LNP/BP Standards Association

RGB: v0.10 demo (part 4)

Dr Maxim Orlovsky

Chief engineering officer at **LNP/BP Standards Association**,

@lnp_bp, @dr_orlovsky

FBDE A843 3DDC 1E69 FA90 C35E FFC0 2509 47E5 C6F7

RGB v0.10 release progress

	Readiness	Released	Presentation
Consensus (Core lib)	✅	9 Feb	Part 1
Standard library	✅	9-10 March	Part 2
Wallet library	✅	22 March	Part 3
Command-line tool	✅	10 April	Part 4

```
$ rgb
```

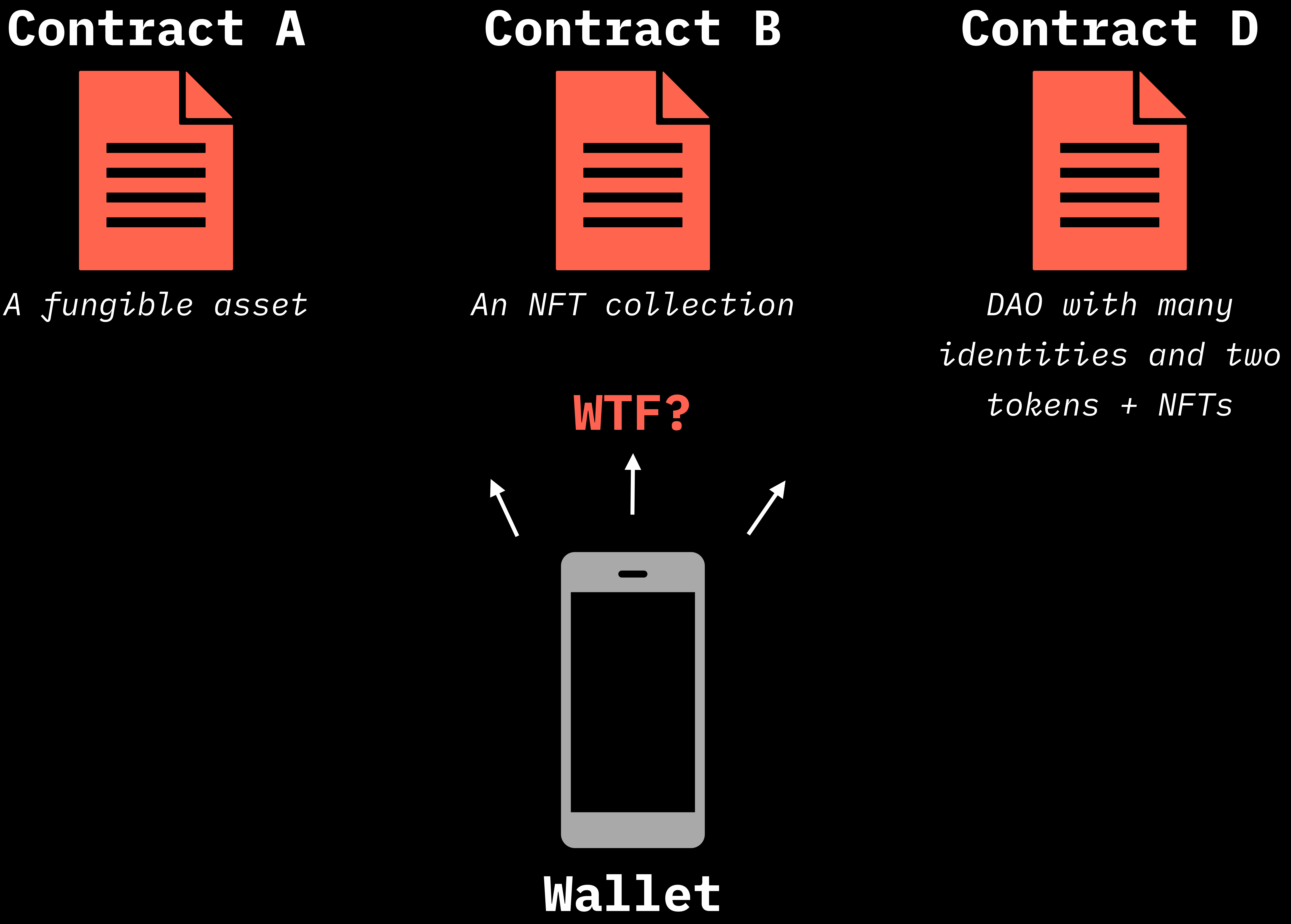
One command to rule all contracts

```
$ git
```

it should work just like git

```
$ cargo install rgb-contracts --all-features
```

How to install



Contract A

A fungible asset

Contract B

An NFT collection

Contract D

DAO with many identities and two tokens + NFTs

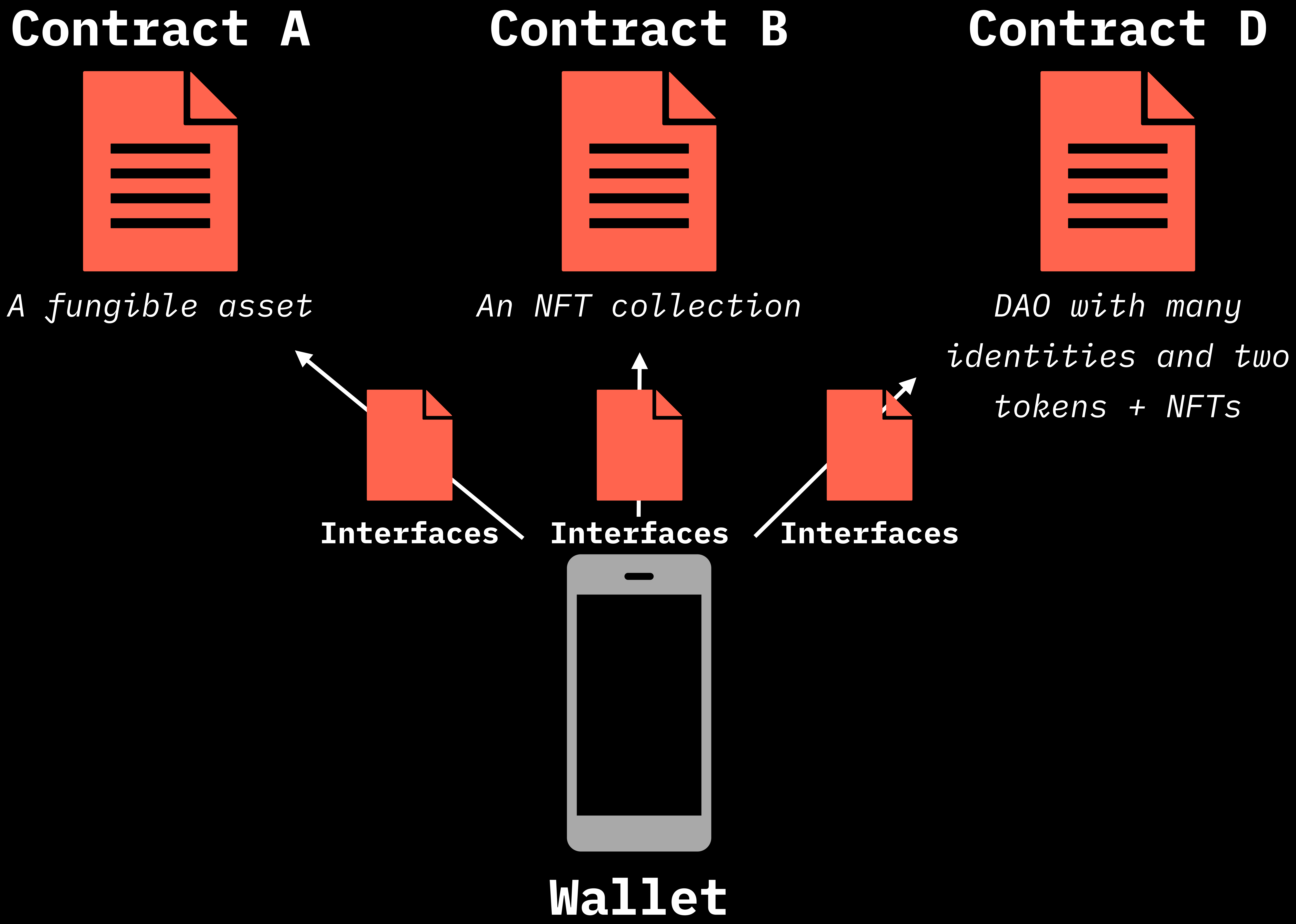
WTF?

