

TOPOLOGY GENERATED NON-FUNGIBLE TOKENS

Blockchain as infrastructure for a circular economy in architectural design

THEODOROS DOUNAS¹, WASSIM JABI² and
DAVIDE LOMBARDI³

¹*Robert Gordon University*

¹*t.dounas@rgu.ac.uk*

²*Cardiff University*

²*Jabiw@cardiff.ac.uk*

³*Xian Jiaotong Liverpool University*

³*Davide.Lombardi@xjtlu.edu.cn*

Abstract. The paper presents a new digital infrastructure layer for buildings and architectural assets. The infrastructure layer consists of a combination of topology graphs secured on a decentralised ledger. The topology graphs organise non-fungible digital tokens which each represent and correspond to building components, and in the root of the graph to the building itself. The paper presents background research in the relationship of building representation in the form of graphs with topology, of both manifold and non manifold nature. In parallel we present and analyse the relationship between digital representation and physical manifestation of a building, and back again. Within the digital representations the paper analyses the securing and saving of information on decentralised ledger technologies (such as blockchain). We then present a simple sample of generating and registering a non-manifold topology graph on the Ethereum blockchain as an EC721 token, i.e. a digital object that is unique, all through the use of dynamo and python scripting connected with a smart contract on the Ethereum blockchain. Ownership of this token can then be transferred on the blockchain smart contracts. The paper concludes with a discussion of the possibilities that this integration brings in terms of material passports and a circular economy and smart contracts as an infrastructure for whole-lifecycle BIM and digitally encapsulates of value in architectural designPlease write your abstract here by clicking this paragraph.

Keywords. Blockchain; Tokenisation; Topology; Circular Economy; decentralisation.

1. Introduction

Building information Modelling has been presented as a whole-lifecycle paradigm that encompasses all aspects of the life of a building asset, from conception, to design to construction and then operations. While there has been massive progress

in the past twenty years in terms of the information that is digitally available to architects and designers, to optimise building design and construction, the impact of the existing problems has been accentuated: building waste (Adams 2017), the impact of the embodied carbon in built environment on climate (Anderson 2014) and the need for integration of digital and physical systems in architectural design (Dounas et al, 2020). Building Information Modelling platforms have the tendency to have a high computational footprint, further increasing complexity within the information space that an architect needs to navigate and process (Aish 2018). However recent examples of lightweight, modular, computational design processes provide an alternative view towards reducing the information complexity needed in architectural design, in a computational paradigm that resembles the Unix tools chain set.

2. Background

2.1. TOPOLOGY, GRAPHS AND TOPOLOGIC

Central to the idea of relational architectural representation is the idea of topology. Within that one distinguishes between manifold topology (Kantor 2005) and non-manifold topology (AIsh et al 2018). Within the birth of topology by Euler, when he asked whether a person could cross all bridges of today's Kaliningrad by crossing each bridge only once, lies the elegance of topological representation: one uses topology to represent an object, economically. Libraries for Non-Manifold topology, such as Topologic, (Jabi 2019) allow the connection of topology with energy or structural simulations, massively reducing the computational footprint and complexity of the building representation. At the core of the Topologic idea lies the notion that "Buildings enclose and partition space and are built from assemblies of connected components" (Aish et al 2018). For each cell in a building, one can generate a non-manifold topology where a cell is analysed into 12 edges, 6 planes of 4 walls and 2 horizontal planes, plus at least one opening. [Figure 1]

