Self-descriptions for Interoperability and Security

Using WoT TD and MUD with RIOT

RIOT Summit 2021
Hugo Damer, Jan Romann, University of Bremen

About us

- Student computer science project at University of Bremen
- Supervised by Prof. Dr. Carsten Bormann, Prof. Dr. Ute Bormann and Dr. Olaf Bergmann
- Goal: Improve IoT security and usability
  - Return improvements to the open source community
- Results from the bachelor phase

Universität Bremen

NAMIB

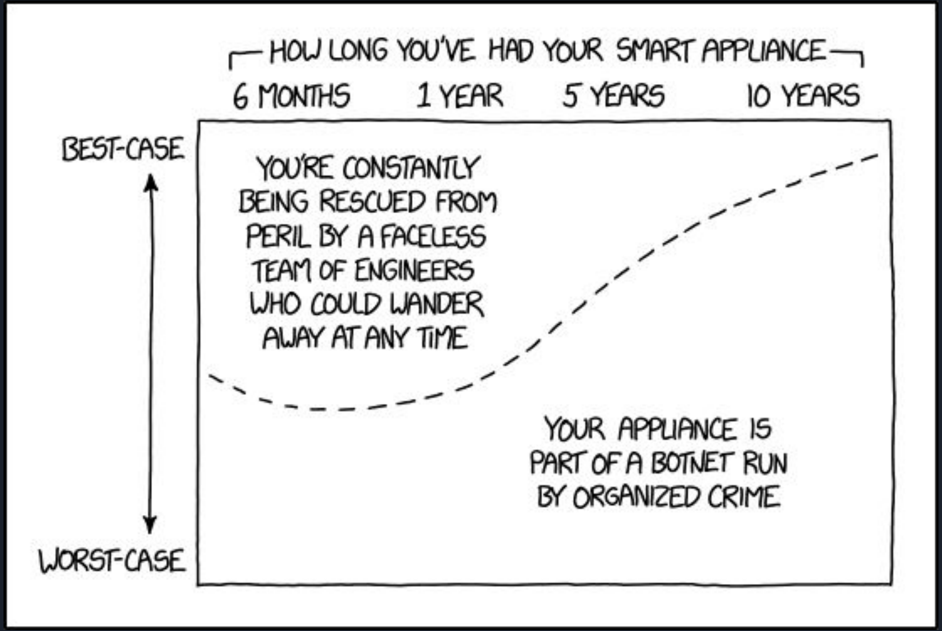
Network Access Makes IoT Better

Motivation

Two main problems:

Security and Interoperability

Problem 1: Security


HOW LONG YOU'VE HAD YOUR SMART APPLIANCE
6 MONTHS
1 YEAR
5 YEARS
10 YEARS
BEST-CASE
WORST-CASE
YOU'RE CONSTANTLY BEING RESCUED FROM PERIL BY A FACELESS TEAM OF ENGINEERS WHO COULD WANDER AWAY AT ANY TIME
YOUR APPLIANCE IS PART OF A BOTNET RUN BY ORGANIZED CRIME

Source: https://xkcd.com/1966/

Devices often are ...

... poorly secured

... untrustworthy

... receiving only limited support

Problem 2: Interoperability

Source: https://unsplash.com/photos/GbAEJUJKJ88

Devices are often ...

… only compatible with their own ecosystem

… using closed/proprietary standards

… not able to interact with each other

Enter: Self-descriptions

- Open standards
- Web of Things Thing Description (WoT TD)
  - W3C Recommendation
  - Describe what a device can do
  - Expose capabilities
- Manufacturer Usage Description (MUD)
  - RFC 8520
  - Describe what a device needs to fulfill its function
  - Reduce attack surface

Standards

WoT TD

- TD as “Entry Point” (cf. index.html)
- Serialized as JSON
- Consumers interact with Thing based on TD

See: Sebastian Kaebisch, Takuki Kamiya, Michael McCool, Victor Charpenay and Matthias Kovatsch, »Web of Things (WoT) Thing Description «, W3C Recommendation, Apr. 2020, URL: https://www.w3.org/TR/2020/REC-wot-thing-description-20200409/

TD Structure

```
{
  "@context": [ ... ],
  "title": "NAMIB Sensor",

  "securityDefinitions": { ... },
  "security": [ ... ]

  "properties": { ... },
  "actions": { ... },
  "events": { ... },
}
```

