TradeTrust Tech Webinar Series

Webinar #4

5th Aug 2020

Infocomm Media Development Authority, Singapore

GOVTECH SINGAPORE

SG:D EMPOWERING POSSIBILITIES

IMDA INFOCOMM MEDIA DEVELOPMENT AUTHORITY

AGENDA

- OpenAttestation Document Format
  - Document Integrity
  - Issuer Identity
  - Document Status
- SDK Overview, Code Walkthroughs
  - oa-verify
  - document-store
  - token-registry
  - decentralised-renderer-react-components
- Q & A

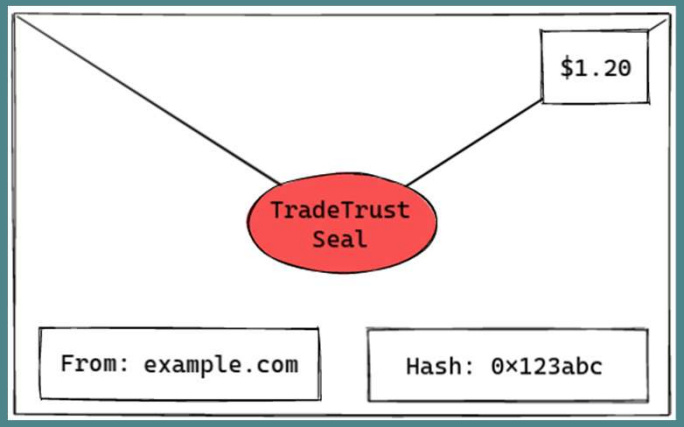
These Slides

GOVTECH SINGAPORE

SG:D EMPOWERING POSSIBILITIES

INFOCOMM MEDIA DEVELOPMENT AUTHORITY

OpenAttestation Document Format



1. Assurance that content is intact
   [*DOCUMENT_INTEGRITY*]

1. Assurance of issuer identity
   [*ISSUER_IDENTITY*]

1. Assurance of Document Status
   [*DOCUMENT_STATUS*]

GOVTECH SINGAPORE

SG:D EMPOWERING POSSIBILITIES

INFOCOMM MEDIA DEVELOPMENT AUTHORITY

Document Verification Demo

- Valid Document
- Document with unauthorised modifications
- Document with false issuer identity
- Document that was not issued

GOVTECH SINGAPORE

SG:D EMPOWERING POSSIBILITIES

INFOCOMM MEDIA DEVELOPMENT AUTHORITY

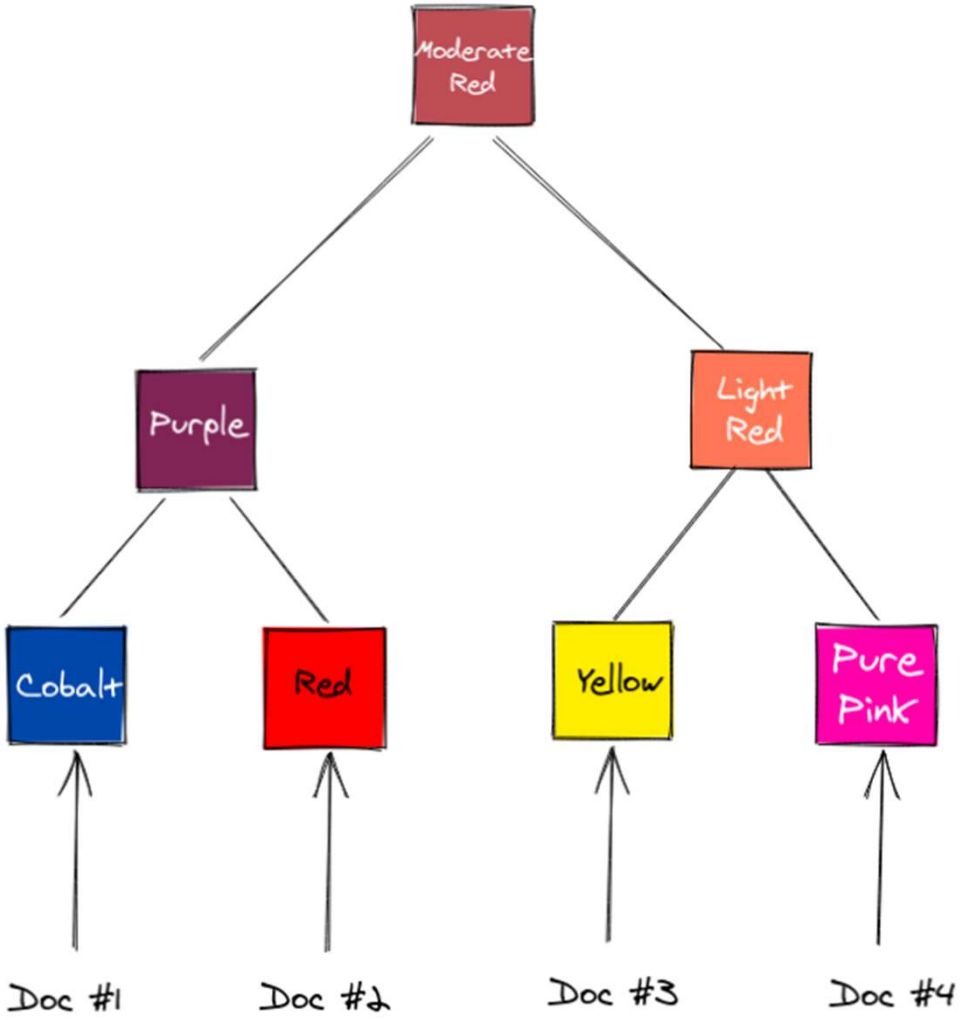
[*DOCUMENT_INTEGRITY*]: Concept #1 - Hashes

