



# TRADETRUST

## Tech Seminar

### Jul 2020

Infocomm Media Development Authority, Singapore





# Verifiable documents

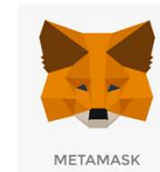
- Slides
  - <https://bit.ly/tt-webinar-2-slides>
- Workshop materials
  - <https://bit.ly/tt-webinar-2-workshop>

# ▼ Prerequisites

- Nodejs
- GitHub account
- Basic cli usage knowledge
- Optional
  - Metamask account (wallet)
  - Domain name (issuer identity)
  - Netlify account (custom document renderer)



*Ether (gas) = cryptocurrency = digital currency.*



*Some of the platforms referenced in this workshop.*

# Verifiable document + Open Attestation CLI (oa-cli)

- Do read these:
  - <https://openattestation.com/docs/verifiable-document/overview>
  - <https://github.com/Open-Attestation/open-attestation-cli>



# What you will learn

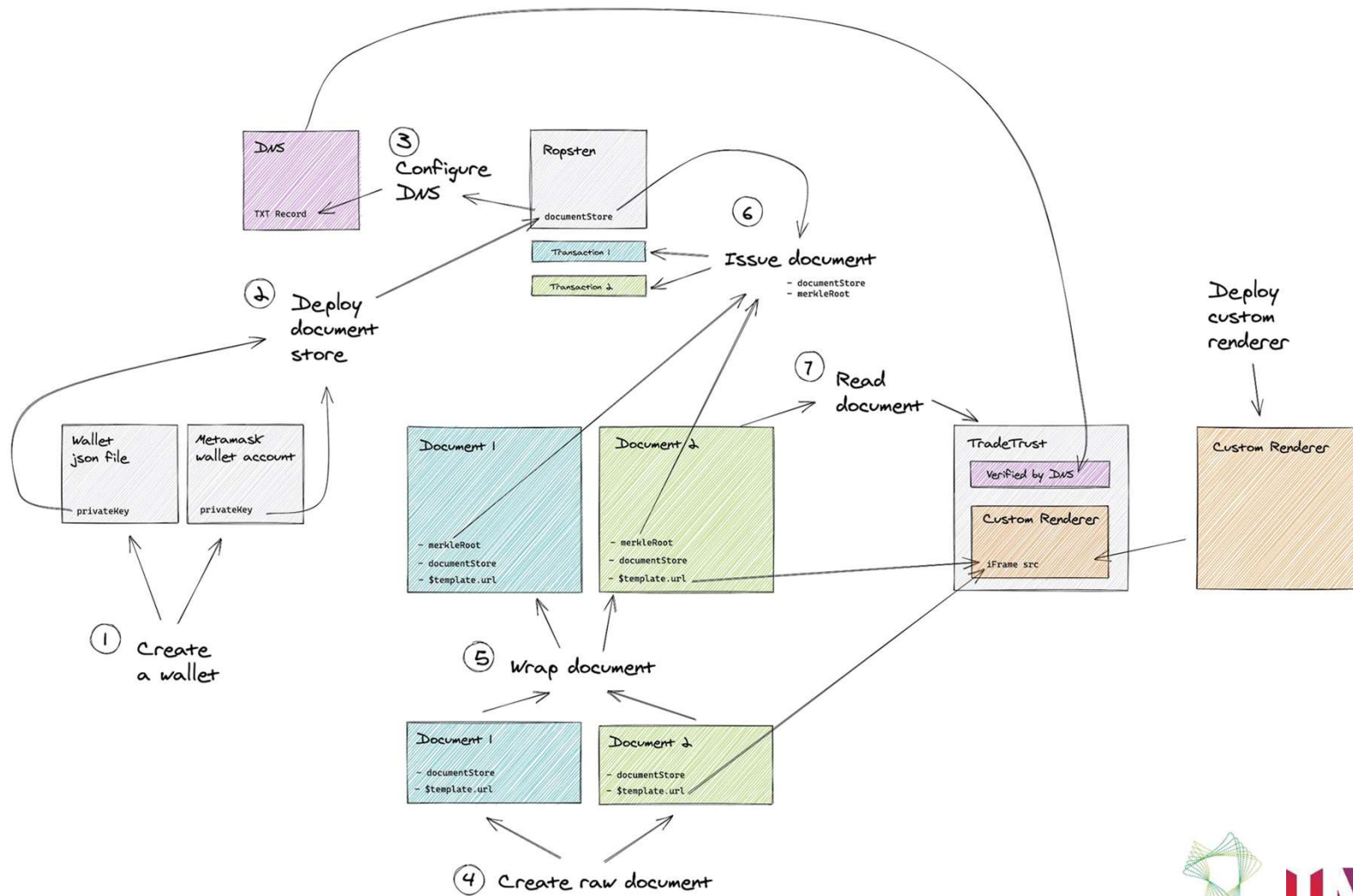
1. Create a wallet
2. Deploy document store
3. Configure DNS
4. Create raw document
5. Wrap document
6. Issue document
7. Read document



## Additional topic

1. Deploy custom renderer
  - Deploy a custom document renderer on Github, with Netlify
  - Make edits to custom document renderer

# Overview



# Introduction



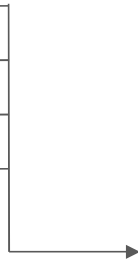




# What can be considered a verifiable document?

- Examples can be:

- E-invoice
- Certificates
- Packing List
- Purchase Order



```
1  {
2  >  "issuers": [ ...
11  ],
12  >  "$template": { ...
16  },
17  "name": "John Doe",
18  "institute": "Institute of John Doe",
19  >  "foo": { ...
21  }
22 }
```

*A basic json file example with minimal key value pairs.*



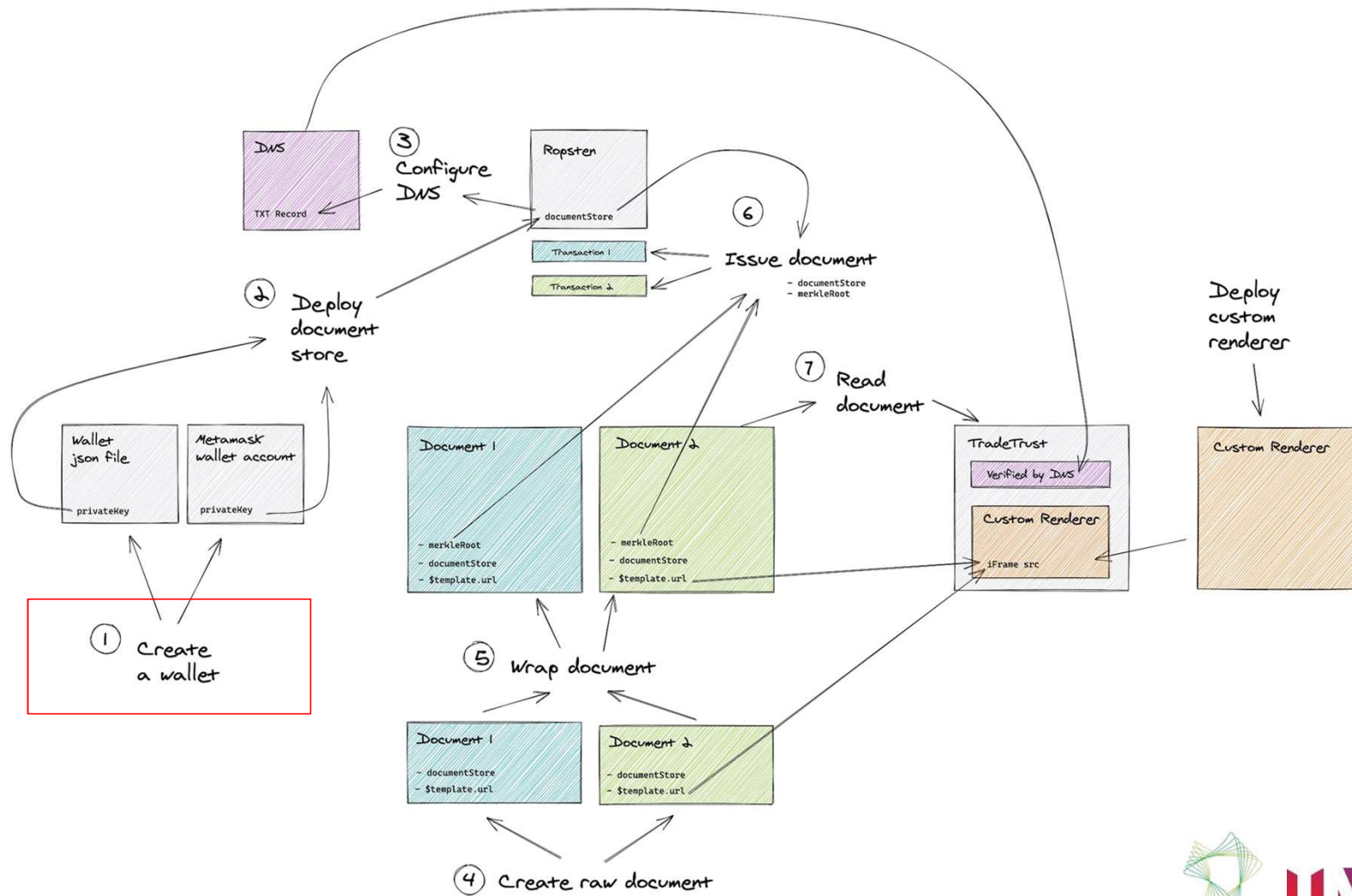
# Install open-attestation (oa-cli)

