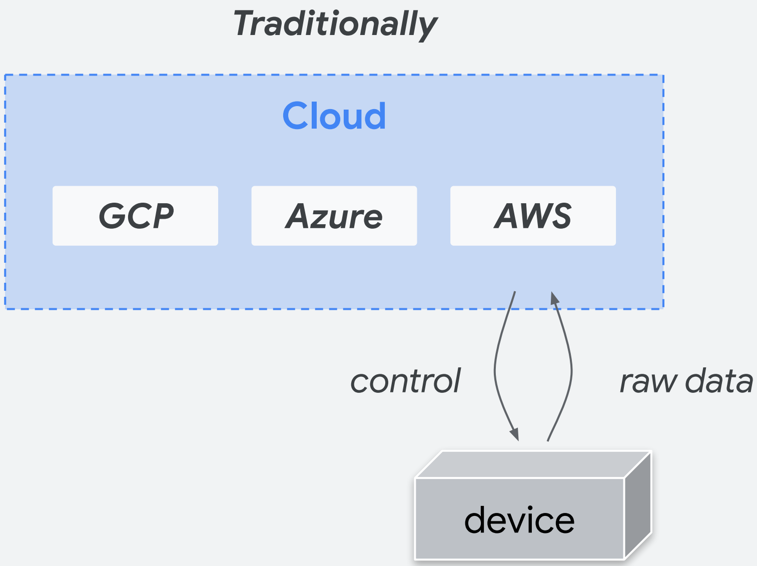
Challenges for Scaling TinyML Deployment (Part 1)

Port of Felixstowe
MSC SUEZ
PANAMA
M S C
Port of Felixstowe
ANNABA
MONROVIA
ELITE

Cloud Computing Paradigm

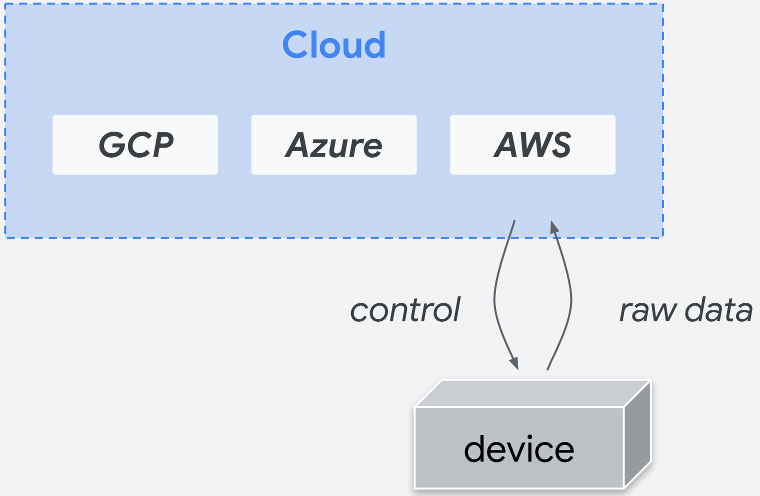
- Device == endpoint



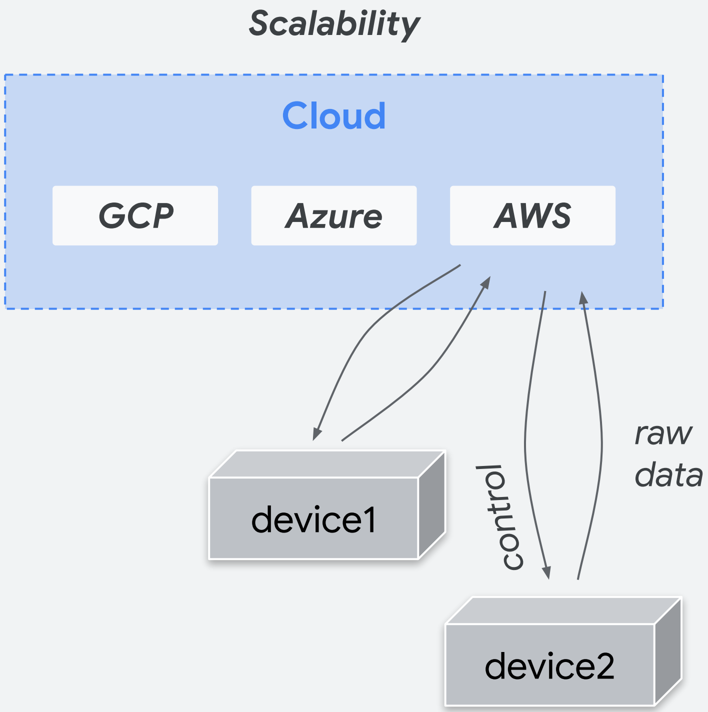
Cloud Computing Paradigm

- **Device == endpoint**
- Device is typically running some complex OS stack
- Device is probably a x86 or ARM-based processor that is widely deployed

