

This proposal concerns the financing of the ongoing development and maintenance of **Polkadot-API** for 2025.

Summary.....	1
Overview.....	1
History and team.....	2
History.....	2
Team.....	3
Josep M Sobreperé.....	3
Víctor Oliva.....	3
Carlo Sala.....	4
Project objectives, status and direction.....	4
Objectives.....	4
1. "Light-Client First" with an Excellent Development Experience.....	4
2. Modular and Composable Libraries with Clear Boundaries.....	4
3. Creation of Performant DApps Without Compromising User Experience.....	5
4. Education through Documentation, Tutorials, and Examples.....	5
5. Equipping Power-Users and Builders with adequate Tools.....	5
Status.....	5
Direction.....	5
Polkadot-API.....	5
Polkadot-API SDKs.....	6
PAPI Console.....	7
Note on transparency.....	7
Milestones and budget.....	8

Summary

Proponent	16JGzEsi8gcySKjpmxHVrkLTHdFHodRepEz8n244gNZpr9J
Beneficiary Address	167tDhLwaQ6kmgau1zwWx48Ux7CS1rxC9jW5kn8PeRZWAVUy
Date	27/02/2025
Requested Amount	630 736 €

Overview

Polkadot-API is a project conceived to build upon and surpass the foundational achievements of the Polkadot-Js project. At its essence, Polkadot-API is a meticulously

crafted suite of libraries, each designed to be composable, modular, and fundamentally aligned with a "light-client first" philosophy.

Our overarching mission is to empower dApp developers by equipping them with a comprehensive array of tools, pivotal for creating truly decentralized applications. This initiative places a strong focus on elevating the developer experience.

During 2024 we delivered the first stable version of polkadot-api, as well as contributed to other utilities to empower developers. We have also started the first batch of Polkadot-API SDKs, higher-level libraries covering common business logic across different workflows.

This proposal outlines the ongoing maintenance, new features, and strategic initiatives we've identified based on valuable community feedback.

History and team

History

The project unofficially began on October 29, 2021, marked by the [first](#) repository commit by [Josep](#), a recent recruit to Parity's Substrate-Connect team. His initial foray involved creating the [scale-ts library](#), which was conceived to offer a modular, composable, and strongly-typed solution for encoding/decoding SCALE-encoded data. This effort, essential for a seamless development experience, was part of an experimental side project named unstoppable-js, which Josep used for learning and exploration, yet with aspirations for greater impact.

Simultaneously, Josep's contributions to @substrate/connect deepened his understanding of the significant disparity between smoldot and the JSON-RPC API used by PolkadotJS. Recognizing the need for a library that could efficiently interact with a new JSON-RPC API specification, which Pierre, the creator of smoldot, was developing. This pursuit eventually culminated in the birth of the Polkadot-API project.

In early 2024, Josep recently opted to leave Parity to expedite the Polkadot-API's development with [Victor Oliva](#), a long-time collaborator in open-source software. After a couple of months, [Carlo Sala](#) joined the team and they have been working full-time since then in the development of Polkadot-API, focusing on the milestones outlined in the previous opengov proposals for 2024.

It's important to note that the Polkadot-API team will maintain its strong collaborative ties with smoldot and other teams in the community ensuring continued synergy and advancement in the ecosystem.

Team

Josep M Sobreperere

Joining Parity Technologies in September 2021, Josep has made significant contributions to numerous key projects. His work encompasses contributions to [@substrate/connect](#), [Polkadot-API](#), the new [JSON-RPC API](#), the [Validator Selection Tool](#), [smoldot](#) and [PolkadotJS](#).

As the architect of the Polkadot-API project, Josep played a pivotal role in leading the evolution of [@substrate/connect](#) into the first [Polkadot Provider](#) compliant extension, a transformation he spearheaded with the crucial help of Matias Volpe.

In addition to his tenure at Parity, Josep, in collaboration with Victor Oliva, has co-authored the [React-RxJS](#) project and contributed to many influential open-source libraries, including [RxJS](#), [React-Router](#), [React-Redux](#) and [Reselect](#).

Recently, he successfully [graduated with distinction](#) from the 2024 Polkadot Blockchain Academy in Singapore.

- GitHub profile: <https://github.com/josepot>
- E-mail address: sobreperere@protonmail.com
- Matrix account: [@josepot:matrix.org](https://matrix.to/#/@josepot:matrix.org)
- DOT address: [15roJ4ZrgrZam5BQWJqiGHpgp7ShFQBRNLq6qUfiNqXDZjMK](#)

Víctor Oliva

Víctor is a software developer specialized in handling real-time data, working as a consultant since 2016 building trading platforms. During this time he co-authored React-RxJS with Josep, and used it as the primary state management library to successfully launch two trading platforms into production, one of them having a volume of 12 billion USD monthly, with around 3500 filled orders per day.