Meta-data

Security configuration

Interaction affordances

See: Sebastian Kaebisch, Takuki Kamiya, Michael McCool, Victor Charpenay and Matthias Kovatsch, »Web of Things (WoT) Thing Description«, W3C Recommendation, Apr. 2020, URL: https://www.w3.org/TR/2020/REC-wot-thing-description-20200409/

Interaction affordances

- Properties
- Actions
- Events

- Expose the capabilities of the Thing
- Mandatory “forms” field for protocol bindings
  - Map affordances to protocols and resources

See: Sebastian Kaebisch, Takuki Kamiya, Michael McCool, Victor Charpenay and Matthias Kovatsch, »Web of Things (WoT) Thing Description«, W3C Recommendation, Apr. 2020, URL: https://www.w3.org/TR/2020/REC-wot-thing-description-20200409/

Interaction affordances: Properties

```
{
  ...,
  "properties": {
    "temperature": {
      "type": "number",
      "unit": "C",
      "forms": [
        {
          "href": "coap://example.org/temperature”
        }
      ]
  },
  ...
}
```

Schema information

Protocol bindings

See: Sebastian Kaebisch, Takuki Kamiya, Michael McCool, Victor Charpenay and Matthias Kovatsch, »Web of Things (WoT) Thing Description«, W3C Recommendation, Apr. 2020, URL: https://www.w3.org/TR/2020/REC-wot-thing-description-20200409/

Actions

Meta data

```
{
  ...,
  "actions": {
    "setState": {
      "description": "Set the state of the lamp",
      "input": { ... },
      "forms": [
        {
          "href": "coap://example.org/toggle"
        }
  },
  ...
}
```

See: Sebastian Kaebisch, Takuki Kamiya, Michael McCool, Victor Charpenay and Matthias Kovatsch, »Web of Things (WoT) Thing Description«, W3C Recommendation, Apr. 2020, URL: https://www.w3.org/TR/2020/REC-wot-thing-description-20200409/

Obtaining TDs

- `/.well-known/wot-thing-description`
- `/.well-known/core`
- CoRE Link Format
  - Content-Type `application/td+json`