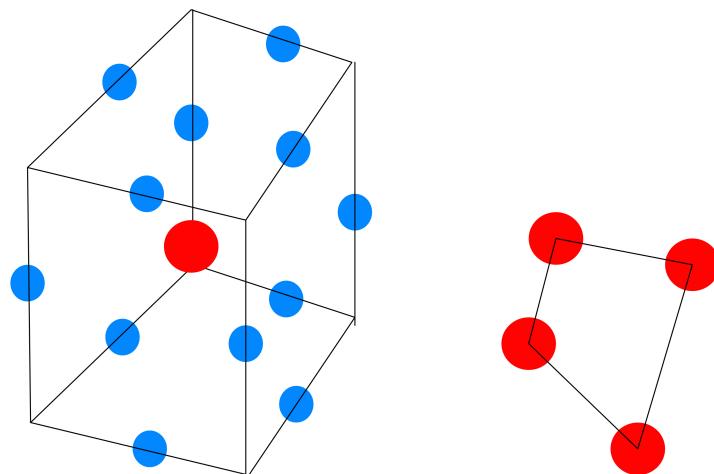


Figure 1. Figure 1: Topological representation of a cell complex- each blues dot represents a cell face or edge, while the red dot represents the cell. The four dots represent then a cell-complex of four cells.

2.2. BLOCKCHAIN, SMART CONTRACT AND TOKENS

Distributed ledgers are essentially distributed databases where participants in the database hold copies of it and there is a mechanism to synchronise and achieve consensus between these copies. The difficulty to achieve consensus in a decentralised ledger is due to the possibility that at least two copies of the ledger might have values that are different added to it. Consensus mechanisms follow in many cases scenarios of the “byzantine Generals problem” (Nakamoto 2009). Blockchains are a special version of distributed ledgers which got initially invented to facilitate the idea behind digital cash in Bitcoin (Nakamoto 2009). Inspired by the Bitcoin blockchain, where the blockchain operates additions and subtractions, A subset of bitcoin programmers created the Ethereum blockchain which acts as a decentralised, global, distributed computing platform capable of any Turing Complete computation.

A blockchain is a decentralised database, where multiple nodes on a computer network hold part of the data or the full data set. Due to its decentralised and distributed nature, Blockchains require a mechanism to synchronise the nodes, and have a single version of the data emerge. Thus algorithmic measures have been implemented that are based on difficult to solve cryptographic problems, or in a collective signing of the next version of the data using again difficult to tamper with cryptography. The former consensus mechanism is called Proof

of Work, since the miners, i.e the validators of the data solve and algorithm, while the latter Proof of Stake since the validators signing the data “stake” an amount of cryptocurrencies as incentive for participation. After validation of the chain, a new block is “minted”, i.e created, which is cryptographically connected with the previous block, as each new block contains a cryptographic hash of the previous one. This ensures that no single node can re-edit the data contained in the transactions but also incentivises the network participants to strive for truth, as each true block minted is rewarded with new cryptocurrency. Beyond tallying cryptocurrencies transactions blockchains can also automatically execute pieces of code called “smart contracts” and it is this feature that makes blockchains compelling in a design and computation context. Our main implementation is based on the Ethereum blockchain [Antonopoulos 2018] because it offers currently various advantages: it has a full toolset to develop smart contracts on, excellent tooling for developers, and full features for the user interfaces tools, where one can write code in javascript frameworks or on python, taking advantage of the respective libraries. From a computational design point of view, we use the Ethereum blockchain as a state machine, i.e a Turing complete machine that allows external input to alter its state. As such, with the feature of smart contracts it is perfectly possible to record and execute code on the Ethereum blockchain, either using python, or the native, purpose built language of Ethereum, solidity. Beyond the hype, reports have described the potential of Blockchain (Cooper 2018) to profoundly affect the digital infrastructure (Kinnaird et al 2017) that runs the build environment but also the digital tools that we currently use to design architecture. Ethereum smart contracts are essentially the equivalent of classes of code that execute specific functions. They are inheritable, can act as factories or libraries for other contracts, and normally one would need a collection of smart contracts to build complex software constructs. Each contract resides in a distinct Ethereum address and is addressable by sending a transaction to that address, with or without invoking a specific function of the contract.

With the use of Smart contracts and the immutability of the blockchain, the creation of digital currency in the form of digital tokens is possible, for example by using the ERC20 standard on Ethereum (Vogelsteller et al. 2015). In parallel, the creation of distinguishable, non-fungible, unique tokens, is possible as well by using the ERC721 standard (Entriken et al. 2018). We present both token standards so that the distinctiveness and uniqueness of ERC721 can be made clear to the reader.