Traditionally

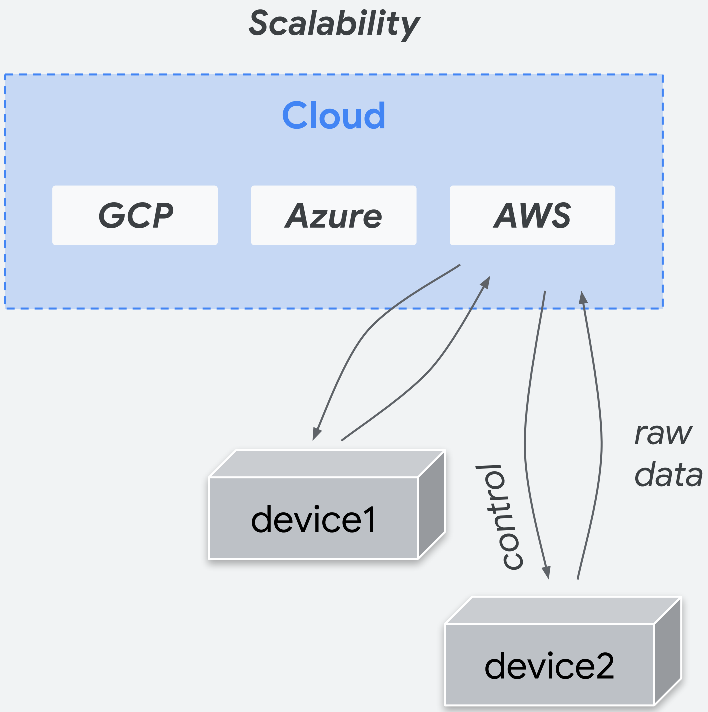


Cloud Computing Paradigm



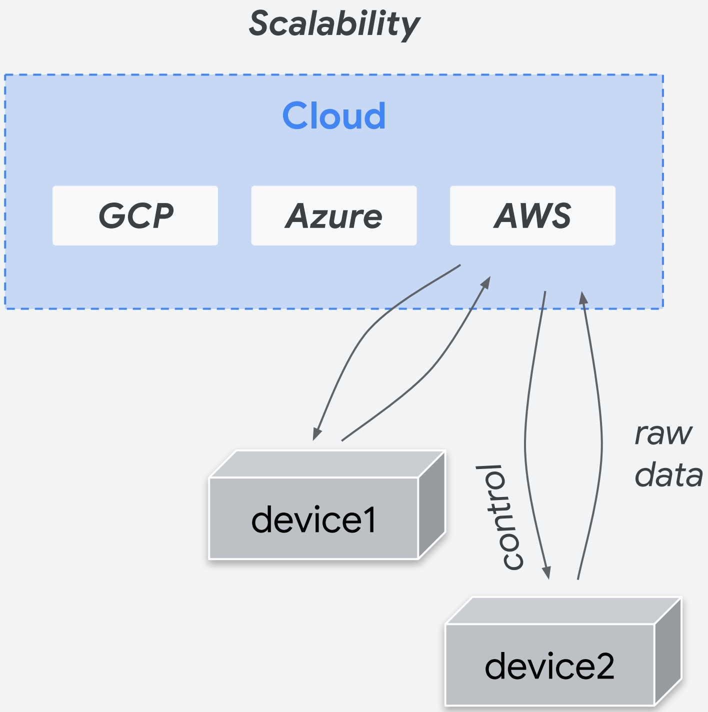
Cloud Computing Paradigm

- If we want scalability, we replicate the inference points

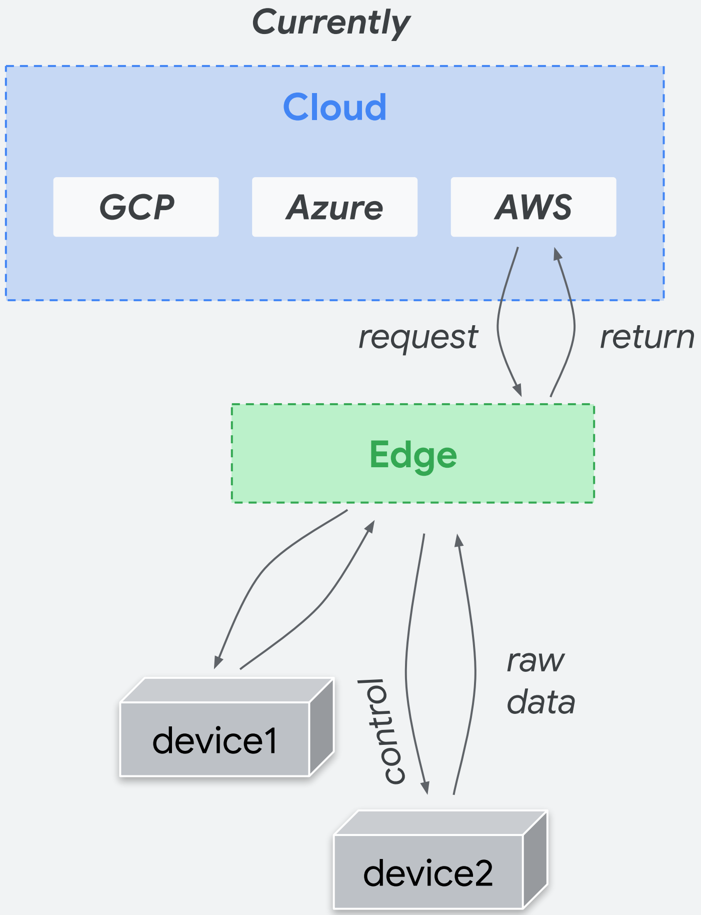


Cloud Computing Paradigm

- If we want scalability, we replicate the inference points
- Containers help with scalability on the cloud server side

