European Commission | HORIZON EUROPE

MLSysOps

Object Detection on MLSysOps: Adaptive Policies for Hardware Backends using vAccel

Anastassios Nanos - Alexandros Patras

NUBIS - UTH

Outline

- Motivation / Problem Statement
- Overview of the Solution
- MLSysOps Architecture
- vAccel Plugin System
- Runtime Policy Demonstration
- Observations & Benefits
- Takeaways / Conclusion / Q&A

Motivation / Problem Statement

- Challenge:
  - object detection pipelines underutilize heterogeneous hardware
- Performance, energy, and QoS inefficiencies in traditional setups
- Need for adaptive, cross-node orchestration

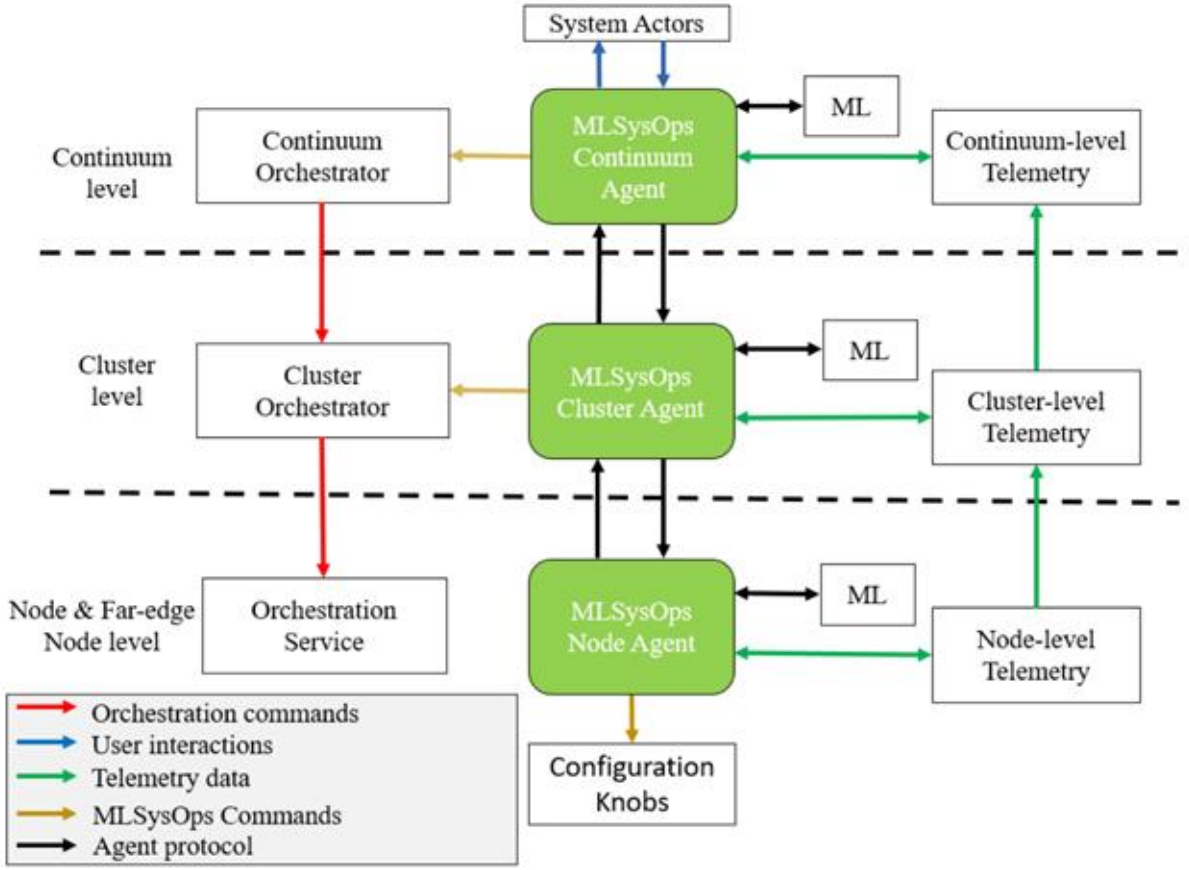
Overview of the Solution

- Key idea:
  - policy-driven, adaptive object detection across CPUs, GPUs accelerators

