Snowbridge Funding Proposal

Funding proposal for operation as a common-good bridge on Kusama and Polkadot

DOT Address: 12UDxESUsqUZi5R5MpovTZLpAWZXPVWHJQkEprcT6MVdKh3A
KSM Address: Gbb7iGbd6Q7o1fGcT2Re4L6RXJ3U7A32itSLMjsUJzmq5LD
Requested USD on Kusama for first on-chain proposal: USD 612,000.00 based on EMA7 rate by Subscan.
Requested USD on Polkadot for first on-chain proposal: USD 2,448,000.00 based on EMA7 rate by Subscan.

Overview

Executive Summary

Snowbridge is a fully trustless bridge between Polkadot networks and Ethereum.

The proposal covers a request for a 2 years worth of runway for all operations for the Snowbridge team *(1 year retroactive, 1 year in future),* with the goal to launch as a common-good bridge on Kusama and Polkadot.

Our bridge will launch on the Bridge Hub, a common-good parachain that is being developed by Parity for Kusama and Polkadot.

It also includes a motivation for supporting and promoting Snowbridge, a long term roadmap, governance and economic plans as well as incentives for successful execution, launch and running of the project.

Purpose

What is Snowbridge

Snowbridge is a set of libraries and services for trustless bridging between Ethereum and Polkadot.

Snowbridge brings real innovation to cross-chain interoperability, not just for Polkadot, but for the wider blockchain ecosystem, being the only Ethereum bridge that is fully trustless, decentralised, fast and cheap. Snowbridge runs entirely using on-chain light clients with no trusted third party involved. Many other Ethereum bridges exist, however they all compromise in some way:

1. Some Ethereum bridges trade off both trustlessness and decentralisation by relying on a set of trusted parties in a multisig (eg: Chainbridge, Anyswap, Polygon, Terra). These risk both being regulated away and involve massive counterparty risk for all users. They also completely degrade the trust and decentralisation of any assets that flow through them.

2. Some Ethereum bridges mitigate this counterparty risk by adding a secondary collateral, insurance or bond/slashing system to their multisig or optimistic design. In Axelar, LayerZero, Chainlink and Gravity Bridge users are not insured, and although there may be slashing to disincentivize fraud and partial collusion, if a majority of signatories collude then fraud cannot necessarily even be detected. These mechanisms do make them more trustless, but trade off on capital efficiency and will lead to higher hidden costs. They also thus degrade the capital efficiency of any assets that flow through them. They often also may depend on fraud proofs and so have long confirmation times of at least a few hours which introduce a systematic risk if an attacker can censor them for longer than their optimistic confirmation time. They are also collateral-constrained, and can also only secure as much state and value as their collateral allows for. This also means that they need to be able to accurately measure the economic value of the assets and state they secure, as well as their exchange rates, which brings an additional challenge.

3. Some Ethereum bridges are fully trustless with full on-chain light clients, but still use an optimistic design to reduce computational costs (eg: NEAR Rainbow bridge). This trades off latency and confirmation time, and also introduces the same systematic risk if an attacker can censor them for longer than their confirmation time.

Snowbridge aims to make no compromises and have minimal trade offs. It will be more trustless than bridges in category (1). It should likely be cheaper, faster and more trustless than bridges in category (2). It should be faster and more secure than bridges in category (3). At worst, the main trade off once fully deployed and completed is that it could be slightly more expensive than bridges in category (1) and (3).
The few other cross-chain interoperability projects today similar to this include only protocols like IBC and eventually XCM which are not yet practical to run on Ethereum.

Deploying Snowbridge may take longer than other bridges, but as we all learned, the tortoise beats the hare.

Snowbridge is general purpose

Snowbridge is also general-purpose and bridges arbitrary state, rather than assets. This means it can connect Ethereum dApps to non-ethereum chains, allowing for cross-chain smart contract calls and becoming a platform for cross-layer-1 applications rather than just an asset bridge.
The light-client based design also means that Snowbridge's trust model is independent of the assets it bridges. It can bridge any value of assets and scale up TVL without any major bottlenecks or barriers related to some underlying collateral or token value.

Systematic risks to the Polkadot ecosystem from existing bridges

There is high demand for Ethereum bridges across parachains and other substrate chains and some chains have already started to introduce more centralised bridges as a stop-gap. Without Snowbridge, these stop-gaps introduce multiple systemic risks to Polkadot and the Substrate ecosystem as a whole. This makes Snowbridge a critical component for the Polkadot and Substrate ecosystem. Specifically, Snowbridge mitigates:

- **Counterparty risk:** Existing bridges all use some kind of multisig, meaning that assets flowing across those bridges now have counterparty risk in those signatories. A hack or loss of keys could result in major asset losses to those substrate chains and their users and lead to wider sentiment issues for Polkadot as a whole *(eg: Much like how there is now a sentiment around dApps running on Binance Chain being much more dangerous due to repeated hacks and low-quality projects)*
- **Regulatory risk:** These multisigs also introduce regulatory risks onto the parachain teams that use them and the multisig participants. They could be regulated and so forced to shut down or restrict access, with implications for the security and decentralisation of all assets that flow through those bridges and parachains.
- **Product-Market fit risk:** The biggest risk is actually not legal or trust related, but cuts into the core value proposition of Polkadot. Polkadot's value proposition is that it leases out high security, trustless and decentralised consensus to other chains. The existing bridges degrade these trust and security properties of any parachains that use them, as most value that exists on those parachains will flow through their Ethereum bridge that is not secured by Polkadot. If these existing bridges do remain secure and do not become

impacted by regulations, while the market continues to accept their trust assumptions and design, then this undermines the whole purpose of Polkadot. If the existing bridges remain successful, the parachains that use them will be effectively wasting their slot, gaining minimal additional security while paying for a Polkadot slot/lease and would become better off and more efficient if they eject from Polkadot to become a sovereign chain in future. There is a systematic risk to Polkadot if it does not influence the existing parachains and their users to care about the trust properties of their bridges.

New Opportunities for Polkadot

Snowbridge's main eventual deployment will be on Polkadot, and live under Polkadot consensus. This means that any assets that flow through Snowbridge are secured by Polkadot and essentially bring new business to the network. As Snowbridge generates more interest and asset flow both into and out of the Polkadot ecosystem, it makes use of Polkadot's security and extends Polkadots security outwards.

Assets flowing into Polkadot

The biggest expected use case is asset flow from Ethereum into Polkadot's ecosystem, as people start to tap into the more secure, faster and cheaper protocols and parachains that live on Polkadot. The opportunity here is straightforward and easy to understand. More assets means greater user activity, increasing network effects and positive feedback loops.

