Smart Contracts

Smart contracts in Aztec are privacy-first, and can include both public and private elements. They are written in Noir framework called Aztec.nr, and allow high-level programs to be converted into ZK circuits.

On this page, you'll learn how Aztec executes smart contracts for privacy and efficiency:

- Role and structure of smart contracts within Aztec
- Intro into Noir programming language and how it converts to circuits
- The Aztec Kernel
- Transaction flow and confidentiality

Defining Aztec smart contracts

A "smart contract" is defined as a set of public and private functions written as Noir circuits. These functions operate on public and private state stored by a contract. Each function is represented as a ZK SNARK verification key, where the contract is uniquely described by the set of its verification keys, and stored in the Aztec Contracts tree.

Noir is a programming language designed for converting high-level programs into ZK circuits. Based on Rust, the goal is to present an idiomatic way of writing private smart contracts that is familiar to Ethereum developers. Noir is under active development adding features such as contracts, functions and storage variables.

The end goal is a language that is intuitive to use for developers with no cryptographic knowledge and empowers developers to easily write programable private smart contracts.

There are no plans for EVM compatibility or to support Solidity in Aztec. The privacy-first nature of Aztec is fundamentally incompatible with the EVM architecture and Solidity's semantics. In addition, the heavy use of client-side proof construction makes this impractical.

Further reading

Read more about writing Aztec contracts here . Edit this page

Previous Transactions Next Contract Creation