- Represent a document by a unique colour
- Combining two colours results in a third colour
- Cannot unmix third colour to determine which two colours were mixed to create it

GOVTECH SINGAPORE

INFOCOMM MEDIA DEVELOPMENT AUTHORITY

[*DOCUMENT_INTEGRITY*]: Concept #2 - Merkle Trees, Creation

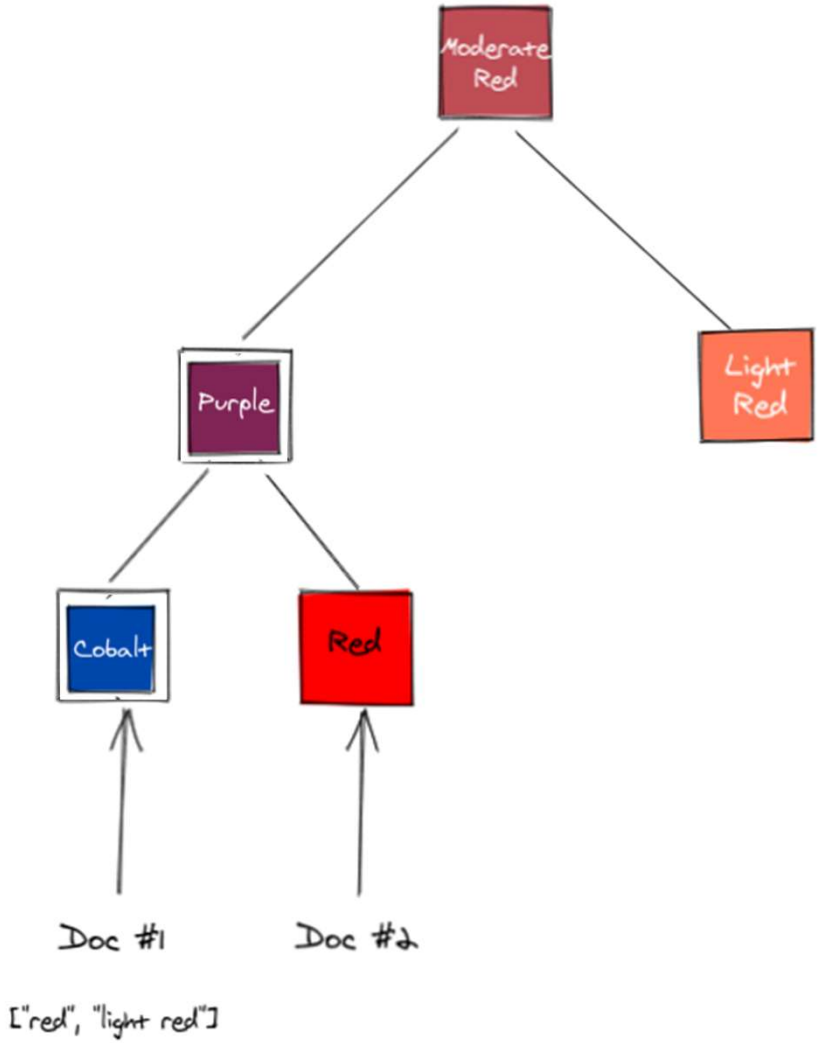


- Example: 4 Documents in batch
- Record “Moderate Red” on the blockchain
- No other colour recorded on the blockchain
- Each document records the minimum set of intermediate colours to arrive at “Moderate Red”
- E.g: Doc #1 will have [“red”, “light red”]
- E.g: Doc #3 will have [“pure pink”, “purple”]

