

STROM DAO

# Training: Fury Network Introduction

Stefan Thon

# Recap



# Decentralisation

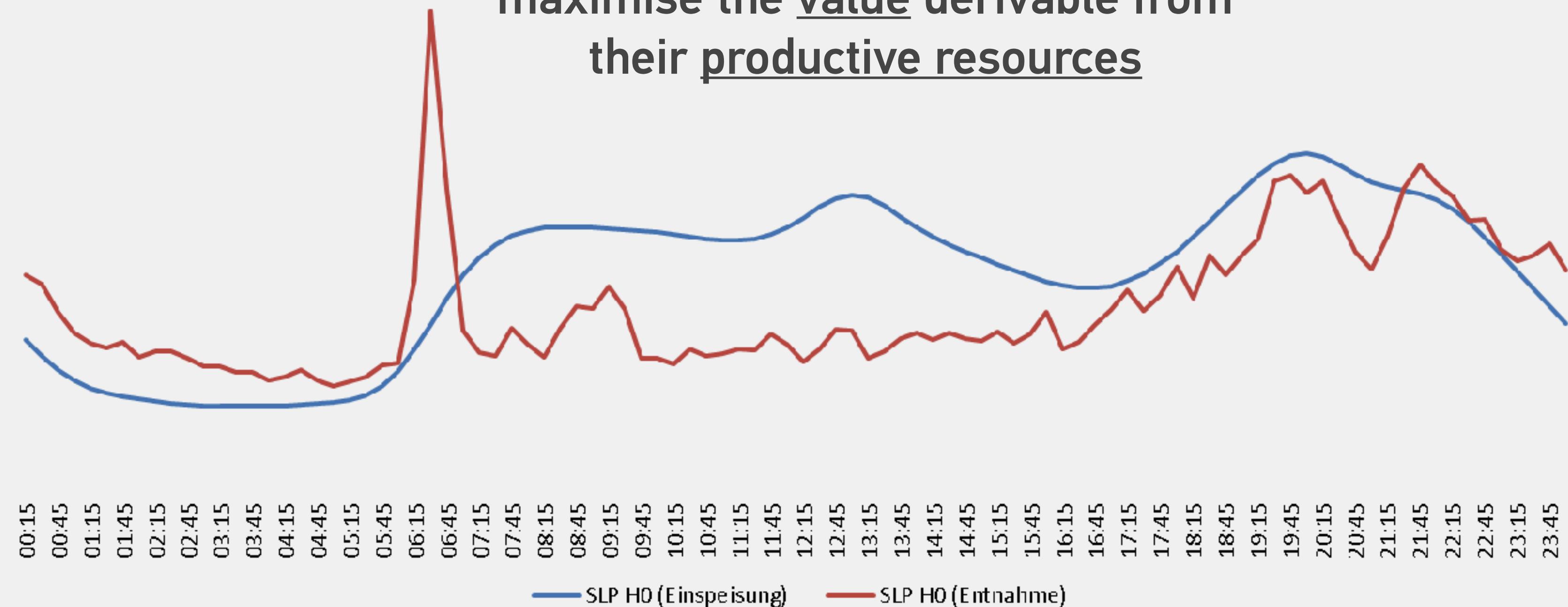


# Market integration

People demand greater agency in shaping their energy and supply systems.\*

- \* There is a latent but increasing desire to more actively engage with energy markets: Self supply, battery storage, power to heat, electric vehicles, demand response peer-2-peer trading...

People demand market access to  
maximise the value derivable from  
their productive resources



# Market access, requires access to the means of market communication

Regelungen zur Adressierung

**EDI@Energy Kommunikationsrichtlinie**  
**Verfahrensbeschreibung zur Abwicklung des**  
**Austauschs von EDIFACT-Dateien**

**We need to evolve the language (protocol)  
of energy market communication**

**Enabling equal market access and opportunity  
for everybody, regardless of scale**

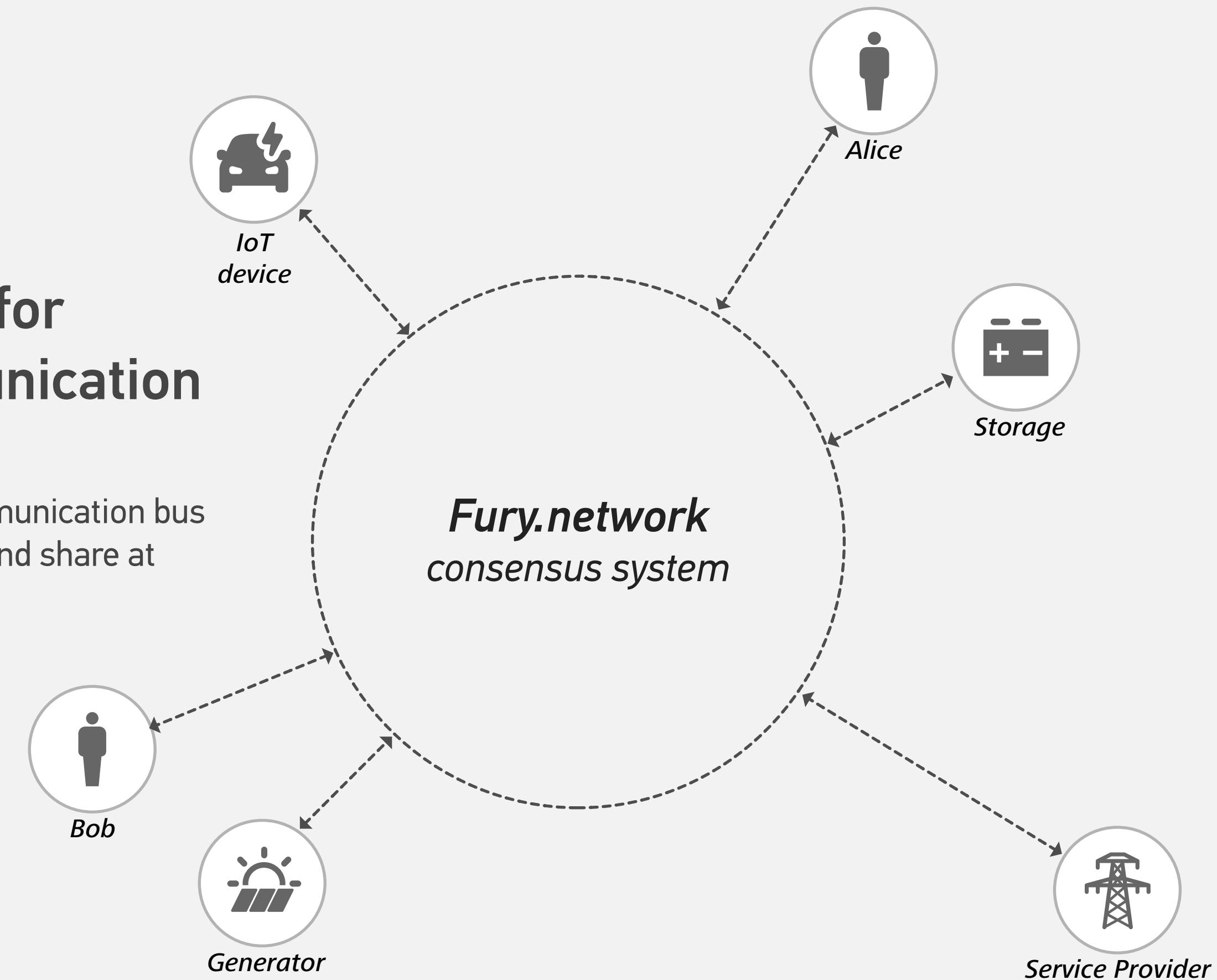
# A consensus system for energy market transactions

FURY NETWORK

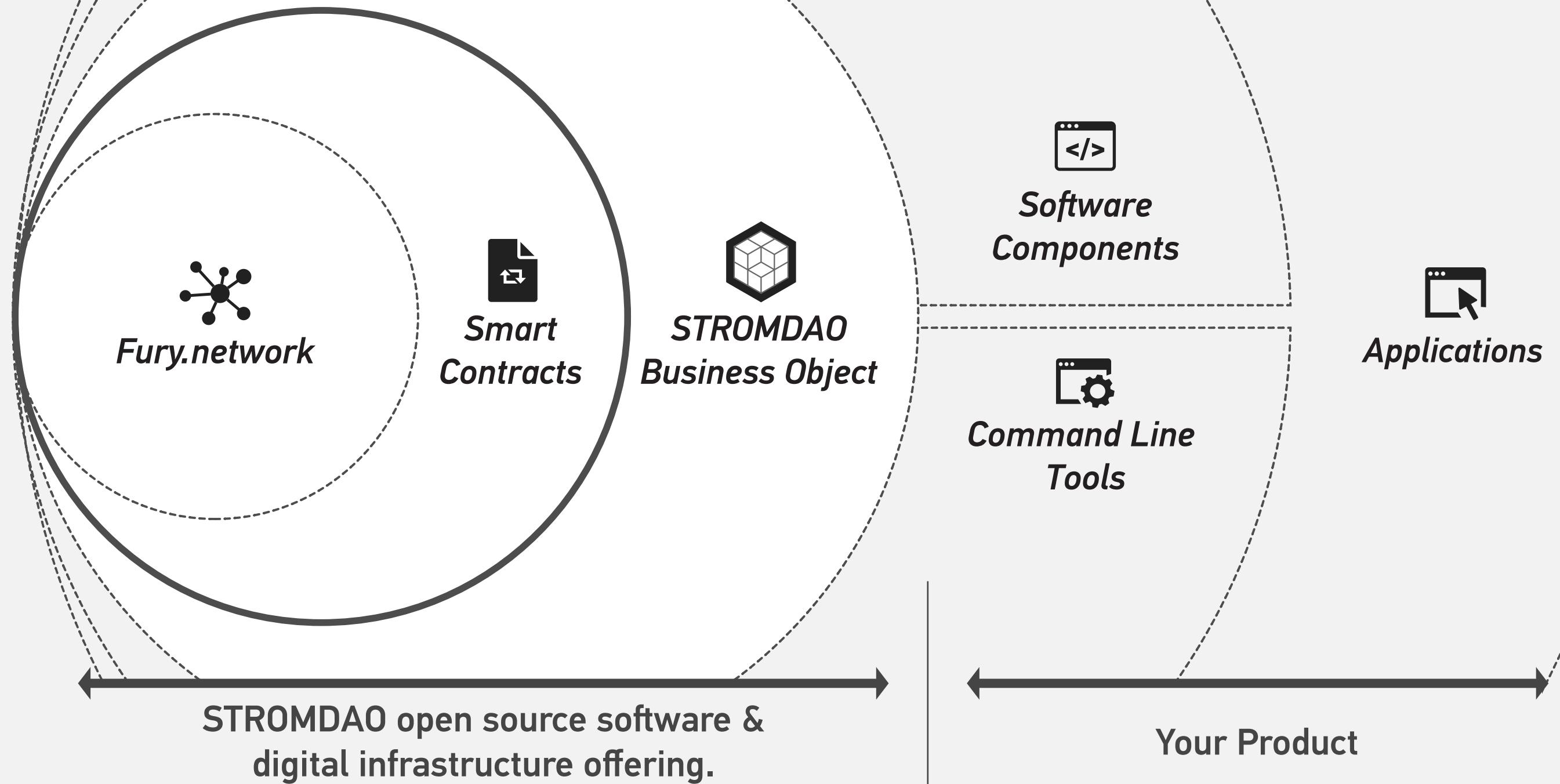
# A blockchain network for energy market communication