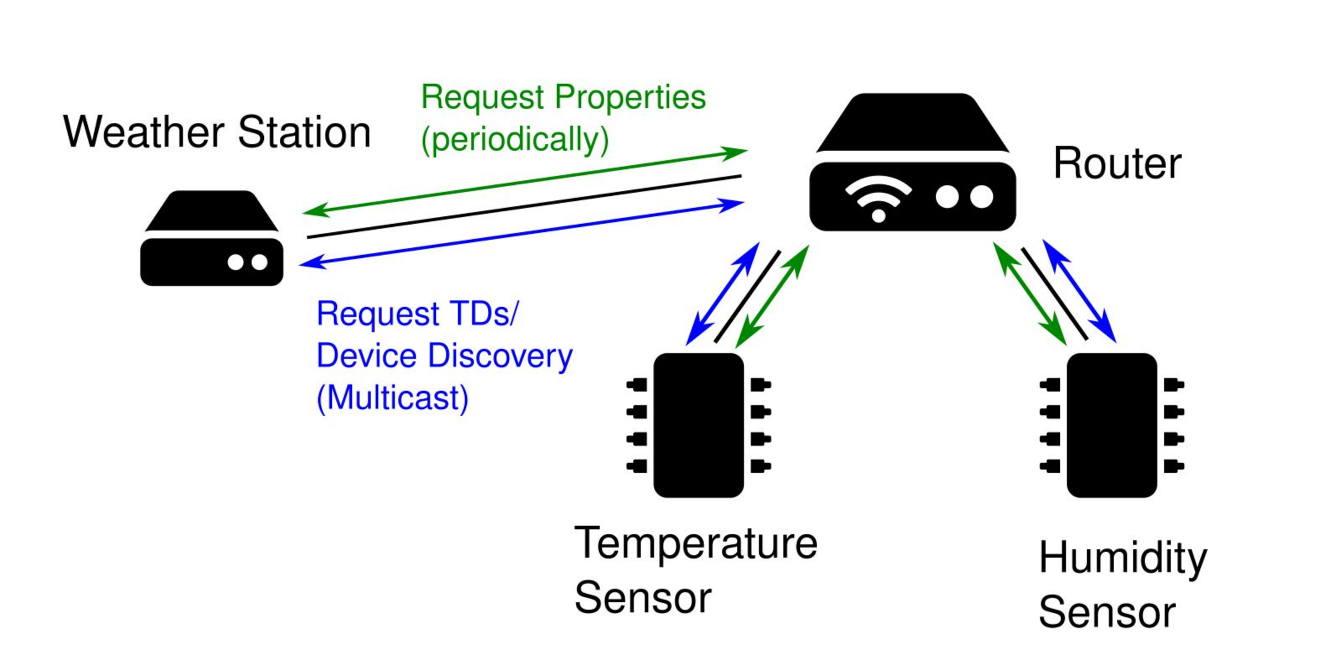
→ Enables multicast discovery

See: Andrea Cimmino, Michael McCool, Farshid Tavakolizadeh and Kunihiko Toumura, »Web of Things (WoT) Discovery«, W3C Working Draft, Jun. 2021, URL: https://www.w3.org/TR/2021/WD-wot-discovery-20210602/

WoT in RIOT

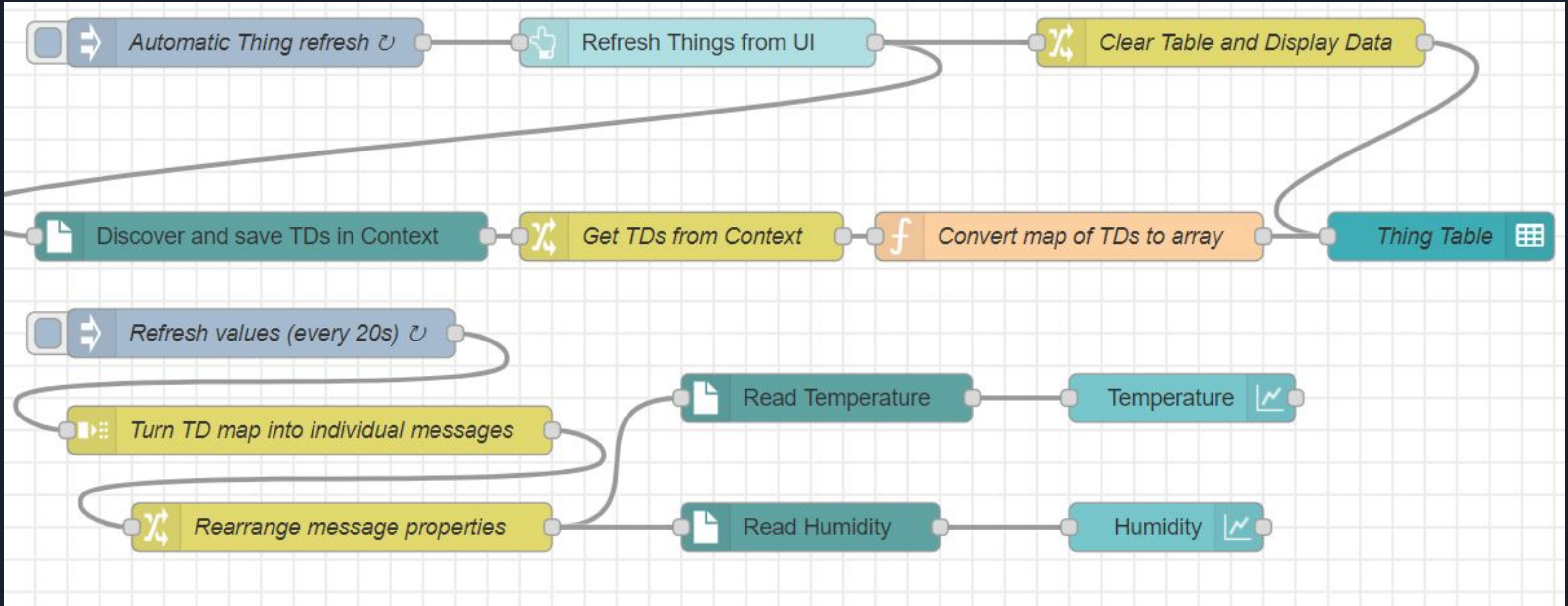
- Upcoming RIOT-Module (still WIP)
  - Collaboration with Philipp Blum (@Citrullin)
- Serialization of TD as JSON at runtime
- CoAP support using Gcoap
- Code generation at compile time

