

TOPOLOGIC, BLOCKCHAIN AND IPFS FOR DIGITAL TWINS

TIMESTAMPS & OTHER STORIES

27 MAY 2024:W3C LINKED BUILDING DATA COMMUNITY GROUP MEETING

DR THEO DOUNAS
UNIVERSITY OF ANTWERP
ADVENTUROUS SYSTEMS

TEAM

Academia



Professor Wassim Jabi

B.Arch, M.Arch, Ph.D

Chair in Computational Methods in Architecture
[Welsh School of Architecture](#)

JabiW@cardiff.ac.uk

+44 29208 75981

Bute Building, Room 1.38, King
Edward VII Avenue, Cardiff, CF10 3NB

Available for postgraduate
supervision



Davide Lombardi

Head, Senior Associate Professor
[Department of Architecture](#)

Phone

+86 (0)512 81880463

Email

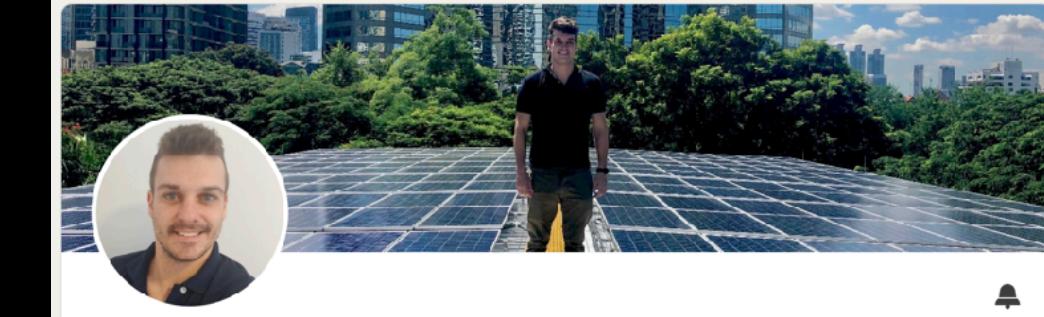
Davide.Lombardi@xjtu.edu.cn

[View Scopus Profile](#)

2015-

2017-

Adventurous Systems (2023-)