2.2.1. *ERC20*

ERC20 tokens issued by a smart contract can be interchangeable between them, in the same manner that currency has: An ERC20 Token of type A is equal and indistinguishable with another Token A, and there is no manner in which one can distinguish on the blockchain between ERC20 tokens of the same type. The Ethereum foundation that governs the standards for the Ethereum blockchain, has defined three optional and six mandatory rules that ERC-20 tokens should follow so they can adhere to the standard. The optional are the Token name, Symbol, and allowing for decimal subdivision, up to 18 decimal places. The mandatory rules

are:

1. The “totalSupply” i.e the designer of the token has to define how many tokens all together will exist and this has to be a finite number.
2. The “BalanceOf” records how many tokens an account has
3. The “Transfer” allows for the transfer of the token
4. The “TransferFrom” records the initialling account
5. The “Approve” cross-checks a transaction against the total number of tokens recorded in the blockchain
6. The allowance checks the balance of an account before a transaction takes place and will cancel a transaction before it takes place

2.2.2. ERC721

ERC721 Tokens are the non-fungible tokens (NFT) that are encapsulated inside a smart contract. Non-Fungible means that one token is not exchangeable with another. Within a smart contract ERC721 are represented using “structs”, a computational entity that can contain a series of other properties in the form of variables. The ERC721 specification itself describes that one of the potential uses of the standard is the representation of physical objects, such as houses or unique artwork, where one ETH address on the blockchain network has ownership of the token, including the potential for an ERC721 token to be owned by a contract. ERC721 use a unique numerical identifier in the form of an unsigned integer. The combination of the contract address and the unique identifier (ETH Address, uint) stands for a global identifier for the token, as ETH addresses are also unique. Key property of the ERC721 is that it is transferable (Openzeppelin 2020).

3. Topology and ERC721: Implementation.

Within our computational analysis, we use a series of encapsulated graphs to represent buildings, using the topology library topologic.app. Each node in the root graph corresponds to a space in an existing building or a building under design. The second layer of graphs stems the initial graph, and in a series of layers and connections of nodes to components of the building, represent in graph form the whole ontology of the building. Consequently a series of tokens that represent the nodes to the graph get created on a contract we control on the Ethereum Blockchain. While tokens that represent building components can be interchangeable, the tokens that represent unique nodes in the building, and the root of the graph, are non-fungible, and unique, i.e. not interchangeable. [Figure 2]

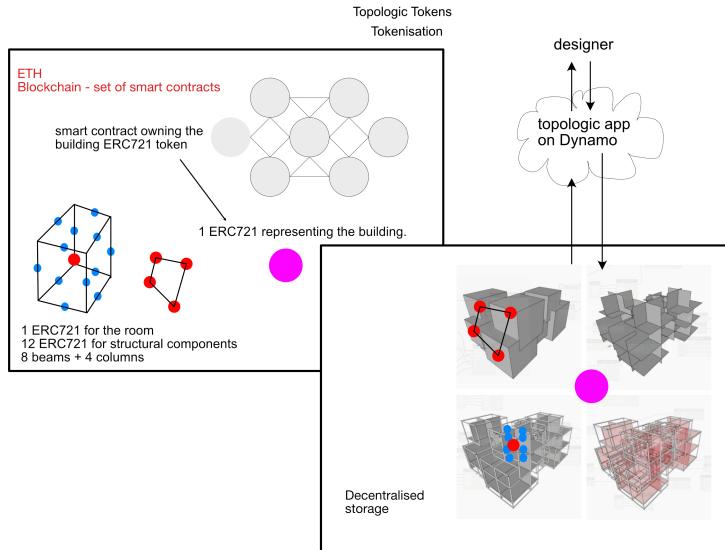


Figure 2. Figure 2 : Conceptual representation of the Topologic Token prototype.

