Outsmarting your Smart Meter

UtrechtJUG // October 27, 2017
Maarten Mulders

Maarten Mulders

@mthmulders

architect | lead software engineer | trainer | speaker

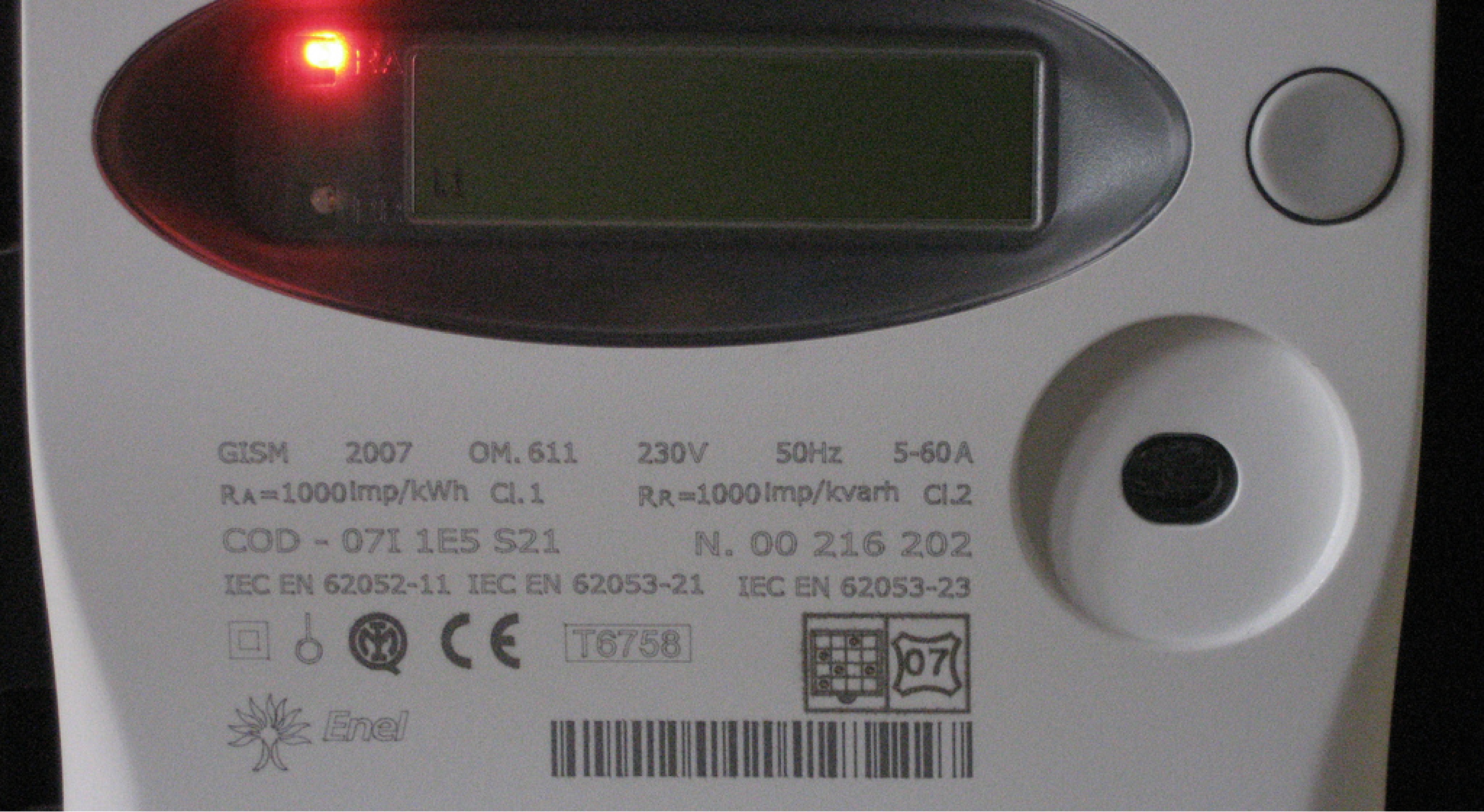
InfoSupport

Solid Innovator

What's Up?