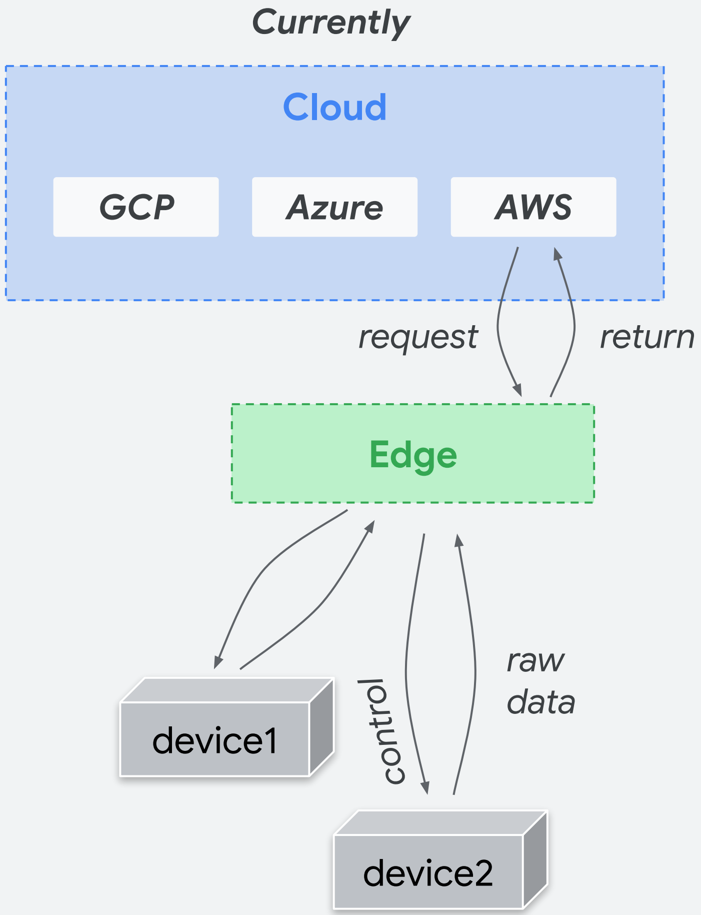


Edge Computing Paradigm

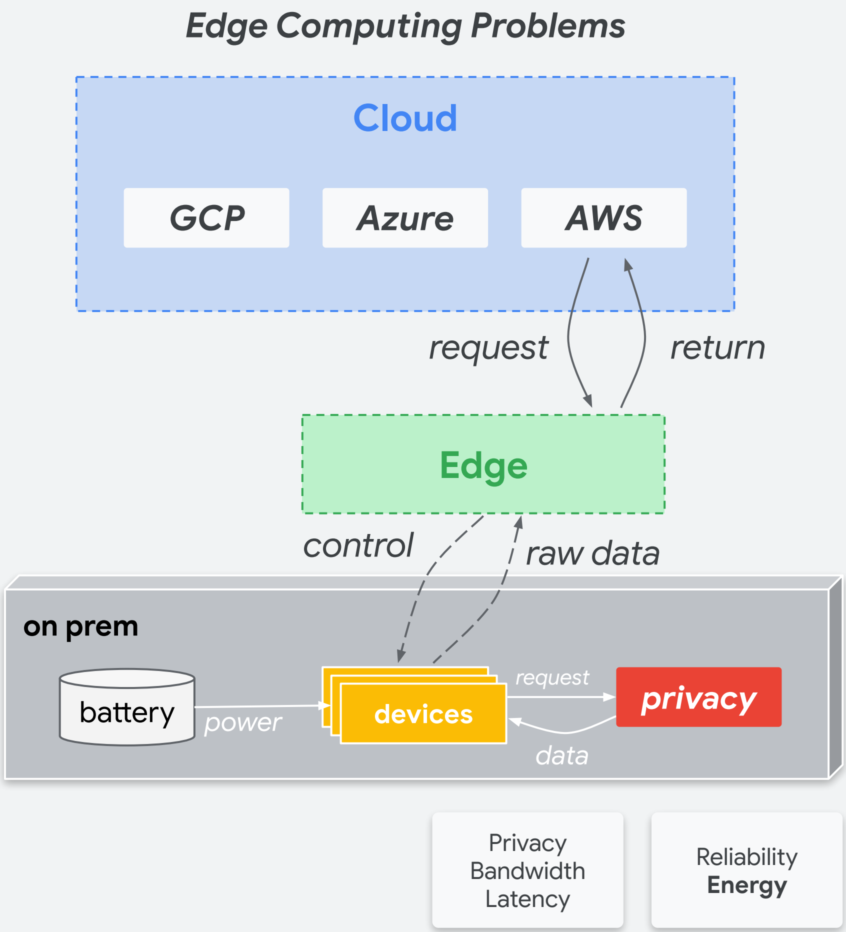


Edge Computing Paradigm

- Deployment is done behind the edge server that connects an enterprise to the cloud network
- Devices are plugged into the wall for power