GOVTECH SINGAPORE

INFOCOMM MEDIA DEVELOPMENT AUTHORITY

[*DOCUMENT_INTEGRITY*]: Concept #2 - Merkle Trees, Verification

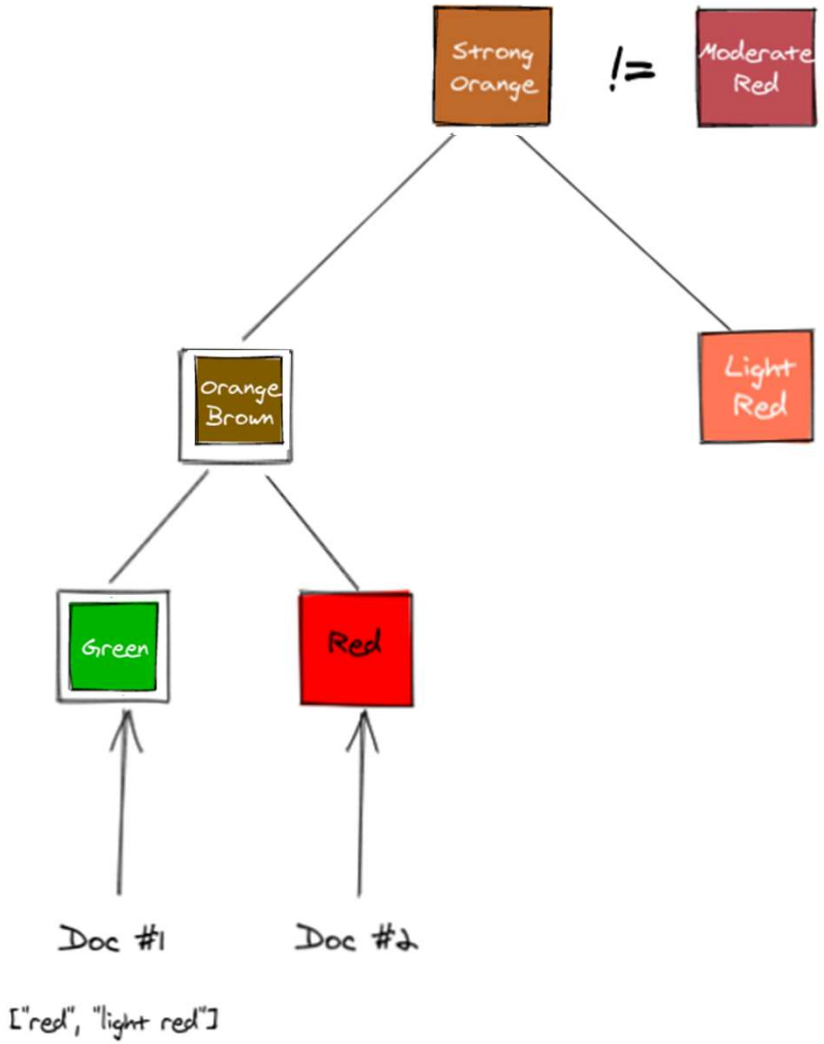


- Doc #1 records that his sibling is “red”
- His “uncle” is “light red”
- Verifier applies algorithm to given document contents and arrives at “cobalt”
- “cobalt” + “red” = “purple”
- “purple” + “light red” = “moderate red”
- Verification Success!

GOVTECH SINGAPORE

INFOCOMM MEDIA DEVELOPMENT AUTHORITY

[*DOCUMENT_INTEGRITY*]: Concept #2 - Merkle Trees, Falsification



- Doc #1 records that his sibling is “red”
- His “uncle” is “light red”
- Doc #1 modified and integrity affected
- Verifier applies algorithm to given document contents and arrives at “green”
- “green” + “red” = “orange-brown”
- “orange-brown” + “light red” = “strong orange”
- Verification Failed!

