
Edge computing

— The way forward for Eclipse IoT —

Agenda

- Introduction to Edge computing
- Open source on the Edge
- Way forward

Intro

Where we are today?

- Everything connected to the core cloud
 - Websites
 - Mobile phones
 - Even IoT devices in most cases

How we got here?

- Mainframes (centralized)
- Client server (distributed)
- Cloud computing (centralized)
- Edge computing (distributed)

**Edge is everything outside of the
core cloud**

**Bring compute resources closer
to the source**

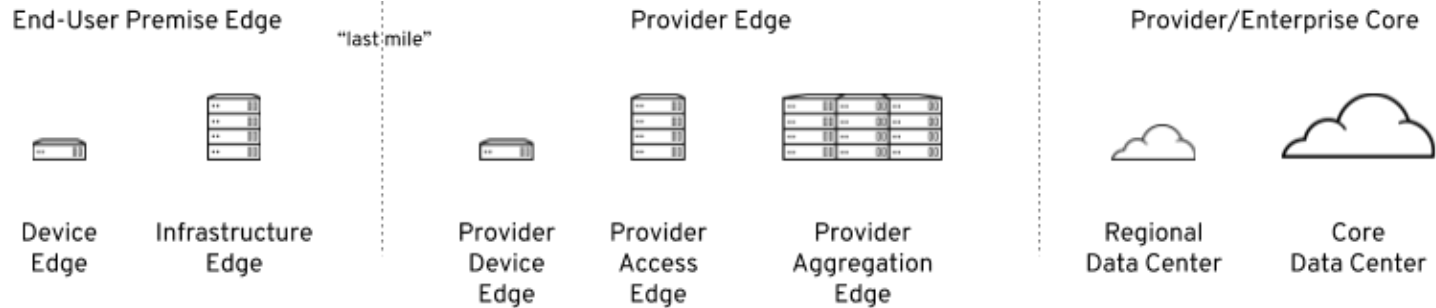
Key triggers?

- IoT
- Much more data
- Need for real time processing
- Much more compute resources

Key enablers?

- Cloud native computing
- 5G
- Machine learning
- Inexpensive, power-efficient hardware (SoC modules, etc.)
- ...

THERE ARE MANY EDGES



Latency

- React locally on sensor or scheduled events
- Compute offload
 - Schedule resource intensive tasks on the dedicated hardware on the Edge
 - Example AR/VR renderings
- Machine learning
 - Cloud trained models - executed on the Edge
 - Edge specific training (environment and data policies)

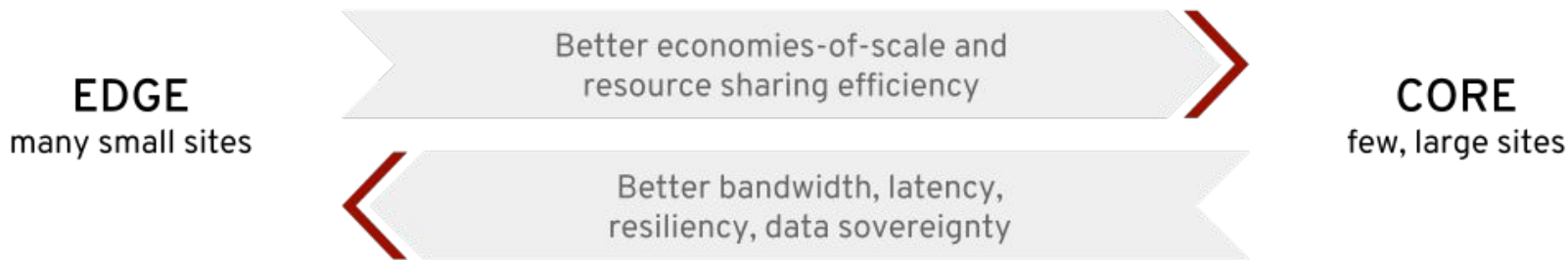
Reliability and HA

- Buffer and batch
 - Store and forward
 - Brokers on Edge nodes
- Caching
 - Local (partial) databases on Edge nodes
 - Sync data with the cloud and other Edge nodes

Data preprocessing

- Data sensitivity
 - GDPR
- Convert data to general structured messages
 - Normalize data structure
- Data analytics
 - Send only relevant data
 - Combine multiple sources
- Add metadata
 - Location, Identity, Security

WHAT IS EDGE COMPUTING?



