Arbitrum FAQ

{"@context":"https://schema.org","@type":"FAQPage","mainEntity":[{"@type":"Question","name":"Why do I need ETH to use the Arbitrum network?","acceptedAnswer":{"@type":"Answer","text":"

ETH is the currency used to pay gas fees on Arbitrum, and all Arbitrum transactions are powered by ETH. You can bridge ETH (and other tokens) from Ethereum to Arbitrum through **Arbitrum's bridge**.

\n\n

\n\n"}},{"@type":"Question","name":"Do I need to pay a tip or priority fee for my Arbitrum transactions?","acceptedAnswer":{"@type":"Answer","text":"

Since transactions are processed in the order that the Sequencer receives them, no priority fee is necessary for Arbitrum transactions; if a transaction does include a priority fee, it will be refunded to the transaction's origin address at the end of the execution.

\n\n"}},{"@type":"Question","name":"How can I see the balance of ETH and other tokens in my wallet on Arbitrum?","acceptedAnswer":{"@type":"Answer","text":"

Most wallets are \"connected\" to one given network at a time. To view your ETH or token balances, ensure that you are connected to the appropriate Arbitrum chain. In MetaMask, you can switch networks via the \"networks\" dropdown. In this dropdown, select your desired network (either Arbitrum One or Arbitrum Nova for our mainnet networks). If your desired network hasn't been added to your wallet yet, you can add it at https://bridge.arbitrum.io/.

\n\n

\n\n"}},{"@type":"Question","name":"What happens if I send my funds to an exchange that doesn't support Arbitrum?","acceptedAnswer":{"@type":"Answer","text":"

If you send the funds and the receiving wallet/exchange doesn't support the Arbitrum network you are sending funds through, there is unfortunately nothing that we can do to recover your funds. You would need to contact the wallet/exchange support and see if they can do anything to help you retrieve the funds.

\n\n

\n\n"}},{"@type":"Question","name":"Does Arbitrum have a mempool?","acceptedAnswer":{"@type":"Answer","text":"

The Arbitrum Sequencer orders transactions on a first come, first served basis; the Sequencer inserts transactions into a queue based on the order they are received and executes them accordingly. This queue thus exists in lieu of a mempool. The Sequencer's queue has no space limit; transactions on the queue will eventually timeout and be discarded if not executed in time. Under normal conditions, the queue is empty, since transactions are executed near-instantaneously.

\n\n

\n\n"}},{"@type":"Question","name":"What's the difference between Arbitrum Rollup and Arbitrum AnyTrust?","acceptedAnswer":{"@type":"Answer","text":"

Arbitrum Rollup is an Optimistic Rollup protocol; it is trustless and permissionless. Part of how these properties are achieved is by requiring all chain data to be posted on layer 1. This means the availability of this data follows directly from the security properties of Ethereum itself, and, in turn, that any party can participate in validating the chain and ensuring its safety. For more information, see Inside Arbitrum Nitro.

\n\n

By contrast, Arbitrum AnyTrust introduces a trust assumption in exchange for lower fees; data availability is managed by a Data Availability Committee (DAC), a fixed, permissioned set of entities. We introduce some threshold, κ, with the assumption that at least κ members of the committee are honest. For simplicity, we'll hereby assume a committee of size 20 and a κ value of 2:

\n\n

If 19 out of the 20 committee members *and* the Sequencer are malicious and colluding together, they can break the chain's safety (and, e.g., steal users' funds); this is the new trust assumption.

\n\n

If anywhere between 2 and 18 of the committee members are well behaved, the AnyTrust chain operates in \"Rollup mode\"; i.e., data gets posted on L1.

\n\n

In what should be the common and happy case, however, in which at least 19 of the 20 committee members are well behaved, the system operates without posting the L2 chain's data on L1, and thus, users pay significantly lower fees. This is the core upside of AnyTrust chains over rollups.

\n\n

Variants of the AnyTrust model in which the new trust assumption is minimized are under consideration; stay tuned.

\n\n

For more, see Inside AnyTrust.

\n\n

\n\n"}},{"@type":"Question","name":"How can I check the status of my cross chain message?","acceptedAnswer":{"@type":"Answer","text":"

You can check the status of *any* Arbitrum cross chain message at https://retryable-dashboard.arbitrum.io/ (you will also be able to execute the cross chain message there, if applicable).

