

# NFT Chain

hr@nftchain.com

October 2022

## Abstract

Non Fungible Tokens (NFT) are commonly referred to as unique digital assets whose “certificates of ownership” are stored on a blockchain. While the first usage of the word NFT dates back to 2014, the term became popular among the crypto crowd in the 2017 thanks to ERC-721 and reached the masses in early 2021 with popular collections like Cryptopunks, Crypto Kitties and Beeple’s digital art pieces. While we like digital art and collectibles, we think NFTs should evolve into more sophisticated Digital Objects that can be composed into more complex objects, decomposed into smaller pieces, implement interfaces, be borrowed/loaned, be transferred (sold/rented/gifted) or be non transferable, be linked to physical objects and a lot more.

NFT Chain is a blockchain that allows to define next generation NFTs, that could span from digital certificates to video game digital assets, from community utility tokens to bingo cards, from digital art to coding proficiency badges.

NFT Chain is also committed to overcome the structural limitations that NFTs have today. For example NFTs will be fully backuppable, data could be stored on chain, issuer verification/certification, contractual agreements (such royalties sharing policies) could be defined on chain and much more.

## 1 Introduction

### 1.1 About the white paper

This is a living document. We will update it periodically with our thinking and project progress.

### 1.2 Preamble: Why do NFTs exist? What’s the point?

Bitcoin was successful because it solved a problem. Transactions could be made quickly, cheaply, and without the need for a third party: no need to trust anyone. This allowed people to conduct transactions without having to worry about censorship or the possibility of their information being stolen.

Being an open permissionless protocol, bitcoin allowed for a rich ecosystem to develop quickly, wallets, exchanges, payments, etc. Following Bitcoin, Ethereum introduced Smart Contracts into the picture. Additional innovations led to the invention of DAOs, ICOs, Defi, and NFTs to name the most successful. All shared the same properties of Bitcoin: Permissionless, censorship-resistant, open, and pseudo-anonymous.

NFTs came into existence to benefit from all the properties of Blockchain with the additional property of non-fungibility. The initial successful projects involved digital art, collectibles, video games, virtual land, and ownership rights to name a few. The main reason people started using NTFs were:

- **Status symbol and Identity expression.** People, or better, some people, like to display their style and wealth through visual things such as watches, handbags, clothing, cars, etc. NFTs allowed for the first time people to boast their status online by applying avatars worth as much as Ferraris. Additionally, by committing to an NFT project collectors would signal their commitment to a specific project, style, value, or whatever the project meant to them.
- **Collection.** People collect things for a variety of reasons. Some people collect things because they appreciate the beauty or rarity of the objects, while others collect things as a hobby or to display their wealth (see point above). Whatever the reason, there is no question that people have been collecting things for centuries and have assigned value to things that go beyond their utility value.
- **Financial incentive/Speculation.** Numerous people invested in NFT because they anticipated its widespread adoption. They believed that the token would be used by a lot of people in the near future, and thus bought it at a low price with the hopes of selling it at a much higher price in the future.

### 1.3 What is NFT Chain?

NFTs are cool but they are nowhere near being perfect. We think we can improve them quite a bit by introducing some cool features. We are building a custom blockchain for NFTs only. You can own NFTs, NFTs can own NFTs, and you can build your own construction of NFTs, a bit like LEGOs. You can borrow your NFT or rent one. Scammers can't do much because the transfer functions are safe and standard. Transactions fees are low and NFTs are easy to create.

NFT Chain aims at building a custom blockchain for solving most of the existing NFTs limitations and introducing new concepts to give NFTs superpowers. Here is a list of the superpowers we aim to build:

- **Composability:** By being able to easily combine different tokens together, you can create more complex and interesting ecosystems. This also allows for a greater degree of flexibility and creativity in how you use your tokens. Examples of composable NFTs could be plots of land-owning buildings or avatars owning accessories such as glasses or earrings. NFT chain implements this property natively.
- **Security by design:** ERC-721-based NFTs allow for great ". Being smart contracts anything can be added, and malicious actors started to introduce functions to steal NFT and other assets. One of the most common attacks consists in sending to the victim an unwanted NFT. Once the victim tries to sell it or send it away, a hidden function in the contract takes the unwanted NFT and the other valuable assets to a predefined wallet controlled by the attacker. NFT Chain allows for a subset of functions to be performed, limiting the flexibility but also drastically increasing the security.
- **Defi Ready:** Successful NFTs will be traded and exchanged. At the moment the most used marketplaces are centralized. Artists and users rely on the centralized marketplace for performing correct auctioning and royalties payouts. This could be programmed in the NFTs natively so that royalties could be paid automatically to the artists, or NFTs could be traded and exchanged without needing intermediaries. Decentralized exchanges for NFTs. Valuable NFTs could be also utilized as collaterals for loans.
- **Rentability:** Imagine I own an impressive collection of weapons in a metaverse game; I can use only one at a time; what if I could rent my NFTs in exchange for tokens?

- **Scalability:** Ethereum block space is expensive! Building on substrate allows us to add multiple parachains as we need and develop custom Zero Knowledge scalability solutions akin to Rollups in Ethereum.
- **Multichain:** We aim at building the best-in-class NFT blockchain. But we believe the future is multichain. With wrappers and bridges, collectors will be able to transfer their ERC-721 to NFT-chain back and forth, allowing them to access multiple ecosystems.
- **Easy to use, explain it to me like I'm 5:** Creating an NFT on Open sea is quite easy I have to admit, however, those NFTs are rarely worth much. We want to build a set of tools to enable builders to create more complex projects. Templates, Tutorials, UIs, etc.

## 2 Technical Details

### 2.1 Blockchain

NFT Chain is being developed on the Polkadot/Substrate/Rust stack, aiming to become a Parachain on Polkadot.

Why a blockchain? Because we think digital objects must be owned by people and not by companies.

Why Polkadot/Substrate? Because we wanted to take advantage of an existing ecosystem that doesn't impose many limits. After a few months of deep dives we decided to join the DOT Family. Other alternatives considered: CosmosSDK, Ethereum Clone, Solana.

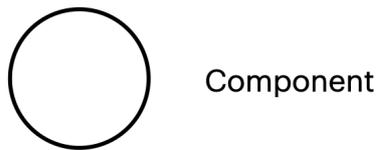
Why a parachain and not a layer 1 chain? Because we want to exploit shared security and the existing Polkadot ecosystem. NFT Chain will still be a very infrastructural project, with other projects and blockchain layers built on top. Being a Dot parachain won't be a limitation even in case of a failure of the Dot ecosystem.

How do we plan to win a parachain? The founders are also large Dot holders and supporters of the project. Willing a parachain auction should be relatively easy even without a crowdloan.

### 2.2 Composable NFTs

In this section we're introducing the main NFT Composability concepts we built NFT Chain around.

### 2.2.1 Component

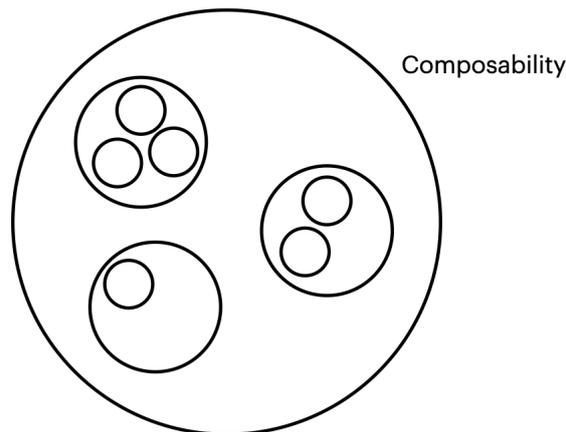


A basic unit. Most of the actual data storage from our users will be done by creating and modifying components. The highest abstraction level for a component is: it is a dictionary with string keys. Going a bit more into details, the dictionary is split in two sections:

- A section which contains always the same keys, in a predefined format, and is required to be filled at creation time. This section is at present day called the Info of a component, is not a dictionary, but a static struct, and contains
  - uid - An immutable component id, which is different from the id of any other component
  - creator - An immutable account id, which corresponds to the account who created the component
  - template - A template id option, which, if Some(\_), corresponds to the template whose requirements the component has to satisfy at any given time. The None variant means that the component is free to do whatever it wants.
  - owner - A mutable account id
  - max\_borrows - A mutable non-negative integer.
  - max\_incoming - A mutable non-negative integer
  - max\_outgoing - A mutable non-negative integer
  - editors - A permission structure.
- A section which contains optional keys, which are in principle all mutable, can be created and destroyed at any time, and can in principle have any value. This section is called the Properties of a component.