Wallet

Interface

Human- and wallet-readable information about the contract (state, operations)

“Interface” or “trait” in context of OOP languages



Contract A

A fungible asset

Contract B

An NFT collection

Contract D

DAO with many identities and two tokens + NFTs

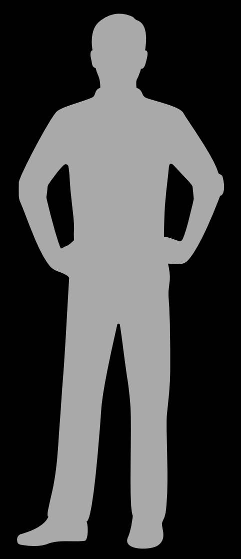
Interfaces

Interfaces

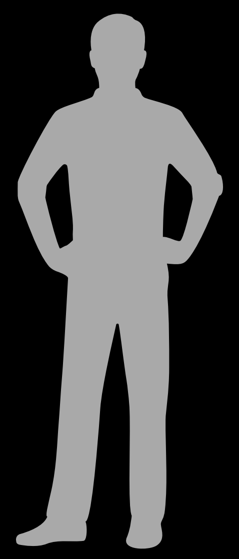
Interfaces

Wallet

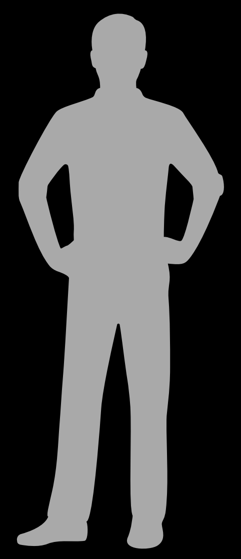
RGB users



Developers

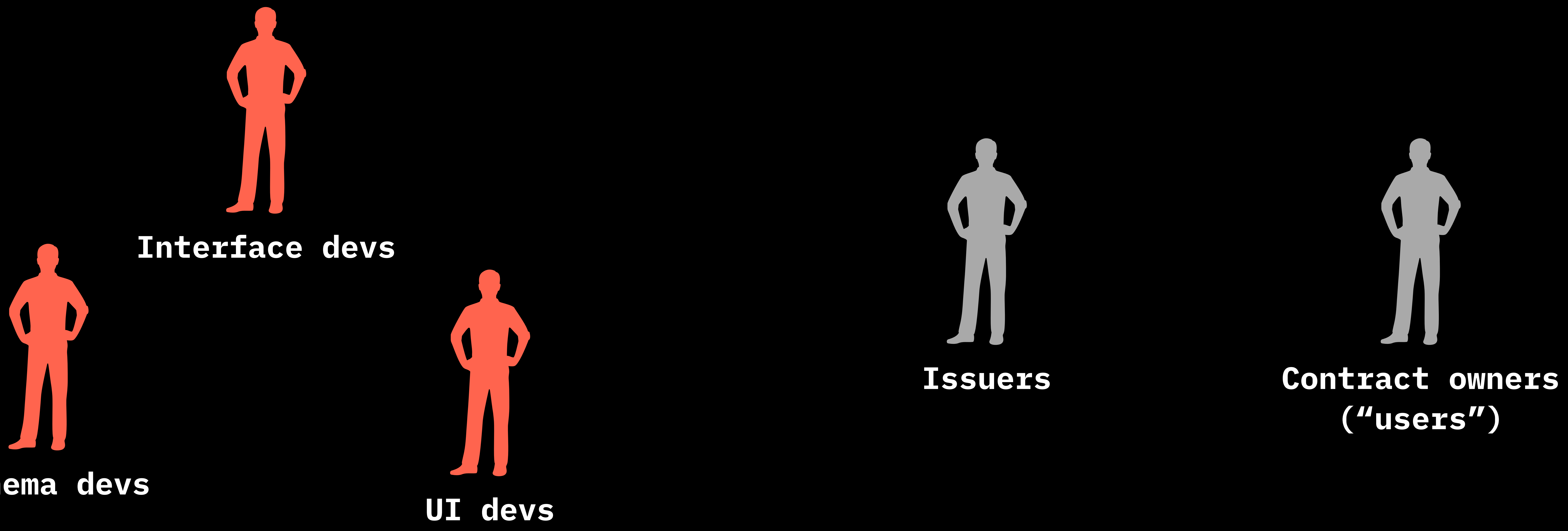


Issuers



Contract owners
(“users”)

RGB users



Interface devs

Schema devs

UI devs

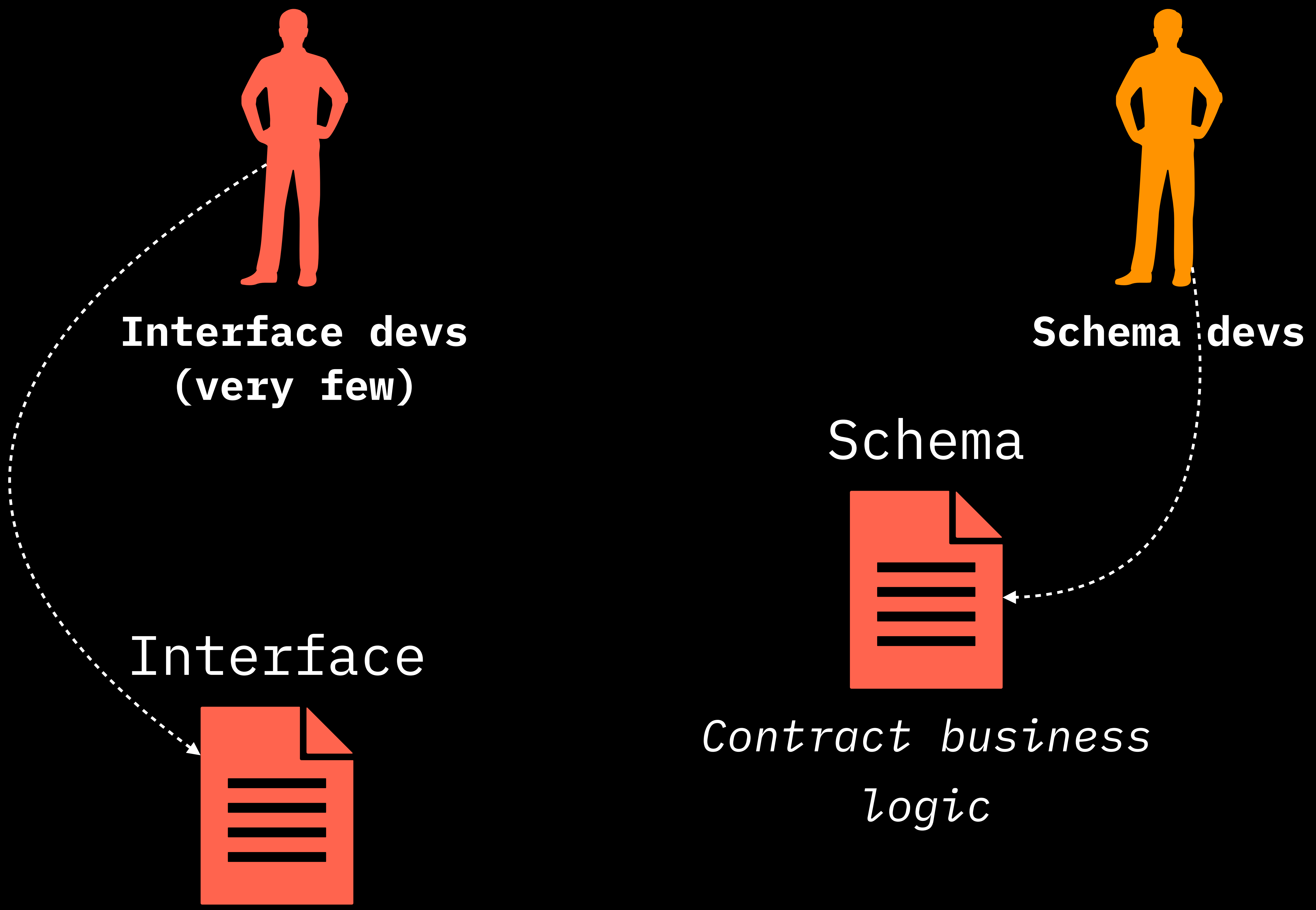
Issuers

Contract owners
(“users”)

**Interface devs
(very few)**

Interface

Instructions on how to operate with contracts



Interface devs (very few)

