

# **PROTOTYPE: ENERGY BLOCKCHAIN TRADING**

System Architecture and User Interface

**2018W 053235-1 Anwendungen aus semantischen  
Technologien**

a00851190 - Ivan Igic

a01026759 - Christian Dienbauer

a01251567 - Felix Fally

# Contents

<b>General Description</b>	<b>2</b>
Domain	2
Scenarios of the domain	2
Description of the system	3
<b>Use Case View</b>	<b>4</b>
<b>Logical View</b>	<b>7</b>
Class Diagram	7
Description	7
<b>Process View</b>	<b>9</b>
Activity Diagrams	9
<b>Development View</b>	<b>10</b>
Component Diagram	10
Description	10
Web Browser	10
<b>Physical View</b>	<b>12</b>
Deployment Diagram	12
Description	12
<b>ADOxx Metamodell</b>	<b>13</b>
Description of the domain-specific modelling language	13
Classes	13
Relation Classes	13
Classes: Notation and Description	14
Relation Classes: Notation and Description	18
<b>User Interface</b>	<b>21</b>
Mockup Webpage	21
<b>Validation Method: Simulation of the system</b>	<b>22</b>

# 1. General Description

## 1.1. Domain

Due to the recent advances in household-level renewable electricity generation technology, a new type of market based on peer-to-peer (P2P) electricity trading between households will emerge. In this prototype we design a potential technical framework for energy trading on a peer to peer network.

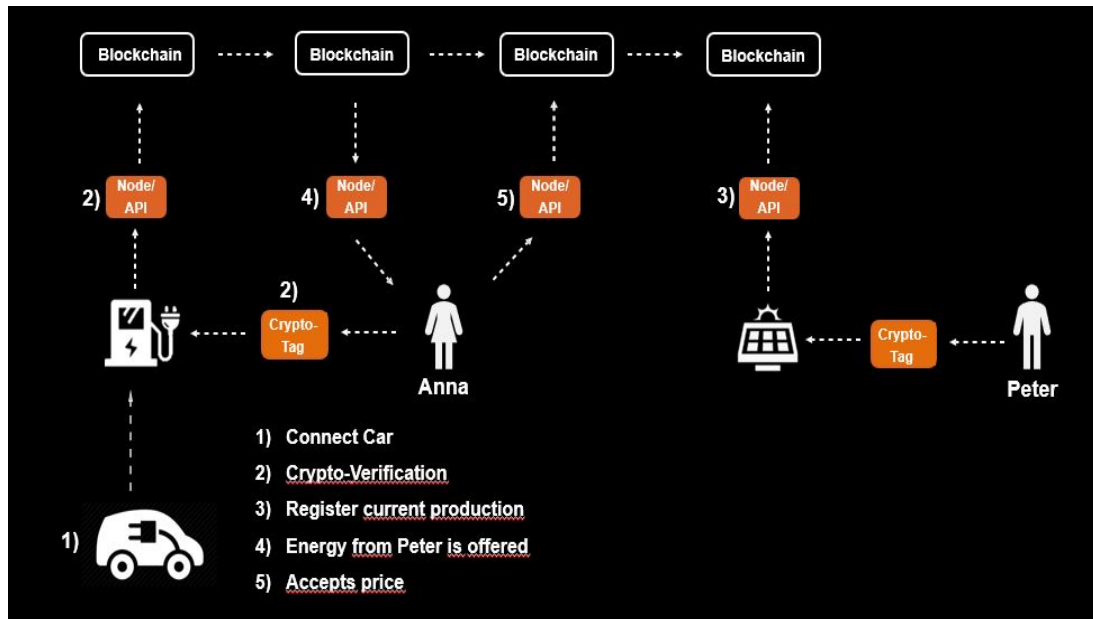
## 1.2. Scenarios of the domain

### Scenario Anna:

Anna needs to charge her electric car. She connects her car with the charging station and identifies herself as a customer. She automatically signs a contract with someone who produces energy. The Charging Station charges the car according to the system. The settlement between Anna and the energy provider happens automatically. So the electric car has been charged eliminating the middleman.

### Scenario Peter:

Peter registers him and his Photovoltaic System on the the system. He agrees to sell energy within the community and to the grid. When someone wants to buy the generated energy the system handles the sale. A contract is signed on the blockchain and the energy is transmitted accordingly. The contract is settled. The produced energy is sold eliminating the middleman.