*Centralize where you can,
distribute where you must*

Key applications?

- Large scale IoT and IIoT
- Smart infrastructure
- Gaming industry
- VR/AR
- AI/ML
- Automotive / Autonomous vehicles
- Security and Surveillance
-

Challenges

- Infrastructure
 - How to manage resources (nodes and clusters) on the Edge?
- Control plane
 - How to manage workloads on the Edge?
- Data plane
 - How Edge sites communicate with the cloud and between themselves?

Challenges

- Resources
 - Limited number of nodes on the Edge
 - No “bursting” to newly provisioned capacity like a public cloud or large datacenter
 - Workloads typically have a wide range of priorities
 - Need more emphasis on prioritization, triage
- Network
 - Network capacity can be limited, and variable
 - Like resources, different workloads can have different network policies/priorities

Security

- Unattended operation
- Physical security
- Purity of images
- Secure delivery of secrets
- Unauthorized microservices
- Controlled access to resources
- Guaranteed remote shutdown

Microservices

- Deployment
- Resources
 - Pod priorities
- Communication
 - VPN
 - VAN
- Security
 - Matching microservices to edge hardware
 - Unauthorized outbound

Open source for the Edge

Open source for the Edge

Eclipse ioFog



Where does it fit?



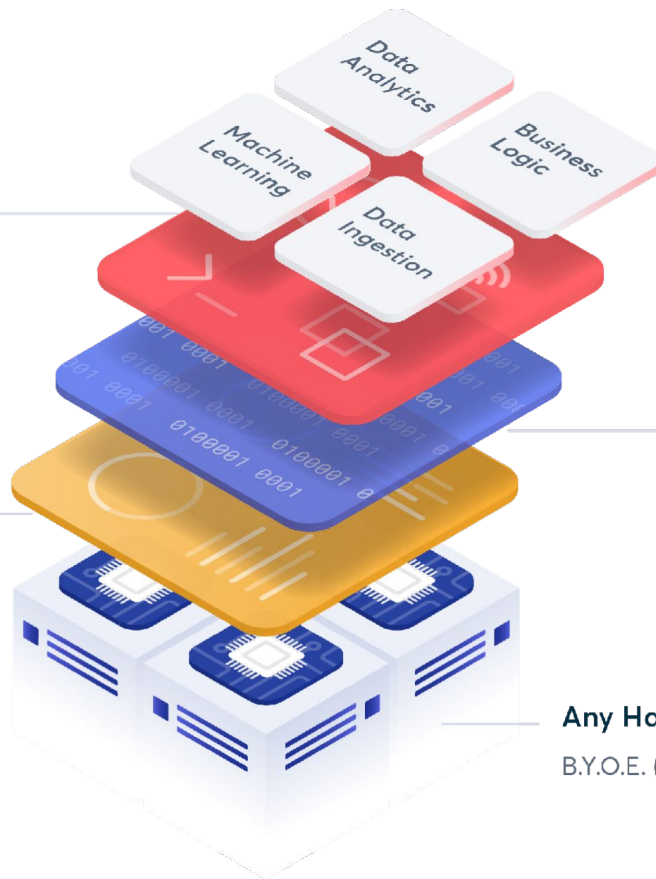
What does it do... on the lowest level?

Application Framework

- Microservice Manager
- Microservice Authorization
- Remote Config & Debug
- Testing/Version Control
- Plug & Play Installation

Edge Kernel

- Resource Manager
- Hardware Abstraction
- Secure Device Connectivity
- Health Monitoring
- Edge Hardware Security