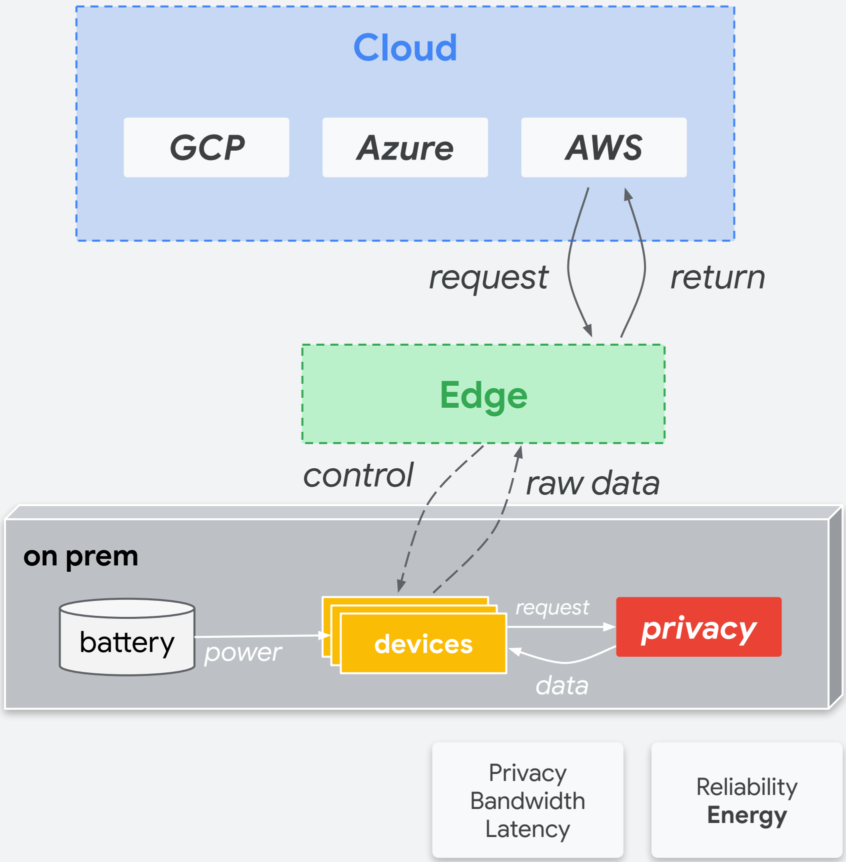
Edge Computing Paradigm



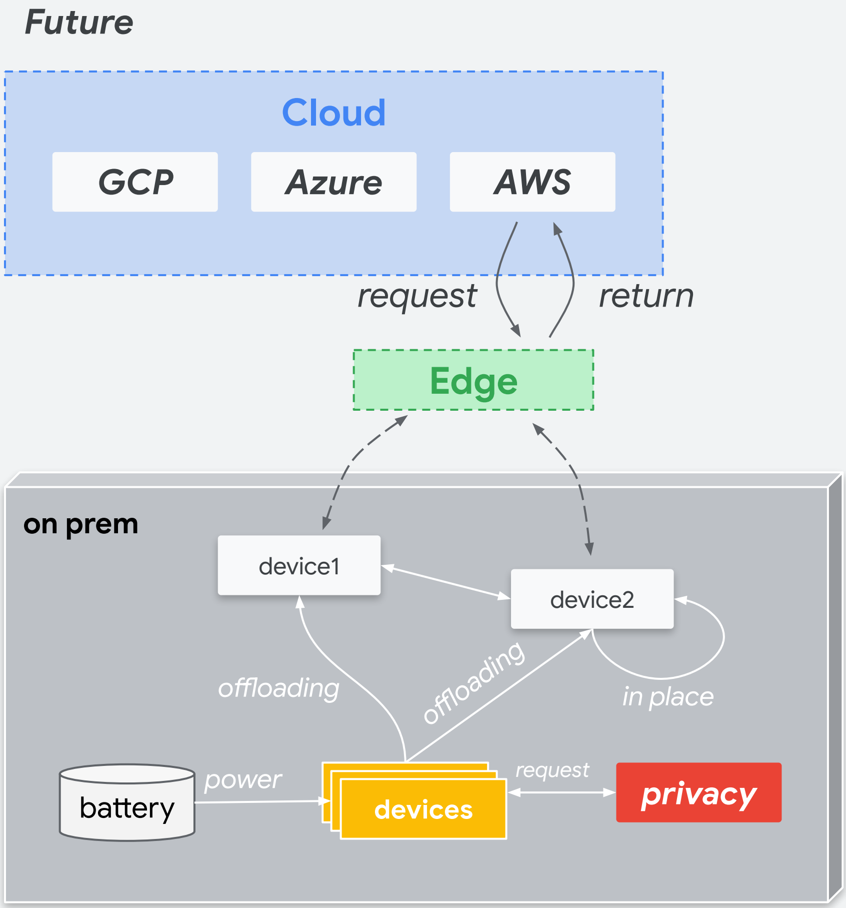
Edge Computing Paradigm

- Devices are connected to the edge server
- Data is transferred from the endpoint devices to the edge server continuously
- Problem is with the energy consumption of the devices

