UVA ENGINEERING
LINK LAB

UNIVERSITY of VIRGINIA

NexusEdge: Leveraging IoT Gateways for a Decentralized Edge Computing Platform

Nabeel Nasir, Victor Ariel Leal Sobral, Li-Pang Huang, Bradford Campbell
University of Virginia

7th ACM/IEEE Symposium on Edge Computing (SEC), 2022

A look back at last year's SEC Keynote

The Computing Continuum

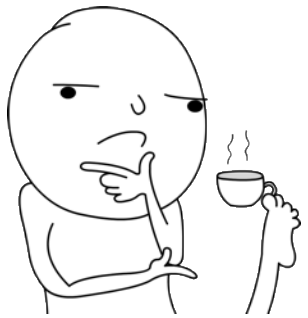
IoT/Edge → **Fog** → **HPC/Cloud/Instrument**

Size	Nano	Micro	Milli	Server	Fog	Campus	Facility
Example	Adafruit Trinket	Particle.io Boron	Array of Things	Linux Box	Co-located Blades	1000-node cluster	Datacenter
Memory	0.5K	256K	8GB	32GB	256G	32TB	16PB
Network	BLE	WiFi/LTE	WiFi/LTE	1 GigE	10GigE	40GigE	N*100GigE
Cost	$5	$30	$600	$3K	$50K	$2M	$1000M

Count = 10^9
Size = 10^1

Count = 10^1
Size = 10^9

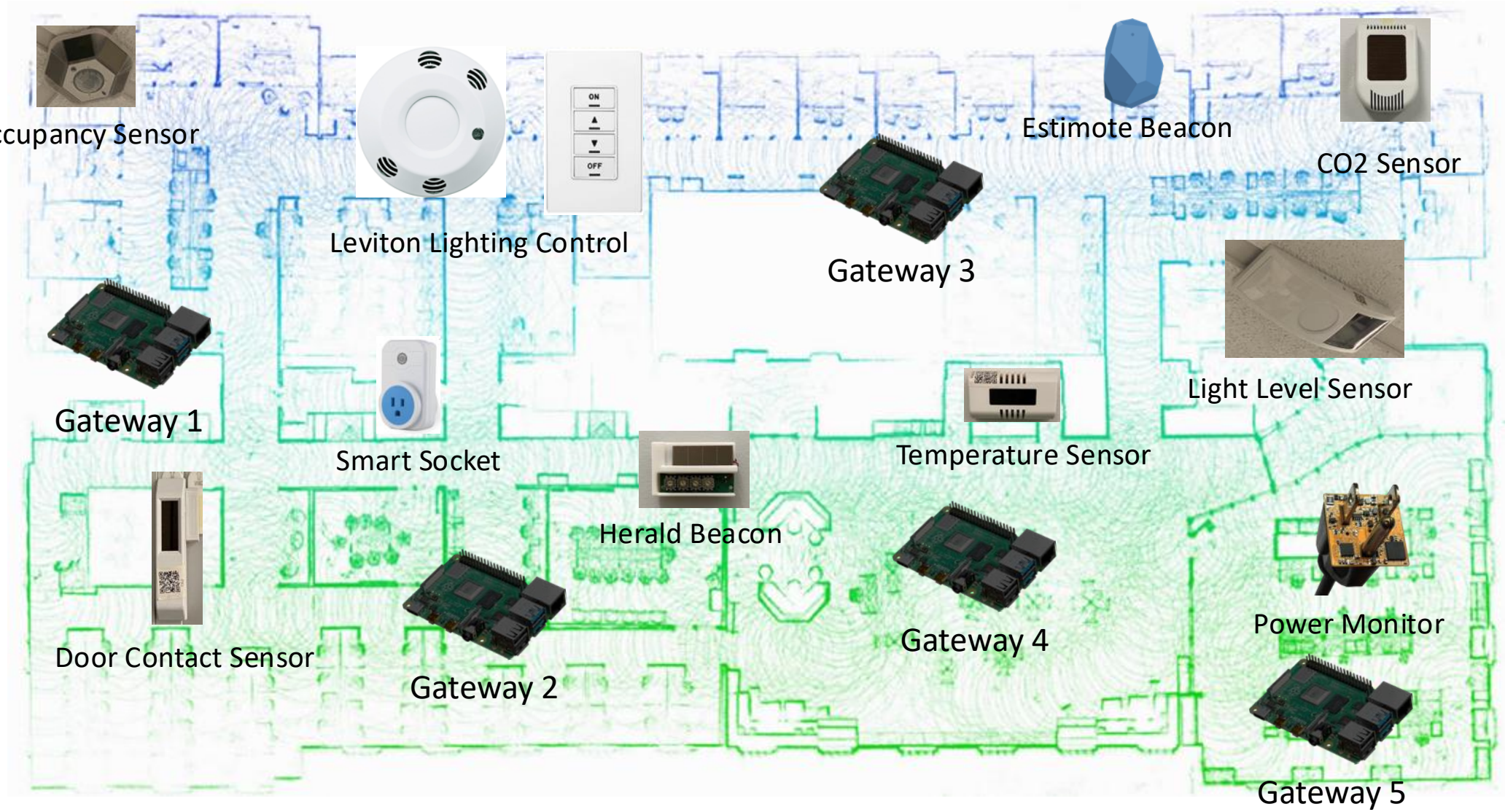
Raspberry Pi
4GB
WiFi
$40



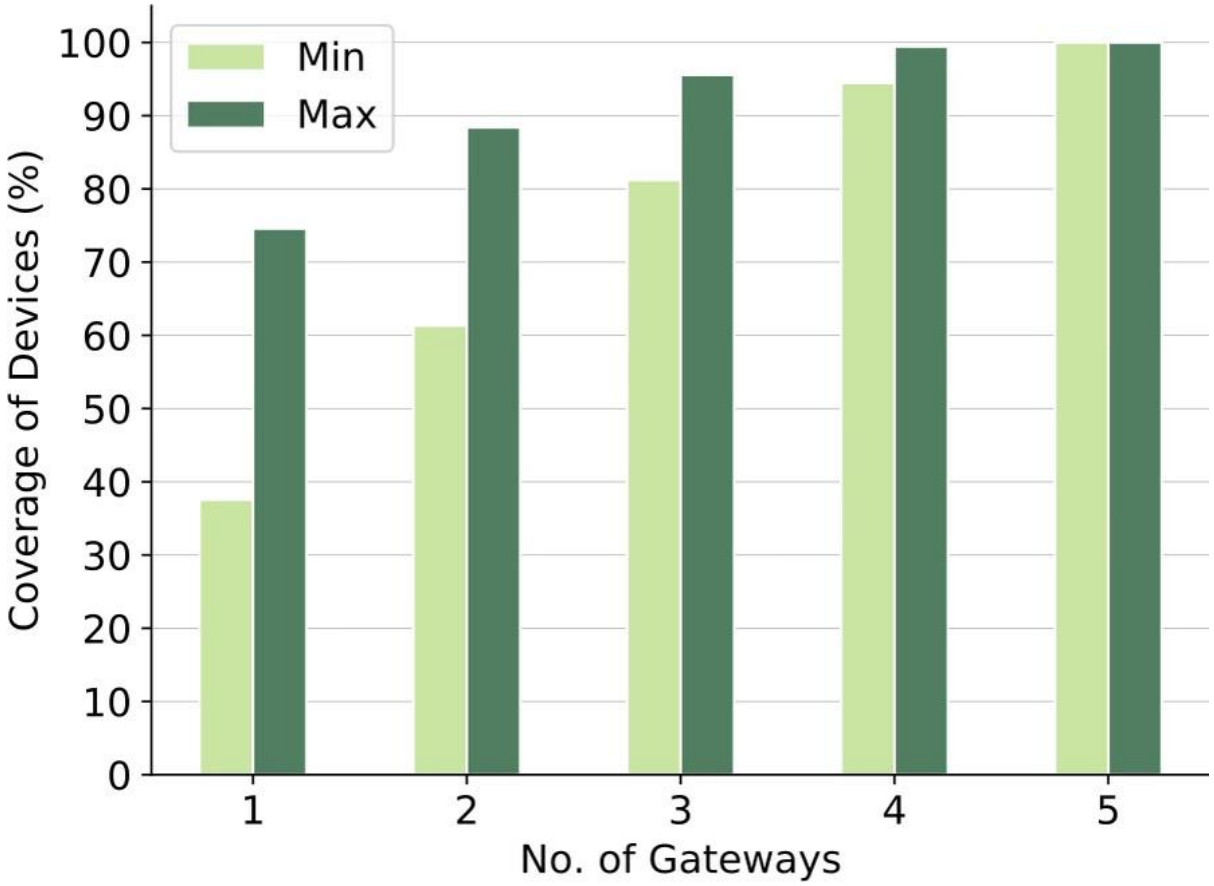
Can we utilize Gateways for Edge Computing?

Pete Beckman et al., "Harnessing the Computing Continuum for Programming Our World." *Fog Computing*, 1st ed., Wiley, 2020

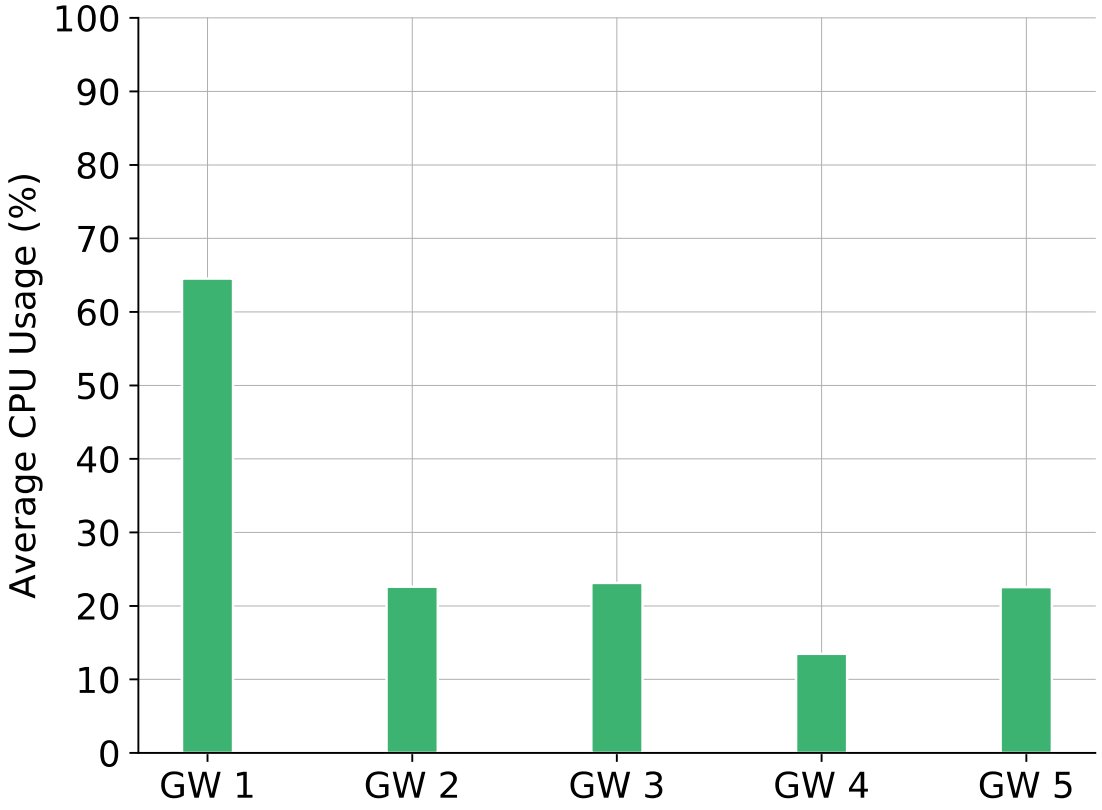
Collecting Data from ~200 IoT Devices in Our Test Bed