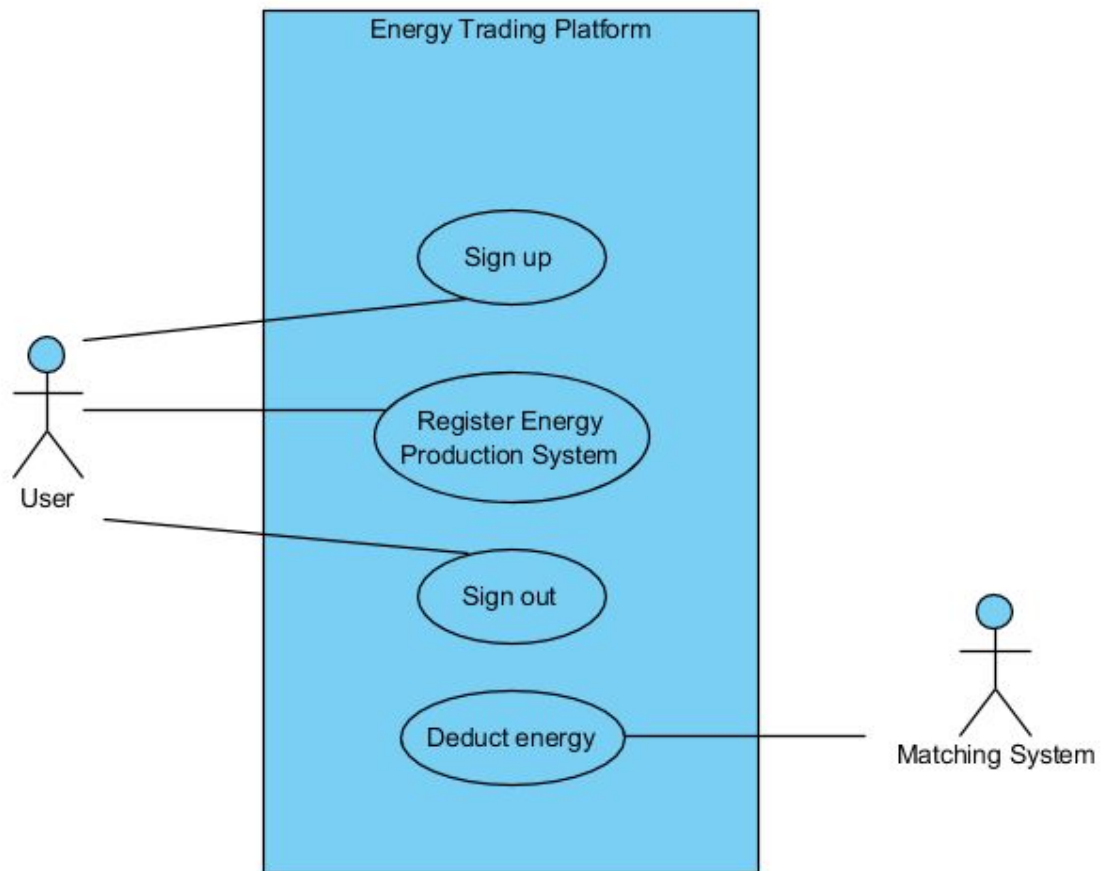
### 1.3. Description of the system

The aim of the project is to implement a simplified peer-to-peer trading system which enables user to trade energy on a private blockchain. The system should simulate the producing, consuming and trading of energy, as it is not possible for us to provide all the different components which exist in real systems, for example photovoltaics, charging stations, crypto tags etc.

The peer-to-peer system should contain following components:

- Web-application: Web-application with user-interface for user-registration and entering (uploading) smart home data
- Ontology: Data structure of the system
- Private Blockchain: Platform for trading and matching prosumers
- ADOxx Metamodel: Metamodel for modelling own smart home model
- Simulation tool: Simulate scenarios for energy consuming, producing and trading

## 2. Use Case View



<b>Use Case:</b>	Sign up
<b>Use Case ID:</b>	UC1
<b>Actor(s):</b>	User, System
<b>Brief description:</b>	A customer inserts his registration data via the web interface and submits.
<b>Pre-conditions:</b>	-
<b>Post-conditions:</b>	The customers data will be placed into the system.
<b>Main Success Scenario:</b>	<ol style="list-style-type: none"><li>1. The customer selects "sign up".</li><li>2. A new order form will be displayed.</li></ol>

	<ol style="list-style-type: none"> <li>3. The user fills blanks in</li> <li>4. The user submits the order form for processing.</li> <li>5. The data is then passed on.</li> </ol>
<b>Extensions:</b>	The process can be cancelled
<b>Priority:</b>	high
<b>Performance Target:</b>	successful registration
<b>Issues:</b>	-

<b>Use Case:</b>	Register Energy Production System
<b>Use Case ID:</b>	UC2
<b>Actor(s):</b>	User, System
<b>Brief description:</b>	A customer inserts his production system data via the web interface and submits.
<b>Pre-conditions:</b>	The user has signed up.
<b>Post-conditions:</b>	The customers data will be placed into the system.
<b>Main Success Scenario:</b>	<ol style="list-style-type: none"> <li>1. The user selects "Register Production System".</li> <li>2. A new order form will be displayed.</li> <li>3. The user enters the "Production System Data"</li> <li>4. The user submits the order form for processing</li> <li>5. The data is then passed on.</li> </ol>
<b>Extensions:</b>	The process can be cancelled

<b>Priority:</b>	high
<b>Performance Target:</b>	successful registration
<b>Issues:</b>	Data can be written into ontology

