Parallel

Project Description

Parallel is a mobile substrate client with Point of Sale(PoS) to interact with Parity blockchains.
Parallel's team contributes to substrate by developing JSON-RPC clients(like web3.js in ethereum) and mobile integration. We want to improve Substrate client library so that developing DApps for Substrate is as easy as developing DApps on Ethereum. Our project builds

- better account management support for Polkadot API(e.g. account backup and recovery)
- Native mobile client for soverign mobile wallets
- Significantly improved developer documentation for Client API

Solution

Parallel will solve the problems by contributing to client-side of the blockchain so that the Parity team can focus on its blockchain development.

We solve the problem by providing:

- JSON-RPC API development/documentation for dapp development
- General client for end-users to use Substrate which accepts the newest implementation of the client updates

- Improvement on adoption on client for monitoring and interacting with runtime api endpoints in Substrate

There are JSON-RPC endpoints that are not documented yet here in offical documentation. We will focus on revealing what the rest of endpoints do.

As we develop and adopt mobile client in beta stage, we will be able to able to provide a much faster rate of contribution to future updates such as parachain development, interacting with participant roles, staking mechanism. In addition, we will have our own client app so that the adoptions can be delivered to the users from beta stage to future.

We will improve current Substrate api to provide more information on Substrate runtime. Current api system does not support looking up the runtime api endpoints and inputs, and only looking compiled wasm byte code is supported. We will make another endpoints in Substrate which retrieves an interface that DApp and Substrate can communicate like ABI in ethereum.

Team members

Ling Qing Meng

Tony Rose

Yuri Lazarini

Nick Hyungsuk Kang

Dillon Settle

Rico Chen

Nagu Thogiti

Team Website

https://decentral.solutions

Legal Structure

C Corporation, Incorporated in Delaware

Decentral, Inc.

119 Michael Way

Santa Clara, CA

95051

Team’s experience

Parallel team consists of members with 20 years of experience who launched Apple/Android pay across enterprise platforms, were awarded for a blockchain patent at Vantiv, worked with the largest bank in Latin America, lobbied California and Wyoming house and Senate for common-sense blockchain legislation, and organized conferences and hackathons driving blockchain technology and regulation advancement, amongst other accomplishments.

Team Code Repos

https://github.com/decentral-inc/

Team LinkedIn Profiles

Ling Qing Meng(https://www.linkedin.com/in/ling-qing-meng-90a35552/),
Tony Rose(https://www.linkedin.com/in/tony-rose/),
Yuri Lazarini(https://www.linkedin.com/in/yuri-lazarini-900b9525/),
Nagu Thogiti(https://www.linkedin.com/in/naguthogiti/),
Rico Chen(https://www.linkedin.com/in/ricochenx/),
Hyungsuk Kang(https://www.linkedin.com/in/nick-kang-5217a7103/)

Current Development Plan

Background

A robust, more extensive substrate client API development library will lead to increased developer activity and contribution. The sooner we do it, the more long term benefit we have to gain. To give an analogy

> *Here is a simple description for what web3 in ethereum does*

1. > *on web3 library, developer executes function like register_account()*

2. > *the web3 formatter encodes interface to JSON-RPC2.0 payload corresponding to the function*

3. > *the payload is sent from the RPC client provided by the web3 library*

4. > *web3 receives the response and shows the result*

As such, this is how web3.js functions as a client API development library for Ethereum. Polkadot API is also following the same process using encoder here. We plan to build this in Swift for fully native mobile integration.