Microservices

Any Software

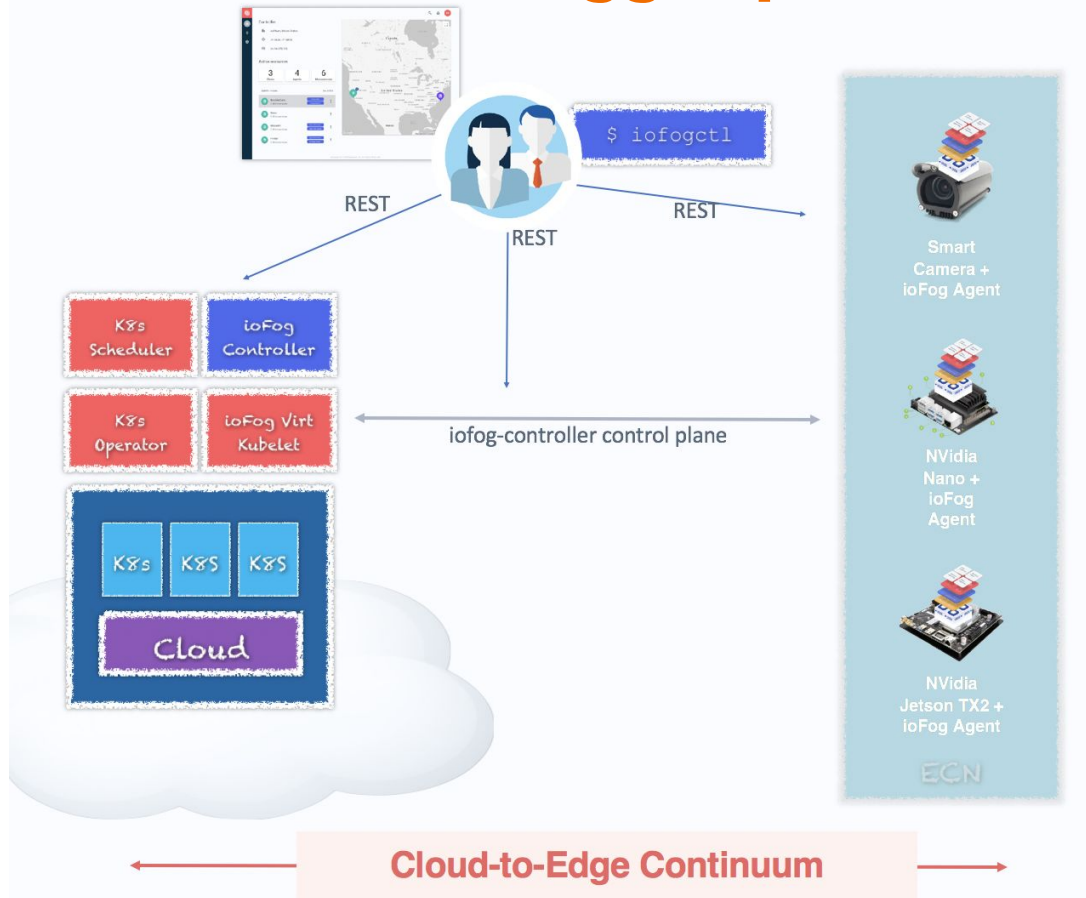
Fog Services

- Distributed Data Bus
- REST APIs
- Real Time APIs
- P2P Fog Networking
- Flexible Storage

Any Hardware

B.Y.O.E. (Bring your own edge).

What does it do... in the bigger picture?



What does it do... to development lifecycle?

EdgeOps Lifecycle

Development

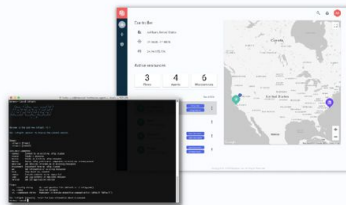
EdgeOps

Developer Tools



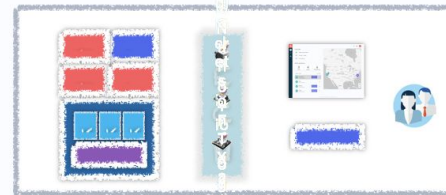
- ioFog SDKs (Python, Node, Java)
- EdgeAI DevKits for Nvidia and Intel OpenVino platforms
- Example EdgeAI Applications
- Documentation
- Video Series

Developer Edition



- Single Node Version
- Same Features as Cloud/Edge
- Open Source
- Spin up local ECN for development and testing
- Integrate with IDEs

