

txThings – CoAP library based on Twisted or asyncio

Florian Schweikert, Christian Amsüss

2014-06-04

What does that mean?

- ▶ CoAP library
- ▶ written in Python
- ▶ based on Twisted *or*
- ▶ based on asyncio

Who we are

- ▶ Christian Amsüss
 - ▶ c.amsuess@energyharvesting.at
 - ▶ writes asyncio port for
Energy Harvesting Solutions GmbH
-
- ▶ Florian Schweikert
 - ▶ kelvan@ist-total.org
 - ▶ member of OSDomotics
uses txThings to connect IoT to classic internet

Python

- ▶ dynamic
- ▶ easy & fast development
- ▶ powerful
- ▶ many libs



Python

import antigravity

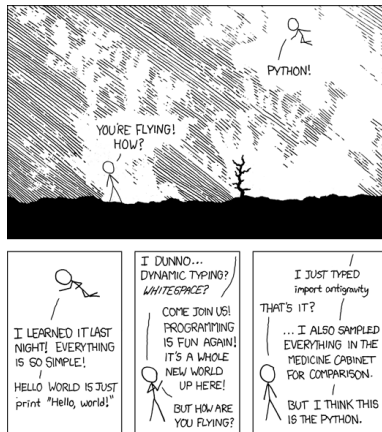


Figure: <http://xkcd.com/353/>

Introduction

CoAP in n lines

Some implementation details

Backends, Future

Appendices

Prerequisites

for the twisted version

- ▶ CoAP-18 nodes (13 should work too)
- ▶ IP(v6) connection to the nodes
- ▶ python2
- ▶ twisted
- ▶ txThings

Accessing a resource in 11 lines

```
endpoint = resource.Endpoint(None)
protocol = coap.Coap(endpoint)
reactor.callLater(0, start)
reactor.listenUDP(61616, protocol,
                 interface='::')

reactor.run()
```


Accessing a resource in 11 lines

```
def start(protocol):
    request = coap.Message(code=coap.GET)
    request.opt.uri_path = ('sensors', 'battery')
    request.remote = ("2001:db8::42",
                     coap.COAP_PORT)
    #request.setObserve(observeCallback)
    d = protocol.request(request)
    d.addCallback(printResponse)
    #d.addErrback(noResponse)
```

Clock server in 12 lines

```
class TimeResource(resource.CoAPResource):
    def __init__(self):
        resource.CoAPResource.__init__(self)
        self.visible = True
        self.observable = True

        self.notify()
```

Clock server in 12 lines

```
def notify(self):
    self.updatedState()
    reactor.callLater(60, self.notify)

def render_GET(self, request):
    response = coap.Message(
        code=coap.CONTENT,
        payload=str(datetime.now()))
    return defer.succeed(response)
```

What's possible?

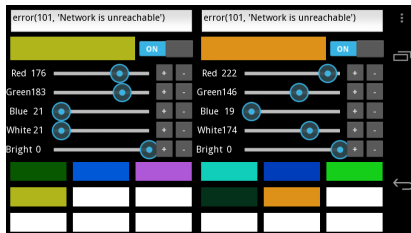
- ▶ everything

What's possible?

- ▶ everything
- ▶ ...
- ▶ WakeOnLan
- ▶ weather forecast
- ▶ departures of public transport
- ▶ calendar
- ▶ new mail counter

GUI/kivy

- ▶ kivy
- ▶ Possibility to build APK
- ▶ should work with QT, etc



Introduction

CoAP in n lines

Some implementation details

Backends, Future

Appendices

Options

```
def render_GET(self, request):  
    if self.etag in request.opt.etags:  
        result = Message(code=VALID)  
    result.opt.etag = self.etag  
    # result.payload = ''  
    return result
```


Blockwise transfer

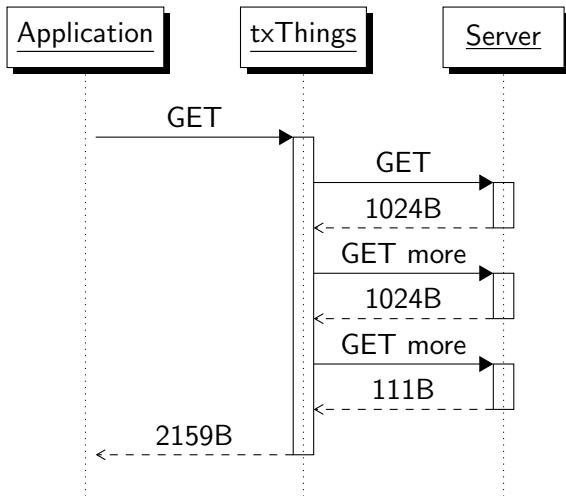


Figure: Abstraction of blockwise transfer

Piggy-backed responses

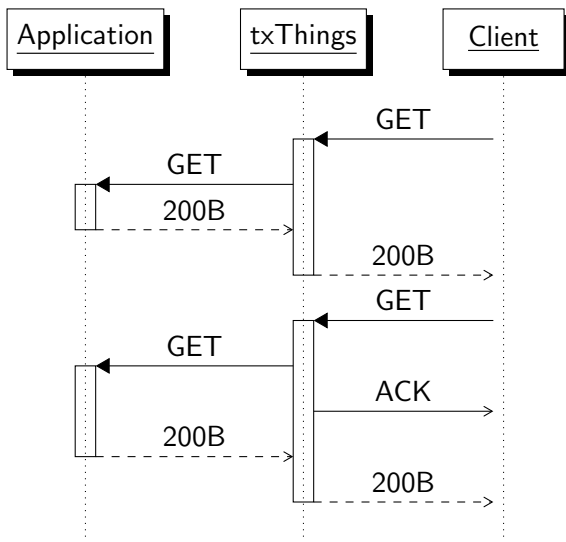


Figure: Piggy-backed vs. separate responses

Introduction

CoAP in n lines

Some implementation details

Backends, Future

Appendices

txThings on twisted

- ▶ Python 2
- ▶ GUIs with kivy
- ▶ easy to integrate in existing twisted servers

txThings on asyncio

- ▶ Python 3
- ▶ only depends on standard library
- ▶ GUIs with GTK

but

- ▶ not completely ported yet

Future of txThings

YOU

Introduction

CoAP in n lines

Some implementation details

Backends, Future

Appendices

Links

- ▶ txThings code
<https://github.com/siskin/txThings>
- ▶ asyncio branch code
<https://github.com/chrysn/txThings>
- ▶ asyncio branch documentation
<http://txthings-asyncio.readthedocs.org/>
- ▶ Event information about the talk
<http://www.meetup.com/IoT-Vienna/events/183209612/>
- ▶ kivy resources
<http://kivy.org/>
<http://kivy.org/docs/guide/android.html>
- ▶ OSDomotics examples
<http://osdwiki.open-entry.com/doku.php/en:projekte:coaponpc#txthings>

Sources, Credits, License

Slides source

embedded in PDF, extract with your viewer or
`pdftk unpack_files *.pdf`

Copyright

Slides: 2014 Florian Schweikert, Christian Amsüss, [CC-BY-SA-3.0](#)

Python logo: from [Python website](#), see [PSF Trademark Policy](#)

xkcd comic: Randall Munroe, [CC-BY-NC-2.5](#)

txThings: 2012-2014 Maciej Wasilak

<http://sixpinetrees.blogspot.com/>