\n\n

You'll need the transaction hash of the \"initiating transaction\": the L1 transaction hash for an L1-to-L2 message (e.g., a deposit), or the L2 transaction hash for an L2-to-L1 message (e.g., a withdrawal).

\n\n

\n\n

If you cross-chain message was initiated from https://bridge.arbitrum.io/, you can also check its status / execute it at that site in the transaction history tab.

\n\n

\n\n"}},{"@type":"Question","name":"If there is a dispute, can my L2 transaction get reorged / thrown out / \"yeeted\"?","acceptedAnswer":{"@type":"Answer","text":"

Nope; once an Arbitrum transaction is included on L1, there is no way it can be reorged (unless the L1 itself reorgs, of course). A \"dispute\" involves Validators disagreeing over execution, i.e., the outputted state of a chain. The inputs, however, can't be disputed; they are determined by the Inbox on L1. (See Transaction Lifecycle)

\n\n

\n\n"}},{"@type":"Question","name":"...okay but if there's a dispute, will my transaction get delayed?","acceptedAnswer":{"@type":"Answer","text":"

The only thing that a dispute can add delay to is the confirmation of L2-to-L1 messages. All other transactions continue to be processed, even while a dispute is still undergoing. (Additionally: in practice, most L2-to-L1 messages represent withdrawals of fungible assets; these can be trustlessly completed *even during a dispute* via trustless fast \"liquidity exit\" applications. See L2-to-L1 Messages).

\n\n

\n\n"}},{"@type":"Question","name":"Are \"Sequencers\" the same entities as \"Validators\"? Can a centralized Sequencer act maliciously (e.g., steal all my money)?","acceptedAnswer":{"@type":"Answer","text":"

No and no!

\n\n

An Arbitrum Chain's Sequencer(s) and Validators and completely distinct entities, with their own distinct roles.

\n\n

The Sequencer is the entity granted specific privileges over ordering transactions; once the Sequencer commits to an ordering (by posting a batch on Ethereum), it has no say over what happens next (i.e., execution). A malicious/faulty Sequencer can do things like reordering transactions or *temporarily* delaying a transaction's inclusion — things which could be, to be sure, annoying and bad — but can do nothing to compromise the chain's safety.

\n\n

The *Validators* are the ones responsible for the safety of the chain; i.e., making staked claims about the chain state, disputing each other, etc.

\n\n

Currently, on Arbitrum One, the Sequencer is a centralized entity maintained by Offchain Labs. Eventually, we expect the single Sequencer to be replaced by a distributed committee of Sequencers who come to consensus on transaction ordering. This upgrade will be an improvement; we don't want you to have to trust us not to reorder your transactions. However, it also isn't *strictly* necessary for Arbitrum One to achieve its most fundamental properties.

\n\n

In other words: ***An Arbitrum Rollup chain with a centralized Sequencer could theoretically still be trustless!***

\n\n

Which is to say — the more important thing than decentralizing the Sequencer, i.e., the thing you ought to care more about — is decentralizing the *Validators*.

\n\n

Arbitrum One's validator set is currently allowlisted; over time, we expectgovernance to expand the allowlist and eventually be removed entirely.

\n\n

For more info see \"State of Progressive Decentralization\".

\n\n

\n\n"}},{"@type":"Question","name":"Why was \"one week\" chosen for Arbitrum One's dispute window?","acceptedAnswer":{"@type":"Answer","text":"

Generally, some amount of time is necessary for the Arbitrum validators to dispute an invalid assertion.

\n\n

A week is expected to be more than enough time for validators to carry out an interactive dispute, assuming they don't encounter difficulty in getting their transactions included on L1. One week was chosen following the general consensus among the Ethereum research community — as well as other layer 2 projects — to provide enough time for the community to socially coordinate in the case of a coordinated Ethereum-staker censorship attack.

\n\n

\n\n"}},{"@type":"Question","name":"What's the state of Arbitrum One's decentralization?","acceptedAnswer":{"@type":"Answer","text":"

See **\"State of Progressive Decentralization\"**, or check out the work of our friends at**L2BEAT.**

\n\n

\n\n"}},{"@type":"Question","name":"Are there any Fiat on-ramps that support Arbitrum?","acceptedAnswer":{"@type":"Answer","text":"

Yes, you can find a list of Fiat on-ramps that support Arbitrumon our portal.

