Nila & Algorand

A light node network to sync items in a Merkle tree.

Challenge

How to solve synchronization issues in many-to-many relationships between spot marketplaces and warehouses that store perishable food commodities.

- At this moment, the agricultural supply chain in India has two main innovation drivers; the first are cooled storage spaces for perishable crops, sometimes as small as a single village container, the second are the trade and direct sales platforms that expose produce to a open market outside of the state regulated APMCs.
- Many e-mandis (digital marketplaces) are listing products from many warehouses. Warehouse stock is sold on several e-mandis. One can understand that keeping an up-to-date product inventory is very complex, also because storage house operators have limited digital skills.
- Proof of location is important due to high logistical costs.

Solution

A 'light-node' Merkle tree network managed by a TEALscript Algorand contract. <u>The contract removes the need for each node to fully sync the entire tree.</u>

technical:

- A warehouse contract has a fixed size that is set on creation, *equal* to the warehouse size. It cannot be altered.
- A node can verify a product location by *matching* the on-chain root (stored in a global variable) to its own Merkle path.
- A node is able to remove a leaf by calling the contract and providing *Merkle proof*.
- Once a leaf is updated/removed, a node communicates via a websocket connection the Merkle path and own route to other nodes opted into the contract. This requires the nodes *to be online at all times*. In order to receive / validate and update their own fragmented tree.