Gateways are Necessary but Underutilized

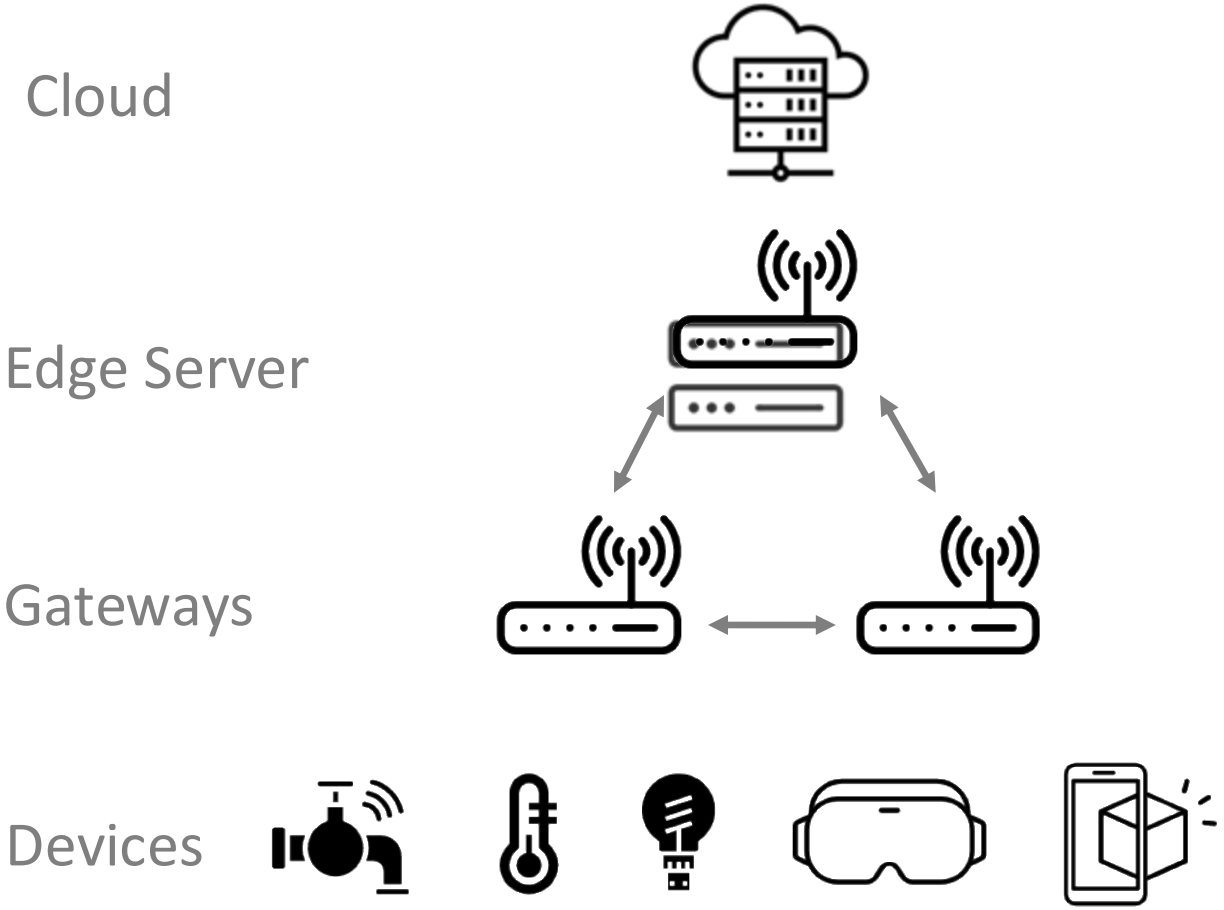


Gateways necessary to cover IoT devices



Have underutilized computational power

Maximize Utility of Gateways for EC before relying on Servers



Gateways offer Some Advantages over Server Machines

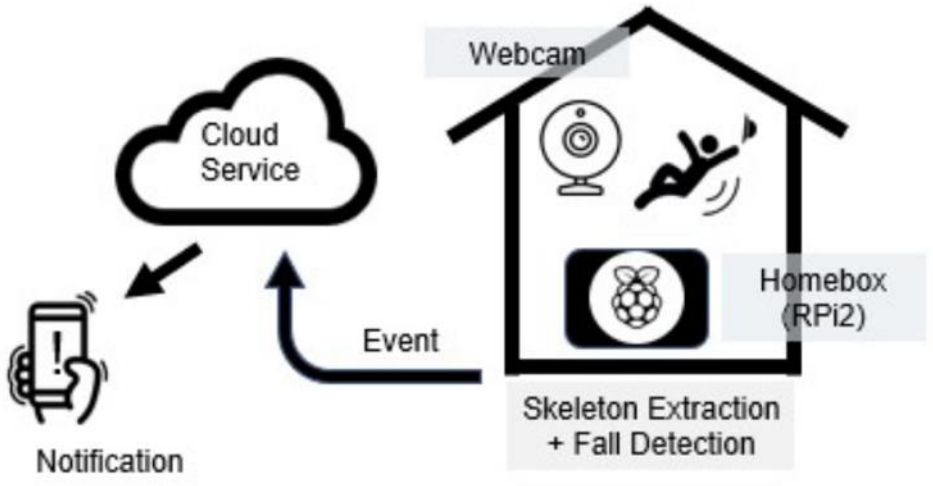
- Better deployment flexibility
- Wireless radios to communicate with IoT devices (one hop from devices)
- Substantially lower cost (servers are $1000+)
- No need to streaming all data to a centralized point
- Also, IoT gateways are becoming better performant

What kind of Edge Applications can we run on these gateways?

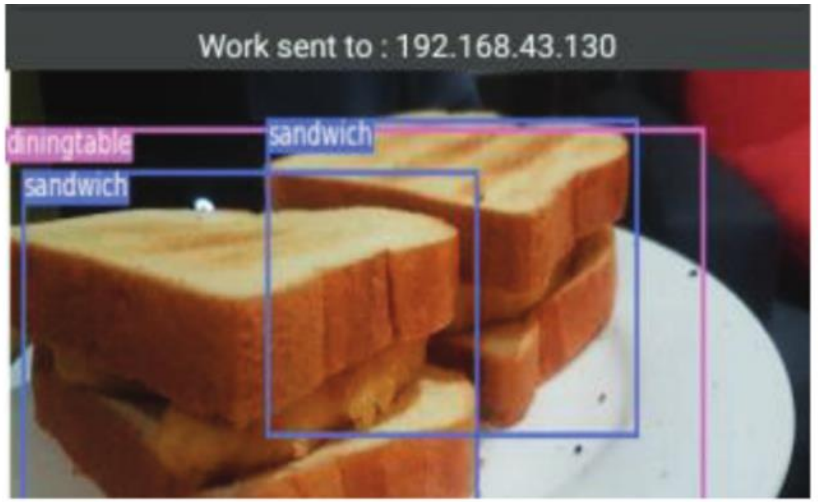
- Not suitable for:
  - Compute-intensive
  - High storage apps
- Examples: Live Video Analytics, Augmented Reality

- Suitable for:
  - Long running applications
  - Operates on streaming data
  - Low to Moderate storage
  - Low to Moderate compute

Machine Learning at the Edge



Elderly Fall Detection [1]

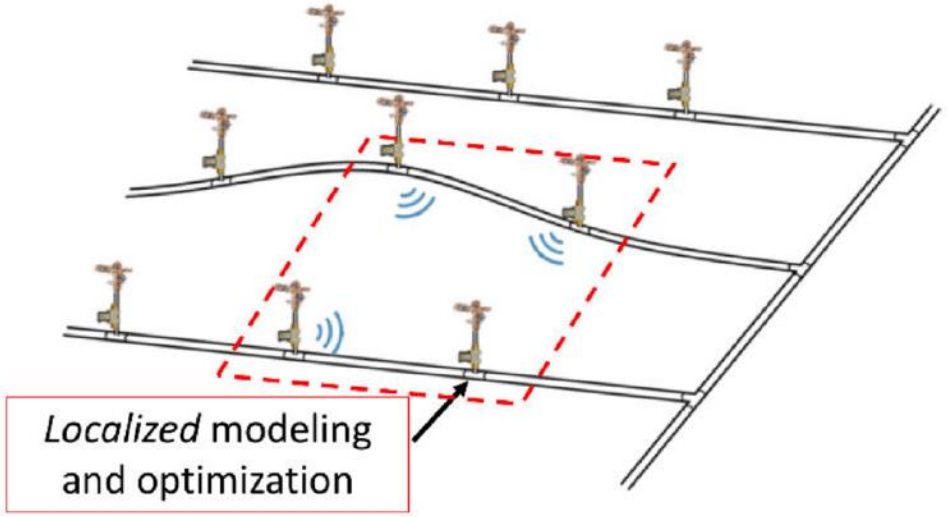


Object Detection [2]

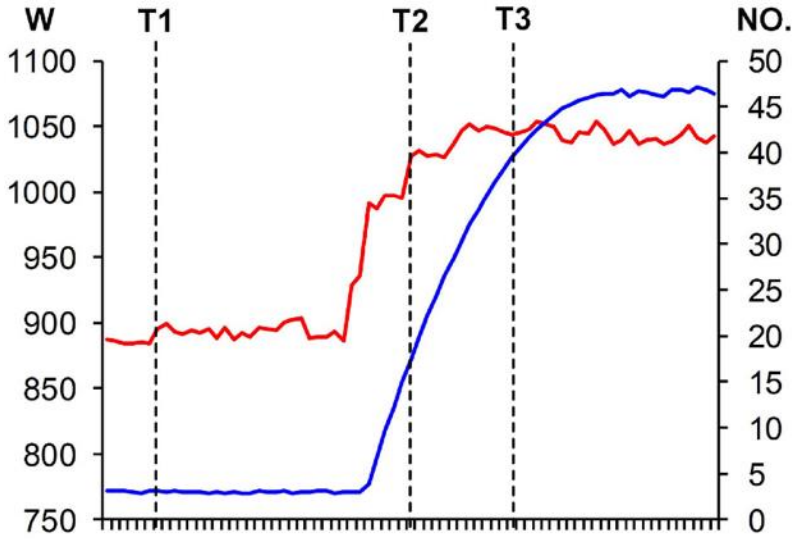
[1] Hsu et al. “Fallcare+: An iot surveillance system for fall detection”, IEEE International conference on applied system innovation (ICASI), 2017.

[2] S. Tuli et al. “Edgelens: Deep learning based object detection in integrated iot, fog and cloud computing environments”, IEEE ISCON, 2019.

Large Scale Sensing and Actuating



Distributed Soil Moisture Control [5]

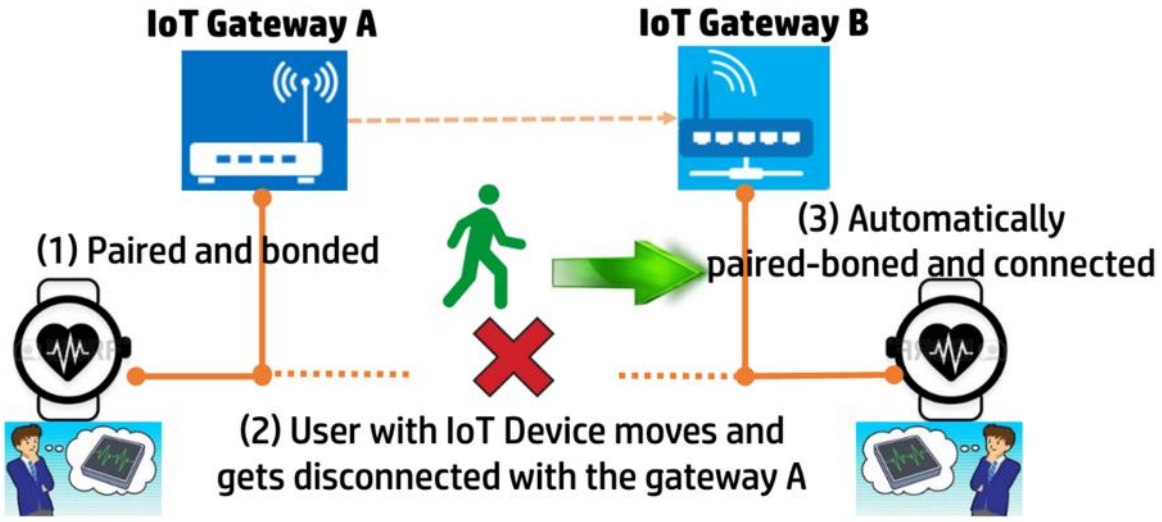


Occupancy based Building Energy Forecasting [6]

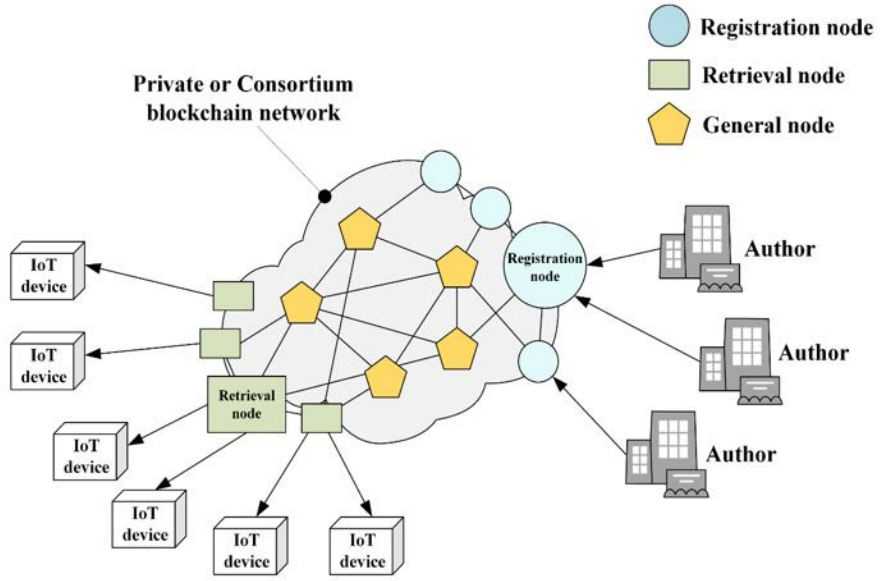
[5] D.A.Winkler et al. “Wisdom: watering intelligently at scale with distributed optimization and modeling,” ACM SenSys, 2019.

