

Trusted & Untrusted Core

More info coming soon Now that we have a better understanding of how Secret is leveraging SGX, let's see how the TEE and enclaves works with the Trusted and Untrusted cores.

Untrusted Core

- Responsible for running the Cosmos SDK and Tendermint.
- Contains code for creating and managing the enclave (load and destroy).
- Can call the CosmWasm module and kick off Secret Contract execution within the enclave.
-

Trusted Core

- Responsible for executing Secret Contracts.
- Responsible for SGX-specific mechanisms: Remote Attestation and Sealing.
- Able to make read/write calls from the Tendermint state at any point during execution.
- The enclave only stores the seed. The enclave may also store the local node's key pair.
-

Note: During contract execution, only the state of the contract being executed can be changed. Other contracts can be queried (i.e. run code that can't change the state of another contract) synchronously, but calls to other contracts and requests for transactions can only be queued. Those operations will happen after the contract has finished running. This is intentional as it prevents a lot of bugs, like the re-entrancy bugs plaguing Ethereum.

Below is a diagram of how the Untrusted and Trusted behave on a User transaction and Secret Contract execution:

Trusted Execution Environments on Secret Network. \

Last updated 1 year ago On this page * [Untrusted Core](#) * [Trusted Core](#)

Was this helpful? [Edit on GitHub](#) [Export as PDF](#)