Introduction to Plutus

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.

Plutus smart 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).

Off-chain code can be written using the Plutus Application Framework (PAF), and this code is then compiled by the GHC (Glasgow Haskell Compiler), whereas on-chain code is compiled by the Plutus compiler into Plutus Core.

Plutus Core is the scripting language used by Cardano to implement the EUTXO model. It is a simple, functional language similar to Haskell, and a large subset of Haskell can be used to write Plutus Core scripts.

The Plutus Application Framework (PAF) provides easy access to services that are commonly used by Plutus applications. Applications deployed using the framework’s libraries can be run on the Plutus application backend, which provides runtime support for access to the blockchain and other concerns such as persistence, logging, and monitoring.

The Plutus Application Backend (PAB) is currently being developed and will execute the off-chain component of Plutus applications. It will manage application requests to the wallet backend and node, store the application state, and offer an HTTP API for managing application instances.

Plutus provides considerable security advantages. It delivers an easier, more robust way to show that your smart contracts are correct and will not encounter the problems found in previous smart contract language design.

Additionally, in contrast to Ethereum, Plutus supports user-defined tokens (both fungible and non-fungible) natively which does not require an accompanying smart contract.