Example Setup

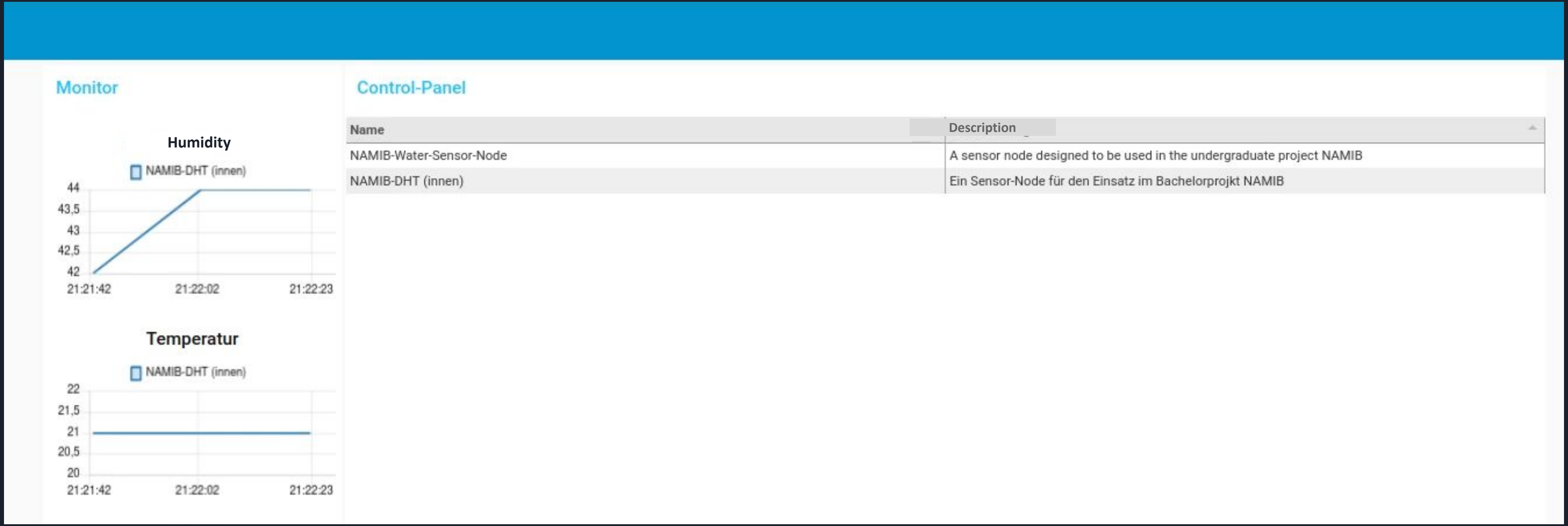


Icon Packs used: Font Awesome (https://github.com/FortAwesome/Font-Awesome , License: CC BY 4.0), Gnome Symbolic Icons (https://gitlab.gnome.org/Archive/gnome-icon-theme-symbolic/-/tree/master, License: CC-BY-SA 3.0)

Weatherstation Logic


Automatic Thing refresh ↻
Refresh Things from UI
Clear Table and Display Data
Discover and save TDs in Context
Get TDs from Context
Convert map of TDs to array
Thing Table
Refresh values (every 20s) ↻
Turn TD map into individual messages
Read Temperature
Temperature
Rearrange message properties
Read Humidity
Humidity

Weather Station UI



Monitor

Control-Panel

Name	Description
NAMIB-Water-Sensor-Node	A sensor node designed to be used in the undergraduate project NAMIB
NAMIB-DHT (innen)	Ein Sensor-Node für den Einsatz im Bachelorprojkt NAMIB

Manufacturer Usage Description (MUD)

- Specified in RFC 8520
- Manufacturer-provided device description of which network access is necessary
- Serialised as a JSON file (MUD-File) following a YANG data model
- Mainly consists of Access Control Lists (ACLs) aside from meta information

See: Eliot Lear, Ralph Droms and Dan Romascanu, »Manufacturer Usage Description Specification«, IETF, RFC 8520, Mar. 2019, DOI: 10.17487/RFC8520,URL: https://rfc-editor.org/rfc/rfc8520.txt

