COMPUTE UTILITIES AND EDGE WORKERS

Mechanisms to Unify Cloud, Edge, and Client Web Programming Models

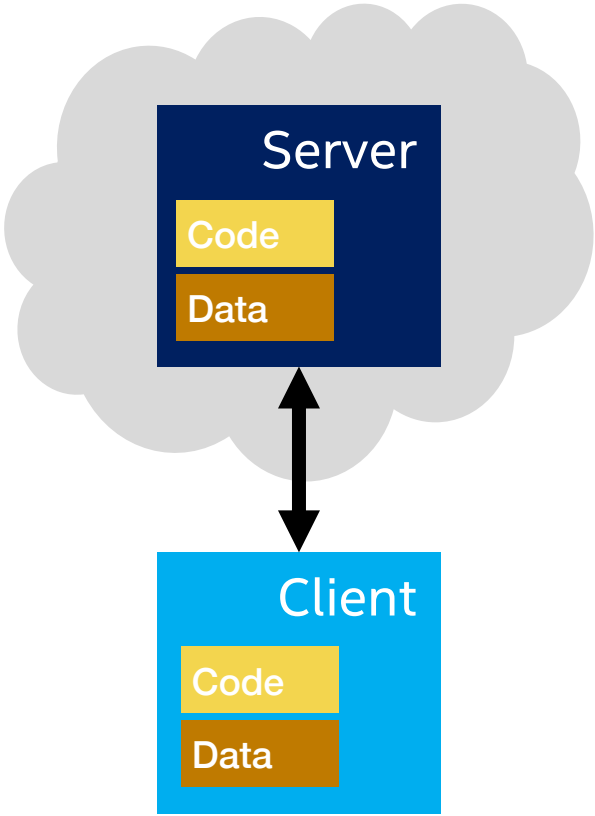
Michael McCool and Sudeep Divakaran

Intel Corporation

October 22, 2021

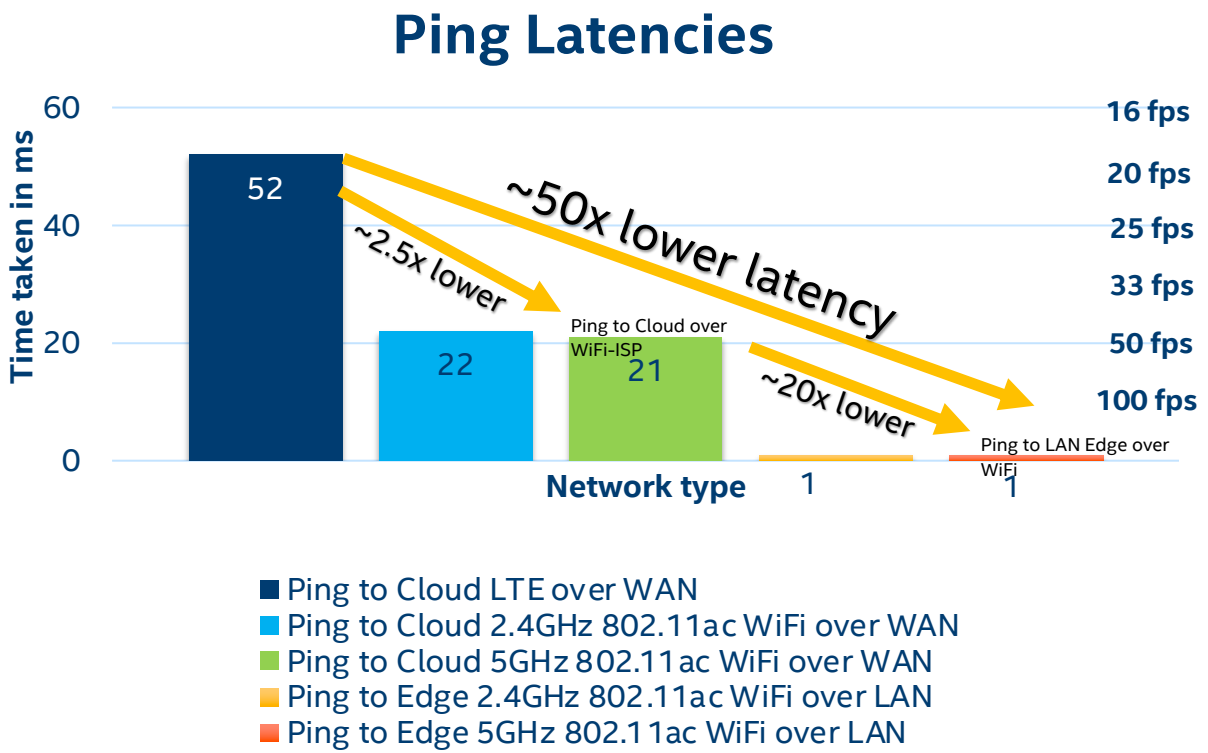
Web Programming Models: Cloud, Client, and Edge

- Edge provides additional resources
- Edge resources are *location sensitive*
  - *Network latency*
  - *Ownership*
- ***WHY use edge resources?***
- ***WHO uses edge resources?***
- ***HOW to use edge resources?***



Why Use the Edge?

- Privacy
- Performance
  - More available power
  - Higher thermal envelope
  - Parallelism
  - Use of special features
  - Use of accelerators
- Latency

