

EdgeFaaS Collaborative R&D for Open Edge Scenarios

Arm & China Unicom

Tina Tsou	Mingxuan Li
Arm	China Unicom

Connect. Collaborate. Accelerate.

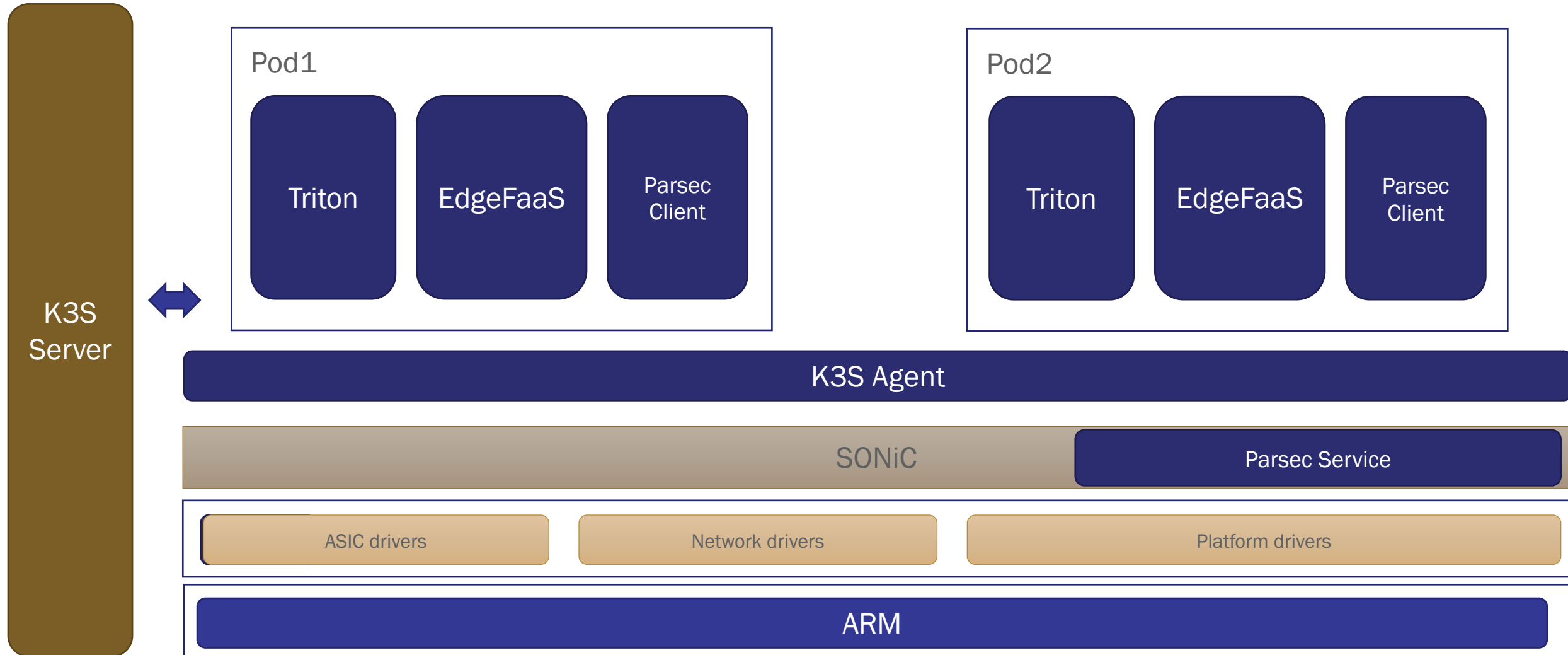


OPEN
Compute
Project®



OPEN
Compute
Project®

EdgeFaaS Collaborative R&D for Open Edge Scenarios SoW

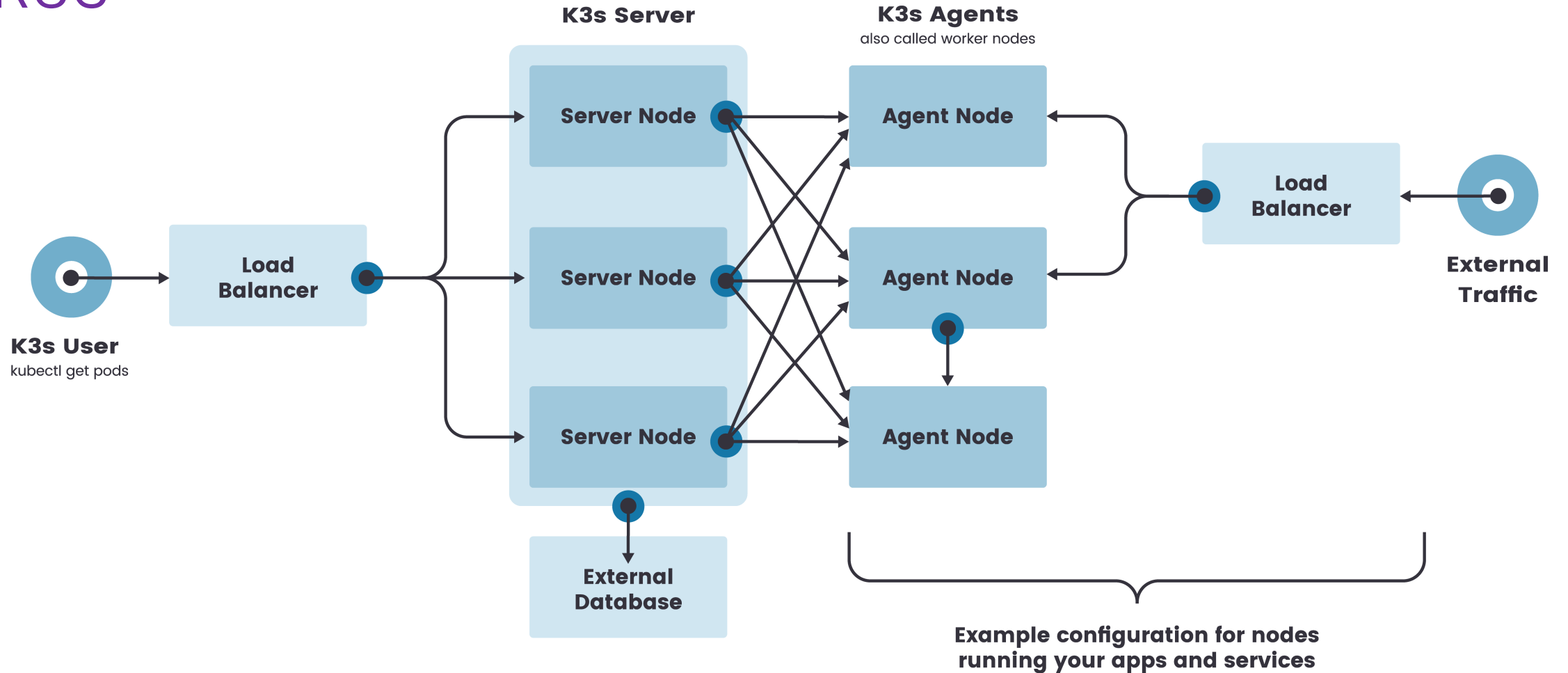


EdgeFaaS Collaborative R&D for Open Edge Scenarios SoW

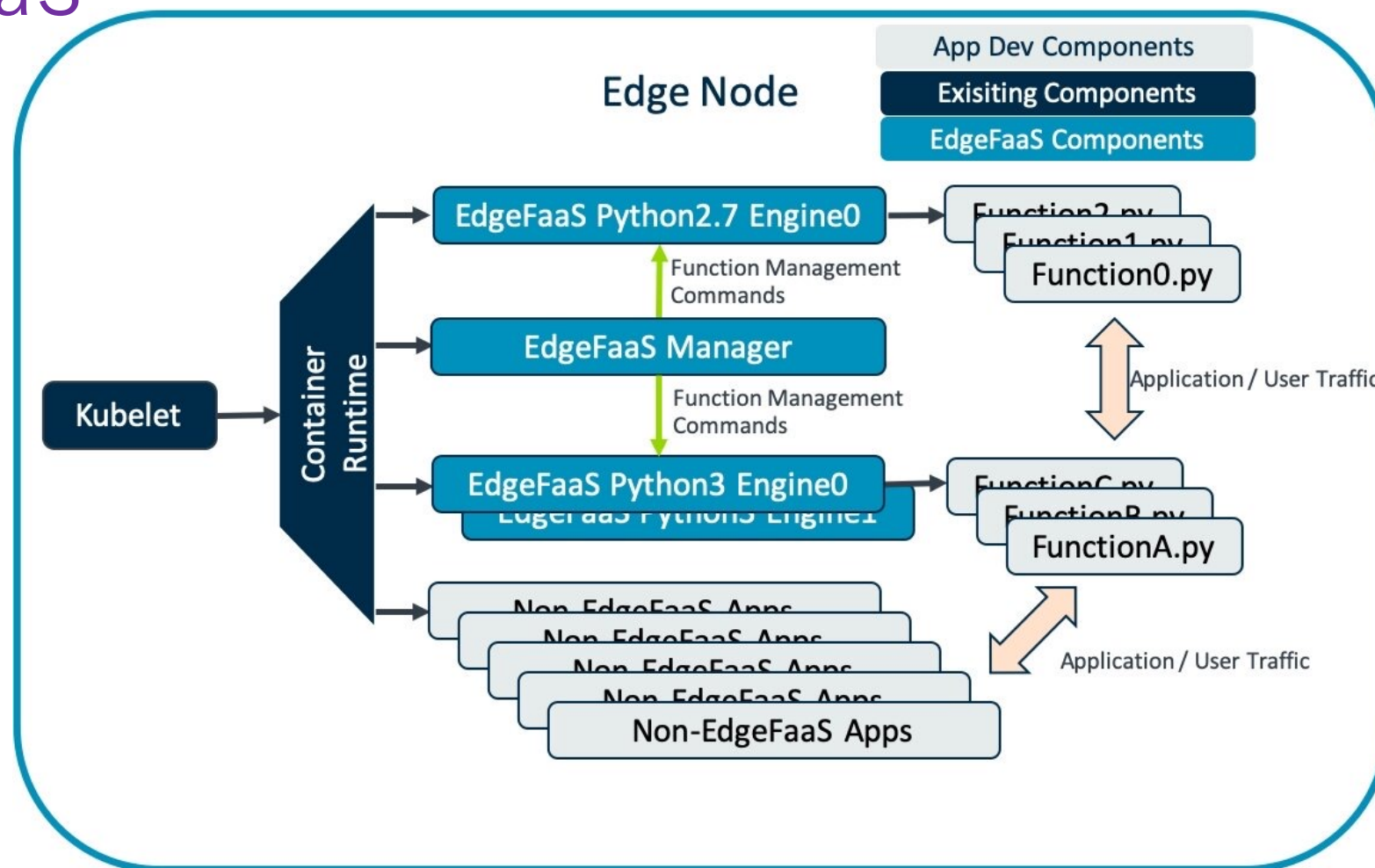
- Task1: Integrate K3S
 - 1.1 K3S porting on SONiC platform.
 - 1.2 Run K3S server and K3S client on different SONiC devices.
 - 1.3 Create and run 2 Pods with K3S on one SONiC device.
- Task2: Integrate Parsec
 - 2.1 Check in parsec code for Akraino SW release5.(CI/CD)
 - 2.2 Port parsec to SONiC platform
 - 2.3 Run parsec service directly on SONiC.
 - 2.4 Run parsec client on a K3S pod, and make parsec client communicate success with parsec service.
 - 2.5 Run multi parsec clients on K3S pods, and make parsec clients communicate success with a same parsec service.
- Task3: Integrate EdgeFaaS
 - 3.1 Check in EdgeFaas code for Akraino.(CI/CD)
 - 3.2 Port edgeFaaS to SONiC platform.
 - 3.2 Run two edgeFaaS functions on different K3S pods, and make the edgeFaaS functions communicate success.
- Task4: Integrate Triton
 - 4.1 Check in Triton code for Akraino release6.(CI/CD)
 - 4.2 Port Triton to SONiC platform.
 - 4.3 Run two triton test case on different K3S pods.



