Lesson 9 - MINA / Aleo

Week 3

Lesson 9 : Mina / Aleo

Lesson 10 : zkEVM Solutions

Lesson 11 :Risc Zero / Circom

Lesson 12 :zkSNARKS Theory

Today's topics

- zk Oracles
- Off chain storage
- O1js - Advanced Concepts
- Protokit
- Decentralised Exchanges on Mina
- Aleo introduction
- Aleo development

> # zkOracles
>
> **See video**
>
> **See tutorial**
>
> **The idea behind an oracle is to allow a zkApp to get data from any HTTPS data source.**

This is a similar idea to that of DECO or Pragma Oracle

The tutorial creates an oracle that

- Fetches data from the desired source
- Signs it using a Mina-compatible private key
- Returns the data, signature, and public key associated with the private key
- Allows the signature to be verified by the zkApp

Off chain storage

There are a number of community projects in this area looking to provide simple solutions using existing decentralised storage providers such as IPFS and Filecoin.

Tutorial 6 walks through an example that uses a merkle tree to store data
For validation the merkle root is stored on chain, the tree is stored off chain.

Advanced Concepts

Recursion

See Tutorial Documentation

Recursion brings use cases such as

- high-throughput applications through rollups,
- Proofs of complex computations,
- Multi-party proofs.

We generally think of one prover and one verifier, multi party proofs allow multiple provers to recursively update a proof off chain and then send it to be verified on chain. We will look at the idea of collaborative SNARKS more generally in a later lesson.

Concurrency

One difficulty with off chain computation is synchronising state updates.

In Mina these are handled via Actions and a Reducer

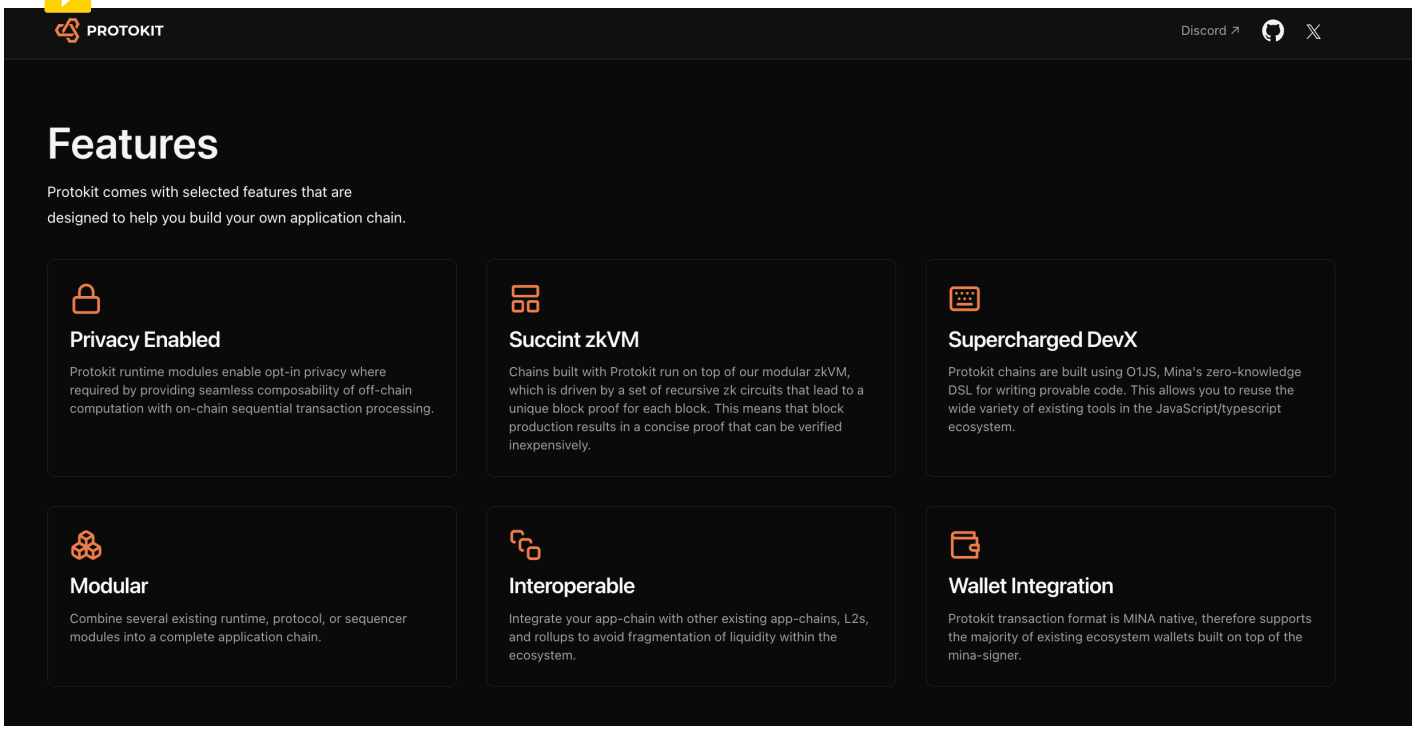
See Docs for an example.