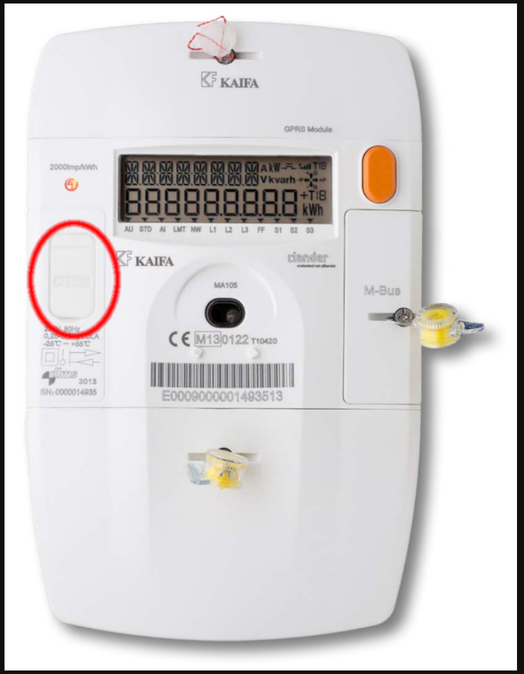
- Background
- Connecting the Smart Meter
- Building a Dashboard
- Questions

kWh
3104572
19617
EA
30
72
Drehstromzähler
Form C14U
Nr 28855257
3x220/380V
10(60)A
50 H
96 U/kWh
Schltg. 400
197


RA
RR
L1
GISM 2007 OM. 611 230V 50Hz 5-60A
RA=1000imp/kWh Cl.1 RR=1000imp/kvarh Cl.2
COD - 07I 1E5 S21 N. 00 216 202
IEC EN 62052-11 IEC EN 62053-21 IEC EN 62053-23
T6758
07
Enel

Connecting the Smart Meter

Pin #	Signal name	Description
1	+5 V	Power supply
2	Request	Input
3	Data GND	Ground
4	not connected	
5	Data	Output
6	Power GND	Power supply