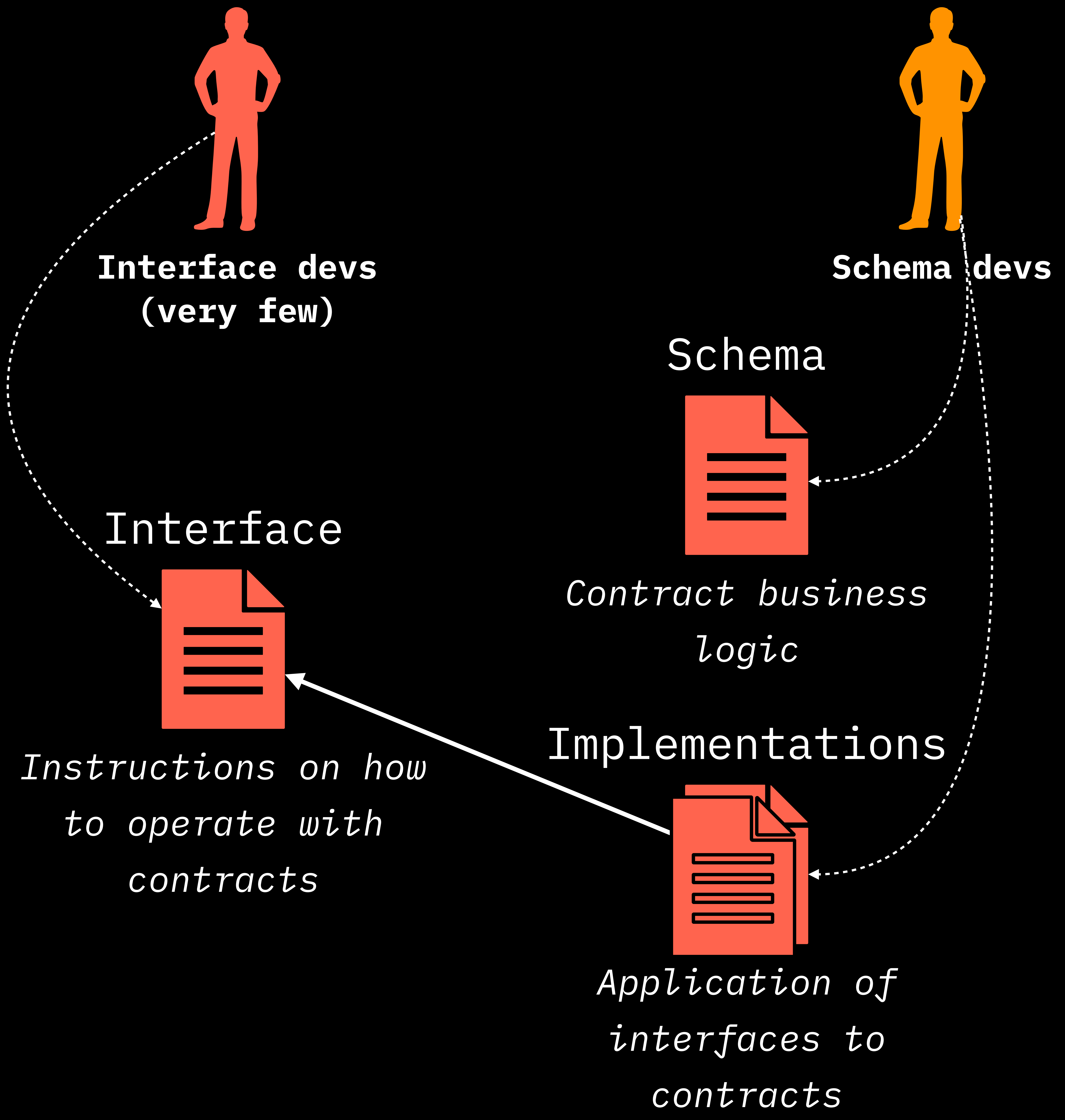
Schema devs

Schema

Contract business logic

Interface

Instructions on how to operate with contracts



Interface devs (very few)

Schema devs

Schema

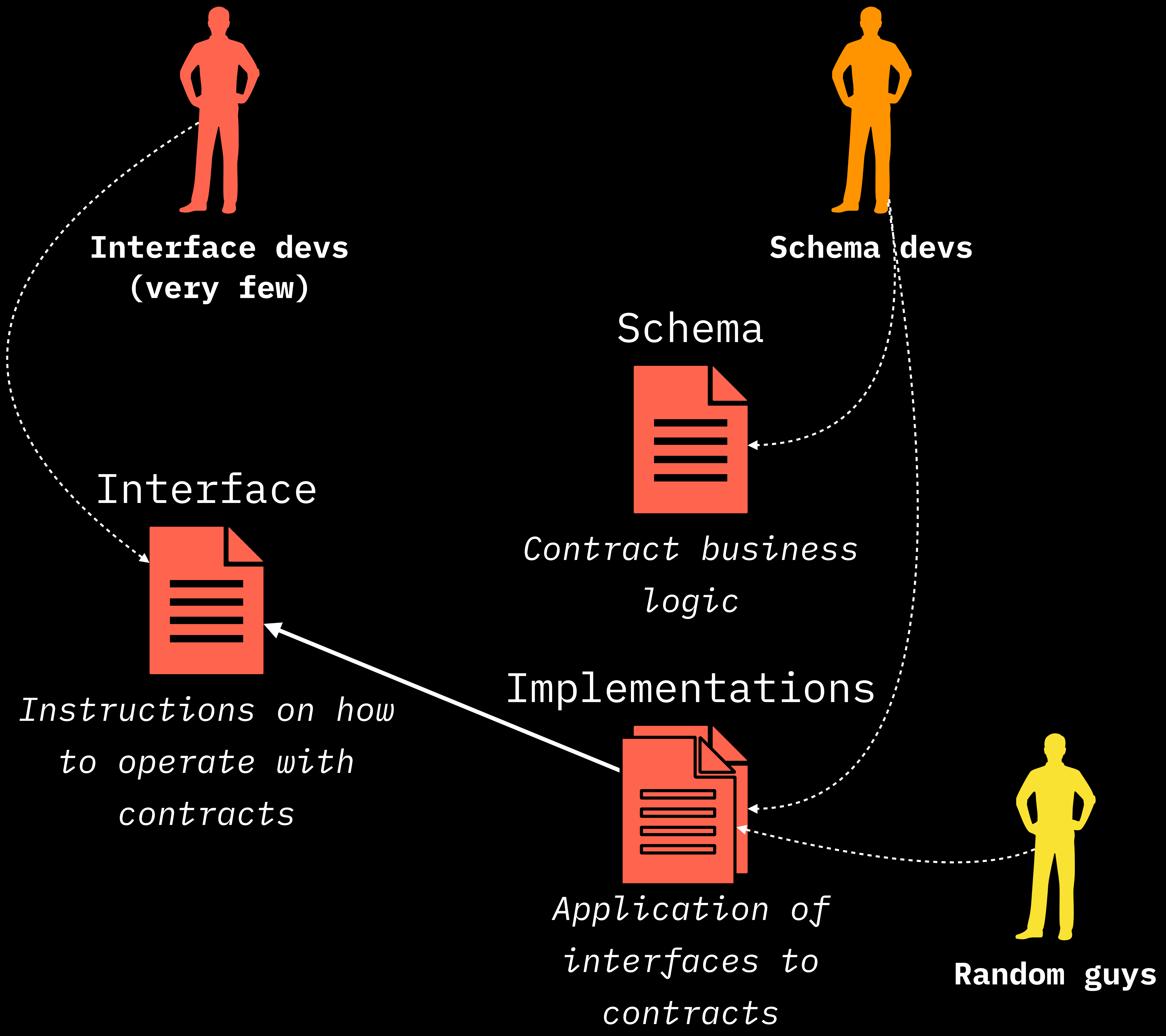
Contract business logic

Interface

Instructions on how to operate with contracts

Implementations

Application of interfaces to contracts



Interface devs (very few)

Schema devs

Schema

Contract business logic

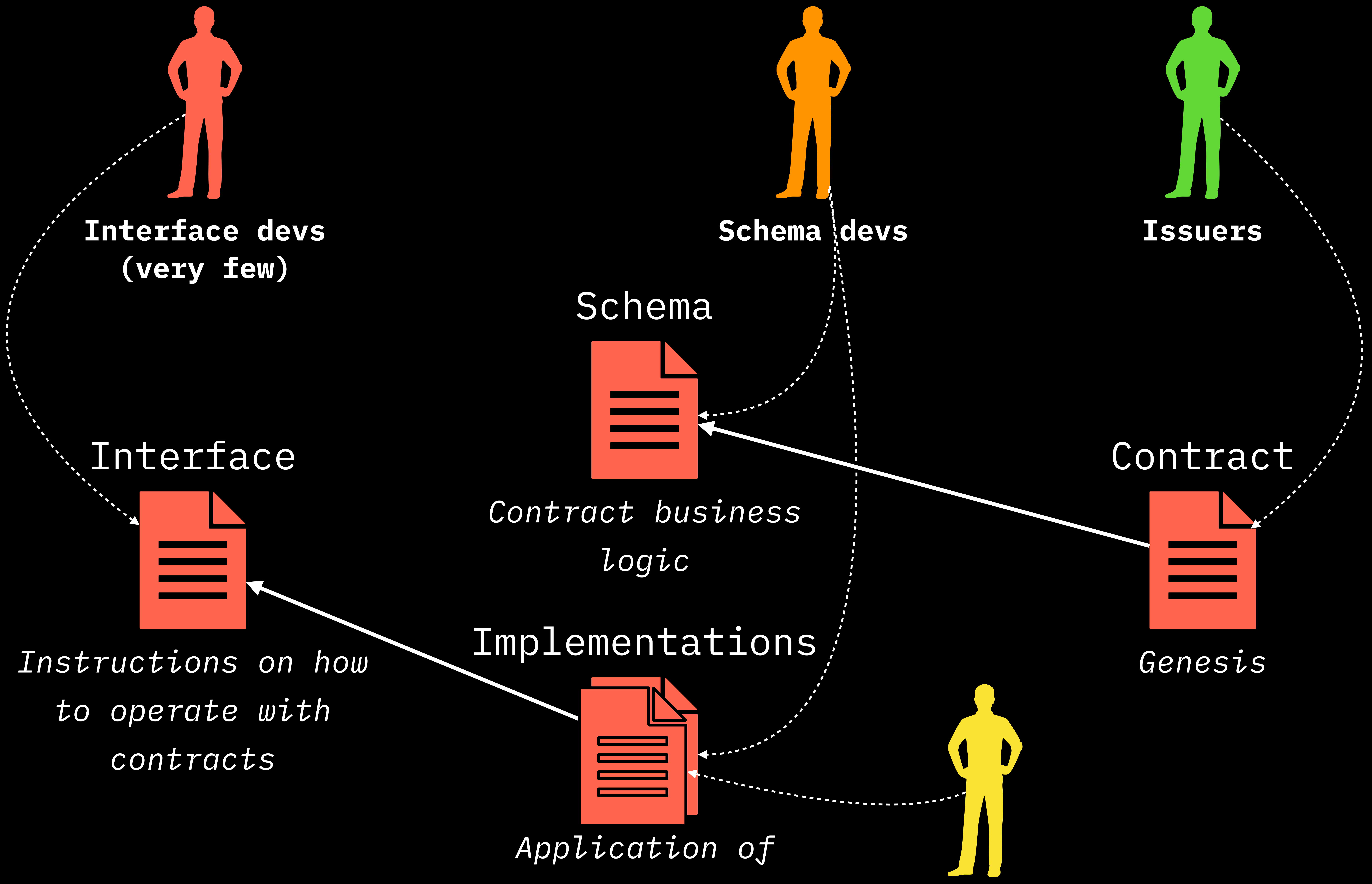
Interface

Instructions on how to operate with contracts

Implementations

Application of interfaces to contracts

Random guys



Interface devs (very few)

