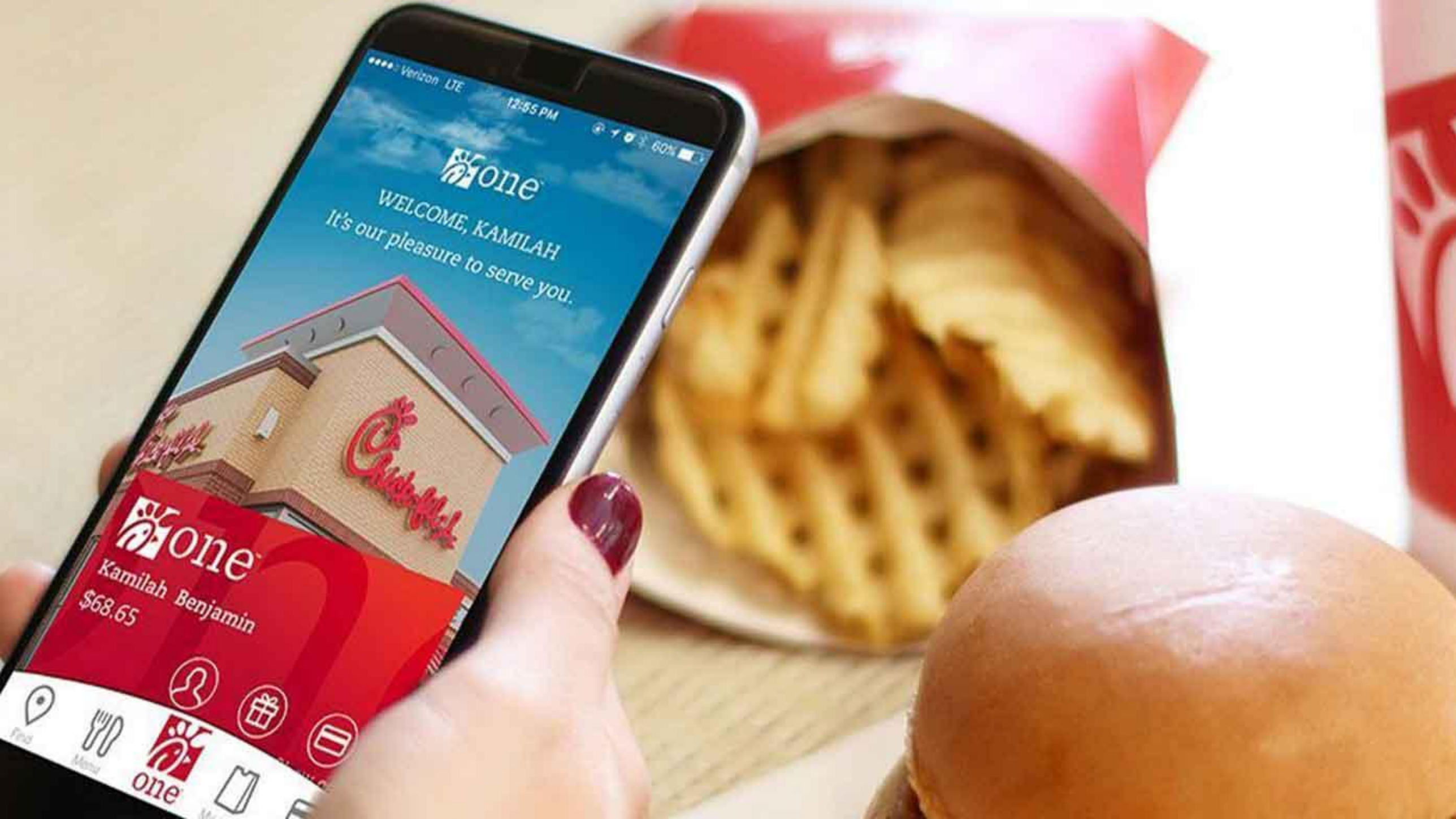

“Can’t we just use the cloud?”

