CARDANO , BLOCKCHAIN AND PLUTUS

- What is Cardano?
- What is a blockchain?
- What are DApps?
- What is ADA?
- Cardano and native tokens
- What is Plutus?
- Plutus vs Glow vs Marlow
- Writing smart contracts using Plutus
- What is the Plutus Platform?
- What are Plutus language versions?

The Cardano blockchain is one of the fastest-rising and most widely used blockchains in 2022. The Cardano blockchain is a third-generation, proof-of-stake blockchain platform and home to the ADA cryptocurrency.

What is Cardano?

The Cardano blockchain is a new blockchain based on the Shelley consensus algorithm. It's designed to be a secure, scalable, and easy-to-use platform for building and running decentralized applications.

What is a blockchain?

A blockchain is a distributed database of records linked together by a series of cryptographic proofs. The records are called blocks; the blockchain is a collection of blocks.

What are DApps?

A decentralized, distributed app is an application that is not controlled by a single entity. Otherwise, a DApp is just like any conventional app. A DApp can be a web app, command-line app, desktop app, or mobile.

The difference between a DApp and a conventional app is that the conventional app is controlled by a single entity. A DApp has a blockchain as its backend.

For example, a React to-do app might have its data (the to-dos) stored in an Ethereum blockchain. Or, you could build a voting system in Angular and have its data stored in the Cardano blockchain.

What is ADA?

ADA is the native cryptocurrency of the Cardano blockchain. You can use and transfer ADA on the Cardano blockchain.

CARDANO , BLOCKCHAIN AND PLUTUS

ADA can be used as both a cryptocurrency and a token. ADA being used as Cardano's native currency means that it is:

- Accepted as fee-payment
- Accepted to make deposits
- The only currency in which rewards are distributed

Just like ETH is divided into denominations (e.g., Wei), ADA has denominations as well. Lovelace is the smallest unit of ADA; 1 ADA is equal to 10^(-18) Lovelace, which can be divided into a total of 1,000,000 Lovelaces.

Cardano and native tokens

The Cardano blockchain makes it possible to build native tokens. These tokens can be used to represent a value and to trade on the Cardano blockchain.

Tokens can be fungible or non-fungible. Fungible tokens represent a value, whereas non-fungible tokens represent real world items (e.g., a car or a piece of artwork).

In the following sections, we'll walk through how to write and create smart contracts on the Cardano blockchain using the Plutus programming language.

What is Plutus?

Plutus is a programming language used for writing smart contracts on the Cardano blockchain. Other programming languages you can use for the Cardano blockchain include Glow lang, Marlowe, etc., but Plutus is the fastest among them.

Plutus is based on Haskell, so using it requires prior knowledge of Haskell programming.

Plutus vs Glow vs Marlow

Marlow and **Glow** from my understanding are both domain-specific languages that can be used to build smart contracts and Dapps, whereas **Plutus** is not a domain-specific language but can still be used to build smart contracts.

Marlow lets people create financial smart contracts on the blockchain. **Glow** allow to build smart contracts with a domain-specific language without the financial parameter.

Plutus is specific to Cardano while **Glow** works across multiple blockchains. Saying this, **Glow** is a 'domain specific language' which is only meant to be used for programming DApps while **Plutus** is a Haskell library.

Marlowe may be useful for a person writing a one-off contract between two people/organisations, while **Glow** may be useful for a developer who wants to write a dapp used by many people.

CARDANO , BLOCKCHAIN AND PLUTUS

Marlowe vs Plutus vs Glow vs Solidity

- Marlowe is a DSL for financial contracts
- Plutus is a turning complete programming language
- Glow is a Cordova alike DSL that will compile to many smart contract platforms
- Solidity is a Flow/TypeScript alike language from Ethereum Alliance (also turning complete)

Writing smart contracts using Plutus

Let's walk through how to write smart contracts using the Plutus programming language. These apps can run off-chain and manage active contract instances.