Hico Mc Donald

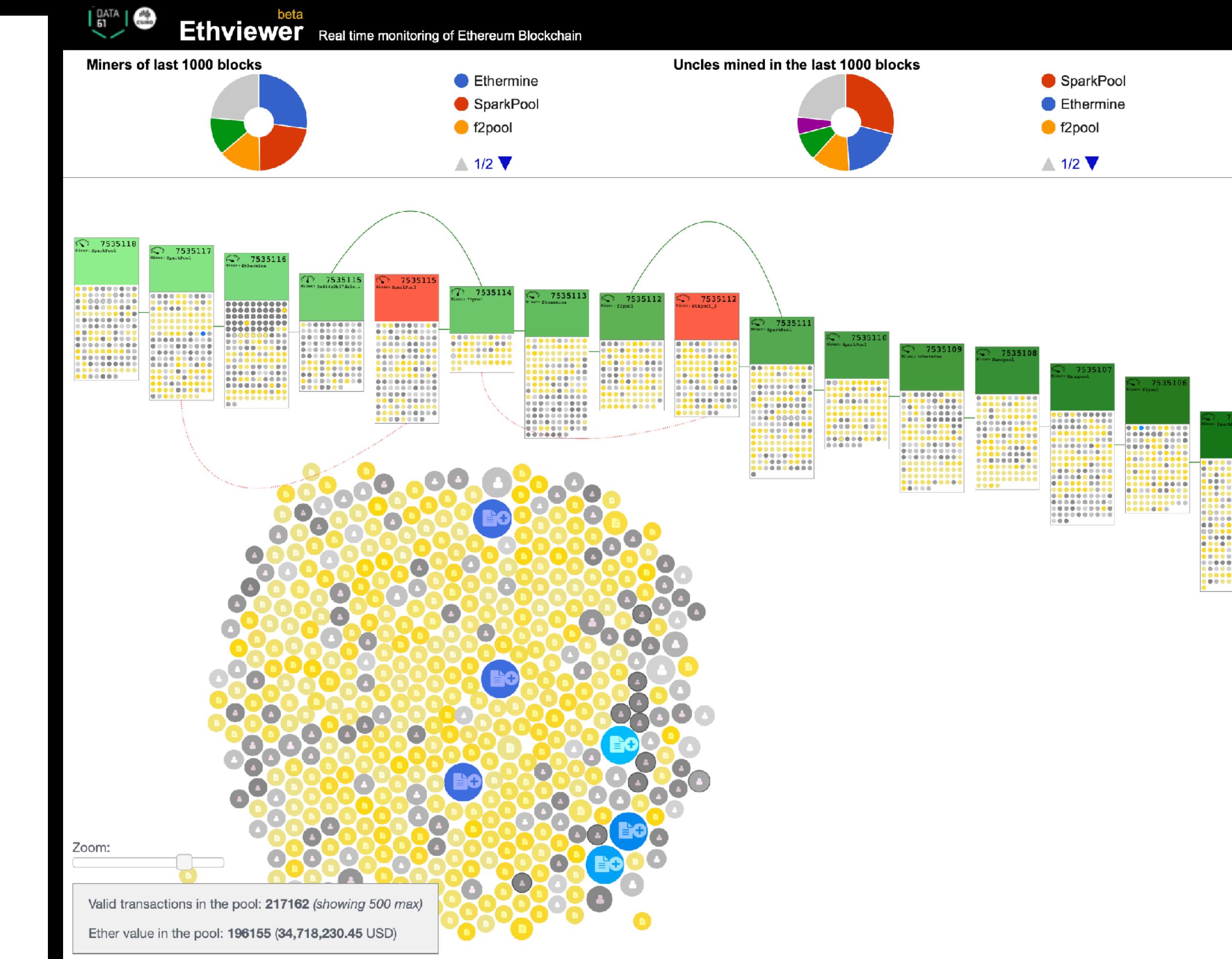
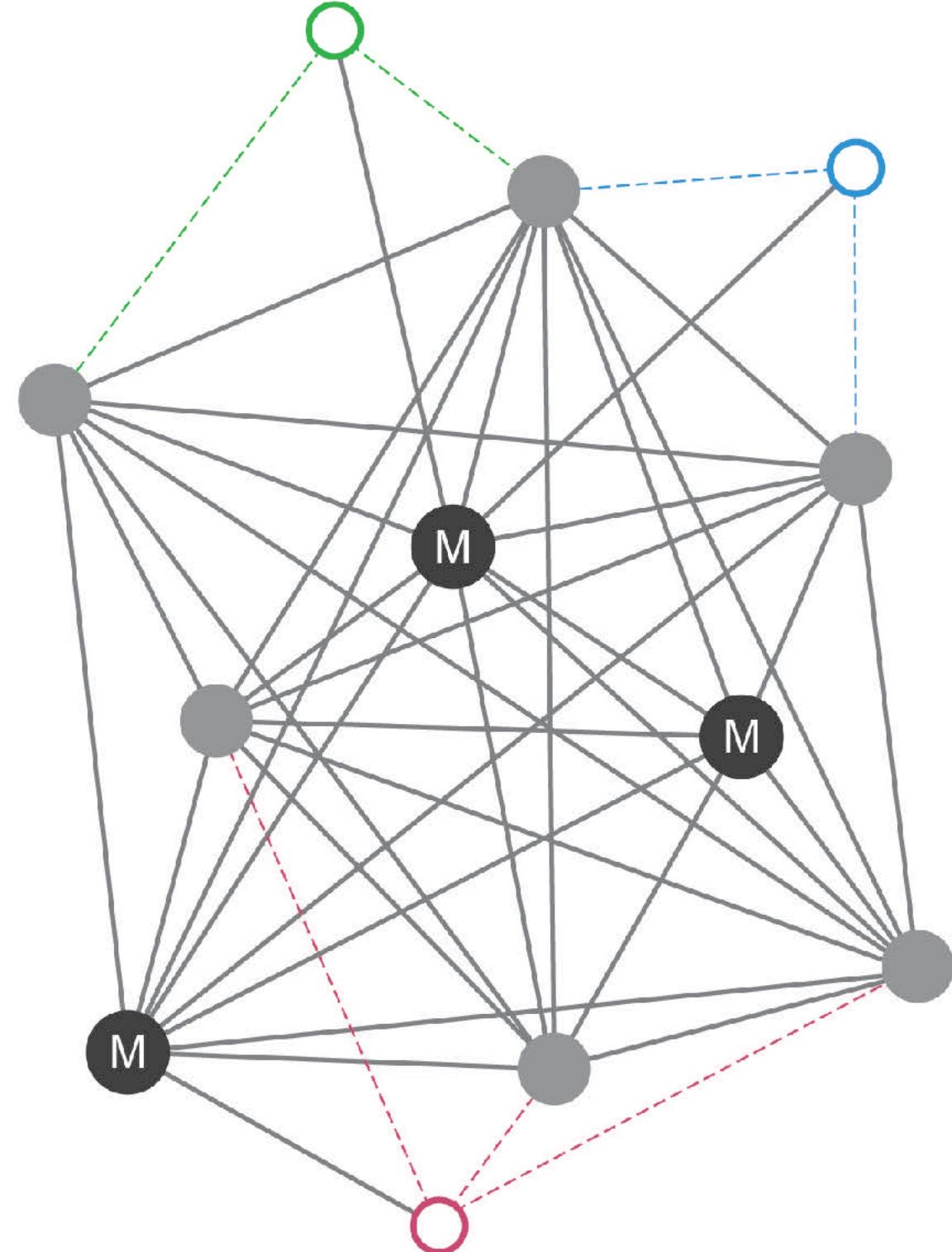
Web3 Partnership Manager | Construction Project Manager | Web3
Blockchain Developer | Civil Engineer | React & Solidity

Cape Town, Western Cape, South Africa - [Contact info](#)

588 followers · 500+ connections

2021-

BLOCKCHAIN AND SMART CONTRACTS



DESIGN
OF ECONOMIC UTILITY



TOKENS

(WHY) BLOCKCHAIN AND SMART CONTRACTS IN AEC

- Immutability
- Trust & Security
- Governance: Common Pool Resources Problems (design and construction projects, circular economies)
- Incentive alignment amongst stakeholders

WHY NOT BLOCKCHAIN

- Very Expensive to store data
- Do not make sense for projects with one stakeholder or aligned incentives.

IPFS

QmHash/bafyHash

Browse

?

?

