OPEN CONNECTIVITY FOUNDATION™

RESTful Interaction

A viewpoint from OCF perspective

March 10, 2017

JinHyeock Choi

RESTful Architecture Style

Entity handler

Physical entity

Client

Server

REST Architecture Style

- Addressable resources
- A uniform, constrained interface
- Representation based manipulation
- Communicate statelessly
- Hypermedia State Engine

Request

Response

CRUD & N operation

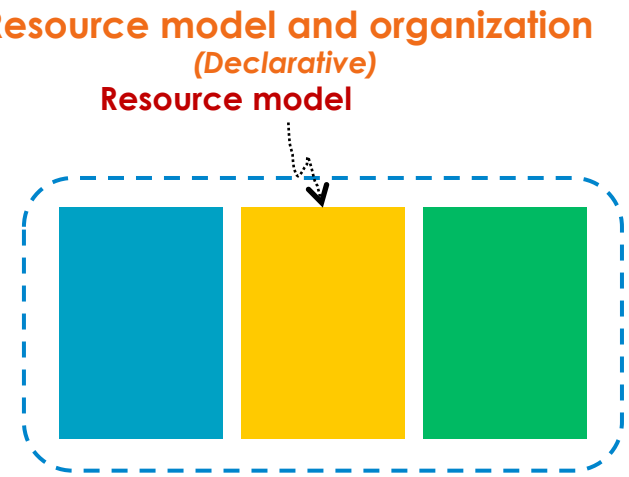
```
{
  "n": "myRoomTemperature",
  "rt": "oic.r.temperature",
  "if": "oic.if.a",
  "id": "example_id_xyz",
  "temperature": 23,
  "units": "C",
  "setValue": 25
}
```

Resource (representation)

Resource (representation)

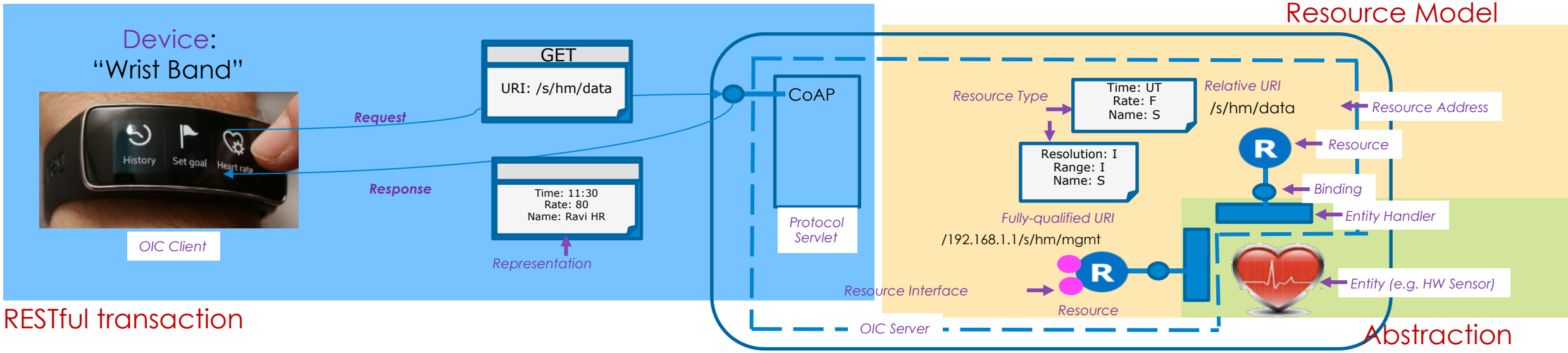
- RESTful Architecture (Representational State Transfer)
  - Resource based operation
    - Real world ‘entity’ is represented as ‘Resource’
  - Resource manipulation via Request/ Response: CRUDN