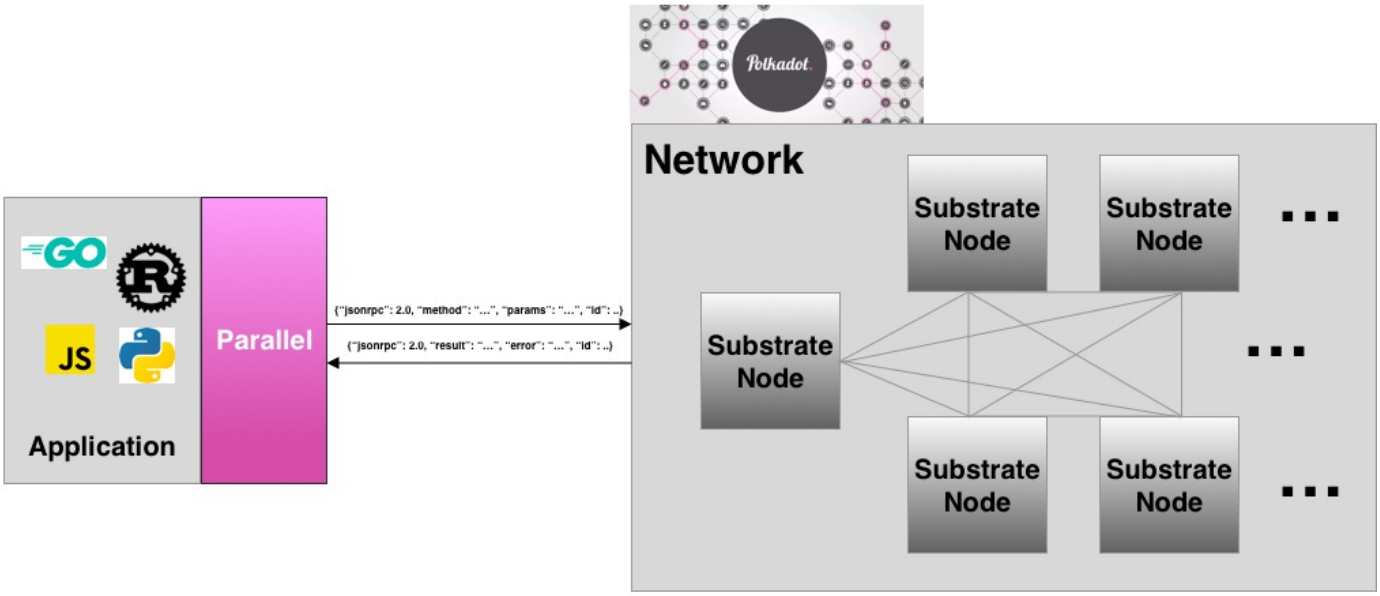
Core Tasks of our Development Plan

1. Develop Polkadot RPC client for account management, mobile key generation, encryption/decryption (1 month)
2. NFC and Bluetooth module in Substrate to listen and authenticate payments (1.5 months)
3. Document rest of JSON-RPC api endpoints in Substrate (1.5 months)
4. Design and document interface for client to interact with Substrate runtime module (1.2 months)
5. Develop interface between client and the runtime module with wallet onboarding (3 montha)
6. Build wallet app for multiple substrate assets(e.g. ERC721 runtime, default currencies on each substrate network) (3

months)

7. Runtime API for building interface between substrate and client(1.2 months)

Key Takeaway: We will build for Substrate, a formatter and rpc-client in multiple languages(javascript, python, go, rust) for developers so that they do not need to rebuild the client for the network(e.g. libp2p).



Further Development

IF we are selected for the grant, here are the development tasks we can work on.

- Point of Sale(PoS) network with NFC to make polkadot network's asset can be transferred in real life
  Point of Sale has become a trend in cryptospace and it is one of the most effective use cases for a blockchain.
  If we make polkadot network supported PoS, this would expand polkadot ecosystem with robust real-life use cases.

- Wearable interface for asset exchange on polkadot
  Payment channel for digital assets that can be used in real life is now a field yet to be developed. We plan to build decentralized POS payment infrastructure using current platform and independant device. Right now we have partnerships with payment processors such as Vantiv and Worldpay.

Development Roadmap

[10k] 1. Develop Substrate RPC client for account management, mobile key generation, encryption/decryption in Swift (1 month)
[3k] 2. Document rest of JSON-RPC api endpoints in Substrate (1.5 months)
[4k] 3. Design and document interface for client to interact with Substrate runtime module (1.2 months)
[4k] 4. Develop interface between client and the runtime module with wallet onboarding (3 montha)
[11k] 5. Build wallet app for multiple substrate assets(e.g. ERC721 runtime, default currencies on each substrate network) (3 months)
[15k] 6. PoS Hardware integration for wallet app(1.2 months)

[12k] 7. Apple Watch + End to End integration(2 months)

`[59k] Total Apple Watch App Development + Apple End to End Integration Software`

[12k] Showcase with an react app(e.g. block explorer)

[20k] Develop mobile polkadot OS in React Native

[10k] Travel budget

`[42k] Total Blockchain software development for Point of Sale system`

Intended language of development

We plan to implement JSON-RPC api in swift for mobile integration, and other programming language implementation for Polkadot API can be developed for building the PoS network.