LNP/BP Standards Association

RGB: from the past to the future

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 back in the day

“Colored coins on Lightning”

RGB today: smart contracts platform

- **Turing-complete**, functional-style **programmability for bitcoin**
- **Offchain means scalable**: LN & client-side-validation
- **Private**: zero-knowledge & confidentiality

Compatibility

Taproot

Multisig wallets

Schnorr signatures

MuSig2

Hardware wallets

Miniscript

Atomic swaps

Adaptor signatures

Submarine swaps

Lightning Network

DLCs

CoinJoin

eltoo

Cross-input signature aggregation

PayJoin

SIGHASH_ANYPREVOUT

Cross-chain swaps

RGB has become community & ecosystem

3 new RGB wallets

MyCitadel · DIBA · Iris

2 dev kits

RGB TOOLS

4 exchanges integrating RGB

25+ active GitHub contributors, hundreds of stars

Contributors 24

+ 13 contributors

rgb-core Public

RGB Core Library: consensus validation for private & scalable client-validated smart contracts on Bitcoin & Lightning

Rust 98 18

rgb-node Public

RGB node for both servers and mobiles

Rust 88 34

First investments

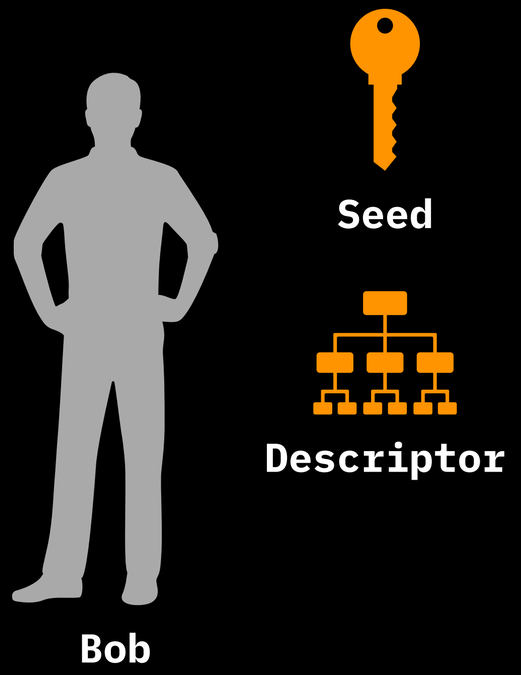
Fulgur Ventures · Draper Associates

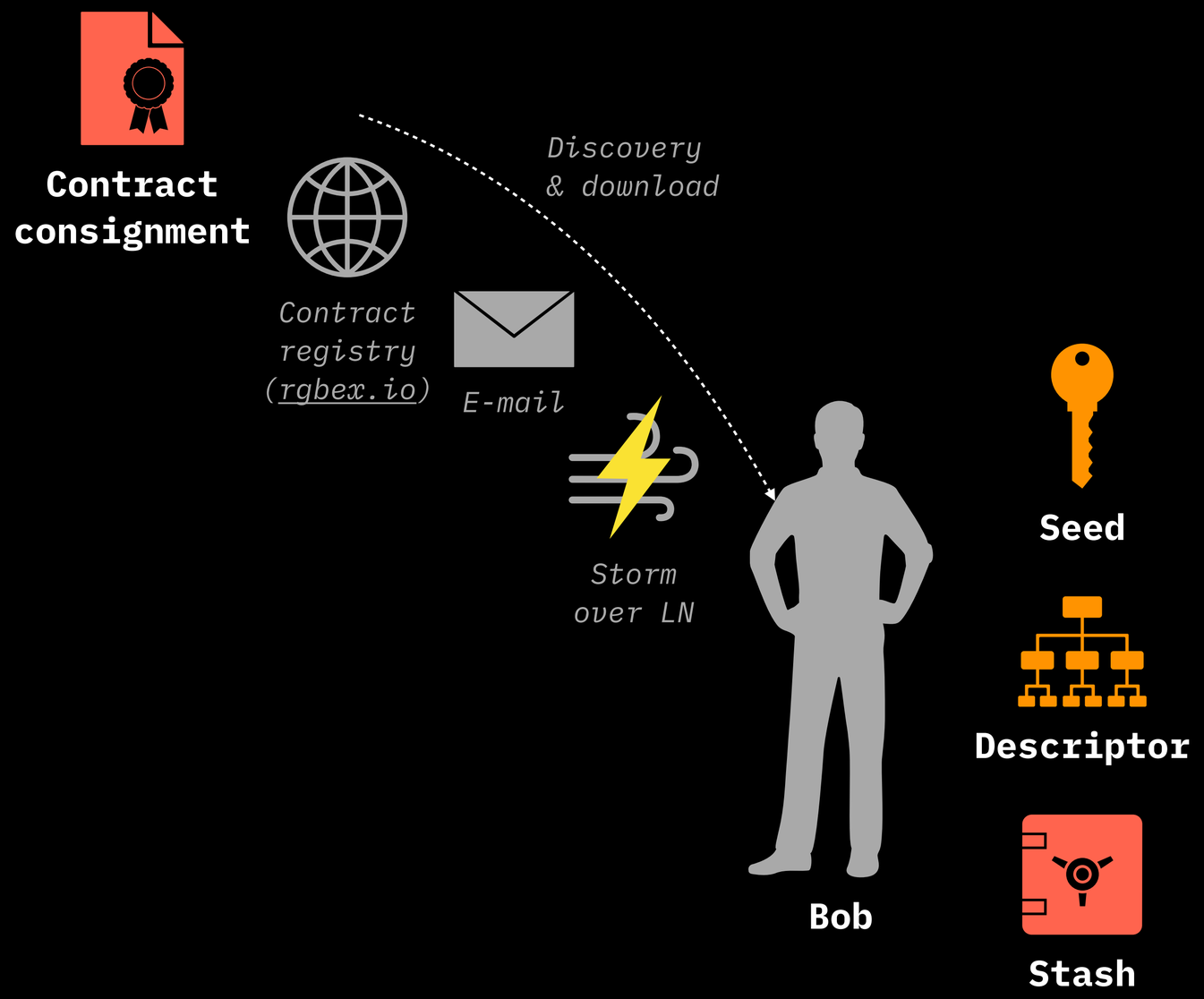
Extensive media coverage

Surfin' Bitcoin · bitcoin amsterdam · #BH2022 · Bitcoin Magazine

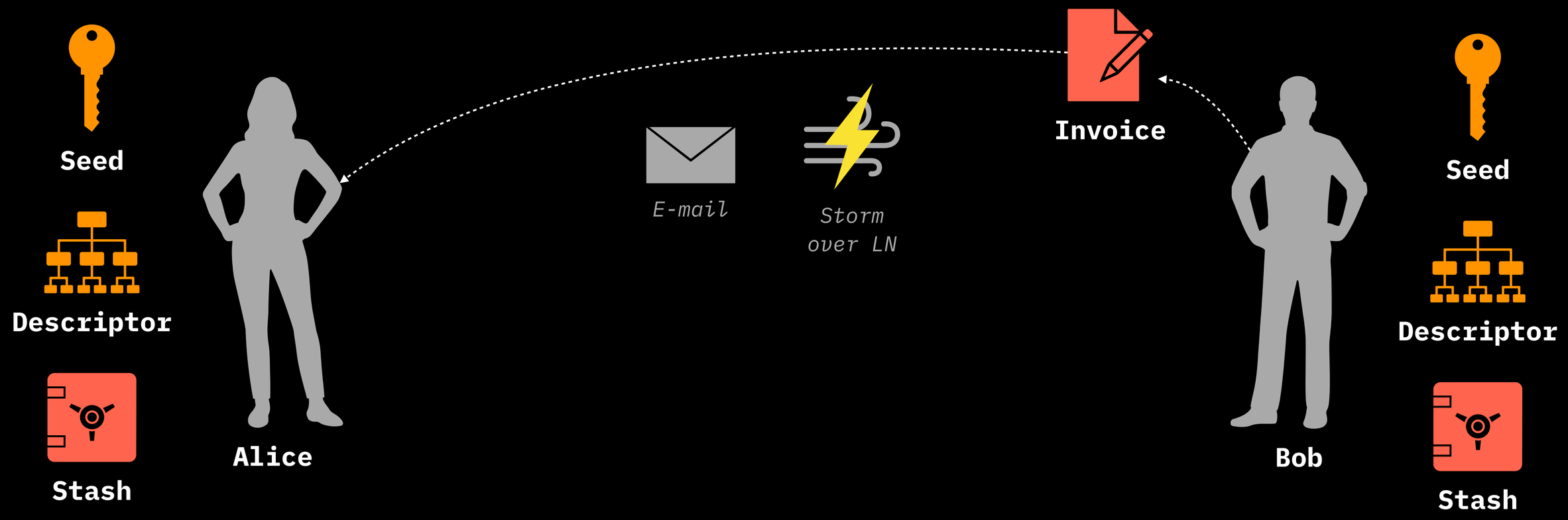
RGB	Taro
In development since 2016	Announced last year
~1 mln of external funding, ~500k of developer funds	>80m of funding
Original concept by Peter Todd & Giacomo Zucco	RGB copycat
Cross-industry effort (5+ competing companies involved)	Internal Lightning Labs project
Released Jul 2022, alpha in 2019, beta in 2021	Alpha in 2022
Smart contracts - **Can do DeFi (DEX, AMM, algorithmic stablecoins etc)** - **Can do DAO**	**Just tokens and nothing more**

