# Musicmap Enhancement Protocol

## Table of Components

# Motivation

The goal of this document is to provide a top-level architecture of the musicmap marketplace, and explore certain building blocks.

Musicmap is a marketplace that offers free or commons data between providers and consumers, also described as contributors. It can be seen as a fairly simple marketplace that does not hold physical data, does not require proof of consumption, nor deals with complex and secure smart contracts for deployment. On the other hand, it will have one the most complex structures of on-chain curated data registries (TCR) and a very challenging dynamic UI linked to the registry framework.

Musicmap will share non-physical data (URL, links) primarily between the common (i.e. non-business) public (C2C). Providers (aka Contributors in this case) are anyone who is motivated to bring structure in a chaotic mass of digital music archives, and who wants to contribute to musicmap's goals as described in the original purpose paragraph (more focus on long tail artists, informing the commons about music history and sociology, building a platform to interact with other (decentralized) music services,...). Consumers in this case (aka Users) are basically anyone who likes music.

Contributors can create their own lists based on genre, which will automatically deploy the right smart contracts and - if needed - a new token. Contributors can also build communities around these genre lists and organize or govern towards certain goal (like tribes), e.g. making an agenda for concerts and parties, making a list of new and unknown artists, rallying community meetups, etc.

The primary ambition however, is to establish a very attractive, decentralized, scalable, and dynamic platform that acts as a browser or explorer to discover music in an active (as opposed to passive-algorithmic) manner.

# Architecture Overview

*This chapter explains the various backend components (some of which have a curation/governance aspect which will be further addressed in a separate chapter), and the top-level architecture of the network.*

## Introduction

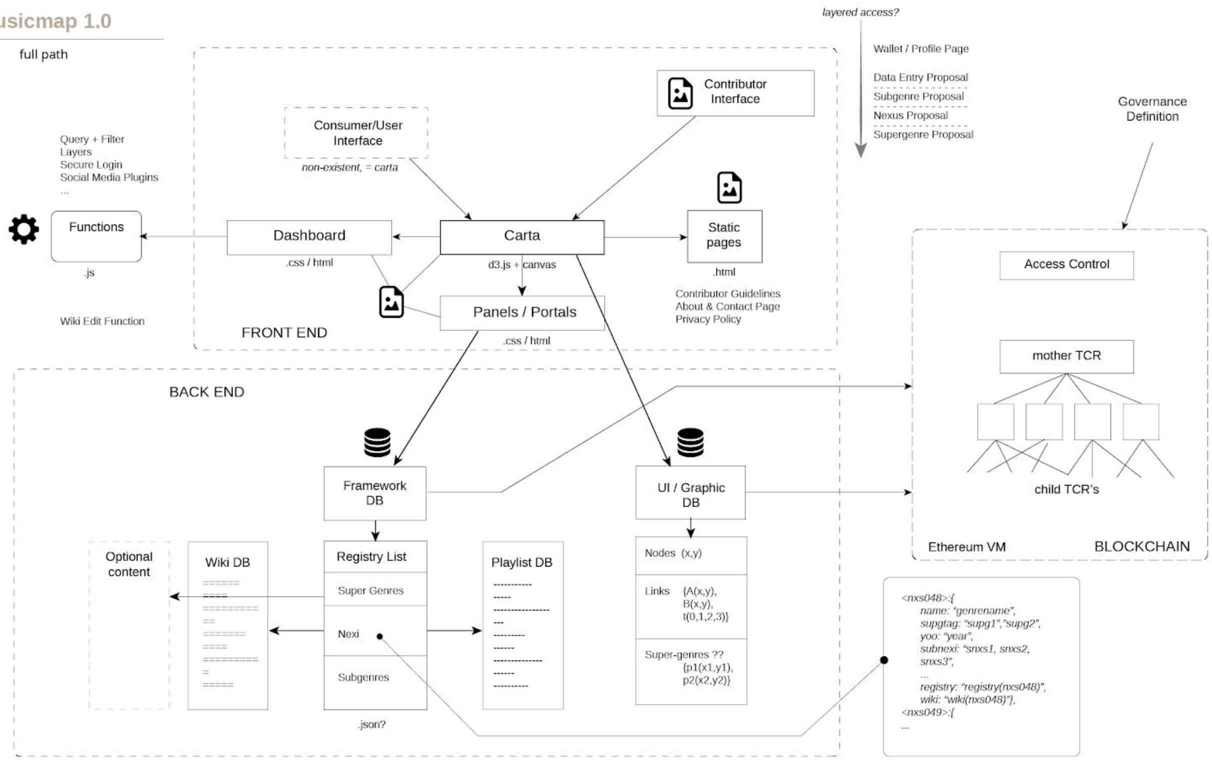
Because the success of the network depends on easily finding new music in an overwhelming ocean of datafiles, the hierarchical structure will be complex but also mandatory to tackle this problem and provide good UX. The proposed structure is an interlinked graph tree of dependencies that works as a nested TCR[1] or a knowledge graph[2]. Further insight needs to be gained which approaches will be used to define the dependencies (e.g. links curated in separate TCR, parent tags, knowledge graph API,...).

[1] https://medium.com/coinmonks/token-curated-registry-tcr-design-patterns-4de6d18efa15

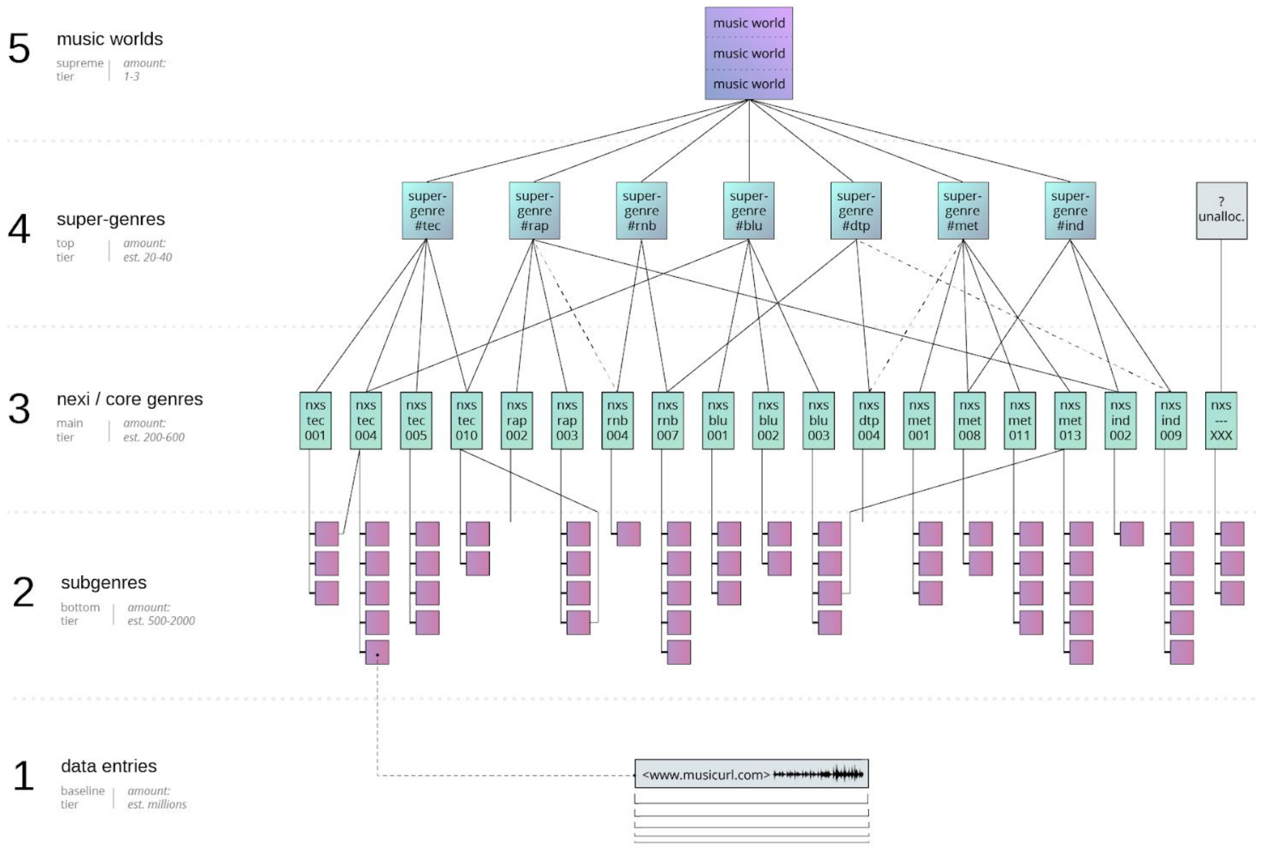
[2] https://grakn.ai/

## musicmap 1.0

full path



```
<nxs048>:{
     name: "genrename",
     supgtag: "supg1","supg2",
     yoo: "year",
     subnexi: "snxs1, snxs2,
     snxs3",
     ...
     registry: "registry(nxs048)",
     wiki: "wiki(nxs048)"},
<nxs049>:{
...
```



*Complete overview of the top-level architecture.*

## Tier 1: The Music Universe = Marketplace

*Supreme Level / Music Planets or Worlds*

The highest tier will act as a mother registry whose attributes will be transferred to all child registries. It is OPTIONAL though NOT RECOMMENDED (more friction) that proposed Data Entries must first be accepted in the mother registry before being eligible for allowance in a child registry (layered TCR).

In order to have a truly decentralized framework, it is mandatory that even the highest level components (super-genres, see further) are curated and subject to change by vote of community. Curation of such a top-level list of super-genres CAN be done in one category (TBD) or possibly in multiple (static and non-curated) categories, which are called Music Worlds or Music Planets henceforth. The recommended number of such Music Planets is three: Classical & Utility Music, World & Folk Music, and Popular Music. These worlds are proposed from a UI perspective (to tackle the problem of an overcomplicated UI map).