MUD File Example

```
{
  "ietf-mud:mud": {
    ...
  },
  "ietf-access-control-list:acls": {
    ...
  }
}
```

Example file taken from Section 9 of: Eliot Lear, Ralph Droms and Dan Romascanu, »Manufacturer Usage Description Specification«, IETF, RFC 8520, Mar. 2019, DOI: 10.17487/RFC8520,URL: https://rfc-editor.org/rfc/rfc8520.txt

MUD File Example - Metadata

```
{
  "ietf-mud:mud": {
    "mud-version": 1,
    "mud-url": "https://lighting.example.com/lightbulb2000",
    "last-update": "2019-01-28T11:20:51+01:00",
    "cache-validity": 48,
    "is-supported": true,
    "systeminfo": "The BMS Example Light Bulb",
    ...
  },
  ...
}
```

Example file taken from Section 9 of: Eliot Lear, Ralph Droms and Dan Romascanu, »Manufacturer Usage Description Specification«, IETF, RFC 8520, Mar. 2019, DOI: 10.17487/RFC8520,URL: https://rfc-editor.org/rfc/rfc8520.txt

MUD File Example - ACL

```
{
  ...
  "ietf-access-control-list:acls": {
    "acl": [
     {
        "name": "mud-76100-v6to",
        "type": "ipv6-acl-type",
        "aces": {
          "ace": [ … ]
        }
     }
    ]
  }
}
```

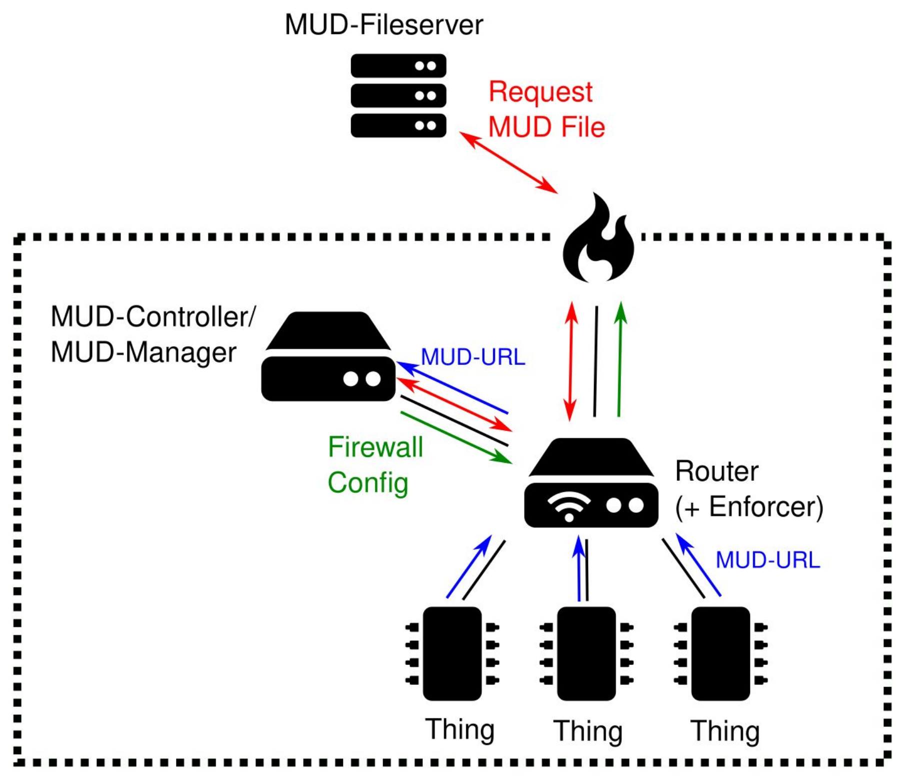
Example file taken from Section 9 of: Eliot Lear, Ralph Droms and Dan Romascanu, »Manufacturer Usage Description Specification«, IETF, RFC 8520, Mar. 2019, DOI: 10.17487/RFC8520,URL: https://rfc-editor.org/rfc/rfc8520.txt