# No.


Chick-fil-A
Your skills. Our team. A perfect match.
Drive-Thru
Lane One
Lane Two
8DZU918
8AFP345

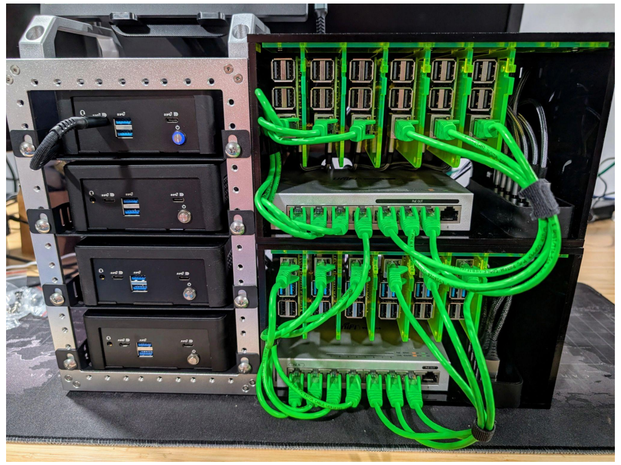

Verizon LTE
12:55 PM
60%
one
WELCOME, KAMILAH
It's our pleasure to serve you.
one
Kamilah Benjamin
$68.65
one