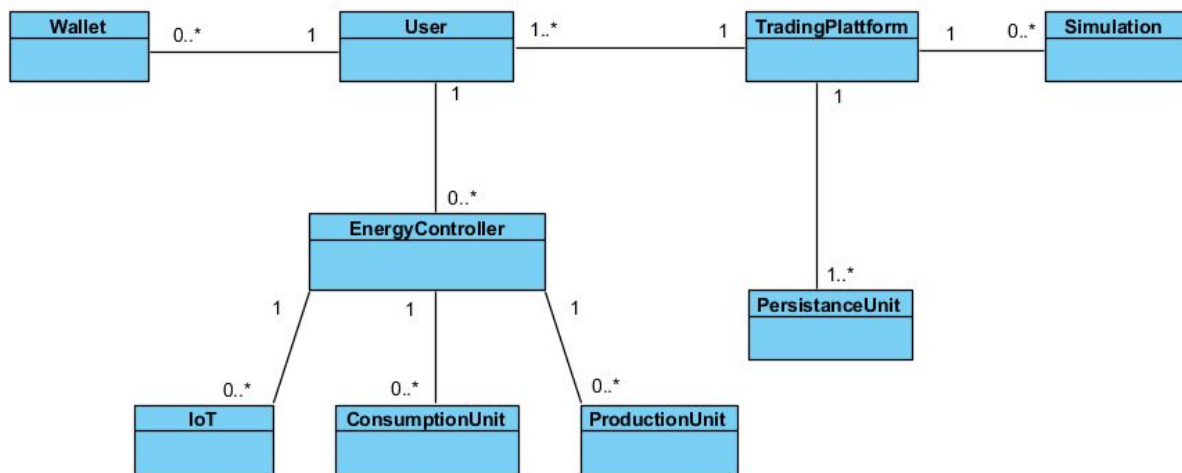
<b>Use Case:</b>	Sign out
<b>Use Case ID:</b>	UC3
<b>Actor(s):</b>	User, System
<b>Brief description:</b>	The customer signs out.
<b>Pre-conditions:</b>	The user has signed up.
<b>Post-conditions:</b>	The customers data will be removed from the system.
<b>Main Success Scenario:</b>	<ol style="list-style-type: none"> <li>1. User logs in</li> <li>2. The user selects "sign out"</li> <li>3. A new window is displayed to confirm the step</li> <li>4. The user submits</li> <li>5. The users data is removed from the system.</li> </ol>
<b>Extensions:</b>	The process can be cancelled
<b>Priority:</b>	high
<b>Performance Target:</b>	successful Data Removal
<b>Issues:</b>	Remove Data from Ontology

<b>Use Case:</b>	Deduct Energy
<b>Use Case ID:</b>	UC4
<b>Actor(s):</b>	Subsystem Matching System
<b>Brief description:</b>	Matching Prosumers on the Trading Platform
<b>Pre-conditions:</b>	User produces energy, another user consumes energy
<b>Post-conditions:</b>	Blockchain transaction is being executed on the platform and energy is being traded

<b>Main Success Scenario:</b>	<ol style="list-style-type: none"> <li>1. Producer produces energy</li> <li>2. Consumer needs energy</li> <li>3. Energy flows from producer to consumer</li> <li>4. Matching algorithm is executed</li> <li>5. Matching is performed</li> <li>6. Blockchain transaction is conducted</li> </ol>
<b>Extensions:</b>	Sell energy for longer period or sell energy contracts
<b>Priority:</b>	high
<b>Performance Target:</b>	Successful energy trading
<b>Issues:</b>	Matching problem

### 3. Logical View

#### 3.1. Class Diagram



#### 3.2. Description

##### **Class: User**

Represents base data of a user like name, age, address, etc.

##### **Class: Wallet**

In order to deal with transactions, a user must have at least one crypto wallet. Data will be stored within this class.



**Class: ConsumptionUnit**

Represents consumption units like a charging station or a washing machine. Information to maximal, current and mean consumption as well as date of register.

**Class: ProductionUnit**

Analog to consumption unit except that there is energy produced.

**Class: IoT**

Describes things like sensor data to get information of the current state of an environment e.g. temperature

**Class: EnergyController**

This class unites the different things of a user and statistics of energy usage and the environment.

**Class: TradingPlattform**

This is the main class of the system which comprises all relevant classes to run the system.

**Class: PersistenceUnit**

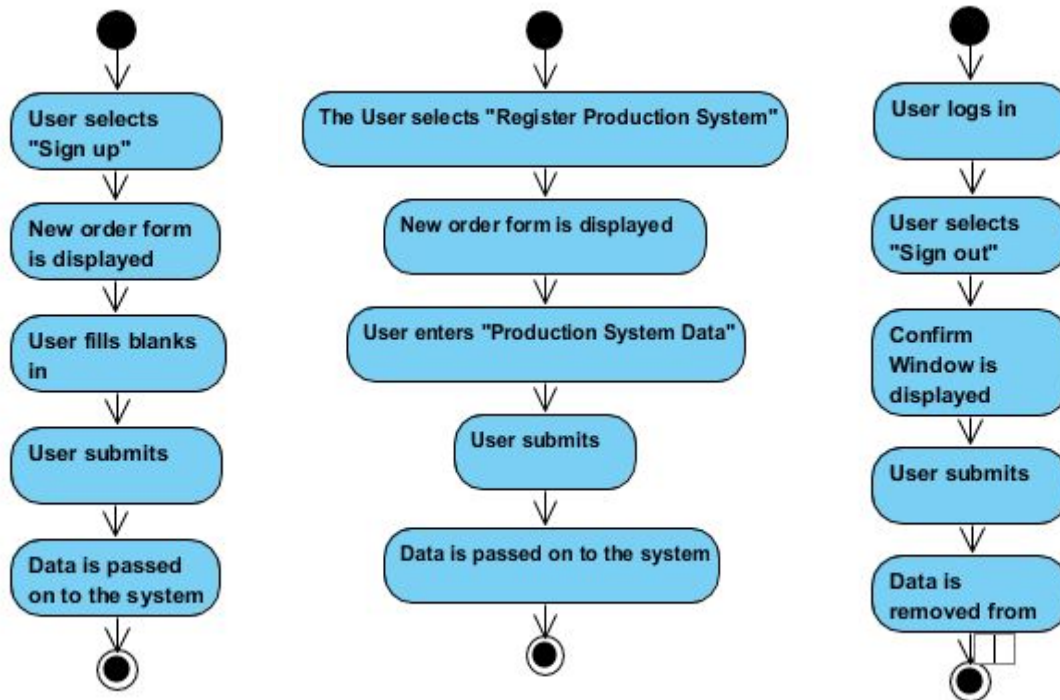
Is used to store and query information of the system by using the Owlready2 library.