## Tier 2: Super-Genre = Mini Marketplace

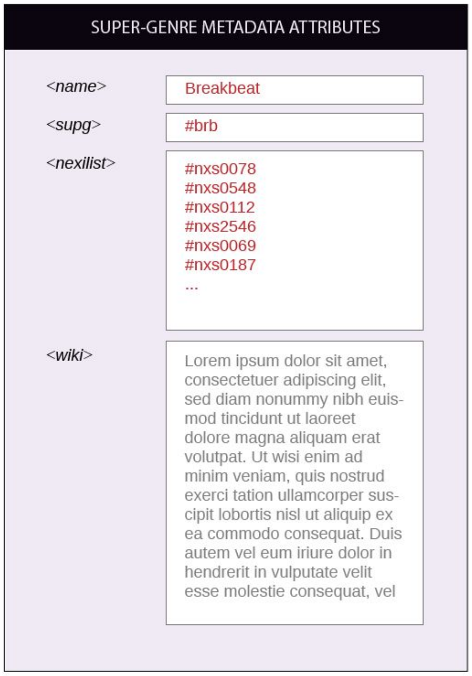
*Nexus Group / Top level framework / Curation Market / Big Community*

A Super-Genre is a list of different genres (“Nexi”, see further) that belong to the same group, thus it is a genre group or top-level hierarchy style to allow easier exploration and to structure the high number of genres. It is important to introduce such higher level of hierarchy in order to facilitate the browsing experience and to further enrichen the possibilities.

At its core, a Super-Genre is a curated list of different Nexi. These lists SHOULD be curated, but they SHOULD NOT be curated according to the same rules as a Nexus. It is expected that the curation aspect is much more minimal and less open to debate.

A Super-Genre SHOULD have a linked content information pane or page (“Wiki”), to allow contributors to make the right choice in adding Nexi.

A Super-Genre MUST NOT have any Data Entries.



SUPER-GENRE METADATA ATTRIBUTES

| | |
|---|---|
| <name> | Breakbeat |
| <supg> | #brb |
| <nexilist> | #nxs0078<br>#nxs0548<br>#nxs0112<br>#nxs2546<br>#nxs0069<br>#nxs0187<br>... |
| <wiki> | Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel |

A Super-Genre CAN have the optional feature of owning a separate, generic Nexus that holds Data Entries that are impossible to attribute to an existing Nexus, i.e. these songs are too eclectic to put into a genre, but can belong to a super-genre (“non-allocated”). This can allow users to browse for interesting music.

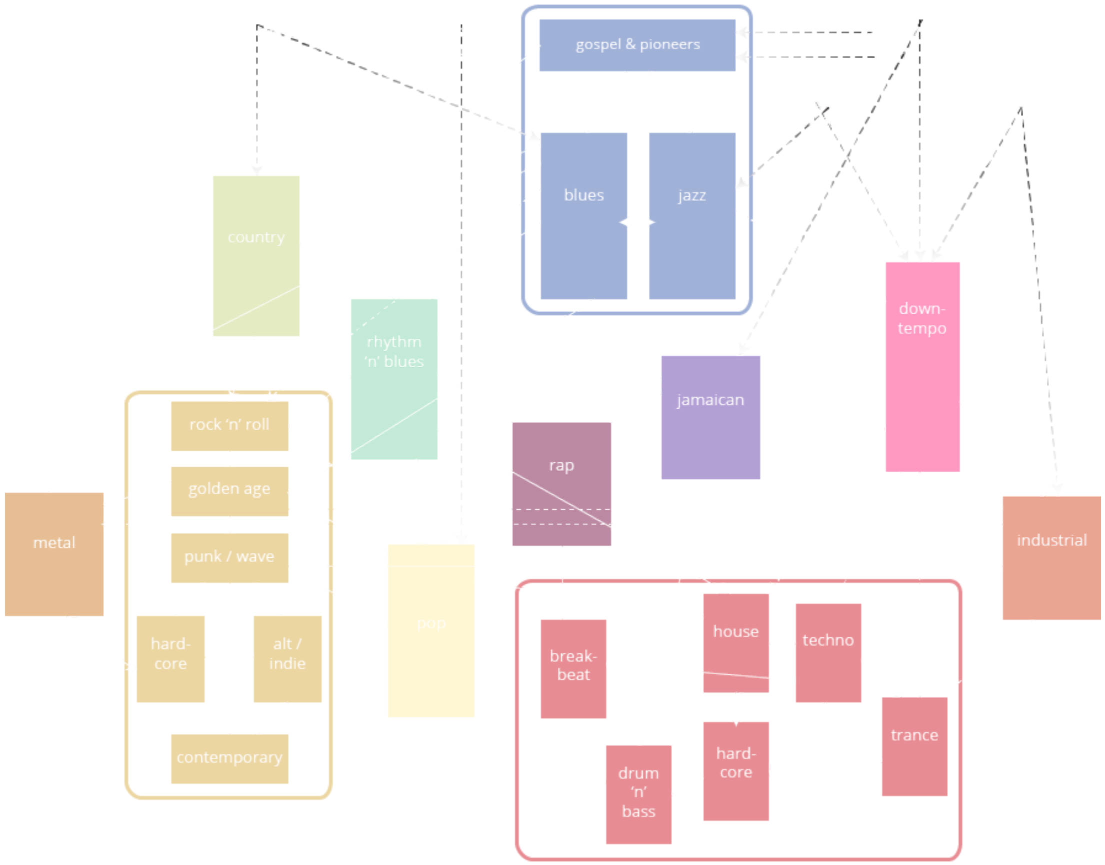
A Nexus can belong to multiple, different Super-Genres.

Because curation of Super-Genres happens on a higher level than curation of Nexi, the curation rules SHOULD be more strict (e.g. 75% majority, different set of curators).

From the musicmap.info methodology page:

*More than 230 music genres is still too abundant to create a comprehensible structure that allows easy orientation. The need for a covering framework is an issue that will be addressed in this chapter. Certain (though few) visual genealogies choose not to implement such framework, and do not (or vaguely) display clusters of related music genres. When dealing with rather detailed genealogies such as musicmap, omitting a visual framework would seriously harm any practical use the map might have.  Fortunately, this framework already exists as almost all genres belong to greater, well-known "areas" in the musical network, what we will call super-genres. Super-genres are simply the parent genre of any given genre; a higher-level, overarching family.*

*A number of super-genres are so large that they might be divided into smaller groups, roughly the size of other super-genres. We will call these the primary genre clusters, represented by the primary colors: Blue Note (blue), Rock (yellow), and Electronic Dance Music (EDM) or Dance (red). For Blue Note (Blues, Jazz, & Gospel/Pioneers) the clustering is debatable, as the difference between its super-genres is much larger than within Rock and Dance. However, there are plenty of common characteristics to unite these super-genres as one family: period and country of origin, the use of "blue guitar notes", instrument set-up, and a strong mutual influence. Rock music already forms a strong family, of which the subdivision instead of the clustering is debatable. Some Rock super-genres are already well-defined (Hardcore, Alternative, Classic or Golden Age Rock). The other Rock super-genres become transparent as the remaining periods of time, either before or after previously mentioned super-genres. For the last genre cluster, EDM/Dance, there is hardly any debate: these super-genres both form a strongly connected family while they are also clearly separate entities with their own identity and subculture. Lately, the term EDM has received a new and narrower meaning as mainstream-friendly dance music, which has provoked aversion for the term. But it is recommended to maintain EDM's original definition, which is exactly what it stands for.*



*[Blueprint of the larger underlying currents or genre families in the history of popular music, forming the different super-genres Straight lines resemble more important evolutions than dotted lines. Double arrows implicate strong mutual influences.]*

*The correct determination of the different super-genres is very important, but much less difficult than determining the correct number of genres. The same super-genres return in various lists, summaries, other genealogies, literature, although sometimes with other names or a slightly different combination of genres. The boundaries of these super-genres are not rigid, after all. Just like smaller, regular genres, super-genres should be considered as fluid concentrations in the musical universe, fading over into each other.*

*It is very plausible that UK Garage & UK Bass will rip itself loose as an immortal and distinct super-genre in time, just like Metal did from Rock. It is however too loose momentarily and a little bit too soon to tell. It is still possible that it will fade out with remnants dispersing back into Breakbeat, Drum 'n' Bass or other super-genres. If not, UK Garage should seriously be considered as the 24th super-genre, depending on future evolutions and how its interaction with contemporary Hip-Hop music (Trap, Ghettotech, Wonky) will turn out. And with Future Bass being more eclectic than anything else, it is hard to predict what that future will hold. For now, these genres will be placed between Breakbeat and Drum 'n' Bass, which is where they originated.*

## Tier 3: Nexus = Core Genre

*Core Genre / Data List / TCR / Medium level framework / Strong Community*

At the heart of the platform is a given music genre, which will act as a hub with different functionalities. Because of the interconnectability and the versatile nature of it, we will call this component NEXUS henceforth.

A Nexus serves as a community platform to organize and bring together people around a common goal: to share music and knowledge around a certain genre. Each Nexus can be regarded as a tribe in its own right.

A Nexus MUST have a compiled list of multiple Data Entries (the more the better) which MUST have one and only one thing in common: that they belong to the same genre. In order to maintain high quality lists, these must be curated. In the case of Hybrid Entries, the dominant *<nxs>* tag will prevail OR a threshold is selectable the User.

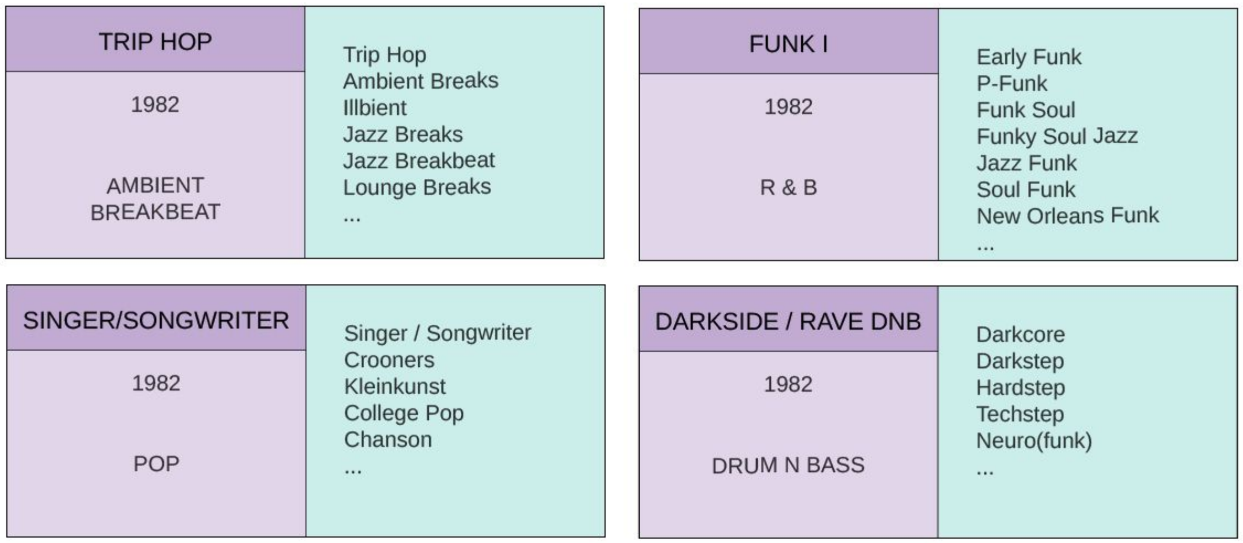
A Nexus MUST have a clear, correct and non-disputable name that allows users to identify the given genre. Contributors SHOULD select the most common name.

