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

Aliaa Essameldin
Mohammed Nurul Hoque
Khaled A. Harras
Carnegie Mellon University
What are IoTs?

Subset of fields in which IoTs are being used
What are IoT Middlewares?
More Than The Sum Of Its Things

IoT Middleware

Non Exhaustive List of Solutions That existed by 2016 to address Common IoT Challenges

IoT Middleware | Classic Solutions

Limitations of Classic Solutions:
1. Delay
2. Cost
3. Privacy

Non Exhaustive List of Solutions That existed by 2016 to address Common IoT Challenges

More Than The Sum Of Its Things

IoT Middleware | Moving to the Edge

Comparison of Total 3-year Management & Processing Costs of Cloud-only vs. Edge + Cloud with 95% Edge Data Reduction (200 Miles Distance)

Cloud + Edge Computing is 36% of the Cost of Cloud-only Computing when the Reduction in Data Volume is 95%.

3-year Cost of Managing & Processing IoT Data

<table>
<thead>
<tr>
<th></th>
<th>Cloud-only</th>
<th>Edge + Cloud</th>
</tr>
</thead>
<tbody>
<tr>
<td>$90,000</td>
<td>$80,531</td>
<td>$28,927</td>
</tr>
</tbody>
</table>

Source: © Wikibon IoT Project. Reference Models AWS IoT Service & Pivot3 Server SAN. Assumptions: Edge reduces IoT Traffic by 95%.

Azure IoT Edge

Cisco IOx Edge Solution

Industry Leaders such as Cisco, Apache and Azure started providing IoT Solutions at the Edge

Carnegie Mellon University NSL
IoT Middleware | What’s Next?

Existing Solutions

Camera Surveillance

Smart Video-conferencing

Emotion Detection Journal

Middleware

Middleware

Middleware
IoT Middleware | What’s Next?

Existing Solutions

- Camera Surveillance
- Smart Video-conferencing
- Emotion Detection Journal

Middleware

Carnegie Mellon University
IoT Middleware | What’s Next?

Existing Solutions

App 1

App 2

App 3
More Than The Sum Of Its Things

IoT Middleware | Overcoming the Limit of the Thing

Existing Solutions

Our Vision

Sensors

Processors

Applications

App 1

App 2

App 3

App 1, Usr1

App 2

App 3

App 1, Usr2

App 4

App 5

Carnegie Mellon University

SEC ‘20
Vision: Develop an IoT Middleware Solution which operates at the Edge which fully capitalizes on the overall resource pool available.
The Hive Middleware

The Hive will interface with applications as well as local resources on a device to create the abstract pools presented in our vision.
1. Data Stage: Decouple Applications and Sensors
Hive Architecture | Seamless Data Exchange

1. Data Stage: Decouple Applications and Sensors
Hive Architecture | Sharing Computational Resources

2. Processing Stage: Decouple processors and devices
2. Processing Stage: Decouple processors and devices
Hive Architecture | Optimizing Resource Utilization

Worker Bee

Seeker Interface
- App Request Handler
- Data Seeking Client
- App DB
- Data Manager

Provider Interface
- Data Request Handler
- Data-Providing Server
- Hardware Modules
- Data Manager

Worker Core
- Providers Tracker
- Agents Tracker
- Resource Monitor
- Service Discoverer

Comp. Manager
- Comp. Client
- Offloader

Comp. Agent
- Comp. Server
- Software Modules

I. DATA STAGE
II. PROCESSING STAGE
3. Core Stage: Connect the decoupled elements in an optimal manner
Hive Architecture | Optimizing Resource Utilization

Queen Bee will also have the regular architectural components, since it is simply one of the functional devices of the Hive.
Queen Election

- Distributively elected node
- First come-first-serve basis
- In case of a tie, the algorithm selects the Queen with the highest up-time
Hive Protocol

More Than The Sum Of Its Things

SEC ‘20

Carnegie Mellon University

NSL
Prototype Evaluation
Queen Election

5 Pi’s boot up at the same time.

Kill queen to trigger re-election.
More Than The Sum Of Its Things

Vokaturi on Hive

![Diagram showing Hive and single mic comparison]

**Single mic**

<table>
<thead>
<tr>
<th>Confidence Level (%)</th>
<th>Time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>80</td>
</tr>
</tbody>
</table>

**Hive**

<table>
<thead>
<tr>
<th>Confidence Level (%)</th>
<th>Time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>80</td>
</tr>
</tbody>
</table>

Hive senses on Laptop

Hive senses on Pi

Carnegie Mellon University
Video Module Implementation

Face detection software module in OpenCV.

Seeker picks stream with highest confidence.

Timing from detection to app, with NTP.
More Than The Sum Of Its Things

Best Face View

Carnegie Mellon University
Resource Usage

Streaming from camera to many apps on the same device.

Detection algorithm runs once in Hive, vs. once per app without hive.
Scalability

Many seekers, one provider. Breaks when bandwidth exceeded. Similar results for one seeker, many providers.
Conclusion
Conclusions

Existing frameworks do not utilize the full potential of IoT.

Decoupling applications, sensors and processors using the a generic Architecture and Protocol like the Hive’s can

1. overcomes these limitations
2. enable a new generation of host-independent apps.
3. saves costs while introducing minimal overhead
Thank You.

For questions and comments, please contact us on:

essameldin@cmu.edu  mnur@cmu.edu  kharras@cmu.edu