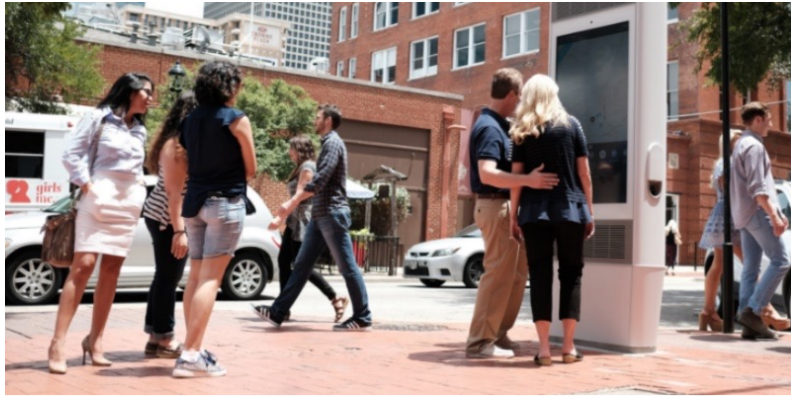


Who: Public Use Cases

Retail

Store customer searching for item in store inventory based on an image

City



City visitor accessing air quality data along jogging route

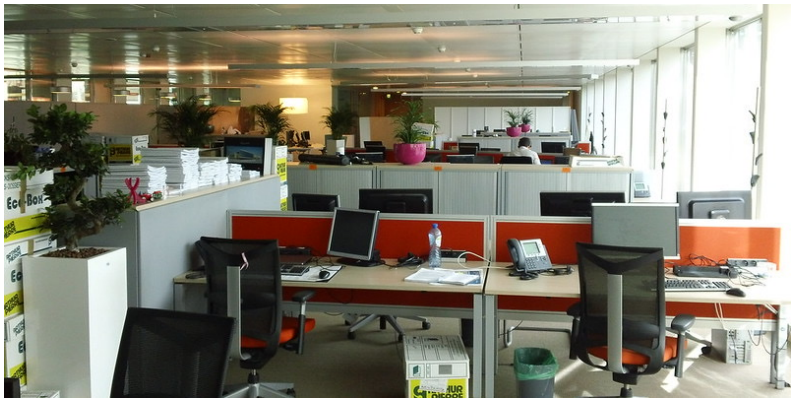
- These use cases might also benefit from background execution and access to IoT devices from the edge worker, e.g. to monitor air quality continuously and generate notifications.
- This presentation, however, primarily focuses on edge workers for performance enhancement.

Who: Private Use Cases

Home

Home resident (1) Playing game and video chatting while exercising (2) Offload of video analytics for robot navigation.

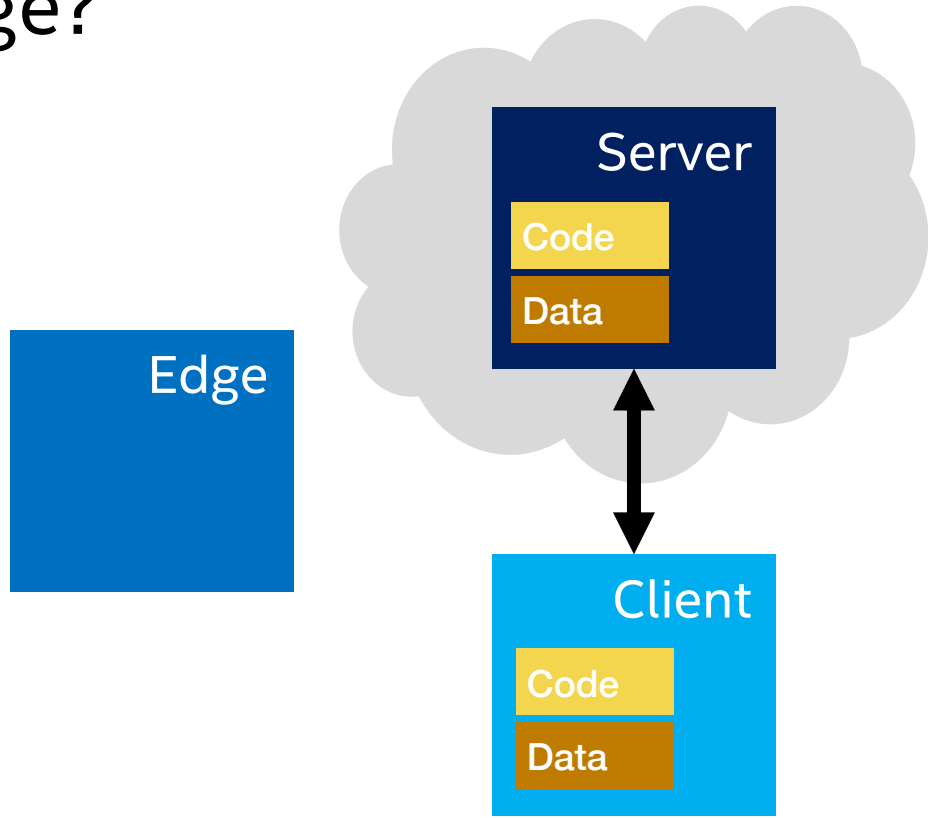
Office



Office worker accessing business and engineering data to fulfill customer order

- These use cases require or can benefit from privacy, although this also depends on the edge deployment model.

How to Program the Edge?



How to Program the Edge?