LETTUS
SUGJEST
CHIKIN

# What is edge computing?

# What do we mean by edge?


Minirack Project



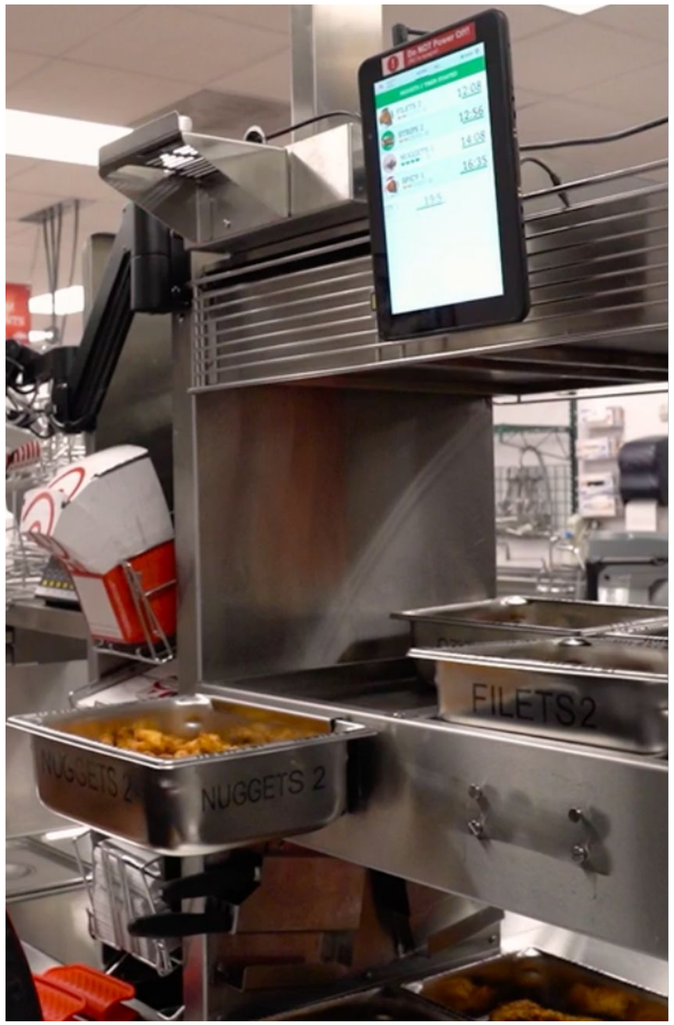
Lior Kamrat's Project

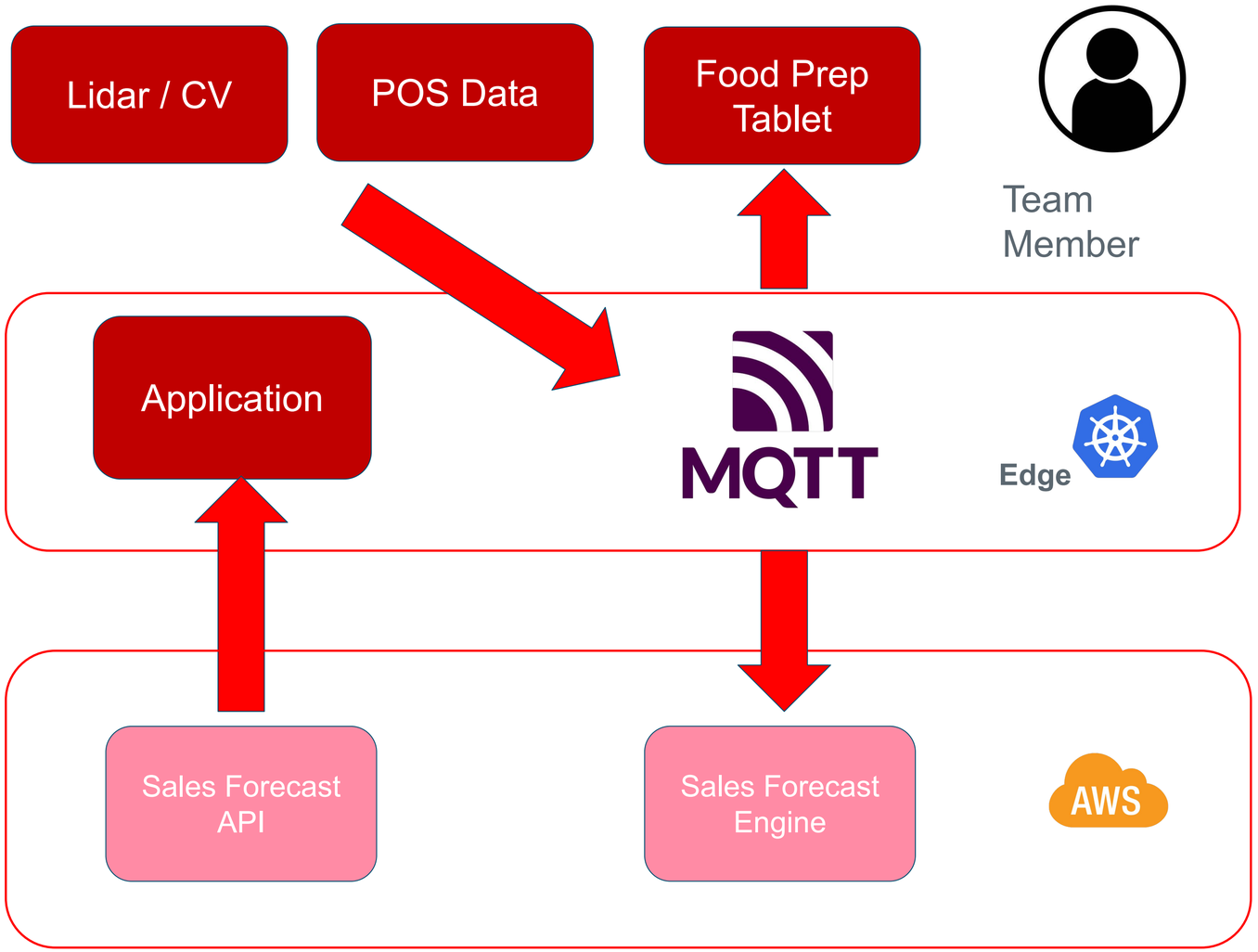
Chick-fil-A NUC Cluster

# Why invest in the edge?

# **Capability:** The desire to do things you can't currently do



# Cost

| | | |
|---|---|---|
| **WAN Bandwidth +** | | **Local Network +** |
| **Cloud Storage +** | **vs.** | **Local Storage +** |
| **Cloud Compute** | | **Local Compute** |

