CAST CHALLENGE

Use Case 2: Delivery Versus Payment

CADMOS

July 2023

CONTENT

- Key Design Drivers
- Assumptions
- Implemented features

Key Design Drivers

Customizable and Compatible

Why?	Flexibility to adapt to a changing regulatory framework and agnosticism to a given token implementation.
Implications:	set assumptions to a minimum level, adding specific features through a modular 'satellite-driven' approach, use standardized interfaces.

Atomic

Why?	Instant 'DVP' is one of the most promising use-case for blockchain based capital market, as it enables features such as *flashswaps* (no more cash needed on the broker side and no counterparty risk).
Implications:	assumptions on asset and cash technical set-up.

Secure

Why?	Business Continuity Plan and privacy improved as well as operational risk limited thanks to public blockchain capabilities.
Implications:	framework as much as possible on-chain and use of solidity best practices.

Thrifty

Why?	Ensure scalability through low transaction costs and focus on operational simplicity.
Implications:	trade-off between architecture complexity and transaction costs.

Assumptions

Atomicity: DvP is contained in one transaction - if D or P fails, the whole transaction is cancelled

I - Cash and Security Token can be atomically transferred

- Transfer of the security token and of the cash token can be executed within one blockchain transaction
- If either the transfer of the cash token or of the security token fails, then the whole transaction is cancelled

⇒ Cash Token compatibililty:

✓ Standard ERC-20/StableCoin

✓ So-Cash

? EUR-CV (can be done depending on EURCV implementation)

II - Cash and Security Token are on the same EVM-compatible chain

DVP is atomically done in a single transaction, another setup (e.g. HTLC) needed for cross-chain settlements

Implemented feature
satellite contracts

Question: how to embed a Forex operation within the DVP?

I – Satellite contracts to add constraints and capabilities

- The DvP smart contract itself has minimal constraints offering a wide applicability
- Additional constraints and arbitrary business logics can be added using delegation to optional satellite smart contracts
- Satellites are the technical representation of legal agreements and business logic modules.
- Atomicity is maintained, as the executions of satellite modules is bundled into the atomic DvP transaction
- If the execution of one module fails, the whole DvP transaction is cancelled
- Example of satellite contracts:
  - Forex module (implemented in our code) to allow trades involving cross currency DvPs
  - Compliance module, for any additional compliance or regulatory check such as security token eligibility for a given cash token (cf. Coinvertible),
  - Flashswaps to aggregate different DvPs: multiple counterparties, or broker-intermediated transactions with no capital requirement from the broker

Implemented feature
On chain Metadata(1/2)

II - Trade data broadcasted on chain and encrypted to leverage blockchain properties (implemented in our code)

- Metadata is published on-chain (as a log) during payment initialization.

- We ensure privacy by making Metadata accessible only to those holding the decryption key, that can be attributed to any stakeholder (counterparties, brokers, custodians, regulators etc.).

- Technically, the metadata is encrypted using a hybrid cryptosystem (inspired by PGP):
  - First the data is symmetrically encrypted (AES)
  - We then asymmetrically encrypt it using the stakeholders' public keys (obtained from ETH addresses).

- Buyer and Seller must validate the metadata for the DvP to take place.

- Stakeholders can maintain an archive node on the chosen network and keep local backups for Business Continuity Plan Purpose

- Advantages:
  - Immutable Audit Trail (Metadata cannot be changed)
  - Metadata is directly published by the DvP smart contract and do not rely on an external system

Implemented feature
On chain Metadata(2/2)

Putting data on chain: what about the gas cost?