Internal name was CMYK

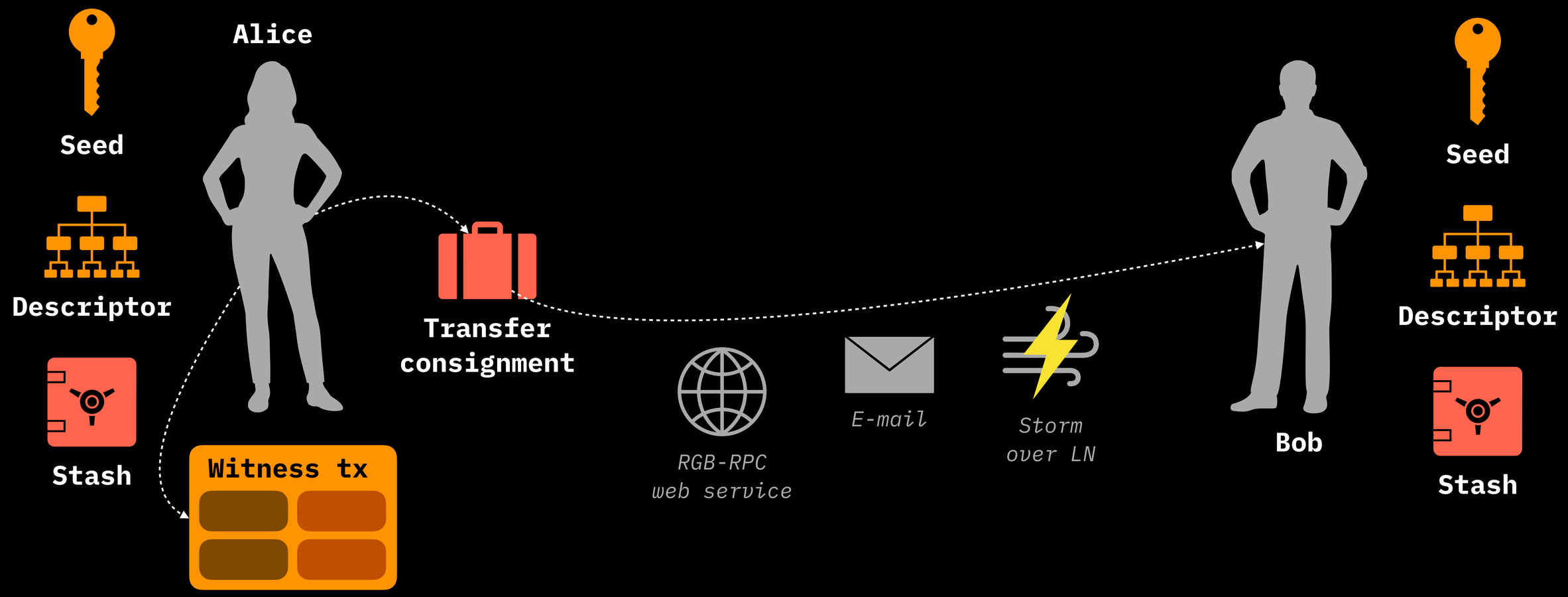

Seed
Descriptor
Bob


Contract consignment
Discovery & download
Contract registry (rgbex.io)
E-mail
Storm over LN
Bob
Seed
Descriptor
Stash