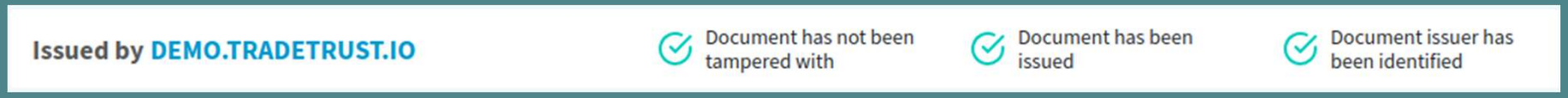
[*DOCUMENT_INTEGRITY*]: Document Format


```
"signature": {
  "type": "SHA3MerkleProof",
  "targetHash": "6f8b1ffc45aa9b5b92b105dfa172c163d29de3e680750eae5e1de7b307483b79",
  "proof": [
    "649879f5c8d26e455e30aa8064b01bef142c29cc8c387b409e9467bf31dc00c4",
    "51a4fa112637b683dcbc898bad86867d26363a76bd5b0504ae95704ba1f8fddb",
    "9d75bcb736099c86ac69312dc2a09ed57baaceaad6a7ffb5df12d8da06b71683",
    "066e724d827caca297413e14ea955fd51ccaf37da5719d3dec787cbb6afed00f",
    "d281a2430c78da6a48cd3291e10080704af5cc59d635b44613fc396c1a244bfe"
  ],
  "merkleRoot": "14d30e6edff6ab6fc0ff1fa848024343739bd9652229cca041b309e360778567"
}
```


