Polkadot

Proposal: Multix - A simple interface for complex multisigs

Proponent: Tbaut
Beneficiary: ChainSafe Systems
14gaKBxYkbBh2SKGtRDdhuhtyGAs5XLh55bE5x4cDi5CmL75
Date: Feb 2023
Requested DOT: 23317 DOT, representing 151 213 USD, EMA-30 DOT (6.485$/DOT) using Subscan on the day of submission
Short description: Multix is an interface to manage complex multisigs. It is an early version, live today. This proposal is to extend it and make sure it caters for most business needs.

Polkadot and Kusama have the lego pieces ready to use for enterprises and DAOs to create multisigs that are flexible enough for their needs. However, the actual day-to-day handling of these multisigs is a huge burden. It is not user friendly, error prone, and time intensive. From our interviews with businesses and DAOs that use multisigs, they all use at least one proxy, and often even many more. This gives them the flexibility they need, but it also makes the management of the multisig even harder. Also, we believe that flexible multisigs should be accessible to everyone, not only businesses. Multix was built to answer those issues. In a record time, the team has built a trust-minimized interface that can already be used today to create very flexible multisigs with a good UX. While it is an early version, it showcases the power of what can be done. If you want to know more about it see this article.

Multix is flexible - based on proxy, trust minimized - no application specific server or database and, is already live!

Context of the proposal

ChainSafe Systems founded in 2017 in Toronto, is a Canadian blockchain development studio with more than 100 employees worldwide. Our vision is to move the various blockchain ecosystems closer, building bridges and increasing client diversity by providing alternative protocol implementations. Chainsafe has been building web3, and in particular Polkadot products and infrastructures for years. They are currently involved in many Polkadot related projects, from enhancing the privacy of the Polkadot-js extension to building the Go implementation of the Polkadot host called Gossamer. Many team members are also part of the Kusama fellowship.

Initially funded through a Web3 Foundation grant for one month, the team has demonstrated

that it was possible to build a trust-minimized interface that is easy to use to manage complex multisigs.

Problem Statement

Many teams in the ecosystem, Chainsafe included, need to manage DOTs securely. While other ecosystems such as Ethereum have very well known and respected non custodial multisig solutions such as Gnosis Safe, no such solution exists for Polkadot. Businesses and tech-savvy users can create flexible multisigs, but their management is very cumbersome, and error prone.

Multix is not the first product trying to tackle this issue, but it is the first to actually answer businesses' needs, and doing it in a trust minimized way. Also, these are not just words, an early version is already live!

When using a multisig, you may encounter these issues

- Building a Multisig with Polkadot-js/apps may not be very straight-forward
- Users need to first add all the signatories in their address book
- Seeing and reviewing multisig transactions requires external communication between signatories, and the last user that will submit the transaction needs to copy/paste the extrinsic's information

These issues are well known by multisig users nowadays, and Polkadot-js/apps, while not being the most user friendly interface, actually still handles transactions well. No need to go to the extrinsics tab to craft a multisig transaction.

Now, the number one problem that is mentioned by the persons we interviewed, is the fact that any multisig created by a set of signatories is deterministic. Meaning that if you wish to change one signatory, or even only change the threshold of the signatory, then the multisig address will change. Because Businesses, DAOs, and even single users prefer to have one address and keep it for a long time, they need to use a proxy. Read the Multix article to get a mental model about proxies.

While proxies are very powerful, they are also cumbersome to use. Creating or using a complex multisig involving one or more proxies is where users actually need to start building their transactions with the extrinsics tab from Polkadot-js/apps. The transactions are very hard to decode with human eyes because the information about the call that will result from the transaction is buried in other complex calls. This is a security threat.

Multix has been built with a proxy at its core, to answer these issues. An early version is live, on Rococo, we want to enhance it and bring it to Polkadot.

Solutions & Objective

Multix is very young, yet it has already pushed the UX boundaries for managing and handling multisigs.

- Creating a new multisig takes seconds.
- Users get access to a multisig with a proxy out of the box allowing them to change the signatories or the threshold at any time without having to move their funds or even communicate about this change.
- New multisig transactions can be made easily from the interface
- Proposals appear automatically on the interface, the calls are decoded, shown in a user-friendly way, and can be reviewed and signed for easily.
- No need to pass around any data. Signatories don't need to be in contact.
- Multix is open and interoperable. It is compatible with multisigs created manually, it also supports multisigs without proxies.
- No server other than a Subsquid indexer is used. All the information displayed by the interface comes from the chain.
- It's open-source already, Multix faces some competition, but if our code is re-used, it benefits the ecosystem (Please give us attribution though!).

This list is what Multix already offers. While it is already very useful, it is only an early version and the beginning of a long journey. Many feature requests have been made and we list them in the implementation & Milestones section below.

The goal is to build as much as what has been requested, to cater for most use-cases.

