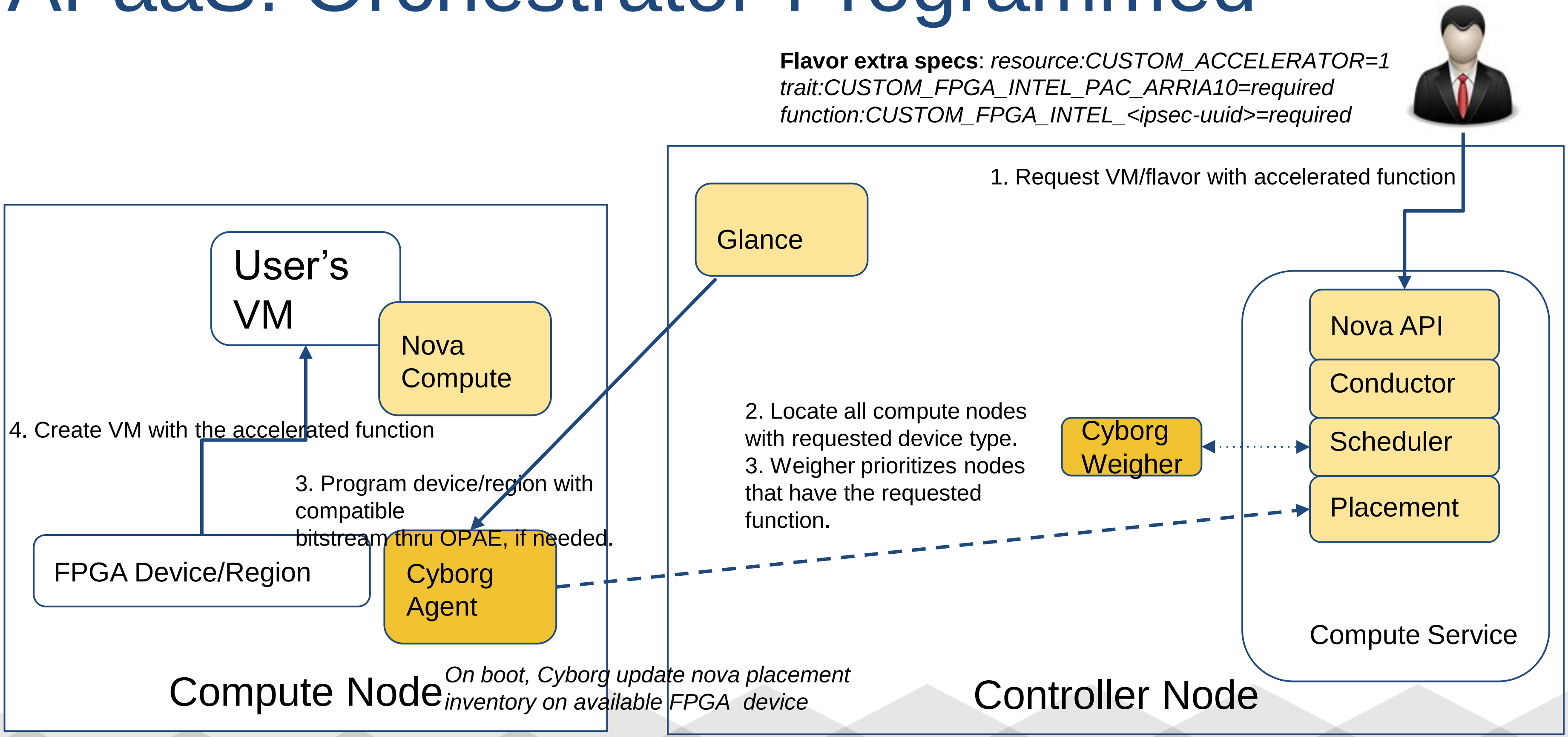
2. Search for compute nodes with available accelerated function