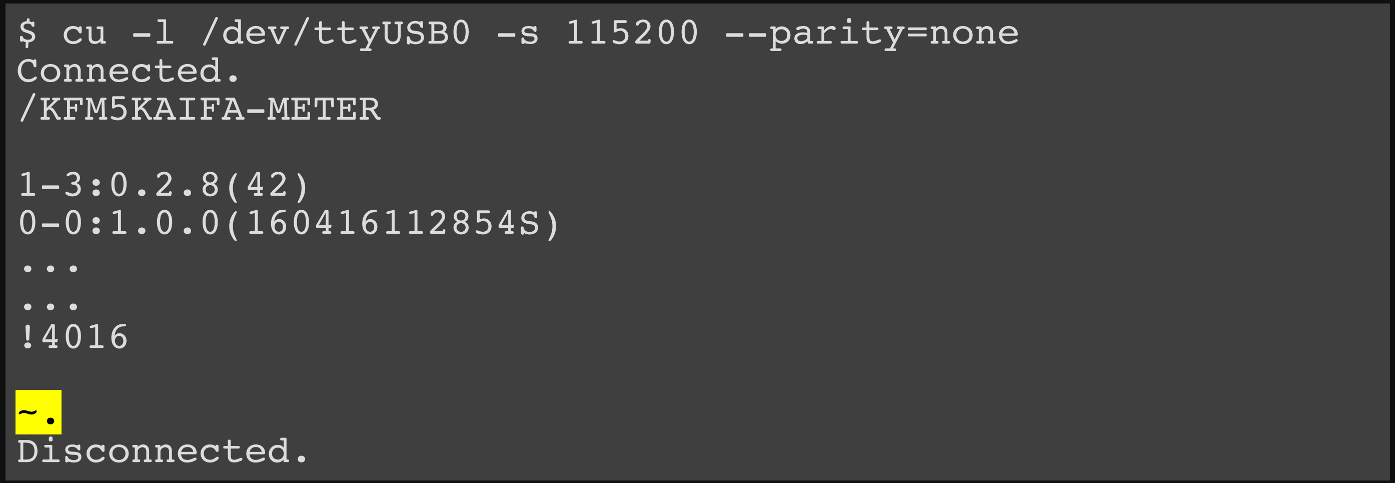
Reading data

eBay shopping list

1. RJ11-to USB female
2. USB male-to-male

Total: $ 4.08 (€ 3.50) **incl. shipping**

Reading data

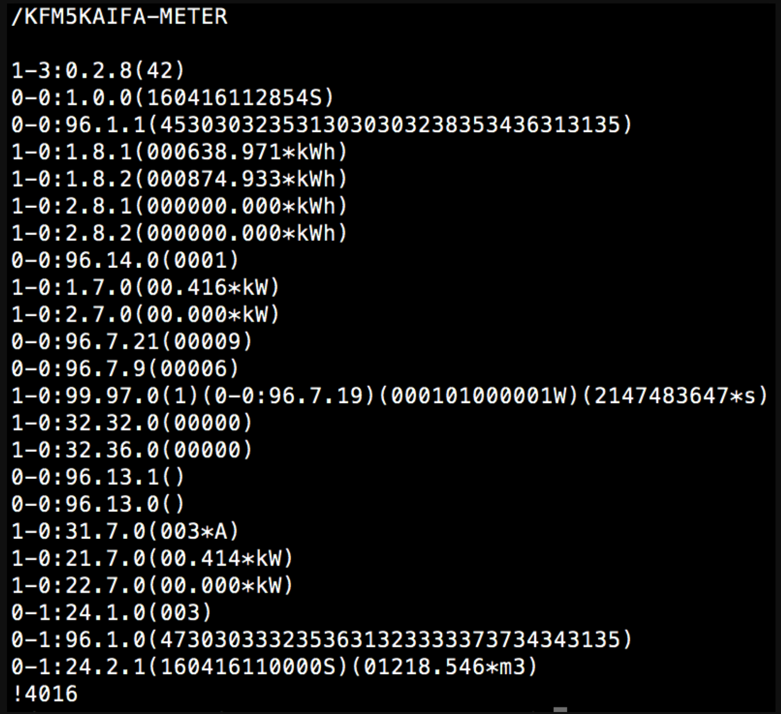


```
$ cu -l /dev/ttyUSB0 -s 115200 --parity=none
Connected.
/KFM5KAIFA-METER

1-3:0.2.8(42)
0-0:1.0.0(160416112854S)
...
...
!4016

~.
Disconnected.
```

Interpreting data



```
/KFM5KAIFA-METER

1-3:0.2.8(42)
0-0:1.0.0(160416112854S)
0-0:96.1.1(4530303235313030303238353436313135)
1-0:1.8.1(000638.971*kWh)
1-0:1.8.2(000874.933*kWh)
1-0:2.8.1(000000.000*kWh)
1-0:2.8.2(000000.000*kWh)
0-0:96.14.0(0001)
1-0:1.7.0(00.416*kW)
1-0:2.7.0(00.000*kW)
0-0:96.7.21(00009)
0-0:96.7.9(00006)
1-0:99.97.0(1)(0-0:96.7.19)(000101000001W)(2147483647*s)
1-0:32.32.0(00000)
1-0:32.36.0(00000)
0-0:96.13.1()
0-0:96.13.0()
1-0:31.7.0(003*A)
1-0:21.7.0(00.414*kW)
1-0:22.7.0(00.000*kW)
0-1:24.1.0(003)
0-1:96.1.0(4730303332353631323333373734343135)
0-1:24.2.1(160416110000S)(01218.546*m3)
!4016
```

```
/???5<identification>

<data> (repeated)
!<CRC>
```

identification: vendor-specific, not specified

data:

```
OBIS-reference(value)
```

CRC16-checksum over `/` to `!`

Parsing data

This is easily parsed with Scala's **parser** combinators

Input:

```
000635.311
```

Parser:

```
def number = """\d*\.?\d*""".r ^^ { BigDecimal(_) }
```

Parsing data (ctd)