Assets flowing out of Polkadot

The basic example of this is assets that originate on Polkadot (eg: Acala/Moonbeam assets) that flow to Ethereum and get used within Ethereum protocols. This basic use already extends Polkadot's security blanket outwards into the Ethereum ecosystem. As these assets become used in DeFi protocols or AMMs providing liquidity, the payoff that an attacker stands to gain by manipulating or controlling these assets increases. This implies that the effective cost of fraud for these assets needs to increase too in order to retain a balanced theoretical crypto-economic security.

Increasing the cost of fraud should directly lead to increased revenue for Polkadot in the long term. Of course, at the moment this balance is not that relevant in practice and so may be difficult to understand, so let's elaborate a bit further:

A more dynamic security marketplace

- Currently, Polkadot sells security under a fairly static model. A chain can buy a fixed-lease slot for a period of time. There is a slot auction, but once a slot is acquired, it gives the same kind of fixed term security based on turnkey slashing conditions and incentives to all chains. A particular chain is not able to configure its required slashing conditions and levels in a bespoke way. This means that all chains get the same kind of security and share it equally, irrespective of how much security they actually need.

- The above setup means that some chains are using their security more efficiently than others and getting a better deal for the cost they pay. Similarly, some chains are overpaying: A chain that has a smaller amount of assets at risk may only need to slash a smaller portion of Polkadot's stake to prevent fraud, and may in fact be better off moving to a cheaper chain like Kusama. Alternatively, it may want to specify a maximum amount of slashable stake that it reserves from the relay chain and get a better deal for reserving less stake. This kind of configurability is not possible at the moment as Polkadot's slashing conditions are the same across all chains.
- Parathreads will come out in time, which do make the market more dynamic in terms of how long/how often a chain purchases security, but longer term we can imagine that a dynamic market around slashing conditions and reserved stake could also become possible.
- Longer term as there are more chains selling security and more competitors, some metric like value-for-money-per-reserved-stake could become relevant for how and where chains choose to buy security from, especially if layer-2/shared security style standardisation evolves to allow a more free, flexible framework-agnostic market. We're already starting to see this possibility today in new chains that are looking to sell security like Octopus Network and Interchain Security over IBC.

Imagine long term a dynamic market like this does exist, it should be more clear to see how Snowbridge's asset flow out of the Polkadot ecosystem will increase demand and eventually revenue for Polkadot's security.

History

Snowbridge originally started as a W3F grant project in early 2020. The design targeted a security model with minimally trusted actors and a stronger focus on cryptographic proofs and data structures. It completed the initial grant with a working bridge in September 2021, however given dependencies on new features in Polkadot (BEEFY) has been delayed in our ability to launch. Since then, the team has continued to run independently and continued to maintain and update the existing codebase, scale up with new developers, and make improvements with new functionality to the bridge, including shipping a new architecture to support post-merge Ethereum.

Technical Details & Design

This proposal is primarily focused on governance, economics and launch. For an overview of the technical design, visit Technical Design.

Our main repository is at https://github.com/Snowfork/snowbridge

Economics

Our common-good bridge will act as a hub for cross-chain communication between Polkadot parachains and Ethereum.

In the long-term, the bridge aims to be self-sustaining by generating revenue from usage of the bridge.

The motivation for a common-good bridge also aligns with our motivation to allow the bridge to operate cheaply with as minimal overhead as needed and avoid rent-seeking unless needed, ideally making resources available to everyone fairly as a public good.

Operational and Usage Costs

This section covers costs that the bridge and its users will need to cover in order for the bridge to function. Specifically, these are costs that cannot be directly controlled by on-chain components, as they are charged by outside parties and services.

We classify these costs into 3 categories:

- **Operational costs:** These are costs that are essential and required to be covered in order for the bridge to remain operational and secure. If for whatever reason these costs cannot be covered, and nobody is willing to pay for them, the bridge will not be sustainable. These are costs that should be shared and covered as a common good.
- **Individual usage costs:** These are costs that an individual user of the bridge may incur as part of their usage of the bridge. These costs can be paid per-user/per-parachain, or on a case by case basis, or on demand as needed.
- **Shared value add costs:** These are costs that also apply based on usage, however that may be shared across users or support multiple users when paid.

Operational costs

There are 3 primary operational costs. These will need to be covered long term in order for the bridge to remain operational and secure:

- **Real-world costs:** This includes all the costs associated with off-chain activities, including running the bridge, infrastructure, maintenance, development, support and all the kinds of things needed to operate a complex system and run a real-world team to do so. It also includes potential requirements for incentives and upside for all entities involved in this. This cost will need to be covered by long term treasury funding, and is detailed later in this document as the core funding request for this proposal.
- **BEEFY header relay:** This is the cost for keeping the beefy light clients on the ethereum side of the bridge up to date with the latest block headers from Polkadot. It involves paying for Ethereum transactions in ETH/gas. The cost is block/time based, ie, each time the header is updated a cost is incurred and so is tied to the desired maximum

confirmation time of the bridge. This is a flexible cost, as the bridge can choose how frequently to update the header relay and finalise the bridge., but we will need to ensure we cover at least a baseline cost for a minimum acceptable confirmation time (eg: 1 hour) to ensure security. This cost is also impacted by Ethereum gas prices. In practice, we expect between $300,000 and $1 million per year to be sufficient to cover this, but it does depend on gas fluctuations. This cost is not part of this proposal, and an additional future proposal closer to launch time for this liquidity should be expected.
- **Ethereum header relay**: This is the cost for keeping Ethereum light clients on the Polkadot side up to date with the latest beacon and execution headers from Ethereum. It involves paying for Parachain transactions in DOT. We expect this cost to be significantly cheaper than the BEEFY header relay, given the lower transaction costs associated with Substrate-based chains.

Individual Usage Costs

These are costs that apply each time the bridge is used.

- **Incentivized Channel:** This is the cost for submitting a message on-chain and paying the fees imposed by our incentivized channel. This cost can be covered on a case by case basis, per user/per parachain and does not need to be covered globally, as described above.
- **Basic Channel:** With the basic channel, every user has their own message delivery lane, and so cost can be charged and covered independently by individual users or parachains themselves. Similarly, any user can run a relayer for themselves. Using the basic channel may require more technical effort for users and may have slightly more cost overhead at scale with a large number of users, but it does open a completely trustless, independent path for users who want that and has other benefits, so we do hope to automate this for all users longer term.

Shared Value Add Costs

These are costs that are beneficial for the bridge if they are incurred, and are likely to benefit all users, but they are not critical hard requirements:

- **Bridge acceleration:** Beyond the baseline BEEFY Header relay costs for long-term confirmation, it is possible to accelerate the bridge to get faster confirmation times. An additional cost can be paid to submit a new BEEFY Header at any time, accelerating the bridge to confirm it faster than the baseline confirmation time. The bridge is accelerated globally, and all users and all messages benefit from this acceleration. The cost could be covered globally, but could also be paid on demand whenever needed, by any user or parachain. If there is demand from Polkadot users for consistent, reliable acceleration, then the treasury could cover this in a future proposal.
- **Incentivized channel:** Our bridge supports two different kinds of message channels, a basic channel where every user’s message is independent, and an incentivized channel

where users share the same channel**.** With the incentivized channel, all user's costs must be covered in order for the channel to function.

