Documentation Index

Fetch the complete documentation index at: /llms.txt

Use this file to discover all available pages before exploring further.

Skip to main content
Arcium Docs home page



Arcium Docs



Search... ⌘K
Ask Assistant

- arcium.com
- X @arcium

Search...
Navigation
INTRODUCTION
Basic concepts

Documentation

Developers

INTRODUCTION

- Overview of the Arcium Network
- Key features and use cases
- Basic concepts

GETTING STARTED

CRYP2NOVA

Archived copy · cryp2nova.com

- How to use this documentation
- Architecture overview
- Network stakeholders

MPC EXECUTION ENVIRONMENTS (MXEs)

- Overview
- MPC protocols
- MXE encryption

CLUSTERS

- Overview
- Cluster migration
- Sybil resistance
- Incentivization
- Permissioned clusters

ARX NODES

- Overview
- Configuration and security
- Performance and incentives

COMPUTATIONS

- Computation tasks
- Defining and commissioning computations
- Lifecycle of an Arcium computation
- Pricing and incentives
- Queue ordering and fault handling

SOLANA INTEGRATION AND MULTICHAIN COORDINATION

- Solana integration: orchestration and execution
- Multichain expansion

STAKING

- Overview

On this page

CRYP2NOVA

Archived copy · cryp2nova.com

- Understanding the encrypted supercomputer
- Arcium and Arx nodes
- Multi-party computation (MPC)
- Trustless execution
- Clusters and MXEs (MPC eXecution Environments)
- Handle faults
- Epochs

INTRODUCTION

Basic concepts

Copy pageCopy page

Core concepts: MPC, MXEs, Arx nodes, Clusters, and secret sharing

Copy pageCopy page

On this page, we'll provide an overview of concepts core to Arcium, including:

1. **MPC eXecution Environments (MXEs):** Accounts that group a program, computation definitions, key material, and an execution Cluster.
2. **arxOS:** The encrypted, distributed operating system powering Arcium, which lets anyone run encrypted computations. It relies upon other concepts central to Arcium and covered below, including **Arx nodes** and **Clusters.**
3. **Arcis:** The Rust framework and compiler for writing encrypted computations.



Understanding the encrypted supercomputer

Arcium describes the network as an "encrypted supercomputer." Arx nodes act as processors, arxOS coordinates their offchain execution, MXEs group application-specific computations, and Arcis provides the developer framework.

Arcium and Arx nodes

The Arcium Network consists of decentralized **Arx nodes** that perform computations on encrypted data. Derived from the Latin word "arx," meaning fortress, each Arx node represents a secure point in the network. However, the true power of Arx nodes comes from their collaboration within the decentralized network.

Archived copy · cryp2nova.com

- **Staking:** Arx nodes stake to activate, join Clusters, receive rewards, and participate in leader selection.

Multi-party computation (MPC)

Multi-Party Computation (MPC) is the cryptographic backbone of Arcium. It allows multiple parties to jointly compute functions while keeping their inputs private, preserving data confidentiality throughout the process.

- **Secret Sharing** is a key cryptographic method within MPC, used to split data into fragments distributed across Arx nodes. No individual node has access to the full data.

Trustless execution

Cerberus distributes trust across the Cluster. Under the assumption that at least one participating Arx node is honest, it preserves privacy and aborts on detected protocol faults rather than producing a corrupted result.

Clusters and MXEs (MPC eXecution Environments)

Clusters are groups of Arx nodes that execute MPC tasks. An **MPC eXecution Environment (MXE)** binds application-specific computation definitions to one active Cluster. Its authority can explicitly migrate it to another Cluster through key recovery.

Handle faults

Cerberus uses a detect-and-abort model. A protocol fault can stop a computation, so applications handle failure, retry, queue expiry, and migration in their lifecycle logic.

Epochs

Timing in the Arcium Network is organized into fixed-duration **Epochs**, which serve as the framework for scheduling computations, distributing rewards, and managing token lock-ups. This consistent timing ensures the efficient and fair execution of tasks across the network.

CRYP2NOVA

Archived copy · cryp2nova.com

Archived copy · cryp2nova.com

Key features and use cases
How to use this documentation

Archived copy · cryp2nova.com

Archived copy · cryp2nova.com

Ask a question...

⌘I

Powered by



This documentation is built and hosted on Mintlify, a developer documentation platform

CRYP2NOVA

Archived copy · cryp2nova.com