1. Make sure you have nodejs installed
2. Install open-attestation-cli globally
  - `npm install -g @govtechsg/open-attestation-cli`
    - <https://openattestation.com/docs/component/open-attestation-cli>
    - <https://github.com/Open-Attestation/open-attestation-cli#setup>
  - `npx -p @govtechsg/open-attestation-cli open-attestation <arguments>`
3. Check if successfully installed
  - `open-attestation --version`
  - `open-attestation --help`

# Create wallet



# Create wallet





# Create wallet (oa-cli)

1. Generate a wallet json file
  - open-attestation wallet create --output-file wallet.json --fund ropsten
2. Create a password for your wallet
3. Save your wallet public address somewhere in case you forget : )
4. **Never lose your wallet file, keep it safe**

```
{ } wallet.json X
{ } wallet.json > ...
1 [{"address": "b4ab7b9446754894cc93ff925eb0fb08f8e1d029",
  "id": "34eb9a2-5120-4020-9130-aaa40e14ezec", "version": 3,
  "Crypto": {"cipher": "aes-128-ctr", "cipherparams":
    {"iv": "415a76189dff3165f04d8060062bf2cf"},
    "ciphertext": "a87d7bbff5a3767b144d8e58239e64ddaa90558f125a
    7e98cbc726ac896d2891", "kdf": "scrypt", "kdfparams":
    {"salt": "58e104a0327801de0b466bb26825d8162067043957f576878
    41a8a1e104f035c", "n": 131072, "dklen": 32, "p": 1, "r": 8},
    "mac": "d26b12d3bb367f7a821251de89a74242b5e7997ec01691468cf
    13e353d8bfb92"}, "x-ethers": {"client": "ethers.js",
    "gethFilename": "UTC--2020-07-15T08-01-05.
    0Z--b4ab7b9446754894cc93ff925eb0fb08f8e1d029",
    "mnemonicCounter": "0e401fe810822a5456296b96cc5b15ce",
    "mnemonicCiphertext": "203147dba4b2ac02de6b48a069e05bbd",
    "path": "m/44'/60'/0'/0/0", "version": "0.1"}}]
```

<https://ropsten.etherscan.io/address/0xb4ab7b9446754894cc93ff925eb0fb08f8e1d029>



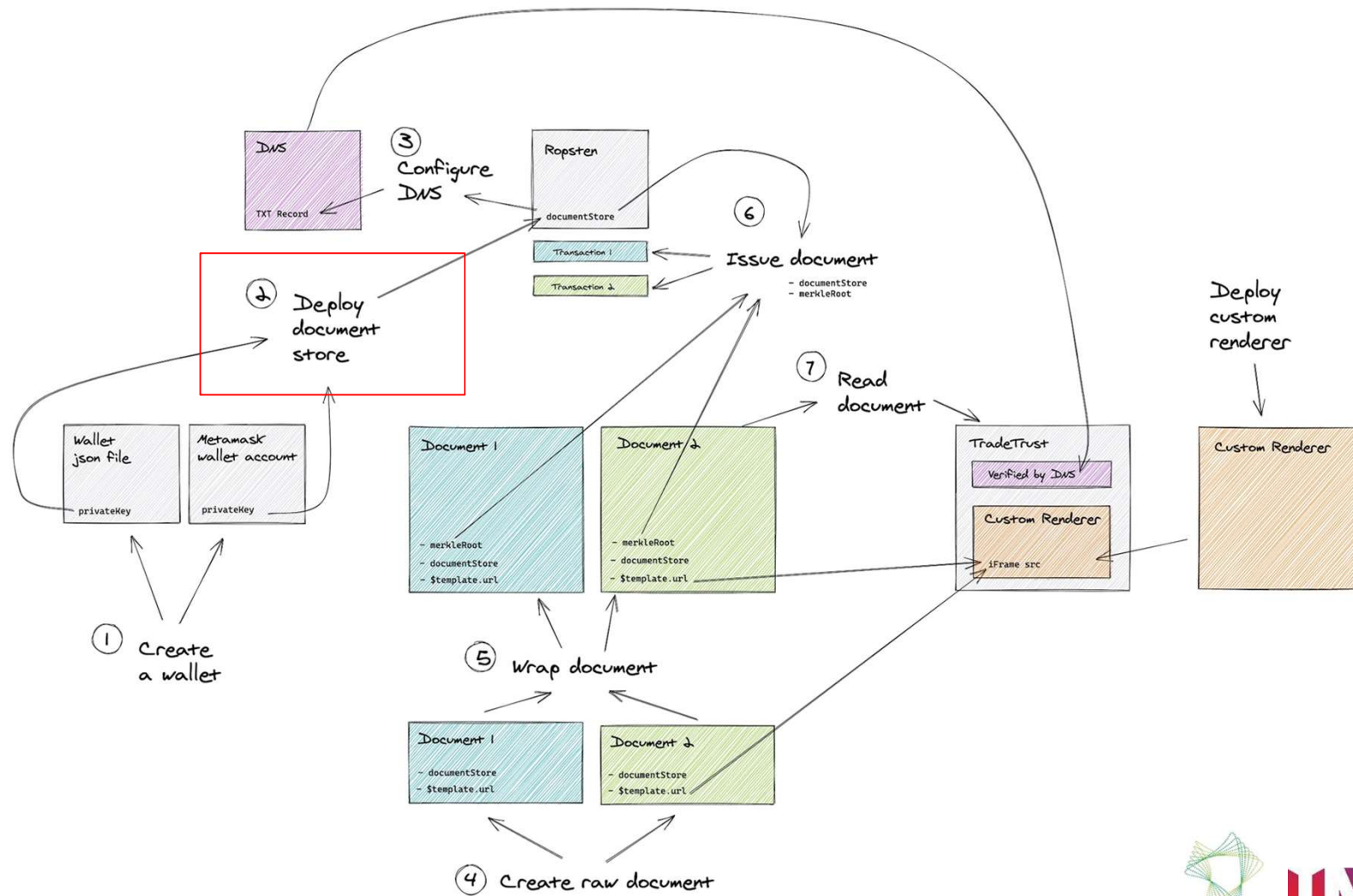
# Create wallet (metamask)

