TL;DR

The goal of this thread is to keep you updated on the front-end interface project for Arbitrum that we at WakeUp Labs are developing.

Lately, we proposed creating an interface that allows users to push transactions through even when the Arbitrum sequencer is offline. After posting our proposal on Tally, we were chosen to develop it with support from the Arbitrum community for WakeUp Labs to take on this project.

[

WakeUp _ Arbitrum

1200×627 26.1 KB

](https://global.discourse-cdn.com/flex029/uploads/arbitrum1/original/2X/4/4c12dd94458fbff926056f620b3da503370b3ec8.png)

What's the project about?

This project aims to strengthen the Arbitrum ecosystem by developing an interface that allows users to submit transactions directly to Layer 1 (L1) even if the Arbitrum sequencer, responsible for handling transaction submissions, experiences downtime or issues. The main objective is to address the risk of dependency on a single point of failure presented by the sequencer.

The proposal focuses on decentralization and empowering users through an open-source approach, enabling other teams to deploy their own versions of the interface. This initiative not only enhances the network's resilience but also provides users with greater control and confidence in the Arbitrum ecosystem. By enabling direct transaction submission to L1, users can bypass potential issues with the sequencer or any attempts at censorship, thereby ensuring uninterrupted transaction capabilities. This approach also upholds Arbitrum's trustless security model, reinforcing the integrity of its Rollup technology even during operational challenges.

Arbitrum's Forum

Last February, we submitted a proposal to the Arbitrum Foundation Forum. It garnered considerable attention, with numerous exchanges and suggestions from community members that helped us improve our proposal with their feedback. This was done to assess the sentiment related to our proposal and to enhance it based on community feedback.

You can check the full proposal here

Tally: Front-end interface to force transaction inclusion during sequencer downtime[Proposals

](/c/proposals/7)

[Non-Constitutional AIP] Front-end interface to force transaction inclusion during sequencer downtime Abstract WakeUp Labs proposal outlines the development of a simple but effective front-end interface that enables users to force-include transactions directly on L1 when the Arbitrum Sequencer is down, aiming to increase trust among end-users navigating the Arbitrum ecosystem. Inspired by a few community discussions (Link 1 & Link 2) , and validated historical events like: Sequencer Downtime 1,…

We've also held an open call to discuss and share all the insights of our proposal to Arbitrum in a meeting

[

](https://www.youtube.com/watch?v=vL3nHpsJXXU)

Later on, we conducted a successful snapshot sentiment check that you can checkhere.

With over 99M votes in favor and less than 1% against.

The Tally formal proposal

To move forward, we posted our proposal on the Tally platform. After giving Arbitrum members time to vote on its value to the community, we were selected to develop the interface.

Tally

Tally | Arbitrum | Front-end interface to force transaction inclusion during...

Front-end interface to force transaction inclusion during sequencer downtime

Development Roadmap

Technical Research, POC, Documentation and Testing

We'll take a deep dive into the Delayed Inbox functionality and its integration, as well as create a proof of concept (PoC). The WakeUp Team's research will be made public, with a summary shared in an X (Twitter) thread and a detailed article posted in the OSS GitHub repository once it's completed. This document will also outline the necessary security measures for the development process.

Endpoint Development

Creation of endpoints to interact with the backend logic.

Logic Integration and Testing

Connecting the frontend to the endpoints and verifying the operational logic, and test heavily the MVP.

Design Phase

Additional time allocated for the design process, ensuring a user-friendly experience.

UI Development

We'll craft an aesthetic and intuitive user interface inspired by existing high-quality designs. To enhance user security, we'll add a banner at the top of the website to verify the URL and guard against phishing sites.

Enhancing Documentation Clarity and Usability

We will finalize and clearly document the open-source codebase. As mentioned above, any team should be able to run their own instance of the service, so it's crucial that everything is clearly explained. We'll also include a list of URLs for instances set up by others that we have verified as correctly implemented.

Service & Maintenance after Launch

We are committed to hosting the site on either Vercel or AWS, ensuring continuous operation 24/7 for a minimum of 24 months. Once we achieve a stable version that has been thoroughly tested, we will also deploy an instance on IPFS to offer a fully decentralized option. While this may not be the most performant choice initially, it will be available for at least these 2 years, with plans to extend indefinitely after evaluating its performance.

At WakeUp Labs we will initially refine the website based on user and Arbitrum community feedback, conducting ongoing testing to maintain reliability. Any issues such as downtime or crashes will be promptly addressed by our team for at least the next two years.

Estimated Timeframe

Approximately 4.5 months for full implementation, including design phases, followed by 3 months dedicated to post-launch iterations. We will then commit to hosting, monitoring, and addressing any possible errors for 24 months after launch.

Conclusions.

Developing a front-end interface to facilitate force-including transactions during Sequencer downtime is a crucial initiative within the Arbitrum ecosystem. This proposal aims to empower users by ensuring they can execute transactions even when the Sequencer is unavailable. This effort aligns closely with Arbitrum's objectives of decentralization and bolstering user autonomy.

To ensure the solution is effective during critical scenarios, it's essential that other projects can independently deploy their own instances of this service. This is why the project is conceived as open-source from inception, and we are committed to actively supporting teams interested in adopting it.

What's Next?

As we embark on the first phase — Technical Research, POC, Documentation, and Testing — we will keep you updated with all the progress!

Also, you can check out our general communications thread on this forum to learn more about us and what we’re building on Arbitrum!