Using templates the creator of the component can restrict the mutability of Info and Properties of a component, and can also restrict the keys and values which can be encoded in the properties of a component.

### 2.2.2 Composability

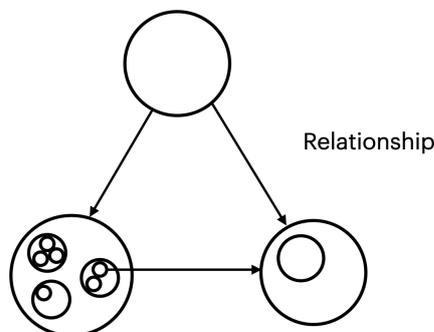


A mutable hierarchy on the components. Ideally it allows users to say that a multi-part object is made of its components (meaning overload with the previously defined Component is intentional). Currently, the plan is to implement composability by associating, to each component id X, a set of component ids S(X), and ensuring that, at every moment,

- the graph ( C : C is a component id, (X, C): X, C are component ids, and S(X) contains C ) is a forest.
- for every component X, and every C in S(X), the owner of C is the same as the owner of X

The main use case (so far) for this structure is that it allows users to instruct the blockchain to perform the same action for many components at the same time, for example it allows to change the ownership of a component and all the components in its associated set with a single extrinsic call from the user.

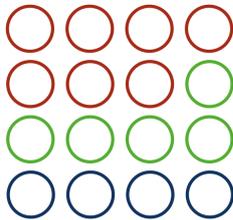
### 2.2.3 Relationships



An encoding of the interaction between components. It can be achieved through two entities: edges and (still work in progress) references.

- Edges are at the moment conceived as a (ComponentId, String, ComponentId) tuple, and an edge (A, B, C) means that component A refers to component C with the name B, and C is aware that A is referring to it.
- References (WIP) are at the moment regular properties of a component, with the type of their value being set to ComponentId. If a component A has a (B, C) reference in its properties, it means that component A refers to component C with the name B. The main difference with edges is that C is not necessarily aware of the existence of A.

## 2.2.4 Templates



Templates

Sets of rules which limit the states that some components can exist in. In essence templates are blueprints for objects.

Static rules are divided in the 4 parts that form a component:

- Info: For each info you can specify a constraint but also a default value.
- Properties: For each property you can specify a different rule composed by:
  - Name: This value will be the identifier of the property we are defining inside the component. Could also be "\*", this will match any type of name different from the other specified properties
  - DataType: Required type and constraint for that property
  - Default: Value used for the property if not found in the component.
  - Required: Boolean value, represents whether or not a specific property is required to be present. Every property specified in this field will be required in the component
- Outgoing edges: Like for each property also for each outgoing there will be an entry that describe all the possible constraints:
  - Name: Required name for the edge that create the connection with the outgoing

component. Could be \*, that will match all the outgoing edges names not already specified in the other entries.

- Template Id: Required Template Id that must be respected by the outgoing connected with the previous name.
- Required: Boolean value, represents whether or not a specific edge is required to be present.
- Incoming Edges: Here we accept a set of TemplateId and at least one of them must be respected by the incoming components.

Inside Info and Properties we are able to specify the requested Type and the possible Constraint through the DataType, this string requires format that allows identification of the requested type and also a possible constraint on it.

The accepted DataTypes are: 64 bit integer value ('Int'), Strings ('Str') and every other type ('\*').

Those data for now have to follow the evalexpr syntax but in the next alpha will be updated with SCALE.

The constraints instead follow the type and must be inside parentheses:

- ValueInt: the integer value must be equal to the one specified in this constraint. Syntax: '(10)'
- ValueStr: same as ValueInt but with a String as argument Syntax: '("ciao")'
- Range: The value should be inside the specified range, including extremes Syntax: '(0..10)'

Not all Constraint is accepted by any DataType. Int works with: ValueInt and Range where the behaviour is obvious, different for String that accept every Constraint but ValueInt and Range interact with the length of the specified string. The last DataType is \* but it does not support any Constraint.

Dynamic rules instead, for now, is an unique expression with a boolean result, this expression will be evaluated following a precise syntax and in the component context.

