Differences between Arbitrum and Ethereum: Overview

Arbitrum is designed to be as compatible and consistent with Ethereum as possible, from its high-level RPCs to its low-level bytecode and everything in between.Decentralized app (dApp) developers with experience building on Ethereum will likely find that little-to-no new specific knowledge is required to build on Arbitrum.

This section describes the differences, perks, and gotchas that devs are advised to be aware of when working with Arbitrum. This first page serves as an overview of where you might these differences, with links to the relevant pages when needed.

Block numbers and time

Time in L2s is tricky. The timing assumptions one is used to making about Ethereum blocks don't exactly carry over into the timing of Arbitrum blocks. SeeBlock numbers and time for details about how block numbers and time are handled in Arbitrum.

RPC methods

Although the majority of RPC methods follow the same behavior than in Ethereum, some methods might produce a different result, or add more information, when used on an Arbitrum chain. You can find more information about these differences inRPC methods .

Solidity support

You can deploy Solidity contracts onto Arbitrum just like you do Ethereum. There are only a few minor differences in behavior. Find more information about it inSolidity support .

Fees

The fees an Arbitrum transaction pays for execution essentially work identically to gas fees on Ethereum. Arbitrum transactions must also, however, pay a fee component to cover the cost of posting their calldata to the parent chain (for example, calldata on Arbitrum One, an L2, is posted to Ethereum, an L1). Find more information about the two components of gas fees inGas and fees andL1 pricing .

Cross-chain messaging

Arbitrum chains support arbitrary message passing from a parent chain (for example, a Layer 1 (L1) like Ethereum) to a child chain (for example, a Layer 2 (L2) like Arbitrum One or Arbitrum Nova). These are commonly known as "L1 to L2 messages". Developers using this functionality should familiarize themselves with how they work. Find more information about it inL1 to L2 messaging .

Similarly, Arbitrum chains can also send messages to the parent chain. Find more information about them inL2 to L1 messaging and the outbox .

Precompiles

Besides supporting all precompiles available in Ethereum, Arbitrum provides L2-specific precompiles with methods smart contracts can call the same way they can solidity functions. You can find a full reference of them inPrecompiles .

NodeInterface

The Arbitrum Nitro software includes a specialNodeInterface contract available at address0xc8 that is only accessible via RPCs (it's not actually deployed on-chain, and thus can't be called by smart contracts). Find more information about this interface inNodeInterface . Edit this page Last updatedonMar 22, 2024 Previous Cross-chain messaging overview Next Block numbers and time