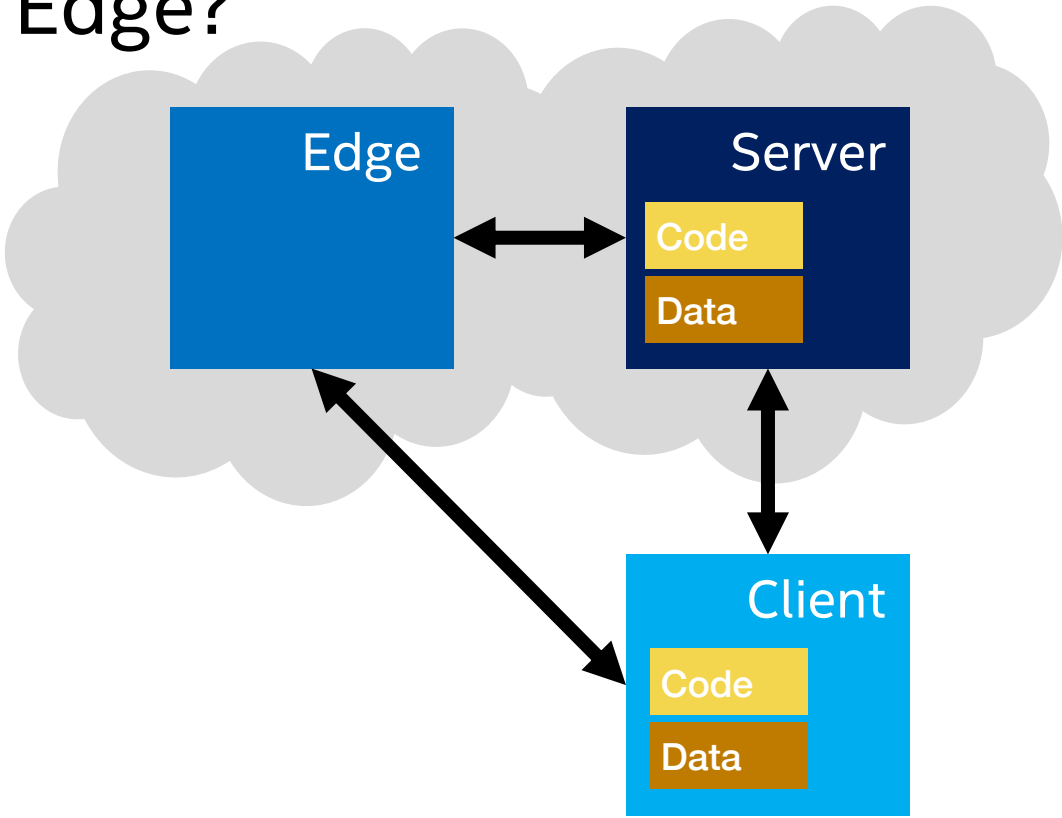
Extending the Cloud

Developer chooses when and where to run code

Deployment at scale

Centrally managed hardware

→ *Service model*



How to Program the Edge?

