Beetle
Sharing, flexibility and access control for Bluetooth Low Energy

Amit Levy  James Hong  Laurynas Riliskis
Philip Levis  Keith Winstein

Stanford University

June 24, 2016
Meet Grace
Meet Grace
Meet Grace.
Meet Grace
Meet Grace
Meet Grace
Meet Grace
Meet Fabian
Meet Fabian
Meet Fabian
Meet Fabian
Meet Esther
Meet Esther
Meet Esther
Meet Esther
No extensibility, no access control, limited communication
Limitations of Bluetooth Low Energy (BLE)

1. Applications must run on the gateway
2. One application at a time
3. All-or-nothing access

**Peripherals need to be simple**
- Low cost
- Low power
- Hard to update

**Gateway oblivious to peripheral functionality**
- How to virtualize generally?
- Who can access what?
The gateway should enable flexible and safe sharing
Beetle

Sharing

Applications can safely share access to peripherals without explicit coordination.

Access Control

Users can specify fine-graned access policies on peripherals and applications.

Communication flexibility

Many-to-many communication between peripherals, gateway applications and cloud applications.

Backwards compatible

 Doesn’t require changes to existing peripherals or applications.
Current Gateways: Explicit support

Dedicated drivers understand a class of peripherals:

- Printers
- Keyboards
- Disks
Current Gateways: Explicit support

Dedicated drivers understand a class of peripherals:

- Printers
- Keyboards
- Disks

Pros

- Safe virtualization
- Access control

Cons

- Need a trusted module for each device class
- Has worked OK for desktops, but does is scale?
For special case peripherals, gateway provides an exclusive, raw channel:

- Oscilloscope, hardware debugger
- USB breathalyzer, USB blood pressure monitor
- Most IoT peripherals (door locks, lights, fitness bands)
Current Gateways: Exclusive access from one application

For special case peripherals, gateway provides an exclusive, raw channel:

- Oscilloscope, hardware debugger
- USB breathalyzer, USB blood pressure monitor
- Most IoT peripherals (door locks, lights, fitness bands)

**Pros**

- Don’t need explicit support for each new device class

**Cons**

- Can’t allow multiple apps to use peripheral
- App gets complete control over peripheral
## Current Gateway Systems

<table>
<thead>
<tr>
<th>OS</th>
<th>Explicit Driver</th>
<th>Exclusive Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>HomeOS</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Linux</td>
<td>✓</td>
<td>✓ (BLE sockets)</td>
</tr>
<tr>
<td>Android</td>
<td>✓ (Google Health)</td>
<td>✓</td>
</tr>
<tr>
<td>iOS</td>
<td>✓ (Health/HomeKit)</td>
<td>✓</td>
</tr>
</tbody>
</table>

**Explicit support**  
Impractical for operating systems to anticipate new device functionality

**Exclusive access**  
No sharing, no access control, inflexible communication
Bluetooth Low Energy application protocol
Bluetooth Low Energy application protocol

Attribute Protocol (ATT):

- Key-type-value store
- Transactional flow-control
- READ, WRITE, NOTIFY, FIND-BY-TYPE...
Bluetooth Low Energy application protocol

Attribute Protocol (ATT):

- Key-type-value store
- Transactional flow-control
- READ, WRITE, NOTIFY, FIND-BY-TYPE...

Generic Attribute Profile (GATT):

- Structure over attributes
- Characteristic: Groups attributes for a value
- Services: Groups related characteristics
Bluetooth Low Energy application protocol

Attribute Protocol (ATT):
- Key-type-value store
- Transactional flow-control
- READ, WRITE, NOTIFY, FIND-BY-TYPE...

Example, heart rate monitor

Generic Attribute Profile (GATT):
- Structure over attributes
- Characteristic: Groups attributes for a value
- Services: Groups related characteristics
Bluetooth Low Energy application protocol

Attribute Protocol (ATT):
- Key-type-value store
- Transactional flow-control
- READ, WRITE, NOTIFY, FIND-BY-TYPE...

Generic Attribute Profile (GATT):
- Structure over attributes
- Characteristic: Groups attributes for a value
- Services: Groups related characteristics

Example, heart rate monitor

Find Char with UUID HRM
Bluetooth Low Energy application protocol

Attribute Protocol (ATT):
- Key-type-value store
- Transactional flow-control
- READ, WRITE, NOTIFY, FIND-BY-TYPE...