Schema devs

Issuers

Schema

Contract business logic

Interface

Instructions on how to operate with contracts

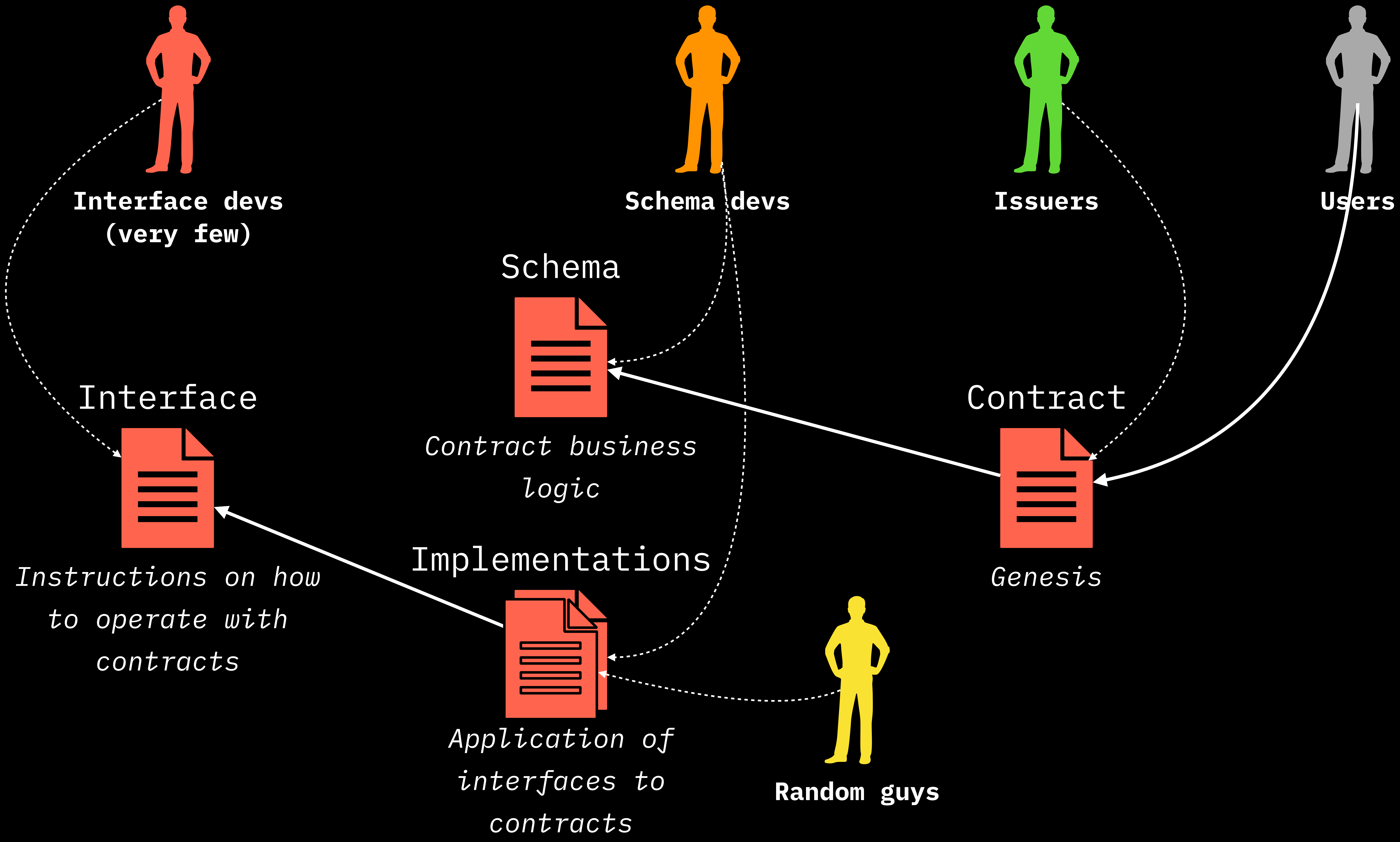
Contract

Genesis

Implementations

Application of interfaces to contracts

Random guys



Interface devs (very few)

Schema devs

Issuers

Users

Schema

Contract business logic

Interface

Contract

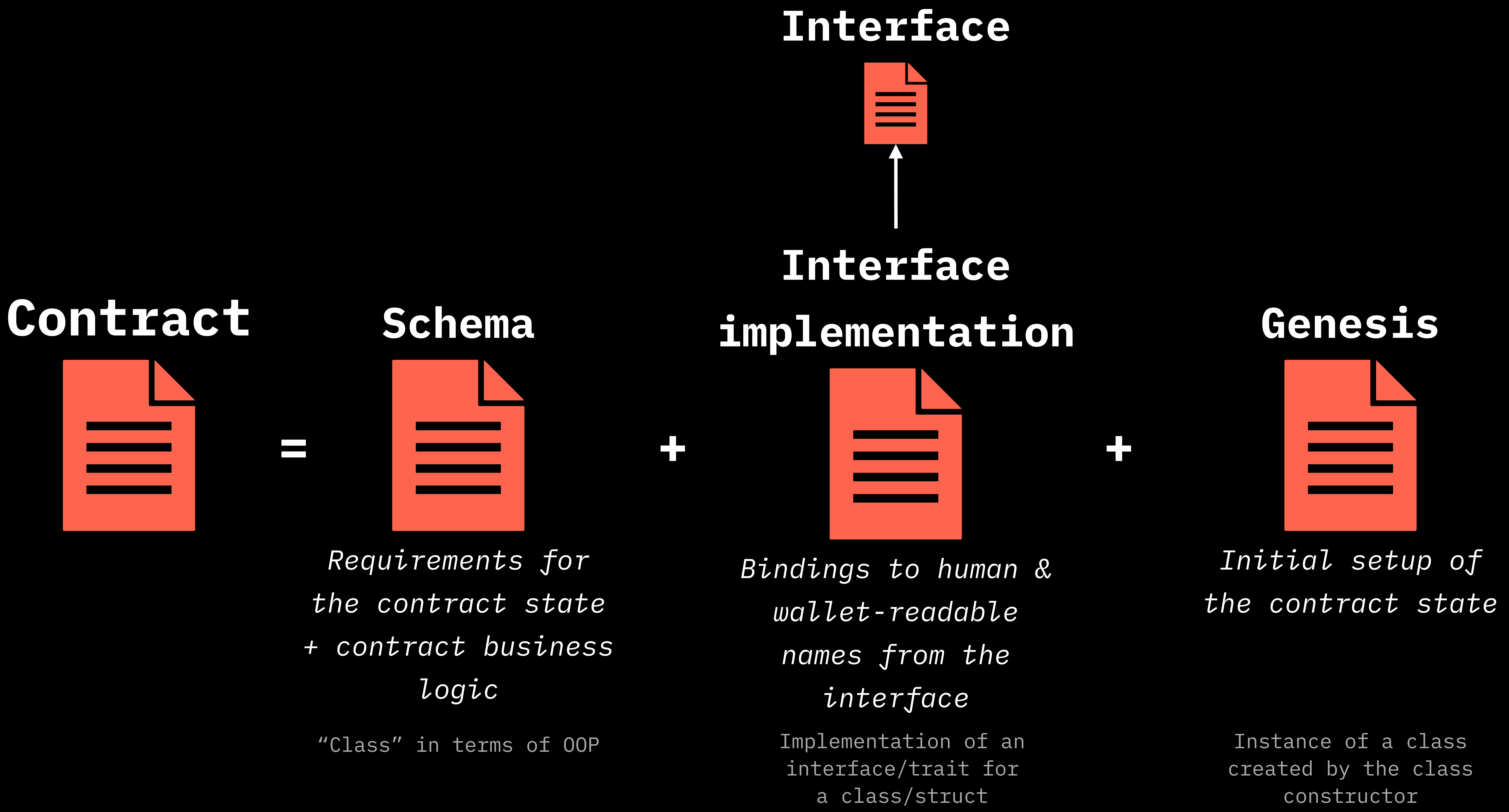
Instructions on how to operate with contracts