\n\n

\n\n"}},{"@type":"Question","name":"How many blocks are needed for a transaction to be confirmed/finalized in Arbitrum?","acceptedAnswer":{"@type":"Answer","text":"

There are two levels of finality in atransaction lifecycle:

\n\n

- Soft finality: once the Sequencer receives and processes a transaction, it emits a receipt through the Sequencer's feed. At this point, if the Sequencer is trusted, the transaction will not be reordered and the state of the chain after processing the transaction can be determined.
\n
- Hard finality: at this stage, assuming there's at least one well-behaved active Arbitrum validator, the client can treat their transaction's finality as equivalent to an ordinary Ethereum transaction.
\n

\n

\n\n"}},{"@type":"Question","name":"Where can I find stats for Arbitrum?","acceptedAnswer":{"@type":"Answer","text":"

Although we currently don't maintain any stats dashboard for Arbitrum, you can find manycommunity created dashboards in Dune.

\n\n

\n\n"}},{"@type":"Question","name":"Will transactions with a higher \"gas price bid\" be confirmed first?","acceptedAnswer":{"@type":"Answer","text":"

There is no notion of mempool on Arbitrum, transactions are processed on a first come first served basis by the Sequencer. Thus, the gas price bid parameter does not influence the order in which a transaction is processed.

\n\n

\n\n"}},{"@type":"Question","name":"Where can I find a list of the current validators of the Arbitrum chains?","acceptedAnswer":{"@type":"Answer","text":"

Validation on both Arbitrum One and Arbitrum Nova is currently allow-listed to a committee of public entities. You can see the list of validators **here**. Governance currently has the power to change this status.

\n\n

\n\n"}},{"@type":"Question","name":"Where can I find the current Data Availability Committee members?","acceptedAnswer":{"@type":"Answer","text":"

The Arbitrum Nova chain has a 7-party DAC, whose members can be seen**here**. Governance has the ability to remove or add members to the committee.

\n\n

\n\n"}},{"@type":"Question","name":"Can I withdraw my funds from Arbitrum back to Ethereum without going through the Sequencer? What about funds that are in a contract?","acceptedAnswer":{"@type":"Answer","text":"

Yes, it is possible to permissionlessly send a message from Ethereum to be executed on Arbitrum, while bypassing the Sequencer. You can do this by using the DelayedInbox contract and force-including the message after a certain amount of time has passed (currently ~24 hours). You can find more information about this behavior here.

\n\n

Keep in mind that you can execute any message in this way, be it a withdrawal of funds back to Ethereum, or a call to a contract.

\n\n

You can also find an example of force-inclusion inthis tutorial.

\n\n

\n\n"}},{"@type":"Question","name":"Are there any plans to reduce the time a transaction needs to wait before being able to be force-included from Ethereum into the Arbitrum chain, bypassing the sequencer? (Currently 24 hours)","acceptedAnswer":{"@type":"Answer","text":"

The mechanism that allows force-including transactions from Ethereum (bypassing the sequencer) is intended to be used in very rare cases, especially when it is expected that the sequencer will not be operational again, so that users have a way of interacting with Arbitrum in a trustless way.

\n\n

When using this mechanism, if the sequencer is down for longer than the time window for force-including transactions from Ethereum, the moment it is online again, it can lead to a reorganization of blocks in Arbitrum (it would have received transactions timestamped before the force-included one).

\n\n

24 hours was chosen because it provides a comfortable period of time for the team running the sequencer infrastructure to fix any bugs that may cause the sequencer to not work. While there aren't any active initiatives to lower that time, the decision ultimately falls in the hands of the Arbitrum DAO, who has discussed the topic in their governance forum (see here for more information).

\n\n

In any case, we could also analyze why would someone use this mechanism having an honest and functional sequencer. For instance, if the reason is a distrust of the sequencer, a centralised agent as of now, one potential solution could be to decentralize the sequencer instead of reducing the force-inclusion delay time.

