

Introduction

What you can do with CAPE and how it works Espresso Systems has developed the [Configurable Asset Privacy \(CAP\) protocol](#) , which enables digital assets to have customized privacy properties.

The first application of CAP is CAPE (Configurable Asset Privacy for Ethereum), a smart contract application currently implemented on Arbitrum's Goerli testnet.

With CAPE, asset creators can create brand new assets on CAPE (domestic CAPE assets) or sponsor wrappers to give existing ERC-20 assets privacy features.

Users can wrap CAPE assets into wrappers , locking their existing Ethereum tokens and minting them 1:1 within the CAPE system. CAPE assets can then be unwrapped back into Ethereum tokens with no more privacy guarantees.

Every asset within CAPE has a viewing policy associated with it that designates who can see select transaction data. Freezing and other types of policies are also possible.

In CAPE, users own asset records which specify the asset type, the amount, and the owner's address. The system publishes short, hidden representations of these records called commitments . A transaction is composed of:

- a list of nullifiers
- which mark the inputs as spent without revealing their content
- new output record commitments to be published, and
- a zero-knowledge proof that ensures the nullifiers and the new records are correctly computed, and also that the privacy policies for those records are satisfied.
-

Rather than submitting CAPE transactions directly to the CAPE Ethereum smart contract, users instead submit transactions to a relayer . The relayer then forwards CAPE transaction data to the CAPE smart contract to be validated. An observer of the system would not be able to see any of the details of the transaction unless they possessed a designated viewing key. Also, since the relayer submits CAPE transactions to the CAPE smart contract and pays the accompanying Ethereum gas fee, the Ethereum transaction fee is not linkable to the original user on the blockchain.

[Previous FAQ & troubleshooting](#) [Next Asset Creation + Definition](#) Last updated 1 year ago On this page