A Nexus SHOULD have multiple name tags to further reduce confusion and to avoid unnecessary sprawl of Nexus creation. There are two kind of tags:

*Synonym tags*: name tags that are more or less a synonym for the genre

*Subgenre tags*: any subgenre that belongs to the parent genre (Nexus).

All tags SHOULD be compiled in a curated list of their own. When creating a Nexus, it is recommended that Contributors have the corresponding tags in the back of their head. When Tier 4 is omitted, then subgenre tags and synonym tags can be combined as subtext as in the following graphic example:



TRIP HOP

1982

AMBIENT BREAKBEAT

Trip Hop
Ambient Breaks
Illbient
Jazz Breaks
Jazz Breakbeat
Lounge Breaks
...

FUNK I

1982

R & B

Early Funk
P-Funk
Funk Soul
Funky Soul Jazz
Jazz Funk
Soul Funk
New Orleans Funk
...

SINGER/SONGWRITER

1982

POP

Singer / Songwriter
Crooners
Kleinkunst
College Pop
Chanson
...

DARKSIDE / RAVE DNB

1982

DRUM N BASS

Darkcore
Darkstep
Hardstep
Techstep
Neuro(funk)
...

*Examples of possible Nexi with a visible list of both synonym and subgenre tags within the same container to allow quick identification for a faster browsing UX.*

A Nexus SHOULD have a linked content information pane or page ("Wiki"), to allow contributors to make the right choice in adding Data Entries.

A minimum or maximum amount of Data Entries per Nexus is OPTIONAL (not recommended). Minimum threshold might be desirable before allowing front-end appearance.

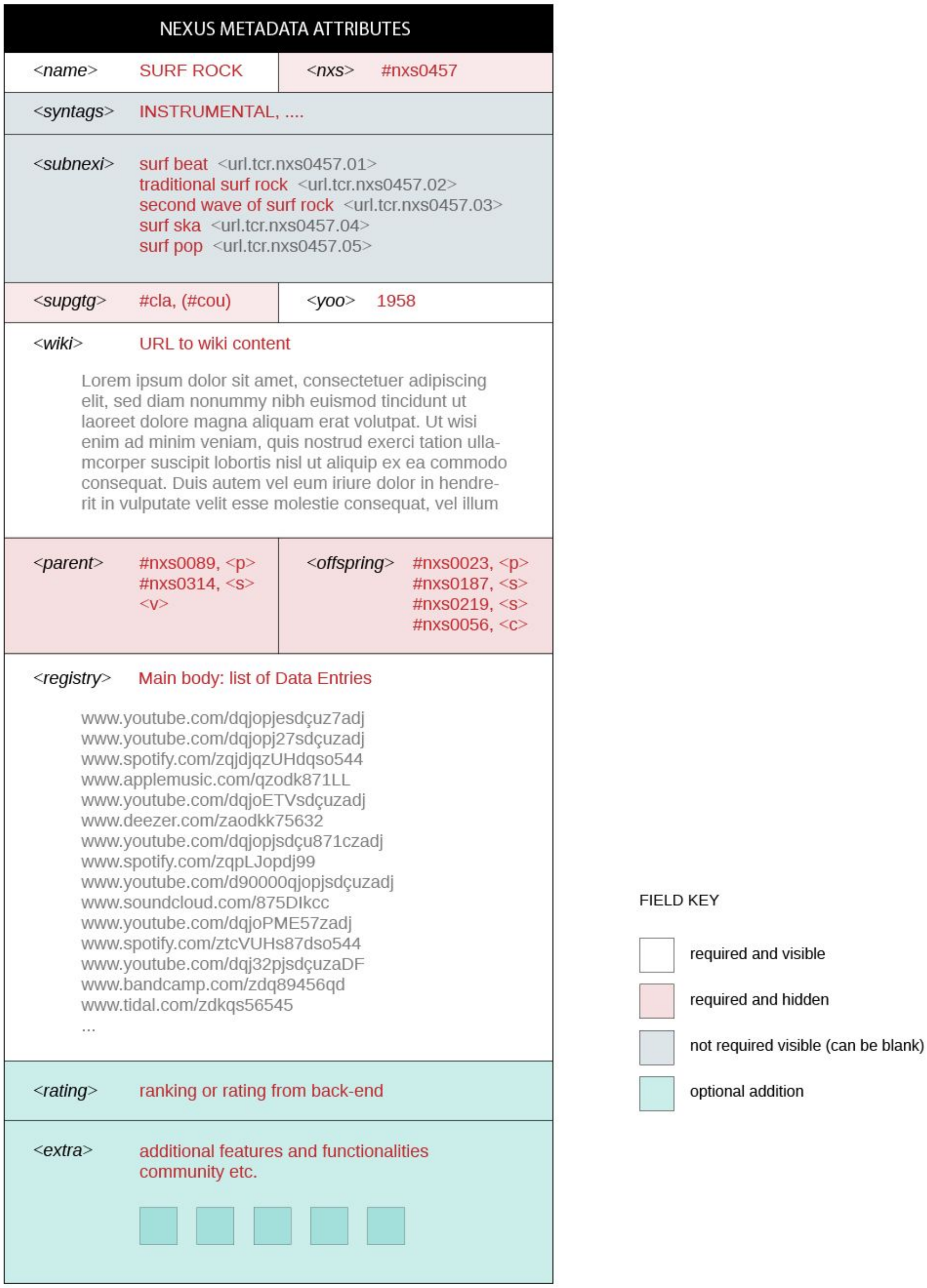
A Nexus CAN have additional future implementations *(forum, heatmaps, festival info, concert info, artwork sharing, list of most popular entries, special search).* This is certainly no part of the MVP.

Each Nexus MUST have its own search function.

When an infinity of lists is allowed, a Nexus MUST have a built-in reputation to allow the better ones to appear on the surface, i.e.: to become selected to appear (first) on the front-end.

A Nexus thus has the following metadata attributes (more CAN be added):

<*name*>: written title name, all caps preferred
<*nxs*>: nexus string: unique codename string; e.g. *tec014*
<*syntags*>: synonym tags that appear as subtitles
<*registry*>: curated list of all data entries, the main datastore body
<*yoo*>: year of origin, curated as part of wiki
<*wiki*>: text information (history, description) about the genre (link to text database)
<*subnexi*>: curated registry of possible subgenres (can be void)
<*supgtag*>: string(s) of the supergenre to which the nexus belongs, major vs minor is optional
<*parent*>: other Nexi from which the genre originated (influences) and corresponding type
<*offspring*>: other Nexi that were influenced by the genre (derivatives) and corresponding type
<*rating*>: community vote and/or popularity proof (by traffic)
<*extra*>: additional features can be added later to let Nexi evolve into powerful tribes



NEXUS METADATA ATTRIBUTES

<name> SURF ROCK <nxs> #nxs0457

<syntags> INSTRUMENTAL, ....

<subnexi> surf beat <url.tcr.nxs0457.01>
traditional surf rock <url.tcr.nxs0457.02>
second wave of surf rock <url.tcr.nxs0457.03>
surf ska <url.tcr.nxs0457.04>
surf pop <url.tcr.nxs0457.05>

<supgtg> #cla, (#cou) <yoo> 1958

<wiki> URL to wiki content

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum

<parent> #nxs0089, <p>
#nxs0314, <s>
<v>

<offspring> #nxs0023, <p>
#nxs0187, <s>
#nxs0219, <s>
#nxs0056, <c>

<registry> Main body: list of Data Entries

www.youtube.com/dqjopjesdçuz7adj
www.youtube.com/dqjopj27sdçuzadj
www.spotify.com/zqjdjqzUHdqso544
www.applemusic.com/qzodk871LL
www.youtube.com/dqjoETVsdçuzadj
www.deezer.com/zaodkk75632
www.youtube.com/dqjopjsdçu871czadj
www.spotify.com/zqpLJopdj99
www.youtube.com/d90000qjopjsdçuzadj
www.soundcloud.com/875DIkcc
www.youtube.com/dqjoPME57zadj
www.spotify.com/ztcVUHs87dso544
www.youtube.com/dqj32pjsdçuzaDF
www.bandcamp.com/zdq89456qd
www.tidal.com/zdkqs56545
...

<rating> ranking or rating from back-end

<extra> additional features and functionalities
community etc.