\n\n

\n\n"}},{"@type":"Question","name":"What is the difference between an L2 block and a RBlock?","acceptedAnswer":{"@type":"Answer","text":"

An L2 block is very similar to the concept of an L1 block. These blocks are generated by validator nodes of Arbitrum by executing the state transition function on sequenced transactions. The structure of an L2 block is similar to that of an Ethereum block, with a few differences that you can see here.

\n\n

On the other hand, an RBlock is a distinctive block that is transmitted back to L1 to serve as a fingerprint of the most recent state of the Arbitrum chain. It comprises an assertion of the present state root of the Arbitrum chain and other essential information pertaining to withdrawals and challenges. The structure of RBlocks can be viewed here.

\n\n

These RBlocks are also generated by validators, but they are appended to the L1 chain. Other validators canchallenge them during a specific time frame of approximately one week if they discover that the current state hash of the chain varies from the one that was initially claimed. Once the challenge period elapses, the RBlock is confirmed on L1.

\n\n

\n\n"}},{"@type":"Question","name":"Why do Arbitrum chains enforce a speed limit? Isn't it better that the speed grows without limits?","acceptedAnswer":{"@type":"Answer","text":"

The transaction lifecycle sets a limit that we have to take into account: validators have to execute each transaction, get the status of the chain, and post an assertion to Ethereum every certain amount of time. If the speed of the chain increases too much, there is a risk that validators won't have enough computation power to process all transactions in a timely manner, and will fall behind on validating them, which would cause the chain to delay confirmations of its state.

\n\n

\n\n"}}]}

Why do I need ETH to use the Arbitrum network?

ETH is the currency used to pay gas fees on Arbitrum, and all Arbitrum transactions are powered by ETH. You can bridge ETH (and other tokens) from Ethereum to Arbitrum throughArbitrum's bridge .

Do I need to pay a tip or priority fee for my Arbitrum transactions?

Since transactions are processed in the order that the Sequencer receives them, no priority fee is necessary for Arbitrum transactions; if a transaction does include a priority fee, it will be refunded to the transaction's origin address at the end of the execution.

How can I see the balance of ETH and other tokens in my wallet on Arbitrum?

Most wallets are "connected" to one given network at a time. To view your ETH or token balances, ensure that you are connected to the appropriate Arbitrum chain. In MetaMask, you can switch networks via the "networks" dropdown. In this dropdown, select your desired network (either Arbitrum One or Arbitrum Nova for our mainnet networks). If your desired network hasn't been added to your wallet yet, you can add it athttps://bridge.arbitrum.io/ .

What happens if I send my funds to an exchange that doesn't support Arbitrum?

If you send the funds and the receiving wallet/exchange doesn't support the Arbitrum network you are sending funds through, there is unfortunately nothing that we can do to recover your funds. You would need to contact the wallet/exchange support and see if they can do anything to help you retrieve the funds.

Does Arbitrum have a mempool?

The Arbitrum Sequencer orders transactions on a first come, first served basis; the Sequencer inserts transactions into a queue based on the order they are received and executes them accordingly. This queue thus exists in lieu of a mempool. The Sequencer's queue has no space limit; transactions on the queue will eventually timeout and be discarded if not executed in time. Under normal conditions, the queue is empty, since transactions are executed near-instantaneously.

What's the difference between Arbitrum Rollup and Arbitrum AnyTrust?

Arbitrum Rollup is an Optimistic Rollup protocol; it is trustless and permissionless. Part of how these properties are achieved is by requiring all chain data to be posted on layer 1. This means the availability of this data follows directly from the security properties of Ethereum itself, and, in turn, that any party can participate in validating the chain and ensuring its safety. For more information, seeInside Arbitrum Nitro .

By contrast, Arbitrum AnyTrust introduces a trust assumption in exchange for lower fees; data availability is managed by a Data Availability Committee (DAC), a fixed, permissioned set of entities. We introduce some threshold,K , with the assumption that at leastK members of the committee are honest. For simplicity, we'll hereby assume a committee of size 20 and aK value of 2:

If 19 out of the 20 committee membersand the Sequencer are malicious and colluding together, they can break the chain's safety (and, e.g., steal users' funds); this is the new trust assumption.