Logical organization: 3 part approach



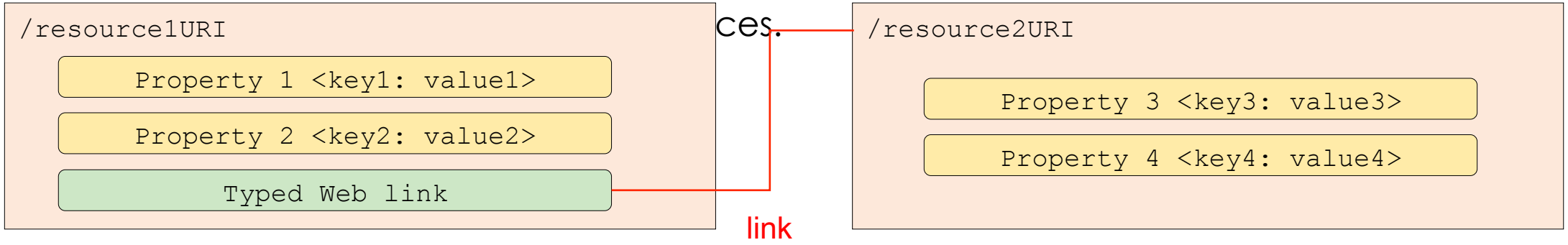
Communication and Interoperability
(Protocol processing & Messaging)
RESTful transaction

Device abstraction (entity handler)
(Imperative)
Abstraction



Resource (instance)

- Resource:
  - "Resource" means a specific "resource instance".
- Resource features: a resource is determined with the following features.
  - **URI**
    - In general, a resource can have any URI but some special resources have pre-defined URI.
  - **Property**
    - <Key: Value> pairs which characterize the resource.
  - **Typed web link**



Resource (instance)

- Resource:
  - "Resource" means a specific "resource instance".
- Resource features: a resource is determined with the following features.
  - **URI**
    - In general, a resource can have any URI but some special resources have pre-defined URI.
  - **Property**
    - <Key: Value> pairs which characterize the resource.
  - **Typed web link**
    - establish a relationship among resources.

/myLightSwitch

```
{
  "n": MyRoomLightSwitch",
  "rt": "oic.r.switch.binary",
  "if": "oic.if.a",
  "id": "b.switch_TF38_3",
  "value": "true"
}
```

Resource Type

Resource Type is a class or category of Resources & specified with
i) The table for Resource Type & ii) The table for associated properties

Ex) "Binary Switch" Resource Type Specification

fixed URI	Resource Type Title	Resource Type ID ("rt" value)	interfaces	Description	Related Functional Interaction	M/CM/O
none	Binary switch	oic.r.switch.binary	oic.if.a	Binary switch to turn on-off the device to which it's associated.	Actuation	O