A blockchain that acts as a single communication bus that all market participants each own and share at the same time

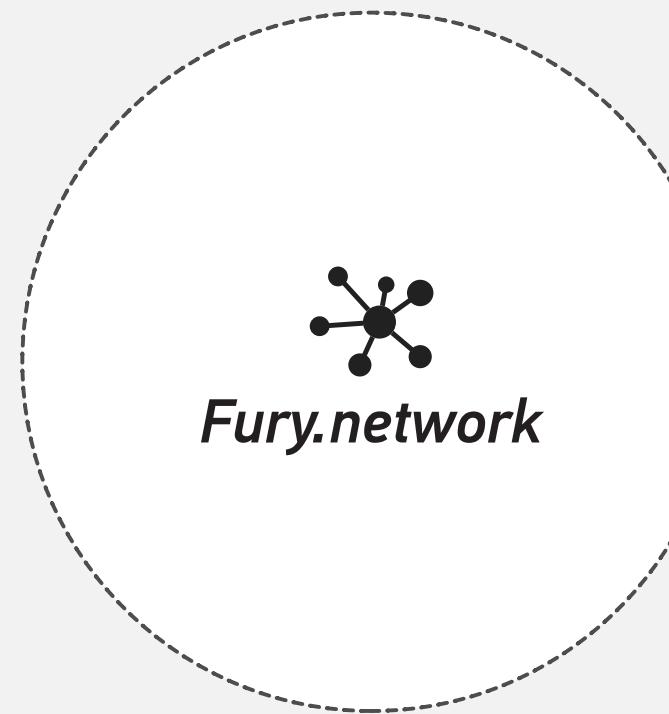
***Fury.network***  
consensus system



# Fury energy market consensus system



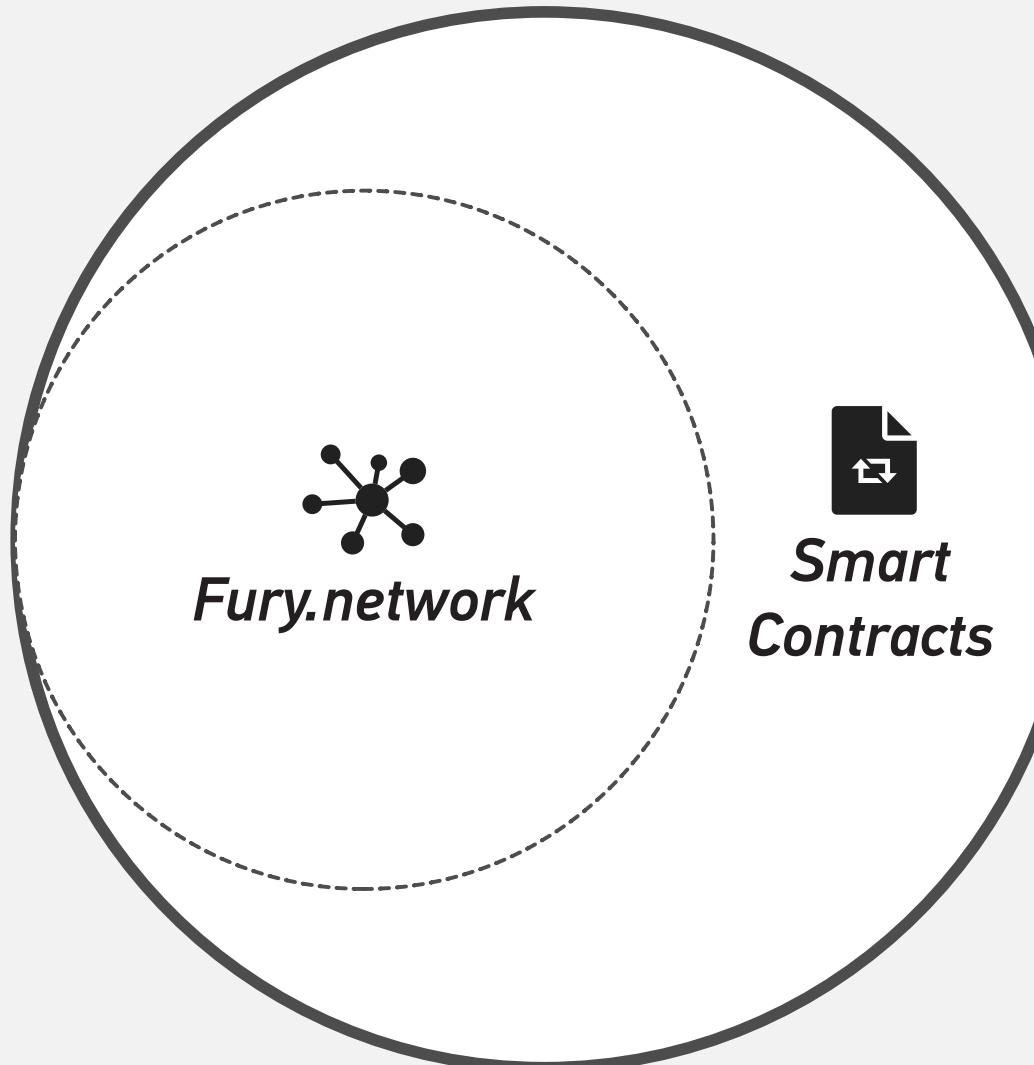
# Fury consensus system



## Fury Energy Blockchain

It is Ethereum under the hood.

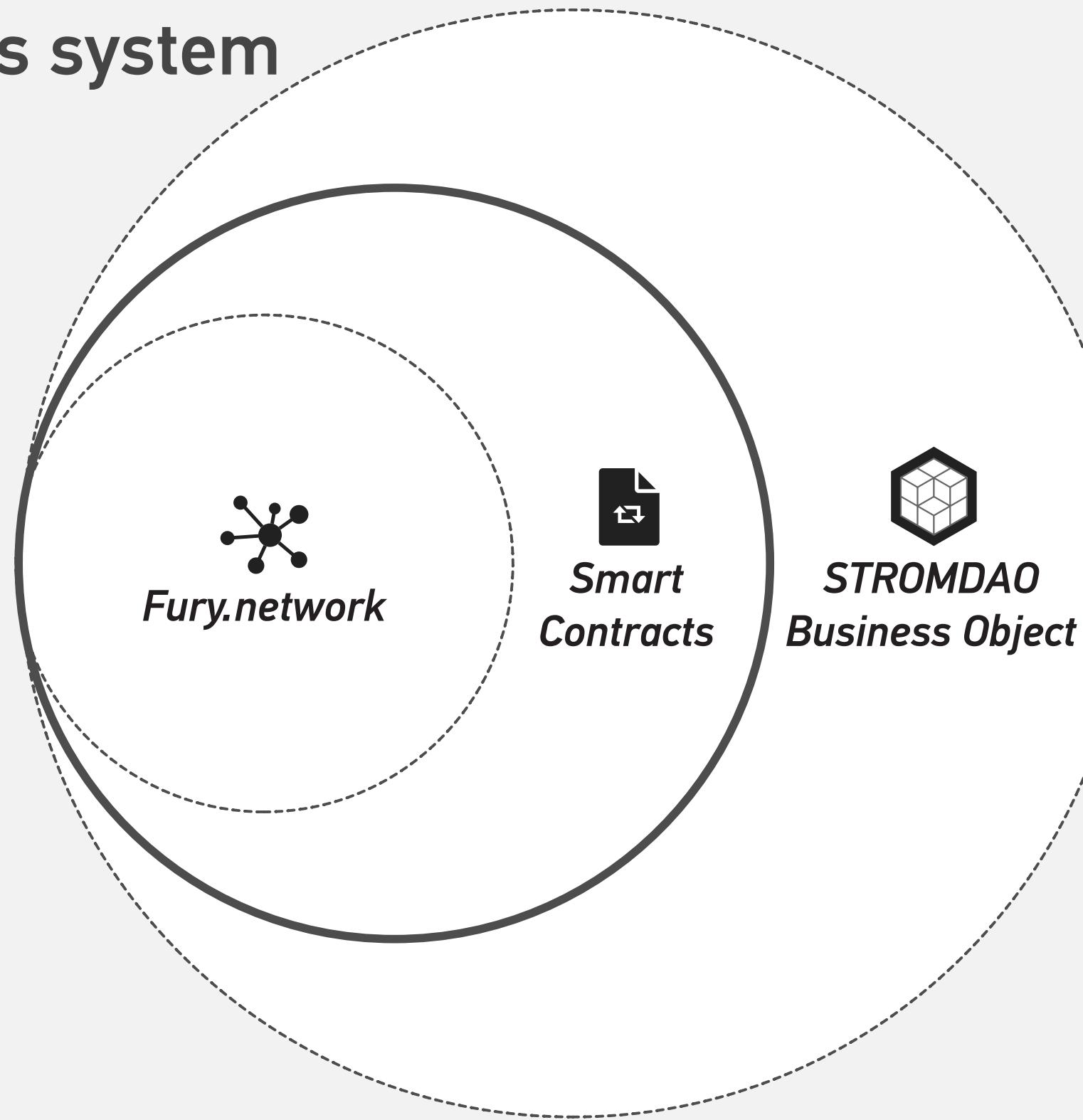
# Fury energy market consensus system



## Fury Native Smart Contracts

Deployed into the network. Ready to use.