FIELD KEY

□ required and visible

🟥 required and hidden

▨ not required visible (can be blank)

🟩 optional addition

From the musicmap.info introduction page:

*GENRE NAME*

*The most common name is being used as title with sometimes an alias. Lesser used synonyms are listed in the description, as well as common subgenres. The order of / and & is important, for instance "Ska-Punk / Two-Tone, & Ska-Core", implies that Ska-Punk and Two-Tone are (more or less) synonymous whereas Ska-Core is something else (though closely related).*

*YEAR OF ORIGIN*

*The exact year of origin is very disputable or sometimes even uncertain. Musicmap tries to ignore singular avant-garde pioneers and lists the year that at least two or more different artists release work within the genre. As a general rule of thumb, the listed year of origin comes after the very first experimentations shaping the genre and before the genre becomes known to the general public. When the genre name mentions two distinct genres with proximate but different years of origin, the point in between is taken. When the situation is more complex (proto-phases, timespans, different waves), more accurate information can be found in the description. When you click on the date, the Carta will take you to the genre's location.*

*SUPER-GENRE TAG*

*This mentions the parent super-genre. Certain genres might belong to two different super-genres (hybrid genre). When one super-genre is strongly favored over the other (= master super-genre), the latter will be written in lowercase and between parentheses (= slave super-genre), though it still would be acceptable to regard this one as a parent genre. The order is of no importance. Only master super-genres are taken into account for the genre lists of each super-genre.*

*DESCRIPTION TEXT*

*The description focuses briefly on the historical and sociological context. Furthermore, it tries to list the characteristics of the genre without becoming too technical. The whole aim of the description is to explain each genre in a manner anyone can understand and to pinpoint the necessary nuances such as temporal framework and the influence of following genres in later phases (which cannot be made visually clear on the Carta). Related subgenres are listed when applicable. Keywords are written in bold, as an attempt to reach the very essence of the genre.*

*EXAMPLE SONGS*

*For each genre, nine to twelve prime examples of songs are listed. Whenever the genre name lists two or more distinct genres, the examples will be divided among these genres. These examples are not always the most well-known songs of said artists or the first within the genre, but rather fitting references for the genre characteristics. Although it is perfectly possible to place the same artist in two different genres, this never occurs on musicmap, to allow a greater diversity of artist examples. Songs are deliberately chosen instead of albums or artists, as these rarely can be fitted into one genre category.*

## Tier 4: Subnexus = Datastore Fragmentation

*Subgenre / Fragmentation / Niche-Hardcore community / Low level framework*

Contributors can propose a new sub-Nexus within an existing one to allow the creation of subgenres and build (niche) communities around those, further increasing the level of detail/hierarchy and enrichen the user experience.

A Subnexus MUST NOT exist without a parent Nexus, i.e. the parent Nexus must first be created before a Subnexus can be proposed. A Subnexus can belong to two different Nexi.

A Subnexus MUST have a clear, non-disputable genre name, which is the most used name.

Nexi SHOULD be transferred to SubNexi when necessary (by vote), which basically means they are Demoted. Demotion/Promotion process requires further investigation.

A Subnexus SHOULD NOT have the same functionalities as a Nexus. It is recommended to keep these minimal: meaning it is only a list.

The Subnexus tier (Tier 2) is OPTIONAL, and might be omitted in favor of non-interactive subgenre tags within the Nexus. The danger of omitting this tier however, is that contributors will create new Nexi that are actually a part of an existing Nexus, and thereby seriously endangering the overview and UX.

## Tier 5: Data Entry = Dataset

*Music Song / Data Point / Proposal with Stake*

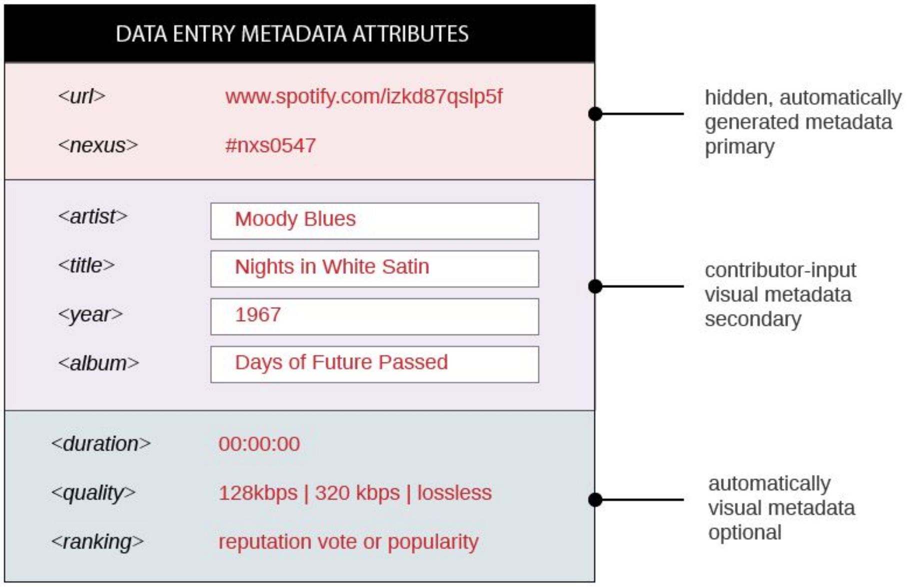
A Data Entry is a single item of data, which in this case will be a link (URL) to a third party music content provider. It is the smallest type of container within the framework, and thus indivisible like an atom.

A Data Entry MUST link to a single work of music, i.e. a song. It MUST NOT link to albums.

A Data Entry MUST NOT link to illegal/pirated work.

A Data Entry MUST NOT link to anything non-audio (festivals, written content, artwork,…)
It is an OPTIONAL feature feature to allow non-audio Data Entries tied to a specific Nexus, which will be compiled in a different list.

Each Data Entry MUST have the following metadata: <url>, <nxs>, <artist>, <title>, <year>, <album>
Most metadata is added by the Contributor and will also be curated. Whenever possible, the secondary visual metadata SHOULD be extracted from the source to reduce work.



DATA ENTRY METADATA ATTRIBUTES

| | |
|---|---|
| <url> | www.spotify.com/izkd87qslp5f |
| <nexus> | #nxs0547 |
| <artist> | Moody Blues |
| <title> | Nights in White Satin |
| <year> | 1967 |
| <album> | Days of Future Passed |
| <duration> | 00:00:00 |
| <quality> | 128kbps \| 320 kbps \| lossless |
| <ranking> | reputation vote or popularity |

A Data Entry is proposed by a Contributor and can be accepted or rejected by the community.

Anyone SHALL be able to propose a Data Entry. Friction to propose Data Entries must be minimal.

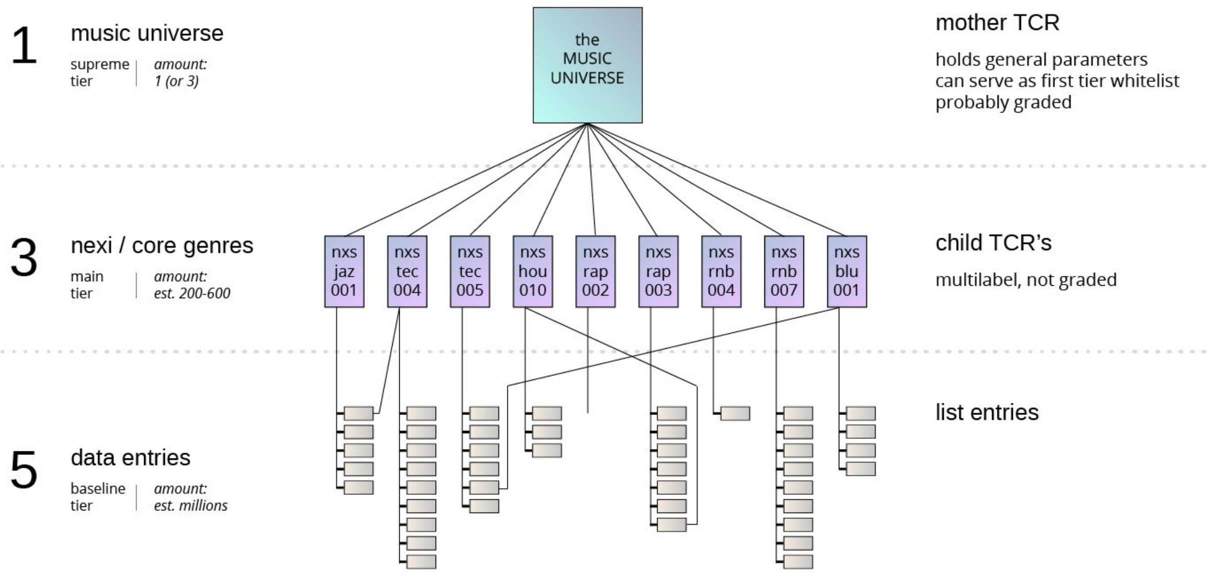
Contributors SHALL NOT propose Data Entries outside a specified Nexus or Subnexus.

To gain accelerated growth, the platform SHOULD allow multiple proposals of Data Entries in one action by a given contributor.

A Data Entry SHOULD be able to belong to multiple, different Nexi. The consequences and implementation of such mechanics require further investigation. This might allow a visual representation of the genre genetics of a certain song or Data Entry (e.g. 72% Boogie, 20% Nu-Disco, 8% Synthpop) and can allow for some very interesting search functions that suppress the possible dangers of pigeonholing. A maximum of 2 or 3 Nexi is highly recommended. If this is allowed, we will call such entry a HYBRID ENTRY.