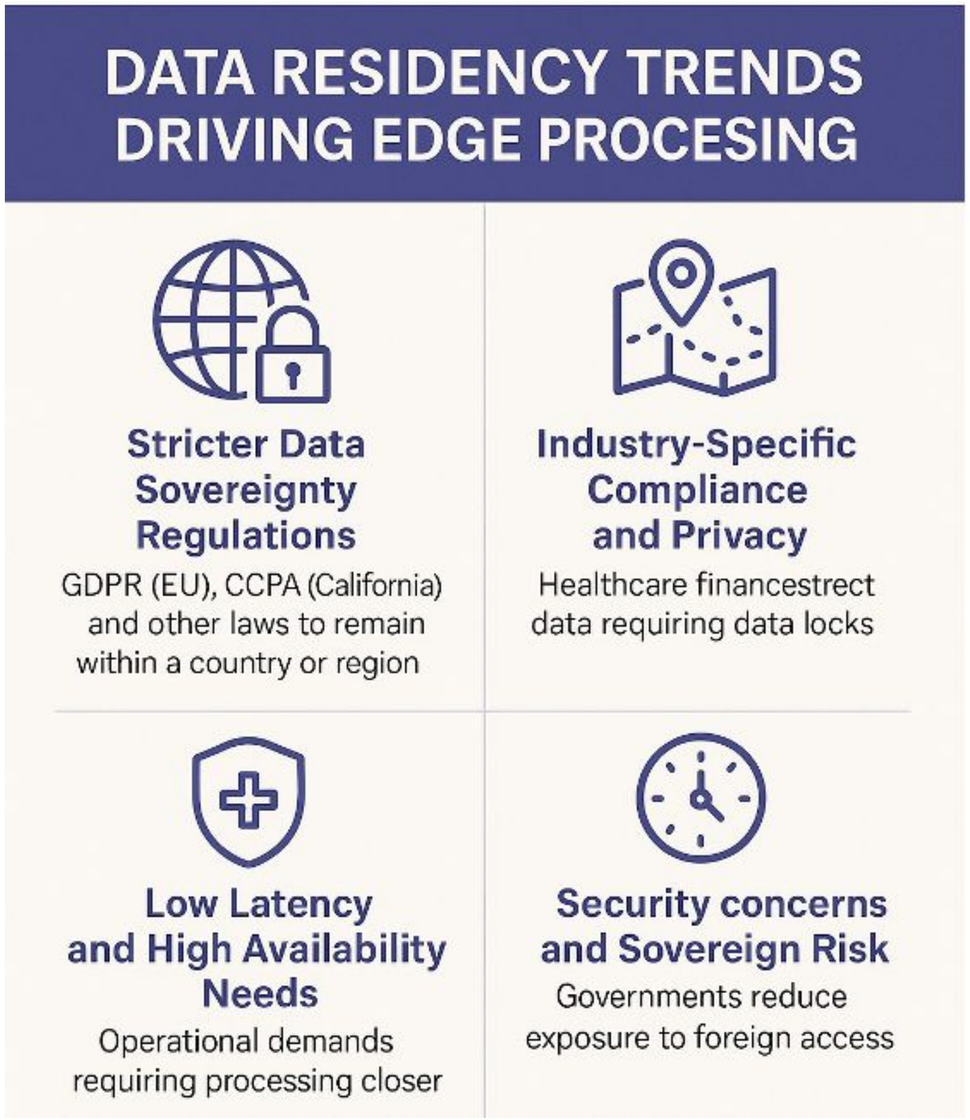
 

# Compliance

Data is an **asset**.