Production Runtime



- ioFog Edge Computing Fabric
- Universal Compute Platform
- 100% Open Source
- Runs on bare metal or Cloud
- ioFog controller hosted as a service or integrated with customer's K8S
- Automated EdgeOps lifecycle management

Introducing the Edge Compute Network (ECN)

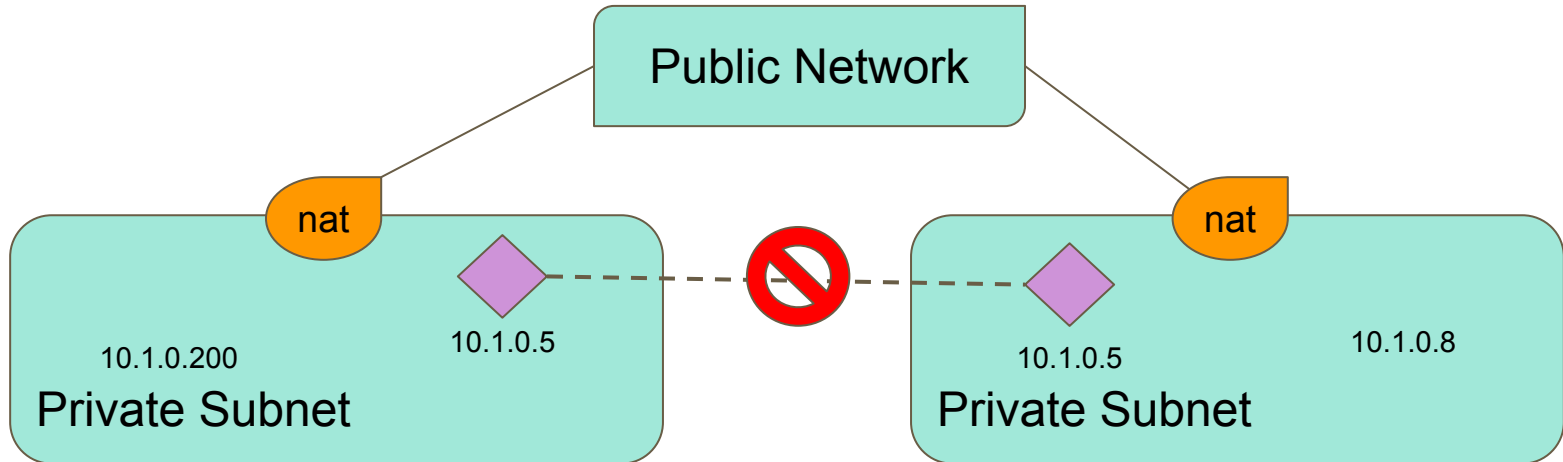
The dashboard displays the following information:

- Controller:**
 - Location: Chicago, United States
 - Coordinates: 41.8781, -87.6298
 - IP Address: 35.209.224.116
- Active resources:**
 - Flows: 2
 - Agents: 3
 - Microservices: 6
- Agents - 3 nodes:**
 - pi-agent1:** 2 Microservices. Actions: heart-rate-mon..., heart-rate-view...
 - pi-agent2:** 2 Microservices. Actions: openvino-cam..., Network for ML...
 - pi-agent3:** 2 Microservices. Actions: ai-image-result..., Network for ML...