Example of metadata set

```
Buyer - Name - Physical Address - LEI
Seller -  Name - Physical  Address - LEI
Asset - Asset EVM Address
Cash - EVM Cash Address
Blockchain Id (‘chain id’)
Quantity
Price
Time
20 - MT202
21 - MT202
```

On-Chain Metadata storage options:

- Smart Contract Storage (SSTORE):
  - Gas Cost ~20’000/byte
  - ✓ Can be queried by another smart contract
- Smart Contract Binary (SSTORE2):
  - Gas Cost ~ 35’000 + 205/byte
  - ✓ Can be queried by another smart contract
- Event Log:
  - Gas Cost ~ 375 + 8/byte
  - Estimation at current ETH price : 0.18$
  - ✗ Cannot be queried by another smart contract

**We implemented compression with zlib to save around 30% in size and gas*

Implemented feature
Clone Factory Pattern

III – Cheap deployment of DvP smart contract to improve scalability (implemented in our code)

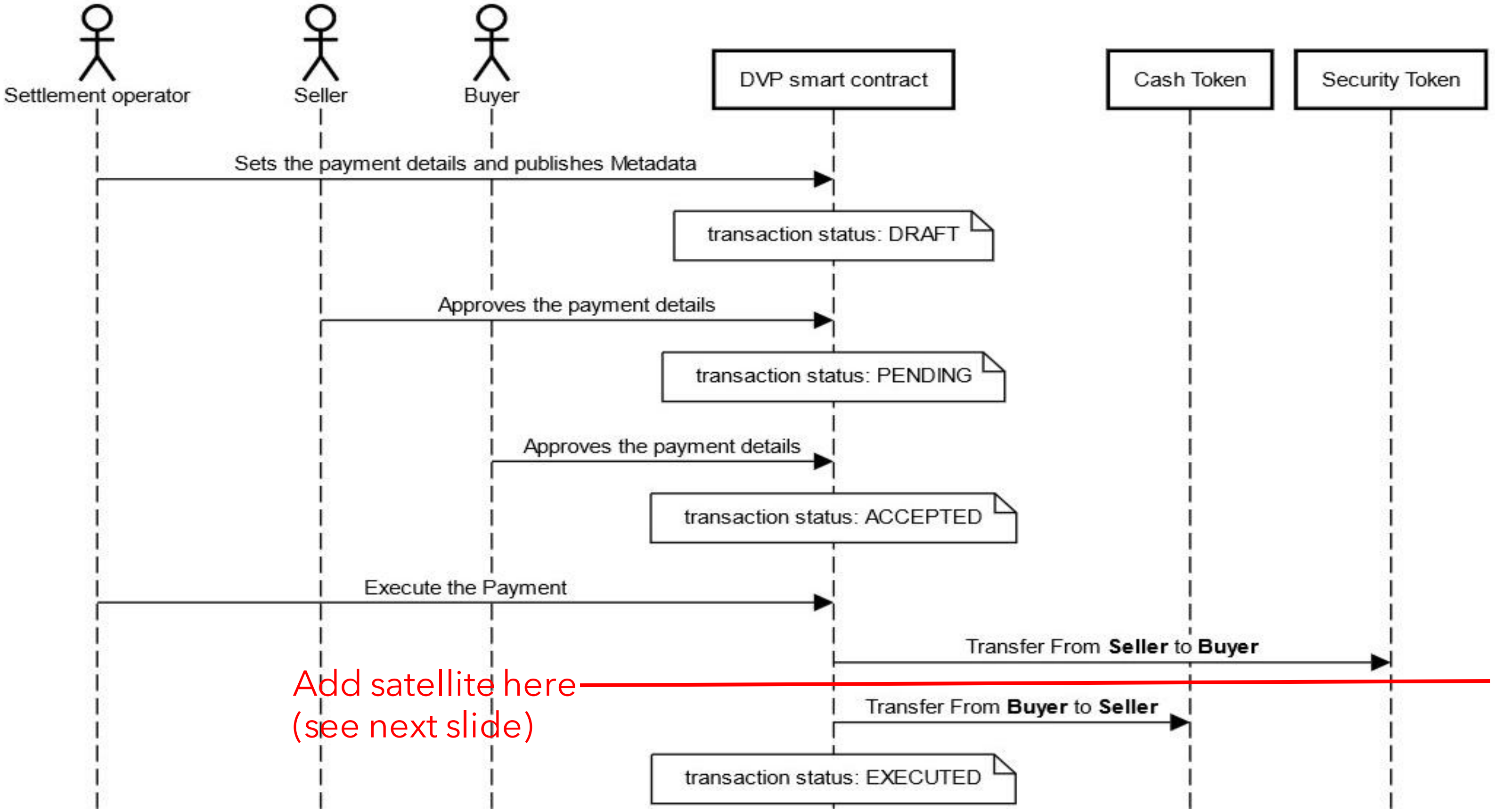
- Deploying the complete DVP smart contract for each operation is relatively expensive :
  - at current ETH price, contract deployment costs 77.33$
- A better solution is to use a "clone-Factory pattern" as it is:
  - cheaper since we only deploy a proxy to the DvP implementation instead of the full logic.
  - at current ETH price, proxy deployment costs 4.61$
  - more Secure with regards to bytecode whitelist (see so-bond contract whitelisting scheme) since we do not use a constructor.