- runtime adaptivity,
- efficiency,
- energy-aware,
- QoS-aware

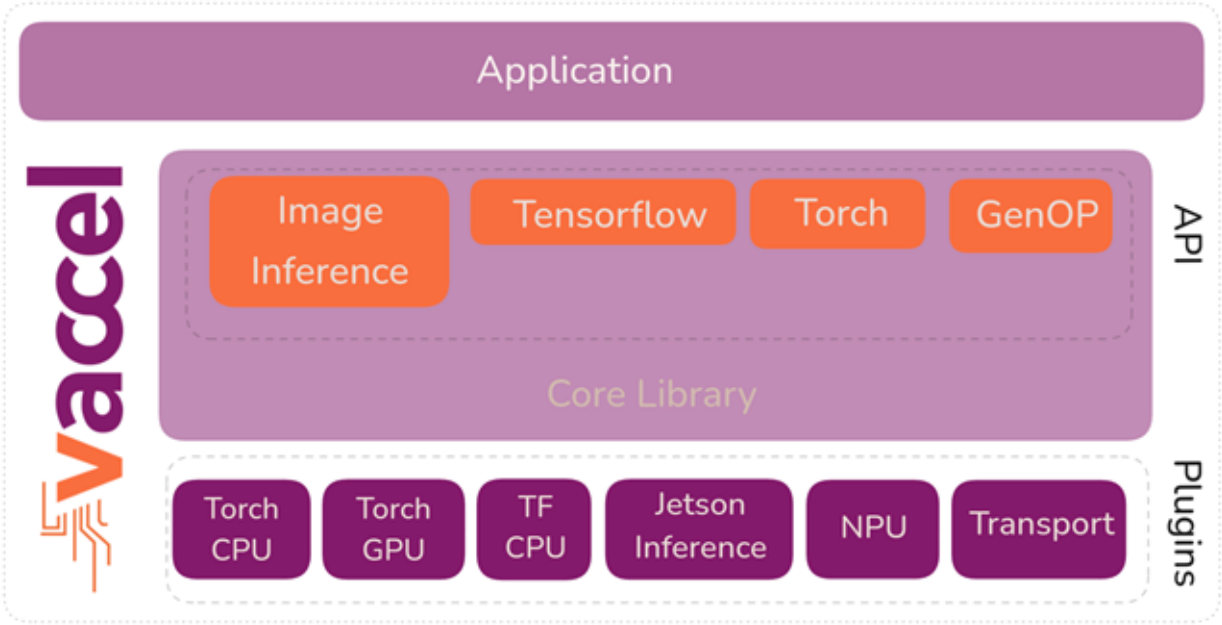
MLSysOps Architecture

- Hierarchical agents on nodes and clusters.
- Telemetry collection: scene changes, workload, hardware availability, QoS, energy.
- Policy engine: runtime-configurable rules.

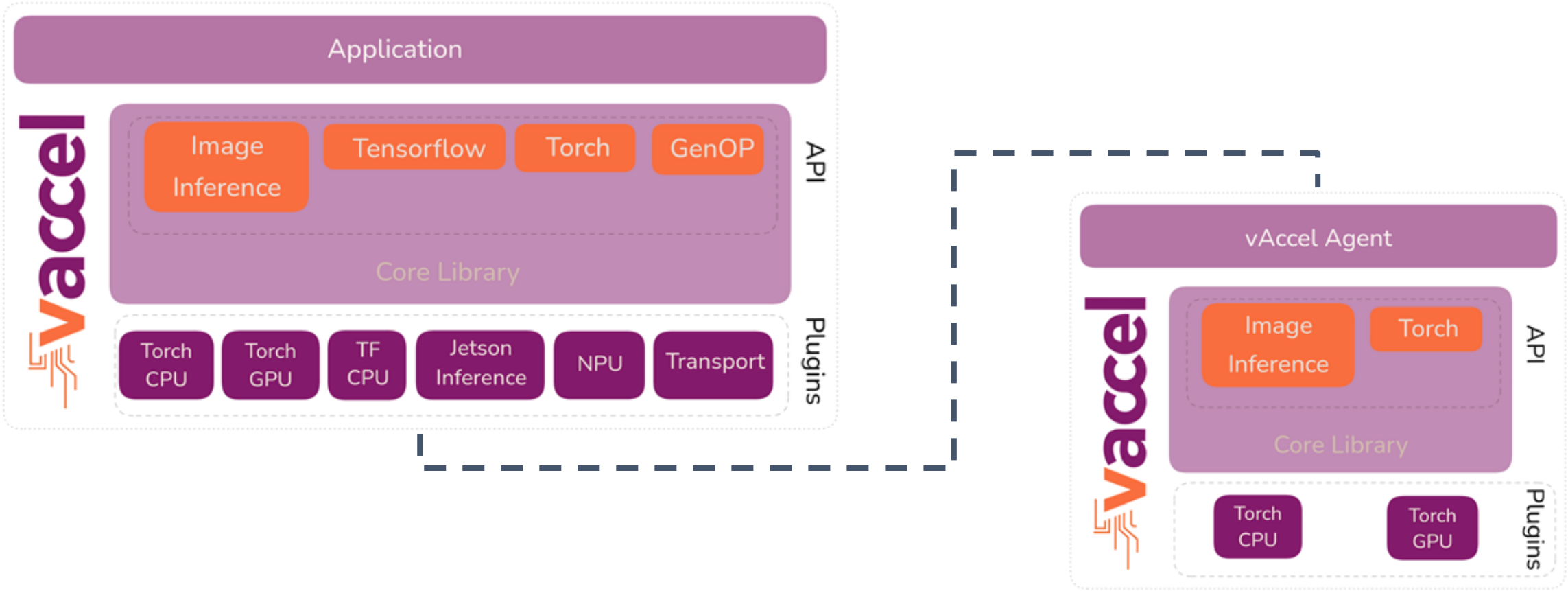


vAccel Plugin System

- Dynamically swappable inference backends
- Policies trigger plugin changes based on telemetry
- Minimal application changes