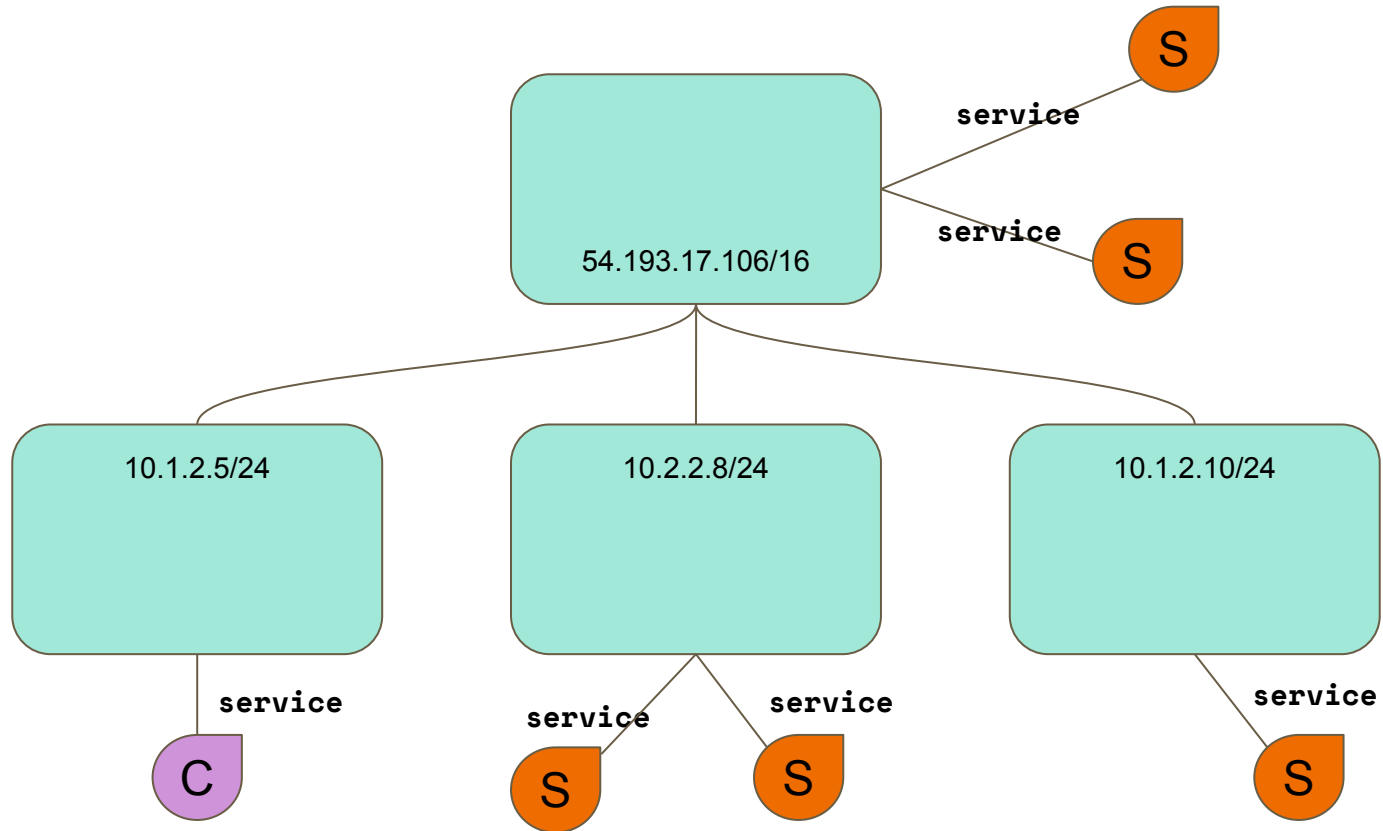
The right side of the dashboard features a map of the United States with a purple location pin in the Midwest and a green location pin in California. The map includes state names and a search bar at the top right.