  - **Execution cost:** In order for users to send messages across the bridge to Ethereum, all messages need to be executed there, with their gas fees covered. This cost is covered by users when they submit a message, so it should always be covered.
  - **Relayer incentive:** Besides for the cost paid to submit, we do still need someone to relay the messages. The incentivized relayer is permissionless and can be run by anyone, but incentivizing the running of a shared relayer to support all users would be of benefit. This should also only require a single honest party, and that the operational cost to run is low, so this will be minimal and will be added as an addition in a future liquidity proposal.
- **Insurance and bug bounties:** We may want to cover some level of insurance for assets within the bridge, as well as bug bounties to incentivize white-hat hackers to reveal any potential bugs/issues they discover. This will be included in a future liquidity funding proposal closer to time of launch.
- **Auditing:** We will want to cover costs for various audits of the bridge. This is detailed later in the document, as part of this proposal.

Parachain-controlled costs and potential sources of revenue

These are costs that are controlled by the parachain, and are potential mechanisms that the parachain can use to extract value from users and generate a source of revenue.

The primary resource that the bridge will be providing is cross-chain messaging, and so message throughput is the primary scarce resource that will be demanded.

We aim for the bridge to ideally only charge users based on market demand, and to prevent sybil attacks. It will initially aim to operate freely with funding from the treasury, but as demand grows, become self-sustaining.

Cross-chain message fees

These are fees that can be charged for cross-chain messages. They will likely be priced on resource usage and message size.

We have considered a fee that is based on the actual contents of a message, for example, charging a basis-points/percentage fee of the value when fungible assets are transferred, but this is unlikely to be effective given that the bridge is designed to support arbitrary message transfer and arbitrary applications, so sticking to resource-based fees is likely best

Parachain transaction fees

The bridge is based on a hub model, where all messages between Polkadot and Ethereum will flow through the bridge parachain. This means that transaction fees for including transactions into the Parachain can also be a source of revenue.

XCM fees

The bridge will communicate with other parachains via HRMP and require XCM messages as part of usage. This is another mechanism where fees can be captured.

Pricing model

The above section describes mechanisms through which value can be extracted, but it does not specify a pricing model. Revenue is sourced through usage of the bridge, but there are still various models that can be used for pricing, including:

- **On demand, Pay as you go:** Charge end users and parachains on a pay as you go basis, with each message, transaction or XCM they send on a case by case basis. This could be priced as a fixed fee, updated infrequently, first come first serve, or could be priced via a typical fee market model like Ethereum’s gas market.
- **Presale model:** Similar to the relay chain slot model, whereby parachains can reserve a guaranteed resource allocation for themselves long term, up front, by pre-purchasing resources for a period of a few months.
- **Free:** The parachain could allow some messages to be free, or allocate a portion of resources to a free bucket. In blockchains, a fee is typically required as a DDoS protection mechanism, so with a free bucket we would need an alternative mechanism for this (eg: a free allocation per user based on a sybil-resistant identity like Encointer’s common good chain).

The ideal configuration of these options will heavily depend on the use cases and demands from applications and the market, so deciding on it this early is likely not the best choice. Having said that, we do believe that in order for the bridge to remain censorship resistant and trustless, it should ensure that some throughput is made available as a public good, rather than allowing for a few big parachains to buy up all the resources.

We imagine that the long term pricing model will involve splitting resources into a few different buckets, with a mix of pricing models that is likely to look very similar to how the Parachain vs Parathread pricing model of Polkadot itself is structured.

Model for Initial Launch

If users or teams send messages via our basic channel, they can relay messages themselves at their own expense. This provides flexibility for advanced integrators who for example may wish to subsidise the bridging activity of their users.

If individual users send messages via our incentivized channel, then they will be charged fees as described below.

Parachain→Ethereum direction

1. Outbound messages are accepted and processed by the parachain on a first-come, first-served (FIFO) basis.
2. There is a maximum limit on the number of messages that can be accepted every commitment cycle (1 to a dozen blocks). Beyond this limit, extrinsics which attempt to submit more messages will fail. Users will need to retry these extrinsics until their message is accepted.
3. The price (in wrapped Ether) for accepting these messages is fixed, though may be changed by governance. There will be considerable margin in the price to accommodate risks of sharp gas price fluctuations.

Depending on our delivery progress, we may launch with the automated Gas Price Estimator for dynamic pricing of messages, to better track gas prices on Ethereum.

Fees captured by the parachain will go into a treasury pallet and be used to cover the above operational and usage costs.

Ethereum→Parachain direction

1. Outbound messages are accepted and processed by our Ethereum contracts on a first-come, first-served (FIFO) basis.
2. There are no limits on the number of messages that can be accepted per block.
3. The price (in wrapped DOT) for accepting these messages is fixed, though may be changed by governance.

Usability

We understand that charging wrapped Ether and wrapped DOT for transfers may impose a burden on usability. However as a trustless and decentralised bridge, we can't rely on any fee model that depends on centralised oracles.

The usability can be mitigated in frontends by allowing users to buy wrapped DOT from Uniswap before submitting their bridge transaction. Since Uniswap doesn't rely on Oracles at all, this still preserves the decentralised nature of the incentivized channel.

Governance

Overview

As a trustless bridge, we need our governance mechanisms to be appropriately decentralised, with checks and balances.

Since we have on-chain components on both Polkadot and Ethereum, we also need a form of cross-chain governance.

Polkadot Governance and Upgrades

As a common-good bridge, we will rely on the governance of Polkadot itself. Specifically this means the Gov2 decentralised governance system that is soon to be deployed to Kusama and Polkadot.

This will allow for governance participation by both ordinary users and domain experts in the Polkadot fellowship.

Ethereum Governance and Upgrades

Our bridge has a significant number of contracts on the Ethereum, and these contracts need to be able to evolve along with the parachain side.

As a prime example, Polkadot and BEEFY consensus algorithms will change, and so we need to make sure our smart contracts support these changes over time.

In general, the design for our upgradable smart contracts is quite simple, relying on versioning of immutable contracts. This stands in contrast with the proxy upgrade pattern, which while being very popular does have a lot of drawbacks.

Smart contract upgrades and configuration changes will be triggered by governance on Polkadot, through the use of cross-chain messaging secured by the bridge itself. In effect this means that there will be no governance authority on the Ethereum side. No multisig governance controller or anything like that.

Obviously this has implications if governance messages cannot be delivered to Ethereum for any reason. There are various fallback options under consideration. Some of these options would need to be trusted in order to work properly and so we'd require a further proposal and community vote to determine if that is something the community desires.

