

D-REX (Distributed Real-time Energy Exchange)

Abstract: Provide a marketplace for financiers and providers to issue and purchase electricity futures. By creating a market for microfinancing generation of electricity we can crowdfund infrastructure to provide electricity for the 1.5 billion people in the world currently without.

User Groups:

- 1) **Consumers:**
 - a) Consumers are the people interacting with the electric generators to receive electricity. They spend *Solar Tokens* using their mobile client (or SMS codes) at generators to pay for electricity in kilowatt hour chunks.
 - b) They can also create requests for solar generators in their region, detailing *location*, *wattage usage*, *reason* for electricity need, etc, which can be bid on by providers and funded by financiers.
- 2) **Providers:**
 - a) Providers bid on requests from consumers. They provide bids detailing type of solar energy unit, cost of installation, time for installation, their insurance, etc.
 - b) (FUTURE) Need to be verified or have some reputation modeling or have some sort of insurance.
- 3) **Insurers:**
 - a) Planned for the future
 - b) Insure providers or financiers
 - c) Provide derivatives of energy futures. Sell part ownership of many utilities for n time.
- 4) **Financiers:**
 - a) Can be large institutions (Royal Bank of Scotland, Deutsche Bank, etc) or anyone with money. Can even be part or fully owned by the community itself.
 - b) Get the emotional payout of helping bring electricity to the underprivileged as well as the financial incentive into micro financing infrastructure.

Specs:

Request: {

```
"Location": STRING,  
"Creator": STRING, //address of the owner who submitted the tx  
"WATTAGE_REQ": INT,  
"REASON": STRING
```

}

Bid: {

```
"ProviderAddr": STRING,  
"Unit_type": STRING,  
"Cost_of_installation": INT
```

}

Process:

- 1) **Consumers** make a *request* for a new generator. They list the location, wattage requirements, reason for request, etc.
- 2) **Providers** look through *requests* and bid on them. They include the type of generator, cost of installation, time for installation, insurance, etc. This bidding process is open for **ONE WEEK**. Consumers can then accept one of the bids, at which point...
- 3) **Financiers** bid on consumer accepted *requests*. This process is open for **ONE MONTH**. If the request is successfully funded a *generator* contract is created to represent the pending device.
- 4) The **Financiers** that own the the generator should all call the smart contract and submit their preferred cost for electricity the smart contract should charge. The price is the average of all submitted values weighted by the ownership stake each financier has. If no **financier** has submitted a price request, then the average is just 0 and the generator runs for free. If only some of the **financiers** have provided a price, it's just the average of the ones provided. If you don't vote, you don't get a voice and must accept the price set by your co **financiers**.
- 5) The **Provider** that won the bid then installs the generator in the location specified.
- 6) The **Consumer** that issued the request than approves the installation by instantiating the generator by sending an initial token to it. The first tx MUST be by the **Consumer** that issued the request. After initialization, the generator can be funded by anyone and used by anyone.
- 7) **Consumers** can then fund the *generator* get electricity. They pay in *solar tokens* which are then disbursed to **financiers** proportionate to their ownership stake in the generator. Consumers can buy these *solar tokens* from exchanges.
- 8) **Financiers** make money by selling their solar tokens on exchanges to anyone who wants to buy. *Solar tokens* have a consumptive value on the platform, not just a speculative value, so there's always a demand.

Tech:

- 1) Contracts:
 - a) Solar Tokens: ERC 20 token contract. ICO a limited supply of **x million** .
 - b) MarketPlace (split into two sections):
 - i) New Generators: Consumers make requests with the given metadata (location, wattage, reason for request, etc). Request is bid on and then awarded.
 - ii) Old Generators: **Financiers** can sell all or part of their ownership of their generators or even market futures and based on the generators ownerships they own.
 - c) Generators:
 - i) Each generator is it's own smart contract. Issues 10,000 (100.00%) ownership tokens at inception based on proportionate financing.
 - ii) GetAvg(): read only function that returns the average price of what the generator should charge for electricity.
 - iii) Status pending until first tx which needs to be done by **Consumer** who created the initial request.
 - iv) (Future) Funds set aside for maintenance / some catch for it.
- 2) Stack:
 - a) Flask Microservice to interact with Qtum RPC
 - b) React front end for UI

Interesting Points:

- Very interesting use case for trusted execution environments. Only the solar generator should be able to make its transactions, someone shouldn't be able to mess with it.
- Smart Contracts built with generic "Utility Model". Crowdfunding electricity now, but could crowdfund any utility (Water Mill, Textile Machines, etc...)
- Providers can have radically different ways to build units and have varying reputation within a community.