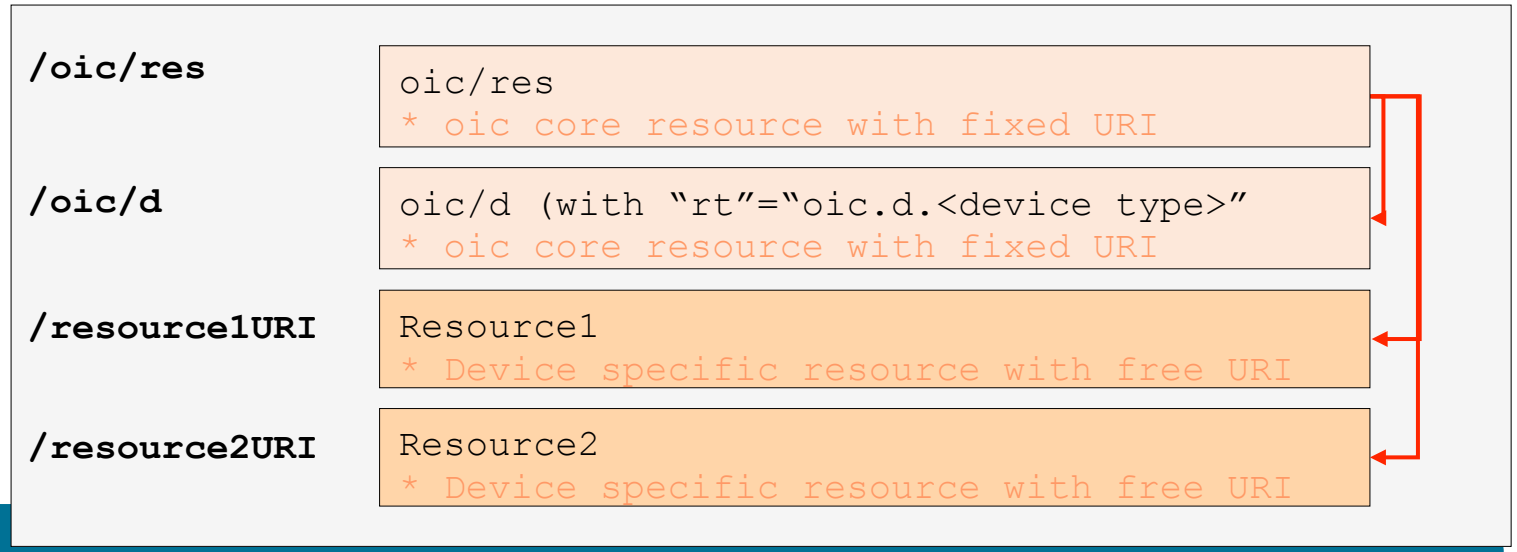
Binary switch properties

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Name	n	string			R	No	Human friendly name For example, "My Light Switch"
Resource Type ID	rt	string			R	yes	Represent a specific resource type.
Interface	if	string			R	Yes	Unique identifier for device (UUID)
Value	value	Boolean			RW	yes	Status of the switch

OIC device

- OIC device is specified
  - with i) core resources & ii) device specific resources.
    - Device type specified by “oic.d.<device type>”, (e.g., oic.d.light)
  - For example “light device (oic.d.light)” shall have mandatory
    - Core resource ① oic/res, ② oic/d (with fixed URI)
    - Device specific resource ③ binary switch (with free URI)

OIC device construct



Device example: light device (oic.d.light)

/oic/res

```
[{
  "di": "example_device_id",
  "links": [
    { "href": "/oic/d",
      "rt": "oic.d.light",
      "if": "oic.if.r",
      "rel": "hosts"},
    { "href": "/myLightSwitch",
      "rt": "oic.r.switch.binary",
      "if": "oic.if.a",
      "rel": "hosts"},
    { "href": "/myLightBrigtness",
      "rt": "oic.r.light.brightness",
      "if": "oic.if.a",
      "rel": "hosts"}
  ]
}]
```

/oic/d

```
{
  "n": "myRoomLightDevice",
  "rt": "oic.d.light",
  "if": "oic.if.r",
  "di": "example_device_id",
  "icv": "oic.1.5"
}
```

/myLightSwitch

```
{
  "n": MyRoomLightSwitch",
  "rt": "oic.r.switch.binary",
  "if": "oic.if.a",
  "id": "b.switch_TF38_3",
  "value": "true"
}
```

/myLightBrightness

```
{
  "n": MyRoomLightBrightness",
  "rt": "oic.r.light.brightness",
  "if": "oic.if.a",
  "id": "light_brightenss_TF38_3",
  "value": 30
}
```

Discovery procedure with oic/res

- Device discovery with “oic/res”
  - A response to “oic/res” carries device type in “rt” of “oic/d” in addition to all resource information.

GET /oic/res

RESPONSE

```
[{
  "di": "example_device_id",
  "links": [
    {"href": "/oic/d",
      "rt": "oic.d.light",
      "if": "oic.if.r",
      "rel": "hosts"},
    { "href": "/myLightSwitch",
      "rt": "oic.r.switch.binary",
      "if": "oic.if.a",
      "rel": "hosts"},
    { "href": "/myLightBrigtness",
      "rt": "oic.r.light.brightness",
      "if": "oic.if.a",
      "rel": "hosts"}
  ]
}]
```

/oic/res

```
[{
  "di": "example_device_id",
  "links": [
    {"href": "/oic/d",
      "rt": "oic.d.light",
      "if": "oic.if.r",
      "rel": "hosts"},
    { "href": "/myLightSwitch",
      "rt": "oic.r.switch.binary",
      "if": "oic.if.a",
      "rel": "hosts"},
    { "href": "/myLightBrigtness",
      "rt": "oic.r.light.brightness",
      "if": "oic.if.a",
      "rel": "hosts"}
  ]
]
```

/oic/d

```
{
  "n": "myRoomLightDevice",
  "rt": "oic.d.light",
  "if": "oic.if.r",
  "di": "example_device_id",
  "icv": "oic.1.5"
}
```

/myLightSwitch

```
{
  "n": MyRoomLightSwitch",
  "rt": "oic.r.switch.binary",
  "if": "oic.if.a",
  "id": "b.switch_TF38_3",
  "value": "true"
}
```