Implementations

Genesis

Application of interfaces to contracts

Random guys



Contract = **Schema** + **Interface implementation** + **Genesis**

Interface

Schema

Requirements for the contract state + contract business logic

“Class” in terms of OOP

Interface implementation

Bindings to human & wallet-readable names from the interface

Implementation of an interface/trait for a class/struct

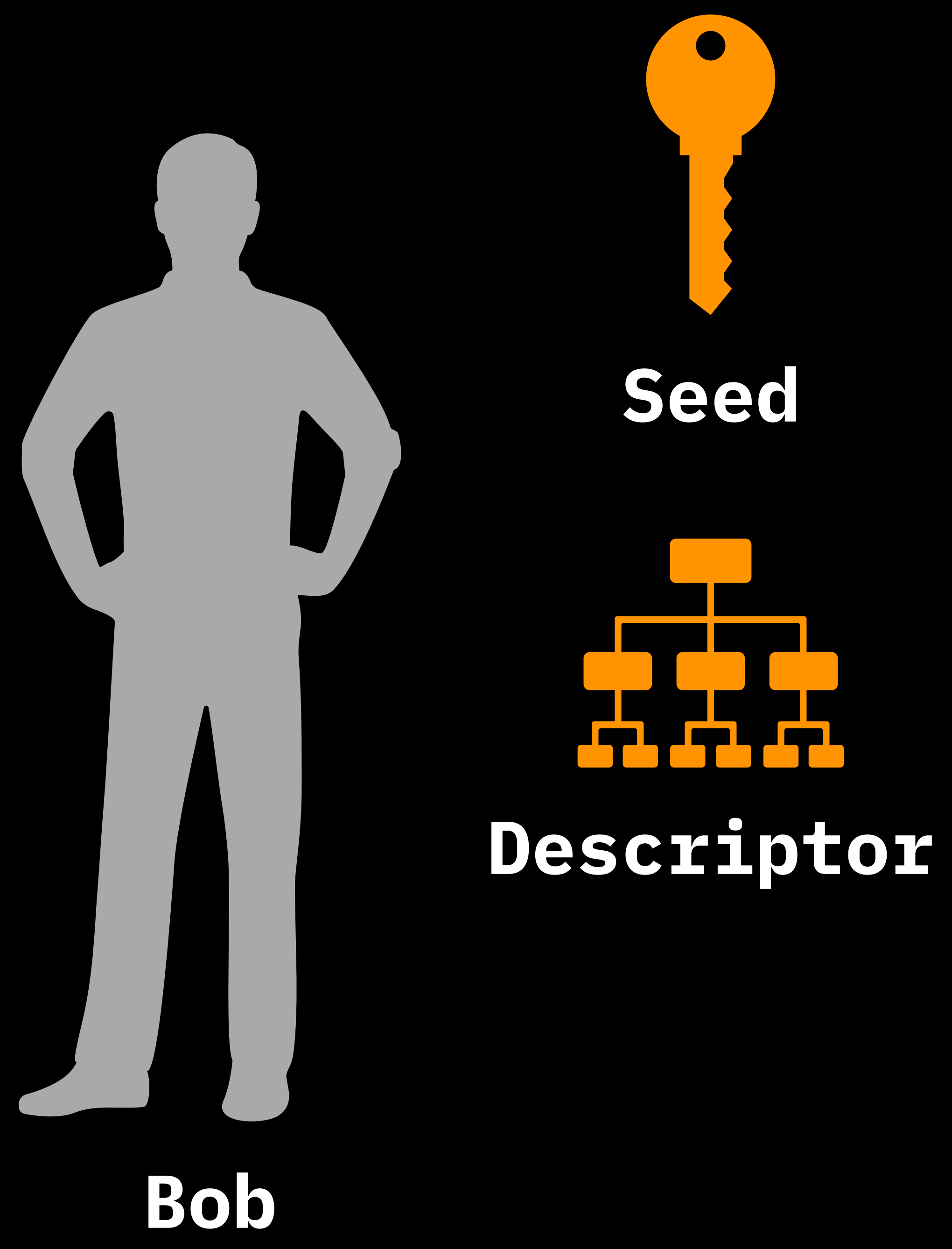
Genesis

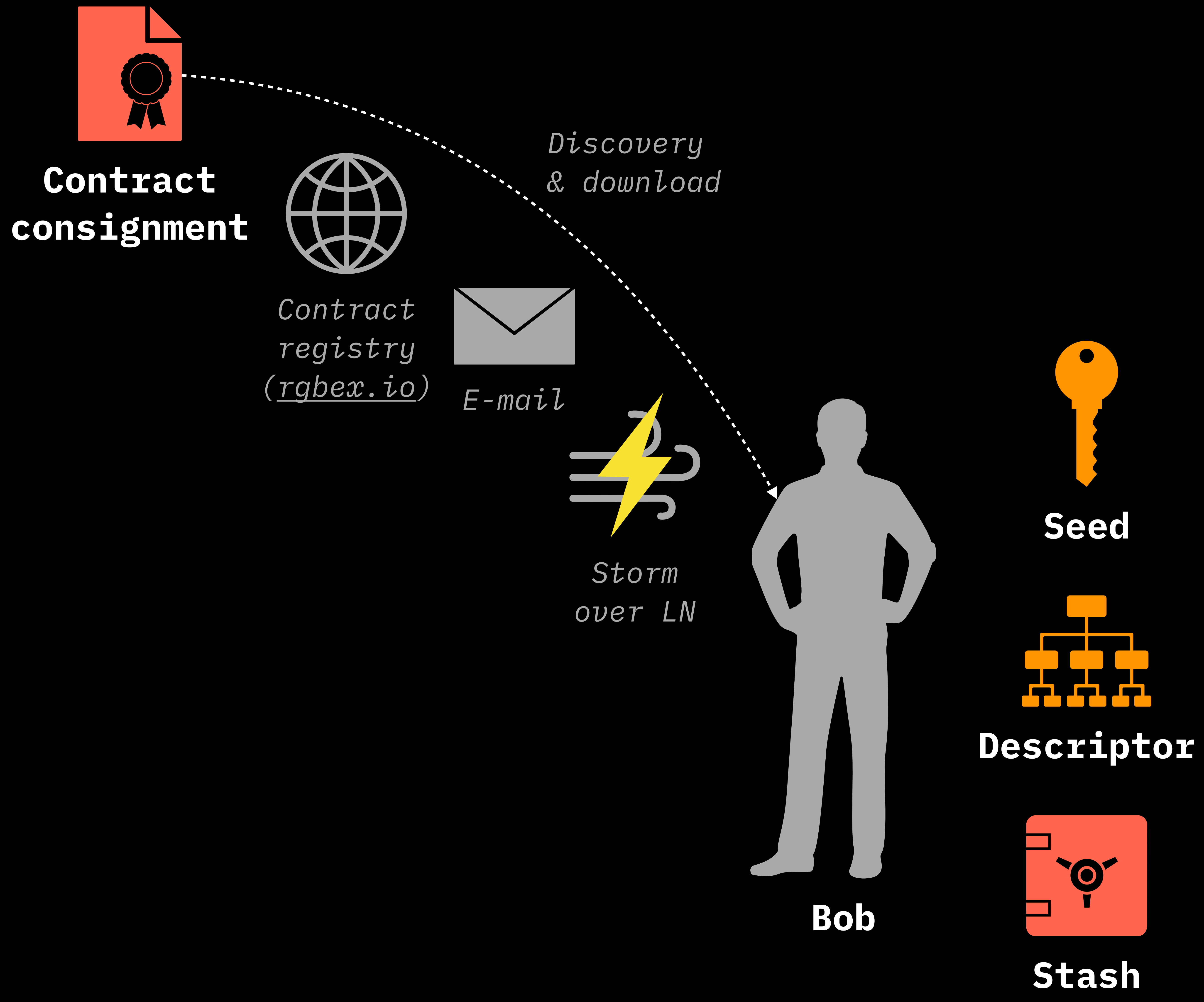
Initial setup of the contract state

Instance of a class created by the class constructor

Contract components overview:

Contract components	Meaning	OOP terms	Ethereum terms
Interface	*Contract semantics*	Interface (Java), trait (Rust), protocol (Swift)	ERC* standards
Schema	*Contract business logic*	Class	Contract
Interface implementation	*Mapping semantics to business logic*	Impl (Rust), Implements (Java)	ABI
Genesis	*Initial contract state*	Class constructor	Contract constructor


Seed
Descriptor
Bob



Contract consignment

Contract registry (rgbex.io)

