

Outsmarting the Smart Meter



JavaCro '16 // May 19, 2016
Maarten Mulders // [@mthmulders](#)

Agenda

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

3 1 0 4 5 7 2 kWh



19617
EA
30
72

Drehstromzähler

Form C14U
3x220/380V
96 U/kWh

Nr 28855257
10(60) A
Schltg. 400

50 Hz
1972



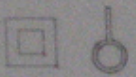
Yellow sticker: EA 30 72



Yellow sticker: EA 30 72



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



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

```
$ 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:

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

Parsing data (ctd)

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

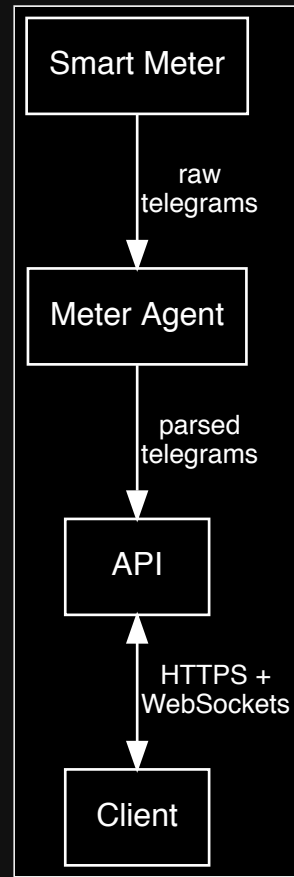
```
def make          = "/" ~> "[A-Za-z0-9]{3}".r ^^ { String(_) }
def identification = "5" ~> ".*".r      ^^ { String(_) }
def header = make ~ identification ^^ {
  case make ~ identification => P1Header(make, identification)
}
```

Parsing data (ctd)

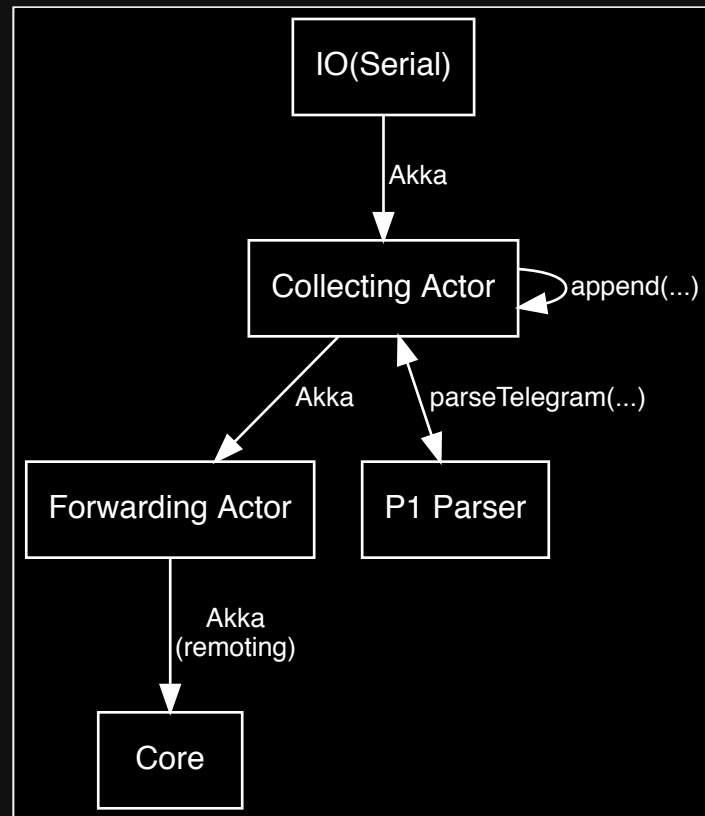
This is easily parsed with Scala's **parser combinators**