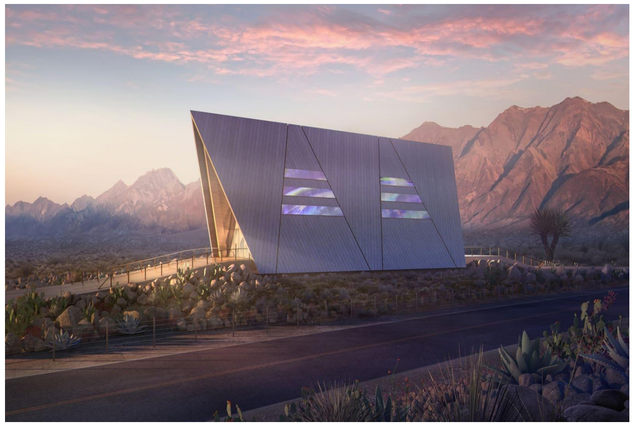
Data is a **liability**.

The industry is trending towards more requirements that certain types of data remain in **specific localities**.



# Convergence: What will it mean for your business if…?



+

+

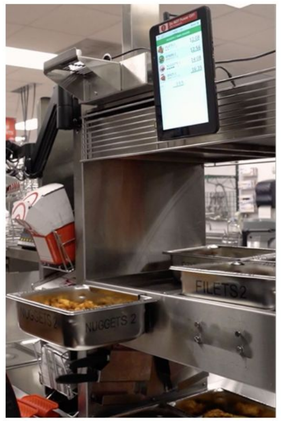
**Sufficient Electrons**

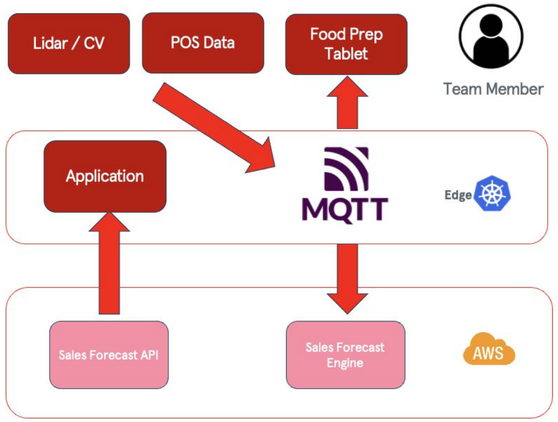
**Superintelligence Breakthrough**

**Robotics**

# Capability + Cost + Compliance + Convergence

**Capability:** The desire to do things you can't currently do






## Compliance

Data is an **asset**.
Data is a **liability**.

The industry is trending towards more requirements that certain types of data remain in **specific localities**.


## Cost

| | | |
|---|---|---|
| **WAN Bandwidth +** | | **Local Network +** |
| **Cloud Storage +** | **vs.** | **Local Storage +** |
| **Cloud Compute** | | **Local Compute** |

## Convergence: What will it mean for your business if…?

**Sufficient Electrons**

+

**Superintelligence Breakthrough**

+

**Robotics**

# What does it take?

## (to get to the edge?)

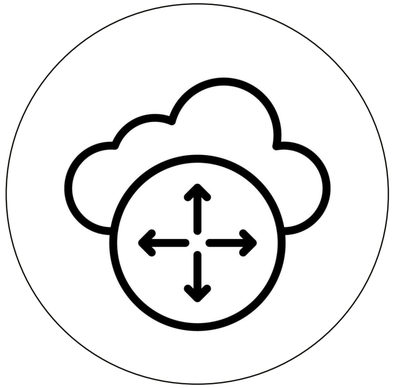
 

# Constraints at the Edge

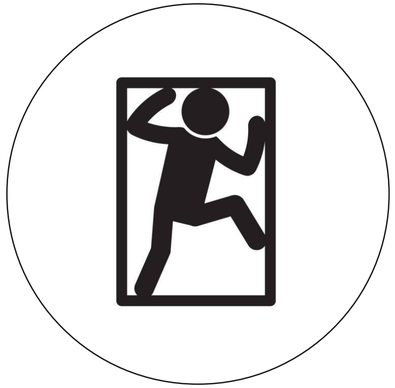
The reality at the edge is one of constraint. **Constraints** breed **creativity**.