K3S

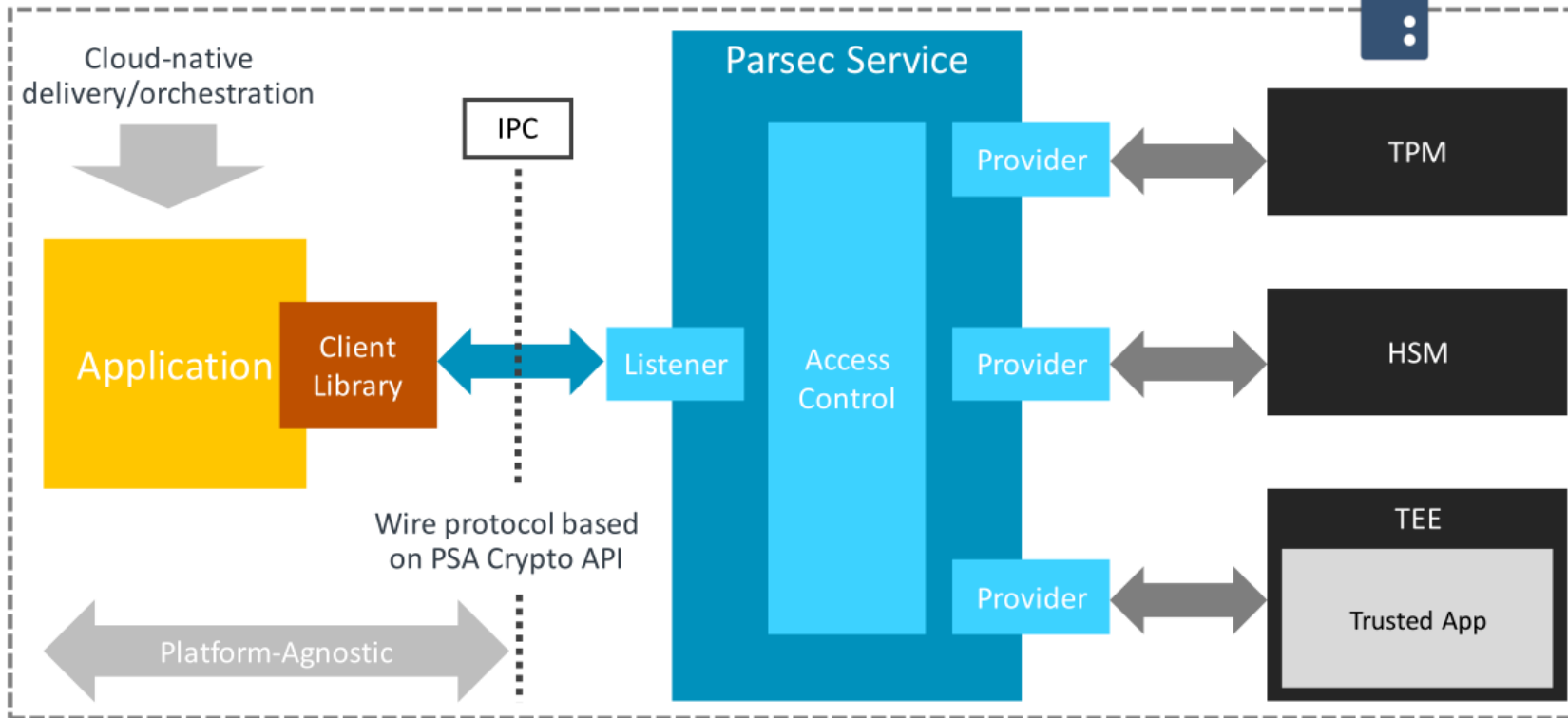


EdgeFaaS



Parsec

Service Architecture



5

© 2021 Parsec Project Contributors



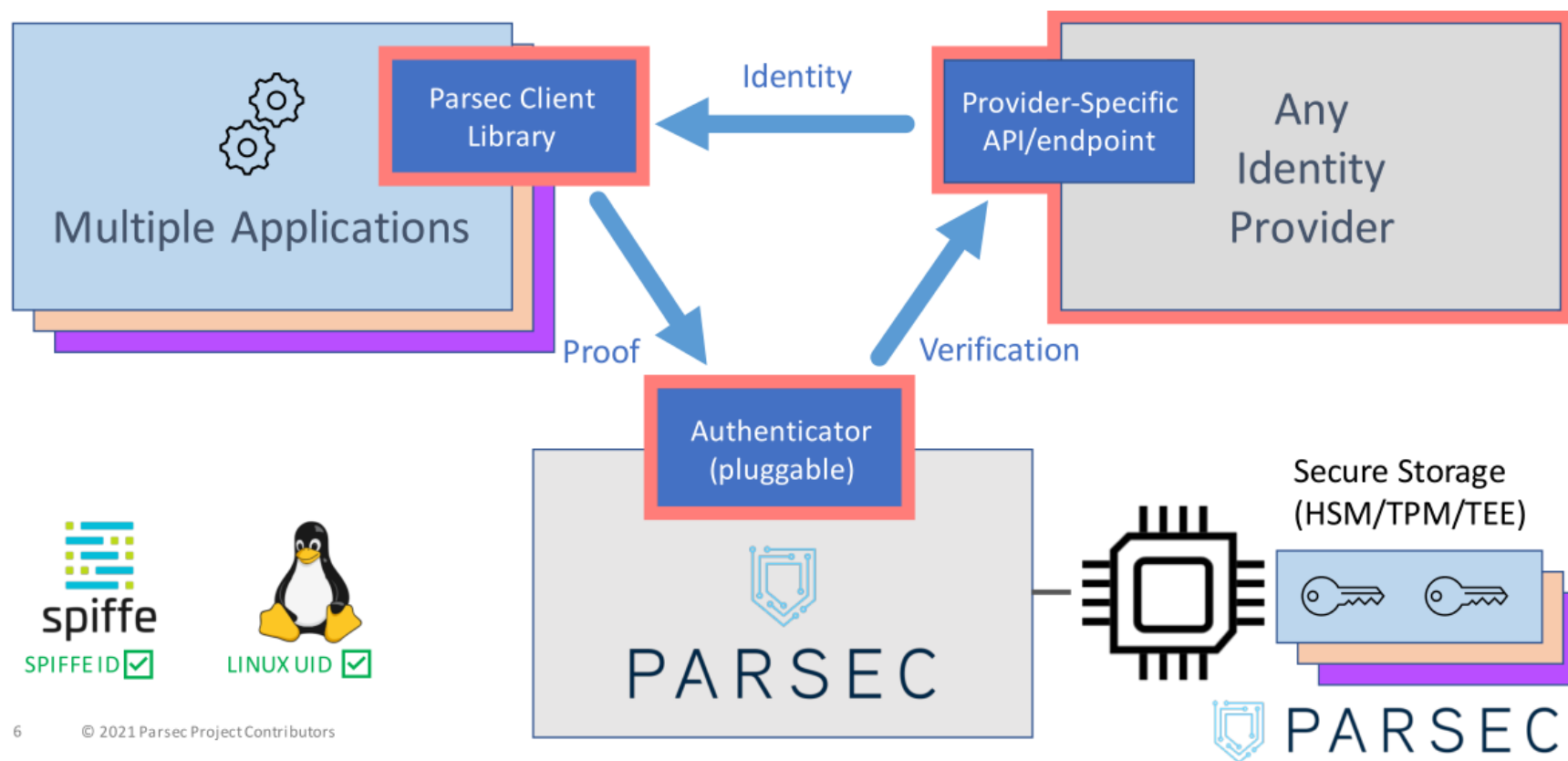
Parsec is an open-source initiative to provide access to secure services and cryptographic operations in a platform-agnostic way, with a key goal being to make it easy for developers to build and run secure applications.

Parsec runs as a service on an edge device, responding to client requests from applications running on the host that are posted to the transport endpoint by one of the language-native client libraries. These are key to Parsec's native support in various languages.

Connect. Collaborate. Accelerate.

Parsec

Application Identity and Multi-Tenancy

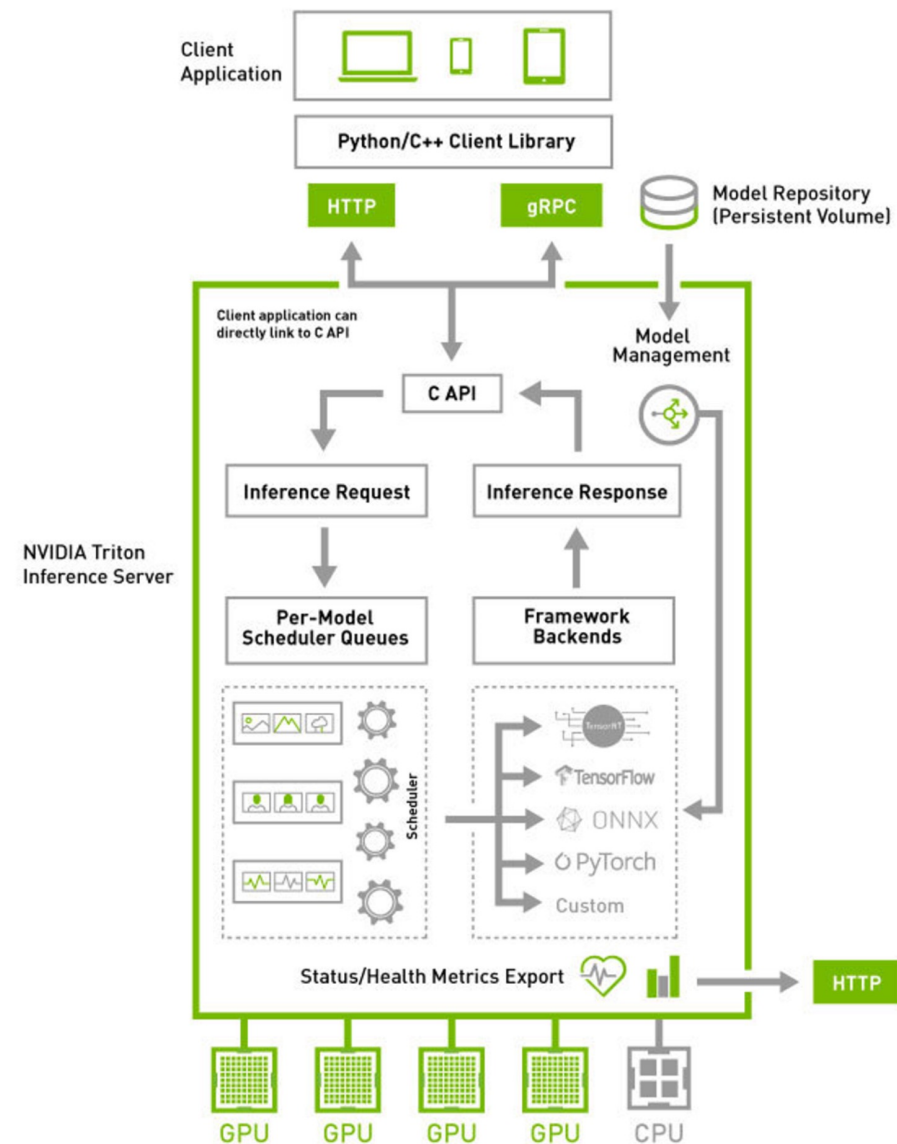


Triton for Embedded Devices

› Triton is supported on Arm embedded platforms: Jetson Xavier, Jetson Nano, Raspberry Pi

› Supported Features:

- TensorFlow 2.4, TensorFlow 1.15.5, TensorRT 7.1, ONNX RT 1.8, and Custom backends
- Model Ensembling
- CPU or GPU support
- Shared Library and C API
- C++ and Python client libraries and examples





OPEN
Compute
Project®

China Unicom Use Case

➤ EdgeFaaS for Open Edge Scenarios

About me



Github: <https://github.com/limx59>

Mingxuan Li, Senior engineer of the Future Network Research Center of China Unicom Research Institute, mainly engaged in research on cloud native, cloud computing, computing power network, big data, and mobile Internet. I mainly track open source projects such as OpenStack, CNCF, LFN, and participate in ITU international standards, CCSA industry standards, etc. At present, I have published more than 20 papers, applied for more than 20 patents, more than 10 authorized patents, applied for 5 software copyrights, and jointly led the formulation of more than 10 industry standards.

