Economics of Disputes in Arbitrum BOLD

The following document explains the economics and denial-of-service mechanisms built into Arbitrum BOLD. It covers trade-offs Arbitrum has to make to enable permissionless validation, explaining the key problems in an accessible way.

Background

Arbitrum One (Arb1) is currently one of the most widely adopted Ethereum scaling solutions, with ~$18bn USD in total-value-locked at the time of writing. Not only do its scaling properties make it popular, such as its 250ms block times, but so do its security properties and its approach to decentralization. Currently, Arbitrum One is governed by the Arbitrum DAO, one of the most active and robust onchain organizations.

However, Arbitrum has not yet achieved its full promise of decentralization. Currently, withdrawals from Arbitrum One back to Ethereum are verified by a permissioned list of validators. These validators can still challenge invalid withdrawals, but the system prevents anyone outside this list from holding them accountable. This limits Arb1 to being categorized as a Stage 1 rollup, according to the L2Beat website, meaning it still has training wheels preventing it from reaching its full potential.

The reason why Arb1 is called “optimistic” is because claims about its state settle to Ethereum after a period of ~7 days, in which they can be disputed. To make an analogy, a check can be cashed right away, but can be taken to court to dispute if there is a problem within a certain time frame. Because Arbitrum’s state is deterministic, a validator that is running a node and following the chain will always know if a posted claim is invalid. A key decentralization property is allowing **anyone** that knows the correct claim to challenge invalid claims and ***win***. This preserves the correct history of Arbitrum settling to Ethereum and protects the integrity of users’ funds along with their withdrawals using a “single honest party” property. As long as there is a single entity following the chain and willing to dispute a claim, Arbitrum’s security guarantees are maintained.

Today, the security properties of Arb1 are defined by the size of the permissioned set of validators it has. Validators could collude, could settle an incorrect history, and users have no recourse aside from the Arb1 security council stepping in. To elevate Arb1’s decentralization, it needs a different approach.

In the fall of ‘23, Offchain Labs announced Arbitrum BOLD, a brand new, dispute resolution protocol built from the ground up that will bring Arbitrum chains to the next level of decentralization. BOLD, standing for Bounded Liquidity Delay, allows for **permissionless** validation of Arbtrum chains. This means the DAO can remove the list of allowed validators and let anyone challenge claims made about Arbitrum states on Ethereum, and win against them if they are incorrect.

In this document we’ll go deep into understanding the economics and tradeoffs that Arbitrum must make in order to enable permissionless validation.

Settling Arbitrum States to Ethereum

We frequently state that “Arbitrum settles its state to Ethereum”, and we’ll elaborate on what that exactly means. All Arb1 transactions can be recreated by reading data from Ethereum L1, as compressed batches of all L2 txs are frequently posted to Ethereum. Once a batch transaction is included in a finalized block on Ethereum, its history will never get reverted on Arbitrum. Ethereum, however, does not know if a batch posted to it refers to a correct Arbitrum history. For verifying batch integrity, there is a separate process that actually confirms batch correctness on Ethereum.

Separately, entities known as validators are checking the correctness of batches by following the Arbitrum chain. Approximately every hour, they post something called an “assertion” which attests to the validity of a batch, saying “I have verified this batch”. As Ethereum has no knowledge of what is actually correct on Arbitrum, it allows ~7 days for anyone to come in and dispute one of these assertions.

Withdrawing Assets Back to Ethereum from Arbitrum

Users of Arb1 that have bridged assets from Ethereum can withdraw said assets at any time. However, for this withdrawal to be fully executed, a withdrawal claim has to correspond to a confirmed assertion on Ethereum. For instance, Alice starts a withdrawal transaction on Arb1, which gets posted in a batch on Ethereum. Then, a validator will post an assertion about that batch on Ethereum 1 hour later. The assertion has a ~7 day window in which anyone can dispute it, otherwise, it will get confirmed. After that window passes, Alice will receive her withdrawn assets on Ethereum and is free to use them as she pleases.