Edge networking

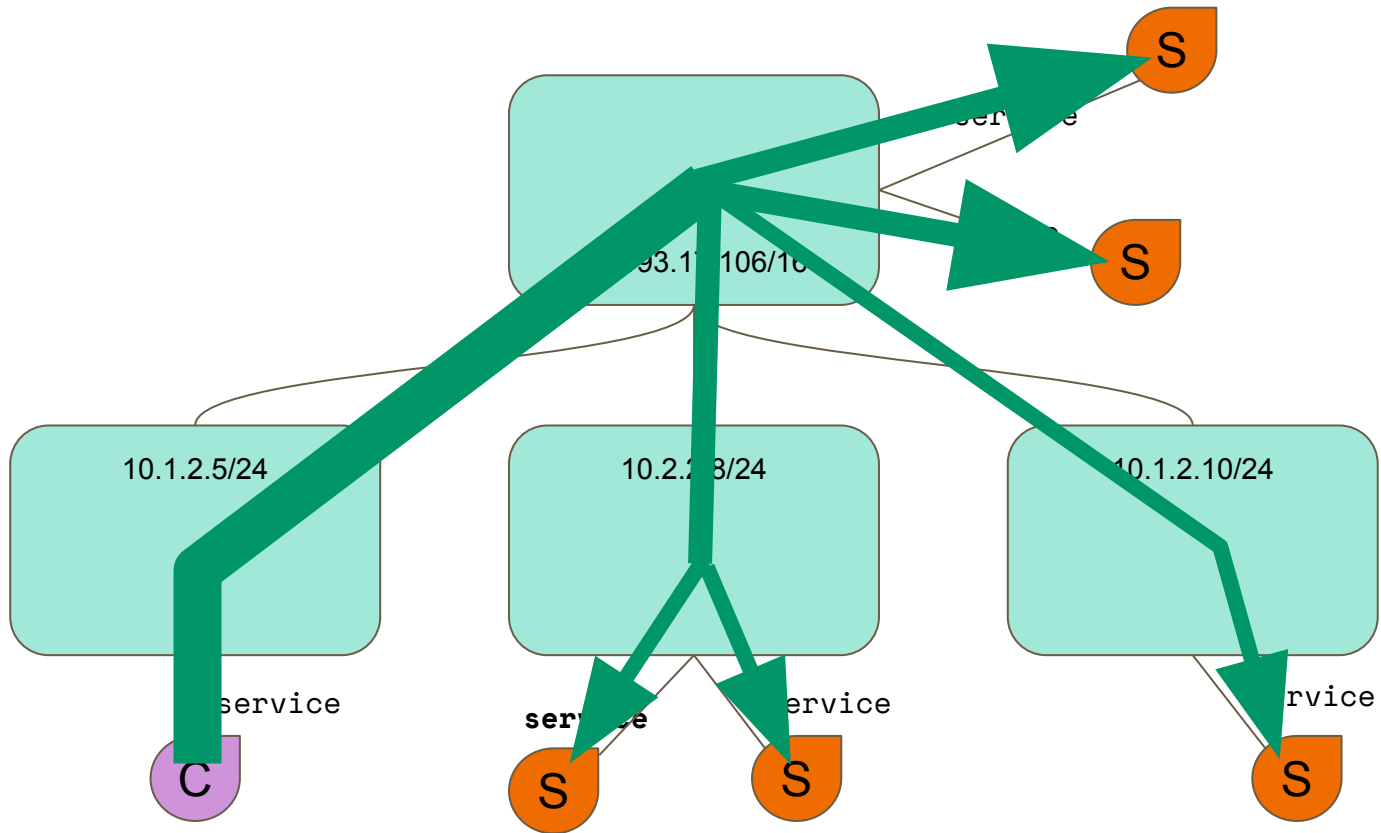
- Hybrid cloud, microservice architecture, agile integration, etc.
 - Not client/server
 - Services/processes want to be deployable and addressable everywhere (north/south/east/west)
- Edge computing - Lots of private subnetworks



Application Layer Addressing



Application Layer Addressing



Implications of Application Addressing

- Security
 - Access control for addresses - at the service/process/business resolution
 - Locked-down network membership - Mutual TLS for inter-site connections
 - Cross-cluster applications not exposed via Kube networking
 - Public exposure limited to ingress
 - Trusted and untrusted edges
- Management
 - Metrics collected at business resolution

Skupper.io

- Operational Ease
 - Easy to deploy in a multi-cluster network
 - No advanced networking (SDN, VPNs, Tunnels, Firewall rules, etc.)
 - No need for elevated or admin privileges
 - No problem with overlapping CIDR subnets or mixes of IPv4 and IPv6
 - No single point of failure - use redundant topology
- Not just for messaging
 - Proxy maps HTTP, TCP, UDP, etc. to AMQP
- <http://skupper.io>
 - Examples, demo-videos, etc.
 - New, emerging project

