

BLE (Bluetooth Low Energy) URI Scheme and Media Types

draft-bormann-t2trg-ble-uri-00

Carsten Bormann & Ari Keränen

T2TRG @ Lisbon

24th September, 2016

Background

- Bluetooth Low-Energy (BLE): popular radio technology for constrained devices
- Resources of BLE devices can be accessed over IP (RFC7668) or via gateways
- How about locally connected devices and web technologies?
- This draft: straw man proposal of BLE URI scheme and media types

BLE attributes, characteristics, etc.

- An **attribute** is a piece of labeled addressable data, handled by the Attribute Protocol
- A **characteristic** is a grouping of attributes, handled by the Generic Attribute Profile
- A **service** is a grouping of characteristics, handled by the Generic Attribute Profile
- A **profile** is a set of rules on how to setup/use services at the client

Example

- Passive scan for nodes:

```
GET ble:/gap/nodes/passive
```

- Results in node list (here with two nodes, second having GAP attributes):

```
[{  
  href: "ble:/b1nodes/42",  
  bdaddr: "Zm9vYmFy"  
}, {  
  href: "ble:/b1nodes/16",  
  bdaddr: "GfEgeD21",  
  ad: [ ["foo", "Zm8"],  
        ["bar", "Zg"] ]  
}]
```

server chooses



Example (contd.)

- GET for one node's services:

```
GET ble:/blnodes/16/services
```

- Results in service list (here two services):

```
[{
  href: "ble:/blnodes/16/s1",
  uuid: "fEgeZm9vGfEgYmyF"
},{
  href: "ble:/blnodes/16/s7",
  uuid: "gGhrdgE32FgaFfga"
}]
```

Example (contd.)

- GET for one service's characteristics:

```
GET ble:/blnodes/16/s7/characteristics
```

- Results in (here only one characteristic):

```
[{  
  href: "ble:/blnodes/16/c5",  
  uuid: "m9vGmyFffEgeZEgY",  
  properties: "K"  
}]
```

Example (contd.)

- GET for characteristic's value:

`GET b1e:/b1nodes/16/c5`

- Result: data

...?

Data types

- Many characteristics have simple data
 - conserving bits
 - uint8, uint16, utf8, ...
- Some simple bitmaps
- Some more complex
 - e.g., "body composition measurement" and "-feature" characteristics: 1, 3, and 4 –bit fields; several uints, time stamps, etc.

Questions

- Sensible approach to pursue?
 - "locally useful" URI scheme
- How do we get data in and out?
 - Bluetooth details needed by client?
 - Abstractions by BLE layer?
- (How to structure URIs)