```
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")
```

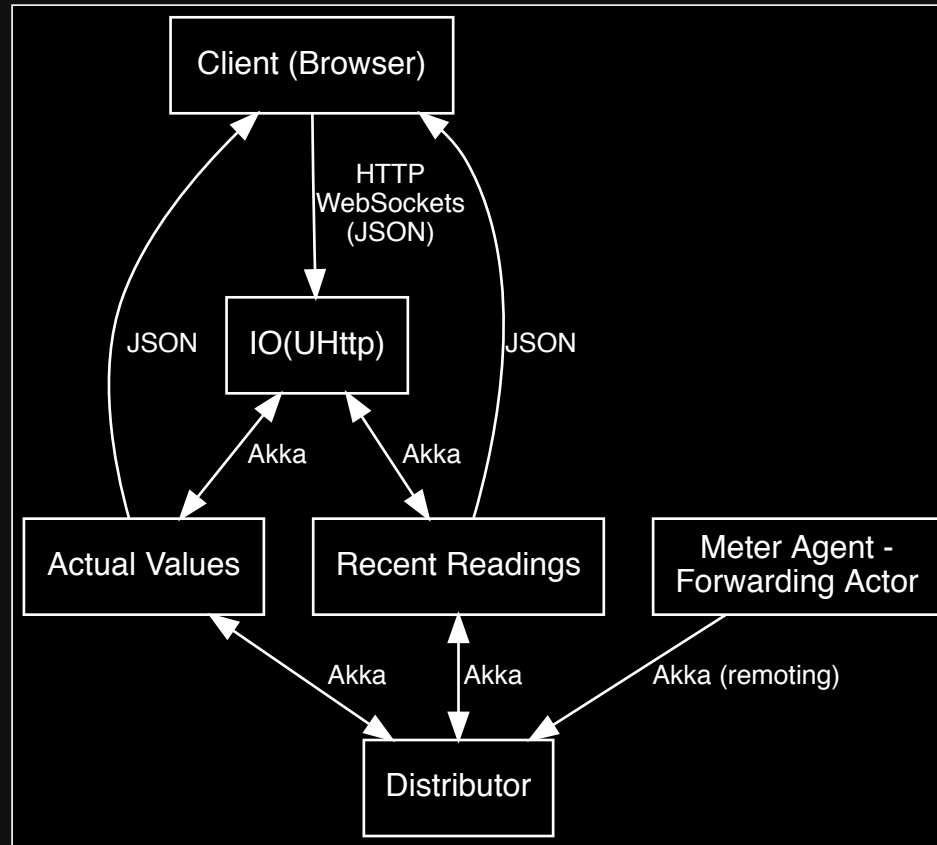
Architecture



Meter Agent



Core



Web Dashboard

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

Web Dashboard

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

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

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

export default new CurrentReadingsService();
```