Simple Atomic DVP

Example: buyer wants to buy so-bond from seller and pays in so-cash

Starting point:

- Buyer must approve in advance the DVP smart contract to spend its cash.
- If cash is so-cash then we suppose that the required banking relationships have been established and the needed pre-approvals made.



Settlement operator

Seller

Buyer

DVP smart contract

Cash Token

Security Token

Sets the payment details and publishes Metadata

transaction status: DRAFT

Approves the payment details

transaction status: PENDING

Approves the payment details

transaction status: ACCEPTED

Execute the Payment

Transfer From **Seller** to **Buyer**

Transfer From **Buyer** to **Seller**

transaction status: EXECUTED

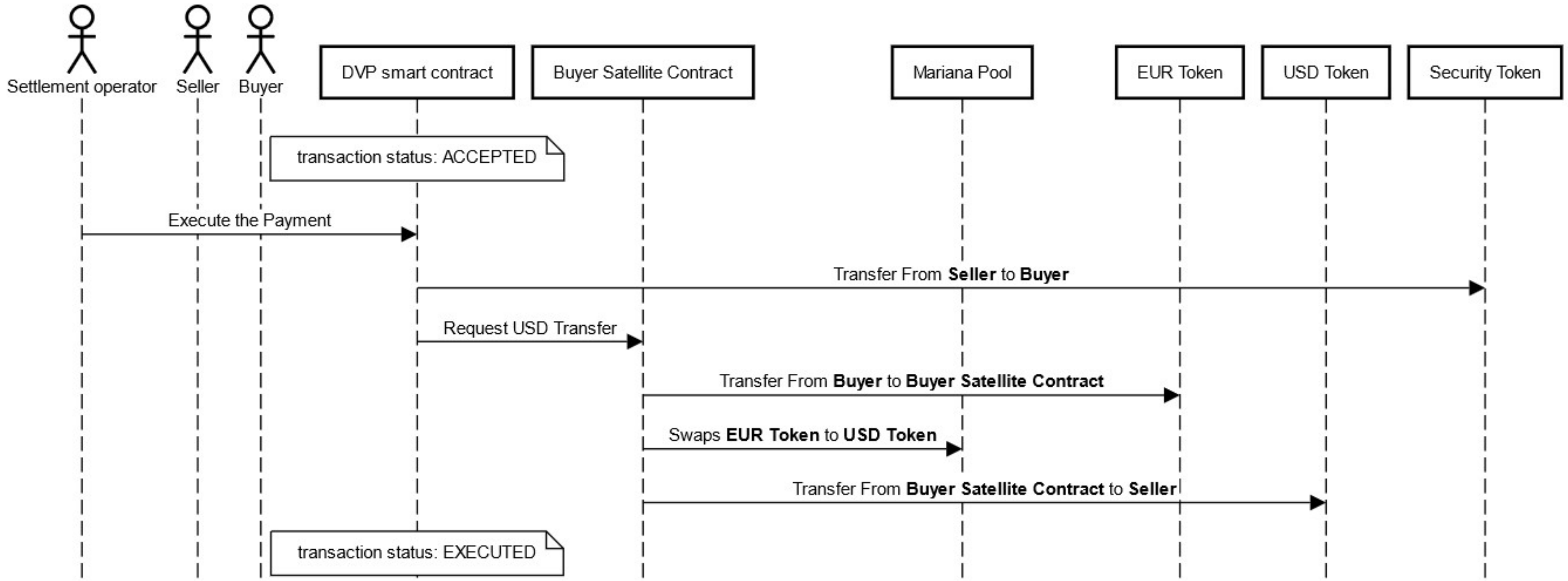
Add satellite here
(see next slide)