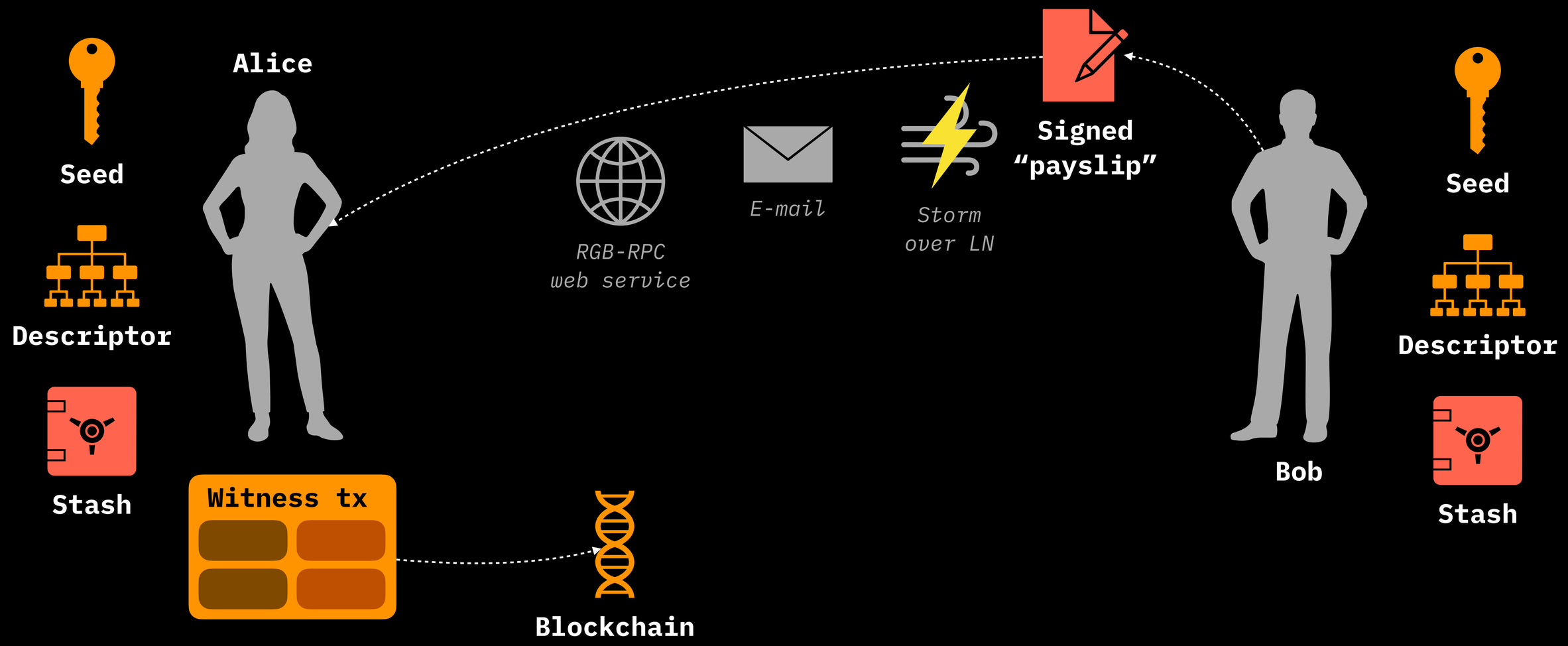
Payment round 1



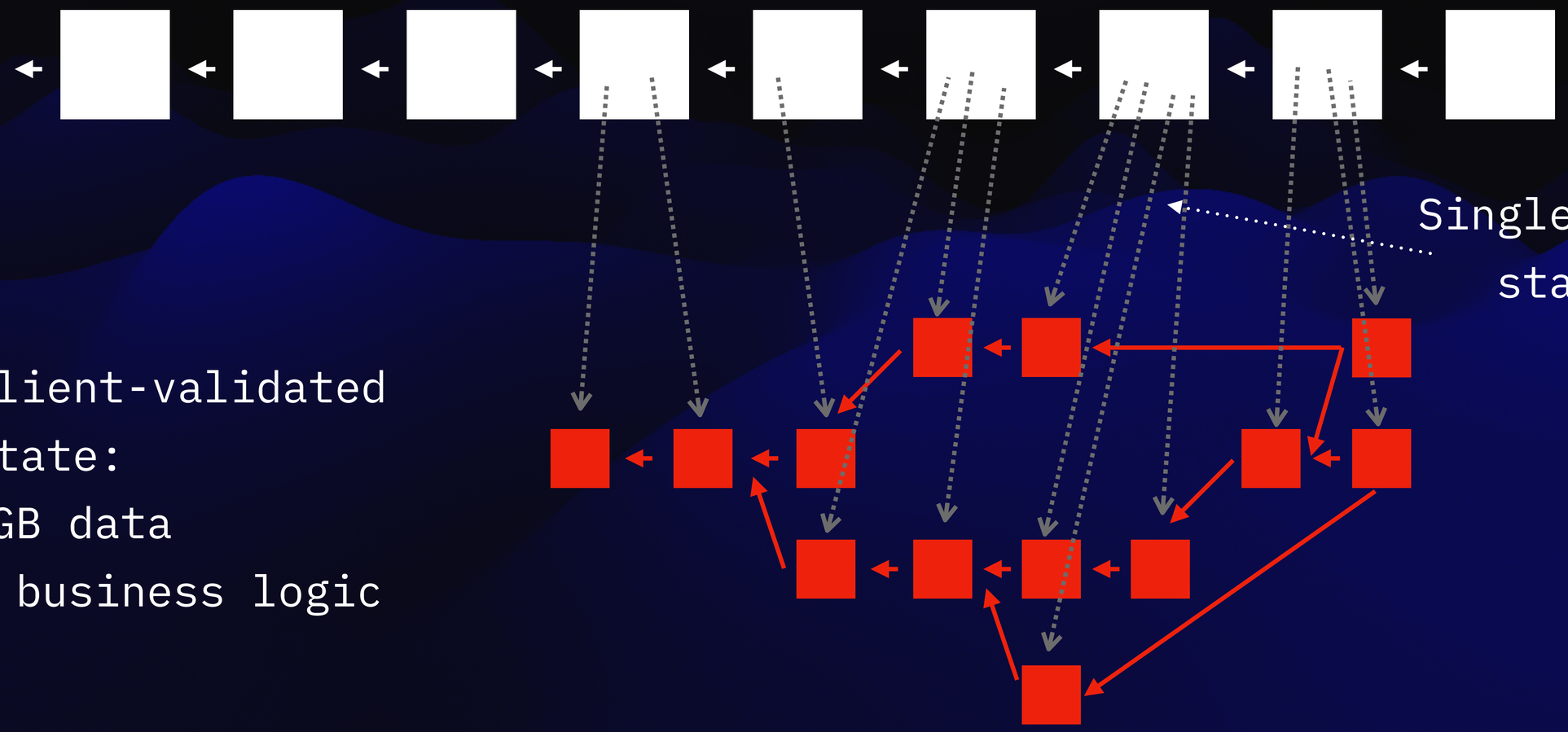
Payment round 2



Payment round 3 (optional)



Bitcoin blockchain:
state ownership



Single-use seals:
state bindings

Client-validated
state:
RGB data
& business logic

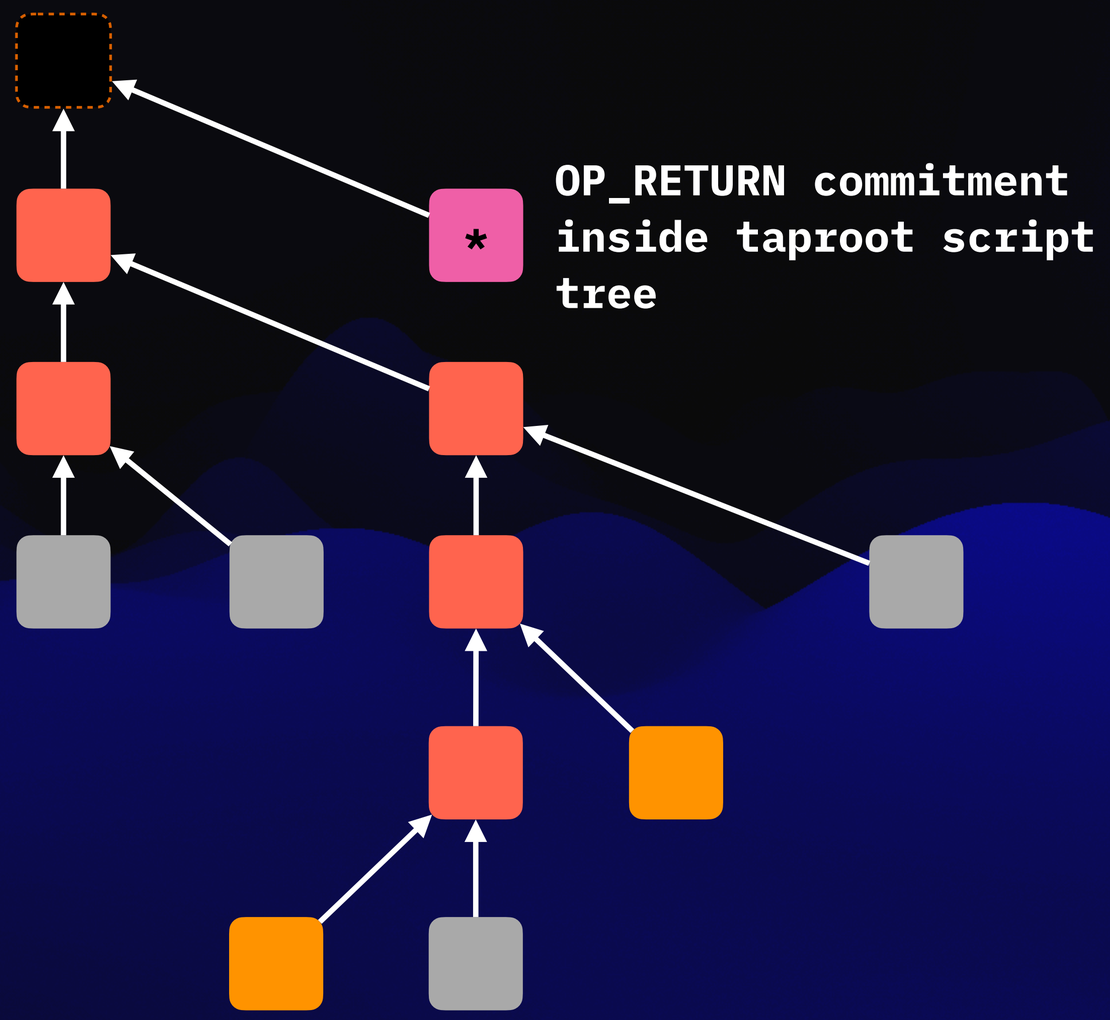
~~OP_RETURN~~

~~Pay-to-contract~~

~~Sign-to-contract~~

Tapret

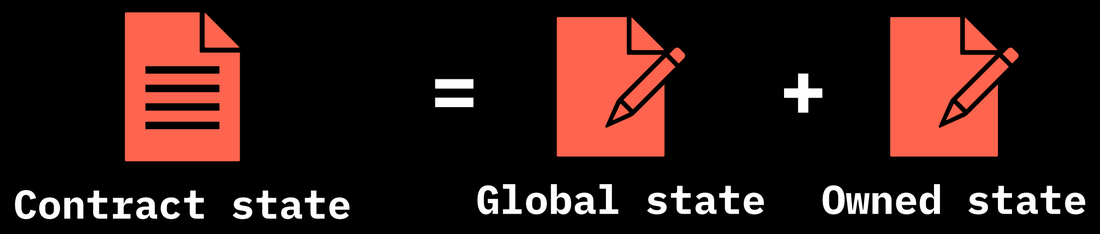
Novel type of private commitments not using blockchain space and affecting keys/signatures



Contract state

- **Global state**: nobody owns, everyone knows
  Examples: asset name

- **Owned state**: someone owns, nobody else knows
  Examples: asset amounts, voting rights, NFT content



Contract A

A fungible asset

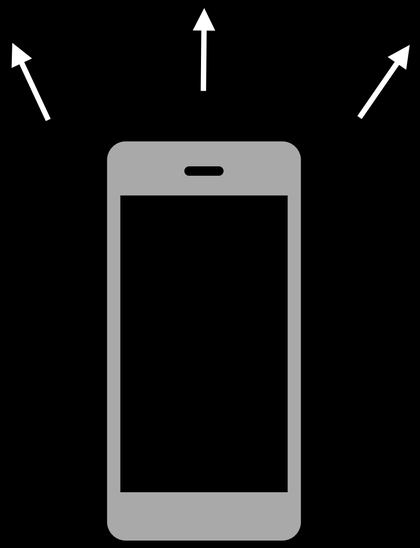
Contract B

An NFT collection

Contract D

DAO with many identities and two tokens + NFTs

WTF?