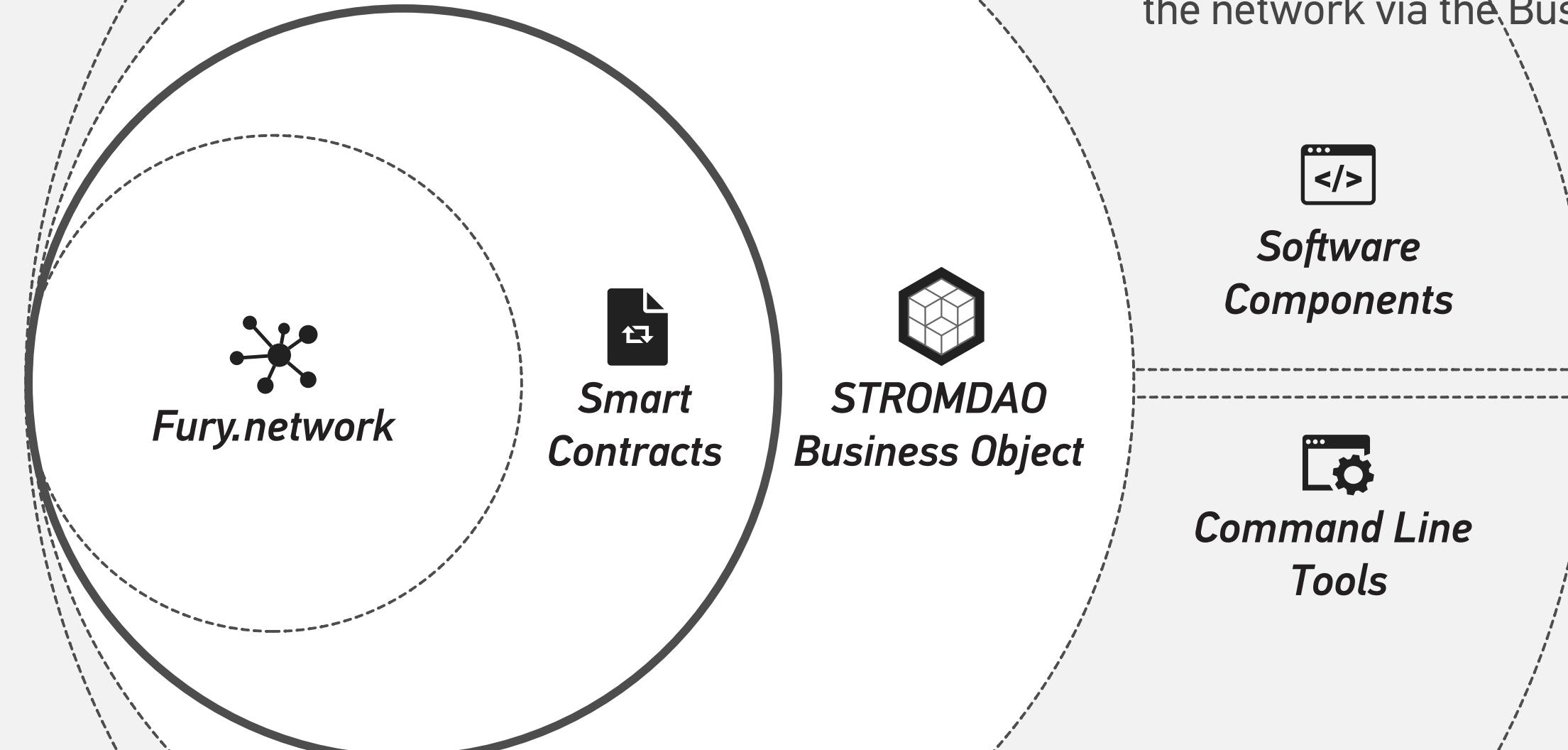
# Fury energy market consensus system



## Integration Library

Abstraction layer between blockchain technology and app business logic.

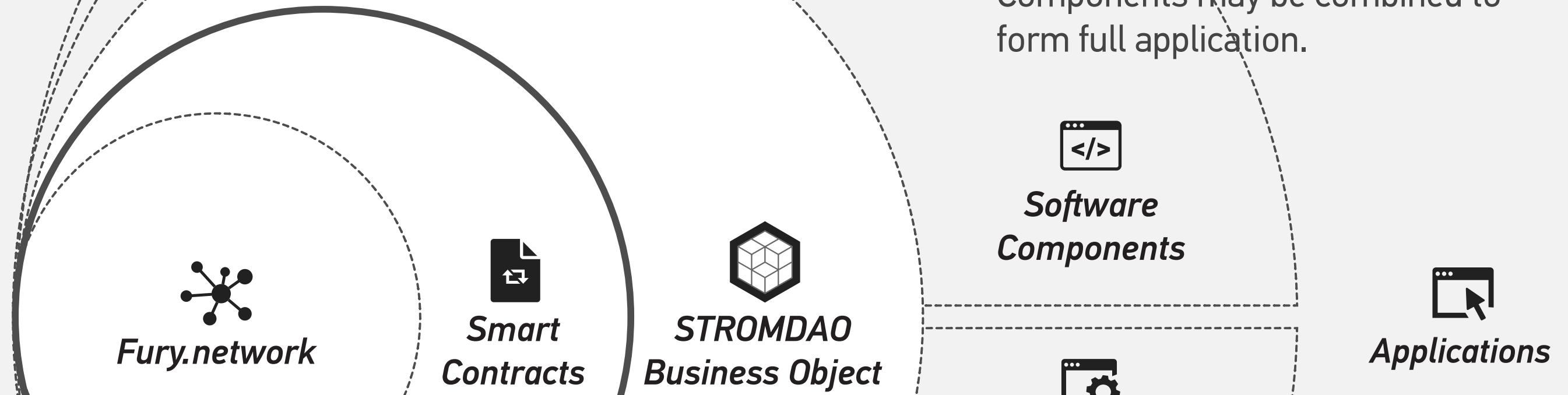
# Fury energy market consensus system



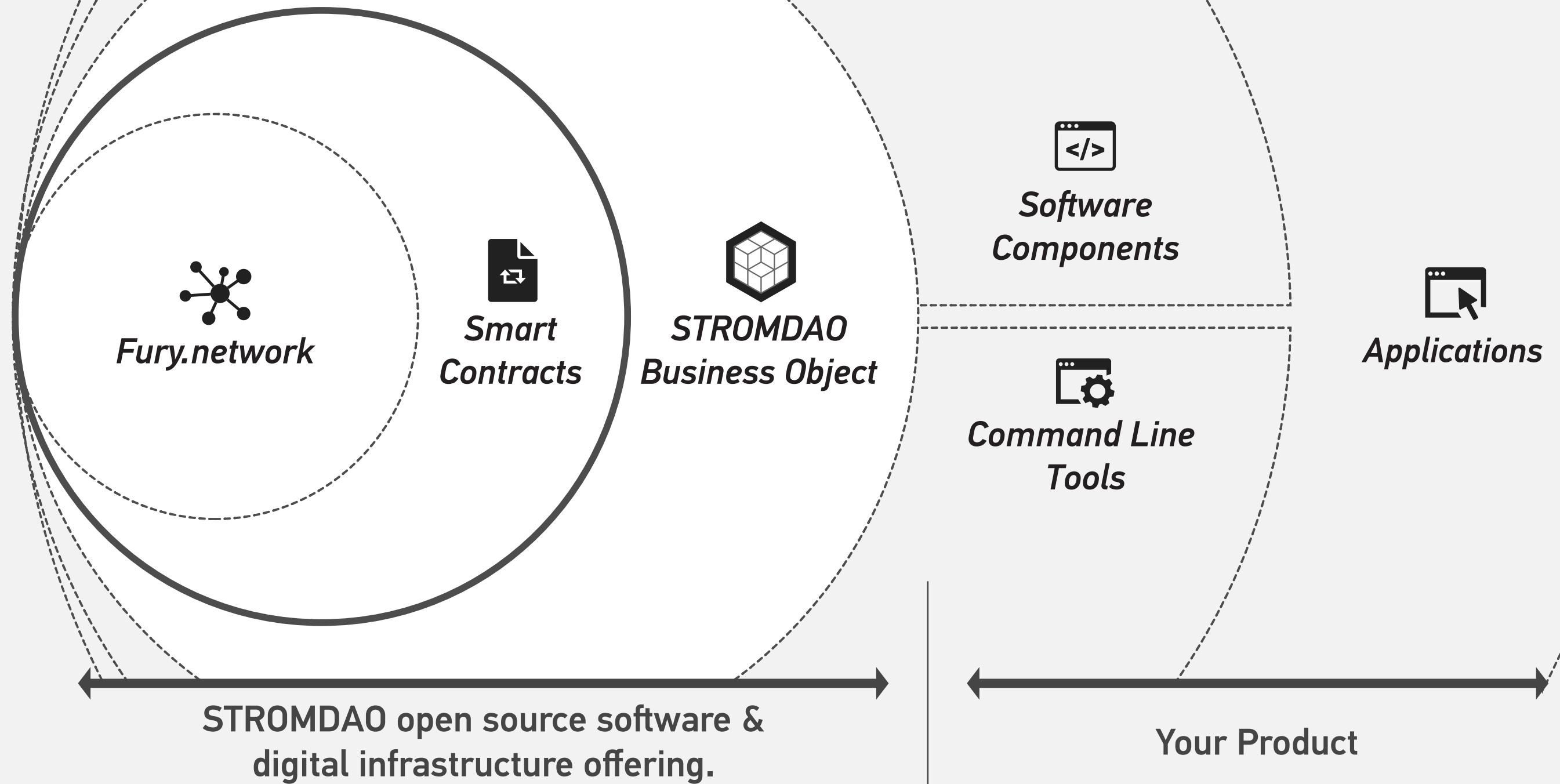
## Software components

Custom software accesses with the network via the Business Object.

# Fury energy market consensus system



# Fury energy market consensus system



# Why Fury?

# Open Access, Coinless, PoA

Proof of Authority (PoA) implementation of the Ethereum protocol with zero transaction cost. Access to the network is unrestricted.



## FURY NATIVE CONTRACTS

# Smart Contracts enforcing market processes and statutory regulations

Network-native smart contracts encapsulate energy sector specific market logic that regulates and effects the transfer of value.

```
1 pragma solidity 0.4.18;
2 contract SimpleMultiSig {
3
4     uint public nonce;           // [only] mutable state
5     uint public threshold;       // immutable state
6     mapping (address => bool) isOwner; // immutable state
7     address[] public ownersArr;   // immutable state
8
9     function SimpleMultiSig(uint threshold_, address[] owners_) public {
10        require(owners_.length <= 10 && threshold_ <= owners_.length && threshold_ != 0);
11
12        address lastAdd = address(0);
13        for (uint i=0; i<owners_.length; i++) {
14            require(owners_[i] > lastAdd);
15            isOwner[owners_[i]] = true;
16            lastAdd = owners_[i];
17        }
18        ownersArr = owners_;
19        threshold = threshold_;
20    }
21
22    // Note that address recovered from signatures must be strictly increasing
23    function execute(uint8[] sigV, bytes32[] sigR, bytes32[] sigS, address destination, uint
24    value, bytes data) public {
25        require(sigR.length == threshold);
26        require(sigR.length == sigS.length && sigR.length == sigV.length);
27
28        // Follows ERC191 signature scheme: https://github.com/ethereum/EIPs/issues/191
29        bytes32 txHash = keccak256(byte(0x19), byte(0), address(this), destination, value, data,
30        nonce);
31
32        address lastAdd = address(0); // cannot have address(0) as an owner
33        for (uint i = 0; i < threshold; i++) {
34            address recovered = ecrecover(txHash, sigV[i], sigR[i], sigS[i]);
35            require(recovered > lastAdd && isOwner[recovered]);
36            lastAdd = recovered;
37
38        // If we make it here all signatures are accounted for
39        nonce = nonce + 1;
40        require(destination.call.value(value)(data));
41    }
42    function () public payable {}
43 }
```

# STROMDAO BUSINESS OBJECT

# Software libraries to integrate blockchain technology into existing IT systems

Open-source software libraries and tools that facilitate integration of the blockchain back-end into existing IT systems and business processes.

