

# Arbitrum

## 1. What is Arbitrum?

Arbitrum is a Layer 2 blockchain built on top of Ethereum, and its purpose is to make using Ethereum faster, cheaper, and more scalable. Ethereum, while extremely secure and decentralized, has some limitations. It can only handle a limited number of transactions per second (approx 20-40), and as more people use it, the network becomes congested, and gas fees increase significantly.

To solve this problem, projects like Arbitrum have been created. They are called Layer 2 solutions because they are built "on top" of the main Ethereum network (which is known as Layer 1). Layer 2 solutions handle most of the work -like processing transactions -and then send a summary of this work back to Ethereum for final confirmation. This allows Arbitrum to benefit from Ethereum's security, while also offering faster speeds and lower costs.

So, Arbitrum acts like an assistant to Ethereum: it handles all the traffic, clears the road, and only sends the final report to Ethereum, which acts as the ultimate judge of whether the transactions are valid.

## 2. How does Arbitrum Work?

Arbitrum uses a technology called a rollup. The basic idea of a rollup is to take many individual Ethereum transactions, bundle them together into one big group (like rolling up a sheet of paper), and send a single summary of that group to the Ethereum network.

This makes things more efficient because Ethereum doesn't have to check every single transaction. It only has to check the summary. Arbitrum handles all the detailed work off-chain (outside of Ethereum), and only interacts with Ethereum when needed.

### 2.1. Arbitrum Rollups

Arbitrum Rollups are a special type of Layer 2 technology built on top of Ethereum. They are designed to solve Ethereum's biggest problems - high gas fees and slow transaction speed without compromising on security.

Let's say hundreds of users are doing transactions. If they all directly used Ethereum, each transaction would go one by one through Ethereum's slow, expensive road. But Arbitrum works differently.

Instead of sending each transaction to Ethereum, Arbitrum collects a bunch of transactions off-chain - kind of like gathering 500 people who all want to go to the same destination. Then, it rolls (or compresses) all those into one big summary and sends that summary to Ethereum.

This summary contains:

1. What happened (the result of those 500 transactions).
2. Proof that it was done correctly (not just random claims)

That's why it's called a Rollup because it rolls up many transactions into one, saving time and money.

## 2.2. How does Ethereum know those transactions were valid?

Ethereum itself does not verify every little step that happened in Arbitrum. Instead, it relies on a clever system where Arbitrum posts the result and says, "Here's what we did."

But what if someone lies?

That's where the fraud-proof mechanism comes in. In Arbitrum's system, when a batch of transactions is posted to Ethereum, there's a short waiting period (called the challenge period). During this time, anyone can step in and say, "Wait! Something's wrong here."

If someone challenges the result, Arbitrum runs a dispute game -a back-and-forth process between the original poster and the challenger. They keep narrowing down where the mistake might be until the system finds the exact step that was wrong.

If the poster was wrong, their post is rejected, and the challenger gets rewarded. If the poster was right, the challenger gets punished. This creates a system where everyone is motivated to be honest, because lying would cost them money.

## 2.3. Arbitrum Fraud Detection Process

1. Batch Submission
  - Sequencer processes transactions off-chain and posts state commitment to Ethereum
  - Claims: "These transactions change state from A to B".
2. Challenge Period
  - 7-day window for validators to verify results.
  - Validators re-execute transactions independently.
  - Compare results with sequencer's claims.
3. Fraud Detection
  - If the validator's result  $\neq$  sequencer's result  $\rightarrow$  submit a challenge with stake.
  - Interactive dispute process begins.
4. Interactive Fraud Proof
  - Binary search narrows dispute to single problematic transaction.
  - Ethereum executes that one transaction to determine truth.
5. Resolution
  - Fraud proven: fraudulent state reverted, malicious party loses stake.
  - False challenge: challenger loses stake, original state stands

## 3. Why is Arbitrum Cheaper?

Arbitrum helps lower user transaction costs by reducing the strain on the parent chain. The primary way it does this is by processing transactions in batches. A batch can contain several hundred child chain transactions and gets submitted as one parent chain transaction. This batching makes interacting with the parent chain cheaper since you save on overhead costs compared to submitting each transaction individually.

