Orbit chain ownership

Achain owner of anOrbit chain is an entity that can carry out critical upgrades to the chain's core protocol; this includes upgrading protocol contracts, setting core system parameters, and adding & removing other chain owners.

An Orbit chain's initial chain owner is set by the chain's creator when the chain is deployed.

The chain-ownership architecture is designed to give Orbit chain creators flexibility in deciding how upgrades to their chain occur.

PUBLIC PREVIEW, MAINNET READY Orbit chains are nowMainnet ready ! Note that Orbit is still apublic preview capability - the Orbit product and its supporting documentation may change significantly as we capture feedback from readers like you.

To provide feedback, click theRequest an update button at the top of this documentjoin the Arbitrum Discord , or reach out to our team directly by completingthis form .

Architecture

Chain ownership affordance is handled viaUpgrade Executor contracts.

Each Orbit chain is deployed with two Upgrade Executors — one on the Orbit chain itself, and one on itsparent chain. At deployment, the chain's critical affordances are given to the Upgrade Executor contracts.

Some examples:

- The parent chain's core protocol contracts are upgradeable proxies that are controlled by a proxy admin; the proxy admin is owned by the Upgrade Executor on the parent chain.
- The core Rollup contract's admin role is given to the Upgrade Executor on the parent chain.
- The affordance to call setters on the ArbOwner procompile — which allows for setting system gas parameters and scheduling ArbOS upgrades (among other things) — is given to the Upgrade Executor on the Orbit chain.

Calls to an Upgrade Executor can only be made by chain owners; e.g., entities granted theEXECUTOR_ROLE affordance on the Upgrade Executor. Upgrade executors also have theADMIN_ROLE affordance granted to themselves, which lets chain owners add or remove chain owners.

With this architecture, the Upgrade Executor represents a single source of truth for affordances over critical upgradability of the chain.

Upgrades

Upgrades occur via a chain owner initiating a call to an Upgrade Executor, which in turns calls some chain-owned contract.

Chain owners can either callUpgradeExecutor.executeCall , which will in turn call the target contract directly, orUpgradeExecutor.execute , which will delegate-call to an "action contract" and use its code to call the target contract.

Ownership flexibility

A chain owner is simply an address; it is set by the Orbit chain's deployer and can represent any sort of governance scheme. I.e., it could be an EOA (as is set via theOrbit Quickstart), a Multisig, a governance token system, etc.

The Arbitrum DAO governed chains, while not Orbit chains themselves, use a similar architecture and upgrade pattern as Orbit chains, with both a governance token and a Multisig (aka, the "Security Council") as chain owners. For more info and best practices on action contracts, see"DAO Governance Action Contracts" .

(NOTE: The DAO Governed chains' Upgrade Executor contracts don't have the.executeCall method; only the.execute method) Edit this page Last updatedonMar 7, 2024 Previous Keyset generation (AnyTrust chains) Next Orbit FAQ