E-mail

Storm over LN

Discovery & download

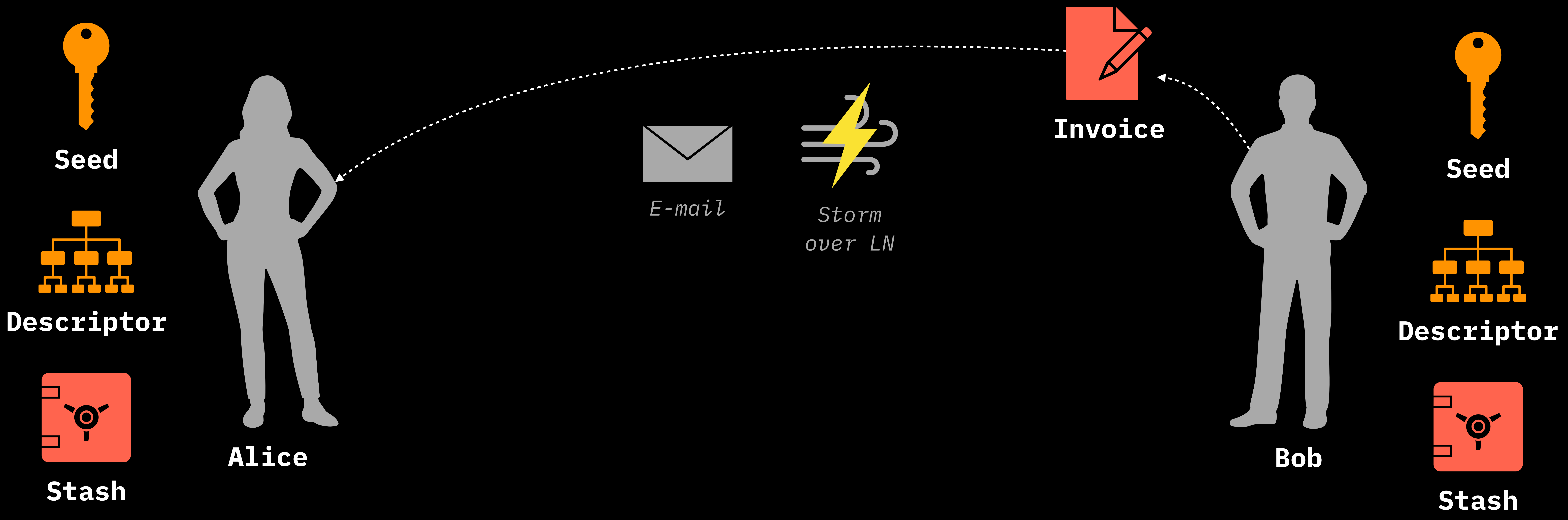
Bob

Seed

Descriptor

Stash

Payment round 1



Seed

Descriptor

Stash

Alice

E-mail

Storm over LN

Invoice

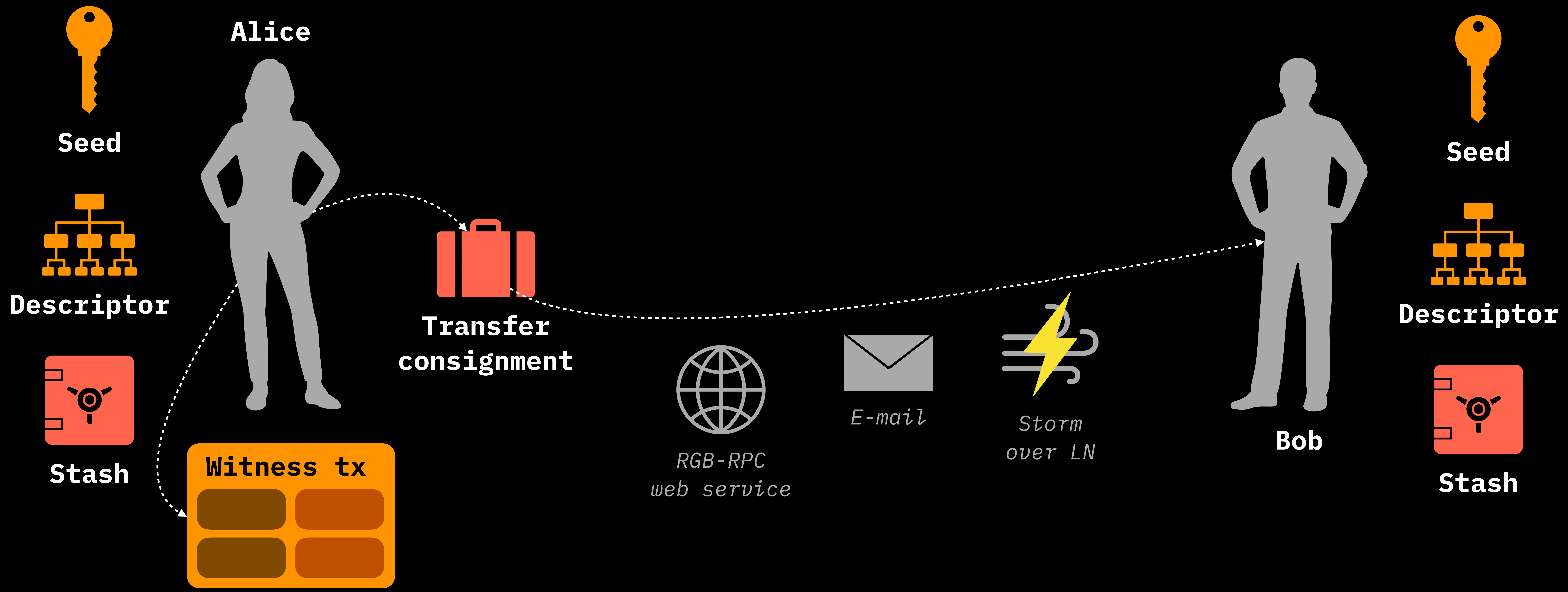
Bob

Seed

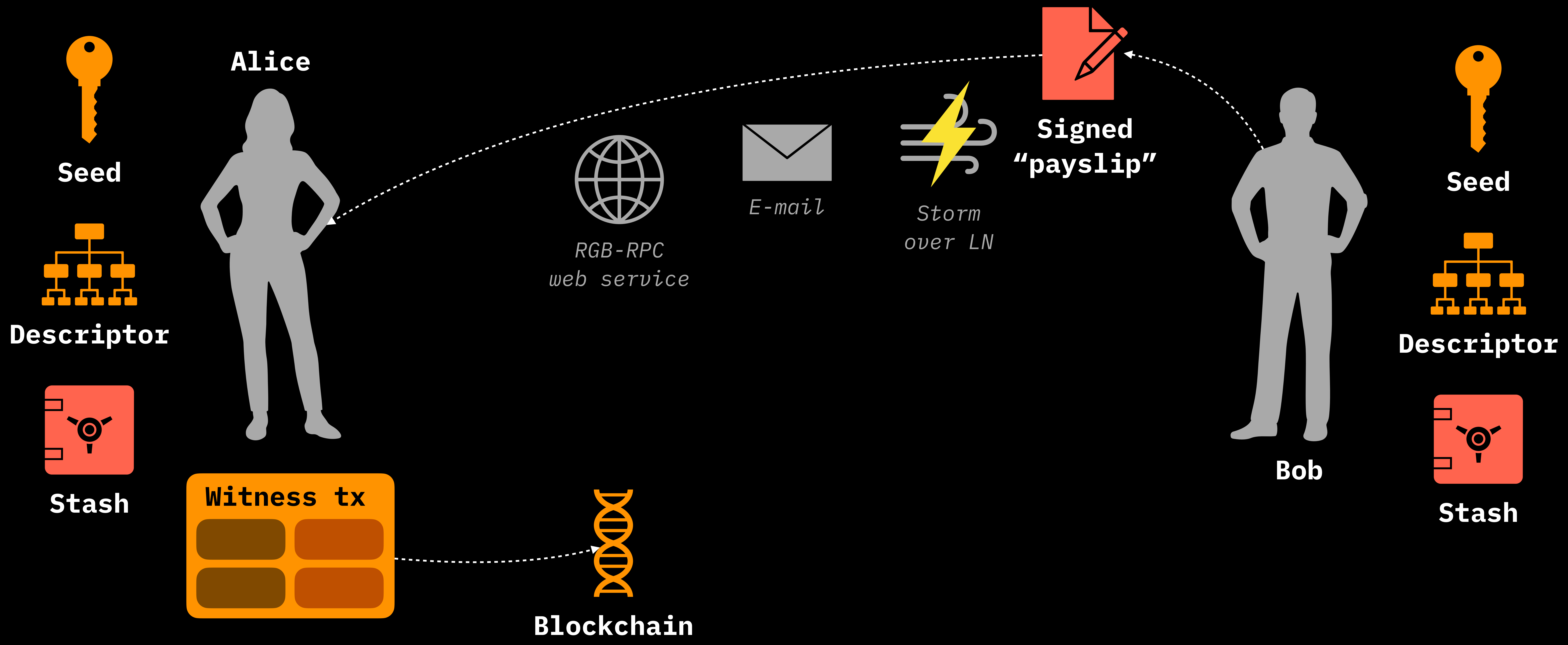
Descriptor

Stash

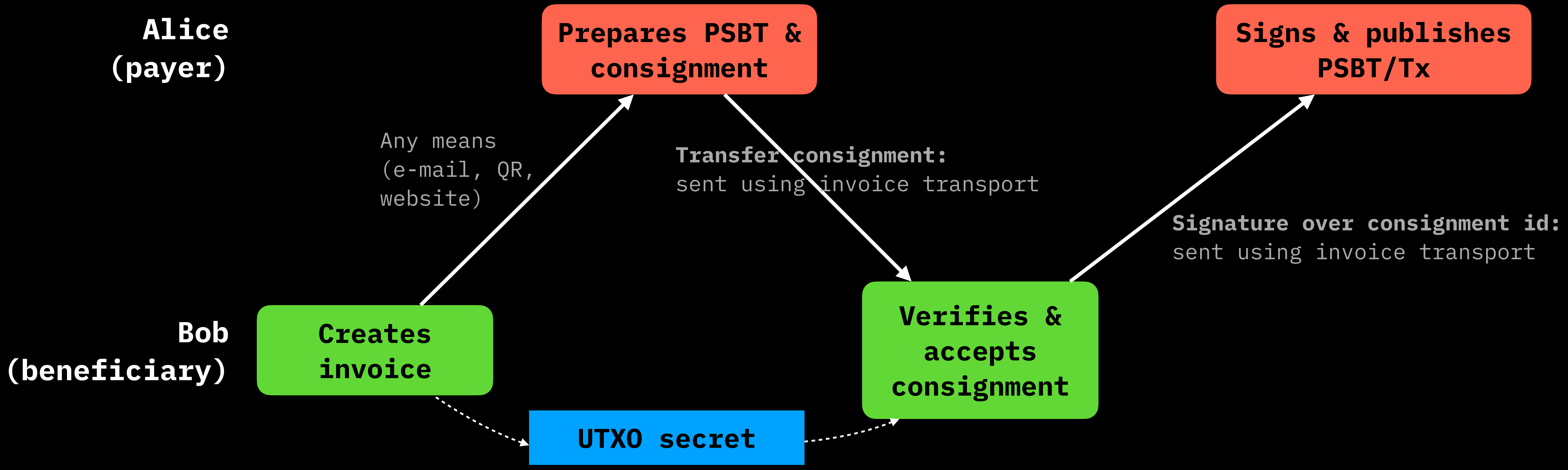
Payment round 2



Payment round 3 (optional)

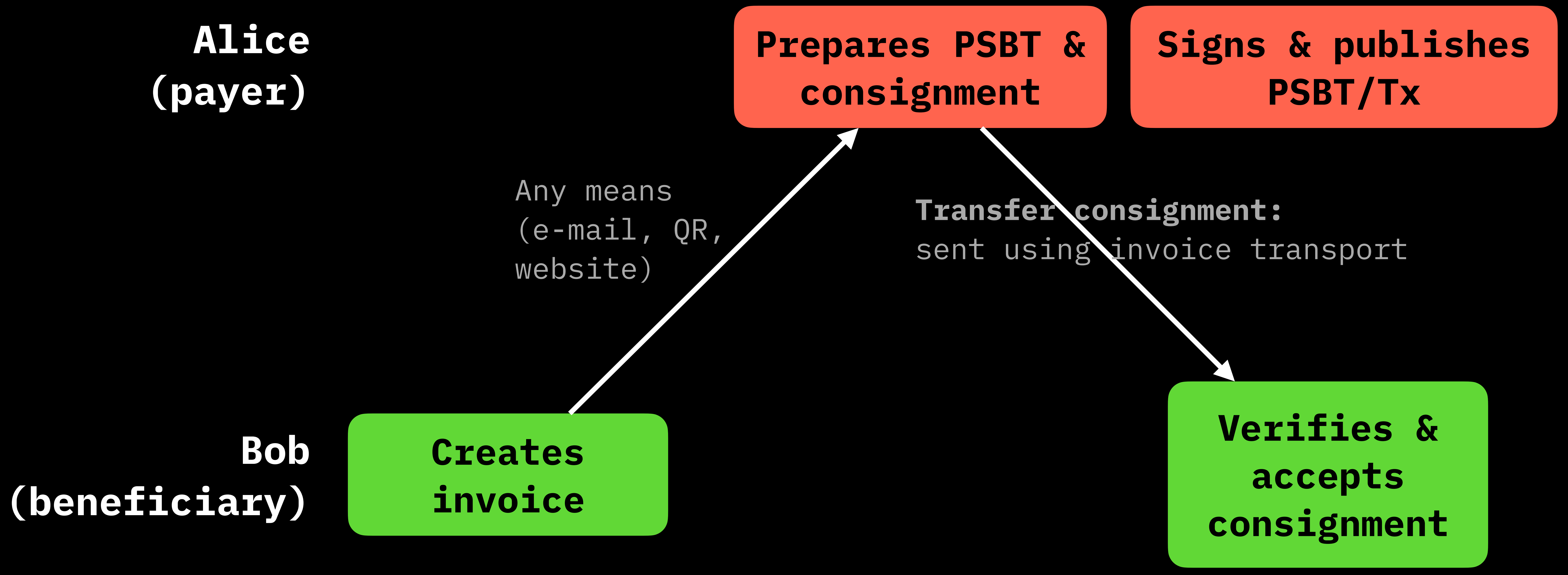


Transfer workflow



Alice
(payer)
Bob
(beneficiary)