1. Install Metamask extension
  - <https://metamask.io/>
1. Create some wallet accounts
2. Request for some ethers
  - <https://faucet.ropsten.be/>
  - <https://faucet.metamask.io/>

# Deploy document store



# Deploy document store







# Deploy document store (oa-cli)

## 1. Deploy document store with wallet.json

- Command
  - `open-attestation deploy document-store "<storeName>" --network <mainnet || ropsten> --encrypted-wallet-path <pathToWalletJson>`
- Example
  - `open-attestation deploy document-store "My first document store" --network ropsten --encrypted-wallet-path wallet.json`

## 1. Save your document store address somewhere in case you forget : )



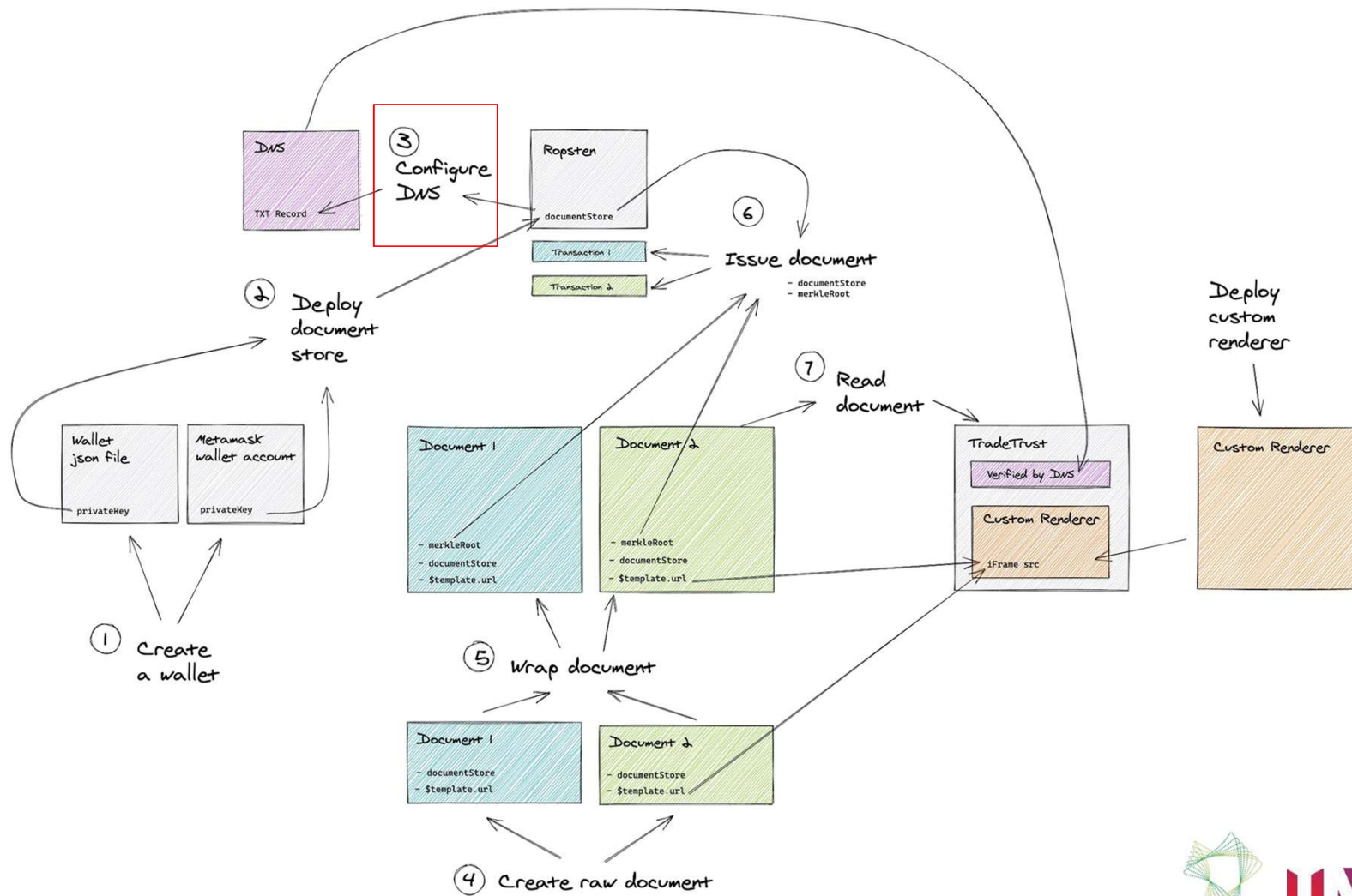
# Deploy document store (metamask)

1. Get private key from metamask wallet account
2. Deploy document store with privateKey
  - Command
    - `export OA_PRIVATE_KEY=<privateKey>`
    - `open-attestation deploy document-store "<storeName>" --network <mainnet || ropsten>`
  - Example
    - `export OA_PRIVATE_KEY=2F12345678`
    - `open-attestation deploy document-store "My first document store" --network ropsten`
3. Save your document store address somewhere in case you forget : )

# Configure DNS



# Configure DNS





# Configure DNS (oa-cli)

## 1. Create temporary DNS record

- Command
  - `open-attestation dns txt-record create --address <documentStore> --network-id 3`
- Example
  - `open-attestation dns txt-record create --address 0x4B563cAE8F6D4E1a31B1e9217C9DD7DC371c0C1D --network-id 3`

## 2. Verify TXT record

- Command
  - `open-attestation dns txt-record get --location <domainName> --networkId 3`
- Example
  - `open-attestation dns txt-record get --location impressive-salmon-egret.sandbox.openattestation.com --networkId 3`

```
SimsMBPgovtech:workshop simboonlong$ open-attestation dns txt-record get --location brew.tk --networkId 3
```

(index)	type	net	netId	addr	dnssec
0	'openatts'	'ethereum'	'3'	'0xF78a7713591517288A950874658728910b1c98dA'	false
1	'openatts'	'ethereum'	'3'	'0x6D31C978c08e929e458AE9F276C875c9919214C9'	false

CLI: Pinging TXT record value to see if added successfully.



# Configure DNS (domain registrar)

## 1. Add a TXT record

- Value
  - `openatts net=ethereum netId=<networkNumber> addr=<documentStoreAddress>`
- Example
  - `openatts net=ethereum netId=3 addr=0x4B563cAE8F6D4E1a31B1e9217C9DD7DC371c0C1D`

## 2. Wait awhile for DNS to propagate

- Example used in this workshop is from <https://www.freenom.com/>

## 3. Verify if TXT record successfully added

- <https://dns.google.com/>

Add Records

A

1200

openatts net=ethereum netId=3 addr=0xF78a7713591517288A950874658728910b1c98dA

[+ More Records](#) [Save Changes](#)

*Adding TXT record, the example used here is freenom.*

```
{
  "name": "brew.tk.",
  "type": 16,
  "TTL": 1199,
  "data": "\"openatts net=ethereum netId=3 addr=0x6D31C978c08e929e458AE9F276C875c9919214C9\""
},
{
  "name": "brew.tk.",
```