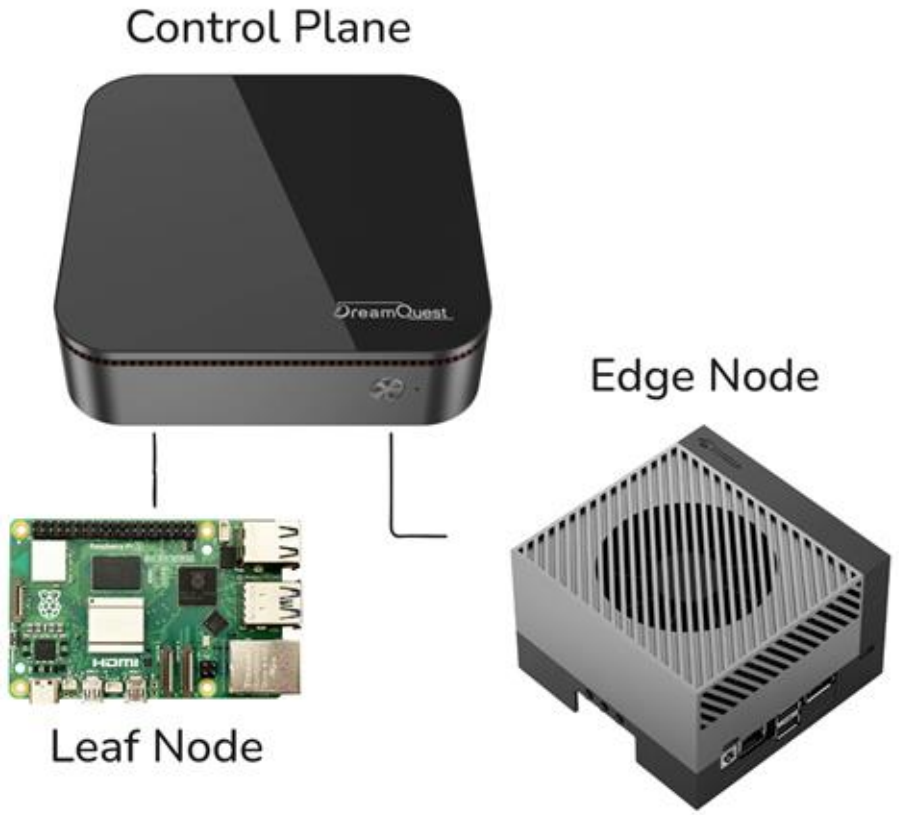
vAccel Plugin System

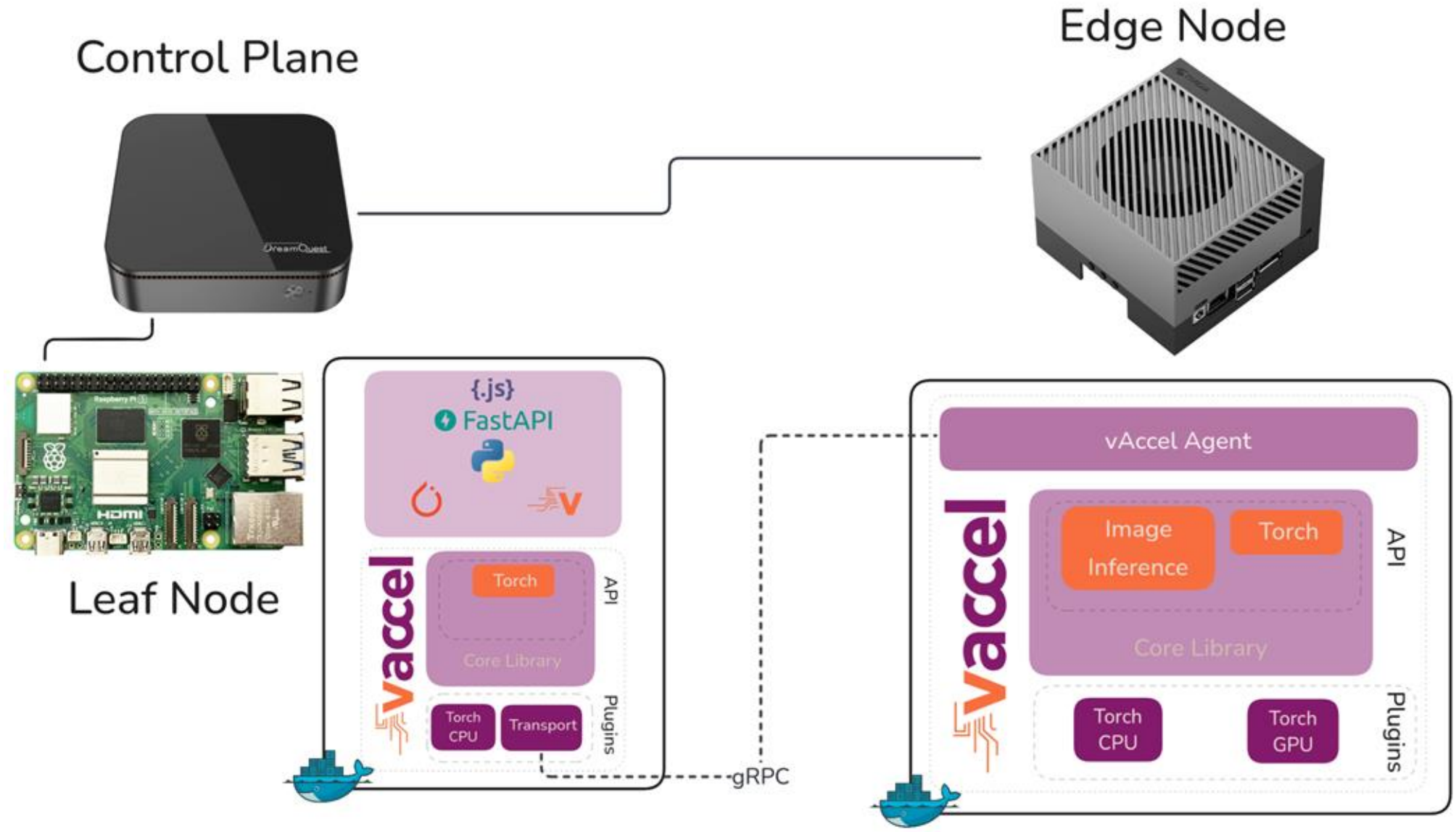


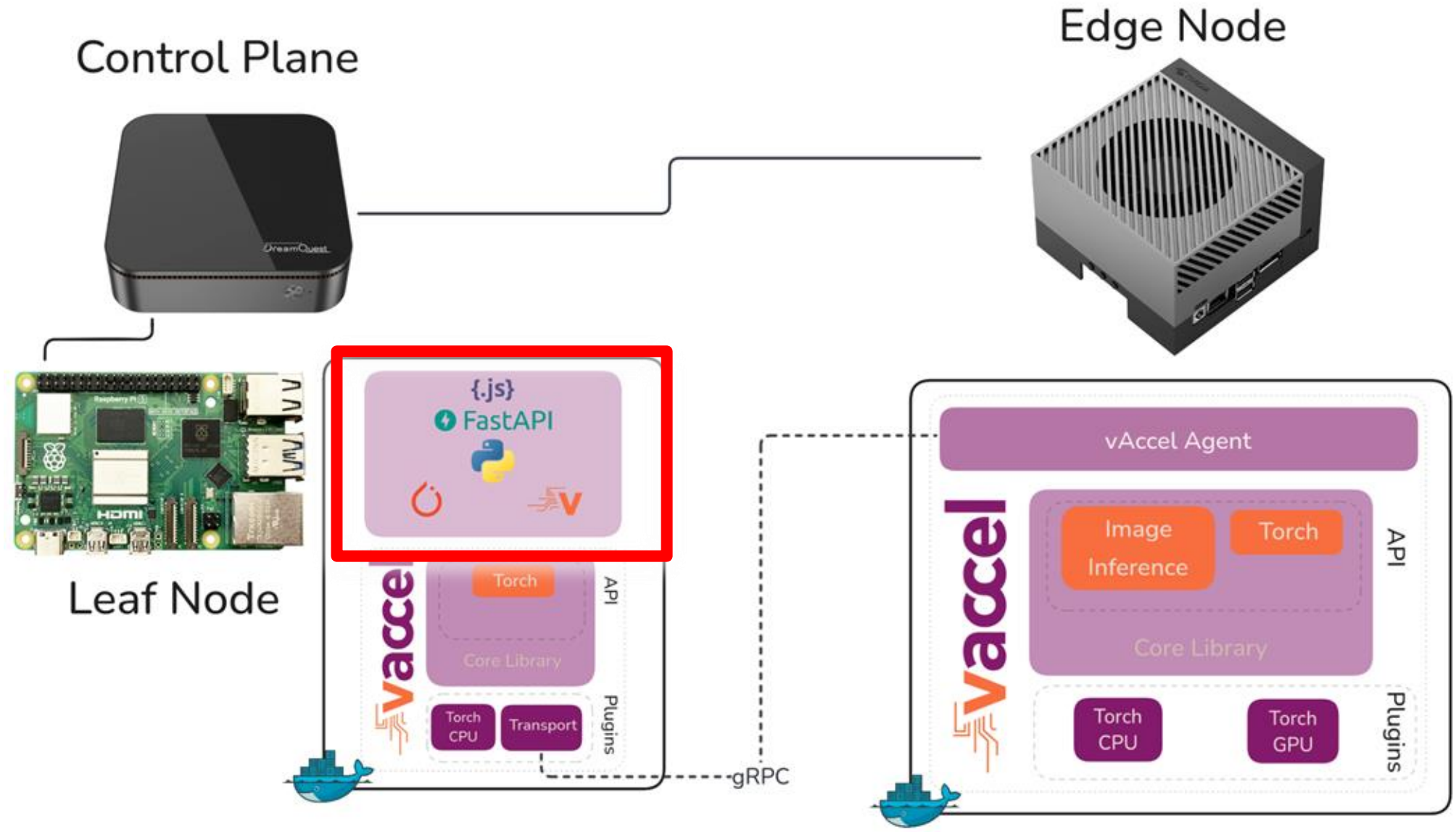