[6] Y. Tang et al. “Establishment of enhanced load modeling by correlating with occupancy information,” IEEE Transactions on Smart Grid, 2019.

Inherently Distributed Applications



BLE Pairing Handoff across IoT Gateways [3]



Distributed Firmware Update for IoT Devices [4]

[3] S. R. Hussain et al. “Seamblue: Seamless bluetooth low energy connection migration for unmodified iot devices,” ACM EWSN, 2017.

[4] S. Choi et al. “Blockchain-based distributed firmware update architecture for iot devices,” IEEE Access, 2020.

Building An Edge Platform on Gateways poses some Challenges

1. Scaling up with multiple gateways has configuration overhead
2. IoT devices use heterogeneous communication techniques
3. Application development on a distributed network is complex
4. Managing apps and devices without a central point is hard
5. Gateways are not as resilient as server machines

Middleware on Gateways to collaborate and execute applications!

Some related works exist, but have limitations

Theme	Paper	Limitations
Gateway Coordination	Ooi et al. [1]	- Cannot execute edge applications
	Clemente et al. [2]	- Doesn't handle device heterogeneity - Interfaces with devices & apps not defined
IoT Middleware	ThingsJS [3]	- Excludes low-power and harvesting devices - Overhead for obtaining device streams
	Hive Middleware [4]	- Not resilient to failures - Data locality not considered in scheduling

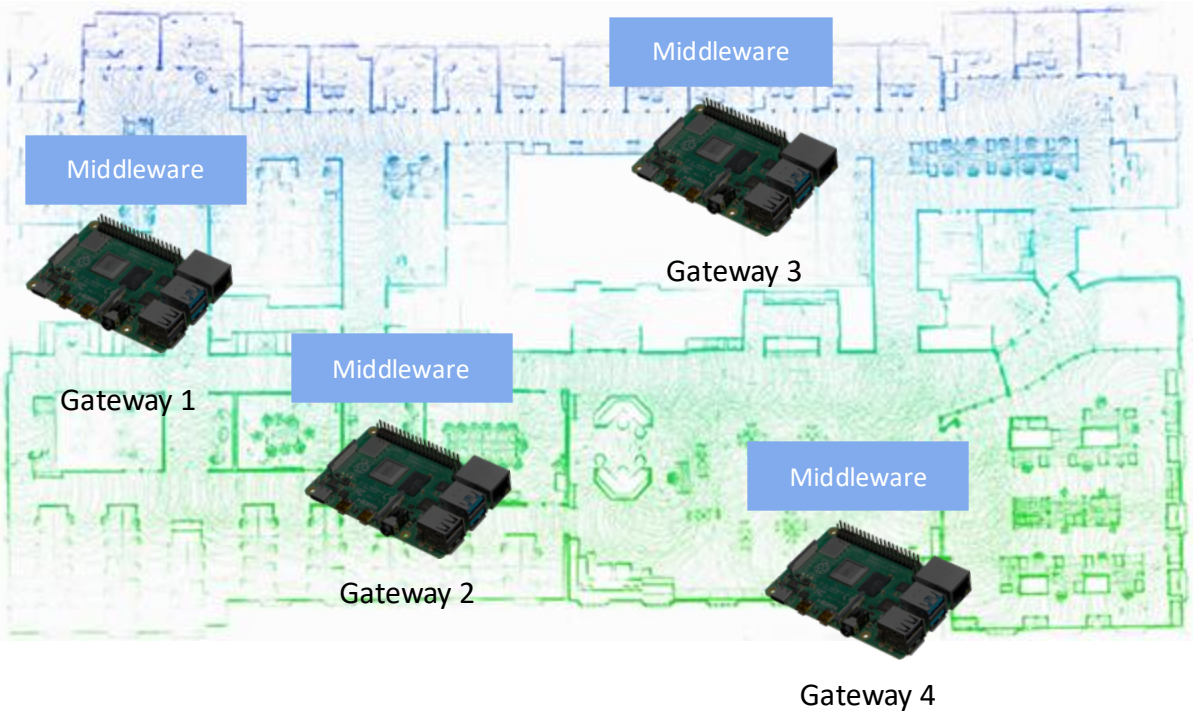
[1] Ooi et al. "A collaborative iot-gateway architecture for reliable and cost effective measurements," IEEE Instrumentation & Measurement, 2019.

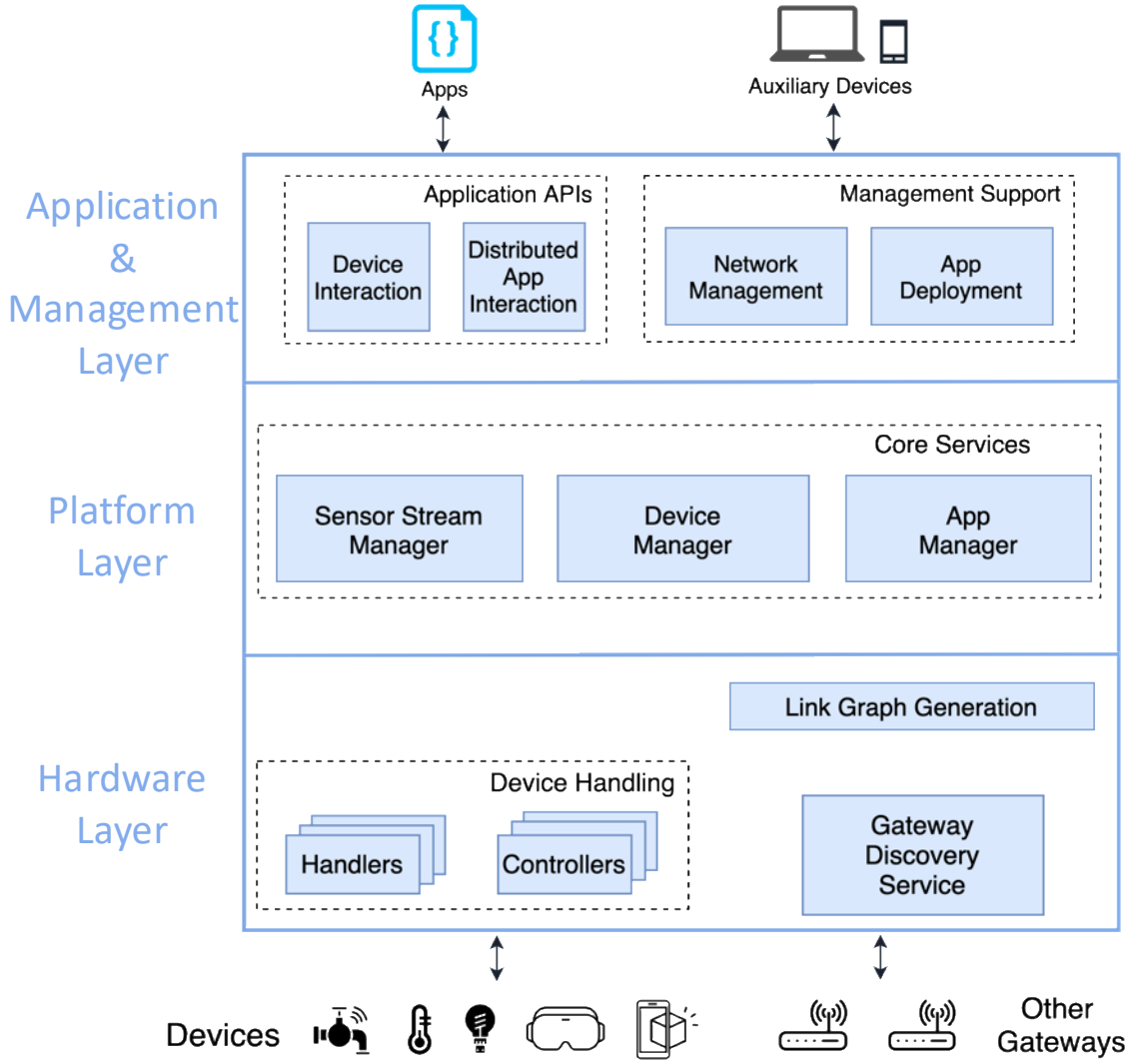
[2] Clemente et al. "Fog computing middleware for distributed cooperative data analytics", IEEE Fog World Congress (FWC), 2017.

[3] Gascon-Samson et al. "Thingsjs: Towards a flexible and self-adaptable middleware for dynamic and heterogeneous iot environments", 4th Workshop on Middleware and Applications for the Internet of Things, 2017.

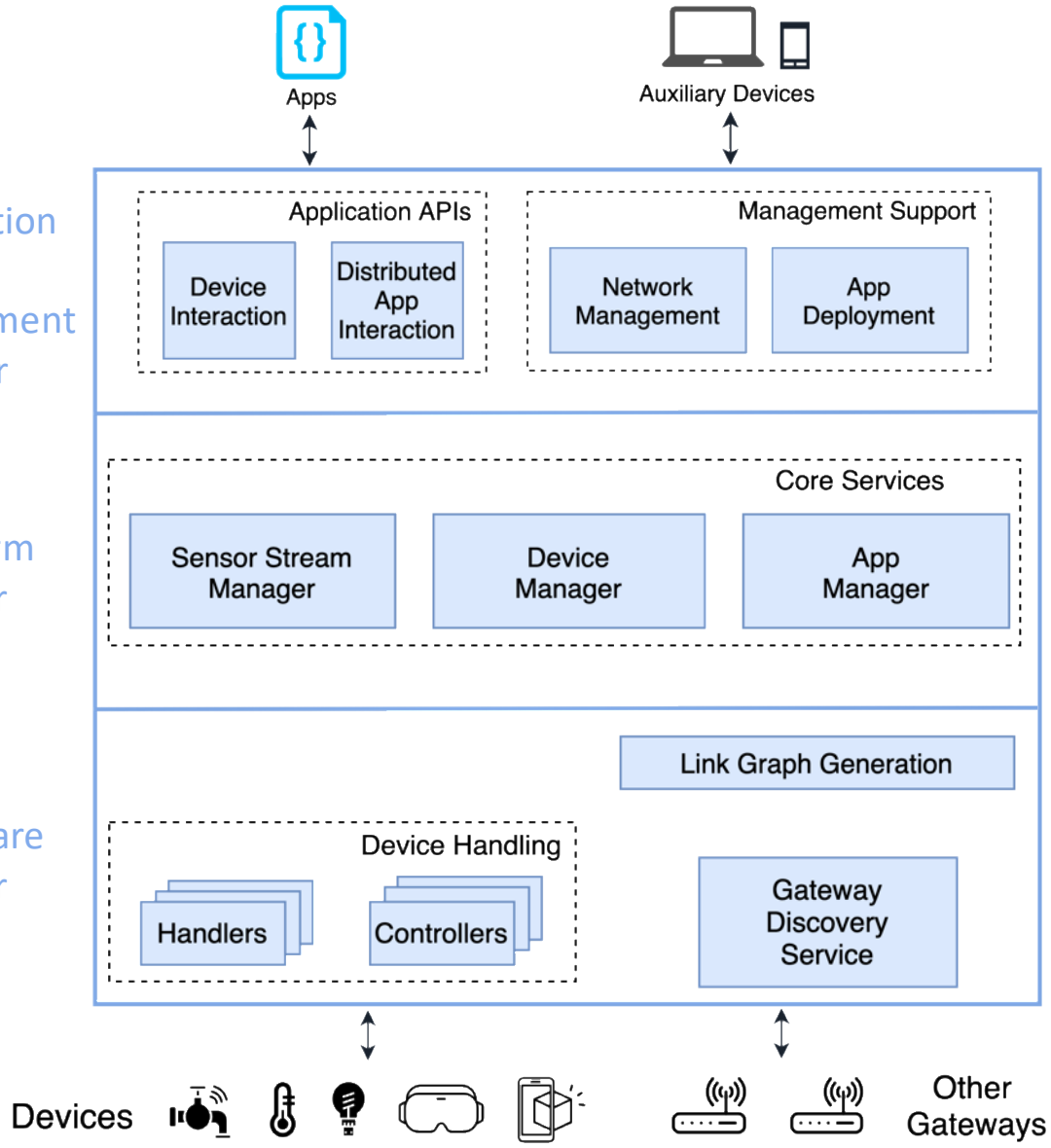
[4] Essameldin et al. "More than the sum of its things: Resource sharing across iots at the edge", IEEE/ACM Symposium on Edge Computing (SEC), 2020.