This is easily parsed with Scala's parser **combinators**

Input:

```
1-0:1.8.1(000635.311*kWh)
```

Parser:

```scala
def elecCons1 = "1-0:1.8.1(" ~>
  """\d*\.?\d*""".r <~ "*kWh)" ^^ { BigDecimal(_) }
```

Parsing data (ctd)

This is easily parsed with Scala's **parser combinators**

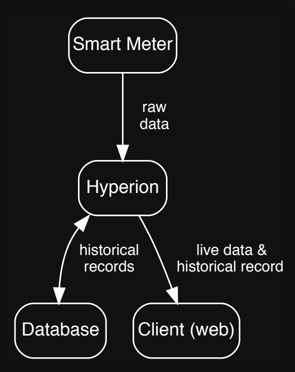
```
def make                 = "/" ~> "[A-Za-z0-9]{3}".r ^^ { String(_) }
def identification       = "5" ~> ".*".r             ^^ { String(_) }
def header = make ~ identification ^^ {
  case make ~ identification => P1Header(make, identification)
}
```

Parsing data (ctd)

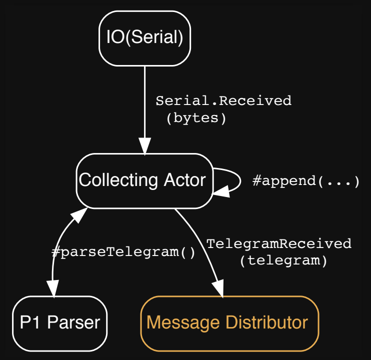
This is easily parsed with Scala's **parser combinators**

```scala
private def telegram = header ~ metadata ~ data ~ checksum
private def parser: Parser[P1Telegram] = telegram ^^ {
  case header ~ metadata ~ data ~ checksum =>
    P1Telegram(header, metadata, data, checksum)
} | failure("Not all required lines are found")
```

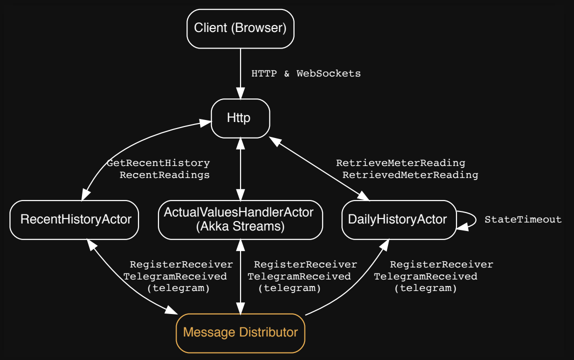
Architecture



Hyperion (1)


IO(Serial)
Serial.Received
(bytes)
Collecting Actor
#append(...)
#parseTelegram()
TelegramReceived
(telegram)
P1 Parser
Message Distributor

Hyperion (2)



Client (Browser)

HTTP & WebSockets

Http

GetRecentHistory
RecentReadings

RetrieveMeterReading
RetrievedMeterReading

RecentHistoryActor

ActualValuesHandlerActor
(Akka Streams)

DailyHistoryActor

StateTimeout

RegisterReceiver
TelegramReceived
(telegram)

RegisterReceiver
TelegramReceived
(telegram)

RegisterReceiver
TelegramReceived
(telegram)

Message Distributor

Web Dashboard

- Separate ui and data
- Single page app
- Small components (UI and logic)

ES6 classes

```
class CurrentReadingsService {
    constructor() {
        const hostname = config.apiLocation();
        this.base_url = `wss://${hostname}/api/actual`;
    }

    connect(cb) {
        this.ws = new WebSocket(this.base_url);
        this.ws.onmessage = (message) => cb(JSON.parse(message.data));
    }

    disconnect() {
        this.ws.close();
    }
}

export default new CurrentReadingsService();
```