Protokit

See Docs

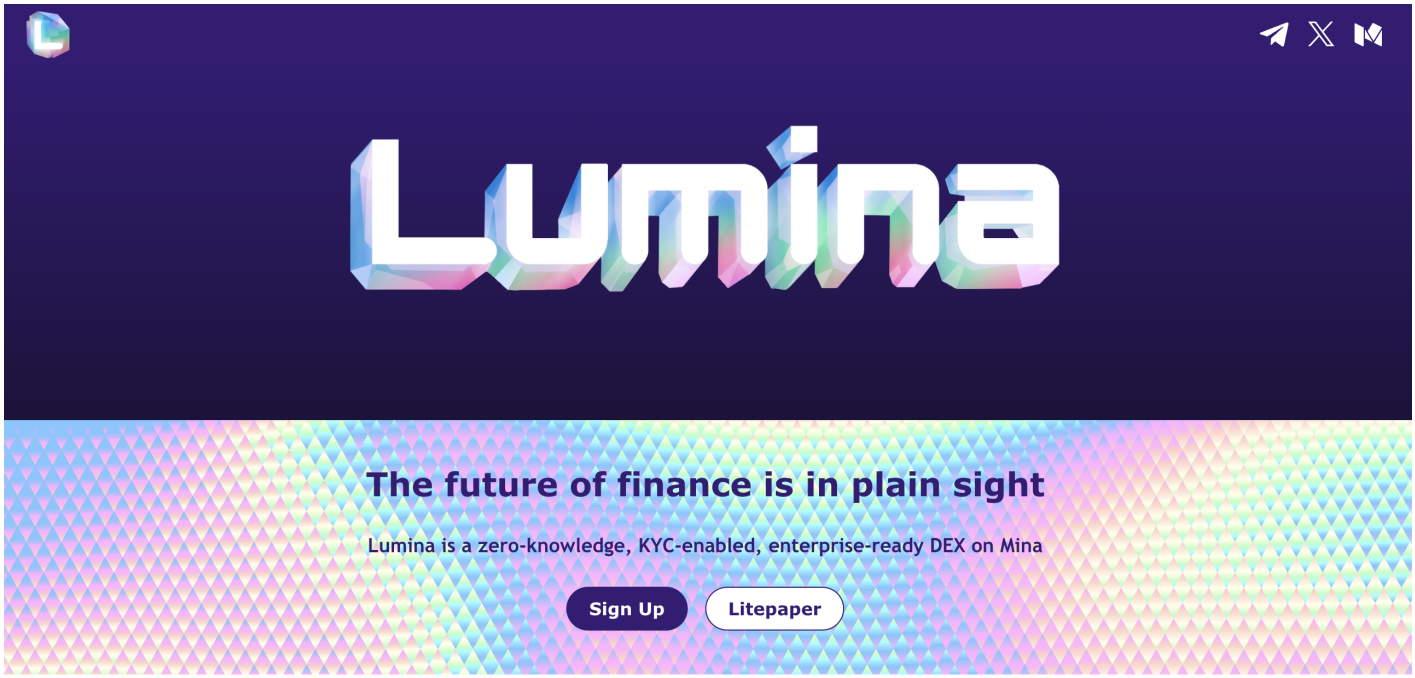
Protokit is a project that came out of a recent Mina builder program.

Protokit enables developers to build zero-knowledge, interoperable and privacy preserving application chains with a minimal learning curve.