E-mail: limx59@gmail.com

limx59@chinaunicom.cn

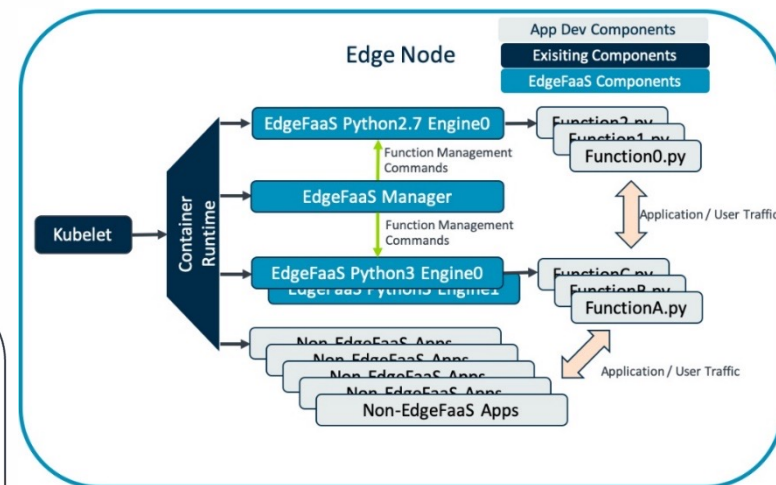
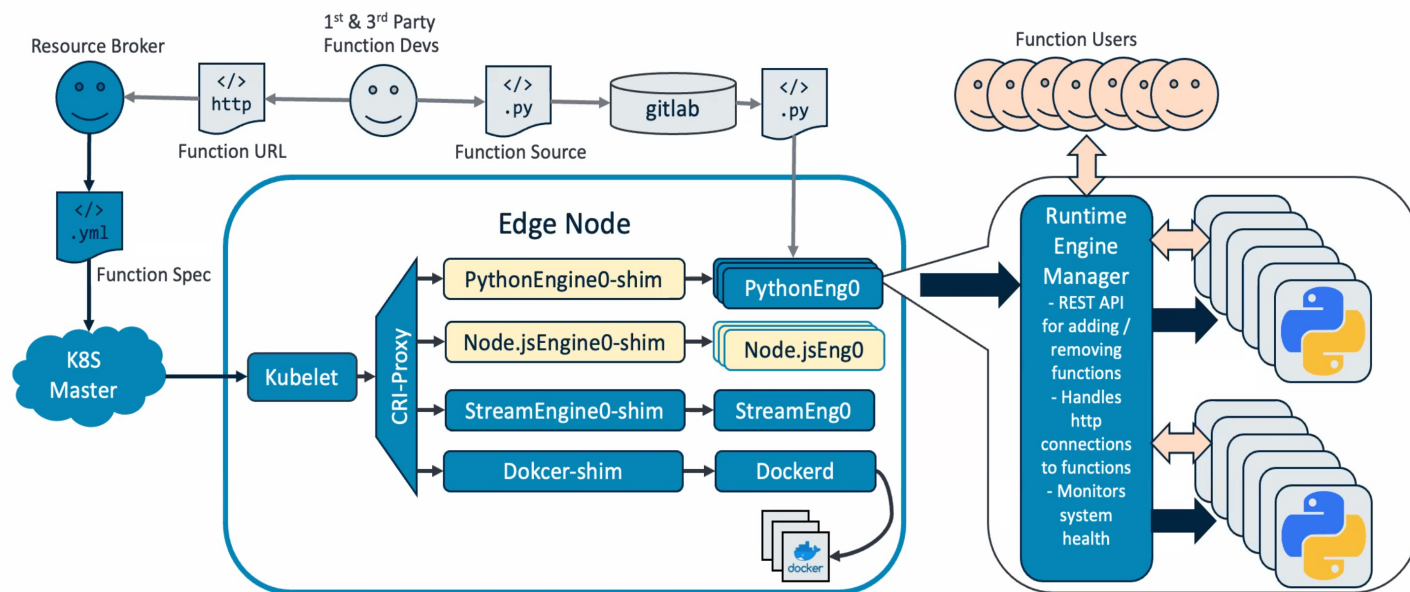
Connect. Collaborate. Accelerate.

Scenarios Analysis

EdgeFaaS is to open edge computing based on "function as a service" capabilities, which is different from traditional MEP in:

- **Different calling ways :** Compared with PaaS in the application store, through function as a service, and through function calls to achieve capability opening.
- **Lightweight:** OpenFaaS and K3S are integrated to realize **"Computing Power as a Service"** at the edge.
- **Different billing models :** Realize computing power transactions according to the function call billing model, breaking the traditional virtual machine leasing method

Technical architecture



EdgeFaaS system architecture

Connect. Collaborate. Accelerate.

Scenarios Requirements

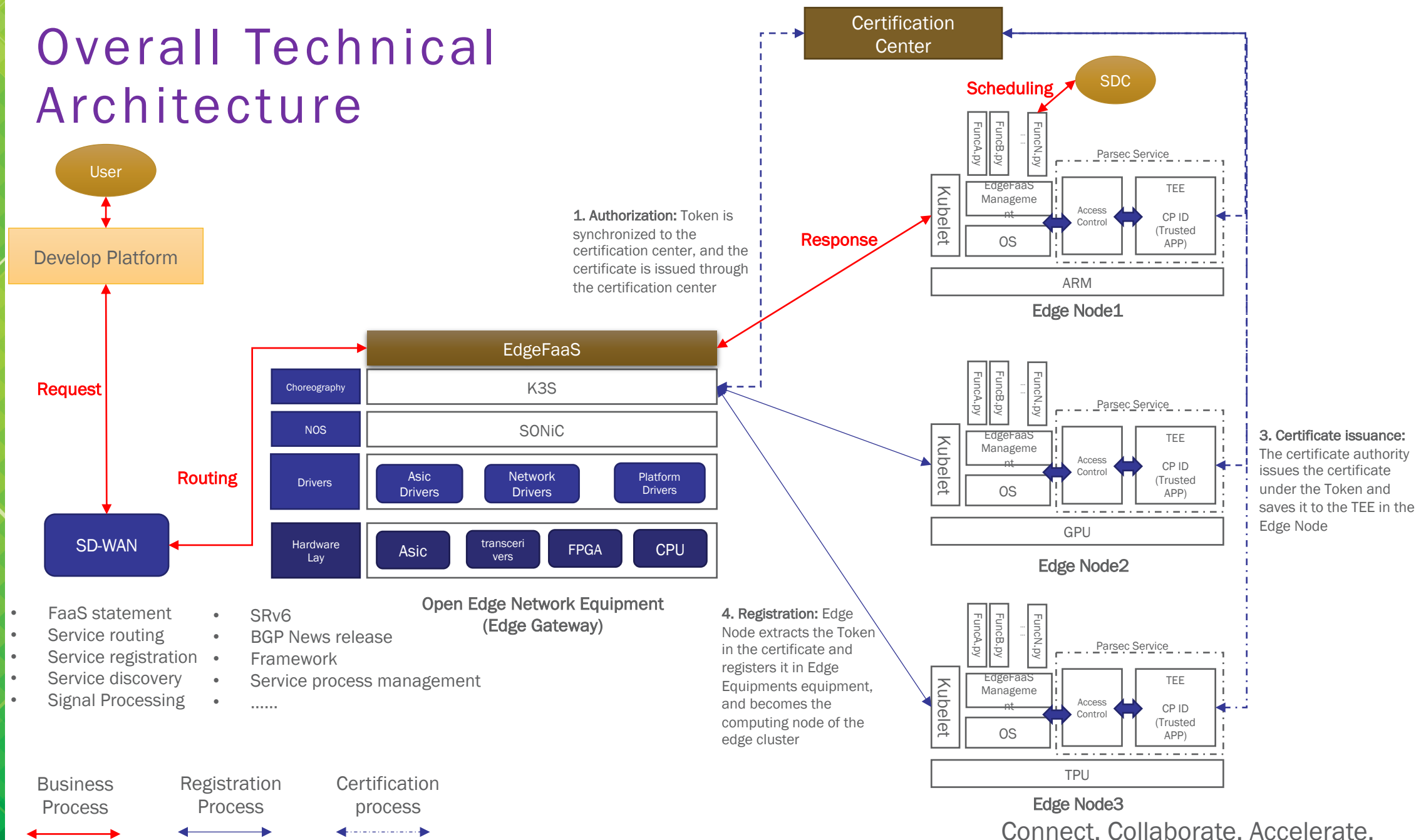
- › Scenario 1: EdgeFaaS on SONiC
 - › 1.1 K3S porting on SONiC platform.
 - › 1.2 EdgeFaaS Open Edge Gateway API
 - › 1.3 SONiC declares function API in SD-WAN network.

- › Scenario 2: Computing Power Certification on Parsec at the Edge
 - › 2.1 Edge Node implements certificate authentication
 - › 2.2 Certificate preservation and anti-tampering based on Parsec
 - › 2.3 Edge Node is registered to K3S with certificate

- › Scenario 3: Computing power service based on EdgeFaas
 - › 3.1 Realize the function call of edge computing power based on EdgeFaaS.
 - › 3.2 Realization of billing for function operation based on license.
 - › 3.3 Realize cloud-side collaboration based on EdgeFaaS.

- › Scenario 4: SDC Smart Monitor based on Triton
 - › 4.1 Realize AI algorithm based on Triton running on SDC
 - › 4.2 Realization of GPU on the Nvidia nano sharing process based on Triton

Overall Technical Architecture





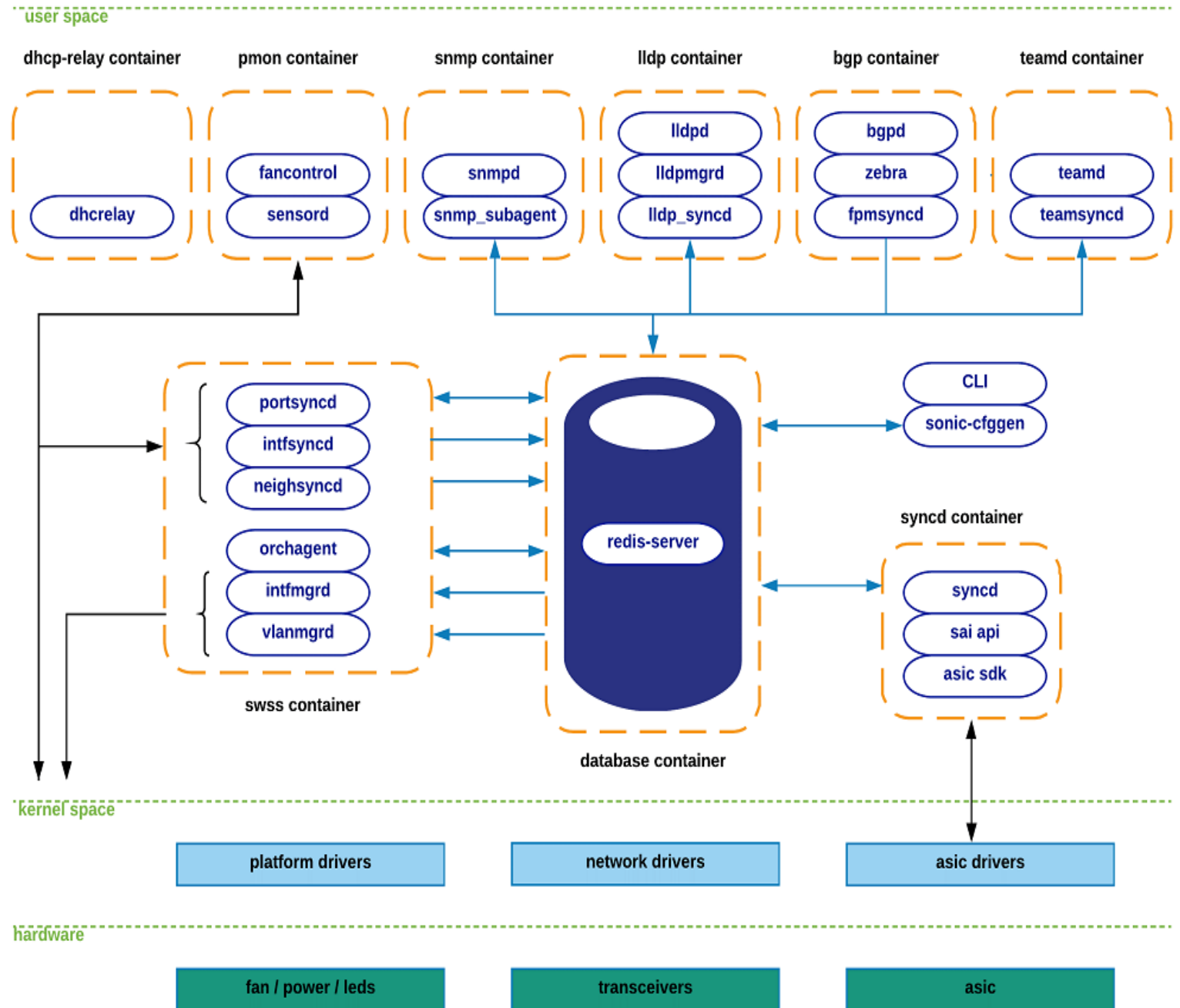
SONiC

SONiC is an open source network operating system based on Linux that runs on switches from multiple vendors and ASICs.