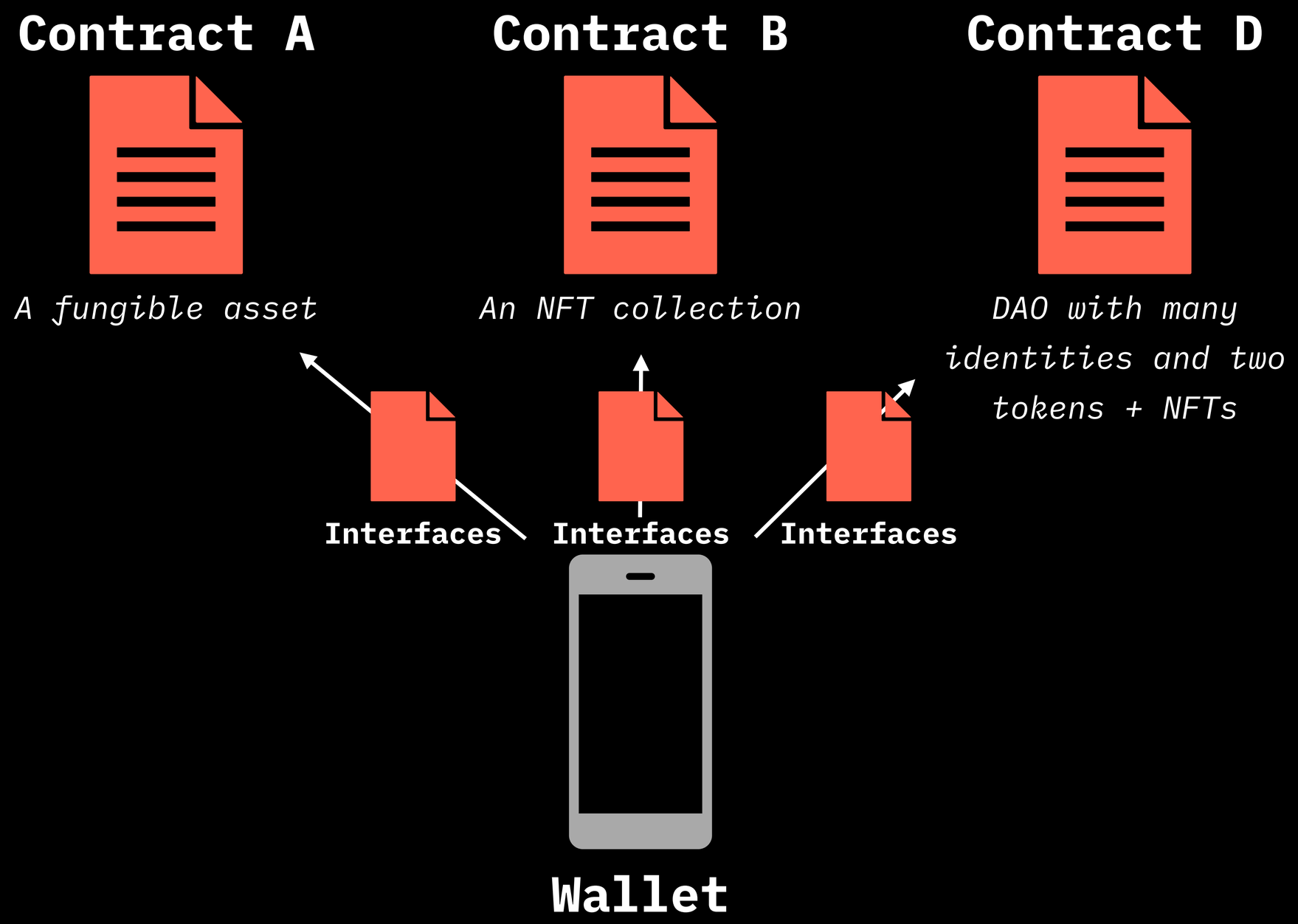


Wallet

Interface

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

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

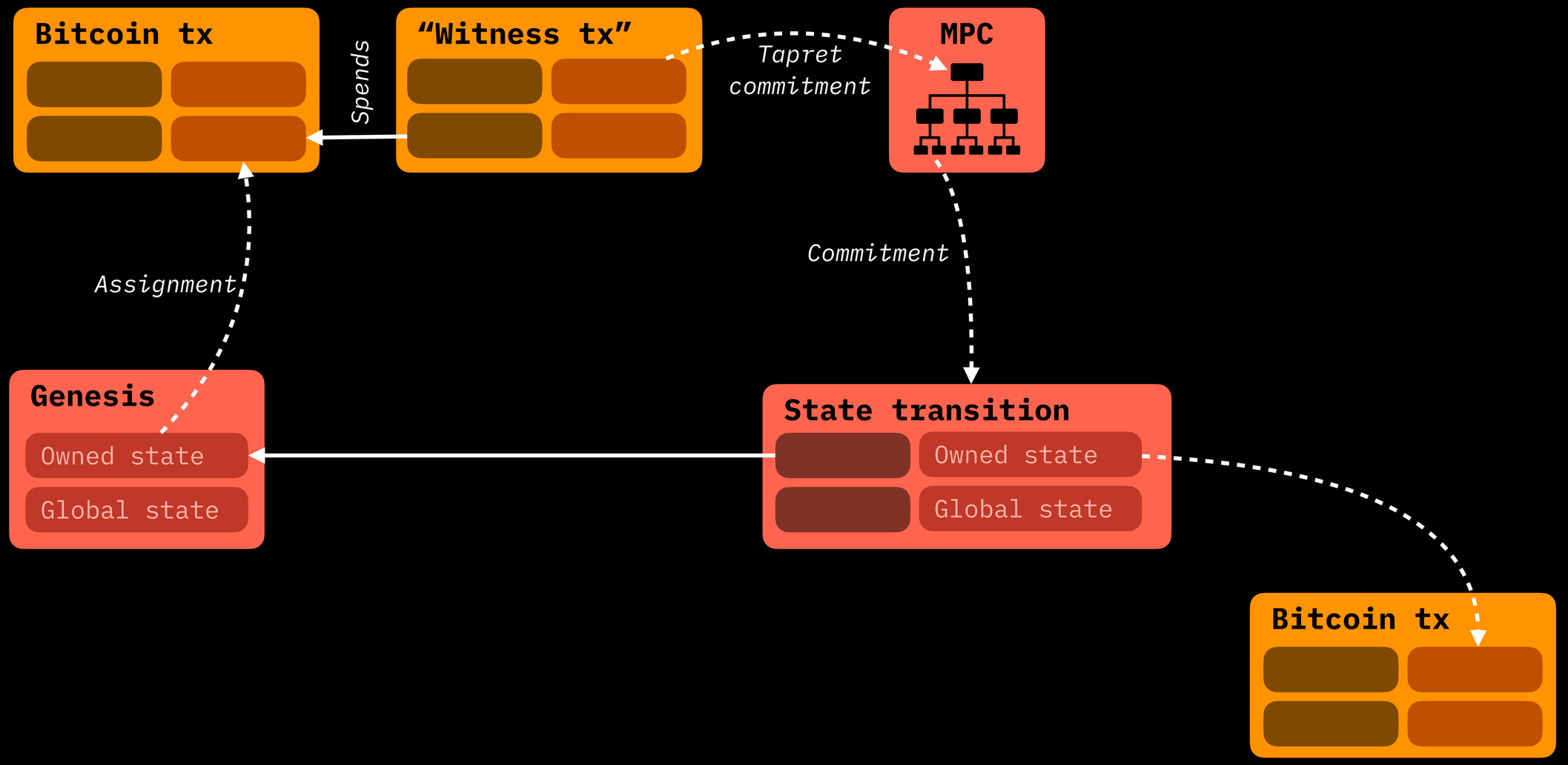

Contract A
Contract B
Contract D
A fungible asset
An NFT collection
DAO with many identities and two tokens + NFTs
Interfaces
Interfaces
Interfaces
Wallet