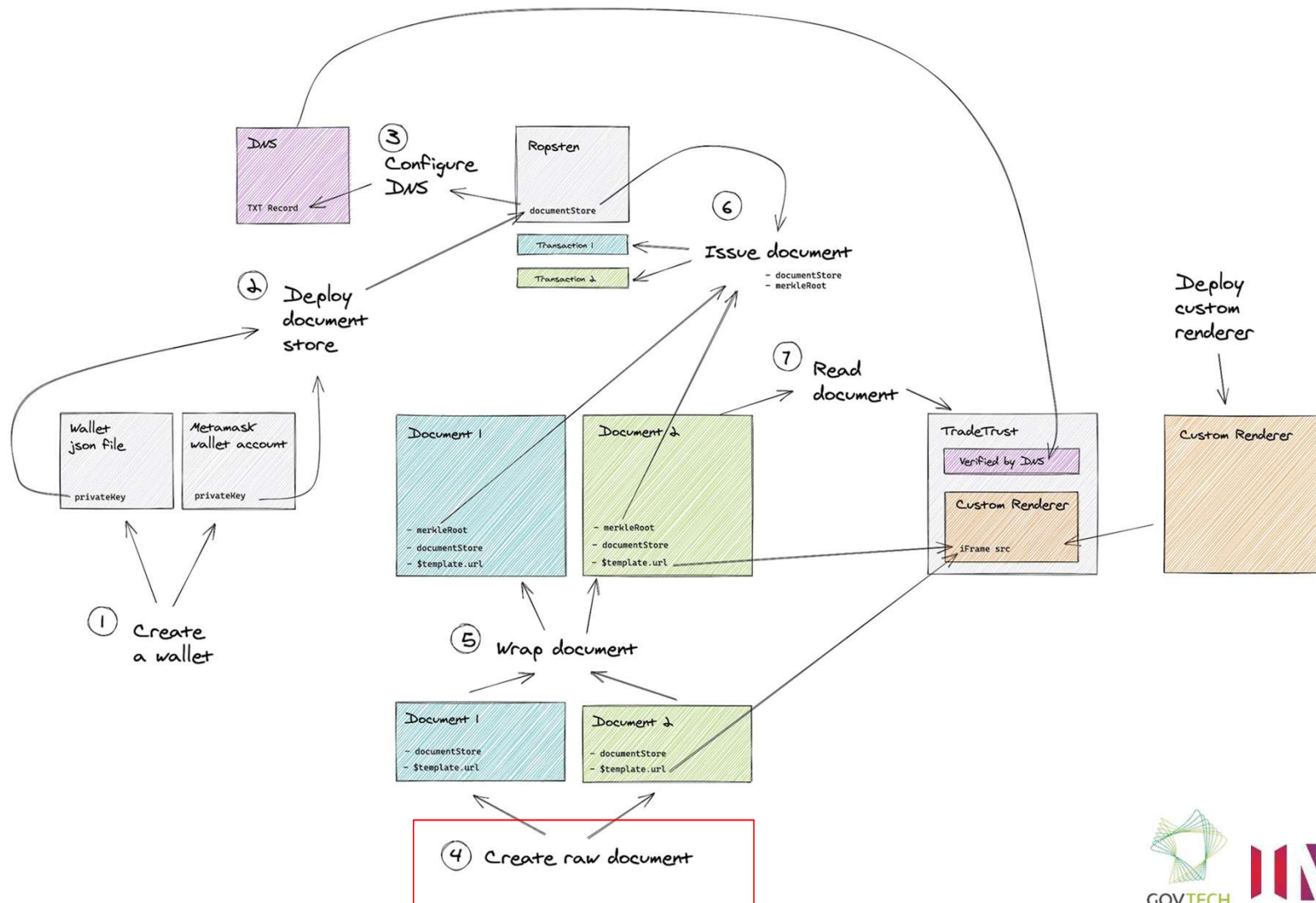
*Verified TXT record at dns.google.*

# Create raw document





# Create raw document







## Create raw document

- Open attestation document schema at:
  - <https://schema.openattestation.com/2.0/schema.json>
- Test your raw document against full schema at:
  - <https://www.jsonschemavalidator.net/>

```
1 {  
2   "version": "https://schema.openattestation.com/2.0/schema.json",  
3   "data": {  
4     "$template": {  
5       "name": "cc105e63-583e-4c6b-831d-53b9ab1c6859:string:main",  
6       "type":  
7         "10c1014d-507f-4122-b261-31eea388764e:string:EMBEDDED_RENDERER",  
8       "url": "0f71e8a8-5c8a-49af-bc44-abb67035ddc4:string:https://  
9         tutorial-renderer.openattestation.com"  
10    },  
11    "recipient": {  
12      "name": "86dc70f4-002b-447f-b548-89b91f667d97:string:John_Doe"
```

OA schema version defined in a wrapped document.



# Create raw document

- Issuers (Required)
  - Domain name (location)
  - Document store address (documentStore)
- \$template
  - Custom renderer location (url)

```
1  {
2    "issuers": [
3      {
4        "identityProof": {
5          "type": "DNS-TXT",
6          "location": "brew.tk"
7        },
8        "name": "Store name",
9        "documentStore": "0xF78a7713591517288A950874658728910b1c98dA"
10     },
11   ],
12 >  "$template": {--
16   },
17   "name": "John Doe",
18   "institute": "Institute of John Doe",
19 >  "foo": {--
21   }
22 }
```

*Issuers key is required.*

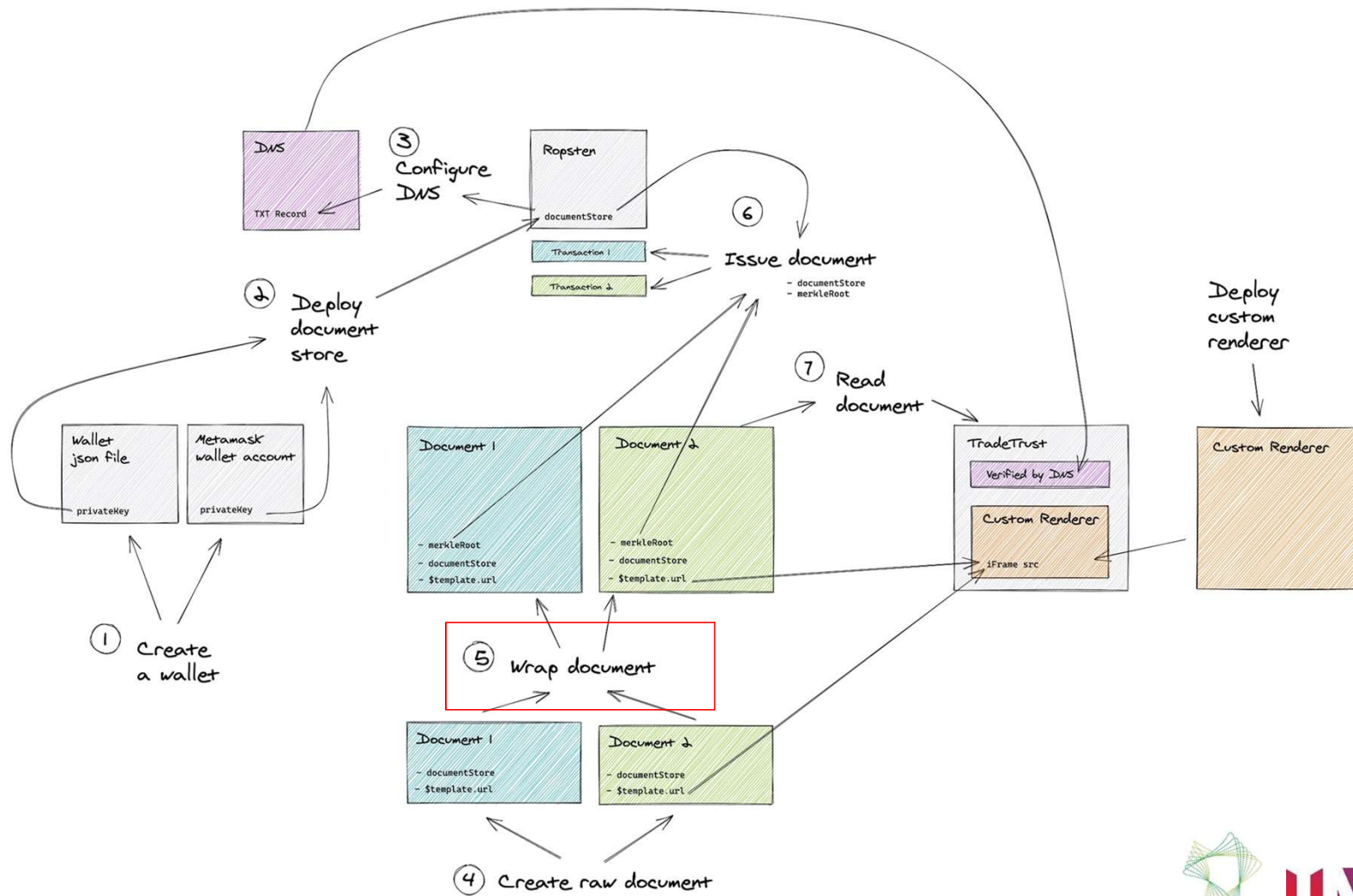
```
1  {
2 >  "issuers": [--
11 ],
12  "$template": {
13    "name": "custom",
14    "type": "EMBEDDED_RENDERER",
15    "url": "https://mystifying-swartz-b01fbb.netlify.app"
16  },
17  "name": "John Doe",
18  "institute": "Institute of John Doe",
19 >  "foo": {--
21  }
22 }
```