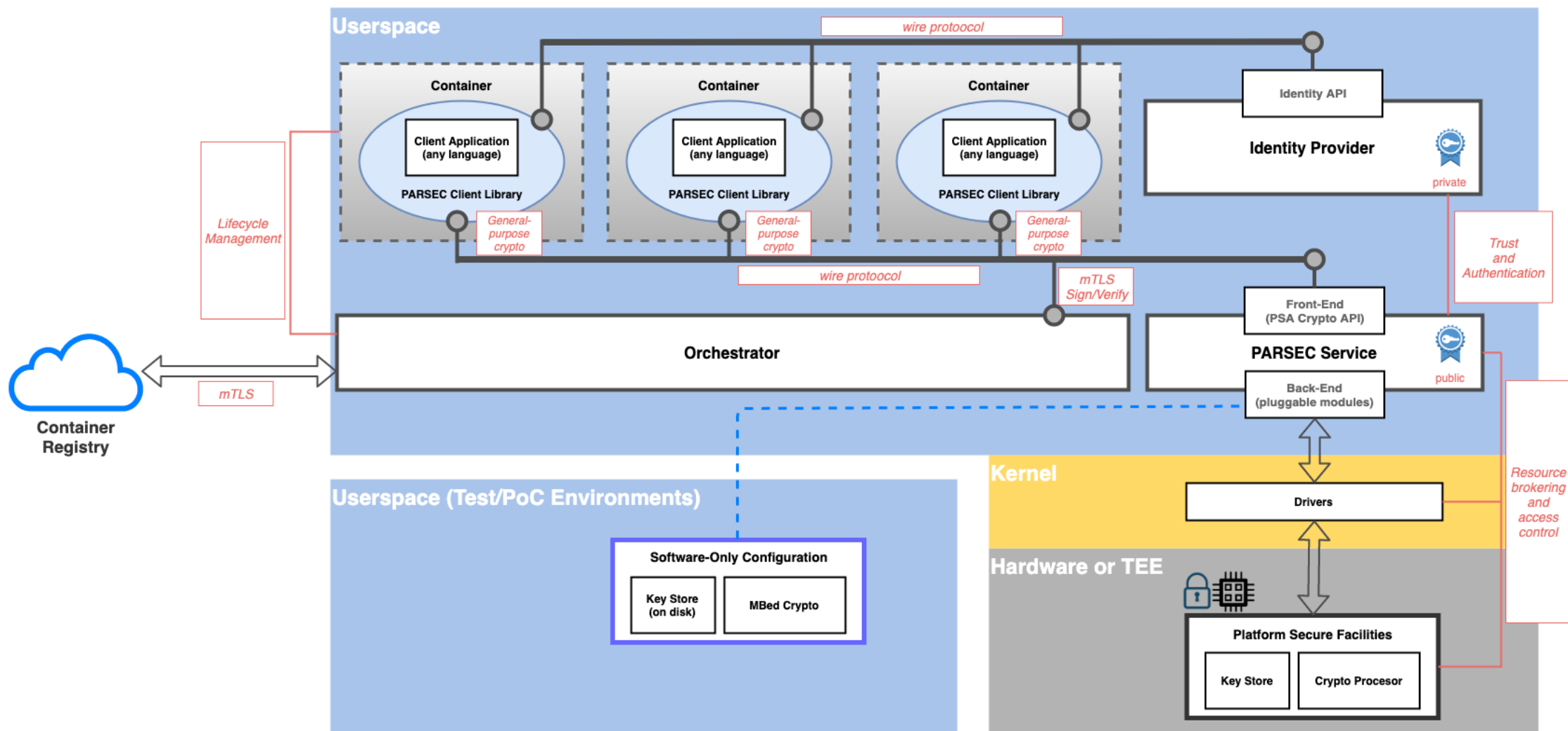
SONiC offers a full-suite of network functionality, like BGP and RDMA, that has been production-hardened in the data centers of some of the largest cloud-service providers.

SONiC offers teams the flexibility to create the network solutions they need while leveraging the collective strength of a large ecosystem and community.

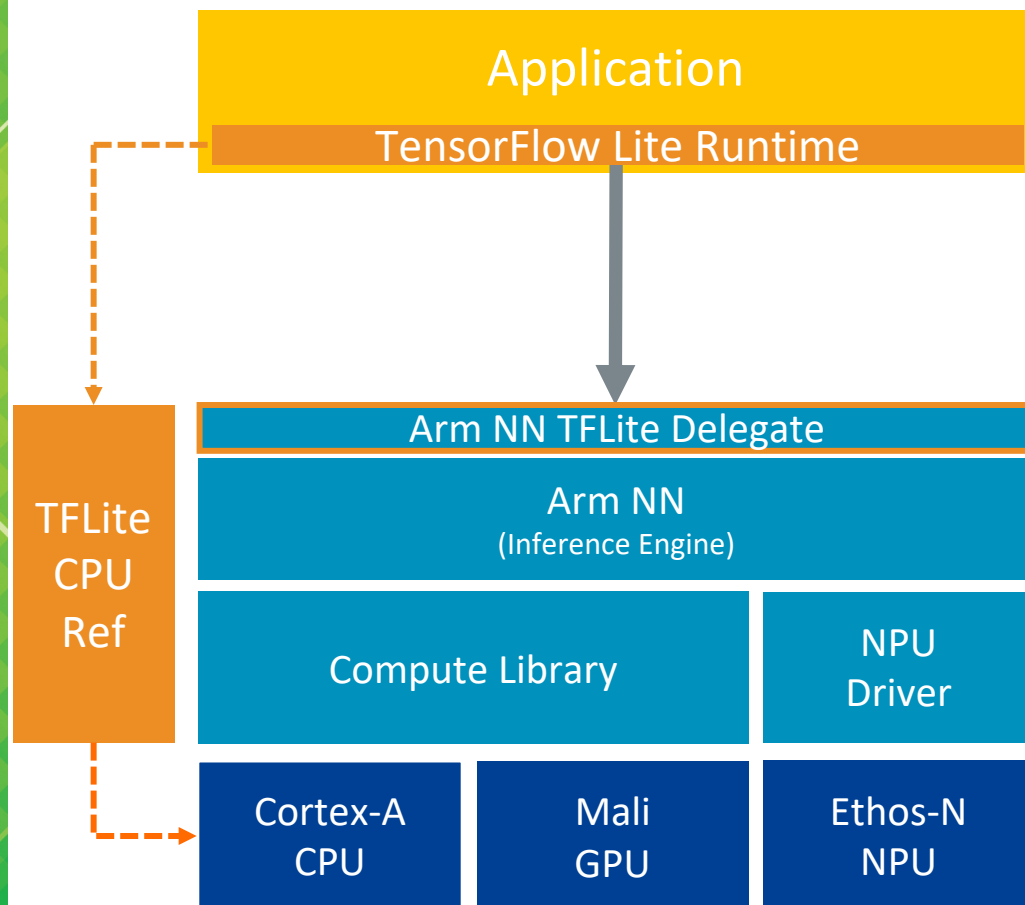
<https://github.com/Azure/SONiC>



Parsec Certification



TensorFlow Lite on SDC



- The Arm NN TFLite Delegate can plug directly into the TFLite Runtime
- Greater flexibility for Android developers over NNAPI
- Arm specific CPU and GPU optimizations accessible through Arm NN
- All TFLite models can be supported:
 - Key operators accelerated through Arm NN and ACL
 - Unsupported operators processed through TFLite CPU Ref
- Also suitable for non-Android environments

- Object Detection based on TFLite running on Raspberry Pi
 - Recognition rate: 66%



Connect. Collaborate. Accelerate.