MUD File Example - ACE

```
{
  ...
        "aces": {
          "ace": [
            {
              "name": "cl0-todev",
              "matches": {
                "ipv6": { "ietf-acldns:src-dnsname": "test.example.com", "protocol": 6 },
                "tcp": {
                  "ietf-mud:direction-initiated": "from-device",
                  "source-port": { "operator": "eq", "port": 443 }
                }
              },
              "actions": { "forwarding": "accept" }
            }
          ]
        }
  ...
}
```

Example file taken from Section 9 of: Eliot Lear, Ralph Droms and Dan Romascanu, »Manufacturer Usage Description Specification«, IETF, RFC 8520, Mar. 2019, DOI: 10.17487/RFC8520,URL: https://rfc-editor.org/rfc/rfc8520.txt

MUD Architecture

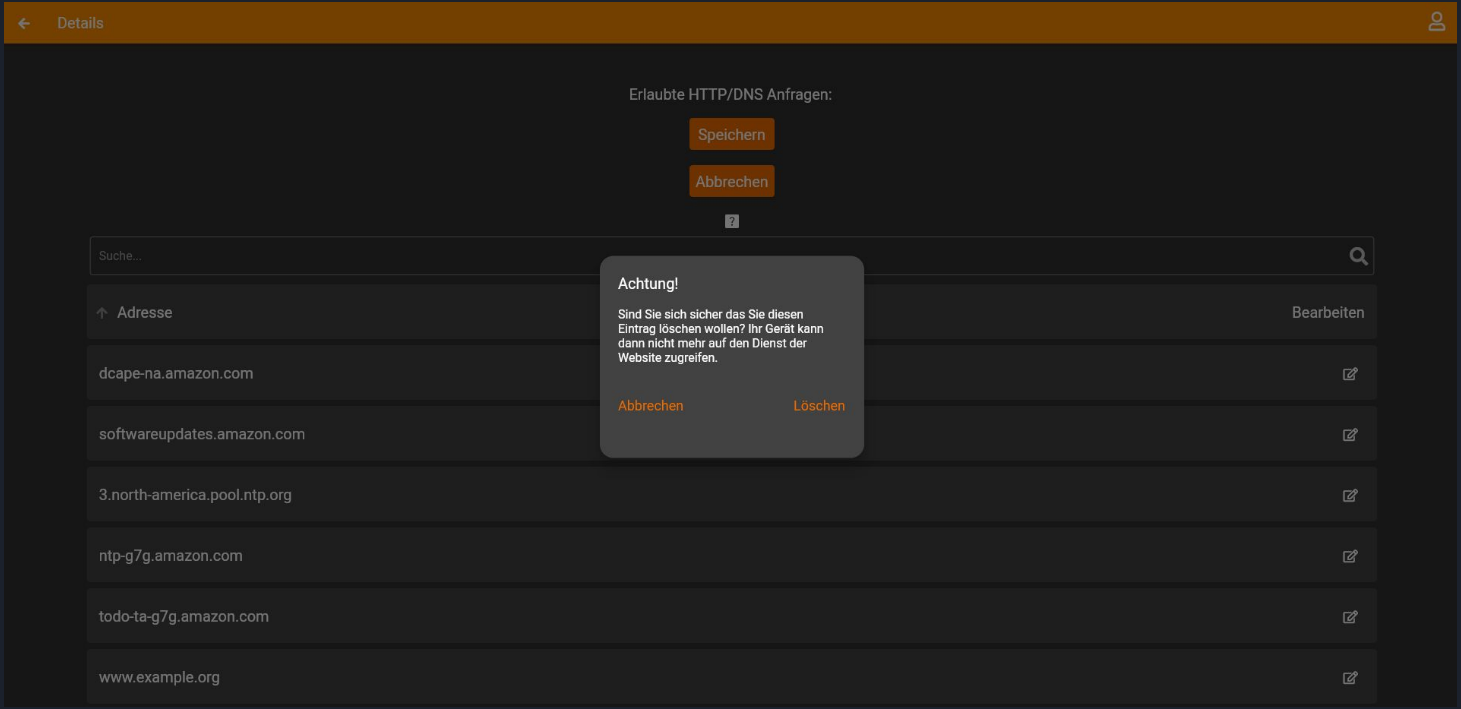


See Section 1.8 of: Eliot Lear, Ralph Droms and Dan Romascanu, »Manufacturer Usage Description Specification«, IETF, RFC 8520, Mar. 2019, DOI: 10.17487/RFC8520,URL: https://rfc-editor.org/rfc/rfc8520.txt
Icon Packs used: Octicons (https://github.com/primer/octicons, License: MIT), Font Awesome (https://github.com/FortAwesome/Font-Awesome , License: CC BY 4.0), Gnome Symbolic Icons (https://gitlab.gnome.org/Archive/gnome-icon-theme-symbolic/-/tree/master, License: CC-BY-SA 3.0)