In the context for now we can access only to the current component's info and properties but with some updates we will manage also the same access to the outgoings through the edge name and also to incomings.

Example:

```
self.max_incoming < 10 &&
self.props.color == \red"
```

Explanation:

- self: stands for the current component
- max\_incoming / props: name of the component info
- color: the special third keyword could be used only if you are using the props info, this will be used as key in the properties map

The '.' character acts as a divider with respect to keywords, this implies that info and property names will not be able to contain dots in their entirety.

### 3 Tokenomics

#### 3.1 Disclaimer

As of now this section is just a guideline of what's coming. Nothing in this section should be taken as final and parameters might change. As the project progresses things are going to be finalised and more details will be added.

#### 3.2 NFT token

- Ticker: NFT
- Supply: 1,000,000,000
- Inflation: TBD
- Relay chain: Polkadot
- Supply profile: TBD
- Anticipated launch: Q3 2023
- Small unit: 1e10 Plank (1 Plank 1e10 = 1 NFT)
- Transaction fee: x% burned - y% treasury - z% validators fee

#### 3.3 Utility

NFT is the native utility token that powers the NFT chain.

- Transactions fees
- Node incentivisation
- Governance
- Other on-chain block-space consuming services

### 3.4 Inflation

The purpose of inflation in the NFT Chain is to pay for ongoing security needs of the network. The primary security budget items are to pay for a parachain slot on an ongoing basis, and to incentivise collators to provide collation (block production) services to support the NFT chain network. Part of the inflation goes towards incentivising collators, another part goes to the parachain bond reserve to accumulate on chain funds to pay for a parachain slot in perpetuity, the remaining is for users that stake their NFT tokens and help power the collator selection process. Additionally a percentage of the transaction fee could be burned to lessen the dilution of the token holders.

The final inflation rate will be determined by a number of factors;

- Expected Staking rate: X ideal
- Expected annualised rate of return: I ideal
- The inflation rate when the Staking rate is 0: R0
- Attenuation rate d

The formula to calculate the final inflation follows:

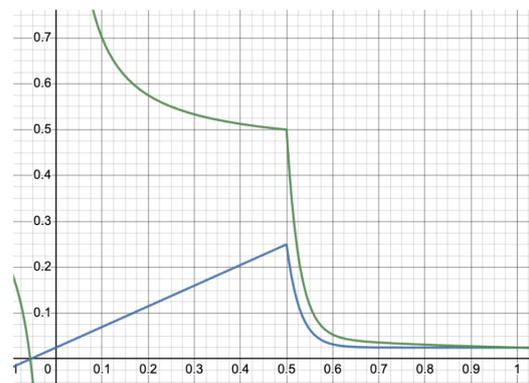
$$I_{NPOS}(x) = \begin{cases} I_0 + (I_{NPOS}(\chi_{ideal}) - I_0) \frac{x}{\chi_{ideal}} & \text{for } 0 < x \leq \chi_{ideal}, \text{ and} \\ I_0 + (I_{NPOS}(\chi_{ideal}) - I_0) \cdot 2^{(\chi_{ideal}-x)/d} & \text{for } \chi_{ideal} < x \leq 1 \end{cases}$$

$$i(x) = I_{NPOS}(x)/x.$$

Those factors are yet to be determined; For example, assigning values of:

- X ideal = 0.5
- I ideal = 0.05
- R0 = 0.025
- d = 0.02

the resulting curve follows:



The following figure shows the relationship between the simulated inflation rate, Staking rate, and annualised rate of return, the X-axis is the Staking rate, the blue line is the annual inflation rate, and the green line is the annualised rate of return.

### 3.5 Distribution

The token distribution will take into account the team, the auction, the ecosystem development, and some additional funding if needed.

### 3.6 Consensus algorithm

NFT chain uses Polkadot nominated proof of stake (NPoS), an adaptation of proof of stake (PoS) in which an unlimited number of token holders can participate as nominators, backing a large but limited set of validators (expected to be in the order of hundreds at genesis). This scheme allows for a massive amount of stake to back validators, much higher than any single user's holding, thus rendering the network more secure.

Nominators, who share the economic rewards as well as possible slashings with the validators they back, are economically vested in the security of the system and are thus economically incentivized to watch over the validators' performance.