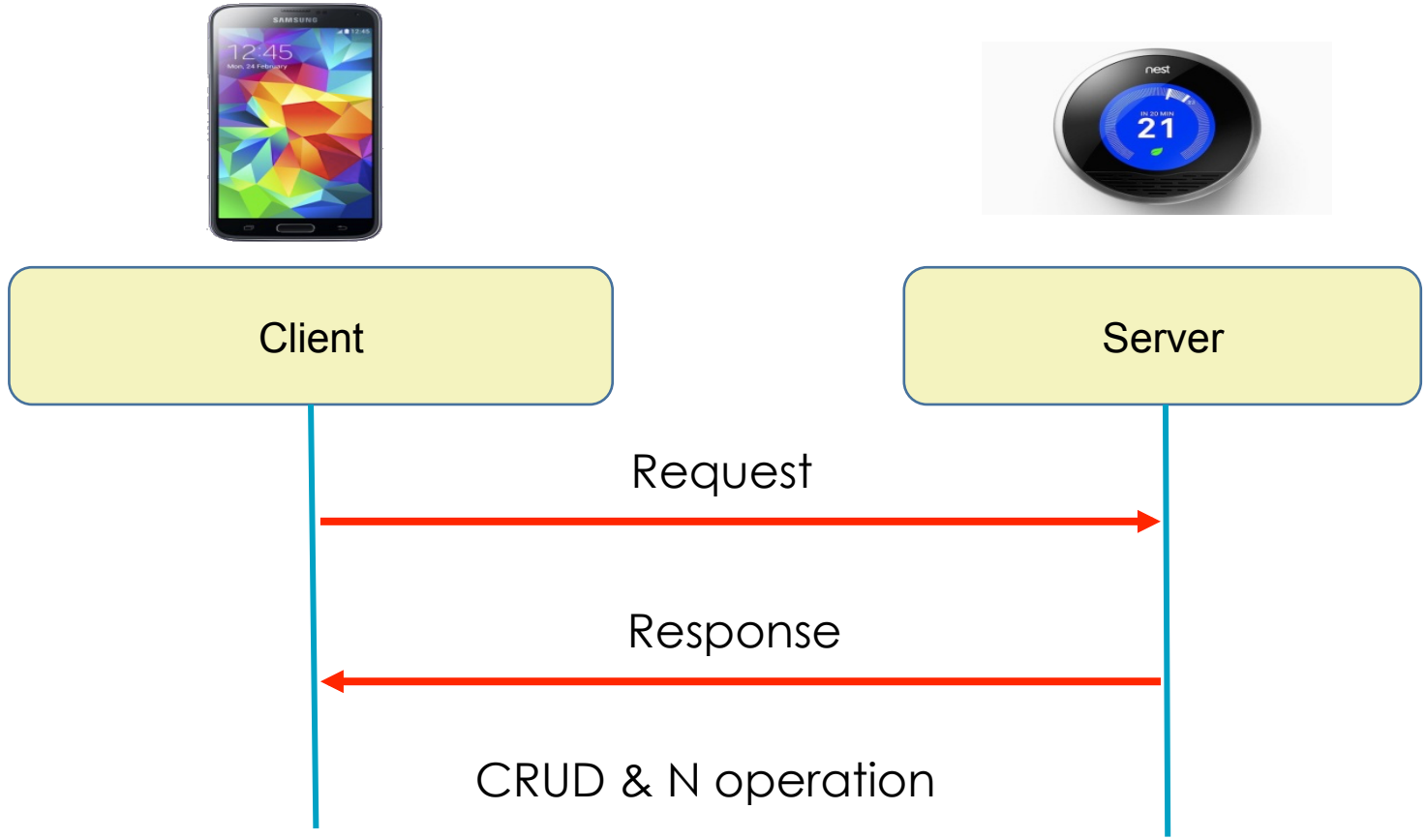
/myLightBrightness

```
{
  "n": MyRoomLightBrightness",
  "rt": "oic.r.light.brightness",
  "if": "oic.if.a",
  "id": "light_brightenss_TF38_3",
  "value": 30
}
```