GOVTECH SINGAPORE

INFOCOMM MEDIA DEVELOPMENT AUTHORITY

Issuer Identity Assurance

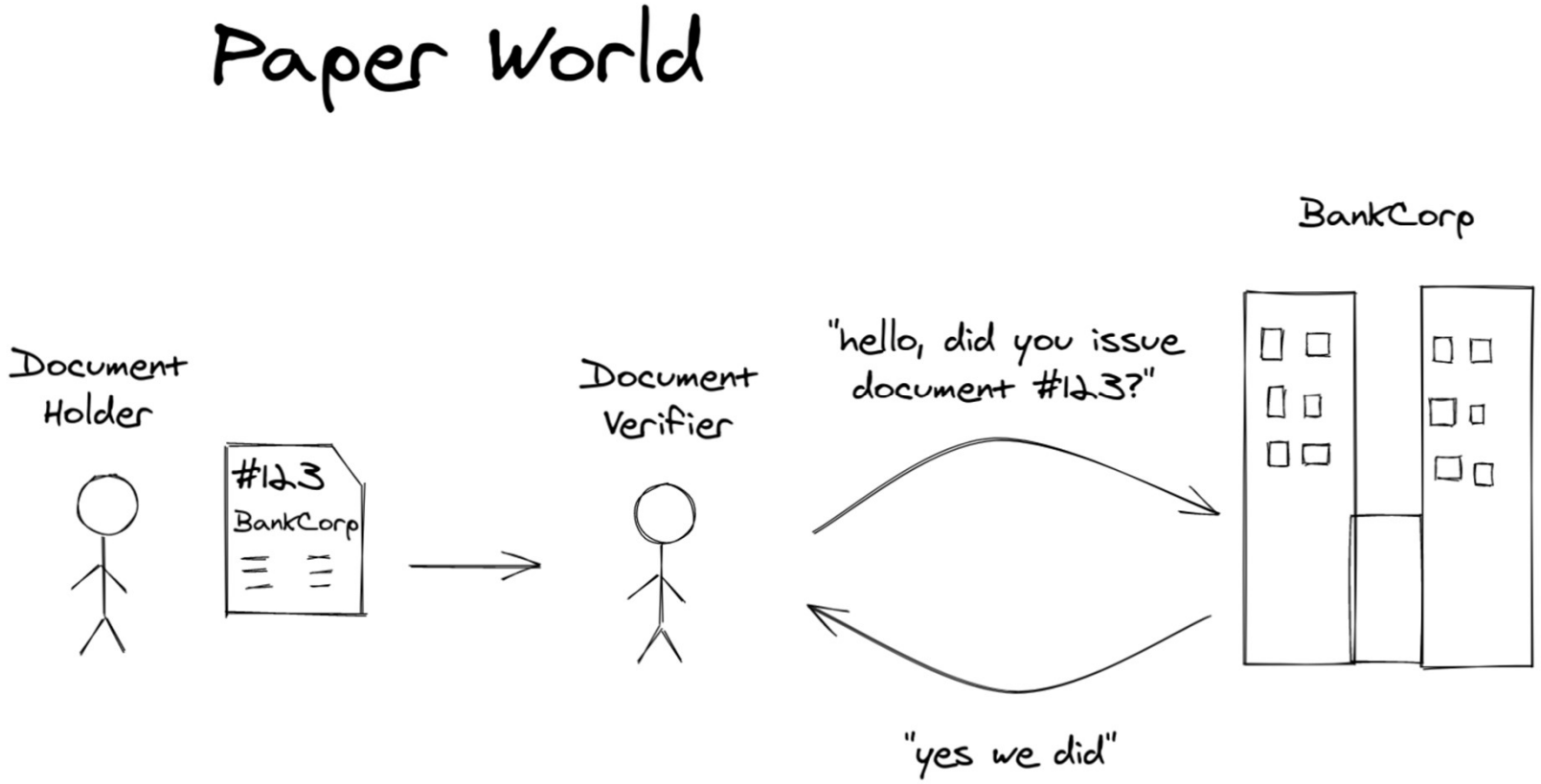


GOVTECH SINGAPORE

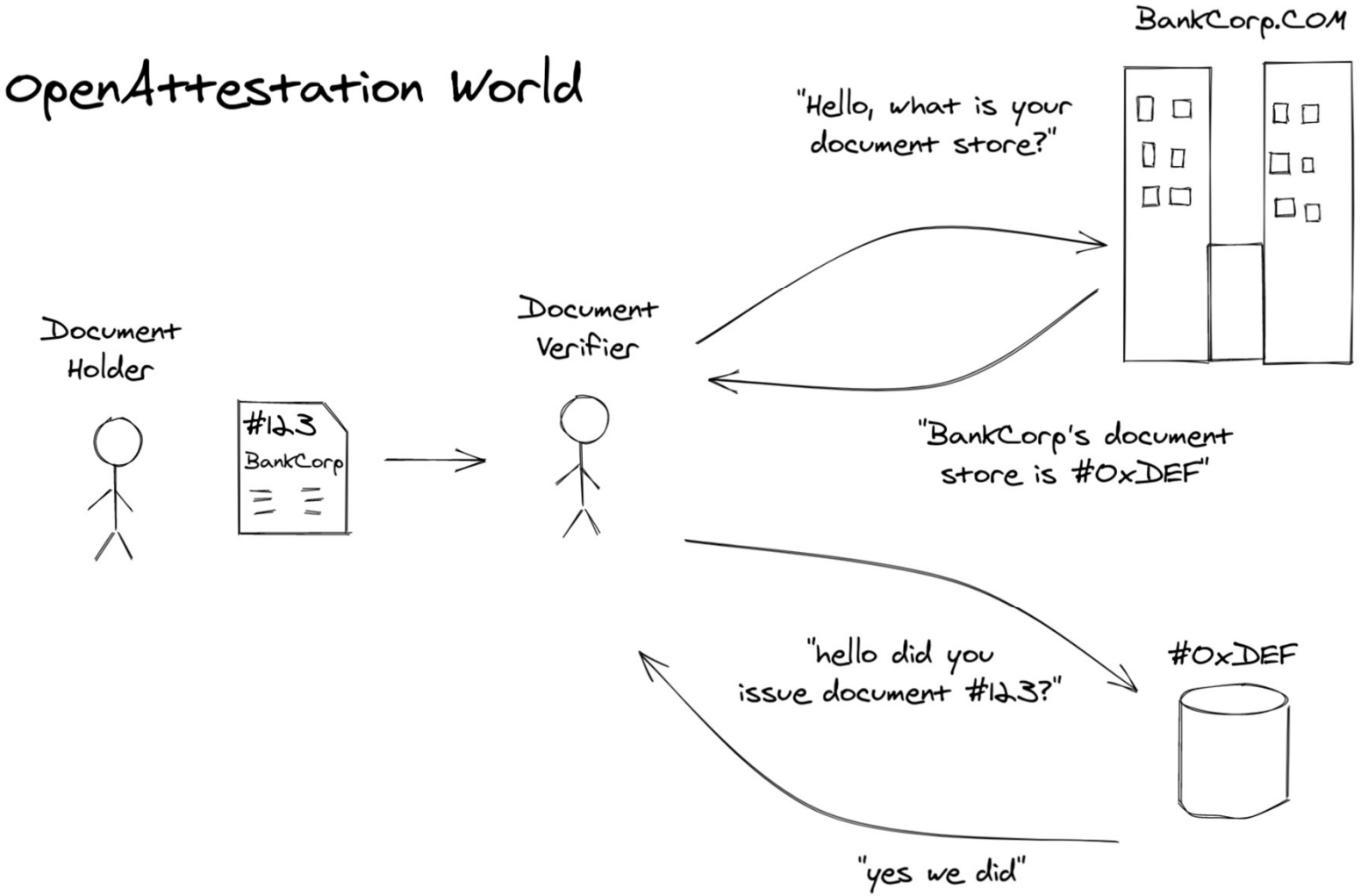
SG:D EMPOWERING POSSIBILITIES

INFOCOMM MEDIA DEVELOPMENT AUTHORITY

[*ISSUER_IDENTITY*]: Concept - Dialing back to confirm identity



[*ISSUER_IDENTITY*]: Concept - Dialing back to confirm identity



[*ISSUER_IDENTITY*]: Document Format

```
"issuers": [
  {
    "name": "434808bb-73d9-...:string:Demo Issuer",
    "documentStore": "630a5665-...:string:0x8bA63EAB43342AAc3AdBB4B827b68Cf4aAE5Caca",
    "identityProof": {
      "type": "462fcc43-...:string:DNS-TXT",
      "location": "8b96e8b5-13f4-...:string:demo.tradetrust.io"
    }
  }
]
```

Pop Quiz!

http://etc.ch/B6rD/

GOVTECH SINGAPORE

SG:D EMPOWERING POSSIBILITIES

INFOCOMM MEDIA DEVELOPMENT AUTHORITY

Part II - SDK Overview

GOVTECH SINGAPORE

SG:D EMPOWERING POSSIBILITIES

INFOCOMM MEDIA DEVELOPMENT AUTHORITY

SDK Overview

- oa-verify - Library that takes in document and returns verification results
- document-store - Repository with Document Store smart contract and helpers
- token-registry - Repository with Token Registry smart contract and helpers
- ethers-contracts-hooks - Utility library for using smart contracts in React
- decentralized-renderer-react-components - Library for custom renderers
- dnsprove - Library that returns OA DNS-TXT results for a given domain
- oa-encryption - Library that handles OA Common Encryption
- oa-functions - Infrastructure templates for OA functions such as verify

Open-Attestation/oa-verify Library

- A library for verifying Open-Attestation documents (TradeTrust is a subset)