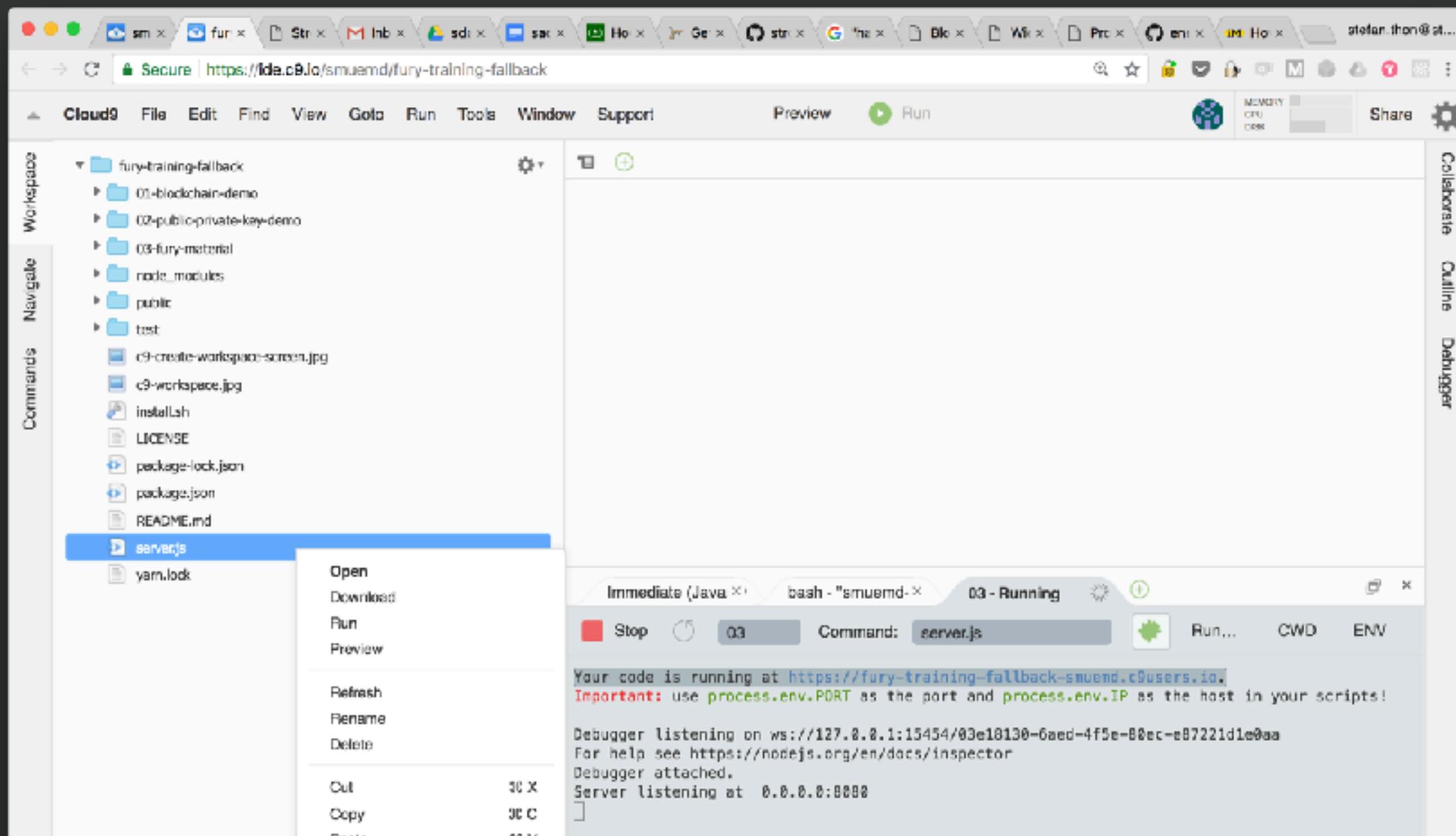
```
26 const businessObject = document StromDAOBO
27
28 /**
29 * function createFuryNode (businessObj, bus
30 // return new businessObj.Node(businessObj
31
32 /**
33 * Routes description */
34 const routesDescription = [
35 { name: 'Home', route: '/' },
36 { name: 'Login', route: '/login' },
37 { name: 'Item', route: '/item/:id' }
38 ]
39
40 /**
41 * hard coded ethereum addresses for testing*/
42 const ethAddresses = {
43 mp: '0x83F8B15eb816284ddcF2ff005Db7a19196d86ae1',
44 smpc: '0x2F516D1e3dcB330BB44c00cb919ab5081075C77E',
45 sk: '0x19BF166624F485f191d82900a5B7bc22Be569895'
46
47 /**
48 * settings export */
49 export default {
50 apiHost: apiHost,
51 businessObject: businessObject,
52 createBusinessObjectConfig: createBusinessObjectConfig,
53 ethAddresses: ethAddresses,
54 // routePrefix: '/',
55 defaultRoute: '/',
56 routesDesc: routesDescription
57 }
```

# Praxis

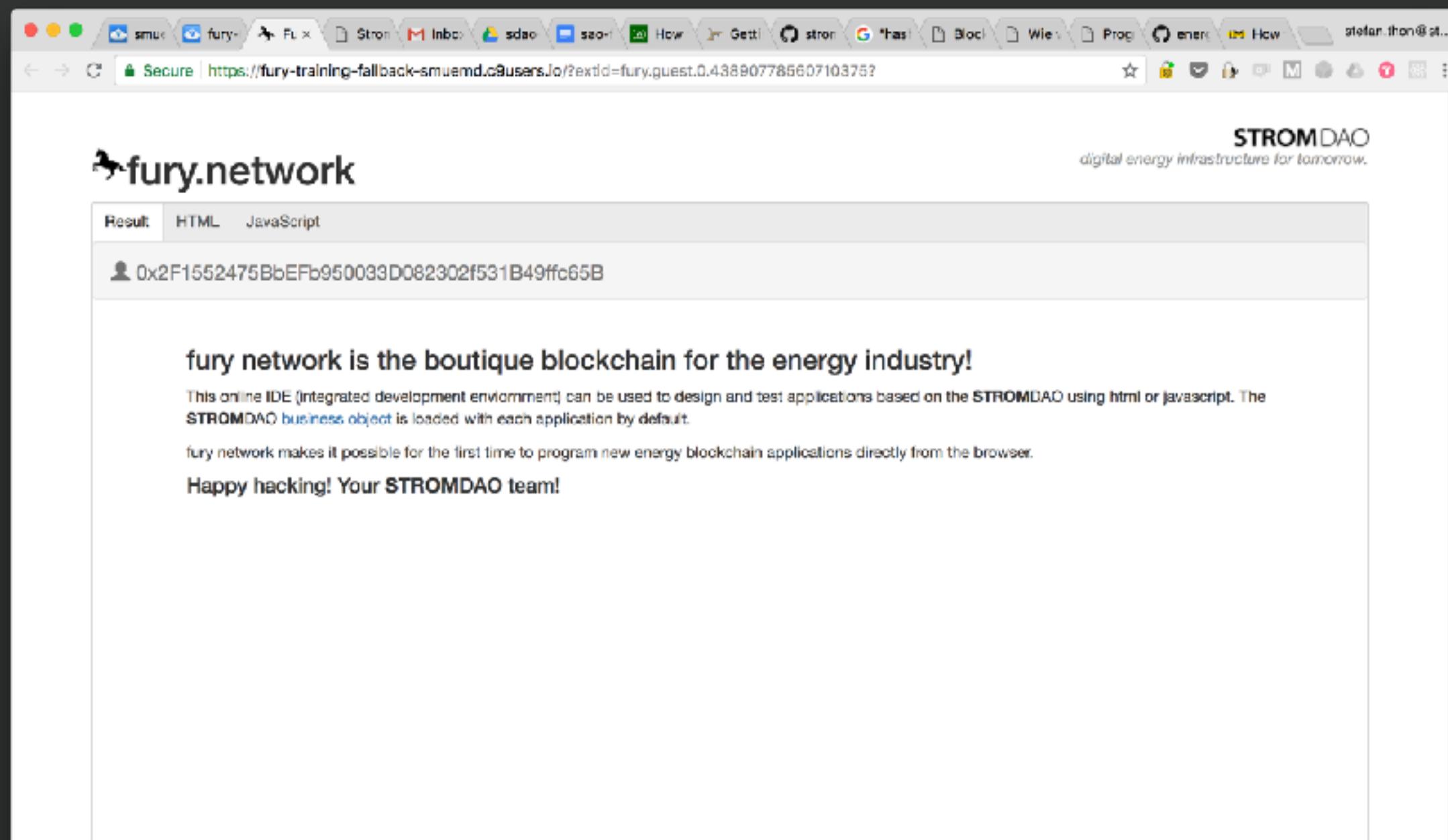
# Agenda

1. Softwarekomponenten auf [c9.io](#)
2. Web Zählerstand
3. Hallo Welt Beispiel (Stromzähler in der Blockchain)
4. STROMDAO Business Objekt
5. Das Stromkonto
6. Entwicklung eines Stromtarifes