Bridge Hub

Our bridge will launch on the Bridge Hub, a common-good parachain that is being developed by Parity for Kusama and Polkadot.

This choice helps reduce the operational load on our team, so we don't have to run our own parachain collators and associated monitoring infrastructure.

Launch Plan

We will deploy to Kusama, followed by Polkadot. Each deployment will be separate and isolated from each other. Both deployments will bridge to Ethereum mainnet. There are currently no plans to bridge to other Ethereum networks.

Upon launch, the bridge will support bidirectional transfers of the following assets: Ether, ERC20 tokens, and DOT or KSM. A client library will also be provided to allow third-party teams to build their frontends around the bridge.

The launch will be followed by future upgrades that implement more of the features requested by the community. These features are described in the roadmap section.

Rococo

We already have the bridge deployed to the Rococo testnet, bridging to Goerli, the main test network for Ethereum (since Ropsten was deprecated after the merge). Our main use for Rococo is for testing and XCM integrations.

While development continues, we expect the bridge to break or be reset fairly often.

Kusama

Even though Polkadot is the end goal, we see Kusama as a very important stepping stone for our bridge, where we can exercise it with real users and assets, but within certain limits.

By the time we deploy on Kusama, our full security model and security audits will be in place. However, there will have been minimal testing with real world assets and scenarios. So we envision the following operating model:

- There will initially be strict limits on the TVL locked into the bridge. As confidence in the bridge grows, this limit will be progressively increased.
- There will be some level of insurance against an exploit, funded via an extra liquidity proposal.
- White-hat hackers will be incentivised to exploit the bridge.

Polkadot

We want our bridge to bake on Kusama for at least a few months before deploying on Polkadot.

As with Kusama, the maximum-allowed TVL will start off low, and then increase over the year, following a similar launch process to Kusama.

An alternative to TVL limits is rate-limiting of withdrawals. There is more discussion about this in our roadmap section.

After Polkadot

Once our Polkadot bridge is live and stable, we may consider deprecating the Kusama bridge and migrating over to the Polkadot bridge, such that Kusama↔Ethereum bridging gets replaced by Kusama↔Polkadot↔Ethereum bridging with messages all going via Polkadot, but this is a decision that can be made longer term in the future.

Roadmap

Overview

This section covers the long term roadmap of the project, broken down into milestones, each of which is roughly 6 months in duration. While we have a big list of features we'd like to develop, we also invite the community to share their own ideas. That's why we've only listed 12-18 months worth of improvements and features.

The order and content of our milestones is by no means final, and we expect shifts to occur. However, launching on Kusama and Polkadot is a priority, and all proposed feature development needs to fit into that schedule.

These are the estimated dates for launching the bridge. Note there is at least a 3-month gap between launching on Kusama and Polkadot.

1. Already Live - Full activation of bidirectional bridge on Rococo
2. Apr 1, 2023 Full activation of bidirectional bridge on Kusama
3. Jun 1, 2022 Full activation of bidirectional bridge on Polkadot

These dates are intended to be *flexible*. There are many various internal and external factors that could influence delivery:

- Our delivery progress on technical milestones
- Parity's deployment of BEEFY onto Kusama and Polkadot
- Scheduling of security audits

Milestone M1

This milestone lists projects for the remainder of 2022 and early 2023.

Activation of Ethereum PoS light-client on Rococo

Our Ethereum PoS light client is mostly complete, but there are some final fixes to make and some operational work to deploy it to Rococo. For example, we need to run our own beacon and execution nodes for Ethereum.

Clients

In the past we developed a demo webapp for using the bridge. However we deprecated it after focusing on launching as a common-goods project.

We imagine that parachain teams and dapps will want to have more control over the interface, with the bridge being an intermediary step rather than a primary point of users, so instead, we want to provide the building blocks in the form of a library.

The library will be written in Typescript and will provide the necessary client APIs for initiating and monitoring bridge transactions.

Governance on Ethereum

We need to be able to govern changes to the Ethereum side of the bridge. This is important for controlling upgrades to Ethereum contracts. For example, BEEFY will evolve over time, and so we duly need to upgrade our BEEFY light client in tandem.

Please refer to our proposed Governance model for more details.

Upgradable Smart Contracts on Ethereum

This needs to be fully implemented before we can go live. Our upgrade design is quite simple. Smart contracts will remain immutable. Contracts can be updated to refer to new versions of contracts they depend on.

Complicated and error-prone approaches such as the proxy pattern will not be used.

Circuit Breaker

After surveying many bridge and defi hacks it has become increasingly apparent that some kind of dynamic circuit breaker on asset withdrawals will be an important security addition to the bridge. It would provide defence in depth, in case our proof-based security model is subverted for any reason.

The circuit breaker could take the form of a simple rate-limiter on withdrawals, or a more sophisticated approach. Of course, these may impose limits on throughput of the bridge and be a potential vector for censorship, so it needs to be carefully designed and implemented.

Since our bridge has a layered design, the circuit breaker will be implemented mostly in our high level application layer.

Limits on total amount locked (TVL)

While the bridge is in its early stages of operation and evaluation, we want to limit the TVL. Implementing functionality to limit and configure the asset applications will be needed. As confidence in the bridge grows, the TVL limits can be increased via governance.

Depending on the progress of our circuit breaker, we may choose not to implement TVL limits, as they both tackle the same problem.

Adversarial testing

Our tests, while extensive, have mostly focused on happy paths and the most obvious adversarial attacks. We think it would be valuable to add tests with a more adversarial mindset, including tests with malicious inputs and fuzz testing. Some of these will be unit and component tests, others will be integration tests on the Rococo testnet.

Milestone M2

Production Infrastructure

For our testnet, we currently deploy our parachain nodes, ethereum nodes, and off-chain services using our own infrastructure platform, built using AWS EC2, Hashicorp Nomad, Consul and various other cloud-native tooling. This stack is working well for us, but as Parity is building out shared infrastructure and tooling for common-good chains, we'll likely want to migrate over to their solution.

If we choose to remain with our own infrastructure platform, we'll need to improve it to support production workloads, and add multi-region redundancy.

Security Audits

The Snowbridge codebase has previously undergone 2 security audits in 2021 for fulfilment of its initial W3F grant, but our codebase has evolved considerably since then, and so those audits are now quite obsolete.

As such, we plan to secure new audits for our code right before deploying onto Kusama. After the initial deployment, we also plan to hold regular audits, especially for major ongoing protocol-level changes.

Given that our codebase is bifurcated into parachain and solidity streams, and involves a complex protocol we need auditing to cover both streams and the comprehensive protocol.

The team will support external auditors and any changes that come about through the audits.

