

Beacon Kusama - Report

Report 2

October 2021 – April 2022

This report outlines the effort of Papers to fulfill the requirements of the Kusama Bounty “#3 Mobile Signers / Injectors / Walletconnect equivalent”¹.

Important note

Due to the uncertainties around the distribution of the bounty, as it was initially planned to be split up into multiple sub bounties, Papers had to temporarily halt the development in October and November until a viable solution was reached. The development was picked up as soon as a first installment had been made, covering the development cost until that point.

1. Intro

During the requirements engineering phase changes it became apparent that there will have to be a scope change from the initial proposal.

This report outlines the deliverables for the **Phases 3 to 6** of this adapted roadmap.

2. Tasks

The following tasks were completed during the development phases under this bounty.

2.1. Beacon Kusama

Beacon Kusama focuses on enabling wallet and decentralized application developers to build on the Beacon functionalities to simpler interact between wallets and dApps.

Status Report 1

During the requirements engineering phase, conducted in the first part of the work, and outlined in Report 1, the development work was structured.

One main takeaway was that in the initial proposal the mobile native Beacon SDKs for Swift and Kotlin were not in-scope. As many wallets utilize these languages to build their mobile first wallets, the decision was made to change out the browser extensions (Spire) related parts with the integration of the Swift and Kotlin SDKs. The work on these native SDKs focuses on the WalletClient implementation.

Beacon SDK (Typescript)

- Beacon SDK Typescript²

The Beacon SDK implements the following:

¹ <https://kusama.polkassembly.io/bounty/3>

² <https://github.com/airgap-it/beacon-sdk>

- WalletClient – for wallet developers
- dAppClient – for dApp developers

The Beacon SDK v3.0.0 brings full support for multiple blockchain,

The Beacon network and SDK have always been primarily focussed on providing a secure and flexible communication method between dApps and wallets. Blockchain specific functionalities have always been kept to a minimum to make sure external changes eg. blockchain upgrades don't affect the connection.

In Beacon SDK v3.0. this approach was taken to the next level. The blockchain specific logic was separated from the communication logic. This allows for new blockchains and new types of messages to be added more easily.

The first additional blockchain that was added was Substrate, which will support Kusama and Polkadot, as well as any other substrate-based chains.

Additional work has been done in splitting up the Beacon SDK into multiple packages, to allow developers to use only the necessary dependencies.

The following packages were introduced:

```
@airgap/beacon-blockchain-substrate
@airgap/beacon-core
@airgap/beacon-dapp,
@airgap/beacon-transport-matrix
@airgap/beacon-transport-postmessage
@airgap/beacon-types
@airgap/beacon-ui
@airgap/beacon-utils
@airgap/beacon-wallet
```

A dApp developer for example does not need the “beacon-wallet” package thus also making the development process and their application more lightweight.

Beacon Substrate messages

The Beacon SDK utilizes multiple messaging schemes and are for example available in the beacon-blockchain-substrate³ package of the Typescript version.

- PermissionRequest
- PermissionResponse
- SingPayloadRequest
- SignPayloadResponse
- TransferRequest
- TransferResponse

These message definitions are specifically created for the Substrate protocol. An example of a TransferRequest:

```
export interface
SubstrateTransferRequest
extends
```

³ <https://github.com/airgap-it/beacon-sdk/tree/v3.0.0/packages/beacon-blockchain-substrate/src>

BlockchainMessage<'substrate'>

```
{  
  
    blockchainData: {  
        type: SubstrateMessageType.transfer_request  
        scope: SubstratePermissionScope.transfer  
        sourceAddress: string  
        amount: string  
        recipient: string  
        network: {  
            genesisHash: string  
            rpc?: string  
        }  
        mode: 'submit' | 'submit-and-return' | 'return'  
    }  
}
```

Beacon example dApp

The Beacon example dApp acts as a Beacon reference implementation for dApp developers. It is a fork⁴ of the PolkadotJS app and adds the Beacon SDK for request in addition to the PolkadotJS. The example dApp uses Beacon SDK v3.0.0 that has support for Fearless and Nova wallet.