Mechanism demonstration

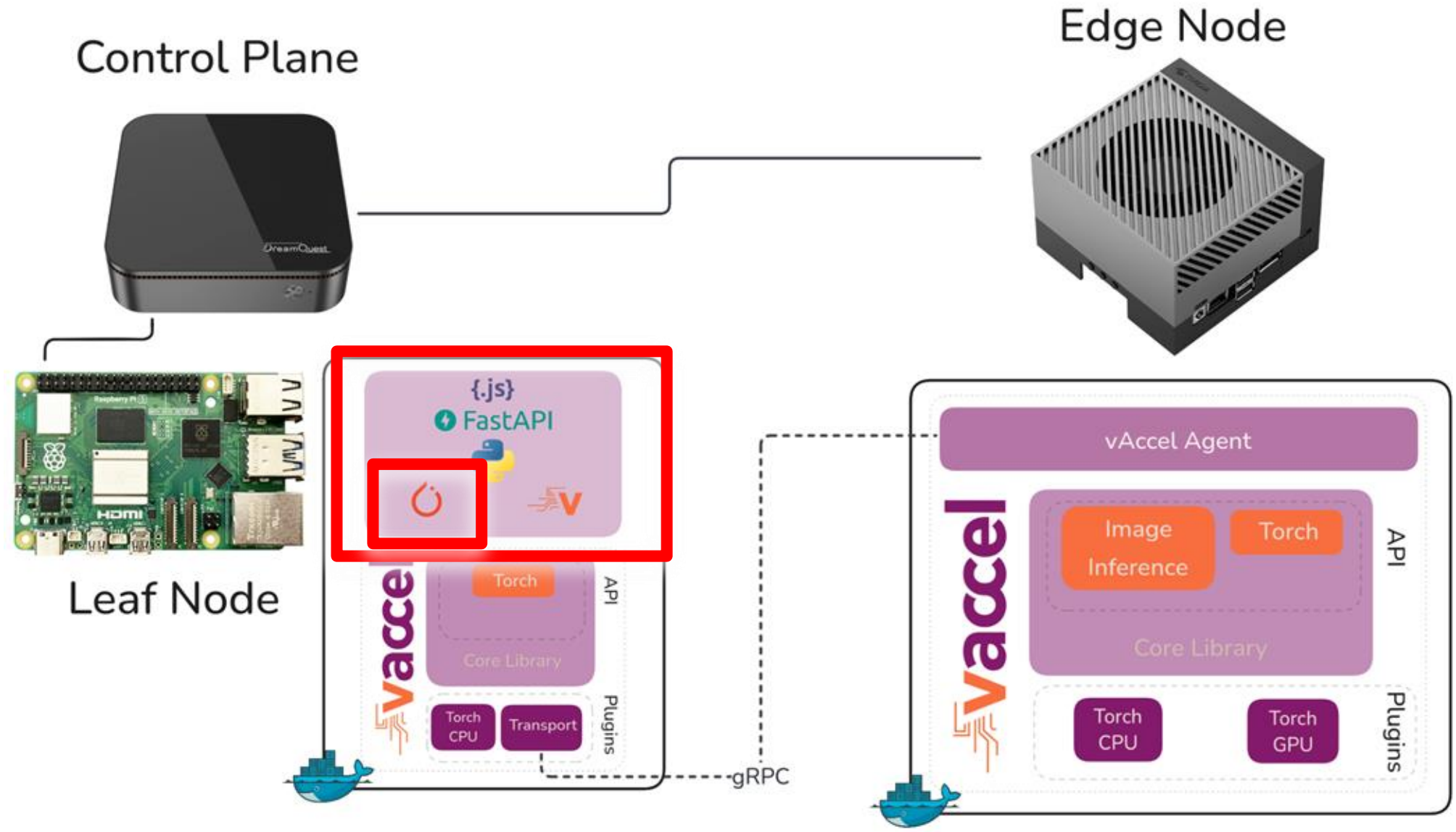
Basic vAccel functionality

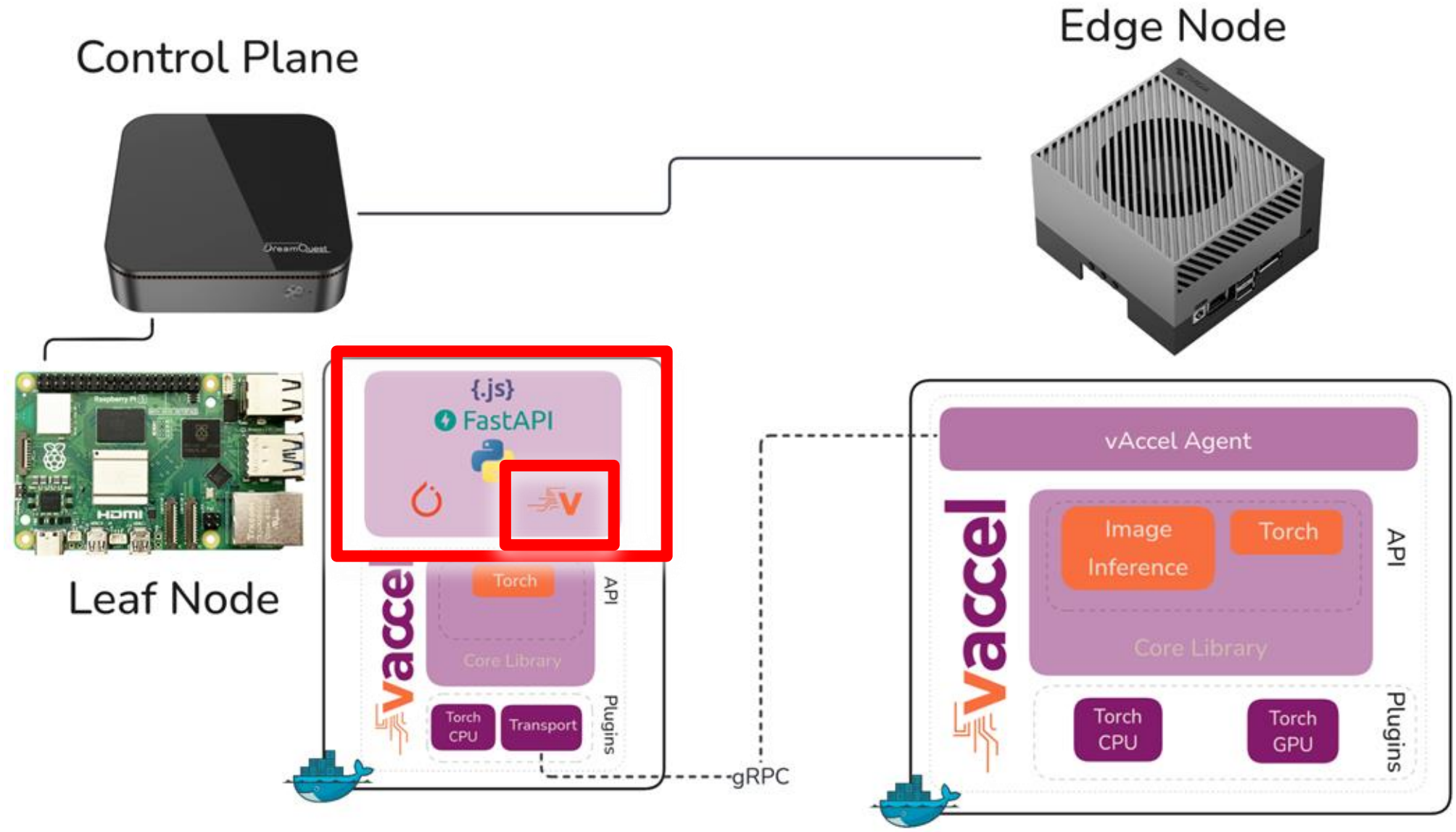
- Basic CV application running object detection using YOLO models on:
  - locally (RPI5) - stock
  - locally (RPI5) - vAccel
  - remotely (RPI5 -> jetson) - vAccel