During the time Victor has been working on Polkadot-API he has made important contributions to the project, especially in code generation and generating unique checksums.

Additionally, Victor has made significant contributions to various open-source projects, including RxJS ([6049](#), [6275](#), [1381](#), [6289](#), [7114](#)), chai ([1242](#)), SockJS ([472](#)), webpack ([ts-plugin](#), [17728](#)) and fakehost ([19](#)).

Recently, he successfully [graduated with distinction](#) from the 2024 Polkadot Blockchain Academy in Singapore.

- GitHub profile: <https://github.com/voliva>
- E-mail address: v.oliva@proton.me
- Matrix account: [@voliva:matrix.org](https://matrix.to/#/@voliva:matrix.org)
- DOT address: [16JGzEsi8gcySKjpmxHVrkLTHdFHodRepEz8n244gNZpr9J](#)

Carlo Sala

Carlo is a software developer who joined the Polkadot space over two years ago inside Zondax. He contributed to the Ledger HW integration for [Polkadot](#), [Kusama](#), and many other chains in the ecosystem inside Zondax.

He contributed as well to shape the future of merkleized metadata in both [RFC #46](#) and [RFC #78](#) along with other colleagues in Zondax, Parity, and Alzymologist teams.

Outside of crypto space, Carlo is the current core maintainer of [Oh My Zsh](#), an enormous open source project along with two other maintainers. Some headlines of it:

- It has over 176k stars (it's the #21 repo by stars in the whole GitHub space!)
- It has over 2400 contributors. One of the tasks of Carlo is to manage a fair amount of issues and PRs in the repo.

- GitHub profile: <https://github.com/carlosala>

- E-mail address: carlosalag@protonmail.com

- Matrix account: [@carlosala:matrix.org](https://matrix.org/@carlosala:matrix.org)

- DOT address: [16JskuojL6mSp6HNcjiHYa9jqksWbLD8L9YGWU1ppiPWQ9sa](#)

Project objectives, status and direction

Objectives

The Polkadot-API project is ambitiously pursuing five high-level objectives:

1. "Light-Client First" with an Excellent Development Experience

'Light-Client First' refers to the strategic emphasis on optimizing our library specifically for light clients. This approach leverages the [new JSON-RPC API](#), renowned for its light-client friendly design. By being built on top of the new API, our library not only performs exceptionally well with light clients but also maintains compatibility and high performance with other providers that adhere to the new JSON-RPC API standards, including centralized RPC Nodes. This dual compatibility ensures that while our focus is on light clients — offering a streamlined, efficient, and resource-conscious solution — the library retains full functionality and effectiveness in broader contexts as well.

Our goal is to offer a high-level API that simplifies the complexities of the new JSON-RPC API. Feedback indicates notable success in this area. We are actively refining our API, addressing feedback, and adding in more features.

2. Modular and Composable Libraries with Clear Boundaries

Significant progress has been made in developing modular libraries. Our focus remains on continuous improvement, ensuring each library is refined, self-contained, and easily integrable.

3. Creation of Performant DApps Without Compromising User Experience

We've made strides in developing [DApps with minimal bundle sizes](#) and rapid loading times. The next step involves optimizing bundle size of the main chunk, to ensure small loading times and improve user experience. While there's still work to be done, our current trajectory in enhancing DApp performance is promising.

4. Education through Documentation, Tutorials, and Examples

We plan to extend the current documentation, by adding more recipes, examples and a detailed API reference to guide users effectively with the release of our stable release. This will be a cornerstone in providing the necessary resources for developers to utilize Polkadot-API effectively.

5. Equipping Power-Users and Builders with adequate Tools

Many projects built with Polkadot-API ([Polkadot Fellows](#), [Delegit](#), [Kheopswap](#), [Papi Console](#), [Polkadot Bounties](#)) have highlighted the potential of our libraries to develop powerful tools for the ecosystem. Moving forward, a key priority will be showcasing their practical applications within this environment.

We believe it is essential to develop these tools alongside the foundational libraries to establish the right abstractions. This approach will enable other teams to create their own tools and dApps effectively.

Status

The current status is explained at length in the [H2-2024 update of the project](#). The project is still in active development.

Direction

Polkadot-API

To ensure transparency and maintain trust within the Polkadot community, we are committed to continuing our established communication practices. This involves periodically posting detailed updates on the Polkadot forum, where we share insights into our progress, address challenges, and outline future plans. Additionally, we will keep our GitHub issues consistently updated, providing real-time visibility into our development process and current focus areas. These practices are integral to our project's direction, ensuring that we remain aligned with the community's expectations and needs while fostering an environment of open collaboration and continuous improvement.