Edge Computing Problems

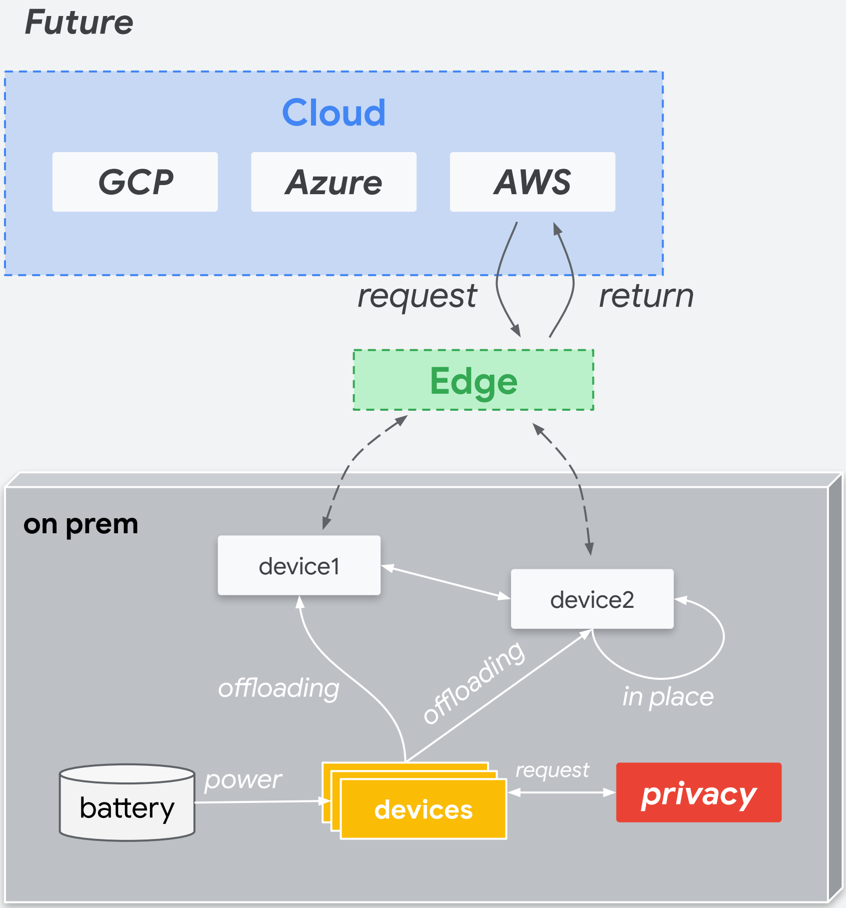


Embedded Computing Paradigm

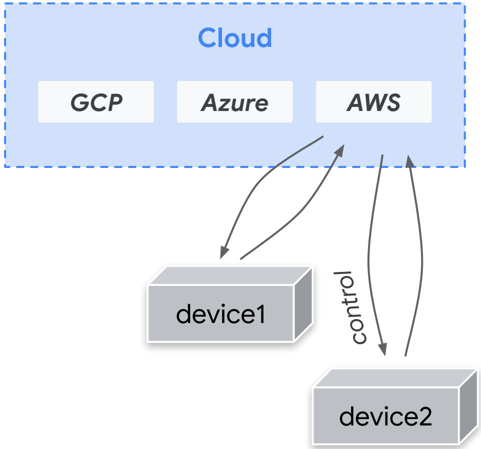


Embedded
Computing Paradigm

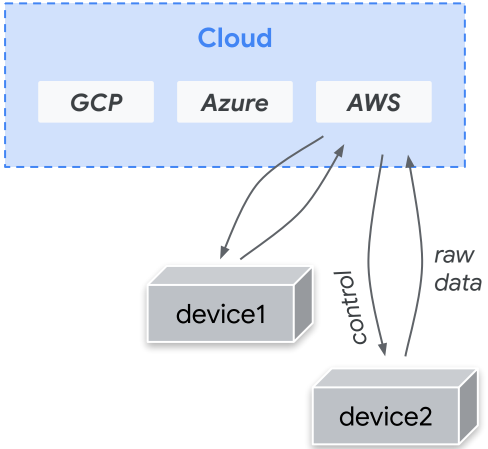
- Endpoint devices offload the compute to more servers using low-energy protocols to conserve energy
- The more power-hungry devices are connected to the wall power
- They connect over the edge to send the data to the cloud server