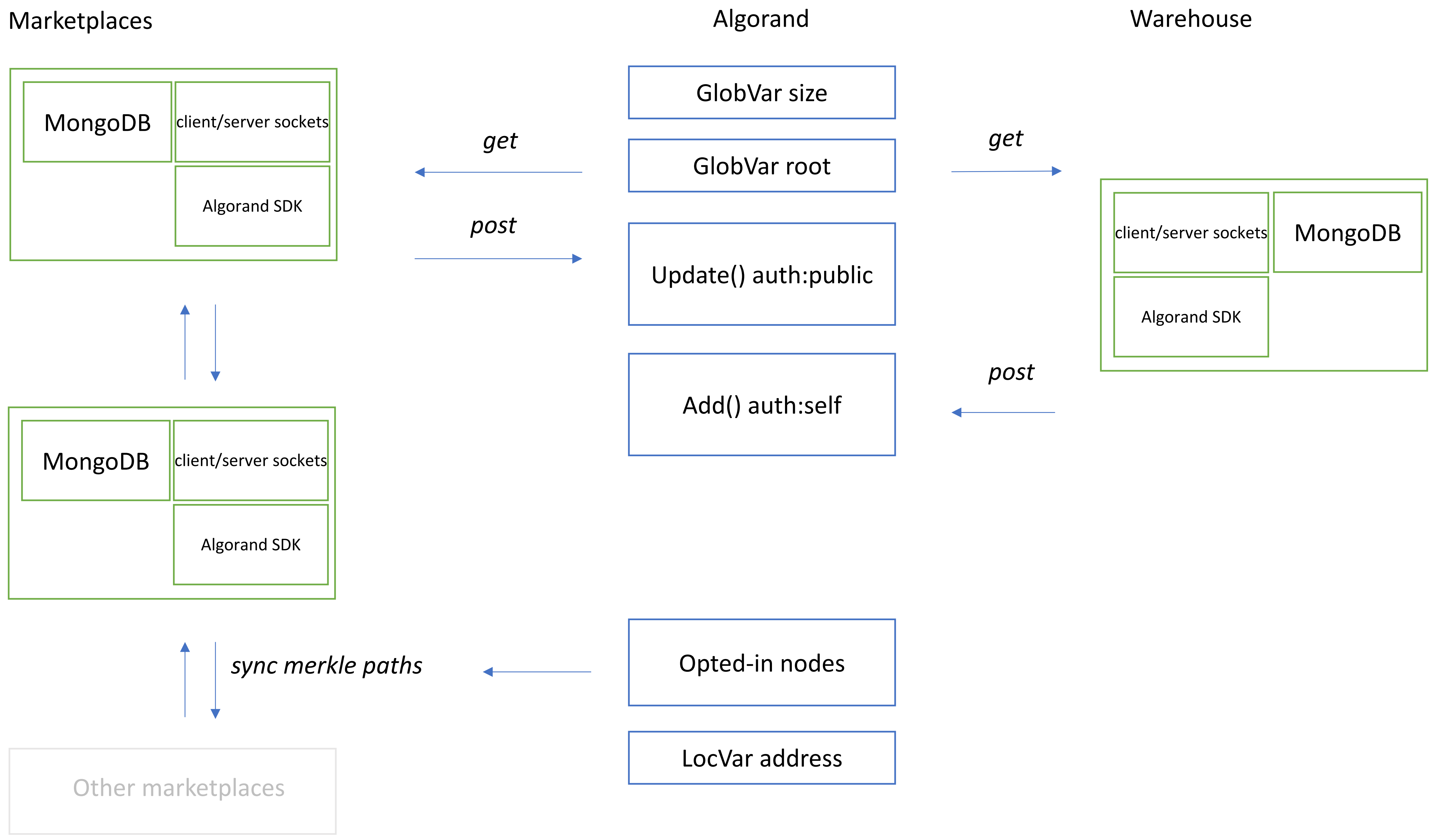
Solution

operational:

- The solution proofs that decentralized applications are *more efficient and faster* then their centralized sibling in a low-trust environment. Three design conditions are important to achieve this:
    1. Use a fixed size tree.
    2. Let sellers remove the stock, independently of the warehouse operators.
    3. Let operators confirm stock has been physically removed, before they can add a new item on a *empty* position.

Effects:

1. To limit the tree size forces the operator to discard unsold items.
2. To let sellers remove stock speeds up the process and avoids time lags on the operator side.
3. To require confirmation on new products assures delivery status and syncs the physical – digital state.


Marketplaces
Algorand
Warehouse
MongoDB
client/server sockets
Algorand SDK
GlobVar size
GlobVar root
Update() auth:public
Add() auth:self
get
post
get
post
client/server sockets
MongoDB
Algorand SDK
MongoDB
client/server sockets
Algorand SDK
sync merkle paths
Opted-in nodes
LocVar address
Other marketplaces

What is Nila

Nila delivers web3 technologies for the agricultural supply chain, with a primary focus in India. Nila runs on Algorand, and has been supported in the past by the foundation.

- Nila has about 400 **active** users onboarded, ready to experiment. They are part of 2 producer cooperations we train and advice ourselves. So a close-knit community to experiment.

Because of the users Nila serves, UX is extremely important.

- Nila operates in areas with low digital penetration and low trust. The impact on these groups can be enormous, but is very hard to achieve. Our software has to be shared, nimble, lightweight and uncomplicated.
- This can be achieved primary by automation, we for example delineate field borders completely automatically, only a single coordinate is needed. The web3 ethos of - code is law - fits in ecosystems where pay-offs are highly disparate.

Nila features

- **Single point fieldbordering**

- **Automated field monitoring**

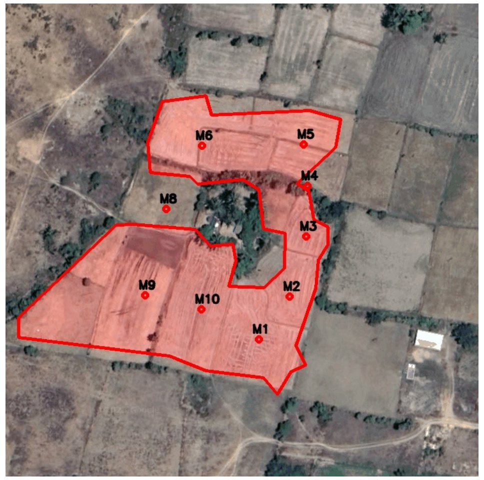
1. Event detection (planting, harvesting,etc.)
2. Crop detection
3. Crop health detection
4. Yield estimation

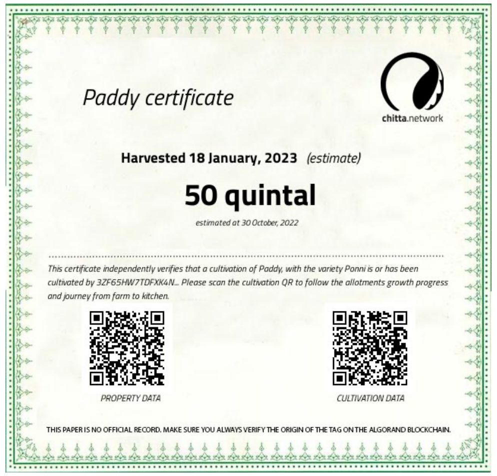
- **Digital wallets via WhatsApp**

1. UPI payment bridge (Indian Rupees)
2. Transactions using 4 digit pin and QR codes.
3. Private keys recoverable by farm cooperatives MSI
4. Offline use possible (GPS required for some uses)