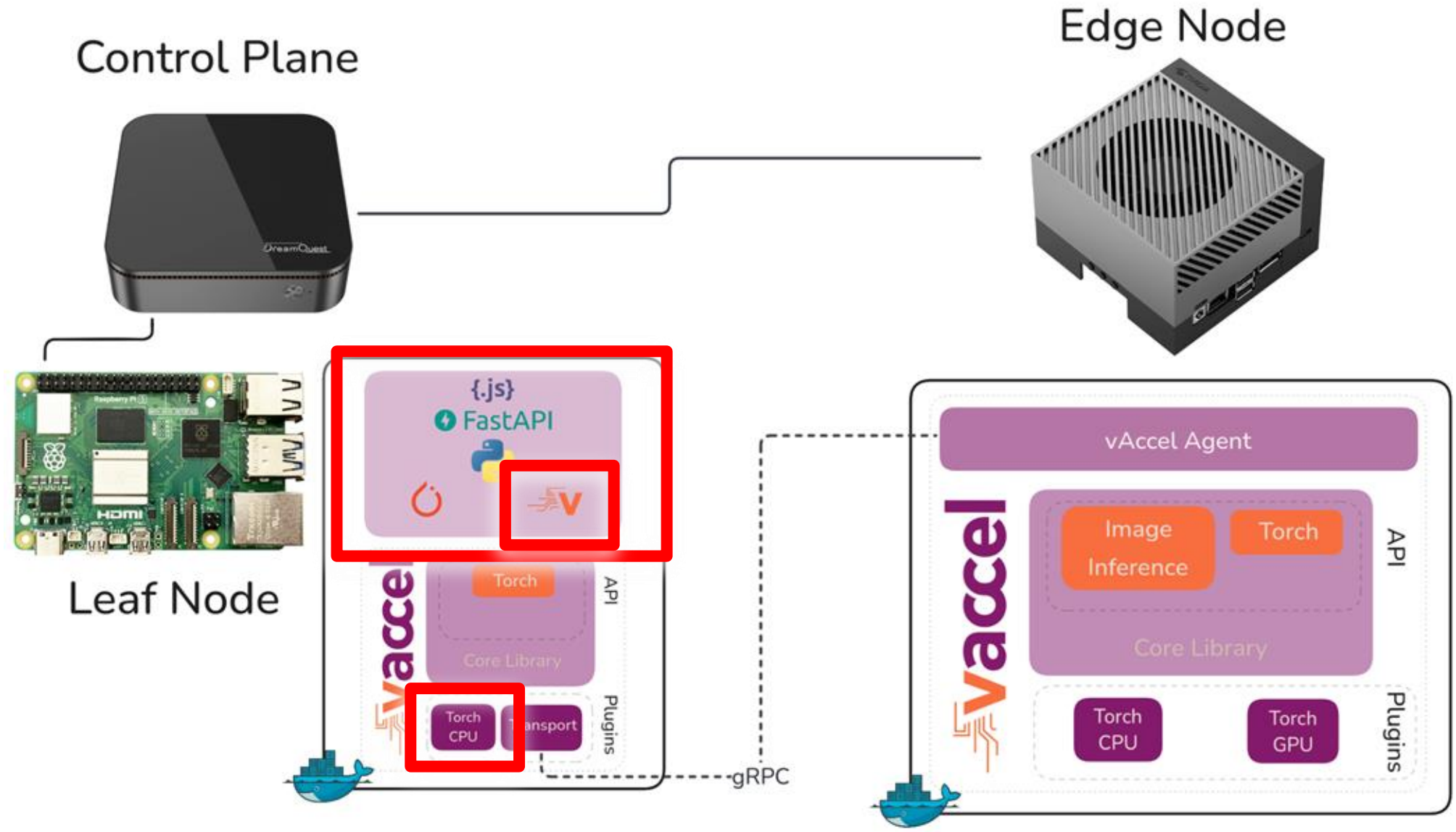


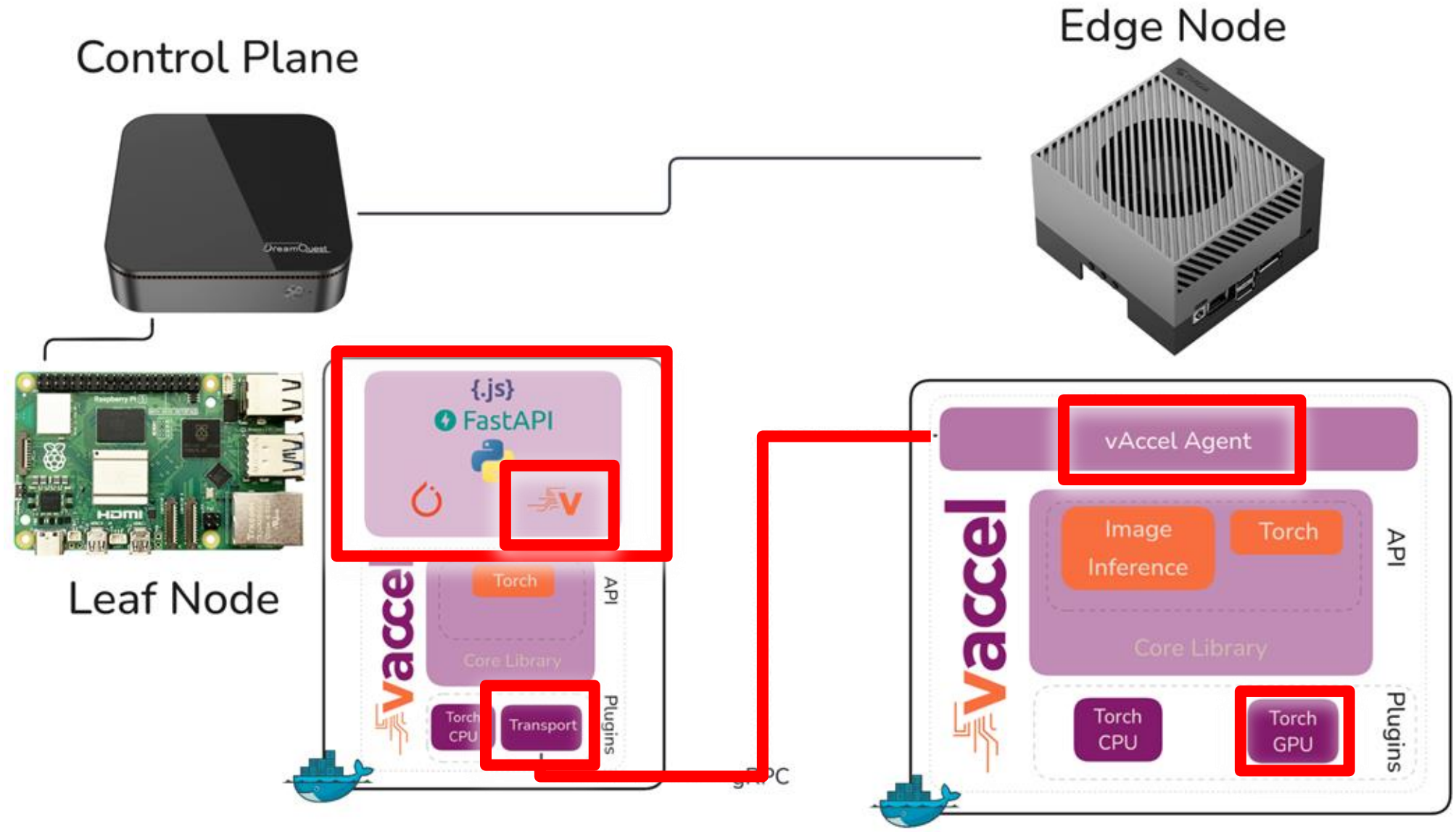

Control Plane
Edge Node
Leaf Node
{.js}
FastAPI
vaccel
Torch
Core Library
API
Plugins
Torch CPU
Transport
gRPC
vAccel Agent
Image Inference
Torch
Torch GPU


Control Plane
Edge Node
Leaf Node
{.js}
FastAPI
vaccel
Torch
Core Library
API
Plugins
Torch CPU
Transport
gRPC
vAccel Agent
Image Inference
Torch
Torch GPU


Control Plane
Edge Node
Leaf Node
{.js}
FastAPI
Torch
API
Core Library
vaccel
Torch CPU