Current interfaces by LNP/BP Association

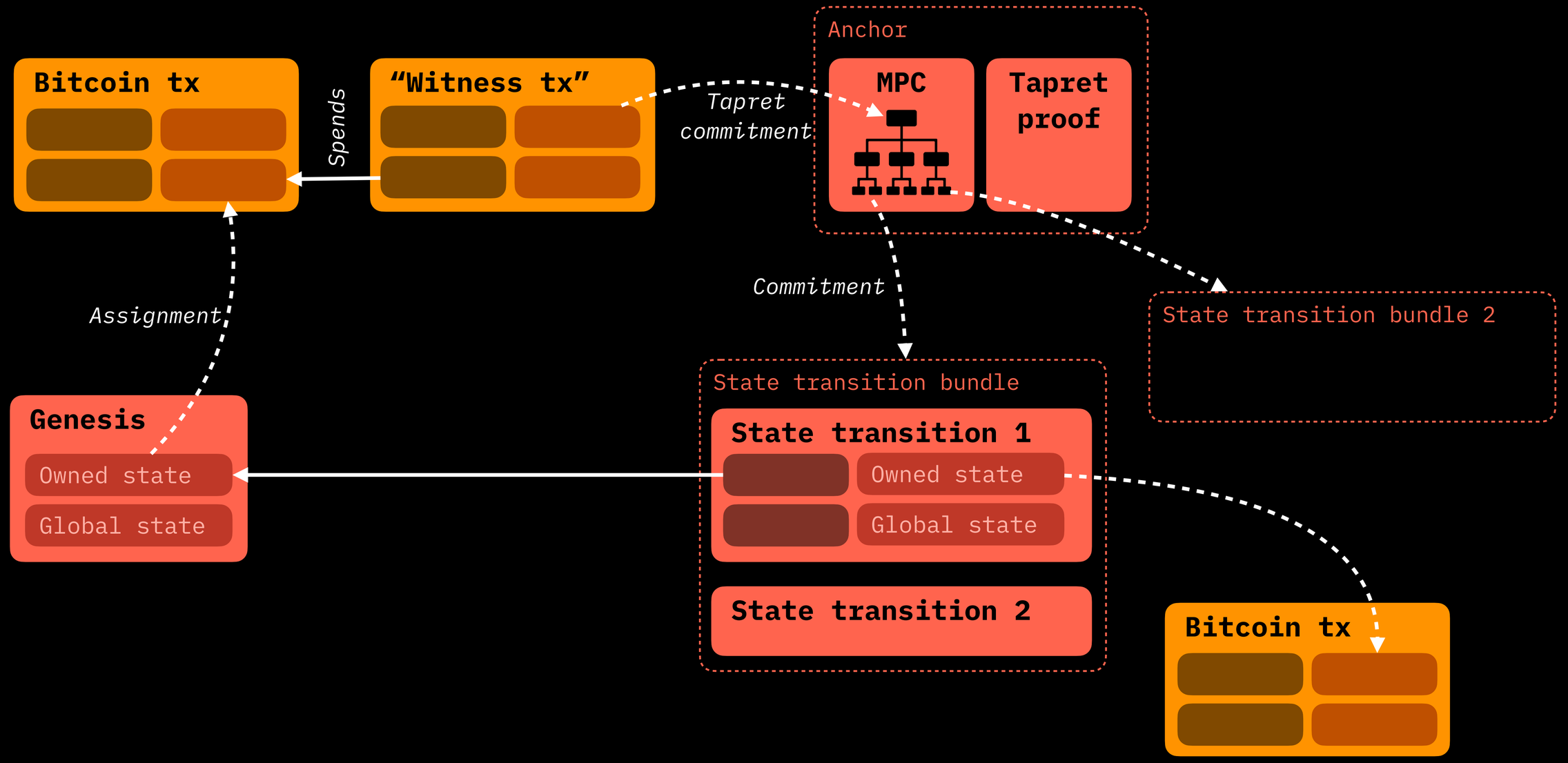
- **RGB20**: fungible assets - like company shares or shitcoins
- **RGB21**: digital collectibles - like NFTs, books, music, shitposts & stupid memes - but without wasting blockspace (!) like in ordinals
- **RGB22**: digital identity
- **RGB23**: provable history of operations (OpenTimeStamps squared)
- **RGB24**: global domain name system (like ENS, but much better)

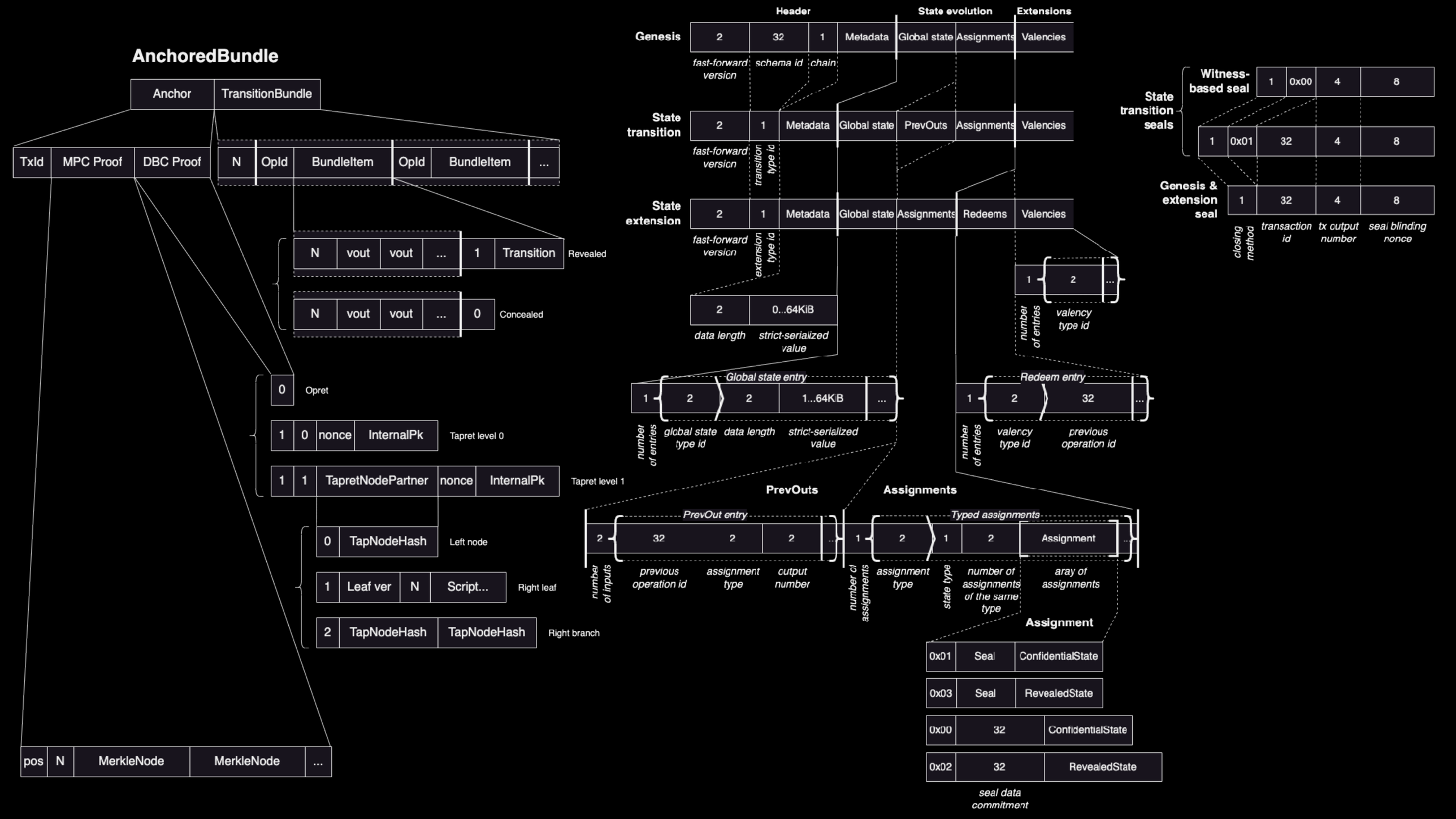
These interfaces are created by us and shipped with RGB standard library. But anyone can create their own interface without the need to contact us!

Anatomy of RGB operation



Anatomy of RGB operation




AnchoredBundle
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
...
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
PrevOut entry
2
32
2
2
...
number of inputs
previous operation id
assignment type
output number
Assignments
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

RGB

GET STARTED

Post-blockchain smart contracts

Why RGB?

Scalability

RGB can scale in terms of transaction throughput, data size and network size. It doesn't keep any data on-chain (in any blockchain); it is sharded from the day 1 and is fully interoperable with layer 2 scalability solutions.

Privacy

No chain analysis is possible due to the absence of transaction graph in blockchain. RGB uses zero-knowledge to protect the history of a fungible state. With RGB, user is always in-charge what and when to disclosure parts of the history and state, if needed.

Bitcoin & Lightning

RGB is a native member of Bitcoin and Lightning network ecosystem, brining rich smart contracts in a scalable way to the World's most secure and censorship-resistant cryptocurrency.

