FUNCTIONAL DESIGN SPECIFICATIONS

Rev #: A
Date: 28-August-2018

Project Title: Smart Grid on Intelligent Token Design
Customer: SERENITY SOURCE
End User: SERENITY SOURCE

Automata
U 12, 5 The Avenue, Mt Druitt, NSW, Australia 2770
Tel: +61 478 718 220
Email: contact@automata.bz URL: www.automata.bz

Contents

REVISION HISTORY

A	28-Aug-2018	Issued for Approval	AS	SN	SN			
Rev #	**Date**	**Description**	**WRT By**	**RVW By**	**APP By**	**APP By**	**APP By**	**Comments**
			SOAP			**CUST**	**END USER**	

Introduction

This project is going to be designed on the true foundations of immutable and decentralized peer to peer ledger which people can trust and use for exchange of energy (electricity) to run their homes. Here we find the true purpose of block chains which is the basis of trust for our assets (energy/electricity) and financial transactions. Smarts contracts have the capability to keep all the promises defined by mutual interests of the prosumers and the consumers. This will involve making up an internet of smart meters running smart contacts and block chain maintaining the mutual consensus.

Users can interact with smart dashboard DApp (Decentralized Application) where buyers meet sellers and vice versa. They can access the energy market/exchange on their mobile devices which are simultaneously connected with their smart meters via DApp and ultimately everything is running on the block chain. Users will be provided with the functionality where while with a touch of finger they can sell and buy energy assets on a trust based secure system. This whole process is going to be cyclic where you produce something and you list it on the market and you get back a reward.

Now here is the part which deserves the most detail. Let's first construct a scenario of demand and supply which will define our true needs and our solution to fulfil the prosumer/consumer needs. For instance, suppose that there are two users, User A & User B. User A wants to send/sell some/some/all its watt/Hrs produced and User B has registered for purchase. A market DApp will help both User A & User B to list their energy assets on the exchange and buy them accordingly.

DApps are directly running on the block chain network which relates to the smart contract (with ERC-20 token functionality). User A will generate a send request on DApp and some watt/Hrs (As per User A request) will be listed on the exchange with a tag/public/wallet address and User B can see and verify the tag address and make a direct connection with the User A using the peer-to-peer connectivity of the Ethereum network.

A remote connection with one smart contract will be maintained during the assets/token transaction. Smart contract will be running on the Ethereum network which guarantees the decentralization without the involvement of a third party. Smart contract (which will be running on our own Grid-Hub with a full Ethereum Node with the capacity to verify the energy consumption or availability on its own computing meter) will verify the energy received by the User A smart meter and then it will verify the number of tokens sent by the User B wallet. Then after the transaction inputs are verified and checked according to the exchange rate calculations the transaction is performed.

Here a transaction is defined as the number of tokens sent over the network and the energy received received.

Design Elements:

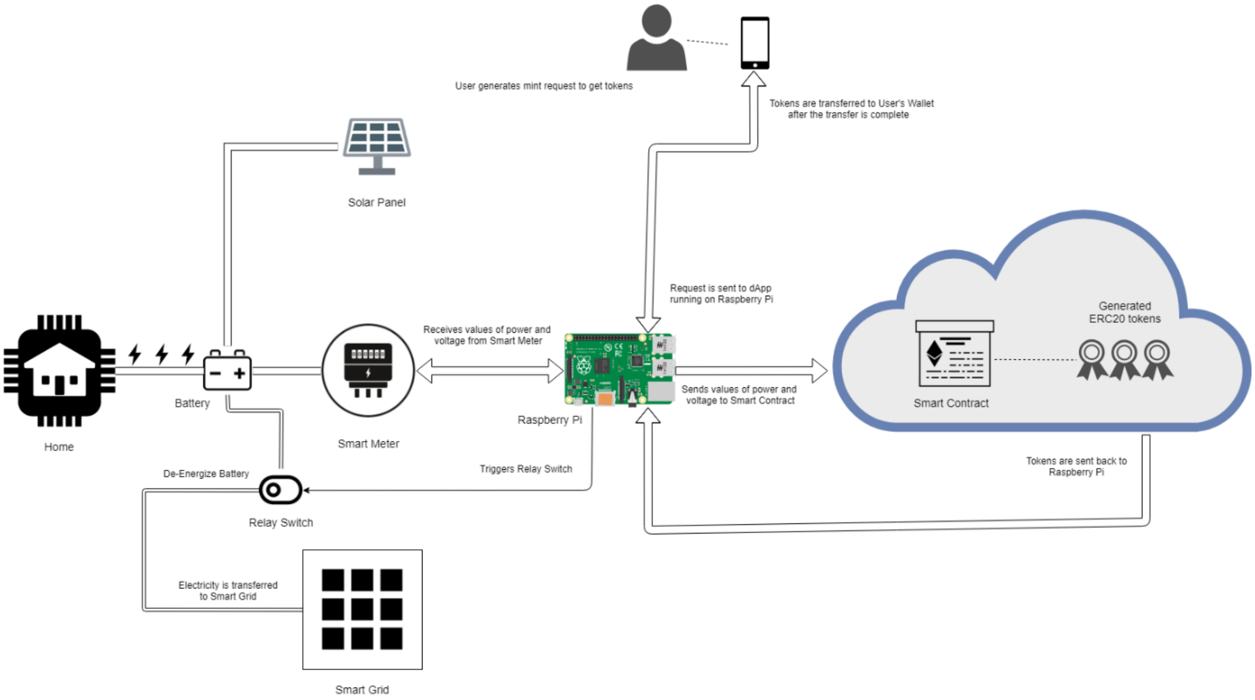
- Smart Meters
- Raspberry Pi
- Ethereum VMs on Smart Grid Nodes
- DApp Dashboard

Basic Principle

Following is the architectural design idea of our projects demo submission:

We plan to connect high speed internet to smart computers (Raspberry Pi) which will receive/send requests to a smart contract via wattage received from a smart meter. A relay switch will energize or de-energize the battery (connected to a smart meter). And the rest of the assets/token exchange is going to be executed on the block chain network via smart contract. A DApp (Running on the Ethereum Node) will receive data from the smart meter of the User and will keep check of all the energy units produced by the battery and it will perform the exchange for the user. A signal will be sent over the network to a smart contract. Here user generates a request from the DApp (provided with a wallet) to their smart meters and this request will be processed on the computer (Raspberry Pi) attached with the smart meter. And then it will be broadcasted to the Block chain Network.