NexusEdge Middleware on Gateways to Handle Challenges





Middleware on Gateways to Handle Challenges



Gateway discovery for scalability

Supports IoT device handling

Encapsulates current network

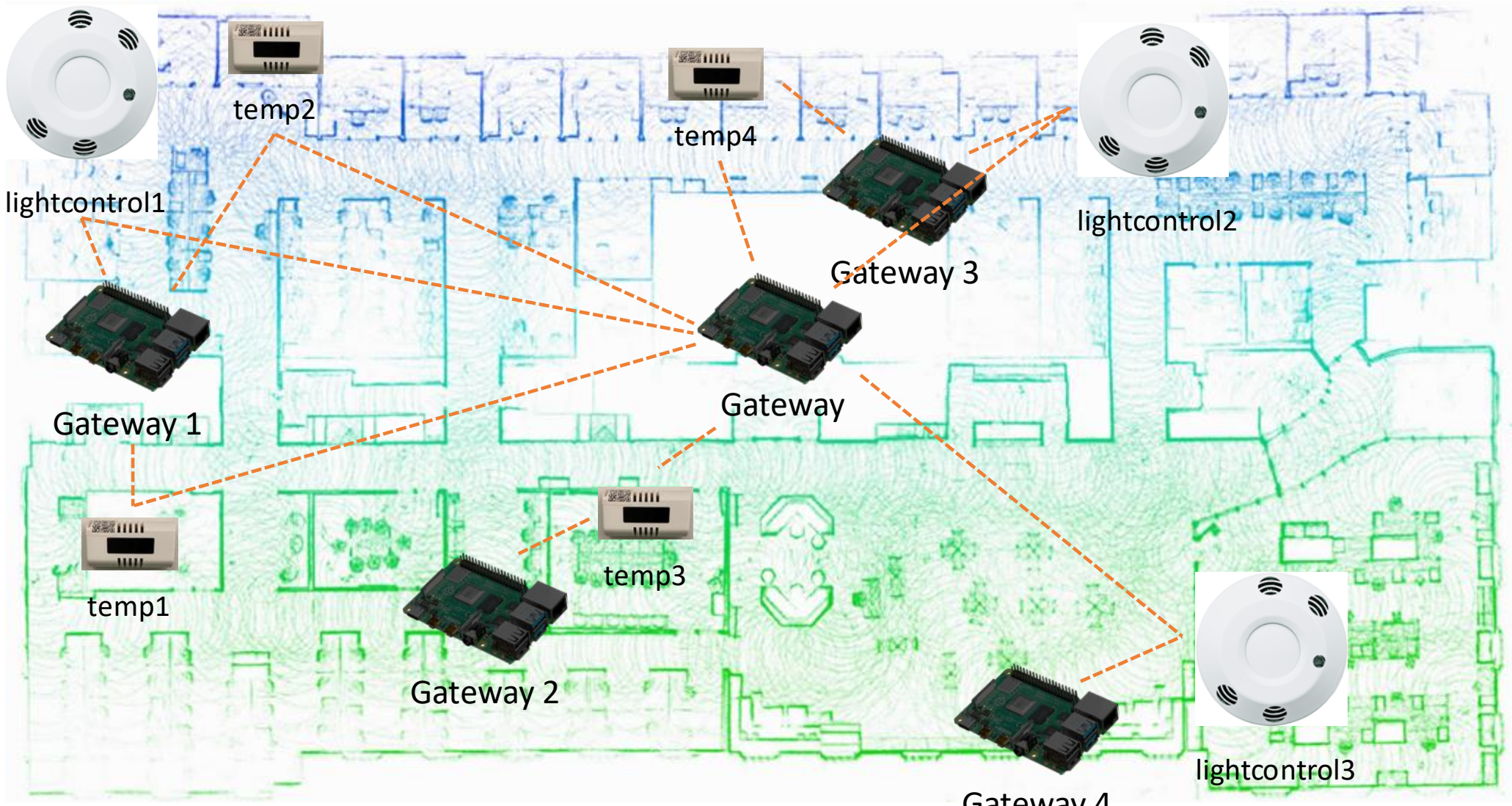
Using a Link Graph to Encapsulate Network Information

- IoT environments are dynamic: devices and gateways could join and leave

- For gateways to coordinate, each gateway needs to know:
  - What other gateways are present
  - What other devices are present
  - Which gateways can communicate with which devices

- Generating link graph: Ask neighbors, they ask their neighbors,…

Next Challenge: Give Developers an Abstracted Single Gateway View



APIs to Provide Simplicity in Device Interaction

Receive data streams from temp4

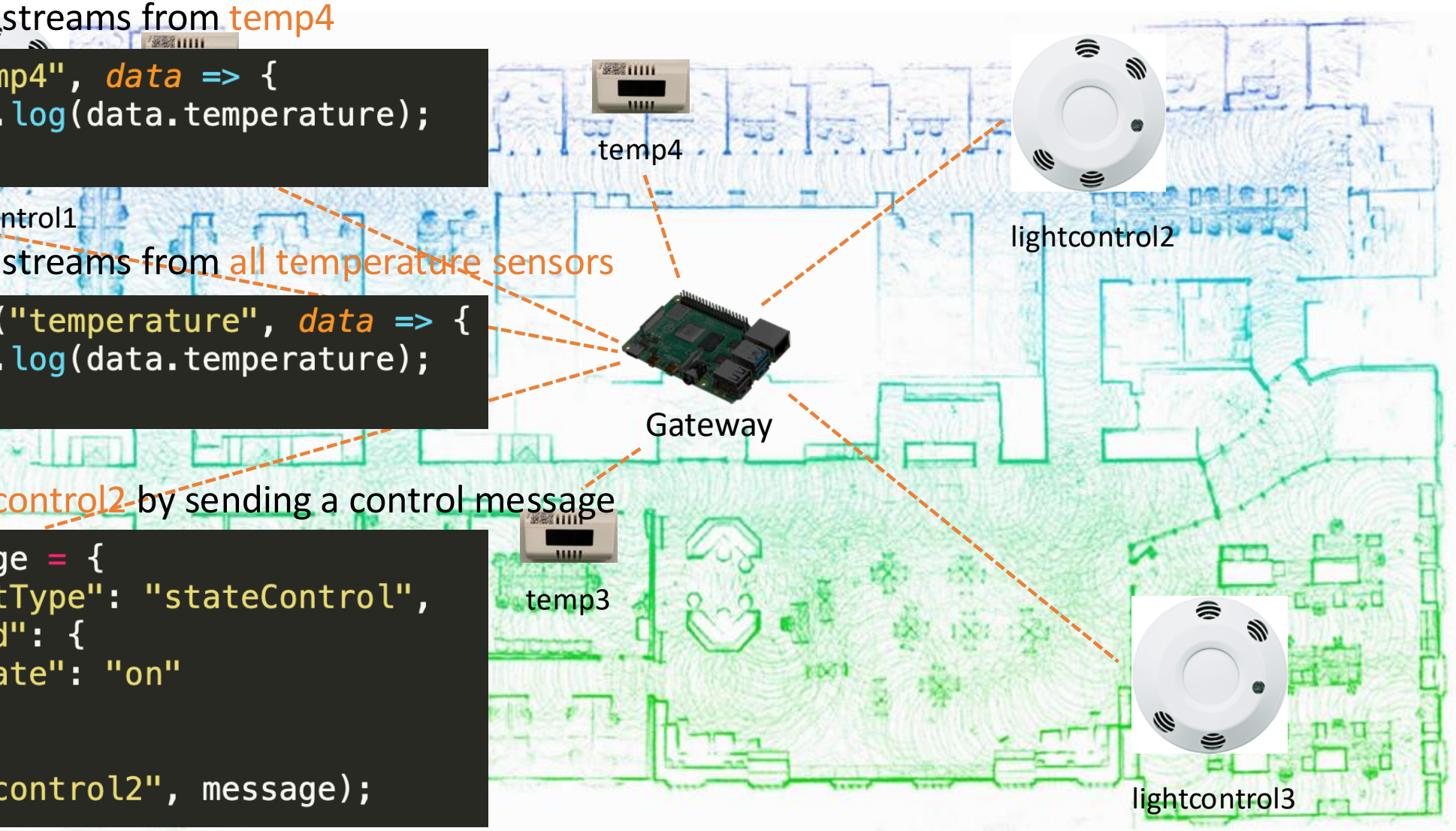
```
receive("temp4", data => {
    console.log(data.temperature);
});
```

Receive data streams from all temperature sensors

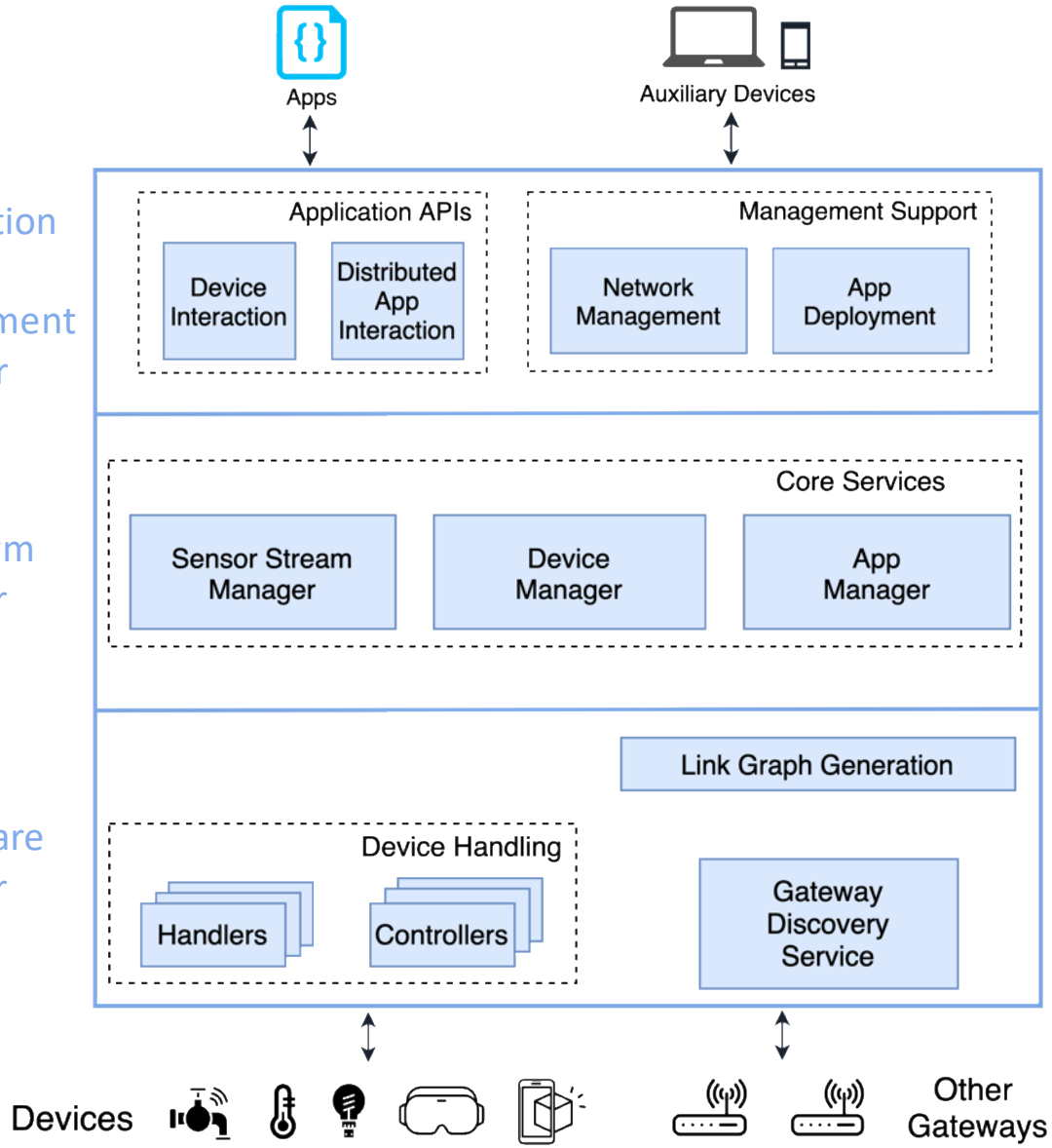
```
receiveType("temperature", data => {
    console.log(data.temperature);
});
```

Turn on lightcontrol2 by sending a control message

```
const message = {
    "requestType": "stateControl",
    "payload": {
        "state": "on"
    }
};
send("lightcontrol2", message);
```