Build with RGB

RGB was designed to allow everything what is possible with blockchain-based smart contracts (like in Ethereum and other systems) – but in the scalable, robust and private way. With RGB, you can do the following categories of smart contracts (and much more):

Tokens

NFTs

DeFi

DAOs

RGB Blackpaper

Turing-complete, Scalable & Confidential Smart Contract Layer for Bitcoin & LN

Maxim Orlovsky 1,2; Peter Todd 1; Giacomo Zucco 1; Federico Tenga 3; Olga Ukolova 1,2

1.LNP/BP Standards Association

2.Pandora Prime

3.iFinex Inc

Abstract

This paper proposes a novel "post-blockchain" smart contract system, named RGB. It is based on the concept of client-side-validation, separating contract state and operations from the consensus level. With this approach, contracts are sharded (each contract is a standalone shard), kept and validated only by contract participants, bringing native privacy and scalability mechanism, exceeding abilities of all existing blockchain-based smart contracts while not compromising on security or decentralization. RGB as a smart contract system doesn't require any specific token/coin to operate and is implemented on top of bitcoin blockchain, as the most secure decentralized system. It also can operate on top of layer-2 protocols, such as sidechains, lightning network and other future protocols. It is fully compatible with all existing bitcoin technologies (scriptless scripts, DLCs, atomic swaps) and future possible bitcoin softforks and doesn't require any changes to the base bitcoin layer. RGB uses specially-designed functional registry-based RISC virtual machine AluVM, which is Turing-equivalent* and is able to operate global state with the same availability guarantees as with existing blockchain-based systems. RGB has a strong privacy-preserving emphasize, using modified form of Blockstream's confidential transaction technology (based on Pedersen commitments, enhanced with Bulletproofs++ range proofs) and cryptographic hash concealments for non-fungible state, such that even contract participants do not see full information about past contract history - while still be able to validate it. The paper presents an initial implementation of RGB technology and discusses possible applications of it for building rich, private, scalable and censorship-resistant applications on top of bitcoin and lightning network, including bitcoin finance ("BiFi") and non-financial forms of smart contracts.

* in the same terms as EVM and WASM-based smart contracts, i.e. nearly computationally universal, bound by number of operation steps, measured by gas consumption in Ethereum-like systems, and by accumulated computational complexity measure in case of AluVM.

We are the largest Bitcoin tech non-profit in Switzerland

github.com/LNP-BP

Search or jump to... / Pull requests Issues Marketplace Explore

LNP/BP Association

Non-profit supervising layer 2 & 3 protocols on Bitcoin & Lightning Network

Bitcoin https://lnp-bp.org @lnp_bp info@@lnp-bp.org

Unfollow

Overview Repositories 43 Packages People 22 Teams 5 Projects 13 Settings

.github / profile / README.md

LNP/BP[1] Standards Association

We are Swiss non-profit **supervising layer 2 & 3 open standards and protocols for Bitcoin & Lightning Network**. We are creators of L2 and L3 protocols like RGB, Bifrost, Storm, Prometheus, Kaleidoscope and active builders of **#BiFi** (bitcoin finance) ecosystem on Lightning. We manage set of LNPBP standards and their opensource reference implementations under permissive MIT & Apache2 licenses. The Association was founded by @dr-orlovsky and @giacomozucco in 2019. You can read more about us on our website, lnp-bp.org and follow us on Twitter @lnpbp.

LNPBP Standards

The current list of standard can be found here. You can:

- submit a new standard proposal
- discuss preliminary ideas about new standards
- follow announcements about standard releases
- write about your implementation of one of the standards
- ask questions
- peer review & audit existing standards

People

View all

Invite someone

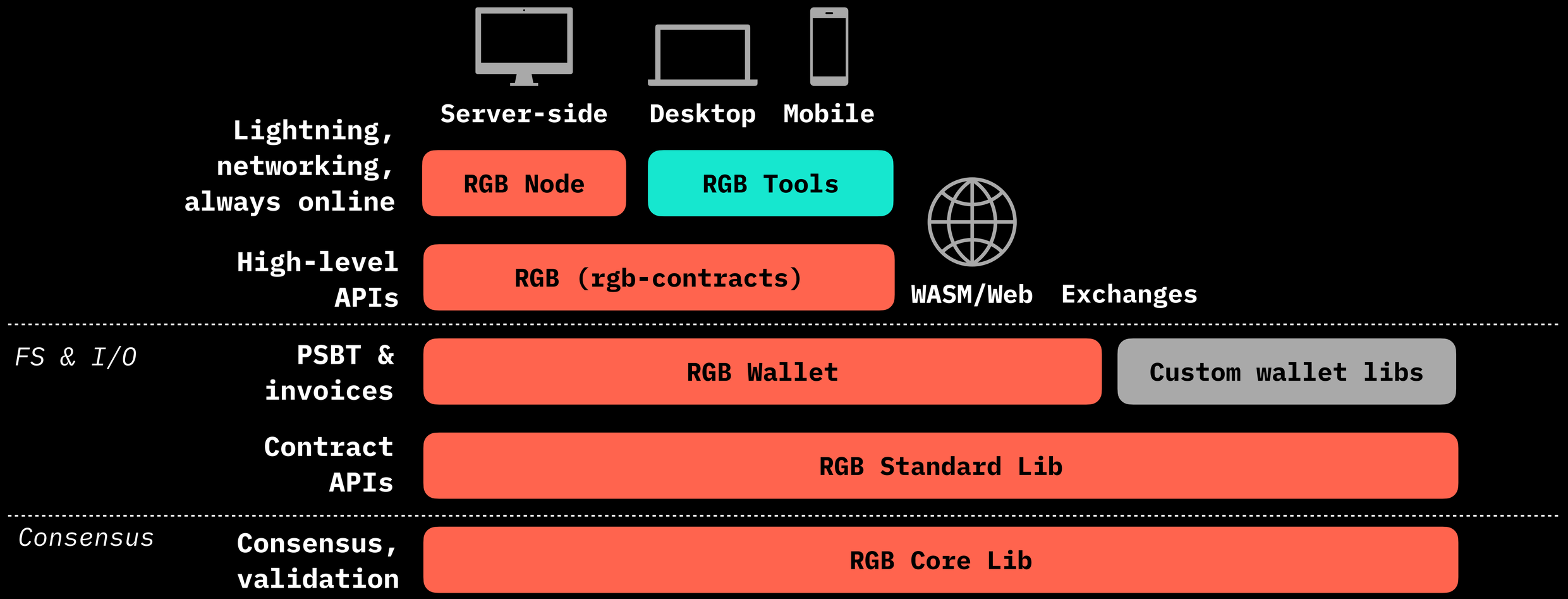
Top languages

Rust C Python Swift Dockerfile

Most used topics Manage

bitcoin client-side-validation lnp-bp lightning-network distributed-systems

RGB library stack for app devs



RGB on GitHub

rgb Public

RGB smart contracts: client-facing library & command-line for desktop and mobile

Rust 1 2

rgb-node Public

RGB node - the official server-side implementation

Rust 108 36

rgb-wallet Public

RGB wallet & standard libs for web & low-level integrations

Rust 9 9

rgb-core Public

RGB Core Library: consensus validation for private & scalable client-validated smart contracts on Bitcoin & Lightning