3. Create VM with allocated accelerated function

On boot, Cyborg updates Nova Placement inventory on available FPGA functions

AFaaS: Orchestrator-Programmed

Flavor extra specs: *resource:CUSTOM_ACCELERATOR=1*
trait:CUSTOM_FPGA_INTEL_PAC_ARRIA10=required
function:CUSTOM_FPGA_INTEL_<ipsec-uuid>=required



1. Request VM/flavor with accelerated function

2. Locate all compute nodes with requested device type.
3. Weigher prioritizes nodes that have the requested function.

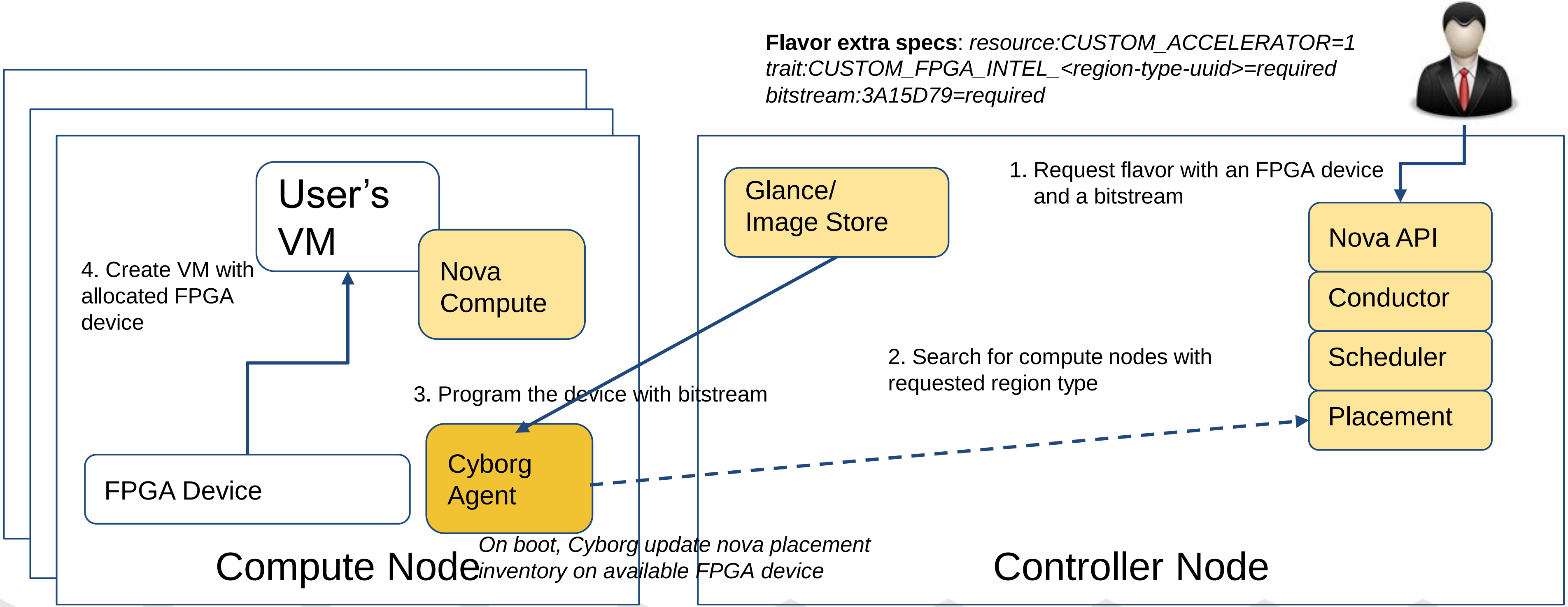
3. Program device/region with compatible bitstream thru OPAE, if needed.