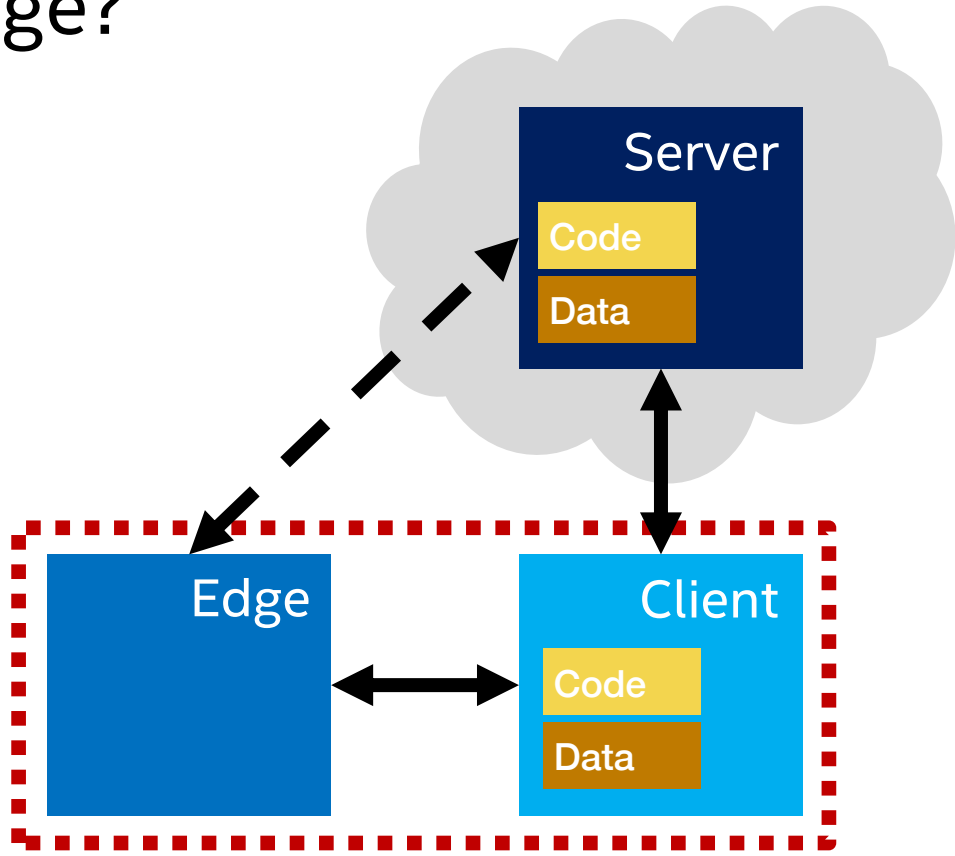
Extending the Client

User chooses when and where to run code

Control over private data

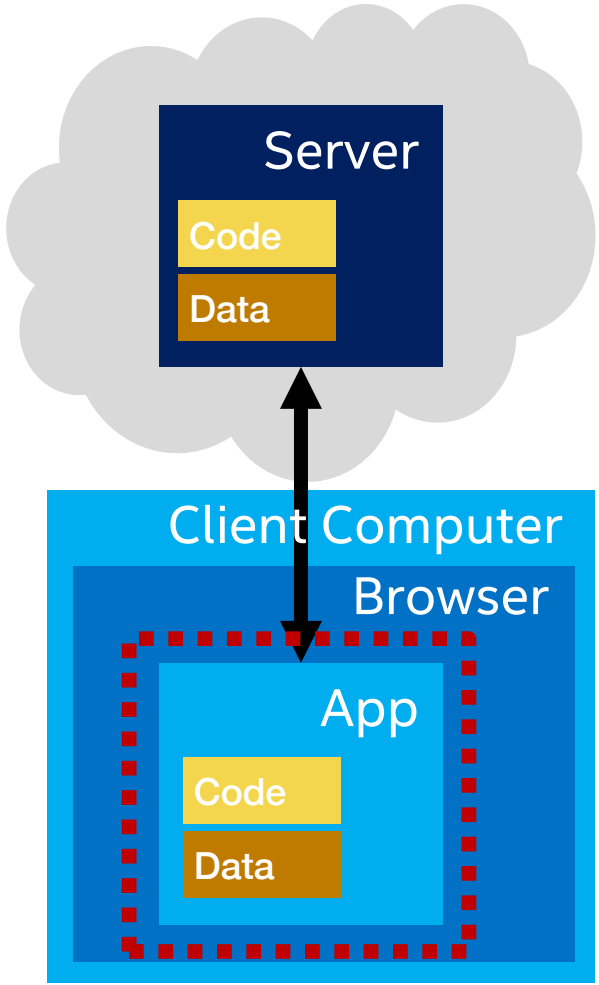
Use of privately-owned hardware

→ *Application model*



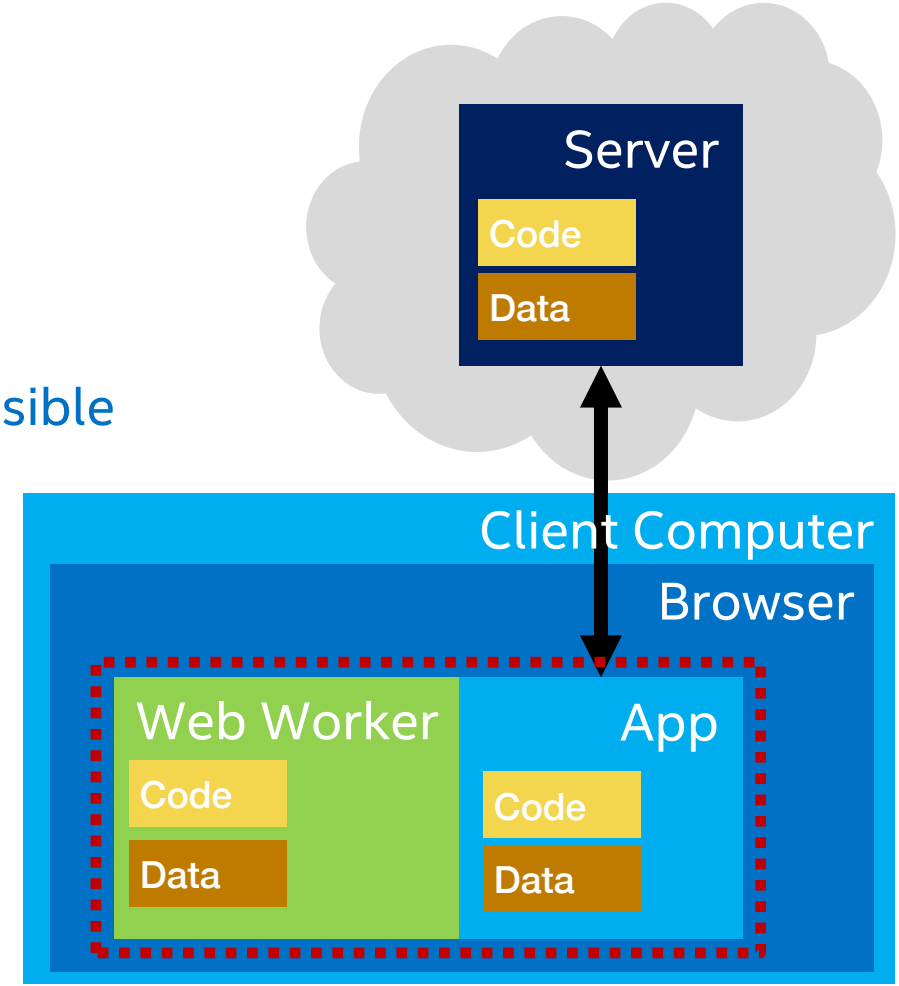
Web Browser: Runtime Platform

- Browser runs on client computer
- Runtime platform for web app
- Provides sandboxed execution context