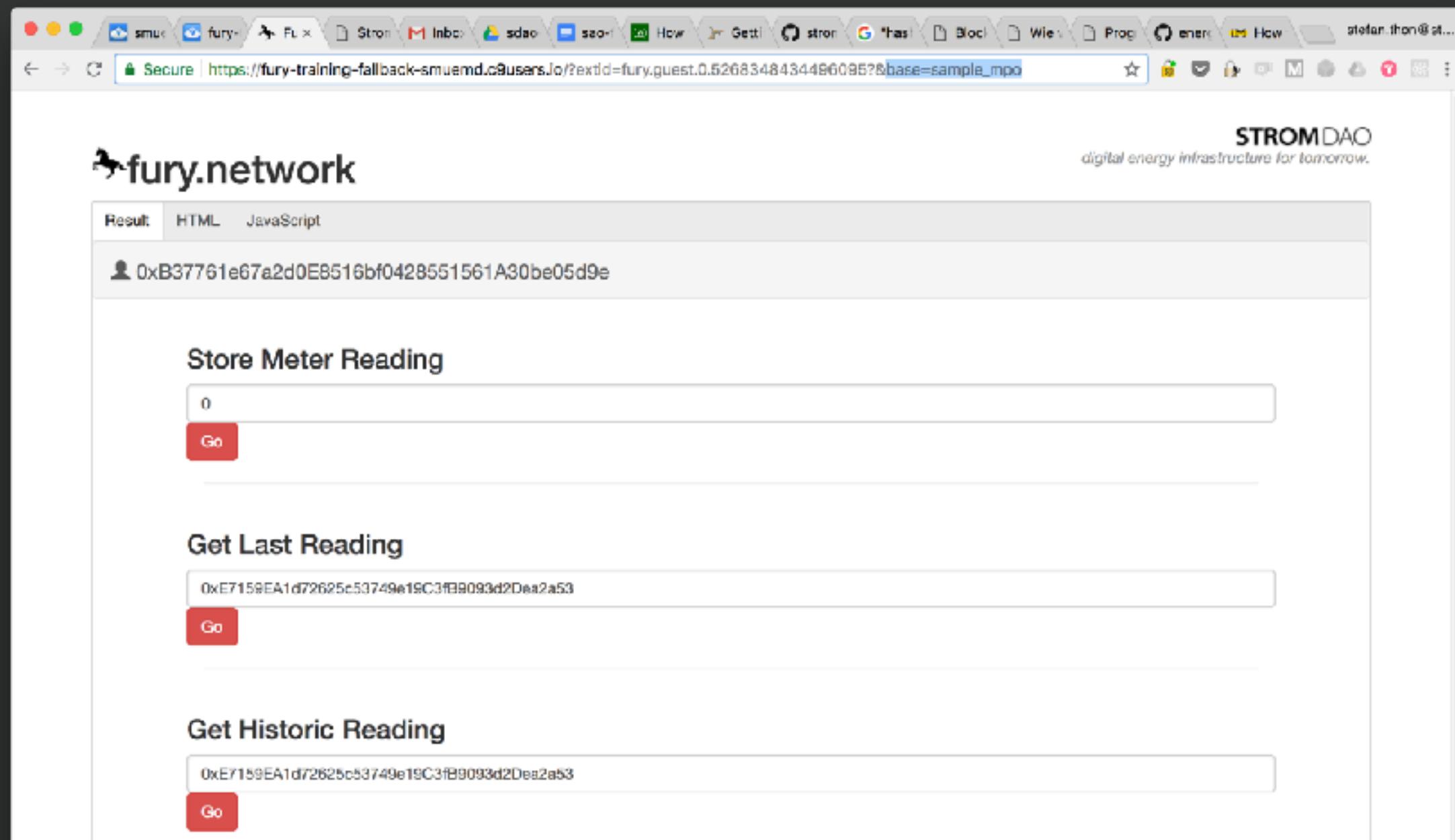
# C9 Server starten



# Preview running application auf c9users.io

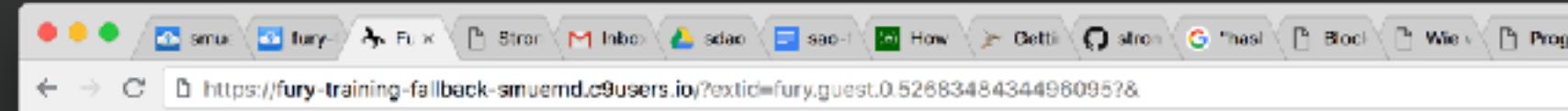


**Open [...].c9users.io/?base=sample\_mpo**



# fury.network

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



The screenshot shows a browser window with the URL <https://fury-training-fallback-simuernd.c9users.io/?extid=fury.guest.0.5268348434496095&>. The page title is "fury.network". Below the title, there is a navigation bar with tabs: Result, HTML, JavaScript (which is selected), and another tab that is partially visible. The main content area contains a large amount of code, likely a client-side script, which is color-coded for syntax highlighting. The code includes several functions, loops, and conditional statements, primarily using jQuery and the StromDAO Node.js library.

```
1 $qparams = function(name){  
2     var results = new RegExp('[\?&]' + name + '=([^#]*?)').exec(window.location.href);  
3     if (results==null){  
4         return null;  
5     }  
6     else{  
7         return decodeURI(results[1]) || 0;  
8     }  
9 }  
10  
11 var extid='fury.network.'+Math.random();  
12  
13 if($qparams("extid")!=null) {  
14     extid=$qparams("extid");  
15 }  
16 var node = new document.StromDAOBO.Node({external_id:extid,testMode:true,rpc:"https://demo.stromdao.de/rpc"  
17  
18 // Fill View (HTML) using JQuery  
19 $('.account').html(node.wallet.address);  
20  
21  
22 $('#go').on('click',function() {  
23     // Create Instance with external ID 1234 and existing private Key (relink)  
24     $('#go').attr('disabled','disabled');  
25     $('#result').show();  
26  
27     node.mpr().then(function(mpr) {  
28         mpr.storeReading($('#reading').val()).then(function(o) {  
29             console.log(o);  
30             $('#go').removeAttr('disabled');  
31         });  
32     });  
33 });  
34  
35 node.mpr().then(function(mpr) {  
36     mpr.readings(node.wallet.address).then(function(o) {  
37         $('#result').text(o);  
38     });  
39 });  
40  
41 // Load View (HTML) using JQuery  
42 $('#view').load('view.html');
```

# Programmatische Messdatenverwaltung mit 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>

# Installation

```
$ npm i -g stromdao-bo-mpo
```

# Help

```
$ stromdao help
```

```
Stefans-MacBook-Pro:fury-training smuemd$ stromdao help
```

## Commands:

help [command...]	Provides help for a command.
exit	Exits application.
store [options] <meter_point_id> [reading]	Stores Meter Point
retrieve <meter_point_id>	Retrieves Meter Point
account [options] <meter_point_id>	Get Address and keys
credit <meter_point_id> <amount>	Add credit to Meter Point
receipts <filename>	Exports transaction history
balancing [options] <meter_point_id>	(Sub) Balance Group
tokenize [options] <meter_point_id>	Derive digital asset
cutokenize [options] <meter_point_id>	Derive digital utility
set [options] <meter_point_id>	Create and link a offset
open [options] <meter_point_id>	Opens Webbrowser with URL
discovergy [options] <meter_point_id>	Links Meter Point to a device
infrastructure <infrastructure_node> <meter_point>	Assigns Infrastructure
list	List of managed meter points
httpservice	Start Lacy Webservice
webuser [options] <meter_point_id>	Create a new webuser
backup <zipfilename>	Exports local storage

```
Stefans-MacBook-Pro:fury-training smuemd$ █
```

# Aufgabe 1: Hallo Welt

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

# Aufgabe 1: Hallo Welt

```
$ stromdao store Z01 1234
```

```
Stefans-MacBook-Pro:fury-training smuemd$ stromdao store Z01 1234
Mandated Z01 0x915198b602bac7d807247e0655f2559c72111c7305bc0d2aca8a9af13b64808c
TX: 0x06209e2942d2ce9047c9dcdf74cad1ae4e5db6c7bb8780abd0a2794ae22eb166
```

Für den Zähler mit der Kennung Z01 wurde der  
Zählerstand 1234 in der Transaktion 0x06209...  
geschrieben.

# Aufgabe 1: Hallo Welt

```
$ stromdao account Z01
```

Netzwerkadresse des Zählers abfragen

```
Stefans-MacBook-Pro:fury-training smuemd$ stromdao account Z01
MPIID Z01
Address 0x79cF78D101e9EEB93D5AC527c3c845967782D5e3
Node 0xb57ea39ccCD9837CA5b7600aa9017c3385A8fb75
BLK 0x6a1A548Cf7e3a2713080f5f5E0fd533bA954665c
Private Key PKI 0xc0ba74ff7b19c16643bd45d1bd56e0e344e4d29c00d274120
RSA Public Key -----BEGIN PUBLIC KEY-----
MTGfMA0GCSqGSTb3D0EBAQIAA4GNADCBi0KBgQCYgwA1f6BnRPxmZdbrrXTalhpe
```

# Aufgabe 1: Hallo Welt

Abrufen des unter der Netzwerkadresse gespeicherten Zählerstandes auf <https://fury.network>

[https://fury.network/?base=sample\\_mpo](https://fury.network/?base=sample_mpo)

The screenshot shows a web browser window with the URL <https://fury.network/?extId=fury.guest.0.560333906958806&base=sample...>. The page is titled "fury.network" and features three main sections: "Store Meter Reading", "Get Last Reading", and "Get Historic Reading".

- Store Meter Reading:** A form with a text input containing "0" and a red "Go" button.
- Get Last Reading:** A form with a text input containing "0x79cF78D101e9EEB93D5AC527c3c845967782D6e3" and a red "Go" button. Below it, the "Time" is listed as "20/06/2018, 17:26:09" and the "Reading" as "1234".
- Get Historic Reading:** A form with a text input containing "0x79cF78D101e9EEB93D5AC527c3c845967782D6e3" and a red "Go" button. Below it, the "Block #" is listed as "226338" and the "Reading" as "1234".

The browser's address bar shows the full URL, and the top right corner displays the STROMDAO logo with the tagline "digital energy infrastructure for tomorrow".

# Aufgabe 1: Hallo Welt

\$ stromdao retrieve Z01

Zählerstand für den Zähler mit der Kennung Z01 per Kommandozeile abfragen

```
Stefans-MacBook-Pro:fury-training smuemd$ stromdao retrieve Z01
Time: 2018-6-20 17:36:09
Reading: 1234
[ BigNumber { _bn: <BN: 5b2a7469> },
  BigNumber { _bn: <BN: 4d2> },
  time: '1529508969',
  power: '1234' ]
```

# Lernziel

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

# Lokale Entwicklung mit der Fury IDE

Schnelles testen und verproben von Anwender-  
Interaktionen.

# Installation Fury IDE (lokal)

```
$ npm i -g fury.network
```

# Help

```
$ fury help
```

```
Stefans-MacBook-Pro:fury-training smuemd$ 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 (Sto
publish <path>	Publish given subfolder to fury.network
auth <path>	

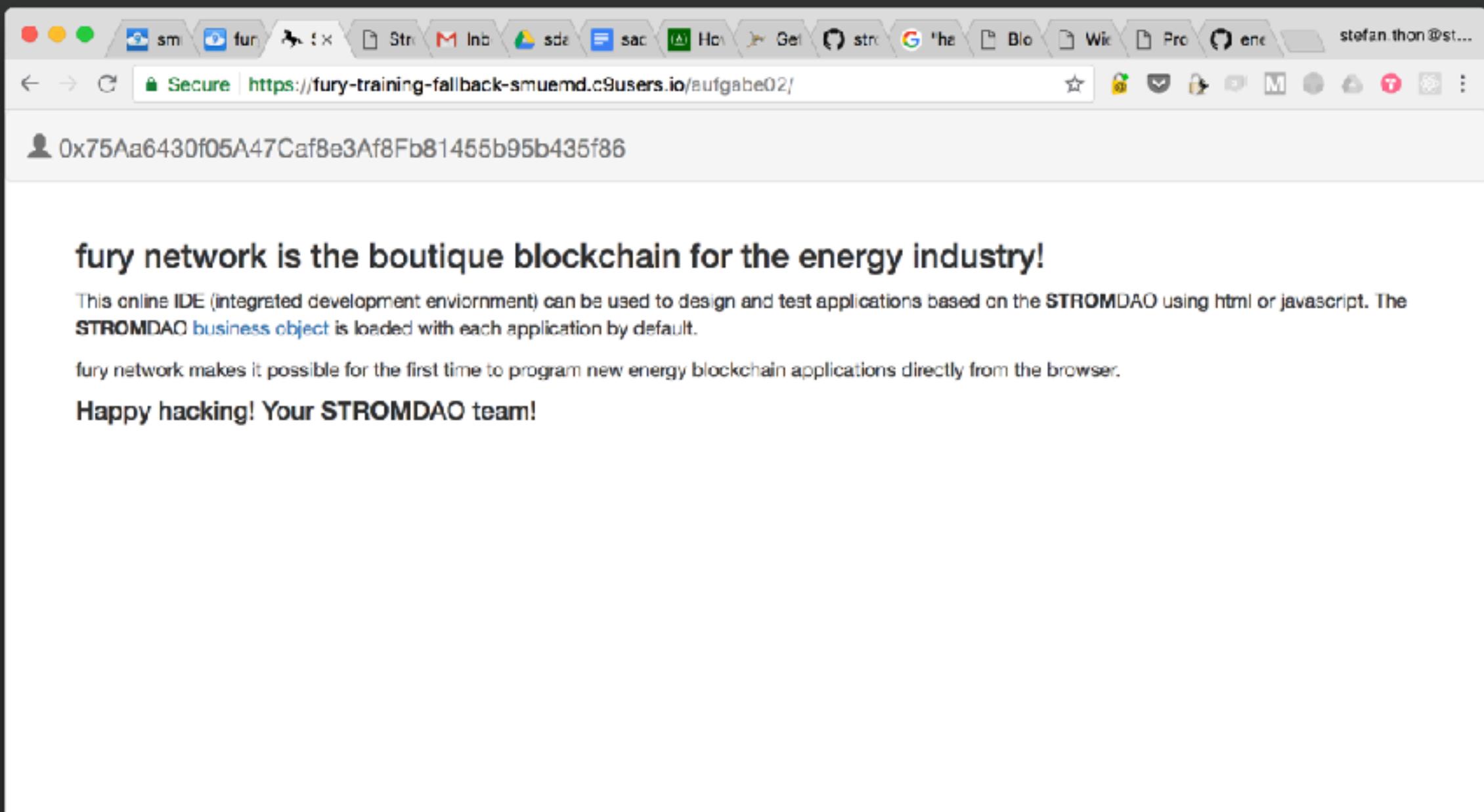
```
Stefans-MacBook-Pro:fury-training smuemd$ █
```

# Initialisierung