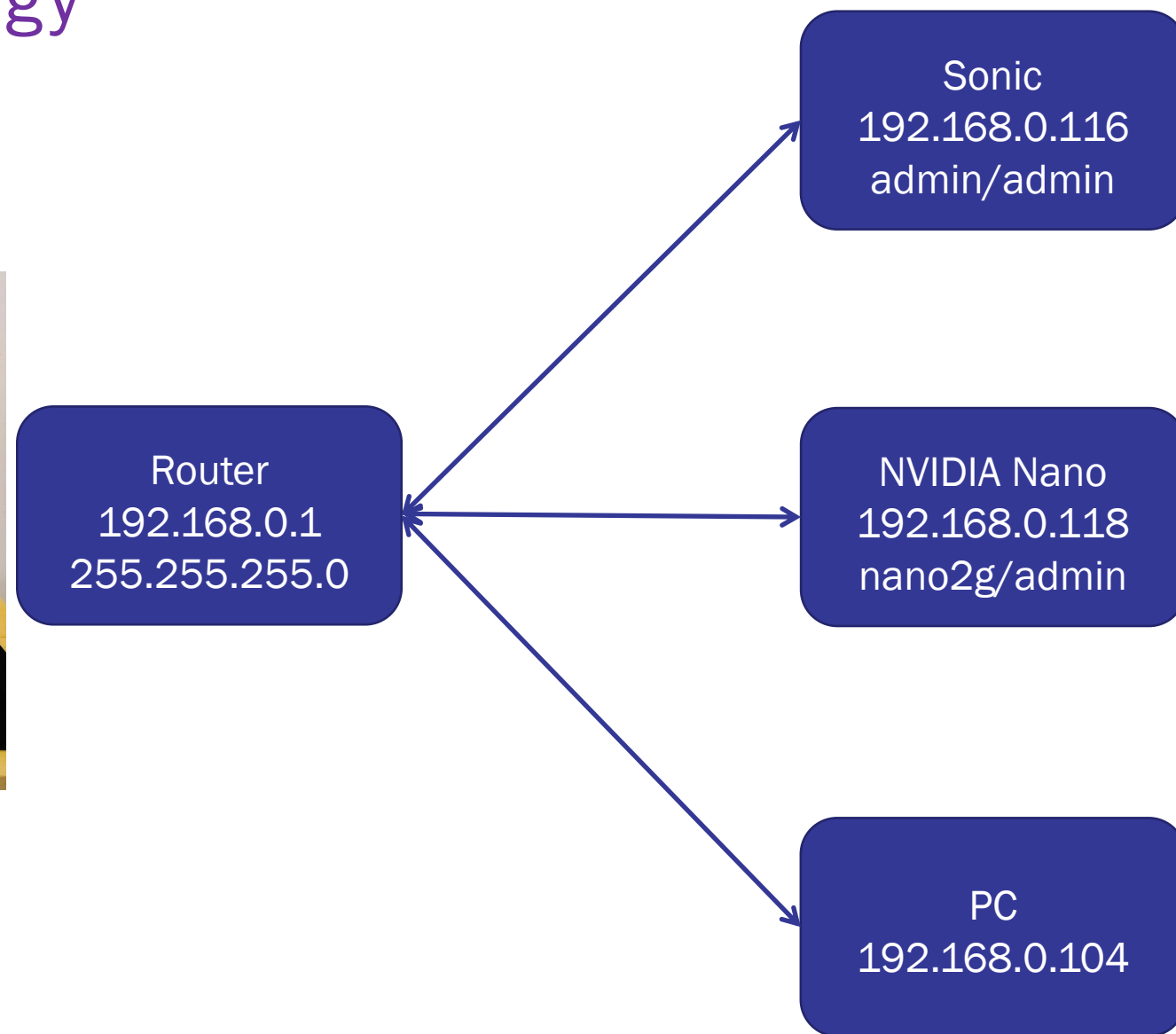


OPEN
Compute
Project®

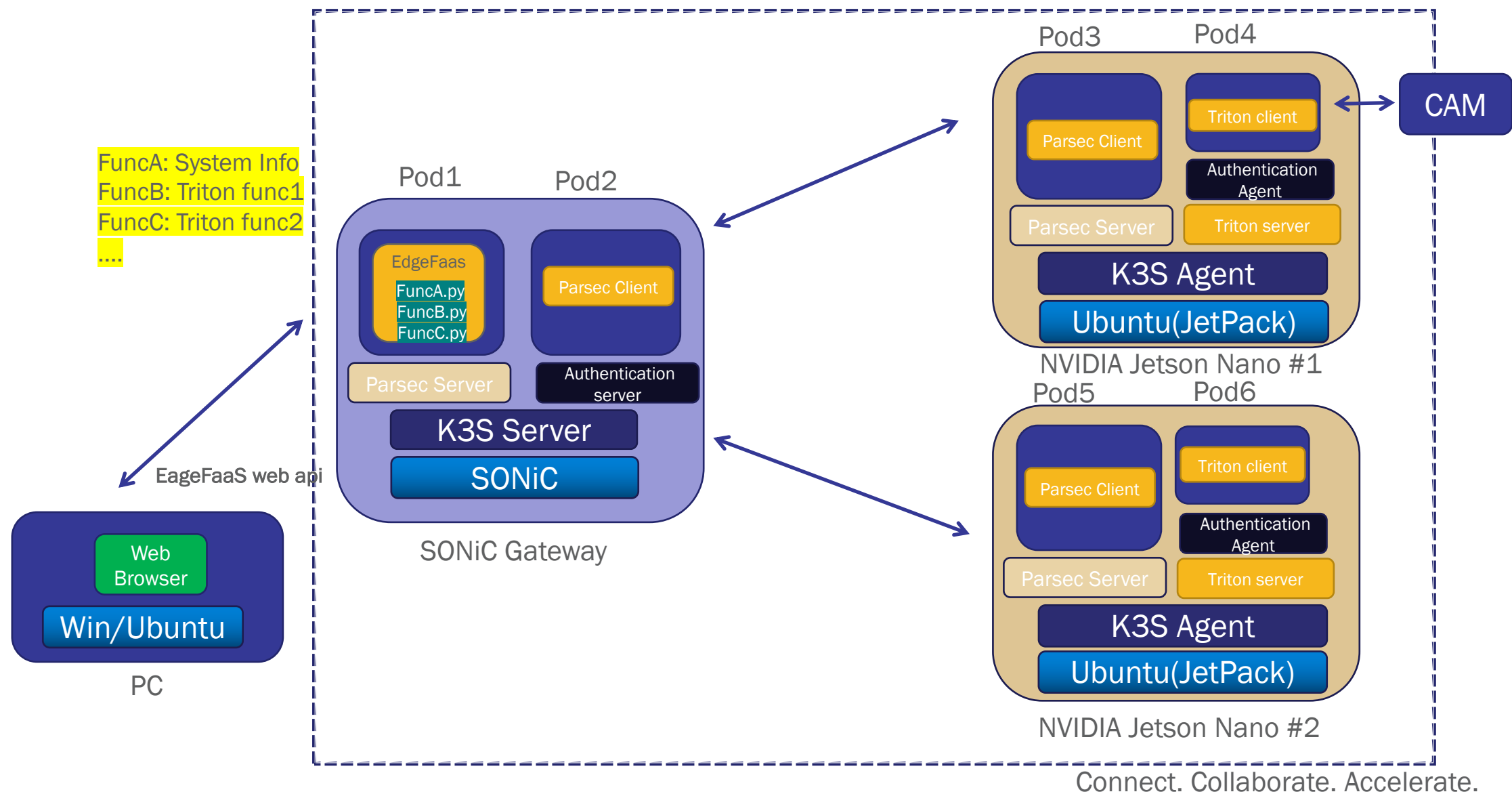
Implementation

➤ EdgeFaaS on SONiC networking test

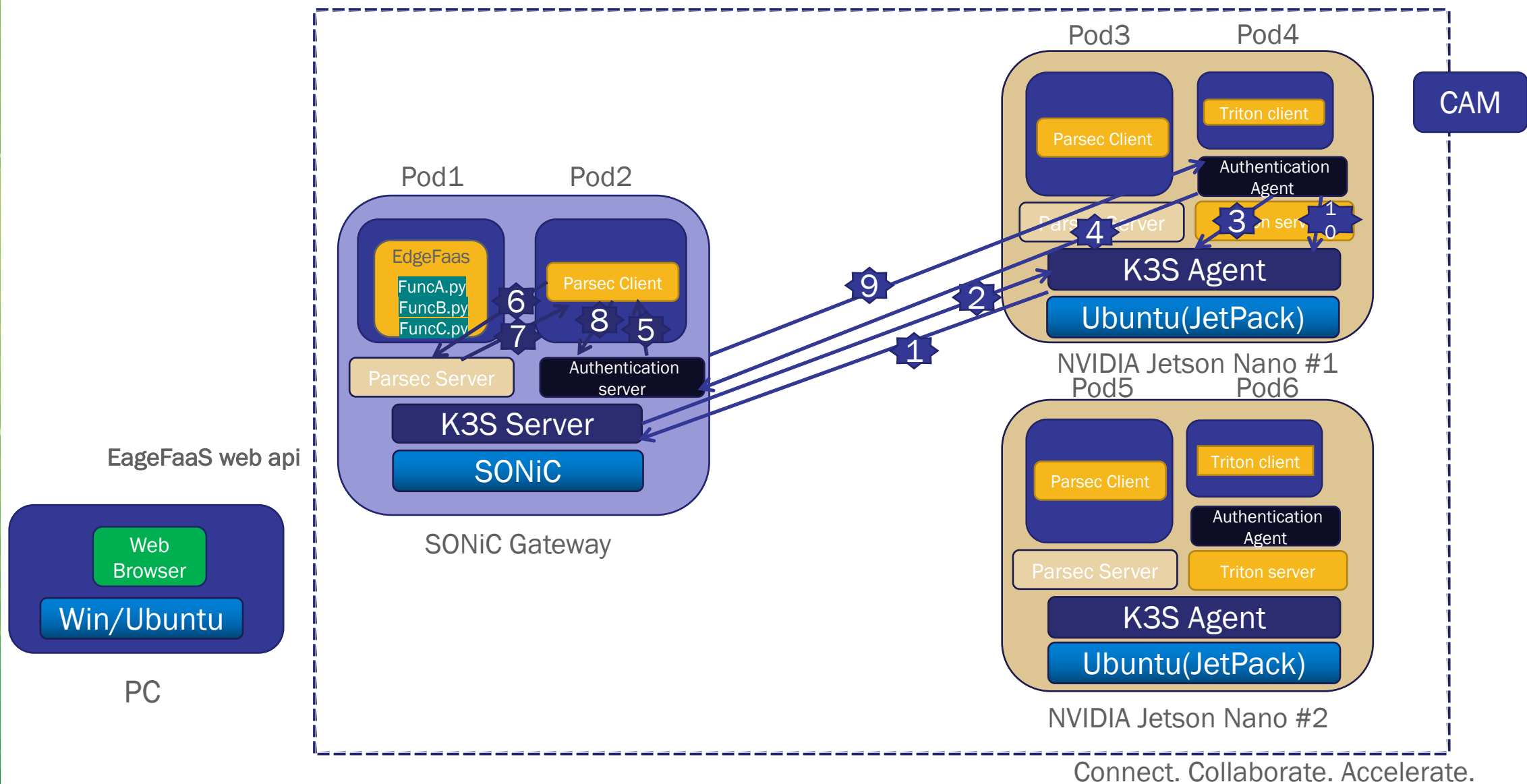
Test Network Topology



Demo App Design

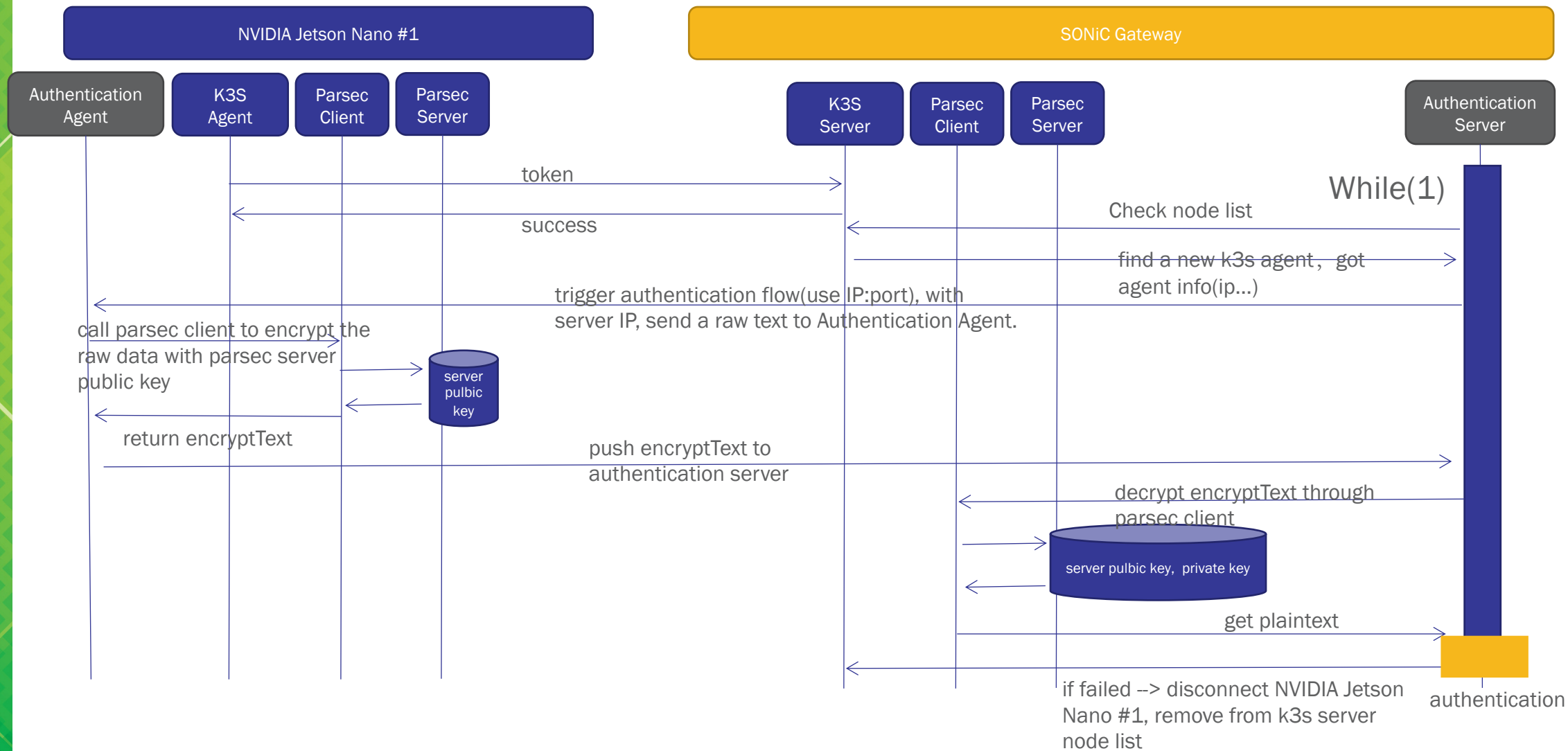


Demo1: NVIDIA Jetson Nano #1 joins network

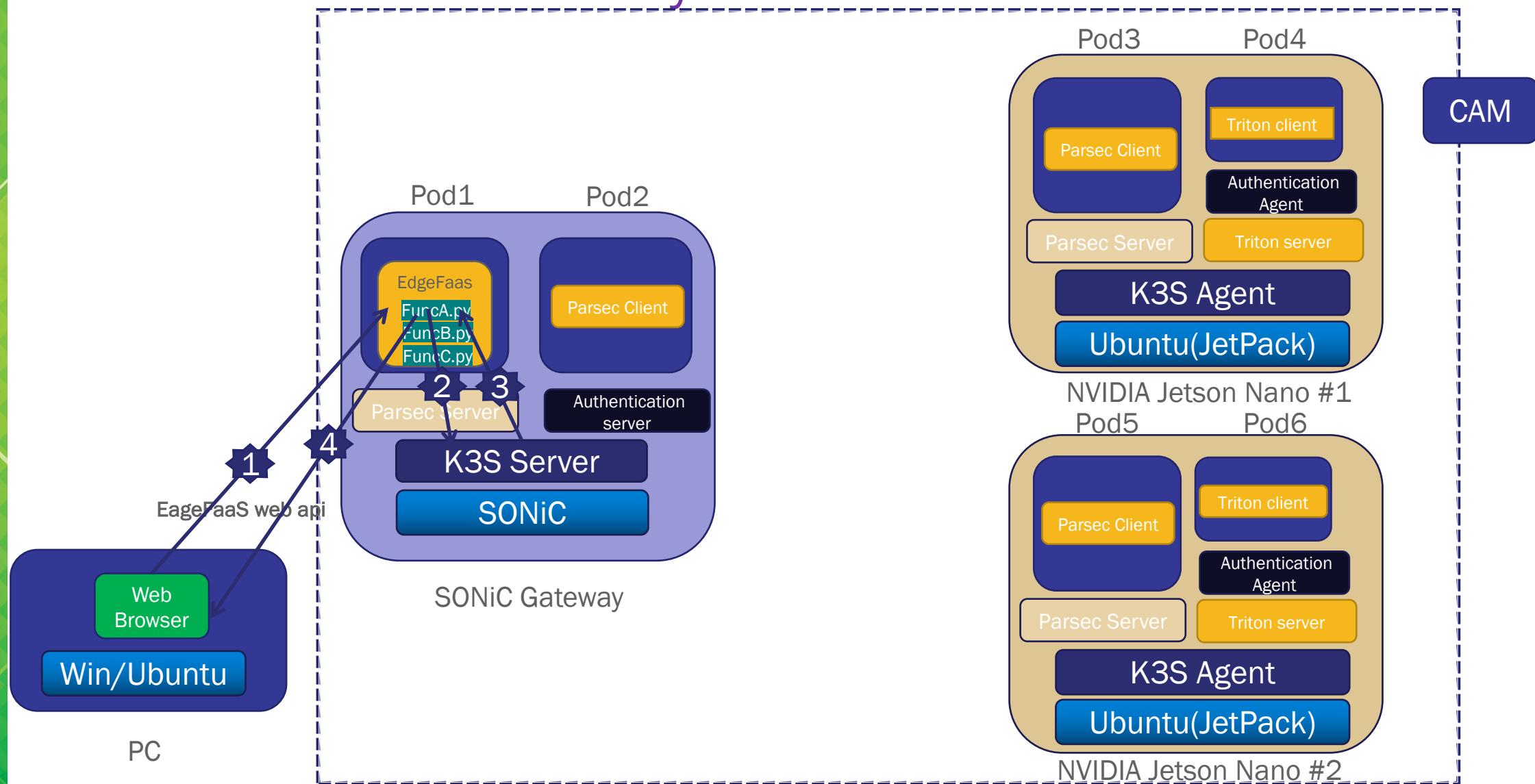




Demo1: NVIDIA Jetson Nano #1 joins network

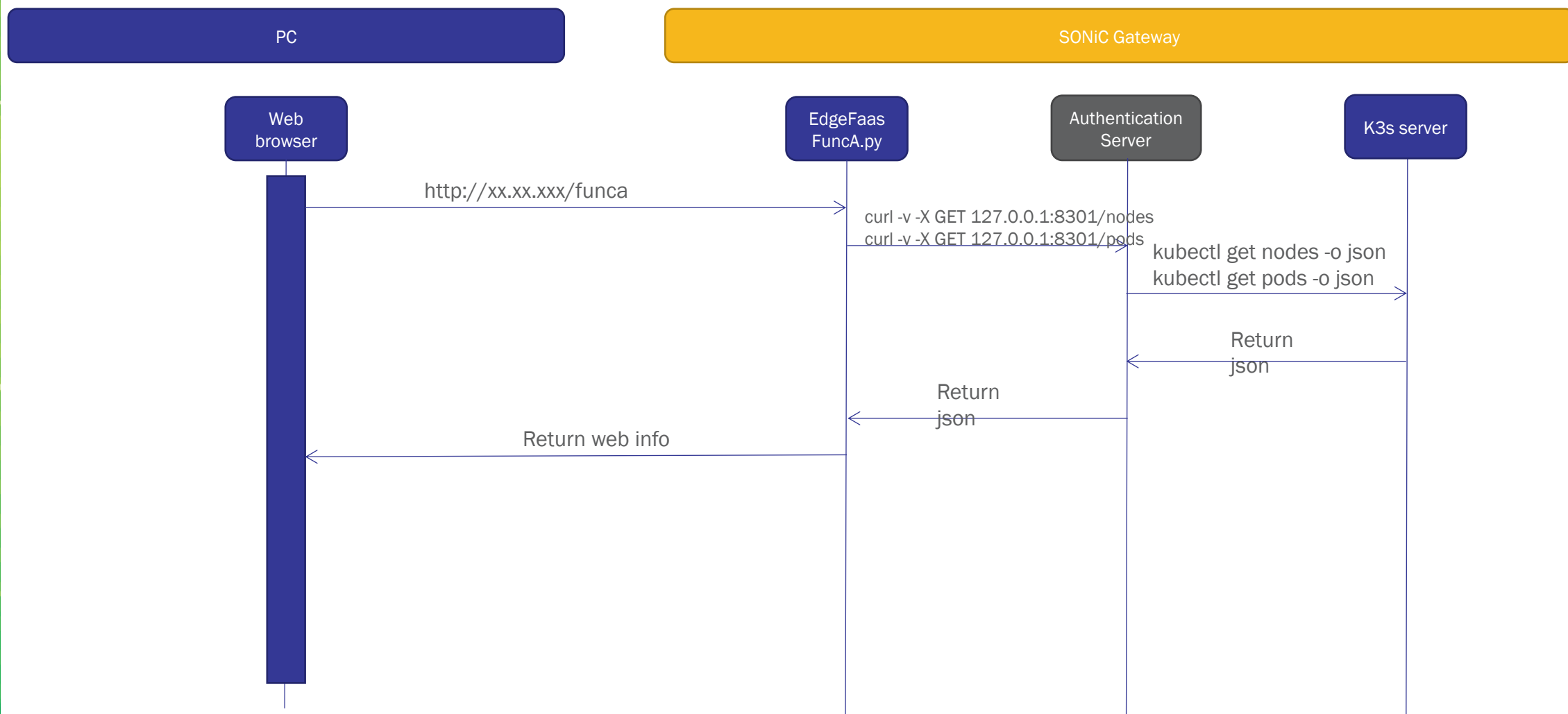


Demo2: Get cluster system info

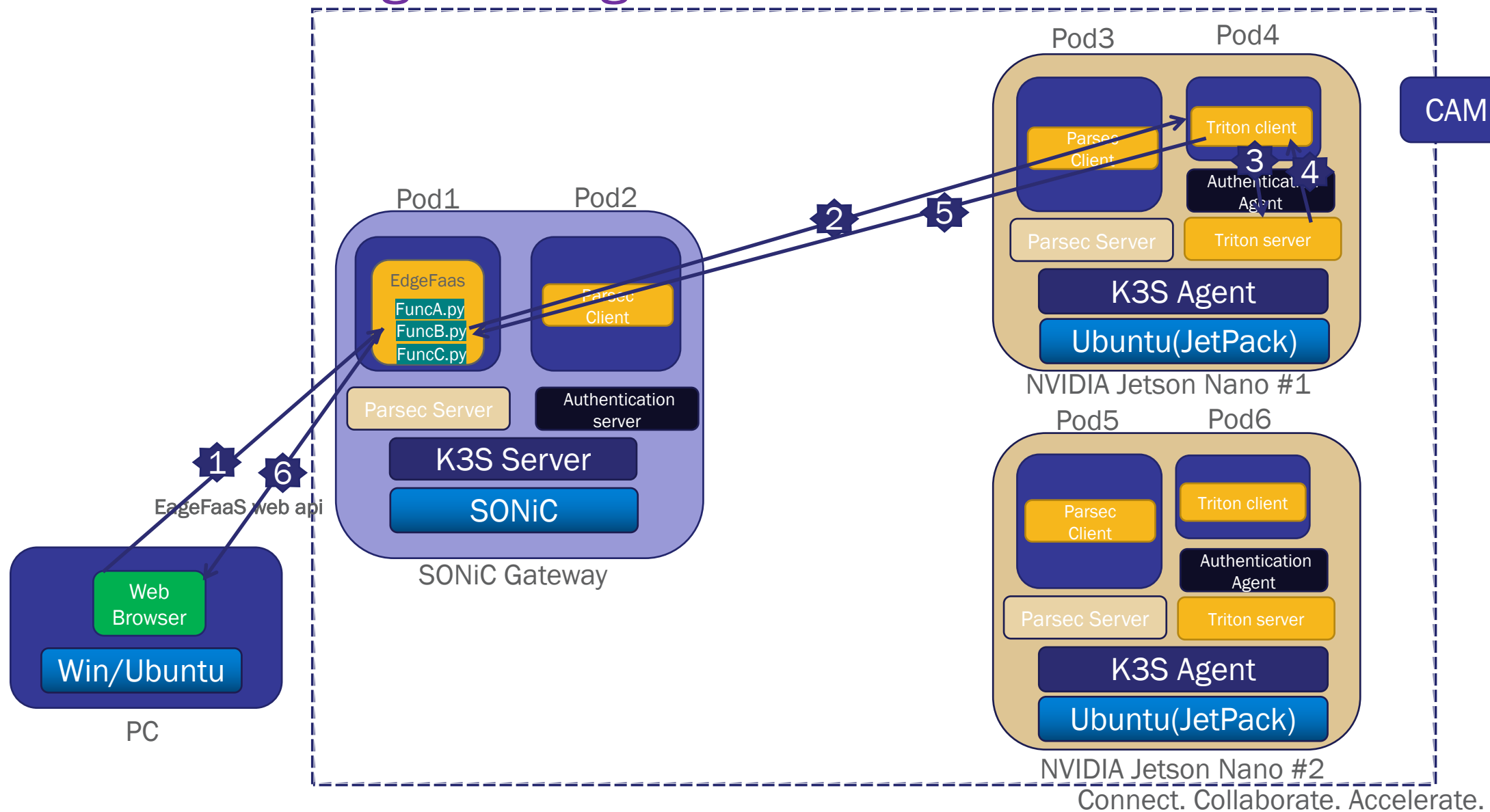


CAM

