Overview

In this post, we’ll explain the necessary steps for upgrading the rules of an Arbitrum chain (namely, its state transition function). Changing the rules of an Arbitrum chain requires two things to happen: Arbitrum nodes must update their software, and an on-chain operation (i.e., a governance proposal) must take place. We’ll compare and contrast upgrades to an Arbitrum Layer 2 with upgrades to Layer 1 Ethereum to illustrate why the on-chain component of an Arbitrum upgrade isn’t just a loose formality, but a requirement.

Ethereum Consensus Changes

Changes to the rules of Layer 1 Ethereum, also known as “consensus changes,” only require that Ethereum nodes software be updated. Not all Layer 1 software updates are consensus changes - only updates that alter which blocks are considered valid are called consensus changes. For example, after a consensus change, a block that would have previously been considered valid may now be invalid and thus rejected, and conversely, a block that would have been considered invalid may now be valid and thus accepted.

It follows that if only a subset of Ethereum nodes “accept” a consensus change (by updating their software accordingly), it’s possible to get a chain-split — a situation in which two conflicting versions of the Ethereum blockchain persist in the network. Consensus upgrades are thus sometimes referred to as “hard forks” or “soft forks” (the distinction between hard and soft forks is nuanced and weird and out of scope for this post).

For Layer 1s like Ethereum, chain splits are resolved not by any technical process, but simply via social consensus. If there’s disagreement about which rules govern Ethereum, it ultimately falls on the community at large — node runners, developers, users, exchanges, etc. — to collectively make the decision. Ultimately, the question of what rules and what blockchain history represents the “true” Ethereum is one that, at the bottom level, comes down to social consensus.

Arbitrum ArbOS Upgrades and the Arbitrum Bridge

Upgrades to the rules of an Arbitrum chain (aka “ArbOS upgrades”) similarly involve Arbitrum node operators coordinating upgrading their software. Again, with ArbOS upgrades, we’re restricting our focus to changes which could alter the validity of Arbitrum blocks (“RBlocks”.) As with Layer 1 Ethereum, if nodes of a given Arbitrum chain are running different versions of ArbOS, this could lead to them producing two different, inconsistent versions of the chain’s history. Unlike with Ethereum, however, there is an “objective” way to determine which of the two competing chains is the canonical one: namely, the Arbitrum Bridge.

An Arbitrum Chain’s bridge is a series of smart contracts that reside on its parent chain; e.g., Arbitrum One’s bridge contracts reside on Ethereum. The bridge is responsible for maintaining information about the state of the Arbitrum chain back on Layer 1. This, among other things, enables users to transfer assets between Layer 1 and Layer 2.

The bridge ensures that its view of the Arbitrum chain is accurate via a system of assertions and fraud proofs. Arbitrum validators can make claims about the state of the chain, and other validators can challenge them. The challenge process is adjudicated by the bridge contracts; you can read about the details here. In brief: validators play an interactive game which narrows down their disagreement to a single computational step, which is then executed in the Layer 1 bridge contracts.

Crucially, this means that the Layer 1 bridge contracts need to be aware of the Arbitrum chain’s execution rules. In the case of an Layer 2 split as described above, only one chain (at most) can be valid in a way that’s consistent with rules codified in the Arbitrum bridge; this means that on only one chain will users be able to withdraw their funds. Likewise, in the case of an ArbOS upgrade, the upgrade to the node software only preserves users’ abilities to withdraw their funds from the bridge if it includes the corresponding update to the Layer 1 bridge contracts.

Conclusion

Where Ethereum is ultimately governed by social consensus, Arbitrum chains have their rules defined and enforced via fraud proofs by smart contracts on Layer 1. Thus, changes are determined not via a subjective social contract, but an actual smart contract. In the case of Arbitrium One, updates to these contracts require that the Arbitrum DAO pass a governance proposal changing the set of rules enforced.