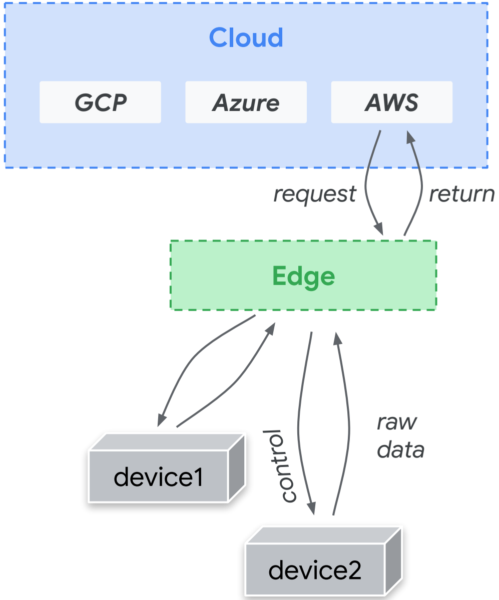
Cloud



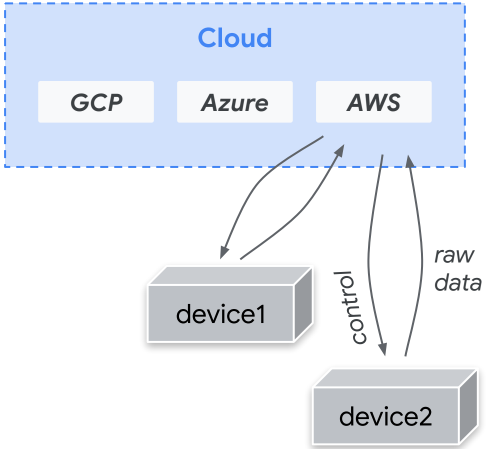
Cloud


Cloud
GCP
Azure
AWS
control
raw
data
device1
device2

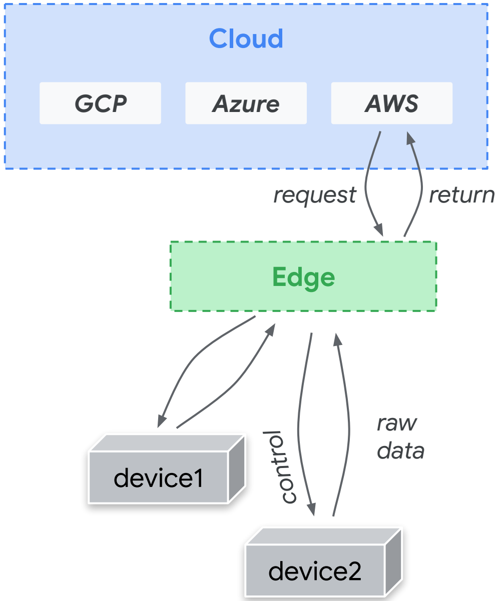
Edge


Cloud
GCP
Azure
AWS
request
return
Edge
control
raw
data
device1
device2

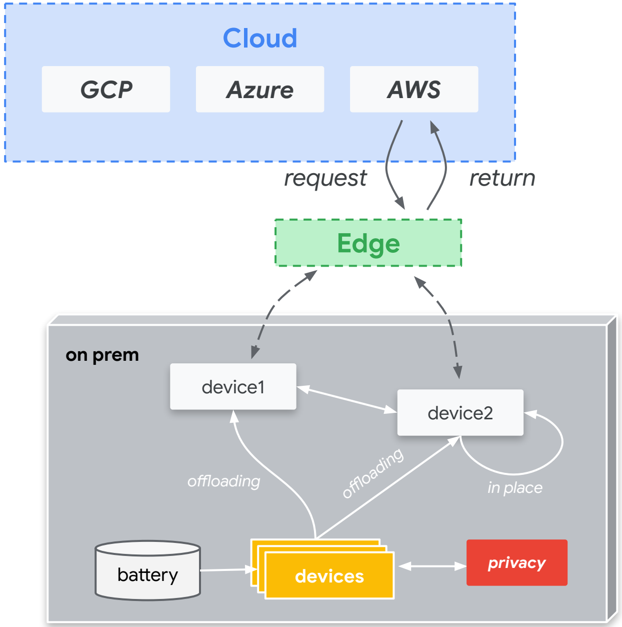
Cloud



Edge

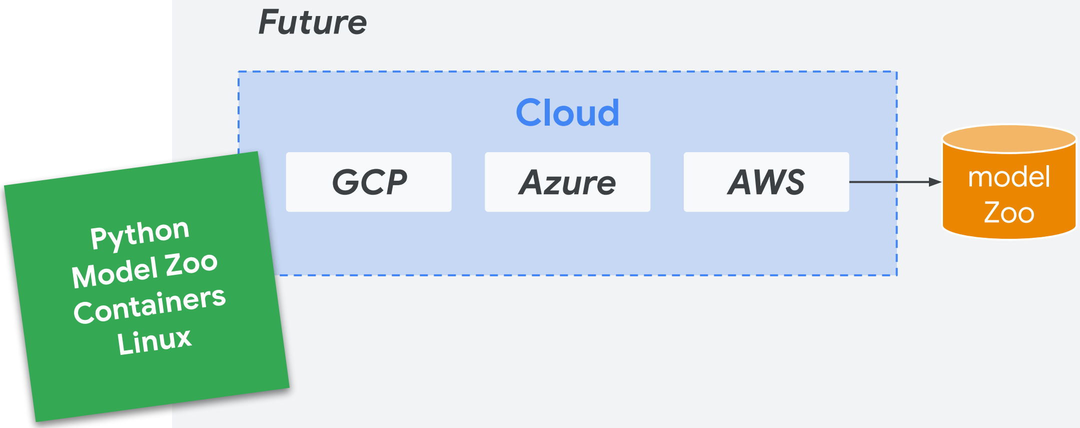


Embedded



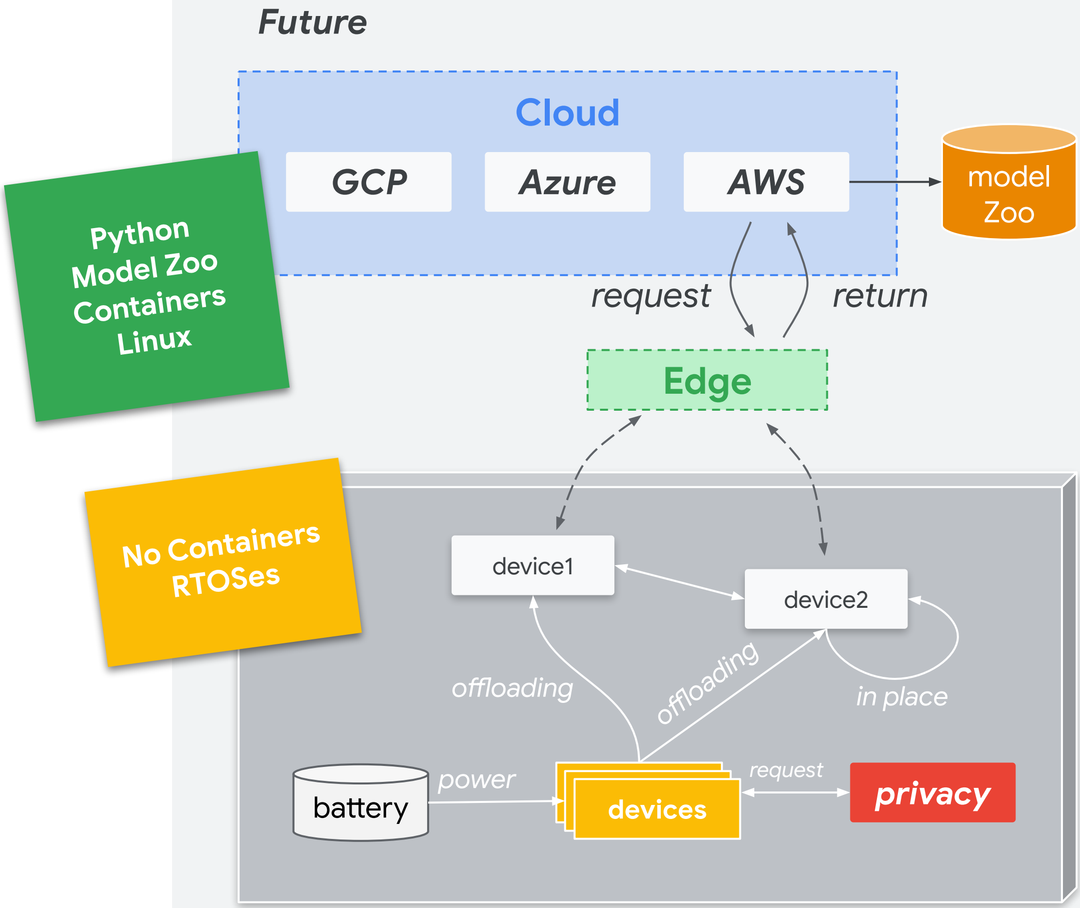
MLaaS Challenges with TinyML

In the **non**-embedded systems side of the world, we have **rich software services** and **operating environments**



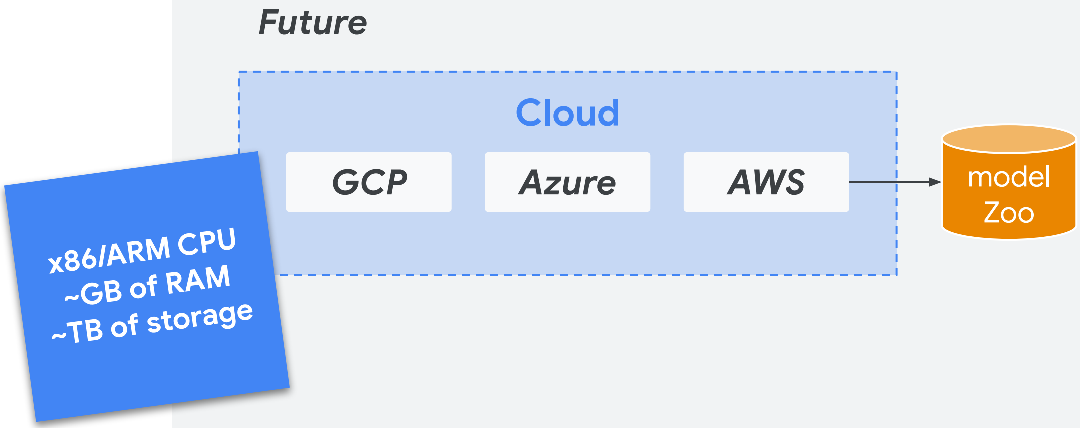
MLaaS Challenges with TinyML

In the **non**-embedded systems side of the world, we have **rich software services** and **operating environments**