78MiB FILES 311MiB ALL BLOCKS + Import

Files

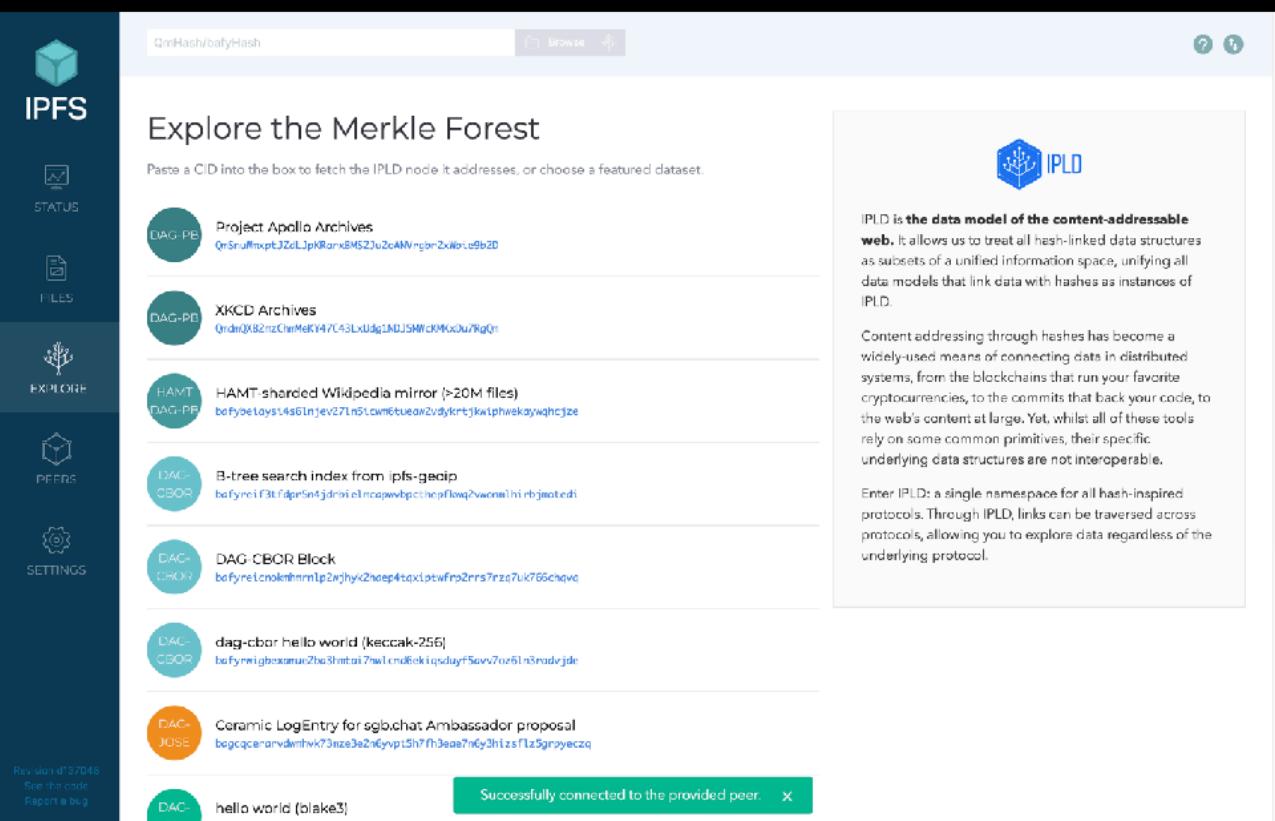
Name ↑	Pin Status	Size
apertures.dxf		30 KIB
furniture.dxf		38 KIB
house.dxf	⚡	24 KiB
house.ttl	⚡	58 KiB
Prvok3.gcode		6 MiB
Spotlight_on_blockchain_2.key		35 MiB
Spotlight_on_blockchain.key		35 MiB
UCHYTKA_0.1mm_PLA_MK3S_35m.gcode	⚡	2 MiB

Imported 4 items

apertures.dxf	30 KiB	✓
furniture.dxf	38 KiB	✓
house.	KiB	✓

Successfully connected to the provided peer.