4. Create VM with the accelerated function

On boot, Cyborg update nova placement inventory on available FPGA device

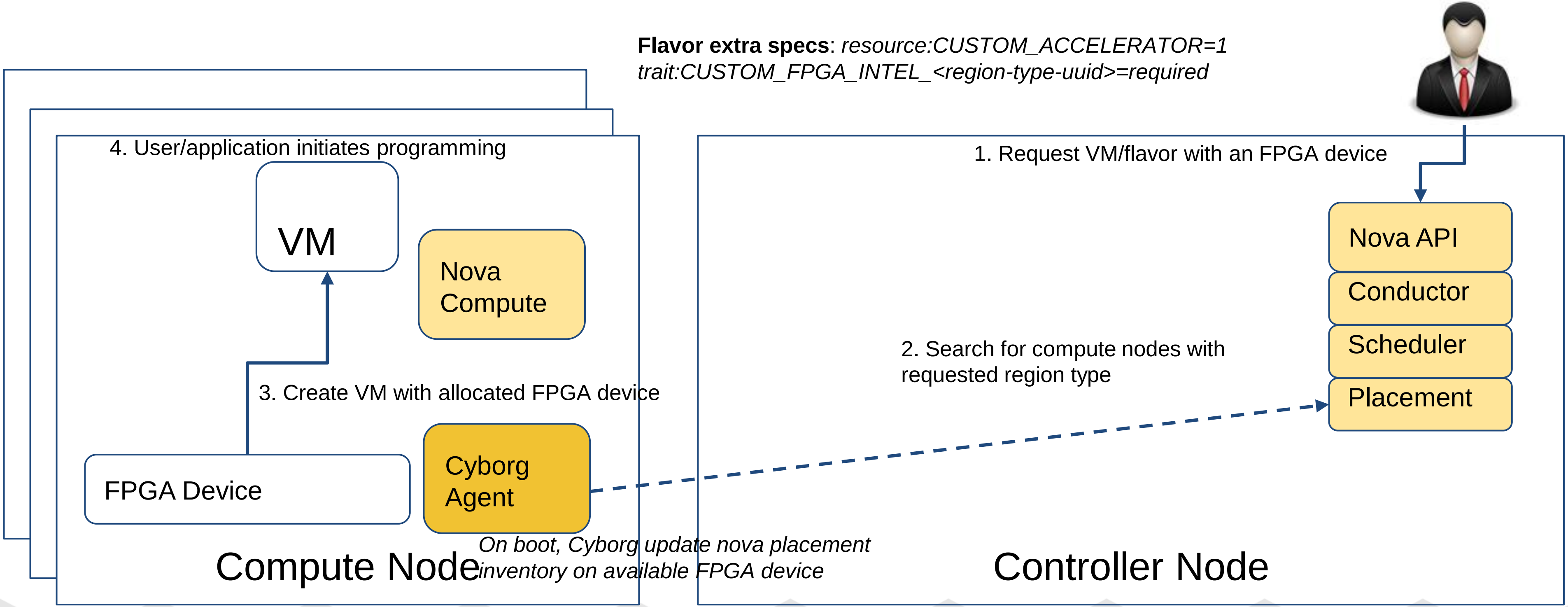
FPGAaaS: Request specifies a bitstream

Flavor extra specs: *resource:CUSTOM_ACCELERATOR=1*
trait:CUSTOM_FPGA_INTEL_<region-type-uuid>=required
bitstream:3A15D79=required


User's VM
Nova Compute
4. Create VM with allocated FPGA device
3. Program the device with bitstream
FPGA Device
Cyborg Agent
Compute Node
Glance/ Image Store
1. Request flavor with an FPGA device and a bitstream
Nova API
Conductor
Scheduler
Placement
2. Search for compute nodes with requested region type
Controller Node

On boot, Cyborg update nova placement inventory on available FPGA device

FPGAaaS: Bitstreams programmed at ru



Flavor extra specs: *resource:CUSTOM_ACCELERATOR=1*
trait:CUSTOM_FPGA_INTEL_<region-type-uuid>=required

4. User/application initiates programming

VM

Nova Compute

3. Create VM with allocated FPGA device

FPGA Device

Cyborg Agent

On boot, Cyborg update nova placement inventory on available FPGA device

Compute Node

1. Request VM/flavor with an FPGA device

Nova API

Conductor

Scheduler

Placement

2. Search for compute nodes with requested region type

Controller Node

QUESTIONS?

Ask on #openstack-cyborg IRC channel

Thank You