Compliant Token Architecture Decisions

When designing the Compliant Token, we are facing an important design decision.
We can either enforce our token's transfer constraints

On-Chain,

Off-Chain, or in a

Hybrid manner.

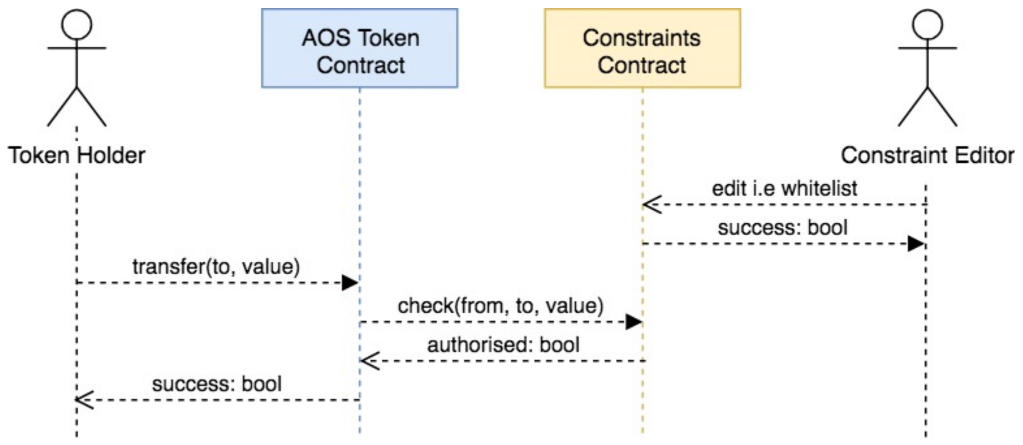
---

1. On-Chain

Editors use private key and i.e. Metamask to directly interact with the Constraint Contract
Transfer constraints are enforced on-chain by the Constraints Contract

+ All security is handled by the private/public-key infrastructure of the blockchain
+ Transfers happen in 1 transaction

- Too many constraints could inflate the average token transaction cost



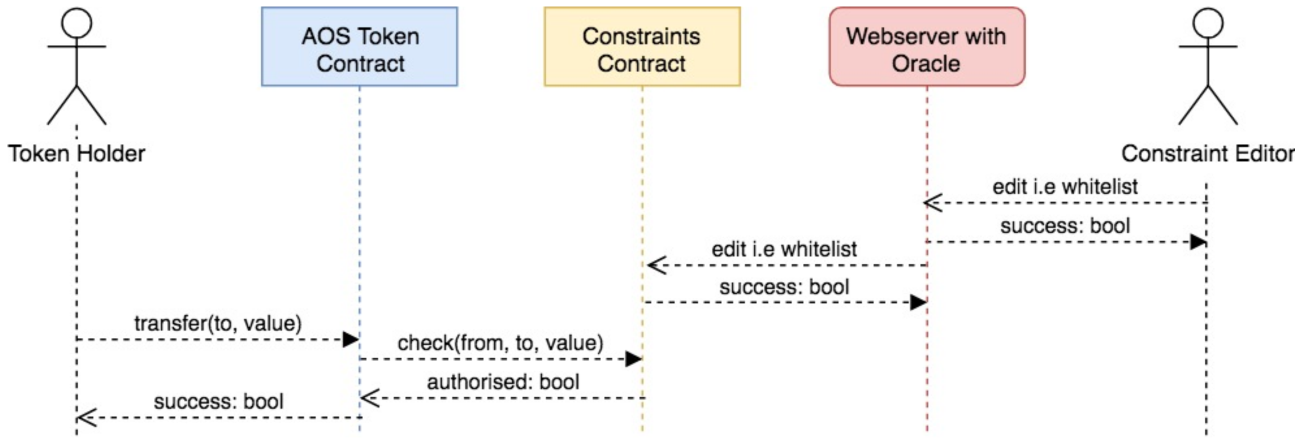
Simon Dosch sd@micobo.com

2. Hybrid

Transfer constraints are enforced on-chain by the Constraints Contract.
Editors use classic password authentication to interact with a website & database off-chain.
The server then acts as an Oracle and pushes i.e. the whitelist constraints into the Constraints Contract

- + Editors do not have to familiarize themselves with Web3.0 technology
- + Transfers happen in 1 transaction

- - Too many constraints could inflate the average token transaction cost
- - Updates not instant / possible redundancy issues between database and blockchain
- - Server operator needs to be funded and trusted

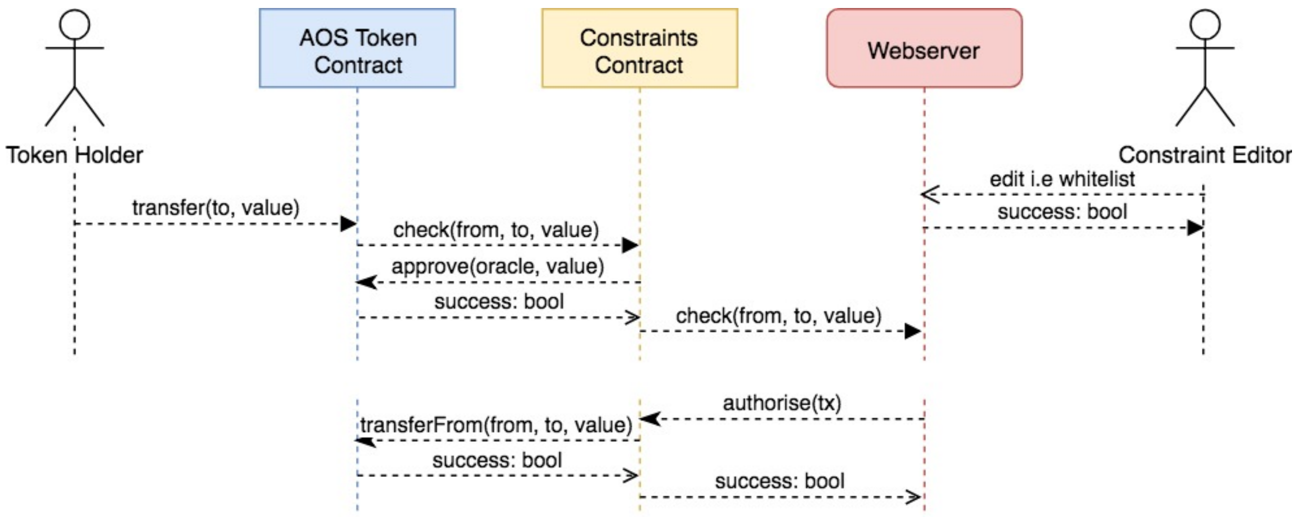


3. Off-Chain

Editors use classic password authentication to interact with a website & database off-chain.
Transfer constraints are enforced off-chain by a program running on a server.
The transfer process has to be split up into two interactions, where transactions are first "queued" (approved) and then authorized and executed.

+ Editors do not have to familiarize themselves with Web3.0 technology
+ More constraints can be checked for, as off-chain computing is not expensive

- Server operator needs to be funded and trusted
- Transfers do not happen in 1 transaction but need to be kept in smart contract storage while waiting for authorisation (this will get complicated!)
- Bloated transfer process



Simon Dosch sd@micobo.com