As such, NPoS is not only much more efficient than proof of work, but also considerably more secure and decentralised than PoS schemes without stake delegation, where only a few whales (owners of a large amount of tokens) can ever become validators.

NFT chain aims at becoming Polkadots parachain and potentially in the future running its own validator network becoming independent.

### 3.7 Parachain Auction Token Details

NFT will be rewarded to participants who support the Parachain Auction with DOT. Upon receipt, a portion of this NFT will be unlocked and transferable, while the remaining portion will be vested and released gradually over the parachain slot duration. Vesting schedule and reward remains to be determined.

## 4 Roadmap

We are currently working following a roadmap that dictated milestones for the next 4 quarters. The initial roadmap was drafted during the first in person full team gathering in Greece, and the milestones names reflect that. We choose greek gods names in

alphabetical order. This roadmap is only a guideline, things might change depending on too many factors.

### 4.1 Q4 2022 Athena

Athena goddess of heroic endeavour! For our first milestone we aim at completing all basic functionalities on chain explained in the chapters above; specifically the borrow pallet, the composable pallet (including templates and composability), and benchmarking. Additionally the white paper will be refined with a first proposal for the tokenomics and on how the network will be secured and operated. The website and the basic social media accounts will be operational. All of this with the blessing of Athena.

### 4.2 Q1 2023 Bia

In Greek mythology Bia is the personification of force. For our second milestone we aim at polishing all aspects of our blockchain, and at introducing utilities such as explorer, documentation, tutorials etc... To stress test our infrastructure we will release the first version of NNS, our flagship project described above. By then the full team will be at full force with the help of Bia.

### 4.3 Q2 2023 Chaos

The personification of nothingness from which all of existence sprang. V1 will be completed, with the launch of the public testnet and NNS V1. What better god to guide us than Chaos?

### 4.4 Q3 2023 Dionysus

Also known as Bacchus, God of wine, fruitfulness, parties, festivals, madness, chaos, drunkenness, vegetation, ecstasy, and the theatre. By Q3 2023 we aim at launching the final version of NFT Chain, and kicking off the growth initiative to bootstrap the ecosystem. At the end of a full year of hard work the community is ready to celebrate under the supervision of Dionysus.

## 5 Project Ideas on top of NFT Chain

Here we want to list a few ideas that could be built on top of NFT Chain infrastructure.

### 5.1 NNS (NFT Name server)

NFT projects are becoming more and more successful, with the Ethereum name server being one

of the most successful. This project maps wallet addresses to human-readable names, making it much easier for users to interact with them. NFT chain names can be purchased for 1 or more years for a fixed price. The owner can renew the ownership when the expiration date approaches. Owners can list their NFT names for sale and ownership can be passed on (like a normal NFT). Wallets can now utilize the human-readable name instead of the address.

NFT names can represent the identities of individuals, companies, associations, DAOs, etc. Example: my name is Giorgio, and I will purchase the name Giorgio.NFT. Now I can assign to it the NFT I want to display, I can add all my certificates and access tokens in one place. More on ENS

## 5.2 Mecenat

Mecenat allows influencers and creators to create a community/network of supporters. A youtuber can create a limited edition NFT to fund his channel. The NFT can give access to exclusive content (like Patreon, OF, etc.) or access to physical events (like VeeFriends), gifts, or anything else.

## 5.3 C.C. (Conspicuous consumption)

C.C. is the first Swiss Luxury NFT brand! Those NFTs don't do much but they are pretty and can be applied as avatars on Twitter. To justify the high cost, the NFT can own some wrapped BITCOIN or ETH. Those cryptocurrencies can be unlocked back only upon the destruction of the NFT. 1 BITCOIN can be represented as an NFT and redeemed at any point in time. A companion Dapp allows for wealth voyeur to quickly look up the price of the C.C. NFT

## 5.4 Coding Badges

Imagine applying for a job and start working with them 30 seconds later. Imagine if a bot on github and other online communities mints proficiency badges on your linked NFT Chain wallet based on the amount - and quality - of code you wrote in a specific programming language. Imagine you're applying for a job that requires Level 5 in Javascript and Level 7 in Python. Instead of relying on self assessment on a resume and expensive, cheatable and error prone coding interviews your future employer can safely assume you know a specific language/framework because the blockchain says so.