Web Dashboard

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

  componentWillUnmount() {
    currentReadingsService.disconnect();
  }

  // continued...
}
```

Web Dashboard

```
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>);
  }
}
```

Web Dashboard

Last updated

Tuesday, 26 April 2016, 22:39:18 Central European Summer Time

Electricity consumption

0.403 kW

Electricity production

0.000 kW

Gas meter

1278.935 m³

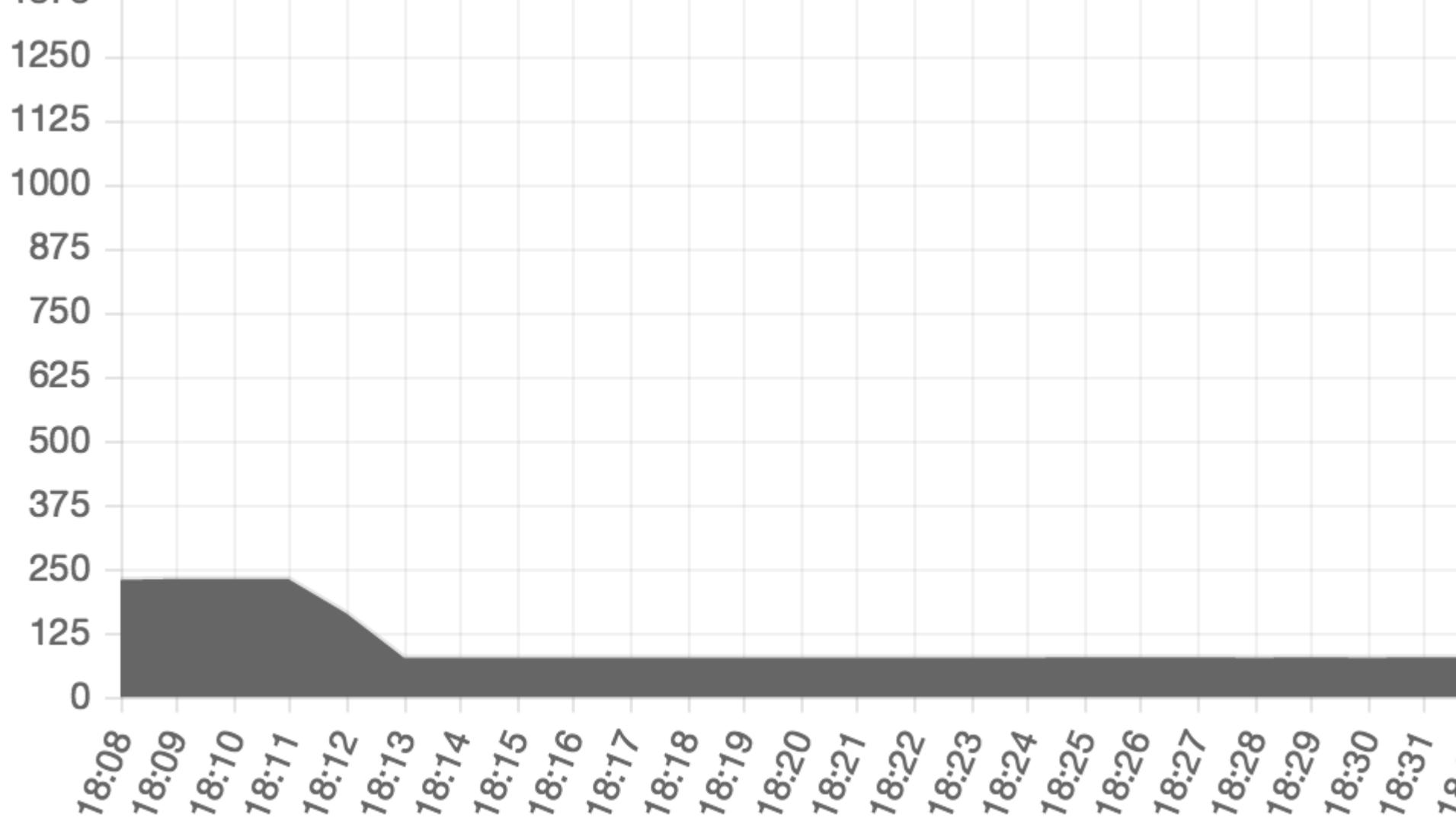
Current tariff

Normal

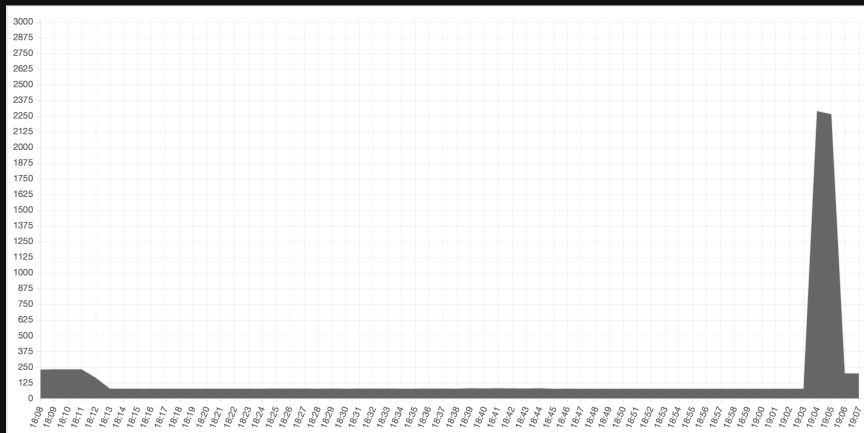
Web Dashboard

3000
2875
2750
2625
2500
2375
2250
2125
2000
1875
1750
1625
1500
1375

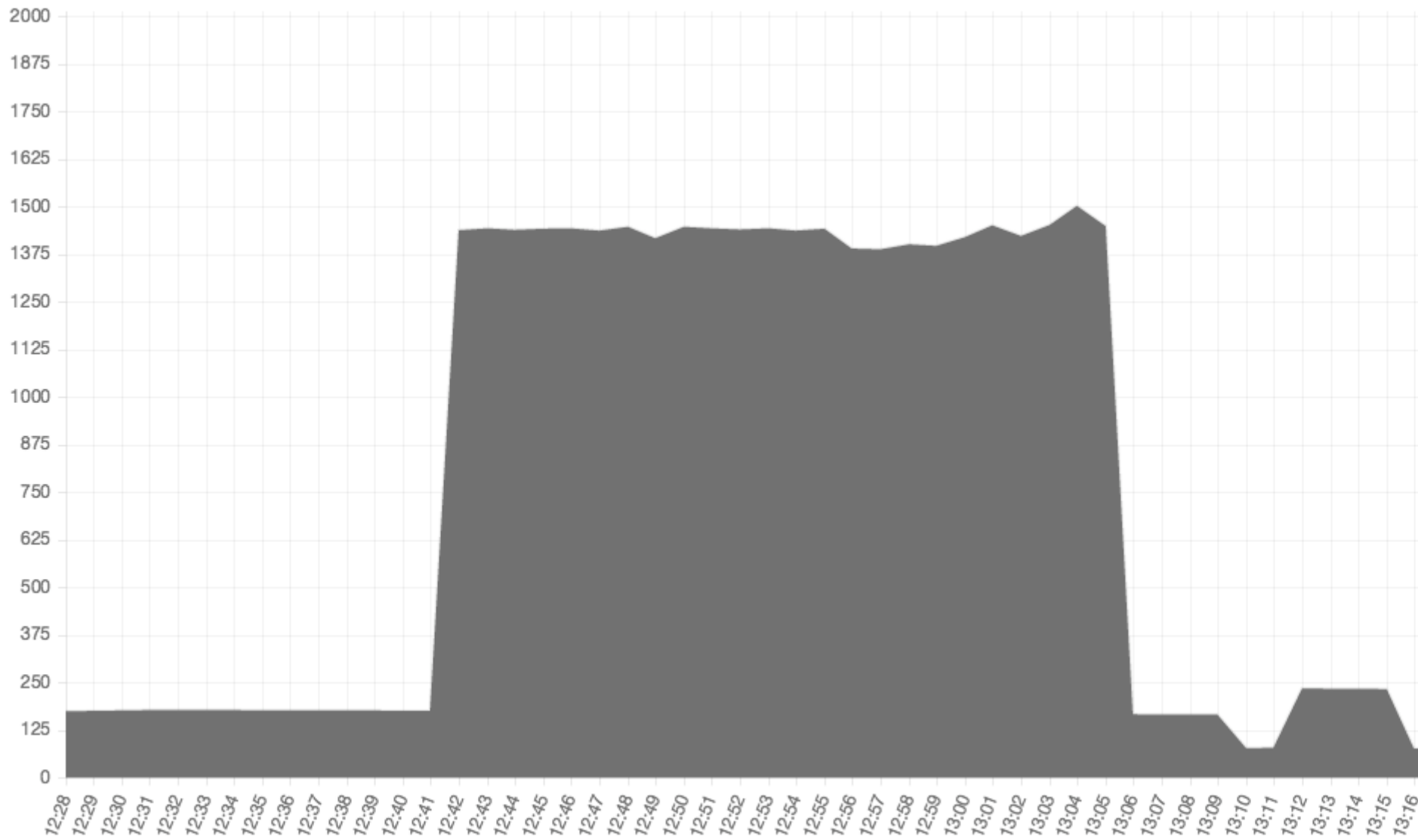




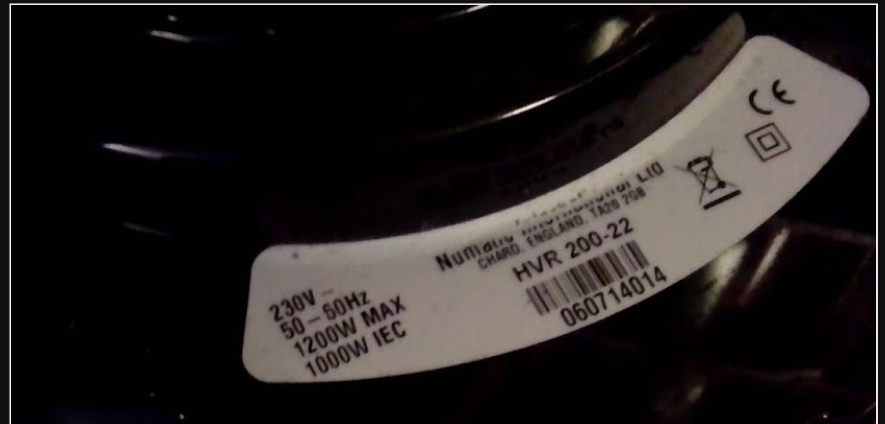
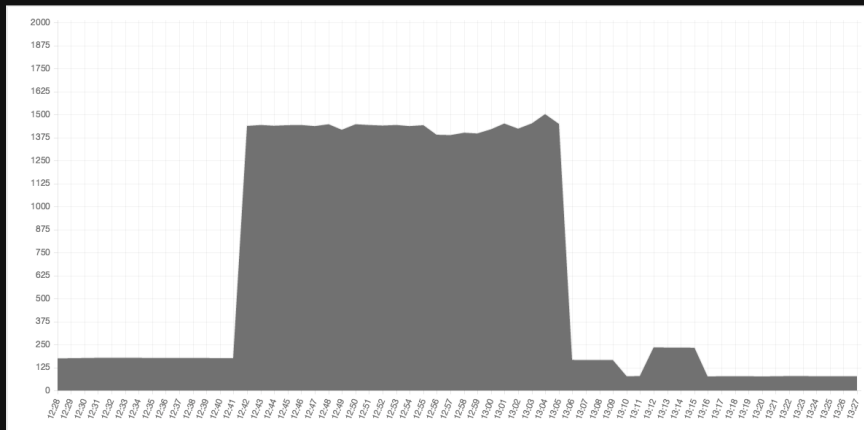
Web Dashboard



Web Dashboard



Web Dashboard



Questions?