CRUDN: generic operation procedure

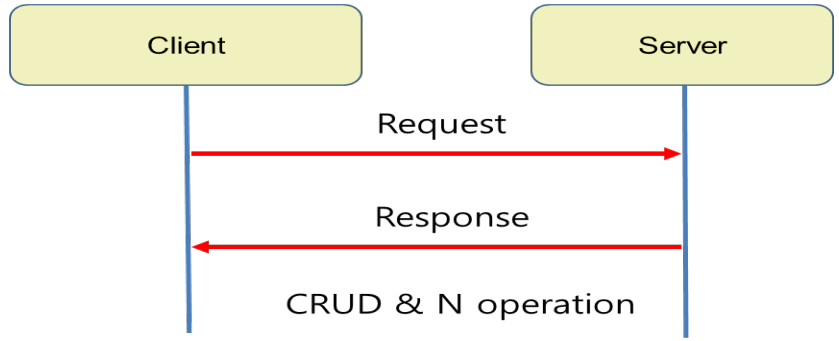
- CREATE
- RETRIEVE
- UPDATE
- DELETE
- NOTIFICATION
  - Subscribed notification
    - A client makes a priori subscription for a target resource of interest to the hosting server.
  - Unsubscribed notification
    - No a priori subscription for the target resource.
    - Ex) Fire alarm or VANET emergency

Generic communication flow scheme



Generic communication flow scheme

(tentative) Request message parameter/ Operation	
Operation (*op*)	operation to be executed, i.e. CRUD&N(?)
Subscription indication (observe opt?)	Request to notify upon resource state change
To (*to*)	The identifier (URI) of the target resource for the operation
From (*fr*)	the identifier of the message Originator in URI (ex) Device ID, Client ID or APP ID
Request Identifier (*ri*)	uniquely identifies a Request message
Content (*cn*)	contents (resource representation) to be transferred
Request expiration Timestamp (*ot*) (?)	Till when the message is valid (?)



Issue: Notification indication
- For CoAP, notification is indicated with observe option & CoAP code.

(tentative) Response message parameter/ Results	
Operation (*op*) (?)	operation to be executed, i.e. N
Response Code *(rs*)	successful, unsuccessful, Unsolicited N (?)
To (*to*)	the identifier of the message Originator, 'fr' in the matching request or multicast address in case of unsubscribed notification
From (*fr*)	The identifier of the target resource, 'to' in the matching request or the identifier or message originator in case of unsubscribed notification.
Request Identifier (*ri*)	uniquely identifies a Request message (or a priori fixed value)
Content (*cn*)	contents (resource representation) to be transferred
Response Expiration Timestamp (ot)?	when the request message expires

CRUDN & POST/PUT/GET/DELETE

No one-to-one correspondence between CRUDN & POST/PUT/GET/DELETE.

- CREATE is performed with POST or PUT
- UPDATE is performed with POST or PUT

- **CREATE with POST**: i) Existing target URI for the resource responsible for creation & ii) the URI of the new resource is determined by the server, forwarded to the client.
- **CREATE with PUT**: i) Non existing target URI for the new resource to be created.
- **UPDATE with POST**: i) Existing target URI for the resource to be updated & ii) partial modification with the payload.
- **UPDATE with PUT**: i) Existing target URI for the resource to be updated & ii) whole replacement with the payload.

UPDATE with POST

- **UPDATE with POST**: i) Existing target URI for the resource to be updated & ii) partial modification with the payload.

POST /existingURI

```
{
   "value": "False"
}
```

OIC server

/existingURI

```
{
  "n": MyRoomFoo",
  "rt": "oic.r.foo",
  "if": "oic.if.a",
  "value": "True"
}
```

UPDATE with POST

- **UPDATE with POST**: i) Existing target URI for the resource to be updated & ii) partial modification with the payload.

POST /existingURI

```
{
   "value": "False"
}
```

OIC server

/existingURI

```
{
  "n": MyRoomFoo",
  "rt": "oic.r.foo",
  "if": "oic.if.a",
  "value": "False"
}
```

UPDATE with PUT

- **UPDATE with PUT**: i) Existing target URI for the resource to be updated & ii) whole replacement with the payload.