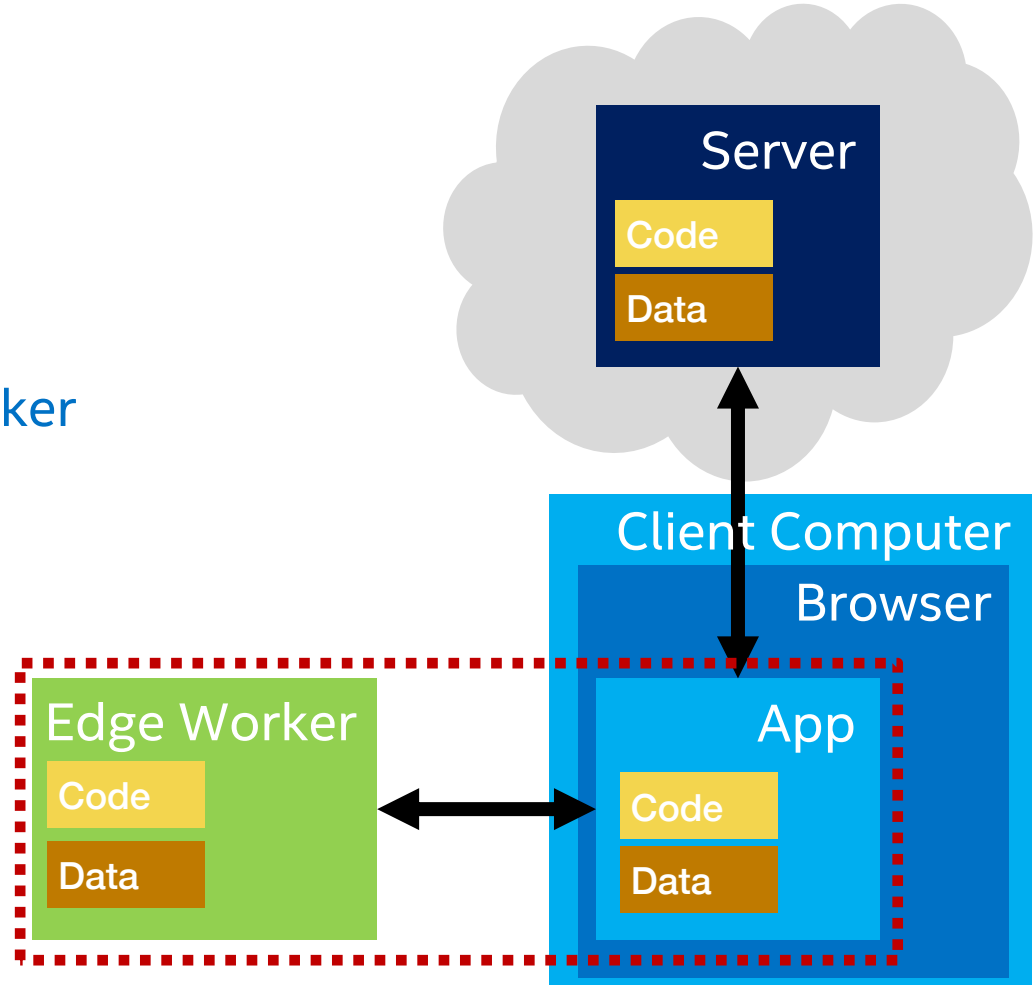
Web Workers

- Move compute to separate thread
- Multiple (parallel) web workers possible



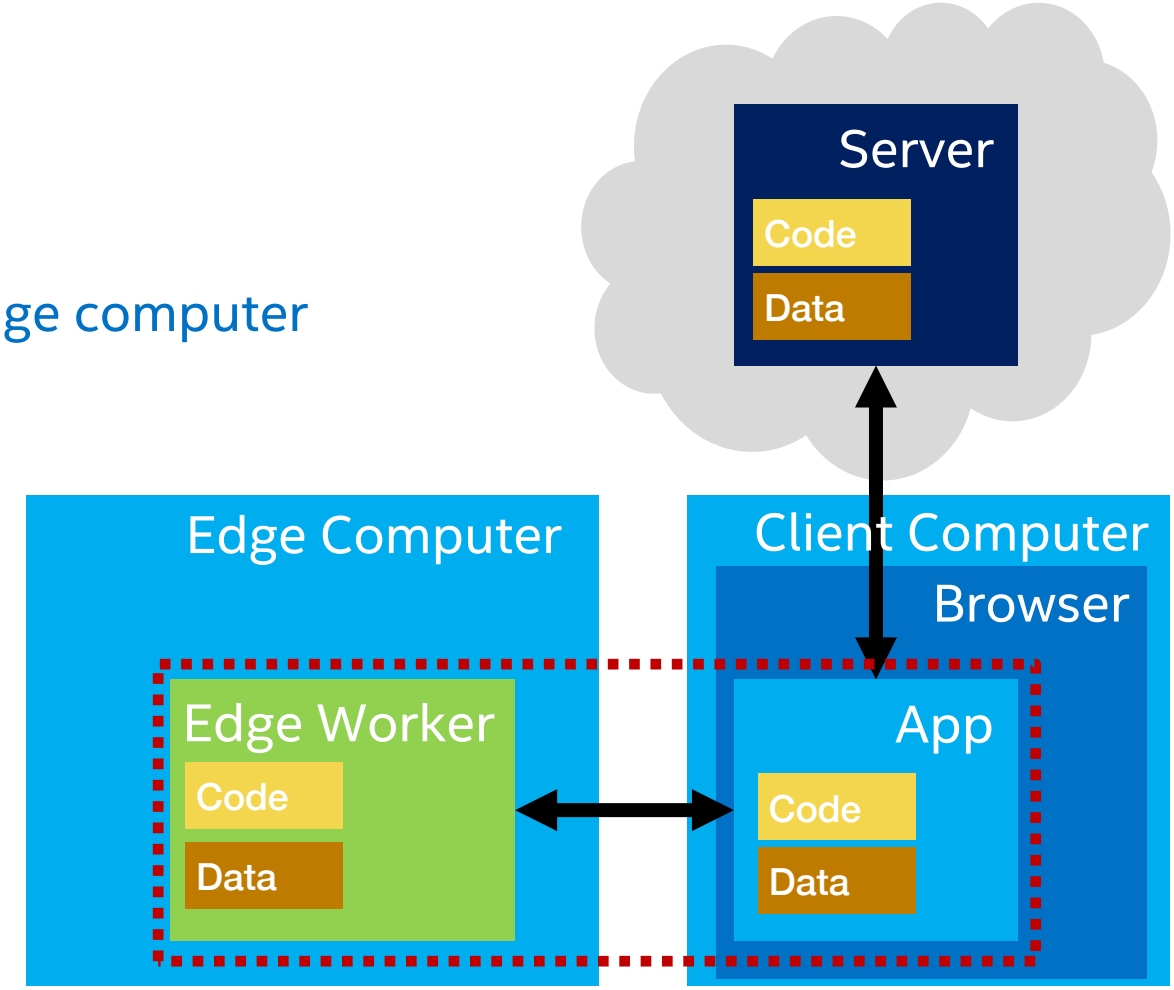
Edge Workers

- Network connected worker
- Same capabilities as web worker
- Runs "somewhere else"