Upon reaching the stable version, we are fully committed to ongoing maintenance and support of Polkadot-API. Our efforts will not only focus on ensuring its continued reliability and performance but also on actively incorporating new features and enhancements in response to evolving community needs and feedback. By staying attuned to the community's requirements, we aim to ensure that Polkadot-API remains a robust and adaptable tool that continues to meet the demands of its users in a dynamic ecosystem.

There might be some API improvements that will require a breaking change, and so a possible v2 of the library. Our philosophy is that if we need to, have small incremental versions with clear migration steps (and possibly automated), rather than accumulating tech debt and release major versions that would be impractical to upgrade.

Some of these changes involve supporting the upcoming Transaction Extensions from Polkadot into the TypedAPI, which will also require an improvement on the signer interface. Another change we are considering is breaking down the TypedAPI into submodules, which could help massively in decreasing bundle sizes for some dApps.

Polkadot-API SDKs

Polkadot has been designed, developed, and enhanced with a focus on nodes, operating costs, and the core development experience, but the end-user has often been overlooked.

Libraries that would facilitate dApp developers' use of Polkadot technology to deliver real value to end users are missing, resulting in a poor user experience. We aim to develop SDKs for the primary pallets across Polkadot/Kusama and their system chains, with a focus on making interactions with Polkadot as straightforward and seamless as using a traditional web2 application.

When creating `polkadot-api`, we strictly focused on building a chain-agnostic library with no business model logic. While low-level endpoints are useful for advanced implementers, we need higher-level APIs that understand the pallet's business logic. The main challenge we've identified in Polkadot is not the UI components but the highly low-level nature of the blockchain's APIs. End users should not need to worry about storage queries, runtime calls, or other low-level implementation details of Polkadot, yet they currently do.

While developing the SDKs for the main use cases of the relay and system chains, we also aim to support the creation of new SDKs tailored for parachains. These chains may have unique business logic for each pallet, or they may need entirely new SDKs designed specifically for their own pallets.

We see this initiative as a natural extension of the work we've done (and will continue to do), bringing significant benefits to the entire Polkadot developer community.

During the last proposal we have released the initial versions of Governance SDK, Accounts SDK, Ink SDK. For the previous proposal we initially underestimated the work needed to develop these SDKs, as it takes a lot of back-and-forth to find the correct abstractions, possibly needing to create side dApps (like [Polkadot Bounties](#)) alongside.

During 2025 we plan to continue developing them until they reach a more stable version.

PAPI Console

During 2024 we released [PAPI Console](#), a dev console built with Polkadot-API that serves both as a tech demo and as console with new visualizations and features. You can hear more about it in our presentation in [Reset Sub0](#).

As it's a console that has gained popularity, we think it's important to spend some time maintaining it and adding some new features:

- Add the ability to connect to multiple chains simultaneously, which would let track things like XCM events across chains
- Improve storage subscriptions. Currently the console doesn't show complex keys, and this is a small feature it's missing. Another idea for storage subscriptions is when a value changes to show a better diff, of what changed and when.
- The Console is also missing RPC Calls. Initially we focused on exposing everything from the chain that's specified in the metadata, which unfortunately doesn't include RPC Calls. These can sometimes be useful, so we will add a new section for RPC Calls, with a specific UI for standardized ones (JSON-RPC Spec) and a generic one for unknown ones, coming through `rpc.rpc_methods`.

Not fully related to the PAPI Console, but we think it might be really useful, is creating some dev tools for monitoring a JSON-RPC Spec connection: displaying on-going subscriptions, operations, adding the ability to tap or send specific messages, etc. Currently we have a "logsProvider" that just logs the messages going through, but it would be quite useful to have a more integrated tool.

Note on transparency

We strive for keeping full transparency of our work and the funds awarded by our previous proposals in the treasury. We keep posting regular updates in the [polkadot forum](#) with details.

The [previous proposal](#) covered the budget for a team of 4 during 6 months. Unfortunately, [we parted ways](#) with one of the members of the team, Hannah, during the second half of October, and the scope of that proposal was affected.

We think the most transparent and fair way to manage this is by discounting the funds for the 4th developer for the time it hasn't worked on it, by taking the proportional value based on the full cost of the proposal. We made the decision for the team to assume some of the additional costs we had from this, like taxes, onboarding costs and some operational expenses.

The proposal covered 183 days, of which Hannah worked 83 days, leaving out 100 days of work for one developer. The refunded proportion should be $100 / 183 = 54.64\%$ for one developer, meaning $54.64\% / 4 = 13.66\%$ of the value of the previous proposal. The value was 540.000€, so the **refunded value should be $13.66\% * 540.000€ = 73.764€$**

We have plans to possibly let the team grow, but rather than requesting an up-front budget for them, we will create new proposals when we find the correct fit, with specific milestones that would cover the specific goals for the new members of the team.