*\$template is needed for rendering document.*

# Wrap document



# Wrap document





## Wrap document

```
1 {
2 >   "issuers": [ ...
11 ],
12 >   "$template": { ...
16 },
17   "name": "John Doe",
18   "institute": "Institute of John Doe",
19 >   "foo": { ...
21 }
22 }
```

Before file is wrapped.



```
1 {
2   "version": "https://schema.openattestation.com/2.0/schema.json",
3   "data": {
4 >     "issuers": [ ...
13 ],
14 >     "$template": { ...
18 },
19     "name": "361e3229-bd0b-4135-8e21-9efbacc23804:string:John Doe",
20     "institute": "bd251286-5716-4c6d-8fdf-9dbe8cacc179:string:Institute of John Doe",
21 >     "foo": { ...
23 }
24 },
25   "signature": {
26     "type": "SHA3MerkleProof",
27     "targetHash": "c04ffd6fba0d3e0f916304e72646097384ec30aa316d27cff4fb616b23831094",
28     "proof": [],
29     "merkleRoot": "c04ffd6fba0d3e0f916304e72646097384ec30aa316d27cff4fb616b23831094"
30   }
31 }
```

After file is wrapped.



# Wrap document

## 1. Batch wrap documents

- Command
  - `open-attestation wrap <inputDocumentPath> --output-dir <outputDocumentPath>`
- Example
  - `open-attestation wrap ./raw-documents --output-dir ./wrapped-documents`

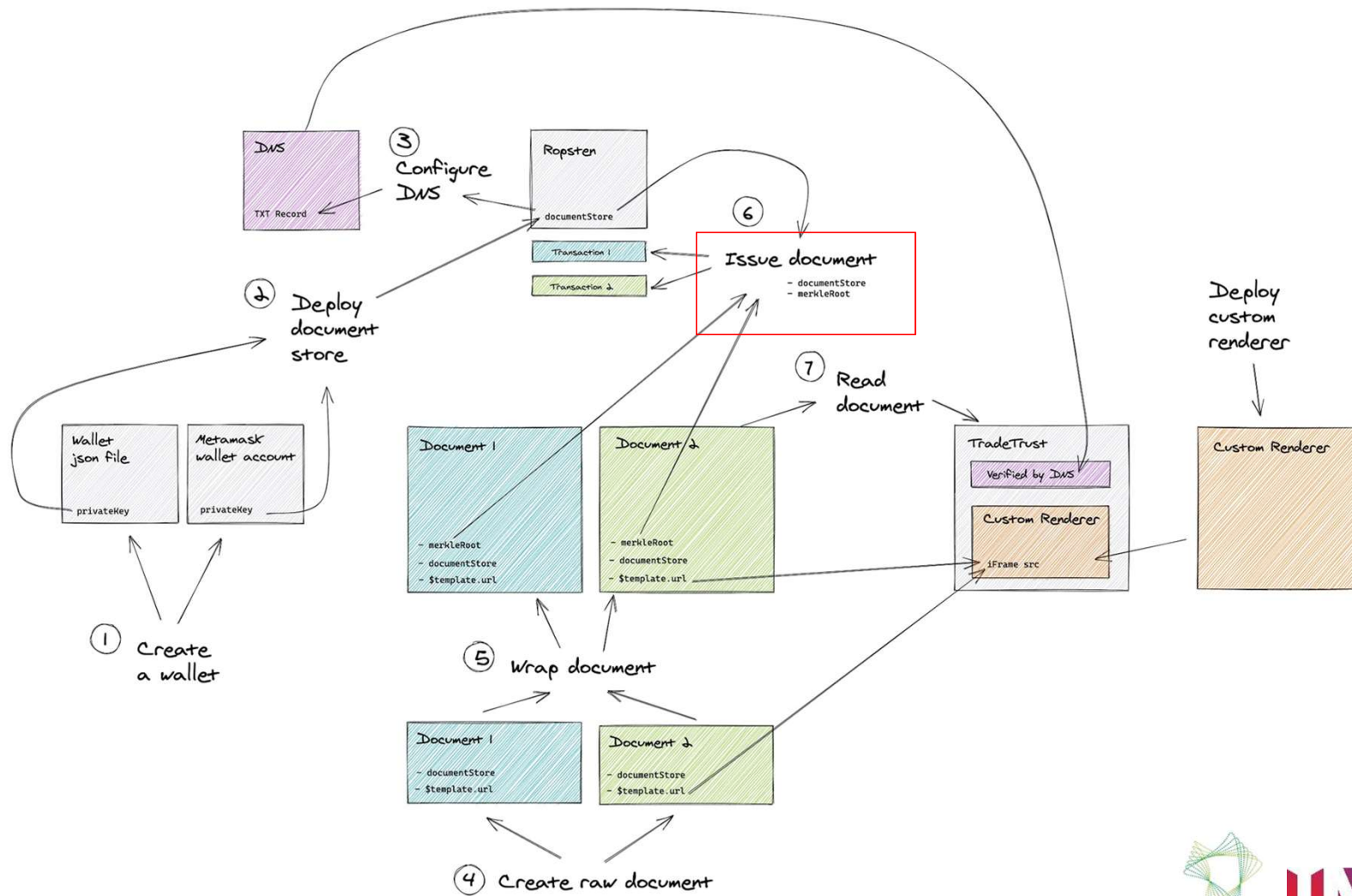
## 2. Single wrap document

- Command
  - `open-attestation wrap <inputDocumentPath> --output-file <outputDocumentPath>`
- Example
  - `open-attestation wrap ./raw-document.json --output-file ./wrapped-document.json`

# Issue document



# Issue document







# Issue document (oa-cli)

- Issue documents with wallet.json
  - Command
    - `open-attestation document-store issue --network ropsten --encrypted-wallet-path wallet.json --address <documentStore> --hash <merkleRoot>`
  - Example
    - `open-attestation document-store issue --network ropsten --encrypted-wallet-path wallet.json --address 0x4B563cAE8F6D4E1a31B1e9217C9DD7DC371c0C1D --hash fd8be91d97c41ecCommand6da8579739af43e9a3113759ef39f494c12b8cdf5cad6e123`



# Issue document (metamask)

- Issue documents with metamask wallet account's privateKey
  - Command
    - export OA\_PRIVATE\_KEY=<privateKey>
    - open-attestation document-store issue --network ropsten --address <documentStore> --hash <merkleRoot>
  - Example
    - export OA\_PRIVATE\_KEY=2F1234567
    - open-attestation document-store issue --network ropsten --address 0x4B563cAE8F6D4E1a31B1e9217C9DD7DC371c0C1D --hash fd8be91d97c41ec6da8579739af43e9a3113759ef39f494c12b8cdf5cad6e123