Decentralised Exchanges on Mina

Lumina



See Site

See Litepaper

Kaupang Dex

This was built using protokit :

Kaupang Dex

Upcoming Mina Projects

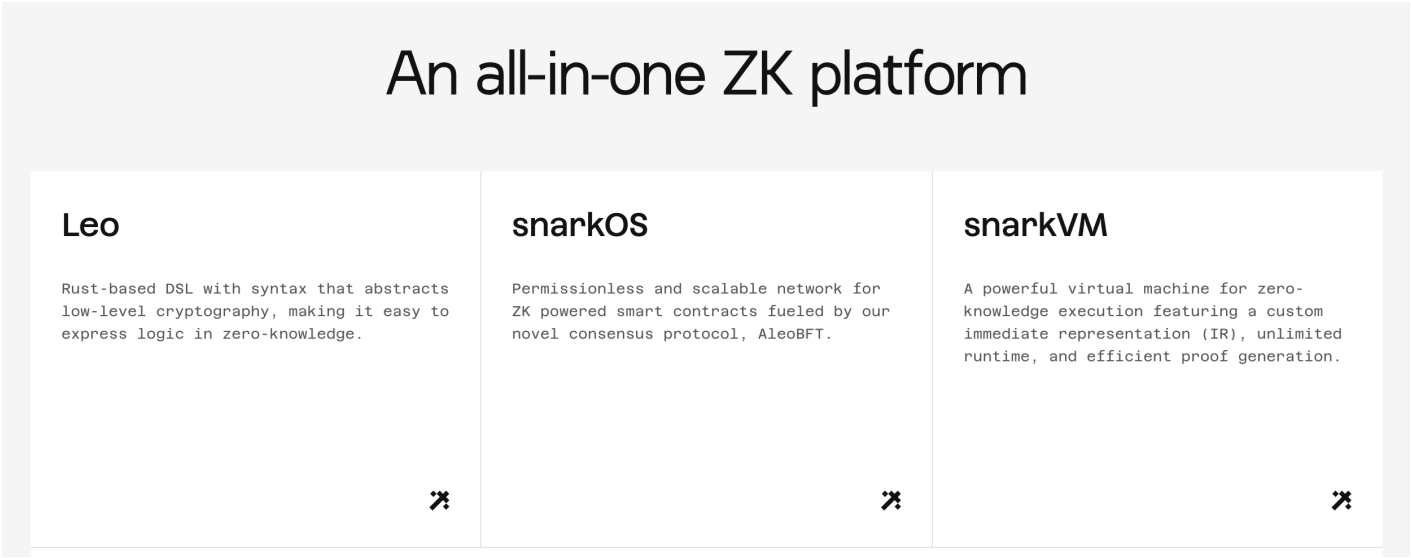
Collaboration with Etonec

A Mina Foundation grant will enable etonec to build a state-of-the-art Zero-Knowledge Proof-ID (zkp-ID) solution with KYC and AML functionality within Lumina DEX, a decentralized exchange built on the Mina Protocol

Community projects / libraries

- **o1js-elgamal** A partially homomorphic encryption library for o1js based on Elgamal encryption: GitHub and npm
- **o1js-pack** A library for o1js that allows a zkApp developer to pack extra data into a single Field. GitHub and npm

Aleo Introduction



Consensus Mechanism

Aleo uses a hybrid Pos / Pow / BFT mechanism called AleoBFT

This article provides an overview.

General concepts

SnarkVM

This is the virtual machine responsible for running programs.

Computation is done off chain and verified on chain.

Transactions

Transactions are used to hold transitions which will update the state of the ledger. They are signed by an

owner and require a fee.

Records

These are the data structure responsible for holding state, they are created and consumed by transitions. In concept they have a similarity with Notes in ZCash and Aztec. The record has an owner and can be encrypted for privacy.

Programs

Programs hold Aleo instructions (an IR).
Program inputs can be private, the state that the program operates on is supplied in records.
The output of the program also includes a proof that the computation proceeded correctly.

Development on Aleo

See Docs

Leo language

See Docs

Leo is a functional, statically-typed programming language built for writing private applications.
It is similar to Rust in syntax, and like other zkp DSLs it has a restricted set of datatypes, but supports

- Integers of different sizes
- Booleans
- Addresses
- Field, Group and Scalar elements
- Signatures

Leo code compiles to Aleo instructions which are an intermediate representation, which then compile to bytecode.
The bytecode consists of AVM codes, which run on the Aleo virtual machine.

Leo installation

See Instructions

Leo Playground

Available at playground

The Leo wallet

See Wallet

See Adapter Docs