Treasury Proposal: Unified Ledger APP for Polkadot

Proponent: 14fhPR28n9EHZitNyf6wjYZVBPwKgcgogVjJPTzvCcb8qi9G
Date: 8.05.2023
Requested DOT: TBD at the time of proposal passing (see details below)
Short description: Provide Polkadot with Ledger APP that works with relay chain and every parachain.
Project Category/Type: Infrastructure
Previous treasury proposals: N/A

1. Context of the proposal

Quite a while ago, there was a forum discussion about the Polkadot ecosystem growth points. Several points must be addressed ASAP inside the ecosystem to help wider adoption, improved end-user experience, and new user acquisition. One of the priority points mentioned was the need for hardware wallet support.

This proposal is focused on the hardware wallet. We want to finalize the development of a unified app for connecting Ledger and the Polkadot ecosystem, and it doesn't need to be updated in case of version upgrades on the Polkadot side.

This proposal is related both to Polkadot and Kusama-based parachains. Within the proposal, we seek some retrospective funding for the work already done, plus milestones-based funding for delivering the entire solution. We will consider separate/complementary proposals to audit the solution and the maintenance funding.

Previous discussions:

Please refer to this Polkadot forum post for details and motivation. This forum topic introduces the proof of concept (POC) and demonstrates the application.

Background:

We are the Equilibrium team. We've been active ecosystem participants for two years plus. We've developed a novel DeFi platform with sophisticated on-chain risk management techniques and introduced a Polkadot-native stablecoin EQD.

We conduct annual Polkadefiance events entirely, devoting our time, effort, and funds (without asking for treasury support).

We have been very active in the bridging domain and have introduced the first trustless cross-chain bridge set up jointly with the Multichain team, which leverages Moonbeam's cross-chain smart contract functionality.

Motivation:

We propose a universal solution that will benefit the entire Polkadot ecosystem. All parachains (including common good ones) and the relay chain will get the unified ledger app, which will allow quicker ledger store integration and ongoing upgrade support from the Ledger team. These points are further elaborated in the problem statement section.

Backers:

The proposed solution has been discussed directly with the Ledger team and given the green light on their side.

20+ parachains are already supported for the blind signing mode by the APP:

Polkadot-based: Polkadot, Statemint, Acala, Equilibrium, Bifrost, Astar, Nodle, Zeitgeist, HydraDX, Manta, Ajuna, Bitgreen, Origintrail, Interlay

Kusama-based: Kusama, Statemine, Karura, Genshiro, Bifrost, Shiden, Basilisk, Calamari, Bajun, Kintsugiб MangataX

2. Problem statement

There should be one universal Ledger app that will work with all parachains. The closest analog is the Cosmos ecosystem, where one application supports most projects. A less similar analog is an ether application that works equally well for all Ethereum projects and EVM networks (Ethereum, Arbitrum, Polygon, you name it)

Current problems:

1. The Ledger team is overwhelmed with work, and every Ledger app update requires a security audit. The Ledger team explicitly stated that they don't want to be in the business of constantly updating each parachain app every time it needs an upgrade.

2. Unique derivation path for each app (a unique account for each relay chain and parachain). Which leads to many problems:

a. Most applications do not have derivatives path selection functionality. So users, who transfer tokens from Kusama to parachain, for example, have their account changed and lose access to their tokens. The entire flow is not obvious to the average user (need to export mnemonic with correct derivation path).

b. The inability to participate in a crowd loan with a Ledger with guaranteed access to its rewards on a parachain without exporting mnemonics to less secure environments (for example, an extension, a mobile application). Most likely, a new project that participates in the crowdloan will not have its app built for a long time, because of which users will have to wait and get a negative experience from the entire ecosystem.
c. Even if developers of parachain/relay chain applications will include functionality for selecting derivation paths in their Ledger applications, this will cause all applications to need to upgrade to support new parachains. The alternative is for users to set the derivation path manually, which is complicated for the average user.

3. The Ledger app is a full-scale decoder, where pallet numbers and their respective methods are hard-coded. This leads to the following problems:

    a. Some transactions become unavailable after the runtime upgrade if there are changes to pallet indices and/or changes to extrinsic types.

    b. These problems in conjunction with slow Ledger support response lead to the fact that users lose access to their assets for a prolonged period.