Example, heart rate monitor

Find Char with UUID HRM
Char at handle 2

Generic Attribute Profile (GATT):
- Structure over attributes
- Characteristic: Groups attributes for a value
- Services: Groups related characteristics
Bluetooth Low Energy application protocol

Attribute Protocol (ATT):
- Key-type-value store
- Transactional flow-control
- READ, WRITE, NOTIFY, FIND-BY-TYPE...

Generic Attribute Profile (GATT):
- Structure over attributes
- Characteristic: Groups attributes for a value
- Services: Groups related characteristics

Example, heart rate monitor

Find Char with UUID HRM
Char at handle 2
Read handle 2

157 BPM
Bluetooth Low Energy application protocol

Attribute Protocol (ATT):
- Key-type-value store
- Transactional flow-control
- READ, WRITE, NOTIFY, FIND-BY-TYPE...

Generic Attribute Profile (GATT):
- Structure over attributes
- Characteristic: Groups attributes for a value
- Services: Groups related characteristics

Example, heart rate monitor

Find Char with UUID HRM
Char at handle 2
Read handle 2
Char value at handle 4
Bluetooth Low Energy application protocol

Attribute Protocol (ATT):
- Key-type-value store
- Transactional flow-control
- READ, WRITE, NOTIFY, FIND-BY-TYPE...

Generic Attribute Profile (GATT):
- Structure over attributes
- Characteristic: Groups attributes for a value
- Services: Groups related characteristics

Example, heart rate monitor

<table>
<thead>
<tr>
<th>Find Char with UUID HRM</th>
<th>Char at handle 2</th>
<th>Read handle 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Char value at handle 4</td>
<td>Read handle 4</td>
<td></td>
</tr>
</tbody>
</table>

![Image of a heart rate monitor and Strava logo]

12 / 26
Bluetooth Low Energy application protocol

Attribute Protocol (ATT):
- Key-type-value store
- Transactional flow-control
- READ, WRITE, NOTIFY, FIND-BY-TYPE...

Generic Attribute Profile (GATT):
- Structure over attributes
- Characteristic: Groups attributes for a value
- Services: Groups related characteristics

Example, heart rate monitor

Find Char with UUID HRM
Char at handle 2
Read handle 2
Char value at handle 4
Read handle 4
157BPM
Beetle: Key insight

BLE application protocol (GATT) is amenable to multiplexing:

- Unified data model
- Standardized data types
- Transactions are meaningful to applications
Beetle Architecture

- Local apps
- Gateway
- Kernel Bluetooth Driver
- Virtual Devices
- Peripherals
Beetle Architecture
Beetle Architecture

Local apps

Gateway

Beetle

Kernel Bluetooth Driver

Peripherals
Beetle Architecture

Local apps

Gateway

Beetle

Kernel Bluetooth Driver

Peripherals

iForgotTheLights
Beetle Architecture

Local apps

Gateway

Beetle

Kernel Bluetooth Driver

Peripherals

iForgotTheLights
Virtual Devices: Just speak GATT

Look like a peripheral to applications.

Look like a gateway to peripherals.

Provide:

- Sharing by multiplexing transactions from different clients
- Access control by mapping handle space
- Flexible communication by allowing transactions over any link
Virtual Devices: Sharing
Virtual Devices: Sharing

- Strava
- HrView

READ REQ: 1
READ RESP: 1
READ REQ: 2
READ RESP: 2
Virtual Devices: Sharing

Strava

HrView

READ REQ: 1

READ RESP: 1

READ REQ: 2

READ RESP: 2
Virtual Devices: Sharing

READ REQ: 1

READ RESP: 1

READ REQ: 2

READ RESP: 2

Strava

HrView
Virtual Devices: Sharing

![Diagram showing data flow between virtual devices]

- **Strava**
  - READ REQ: 1
  - READ RESP: 1

- **HrView**
  - READ REQ: 2
  - READ RESP: 2
Virtual Devices: Sharing
Virtual Devices: Sharing