- Has default verification settings, but can also override the defaults to create your own verifier
- 3 categories of verification (DOCUMENT_STATUS, DOCUMENT_INTEGRITY, ISSUER_IDENTITY)
- Returns a detailed array of what checks were done and the results
- isValid can be used to summarise the results into whether it passed or failed
- Usable in both Node.js and Browser

GOVTECH SINGAPORE

INFOCOMM MEDIA DEVELOPMENT AUTHORITY

Code Walkthrough for oa-verify

- Step 0: Barebones page with a text field for you to enter document data
- Step 1: Set up oa-verify
- Step 2: Call on oa-verify when document data is provided
- Step 3: Parsing verification results

Open-Attestation/Document-Store Library

- Repository contains the smart contract source code
- Also has helper functions for deploying and interacting with it
- Can also connect to existing instance and execute methods on it
- Requires you to have Ethers.js provider (and signer if writing)
- Can be combined with Open-Attestation/ethers-contract-hooks
- Usable in both NodeJS and Browser

GOVTECH SINGAPORE

INFOCOMM MEDIA DEVELOPMENT AUTHORITY

Open-Attestation/Token-Registry Library

- Repository contains the smart contract source code for token-registry and title-escrow
- Also has helper functions for deploying and interacting with it
- Can connect to existing instance and execute methods on it
- Requires you to have Ethers.js provider (and signer if writing)
- can be combined with Open-Attestation/ethers-contract-hooks
- Title Escrow smart contract can be deployed using TitleEscrowFactory
- token-registry is ERC721 compatible, for TradeTrust the owner points to a title-escrow
- Usable in both NodeJS and Browser