Implementation & Milestones

General project phases, mostly running in parallel

1. Design and UX development
2. Features development (see below)
3. Continuous UX interviews and feedback

4. General Multisig education content such as regular article
5. Security R&D

Based on the feedback we collected, here are the most demanded features in no particular order.

Features	**Description**
Design	This is a parallel work to all the other features. Multix currently does not have any design identity. While it worked for an early version, it is important to have its own logo and identity.
Enhanced UX at the creation flow	When a user is done creating a multisig. The time between the multisig being created on-chain and it being visible on the interface and it being visible should be shortened, and we should provide a better UX.
Key rotation flow	This is most probably one of the most important features. We should provide an easy flow to change the signatories or the threshold. Once accepted by all signatories, the multisig is swapped and the proxy address remains unchanged.
Mobile support	While the initial development had this in mind, not all elements are in place, such as a sidebar for smaller screens.
Transaction interface for voting, staking, setting an identity	The interface should make the most common actions first-class citizens. Right now Multix shows in a user-friendly way a form to send tokens. Also, every token transaction is displayed nicely with account destinations and token amounts denominated with the right decimals. Our research shows that the most common actions are Staking, voting, and setting identity. We need to extend the interface to these needs.
Support other networks	As an early version, Multix was deployed on Rococo. The infrastructure should be put in place to be able to switch between networks, starting with Kusama and Polkadot.
Interface for application specific proxies	Some DAOs already have such an architecture: one proxy has several multisigs behind it, to perform specific actions. The full control of the proxy is given on a Multisig with a high threshold. A governance proxy could be set on a proxy from another multisig, with a lower threshold, so that the funds on the proxy could be used for voting requiring fewer signatures

Enhance UI security	Multix is an interface to perform critical actions. Businesses need to use it with confidence. It should be deployed on IPFS. However, regardless of the security measures taken to deploy Multix on IPFS, this should not prevent users from verifying every signature, every time. Ideally on an offline device or hardware wallet.
Proxies for each signatory	Some businesses have very advanced needs and use cases, requiring 1 proxy in front of each account. Although this adds a lot of complexity to the setup, and the interface, this need could be catered for.
Arbitrary extrinsics	Multix already plays well with any multisig proposal made outside of its interface. You can craft an extrinsic in Polkadot-js/apps and Multix will show it and allow you to sign it, eventually requiring `callData` if needed. However, switching interfaces is not ideal and advanced users may be willing to be totally in control.
Historical transaction	Reviewing who voted when, and on what is possible using an explorer today, but having it nicely displayed in the interface would be greatly appreciated.

Team

1. Project manager
2. Developer
3. Designer
4. Devops

The initiator and main developer behind Multix is Thibaut Sardan. He is an ex Parity developer who has contributed to major ecosystem tools from being the first developer of Polkassembly, to one of the main maintainers of Polkadot-js extension, Parity Signer, or its UX centric fork Stylo. Thibaut has a great interest in building user friendly yet highly complex web3 products, constantly bringing security closer to usability, so that users do not have to choose.

Timeline

This proposal is aimed at funding the team for 5 months,

starting from when the proposal passes. While it is hard to predict this far in the future the exact amount of features that will be implemented, we will implement as many features as possible, from the above list.

Payment Details

The requested amount for a 5-month ongoing development grant is 151 213 USD in DOT on the day of submission using the EMA30 rate by Subscan. The allocation is asked for as a single, up-front payment to cover our ongoing development costs and for the team to be able to provide a runway to their developer resources.

The following resource table outlines the expected costs for the Multix team. The table consists of:

- the current base cost to operate the team, with the number of hours for each month,
- A 36% operational overhead to cover the costs of running the project as a team within ChainSafe Systems (e.g., operations, offices, equipment, tooling, infrastructure, legal, travel)
- We estimate that the number of monthly hours for Design will be high at the beginning and decrease a little with time. Similarly, the number of Devops hours will be significant as we make Multix available to more networks and will decrease with time.

Month	Dev h	Design & UX h	PM h	Devops h	Base cost	Cost + 36%
1	160.95	87	13.05	43.5	27 405$	37 271$
2	160.95	87	13.05	21.75	25 448$	34 609$
3	160.95	43.5	13.05	8.7	20 358$	27 687$
4	160.95	43.5	13.5	4.35	19 967$	27 154$
5	160.95	21.75	13.05	4.35	18 009$	24 492$
Total	**804.75**	**282.75**	**65.25**	**82.65**	**111 186$**	**151 213$**

Beneficiary Information

The beneficiary of this proposal is ChainSafe Systems Inc, Toronto, Canada. The on-chain address is 14gaKBxYkbBh2SKGtRDdhuhtyGAs5XLh55bE5x4cDi5CmL75. The funds' manager is Hatcher Lipton, COO at ChainSafe Systems, reachable through hatcher@chainsafe.io for any questions regarding the payment conditions.

For any questions about the technical nature of this proposal, you can message me, Thibaut, at thibaut@chainsafe.io