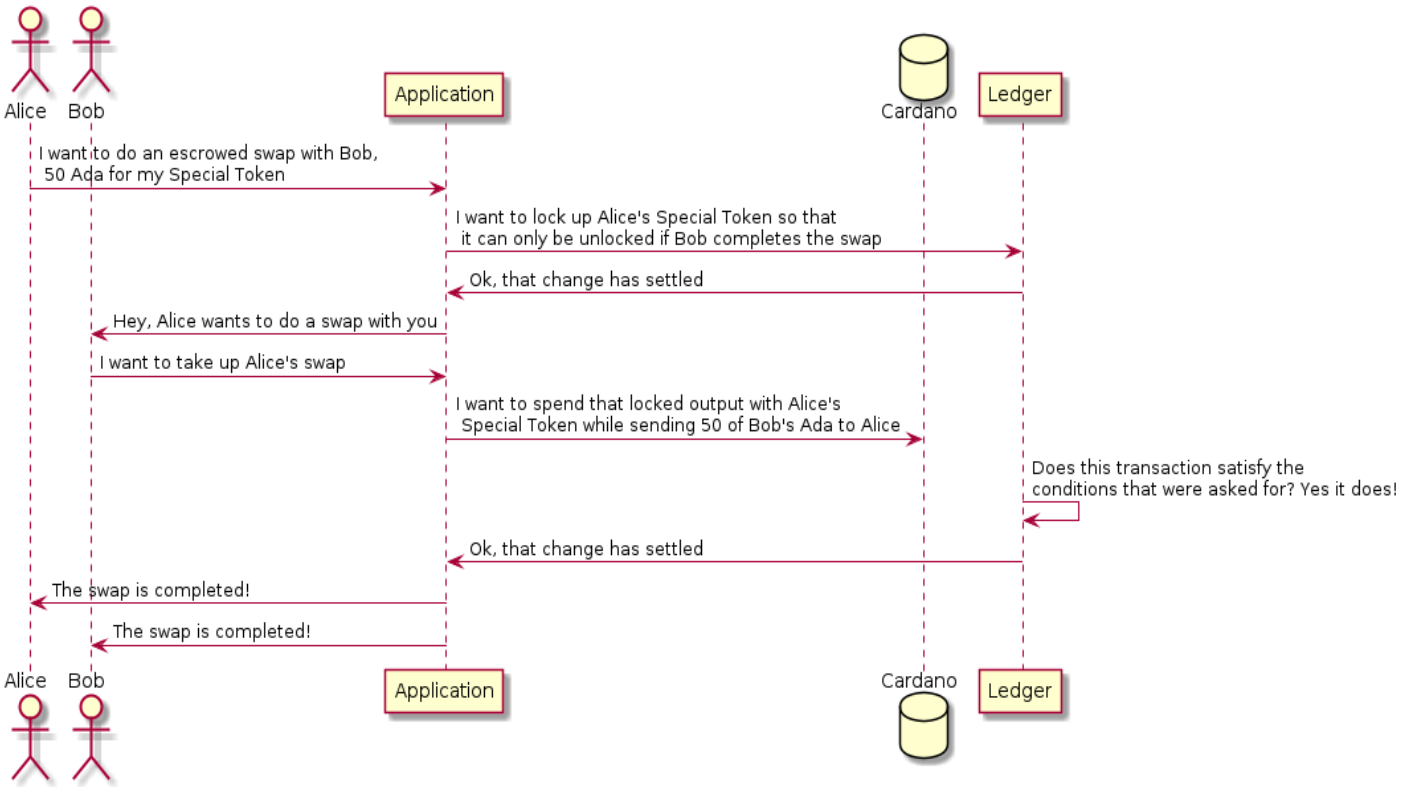
According to Cardano Testnets, Plutus contracts consist of parts that run on the blockchain (on-chain code) and parts that run on a user's machine (off-chain or client code). We'll write our smart contracts in a Plutus online editor and simulator.

Plutus is **the native smart contract language for Cardano**. It is a Turing-complete language written in Haskell, and Plutus smart contracts are effectively Haskell programs. By using Plutus, you can be confident in the correct execution of your smart contracts.

A Plutus smart contract is a Haskell code that runs in a user's wallet and sends code to the blockchain to be run by the nodes in the blockchain. The smart contracts are run on the blockchain, not on the user's machine.

Plutus smart contracts have definitions that are used to define the smart contract and its state. Inside these definitions are endpoints, which define the smart contract's behaviour. These endpoints are functions that are executed by the wallet. They are known as off-chain functions and are used to build transactions and send them to the blockchain