Middleware on Gateways to Facilitate this Shift



Gateway discovery for scalability

Supports IoT device handling

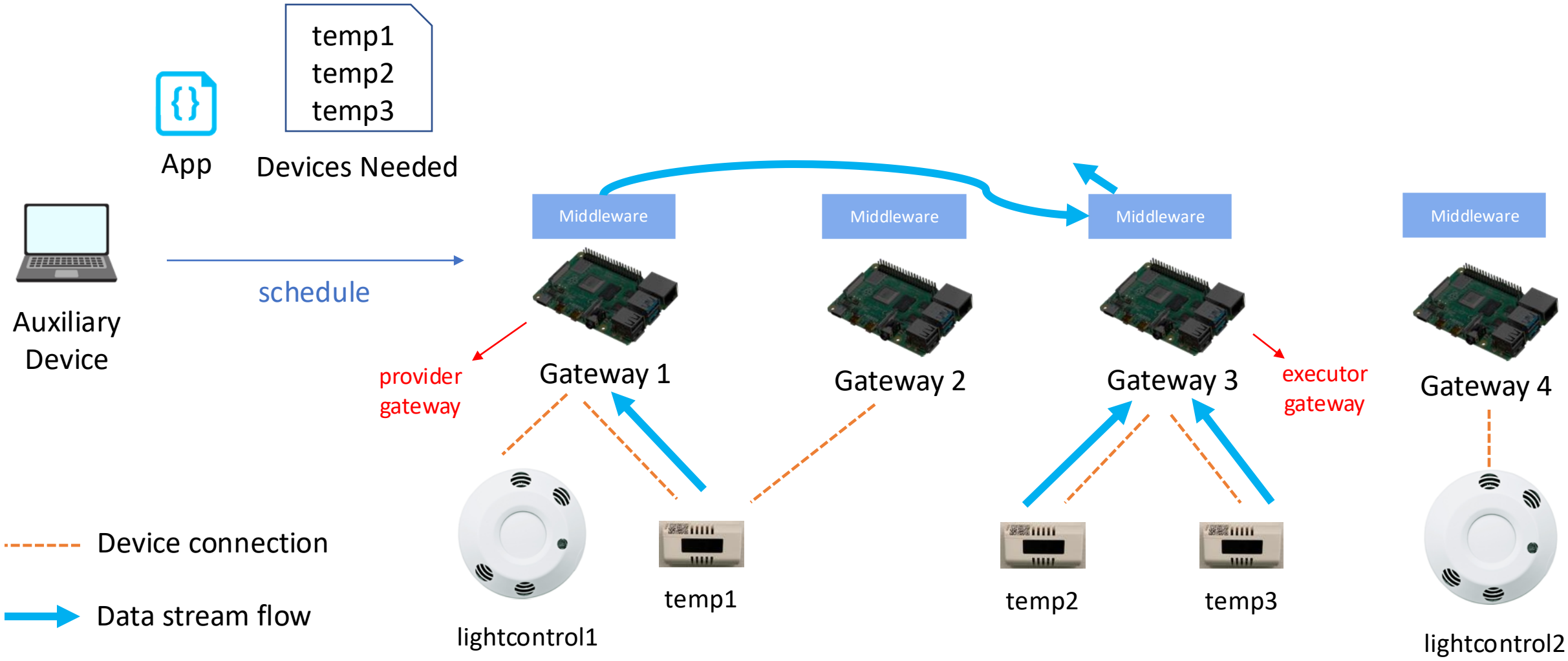
Encapsulates current network

Distributed services to handle devices, applications, and data streams

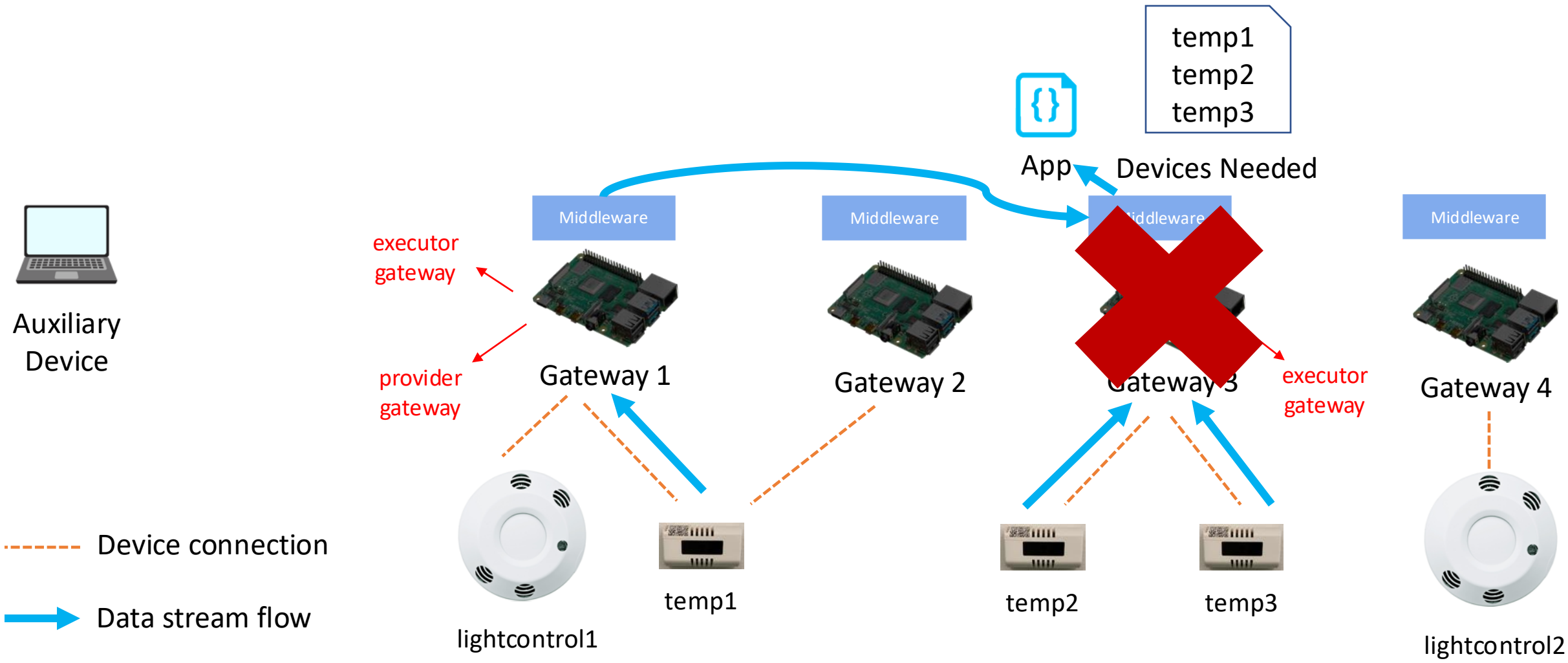
Device interaction API hides underlying network complexity from apps

Provides remote management

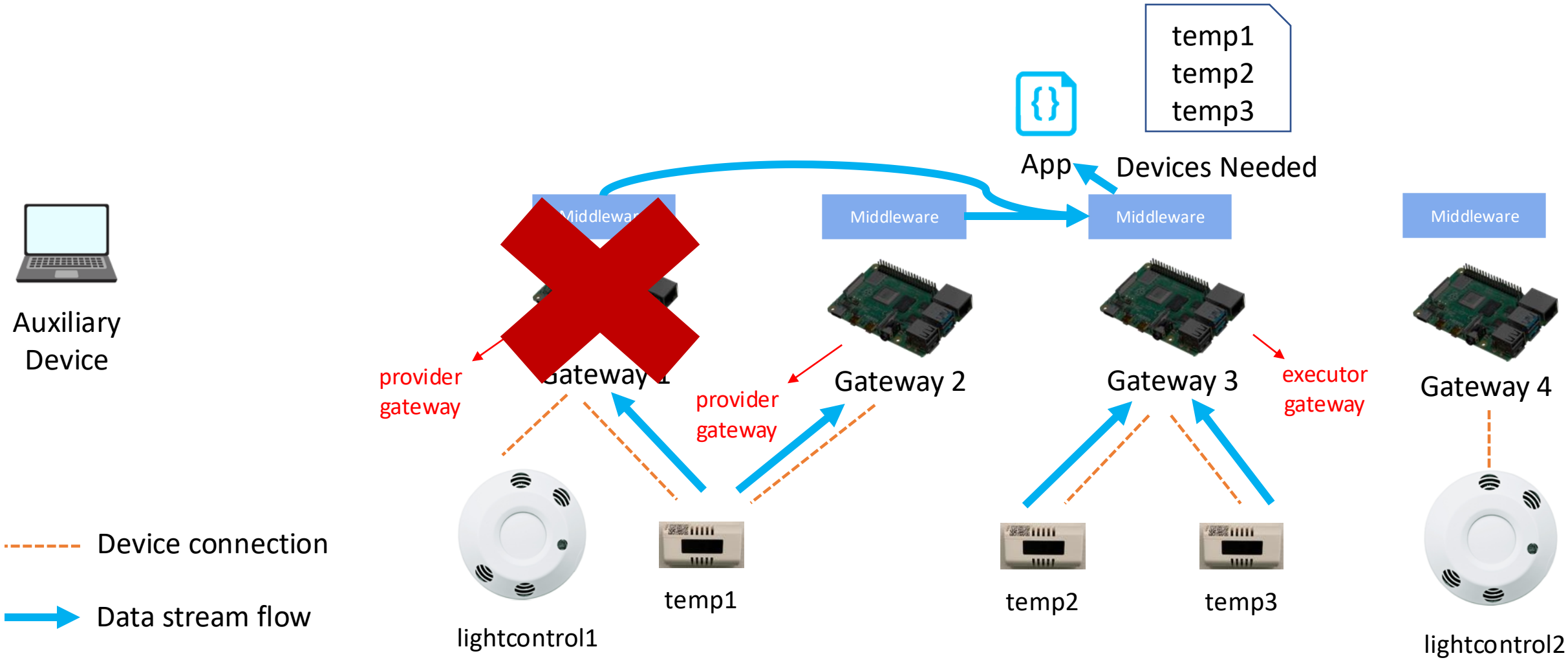
NexusEdge uses Device Locality to Schedule Applications



Resiliency: Migrates App if Executor Gateway Fails



Resiliency: Provide Alternative Streams (if possible) when Provider Fails



NexusEdge is open-source, containerized and Kubernetes-friendly

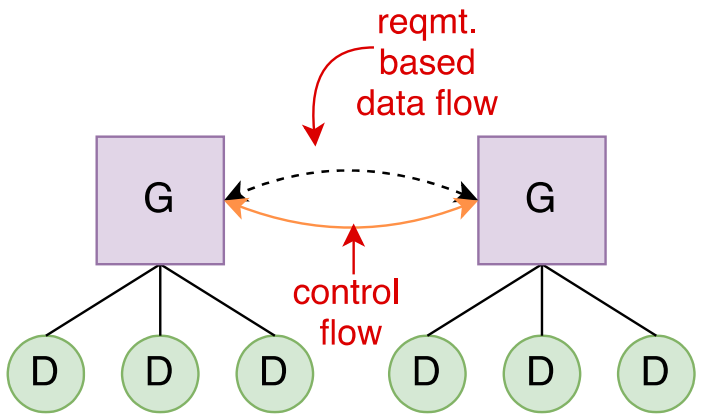
- Implementation is open source
  - github.com/uva-linklab/nexusedge

- Containerized with Docker
  - Handles gateway heterogeneity and improves deployability

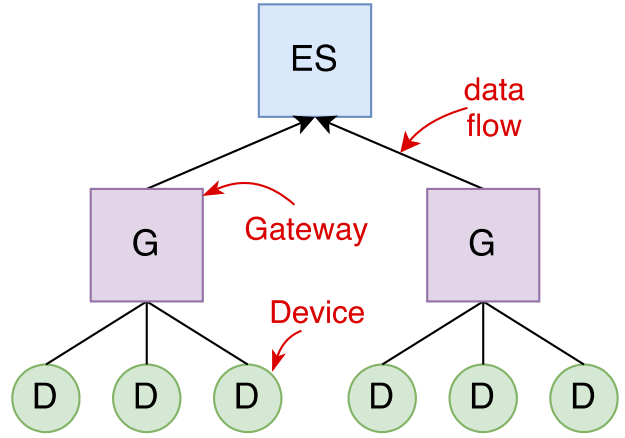
- Can be orchestrated using K3S (Lightweight Kubernetes)
  - DaemonSet to deploy middleware on all gateways

Scalability with Devices and Applications

- Comparison of decentralized NexusEdge vs a centralized server-based architecture