If anywhere between 2 and 18 of the committee members are well behaved, the AnyTrust chain operates in "Rollup mode"; i.e., data gets posted on L1.

In what should be the common and happy case, however, in which at least 19 of the 20 committee members are well behaved, the system operates without posting the L2 chain's data on L1, and thus, users pay significantly lower fees. This is the core upside of AnyTrust chains over rollups.

Variants of the AnyTrust model in which the new trust assumption is minimized are under consideration; stay tuned.

For more, seeInside AnyTrust .

How can I check the status of my cross chain message?

You can check the status ofany Arbitrum cross chain message athttps://retryable-dashboard.arbitrum.io/ (you will also be able to execute the cross chain message there, if applicable).

You'll need the transaction hash of the "initiating transaction": the L1 transaction hash for an L1-to-L2 message (e.g., a deposit), or the L2 transaction hash for an L2-to-L1 message (e.g., a withdrawal).

If you cross-chain message was initiated fromhttps://bridge.arbitrum.io/ , you can also check its status / execute it at that site in the transaction history tab.

If there is a dispute, can my L2 transaction get reorged / thrown out / "yeeted"?

Nope; once an Arbitrum transaction is included on L1, there is no way it can be reorged (unless the L1 itself reorgs, of course). A "dispute" involves Validators disagreeing over execution, i.e., the outputted state of a chain. The inputs, however, can't be disputed; they are determined by the Inbox on L1. (SeeTransaction Lifecycle)

...okay but if there's a dispute, will my transaction get delayed?

The only thing that a dispute can add delay to is the confirmation of L2-to-L1 messages. All other transactions continue to be processed, even while a dispute is still undergoing. (Additionally: in practice, most L2-to-L1 messages represent withdrawals of fungible assets; these can be trustlessly completedeven during a dispute via trustless fast "liquidity exit" applications. SeeL2-to-L1 Messages).

Are "Sequencers" the same entities as "Validators"? Can a centralized Sequencer act maliciously (e.g., steal all my money)?

No and no!

An Arbitrum Chain's Sequencer(s) and Validators and completely distinct entities, with their own distinct roles.

TheSequencer is the entity granted specific privileges over ordering transactions; once the Sequencer commits to an ordering (by posting a batch on Ethereum), it has no say over what happens next (i.e., execution). A malicious/faulty Sequencer can do things like reordering transactions ortemporarily delaying a transaction's inclusion — things which could be, to be sure, annoying and bad — but can do nothing to compromise the chain's safety.

TheValidators are the ones responsible for the safety of the chain; i.e., making staked claims about the chain state, disputing each other, etc.

Currently, on Arbitrum One, the Sequencer is a centralized entity maintained by Offchain Labs. Eventually, we expect the single Sequencer to be replaced by a distributed committee of Sequencers who come to consensus on transaction ordering. This upgrade will be an improvement; we don't want you to have to trust us not to reorder your transactions. However, it also isn'tstrictly necessary for Arbitrum One to achieve its most fundamental properties.

In other words:An Arbitrum Rollup chain with a centralized Sequencer could theoretically still be trustless!

Which is to say — the more important thing than decentralizing the Sequencer, i.e., the thing you ought to care more about — is decentralizing theValidators .

Arbitrum One's validator set is currently allowlisted; over time, we expectgovernance to expand the allowlist and eventually be removed entirely.

For more info see"State of Progressive Decentralization" .

Why was "one week" chosen for Arbitrum One's dispute window?

Generally, some amount of time is necessary for the Arbitrum validators to dispute an invalid assertion.

A week is expected to be more than enough time for validators to carry out an interactive dispute, assuming they don't encounter difficulty in getting their transactions included on L1. One week was chosen following the general consensus among the Ethereum research community — as well as other layer 2 projects — to provide enough time for the community to socially coordinate in the case of a coordinated Ethereum-staker censorship attack.

What's the state of Arbitrum One's decentralization?

See"State of Progressive Decentralization" , or check out the work of our friends atL2BEAT .

Are there any Fiat on-ramps that support Arbitrum?

Yes, you can find a list of Fiat on-ramps that support Arbitrumon our portal .

How many blocks are needed for a transaction to be confirmed/finalized in Arbitrum?