READ REQ: 2

READ RESP: 2

Strava

HrView
Virtual Devices: Sharing

READ REQ: 1
READ RESP: 1
READ REQ: 2
READ RESP: 2

Strava
HrView

READ RESP: 2
Virtual Devices: Sharing

![Diagram of virtual devices sharing data between Strava and HrView]

- **READ REQ: 1**
- **READ RESP: 1**
- **READ REQ: 2**
- **READ RESP: 2**

**Strava**

**HrView**

**Device:**

- Display showing heart rate of 146 BPM.
Virtual Devices: Access Control

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Glucose Char</td>
</tr>
<tr>
<td>3</td>
<td>Glucose value</td>
</tr>
<tr>
<td>6</td>
<td>Battery Char</td>
</tr>
<tr>
<td>7</td>
<td>Battery value</td>
</tr>
</tbody>
</table>
Virtual Devices: Access Control

BatteryMon

<table>
<thead>
<tr>
<th>6</th>
<th>Battery Char</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Battery value</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2</th>
<th>Glucose Char</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Glucose value</td>
</tr>
<tr>
<td>6</td>
<td>Battery Char</td>
</tr>
<tr>
<td>7</td>
<td>Battery value</td>
</tr>
</tbody>
</table>
Virtual Devices: Access Control

BatteryMon

READ REQ: 7

<table>
<thead>
<tr>
<th></th>
<th>Battery Char</th>
<th>Battery value</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2 Glucose Char
3 Glucose value
6 Battery Char
7 Battery value
Virtual Devices: Access Control

BatteryMon

<table>
<thead>
<tr>
<th></th>
<th>Battery Char</th>
<th>Battery value</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

READ REQ: 7

<table>
<thead>
<tr>
<th></th>
<th>Glucose Char</th>
<th>Glucose value</th>
<th>Battery Char</th>
<th>Battery value</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Virtual Devices: Access Control

BatteryMon

READ REQ: 2

6
Battery Char

7
Battery value

2
Glucose Char

3
Glucose value

6
Battery Char

7
Battery value
Virtual Devices: Access Control

BatteryMon

ERR

<table>
<thead>
<tr>
<th></th>
<th>6</th>
<th>Battery Char</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>7</td>
<td>Battery value</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>Glucose Char</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3</td>
<td>Glucose value</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>Battery Char</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>Battery value</td>
</tr>
</tbody>
</table>

READ REQ: 2
READ REQ: 7
READ REQ: 7
ERR 17 / 26
Implementation

Linux

- User-space daemon (C++)
- Applications connect via UNIX domain or TCP sockets
- Access control via database + browser interface

Android

- Modified Android’s Bluetooth Service app
- Applications connect over Intents (IPC)
- Access control via normal user control flow
Applications

- Multi-app Heart Rate Monitor
- Battery Monitor
- Generic Home Gateway
Performance

- One app to one peripheral – no significant overhead
  - 1.7x connection interval with/without Beetle
- Multi-client throughput depends on workload
  - Reads are linearly scalable via caching
- Peripheral-to-peripheral ~2-3x due to extra hop
  - Could be improved with better coordination by gateway
Limitations

- Bluetooth Low Energy specific
  - Relies on properties of the application layer protocol
- Peripherals must conform to GATT transactional semantics
  - But peripheral-specific virtual devices could mask violations
- Access control does not solve secure naming
- Does not offer a management solution
  - Complimentary to systems, like HomeOS
Safe and flexible peripheral sharing

Can and should be provided first-class by the gateway
Discussion

Safe and flexible peripheral sharing

Can and should be provided first-class by the gateway

Lessons for protocol design

- Data model at the right abstraction level
- Standardized data types
- Global namespace
Discussion

Safe and flexible peripheral sharing

Can and should be provided first-class by the gateway

Lessons for protocol design

▶ Data model at the right abstraction level
▶ Standardized data types
▶ Global namespace

Lessons for BLE peripheral builders

▶ Don’t rely on exclusive access for semantics
▶ One GATT transaction per application transaction
Discussion

Safe and flexible peripheral sharing

Can and should be provided first-class by the gateway

Lessons for protocol design

- Data model at the right abstraction level
- Standardized data types
- Global namespace

Lessons for BLE peripheral builders

- Don’t rely on exclusive access for semantics
- One GATT transaction per application transaction

https://github.com/helena-project/beetle
Light Bulb DRM: Philips Locks Purchasers Out Of Third-Party Bulbs With Firmware Update

from the Patch-notes:-'Adds-'buyer's-remorse' dept

The world of connected devices is upon us and things have never been better. Criminals can access your email account by breaking into your fridge. Your child's toys and your television record your conversations and send them to manufacturers' servers, where criminals are (again) able to access them. Your home thermostat goes HAL 9000 and attempts to set your house on fire. And, now, your light bulbs won't do the one thing you expect them to do: produce light.
Are corporations just evil?