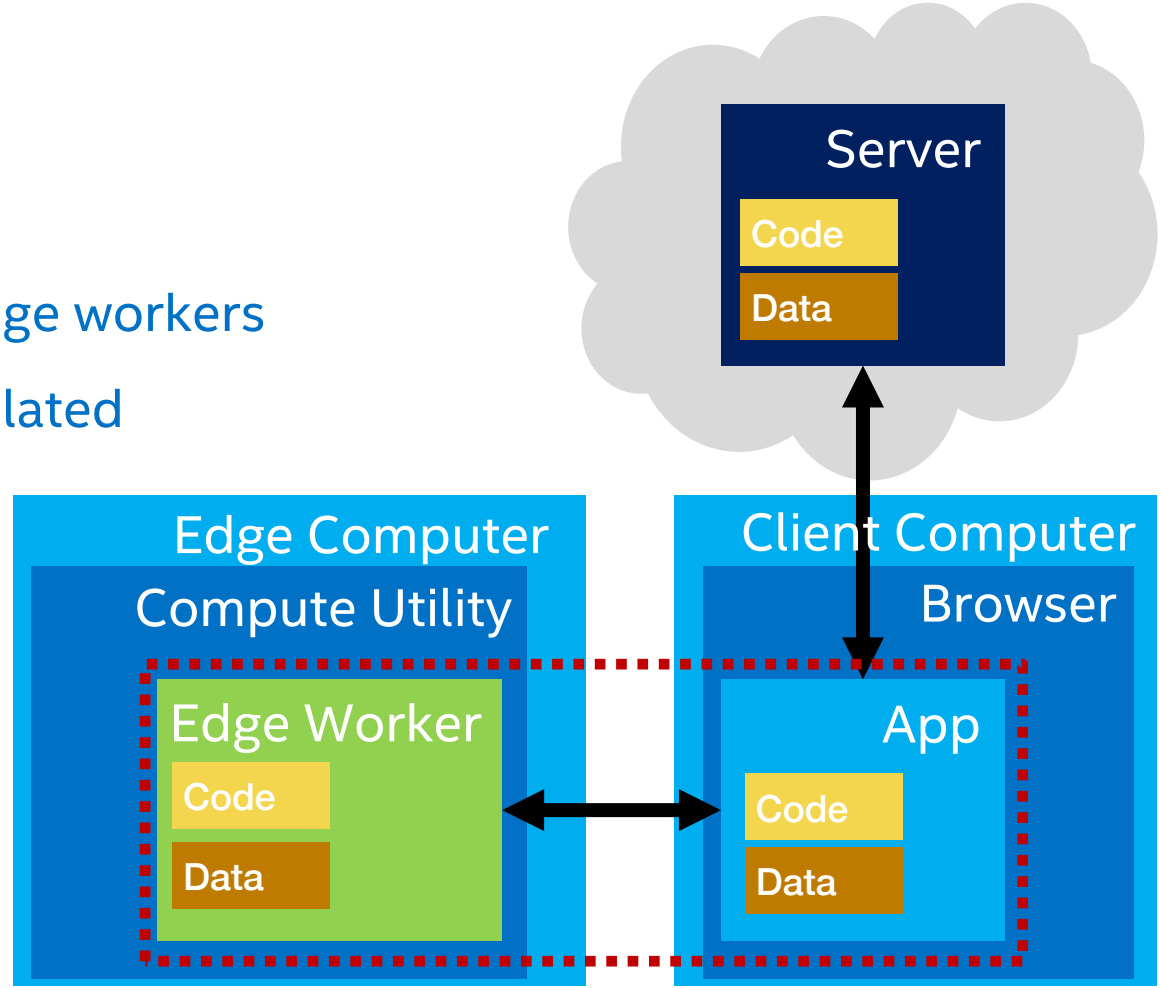
Edge Workers

- *May* run on separate edge computer



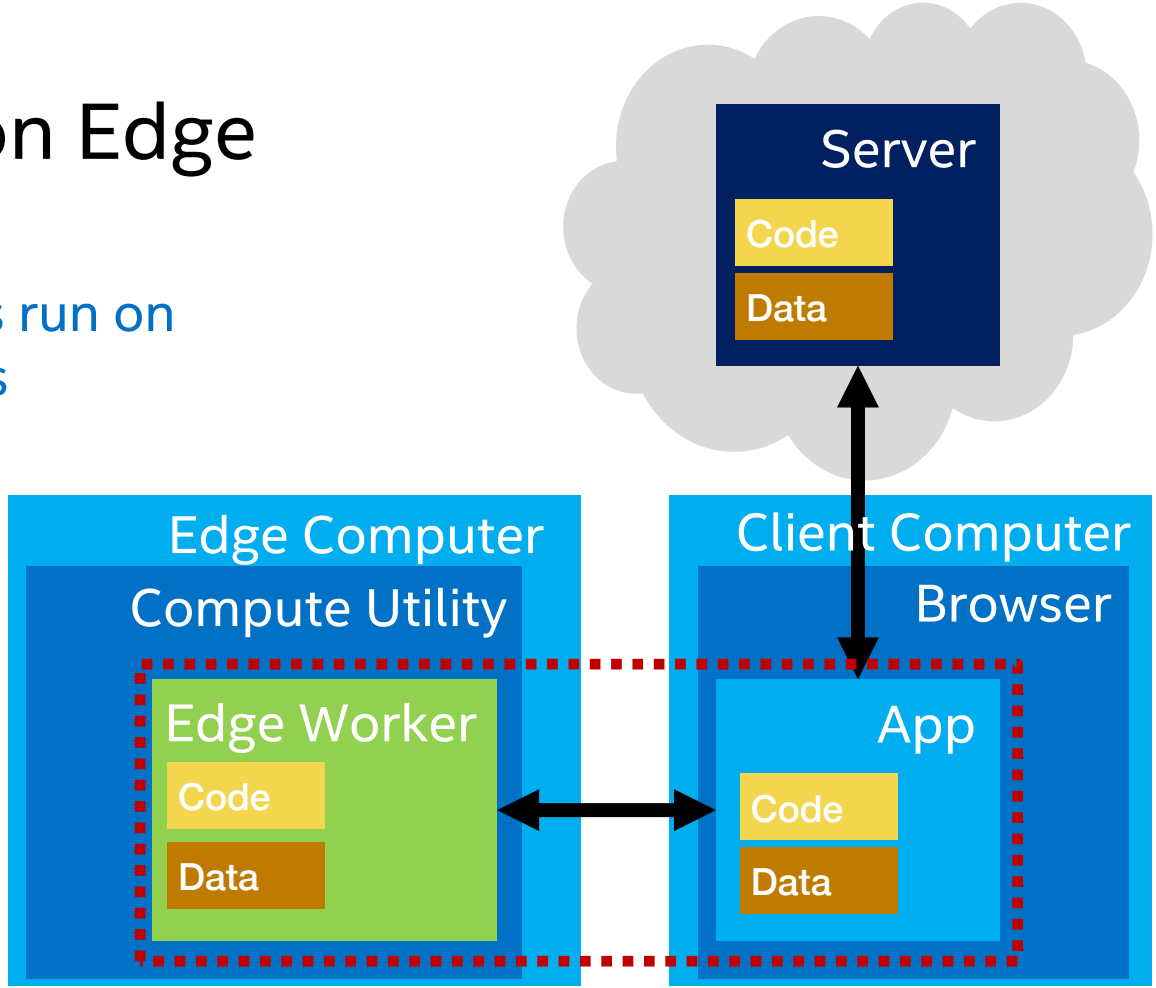
Compute Utility

- Runtime platform for edge workers
- Provides sandboxed/isolated execution context