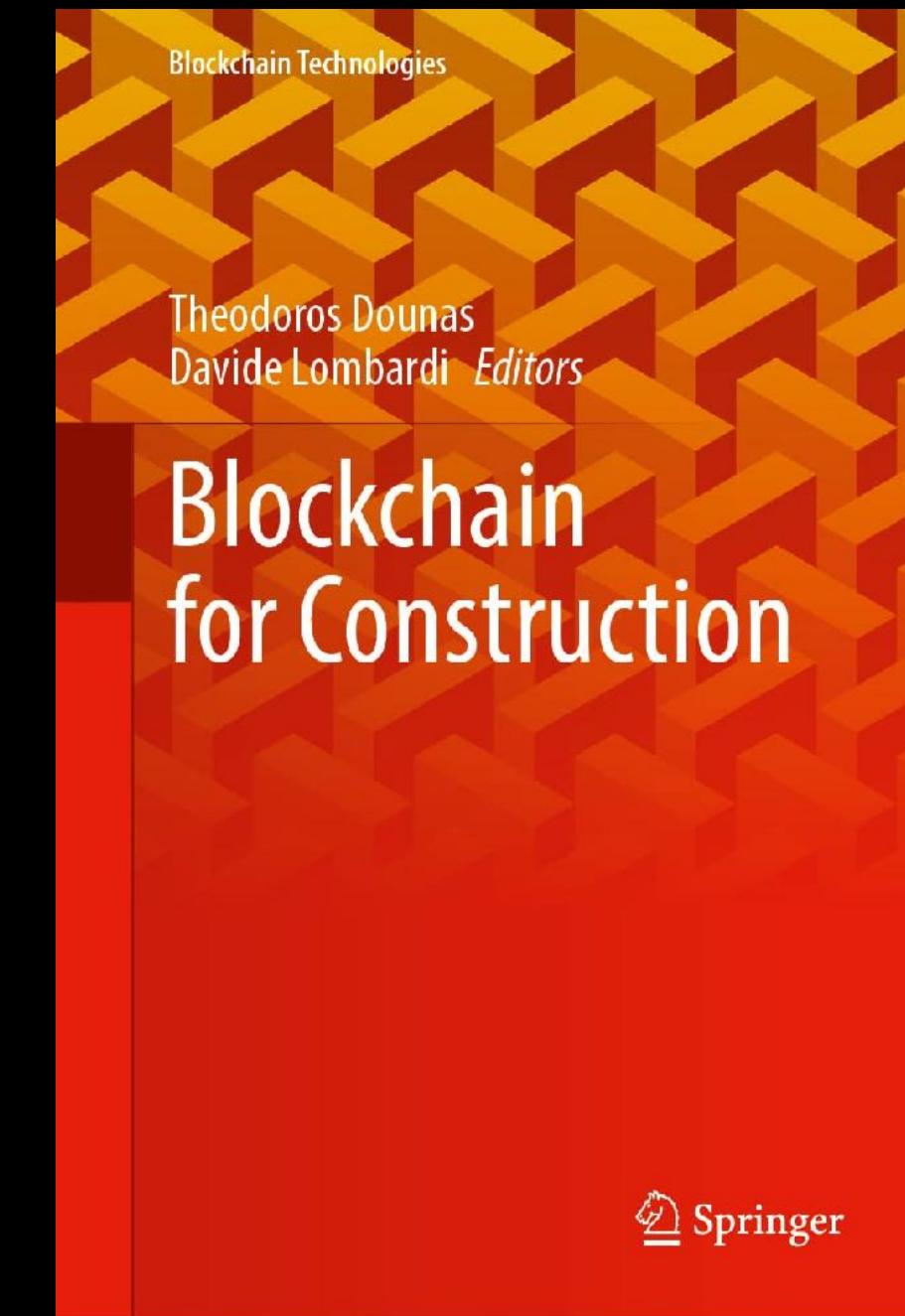
INTERPLANETARY FILESYSTEM



- Role: data storage -Content addressable, rather location addressable
- Infrastructure to host a website, digital twin, file etc
- Read and write linked data to IPFS
- For our purposes: Store linked data, geometry data and other objects

EARLIER WORK

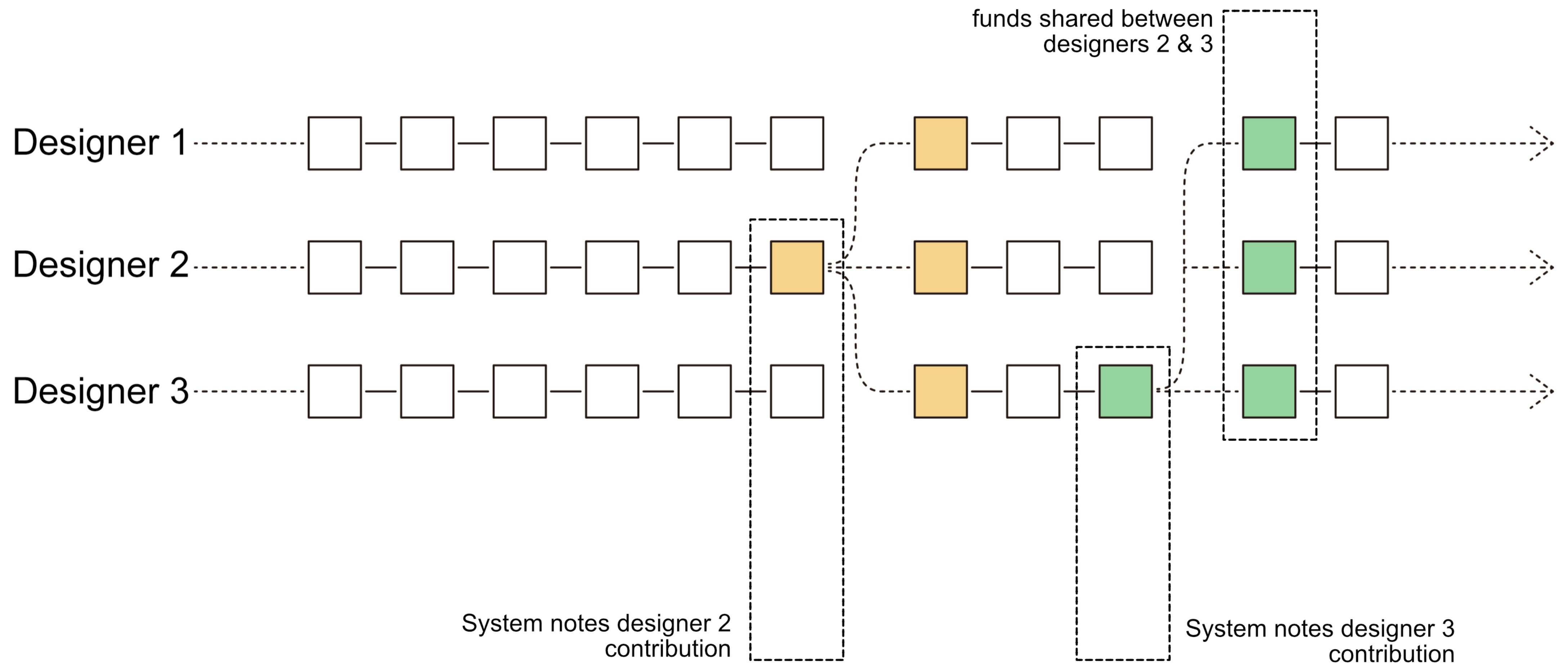
- Blockchain for Architectural Design
- Topologic tokens for Circular Economy
- Collective Digital Factories [Decentralised Autonomous Organisations]
- Crypto-Twins