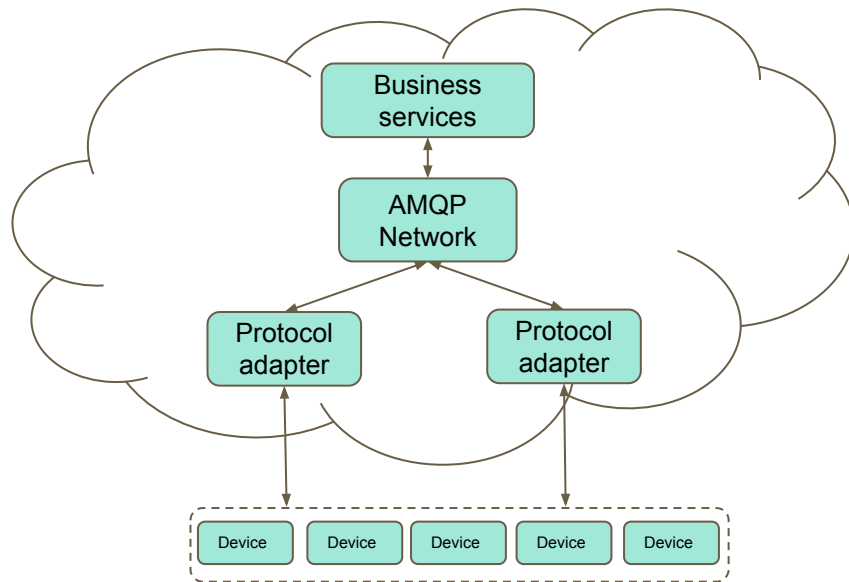
Way forward

Is cloud obsolete?