**Class: Simulation**

Methods and parameter for system evaluation are defined.

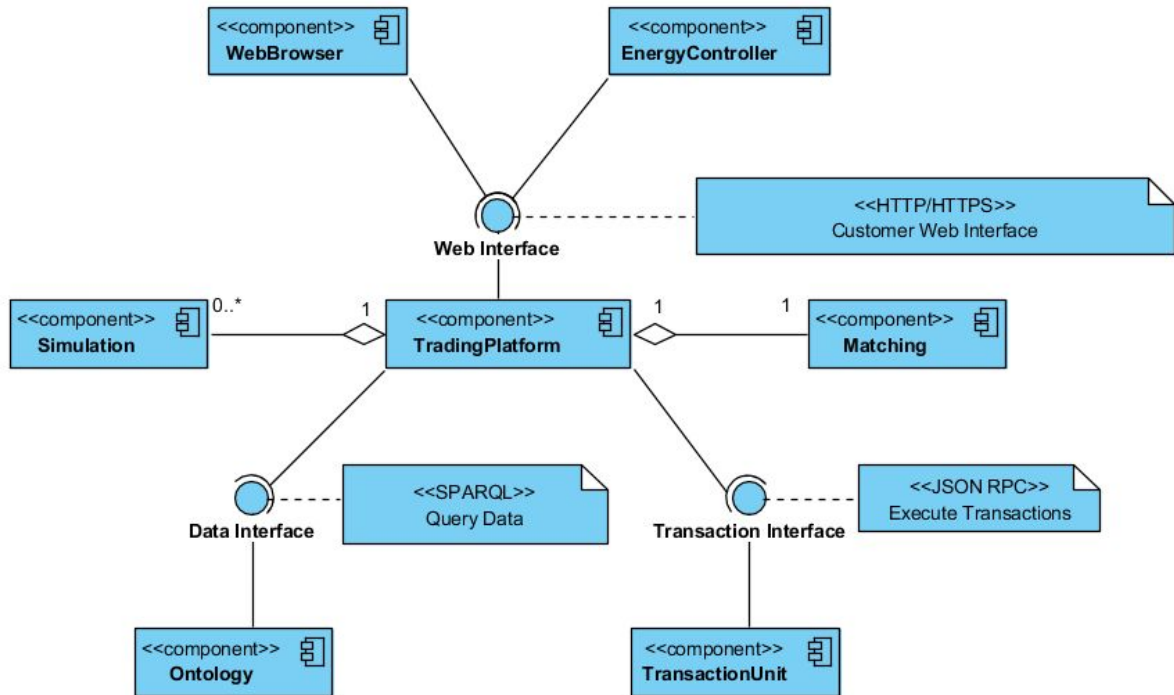
## 4. Process View

### 4.1. Activity Diagrams



## 5. Development View

### 5.1. Component Diagram



### 5.2. Description

#### Web Browser

We will provide the user a graphical user interface via the web browser. This will be the primary interface for user to interact with the system. For the functionality please have a look in the use case description.

#### Trading Platform

This component will be the heart of our system. It will connect all other components with each other and will provide a restful interface to provide its services. Via this api different applications can interact with the system.

#### Energy Controller

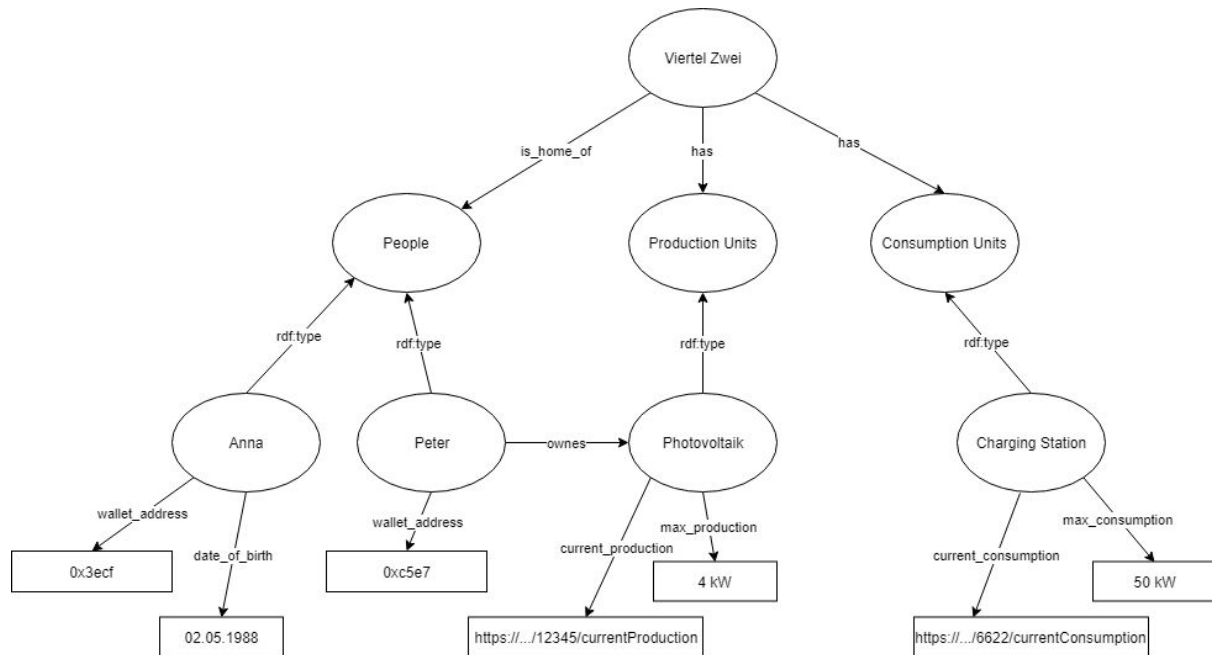
Monitors and controls energy data of a user. Based on this data, usage of different consumption and production units can be determined.