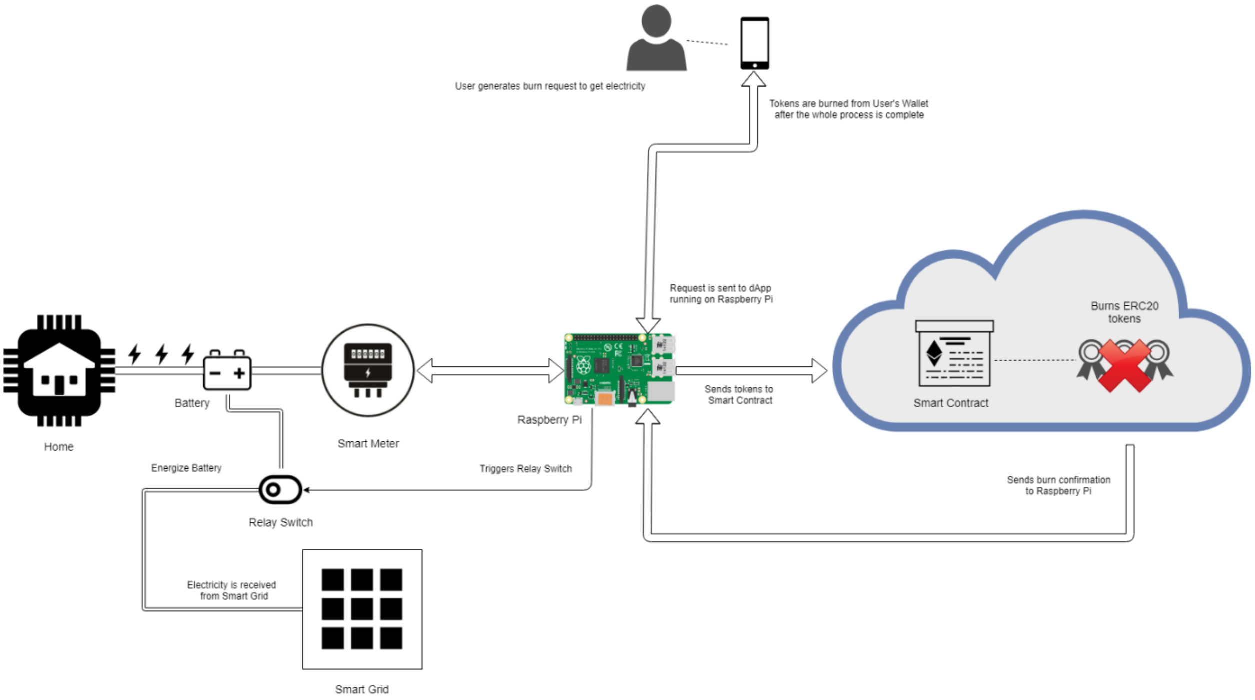
Mint Request Principle


User generates mint request to get tokens
Tokens are transferred to User's Wallet after the transfer is complete
Solar Panel
Request is sent to dApp running on Raspberry Pi
Generated ERC20 tokens
Receives values of power and voltage from Smart Meter
Home
Battery
Smart Meter
Raspberry Pi
Sends values of power and voltage to Smart Contract
Smart Contract
Tokens are sent back to Raspberry Pi
De-Energize Battery
Triggers Relay Switch
Relay Switch
Electricity is transferred to Smart Grid
Smart Grid

Every home is going to be provided with a smart meter which is connected with an internet connection. Smart meters are supposed to transmit data to internally connected Raspberry chip which is running a SPV (Simple Payment Verification) Node or smart wallet. A smart wallet will then make a call to the Ethereum Node (Which is running our smart contract on it) based on the user/meter input. The smart contract will give a kick off call to the relay system. The relay switch will be turned on and it will measure the amount of watts being sent by the user smart meter and it will drain the user battery accordingly (Watts will be stored in the smart grid system). After the drainage process is complete the smart contract will verify the watts received by the smart Grid (Closing the relay system) and smart contract will then mint some tokens to the User wallet address.

This Whole process is automated and will be completely controlled by the Dapp provided to the User. Basic idea here in the diagram is to transmit some amount of energy/electricity to a hub and in exchange for the asset a smart contract will reward some tokens as per market rules defined.

Burn Request Principle


User generates burn request to get electricity
Tokens are burned from User's Wallet after the whole process is complete
Request is sent to dApp running on Raspberry Pi
Burns ERC20 tokens
Smart Contract
Sends tokens to Smart Contract
Raspberry Pi
Smart Meter
Battery
Home
Energize Battery
Triggers Relay Switch
Relay Switch
Sends burn confirmation to Raspberry Pi
Electricity is received from Smart Grid
Smart Grid

In this case we need our battery to be energized and receive some payment tokens back from the token pool. In the previous case the smart Grid was holding the energy information and the wallets were receiving tokens. Here the opposite case applies.

The user will send request to smart contract to buy some watts from the smart grid and then the smart contract will verify whether the required number of watts are present in the smart Grid or not. After the verification is done it will check the user wallet for the number of tokens per watt according the exchange rate and then if it satisfies every rule defined in the contract, the transfer will be executed.

The reverse relay system will be turned on and it will juice up the user battery with the required amount of watts and the smart contract will receive the same number of tokens against the exchange rate from the user wallet.