Rust 117 23

blackpaper Public

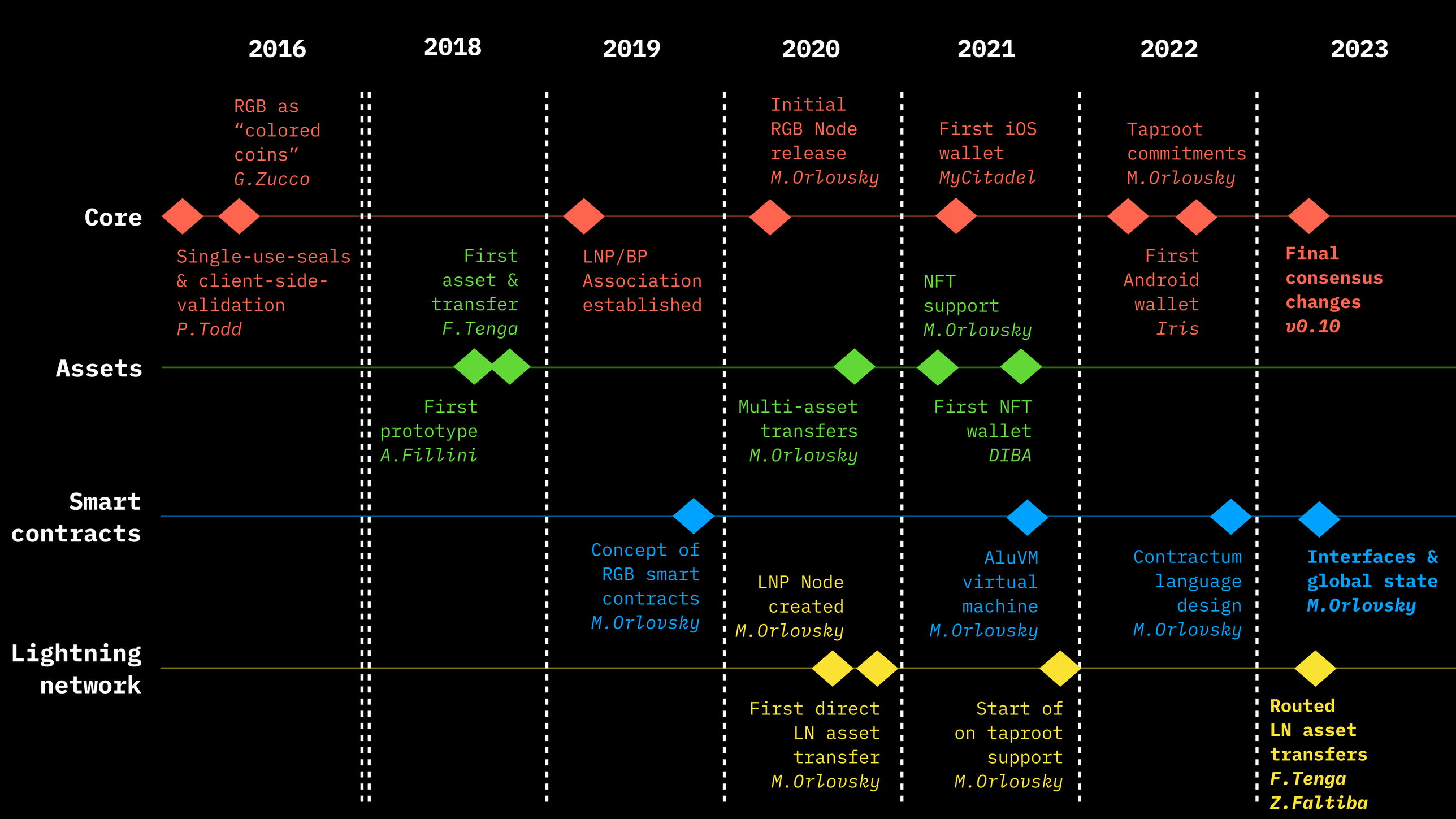
RGB blackpaper

2 1

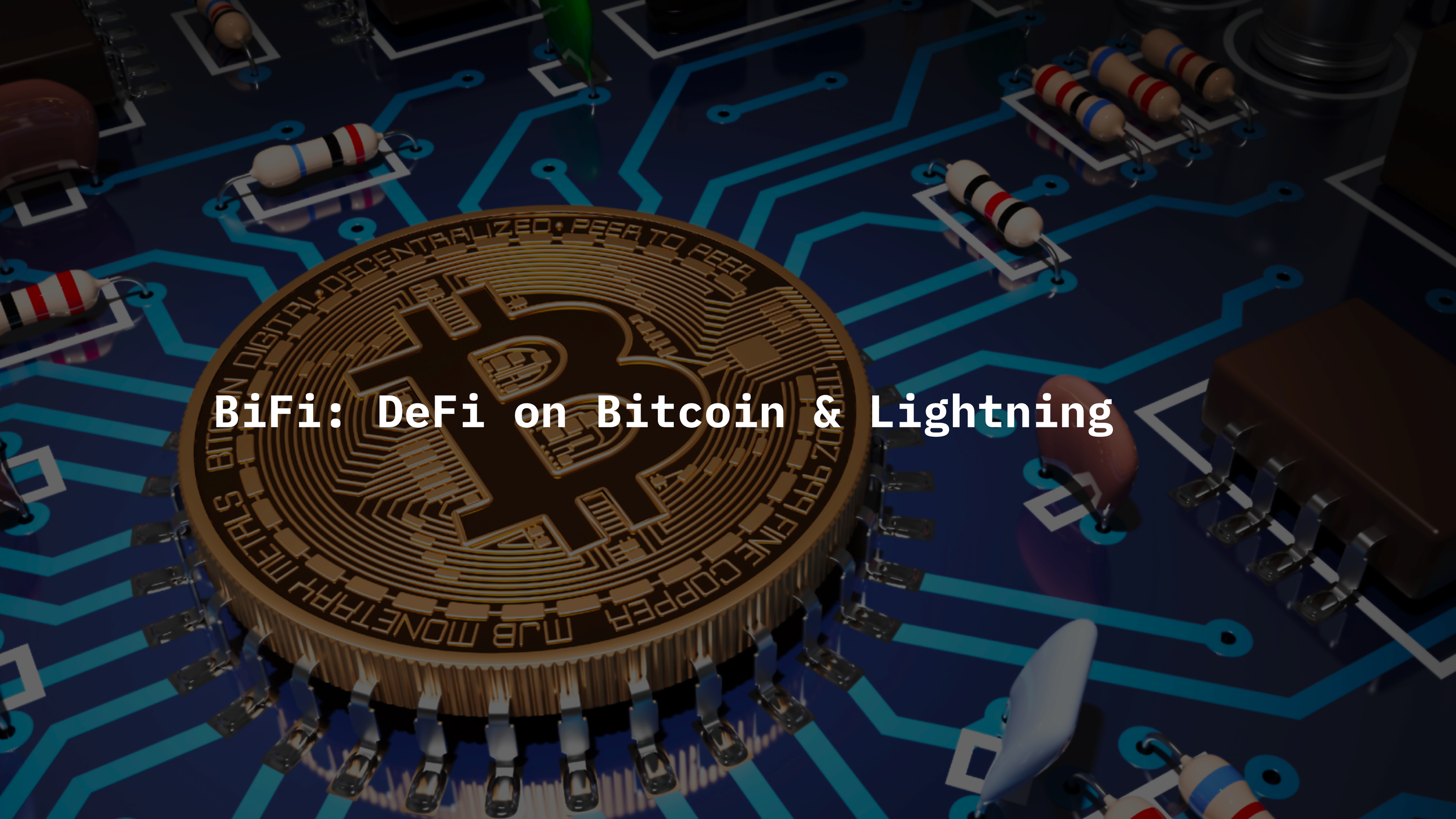
contractum-lang Public

Contractum: RGB contract definition language

25 1


2016
2018
2019
2020
2021
2022
2023
Core
Assets
Smart contracts
Lightning network
RGB as "colored coins" G.Zucco
Single-use-seals & client-side-validation P.Todd
First asset & transfer F.Tenga
First prototype A.Fillini
LNP/BP Association established
Concept of RGB smart contracts M.Orlovsky
Initial RGB Node release M.Orlovsky
Multi-asset transfers M.Orlovsky
LNP Node created M.Orlovsky
First direct LN asset transfer M.Orlovsky
First iOS wallet MyCitadel
NFT support M.Orlovsky
First NFT wallet DIBA
AluVM virtual machine M.Orlovsky
Start of on taproot support M.Orlovsky
Taproot commitments M.Orlovsky
First Android wallet Iris
Contractum language design M.Orlovsky
Final consensus changes v0.10
Interfaces & global state M.Orlovsky
Routed LN asset transfers F.Tenga Z.Faltiba