There are two levels of finality in atransaction lifecycle :

- Soft finality: once the Sequencer receives and processes a transaction, it emits a receipt through the Sequencer's feed. At this point, if the Sequencer is trusted, the transaction will not be reordered and the state of the chain after processing the transaction can be determined.
- Hard finality: at this stage, assuming there's at least one well-behaved active Arbitrum validator, the client can treat their transaction's finality as equivalent to an ordinary Ethereum transaction.

Where can I find stats for Arbitrum?

Although we currently don't maintain any stats dashboard for Arbitrum, you can find manycommunity created dashboards in Dune.

Will transactions with a higher "gas price bid" be confirmed first?

There is no notion of mempool on Arbitrum, transactions are processed on a first come first served basis by the Sequencer. Thus, the gas price bid parameter does not influence the order in which a transaction is processed.

Where can I find a list of the current validators of the Arbitrum chains?

Validation on both Arbitrum One and Arbitrum Nova is currently allow-listed to a committee of public entities. You can see the list of validatorshere . Governance currently has the power to change this status.

Where can I find the current Data Availability Committee members?

The Arbitrum Nova chain has a 7-party DAC, whose members can be seenhere . Governance has the ability to remove or add members to the committee.

Can I withdraw my funds from Arbitrum back to Ethereum without going through the Sequencer? What about funds that are in a contract?

Yes, it is possible to permissionlessly send a message from Ethereum to be executed on Arbitrum, while bypassing the Sequencer. You can do this by using the DelayedInbox contract and force-including the message after a certain amount of time has passed (currently ~24 hours). You can find more information about this behaviorhere .

Keep in mind that you can execute any message in this way, be it a withdrawal of funds back to Ethereum, or a call to a contract.

You can also find an example of force-inclusion inthis tutorial .

Are there any plans to reduce the time a transaction needs to wait before being able to be force-included from Ethereum into the Arbitrum chain, bypassing the sequencer? (Currently 24 hours)

The mechanism that allows force-including transactions from Ethereum (bypassing the sequencer) is intended to be used in very rare cases, especially when it is expected that the sequencer will not be operational again, so that users have a way of interacting with Arbitrum in a trustless way.

When using this mechanism, if the sequencer is down for longer than the time window for force-including transactions from Ethereum, the moment it is online again, it can lead to a reorganization of blocks in Arbitrum (it would have received transactions timestamped before the force-included one).

24 hours was chosen because it provides a comfortable period of time for the team running the sequencer infrastructure to fix any bugs that may cause the sequencer to not work. While there aren't any active initiatives to lower that time, the decision ultimately falls in the hands of the Arbitrum DAO, who has discussed the topic in their governance forum (see here for more information).

In any case, we could also analyze why would someone use this mechanism having an honest and functional sequencer. For instance, if the reason is a distrust of the sequencer, a centralised agent as of now, one potential solution could be todecentralize the sequencer instead of reducing the force-inclusion delay time.

What is the difference between an L2 block and a RBlock?

An L2 block is very similar to the concept of an L1 block. These blocks are generated by validator nodes of Arbitrum by executing the state transition function on sequenced transactions. The structure of an L2 block is similar to that of an Ethereum block, with a few differences that you cansee here .

On the other hand, an RBlock is a distinctive block that is transmitted back to L1 to serve as a fingerprint of the most recent state of the Arbitrum chain. It comprises an assertion of the present state root of the Arbitrum chain and other essential information pertaining to withdrawals and challenges. The structure of RBlocks can be viewedhere .

These RBlocks are also generated by validators, but they are appended to the L1 chain. Other validators canchallenge them during a specific time frame of approximately one week if they discover that the current state hash of the chain varies from the one that was initially claimed. Once the challenge period elapses, the RBlock is confirmed on L1.

Why do Arbitrum chains enforce a speed limit? Isn't it better that the speed grows without limits?

The transaction lifecycle sets a limit that we have to take into account: validators have to execute each transaction, get the status of the chain, and post an assertion to Ethereum every certain amount of time. If the speed of the chain increases too much, there is a risk that validators won't have enough computation power to process all transactions in a timely manner, and will fall behind on validating them, which would cause the chain to delay confirmations of its state. Edit this page Last updatedonMar 7, 2024 Previous WAVM Modules Next Arbitrum glossary