Launch on Kusama

The Kusama launch will involve significant effort and operational work. The process and goals are described in an earlier section: Launch on Kusama

The launch will be phased in over a course of several weeks:

1. Deploy pallets onto Bridge Hub parachain
2. Activate Ethereum->Parachain side of the bridge with safeguards in place
3. Activate Parachain->Ethereum side of the bridge with safeguards in place
4. Remove safeguards (TVL limits) after an appropriate amount of time.

Gas-price estimator

In the Parachain→Ethereum direction we need to cover the cost for our incentivized channel, where users all share the same channel. This cost is paid by the incentivized relayers, but is refunded through fees currently charged to users to cover the costs. The fees are charged based on a fixed gas price that is intended to be managed, but we need to overcharge to ensure we can keep up with unexpected fluctuations.

We'd like to build a more dynamic gas price estimator. Relying on any kind of gas-price oracle would be problematic given their centralised risks. Fortunately, since the recent Ethereum EIP-1559 upgrade, the BaseFeePerGas field is available in the block header and so can be tracked by our Ethereum light client. This means we can more accurately and dynamically upgrade the fee without overcharging as much. Implementing support for this will increase the cost efficiency of the bridge for users.

Permissionless apps

Currently our bridge has a limited set of core on-chain apps for bridging Ether, ERC20 tokens and DOT. These apps live on our parachain and are within its trust domain.

However, our base-layer protocol allows for bridging arbitrary messages. We can allow additional apps to be installed on our parachain, but to truly unlock the power of this feature, we want to allow 3rd-party parachains to message Ethereum smart contracts directly (and vice-versa), without needing to install anything on our parachain.

This feature does have some complexity, specifically, improving our on-chain adaptor between XCM and our Ethereum bridge to handle edge cases around gas limits and XCM failure/error scenarios from a security point of view.

Milestone M3

Launch on Polkadot

Some time after the Kusama launch (roughly 3-6 months), we'll want to activate the bridge on the Polkadot network as common-good parachain

The launch will be phased over a course of several weeks:

1. Acquire a common-good slot and launch a shell parachain (or launch on BridgeHub)
2. Activate Ethereum->Parachain side of the bridge with some safeguards in place
3. Activate Parachain->Ethereum side of the bridge with some safeguards in place
4. Remove safeguards

The bridge will only support cross-chain Ether, ERC20 token and DOT transfers at launch.

Channels V2

Transactions on Ethereum are considerably more expensive than transactions on our parachain. Our bridge reduces costs by batching messages together and then relaying them to Ethereum in a single transaction.

However there is still a limit on the size of these message batches, limiting the amount of messages that can be sent to Ethereum every commitment cycle. Over this limit, our parachain will drop any requests to include new messages. This dropping behaviour does not compose very well with XCM.

We therefore need to devise a method of allocating "bandwidth" to all users and third-party parachains that want to send messages to Ethereum, giving some degree of guarantees on message inclusion. This ties into the pricing model considerations mentioned above, implementing support for these pricing models.

Bridge arbitrary Polkadot assets to Ethereum

Currently our bridge has applications installed for bridging Ether and ERC20 tokens to our parachain, and in the opposite direction, bridging our parachain's native asset (ie, DOT) to Ethereum.

We want to extend this with an application that allows for any asset to be either deposited to our parachain via XCM, to be bridgeable to Ethereum (*For example Acala and Moonbeam tokens: ACA, GLMR).*

We've also seen demand from parachains to be able to keep this functionality on their own chain, rather than on our chain. This kind of usage would be unlocked by the Permissionless apps project.

NFT App

We have a prototype NFT transfer app, but it is not ready for production. It is also limited to supporting ERC721 NFTs. We'll want to improve it and deploy it so that we support transferring NFTs from Ethereum to any Polkadot parachain.

Post M3

BEEFY with BLS signatures

This would greatly improve the performance and gas usage of our BEEFY light client, as it would need only verify an aggregated BLS signature for each BEEFY commitment.

However this depends on support both in Polkadot and Ethereum. For example, EIP-2537 would need to be implemented in Ethereum.

Ongoing Maintenance and Support

Besides for specific milestone deliverables, working in the Polkadot ecosystem on a parachain demands a heavy workload for maintenance and operational support.

As a light-client powered bridge, we have a hard dependency on BEEFY, which is currently being developed by Parity for deployment onto Kusama and Polkadot. We anticipate the BEEFY protocol will evolve, and this would of course involve changes and work on our side as well.

Given that BEEFY is a critical requirement for our launch, we would potentially like to support its development directly. Helping to accelerate that development may be something we prioritise alongside milestone work.

Team

Aidan Musnitzky and Vincent Geddes have been leading the bridge team. We have prepared some biographies for more information on our current team.

Aidan Musnitzky

Aidan is the founder of Snowfork and supports various projects at Snowfork, including supporting and advising the Snowbridge team. Aidan has a background in a range of areas. A long time back, he worked on distributed systems at Amazon Web Services building out functionality for EC2. He's spent the last decade consulting on a wide range of projects, including other Blockchain and Cryptocurrency projects and helped bootstrap various crypto startups and advises an early-stage fund in the space.

Aidan started Snowfork a few years ago and has scaled it up to 20+ engineers across various teams and works on various different projects at Snowfork. Aidan played a primary role in the initial development of Snowbridge during 2020/2021 as an engineer and architect, and now supports the team, advising on technical and strategic leadership on an ongoing basis.

Vincent Geddes

Vincent is the technical lead for Snowbridge, and has been with the project for over 2 years. Previously, Vincent had stints at Oracle and Amazon Web Services. In both roles he specialised in virtualization hypervisors, cloud infrastructure, and monitoring.

As an early AWS EC2 team member, Vincent co-developed the first generation of CPU-burstable instance types, among other projects. At Oracle, Vincent helped integrate the Xen hypervisor in the compute service of Oracle Cloud (first generation), and later led maintenance and feature development work for that aspect of the product.

In the distant past he has also dabbled with compiler and runtime development for languages such as Smalltalk and Scheme.

Alistair Singh

Alistair has been a member of the Snowfork team for one year, working on Snowbridge's XCM implementation as well as onboarding Snowbridge onto Rococo. He completed a Bachelors in Computer Science degree and currently resides in the City of Cape Town. With 13 years of experience developing software in the traditional FinTech spaces, such as insurance and online casino industries.

He was previously employed at Amazon AWS as part of EC2’s Control Plane team, where he obtained experience building and maintaining critical systems at scale. In the computing field, he is passionate about databases, simulations, game development, programming language design, and most recently, blockchain technology.

Clara van Staden

Clara is a backend engineer with 9 years of experience in commercial software. After completing a B.IT degree (graduating top of her class) at the University of Pretoria in South Africa, she joined Globetom to work on many different projects for mainly telecommunications clients. She represented Globetom at the TM Forum, a global telecommunications standard’s body and won several awards for her contributions.