#### Ontology

For our data structure we will use an ontology which should enable us to connect different data within the energy domain. Besides user information, preferences to their energy trading and addresses to their crypto wallets it can comprises information from weather stations and the EPEX Spot (European Power Exchange Spot Market) which can be used for price determination of

an energy unit. All information structured with the ontology can be queried using SPARQL.

Example of how our structure of the ontology can look like:



### Matching

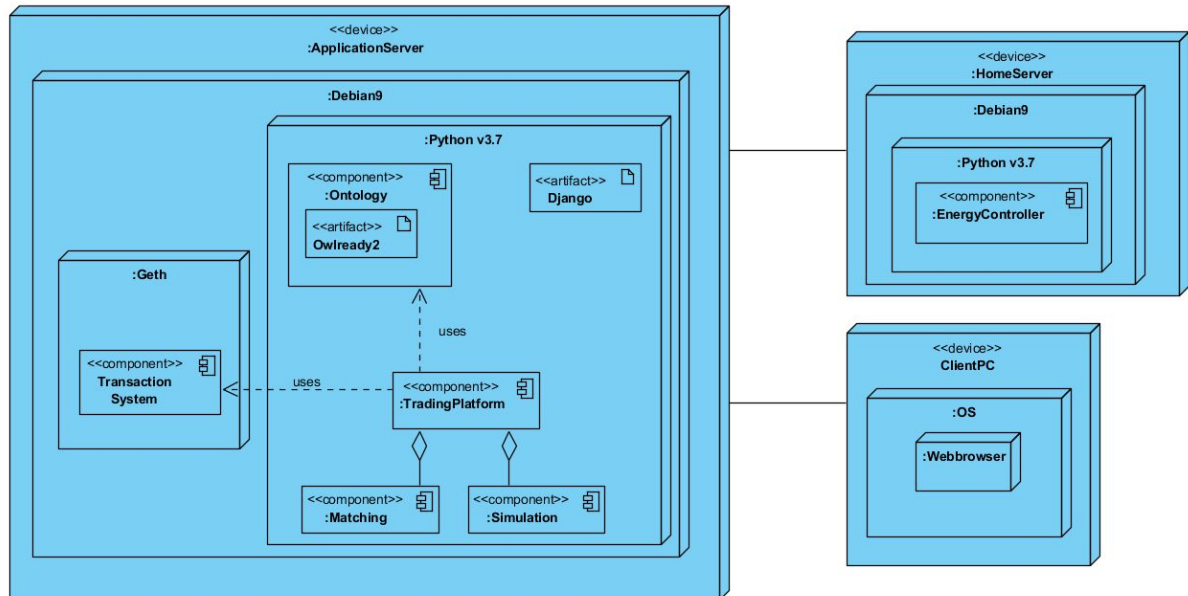
This component will match energy producer with consumer. For the beginning we will use an easy first come first serve algorithm which can be replaced by a more sophisticated algorithm or artificial intelligence later on.

### Transaction Unit

Will handle transactions over the blockchain. Therefore, an ethereum node will be created which can be accessed using JSON RPC.

## 6. Physical View

### 6.1. Deployment Diagram



### 6.2. Description

The trading platform will be hosted on an application server using a debian operating system. For coding we will use python with libraries like 'Owlready2' for building our ontology as well as the 'Django' framework for building our website. Our transaction system will use the geth interface for running a full ethereum node in order to deduct energy.

The EnergyController component will run on a server which will be hosted at client side. Using a similar technology stack as the trading platform.

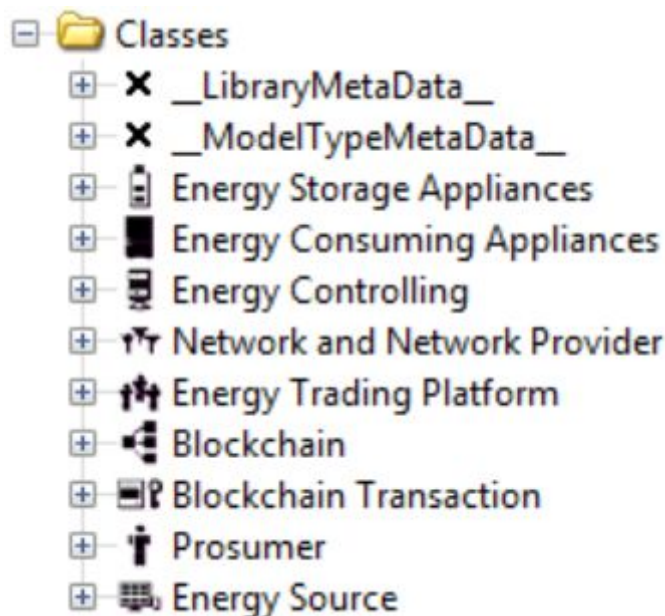
Users can access the system by using http via a web browser.