Our prototype implementation involves executing a simple topologic definition in Dynamo that generates a cell complex of nine cubes, side-10, and then topologically analysing each of the cells into the faces and edges that comprise the cell. [Figure 3]

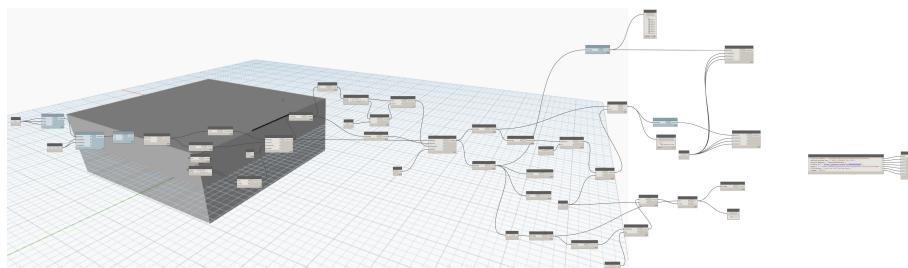


Figure 3. Figure 3: Generating a 9-cell complex on dynamo using topologic-analysing the cell into edges.

We then upload the topology definition on the interplanetary filesystem, a decentralised file repository that is used in decentralised applications, since saving whole files on the blockchain is both difficult but also computationally expensive to the point that it becomes impractical. Files on IPFS are connected with a cryptographic hash generated by the SHA256 algorithm. This hash is a unique representation each file stored on the IPFS storage. By uploading the dynamo

definition on IPFS it is assigned thus a unique hash, based on the nature of the file. We then upload the topology definition on the interplanetary filesystem, a decentralised file repository that is used in decentralised applications, since saving whole files on the blockchain is both difficult but also computationally expensive to the point that it becomes impractical. Files on IPFS are connected with a cryptographic hash generated by the SHA256 algorithm. This hash is a unique representation each file stored on the IPFS storage. On the Ethereum Ropsten Test network we have deployed a Token Minting contract. The tokens generated have beyond their unique number, the following variables: a hash in the form of a string, a price and an address that owns the token. [smart contract code]

```

uint256 public _tokenIds;
uint256 public _buildingItemIds;
mapping (uint256 => buildingItem) private _buildingItems;

//@dev: creating the buildingItem as a struct, tokenURI is the IPFS URI
struct buildingItem {
    address seller;
    uint256 price;
    string tokenURI;
    bool exists;
}

```

To connect the Dynamo software to the Ethereum blockchain we use an intermediary library written in Python, Web3.py and a script that essentially writes the hash to the smart contract. We confirm through the remix.ethereum.org interface that the IPFS hash of the topologic dynamo definition has been minted into a token in our contract. [Figure 4]

```

contract_address = "0x4C7c5343A5814c915c8059547119DA4470E291E1";
wallet_private_key = "PATH_TO_PRIVATE_KEY_FILE";
wallet_address = "XXXXXXXXXXXXXX";
infura_url = "https://ropsten.infura.io/XXXXXXXXXXXXXX";
path = "C:\Users\XXXXXX\anaconda3\envs\dynamo383\lib\site-packages";
IPFSString = "QmYdL2XjEG1QMKqAxp4SGAjVGyVlxrVjfipBhsSnTYafa";
number = 98765;

```
stateMutability": "view", "type": "function", ("inputs": {}), "name": "myUint", "outputs": [{"internalType": "uint256", "name": "", "type": "uint256"}], "stateMutability": "view", "type": "function"}, {"inputs": {}, "name": "owner", "outputs": [{"internalType": "address", "name": "", "type": "address"}], "stateMutability": "view", "type": "function"}]

smartContract = w3.eth.contract(address=contract_address, abi=contract_abi)

receipts = []
nonce = w3.eth.getTransactionCount(wallet_address)
tx_dict = smartContract.functions.setIPFSHash(message).buildTransaction({
 'chainId': 3,
 'gas': 210000,
 'gasPrice': w3.toWei('50', 'gwei'),
 'nonce': nonce,
})
signed_tx = w3.eth.account.signTransaction(tx_dict, wallet_private_key)
tx_hash = w3.eth.sendRawTransaction(signed_tx.rawTransaction)
tx_receipt = w3.eth.waitForTransactionReceipt(tx_hash, timeout=120, poll_latency=0.1)
receipts.append(tx_receipt)
```

```

Figure 4. Python Script on Dynamo, writing an IPFS hash on our ERC721 smart contract.