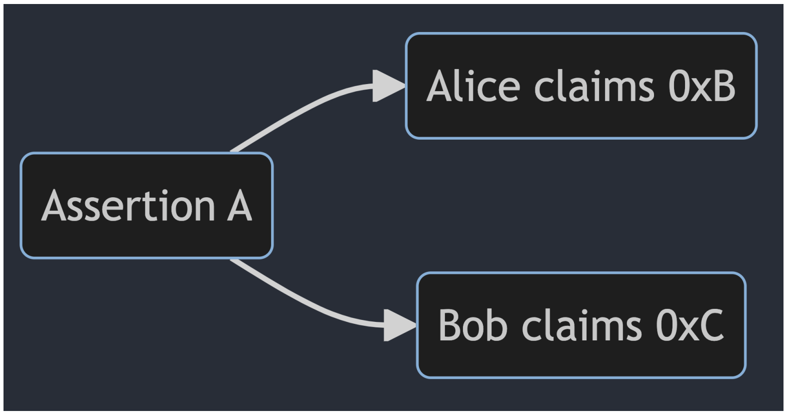
“Settling states" and having a ~7 day dispute window is crucial to ensuring assets can be withdrawn safely. Allowing anyone to dispute invalid claims and wins helps keep withdrawals protected by strong security guarantees, without needing to trust a group of validators. It is this “permissionless validation” what separates Optimistic Rollups from side chains.

The Dispute Period

The reason there is a dispute window for assertions about Arb1 on Ethereum is because Ethereum itself **has no knowledge about what is actually correct** on Arb1. The two blockchains are different domains with different states. Ethereum, however, can be used as a neutral referee for parties to dispute claims about Arb1. The reason why the dispute period is ~7 days is because it is seen as the maximum period of time an adversary could possibly delay Ethereum before social intervention, originally proposed by Vitalik Buterin. This window gives enough time for parties to catch invalid claims and challenge them accordingly.

Dispute Resolution Times

An actual dispute occurs once a party disagrees with an assertion on Ethereum, and posts their own assertion they know to be correct. This creates a “fork” in the chain of assertions, requiring a resolution process. We’ll get into the high-level details of how disputes are resolved soon.



Once an actual dispute is ongoing, it will also take time to resolve, as, once again, Ethereum has no knowledge of the correctness of Arbitrum states. Ethereum must then give sufficient time for parties to provide their burden of proof and declare a winner. The new, Arbitrum BOLD protocol **guarantees that a dispute will be resolved within 7 days** so long as there is an honest party present to defend against invalid claims.

As assertions have a dispute window of 7 days, and disputes require an additional 7 days to resolve, a dispute made at the last second would **delay assertion confirmation to a maximum of 14 days**, or 2 weeks. BOLD is the only dispute protocol we are aware of that guarantees this bound.

The Cost of Delaying Withdrawals

Delaying withdrawals incurs opportunity cost and a worse user experience for users that want to withdraw their assets. In the happy case, where no disputes exist, withdrawals already have a baked-in, 7 day delay. A dispute incurs an additional 7 days on top of that maximum. The problem is that disputes delay *all* pending withdrawals from Arb1 back to Ethereum, not just a single claim. As such, **disputing a claim must have a cost for the initiator** proportional to the opportunity cost they impose on Arbitrum users.

Requiring a Bond to Become a Validator

The entities responsible for posting assertions about Arbitrum state to Ethereum are called validators. If posting assertions were free, anyone could create conflicting assertions to always delay withdrawals by 14 days instead of 7. As such, Arbitrum requires validators to put in a “security deposit”, known as a **bond**, to be allowed to post assertions. Validators can withdraw their bond as soon as their latest posted assertion has been confirmed, and end their responsibilities.

Pricing Bonds

Ensuring assertions are frequently posted is a requirement for Arbitrum, but at the same time, it should not be a privilege that is easily obtained. In terms of pricing this “security deposit” for Arbitrum validators, we choose to do so based on opportunity cost.

To be extremely conservative, in a bank-run scenario, the Arbitrum One bridge contains approximately $5.4bn worth of assets at the time of writing on April 15th, 2024. Assuming funds could earn a 5% APY if invested, the opportunity cost of 1 extra week of delay is approximately $5,200,000 USD. Given this scenario, we recommend a bond for assertion posters somewhere in the range of ~$2.5M - $5M.