Thereafter she joined Over, a distributed startup building a visual communication app. Clara was responsible for building APIs to enable serving content (fonts, project metadata, images) for the web and mobile apps, using Go. Over was acquired by GoDaddy in 2020.

In 2022, Clara joined Snowfork to work on Snowbridge. She is responsible for the work required for the Ethereum upgrade. Over the last year, she has implemented a beacon light client to track the Beacon chain consensus.

David Dunn

I'm an obligate tinkerer: fiddling with workflows, tools & languages is a favourite pastime of mine. I've worked at fintech companies for a few years and joined Snowbridge to change things up by working with blockchains & the Rust programming language. So far so good! Having fully remote work on an open source project is also fantastic.

My current task is to allow multiple accounts to send messages over our basic (non-incentivized) channel. This enables anyone to run their own relayer to forward their messages, which is a key part of the bridge being trustless. Nobody should have to trust that someone will run a relayer (whether for an incentive or not) to get their messages across the bridge.

Retrospective

The project has been self-funded since September 2021, when we completed our last grant deliverable for the W3F. This deliverable was to ship a working end to end trustless Ethereum bridge based on a Proof of Work light client and an old, now deprecated version of BEEFY. Since then we continued to evolve the project, and have completed and embarked on many projects listed further below.

Overall, the core technical focus over the last year has been on the following:

1. Supporting Ethereum PoS Consensus, and deprecating our Ethereum PoW light client.
2. Tracking upstream BEEFY development as it evolves to its final production ready form.
3. Tracking upstream Polkadot, Substrate and Cumulus development as they have evolved.
4. Reworking and refining our core bridging protocols and pallets
5. Maintaining a testnet on Rococo.
6. More powerful off-chain relayer logic supporting wider edge cases and upstream changes in Polkadot and Ethereum itself (e.g. EIP 1559)

Another priority has been building up our team, and developing enough in-house expertise to execute on new features with greater velocity. Given the required skills and experience for operating in our environment, onboarding and growth has involved many months of ramp up time for new developers who have joined us.

Completed or ongoing projects

Light-client for Ethereum Proof-of-Stake Consensus

Now that Ethereum and its testnets have been merged, we can no longer use our PoW light client (which we developed back in 2020). As such we have developed a new light client, tracking the Beacon chain's PoS consensus.

This development has involved close collaboration with the Ethereum consensus community, as we are one of the few teams developing light clients for the Beacon chain.

The PoS light client is working and operational on our Rococo testnet. That said, we expect a few more months of refinement, testing, and auditing before we can consider it ready for production workloads.

See Technical Design for a high-level overview. The light client pallet lives in our codebase here. The off-chain relayer component can be browsed here.

BEEFY light client development

The BEEFY protocol has continued to evolve and improve since we completed our W3F grant deliverables. We have followed closely, keeping our light client up to date, and collaborating with BEEFY developers to troubleshoot issues and test new changes in the protocol.

Our BEEFY light client and related off-chain services have been totally rewritten since the initial versions delivered to W3F. The code is more readable, secure, and robust.

Pull requests

Testnet Operations

For a short while, we maintained our own private relaychain and parachain testnet, which bridged to the Ethereum Ropsten network.

Upon joining the Substrate Builders programme in early 2022, we migrated to the official Rococo testnet, where we help exercise BEEFY. Before Ropsten was merged to support PoS, we also supported full bidirectional bridging.

Our bridge on Rococo hasn't always been operational, given upstream issues such as the merge activity on Ethereum testnets in the past few months (June-August), or some occasional hiccups or compatibility issues with BEEFY as it evolves.

We are getting ready to deploy a new version of our bridge, with the Ethereum PoS light client, to bridge Rococo to the Ethereum Goerli testnet. This should occur sometime in the next few weeks.

Our system also requires a lot of off-chain services such as message relayers.

As part of the Substrate Builders programme, our team has gained a lot of experience operating a parachain on Rococo. Like upgrades, storage migrations, dealing with block production issues, and so on.

Everything is hosted on AWS EC2. HashiCorp Nomad and Consul are used for service orchestration and discovery. Traefik is used for load balancing, SSL termination, etc. There is a lot of complexity here and so in future we'd really like to migrate to whatever infrastructure is being used to host other common-good parachains and adjunct services.

Assets V2

As we gained experience with Substrate and XCM we realised our initial approach to representing Ethereum-native assets on our parachain was deficient, limiting, or would cause more problems down the line.

- Replaced our custom homebrew assets pallet with the upstream pallet-assets from Substrate
- Ethereum-native assets represented using 128-bits instead of 256-bits for better compatibility with XCM.

Pull Requests: #562, #9742

XCM Auto-forwarding

This is a feature which allows Ethereum users to transfer assets on Ethereum directly to third-party parachains, initiated by a single Ethereum transaction.

Without this feature, users would first have to bridge Ethereum assets to our parachain, and then issue a separate transaction with XCM instructions to transfer the wrapped assets to the final destination parachain.

Pull Requests: #533, #654, #556, #563

Permissionless Basic Channel

Our basic channel (See Technical Design) is currently permissioned in the sense that only privileged users may use it to send messages over the bridge. We are nearing the completion of work which makes the channel permissionless, so that any user can use it to send messages over the bridge.

This is a fairly complicated feature, as it involves adding another layer of merkle proofs in our bridging protocol for the Parachain→Ethereum direction.

Pull Requests: #637, #679

Contributions to Go-Substrate-RPC-Client

GSRPC is the main Golang client for Substrate, built and maintained by Centrifuge. As one of the main consumers of this library since 2020, we have contributed significant resources towards keeping it up to date as Substrate evolves.

Since Sep 2021, we have contributed the following: #249, #259

Contributions to Substrate

Besides helping to test the BEEFY protocol, we also added a new feature to the FRAME assets pallet, to help support use cases beyond Statemint: #9742

Other work

Besides new features, we have also invested a lot of resources in maintenance and iterative improvements, including:

1. Improvements to core bridging protocol: #631
2. Updating parachain with new versions of Polkadot/Substrate/Cumulus: #534, #591, etc.
3. Update pallets to FRAME 2.0 macros: #527

There are many other smaller improvement and maintenance tasks. See our repository for a full commit history of work since September 2021.

Funding Motivation

This funding proposal and plan revolves around a central goal: Maintaining and incentivizing a high-performance engineering team that can produce great value for the Polkadot ecosystem with a longer term runway period of 2+ years.

The majority of parachain teams today that do this are funded under a traditional VC-backed model with their own token and incentives that may make it challenging to align with a common-good parachain approach.

Having said that, the VC/token model can be very effective at creating teams with a stable, secure long term runway that gives the team full security and flexibility in their spending, as well as great incentives for upside in the success of the project, leading to projects succeeding and being maintained longer term.

Running a successful parachain, especially for a project related to bridging and interoperability is a complex task that requires significant resources and incentives.