- **Real World Assets**

1. Supply-chain trail
2. Origin certificates
3. Harvest date proof
4. Storage conditions
5. Export documentation




Paddy certificate

chitta.network

Harvested 18 January, 2023 *(estimate)*

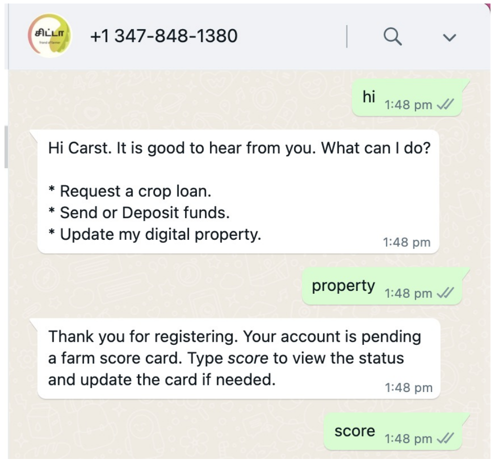
50 quintal

estimated at 30 October, 2022

This certificate independently verifies that a cultivation of Paddy, with the variety Ponni is or has been cultivated by 3ZF65HW7TDFXK4N... Please scan the cultivation QR to follow the allotments growth progress and journey from farm to kitchen.

PROPERTY DATA *CULTIVATION DATA*

THIS PAPER IS NO OFFICIAL RECORD. MAKE SURE YOU ALWAYS VERIFY THE ORIGIN OF THE TAG ON THE ALGORAND BLOCKCHAIN.


+1 347-848-1380

hi 1:48 pm

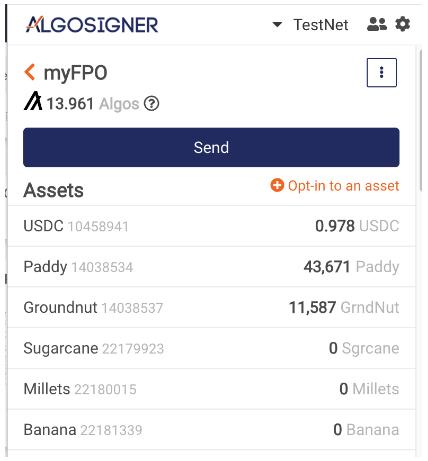
Hi Carst. It is good to hear from you. What can I do?

* Request a crop loan.
* Send or Deposit funds.
* Update my digital property. 1:48 pm

property 1:48 pm

Thank you for registering. Your account is pending a farm score card. Type *score* to view the status and update the card if needed. 1:48 pm

score 1:48 pm


ALGOSIGNER TestNet

myFPO

13.961 Algos

Assets	Opt-in to an asset
USDC 10458941	0.978 USDC
Paddy 14038534	43,671 Paddy
Groundnut 14038537	11,587 GrndNut
Sugarcane 22179923	0 Sgrcane
Millets 22180015	0 Millets
Banana 22181339	0 Banana

Future improvements

The solution has not been validated and is rudimentary in its delivery*. Future work adds the following features:

tasks:

- Warehouse managers can share new items through a different channel.
- Warehouse managers – or RWA asset oracles – can remove unsold or expired items to make place for new items.
- Marketplaces have extended ability to not only remove (= replace with a empty hash) but also update (= replace with a new hash). For example when not the entire stock has been sold.
- Contracts check on-chain payments have been made.
- Contracts use Nila RWA tokens to not only verify location but also pre-harvest conditions.
- Sync Merkle paths without the need for a item index by using right/left prefixes.

*** We pledge the application has only been envisioned for the build-a-bull hackathon (not a continuation of an older app).**

FAQ and tests

LocalNet

- Clone the github repository and run a fleet of nodes, create the application.

TestNet

- Users can easily mistake sending the storage issuer QR for the purchase (verifier) QR.