What a contract looks like

Example Aztec.nr Contract

In keeping with the origins of blockchain, here's an example of a simple private token contract. Everyone's balances are private.

easy_private_token_contract contract EasyPrivateToken

{ use

dep :: aztec :: prelude :: { AztecAddress ,

NoteHeader ,

Map } ; use

dep :: value_note :: { balance_utils ,

value_note :: ValueNote } ; use

dep :: easy_private_state :: EasyPrivateUint ;

struct

Storage

{ balances :

Map < AztecAddress ,

EasyPrivateUint

, }

/* ** initialize the contract's initial state variables./*

[aztec(private)]

[aztec(initializer)]

fn

constructor (initial_supply :

u64 , owner :

AztecAddress)

{ let balances = storage . balances ;

balances . at (owner) . add (initial_supply , owner) ; }

// Mints amount of tokens to owner.

[aztec(private)]

fn

mint (amount :

u64 , owner :

AztecAddress)

{ let balances = storage . balances ;

balances . at (owner) . add (amount , owner) ; }

// Transfers amount of tokens from sender to a recipient.

[aztec(private)]

fn

transfer (amount :

u64 , sender :

AztecAddress , recipient :

AztecAddress)

{ let balances = storage . balances ;

balances . at (sender) . sub (amount , sender) ; balances . at (recipient) . add (amount , recipient) ; }

// Helper function to get the balance of a user ("unconstrained" is a Noir alternative of Solidity's "view" function).
unconstrained fn

getBalance (owner :

AztecAddress)

->

pub

Field

{ let balances = storage . balances ;

// Return the sum of all notes in the set. balance_utils :: get_balance (balances . at (owner) . set) }Source code: noir-projects/noir-contracts/contracts/easy_private_token_contract/src/main.nr#L1-L47 The prelude consists of more commonly imported aztec types that are needed for development. Here is what the prelude includes:

prelude use

dep :: protocol_types :: { address :: { AztecAddress ,

EthAddress } ,

abis :: function_selector :: FunctionSelector } ; use

crate :: { state_vars :: { map :: Map ,

private_immutable :: PrivateImmutable ,

private_mutable :: PrivateMutable , public_immutable :: PublicImmutable ,

public_mutable :: PublicMutable ,

private_set :: PrivateSet , shared_immutable :: SharedImmutable } , log :: { emit_unencrypted_log , emit_encrypted_log } ,

context :: PrivateContext , note :: { note_header :: NoteHeader ,

note_interface :: NoteInterface ,

note_getter_options :: NoteGetterOptions , note_viewer_options :: NoteViewerOptions , utils :: compute_note_hash_and_nullifier as utils_compute_note_hash_and_nullifier } } ; Source code: noir-projects/aztec-nr/aztec/src/prelude.nr#L1-L16 Disclaimer Please note that any example contract set out herein is provided solely for informational purposes only and does not constitute any inducement to use or deploy. Any implementation of any such contract with an interface or any other infrastructure should be used in accordance with applicable laws and regulations. Edit this page