MLaaS Challenges with TinyML

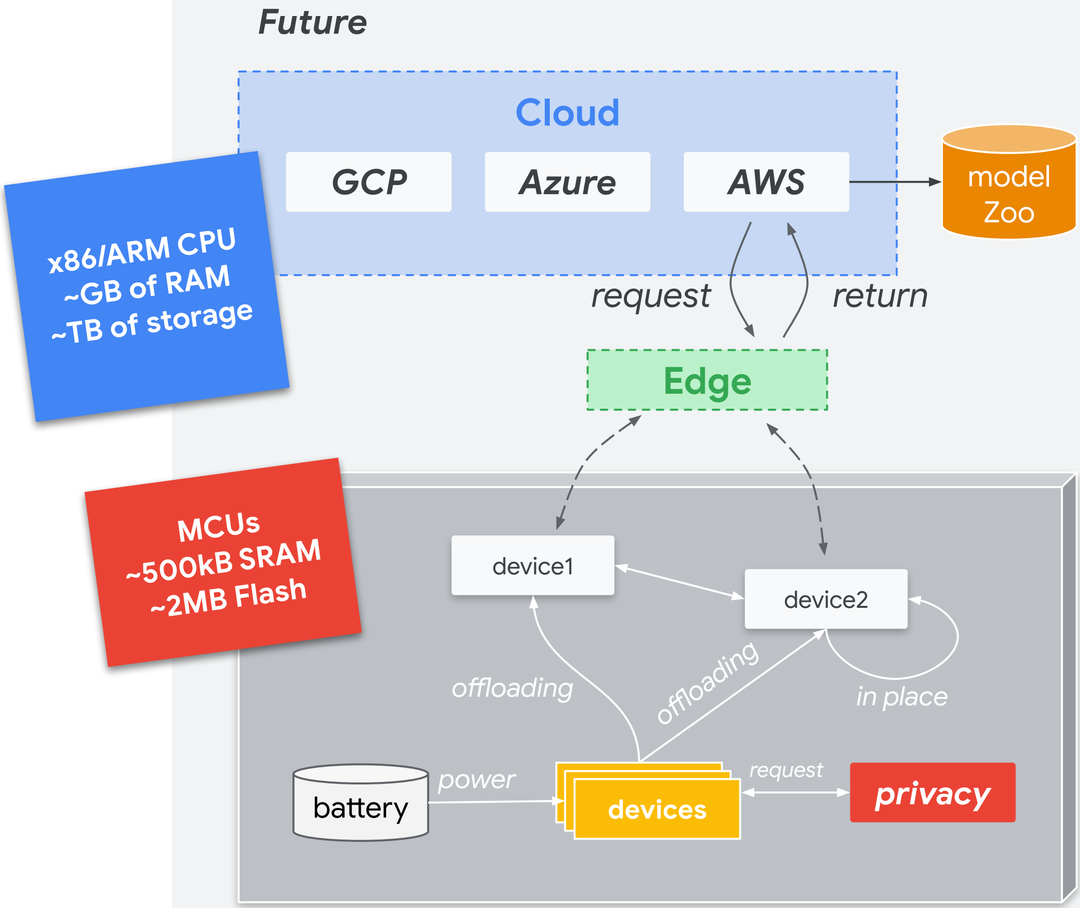
On the cloud side, we have "infinite" amount of hardware resources



MLaaS Challenges with TinyML

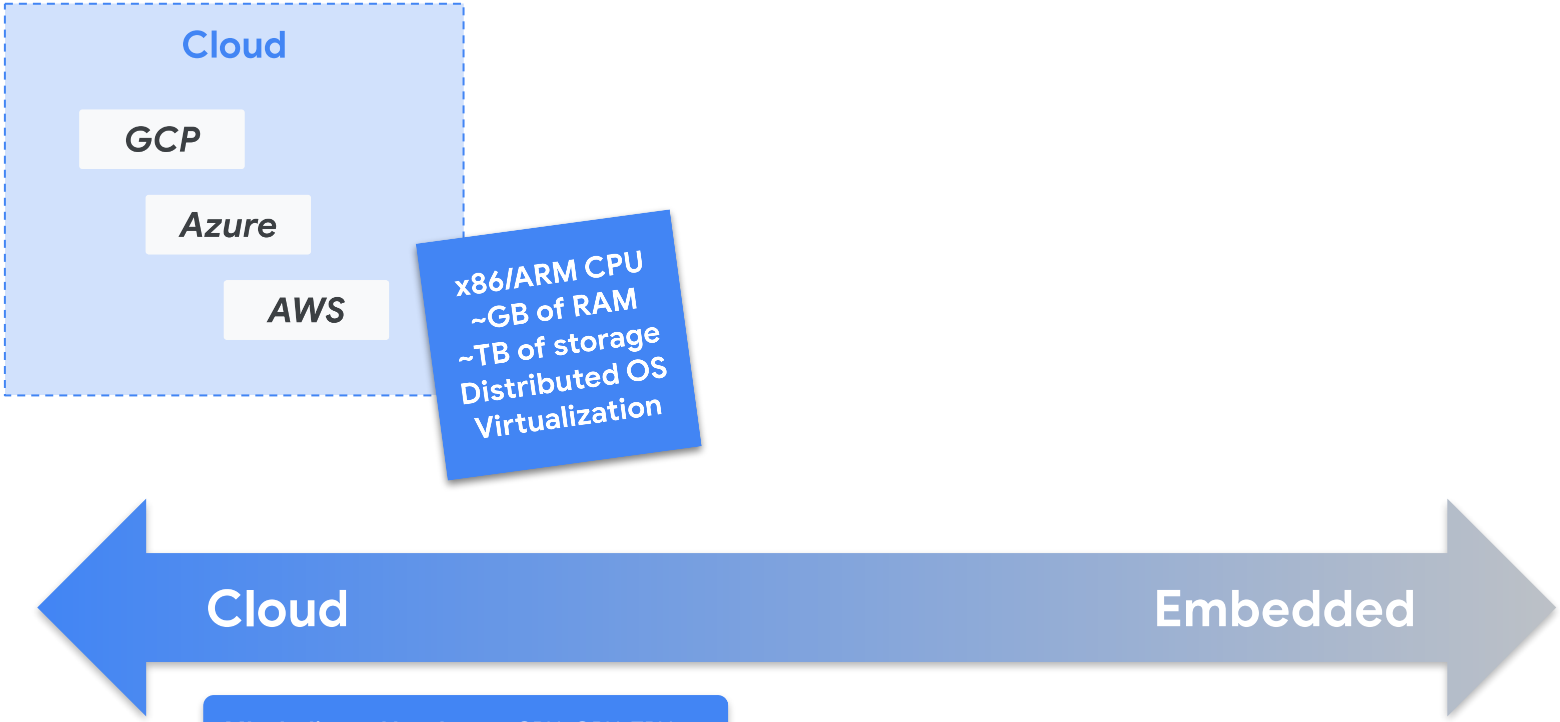
On the cloud side, we have "infinite" amount of hardware resources

