



Nila & Fuel

Agricultural Supply and Finance Protocols on Fuel

State of Nila

- Nila is a digital wallet chatbot operational in Tamil Nadu, India. Nila runs on Algorand, and has been supported in the past by the foundation, but is keen to experiment with other chains.

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.

- Nila users have limited digital skills, they have smartphones, but only use WhatsApp and Youtube.

Nila has solved most the UX when faced with limited user capabilities: WhatsApp conversations, state-of-the-art field bordering algorithms, automated field progress updates and automated track records.

- Nila users and vetted users have very limited money to collateralize loans.

Nila users are smallholder farmers that cannot access crypto lending protocols, as they do not hold any collateral (except the ownership of their lands). We need to use the crypto sector ability to 'hand them assets'.

Nila features

- Single point fieldbordering



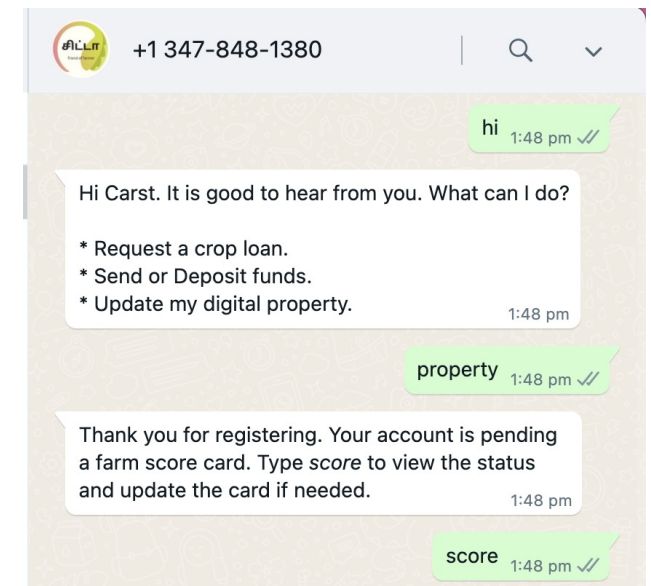
- Automated field monitoring

1. Event detection (planting, harvesting, etc.)
2. Crop detection
3. Crop health detection
4. Yield estimation
5. RWA certificates (on Algorand)



- 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)



The fuel-nila experiment

- Testing of unsecured loans using 3th party 'vouch' signatures
- Testing of debt refinance using crop tokens (fungible utxo-based)

Goal: test the software potential and ability to step into something new from the ground-up.

Why choose FUEL :

- Use of multisig predicates.
- UTXO output as unique feature – not possible with account-based protocols.

Unsecured loans using multisig predicates

Experiment to build a predicate that is instantiated by the borrower, funded by the credit contract and wait to execute until at least 2 signatures from a vetted list of addresses are found.

What worked:

- Instantiates predicate with configurables.
- Implemented witness data.
- Sign predicate by a sponsor
- Deposit funds to the predicate address by the contract

What didn't work:

- Unable to verify a predicate (whitelisted addresses) on-chain. Only the hex string match will not reveal whitelisted signatures are indeed of vetted sponsors.
- Several issues with unclear documentation. The concept of a predicate is hard to grasp, has limited scope.

Result:

- *Unfortunately not a use-case for predicates. Simple sign transaction verified on-chain much more effective. Mainly because the predicate conditions can't be verified by the contract that sends the funds.*

Refinance loan (collateralize with RW assets)

Experiment to refinance loans with crop tokens after a harvest event is signaled. Extension of credit by the sponsor if it is a known storage partner. Use of UTxO data to verify origin and collected data (quantity, conditions), because UTxO perceives fungible tokens as individual.

What worked:

- Oracle contract to establish crop token price
- Limit to vetted sponsors with known storage.
- Overcollateralization of refinanced loan.

What didn't work:

- Contracts are account-based, not UTxOs, asset history has to come from indexer
- Quantity and match with property NFT should be verifiable (not tested)

Result:

- *Experiment worked. However limited in its scope and no improvement over any other 'account-based' chains. Fungible tokens not indistinguishable. Documentation has to be improved.*
- *Successful collateralization with oracle (dummy price of paddy) and refinance action (insert new conditions to on-chain storage).*
- *Property NFT and Paddy RWA not yet transferred to FUEL (residing on Algorand).*
- *Not clear how to use UTxO data in contract, or verify the token history.*

Future of Fuel & Nila

- If rewarded, 50% of the money will go to credit offerings on fuel testnet.
- Choice to move all assets and methods to Fuel dependent on:
 - mainnet eta,
 - stablecoins or Polygon bridge assets.
 - RWA focus of Fuel.
- We have to still a lot to learn about Fuel, Sway and Rust (original Java & Python developers).