## Notes on the 5 Tier architecture and Music Groups:

The previous proposals assume a 5 level or 5 tier architecture (or 4 tier depending on whether to include data entries or not). Tier 4 can be omitted in case of overwhelming complexity. The odd tiers (1,3,5) form the absolute minimum working basis, whereas tiers 2 and 4 are mainly for structuring purposes. A minimum simulation network will probably look like this:



From the musicmap.info introduction page:

*Musicmap attempts to provide the ultimate genealogy of popular music genres, including their relations and history. It is the result of more than seven years of research with over 200 listed sources and cross examination of many other visual genealogies. Its aim is to focus on the delicate balance between comprehensibility, accuracy and accessibility. In other words: the ideal genealogy is not only complete and correct, but also easy to understand despite its complexity. This is a utopian balance that can never be achieved but only approached. By choosing the right amount of genres, determining forms of hierarchy and analogy and ordering everything in a logical, but authentic manner, a satisfactory balance can be obtained. Said balance is always the main subject of discussion in music genre genealogies and the capital reason why an absolute visual reference has been absent thus far (and probably always will be). Musicmap is an ongoing project in search for the perfect balance of popular music genres to provide a powerful tool for educational means or a complementary framework in the field of music metadata and automatic taxonomy.*

*The main conceptual methods of musicmap to achieve a satisfactory equilibrium consist of grouping closely related genres together ("sibling genres"), color coding much larger genre groups ("super-genres"), and introducing a deeper layer of lesser influential subgenres. Hereby the total amount of the intermediate or main genres could be reduced to 234. This is deliberately far from the possible total amount of genres (approximately 600 or more, some sources claim over 1000) to enable easy orientation and good overview. […]*

*Musicmap combines the advantages of large mega genealogies (>500 genres) with those of synoptic overviews (<50 genres) by working with different levels of detail on its visual genealogy, referred to as the "Carta". The upper level displays only the super-genres. The lower level forms the principal subject of the aforementioned balance, where the main genres are listed.*

*[Goal 3:] To improve existing music genre databases and to provide a complementary and necessary framework for automatic music taxonomy.*
*Numerous international companies and academic researchers are currently looking for ways to analyze music databases and implement metadata in order to bring structure to these enormous archives. With the ever growing*

*capacity of electronic music delivery (EMD) systems such as ITunes, Spotify, Soundcloud, Shazam, Tidal, Beatport, Google Play, etc., the need increases for an algorithm-based architecture where automatic taxonomy based on genre is often preferred. Researchers try to develop adequate algorithms that analyze any given audio signal, break it down into its various components, and process a singular outcome: a tag in the form of genre. However, these components need to be checked against a legitimate backbone structure: a complementary framework of music genres like musicmap. Many well-known music websites, players, apps, catalogues, and databases already use a predetermined categorization of music genres. Unfortunately, these categorizations are often imbalanced and inaccurate, with very large genres and obscure subgenres nested on the same level, or even mentioning of two genres that are actually the same. The primary contributor of this imbalance is the translation of an organic, dynamic and sociocultural reality (genres) into a structured, static and mathematical artificiality (hierarchical categories). Musicmap believes that a top-down, hierarchical structure is not impossible and likes to improve on these structures, provided that certain cautions are met to prevent pigeonholing (see more on the Theory page and at #5).*

On the other hand, the 5 levels can interact together as a layered TCR, requiring users to first enter top tier before they can go a deeper level.

The proposed 5 structure forms a challenge to incorporate intermediate groups. A list is either part of a certain tier or it isn't. A more fluent, dynamic variety of grouped Nexi is not allowed with such model. This is especially noteworthy on the Tier 2.5 level, between Nexi and Super-Genres.

Parallel to this, contributors can create MUSIC GROUPS, which serve as a combination of different Nexi, without meeting the strict requirements of a Super-Genre. Music Groups are not visually represented on the Carta. Music Groups can be accessed through a separate pane or UX component that shows a list of user-generated music groups. Music Groups is an OPTIONAL feature for the future.

# Actors

*In the following chapter the various actors and their respective roles will be highlighted to analyze their interaction with the network and to establish a basis for simulation.*

## Consumer / User

Consumers are users of the platform who want to browse for music (=consuming the data). Anyone who likes music can be a consumer. Consumers do not need authentication or liquidity.

## Contributor

Contributors add value to the network by proposing Data Entries or Nexi. Contributors must have a wallet and a minimum liquidity as they will have to put up stake in order to propose a Data Entry. Since we are dealing with very small amounts of data with little value, the stake must be extremely minimal. Since musicmap relies on a very large amount of contributors, and particularly outside the blockchain community, authentication and onboarding must show the lowest possible threshold.

There are different types of Contributors:

*Nexus Contributors:*
They propose new Data Entries within an existing Nexus.

*Super-Genre Contributors:*
They propose new Nexi within an existing Super-Genre.

Protocol to propose and accept new data from super-genre contributors will be much more rigorous and with a substantial higher threshold and different parameters than Nexus contributions.

## Curator

Curators judiciously maintain the different registries by voting bad entries out and good ones in. Curators also need to have a wallet and liquidity. The rules when an actor can become a curator are not yet clear. Normally, curators are bound either to their Nexus or their super-genre and thus can only curate their respective registries or markets. Being a curator across different markets is possible if the curators are active in those markets.

There are different types of Curators:

*Nexus Curators:*
They curate a TCR, thus are only required for binary voting. They either vote on a new proposal of Data Entry for a certain Nexus, or they challenge an existing registry Data Entry.

*Super-Genre Curators:* 3
They curate which Nexi are accepted in the network and which aren't. This is rather a Curation Market

where the best Nexi will surface and be displayed, according to a certain ranking or reputation. Survival of the fittest will eventually widen the gap between the high quality markets and the low quality ones. Thus introducing an observable threshold under which low-quality super-genres will be excluded from the carta and the high quality ones.

How to pick the curators? What when none are presented? Failure of momentum for certain genres. Objectivity? How many min./max.? Exclusivity? Curator Fraud?

## 3rd Parties

3rd parties might be involved either as a Data Consumer or a Data Provider (or both). This can be seen as a B2B interaction on the network. When feeling confident, 3rd party providers can propose their own data instead of letting the community link to their database.

To be completed.

# Back End Components

*This chapter explains the various backend components (some of which have a curation/governance aspect which will be further addressed in a separate chapter), and the top-level architecture of the network.*

## Introduction

Overview of back end components

## Datastore I - Framework Database

Triple Datastore:

1. Registry list with only metadata consisting of three parts. Every entry has pointers to the other two datastores.
    a. Super-genre database
    b. Nexi database
    c. Subnexi database
2. Wiki Datastore. Holds text content.
3. Playlist Datastore. Holds all Data Entries. Holds URLs and address of the data entries.

## Datastore II - Graphical Database

All back end information that is displayed on the Carta goes in this database. This database will become updated after certain application processes are accepted in the TCR framework.

This will consist of three parts:

1. Nodes Registry: all the Nexi are displayed on the carta as two-dimensional coordinates. Therefore each nexus string will have x and y parameters.
    <node0478>: (x,y)
2. Links Registry: holds all the lines (relationships) between the nodes, thus creating a visual knowledge graph or genealogy. Links will be curated by the community.
    <link0145>: {node0478, node8794, type}
3. Super-Genre Registry: holds (colored) areas that display super-genres.

## Other Back End Components

# Front End UI/UX

*This chapter explains the various front-end components and how users should be able to interact with them.*

## Carta ("the Map")

*Visual component / UI/UX / The map / Front-End*

The Carta is the name for the user interface which allows users to browse through the various Super-Genres, Nexi, and Subnexi. The Carta shows the various Nexi in an aesthetic manner, similar to the current version of musicmap.

The Carta SHOULD allow for multiple levels of details. At least two (Super-Genres and Nexi).

The Carta SHOULD NOT differentiate between various Nexi: i.e. each Nexus should be portrayed in a similar manner as the next one.

The Carta can be made dynamic by d3.js combined with Canvas.

The Carta will automatically add new Nexi, but it is RECOMMENDED to wait a certain initiation or maturing period. The process of adding a new Nexus (with checks and balances) on the Carta is one of the more difficult challenges that requires further thinking. When reputation is used (i.e. graded TCR's), the Carta will only display the top reputation Nexi.

The Carta CAN be chronological or not. TBD. This a critical decision that requires careful consideration. From the musicmap.info introduction page:

> *Horizontal timelines provide clear information about which year each genre emerged, although for most genres this is disputable, which is why the timelines are faded in the background and surplus information is added in separate genre descriptions.*

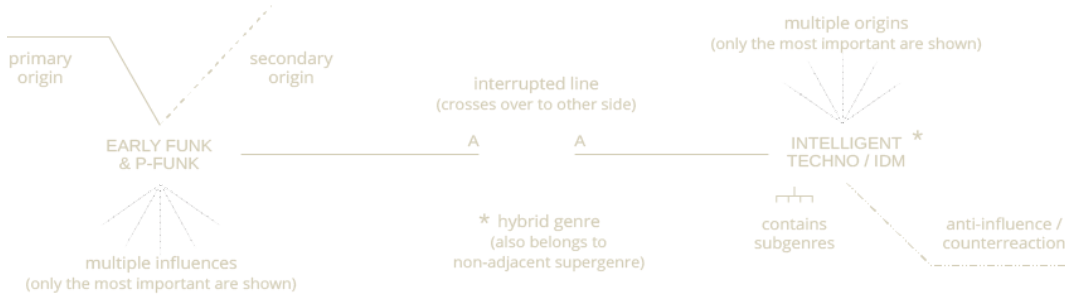
The Carta is designed around the concept of links (see further) and hybrid genres. From the musicmap.info introduction page:

> *The Carta is like a real world map, only the vertical axis represents time (top=past, bottom=future) and the horizontal axis represents the corresponding super-genre. To navigate, simply drag the map where you want to go or zoom in and out. Like a real world map, musicmap is a spherical projection, which implies that the left side crosses over to the right side.*
>
> *When zoomed out, only the super-genres will appear and the main genres and their relationships are hidden. There are even more genre names than visible on the Carta, but these are often subject to debate, too similar to already existing genres, meaningless umbrella terms, or simply very small and obscure, quickly leading to over-classification.*

*If you lose track of date, you can easily check your chronologic context without zooming out by hovering over a thicker horizontal line, which represents the start of a new decade. This will show you the starting year of the corresponding decade. Thinner horizontal lines are placed biannually.*

*Hovering over a genre of interest will highlight all relations from and towards that genre (influences and derivatives) to easily track remote links. Some genres are marked with a small family tree symbol, indicating that there are several substantial subgenres in which the genre further can be divided, or that the genre itself is in fact a combination of several subgenres. Some genres are marked with an asterisk, indicating that they belong to two different super-genres which are not located next to each other.*

*Two or more genres can be located so close to each other that they morph into a continuous zone (very similar genres or "sibling genres"). Sibling genres often share great areas of overlapping, making it hard to separate them. While strictly speaking being different genres, they are mentioned on the Carta as a single entity (using "&"). This as opposed to synonymous genres or genres with negligible differences, where "/" is used instead.*


primary origin
secondary origin
EARLY FUNK & P-FUNK
multiple influences (only the most important are shown)
interrupted line (crosses over to other side)
A
A
* hybrid genre (also belongs to non-adjacent supergenre)
multiple origins (only the most important are shown)
INTELLIGENT TECHNO / IDM *
contains subgenres
anti-influence / counterreaction