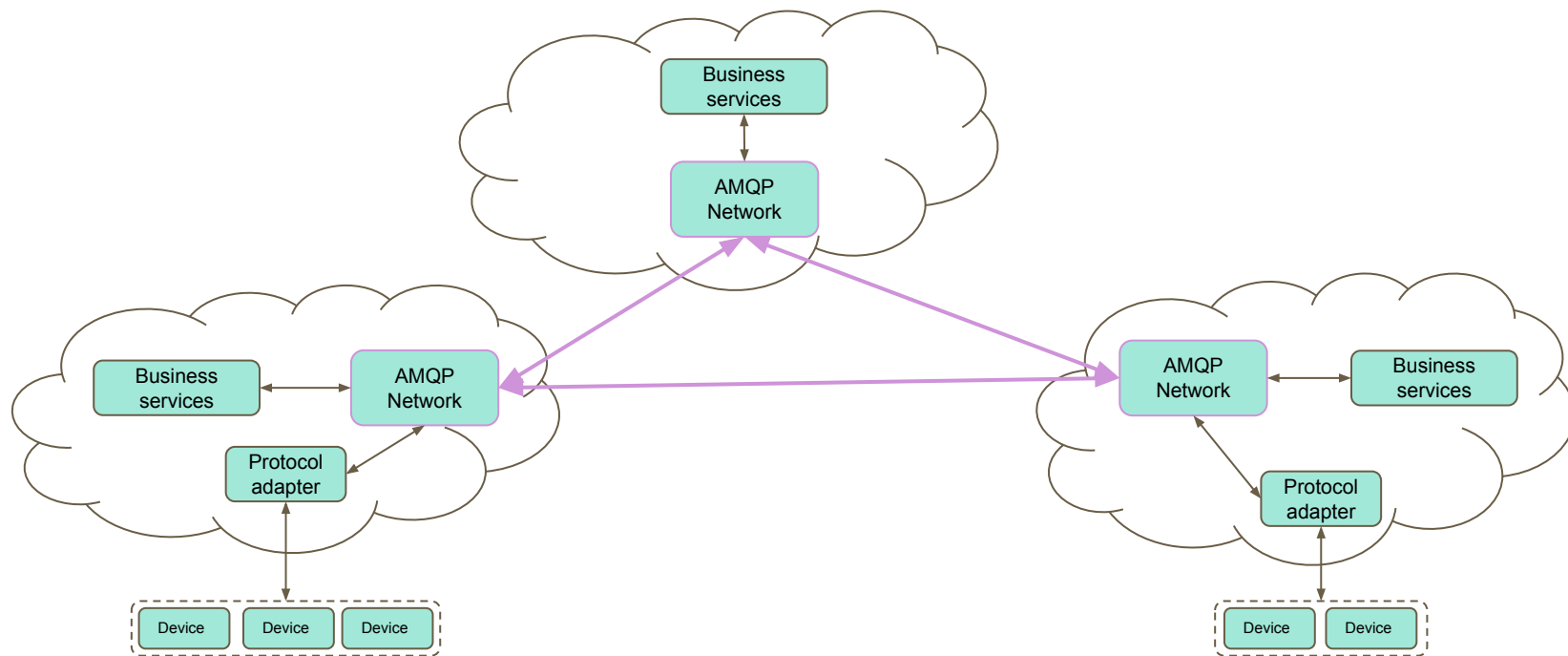
Way forward

- Cloud is not obsolete
- Cloud IoT platforms still needed
 - Hono
 - Ditto
- Work on distributed Edge deployments for IoT services

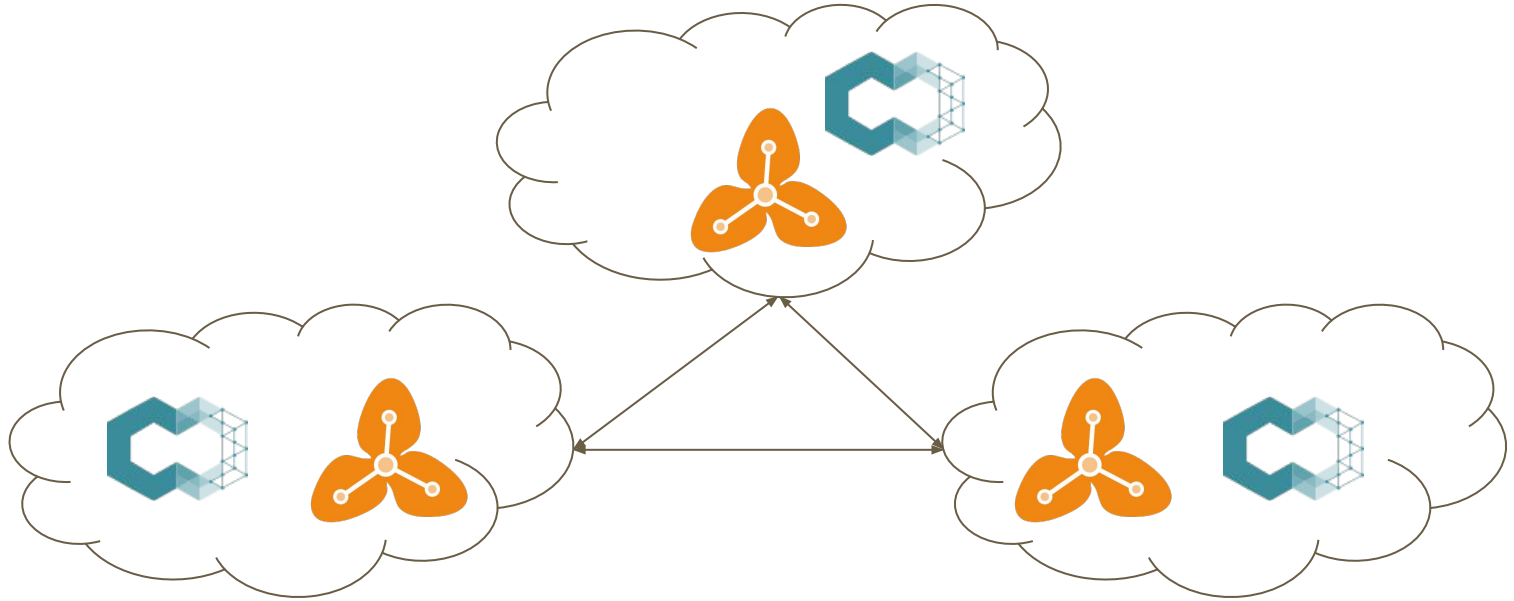
Eclipse Hono



Eclipse Hono



Eclipse Ditto



Next generation "gateways"

- Move to cloud native development of gateways services
- More compute resources
 - More caching
 - More analytics
 - More ML
- CI/CD

Questions?

[@dejanb](#)

[@kiltonhopkins](#)