Transport
Plugins
gRPC
vAccel Agent
Image Inference
Torch
API
Core Library
vaccel
Torch CPU
Torch GPU
Plugins


Control Plane
Edge Node
Leaf Node
{.js}
FastAPI
vaccel
Torch
Core Library
API
Torch CPU
Transport
Plugins
gRPC
vAccel Agent
Image Inference
Torch
Torch GPU

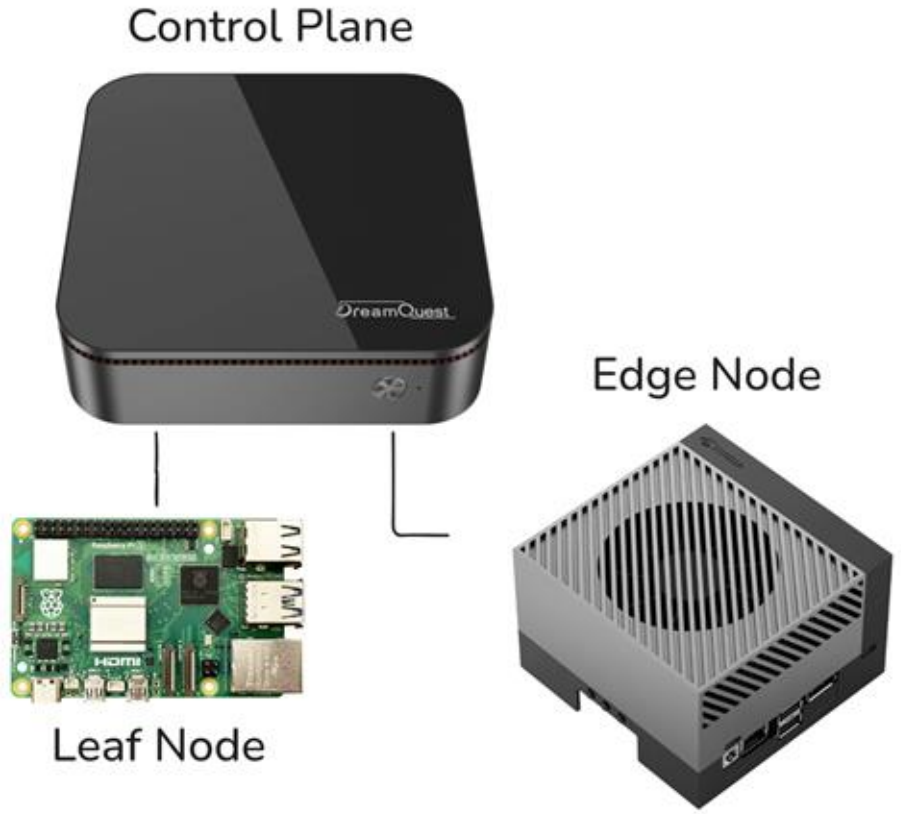

Control Plane
Edge Node
Leaf Node
{.js}
FastAPI
Torch
API
Core Library
Plugins
Torch CPU
gRPC
vAccel Agent
Image Inference
Torch
API
Core Library
Torch CPU
Torch GPU
Plugins