## 7. ADOxx Metamodell

### 7.1. Description of the domain-specific modelling language

This chapter describes the Metamodel for the whole Energy System which was designed with ADOxx. The developed modelling-language supplies functions to model the whole system and describes how all the single parts of the system interact together. The user can then model his system and then export a XML-File. This XML-File can be uploaded to the Peer-to-Peer website and all energy sources and devices where registered automatically.

### 7.2. Classes


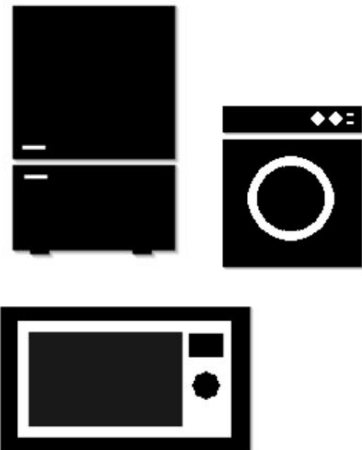


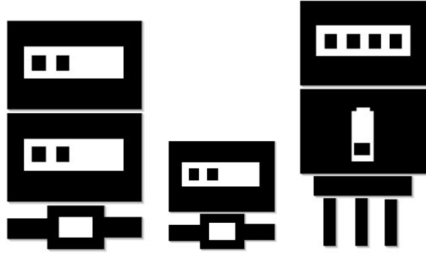
### 7.3. Relation Classes



## 7.4. Classes: Notation and Description

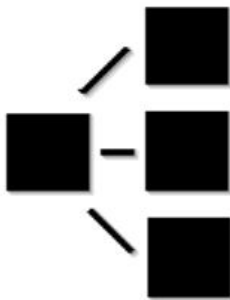


This sub-chapter describes the attributes and their datatypes for the specific class.

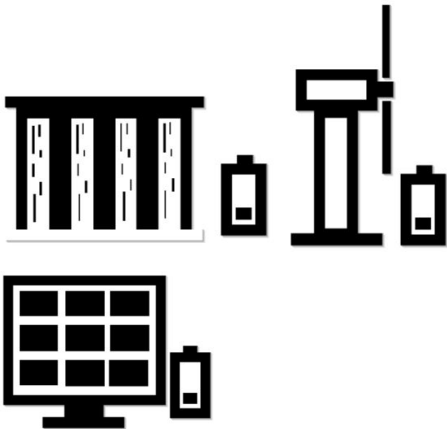
Energy Storage Appliances	Type	Notation
Name Type Description Power Storage Maximum Power Storage Current Storage Begin Storage End Active	Longstring Enumeration Longstring Double Double Datetime Datetime Enumeration	 <p>The Class "Energy Storage Appliances" describes the different devices which can safe energy, especially for the case when the energy production is higher than the consumption. Currently there is just "Batterie" included as a type of "Energy Storage Appliance".</p>
Energy Consuming Appliances	Type	Notation
Name Type Description Power Consuming Maximum Power Consuming Average Power Consuming Current Consuming Begin Consuming End Active	Longstring Enumeration Longstring Double Double Double Datetime Datetime Enumeration	 <p>The Class "Energy Consuming Appliances" describes the different devices which can consume energy. Currently there are "Washing Machine", "Refrigerator" and "Microwave" included as types of "Energy Consuming Appliance".</p>

Energy Controlling	Type	Notation
Name Type Description System Type Counter Task	Longstring Enumeration Longstring Longstring Double Longstring	 <p>The Class "Energy Controlling" describes all devices which monitor the energy flow and control all devices in the system. Moreover, they are initiating the trading on the Trading Platform. Currently there are "Controller Unit", "Subcontroller Unit" and "Smart Meter" included as types of "Energy Controlling".</p>
Network and Network Provider	Type	Notation



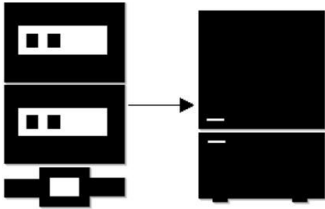


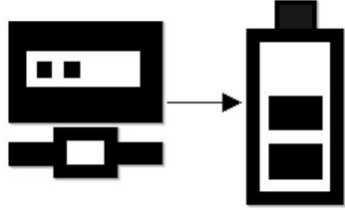
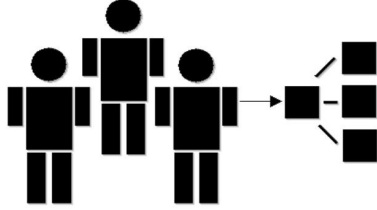
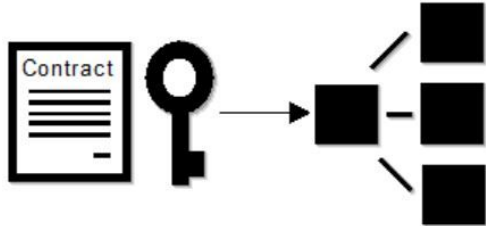
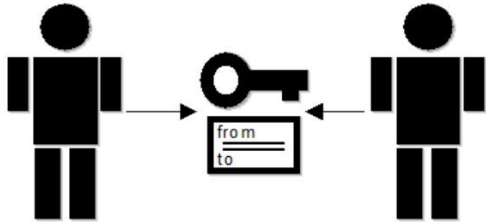
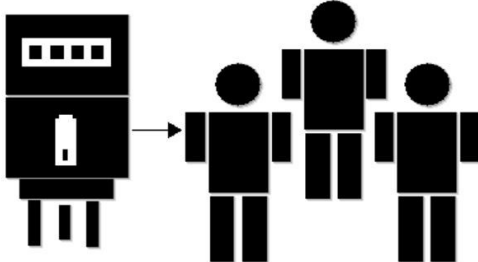
Blockchain	Type	Notation
Name Type Description Cryptocurrency	Longstring Enumeration Longstring Longstring	 <p>The Class "Blockchain" describes the Blockchain which is used for the "Energy Trading Platform".</p>
Blockchain Transaction	Type	Notation
Name Type Description Transaction Fee	Longstring Enumeration Longstring Double	 <p>The Class "Blockchain Transaction" describes all Transactions which can be performed on the Blockchain. Prosumers are initiating the execution of the transactions. Currently there are "Smart Contracts" and "Normal Transactions" included as types of "Blockchain Transaction".</p>
Prosumer	Type	Notation
Name Description Public Address Private Address	Longstring Longstring Longstring Longstring	 <p>The Class "Prosumer" describes all seller and consumer of energy. Prosumers are producing their own electricity and also selling it, if their production is higher than their consumption. In the case when electricity production is low, then electricity is also bought. For both cases the trading platform is used to find the right pair.</p>
Energy Source	Type	Notation