Nest to brick Revolv smart hubs on Sunday, and there's nothing owners can do about it

If you own a Revolv smart hub, then Sunday is the day that Nest will pull the plug on it, and you can kiss your $300 gadget goodbye.

By Adrian Kingsley-Hughes for Hardware 2.0 | June 17, 2016 -- 19:00 GMT (12:00 PDT) | Topic: Google
Are corporations just evil?

GM says you don't own your car, you just license it
Attribute Caching
Attribute Caching

READ REQ: 1

app1

READ REQ: 1

app2

READ RESP: 1

READ RESP: 1

READ RESP: 1
Attribute Caching

READ REQ: 1

READ RESP: 1

app1

READ REQ: 1

READ RESP: 1

app2
Attribute Caching

READ REQ: 1
READ RESP: 1
READ RESP: 1
READ RESP: 1

app1
app2
Attribute Caching

READ REQ: 1
READ RESP: 1
READ RESP: 1
READ RESP: 1

app1

app2
Handling Subscriptions

Subscriptions:

∅

SUBSCRIBE app1

NOTIFY

NOTIFY

NOTIFY

SUBSCRIBE app2

NOTIFY

NOTIFY

UNSUBSCRIBE

UNSUBSCRIBE

UNSUBSCRIBE
Handling Subscriptions

Subscriptions: ∅

- SUBSCRIBE app1
- NOTIFY
- NOTIFY
- NOTIFY
- SUBSCRIBE app2
- NOTIFY
- NOTIFY
- NOTIFY
- UNSUBSCRIBE
- UNSUBSCRIBE
- UNSUBSCRIBE
Handling Subscriptions

Subscriptions:  \( \emptyset \)
Handling Subscriptions

Subscriptions: app1

- SUBSCRIBE
- NOTIFY
- NOTIFY
- NOTIFY

app1

- SUBSCRIBE
- NOTIFY
- NOTIFY

UNSUBSCRIBE

app2
Handling Subscriptions

Subscriptions: app1

app1

app2

SUBSCRIBE
Handling Subscriptions

Subscriptions: app1

app1

app2
Handling Subscriptions

Subscriptions: app1

NOTIFY

app2
Handling Subscriptions

Subscriptions: app1

app1

app2

NOTIFY

UNSUBSCRIBE

UNSUBSCRIBE
Handling Subscriptions

Subscriptions: app1

app1

app2

Subscriptions: ∅

SUBSCRIBE app1

SUBSCRIBE app2

NOTIFY

NOTIFY

NOTIFY

UNSUBSCRIBE

UNSUBSCRIBE

UNSUBSCRIBE
Subscriptions: app1

- SUBSCRIBE
- NOTIFY

- SUBSCRIBE
- NOTIFY
- NOTIFY

- UNSUBSCRIBE
- UNSUBSCRIBE
- UNSUBSCRIBE
Handling Subscriptions

Subscriptions: app1  app2
Handling Subscriptions

Subscriptions: app1  app2

NOTIFY

app1

app2
Handling Subscriptions

Subscriptions: app1 app2
Handling Subscriptions

Subscriptions: app1  app2
Handling Subscriptions

Subscriptions: app1  app2

app1

app2

UNSUBSCRIBE
Handling Subscriptions

Subscriptions: app1

app1

app2

UNSUBSCRIBE
Handling Subscriptions

Subscriptions: app1

Subscriptions: ∅
Handling Subscriptions

Subscriptions:  \( \emptyset \)
Handling Subscriptions

Subscriptions: $\emptyset$

app1

app2

UNSUBSCRIBE
Performance: Caching

99th %tile latency per request vs # concurrent requests

Without Caching

With Caching