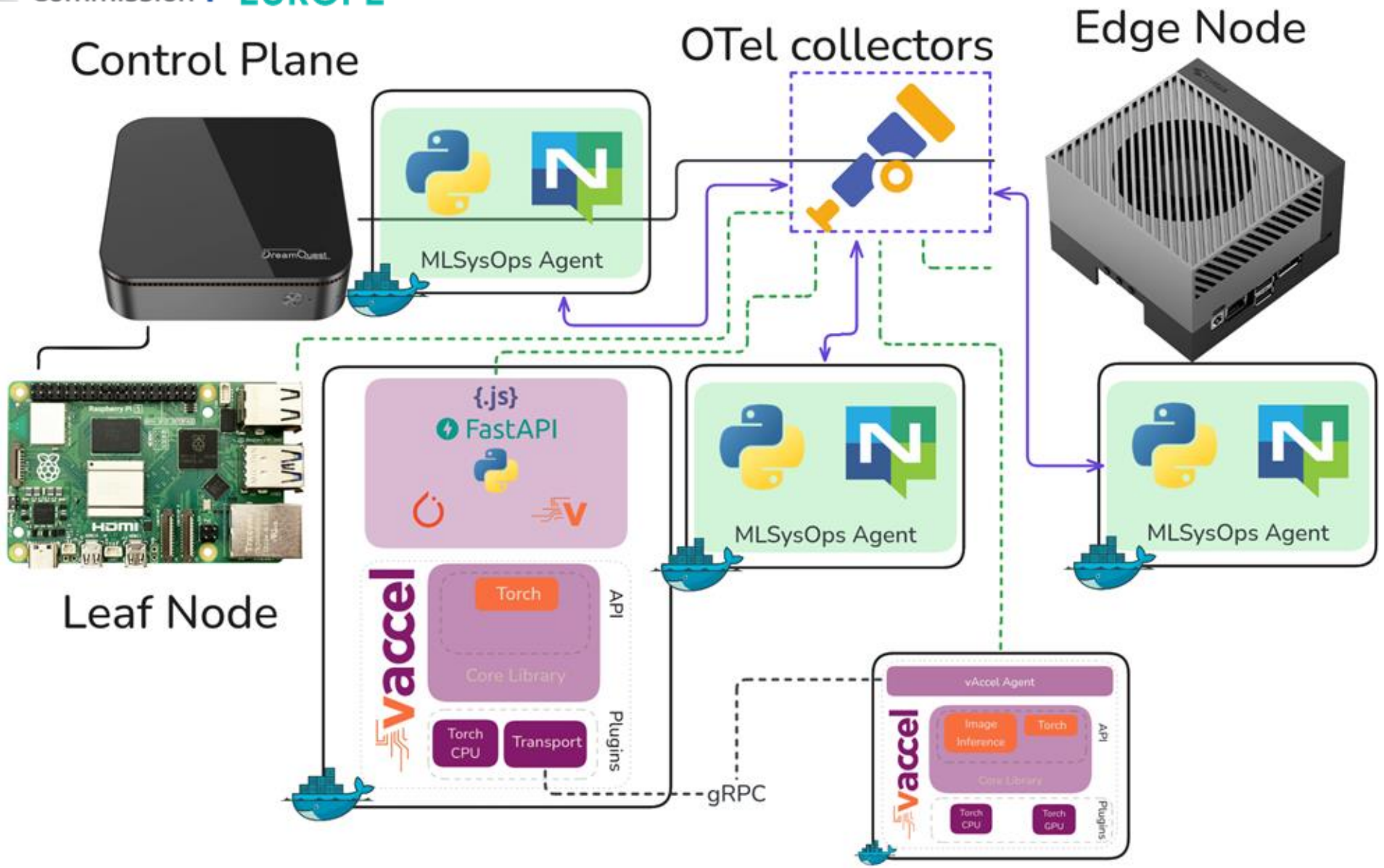

Control Plane
Edge Node
Leaf Node
{.js}
FastAPI
vaccel
Torch
Core Library
API
Torch CPU
Transport
Plugins
gRPC
vAccel Agent
Image Inference
Torch
Torch GPU

Runtime Policy Demonstration

Basic MLSysOps functionality

- Node/Cluster agents observing telemetry data from the cluster:
  - acting on rules defined by the user
  - triggering menchanisms to change runtime behavior




Control Plane
OTel collectors
Edge Node
MLSysOps Agent
Leaf Node
{.js}
FastAPI
MLSysOps Agent
MLSysOps Agent
vaccel
Torch
Core Library
API
Torch CPU
Transport
Plugins
gRPC
vAccel Agent
Image Inference
Torch
API
Torch CPU
Torch GPU
Plugins

Observations & Benefits

- Key outcomes:
  - Faster, smarter detection
  - QoS compliance
  - Energy efficiency
  - Seamless cross-node orchestration
- Simplicity: one policy, multiple accelerators, minimal code changes.

Key Takeaways

- Adaptive AI pipelines are achievable with MLSysOps + vAccel
- Telemetry-driven, policy-based orchestration is powerful and practical
- Play with adaptive inference on your own environment!

mlsysops-eu/mlsysops-framework

nubificus/vaccel

Thank you!
Questions?

This work is partially funded through Horizon Europe Research and Innovation action MLSysOps (GA: 101092912)