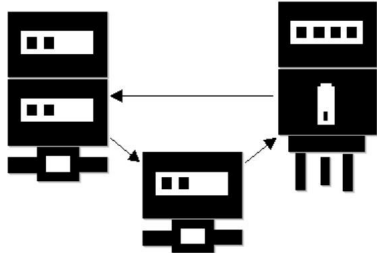
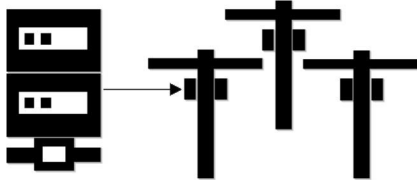
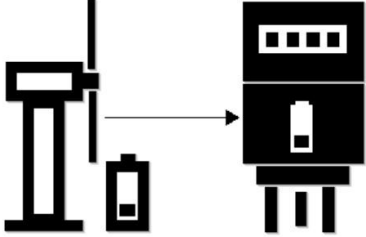
Name Type Description Power Supplying Maximum Power Supplying Average Power Supplying Current Supplying Begin Supplying End Active Power Prodction Maximum Power Production Average Power Production Current Production Begin Production End	Longstring Enumeration Longstring Double Double Double Datetime Datetime Boolean Double Double Double Datetime Datetime	 <p>The Class "Energy Source" describes all the different internal energy production systems the house can has. All energy which is being produced is used to supply the devices in the house, to storage it into batteries or to sell it to other households. Currently there are "Wind Turbines", "Photovoltaic Systems" and "Hydroelectric Plants" included as types of "Energy Source".</p>
---	--	---

## 7.5. Relation Classes: Notation and Description

This sub-chapter describes the Relation-Classes.

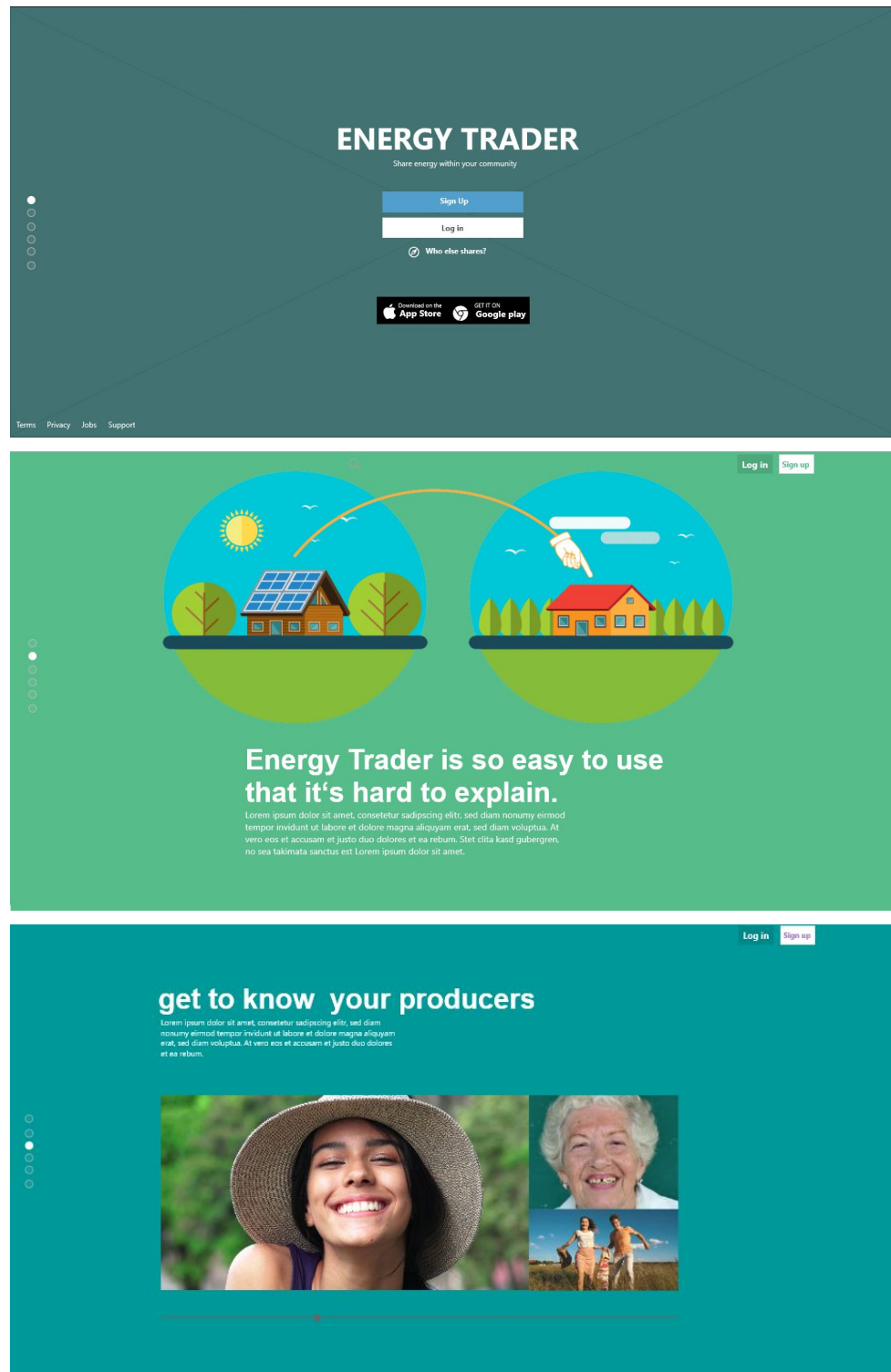
<b>ControllingControlsConsumingAppliances</b>	<b>Notation</b>
The Relation-Class "ControllingControlsConsumingAppliances" describes the relation between the Classes "Energy Controlling" and "Energy Consuming Appliances". This relation should illustrate the control of the controller units over all devices which are connected with the system.	
<b>ControllingControlsStorageAppliances</b>	<b>Notation</b>