<https://andreasgassmann.github.io/apps/#/explorer>

Beacon Swift and Kotlin SDKs

The following two repositories are relevant for the mobile native Beacon SDKs.

- Beacon Kotlin SDK⁵
- Beacon iOS SDK⁶

The Beacon iOS and Kotlin SDKs handle the WalletClient equivalent of the Beacon SDK (Typescript). Besides the development conducted for this and in preparation to handle multiple protocols, these SDKs were split up in multiple modules. Multiple modules enable the developers to select only the modules relevant for them without having to bloat application with unnecessary dependencies.

The native Beacon SDKs focus heavily on the WalletClient with the assumption that native mobile applications are most likely wallets and not dApps.

The project consists of the following modules:

- core - common and base code for other modules
- client-wallet - the wallet implementation of Beacon
- client-wallet-compatible - a supplementary interface for client-wallet for use without Coroutines

⁴ <https://github.com/AndreasGassmann/apps>

⁵ <https://github.com/airgap-it/beacon-android-sdk>

⁶ <https://github.com/airgap-it/beacon-ios-sdk>

- blockchain-PROTOCOL - a set of messages, utility functions and other components specific for PROTOCOL
- transport-p2p-matrix - Beacon P2P implementation which uses Matrix network for the communication
- demo - an example application

In addition to the developer documentation found in the repository and on the Beacon Docs, both repositories contain example applications where the SDK can be tested.

During the initial phase of integration, multiple wallet teams implemented the Beacon SDKs as a proof of concept, this includes:

- Fearless Wallet⁷
- Nova Wallet⁸

Beacon Node infrastructure setup

As Beacon mainly relies on a matrix⁹ based infrastructure, which allow a decentralized approach at running these nodes, an initial testnet was created with two nodes. Request related to the Kusama integration in this phase will be relayed through these nodes until more nodes will be added when the integration becomes production ready.

- beacon-test-node-1.crystal.papers.tech
- beacon-test-node-2.crystal.papers.tech

Beacon SDKs developer documentation

With the Beacon Docs¹⁰ developers have a repository of collected knowledge for Beacon. With the playground simple examples and their behaviour can be tested out directly on the documentation website.

The Beacon Docs has detailed documentation for all three libraries:

- Beacon SDK (Typescript)
- Beacon Android SDK
- Beacon iOS SDK

3. Spent resources

The following outlines the resources used for the following tasks.

Task	Days	Dollar value
Requirement engineering	3 days	3'600
Beacon SDK (Typescript)	24.5 days	29'400
Beacon example dApp	7.5 days	9'000
Beacon Swift and Kotlin SDKs	24 days	28'800
Beacon Node infrastructure	12 days	14'400
Beacon Developer Documentation	14.5 days	17'400

⁷ <https://fearlesswallet.io/>

⁸ <https://novawallet.io/>

⁹ <https://matrix.org/>

¹⁰ <https://docs.walletbeacon.io/>

Total	85.5 days	102'600
--------------	------------------	----------------

4. Next steps

As there are still multiple tasks that should be done, our suggestion is to create a new proposal outlining the next steps and requesting additional funding to carry these out.

These next steps include:

- Integration of Beacon in the PolkadotJS wallet extension
- Integration of Beacon in the AirGap wallet

5. Resources

- Website: <https://www.walletbeacon.io/>
- Beacon Nodes: <https://status.walletbeacon.io/>
- Docs & Playground: <https://docs.walletbeacon.io/>
- GitHub
 - Typescript SDK: <https://github.com/airgap-it/beacon-sdk>
 - Android SDK: <https://github.com/airgap-it/beacon-android-sdk>
 - iOS SDK: <https://github.com/airgap-it/beacon-ios-sdk>