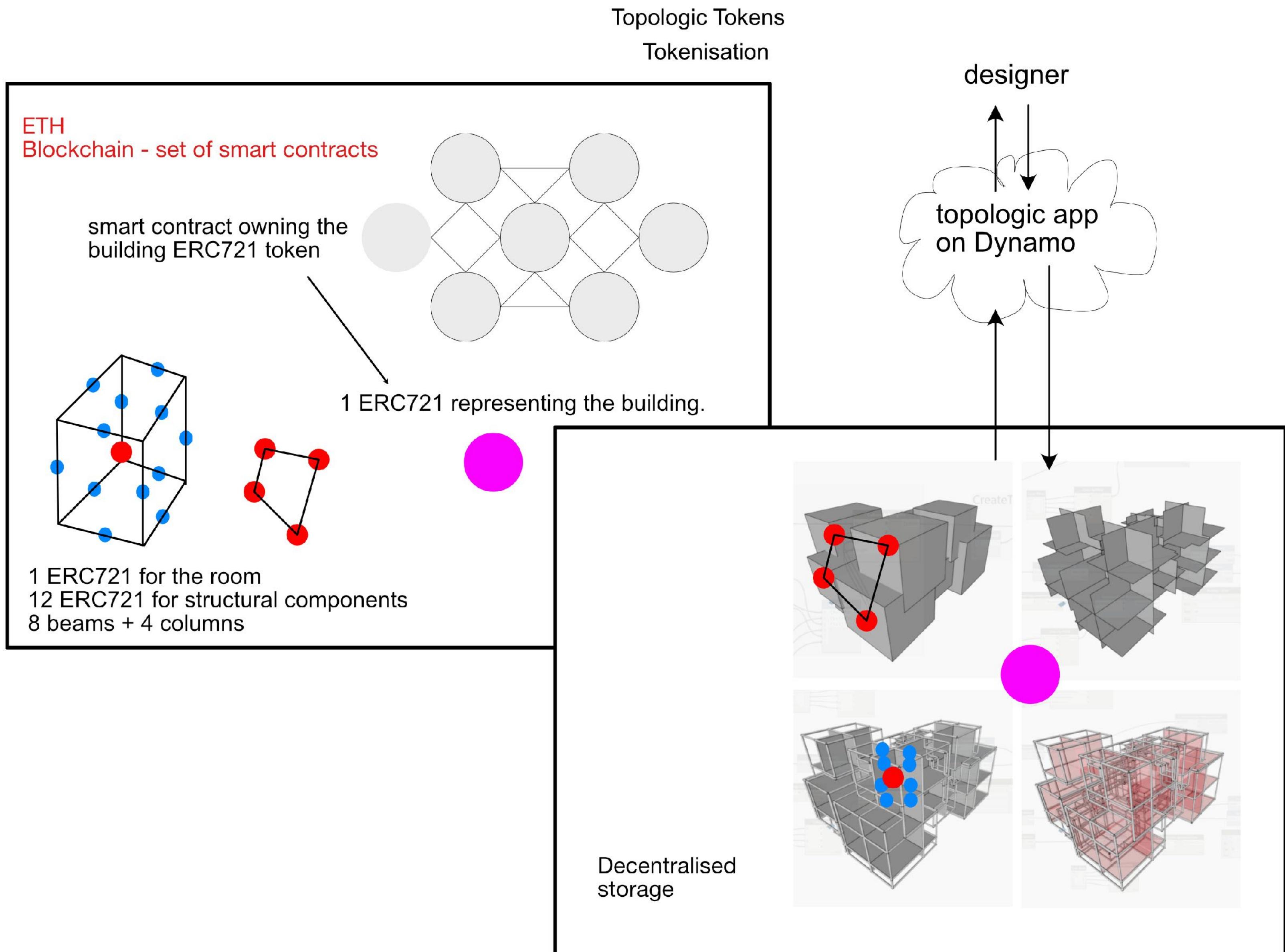
Springer Verlag 2022

ISBN 9811937583 / 9789811937583

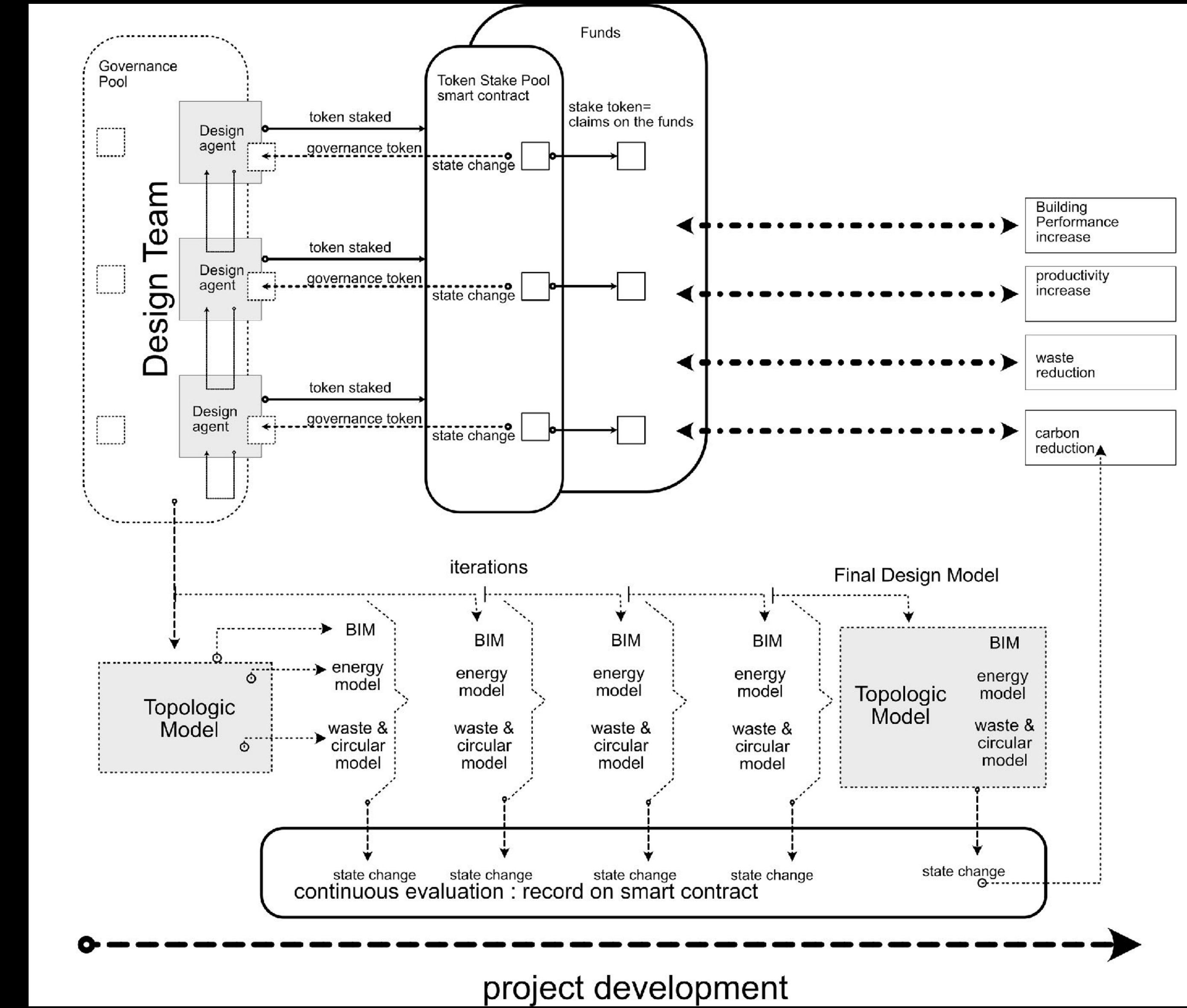
BLOCKCHAIN FOR ARCHITECTURAL DESIGN



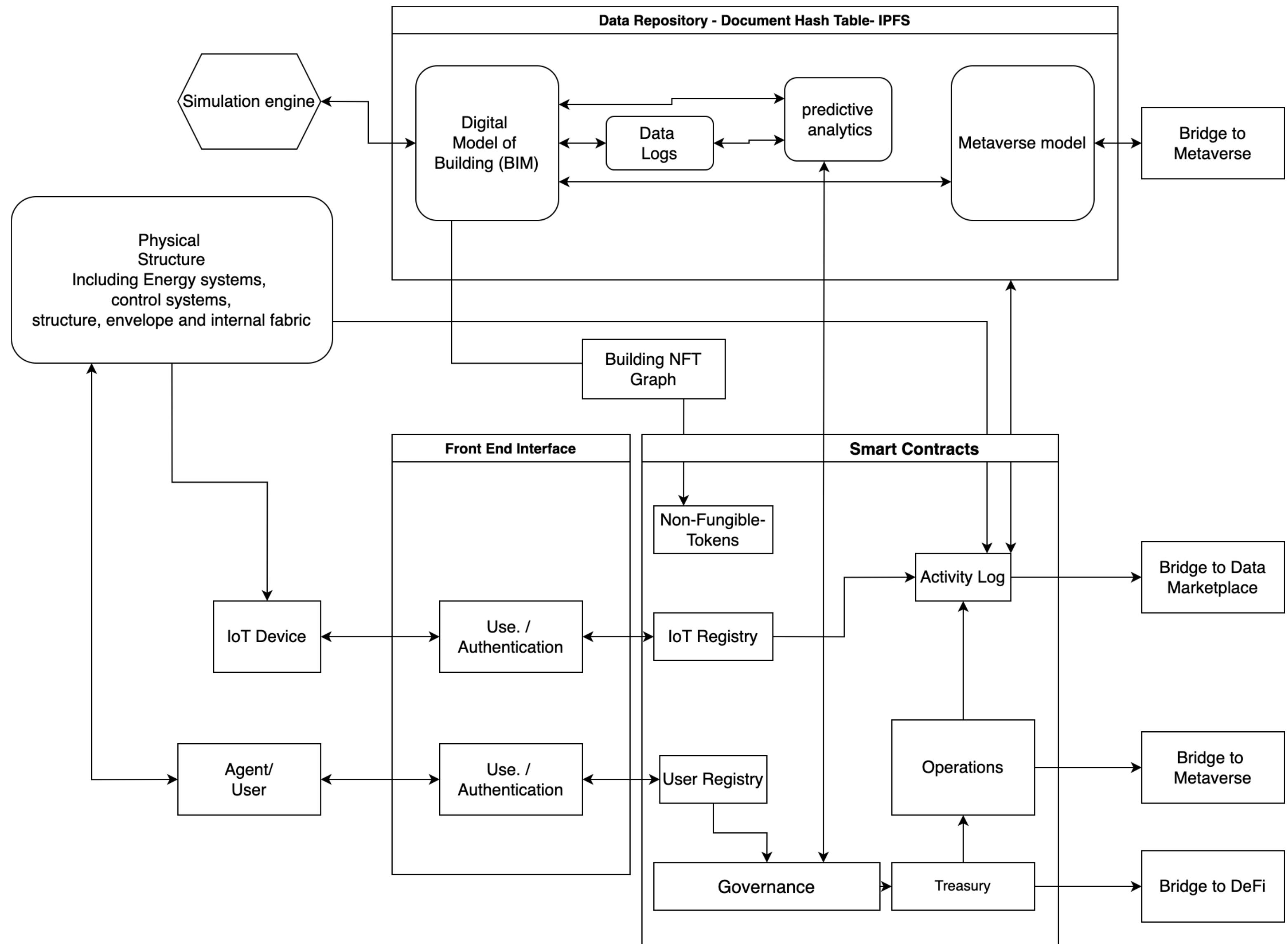
TOPOLOGY TOKENISATION OF BUILDINGS 2021 - CIRCULAR ECONOMY



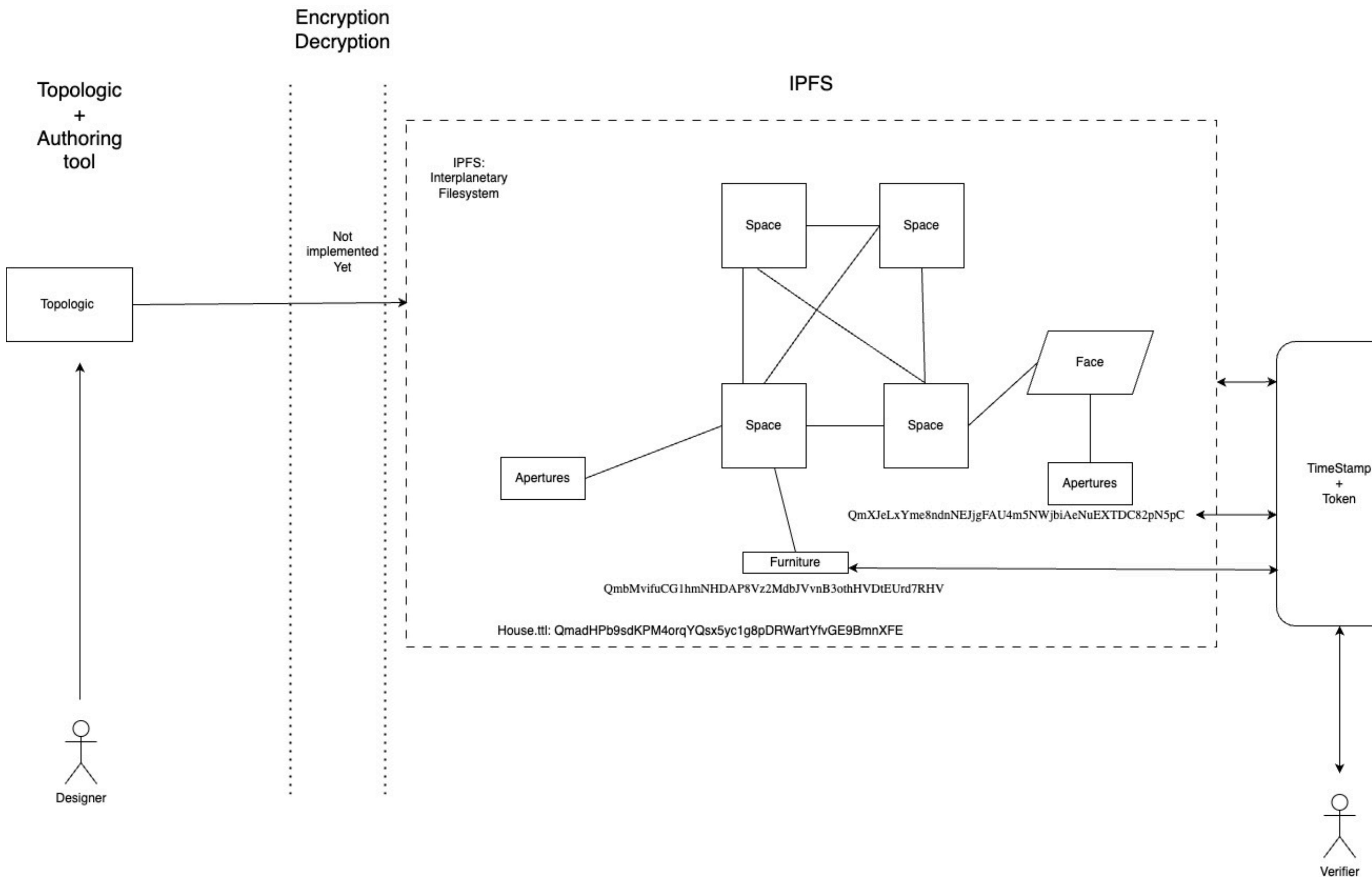
COLLECTIVE DIGITAL FACTORIES



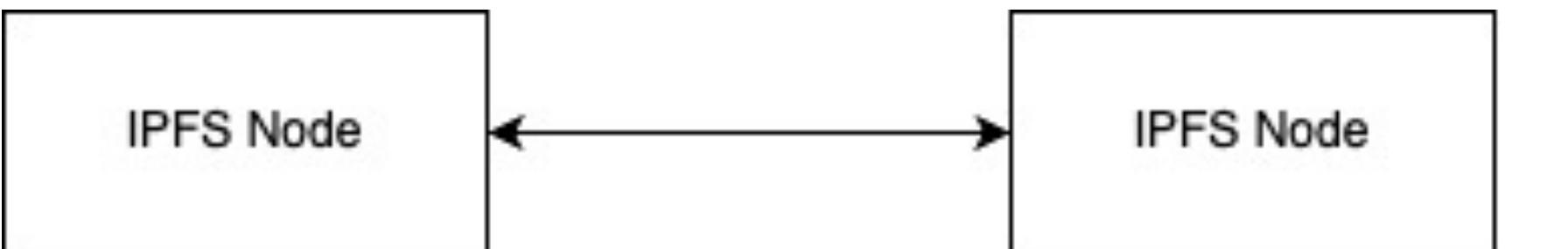
CRYPTO-TWINS



TOPOLOGIC+IPFS+SMART CONTRACTS



IPFS P2P



Terminal window showing Python code for receiving files from an IPFS node:

```

45     "path": "blocks",
46     "shardFunc": "/repo/flatfs/shard/v1/next-to-last/2",
47     "sync": true,
48     "type": "flatfs"
49   },
50   "mountpoint": "/blocks",
51   "prefix": "flatfs.datastore",
52   "type": "measure"
53 },
54 {
55   "child": {
56     "compression": "none",
57     "path": "datastore",
58     "type": "leveldb"
59   },
60   "mountpoint": "/",
61   "prefix": "leveldb.flatfs",
62   "type": "measure"
63 }
64 ],
65 "type": "mount"
66 },
67 "StorageCWatermark": 90,