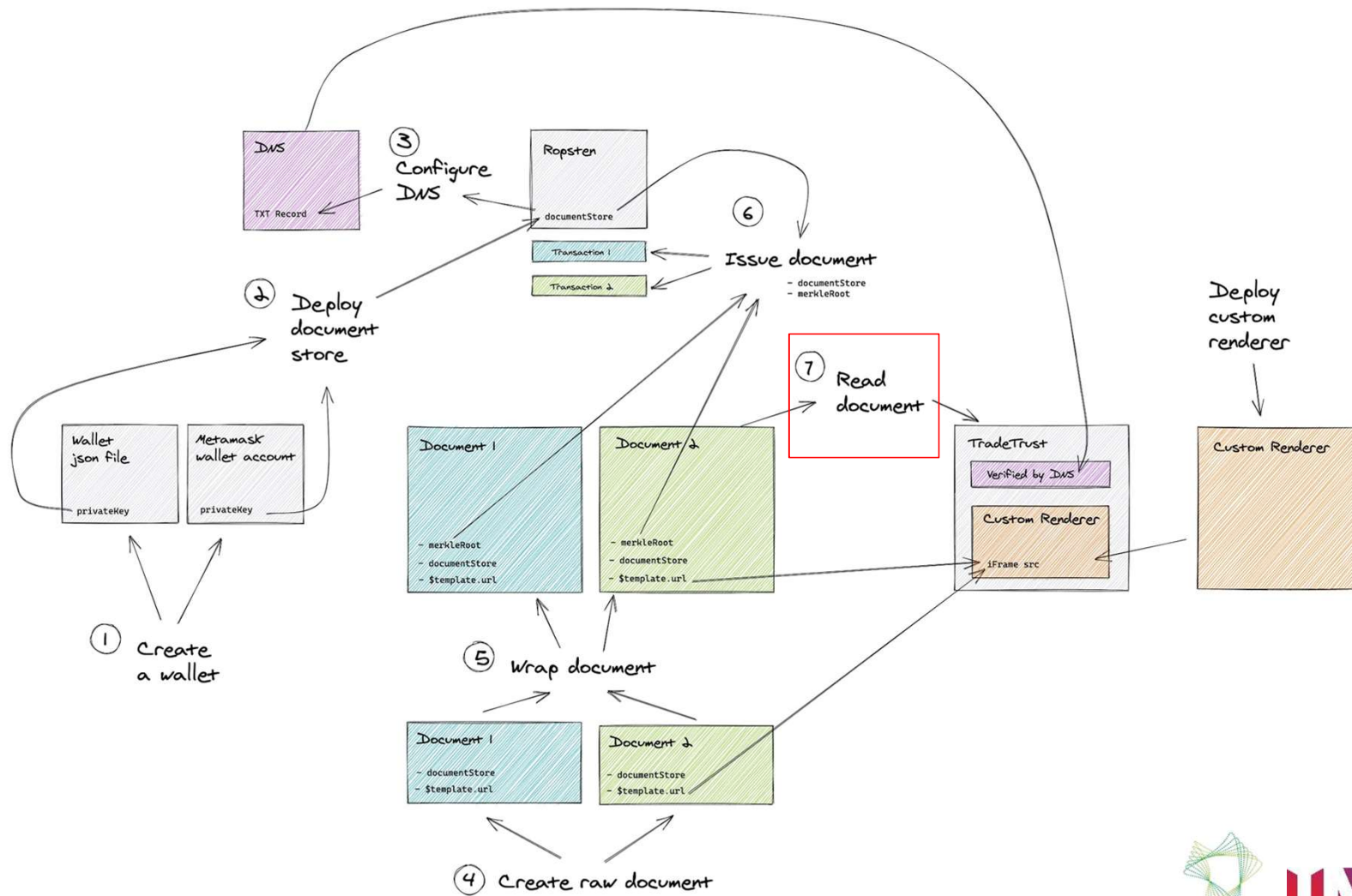
```
SimsMBPgovtech:workshop simboonlong$ open-attestation document-store issue --network ropsten --address
0xF78a7713591517288A950874658728910b1c98dA --key 0x4B563cAE8F6D4E1a31B1e9217C9DD7DC371c0C1D
--hash c04ffd6fba0d3e0f916304e72646097384ec30aa316d27cff4fb616b23831094
[info] Issuing c04ffd6fba0d3e0f916304e72646097384ec30aa316d27cff4fb616b23831094 to document stor
e 0xF78a7713591517288A950874658728910b1c98dA
[warning] Be aware that by using the `key` parameter, the private key may be stored in your machine
's sh history
[warning] Other options are available: using a file with `key-file` option or using `OA_PRIVATE_KEY`
environment variable
[awaiting] Sending transaction to pool
[awaiting] Waiting for transaction 0xb62a6d9ebad66092b8dfa776d6dbe410ad760d029bd2fd8e7dc9a5cc6b9bc6e
6 to be mined
[success] Document/Document Batch with hash c04ffd6fba0d3e0f916304e72646097384ec30aa316d27cff4fb616
b23831094 has been issued on 0xF78a7713591517288A950874658728910b1c98dA
[info] Find more details at https://ropsten.etherscan.io/tx/0xb62a6d9ebad66092b8dfa776d6dbe410ad
760d029bd2fd8e7dc9a5cc6b9bc6e6
SimsMBPgovtech:workshop simboonlong$
```

CLI: Document successfully issued to document store.

# Read document



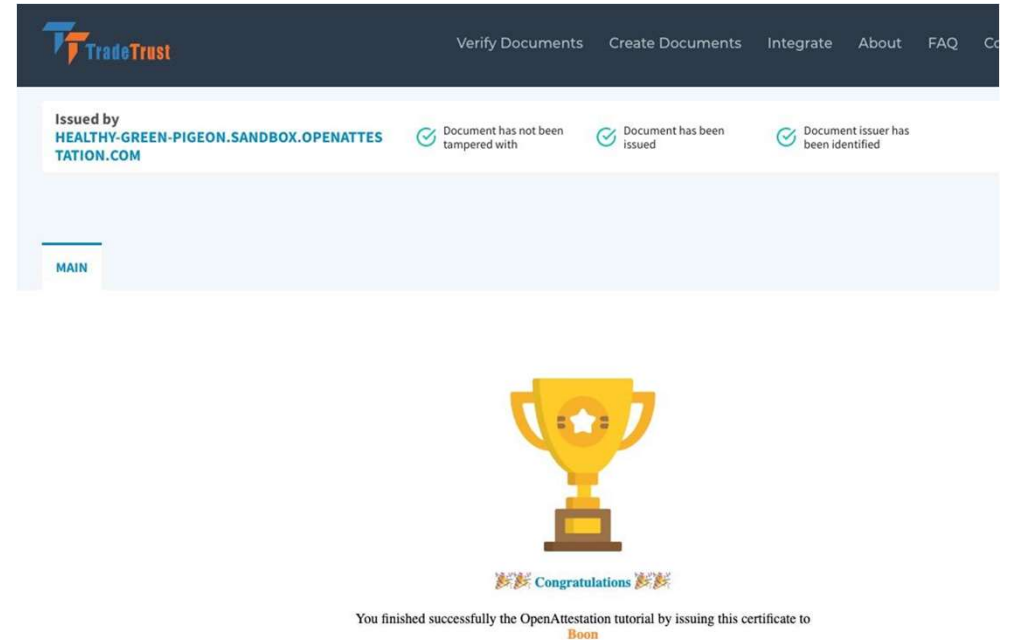
# Read document





## Read document

- Drag and drop your issued wrapped document now, to see the tutorial document renderer in action

