So Many Sensors, So Little Data
the half-baked idea...

Seth Holloway, **Drew Stovall**, Angela Dalton, Christine Julien
Mobile and Pervasive Computing Group
The University of Texas at Austin
{sethh,dstovall,adalton,c.julien}@mail.utexas.edu

First Int’l Wkshp on Software Architectures and Mobility (SAM ‘08)
Presented May 10, 2008 Leipzig, Germany
So many sensors...

- Self contained **sensors** and **actuators** are (almost) affordable
- Data collection on sensor nodes is easy and native
- Communication between sensor nodes is common
- Collaboration between sensor nodes is under research
So little data...

- Communication between sensor nodes and “outside world” is difficult
- Bridging sensor network to PC requires custom code
  - Design
  - Implementation
  - Documentation
  - Testing / debugging
  - Maintenance
Who needs the data?

• Applications require
  – simple, but non-trivial logic
  – long-term, complex data storage
  – interaction with users and devices

• Examples
  – Coffee pot usage monitor (buttons, dials)
  – Home automation (lights, TV controller, etc)
  – Inventory management (RFID scanners)
Conventional solution

- Nodes: bytes ↔ Applications: streams, db’s
- To bridge the gap, custom code to:
  - Serialize into bytes for transmission (node)
  - Route discovery (node)
  - De-serialize/Re-serialize at network bridge (both)
  - De-serialize bytes on server (PC)
- Extra code/complexity prevents average programmers from using sensor data
We have seen this before...

• There is a domain of applications where
  – Programmers *focus* only on the application
  – Vetted *protocols* already exist
  – Existing, interchangeable *tools*
  – Large quantity of *examples*, tutorials, experts
  – *Scalability* issues are already addressed
  – *Users* are already comfortable
We have seen this before...

- There is a domain of applications where
  - Programmers **focus** only on the application
  - Vetted **protocols** already exist
  - Existing, interchangeable **tools**
  - Large quantity of **examples**, tutorials, experts
  - **Scalability** issues are already addressed
  - **Users** are already comfortable

**Web Applications**

---

*So many sensors, So little data (SAM@ICSE ’08)*
Our solution

• Phase 1 – Build application
  – Developer creates application using web technologies
  – Input received via forms
  – Unit tests / manual tests to resolve application issues

• Phase 2 – Receive data from sensors
  – Sensors given URLs and field names of web forms
  – Sensors given (optional) configuration
  – Sensors deliver/ retrieve data via HTTP
Example

So many sensors, So little data (SAM@ICSE ’08)
Example

.../add_cups

Cups Added:

-1

.../num_cups

Cups Available: 10

So many sensors, So little data (SAM@ICSE '08)
Cups Added: -1

Cups Available: 10
Example

.../add_cups
Cups Added: -1

HTTP_post

HTTP_get

.../num_cups
Cups Available: 10

So many sensors, So little data (SAM@ICSE ’08)
Example

So many sensors, So little data (SAM@ICSE ’08)
Challenges

• Higher bandwidth usage vs. custom protocol
  – Custom protocols are not excluded though
  – Migrate to custom protocol when required

• Requires sensor node implementation
  – Cost is amortized once written
Benefits

• Well designed, well understood protocol
  – Solutions to many challenges already built in
• Large range of existing web-application servers
  – Mature, stable, supported, standards based
• Large range of programming tools
  – Editing, commenting, debugging, testing
• Scalability built in
  – Fragmentation, partial downloads, if-modified-since
  – Proxy servers
• Compatibility with internet, home networks, COTS hard/software
So Many Sensors, So Little Data

Seth Holloway, Drew Stovall, Angela Dalton, Christine Julien
{sethh,dstovall,adalton,c.julien}@mail.utexas.edu
http://mpc.ece.utexas.edu

First Int’l Wkshp on Software Architectures and Mobility (SAM ‘08)