Framework overview

A description of CosmWasm and the framework around it as implemented on Secret Network Smart contracts on Secret Network follow the widely usedCosmWasm framework. The framework is built as a module for the Cosmos SDK and allows developers to build smart contracts withRust which compile toWebAssmebly (Wasm). This type of efficient and fast binary, combined with the power of a low-level typed coding language, makes CosmWasm a secure and efficient way for developers to write smart contracts. Contracts on Secret have a public binary just like any other smart contract network, onlythe contract input, output and state is encrypted. The contracts are executed in the Wasm VM as deployed via the Wasmd Cosmos SDK module on the Secret blockchain.

A diagram of the CosmWasm framework - (Image created by PolymerDao and OpenIBC) Secret Contracts slightly differ from any standard CosmWasm implementation as they handle encrypted data, secure enclaves, and the key design around that. Secret currently emulates CosmWasm v1.1.9 and is completely compatible over IBC with other V1.0+ Cosmos Smart contract networks. To learn more about IBC compatible contracts you can readthis section .

Want to learn about how Secret handles the Contract-state encryption and other details that relate to how CosmWasm works on secret? -- Read the implementation documentationhere!

The components of a CosmWasm contract

A Secret contract has 4 different components as listed below each serving a different function. A deeper overview of the implementation of these 4 components/files is given in theContract-components section.

- Instantiate
-
  - the logic that is ran once on initialization of the contract. This will usually be the initial conditions and configuration of the contract.
- Execute
-
  - this is where thetransactional
- logic resides. These are functions that modify contract data
- Query
-
  - these are functions that are ran as read-only and cannot modify contract data
- State/storage
-
  - the long-term storage of the contract. By default, data used by contracts will be lost between transactions.
-

Each contract will contain these basic building blocks, with their logic being built around how they work. You can read more about the interplay of Entry-points as depicted in the graphic below inthis starter guide.

An overview of the interplay of the CosmWasm components with the blockchain data layer. - (Image created by PolymerDao and OpenIBC) Want to get started with CosmWasm development? -Get started with our 4 part crash course. Dont have much time? - Try our guide on the millionaires problem instead!- It teaches everything about the CosmWasm framework, message types and environment setup in a single page.

Last updated9 months ago On this page Was this helpful?Edit on GitHub Export as PDF