BLOCKCHAIN MINI HACKATHON

Entwicklung eines Stromtarifes, welchen reale Stromkunden buchen können.

STROMDAO

AGENDA

1. Installation der Softwarekomponenten
2. Hallo Welt Beispiel (Stromzähler in der Blockchain)
3. **STROM**DAO Business Objekt
4. Das Stromkonto
5. Entwicklung eines Stromtarifes

VORBEREITUNG/INSTALLATION

VORAUSSETZUNGEN

- Text Editor (Zum Beispiel Notepad++, Atom.io,...)
- Node JS Version 7.0 (Download)

INSTALLATION

```
npm install -g fury.network
npm install -g stromdao-bo-mpo
```

UNINSTALL

```
npm remove -g fury.network
npm remove -g stromdao-bo-mpo
```

```
root@stromdao-convertible:/tmp# fury help

  Commands:

    help [command...]     Provides help for a given command.
    exit                  Exits application.
    init <path>           Create a subfolder path and inits its content
    run [options] <path>  Starts local http server for given subfolder (Stop with CTRL+C)
    publish <path>        Publish given subfolder to fury.network
    auth <path>

root@stromdao-convertible:/tmp# stromdao-mp help

  Commands:

    help [command...]                           Provides help for a given command.
    exit                                        Exits application.
    store [options] <meter_point_id> <reading>  Stores Meter Point Reading for given external Meter Point ID.
    retrieve <meter_point_id>                   Retrieves Meter Point Reading for given external Meter Point ID.
    account [options] <meter_point_id>          Get Address an keys for given external Meter Point ID.
    credit <meter_point_id> <amount>            Add credit to Meter Point ledger.
    ledger <meter_point_id>                     Retrieve Ledger Information for meter point id (Stromkonto).

root@stromdao-convertible:/tmp#
```

STROMDAO

FURY.NETWORK

- Browser basierte Entwicklungsumgebung
- Zugriff auf das Business Objekt
- Interaktionen mit Anwendern
- Showcase Entwicklung
- https://fury.network/

STROMDAO-BO-MPO

- Kommandozeilen Tool
- Stromzähler Ablesung
- Messwerte mit Hilfe des Business Objektes verarbeiten
- Tarifierung (Settlement/Clearing)
- https://www.npmjs.com/package/stromdao-bo-mpo

STROM DAO BUSINESS OBJEKT

- Abstraktionsschicht zur Energy Blockchain
- Bibliothek für Energiewirtschaftliche Prozesse
- Konsenssystem
- https://www.npmjs.com/package/stromdao-businessobject

STROMDAO

AUFGABE 1: HALLO WELT

Wir wollen mit Hilfe der Kommandozeile einen Zählerwert in der Blockchain speichern und diesen im Anschluss per Webbrowser abrufen.

AUFGABE 1: HALLO WELT

```
$ stromdao-mp store ZAEHLER1337 1234
TX: 0x67ba79a720202a4b0315fa9c1bc5847a2d18d9e7d664a6b9c50e0501b0108
```

Für den Zähler mit der Kennung ZAEHLER1337 wurde der Zählerstand 1234 in der Transaktion 0x67ba... geschrieben.

AUFGABE 1: HALLO WELT

Abrufen eines Zählerwertes Fury.Network (IPFS)

Adresse des Zählers:

```
$ stromdao-mp account ZAEHLER1337
MPID ZAEHLER1337
Address 0x4c01e6a3649cDEB08029D14bcdeB39366f9317F3
...
```

STROMDAO

AUFGABE 1: HALLO WELT

STROMDAO

LERNZIEL

- Jede Zählerkennung hat eine eindeutige Adresse in der Blockchain.
- Zugriff auf die Blockchain kann via Browser oder Kommandozeile erfolgen.
- Zählerstände bilden bereits einen Konsens.

FURY IDE (LOKAL)

Schnelles Testen und verproben von Anwender Interaktionen.

Anlegen eines lokalen Showcases

```
$ fury init aufgabe2
$ fury run aufgabe2
```

Es wird ein Unterverzeichnis angelegt mit dem Namen *aufgabe2*. Darin ist eine *base.html* und eine *base.js*.

AUFGABE 2: LOKALE NUTZUNG VON FURY

1. Herunterladen der ZIP Datei
2. Entpacken und Überschreiben der beiden Dateien in *aufgabe2*
3. Reload der Seite im Browser

BUSINESS OBJEKT

STROMDAO

BUSINESS OBJEKT

```
node.mpr().then(function(mpr) {
        mpr.readings($('#meterpointaddress').val()).then(function(o) {
             console.log(o);
                d=new Date((o.time.toString())*1000);
                $('#ts').html(d.toLocaleString());
                $('#power').html(o.power.toString());
        });
});
```

Das Business Objekt wird mit *node* angesteuert. Es ist bereits initialisiert, so dass jede Instanz eine eigene Adresse und einen eigenen privaten Schlüssel besitzt.

BUSINESS OBJEKT

```
node.mpr().then(function(mpr) {
        mpr.readings($('#meterpointaddress').val()).then(function(o) {
             console.log(o);
                d=new Date((o.time.toString())*1000);
                $('#ts').html(d.toLocaleString());
                $('#power').html(o.power.toString());
        });
});
```

Im Objekt gibt es verschiedene Domains. Hier wird Meter-Point-Reading (*mpr*) verwendet. Diese Domains entsprechend meist sogenannten Smart-Contracts in der Blockchain.

BUSINESS OBJEKT

```
node.mpr().then(function(mpr) {
        mpr.readings($('#meterpointaddress').val()).then(function(o) {
             console.log(o);
                d=new Date((o.time.toString())*1000);
                $('#ts').html(d.toLocaleString());
                $('#power').html(o.power.toString());
        });
});
```

In diesem Aufruf entspricht *readings* einem Methodenaufruf im Smart Contract.