```
$ cd public  
$ fury init aufgabe02
```

```
Stefans-MacBook-Pro:fury-training smuemd$ cd public/  
Stefans-MacBook-Pro:public smuemd$ fury init aufgabe02  
Swarm listening on /ip4/127.0.0.1/tcp/4003/ws/ipfs/QmRjZCAhMWJQddPFZgu1WAoA3grf4HFiK1NtLVfTcprVz  
Swarm listening on /ip4/127.0.0.1/tcp/4002/ipfs/QmRjZCAhMWJQddPFZgu1WAoA3grf4HFiK1NtLVfTcprVz4  
Swarm listening on /ip4/192.168.178.134/tcp/4002/ipfs/QmRjZCAhMWJQddPFZgu1WAoA3grf4HFiK1NtLVfTcp  
aufgabe02 created  
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjB4NTk0M2E2Mjc5QjNCZEU3M2IwZGFkZjUxMzYzMjM0RjMDJ  
Stefans-MacBook-Pro:public smuemd$ █
```

# Open [...].c9users.io/aufgabe02/

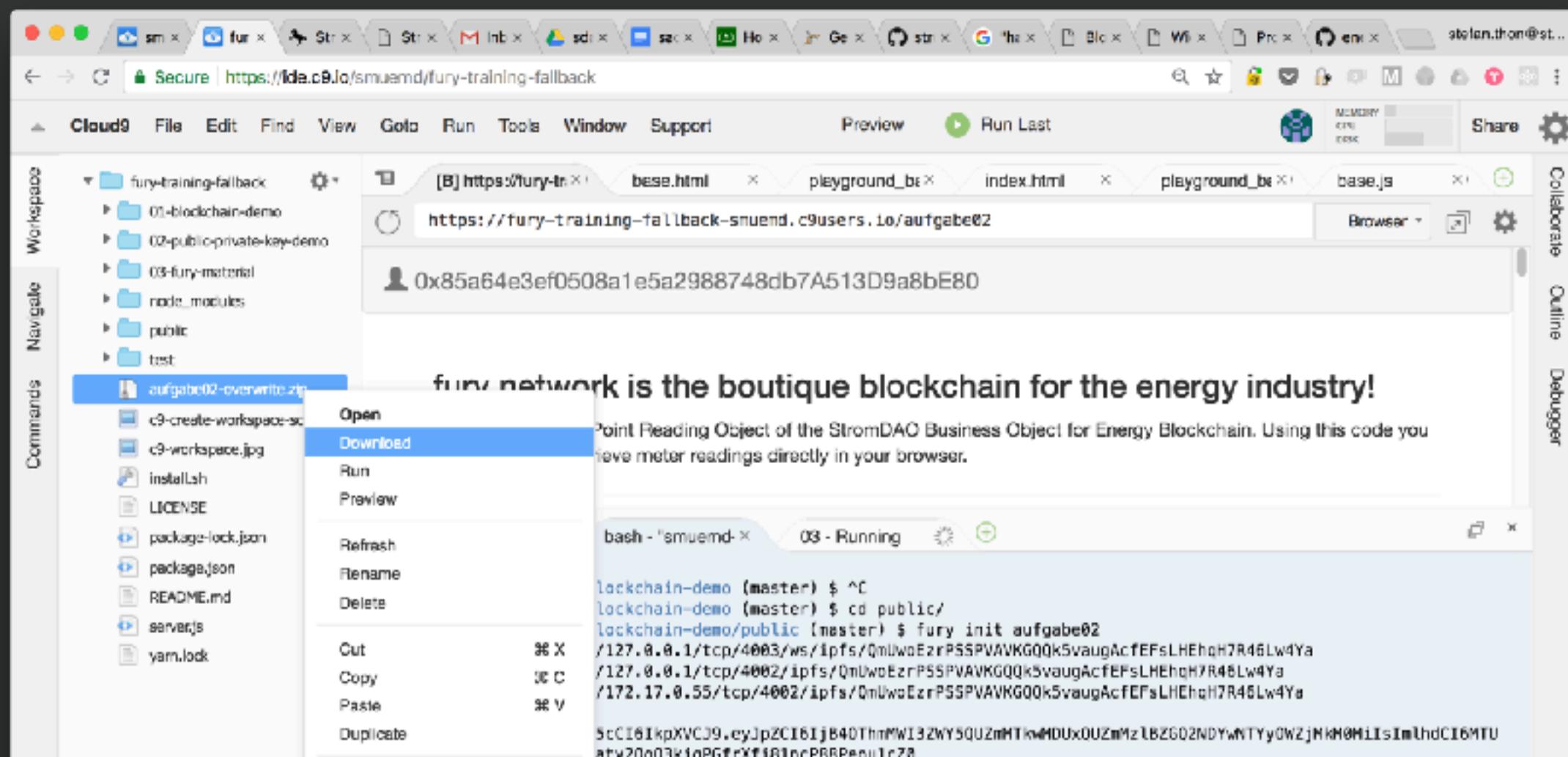


# Aufgabe 2: Entwicklung lokaler Programme

1. ZIP Datei **aufgabe02-overwrite.zip** herunterladen und entpacken
2. Überschreiben der beiden Dateien in **public/aufgabe02**
3. Reload der Seite im Browser

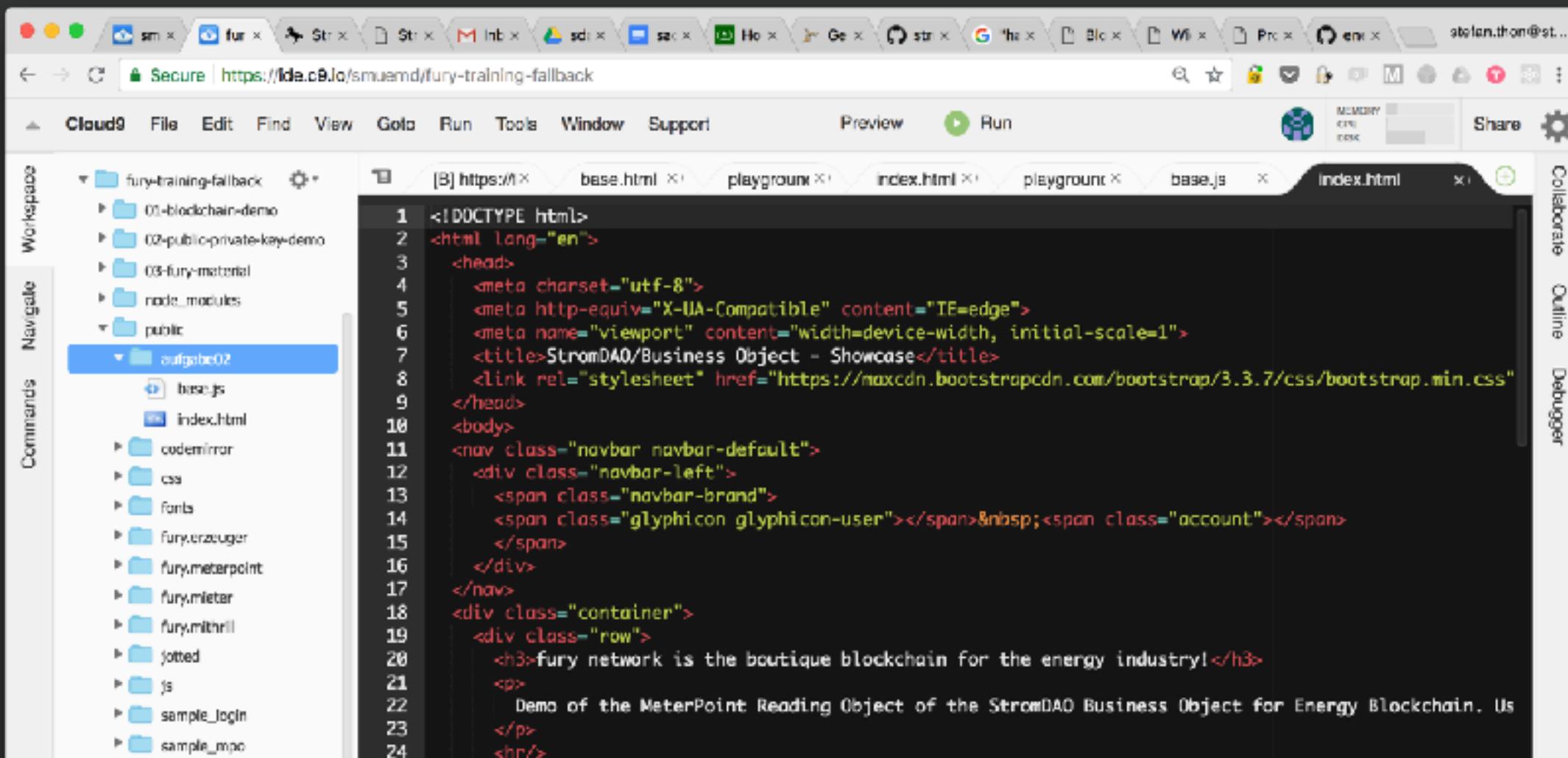
# Aufgabe 2: Entwicklung lokaler Programme

ZIP Datei aufgabe02-overwrite.zip



# Aufgabe 2: Entwicklung lokaler Programme

Überschreiben der beiden Dateien in public/aufgabe02



```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>StromDAO/Business Object - Showcase</title>
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
</head>
<body>
<nav class="navbar navbar-default">
<div class="navbar-left">
<span class="navbar-brand">
<span class="glyphicon glyphicon-user"></span>&ampnbsp<span class="account"></span>
</span>
</div>
</nav>
<div class="container">
<div class="row">
<h3>fury network is the boutique blockchain for the energy industry!</h3>
<p>
    Demo of the MeterPoint Reading Object of the StromDAO Business Object for Energy Blockchain. Us
</p>
<hr/>
```

# Aufgabe 2: Entwicklung lokaler Programme

Reload der Seite im Browser auf [...].c9users.io/aufgabe02/

The screenshot shows a web browser window with the following details:

- Address Bar:** https://fury-training-fallback-smuemd.c9users.io/aufgabe02/
- Content Area:**
  - A header section displays a placeholder BC Address: 0x1500Bc71833b483B7DF903Be6Cc256CfB231f3f9.
  - A main message: "fury network is the boutique blockchain for the energy industry!" followed by a descriptive text about MeterPoint Reading Object.
  - An input field labeled "BC Address" with a placeholder "0x..." and a red "Lookup" button.
  - A "Response" section below the input field.
  - An "Effective Code" section at the bottom containing a snippet of JavaScript code.

# Integrationsbibliothek Business Object

Abstraktionsschicht zur Fury Blockchain für die Umsetzung energiewirtschaftlicher Prozesse.

# Integrationsbibliothek Business Object

Das Business Objekt wird mit *node* angesteuert.

Es ist bereits initialisiert, so dass jede Instanz eine eigene Adresse und einen eigenen privaten Schlüssel besitzt.