Honest parties can always withdraw their bond once their assertions are confirmed. However, adversaries stand to lose the entirety of their bond if they post invalid claims. A large bond size drastically improves the economic security of the system based on these two axes.

Centralization Concerns

Requiring a high bond to post assertions about Arbitrum seems centralizing, as we are replacing a whitelist of validators with instead a system that requires a lot of money to participate in.

However, **BOLD ships with a trustless bonding pool** for assertion posting. That is, any group of honest parties can pool funds into a simple contract that will post an assertion to Ethereum without needing to trust each other. We believe that making it easy to pool the funds to become an assertion poster, without needing trust to dispute invalid claims, does not fundamentally affect the safety or decentralization of BOLD.

We claim optimizing for the unhappy case is more important than the happy case. As there only needs to be one honest assertion poster, we believe it falls into the security budget of the chain to price in a $2M bond to become a validator. It *should* be expensive to delay Arb1 withdrawals, and it should also have a high barrier to entry to perform a key responsibility. As long as disputes can be made in a trustless manner, and trustless pools are available in production, we claim the security properties of assertion posting hold equally.

Resolving Disputes

One of the core properties BOLD achieves is providing a fixed, upper-bound for dispute resolution times. This section will discuss the constraints required to achieve this from first principles.

Dispute Game Overview

Every game between adversarial parties needs a referee – that is, a neutral party that can enforce the rules to declare a fair winner. Arbitrum BOLD relies on Ethereum as its referee, for its properties as the most decentralized, censorship resistant smart contract chain in the world.

When a dispute happens about Arb1 assertions on Ethereum, there is a protocol for resolving them. A dispute, at its core, is about the blockhash of an Arb1 block at a given height. Ethereum does not know which claim is correct, and instead, relies on a dispute game to be played out. The game involves different parties making claims with proof to eventually narrow down their disagreement to a single step of execution within the execution of a block, called a one step proof (OSP). Ethereum can then verify this OSP by itself and declare a winner as the neutral referee[1].

The “rules” of the dispute involve parties making claims with proofs to reach the single point of disagreement. Parties “narrow down” their claims via moves called bisections. After a party has made a bisection, it has nothing else left to do until another party comes in and counters it. The core of the system is that an honest party winning a one step proof leaves the evil party with no other moves to make. Once the honest party has accumulated enough time not-countered, it will be declared the winner.

Compared to other dispute protocols, however, BOLD is **not** a dispute between two specific Ethereum addresses, such as Alice and Bob. Instead, it is a dispute between an absolute,

[1] This concept is how Arbitrum got its name. Ethereum is the final “arbitrator” of disputes about the rollup’s state

correct history vs. an incorrect one. Claims in BOLD are not attached to a particular address or validator, but instead to Merkle commitments of an Arbitrum chain's history. If Alice and Charlie are both honest, and Bob is malicious, Alice and Charlie can play the game as part of a single "team". If Alice goes offline in the middle of a dispute game, for instance, Charlie can continue resolving the game in the honest team's favor at any time, because Charlie and Alice being honest means they will claim and make moves on the correct history. This is why we say BOLD enables "trustless cooperation", as there is no need for communication between parties that are honest. We believe committing a set of chain history hashes instead of a specific hash at a moment in time is crucial for securing dispute protocols.

Spamming the Dispute Game

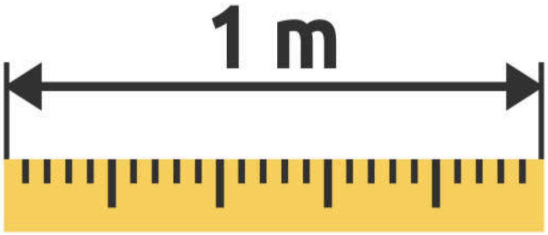
BOLD is a dispute game in which the party that has accumulated ~7 days "not-countered" wins. That is, parties are incentivized to counter any new claims as soon as they appear to "block" their rivals from increasing their timers. For honest parties, responding to claims may sometimes require offchain computational work, which needs access to computational resources such as CPUs. However, evil parties can just make claims that are eventually found to be junk while making honest parties do actual work.