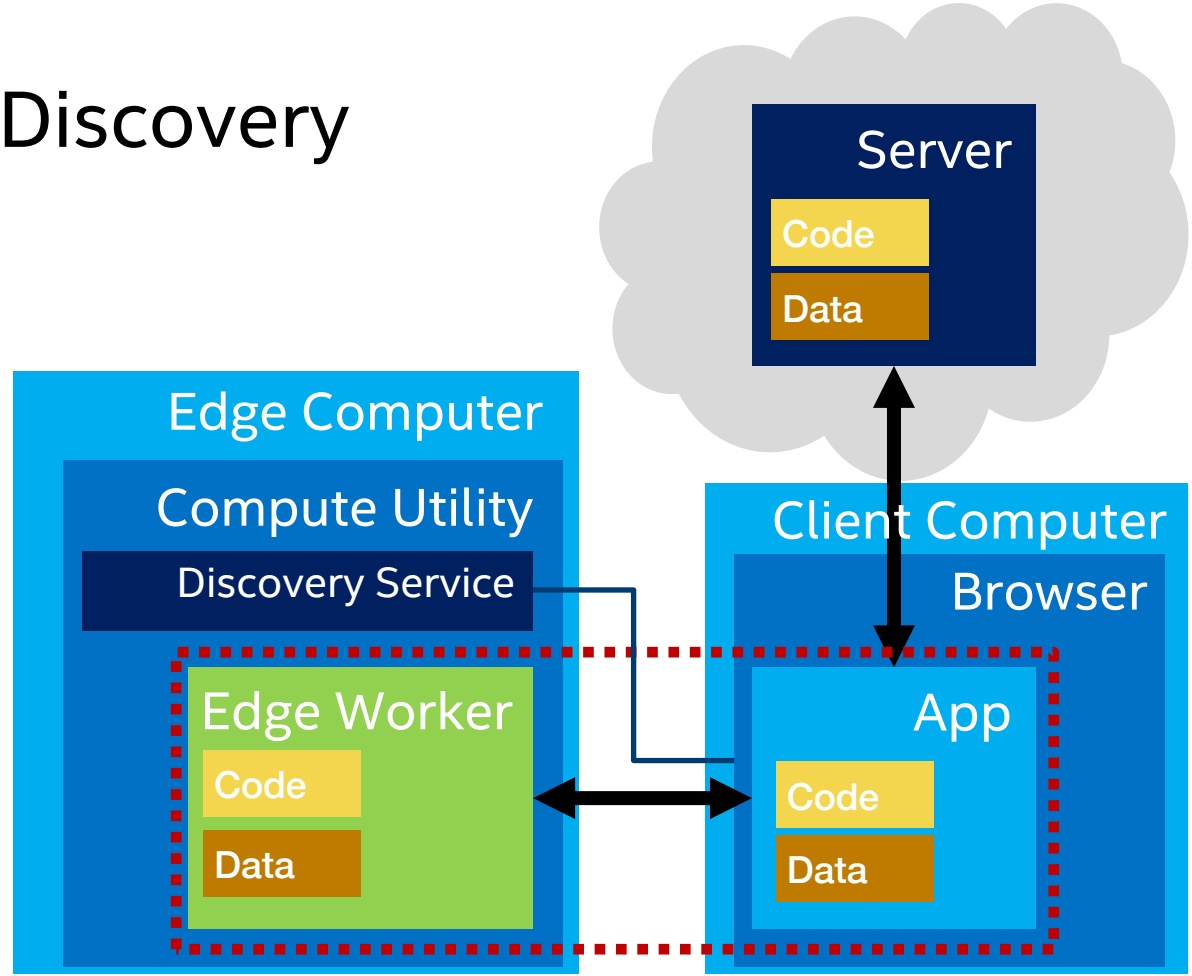
Compute Utility: on Edge

- In general, edge workers run on ***remote*** compute utilities
- ***HOW to find one?***
- ***HOW to decide to use?***



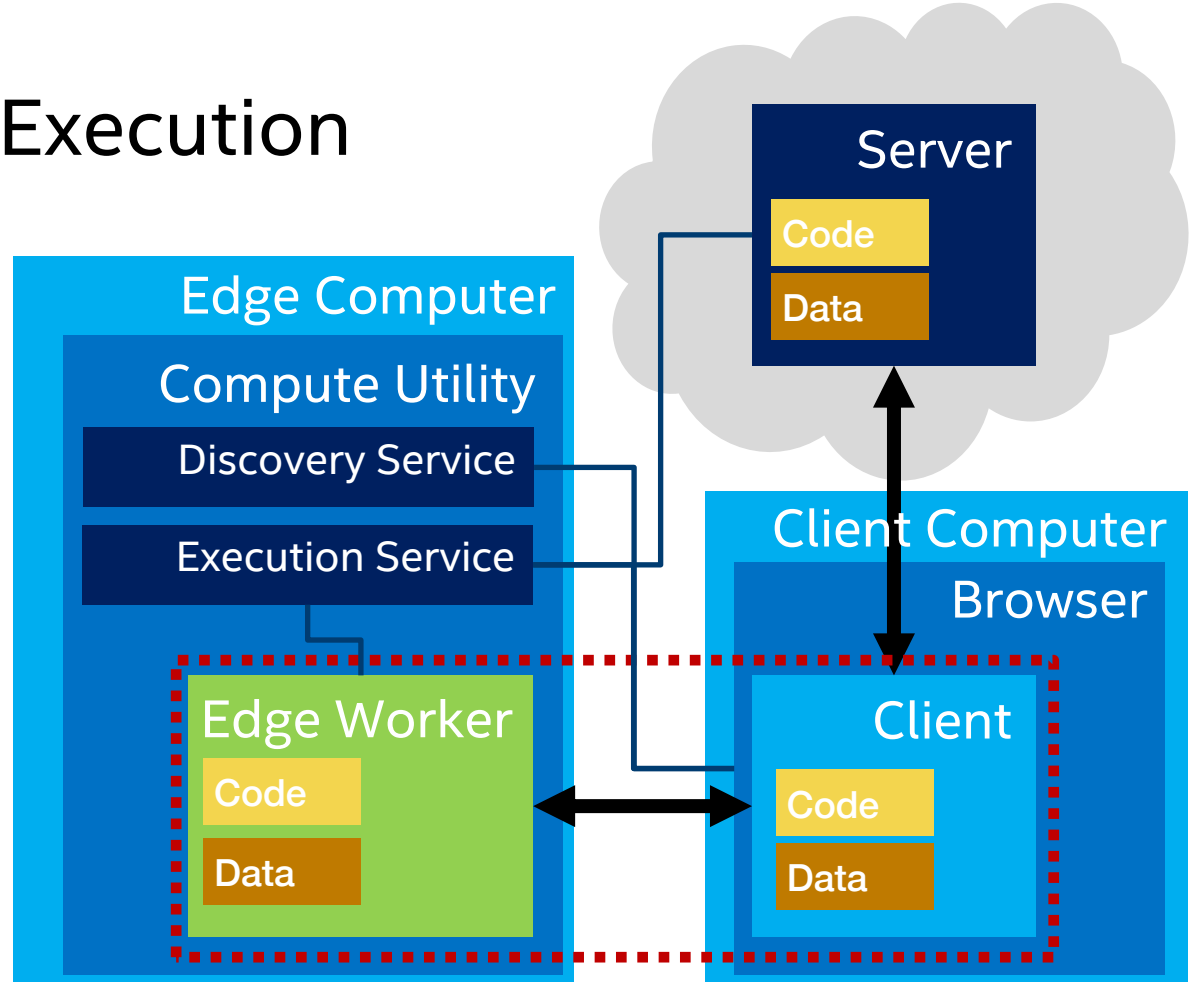
Compute Utility: Discovery

- Find remote compute utilities
- Provide metadata
  - Network
  - Performance
  - Capabilities