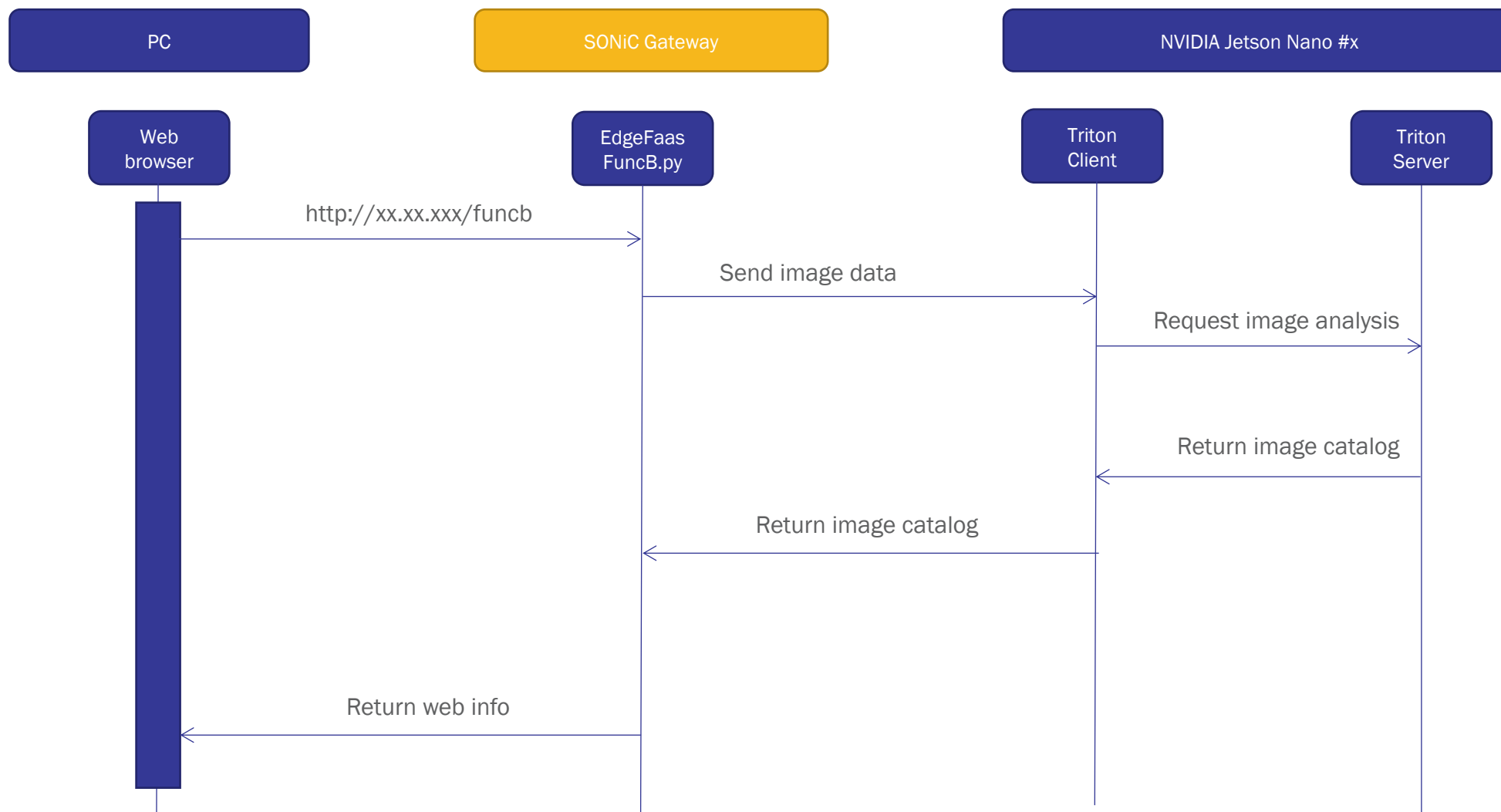
Demo2: Get cluster system info



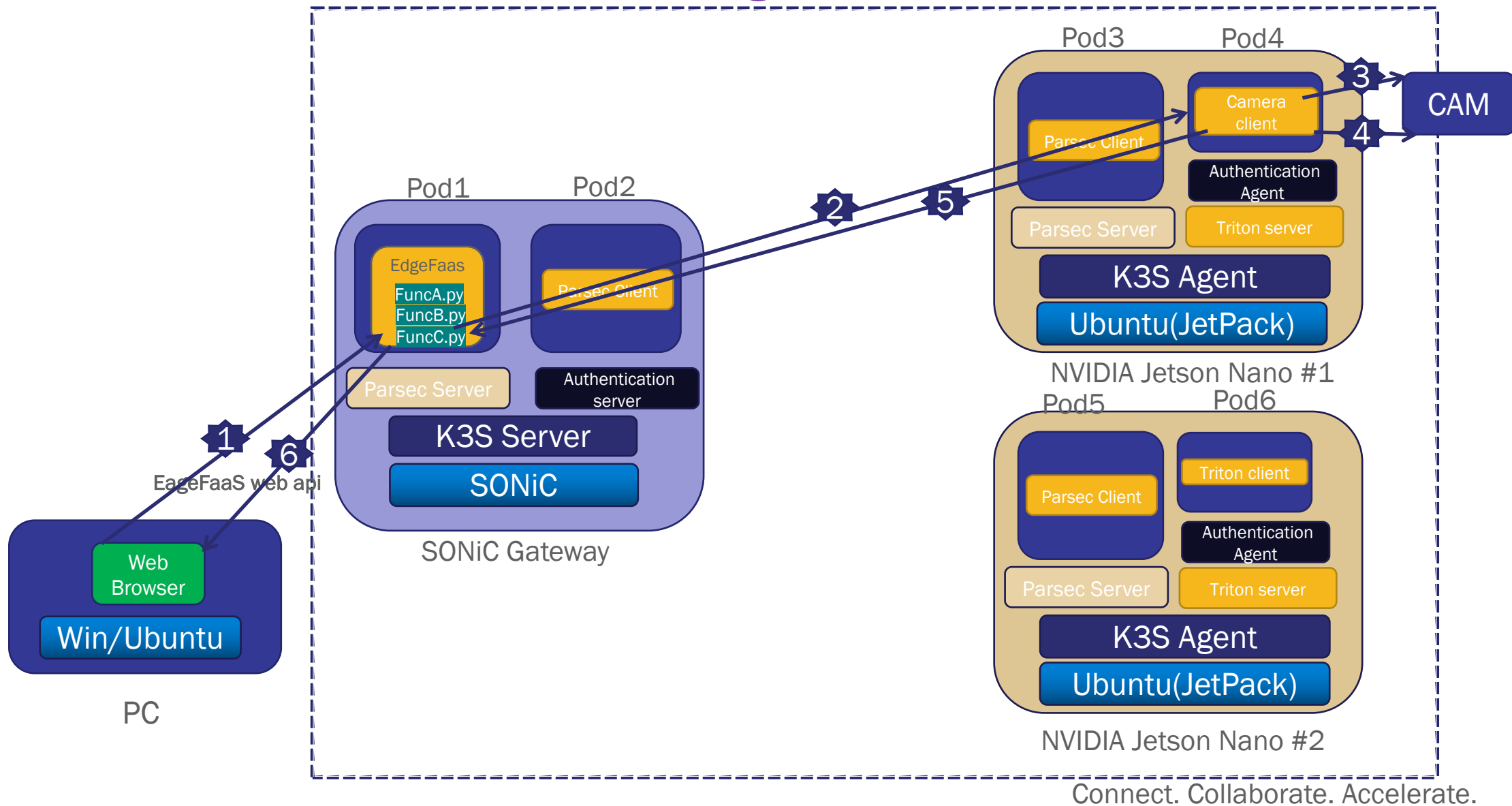
Demo3: Image recognition



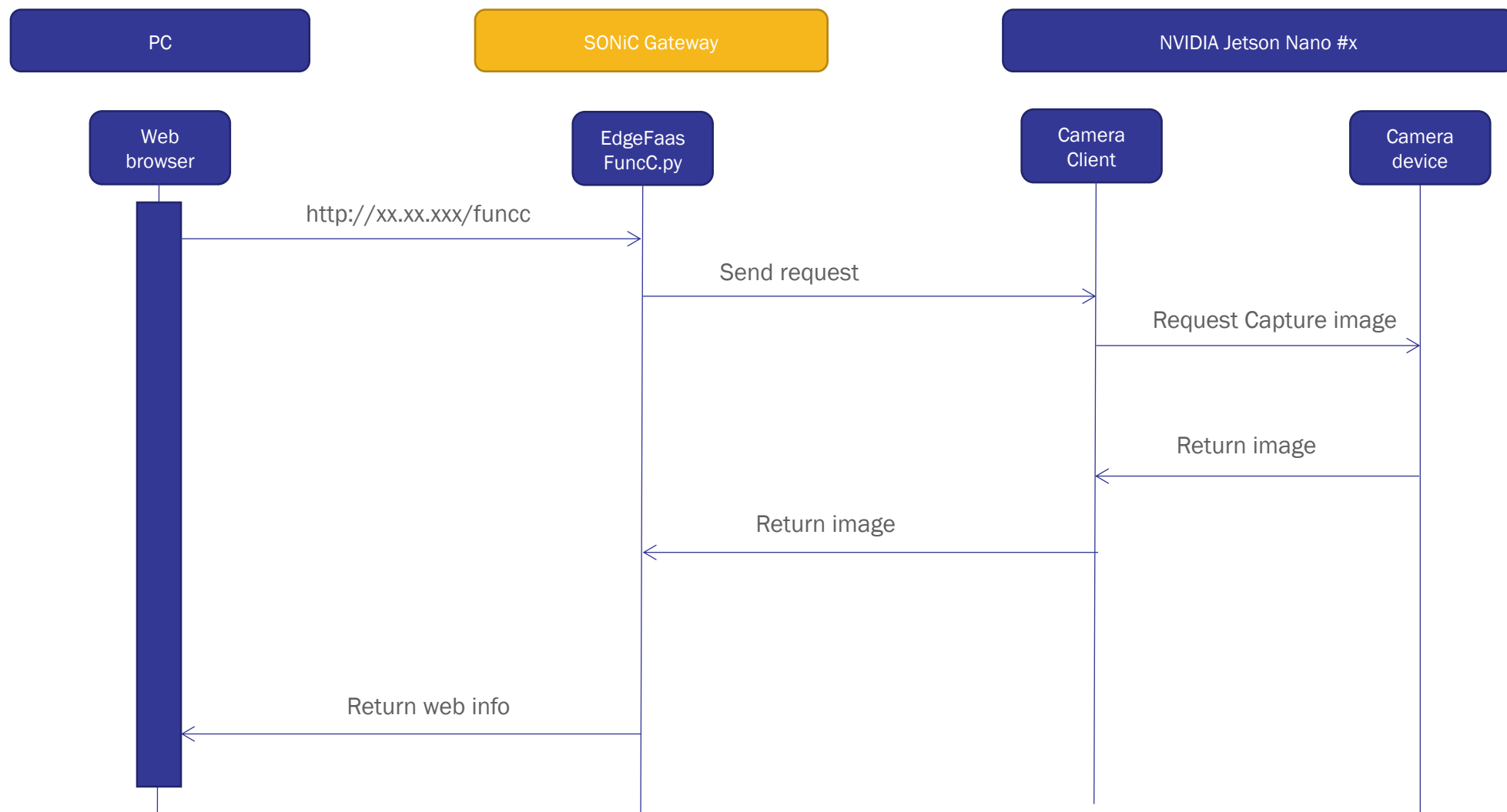
Demo3: Image recognition



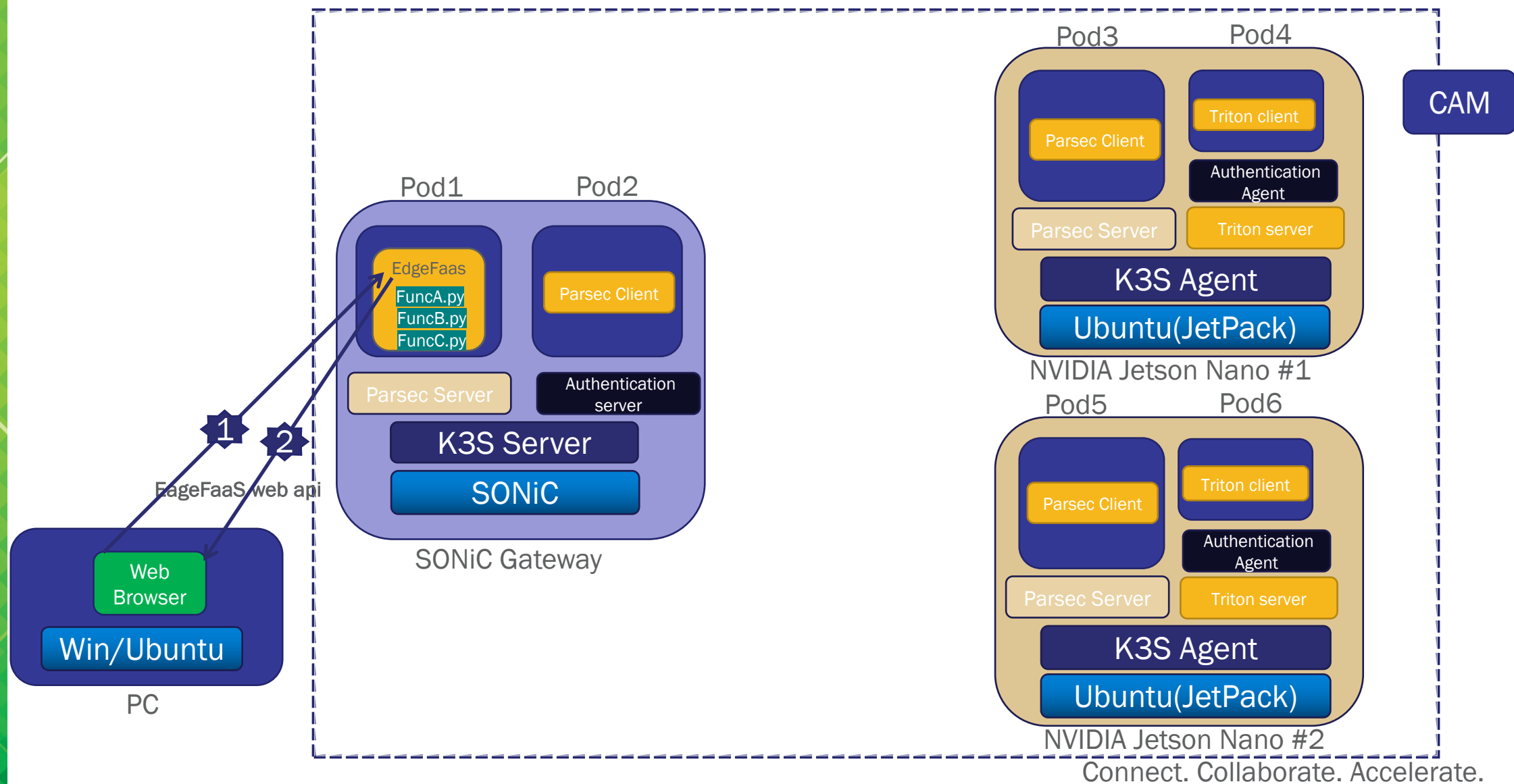
Demo4: Get Camera Image



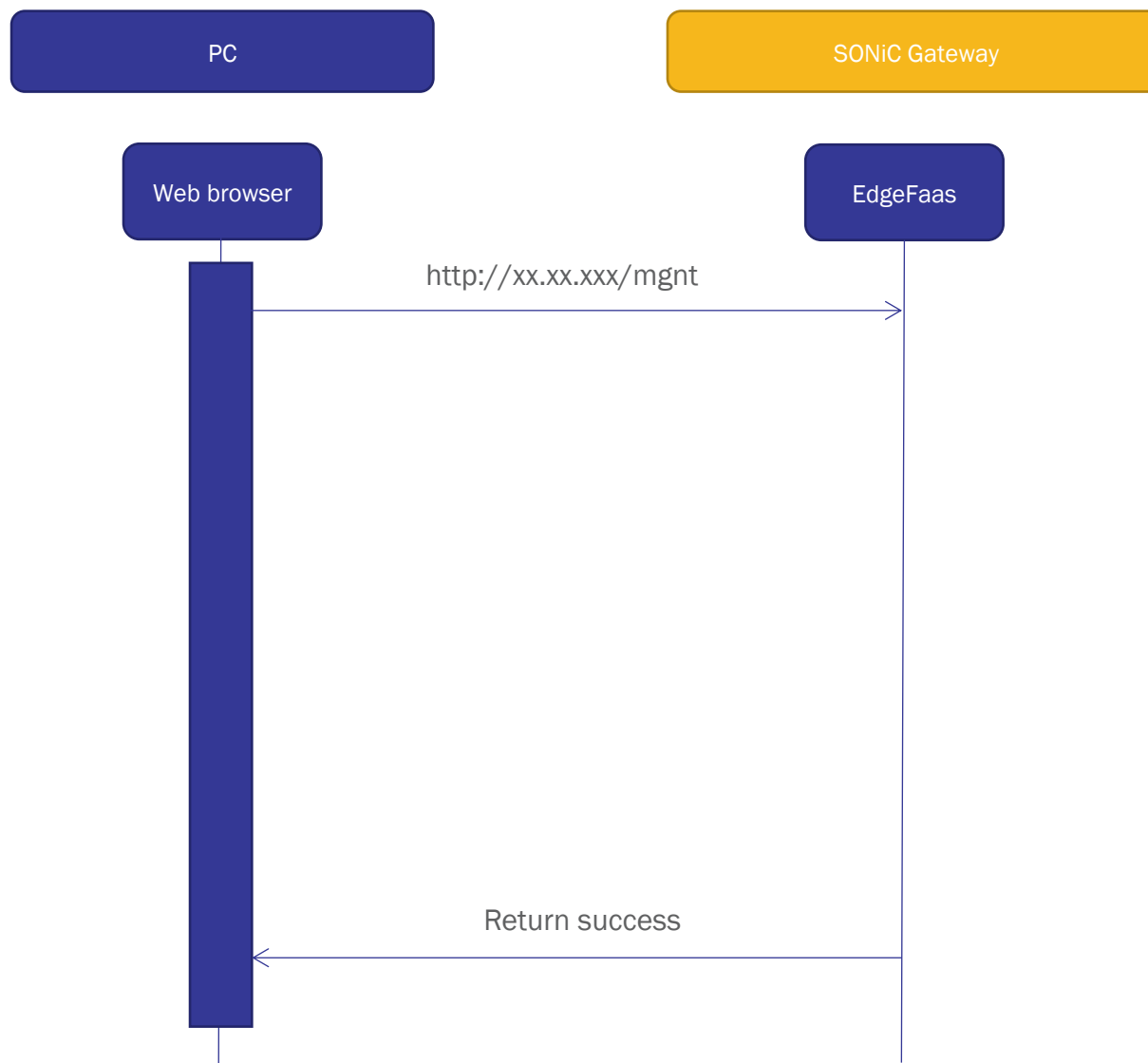
Demo4: Get Camera Image



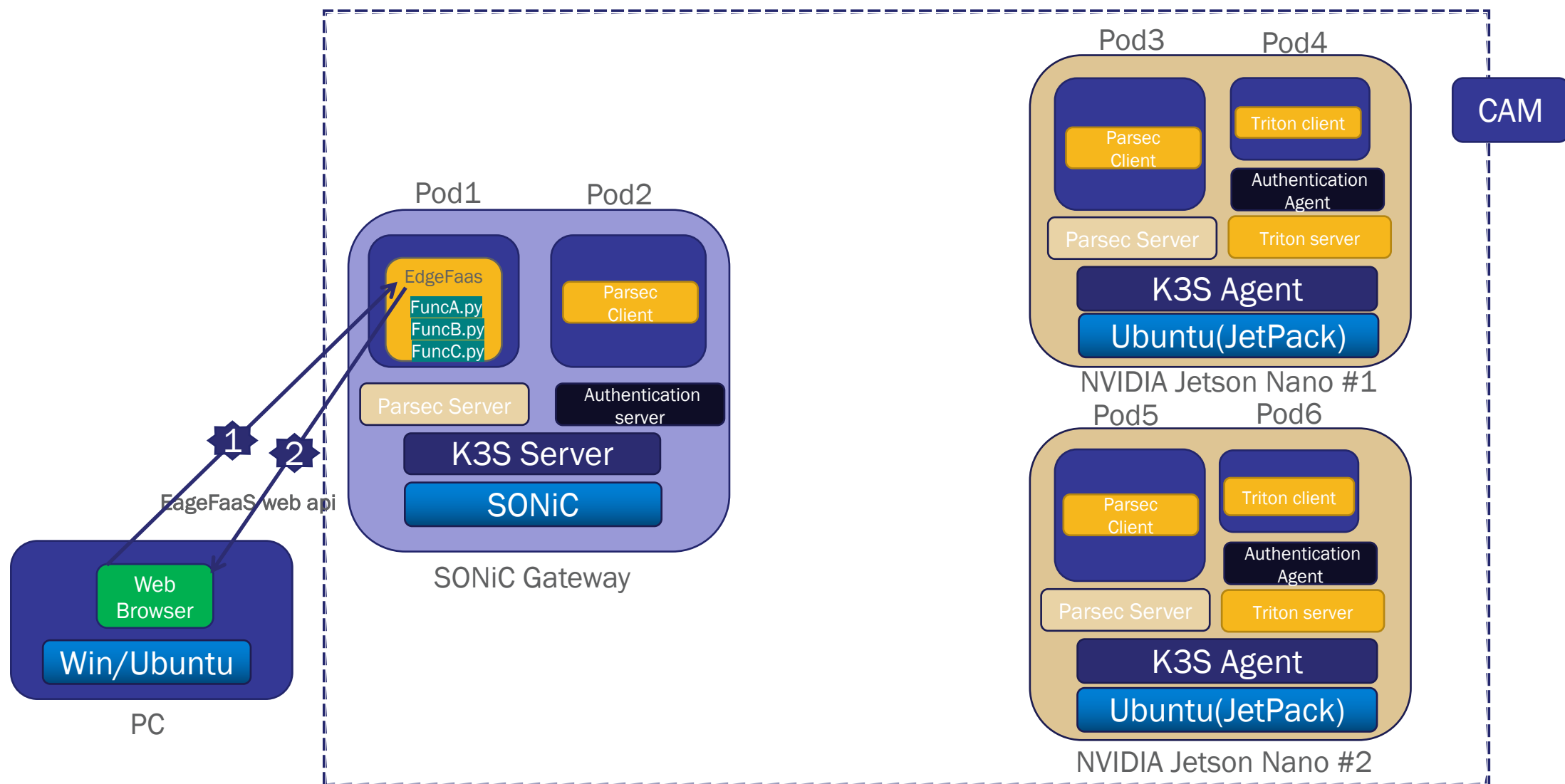
Demo5: Update new Function



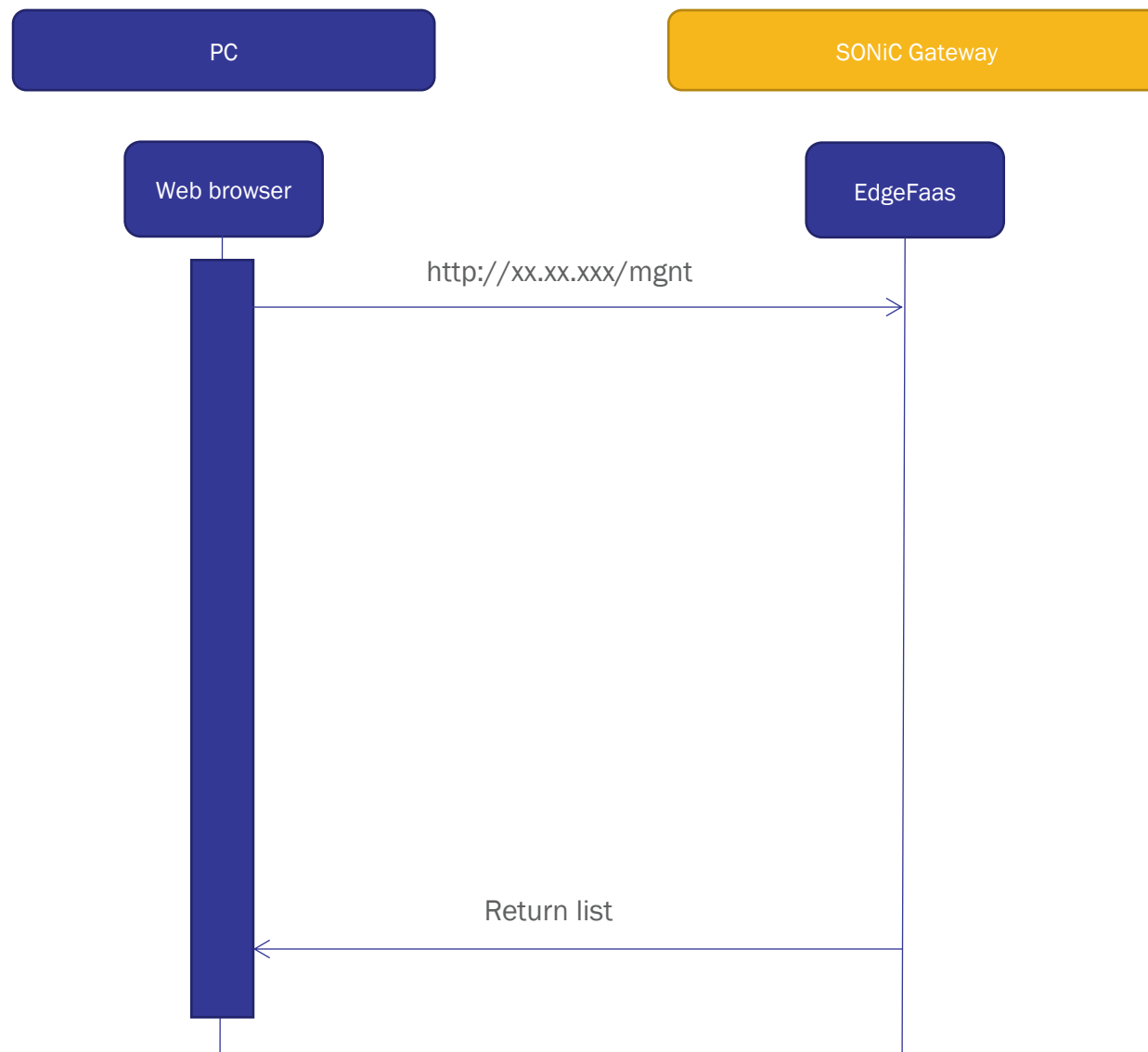
Demo5: Update new Function



Demo6: Get API list



Demo6: Get API list





OPEN
Compute
Project®