```
12
13     var extid = "fury.network." + Math.random();
14     if ($.qparams("extid") != null) {
15         extid = $.qparams("extid");
16     }
17     var node = new document.StromDAOBO.Node({
18         external_id: extid,
19         testMode: true,
20         rpc: "https://demo.stromdao.de/rpc",
21         abilocation: "https://cdn.rawgit.com/energychain/StromDAO-BusinessObj/0.1.0/dist/StromDAOBusinessObj.js"
22     });
23
24     // Fill View (HTML) using JQuery
25     $('.account').html(node.wallet.address);
26     document.go = function () {
27
28         node.mpr().then(function (mpr) {
29             mpr.readings($('#address').val()).then(function (reading) {
30                 console.info('reading: ', { time: reading.time.toNumber(),
31                     var html = "<ul>";
32                     if (typeof reading == "object") {
33                         for (var property in reading) {
34                             if (isNaN(property)) {
35                                 if (property != "length") {
36                                     html += '<li><strong>' + property + '</strong></li>';
37                                 }
38                             if (property == "power") {
39                                 $('#reading').val(reading[property]);
40                             }
41                         }
42                     }
43                 }
44                 html += "</ul>";
45                 $('#result').html(html);
46             });
47         });
48     }
49
50
51
52
```

# Integrationsbibliothek Business Object

Im Objekt gibt es verschiedene Domains.

Hier wird Meter-Point-Reading (*mpr*) verwendet.  
Eine Domain entspricht meist sogenannten  
Smart-Contracts in der Blockchain.

*readings* einem Methodenaufruf im Smart Contract.

[https://github.com/energychain/StromDAO-BusinessObject/blob/master/smart\\_contracts/StromDAO-B0.sol#L118-L142](https://github.com/energychain/StromDAO-BusinessObject/blob/master/smart_contracts/StromDAO-B0.sol#L118-L142)

```
node.mpr().then(function (mpr) {
  mpr.readings($('#address').val()).then(function (reading) {
    console.info('reading: ', { time: reading.time.toNumber(), power
  });
});
```

# Stromkonto

Smart-Contract für Bilanzkreisführung und  
Buchhaltung.

# Stromkonto

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

<https://demo.stromdao.de/introspect.html>

The screenshot shows a web-based electricity account statement. At the top, it displays the account number "0x0b155Ca8919D80138f02D3E9A8256029B263eD46". Below this, the title "Stromkonto Abrechnung" is visible. A summary table provides the following data:

Energie	Arbeitspreis	Geld
-1,183.561 KWh	22.5000 €/KWh	-5.26 €

Below the summary, there is a button labeled "Salden Übertrag an 0x.. Freigeben". The main section, titled "Umsätze", lists seven transactions from June 2018, all to "STROMDAO" at 14.794 KWh and 0.07 €. The transactions are dated 14/06/2018 to 20/06/2018 at times 05:02:44 to 05:02:55.

Konsens	Von	An	Energie	Geld
20/06/2018, 05:02:54	0x4a1018fa6a091699181352d8ba4f1930277a726f	STROMDAO	14.794 KWh	0.07 €
19/06/2018, 05:02:44	0x4a1018fa6a091699181352d8ba4f1930277a726f	STROMDAO	14.794 KWh	0.07 €
18/06/2018, 05:02:55	0x4a1018fa6a091699181352d8ba4f1930277a726f	STROMDAO	14.795 KWh	0.07 €
17/06/2018, 05:02:54	0x4a1018fa6a091699181352d8ba4f1930277a726f	STROMDAO	14.794 KWh	0.07 €
16/06/2018, 05:02:44	0x4a1018fa6a091699181352d8ba4f1930277a726f	STROMDAO	14.795 KWh	0.07 €
15/06/2018, 05:02:44	0x4a1018fa6a091699181352d8ba4f1930277a726f	STROMDAO	14.795 KWh	0.07 €
14/06/2018, 05:02:44	0x4a1018fa6a091699181352d8ba4f1930277a726f	STROMDAO	14.794 KWh	0.07 €

# Aufgabe 3: Verbrauchsbuchung

1. Durchführung der Zählerablesungen.
2. Öffnen einer Fury Instanz für das Stromkonto
3. Kontrolle der Abrechnung.

# Zählerablesung

```
$ stromdao store --workprice 2500000 z01 2234
```

```
Stefans-MacBook-Pro:fury-training smuemd$ stromdao store —workprice 2500000 z01 2234
```

```
AddTx 0x79cF78D101e9EEB93D5AC527c3c845967782D5e3 0x6a1A548Cf7e3a2713080f5f5E0fd533bA954665c 2500000 1000 0x4880bb93a392e2e426
```

```
Stefans-MacBook-Pro:fury-training smuemd$
```

# Öffnen des Stromkontos

```
$ stromdao open -b -p Z01
```

```
Stefans-MacBook-Pro:fury-training smuemd$ stromdao open -b -p Z01
```

```
https://www.stromkonto.net/?sc=0x3d45647c5DE276E71404B077545c42eF5Deaa329&account=0x79cF78D101e9EEB93D5AC527c3c8
```

```
Stefans-MacBook-Pro:fury-training smuemd$ █
```

# Kontrolle der Abrechnung

The screenshot shows a web browser window with a title bar containing various icons and tabs. The main content is a electricity bill statement titled "Stromkonto Abrechnung".

**Stromkonto Abrechnung**

0x79cF78D101e9EEB93D5AC527c3c845967782D5e3

Energie	Arbeitspreis	Geld
-1.000 KWh	0.2500 €/KWh	-0.25 €

**Umsätze**

Konsens	Von	An	Energie	Geld
20/06/2018, 23:54:44	0x79cf78d101e9eeb93d5ac527c3c845967782d5e3	0x6a1a548cf7e3a2713080f5f5e0fd533ba954665c	1.000 KWh	0.25 €

**Anfangssaldo**

Konsens: 225451 - (20/06/2018, 23:54:44)

# Lernziel

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

# Tarifentwicklung

Mit den gezeigten Werkzeugen lassen sich Stromtarife implementieren.

# Settlement Objekt

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

```
node.stromkontoproxy('0x3d45647c5DE276E71404B077545c42eF5Deaa329
  .then(sko => {
    return sko.addTx('von', 'an', 'kosten', 'energiemenge')
  })
  .then(tx => {
    console.info('tx number: ', tx)
  })
  .catch(err => console.error(err))
```

# Settlement Objekt

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

```
Settlement: { tarif:  
  { '196c1b596da6f09e8576bd0ab0a6f3b7':  
    { GeoKey: '196c1b596da6f09e8576bd0ab0a6f3b7',  
      Zipcode: '39110',  
      City: 'Magdeburg',  
      District: '',  
      CustomCode: '',  
      UsageStart: 1,  
      UsageEnd: 100000,  
      BpNet: 88.46,  
      BpGross: 105.26739999999998,  
      BpMonthlyNet: 7.3717,  
      BpMonthlyGross: 8.7723,  
      UpNet: 21.949,  
      UpGross: 26.11931,  
      UphNet: 0,  
      UphGross: 0,  
      BoNet: 0,  
      BoGross: 0,  
      BoNewNet: 0,  
      BoNewGross: 0,  
      BoInstantNet: 0,  
      BoInstantGross: 0,  
      BoPercent: 0,  
      Net: '39106001',  
      Additions: '',  
      Total: 746.9300000000001,  
      TotalWoBonus: 888.8464,  
      TotalNet: 746.93,  
      TotalNetWoBonus: 746.93,  
      TotalGross: 888.8464,  
      TotalGrossNoBonus: 888.8464 } },  
    zipcode: '39110',  
    BpGross: 1052673999.9999998,  
    UpGross: 2611931,  
    account: '0x79cF78D101e9EEB93D5AC527c3c845967782D5e3',  
    node_account: '0x6a1A548Cf7e3a2713080f5f5E0fd533b4954665c',  
    node_wallet: '0xb57ea39cc009837CA5b7600aa9017c3385A8fb75',  
    start:  
    [ BigNumber { _bn: <BN: 5b2acd1f> },  
      BigNumber { _bn: <BN: 8ba> },  
      time: BigNumber { _bn: <BN: 5b2acd1f> },  
      power: BigNumber { _bn: <BN: 8ba> } ],  
    end:  
    [ BigNumber { _bn: <BN: 5b2ad6b1> },  
      BigNumber { _bn: <BN: ca2> },  
      time: BigNumber { _bn: <BN: 5b2ad6b1> },  
      power: BigNumber { _bn: <BN: ca2> } ],  
    cost: 2693712,  
    base: 1000 }
```

# Settlement Datei anlegen

```
$ touch settlement_out.js
```

```
// TODO settlement_out.js content:  
console.log("Node: ", node.wallet.address);  
console.log("Settlement: ", settlement);  
|
```

## Settlement Datei aufrufen

```
$ stromdao store -f settlement_out.js --de 39110 Z01 3234
```

```
Stefans-MacBook-Pro:fury-training smuemd$ stromdao store -f settlement_out-sample.js --de 39110 Z01 3234
```

```
Node: 0x79cF78D101e9EEB93D5AC527c3c845967782D5e3
```

```
Settlement: { tarif:  
  { '196c1b596da6f09e8576bd0ab0a6f3b7':  
    { GeoKey: '196c1b596da6f09e8576bd0ab0a6f3b7',  
      Zipcode: '39110',  
      City: 'Magdeburg',  
      District: '',  
      CustomCode: '',  
      UsageStart: 1,
```

# Weitere Settlement Datei anlegen

