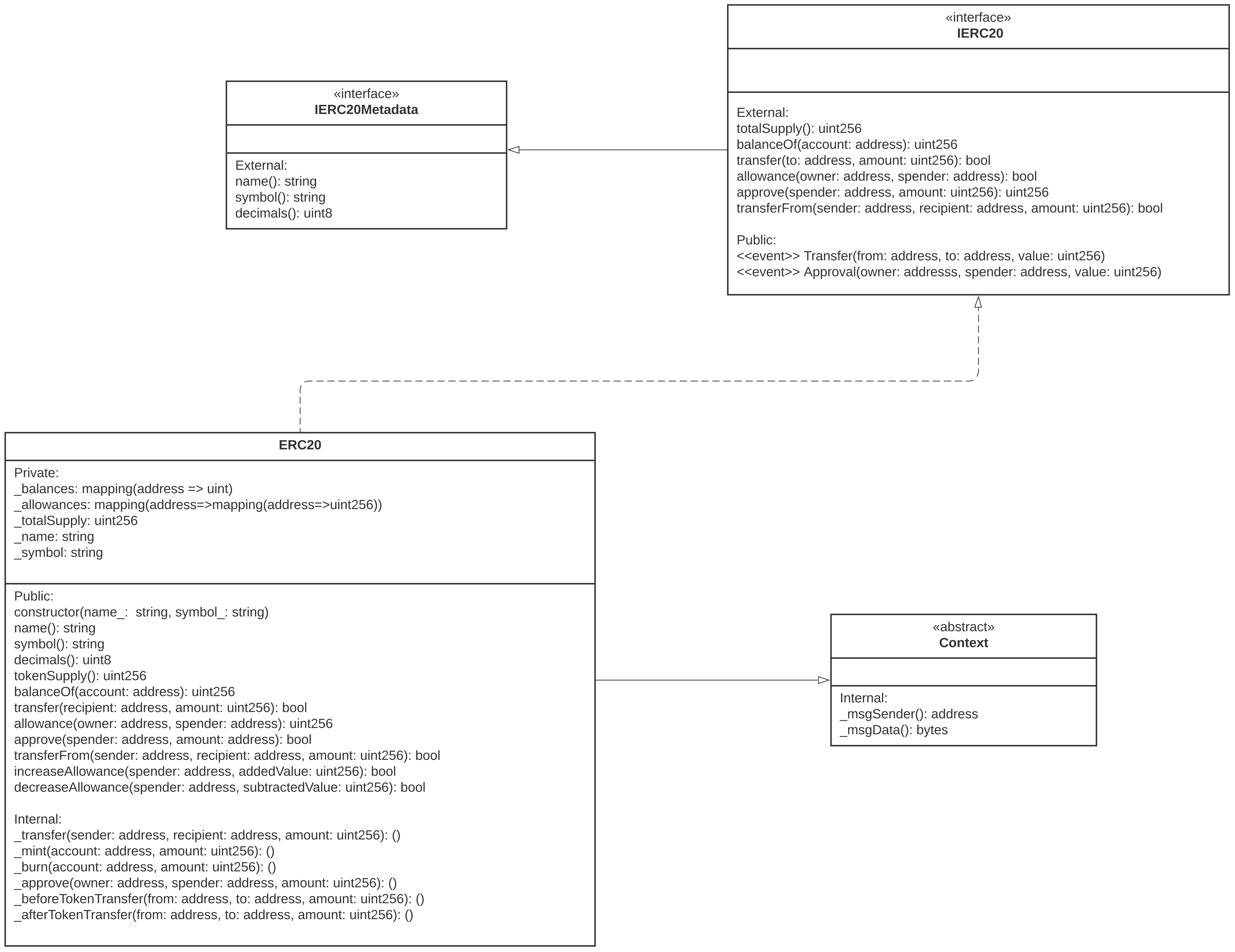
«interface»
IERC20
External:
totalSupply(): uint256
balanceOf(account: address): uint256
transfer(to: address, amount: uint256): bool
allowance(owner: address, spender: address): bool
approve(spender: address, amount: uint256): uint256
transferFrom(sender: address, recipient: address, amount: uint256): bool
Public:
<<event>> Transfer(from: address, to: address, value: uint256)
<<event>> Approval(owner: addresss, spender: address, value: uint256)
«interface»
IERC20Metadata
External:
name(): string
symbol(): string
decimals(): uint8
ERC20
Private:
_balances: mapping(address => uint)
_allowances: mapping(address=>mapping(address=>uint256))
_totalSupply: uint256
_name: string
_symbol: string
Public:
constructor(name_: string, symbol_: string)
name(): string
symbol(): string
decimals(): uint8
tokenSupply(): uint256
balanceOf(account: address): uint256
transfer(recipient: address, amount: uint256): bool
allowance(owner: address, spender: address): uint256
approve(spender: address, amount: address): bool
transferFrom(sender: address, recipient: address, amount: uint256): bool
increaseAllowance(spender: address, addedValue: uint256): bool
decreaseAllowance(spender: address, subtractedValue: uint256): bool
Internal:
_transfer(sender: address, recipient: address, amount: uint256): ()
_mint(account: address, amount: uint256): ()
_burn(account: address, amount: uint256): ()
_approve(owner: address, spender: address, amount: uint256): ()
_beforeTokenTransfer(from: address, to: address, amount: uint256): ()
_afterTokenTransfer(from: address, to: address, amount: uint256): ()
«abstract»
Context
Internal:
_msgSender(): address
_msgData(): bytes