Because evil parties can submit junk that makes honest parties do work, there has to be an economic cost associated with making moves in the dispute game. That is, we need a way to prevent **spam attacks** in dispute games.

The Cost of Moves

When thinking about how to price the bonds required to make claims within disputes, we essentially consider the marginal costs that the honest party incurs for each claim an evil party makes. In the BOLD research paper, this comes out to include information such as the number of adversary moves, multiplied by the gas cost of making bisections, making claims, and some estimates of the offchain computational costs. We deem this the **marginal cost** of a party in a dispute.

In BOLD, the space of disagreements between parties is of max size 2^43. As such, the dispute game has to be played at different levels of granularity to make it computationally feasible.



For instance, say we have two, 1 meter sticks that seem identical, and we want to figure out where they differ. It turns out that they seem identical at the centimeter level, so we need to go down to the millimeter level, then the micrometer level, and then figure out where they differ at the *nanometer* level.

This is what BOLD does over the space of disputes. Parties play the same game at different levels of granularity. At the centimeter level, each centimeter could trigger a millimeter dispute, and each millimeter dispute could have many micrometer disputes, etc. This fans out to a large number of potential dispute games unless spam is discouraged.

Preventing Spam

Given Ethereum knows nothing about which claims are honest or evil until a one step proof is reached, then how can the protocol detect spam and discourage it? A key insight is that honest parties only need to make one honest claim. Honest parties will never spam and create thousands of conflicting claims with themselves. Given this, we can put a price tag on making moves by looking at something called the “resource ratio” between honest and evil parties, as defined in the BOLD research paper

$$\mathcal{R} := \frac{G_{\mathrm{A}} + S_{\mathrm{A}}}{G_{\mathrm{H}} + S_{\mathrm{H}}},$$

This ratio is essentially the gas + staking (or bonding) marginal costs of the adversary to the honest party. This means that certain values input into the equations can lead to **different ratios**. For instance, we can say the adversary has to pay **10x** the marginal costs of the honest party. However, aiming to increase this ratio significantly by plugging in different values leads to higher costs for all parties.

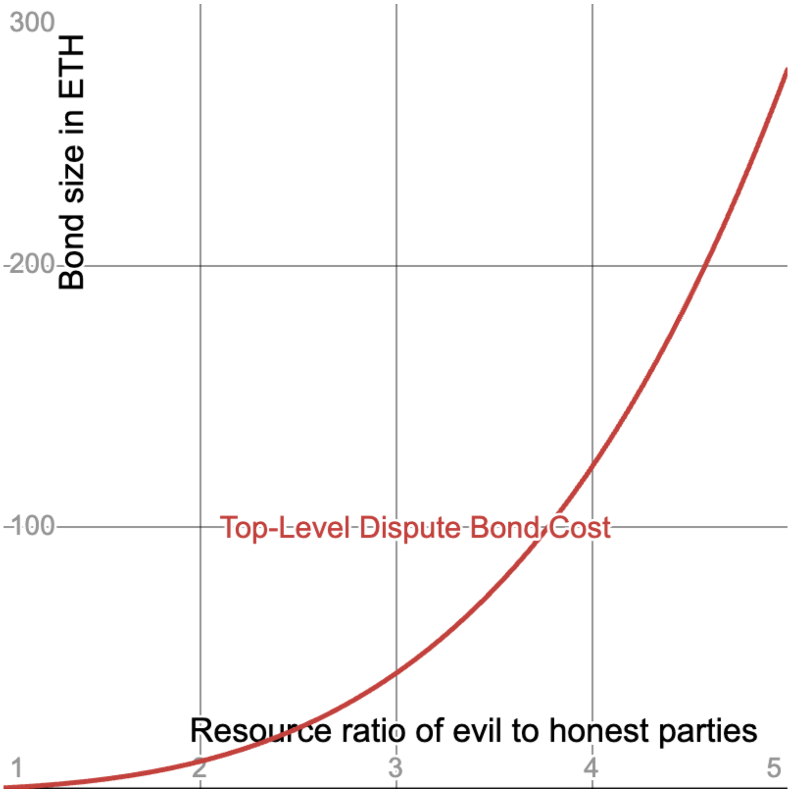
Dispute Mini-Bonds

We require parties to lock up some capital called a “mini-bond” when making big claims in a dispute. These bonds are not needed when making bisection moves, but are critical for posting an initial claim. Pricing these mini-bonds helps us achieve a high resource ratio of evil parties to honest parties.