Notes:

- DvP smart contract is built upon so-bond Bilateral trade smart contract, where we added the Settlement Operator role (see CAST framework).
- Seller or Buyer can initiate the trade.
- Seller or Buyer can be Settlement Operator, in which case "execute" does not need to be called
- Trade Approval includes a front-running protection mechanism to avoid last-minute changes in the Trade details.
- DvP smart contract computes the required cash payment according to the trade token quantity and unit price.

Atomic DVP + FX

Setup: Buyer wants to pay EUR and Seller wants to receive USD; FX is intermediated by an Automated Market Maker (see e.g. BIS 'Mariana Pool' whitepaper).

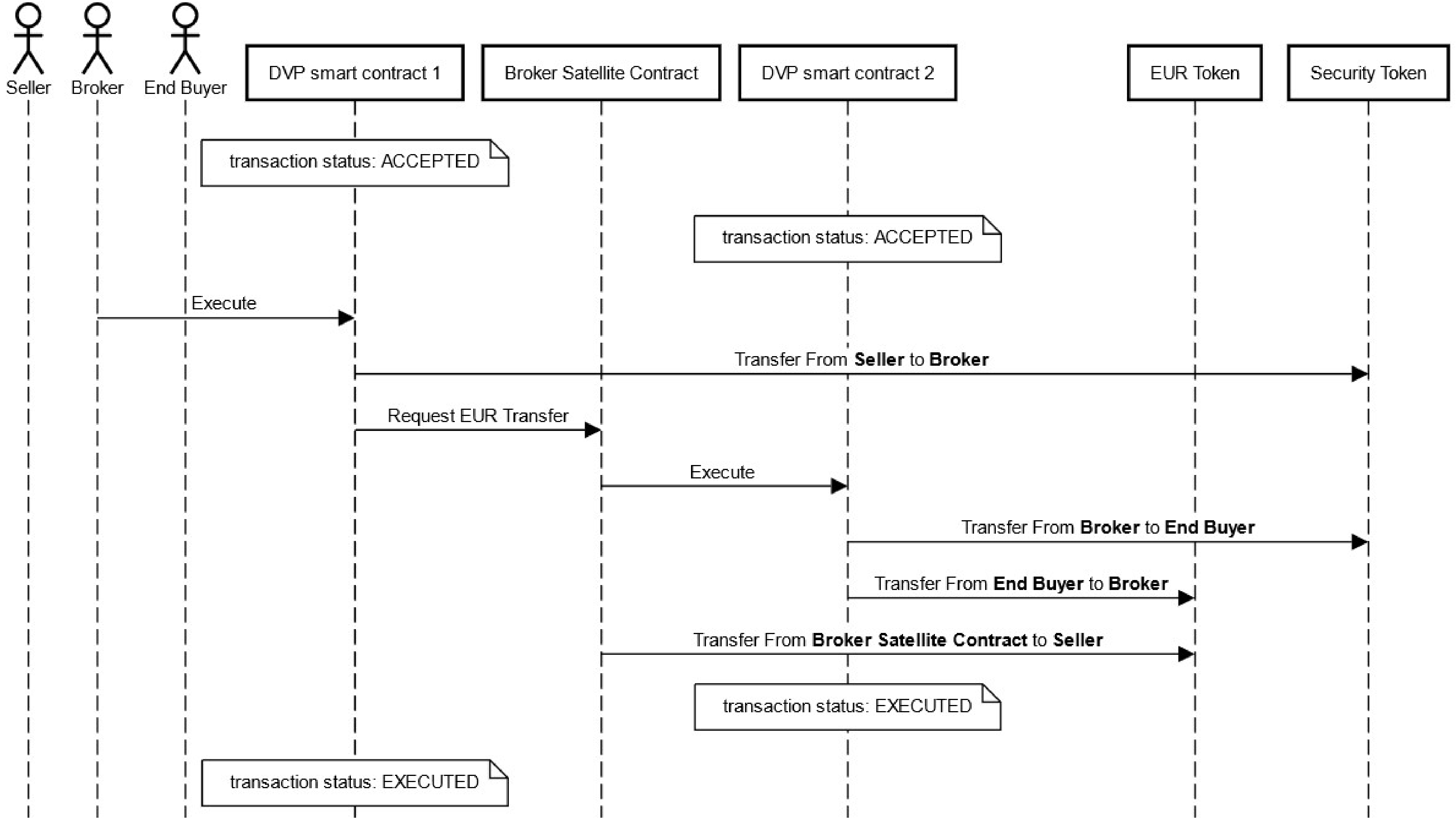

Settlement operator
Seller
Buyer
DVP smart contract
Buyer Satellite Contract
Mariana Pool
EUR Token
USD Token
Security Token
transaction status: ACCEPTED
Execute the Payment
Transfer From **Seller** to **Buyer**
Request USD Transfer
Transfer From **Buyer** to **Buyer Satellite Contract**
Swaps **EUR Token** to **USD Token**
Transfer From **Buyer Satellite Contract** to **Seller**
transaction status: EXECUTED

Notes:

- DVP smart contract must check that the required balance has effectively been transferred
- Instead of interfacing with an AMM, we could directly integrate another DVP smart contract.
- Transferring the security token 'before' the cash is akin to a *flashswap*: the buyer can atomically 'resell' the security token *within the same transaction* to free up the cash needed to pay the seller.

Annex: Flashswap DVP

Setup: Broker intermediates between the Seller and the End Buyer and takes a fee.
The two DVPs can atomically be chained so that the broker does not need to mobilize any capital.


Seller
Broker
End Buyer
DVP smart contract 1
Broker Satellite Contract
DVP smart contract 2
EUR Token
Security Token
transaction status: ACCEPTED
transaction status: ACCEPTED
Execute
Transfer From **Seller** to **Broker**
Request EUR Transfer
Execute
Transfer From **Broker** to **End Buyer**
Transfer From **End Buyer** to **Broker**
Transfer From **Broker Satellite Contract** to **Seller**
transaction status: EXECUTED
transaction status: EXECUTED