<p>The Relation-Class "ControllingControlsStorageAppliances" describes the relation between the Classes "Energy Controlling" and "Energy Storage Appliances". This relation should illustrate the control of the controller units over all devices which are connected with the system.</p>	
<p><b>TradingPlatformPerformsOnBlockchain</b></p>	<p><b>Notation</b></p>
<p>The Relation-Class "TradingPlatformPerformsOnBlockchain" describes the relation between the Classes "Energy Trading Platform" and "Blockchain". This relation should illustrate that the "Energy Trading Platform" performs on the "Blockchain".</p>	
<p><b>TransactionExecutedOnBlockchain</b></p>	<p><b>Notation</b></p>
<p>The Relation-Class "TransactionExecutedOnBlockchain" describes the relation between the Classes "Blockchain Transaction" and "Blockchain". This relation should illustrate that the "Blockchain Transactions" are executed on the "Blockchain".</p>	
<p><b>ProsumerTradesThroughTransaction</b></p>	<p><b>Notation</b></p>
<p>The Relation-Class "ProsumerTradesThroughTransaction" describes the relation between the Classes "Prosumer" and "Blockchain Transactions". This relation should illustrate that the "Prosumer" trades through "Blockchain Transactions".</p>	
<p><b>ControllingDealsOnTradingPlatform</b></p>	<p><b>Notation</b></p>
<p>The Relation-Class "ControllingDealsOnTradingPlatform" describes the relation between the Classes "Energy Controlling" and "Energy Trading Platform". This relation should illustrate that the "Energy Controlling" monitors the energy consumption and can react if energy needs to be sold or bought. The searching for the right partner is done on the "Energy Trading Platform".</p>	

<b>ControllingConnectedWithControlling</b>	<b>Notation</b>
<p>The Relation-Class  "ControllingConnectedWithControlling"  describes the relation between the instances of  the Class "Energy Controlling". This relation  should illustrate that all the "Energy Controlling"  Devices are connected and can interact with  each other.</p>	 <p>The diagram illustrates three energy controlling devices, represented by icons with multiple windows and a base. They are arranged in a triangle, with double-headed arrows connecting each pair of devices, indicating a fully interconnected network where all devices can interact with each other.</p>
<b>ControllingConnectedWithNetwork</b>	<b>Notation</b>
<p>The Relation-Class  "ControllingConnectedWithNetwork" describes  the relation between the Classes "Energy  Controlling" and "Network and Network  Provider". This relation should illustrate that the  "Energy Controlling" Devices monitor and  control the physical connection between the  house and the electricity network.</p>	 <p>The diagram shows an energy controlling device on the left, connected by a single-headed arrow to a network structure on the right. The network structure consists of a central vertical line with three horizontal branches, each ending in a small square, representing a physical connection to an external network.</p>
<b>SourceSuppliesControlling</b>	<b>Notation</b>
<p>The Relation-Class "SourceSuppliesControlling"  describes the relation between the Classes  "Energy Controlling" and "Energy Source". This  relation should illustrate that the "Energy  Controlling" Devices monitor and control the  energy production of the internal "Energy  Source"</p>	 <p>The diagram depicts an energy source on the left, represented by a tall structure with a chimney and a small battery-like icon below it. A single-headed arrow points from the energy source to an energy controlling device on the right, which has a similar icon to the one in the first diagram, indicating that the device monitors and controls the energy production of the source.</p>

## 8. User Interface

### 8.1. Mockup Webpage



## 9. Validation Method: Simulation of the system

This chapter describes the validation method of our project. In order to proof that our system is working we want to implement a simulation tool. This tool should execute the system and simulate the energy production, consumption and trading. The aim is to use the data that was saved before into the ontology. Furthermore should the operator of the simulation tool be able to change the values of the data at runtime. When the operator changes a value the system should react probably. The simulation should contain scenarios to produce, consume and also trade energy.

Technical details:

1. Execution of the simulation via Website
2. Communication between Website and Simulation tool via Rest Api
3. Implementation of Simulation with Python