Thus, we are able to encapsulate a topological representation of a building unto a unique non-fungible ERC721 token. This allows a bidirectional connection between topological representation of buildings on dynamo and smart contracts on the blockchain.

4. Computational Design Scenarios in Architecture using NFTs -opportunities and constraints

The NFT representation of building components and their topological interdependence using Topologic opens a variety of avenues for architectural design. On a new building “A”, the design of the building will be already represented on the blockchain, along with the topologic interdependence of the components. This creates immediately the possibility for collective ownership of the building, by multiple addresses on the Ethereum blockchain. It also creates the first step for the creation of a circular economy of components, as a future designer that requires to re-cycle the building will have at her disposal, the full, immutable record of all components, and their topological interrelationship. Furthermore, if our technique is extended to include existing buildings, the token representation can be used to use buildings as material banks in a circular economy, with the blockchain providing the data provenance layer, i.e the designer can query the blockchain for components needed for her design. In terms of operations the token representation on the blockchain can be used both as a digital twin but also as an asset that funds decentralised finance applications. While our prototype currently is basic and simple we are working on creating templates and scripts for designers that will allow the tokenisation of their building designs in an easy manner. In parallel we believe that the creation of smart contract NFTs for architecture will contribute to jumpstarting the creation of a creative economy for architecture with blockchain(s) as infrastructure, where designers will be able to have better control of their intellectual property, but also where new methods of procurement will foster innovation in design. For this to happen, we would need better tools for connecting designs to the blockchain, the establishment of certain blockchains as the optimum infrastructures for design, and adoption of our tools by a number of architects.

5. Discussion & Future Possibilities

The work we present conceptually and practically connects non-fungible digital artefacts on a blockchain with the topological representation of physical buildings, where the non-manifold topology of the building is key to the generation of the non-fungible token. Within the Blockchain and the smart contracts universe, this NFT is unique, and an address account can own it. This is the equivalent of legal ownership of an artefact. Through a careful orchestration of the equivalence the smart contract to legal contracts, the digital representations that architects use achieve a unique function: the representation is the building and the building is the representation. This analogy opens a wide variety of applications including value management and the creation of a digital economy, where optimisation of structure, embodied carbon, energy, material and real estate management have direct currency on blockchain and decentralised ledger technologies. For example,

ownership of a NFT could be used as legal proof of ownership of a building, or within the architectural design, of a particular design solution. One can imagine global smart contracts that act as registries of ownership of buildings, where complex transactions are simplified by simple transfer of ERC721 tokens between accounts on the Ethereum blockchain.

Within our prototype we only have tokenised a topology file, however each building component of a topology dynamo definition can be an ERC721 token. This greatly reduces the computational burden and complexity of establishing material passports for their use in circular economy applications of the built environment, and simultaneously provides an elegant and immutable manner of transferring and tracking ownership of building components. One easy extensibility of the system is the addition of various other variables that define the material properties of a component uniquely identified on the topology of the building. This directly solves the problem of the provenance of information in circular economies (Debacker et al. 2017) by directly storing information about a building component, creating trust within the value network through the blockchain, by providing transparent and traceable information, and augmenting potential further applications where the value of a component can further be monetised by financial applications on the Ethereum blockchain. A further potential application of our bi-directional prototype is the provenance of information on digital twins. One can construct digital twins directly on the blockchain securing the operational maintenance of building components and the informational provenance of such operation and maintenance. Thus our cryptographically secured mechanism allows the generation of a representation of a building component, and the management of its lifecycle from birth to end of life, including all phase of the lifecycle of a building, from concept to decommission. We believe that our prototype is significant in the sense that provides a two-way, unique, safe, accurate manner of translating between physical and digital environments, where information is validated, and additionally creates the infrastructure to allow a decentralised management of the built environment. It is original in advancing the state of the art of blockchain in construction, providing for the first time a manner to align physical assets with digital assets in a manner which is unique and singular. The methods and prototypes are developed rigorously, stemming from already well-established practices, within a sociotechnical framework that responds to challenges of the fourth industrial revolution.