**Network Connectivity**

**Bandwidth**



**Resource Constraints**



**Physical Space Constraints**

**Lack of On-site Tech Support**

# Challenge #1: You own the entire stack

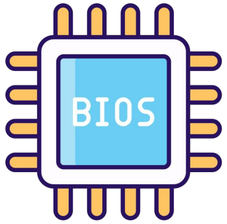
At the edge, you are responsible for solving for all the challenges of infrastructure.

WAN / LAN Network

SW/HW Supply Chain

Hardware

BIOS + Firmware




Secure On-boarding


Operating System

Patching

Cluster Management


Replicated Storage


App Deployment

Identity / PKI Management

Observability

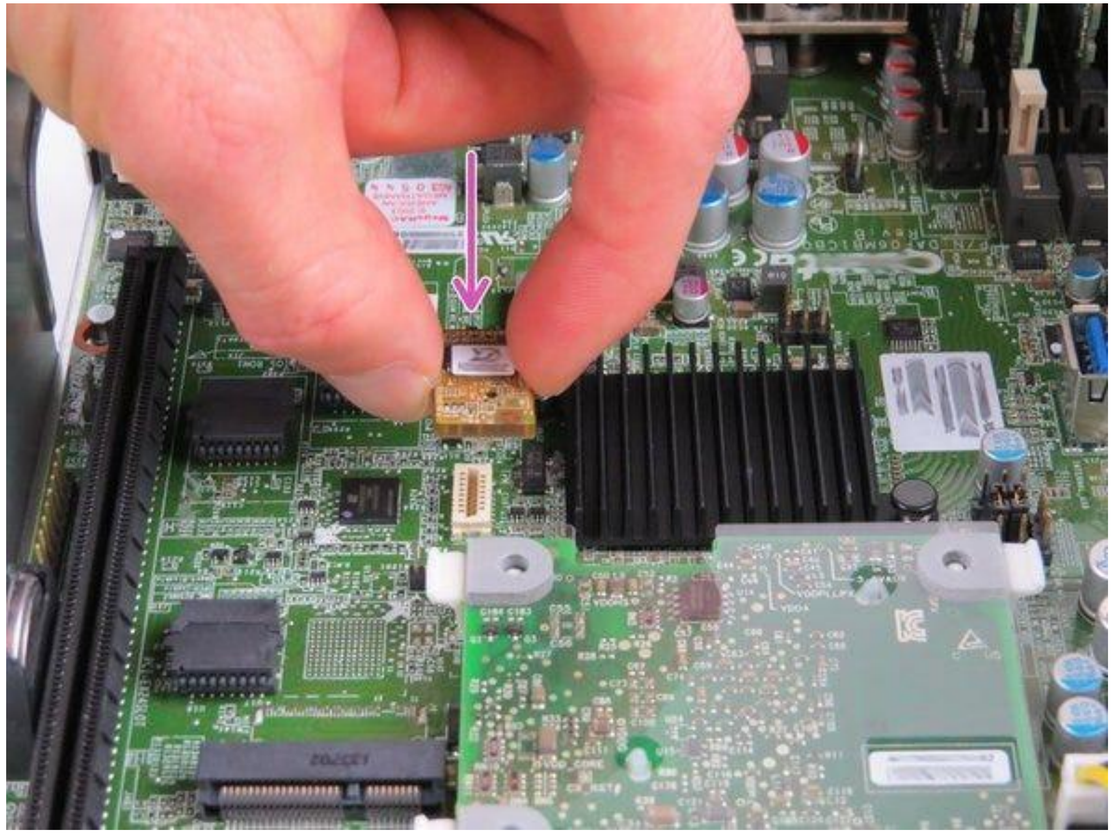
Shared Services

Support

# Challenge #2: Secure On-boarding

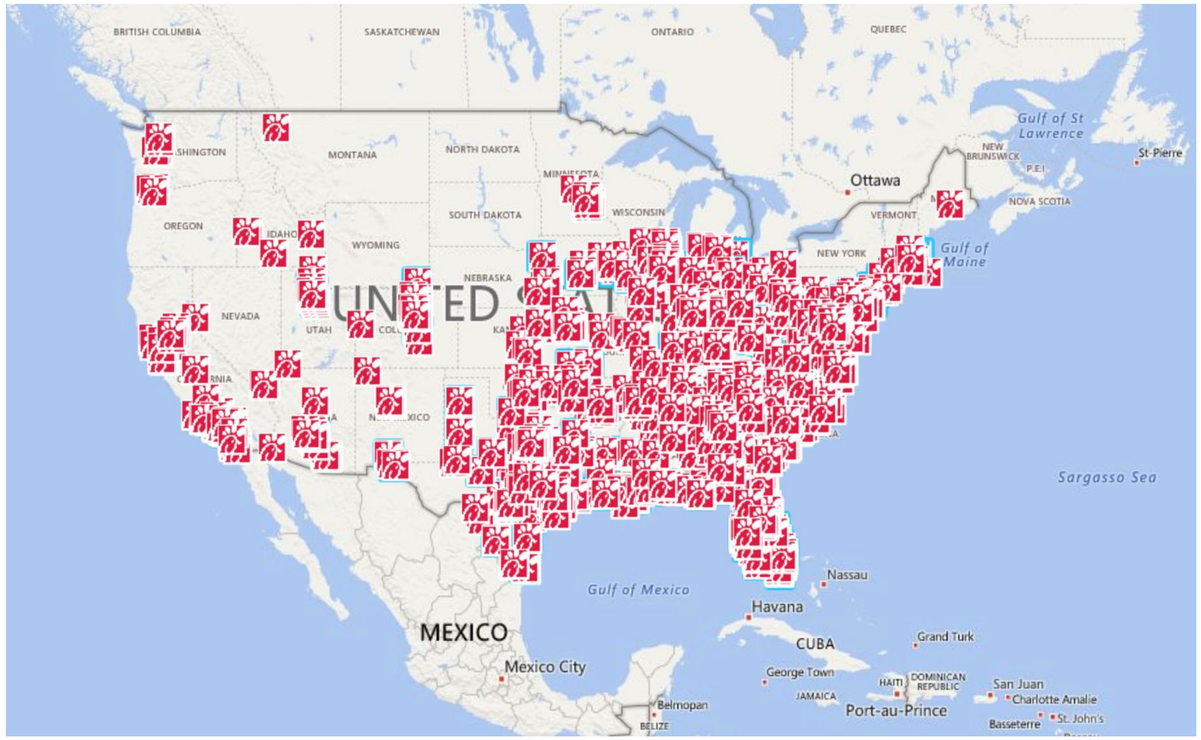
Secure on-boarding of devices (with zero-touch provisioning) is critical to secure operations.



- **Hardware Supply Chain**: Can you trust the suppliers delivering your hardware?
- **Hardware Root of Trust:** Establish trust through TPM and endorsement keys.
- **Measured Boot**: Validate no tampering is taking place.
- **Attestation**: Assert a devices identity to get a certificate.
- **Authenticate and Authorize to Services**: leverage the certificate to swap for other credentials or to authenticate to services.

# Challenge #3: Scaling with Architectural Complexity

Complexity at the edge is extremely hard to manage due to tens-of-thousands of footprints




- **Maximize Homogeneity**: Hardware, OS, network topology, and more.
- **Minimize unique configuration:** Establish trust through TPM and endorsement keys.
- **Automate, Automate!**: Automate as much as you can to remove anything you are certain you will have to do at scale (provisioning, deployments).
- **Bake in support features**: Remote wipes, scale logs up/down
- **Fear the brick**: Take special care around anything that might “brick” devices in the field at scale. Not good.

# Observations + Recommendations

# 1. You can succeed at the Edge with Open Source

Even if you aren't a "tech company". And it is the most **secure** way to do so.