Code Walkthrough for Token-Registry

- Step 0: Barebones with MetaMask provider
- Step 1: Connecting to Token Registry
- Step 2: Deploying Title Escrow
- Step 3: Minting Token to Title Escrow
- Step 4: Transferring Token to Target

GOVTECH SINGAPORE

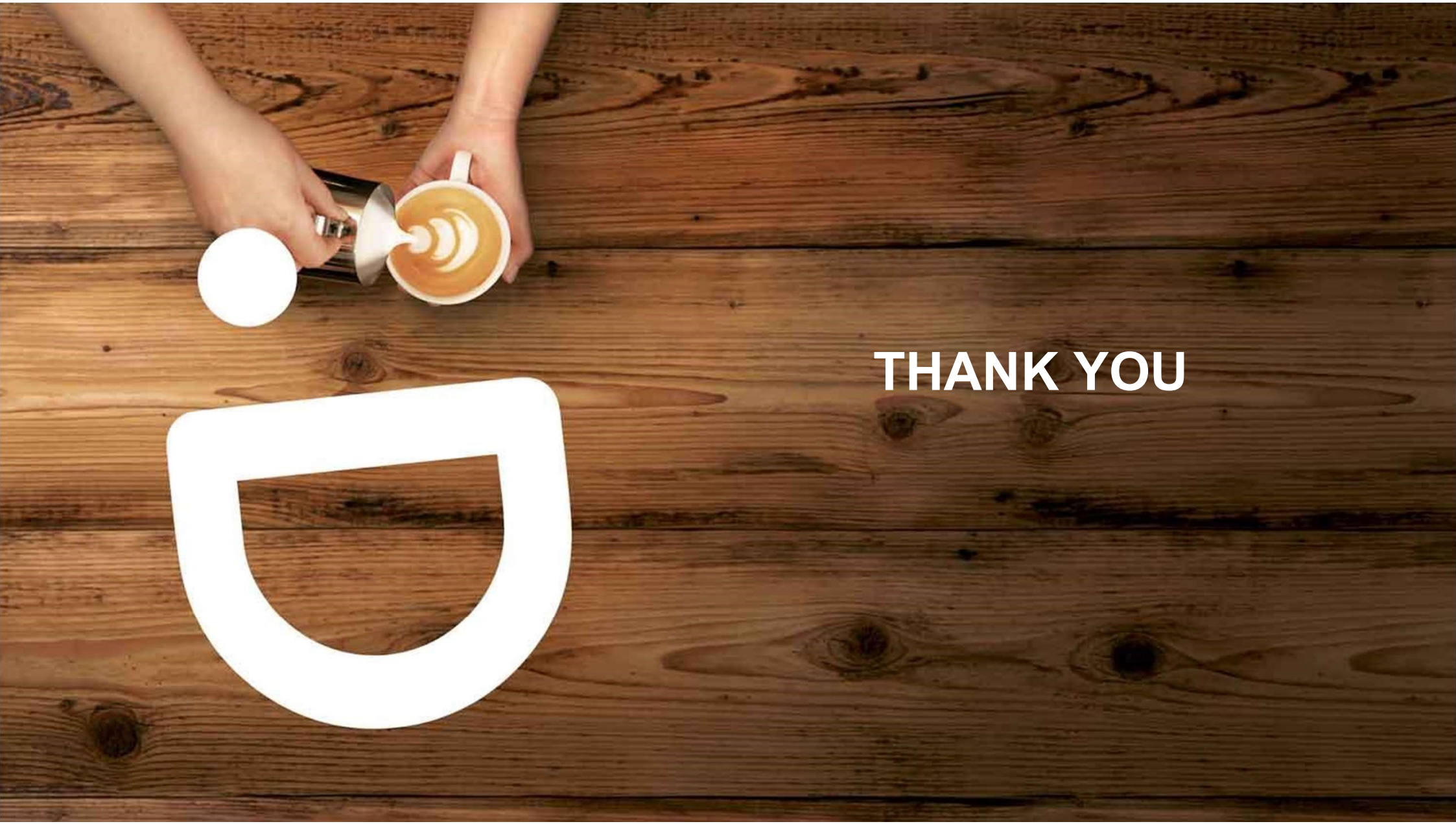
INFOCOMM MEDIA DEVELOPMENT AUTHORITY

Q&A?

GOVTECH SINGAPORE

SG:D EMPOWERING POSSIBILITIES

INFOCOMM MEDIA DEVELOPMENT AUTHORITY


THANK YOU