MUD Controller

- We wrote our own MUD manager implementation
- Consists of…
  - ...Controller
    - runs on e.g. Raspberry Pis
  - ...Enforcer
    - runs on home routers (OpenWRT)

MUD Controller

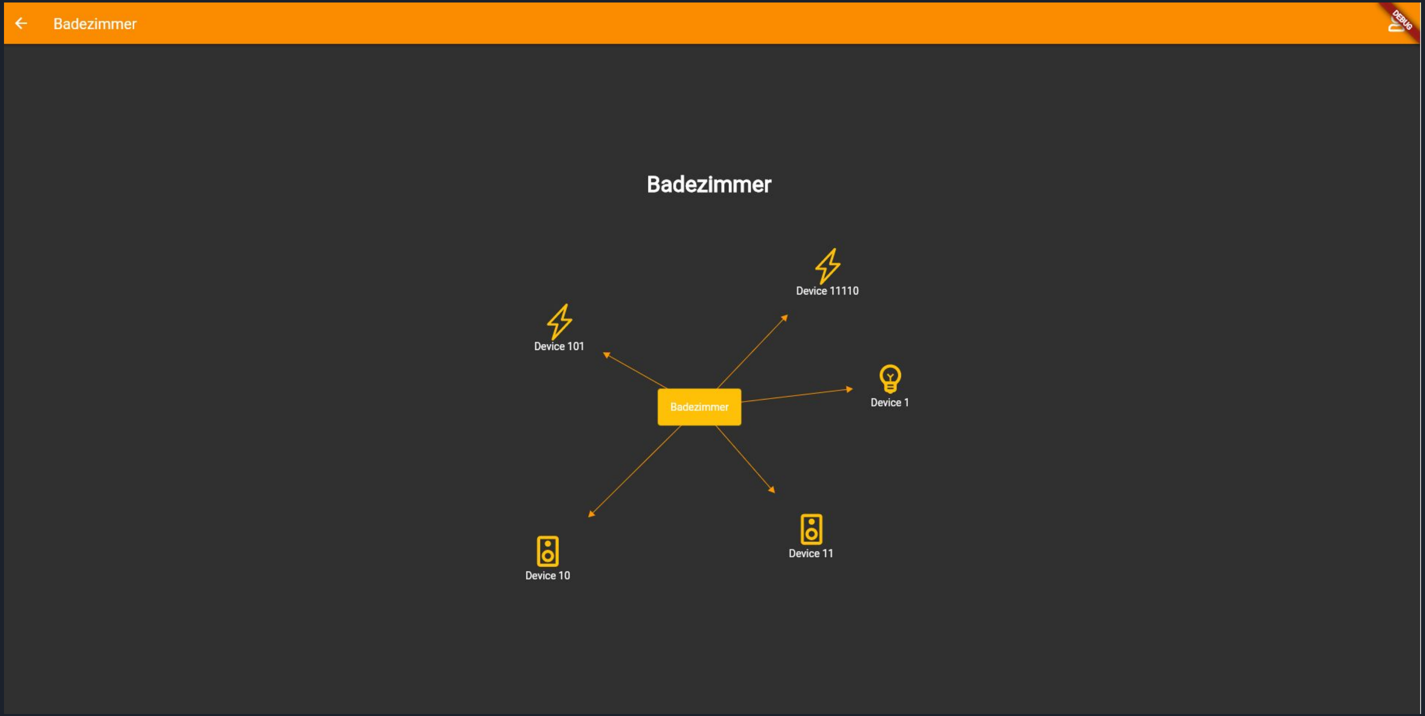


Adresse	Bearbeiten
dcape-na.amazon.com	
softwareupdates.amazon.com	
3.north-america.pool.ntp.org	
ntp-g7g.amazon.com	
todo-ta-g7g.amazon.com	
www.example.org	

Achtung!