The majority of interoperability projects in the blockchain ecosystem have required runways in the tens of millions of dollars, in addition to high potential token upside with valuations on the order of hundreds of millions of dollars to incentivize their team longer term. Snowbridge has decided against seeking any kind of external funding and has been bootstrapped beyond our initial grant for over a year now so that we could maintain optionality in our direction and maintain the potential for alignment as a common good chain.

The purpose of this funding proposal is to request for the Polkadot treasury to provide the kind of funding that can support major public good projects and parachain teams through similar dynamics to VC-funded projects - We're hopeful that this can succeed, and that our project can also play a part in growing a stronger culture around funding high-impact public goods that need a long-term runway and strong incentives.

We also believe that this particular project is a critical piece of infrastructure closely tied to Polkadot's vision, and is essential to uphold Polkadot's value proposition, so would be a highly valuable and impactful investment for Polkadot to make.

Funding Structure

The funding and resources required and requested for this project can be broken down as follows:

A. **Runway funding:** General purpose funding for the team, covering costs for 2 years worth of runway across all areas, including salaries, engineering work, operational work, management, support for general business expenses *(eg: Accounting, Admin, HR, Payroll, Software Tooling and Services used, Hardware costs, etc)*, Infrastructure, Travel, Conferences and Team events. This funding is requested to cover the runway from September 2021 to September 2023. It should be paid out immediately either in stable assets *(ie, USDC or USD)* or as DOT which is safely liquidated as soon as possible, and then managed and controlled by the team directly.

B. **Incentives funding:** Funding for the team to provide incentives for successful execution, running and success of the bridge project and to give the team upside in the success of the project and its impact on the Polkadot ecosystem. This funding will be managed and held by a collective as described below in a later section.

C. **Funding for audits and bug bounties:** Funding for 2 - 3 audits from highly reputable firms for both an initial audit and ongoing audit and security support as needed. This funding will be paid out and managed and controlled by the team directly, liquidated into USD specifically to be used for auditing costs, and can be refunded to the treasury if not fully used.

D. **Funding for insurance:** We'll likely want some insurance coverage for the bridge in the event of loss due to a bug. Insurance may not scale long term, as eventually we hope the bridge's security will stand sufficient on its own, but initially it will be valuable to get from the treasury to add a safety net for users. For the first year, or first few months of the bridge where it operates with something like a $10 million cap on assets, insurance from the treasury could be feasible.

E. **Operator Incentives:** This includes incentives for operators running the bridge. These will be stored on-chain, on both our parachain and in our Ethereum smart contracts with the following expectation:
   a. BEEFY Relay**:** We expect ~$300k to $1million per year needed to cover beefy relaying gas costs and a basic incentive for at least one honest relayer.
   b. Ethereum Header Relay: Should be a magnitude cheaper than the BEEFY relayer.
   c. Incentivized Channel Relayer: Depending on the pricing model, message relayers could be partially or wholly subsidised by the treasury.
   d. Basic Channel Relayers: Our governance model will require the use of our basic channel for cross-chain governance. This will see only intermittent use so costs should be very low.

F. **Liquidity bootstrap:** We will want some funding to bootstrap liquidity for the bridge and for bridged assets to get Ethereum liquidity into Polkadot's DeFi ecosystem, as well as some DOT liquidity into Ethereum's ecosystem. We may also consider liquidity incentives to encourage usage in future too.

The funding request in this proposal covers items A, B, and C.

Runway Funding

Snowfork is requesting **$3,060,000 runway funding** in total to cover the period from Sept 2021 to Sept 2023.

- **$960,000** will apply to the period from Sept 2021 to the start of Sept 2022.
- **$2,100,000** will apply to the period from Sept 2022 to start of Sept 2023

Retroactive funding

The retroactive numbers are based on actual investment already made into the project over the last year, broken down as follows:

Item	**Amount (USD)**
Core Snowbridge Team (This was initially 3 core contributors with 2-3 additional part time contractors in 2021, but has ramped up to currently 5 core contributors across dev and management in 2022)	741300
Supporting Staff This incorporates costs from supporting Snowfork staff and services, including HR, Accounting Services, Tax Services, Recruiting Services, Referrals and other Admin Support Services	107500
Infrastructure	65863

This incorporates all system infrastructure costs, including costs like AWS, Tenderly, Alchemy, ElasticSearch, Infura and a few other hosting services Every dev has their own EC2 instance for development, as our development environment does not work well on laptops.	
Software Services This incorporates all software services used by the team, including costs like Google Workspaces, Slack, Discord, Github, Linear, Lever, Quickbooks, Notion and a few other software services	16760
Travel, Team Events, Conferences This incorporates all costs related to travel, team events, coworking and conferences	13000
Hardware This incorporates all hardware costs, primarily laptops for staff	8500
Admin, Legal, Insurance This incorporates fees and payments related to administrative costs, payroll, legal and insurance related fees	7150
Total	**960073**

The above numbers are more heavily weighted towards recent months as the team has scaled up, and currently runs at ~$85000 per month. Additionally, the project leads have been working for significantly below market rates.

We give a detailed accounting for the work done over the last year in the Retrospective section.

Future Runway

For the future year of runway, we're requesting funding to support full market-competitive pay for the entire team as well as a buffer to support potentially scaling up with an additional 2-3 full time contributors as needed, and additional travel/conference expenses and growing Infrastructure expenses as we scale up our services.

Given this target, our future target runway expectation is at $175,000 per month, with the additional increase allocated almost entirely to engineering and infrastructure related expenses that we plan to ramp up over the coming months.

This results in the $2,100,000 amount for Sept 2022 to start of Sept 2023

The team also expects to maintain full autonomous control over its runway and how to spend it, much like in a traditional fundraise *(provided all costs are driven towards a successful polkadot ethereum bridge and effective team to build and maintain it)* and so the total full amount of **$3,060,000** is requested up front as part of this proposal, and will need to be either received in USDC or quickly liquidated into USD to ensure a guaranteed secure runway for the project independent of market conditions.

We expect to do a similar runway-related proposal in a year for a follow up year of runway, depending on the situation at the time, and so if the team does not utilise the additional buffer of funds requested, nor scale up as quickly as expected, then they may spillover to allow us to delay that extended runway proposal beyond September 2023.

Incentives Funding

With a runway in place for team expenses and job security, there is still the need to ensure that the team is incentivized to feel strong long term ownership of the project and has an incentive in upside related to the success of the project and impact of the bridge, long term successful operation, and incentive in the Polkadot ecosystem as a whole.

We're requesting funding to be allocated for this, covering incentive bonuses for:

- Succeeding at shipping and launching the milestones outlined in our Roadmap
- Overseeing successful operation and maintenance of the bridge *(including being part of governance)*
- Successful long-term stewardship for operation and security of the bridge
- Research & development of future next-gen improvements or evolutions of the bridge

This will consist of several incentive triggers, each with a DOT reward that triggers based on certain conditions. Rewards based on engineering milestones are expected to vest over 2 years, and rewards based on measured success and impact on the Polkadot ecosystem are expected to vest immediately.