CONTACT

BOUERI Nassib
Nassib.boueri@cadmos.finance

FAYE Jonathan
Jonathanfaye@gmail.com

TURK Joseph-André
Josephandre.Turk@gmail.com

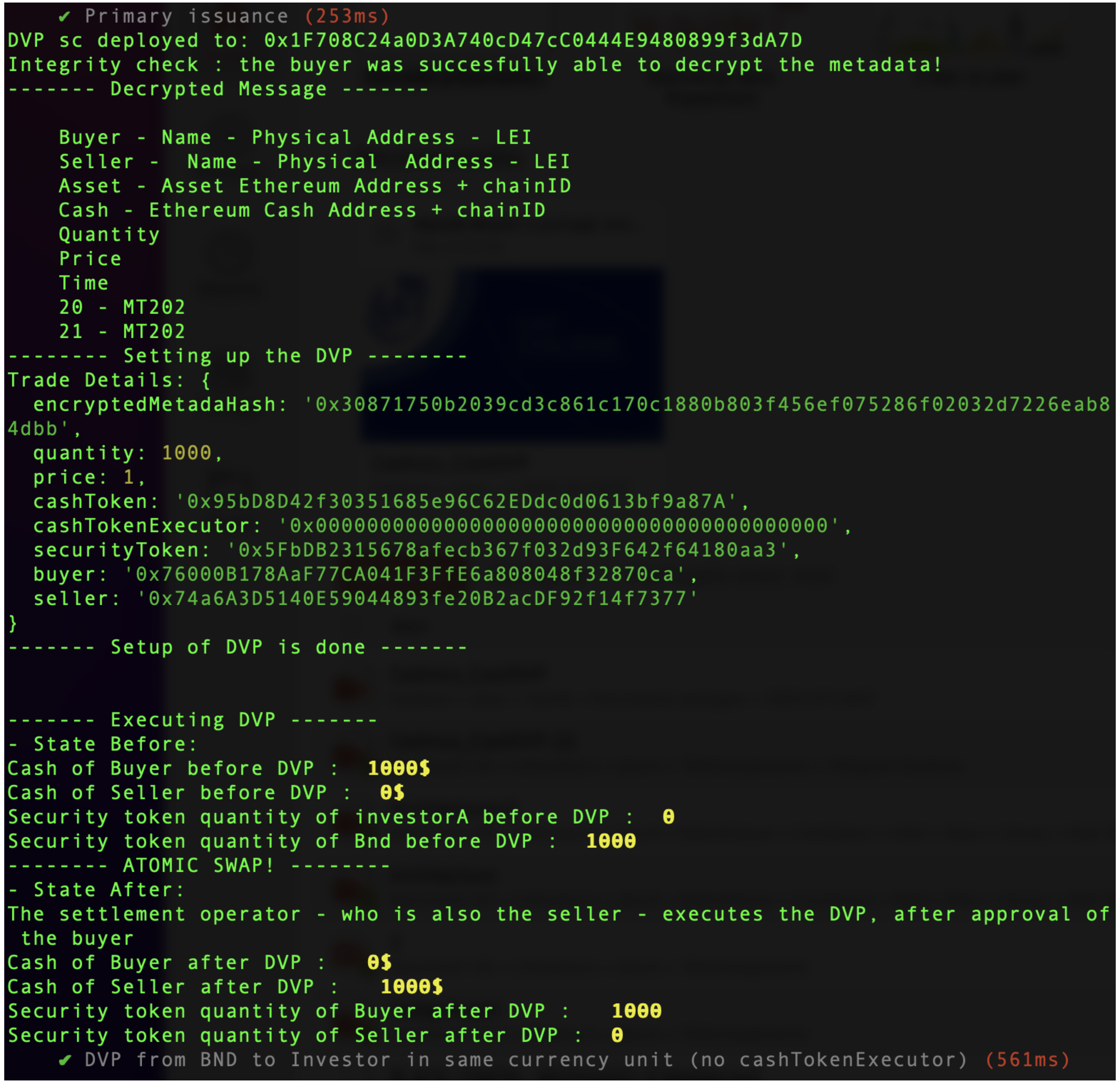
ABOUT CADMOS

CADMOS.FINANCE brings together powerful smart contracts and a scalable tokenization platform to offer a robust Infrastructure for Decentralized Asset Management, and Fundraising.

CADMOS.IO is a full-stack solution provider for the digital asset universe.