Creates invoice
Any means
(e-mail, QR,
website)
Prepares PSBT & consignment
Transfer consignment:
sent using invoice transport
Verifies & accepts consignment
UTXO secret
Signature over consignment id:
sent using invoice transport
Signs & publishes PSBT/Tx

Transfer workflow



Alice (payer)

Prepares PSBT & consignment

Signs & publishes PSBT/Tx

Any means (e-mail, QR, website)

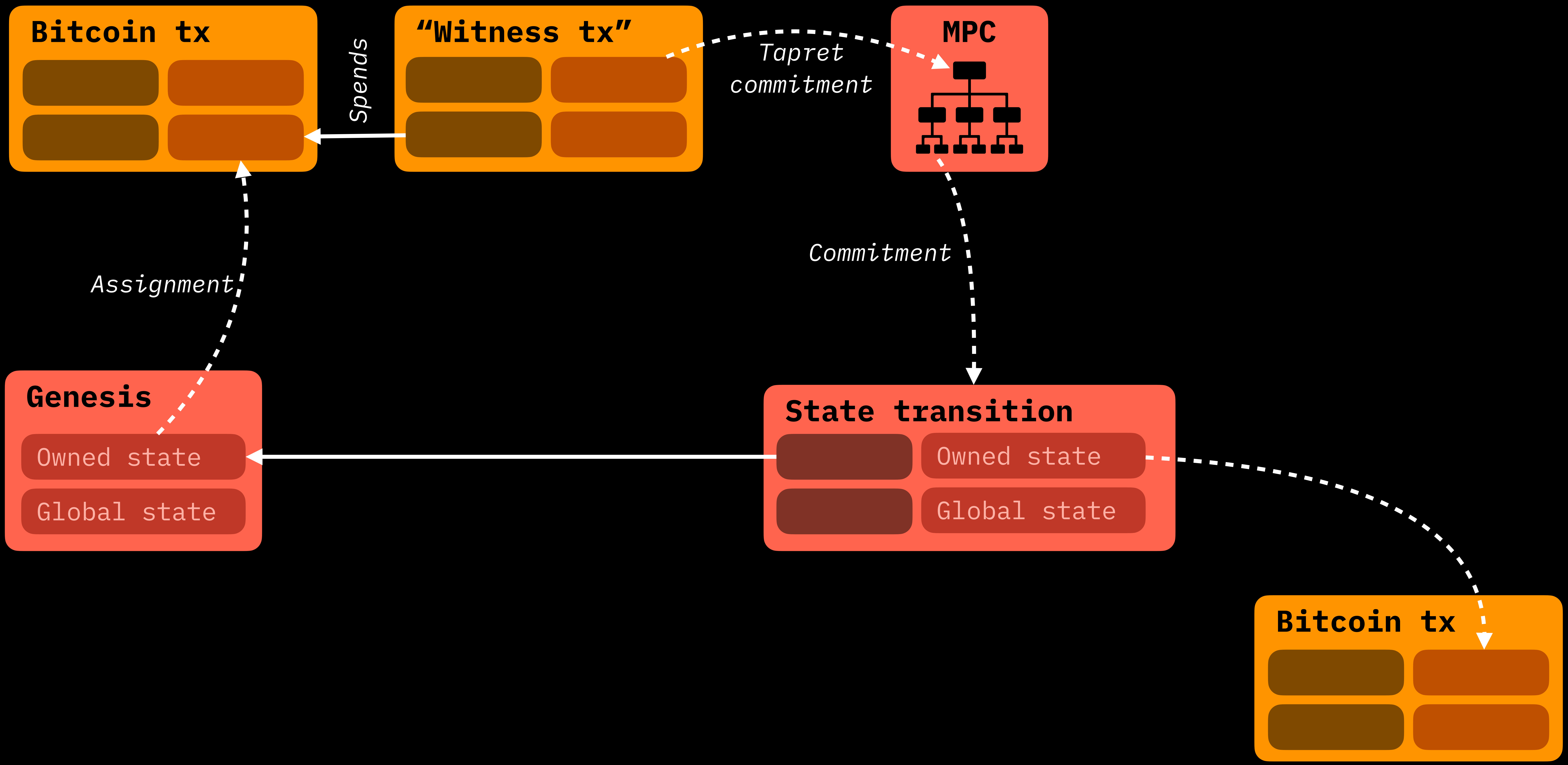
Transfer consignment: sent using invoice transport

Bob (beneficiary)

