IoT Middleware Overview

Jug Venkatesh

System Energy Efficiency Lab

seelab.ucsd.edu
Middleware – or, haven’t we covered this already?

- Communication
- Storage
- Data management
- Software
- Platforms

IoT Applications as a Graph
So why differentiate middleware?

- Unify/standardize protocols
- Infrastructure “glue” to provide abstraction and layers
- API/Interfaces:
  - Connect with existing middleware components
  - Establish interfaces for a component to be developed/modified
Functional middleware building blocks

- Interoperability
- Security/Privacy
- Context Detection
- Device Discovery
- Scalability
Interoperation

- Network
  - Physical layer-agnostic
  - Data agnostic

- Syntactic
  - Data format/structure
  - Data translatability

- Semantic
  - Meaning of data – machine interpretation
Interoperation & Layering

- APIs & abstraction: longstanding software engineering traditions

- Help make interoperability possible:
  - Separates domain responsibilities
  - Identifies requirements for modular implementation (API)
  - Builds upon established requirements/specifications
Context Detection

- Data $\rightarrow$ Knowledge
- Answer fundamental questions about data:
  - **What** does it mean?
  - **Who** needs to know about it?
  - **Why** is it important?
  - **When** did this happen?

- Metadata encapsulates data (more layering!)
  - Enables *automated consumption* by IoT applications
Algorithms for Context Detection

- Typically fall under machine learning:
  - **Model generation:** formulate output based on a functional combination of input
  - **Discrete-event detection:** pattern matching, anomaly detection, clustering
  - **Supervised/Unsupervised learning** of output states
  - **State models:** decision trees, Bayesian networks
Device Discovery

- Identify other nearby devices, and make yourself known
  - Semantic state – *device ontology*

- Virtual device interface
  - Identification/Association
  - Capabilities/RPCs
  - State information about rest of infrastructure

- Enables further communication
  - Transfer of information
  - Routing (P2P)
  - Application-specific implementation changes
Device Discovery Implementations

- Bluetooth Beacons
  - One-directional location/metadata detection

- WiFi Aware
  - One-directional identifier broadcast
  - Metadata can allow for ad-hoc, two-way communication

- Physical Web
  - One-way signal + URL for additional connectivity

- Protocols:
  - DNS-SD (service discovery)
  - Multicast DNS
  - W3C Network service discovery
Security & Privacy

- Fundamental problem:
  - Nodes need to discover other nodes, but still maintain privacy

- Middleware solution: multi-tier privacy (layering yet again!)
  - Management of discoverability, data access
  - Use device semantics to identify accessibility...but how to limit discoverability?
Security Implementations

- Public Key Infrastructure
  - Requires backend access

- Group Key: Authorization before Inclusion
  - Requires new device to register oneotime with backend

- Local Access Control:
  - Local DNS

- Multi-stage access:
  - Public – advertise ID only
  - Semi-Private: Authorization allows metadata access
  - Private: Full access allows data access
Scalability

- Expected:
  - Trillions of devices
  - Exabytes of data

- Current reality:
  - Millions of devices
  - Approaching petabytes of data

- Fundamental scalability problems:
  - **Storage**: persistent? raw? summarized?
  - **Processing**: how much data to use? how do applications scale with more data availability?
  - **Communication**: how much data to transmit? what infrastructure will scale to allow more data transmission while still meeting application deadlines?
Discussion: Middleware relationships