More Than The Sum of Its Things
Resource Sharing Across IoTs at The Edge

Aliaa Essameldin, Mohammed Nurul Hoque, Khaled A. Harras
Growth and Diversity of Applications

Heterogeneous Devices
Middleware
The Objective

Seamless resource and context sharing at the edge
The Hive
What is The Hive?
What is The Hive?

Edge-based middleware system maximizes resource sharing among IoT devices
What is The Hive?

**Data Exchange**
Decouple applications and sensors

**Processing**
Share compute information

**Core**
Gather data and make decisions
Hive Protocol
Governs communication across all bees
Agenda

2. Related Work
3. Hive Motivation and Objectives
4. Hive System Architecture
5. Hive Protocol
6. Preliminary Assessment
7. Video Integration and Evaluation
8. Conclusion and Future Work
Agenda

2. Related Work

3. Hive Motivation and Objectives

4. Hive System Architecture

5. Hive Protocol

6. Preliminary Assessment

7. Video Integration and Evaluation

8. Conclusion and Future Work
Agenda

2. Related Work

3. Hive Motivation and Objectives

4. Hive System Architecture

5. Hive Protocol

6. Preliminary Assessment

7. Video Integration and Evaluation

8. Conclusion and Future Work
When to Hive?
Scenario 1

I walk and talk on a video call. Zoom starts using video from my home camera system.
Scenario 2

Both zoom and my camera’s security software use a common module to run facial recognition, so computation is saved.

*Assume Zoom is using video from my home camera system.
Scenario 3

A game on my phone begins to use compute resources on my laptop
Scenario 4

A mental health monitoring system factors in temperature readings from a Raspberry Pi.
Goals

1. Sensor Data Sharing

2. Application Data Sharing

3. Compute Sharing

4. Optimal Sharing Decisions
Agenda

2. Related Work

3. Hive Motivation and Objectives

4. Hive System Architecture

5. Hive Protocol

6. Preliminary Assessment

7. Video Integration and Evaluation

8. Conclusion and Future Work
Agenda

2. Related Work
3. Hive Motivation and Objectives
4. Hive System Architecture
5. Hive Protocol
6. Preliminary Assessment
7. Video Integration and Evaluation
8. Conclusion and Future Work
1. Track Worker Bees

2. Collect List of Available Sensors From Workers

3. Decide optimal Seeker-Provider Pairings

4. Collect Resource Usage from Workers and Balance It

The Queen (Core) Optimizing the System
The Queen (Core)  
Optimizing the System

1. Track Worker Bees
2. Collect List of Available Sensors From Workers
3. Decide optimal Seeker-Provider Pairings
4. Collect Resource Usage from Workers and Balance It
The Worker Core
Assisting the Queen in Optimizing the System

1. Asks Queen for Seeker-Provider Pairing for this Bee
2. Asks Queen which Agents this Bee should offload to
3. Register this Bee with the Queen
4. Answer Queen’s Query for Resource usage
5. Answer Queen’s Query for Available Sensors
The Seeker Interface
Accessing the Hive’s Resources

- **App Request Handler**: Receives and authenticates the request. Records seeker in Local DB.
- **Data Seeking Client**: Makes the call to the relevant Provider(s) for the request.
- **Local DB**: Responsible for any Data filtering or aggregations which get done.
- **Data Manager**: Returns top providers for the request. Set by Providers Tracker in Worker Core.
The Provider Interface
Answering Data Requests

- **Data Request Handler**: Parses the request.
- **Data-Providing Server**: Collects data from hardware modules and passes it to data manager if needed.
- **Data Manager**: Responsible for any Data operations which need to be done before returning.
- **Hardware Modules**: Data is collected from here.
The Computational Manager
Allowing Access to Compute Resources

Receives processing requests and data streams

Contacts the appropriate Agent Bee (as determined by In Worker Core) and passes results back to Seeker/Provider Interface
The Computation Agent
Performing computation for a manager

Receives processing requests, invokes software components, and passes back results

Contained code snippets that contain different computationally demanding algorithms
Agenda

2. Related Work
3. Hive Motivation and Objectives
4. Hive System Architecture
5. Hive Protocol
6. Preliminary Assessment
7. Video Integration and Evaluation
8. Conclusion and Future Work
Agenda

2. Related Work
3. Hive Motivation and Objectives
4. Hive System Architecture
5. Hive Protocol
6. Preliminary Assessment
7. Video Integration and Evaluation
8. Conclusion and Future Work
The Tradeoff
The Tradeoff

(a) 360p

(b) 480p
Discussion
• **Good Abstractions.** The levels of abstraction align with the context.

• **Rigorously Tested.** Example of thorough experimentation.

• **Software Engineering.** All about the architecture.

• **Security.** Do I want all my devices talking with each other?

• **Applicability.** Not totally convinced by their usage cases.
Any Questions?
What other applications are there for the Hive?
What sort of security would have to be implemented in the Hive architecture?
Hive introduces additional delay in the network. What are some applications where this delay is unacceptable?