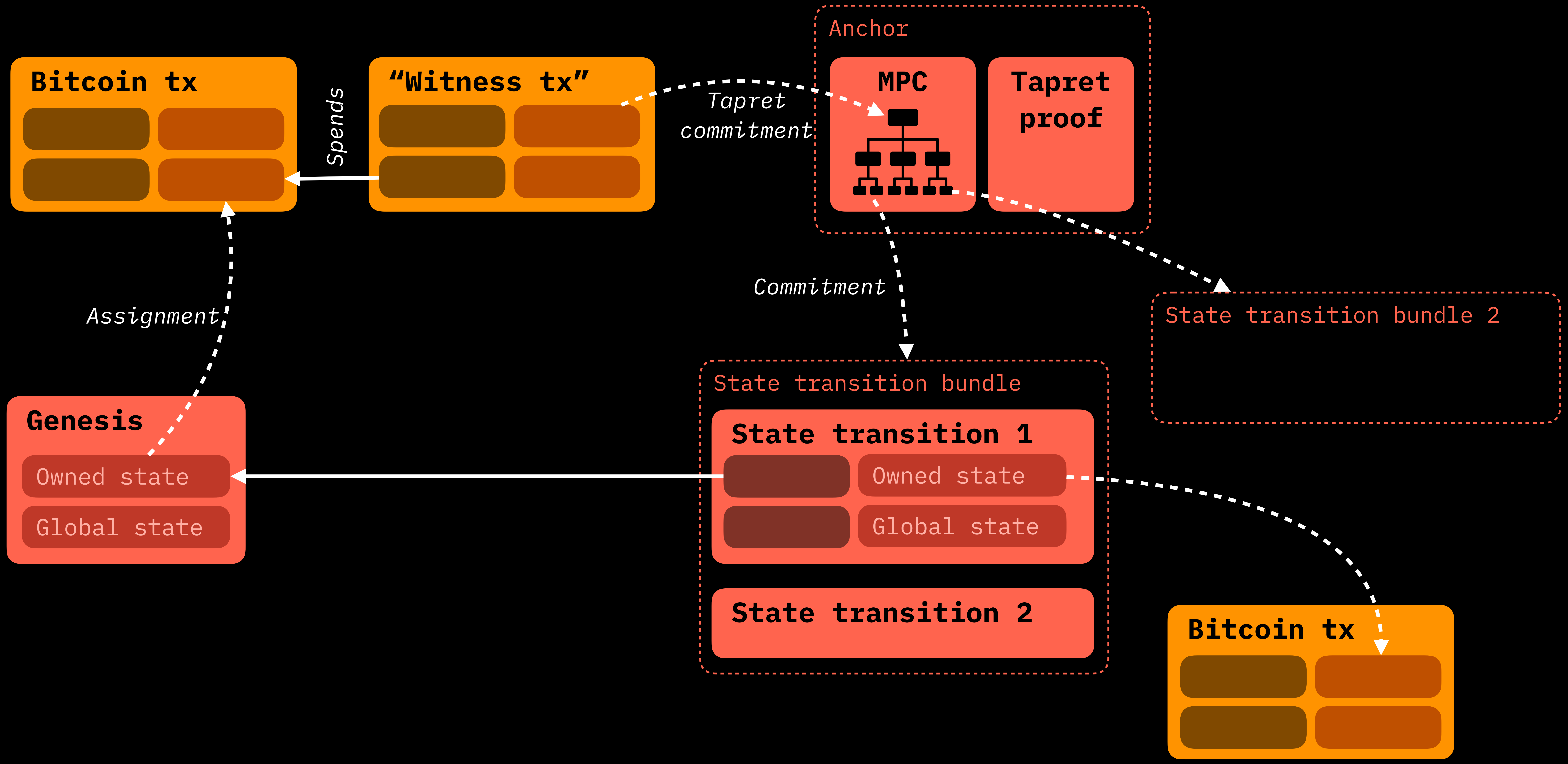
Creates invoice

Verifies & accepts consignment

Anatomy of RGB operation

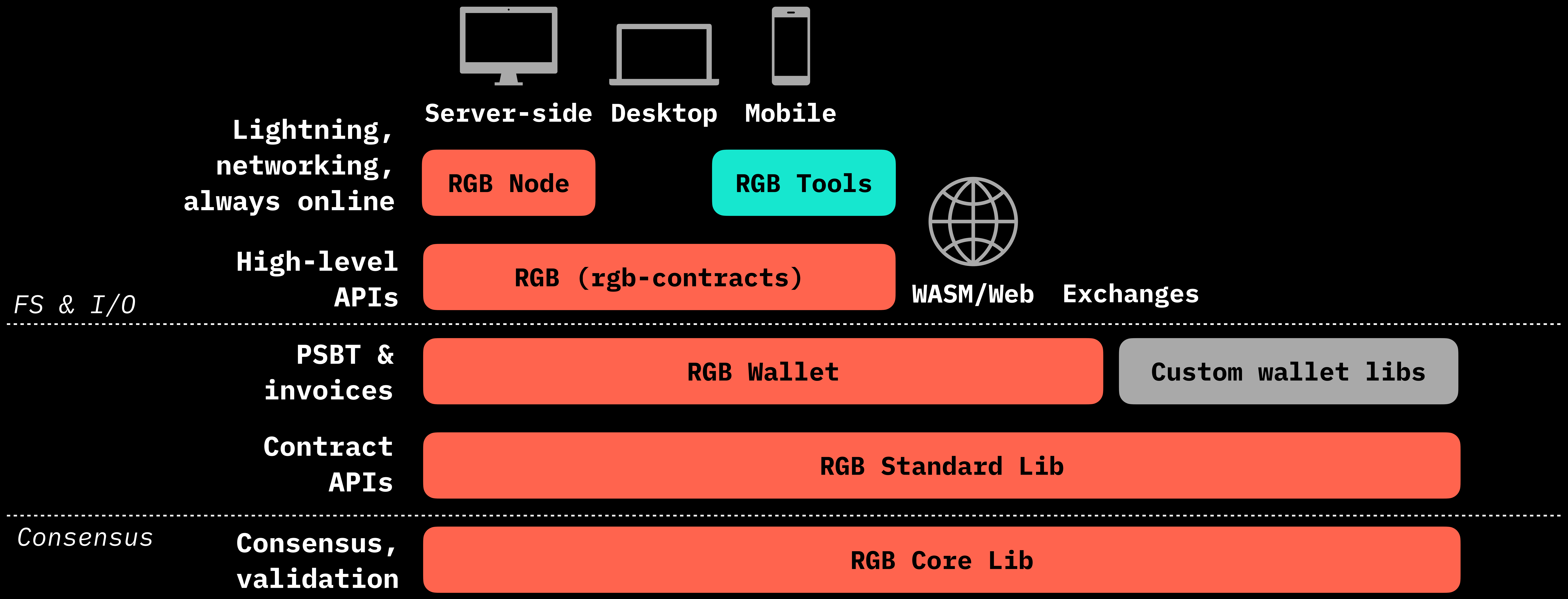


Anatomy of RGB operation



Header
State evolution
Extensions
Genesis
2
32
1
Metadata
Global state
Assignments
Valencies
fast-forward version
schema id
chain
State transition
2
1
Metadata
Global state
PrevOuts
Assignments
Valencies
fast-forward version
transition type id
State extension
2
1
Metadata
Global state
Assignments
Redeems
Valencies
fast-forward version
extension type id
2
0...64KiB
data length
strict-serialized value
1
2
...
number of entries
valency type id
Global state entry
1
2
2
1...64KiB
...
number of entries
global state type id
data length
strict-serialized value
Redeem entry
1
2
32
...
number of entries
valency type id
previous operation id
PrevOuts
Assignments
PrevOut entry
2
32
2
2
...
number of inputs
previous operation id
assignment type
output number
Typed assignments
1
2
1
2
Assignment
...
number of assignments
assignment type
state type
number of assignments of the same type
aray of assignments
Assignment
0x01
Seal
ConfidentialState
0x03
Seal
RevealedState
0x00
32
ConfidentialState
0x02
32
RevealedState
seal data commitment
State transition seals
Witness-based seal
1
0x00
4
8
1
0x01
32
4
8
Genesis & extension seal
1
32
4
8
closing method
transaction id
tx output number
seal blinding nonce
Anchor
TransitionBundle
TxId
MPC Proof
DBC Proof
N
OpId
BundleItem
OpId
BundleItem
...
N
vout
vout
...
1
Transition
Revealed
N
vout
vout
...
0
Concealed
0
Opret
1
0
nonce
InternalPk
Tapret level 0
1
1
TapretNodePartner
nonce
InternalPk
Tapret level 1
0
TapNodeHash
Left node
1
Leaf ver
N
Script...
Right leaf
2
TapNodeHash
TapNodeHash
Right branch
pos
N
MerkleNode
MerkleNode
...

RGB library stack for app devs



Server-side Desktop Mobile

Lightning, networking, always online — RGB Node — RGB Tools

High-level APIs — RGB (rgb-contracts) — WASM/Web — Exchanges

FS & I/O

PSBT & invoices — RGB Wallet — Custom wallet libs

Contract APIs — RGB Standard Lib

Consensus

Consensus, validation — RGB Core Lib