Additionally, Arbitrum posts transaction data on the parent chain in a compressed format. It only decompresses this data within the child chain environment, reducing the amount of information that needs storing on the parent chain.

## **4. Arbitrum Ecosystem**

Arbitrum isn't just one thing. It's a full ecosystem with multiple components that serve different purposes. Here are the major parts:

### **4.1. Arbitrum One**

It's the main and most widely used network in the Arbitrum ecosystem. It's a Layer 2 blockchain, which means it's built on top of Ethereum.

What it does is take lots of transactions -for example, people swapping tokens on a DeFi app or minting NFTs -and batches them together into one big summary. Then it sends this summary back to Ethereum. This process is called a "rollup." Instead of each user paying for their transaction on Ethereum, now hundreds of transactions can share the cost of one. So things become much cheaper and faster for everyone. Arbitrum One still uses Ethereum to secure all its transactions, so the safety level is almost the same. It's good for apps that need strong trust and handle financial transactions like lending, trading, or NFTs. You can think of it as a fast expressway that runs above the main Ethereum highway -it avoids traffic but still leads to the same destination.

### **4.2. Arbitrum Nova**

Arbitrum Nova, which is like a younger sibling of Arbitrum One. It was created for a different purpose. While Arbitrum One focuses on security and decentralization, Arbitrum Nova focuses on speed and low cost, even if that means a small trade-off in decentralization. Nova uses a different technique called AnyTrust, which means instead of sending all the data back to Ethereum, it depends on a small trusted group of validators (called a data availability committee) to make sure the data is available. This makes Nova much cheaper because it doesn't have to store everything on Ethereum. Nova is perfect for things like online games, social media apps, and applications where thousands of small transactions are happening constantly but each transaction isn't worth a lot of money. A great example is Reddit, which actually uses Arbitrum Nova to manage its Community Points. If Arbitrum One is a secure delivery van that takes valuable packages across the city, Arbitrum Nova is like a bike courier delivering flyers super fast, cheap, and efficient, but not meant for carrying gold.

### **4.3. Arbitrum Orbit**

Arbitrum Orbit, which is not a chain itself, but a toolset that lets developers create their own custom blockchains. Imagine you are a developer and you want to build your own version of Arbitrum, maybe for a game, a DAO, or a private network. Orbit allows you to do this. You can take Arbitrum's technology -either the rollup model like Arbitrum One or the AnyTrust model like Nova -and build your own network on top of Arbitrum or even directly connected to Ethereum. You get to choose the rules, the fees, the governance, and even how secure it is. It's like if Arbitrum One is a national highway, Orbit gives you the tools to build your own highway, your own toll system, and your own speed limits. This is especially useful for big projects that want full control but don't want to build everything from scratch. So Orbit is enabling a future where multiple independent networks can exist, all connected through the Arbitrum framework.

## 5. How users send Transactions to the Arbitrum Chain

### 5.1. Submitting Transactions to the Sequencer

The Sequencer is like a fast lane that quickly organizes and processes transactions on Arbitrum.

- **Public RPC** - Free and easy to use, but limited in speed. Good for simple apps or testing.
- **Third-Party RPC** - More powerful (e.g., Infura, Alchemy). Best for apps needing high performance.
- **Self-Hosted Arbitrum Node** - Run your own node for full control. Sends transactions directly via the Sequencer Feed.
- **Sequencer Endpoint** - Fastest method. Sends raw transactions directly to the Sequencer (no load balancer).