Decentralized NexusEdge Architecture



Centralized Edge Server Architecture

Scalability with No. of IoT Devices

- Setup
  - Edge Server: 4 Raspberry Pis as Gateways, 1 Raspberry Pi as Server
  - NexusEdge: 5 Raspberry Pis as Gateway, no Server

- Measure:
  - CPU usage
  - Memory usage
  - Network traffic

Comparison with Centralized Architecture

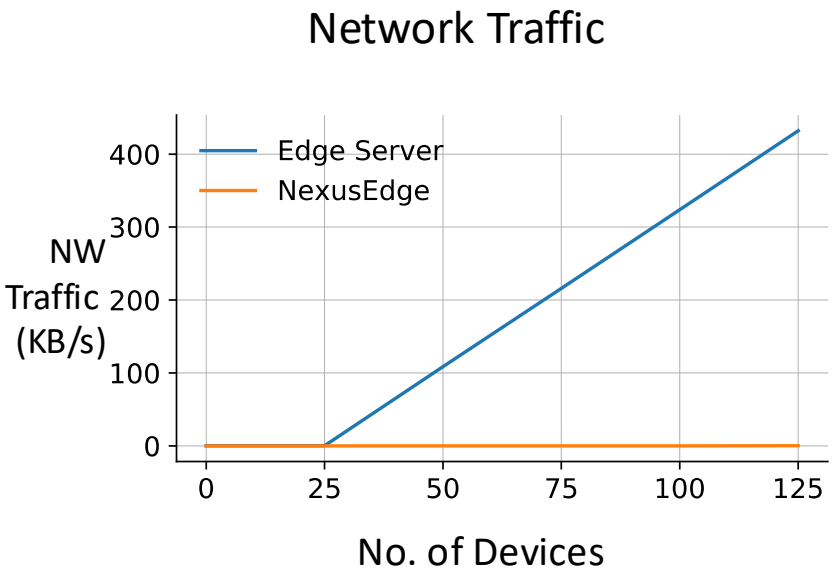
a) Scalability with No. of IoT Devices

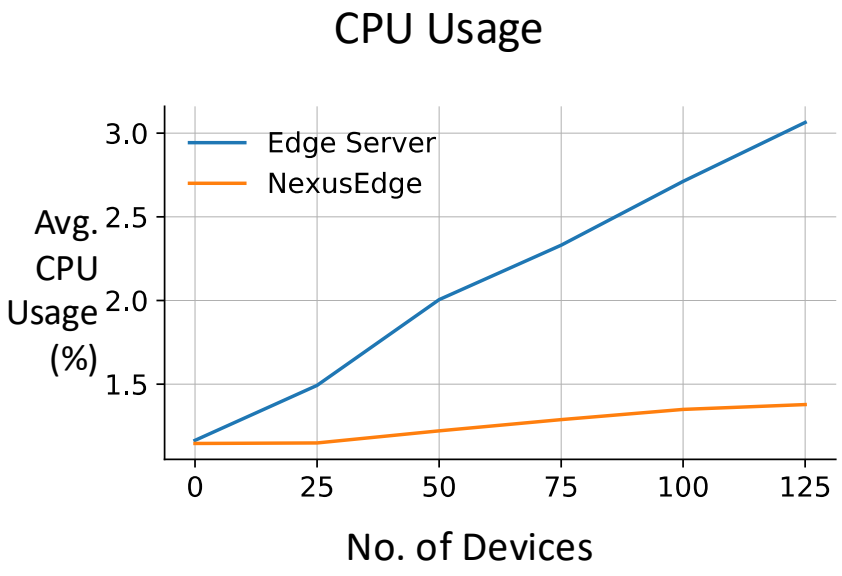
- Devices Simulated for the evaluation:

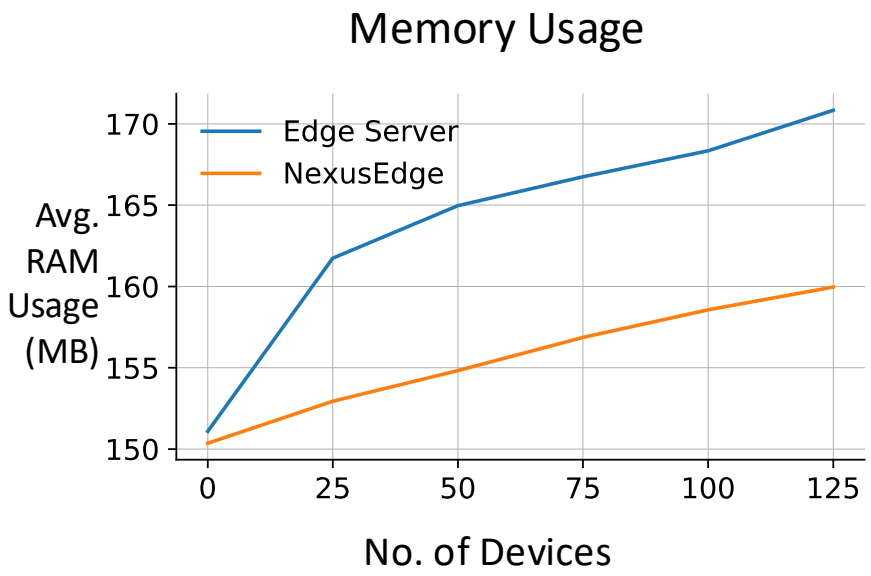
Sensor	Data Rate	Count
Temperature Sensor	100 bytes / 5 mins	50
Occupancy Sensor	100 bytes / 5 mins	50
CO_2 Sensor	100 bytes / 15 mins	10
Smart Meter	500 bytes/s	10
Camera	100 KB/s (assuming 720p at 15fps)	5
		125 (total)

- 125 devices equally distributed on server and gateways, i.e. each hosting 25 devices

Scales well with more IoT devices







- NexusEdge has substantially lower network traffic
  - NexusEdge gateways only stream data when applications require it

- Resource usages slightly higher for Edge Server architecture to handle data flooding

Scalability with No. of Edge Applications

- Setup
  - Edge Server: 4 Raspberry Pis as Gateways, 1 Raspberry Pi as Server
  - NexusEdge: 5 Raspberry Pis as Gateway, no Server

- Measure:
  - CPU usage
  - Memory usage
  - Network traffic
  - End-to-End Latency (from device to application)

Scalability with No. of Edge Applications

- Devices Simulated for the evaluation:

Sensor	Data Rate	Count
Temperature Sensor	100 bytes / 5 mins	50
Occupancy Sensor	100 bytes / 5 mins	50
CO_2 Sensor	100 bytes / 15 mins	10
Smart Meter	500 bytes/s	10
Camera	100 KB/s (assuming 720p at 15fps)	5
		125 (total)

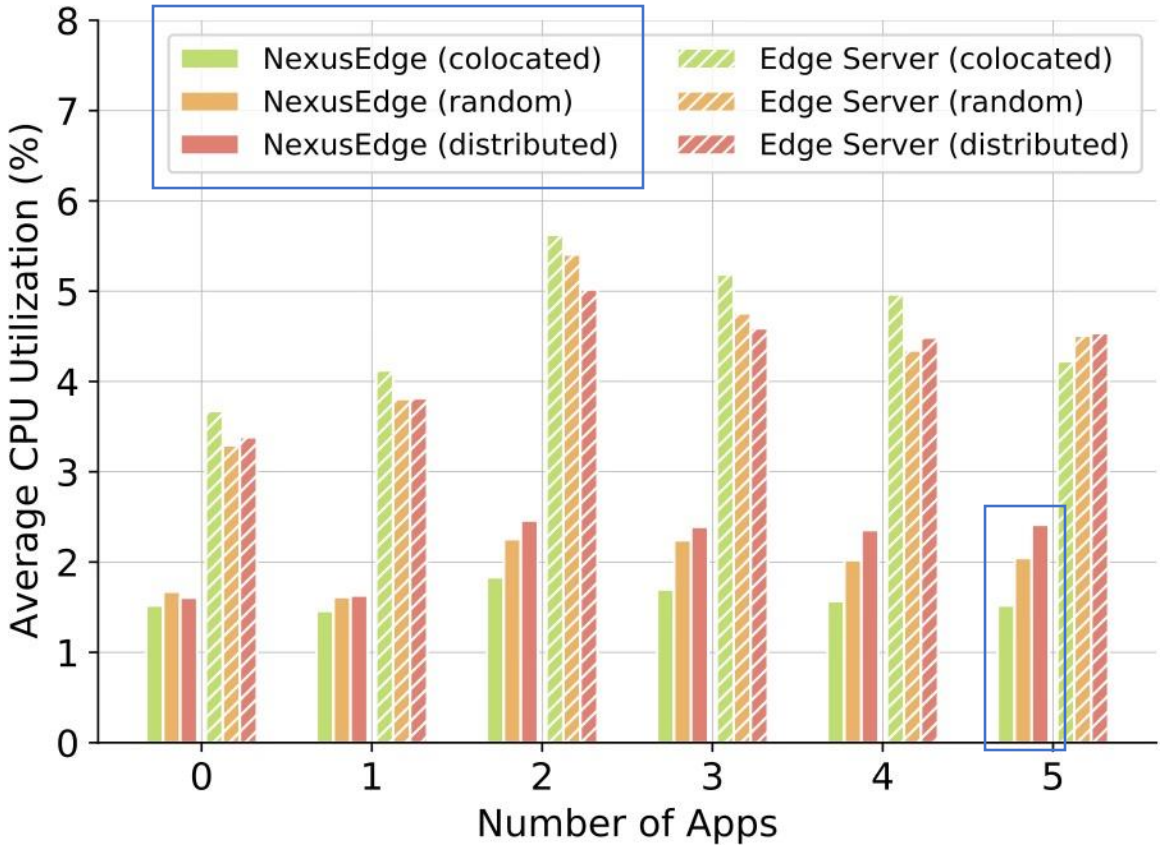
Scalability with No. of Edge Applications

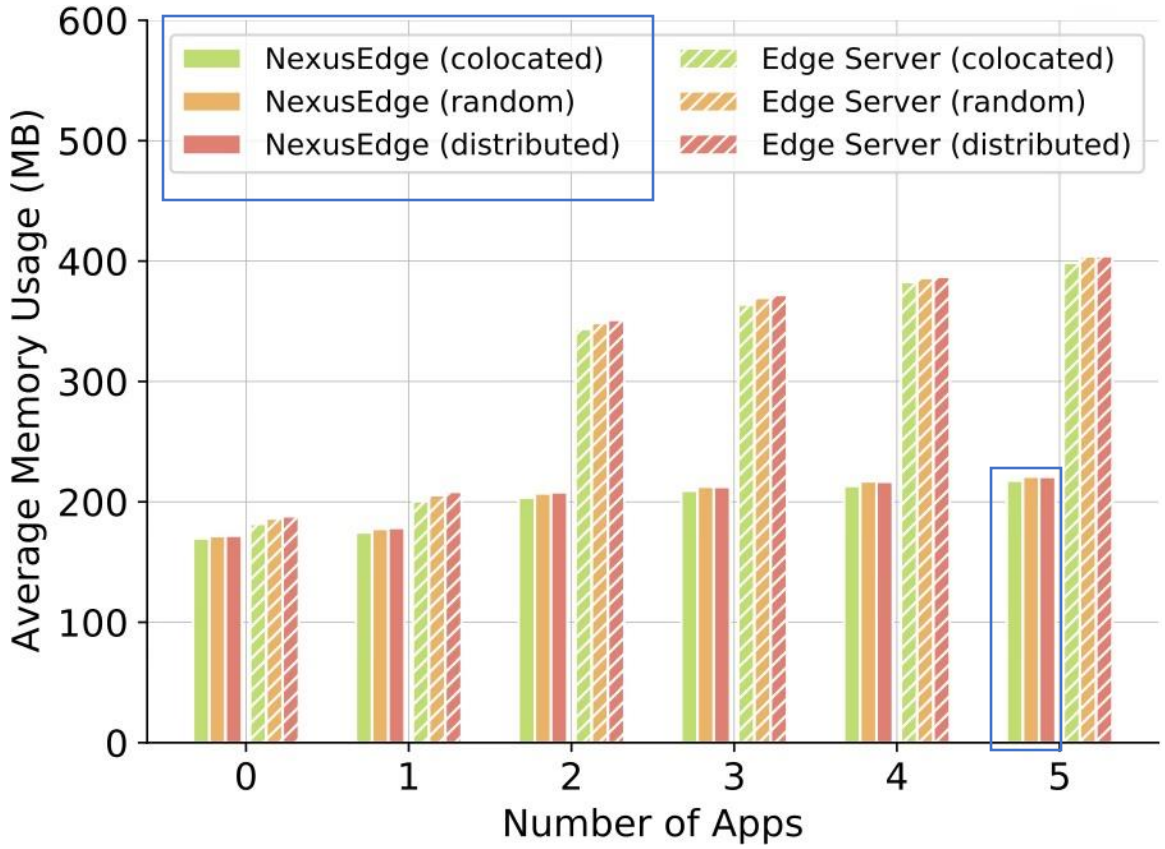
App	Sensors Used
Power meter anomaly detection	10 Smart meters
Object detection in secure areas	5 Cameras
User comfort monitor	50 Temperature Sensors
Air quality monitoring	10 CO_2 Sensors
Room scheduling	50 Occupancy Sensors

b) Scalability with No. of Edge Applications

- Three different device distributions
  - **Colocated**: Devices required for app available on executing gateway/server
  - **Distributed**: Devices required for app equally distributed on all five nodes
  - **Random**: Devices randomly distributed on all five nodes