References

- “Platform for Smart contract development and validation” : 2020. Available from <www.remix.ethereum.org> (accessed December 2020).
- “Smart Contract Address on the Ethereum Ropsten TestNetwork.” : 2020. Available from <[0x4C7c5343A5814c915c8b59547119DA4470E291E1](https://etherscan.io/address/0x4C7c5343A5814c915c8b59547119DA4470E291E1)> (accessed created and accessed December 2020).
- “OpenZeppelin (2020), ERC721 contract template implementation” : 2020. Available from <<https://docs.openzeppelin.com/contracts/2.x/erc721>> (accessed December 2020).
- Adams, T.K., Thorpe, T., Osmani, M. and Thornback, J.C.P.: 2017, Circular Economy in construction: Current awareness, challenges, enablers, *Proceedings for the institution of*

- civil engineers, waste and resource management*, vol. 170, 15-24.
- Aish, R., Jabi, W., Lannon, S., Wardhana, N. and Chatzivasileiadi, A.: 2018, Topologic: tools to explore architectural topology, *Advances in Architectural Geometry*, Chalmers University of Technology, Gothenburg, Sweden, 22-25.
- Antonopoulos, A. and Wood, G.: 2018, *Mastering Ethereum*, O Reilly Media Inc, Sebastopol .
- Cooper, H.: 2018, Blockchain Feasibility and opportunity assessment, *BRE report prepared for the BRE trust*, London.
- Debacker, W., Manshoven, S., Peters, M., Ribeiro, A. and Weerd, Y.: 2017, Circular economy and design for change within the built environment: preparing the transition, *International HISER Conference on Advances in Recycling and Management of Construction and Demolition Waste*, Delft, The Netherlands.
- Dounas, T.: 2020, "Contract Deployment on Etherscan". Available from <<https://ropsten.etherscan.io/tx/0x6b9ffa539b59314b425f444f6933c47dc4be98f6924b52ffc62aff9b4b0f6efc>> (accessed December 2020).
- Dounas, T., Lombardi, D. and Jabi, W.: 2020, Framework for decentralised architectural design BIM and Blockchain integration, *Journal of Architectural Computing, Architecture in the Age of Fourth Industrial revolution* vol. II(<https://doi.org/10.1177%2F1478077120963376>).
- Entriken, W., Shirley, D., Evans, J. and Sachs, N.: 2018, ERC-721 Non-Fungible Token Standard, *Ethereum Improvement Proposals*, accessed Dec 2020: <https://github.com/ethereum/EIPs/blob/master/EIPS/eip-721.md>.
- Jabi, W. and Chatzivasileiadi, A.: 2020, Topologic: exploring spatial reasoning through geometry, topology, and semantics., *Presented at: 5th International Symposium Formal Methods in Architecture (5FMA)*, Lisbon, Portugal.
- Kinnaird, C., Geipel, M. and Bew, M.: 2017, Blockchain Technology, How the Inventions Behind Bitcoin are Enabling a Network of Trust for the Built Environment, *Arup Report*, London.
- Vogelsteller, F. and Butterin, V.: 2015, ERC20: A standard Interface for Tokens., *EIP: Ethereum Improvement Proposals*, Accessed Dec 2020: <https://github.com/ethereum/EIPs/blob/master/EIPS/eip-20.md>.