It is clear that if we can make the cost to the evil party some multiplier of the honest party, we get significant security benefits. For instance, imagine if a $1 billion dollar attack can be defended by simply pooling together $10 million dollars. Is it possible to achieve such a ratio?

Let’s explore the limitations of making the cost to evil parties high vs. that of the honest parties. That is, if we aim to have a constant resource ratio > 1, we have to do the following: if the adversary makes N stakes at any level, they can force the honest party to make N stakes at the next level down where the adversary can choose not to place any stakes at all. In terms of resource ratio, to make the adversary always pay 10x in staking, we need to make the bond amount at one level 10x more than the next. As there are multiple levels, the equations for the bond size include an exponential factor on the desired, constant resource ratio > 1.

Below, we plot the bond size vs. the resource ratio of evil to honest costs. The source for these equations is both in the research paper and plotted here.



If we desire a constant resource ratio of evil to honest costs > 1, the required bond size in ETH increases as a polynomial at a particular challenge level.

Trade Offs

Having a 1000x resource ratio would be nice in theory, but would unfortunately require a bond of 1M ETH ($3.5bn) to open a challenge in the first place, which is unreasonable. Instead, we can explore a more feasible ratio of 10x.

The tradeoff here is the higher the resource ratio we want, the more expensive it is for both honest and evil parties to make claims in disputes. However, claims can **always be made** through a **trustless pool**. Honest parties can pool together funds to participate in disputes.

The Sweet Spot

We estimate that at a resource ratio of 10x, the cost of resolving a single adversarial claim in a dispute would be approximately $16M USD for the honest parties. Honest parties will always be refunded, and possibly rewarded, while evil parties always stand to lose 100% of their bond. The DAO could then consider the cost of incentivizing a single honest staker in the happy case to be the **security budget of the chain**. This means defending against a $1 billion attack would require $100M total to defend, with the entire amount being refunded while the attacker losing all $1bn.

Thinking About Incentives

Although we have made claims with hard numbers about how to price disputes and withdrawal delays in Arbitrum BOLD, we also take a step back and think about the game theoretical assumptions we are making. Arbitrum One is a complex protocol used by many groups of people, with many different incentives. The research team at Offchain Labs, has spent considerable effort studying the game theory of validators in Optimistic Rollup. Honest parties represent everyone that has funds onchain, and they have a huge amount to gain by winning the challenge - as they can prevent the loss of their assets rather than losing them.

A more complex model is proposed which considers all parties staking and their associated costs created by Akaki Mamageishvili and Ed Felten in their paper “Incentive Schemes for Rollup Validators”. The paper looks at what incentives are needed to get parties to check whether assertions are correct. It finds that there is no pure strategy Nash equilibrium, and only a mixed equilibrium if there is no incentive for honest validators. However, the research showed a pure strategy equilibrium can be reached if honest parties are incentivized to **check** results. The problem of honest validators “freeriding” and not checking is well-documented as the verifier’s dilemma. We believe future iterations of BOLD could include “attention challenges” that reward honest validators for also doing their job.

Conclusion

This paper summarizes the rationale behind choosing bond sizes and the cost of spam prevention in Optimistic Rollup dispute protocols. We recommend that bond sizes be high enough to discourage challenges from ever being opened at all, as evil parties will always stand to lose when playing the game. As Arbitrum BOLD does not tie disputes to specific addresses, honest parties can have trustless cooperation to resolve disputes if desired. We posit that making the cost of the evil parties be 10x that of the honest party leads to nice economic properties that help us reason about how to price bonds. Finally, we look at a high-level game theory discussion of Optimistic Rollups and argue that solving the verifier’s dilemma by incentives to honest validators is an important addition to work towards.