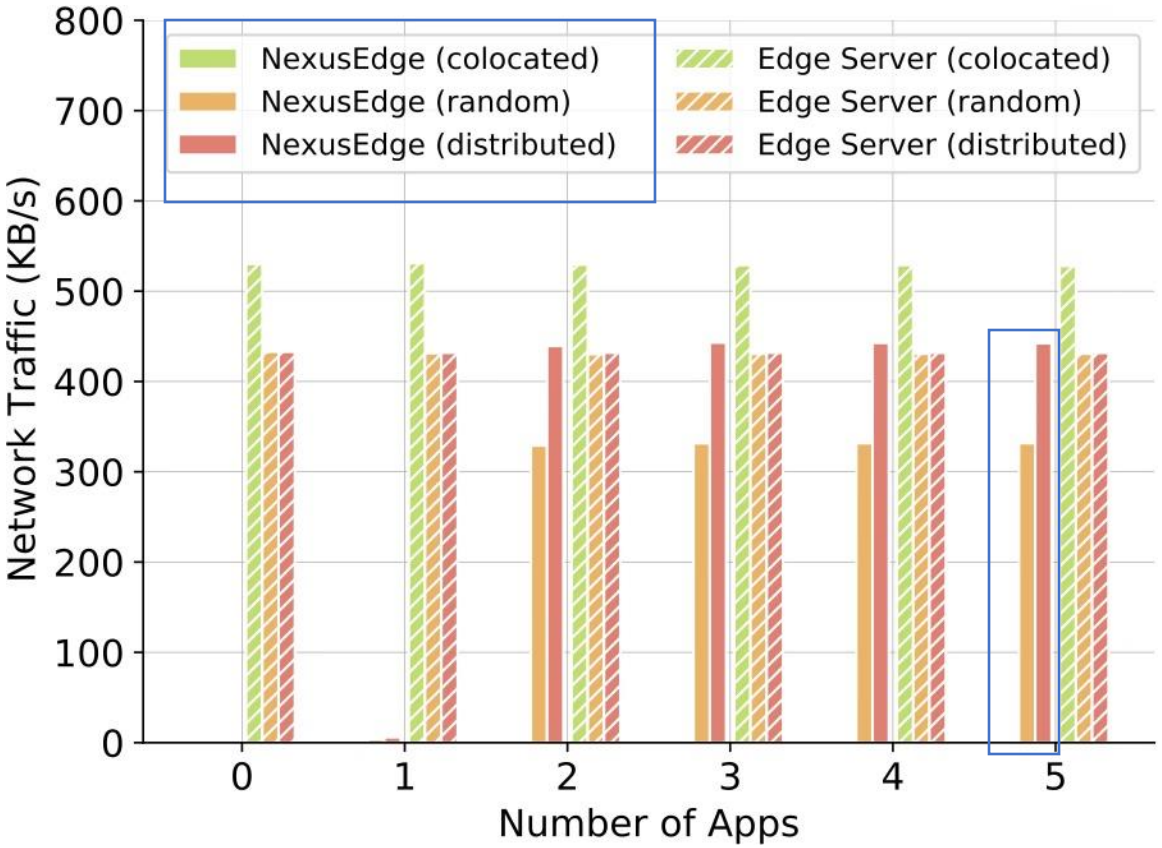
b) Scalability with No. of Edge Applications

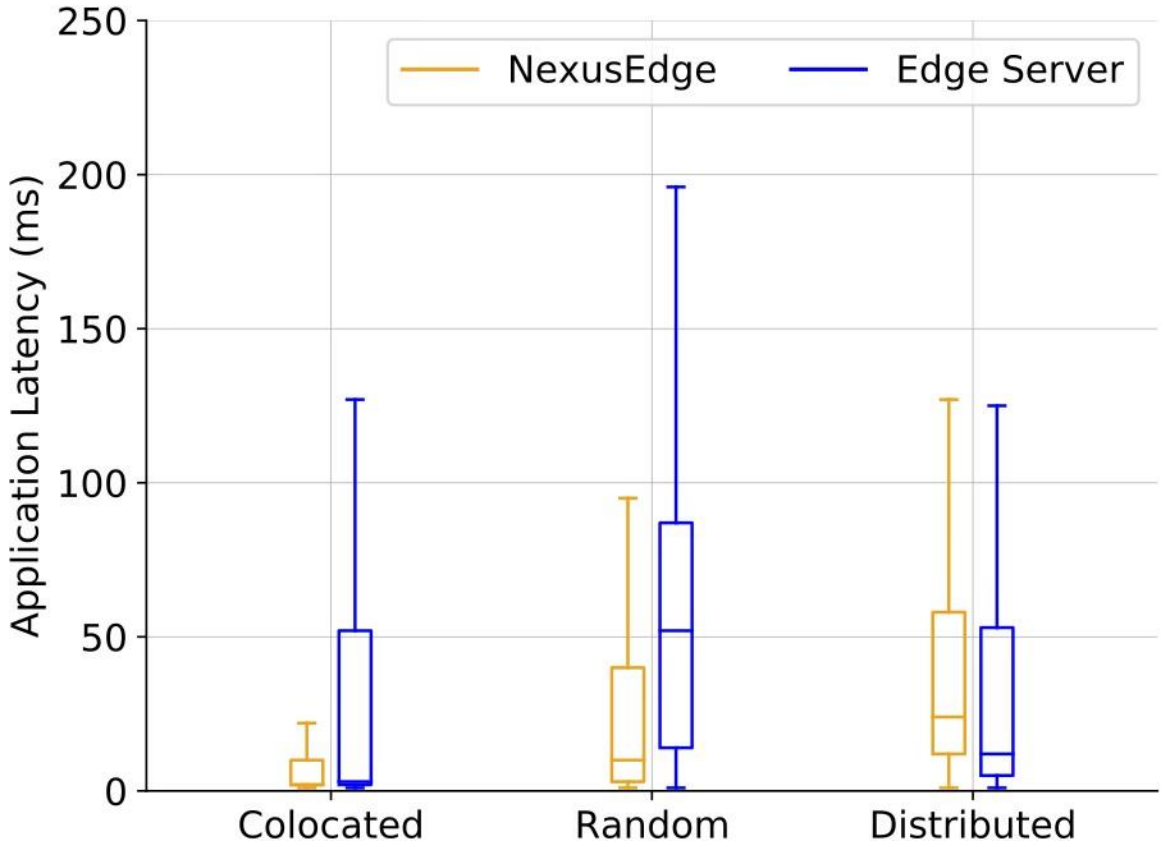




- Memory usage of ES is ~90% higher than NexusEdge, across distributions
  - Reason: Single edge server to execute applications
- CPU and memory usages in NexusEdge are very low, < 3% and 250 MB (out of 4GB) with 5 apps
  - Shows potential to run edge applications

Scales well with more edge applications



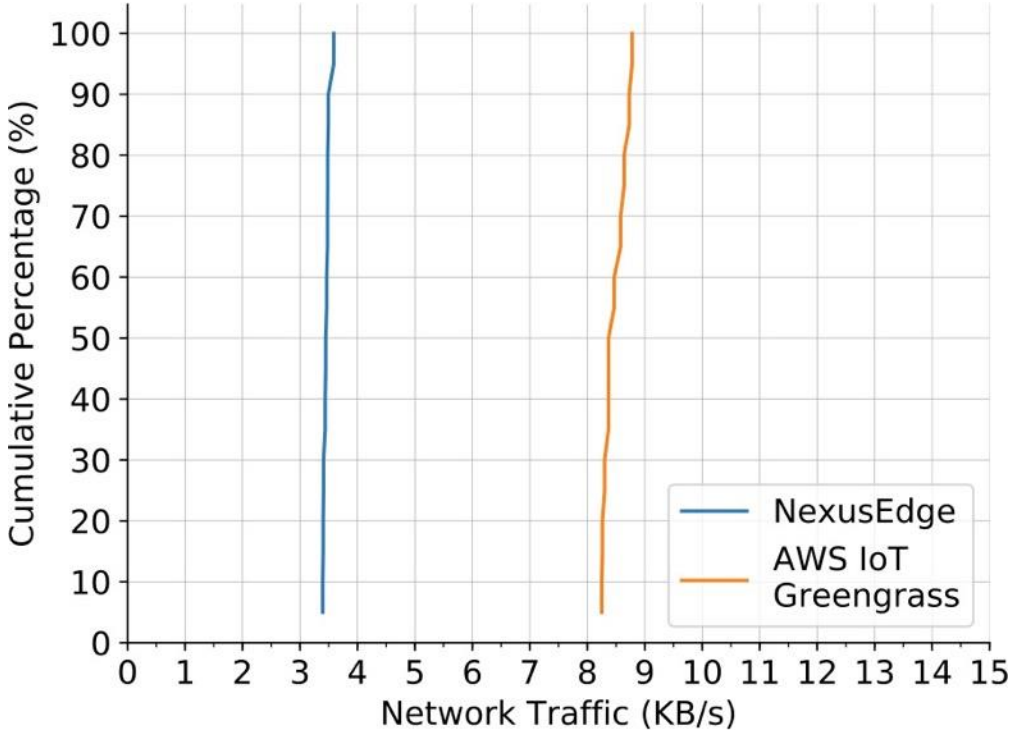


- Centralized architecture suffers from high network traffic
  - NexusEdge utilizes locality of data while scheduling applications
- NexusEdge offers substantially better application latency (except for the distributed case)

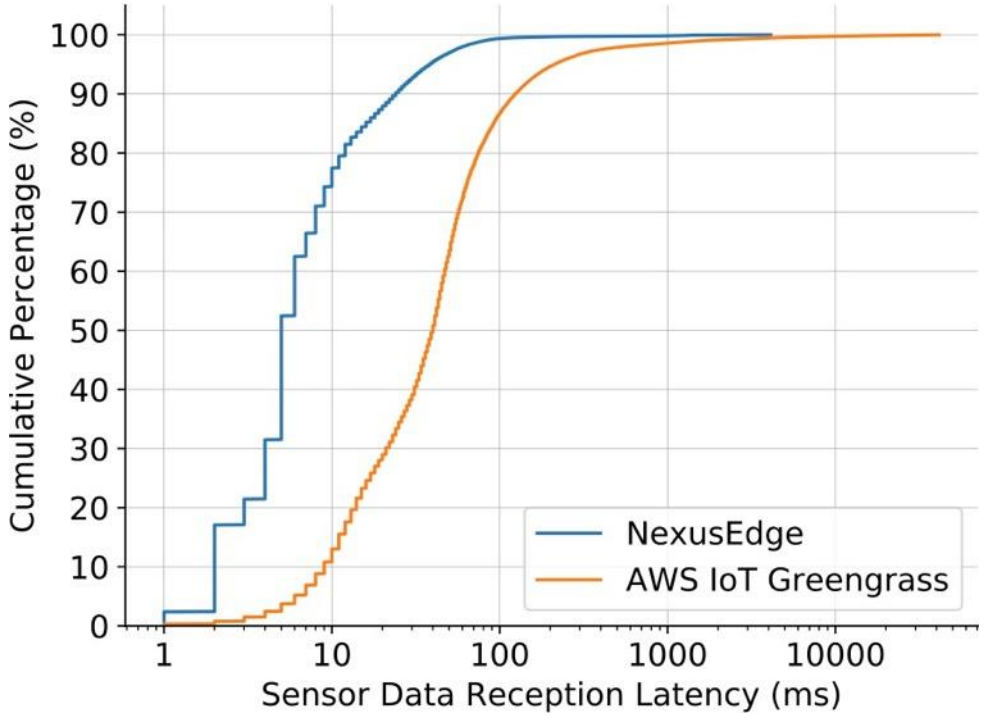
Comparison with AWS IoT Greengrass

- We compare against Amazon's AWS IoT Greengrass
- Edge Computing platform that follows a Centralized architecture
- Evaluation Setup
  - Use 70 sensors from our testbed: temperature and occupancy
  - App: Send alert for suboptimal cooling and wasted energy
- Run application and measure network traffic and end-to-end latency

Improved Traffic and Latency Compared to Greengrass



Mean Traffic: 3.46 KB/s vs 8.48 KB/s
(2.5x lower)