On the other hand, at the embedded scale, we have severe **resource constraints**



Cloud

Embedded



Cloud

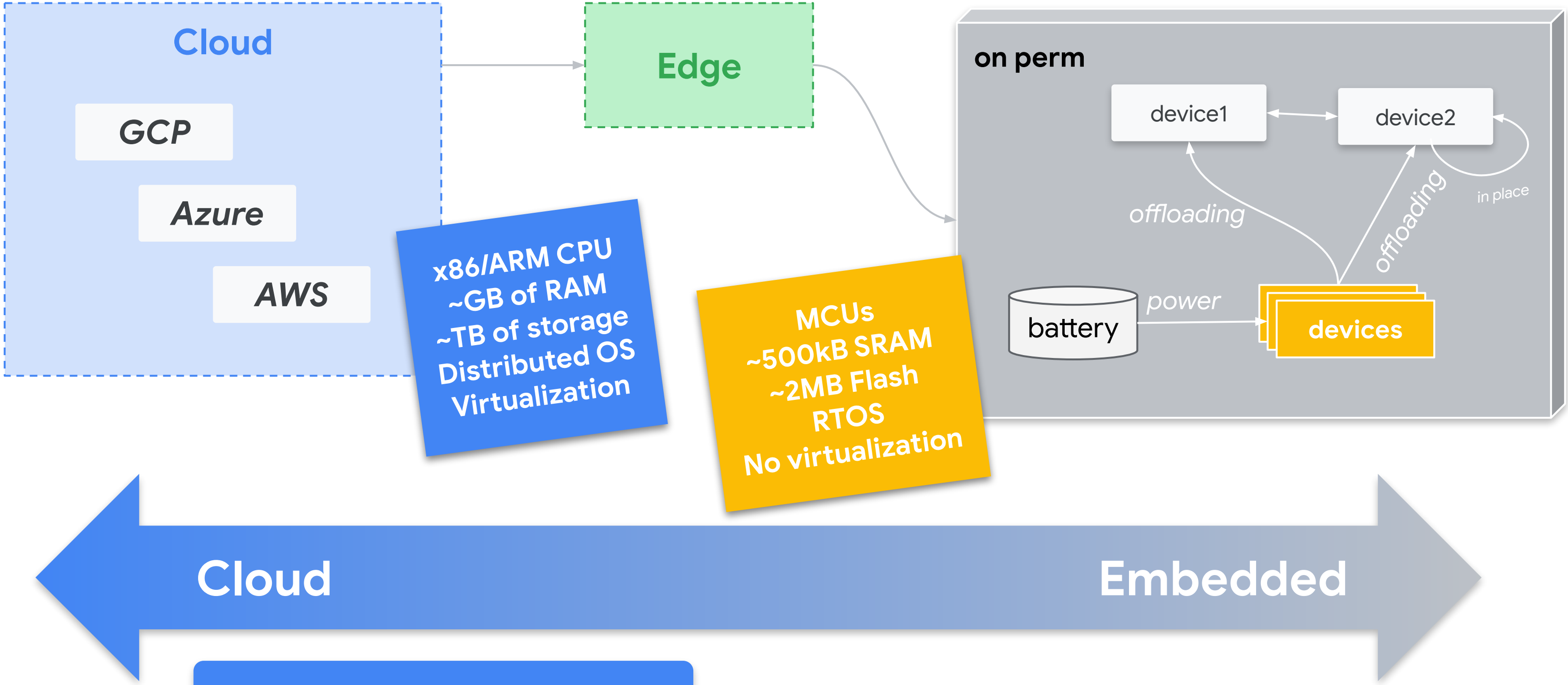
GCP

Azure

AWS

x86/ARM CPU
~GB of RAM
~TB of storage
Distributed OS
Virtualization

ML-dedicated hardware: CPU, GPU, TPU
ML-dedicated software: many tools
ML Tasks ⟶ Data collection and preprocessing, data transformation, model training, model deployment, inference



x86/ARM CPU
~GB of RAM
~TB of storage
Distributed OS
Virtualization

MCUs
~500kB SRAM
~2MB Flash
RTOS
No virtualization

ML-dedicated hardware: CPU, GPU, TPU
ML-dedicated software: many tools
ML Tasks ⟶ Data collection and preprocessing, data transformation, model training, model deployment, inference

No ML-dedicated Hardware
No ML-dedicated software
ML Tasks ⟶ Inference