3. Proposed solution

Make call data decoding on the device optional. The main reason it is necessary to make a distinct Ledger app for each parachain is to decode the call data of every extrinsic.

Ethereum Ledger app, for example, only decodes some of the transactions, except ERC-20 transfers and some popular, frequently used transactions. That is, the user checks data in the Metamask, for example, and approves it on a Ledger.

In the case of the Polkadot ecosystem app, the call data is already being decoded in apps/mobile apps. To be sure that the Ledger has corresponding call data, it's possible to display the hash of the call data both in the extension/mobile application and on the Ledger device. This will eliminate the need to implement an extra decoder for each parachain after the runtime upgrade. An additional security measure that every user

should engage in is the ability to compare the transaction hash provided by the user's wallet / local copy of polkadot.js with the one user sees on the Ledger device itself.

This approach has been confirmed with the Ledger team, if we provide the decoding for balance transfers and keep everything else in the blind sign mode, we can proceed with the Ledger Live application. If the balances transfer translation decoding fails for some reason (the pallet got updated, or some extrinsic have changed their signature), the Ledger device will display an error and suggest signing the transaction hash. This way, we eliminate the need for frequent app updates, even in corner-case scenarios.

The prerequisites are:

- Fulfill security requirements
- Listing in Ledger Live in the developer mode
- Performing a 3rd party security audit of the APP

Comment on the 3rd party audit

We can't assume the cost of the 3rd party audit before we make all of the changes to the APP code outlined in this proposal. So we exclude the audit budget and timeline estimation from this proposal and will apply a complementary proposal for audit funding purposes.

Pallets supported for clear signing

We've compiled a comprehensive list of different asset-related pallets/modules that need to be supported for the clear sign. The full table is available in this spreadsheet. Parachains that display parachain names instead of genesis hash are marked in color. The overall balances usage breakdown by parachain looks the following way (both for Kusama and Polkadot):

Pallet	Used by # of parachains
Balances	64
Assets	25
Tokens	20
Currencies	16
EqBalances	2
EncointerBalances	1
CoreAssets	1

Adding Balances pallet is pretty straightforward, while the support of other pallets will require additional RnD on a case-by-case basis. We will use this fact to outline the grant milestones.

Similar solutions

Zondax proposed something similar, yet his plan is costly, takes considerable time, and there is no open information on complete funding and costs. It has a per-parachain subscription plan and may cost each individual project a considerable budget. At the same time, the proposed solution doesn't deliver a unified app for all parachains, making support from Ledger very demanding. It lacks universal applicability and raises concerns that have been addressed by comments on the forum (link provided above).

Milestones

#	Milestone	Target	Notes
1	Parsing Logic	We will develop a custom app logic, which depending on the parachain and its supported assets modules/pallets will invoke different parsing schemes.	Please refer to the parachains spreadsheet for a comprehensive description of parachains variations of balances. To be released into the public GitHub repo.
2	A clear sign of Balances pallet	We will develop parsers for the most common Balances pallet used by the majority of parachains. We will also develop parsing of eqBalances (Equilibrium's double entry bookkeeping)	To be released into the public GitHub repo.
3	A clear sign of other pallets	We will develop parsing logic for other commonly used asset pallets, namely: Currencies, Tokens, and Assets	This is the most demanding task, as it requires a deep understanding of the logic behind Currencies, Tokens, and Assets pallets and their variations among different parachains. To be released into the public GitHub repo.

4	Full test coverage + CI / CD	We will cover the APP with tests to explicitly demonstrate correct app behavior with all available parachains and respective balance modules.	This is a Ledger pre-listing requirement. Tests along with instructions on how to run them will be available in a separate folder in the public GitHub repo. CI / CD will be configured for the public GitHub repo.
5	Ledger STAX support	The entire app will be ported and tested with the new Ledger STAX device. At the end following devices will be supported: Ledger Nano X, Ledger Nano S, Ledger STAX	This is a Ledger pre-listing requirement Tests along with instructions on how to run the app on Ledger STAX will be made available in a separate folder in the public GitHub repo.
6	Developer mode listing	We will list the app in developer mode in the Ledger app store and fix and address all the comments/concerns of the Ledger team required to get the app approved for listing.	We will work closely with Ledger Security and the dev team throughout the listing process.