PUT /existingURI

```
{
    "value": "False"
}
```

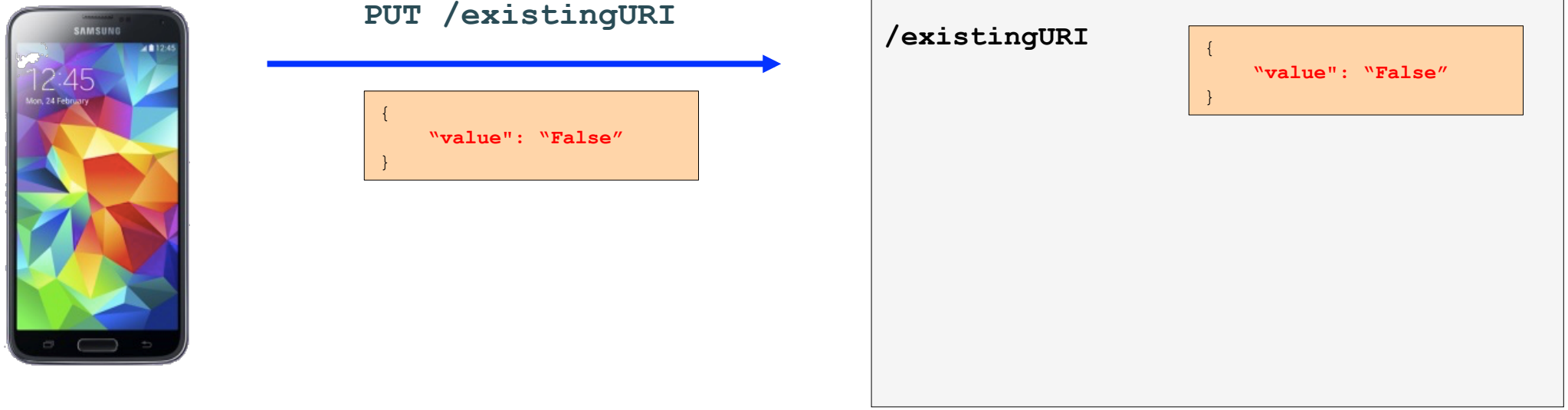
OIC server

/existingURI

```
{
  "n": MyRoomFoo",
  "rt": "oic.r.foo",
  "if": "oic.if.a",
  "value": "True"
}
```

UPDATE with PUT

- **UPDATE with PUT**: i) Existing target URI for the resource to be updated & ii) whole replacement with the payload.



Interface

- (Tentatively) Defines interaction constraints on a Resource
  - Which properties are exposed
  - Which methods are allowed – retrieve-only or updateable
  - How links are processed; oic.if.b
  - Application semantics –
    - for example oic.if.s and oic.if.a for sensors and actuators vs. oic.if.r and oic.if.rw for status and configuration parameters

- Specifies several interfaces
  - Further elaboration or clarification .

Query: “rt” & “if” query

- “if” query specifies to indicate the “interface” to use
  - POST /exampleCollectionURI?if=oic.if.b
- “rt” query specified as a filter

GET /collection

?rt=oic.r.switch.binary

REQUEST

RESPONSE

```
[
  {
    "href": "/BinarySwitchResURI",
    "rt": ["oic.r.switch.binary"],
    "if": ["oic.if.baseline","oic.if.a"],
    "p": {"bm": 1,"sec": false,"tcp": 57276}
  }
]
```

```
{
  "rt": ["oic.wk.col"],
  "if": ["oic.if.ll","oic.if.baseline", "oic.if.b"],
  "links": [
    {
      "href": "/BinarySwitchResURI",
      "rt": ["oic.r.switch.binary"],
      "if": ["oic.if.baseline","oic.if.a"],
      "p": {"bm": 1,"sec": false}
    },
    {
      "href": "/TemperatureResURI",
      "rt": ["oic.r.temperature"],
      "if": ["oic.if.baseline","oic.if.a"],
      "p": {"bm": 1,"sec": false}
    }
  ]
}
```

Further works?

- More minute Resource manipulation
  - Part of Resource RETRIVE or UPDATE
    - Long array update
    - PATCH would be of help
  - Group management
    - Dim down the light, Close the Window & Play the movie
    - Group/Scene/Script/Rule
  - Time dependent operation with delay or reservation(?)
    - Turn off the light 10 min later
    - Action & Actuation Resource(?)
    - Form may help?
- Standardized IETF/IRTF/WoT specification would be of help

OPEN CONNECTIVITY
FOUNDATION™