React Components (1)

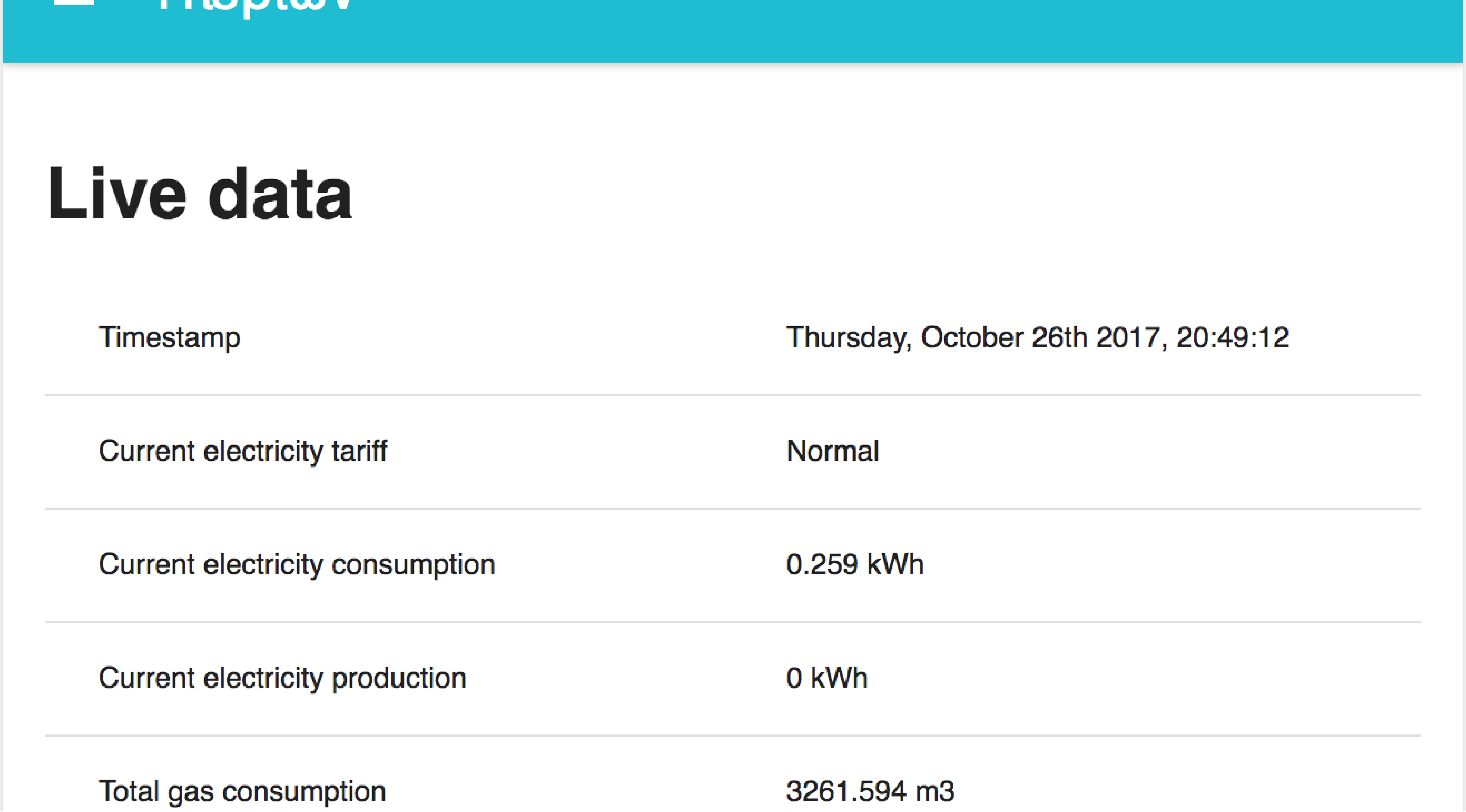
```
class CurrentReadingsPage extends React.Component {
  componentDidMount() {
      currentReadingsService.connect((data) => {
          this.setState({
              currentReading: data,
          });
      });
  }

  componentWillUnmount() {
      currentReadingsService.disconnect();
  }

  // continued...
}
```