RGB tomorrow



BiFi: DeFi on Bitcoin & Lightning

~~DeFi~~

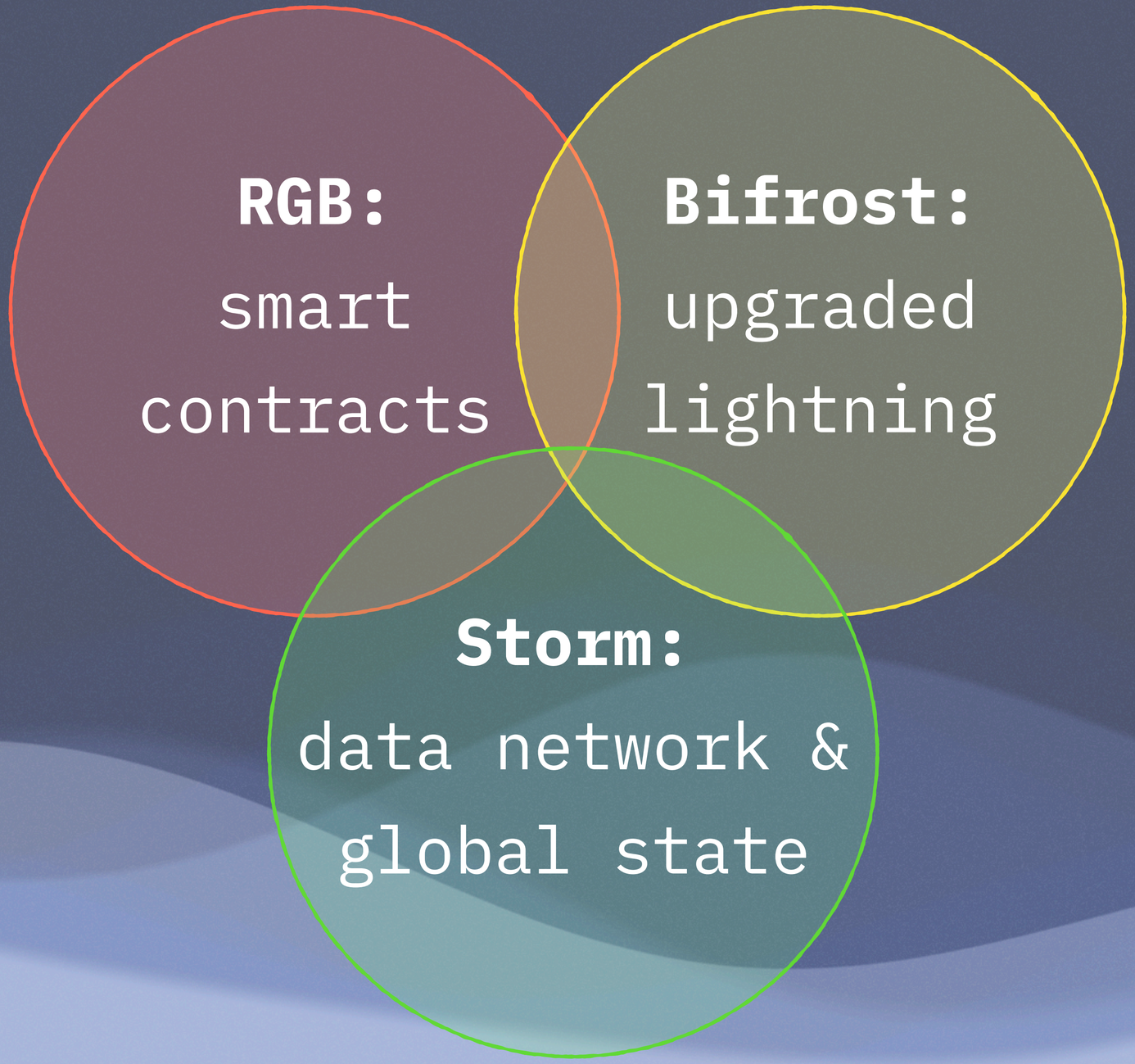
#BiFi

Bitcoin finance

- DEX
- Liquidity pools
- future & option contracts
- AMM & algorithmic stable coins

Based on

- Bitcoin
- RGB programmable assets
- DLC oracles
- Lightning network

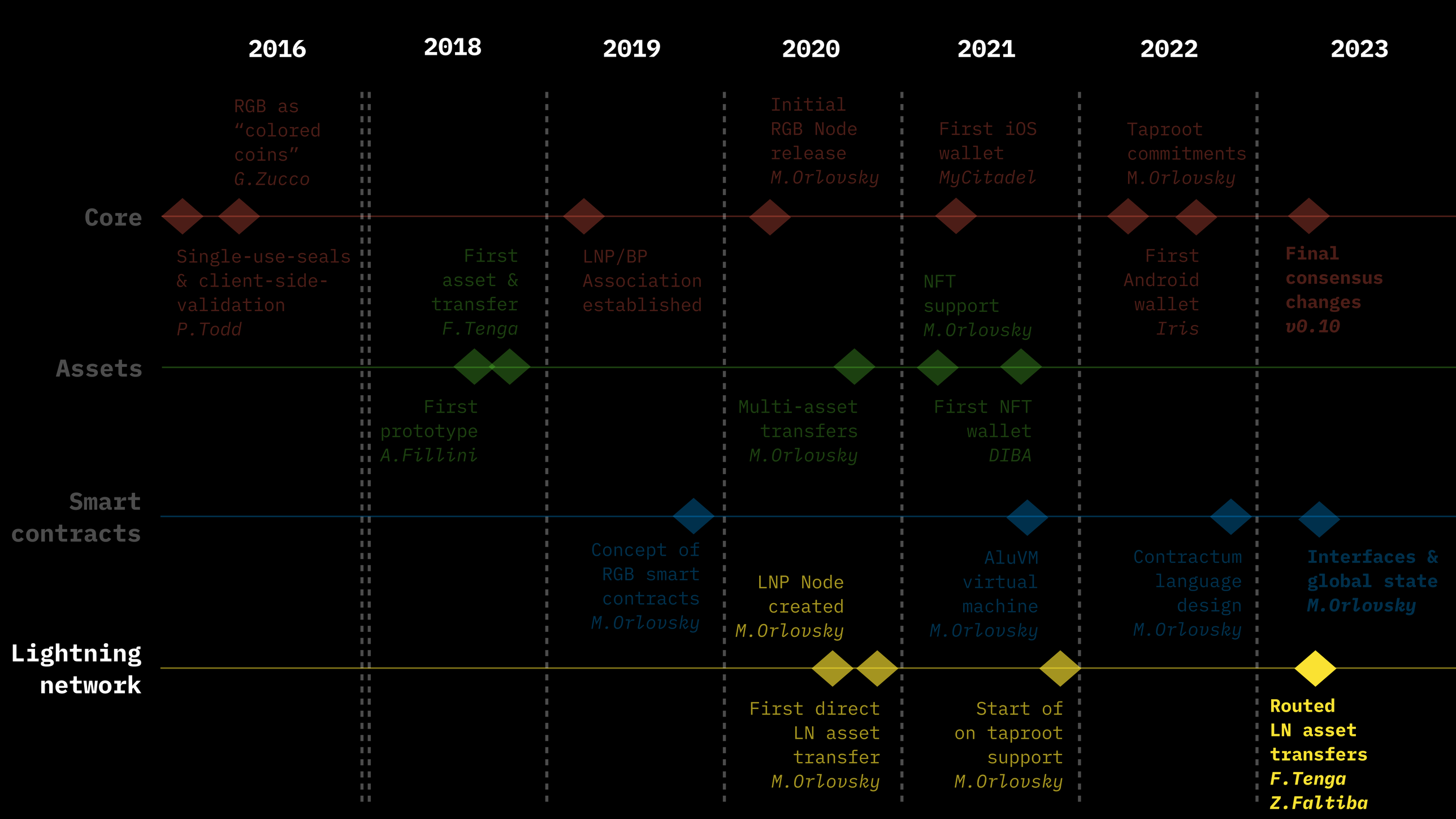

RGB:
smart
contracts
Bifrost:
upgraded
lightning
Storm:
data network &
global state

Contractum language

```
schema DecentralizedIdentity
   owned Identity :: PgpKey
   owned IOYIssue :: Zk64
   -- `Zk64` means 64-bit unsigned integer hidden with zero-knowledge
   owned IOYTokens :: Zk64

   global IOYTicker :: String
   global IOYName :: String

   genesis :: Identity, IOYTicker, IOYName

   op Revocation :: old Identity -> new Identity

   op Promise :: used IOYIssue -> given [IOYTokens]?, remaining IOYIssue?
      assert used == sum given + (remaining ?? 0)

   op Transfer :: spent {IOYTokens} -> received [IOYTokens]
      assert sum spent == sum received
```

```
interface PgpIdentity
  owned Identity :: PgpKey
  exec Revocation :: old Identity -> new Identity

implement PgpIdentity for DecentralizedIdentity
  -- we do not need to put anything here since schema state and operation
  -- names matches interface requirements and the compiler is able to guess
  -- the bindings
```

```
interface FungibleToken:
   global Ticker -> String -- this is similar to schema definition; in fact
                           -- it is a requirement that the schema must provide
                           -- a global state of the String type and link it to
                           -- the "Ticker" name
   global Name -> String

   owned Inflation :: Zk64 -- pretty much the same applies to assigned state
   owned Asset :: Zk64

   op Issue :: Inflation -> [Asset]?, Inflation? -- and operations
   op Transfer :: {Asset} -> [Asset]

-- Specific schema state may use different naming, for instance because a
-- schema can define multiple assets with different names; in that case we
-- will have multiple interface implementations referencing different state.
implement FungibleToken for DecentralizedIdentity
   global Ticker := IOYTicker -- this creates a _binding_ of the state defined
                              -- in the schema (*IOYTicker* in this case) to
                              -- the interface
   global Name := IOYName
   owned Inflation := IOYIssue
   owned Asset := IOYTokens
   op Issue := Promise
   op Transfer -- here we skip `:=` part since the interface operation name
               -- matches the name used in the schema. In such cases we can
               -- also skip the declaration at whole
```


2016
2018
2019
2020
2021
2022
2023
Core
Assets
Smart contracts
Lightning network
RGB as "colored coins" G.Zucco
Single-use-seals & client-side-validation P.Todd
First asset & transfer F.Tenga
First prototype A.Fillini
LNP/BP Association established
Concept of RGB smart contracts M.Orlovsky
Initial RGB Node release M.Orlovsky
Multi-asset transfers M.Orlovsky
LNP Node created M.Orlovsky
First direct LN asset transfer M.Orlovsky
First iOS wallet MyCitadel
NFT support M.Orlovsky
First NFT wallet DIBA
AluVM virtual machine M.Orlovsky
Start of on taproot support M.Orlovsky
Taproot commitments M.Orlovsky
First Android wallet Iris
Contractum language design M.Orlovsky
Final consensus changes v0.10
Interfaces & global state M.Orlovsky
Routed LN asset transfers F.Tenga Z.Faltiba