Bezirk is to the IoT what bio/organic is to agricultural products.

IoT that tastes better

$$
mainstream business models
revolve on the value of data
for service providers

$$$
user-centrism and privacy

$$
© 2016 Robert Bosch LLC and affiliates. All rights reserved.
IoT responsiveness

[Partner Maintenance] Chamberlain MyQ garage doors
Scheduled Maintenance Report for Wink

We’ve been informed that the Chamberlain MyQ garage door web services will be undergoing maintenance on Monday, Nov. 10 01:00 EST (Nov. 9, 22:00 PST). The maintenance is expected to last about 4 hours and users will not be able to use the Chamberlain garage doors from the app during this time.

Period: 3 months ago: Nov 06, 2014 22:32 UTC

IoT reality check fragmentation

corporations

- consumer Data has business value
- little incentive to share that value
not all **Data** is created equal

mainstream IoT
raw **Data** pumped to the cloud

what does that mean to **you**?
mainstream IoT
Loss of privacy

Social reasons for boundaries

reclaim the expected boundaries of privacy
Empower users to reclaim boundaries

Privacy

- Quality: local integration
- Respectful: data ownership

Security model

user-centric Spheres and Pipes

- Pipes between spheres / to cloud
  - secure channels for data & events
- Spheres of trust
  - boundaries of confidentiality

Who knows about this?
- Just you and I.

 Quality: local integration
 Respectful: data ownership

- Quality: local integration
- Respectful: data ownership

Security model

user-centric Spheres and Pipes

- Pipes between spheres / to cloud
  - secure channels for data & events
  - requested by services, authorized by users
  - policy enforced by middleware
  - only authorized exchanges go through
- Spheres of trust
  - boundaries of confidentiality
  - managed by users
  - easy user experience
  - balances usability & security
Introduces boundaries to the IoT for reasons:

- **Social:** privacy
- **Technical:** scope and scale

**Promotes**

- Decentralized models with linked data
- Computation in local/mobile devices

---

**Scope and Scale**

**IoT is not the Internet**

- **History:** successful apps for the internet run on general purpose computers and access remote services e.g. email, web browsing
- **Now:** must a sensor/appliance shoulder the burden of being a peer on the internet? e.g. access control, privacy...
- **Must** a sensor/appliance communicate primarily with remote services?

**IoT ≠ give every device an IP(v6) address**

<table>
<thead>
<tr>
<th>our claim</th>
</tr>
</thead>
<tbody>
<tr>
<td>IoT topology should recognize and support two kinds of communication patterns: <strong>local</strong> and <strong>remote</strong></td>
</tr>
</tbody>
</table>

---
### Scope and Scale addressing a Thing

#### Landscape of Addressing Schemas

<table>
<thead>
<tr>
<th>Address</th>
<th>Applications</th>
<th>Who Receives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node e.g. 172.16.254.1 (IPv4)</td>
<td>Internet routing: IPv4 (1981), IPv6 (1998)</td>
<td>Identified node sender must know recipient’s address</td>
</tr>
<tr>
<td>Geo e.g. (40.426, -79.965, 500) (lat, long, radius)</td>
<td>Sensor networks, safety &amp; disaster response, transportation</td>
<td>Whoever is in the area</td>
</tr>
<tr>
<td>Topic label e.g. “user location”</td>
<td>pub/sub: Java Messaging Service (message centric), Data Distribution Service (data centric)</td>
<td>Whoever subscribes to the topic</td>
</tr>
</tbody>
</table>

**Note:** Different addressing schemas solve different problems.

#### Resolving Requests

Resolving requests via semantic addressing (example: by topic)

- **Trace 1**
  - App
  - User Location
  - User Registry
  - Dishwasher
  - `locateUser.(Bob)`
  - `getUserFace.(Bob)`
  - `getUserLocation.(Bob, <here>)`

- **Trace 2**
  - Dishwasher
  - `locateUser.(Bob)`
  - `getUserFace.(Bob)`
  - `getUserLocation.(Bob, <here>)`

- How a request is resolved depends on the **status** of the environment.
Scope and Scale addressing a Thing

1. Directed, aka unicast: a sender explicitly identifies the service endpoint of the recipient - an opaque handle generated by Bezirk.

2. Publish-subscribe, aka multicast: a sender publishes messages stamped with a semantic address. Bezirk instances deliver to all recipients that match the address:
   - By design, senders have no prior knowledge of the recipients, and obtain no knowledge unless (some of) the recipients reply back.

Semantic Address includes:
- The physical location of recipients as a triple of names
  - region (e.g. "floor1" or "Pennsylvania"), area (e.g. "bedroom" or "greater Pittsburgh"), and landmark (e.g. "window" or "Point State Park").
- The pipes to be traversed, which may
  - exclude pipe traversal (local only),
  - identify one specific pipe to traverse, or
  - include all pipes in the dissemination environment.

(note) supported interaction patterns

- Classic request-reply
- Group request-reply
- Combinations of event/stream request-reply
**Introduces boundaries to the IoT**

for reasons:

→ Social: privacy
→ Technical: scope and scale

**Promotes**

→ Decentralized models with linked data
→ Computation in local/mobile devices

---

**Modeling IoT Data in a mobile world**

model interchange protocols are key

**Dragonfly**

Observe the user’s behavior as he engages the environment

- External service observes the user’s lighting choices (e.g., yellow lights on), and the context in which the choices occurred
- External service shares these observations within the network (and with U)

**Penguin**
Modeling user observations tailored subset of SSN for IoT

W3C Semantic Sensor Network Ontology

W3C SDW-WG Reqs (http://www.w3.org/TR/sdw-ucr/)
REQ 5.18 Lightweight Model for IoT and REQ 5.20 Machine-to-Machine

Modeling IoT Data in a mobile world

Bezirk Henson | 4/11/2016
© 2016 Robert Bosch LLC and affiliates. All rights reserved.

Bezirk Henson | 4/11/2016
© 2016 Robert Bosch LLC and affiliates. All rights reserved.
Modeling user profiles based on UPOS: User Profile Ontology


2nd quarter 2016
SDK, binaries

http://bezirk.com

end-users:
- access cloud services
- upload/download content

you may download

try out new services & use cases

code over
- services / apps
- protocols

 Bezirk middleware

1 player 4 player