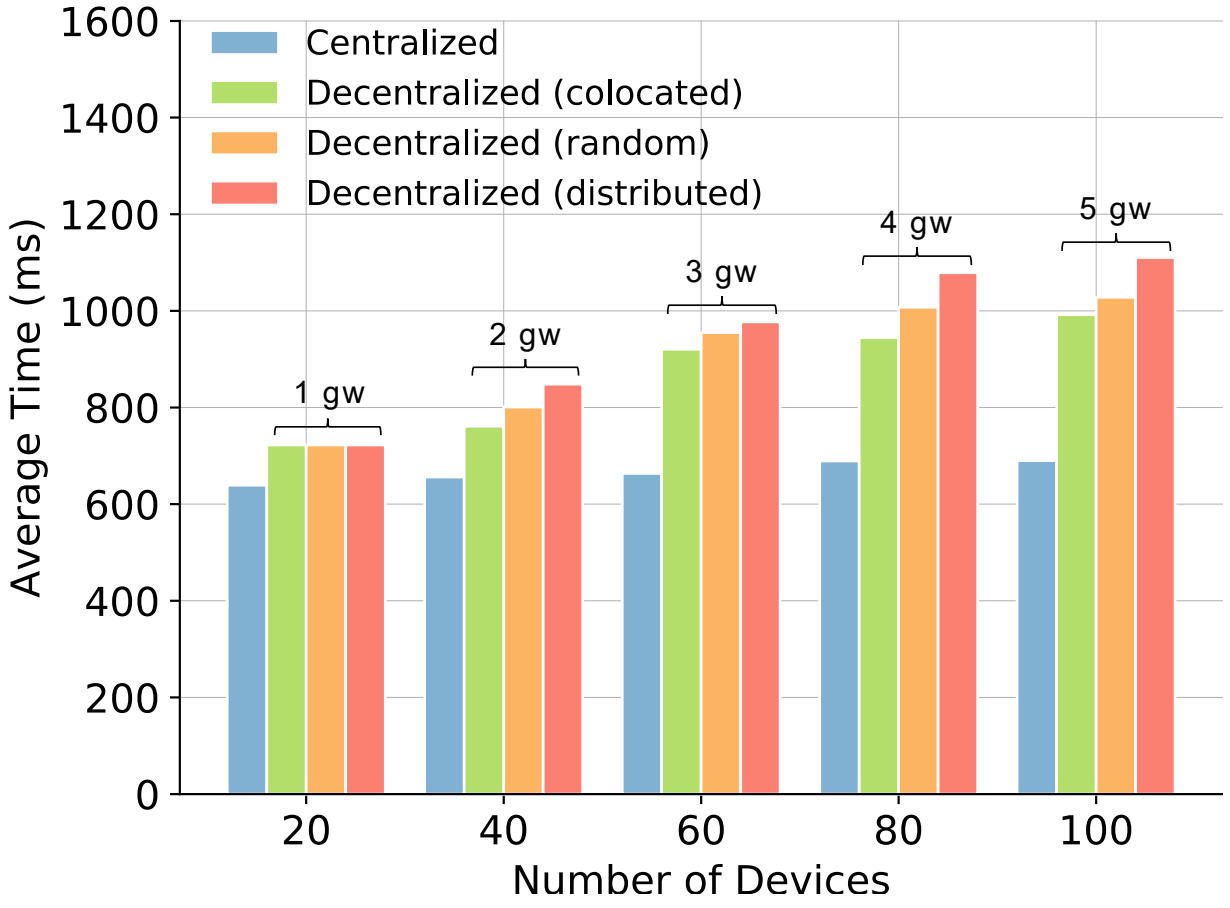


Mean Latency: 13.83 ms vs 142.74 ms
(10x faster)

Decentralization Overhead

- Gateways have some decentralization overhead for gateway coordination
- For servers, no scheduling is needed
- Deploy an app and measure deployment time on server vs gateways

NexusEdge has reasonable Decentralization Overhead



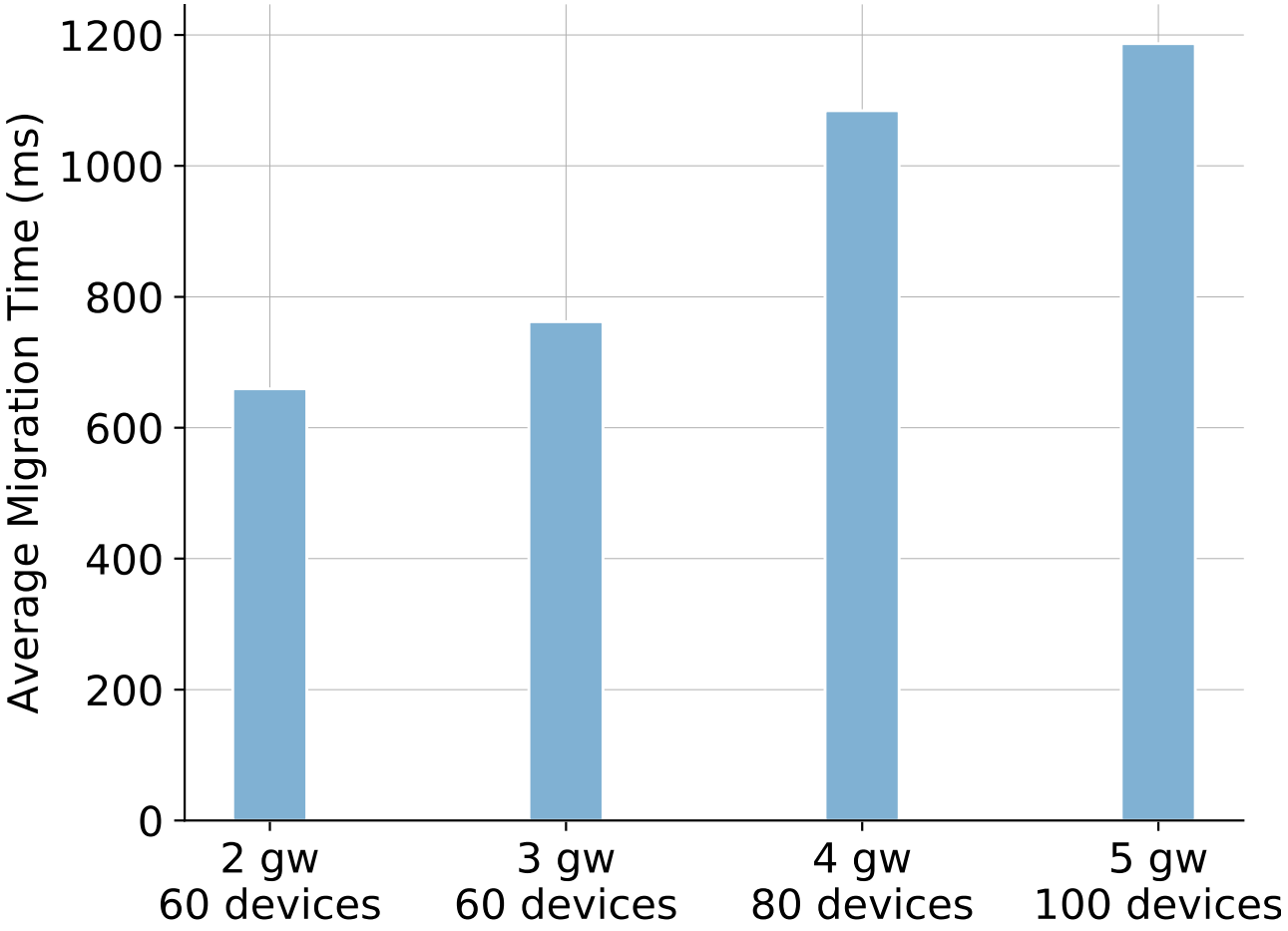
- Deployment time of NexusEdge is 1110.4ms compared to 690.1ms for centralized
- Overhead: 420ms

Resiliency to Gateways Failures

- Measure resiliency response time to failures

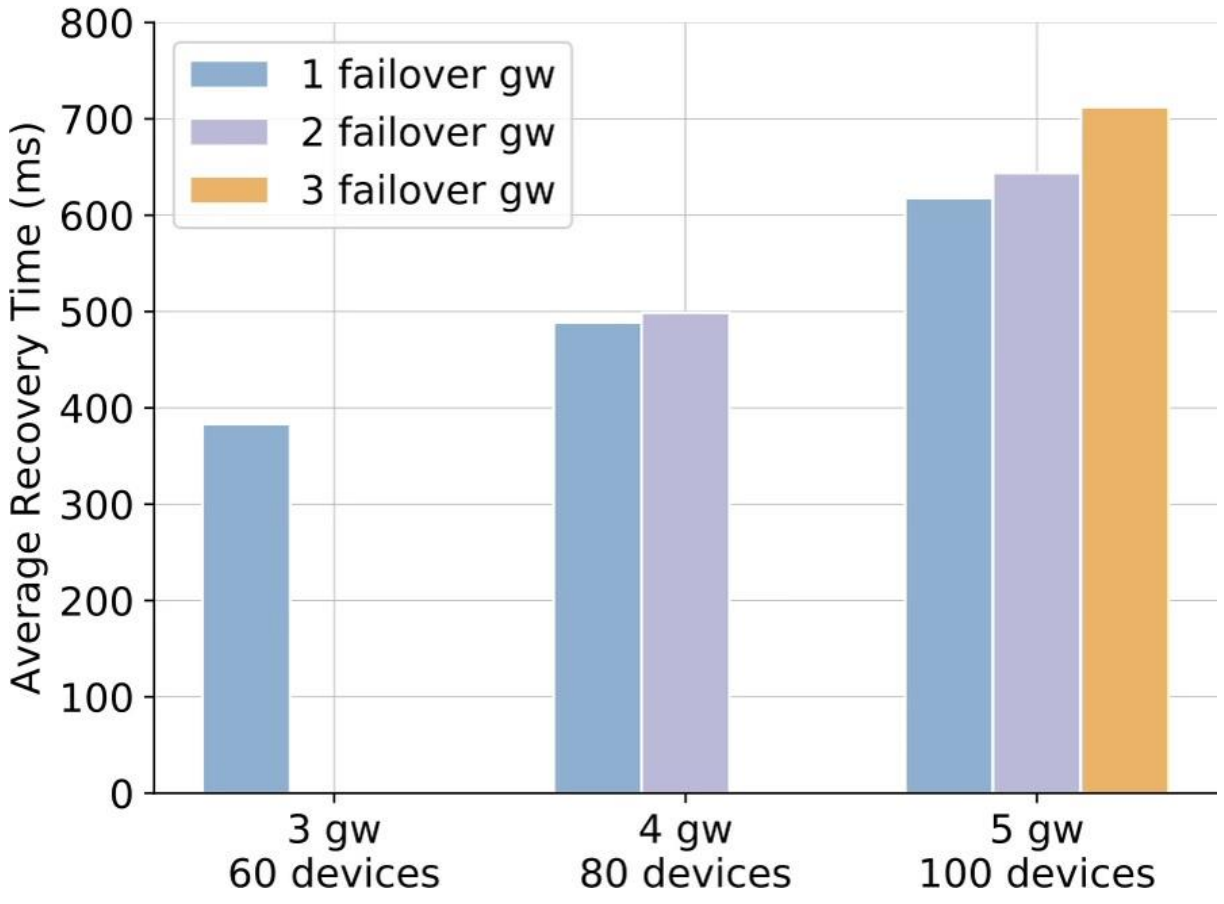
- NexusEdge provides resilience for:
  1. Executor Failure: If a gateway executing an app fails, it’s migrated to a different gateway
  2. Provider Failure: If a gateway providing data streams to an app fails, the middleware tries to reinstate data streams from alternative sources

NexusEdge Quickly Migrates Apps when Gateways Fail



- Fail the executor gateway and measure migration time
- Migrating an app using 100 data streams takes ~1.2s
- Migration time increases linearly

NexusEdge Quickly Reinstates Streams



- Measure response time to reinstate stream from a *failover* gateway
- Fast recovery: 712.3ms even with 5 gateways, 100 devices

Case Study: Distributed Federated ML

- Implemented a distributed federated machine learning without cloud support

- Uses several distributed APIs provided by the middleware

Limitations and Future Work

- Currently offers limited security provisions
  - Interfaces for device registration, discovery, and auxiliary devices could be updated
- Task scheduler to improve QoS of edge apps by utilizing heterogeneity of edge hardware
- Device stream routing for user-driven privacy policy enforcement

Takeaways

- IoT Gateways are ubiquitous and becoming increasingly performant
- Increase their utility in the computing continuum before depending on more powerful machines
- NexusEdge middleware a first step towards a vision to democratize edge computing

Thank You!

I’m on the job market!

Nabeel Nasir
www.cs.virginia.edu/~nn5rh