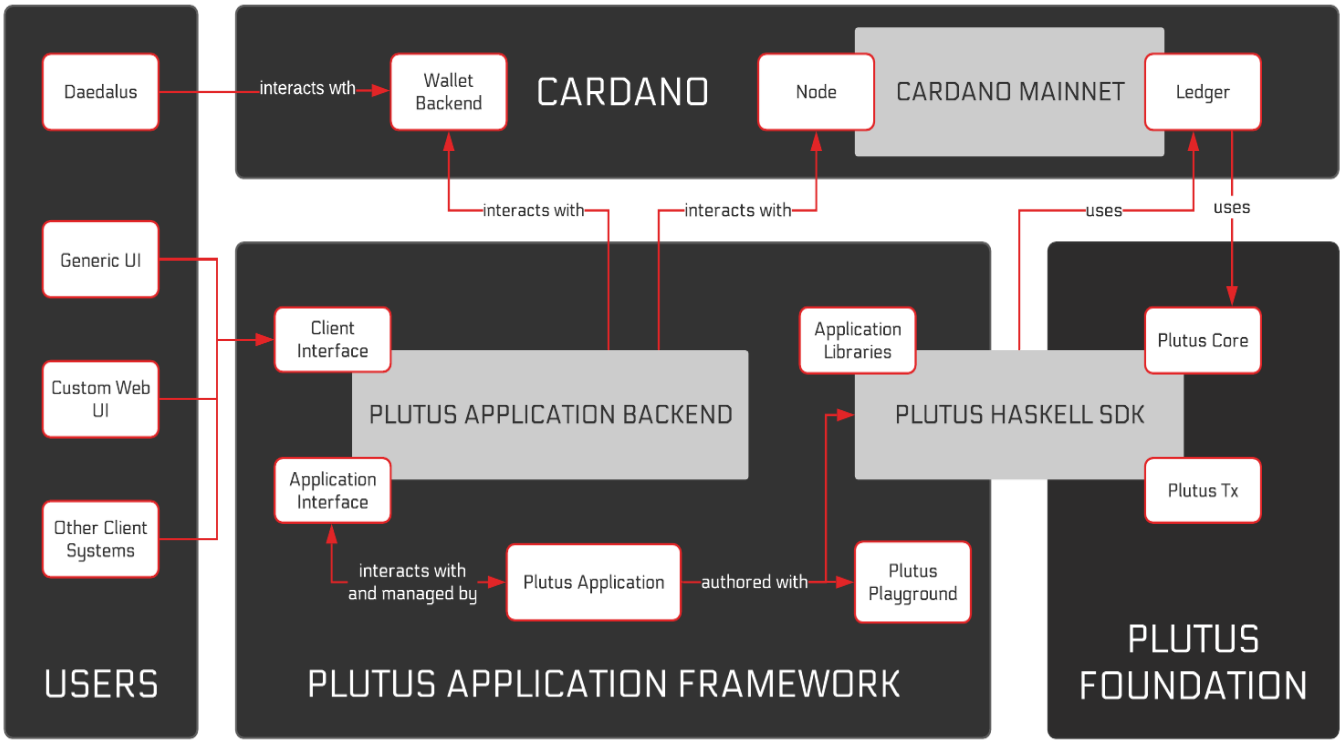
CARDANO , BLOCKCHAIN AND PLUTUS



What is the Plutus Platform?

The Plutus Platform is a platform for writing *applications* that interact with a *distributed ledger* featuring *scripting* capabilities, in particular the Cardano blockchain.

CARDANO , BLOCKCHAIN AND PLUTUS



Conceptually, the Platform breaks down based on which part of the system we're interested in:

- Plutus Foundation: support for writing the trusted kernel of code, and executing it on the chain.
- The Plutus Application Framework: support for writing applications ("Plutus Applications") in a particular style.

In order for an application to run its trusted kernel of logic as a script on a ledger, the ledger needs a way of specifying and executing scripts. Scripts are simply programs, so this means we need a *programming language*.

Plutus Core

In the Plutus Platform, this language is *Plutus Core*. Plutus Core is a variant of the lambda calculus, a well-studied formalism for computing.

> ***Note:*** ***Plutus Core is our "assembly language". Trust me, you don't want to see any! Dealing with that is the compiler's job.***

CARDANO , BLOCKCHAIN AND PLUTUS

> Plutus Core is designed for simplicity, determinism, and to allow careful cost control of program execution. Using the lambda calculus makes it an easy compilation target for functional programming languages, and allows us to have a simple, formally verified evaluator.

Plutus Tx

Writing Plutus Core by hand is not a job for a human! It is designed to be written by a compiler, and the Platform provides a compiler from a subset of Haskell to Plutus Core. This allows you to seamlessly write applications in Haskell, while compiling part of the code to on-chain Plutus Core, and part into an off-chain application.

Supporting “mixed” code in this way enables libraries written with the Plutus Haskell SDK to share logic and datatypes across both parts of the application, reducing the risk of errors significantly.