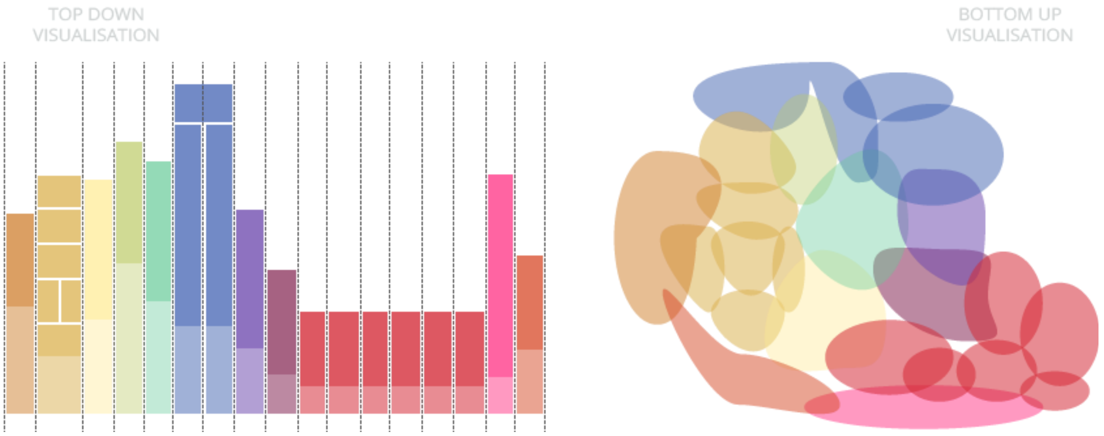
*[example of two different genres as shown on the Carta, with explanation of possible links. Notice the use of "/", indicating synonyms and "&", indicating a combination of two (or more) different but closely related genres into a sibling genre group.]*

2D framework: there are different possibilities in building the visual framework of genres, depending on the decision to make it chronological or not. From the musicmap.info methodology page:

*There are two ways of structuring the popular music network (combining the various super-genres and genres and presenting them in a graphical manner): top-down or bottom-up method.*

*The bottom-up method starts from the determined amount of music genres and places them in a two-dimensional space. Their coordinates are based on the genre-defining characteristics and thus similar genres are located close to each other. Once again, parametrically based programming to create this chart is nigh impossible as the characteristics (parameters) are too broad, not entirely measurable, and even variable in importance (weight). The benefit from this method is that super-genres will eventually emerge as amorphous zones, overlapping and connecting other super-genres at various points. This is a more realistic visualization of the musical genre network, though still limited by two-dimensional constraints.*

*The top-down method starts by laying out the super-genres, usually in a chronologic order. This creates a two-dimensional space where the horizontal axis determines the super-genre and the vertical axis determines the time. The result is a rigid structure with adjacent super-genres in which each genre is placed on a point (or collection of points) that automatically dictates the corresponding super-genre and year of origin.*



*[top-down method and rigid structure (left) versus bottom-up method and amorphous structure (right)]*

*The Musicmap is based on the top-down method instead of the bottom-up for various reasons.*

*Firstly, a true bottom-up method creates an amorphous structure in which chronological order is impossible, unless shaping the form as such, in which case it actually becomes rigid. Chronological order is important, not only to create a better overview and faster navigation, but also to witness the deviant phenomena: when are there many genres, when few? At which point (of time) are there booming events? When does a super-genre cease to perpetuate new styles? And so on.*

*Secondly, the bottom-up method is rather theoretical when it comes to music genres. At one point or another one will always rely on top-down to check the comprehensiveness: are all types of Jazz included? How many genres in Techno are there? Mainly because literature that covers all popular music at once is very, very scarce. Academic literature comes mostly in the form of specialization into one or a few super-genres, thus favoring the top-down method.*

*Thirdly, a rigid structure allows expansion in the vertical axis. This means it can be updated with new genres and adapted to accommodate future genres, hereby greatly increasing flexibility and sustainability. A rigid structure can also easily expand in the horizontal direction, meaning that super-genres can be made broader if need be (to allow corrections, subdivisions and so on). An amorphous structure will enclose certain super-genres, preventing expansion in the future, or correcting if needed.*

*An amorphous structure is better at creating shorter network connections, where all the genres have more strategic positions and the relations between genres becomes more transparent. On top of that, an amorphous structure is also possible of correctly placing hybrid genres: genres that belong to two or more different super-genres. In a rigid structure, these can be serious handicaps, preventing accurate placement of such genres and hindering an accurate amount of links or relationships between genres. Fortunately, there are ways to work around these obstacles and minimalize their negative impact.  Placement on the horizontal axis of super-genres is very critical as the most related super-genres should appear next to each other, allowing as many hybrid genres as possible to be correctly placed. By creating a circular plane, thus connecting the outermost left and right super-genres, more hybrid genres can be placed and shorter connections can be drawn. Hybrid genres belonging to non-adjacent super-genres are placed in either their dominant super-genre or one that makes the most sense, but both their super-genres are mentioned as tags in the description. This is a flaw of the Carta, but a small one that still allows the many advantages of a rigid structure.*

*Finally, a rigid structure allows implementing in existing music databases, although this must be approached with caution. It provides the possibility of transferring its categories (super-genres, and then genres) to a hierarchical database.*

## Link

*Connection / Relationship / UI Vector*

Links are components that are visually represented by vectors between two Nexi. Therefore each link can have a unique name consisting of the two Nexi codenames. Each Nexus thus has two groups of links: links from the past (origins or influences / parent genres) and links to the future (derivatives or evolutions / offspring genres).

Each Link MUST have the following attributes:

<*start*>: origin or parent genre, by <*nxs*>
<*end*>: evolution or offspring genre, by <*nxs*>
<*type*>: type of link (TBD, current types: *primary / secondary / counterreaction / various*)

Links are proposed by the community after the creation of the respective Nexi. In order to create a link, both Nexi MUST exist and be displayed on the Carta.

From the musicmap.info introduction page:

*Special attention was given to inter-categorical relationships, with a different style for primary links (parent genres), secondary links (other influences) and anti-links (backlashes) to make the chart more accurate. […]*

*[goal 2:] To inspire people to explore the world of music, and discover music outside of their comfort zone. Thanks to a tremendous amount of links, and a deliberate compact placement of music genres, people are motivated to track down other genres, within areas they would normally not look. With only a minimal degree of separation, all music genres can eventually be linked to each other.*

Links are not a required component of the interface; it is possible to construct the Carta without links, though NOT RECOMMENDED as it will lo se a lot of its appeal.

## Wiki

*Content / Descriptions*

The Wiki is written content around a Nexus that provides more information: history, description, audio-acoustic features, instruments, geography, subculture, philosophy, and more. Wiki's CAN be formatted in paragraphs according to the previous aspects. The Wiki is a curated text page that not only serves to greatly enrichen the UX and thus attract a bigger community, but also to guide contributors in the right direction when proposing new Data Entries.

From the musicmap.info introduction page (goal 1):

*To inform anyone, regardless of age and education about the basic knowledge of music genres.*
*Musicmap believes that knowledge about music genres is a universal right and should be part of basic education,*

*alongside other forms of art. Because this subject is often not found in school curricula (due to its complexity and a certain disdain for alternative/underground/"low-level" culture), the need for its coverage is high.*

Incomplete, short, or poorly maintained Wiki's SHOULD be marked to attract better curation.

## Nexus Panel

Similar to the back-end, the Nexus panel serves as a hub that holds all functionalities of the Nexus. The Nexus Panel can be regarded as the Portal Page for a certain community or tribe around a specific genre, in which that community can organize and do different (token) governance things.

## Super-Genre Panel

The super-genre panel displays the various attributes of the super-genre, which are essentially two main visible parts: wiki content and curated list of nexi. Users can the super-genre panels as an alternative way of exploring music genres (i.e. without using the Carta, thus in a more encyclopedic manner). Similar to a Nexus Panel, a Super-Genre panel can serve as a portal for an even bigger community.

## Settings & Functions

List of possible functions (integrated in top banner dashboard or left sidebar or hamburger button). Further to be investigated.

- Search Function
- Carta Settings (layers / legend)
- Secure Login for Contributors and Curators
- Carta View (chronological or not)
- User Guidelines
- Legal / Privacy / About
- Connectivity / Social Media

## User Profile Page & Wallet

This is a very simple but secure page that holds basic user information: account name, password change, statistics, wallet assets, purchase and exit options, and link to proposal page.

## Proposal Page

This is a secure page, only accessible by users with an account, to propose new data. On the proposal page users can either propose new Data Entries or Nexi, but only *eligible** options will be enabled (or there will be two different proposal pages if that is recommended). This page has two sections:

*Data Entry Proposal*: 4 text input fields (metadata), automatic display of stake necessary by the amount of proposals, confirmation button

*Nexus Proposal*: the following metadata is a minimum requirement by the contributor and will be subject to curation, and thus a possible basis for denial:

*<name>, <yoo>, <supgtag>, <parent>, <offspring>, <wiki>*

The wiki tag is not a minimum required but a recommended metadata field to prevent flooding of genres and to allow only serious, high-quality proposals, reducing the work of the super-genre curators. Wiki text fields can have a minimum character count requirement (e.g. 1000 incl. spaces).

The <parent> and <offspring> tags can have an additional parameter that explains the relationship, thus establishing the link metadata. For example:

<parent> =

# Governance / Curation

*In the following chapter, the backend components which have a curation aspect will be more examined in detail. The following questions should be answered in this chapter: Which curation building blocks for which components and why? Which parameters will be used? How to implement the proper incentives, bootstrap the registries, and make sure there are no attack vectors?*

*This chapter needs to be completed after simulations, brainstorming and trial environments. The proposed design is entirely theoretical and subject to change.*

## Overview

In its most complex form, musicmap will feature 4 (5 including Data Entries) tiers that form a hierarchial network of Token Curated Registries. This is commonly known as a Nested TCR. There has not yet been much research done regarding nested TCR's. The different tiers and proposed curation building blocks are as follows:

| | | |
|---|---|---|
| 1. Music Universe: | Mother TCR | ((graded)) TCR of Super-Genres |
| 2. Super-Genre: | TCR Group | ((graded)) TCR of Nexi |
| 3. Nexus: | Core TCR (leaf node) | TCR of Data Entries with multilabel |
| 4. Subnexus: | Fragmentation TCR (leaf node) | TCR of Data Entries with multilabel |
| (5. Data Entry: | Listing | ) |

From a curation mechanics point of view, it is assumed that all TCR's eventually spawn from the mother TCR, which will be first (and the only one at that point in time) in place. However, side-consequences of such "top-down" instead of bottom-up approach must be thoroughly researched.