```
$ touch settlement_simple.js
```

```
// TODO settlement_simple.js content:  
  
global.promise = new Promise((res, rej) => {  
    // settlement.cost=10000000; // =1,00 EUR  
    node.stromkontoproxy(global.smart_contract_stromkonto)  
        .then(sko => sko.addTx(settlement.account, settlement.node_account, settlement.cost, settlement.base))  
        .then(tx => {  
            console.info("TX", tx);  
            console.info("From:", settlement.account);  
            console.info("To:", settlement.node_account);  
            console.info("Amount:", (Math.round(settlement.cost) / 10000000).toFixed(6));  
            console.info("Base:", (settlement.end.power.toString() * 1 - settlement.start.power.toString() * 1));  
            res(tx)  
        })  
        .catch(err => rej(err))  
})
```

## Settlement\_simple Datei aufrufen

```
$ stromdao store -f store -f settlement_simple.js  
--de 69256 Z01 4234
```

```
Stefans-MacBook-Pro:fury-training smuemd$ stromdao store -f settlement_simple-sample.js --de 69256 Z01 4234  
TX 0x2d437784107773a3f973f4f5c161afd710cf849bf886ed82c4a3610286b3b23a  
From: 0x79cF78D101e9EEB93D5AC527c3c845967782D5e3  
To: 0x6a1A548Cf7e3a2713080f5f5E0fd533bA954665c  
Amount: 0.257082  
Base: 1000  
Stefans-MacBook-Pro:fury-training smuemd$ █
```

```
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 der Tarifentwicklung muss der *settlement.cost* entsprechend modifiziert werden
- Im Settlement Object sind alle Daten vorhanden, die zur Berechnung notwendig sind
- Es steht hierzu eine Javascript Runtime zur Verfügung
- Auf das Business Objekt kann zugegriffen werden

# Code Beispiel

Es sind auch komplexere Tarif-Implementierungen möglich.

# Aufteilung der Kosten auf mehrere Konten (WG Strom)

```
// WG STROM

let konto1="0x6fCb57AC03CdfA7E26bc6117ceCa83b1516Af2f7";
let konto2="0x94013490Cc86ada8D9154B085D791176F676283C";
let konto3="0x6692BC4915Ba6f07bB309481d2198C5C25e3Fcce";

settlement.cost=Math.round(settlement.cost/3)
global.promise = new Promise(function(resolve2, reject2) {
    node.stromkontoproxy("0x19BF166624F485f191d82900a5B7bc22Be569895")
        .then(sko => {
            return sko.addTx(konto1, node.nodeWallet.address, settlement.cost, s
                .then(tx => sko.addTx(konto2, node.nodeWallet.address, settlement.
                .then(tx => sko.addTx(konto3, node.nodeWallet.address, settlement.
                .then(tx => resolve2(tx))
            })
        .catch(err => reject2(err))
})
```

# 1 % Provisionszahlung an Vertriebspartner

```
// Provision

let vpkonto="0x6fCb57AC03CdfA7E26bc6117ceCa83b1516Af2f7";

global.promise = new Promise(function(resolve2, reject2) {
  node.stromkontoproxy("0x19BF166624F485f191d82900a5B7bc22Be569895")
    .then(sko => sko.addTx(node.wallet.address, node.nodewallet.address, settlement.cost, settlement.base)
      .then(tx => sko.addTx(node.nodewallet.address, vpkonto, Math.round(settlement.UpGross*0.01), settlement.base)
        .then(tx =>
          resolve2(tx))
        )
      .catch(err => reject2(err))
  })
})
```

# Transfer von 6,880 Ct / kWh auf EEG Konto

```
// EEG Buchung

let eegkonto="0x6fCb57AC03CdfA7E26bc6117ceCa83b1516Af2f7";

global.promise = new Promise(function(resolve2, reject2) {
  node.stromkontoproxy("0x19BF166624F485f191d82900a5B7bc22Be569895")
    .then(sko => sko.addTx(node.wallet.address, node.nodeWallet.address, settlement.cost, settlement.base)
      .then(tx => sko.addTx(node.nodeWallet.address,eegkonto,Math.round(settlement.base*688000),settlement.base)
        .then(tx =>
          resolve2(tx))
    )
    .catch(err => reject2(err))
})
```

# Weitere Beispiele

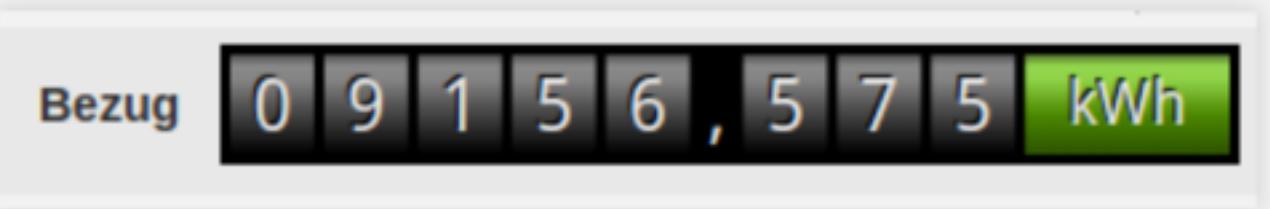
- Strompreis dynamisch nach Verfügbarkeit von Grünstrom  
(vgl. Autostrom, Grünstromtoken)
- Happy-Hour: Jeden Abend um 17:00 Uhr Strom zum halben Preis  
(vgl. settlement\_happyHour.js)
- Stunde Strom geschenkt
- Jede Kilo-Wattstunde ist ein Token (Beteiligung) an einer Anlage  
(vgl. Apex, Microgrid CUX, public/fury.mieter, )

# **Programmierter Stromtarif**



## Stromzähler

- Grundlage eines Vertrages
- Mengenermittlung



## Stromtarif

- Bilanzierung
- Abrechnung von Energiemengen



# Settlement

- Regelt, was mit dem Delta zwischen zwei Ablesungen passiert
- Basiert auf vorher festgelegter Regel.

Bezug

0 9 1 5 6 , 5 6 0 kWh

Bezug

0 9 1 5 6 , 5 7 5 kWh

```
Stefans-MacBook-Pro:fury-training smuemd$ stromdao store -f settlement_happyHour.js --de 39110 DEMO 1956560
Mandated DEMO 0x03fb0e2e62ad8d76edb37f3e65acb9dd235548ea894513ca7609cd505f94d195
TX: 0x71fc06367d04a618baf8f496f7293e42de841b2074341b5ce0e177606cc3a59c
Stefans-MacBook-Pro:fury-training smuemd$ stromdao store -f settlement_happyHour.js --de 39110 DEMO 1956575
TX 0x2965b2e051ded8c90afaf3412a1b5939b47ecdf5f79c5b6325a15adf24ce2971
From: 0x8f1B225419F8C97442b4652345742BE0Ef4ABed1
To: 0xb57ea39ccCD9837CA5b7600aa9017c3385A8fb75
Amount: 0.004172
Base: 15
Stromkonto: 0x3d45647c5DE276E71404B077545c42eF5Deaa329
Stefans-MacBook-Pro:fury-training smuemd$ stromdao open -b -p DEMO
https://www.stromkonto.net/?sc=0x3d45647c5DE276E71404B077545c42eF5Deaa329&account=0x8f1B225419F8C97442b4652345742BE0Ef4ABed1
Stefans-MacBook-Pro:fury-training smuemd$
```

Bezug

0 | 9 | 1 | 5 | 6 , 5 | 6 | 0 kWh

Bezug

0 | 9 | 1 | 5 | 6 , 5 | 7 | 5 kWh

# Programmierung

```
global.promise = new Promise(function(resolve2, reject2) {
node.stromkontoproxy(global.smart_contract_stromkonto)
.then(sko => {
  return sko.addTx(node.wallet.address, node.nodewallet.address, settlement.cost, settlement.base)
})
.then(function(tx) {
  console.log("TX", tx);
  console.log("From:", node.wallet.address);
  console.log("To:", node.nodewallet.address);
  console.log("Amount:", (Math.round(settlement.cost) / 1000000).toFixed(6));
  console.log("Base:", (settlement.end.power.toString() * 1 - settlement.start.power.toString() * 1));
  console.log('Stromkonto: ', global.smart_contract_stromkonto)
  resolve2(tx);
})
.catch(err => reject2(err))
})
```

Bezug 0 | 9 | 1 | 5 | 6 , 5 | 6 | 0 kWh

Bezug 0 | 9 | 1 | 5 | 6 , 5 | 7 | 5 kWh

# Programmierung

```
/*
 * Happy Hour
 * =====
 * 11:00-12:00
 */

let startTime=new Date(settlement.start.time*1000); // time 1st reading
let endTime=new Date(settlement.end.time*1000); // time 2nd reading

console.log("Hours", startTime.getHours(), endTime.getHours());

if(endTime - startTime < 3600000) { // 1 hr
  if((startTime.getHours() === 10) && (endTime.getHours() === 10)) { // both in hour 10
    settlement.cost=0;
  }
}
```

Bezug

0 | 9 | 1 | 5 | 6 , 5 | 6 | 0 kWh

Bezug

0 | 9 | 1 | 5 | 6 , 5 | 7 | 5 kWh

```
Stefans-MacBook-Pro:fury-training smuemd$ stromdao store -f settlement_happyHour.js --de 39110 DEMO 2966575
Hours 3 3
TX 0x287ca272e7cdf3dde66c97324cdf6612b42ba177af61067168e07c02fbcd2d43
From: 0x8f1B225419F8C97442b4652345742BE0Ef4ABed1
To: 0xb57ea39ccCD9837CA5b7600aa9017c3385A8fb75
Amount: 0.000000
Base: 1000000
Stromkonto: 0x3d45647c5DE276E71404B077545c42eF5Deaa329
Stefans-MacBook-Pro:fury-training smuemd$
```

Secure | <https://www.stromkonto.net/?sc=0x3d45647c5DE276E71404B077545c42eF5Deaa329&account=0x8f1...>

0x0b155Ca8919D80138f02D3E9A8256029B263eD46 Profil Ansicht

## Stromkonto Abrechnung

0x8f1B225419F8C97442b4652345742BE0Ef4ABed1

Energie	Arbeitspreis	Geld
-1,010.015 KWh	0.0026 €/KWh	-2.62 €

### Umsätze

Konsens	Von	An	Energie	Geld
21/06/2018, 03:12:39	0x8f1b225419f8c97442b4652345742be0ef4abad1	0xb57ea39cccd9837ca5b7600aa9017c3385a8fb75	1,000.000 KWh	0.00 €
21/06/2018, 03:10:54	0x8f1b225419f8c97442b4652345742be0ef4abad1	0xb57ea39cccd9837ca5b7600aa9017c3385a8fb75	10.000 KWh	2.62 €
21/06/2018, 02:54:44	0x8f1b225419f8c97442b4652345742be0ef4abad1	0xb57ea39cccd9837ca5b7600aa9017c3385a8fb75	0.015 KWh	0.00 €
Anfangssaldo 0.00 €				

[kontakt@stromdao.com](mailto:kontakt@stromdao.com)

# Title dark

## Subtitle dark