68 "StorageMax": "10GB"
69 },
70 "Discovery": {
71   "Mdns": {
72     "Enabled": true
73   }
74 },
75 "Experimental": [
76   "FilestoreEnabled": false,
77   "Libp2pStreamMounting": true,
78   "OptimisticProvide": false,
79   "OptimisticProvideJobsPoolSize": 0,
80   "P2pHttpProxy": false,
81   "StrategicProviding": false,
82   "UrlstoreEnabled": false
83 },
84 "Gateway": {
85   "APICommands": [],
86   "DeserializedResponses": null,
87   "DisableTMLErrors": null,
88   "ExposeRoutingAPI": null,
89   "HTTPHeaders": {},
90   "NoDnsLink": false,
91   "NoFetch": false,
92   "PathPrefixes": [],
93   "PublicGateways": null,
94   "RootRedirect": ""
95 },
96 "Identity": {
97   "PeerID": "12D3KooWu72wB1pKAlhN3ySt5dTSQBzvEKAJcF5DvJPTmZ2vyks"
98 },
99 "Internal": {},
100 "Ips": {
33
34   a = input("Press any key to exit...")
35
36   # when you no longer need to listen for incoming files, clean up resources:
37   file_receiver.terminate()
  
```

Terminal window showing Python code for receiving files from an IPFS node:

```