Budget

Below is the detailed breakdown of cost estimation for each milestone above. Additionally, we seek retrospective funding for some of the work we've done so far (the first milestone in the below table).

An hourly rate of 180 € is applied.

Task / Milestone	Hours	Cost	Notes
Retrospective funding	160 hours	31,680 €	This is retroactive funding that we're seeking for ● RnD, development of the proof of concept of the APP ● Demonstration of the APP ● Multiple discussions with the Ledger

			team and across the ecosystem to come up with the plausible solution that benefits all equally. • Support of 20+ parachains inside the current APP version. • Analytical work related to different assets modules and how to support them
Parsing Logic	160 hours	31,680 €	This is funding that we're seeking for the improvement of the app to support balances/transfers transaction decoding on the Ledger device.
A clear sign of Balances pallet	160 hours	31,680 €	See the corresponding milestone above.
A clear sign of other pallets	300 hours	59,400 €	See the corresponding milestone above.
Full test coverage	300 hours	59,400 €	See the corresponding milestone above.
Ledger STAX support	160 hours	31,680 €	See the corresponding milestone above.
Developer mode listing	160 hours	31,680 €	See the corresponding milestone above.
COSTS		252,000 €	
Uncertainty risk	10%		We seek an additional 10% of the costs to provide some room for the highly uncertain nature of the task and the need to support the 70+ parachains. This uncertainty is included in the final cost estimation for each milestone. (e.g. hours * hourly rate * 110%)
TOTAL COSTS		**277,200 €**	

We request a two-part payment based on the above budget:

- 154,440 € for retrospective funding and the first 3 milestones: (Parsing Logic, Clear sign of Balance pallets, Clear sign of other asset pallets). Payable on proposal approval.
- 122,760 € for the last 3 milestones: (Full test coverage, Ledger STAX support, Developer mode listing). Payable after delivery of milestones 1 through 3.

  The exact amount in DOTs will be calculated when the proposal is submitted on-chain using the 7-day average found on subscan and the current USD <-> EUR exchange rate found on xe.com. The destination address is

  14fhPR28n9EHZitNyf6wjYZVBPwKgcgogVjJPTzvCcb8qi9G

4. Reporting and delivery

#	Deliverable	Timing	Acceptance criteria ~~Funding conditions~~
0	Blind sign POC	Completed	• If the proposal gets approved
1	Parsing Logic	2 weeks (10 business days) from the moment of proposal approval.	• The new code is uploaded to the GitHub repo. • Comments on status are provided to the initial proposal post. • All code-related comments/issues are resolved.
2	A clear sign of Balances pallet	2 weeks (~10 business days) from the moment of funding of the previous milestone.	• The new code is uploaded to the GitHub repo. • Comments on status are provided to the initial proposal post. • All code-related comments/issues are resolved. • Demonstration video/post is provided for Ledger Nano S and

			Nano X devices.
3	A clear sign of other pallets	4 weeks (~20 business days) from the moment of funding of the previous milestone.	• New code is uploaded to the GitHub repo. • Comments on status are provided to the initial proposal post. • All code-related comments/issues are resolved. • Demonstration video/post is provided for Ledger Nano S and Nano X devices.
4	Full test coverage	4 weeks (~20 business days) from the moment of funding of the previous milestone.	• Tests are uploaded to the GitHub repo. • Comments on status are provided to the initial proposal post. • Readme.md provides examples and clear explanations on how to run tests.
5	Ledger STAX support	2 weeks (~10 business days) from the moment of funding of the previous milestone.	• New code is uploaded to the GitHub repo. • Tests are uploaded to the GitHub repo. • All code-related comments/issues are resolved. • Demonstration video/post is provided Ledger STAX device
6	Developer mode listing	4 weeks (~20 business days) from the moment of funding of the previous milestone.	• Comments on status are provided to the initial proposal post. • Users are able to download the app from the Ledger app store under the development mode.