Vitalik outlined 3 stages of training wheels for decentralizing Layer 2 rollups.

There are currently a large number of (optimistic and ZK) rollup projects, at various stages of development. One pattern that is common to almost all of them is the use of temporary training wheels: while a project's tech is still immature, the project launches early anyway to allow the ecosystem to start forming, but instead of relying fully on its fraud proofs or ZK proofs, there is some kind of multisig that has the ability to force a particular outcome in case there are bugs in the code.

Notably this framework has received some buy in from major projects, such as Coinbase.

From the Coinbase perspective, this would be a valuable resources for presenting our users with a standardized risk assessment as they interact and move assets across different rollups. We would be excited to collaborate on fleshing this out and making this a resource that can be standardized across the broader ecosystem. - Jesse Pollak

In all of Aztec's existing governance proposals there are effectively no training wheels, bypassing "Stages 1 & 2" all together, and maybe even "Stage 3".

Let's discuss this in more depth, and consider what kind of training wheels could be used, taking inspiration from "Stage 3", the most decentralized stage outlined within the framework.

This is not a requirement, only hoping facilitate the discussion while the upgrade RFP is live

Primary concern

In the event of a security vulnerability, or buggy code, it may be difficult to patch. This post does not concern itself with malicious governance attacks, as those are discussed in more depth throughout existing work.

Definitions

- Halt - the inability to produce a new rollup and state transition
- Exploit/bug - an issue in the code otherwise resulting in non-expected functionality, but the state continues to progress

It's hard to define generic categories here given the amount of uknown unknowns but :shrug:

Current Comparisons

Currently there are 3 governance proposals.

- Non-governance
- The Republic
- The Empire Stakes Back

Let's discuss each of these and how they handle halts & bugs.

Non-governance

Concern

Halt

Bug

What can be done?

Nothing

Race to exit, or nothing

Non-govnernace stands out because there is no mechanism by which the system can be updated and therefore in the event that the rollup has halted, nothing can be done… Everyone's assets will likely be stuck, forever.

The Republic

Concern

Halt

Bug

What can be done?

Wait 30 days to release fix

Race to exit, or wait 30 days

Note that the Republic outlines a spectrum of optional governance for portals. You could be fucked if you are on a non-governable portal because it could be impossible to create the planned “exit”, depending on the issue. The same is true within the empire stakes back.

The Empire Strikes Back

Concern

Halt

Bug

What can be done?

Wait 30 days to release fix

Race to exit, or wait 30 days

Breakdown

Now that we have a general understanding of the concerns and the proposal’s solutions to these, let’s discuss why this is the case in more depth, and potentially some improvements or solutions to the challenge.

In Lasse’s proposal, The Republic, he defines a 30 day upgrade window to guarantee that if a user has issues with the decisions the Senate is making via governance proposals, they have sufficient time to force exit their assets to L1. This is good and generally addresses the issue of being rugged via malicious governance proposals, for example like we recently saw with Tornado Cash.

In Joe’s proposal, The Empire Stakes back, he leverages the framework Lasse defined and (I believe) has the same 30 day exit for the same reason - malicious governance.

However, neither of these proposal differentiate between a malicious govenrance proposal and the ability to remedy a halted network. In the event that no rollups are prouduced and we must wait 30 days to fix, that would cause a significant amount of harm to the network and it’s reputation, not to mention an inability to access funds for a whole month.

It is important that this is a well understood design decision.

Generally, I think that this can be improved, and is worth consideration

Improvement suggestions

In the event of a halt, I think that within >= 7 days, there should be a mechanism that allows upgrades, by passing the traditional 30 day window.

This was proposed for “Stage 3” rollups in Vitalik’s training wheel framework:

If no valid proof is submitted for >= 7 days (ie. “the prover is stuck”), control temporarily turns over to the security council

In the case of the republic or the empire stakes back, there is not necessarily a security council, but rather a Senate that can update the registry contract which points to the canoncial and current version of the network. Simply put, I suggest we enable them to update the software version registry after the 7 day window in the event of a halt, rather than 30.

Pushing other changes in a hotfix

It is critically important that the scope of these changes are limited to fixing the bugs, and that client/contract/etc developers do not

include any other changes in these releases. I’m unsure how this would be enforced, unfortunately, but assuming that we choose the republic or the empire stakes back, there is a general trust assumption on the senate, anyways.

Other considerations

Vitalik outlines two other criteria that Stage 3 rollups can use.

1. The rollup uses two or more independent implementations of its state transition function (e.g. two distinct fraud provers, two distinct validity provers, or one of each), and the security council can adjudicate only if they disagree - which would only happen if there is a bug

2. If someone submits a transaction or series of transactions that contains two valid proofs for two distinct state roots after processing the same data (ie. “the prover disagrees with itself”), control temporarily turns over to the security council

Both worth considering, but likely a bit more work than a simple halting training wheel.