STROMKONTO

STROMDAO

STROMKONTO

Stromkonto

Konto

Nummer/Account

0x4389320f49264bd84D6f9809787E17988b442B93

Öffnen

Soll	Haben	Saldo
0.0171512	0	-0.0171512

Block	Von	An	Betrag
#58680	0x4389320f49264bd84d6f9809787e17988b442b93	0xedfc2ee5ef07c8bd9309c301e4008475a9d5f523	0.0141015
#58678	0x4389320f49264bd84d6f9809787e17988b442b93	0xedfc2ee5ef07c8bd9309c301e4008475a9d5f523	0.0019157
#58675	0x4389320f49264bd84d6f9809787e17988b442b93	0xedfc2ee5ef07c8bd9309c301e4008475a9d5f523	0.0005602
#58673	0x4389320f49264bd84d6f9809787e17988b442b93	0xedfc2ee5ef07c8bd9309c301e4008475a9d5f523	0.0003128
#58668	0x4389320f49264bd84d6f9809787e17988b442b93	0xedfc2ee5ef07c8bd9309c301e4008475a9d5f523	0.000261

STROMKONTO

- Blockchain (Tokens), zeigen Besitz an (= kein Konzept für Schulden).
- Datentyp ist immer Integer (Ganzzahl).
- Ein Stromkonto zeigt gefilterte Transaktionen an.

AUFGABE 3: VERBRAUCHSBUCHUNG

```
$ stromdao-mp store --auto 69256 ZAEHLER1337 1234
```

- Öffnen einer Fury Instanz für das Stromkonto.
- Durchführung der Zählerablesungen.
- Kontrolle der Abrechnung.

LERNZIEL

- Das Business Objekt abstrahiert die Blockchain und den Energiemarkt.
- Das Stromkonto ist ein Kontenbuch für Energieabrechnungen.
- In der Energy Blockchain existieren vorgefertigte Smart Contracts.

TARIFENTWICKLUNG

STROMDAO

TARIFENTWICKLUNG

- Ein Stromliefervertrag besteht aus einer Vielzahl von Buchungen auf einem Stromkonto
- Buchungen werden auf Basis eines Eingangsvektors (Settlement Objekt) vorgenommen

```
node.stromkontoproxy("0x19BF166624F485f191d82900a5B7bc22Be569895").then(function(sko) {
        sko.addTx(von,an,kosten,energiemenge).then(function(tx) {
            //...
        });
});
```

SETTLEMENT OBJECT

```
{ tarif:
   { '164fb093ddc8e8a0c13605eca8760abc':
      { GeoKey: '164fb093ddc8e8a0c13605eca8760abc',
        Zipcode: '69256',
        City: 'Mauer',
        District: '',
        CustomCode: '',
        UsageStart: 1,
        UsageEnd: 100000,
        BpNet: 102.56,
        BpGross: 122.04639999999999,
        UpNet: 20.089,
        UpGross: 23.90591,
        BoNet: 0,
        BoGross: 0,
        BoNewNet: 0,
        BoNewGross: 0,
        BoInstantNet: 0,
        BoInstantGross: 0,
        BoPercent: 0,
        Net: '65929202',
        Total: 705.23 } },
  BpGross: 1220464000,
  UpGross: 2390591,
  account: '0x0296afd730457408cEBaf0dB2373d541423E0ACf',
  node_account: '0xEDFC2EE5ef07c8Bd9309c301e4008475A9d5F523',
  start:
   Result {
     '0': BigNumber { _bn: <BN: 59bce7c5> },
     '1': BigNumber { _bn: <BN: fb> },
     time: BigNumber { _bn: <BN: 59bce7c5> },
     power: BigNumber { _bn: <BN: fb> },
     length: 2 },
  end:
   Result {
     '0': BigNumber { _bn: <BN: 59bce7f1> },
     '1': BigNumber { _bn: <BN: fc> },
     time: BigNumber { _bn: <BN: 59bce7f1> },
     power: BigNumber { _bn: <BN: fc> },
     length: 2 },
  cost: 4094,
  base: 1 }
```

STROMDAO

TARIFENTWICKLUNG

Datei *settlement_out.js* anlegen mit dem Inhalt:

```
console.log(settlement);
```

Speichern und im Anschluss

```
$ stromdao-mp store -f settlement_out.js --de 69256 ZAEHLER1337 123
```

TARIFENTWICKLUNG

Datei *settlement_simple.js* anlegen mit dem Inhalt:

```
global.promise = new Promise(function(resolve2, reject2) {
        node.stromkontoproxy("0x19BF166624F485f191d82900a5B7bc22Be5
                        sko.addTx(node.wallet.address,node.nodeWall
                                        resolve2(tx);
                        });
        });
});
```

Speichern und im Anschluss

```
$ stromdao-mp store -f settlement_simple.js --de 69256 ZAEHLER1337
```

SKO.ADDTX(VON,AN,WERT,ENERGIE)

- führt die eigentliche Buchung in der Blockchain durch
- kann mehrfach hintereinander für Splitbuchungen aufgerufen werden
- nur bei Kosten über 0 wird eine Buchung ausgeführt
- der Promise erlaubt einen Aufruf von der Komandozeile

SKO.ADDTX(VON,AN,WERT,ENERGIE)

- bei einem Tarif muss der settlement.cost entsprechend modifiziert werden
- im Settlement Object sind alle Daten vorhanden, die zur Berechnung notwendig sind
- es steht der volle Umgang von Javascript (NodeJS) zur Verfügung
- auf das Business Objekt kann zugegriffen werden