For more information visit www.cadmos.finance / www.cadmos.io

Simple Atomic DVP

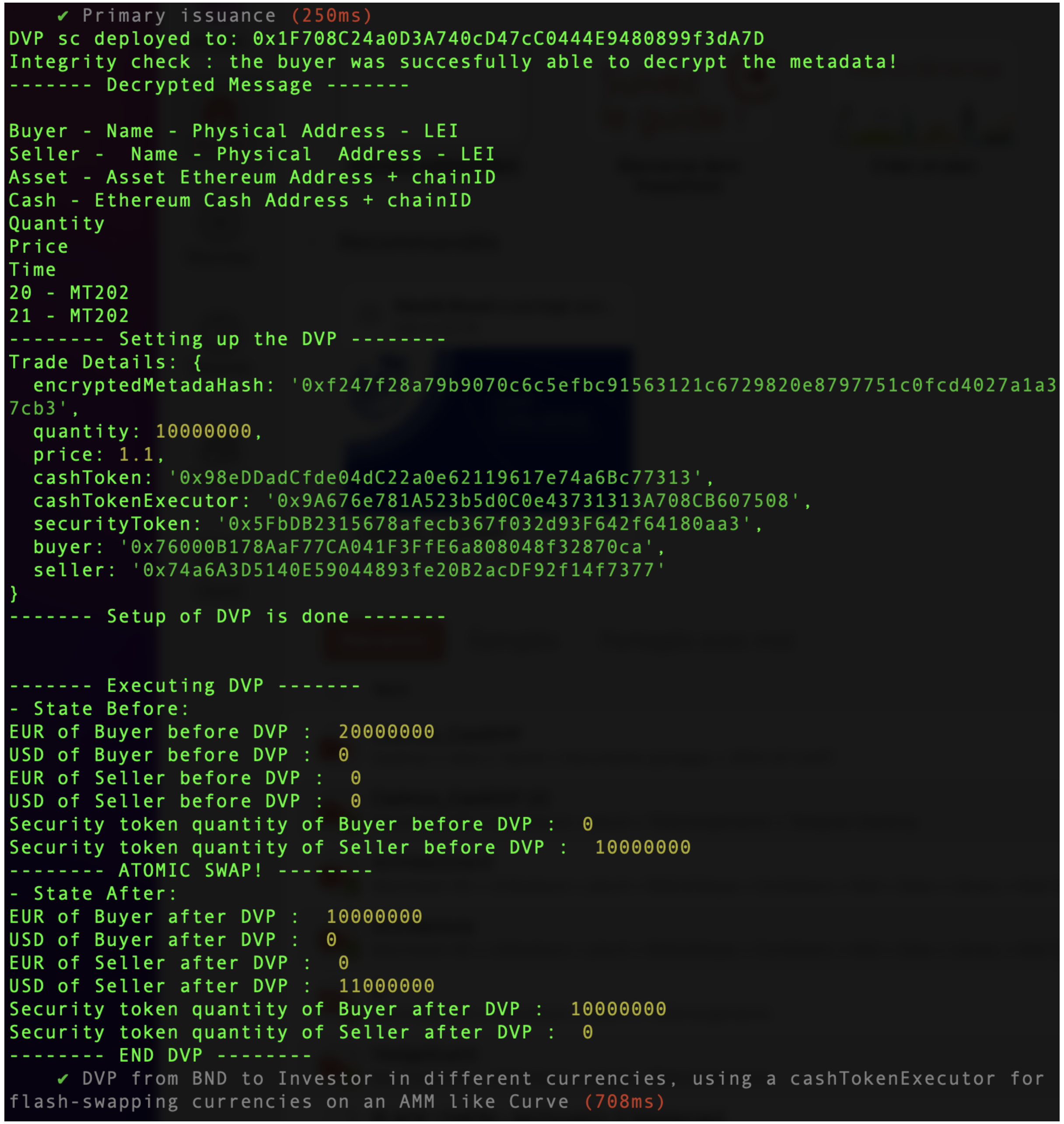


```
    ✔ Primary issuance (253ms)
DVP sc deployed to: 0x1F708C24a0D3A740cD47cC0444E9480899f3dA7D
Integrity check : the buyer was succesfully able to decrypt the metadata!
------- Decrypted Message -------

    Buyer - Name - Physical Address - LEI
    Seller -  Name - Physical  Address - LEI
    Asset - Asset Ethereum Address + chainID
    Cash - Ethereum Cash Address + chainID
    Quantity
    Price
    Time
    20 - MT202
    21 - MT202
-------- Setting up the DVP --------
Trade Details: {
  encryptedMetadaHash: '0x30871750b2039cd3c861c170c1880b803f456ef075286f02032d7226eab8
4dbb',
  quantity: 1000,
  price: 1,
  cashToken: '0x95bD8D42f30351685e96C62EDdc0d0613bf9a87A',
  cashTokenExecutor: '0x0000000000000000000000000000000000000000',
  securityToken: '0x5FbDB2315678afecb367f032d93F642f64180aa3',
  buyer: '0x76000B178AaF77CA041F3FfE6a808048f32870ca',
  seller: '0x74a6A3D5140E59044893fe20B2acDF92f14f7377'
}
------- Setup of DVP is done -------


------- Executing DVP -------
- State Before:
Cash of Buyer before DVP :  1000$
Cash of Seller before DVP :  0$
Security token quantity of investorA before DVP :  0
Security token quantity of Bnd before DVP :  1000
-------- ATOMIC SWAP! --------
- State After:
The settlement operator - who is also the seller - executes the DVP, after approval of
 the buyer
Cash of Buyer after DVP :   0$
Cash of Seller after DVP :   1000$
Security token quantity of Buyer after DVP :   1000
Security token quantity of Seller after DVP :  0
    ✔ DVP from BND to Investor in same currency unit (no cashTokenExecutor) (561ms)
```