Thinking of Application

➤ EdgeFaaS on SONiC in Smart Manufacturing

Connect. Collaborate. Accelerate.

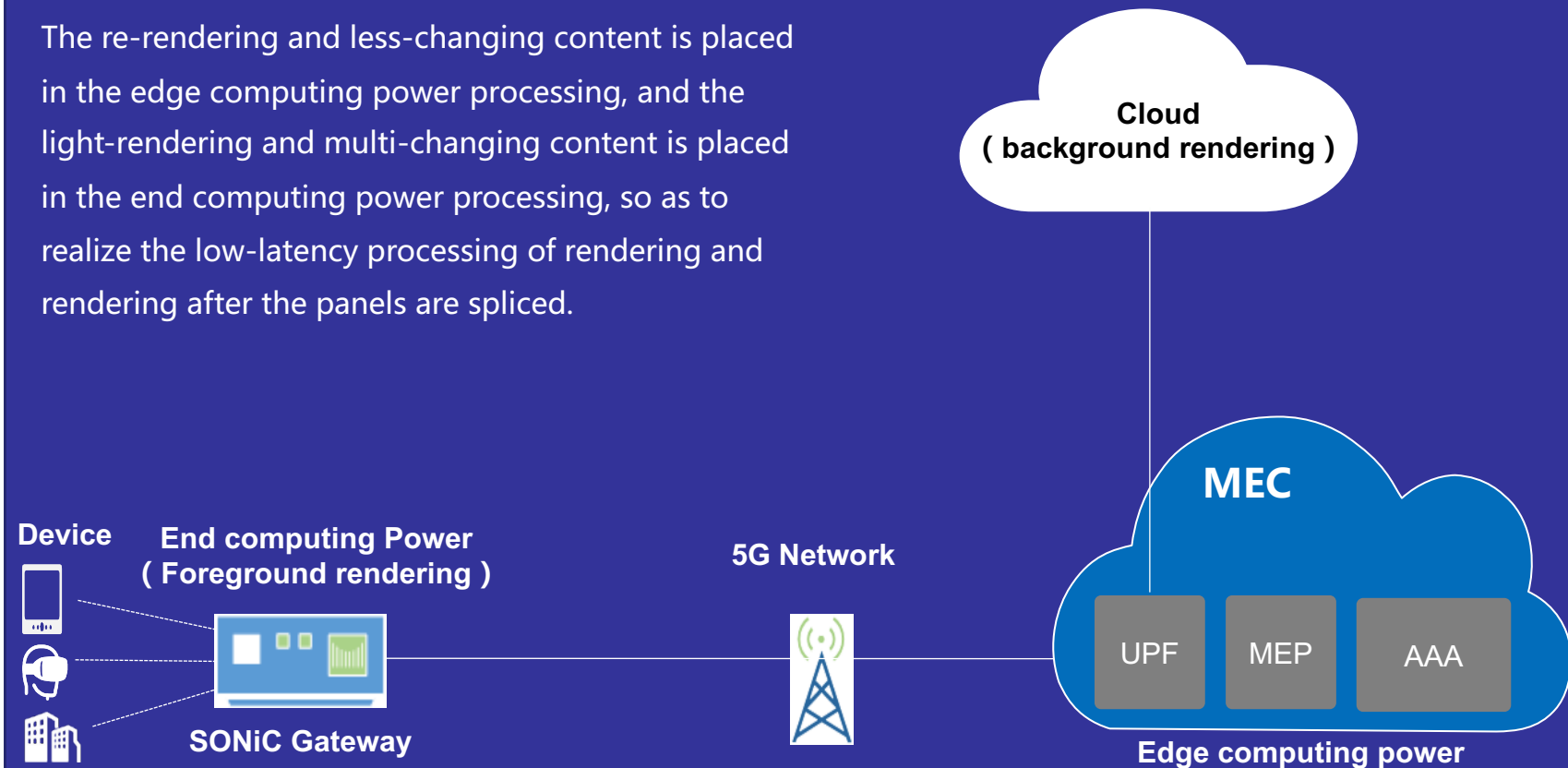
Spatiotemporal splicing positioning technology

Realize the integration of hierarchical computing, intelligent computing power and quality traceability

Production process: Based on 5G communication, deep reinforcement learning and other technologies, open up the existing final assembly production management, use the **area array recognition** and **spatiotemporal splicing positioning technology** of image stitching, and standardize the processing algorithm of the shooting results to realize the segmented construction and overall assembly of components in the mobile environment process.



The re-rendering and less-changing content is placed in the edge computing power processing, and the light-rendering and multi-changing content is placed in the end computing power processing, so as to realize the low-latency processing of rendering and rendering after the panels are spliced.



Q&A

Connect. Collaborate. Accelerate.



OPEN
Compute
Project®