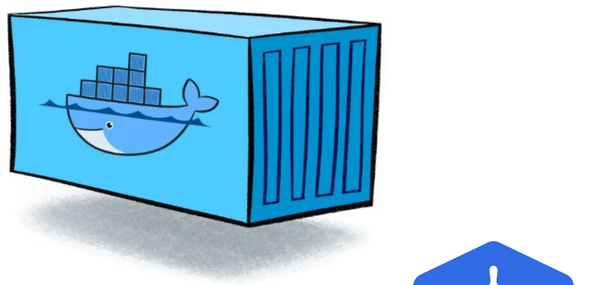


**And so many more awesome projects…**

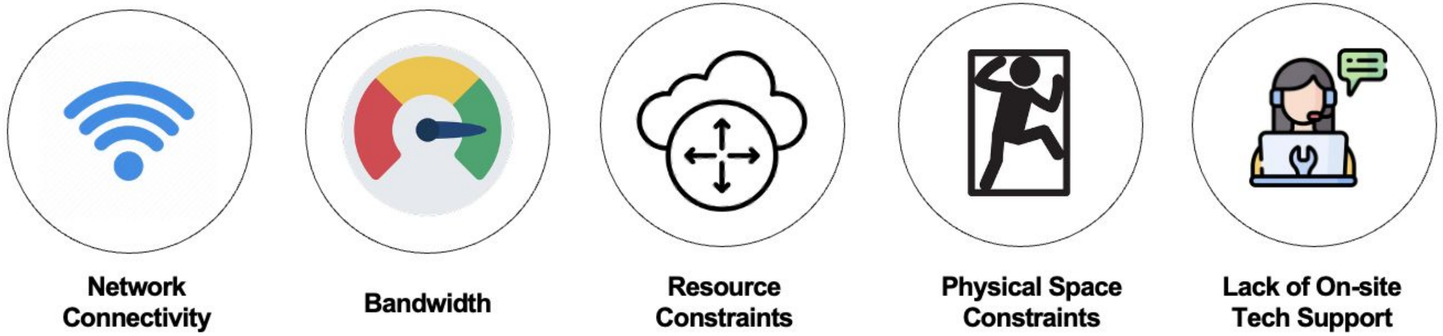
 

# 2. Be as cloud native as possible while managing differences

Maximize the similarities for developer friendliness, don’t underestimate the differences.



CLOUD NATIVE
COMPUTING FOUNDATION



# 3. Not all infrastructure is “undifferentiated”

But it is most certainly “heavy lifting”.

## AWS Design Principles

- **Stop spending money on undifferentiated heavy lifting**: AWS does the heavy lifting of data center operations like racking, stacking, and powering servers. It also removes the operational burden of managing operating systems and applications with managed services. This permits you to focus on your customers and business projects rather than on IT infrastructure.

*Infrastructure is **differentiating** when it enables your business to do something you can’t do otherwise*

# 4. Radically minimize complexity

If you don’t, it will bury you with operational burden.

A few examples of tradeoffs we made at Chick-fil-A to get simplicity…

- Homogeneity of everything (hardware, OS image, config)
- Minimal persistence guarantees / ephemeral data
- Minimal Kubernetes usage
- Ubuntu LTS to ensure security patches and minimize need for full in-field OS upgrade
- Simple GitOps deployment pipelines

# 5. Start secure, stay secure

Or your infrastructure won’t be yours for long.


Hardware Root of Trust


Secure or Measured Boot

Remote Attestation + Zero Trust

## Stay Secure

- Network Isolation + Outbound traffic only
- mTLS / SSL everywhere
- OS and App Patching
- Threat monitoring
- Software Supply Chain (SBOM, sigstore.)
- Secrets Management
- Physical Security

# Time to abandon the cloud?

# Should you abandon the cloud?


- Aggregated Analytics
- ML model training
- Customer-facing applications
- Anything that doesn’t have to be at the edge
- Managing a fleet of edge devices

- Local, real-time decisions
- Operations during network disruptions
- Low-latency applications
- High data volume pre-processing (like IOT)

# Thanks!