Another question arises whether or not to allow "an infinity of lists" and to shift the high quality lists from the low quality ones by grading them by reputation (upvotes or downvotes), thus making a graded TCR or gTCR. The problem with this approach in musicmap is that a subjective hard-to-determine threshold needs to be defined above which the lists will be included in the UI. The search for corresponding reputation parameters also adds further complexity and increases the attack surface. Last but not least, graded TCR's require more effort from the community. Not only do they need to vote listings in or out, they also need to upvote or downvote. It is highly unlikely in a massive ecosystem as musicmap, that such enormous community effort will be achieved on a balanced manner. And what does this reputation mean? It is likely people will vote on Nexi with "likes" instead of "this list is properly curated", further skewing the ecosystem.  It makes more sense to keep it simple and let the high tier TCR's be binary. This however, has two major negative consequences: when a Nexus00 gets outvoted, all the associated data entries will be lost and a huge amount of work can be made undone. Another problem is the creation of two Nexi that are too similar. WIthout a reputation system, they both will be deemed equal but one will be redundant and unnecessary clutter to the UI.

## Minimum component: TCR mechanics of a Nexus

Let's examine the most fundamental, basic aspect of the network: adding a new Data Entry (thus a single song) to a Nexus registry (thus choosing a genre for that song). This is based on the AdChain mechanics and most likely subject to change.

### GENERAL / NO CHALLENGE

When a user wants to become a proposer and eventually a token holder, they need to create a secure user profile. Once logged in to the network, the user can propose a new data entry from an outside source. Each proposition will have a PROPOSER_STAKE attribute - which is assumed for now a fixed parameter (amount, MMT) – attached to the Data Entry and the Proposer Identity.

The required stake SHOULD be automatically calculated based on the number of proposed Data Entries, exchanged into the network's raw currency ($, €, BTC, ETH, or OCN), and deducted from the Proposer's wallet that holds raw currency. This prevents hoarding tokens without using them. Speculators who only buy tokens without using them, are out of luck, because this function SHOULD NOT be allowed. Proposer wallets only hold raw currency which is automatically converted to MMT when – and only when - a data proposal is being made. The price in raw currency for each token is calculated by the bonding curve.

Once a proposition is introduced to the network, all the eligible token holders SHOULD be immediately signaled and the initiation period starts. The initiation period or INIT_PERIOD is a fixed parameter (duration, sec) after which the Data Entry automatically gets accepted and added to the registry. Automatic acceptance is a basic TCR concept that is for musicmap certainly necessary to bootstrap the network and to prevent growth issues.

After the initiation period, the stake goes to a separate special wallet that holds all the user's tokens. This wallet directly reflects the user's contribution to the network and is used to determine the user's holding rights. This means that whoever contributes the most (does the most work) will have the most voting power (see further).

### CHALLENGE ACCEPTED

During the initiation period, the curators will have the chance to challenge the proposal. Anyone who has tokens is an eligible curator, which means anyone who has proposed a data entry at least once that was successfully accepted. If a challenger (or multiple, TBD) arises, they will have to put a matching stake themselves: CHALLENGER_STAKE (=PROPOSER STAKE). The community of other token holders will then vote whether the challenge is justified. Their voting power is a direct derivative of the amount of tokens that they hold. This is why speculators must be prevented in the game.

CHALLENGE REJECTED

## Mechanics of deploying and curating a new core TCR (Nexus)

The following describes more in technical detail all steps in the fundamental curation process within musicmap. These steps consist of two main parts: deploying a new TCR, which means defining and adding a new Nexus to the network, and adding new Data Entries to that TCR (or another existing TCR). For simplicity purposes, only a two-tier framework is taken into account here (singular mother TCR and many Nexi).

1. New Nexus smart Token gets deployed by the smart contract (BANCOR based?). This gets implemented by a Contributor (or do we want a preset framework based on mm 1.0?).
2. Nexus Proposer needs to stake a min deposit of tokens (which tokens?, what price? Dynamic price? Dependant on network size?). The friction here must be significantly higher than proposing a new listing (Data Entry).
3. Within the mother TCR, a new Nexus TCR contract becomes created with its native token, and with generic parameters based on the mother TCR.
4. Token Distribution Definition ?
5. Bootstrapping of registry?
6. The Nexus Registry can now be used for new listings.

   ------------*second part*------------------

7. A Contributor proposes a new Data Entry to a certain TCR (let’s call this Nexus ABC). In order to do so, they must have ABC Tokens.
8. The apply function locks the min_deposit tokens and takes the string (URL) of the data entry
9. Several helper functions perform background checks before the application process starts:
   a. Check duplicity of listing (is the URL not already added to the network?)
   b. Check pending application (isn’t there already a pending application?)
   c. Check whether entered deposit size is according to min_deposit.
   d. Illegal/Pirated Response: check if the content is according to law.
10. After successful checks, a data structure becomes initiated
11. The application stake tokens are transferred from the contributor’s account to the registry
12. The expiry date becomes set with the application stage length

## §Recommended settings for the canonical parameters (1.1)

The following parameters are part of the canonical TCR smart contracts, as introduced by Mike Goldin. These are also the main mechanics used by AdChain or made dynamic by vote in the EasyTCR UI. A review (1.1) has been introduced and a paper for a bigger rehaul (2.0) is on the way. For each parameter, recommended settings for musicmap are described, but - as always - subject to change.

### MIN_DEPOSIT

Must be made dynamic per tier, the higher the tier, the higher the deposit.
When MIN_DEPOSIT is high, MINORITY_BLOC_SLASH must be 100% to prevent vote splitting behaviour!
WIthin a Nexus, MIN_DEPOSIT will be very low (to prevent friction), allowing for a lower MINORITY_BLOC_SLASH.
For SG creation (within the mother TCR), the MIN_DEPOSIT will be very high.
MIN_Deposit = MAX_Deposit? Does it makes sense to allow whales to brute force applications with high stakes? For Data Entries, it is probably advisable to set the deposit at a fixed price.

This parameter can be removed in registry.sol 2.0 which forces contributors to make their own choice of application stake (which can also be zero). This subjective and dynamic deposit is known as a TRUST_POOL

### APPLY_STAGE_LEN

The duration during which a new proposal gets voted in or out of the system. Normally the proposal is automatically accepted if the apply_stage_len ends and there are no challenges.

Very important for high tiers, must be prolongued. Has serious UI implications.
After the application stage, it is assumed that the Data Entry will be copied to the Graphical DB (if accepted), implying that it will be showed on the dynamic UI.
For Data Entries added to a Nexus or Subnexus however, it is advisable to set the application length to zero, thus instantly adding new songs to a certain genre without having to wait for a vote from the community. Since the community has infinite time to challenge the proposal, this makes sense.

### COMMIT_PERIOD_LEN

Infinite by default to allow registered Data Entries to be outvoted.

### REVEAL_PERIOD_LEN

### VOTE_QUORUM

The percentage needed when a vote gets accepted or rejected. 50 = default majority vote.
Be aware that this parameter applies to the token amount, and thus assumes that 1 token = 1 vote.

Quorum must be increased for higher tier. E.g. 67 = ⅔ majority vote. When the community is allowed to change TCR parameters (not recommended), it should be a very high quorum.

### DISPENSATION_PCT

Percentage of the forfeited deposit that is awarded to the winning party (if there is a challenge). In Adchain, this is 50%.
This can be a dynamic function to allow difficult votes to hand out a bigger reward. In that case dispensation_pct increases by the asymmetry of the voting parties.

<50 low risk, low reward / >50 high risk, high reward, better to keep at 50.
Anything too low (10%) or too high (90%) will result in a churn.

### MINORITY_BLOC_SLASH (1.1)

Slash the minority tokens (percentage based). If 100%, all tokens get forfeited to winning side.
Should be 100% when dealing with high MIN_DEPOSIT (high tiers, SG). For lower tiers, this might be better to be lower, so losses are not as bad. Recommended to keep fairly above 0% to prevent vote splitting.

### EXIT_FREEZE (1.1)

Procedures that can be put in place when a data entry exits (removed by the contributor). Normally, the contributor gets the initial token stake back, but these tokens might be freezed for a certain period of time to increase skin in the game.

### TRUST_POOL (2.0)

Replaces the MIN_DEPOSIT parameter. Basically determines a stake range between min_value = 0 and max_value = token_pool. Remains to be seen if this will actually be implemented in the canonical TCR contracts.

## Optional Community-Based Smart Contract Parameters

The following incentives or optional parameters have all been introduced in one way or another by the community, and therefore have a small basis of analysis and research, but - needless to say - never been tested. Some of these parameters are purely theoretical and the possibility of actually writing them into Solidity language has not been tried yet. Some of these are additions to the registry.sol contract, some to the PLCR.sol contract, and some need to be determined. Only incentives or parameters that might be of any use to musicmap have been withheld here, as there are many more possibilities out there.

### ARBITRATION

Allowing the TCR parameters in the smart contract to be changed by "Arbiters" or by community vote. It is probably not recommended.

### VOLUNTARY DILUTION RATE

This allows tokens to be minted for new applicants with no capital. It will dilute the token pool but something else (non-financial) can be asked in return.

### NATURAL CAP

Hardcoded cap at the number of entries, e.g. for highest tiers. Caps are interesting to prevent wild growth. This cap can be made approximately 50 for Supergenres and 1000 for Nexi.

### STAKED_I and CHALLENGED_I

Add staked_i tokens and challenged_i tokens to a data entry, so multiple challengers can challenge a listing. Rewards are then dispensated among all challengers, incentivizing difficult decisions and counterattacking majority votes. This is highly recommended for musicmap.
DUr

### VETTING STAGE

Apply a separate vetting stage before application stage starts to filter certain requirements, metadata and to response against illegal data, similar to Ocean. Part of the illegal response approach.

### INFLATIONARY REWARDS

Provide additional block rewards to challengers or defenders, like Ocean. This probably serves little use in musicmap.