45     "path": "blocks",
46     "shardFunc": "/repo/flatfs/shard/v1/next-to-last/2",
47     "sync": true,
48     "type": "flatfs"
49   },
50   "mountpoint": "/blocks",
51   "prefix": "flatfs.datastore",
52   "type": "measure"
53 },
54 {
55   "child": {
56     "compression": "none",
57     "path": "datastore",
58     "type": "leveldb"
59   },
60   "mountpoint": "/",
61   "prefix": "leveldb.flatfs",
62   "type": "measure"
63 }
64 ],
65 "type": "mount"
66 },
67 "StorageCWatermark": 90,
68 "StorageMax": "10GB"
69 },
70 "Discovery": {
71   "Mdns": {
72     "Enabled": true
73   }
74 },
75 "Experimental": [
76   "FilestoreEnabled": false,
77   "Libp2pStreamMounting": true,
78   "OptimisticProvide": false,
79   "OptimisticProvideJobsPoolSize": 0,
80   "P2pHttpProxy": false,
81   "StrategicProviding": false,
82   "UrlstoreEnabled": false
83 },
84 "Gateway": {
85   "APICommands": [],
86   "DeserializedResponses": null,
87   "DisableTMLErrors": null,
88   "ExposeRoutingAPI": null,
89   "HTTPHeaders": {},
90   "NoDnsLink": false,
91   "NoFetch": false,
92   "PathPrefixes": [],
93   "PublicGateways": null,
94   "RootRedirect": ""
95 },
96 "Identity": {
97   "PeerID": "12D3KooWu72wB1pKAlhN3ySt5dTSQBzvEKAJcF5DvJPTmZ2vyks"
98 },
99 "Internal": {},
100 "Ips": {
33
34   a = input("Press any key to exit...")
35
36   # when you no longer need to listen for incoming files, clean up resources:
37   file_receiver.terminate()
  
```

```

[18]: ft = ipfs_datatransmission.transmit_file( #we can add here progress update and metadata from this example: https://github.com/em
      filepath, peer_id, "my_apps_filelistener")
if ft:
    print("Started Transmission")
else:
    print("Failed to start transmission")
  
```

Started Transmission

PREREQUISITES

- IPFS toolkit
- Ethereum Virtual Machine compatible Blockchains (we use ETH testnets, but any EVM compatible BC will work)
- Smart Contract is written in Solidity
- Either Python or javascript (libraries for communicating with a blockchain node come in either python or js versions web3.py - web3.js / ether.js)