We feel a total of $10,000,000 in exposure to DOT, measured based on the price of DOT at time of proposal, is fair if the project is highly successful and makes a measurable impact on the Polkadot ecosystem, with the majority of exposure being weighted to success and impact triggers rather than engineering milestone triggers.

This is structured as:

- $1,875,000 in long term, vested DOT exposure as a reward for engineering milestone completion and successfully shipping and launching the bridge on both Kusama and Polkadot
- $8,125,000 in long term DOT payouts over 24 months based on successful operation and usage of the bridge and its impact on parachains in the ecosystem

Payout Trigger	**Amount measured in USD @ $7.5 / DOT**	**DOT Payout @ $7.5 / DOT**
P1 - Milestone M1 and related tasks fully completed and reviewed	$375,000	50,000 DOT vesting over 2 years
P2 - Milestone M2 and related tasks fully completed and reviewed, including live Kusama	$562,500	75,000 DOT vesting over 2 years

launch		
P3 - Milestone M3 and related tasks fully completed and reviewed, including live Polkadot launch	$937,500	125,000 DOT vesting over 2 years
P4 - Bridge is live with successful operation on Kusama for 3 months and at least 3 parachains integrated and bridging to Ethereum	$625,000	83,333 DOT paid out on trigger
P5 - Bridge is live with successful operation on Polkadot for 3 months and at least 3 parachains integrated and bridging to Ethereum	$937,500	125,000 DOT paid out on trigger
P6 - Continued successful operation on Polkadot from a period of 3 months after launch to 24 months after launch, with at least 6 parachains integrated and bridging in assets	$312,500 per month for 21 months	41,666 DOT paid out monthly on trigger for 21 months
Total	**$10,000,000**	**1,333,333 DOT**

For all triggers related to successful operation and parachain usage, if for any reason there is a break, delay or security issue or loss of funds from the bridge, that trigger will be paused.
The trigger will then only resume if the bridge resumes in operation successfully and usage and sentiment of the bridge from parachains continues.

Evaluation of milestones and payments

Our milestone deliverables will need to be reviewed, and then funds disbursed accordingly using a *vested.Transfer* call. To structure this process, we anticipate a governance collective being formed from various stakeholders (Parity, W3F, parachain teams, community members, etc). Snowfork will obviously have no representation on this body.

The collective will:

1. Review the scope of the milestones and any requests to change scope
2. Review the milestone deliverables
3. Vote on the whether to mark the milestone as completed satisfactorily
4. Execute the *vested.Transfer* call, transferring funds from the treasury to accounts under Snowfork's control.

For this to work, the review collective should have control over the incentive funds. Given that the collectives functionality is not yet live, we propose to defer payment of the incentive funds until then. Once the collectives functionality is live, the incentive funds should be transferred to an account controlled by the Review collective. If the collectives functionality is not live, or not suitable for use by the time of the first milestone, then voting should be coordinated offchain and the current Polkadot council should execute the *vested.Transfer* on behalf of the offchain collective.

The collective will be subordinate to Polkadot governance. This is a remedy in the case that the collective becomes inactive or if any counterparty wishes to dispute an issue.

Additionally, given that the Polkadot ecosystem involves evolving technologies, we will still need to be able to adapt to changes and maintain an agile development workflow as part of our roadmap. In practice, this means being able to make change requests to planned roadmap items and milestone evaluation criteria if the space changes in unexpected ways. This same review collective will be responsible for evaluating and accepting requests to change the roadmap and milestone deliverables.

Funding for audits

We'd like to get 2 different independent audits from 2 different auditing firms that includes a full initial audit of all aspects of the protocol and all on-chain code on Polkadot and Ethereum. This should also lead to an ongoing relationship with an updated audit every few months as any changes are shipped.

We also think it would be valuable to get an audit of our off-chain code, even though it is less security critical from at least one of the auditors

We have not engaged with or gotten a quote from auditors yet, but we expect this to cost in the realm of $300,000 to $500,000 per auditor.

Given this, we're requesting **$1,000,000** to be allocated for current and future audits, although this can be adjusted in more detail when we have concrete quotes.

Source of funds

Given that the eventual launch will be primarily focused on Polkadot, we expect the majority of this proposal to be covered by the the Polkadot treasury, however it does make sense for some portion to be covered by the Kusama treasury given that the technology will be deployed to and benefit Kusama as well.

We propose for the split to be as follows:

- The runway funding should be split with 20% covered by Kusama and 80% covered by Polkadot ($612,000 vs $2,448,000)
- The incentive funding should be split with the Kusama-specific incentive payouts P2 and P4 being covered by Kusama and the remainder being covered by Polkadot ($1,187,500 vs $8,812,500)

Future Liquidity Proposals

Funds for D, E, F and G will likely be in a follow up liquidity proposal closer to launch time. As an expectation, as mentioned above, we expect this future proposal request to be in the range of ~$1.5M per year for relayer incentives and baseline gas sponsorship costs to keep the bridge active and secure, and an additional flexible amount for insurance, liquidity bootstrapping and growth incentives.

Summary of Funding Requested

We request **$14,060,000**, broken down as follows**:**

- Runway funding:
  - **$960,000** for the period from Sept 2021 to the start of Sept 2022, paid out on the acceptance of the proposal
  - **$2,100,000** will apply to the period from Sept 2022 to start of Sept 2023, paid out on the acceptance of the proposal
- Audit funding
  - **$1,000,000** paid out on the acceptance of the proposal *(or later once auditing details are finalised)*
- Incentive funding
  - **$1,875,000** allocated across successful completion of milestone work and launch of the bridge on Kusama and Polkadot, vesting over 2 years from completion and launch
  - **$8,125,000** allocated for success and impact of the bridge for a 2-year period post launch of the bridge

Long term future

This proposal targets an average long term cost of **$6,500,000** for the team for all expenses and incentives for running the bridge per year.

If we include our additional expected future costs in the future liquidity proposal of **$1,500,000** for sponsoring gas-related costs and incentives for the bridge, this brings us to a total of **$8,000,000** per year cost for the Polkadot ecosystem to keep the Ethereum bridge running and successful long term.

We imagine that, assuming the Polkadot ecosystem continues to grow and that the bridge continues to be used, it is very feasible to imagine that the bridge could capture this amount of yearly value from the various sources of revenue described in this proposal, and that after 2 years it is feasible that the bridge becomes self sustaining and no longer requires sponsorship from the Polkadot treasury.

Additionally, we expect that the bridge may be able to capture even further value than this long term and so could become an additional mechanism for burning or locking up DOT long term reducing the DOT supply further.