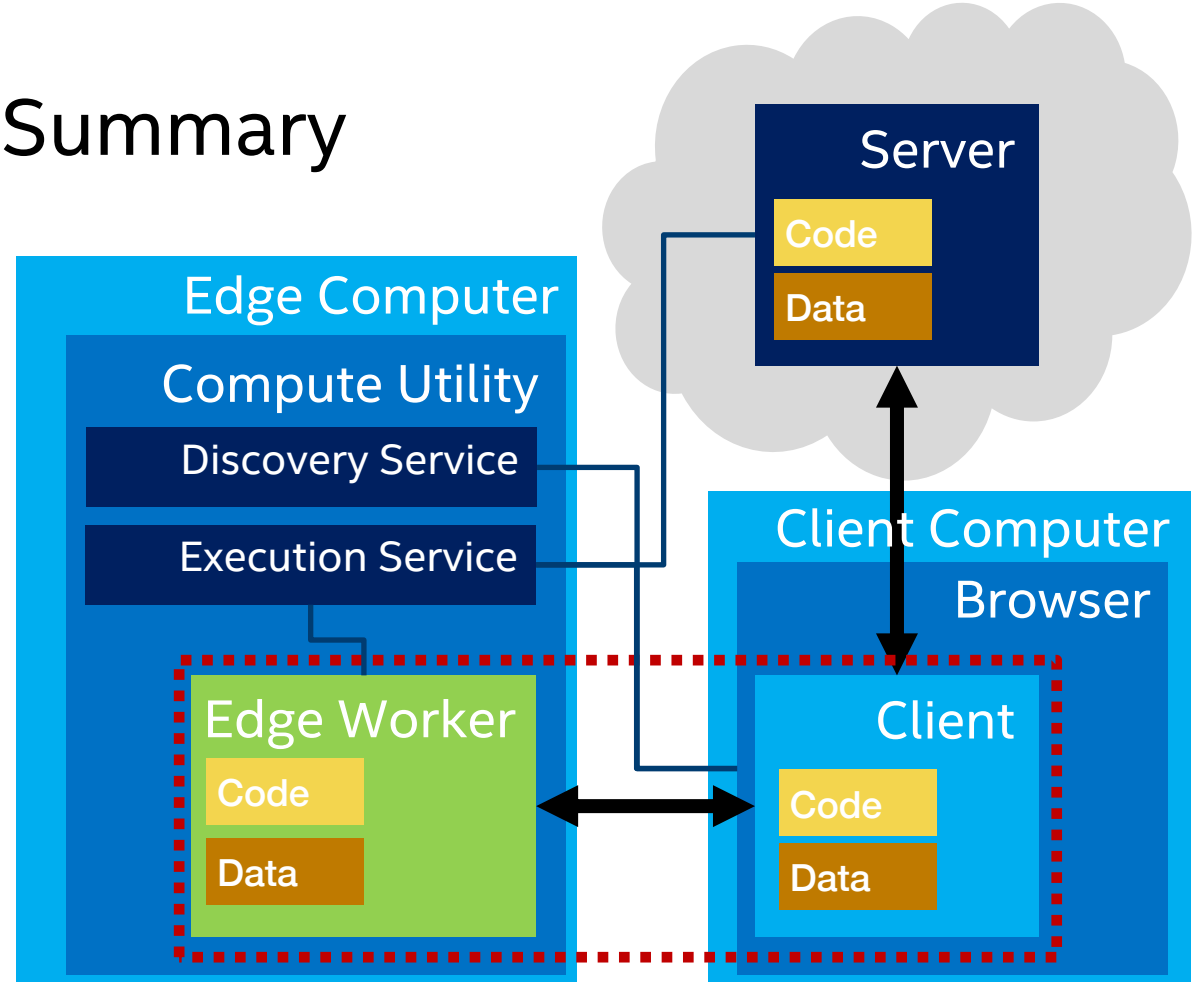
Compute Utility: Execution

- Loads and runs workload
- Workload packaging:
  - Javascript
  - WASM
  - Container images
- Need acceleration



Compute Utility: Summary

- Complementary role to browser
- Like the browser, just a program that can be run anywhere



Standards Work Needed

Network metadata	Can be based on extending IETF QoS standards
Performance metadata	Can be based on benchmark standards
Capability metadata	Standards needed
Discovery service	Can be based on W3C WoT Discovery standard
Execution service	Standard web service needed
Workload packaging	Can be Javascript, WASM, or container images.
Offload rules	Can be implementation specific
Privacy considerations	Legal and adoption requirements

Summary

- Edge workers support widespread access to edge computing resources
  - Provides standardized offload mechanism for web apps
  - Access to advanced hardware features
  - Avoids backhaul and ISP bottlenecks, using high performance 5G or WiFi
- Simple extension of existing browser programming models
  - Most of the standards work needed is in infrastructure
- Complementary to cloud-based edge computing models
  - Still needed to manage compute utilities and the services they depend on
  - Cloud-based models also good for centrally managed systems

Discussion and Extensions

- Migration
  - Moving *running* edge worker instance - VM/container migration
- Persistent background execution/event driven execution
  - IoT orchestration use case – perhaps extend *Service* Workers/PWAs
- Fixed function workers
  - e.g. AI inferencing service
  - Can use same discovery mechanisms
  - Still need performance/network metadata
- Recursive use

Questions?