React Components (2)

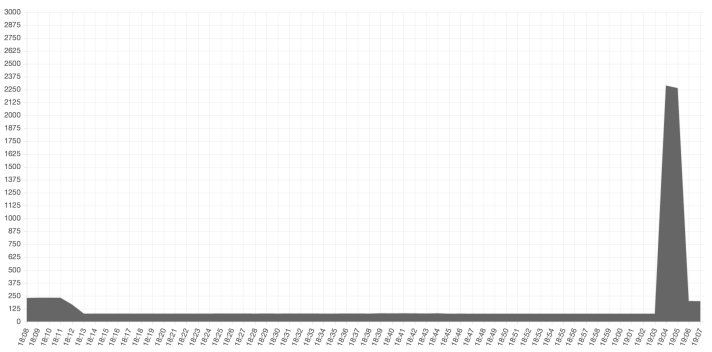
```
class CurrentReadingsPage extends React.Component {
  // continued...
  render() {
    const makeRow = (label, value) => (<Row>
      <Col lg={ 6 }><strong>{ label }</strong></Col>
      <Col lg={ 6 }>{ value }</Col>
    </Row>);
    return (<Grid>
      { makeRow("Last updated",
                formattingService.formatDateFull(ts)) }
      { makeRow("Electricity consumption",
                formattingService.formatNumberPower(consumption)) }
      { makeRow("Electricity production",
                formattingService.formatNumberPower(production)) }
      { gas ? makeRow("Gas meter",
                formattingService.formatNumberGas(gas)) : null }
      { makeRow("Current tariff", tariff) }
    </Grid>);
  }
}
```