### 5.2. Bypassing the Sequencer

You can also skip the Sequencer and submit transactions to Arbitrum through Ethereum itself (called the Delayed Inbox)

- If the Sequencer is down or ignoring your transaction, this method guarantees inclusion.
- You send the transaction to a special Ethereum contract.
- If the Sequencer ignores it for 24 hours, you can force it to include your transaction using the `forceInclude` function.

## 6. What is Sequencer

The Sequencer is like the traffic controller of Arbitrum. It accepts transactions from users, orders them, batches and compresses them, and sends them to Ethereum (the parent chain) for final confirmation. This helps make transactions fast, efficient, and low-cost.

### 6.1. Functions

- **Real Time Feed**
  - Sends instant updates about the order of transactions.
  - Apps and users get quick feedback (called *soft finality*)
- **Batch Processing**
  - Groups transactions into batches.
  - Uses compression (Brotli algorithm) to reduce size and cost.
  - Posts the compressed data to Ethereum.
- **Finality**
  - **Soft Finality**: Fast but temporary; comes from the Sequencer feed.
  - **Hard Finality**: Slow but permanent; comes from posting to Ethereum.

## 7. Gas and Fees

When you make a transaction on Arbitrum, you pay two separate fees. The dual fee structure ensures that Arbitrum can operate sustainably while maintaining security.

- To run your transaction on Arbitrum this is called L2 gas.
- To store your transaction on Ethereum this is called L1 calldata fee.
- Total Fee = L2 Gas Fee + L1 Calldata Fee.

### 7.1. L2 Gas Fee - Arbitrum Execution Fee

This is like paying rent to use Arbitrum's computer so it can run your smart contract or transaction.

#### Formula:

$\text{gasUsed} \times \text{gasPrice}$  (called basefee)

Arbitrum automatically adjusts this gas price:

- If network is busy → gas price goes up
- If quiet → gas price goes down

### 7.2. L1 Calldata Fee

Arbitrum must prove your transaction to Ethereum by posting it there (for security).

This costs money, because posting data to Ethereum is expensive.

So, you must pay the person (called the Batch Poster) who puts your transaction on Ethereum.

#### Execution:

- Transaction gets compressed (smaller size = cheaper cost).
- System counts how many bytes it will take on Ethereum.
- Each byte costs about 16 gas on Ethereum.
- Multiply that by Ethereum's current gas price (basefee).
- That's the L1 calldata cost.
- Then, Arbitrum converts this Ethereum cost into L2 units so you can pay it easily.

## 8. Who Governs Arbitrum

Arbitrum is not controlled by any single person, company, or group. Instead, it is governed by a community of people from all over the world through a system called Arbitrum DAO. A DAO is short form of "Decentralized Autonomous Organization." In simple words, a DAO is like a digital version of a democratic parliament where instead of politicians, the people who hold a special token get to vote on decisions.

In Arbitrum's case, this token is called ARB. Anyone who owns ARB tokens becomes part of the DAO and can participate in the decision-making process. They can vote on things like how the network should evolve, what upgrades should be made, how to spend money from the treasury (which is a large pool of ARB tokens set aside for development), and other key changes.

Let's use a real-world example. Imagine Arbitrum is a big cooperative housing society. All the residents together decide how the building should be maintained, what upgrades to make, who to hire as staff, and how to use the shared funds. But to vote, each resident needs to show a proof of

ownership -in Arbitrum, that proof is the ARB token. The more tokens you have, the more weight your vote carries. However, it's not about power -it's about participation and transparency.

Now, how do people actually make these decisions? Through something called governance proposals. If someone wants to suggest a change like increasing block size, funding a new project, or upgrading the protocol they can write a proposal. Then, other token holders read the proposal, discuss it (often on the Arbitrum DAO forum), and vote "yes" or "no." If the vote passes, the change is implemented. If not, nothing happens.

This whole process is done on-chain, which means everything is recorded and transparent. There's no hidden voting, no secret leadership -it's all open and visible on the blockchain.