Onderwerp:	Operation of FR blockchain | Project process | MPV
Van:	Daniel James Paterson (danieljpaterson@yahoo.co.uk)
Aan:	ben.britton@fieldready.org; andrew.lamb@fieldready.org;
Cc:	anna@nnvnx.com;
Datum:	vrijdag 22 september 10:51 2017

Hi both,

Thank you for your very useful feedback on the process flowchart - I'll digest and incorporate it. In the meanwhile, let me take this opportunity to outline three things for your understanding of the project going forward. Hopefully this tackles any confusion you have Andrew.

Feedback welcome though not essential - the most important thing is that you digest these notes. Read in conjunction with the process flowchart, and note that *contract* and *smart contract* are two very different things. We may want to Skype to cover this email verbally.

1) Operation of the FieldReady blockchain

Here's how I envision the FieldReady blockchain working in simple terms (this understanding will be evolved as we go forward):

- A blockchain *smart contract* is a file (.sol, in the Solidity language) which essentially contains two things: variables and functions. Running a smart contract on the Ethereum blockchain allows the state of the variables to be repeatedly changed and actions to be triggered. The variable state changes and actions are immutably recorded forever.

- Every distributed manufacturing customer order will have its own unique *instance* of the single Field Ready smart contract which this project will design. This file will 'progress' down the (blue) right-hand side of the process flow chart, the state of its variables being updated and actions automatically being triggered as it goes, in order to drive the process forward. For example, the smart contract will contain meta variables such as the number of suppliers, and an array of variables for supplier names, 'quality pass' flags, terms of the supplier contracts, etc.

- A blockchain isn't a database and cannot hold large amounts of data such as supplier contracts and designs. What is it is a way to record and time-stamp pieces of information so they can be proven later. Therefore the supplier

contracts and designs will be hashed (NB it's not just text that can be hashed; files can too) and the hashes recorded on the blockchain in steps 4 and 5. At any time in the future, any party can prove aspects of the contract and design by hashing it and checking that the hash is the same as the one on the blockchain.

- A blockchain smart contract can also interact with and control *wallets*, which is what allows it to automatically send payments from one account to another according to the functions and variables in its code (in the Ethereum currency, *ether*).

2) Project process

1. Define the *future state* of a Field Ready blockchain << *this is what we are doing now.*
2. Decide which elements of the future state are appropriate for the MPV. Goal: an *incremental future state*.
3. Write the Field Ready smart contract (I will disappear for a few weeks while doing this).
4. Run a test simulation.
5. Rejoice that we have done something genuinely innovative in amongst all the noise.

3) Minimum viable product

To manage expectations, this will indeed be minimum:

- I don't envision it having a DAPP (a front-end). We will run the test from wallets or an online design environment. This will be better anyway, as we will see and feel the blockchain in action.
- I don't envision it having any payment integrations to real-world currencies - all payments will be in ether in wallets on the *Ropsten testnet*. We may not even include payments in the MPV based on Ben's feedback.
- I have accumulated 40 *testnet ether* we can use as 'gas'. (Gas is what it costs to run a smart contract on the distributed network - how much gas is used depends on size of contract, type of variables, etc. I don't envision I will put time into 'leaning' the contract to save gas - so long as it works at this point, that is fine).

- Hashing will likely be done manually, not automatically.

For the purists, the summary of all this is:
The goal of this project is to develop a Field Ready smart contract which mimics a basic distributed manufacturing process, and then run it once.

Qué?
Daniel

Daniel James Paterson, MEng

Pico-solar | Digital tech | Blockchain | Lean