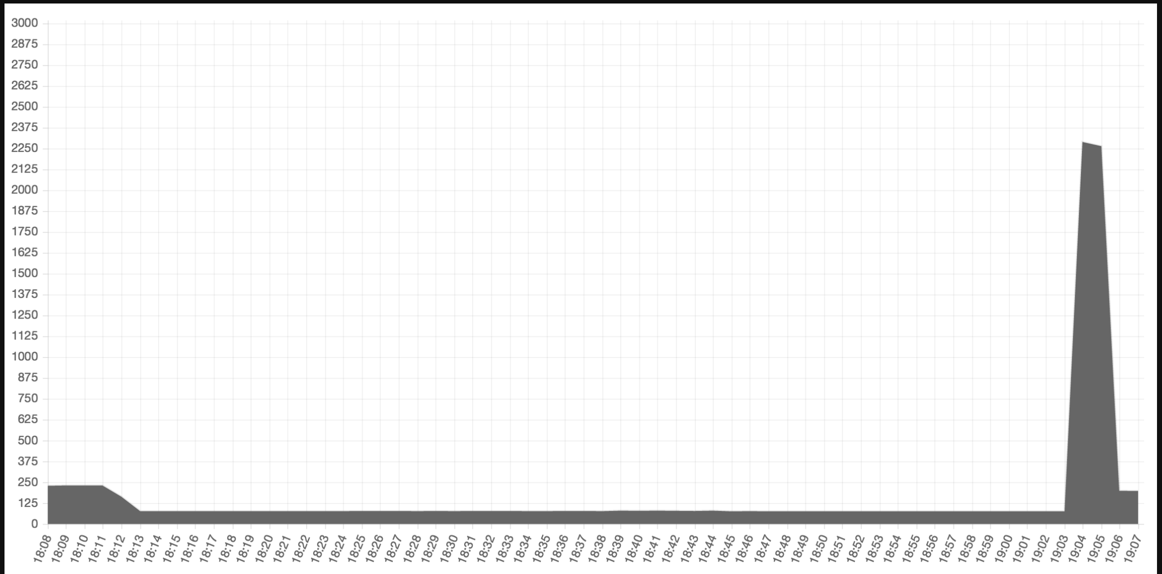
Live data

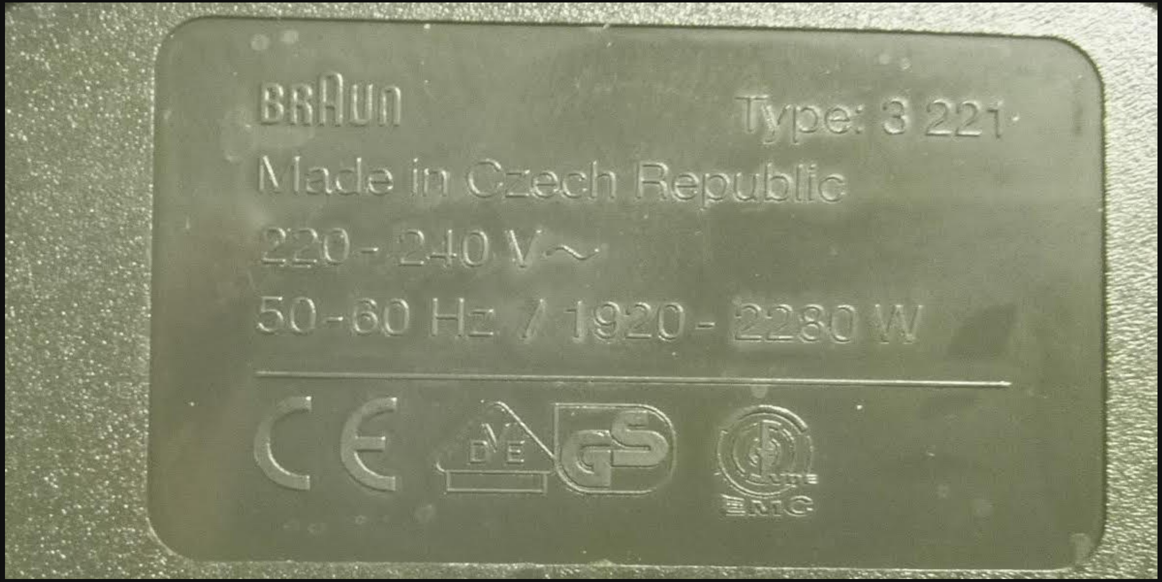
Timestamp	Thursday, October 26th 2017, 20:49:12
Current electricity tariff	Normal
Current electricity consumption	0.259 kWh
Current electricity production	0 kWh
Total gas consumption	3261.594 m3

What's Going On? (1)

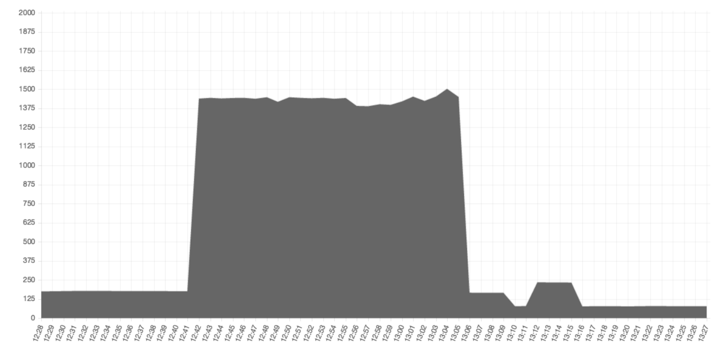


What's Going On? (1)