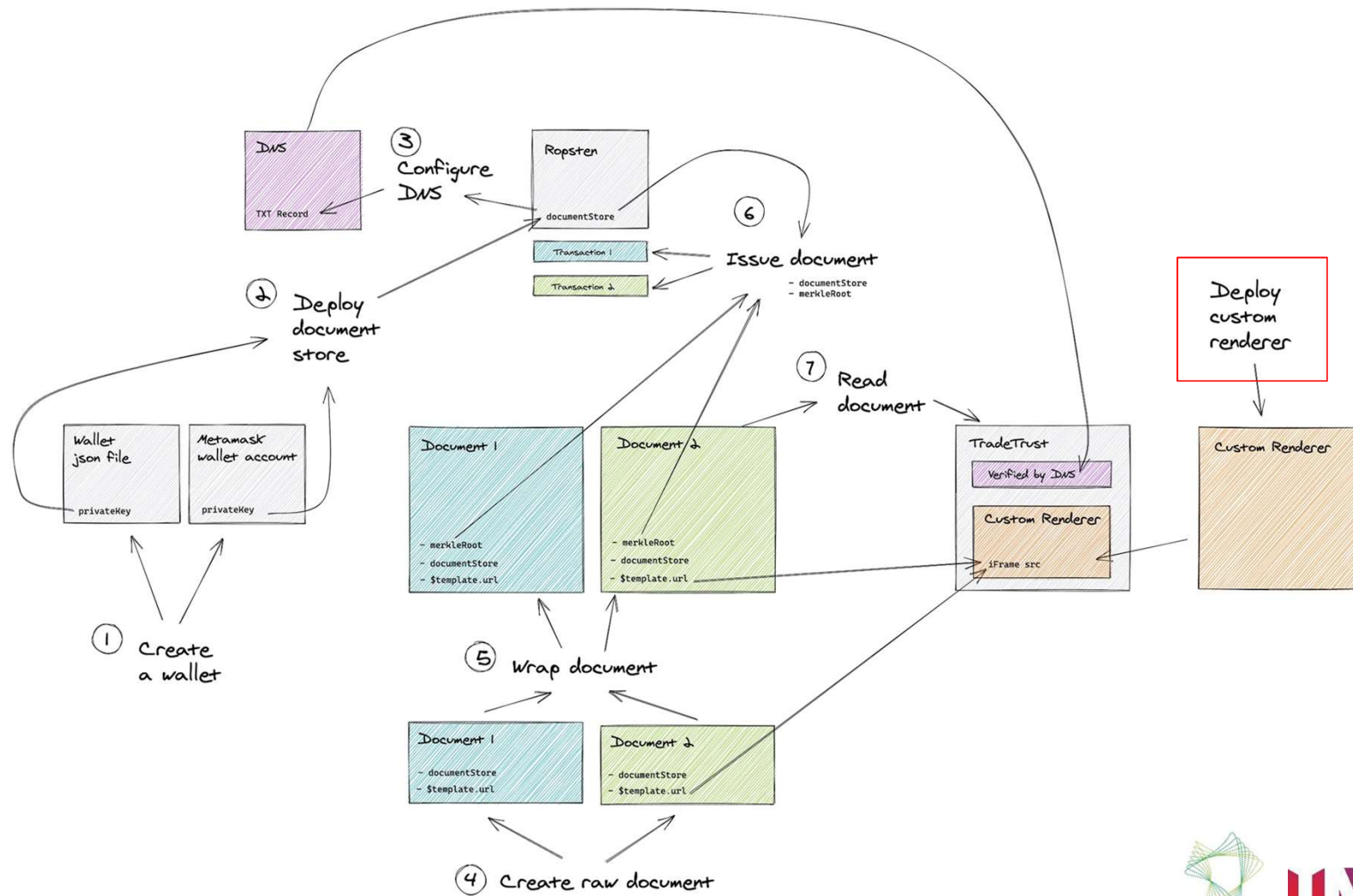


UI: Reading your documents on TradeTrust website.

# Additional topic: Deploy custom renderer

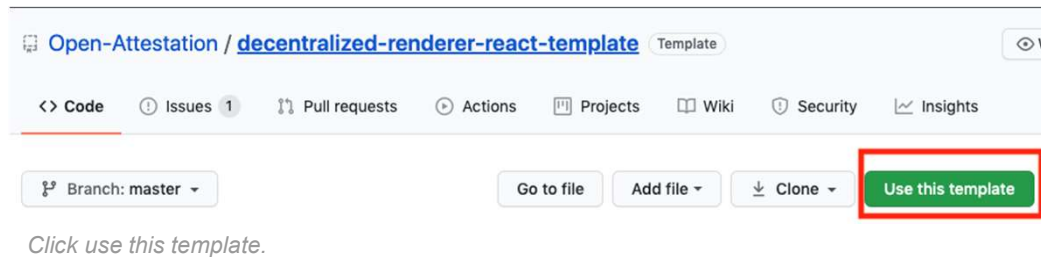


# Deploy custom renderer



# Deploy custom renderer

- Login to Github and use starter template from this repo:
  - <https://github.com/Open-Attestation/decentralized-renderer-react-template>



## Create a new repository from decentralized-renderer-react-template

The new repository will start with the same files and folders as [Open-Attestation/decentralized-renderer-react-template](#).

Owner \* simboonlong

Repository name \* workshop-renderer ✓

Great repository names are short, lowercase, and contain only alphanumeric characters and hyphens. workshop-renderer is available. Inspiration? How about congenial-robot?

Description (optional)

☒ Public Anyone on the internet can see this repository. You choose who can commit.

☐ Private You choose who can see and commit to this repository.

☐ Include all branches Copy all branches from Open-Attestation/decentralized-renderer-react-template and not just master.

Create repository from template

Click create repo from template.



# Deploy custom renderer

- Login to Netlify and add new site
  - Configure netlify access rights to your renderer's repo
  - Add build command + publish directory
    - npm run build
    - dist
- Click deploy to get your public url
  - <https://mystifying-swartz-b01fbb.netlify.app>

The first screenshot shows the Netlify dashboard with a button labeled "New site from Git" highlighted by a red rectangle. Below it, a card for "g's team (6 days ago)" is visible, and a text prompt says "Click New site from Git."

The second screenshot shows the "Create a new site" page. It has a progress bar with three steps: "1. Connect to Git provider", "2. Pick a repository", and "3. Build options, and deploy!". Under "Continuous Deployment: GitHub App", it says "Choose the repository you want to link to your site on Netlify. When you push to Git, we run your build tool of choice on our servers and deploy the result." A search bar shows "simboonlong". Below it, two repositories are listed: "simboonlong/document-renderer-custom" and "simboonlong/git-learn". A red box highlights the link "Configure the Netlify app on GitHub." at the bottom.

The third screenshot shows the "Repository access" section. It has two radio buttons: "All repositories" (unselected) and "Only select repositories" (selected). Below "Only select repositories" is a "Select repositories" button. Under "Selected 3 repositories", three repositories are listed: "simboonlong/workshop-renderer", "simboonlong/git-learn", and "simboonlong/document-renderer-custom". A red box highlights the "simboonlong/workshop-renderer" repository.