Atomic DVP + FX



```
    ✔ Primary issuance (250ms)
DVP sc deployed to: 0x1F708C24a0D3A740cD47cC0444E9480899f3dA7D
Integrity check : the buyer was succesfully able to decrypt the metadata!
------- Decrypted Message -------

Buyer - Name - Physical Address - LEI
Seller -  Name - Physical  Address - LEI
Asset - Asset Ethereum Address + chainID
Cash - Ethereum Cash Address + chainID
Quantity
Price
Time
20 - MT202
21 - MT202
-------- Setting up the DVP --------
Trade Details: {
  encryptedMetadaHash: '0xf247f28a79b9070c6c5efbc91563121c6729820e8797751c0fcd4027a1a3
7cb3',
  quantity: 10000000,
  price: 1.1,
  cashToken: '0x98eDDadCfde04dC22a0e62119617e74a6Bc77313',
  cashTokenExecutor: '0x9A676e781A523b5d0C0e43731313A708CB607508',
  securityToken: '0x5FbDB2315678afecb367f032d93F642f64180aa3',
  buyer: '0x76000B178AaF77CA041F3FfE6a808048f32870ca',
  seller: '0x74a6A3D5140E59044893fe20B2acDF92f14f7377'
}
------- Setup of DVP is done -------


------- Executing DVP -------
- State Before:
EUR of Buyer before DVP :  20000000
USD of Buyer before DVP :  0
EUR of Seller before DVP :  0
USD of Seller before DVP :  0
Security token quantity of Buyer before DVP :  0
Security token quantity of Seller before DVP :  10000000
-------- ATOMIC SWAP! --------
- State After:
EUR of Buyer after DVP :  10000000
USD of Buyer after DVP :  0
EUR of Seller after DVP :  0
USD of Seller after DVP :  11000000
Security token quantity of Buyer after DVP :  10000000
Security token quantity of Seller after DVP :  0
-------- END DVP --------
    ✔ DVP from BND to Investor in different currencies, using a cashTokenExecutor for
flash-swapping currencies on an AMM like Curve (708ms)
```