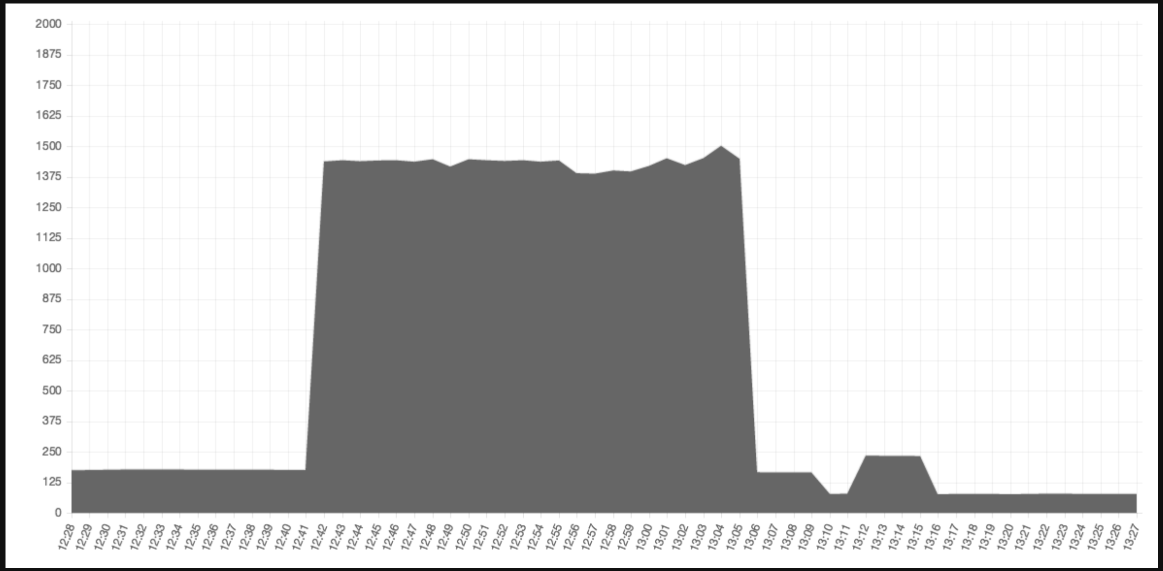


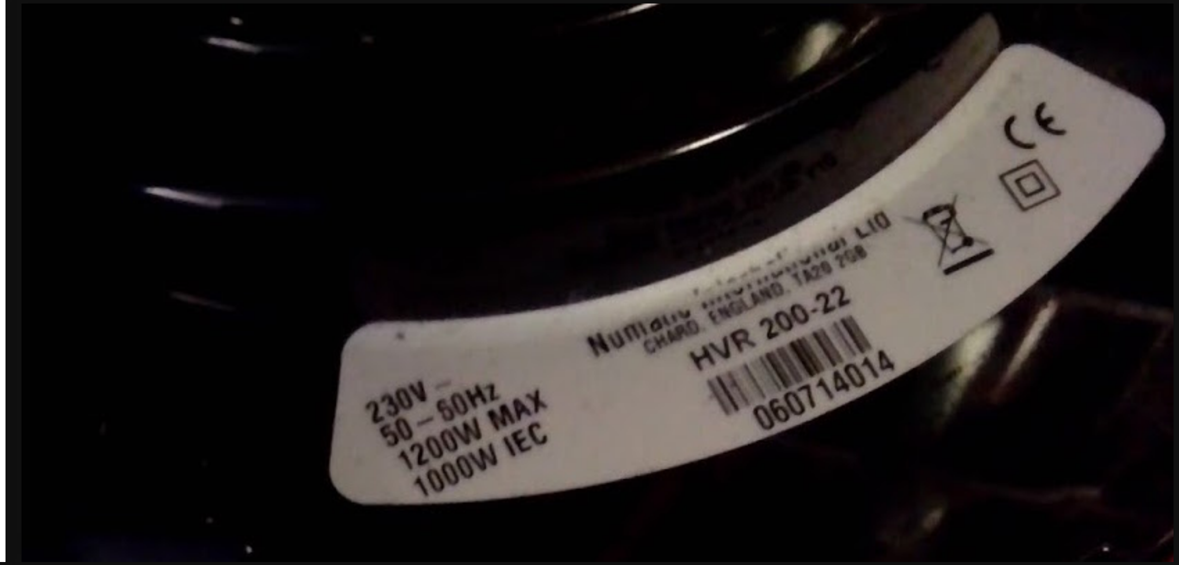


What's Going On? (2)



What's Going On? (2)





Questions?