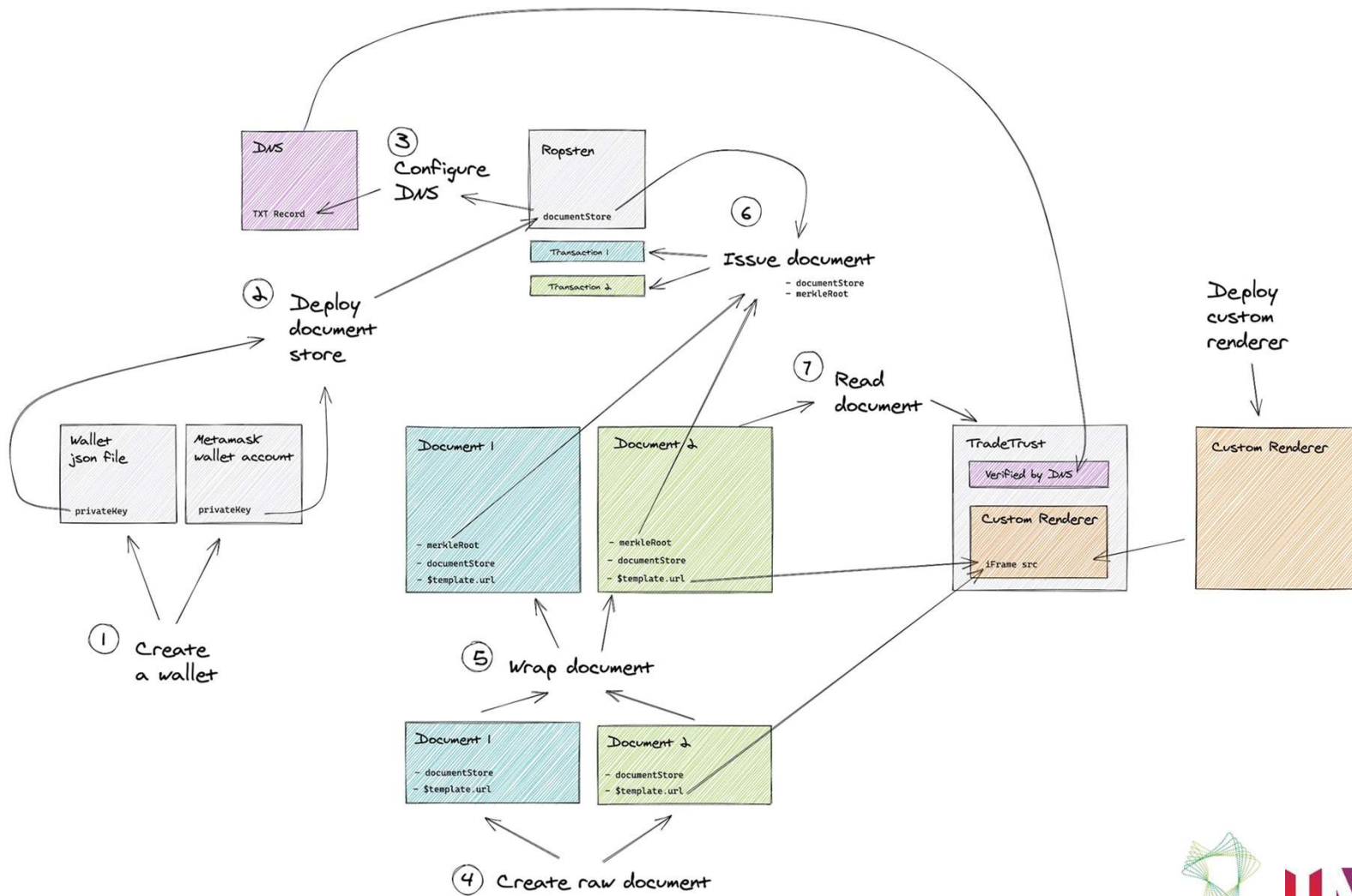
Below the third screenshot, the text "Add Github repo to Netlify." is displayed.



## Deploy custom renderer (let's see)

1. Update new values to raw documents
  - Remember to update \$template.name if it's different : )
1. Run wrap again
2. Issue the documents to blockchain again
3. Drag drop again, you should now see your custom renderer : )
4. Make some edits to your custom renderer and push those up
5. Drag drop to see your reflected changes : )
6. Detailed steps at:
  - <https://openattestation.com/docs/advanced/custom-renderer>

# Recap



# Useful links

## Documentation

- <https://openattestation.com/docs/verifiable-document/overview>

## Open-Attestation CLI

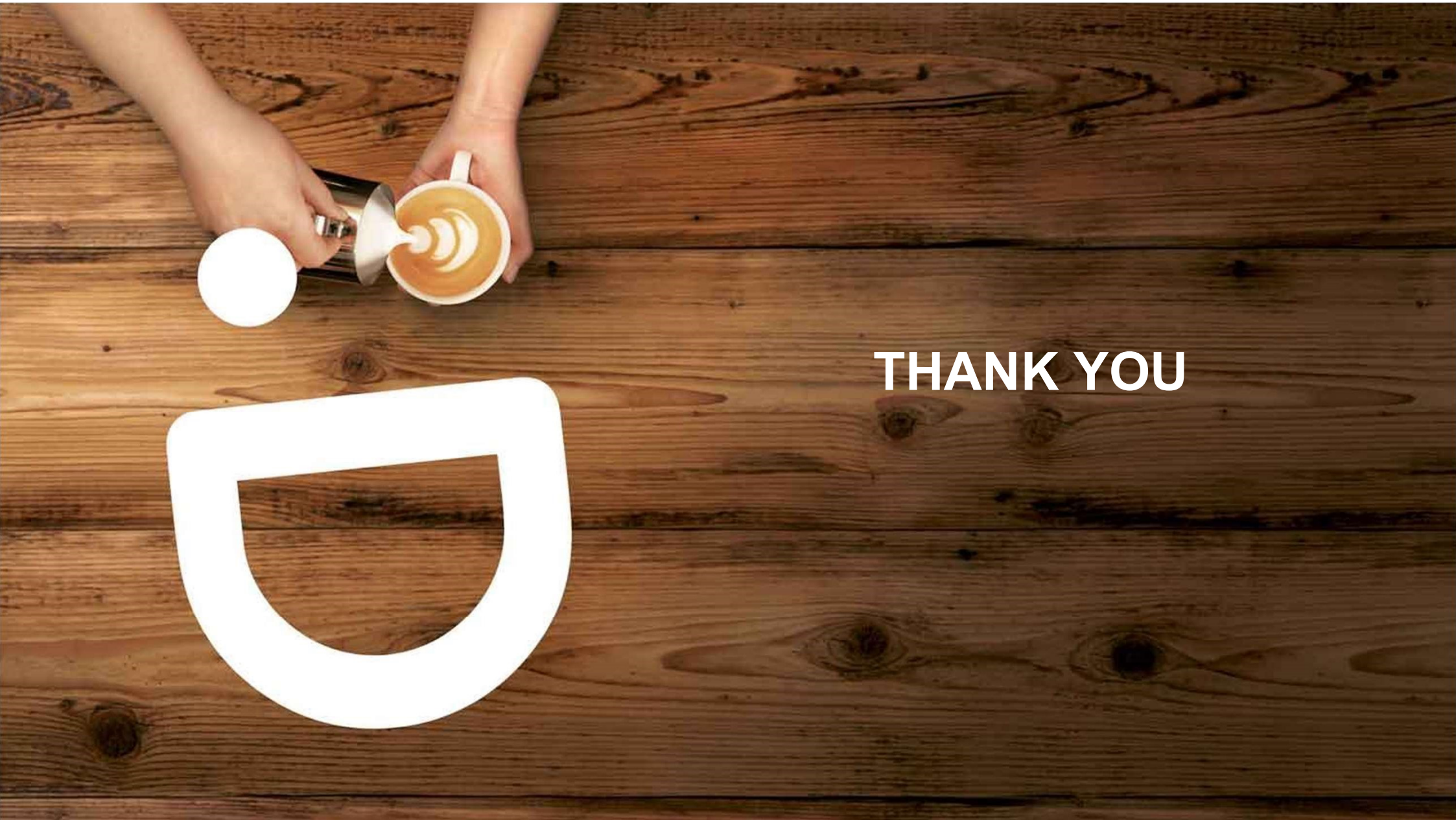
- <https://github.com/Open-Attestation/open-attestation-cli#setup>
- <https://github.com/Open-Attestation/open-attestation-cli#wallet>
- <https://github.com/Open-Attestation/open-attestation-cli#deploying-document-store>
- <https://github.com/Open-Attestation/open-attestation-cli#dns-txt-record>
- <https://github.com/Open-Attestation/open-attestation-cli#wrapping-documents>
- <https://github.com/Open-Attestation/open-attestation-cli#issue-1>

## Renderer Template

- <https://github.com/Open-Attestation/decentralized-renderer-react-template>

## TL;DR

- [https://drive.google.com/drive/folders/117TpQjP5SU0IVsB84A\\_HxUiOGDIjJLW4](https://drive.google.com/drive/folders/117TpQjP5SU0IVsB84A_HxUiOGDIjJLW4)



**THANK YOU**