Sind Sie sich sicher das Sie diesen Eintrag löschen wollen? Ihr Gerät kann dann nicht mehr auf den Dienst der Website zugreifen.

MUD Controller



MUD Legacy-Service

How to deal with devices that do not support MUD?

- Legacy-Service component
  - Can analyse network traffic of non-MUD devices
  - MUD recommendations based on device behavior
  - Based on crowdsourced repository of MUD files
    - https://gitlab.freedesktop.org/sw0rd/MUD-Files

MUD in RIOT

RIOT can already communicate its own MUD-URL!

- Implementation of DHCPv6 MUD option
- Client can request non-temporary addresses
  - MUD-Controller works with leases
- Open PR for stateless DHCPv6

How does the RIOT device know its own MUD-URL?

- Configuration of MUD URL via Makefile or KConfig

Insights

Insights: WoT

- Discovering and using TDs works
- However: No support for composition yet
- JSON not ideal for constrained devices → CBOR

Insights: MUD

- DHCP not ideal for emitting MUD URLs in the IoT
  - SLAAC makes DHCPv6 less relevant
  - Possible alternative: CoRE resource directories
- Keeping track of IP addresses for DNS entries is difficult
  - Especially when dealing with load balancers
- Isolating devices in home networks is difficult
  - VLANs not supported by hardware

Conclusion

- Self-descriptions are very promising
- Can make the IoT more secure and easier to use
- However:
  - Some weaknesses
  - Constrained devices require a bit more effort

→ There is still a lot to do

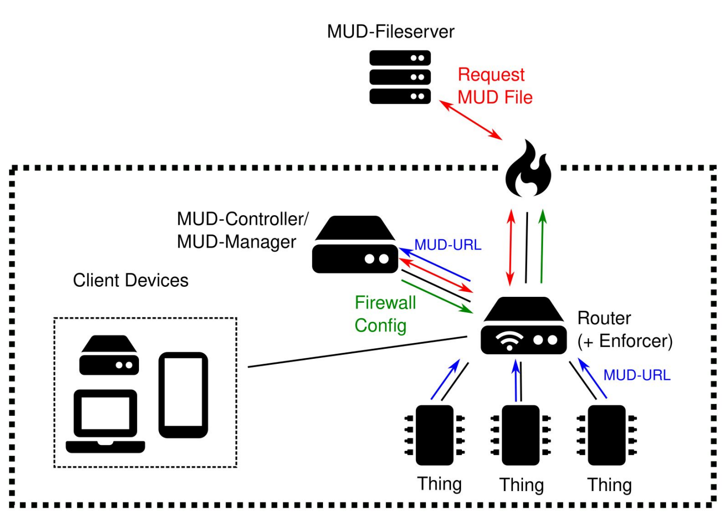
Fork Us on Github

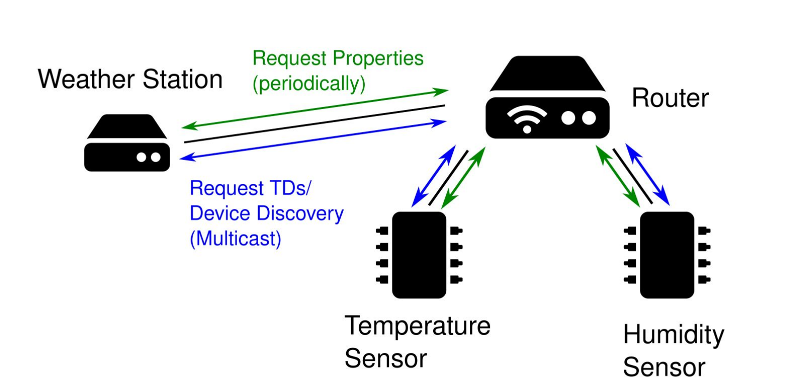
NAMIB

Network Access Makes IoT Better

- https://github.com/namib-project

We're looking forward to your feedback and contributions!





Icon Packs used: Octicons (https://github.com/primer/octicons, License: MIT), Font Awesome (https://github.com/FortAwesome/Font-Awesome , License: CC BY 4.0), Gnome Symbolic Icons (https://gitlab.gnome.org/Archive/gnome-icon-theme-symbolic/-/tree/master, License: CC-BY-SA 3.0)