Milestones and budget

This proposal requests financing for the next 11 months, from February to December 2025. We chose these work items based on what we consider to be the highest priority. However, we are totally open to feedback if you think that priority should be placed somewhere else.

During February, 2 of the developers will be preparing the content for the dApps / off-chain track for the Polkadot Blockchain Academy. So this proposal covers the work of 1 full-time developer during February, which will guarantee continuous maintenance of the project, and the full team (Josep, Víctor and Carlo) for the months from March to December. This means this proposal will cover 150 hours of work for the first month, and 450 hours for each of the other months, a total of 4.650 hours.

We propose an hourly rate of **150€**, which we believe is both reasonable and justifiable given the expertise required for a project of this complexity. The funds will be handled through the [Polkadot Community Foundation](#). The PCF would be responsible for creating the appropriate contractual agreements with our legal entity based in Spain, and managing the funds according to the interest of the Polkadot Treasury and its community.

General maintenance of the Project: - Bug fixes - Improve test suite - Keep changelogs up to date - Improve and automate CI deployments - Keep dependencies up to date - Adapt to the changes of JSON-RPC spec - Report issues to upstream dependencies - Improve code readability and add comments to the code parts that require them. A lot of changes are either unpredictable at the moment, or are too small to be their own milestone.	1 650 Hours	247.500,00 €
Improve documentation: Examples and tutorials	120 Hours	18.000,00 €
Improve code splitting and bundle sizes	120 Hours	18.000,00 €
Support custom hashing functions	40 Hours	6.000,00 €
Get codec for any type defined in the metadata	80 Hours	12.000,00 €
Extrinsic v5	80 Hours	12.000,00 €
PAPI v2: Transaction Extensions	120 Hours	18.000,00 €
PAPI v2: Improve signer	80 Hours	12.000,00 €
PAPI v2: Docs and migration guide	40 Hours	6.000,00 €
PAPI v2: Improve TypedAPI modularity	240 Hours	36.000,00 €
PAPI Console: Maintenance	320 Hours	48.000,00 €
PAPI Console: Multi-chain support	120 Hours	18.000,00 €
PAPI Console: Improve storage subscriptions	120 Hours	18.000,00 €
PAPI Console: RPC calls	80 Hours	12.000,00 €
Polkadot-API Governance SDK Maintenance	120 Hours	18.000,00 €
Polkadot-API Ink SDK Maintenance	80 Hours	12.000,00 €
Polkadot-API Accounts, proxies and multisigs SDKs	160 Hours	24.000,00 €
Polkadot-API Staking SDK	240 Hours	36.000,00 €
JSON-RPC spec Dev Tools	240 Hours	36.000,00 €
Provide support to ecosystem teams and tool builders	600 Hours	90.000,00 €
Proposal total	4.650 Hours	697.500,00 €
Polkadot-API H2 2024 Proposal balance refund		-73.764,00 €
Net	4.650 Hours	623.736,00 €
Polkadot Community Foundation overhead provision		7.000,00 €
Total	4.650 Hours	630.736,00 €

Please note that these milestones are provided as a best-effort estimate, and the reality might differ. The actual development (i.e. new functionalities, code updates, etc) will be prioritized based on reported GH issues and/or community requests. If you were to be unsatisfied with our work, we are open to discussing how we focus our efforts.

The exact amount in USDCs will be calculated when the proposal is submitted on-chain using current USD <-> EUR exchange rate found on [xe.com](https://www.xe.com). The destination address is [167tDhLwaQ6kmgau1zwWx48Ux7CS1rxC9jW5kn8PeRZWAVUy](https://polkadot.subscan.io/address/167tDhLwaQ6kmgau1zwWx48Ux7CS1rxC9jW5kn8PeRZWAVUy), which is the address in Polkadot AssetHub for [Polkadot Community Foundation](https://polkadot.subscan.io/address/167tDhLwaQ6kmgau1zwWx48Ux7CS1rxC9jW5kn8PeRZWAVUy).

The PCF will create the appropriate contractual agreement, which will be signed by both parties. Polkadot-API team will create one invoice for the PCF for the full value of **623.736,00€**, and PCF will transfer the equivalent of 623.736,00€ in USDC, to [12R1XCdgkHysv8Y4ntiXguo4eUYHXjQTmfRjL8FbmezsG71j](https://polkadot.subscan.io/address/12R1XCdgkHysv8Y4ntiXguo4eUYHXjQTmfRjL8FbmezsG71j), which is a multisig controlled by the members of the team. You can use [multix](https://multix.io) to easily verify that the signatories of that address are Victor and Josep.

The PCF will use its part needed from the equivalent of 7.000,00€ in USDC to cover any expenses and operational fees, and will return to the treasury the remaining.