### STAKE_TIMELOCK

Tokens that are being used to vote will be locked for a certain period to increase skin in the game and to prevent token whales with little knowledge from curating purely for speculation.

### ? FIXED_REWARD

Gives a fixed amount to the winning party on top of the distributed tokens to prevent the spread of misinformation or censorship by malevolent actors to decrease the amount of winners.

## Unique Musicmap Smart Contract Parameters

In this section we will introduce highly experimental new parameters that are invented specifically for musicmap.

Further to be determined.

# Token Engineering

*This chapter explains the possible Token economics that form the underlying mechanics for the network, how they are used for creating incentives, and which parameters need to be investigated.*

## Token

Being a unique marketplace, it requires its own token, which will for now be temporarily be called Musicmap Token (MMT, placeholder name). Speculation SHOULD be discouraged and velocity of MMT (using the Tokens for contributions and curation) encouraged. The most active contributors and curators of the network SHOULD be rewarded accordingly with the highest amount of tokens.

## Adding a Data Entry (Proposal within a Nexus)

Whenever a certain Data Entry (or group of Data Entries like a music album) is proposed by a Contributor, he has to put up a certain fixed stake per data entry. That stake will become locked for a certain period of time, in which the curators of the given Nexus decide whether to decline or accept the proposal. If declined, the contributor loses the stake.

Much work for the curators – problem? Automatic acceptance or automatic denial after period? Curators will become lazy and go for the majority vote?

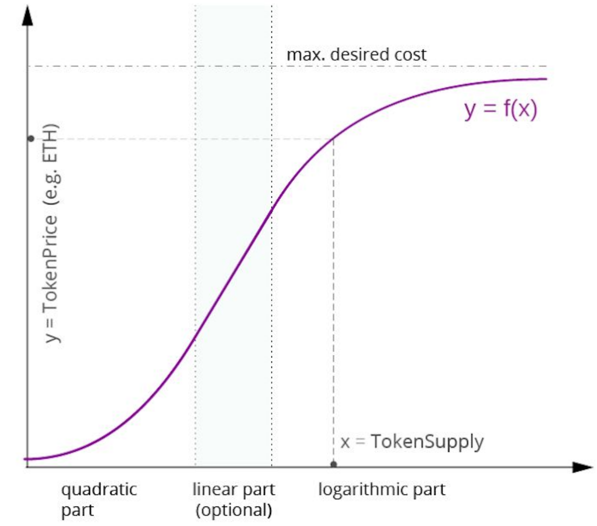
It is possible to only allow a Data Entry to be added to the registry after proposal of at least 2 or 3 different actors (built-in automatic curation at the primary stage).

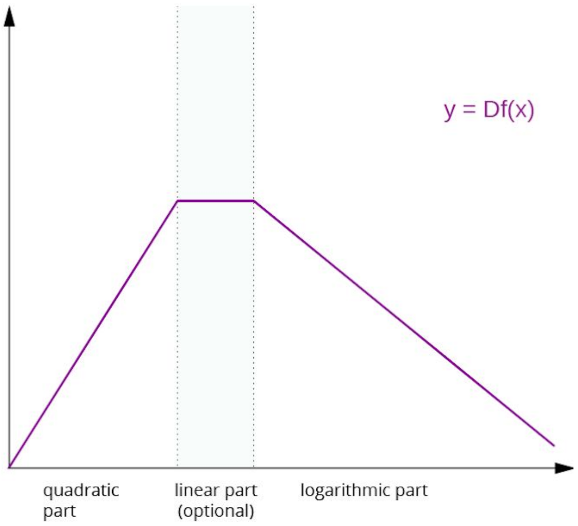
## Bonding Curve

Bootstrapping a certain Nexus list is very important to the overall success. The network requires a large community and is mostly (or strictly) B2C, therefore a good nascence must be ensured by a bonding curve with a quadratic component in the beginning. Because a certain Nexus should allow for an infinite number of entries, a linear component looks attractive, but at a certain point it is a recommended to keep the TokenPrice at an economically desirable maximum threshold. Thus bending the curve again, only this time slower than the quadratic component (by a logarithmic part) to stretch out the obtainable size of the lists.

Each Nexus SHOULD have the same bonding curve (no discrimination – it is better to apply bounties or penalties (see further) on different parameters.

Super-Genres MUST have a different bonding curve than Nexi or none at all => fixed price. This is because they are completely different markets (finite, small-amount lists, high proposal friction).





## Bounties and Penalties

It can be safely assumed that a gap between extensive well-maintained Nexi and small, poorly maintained Nexi will quickly ensue after network launch. Certain music genres will have a much stronger and more active community in the realm of internet and blockchain. Early adaptor rewards within other Nexi might simply not be enough to establish a desired equilibrium at the third tier. Bounties (or penalties) in the form of tokens can be implemented to attract curators and contributors to a certain nexus that does not meet a MIN_SUPPLY after a stipulated time period.

## Creating a Nexus  (Proposal within the network)

Contributors can propose a new Nexus. When creating a new Nexus, the contributors must assign (a) existing Super-Genre(s) to its supgtag attribute, which will be the subject of the stake being put up. The following values are accepted:

Singular Nexus (1 supergenre)
Hybrid Nexus (2 supergenre)
… Nexus (1 dominant and 1 submissive supergenre)

Curators can choose to decline the proposal based on the wrong supgtags, or because the nexus functions as a subnexus to an existing nexus, or simply because the genre is not deemed a real music genre (but rather a playlist or mood).

It is RECOMMENDED that contributors follow a certain set of guidelines when proposing Nexi. From the musicmap.info abstract page:

*The history of Popular Music hardly qualifies as an exact science. It is a retrospective analysis of events that focuses on the underlying forces or common symptoms in the overwhelming production of music records, ignoring nuances and side-effects to grasp a comprehensible structure. This is because (popular) music is far from a static* *phenomenon: it is a constantly evolving, transforming, giant organism. Almost never has a music genre suddenly emerged as a shocking revolution without any trace or evolution in the past. All of them have naturally evolved, mutated, merged, or become (theoretically) extinct. Only the past can be examined of this natural, organic network.*

*Herein one often finds network nodes or concentrations of artists having multiple traits in common and thus forming a genre. More correctly speaking: certain albums or a collection of songs by different artists, since an overwhelming number of musical artists cannot be classified within the constraints of a single genre. Many artists attempt to create a unique and distinctive sound, crossing over into near and distant genres, while also evolving in sound during the course of their albums. This is why, in case-specific literature, the same artist can be a given example for different genres. This means that often artists are located on the connection between two (or more) nodes instead of right on the node (genre) itself.*

*The traits that define a genre are more than a similar sound or summary of technical elements; subculture, fashion, geography, mentality and period of time all qualify as possible trademarks of which a genre in retrospect might be recognized. In other words, music genres can be seen as a concentration in the musical network, as the common denominator of a large enough group of music (not necessarily artists) connected by a specific instrumentation, technique, mentality/ideology, sound, place and/or time. By taking only the most important concentrations into account, a two-dimensional, comprehensible genealogy can be created. Genre definitions must be regarded independently of language. Changing the language of the vocals does not create something new (which is why e.g. French Rap is not listed as a genre), though geographical differences might influence the constraints of the genre.*

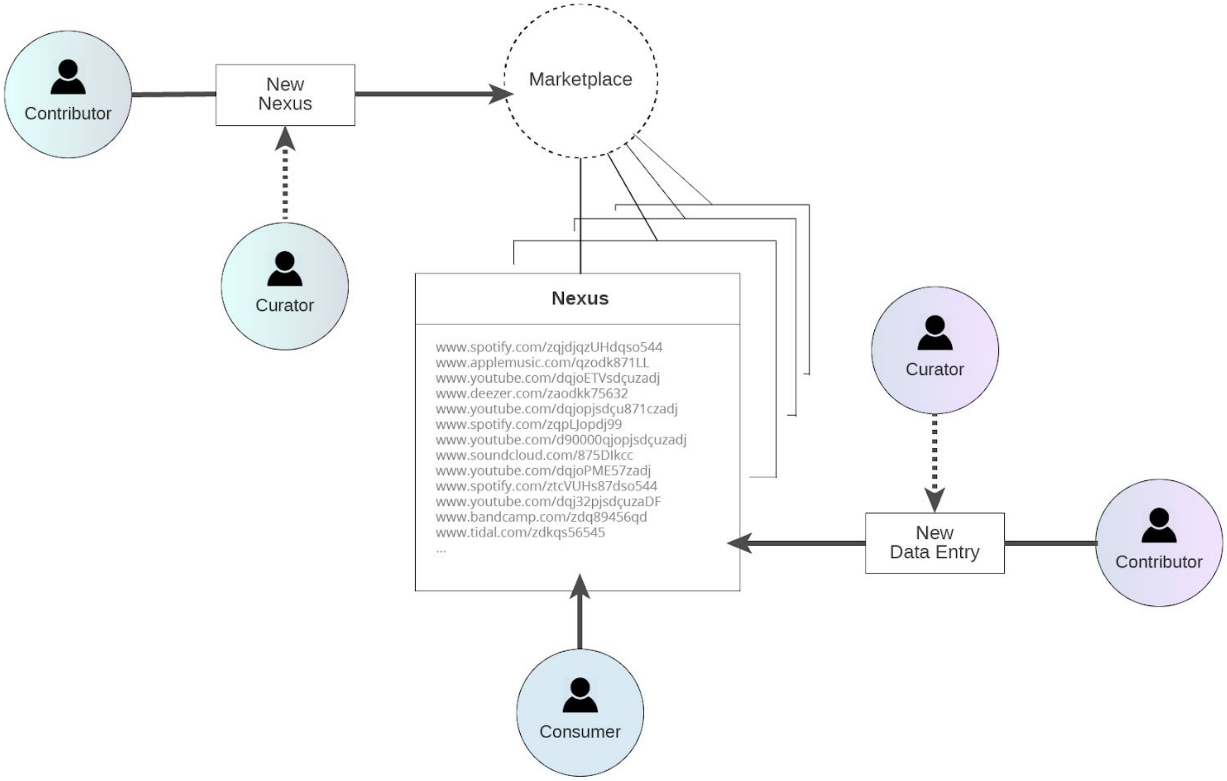
*The amount of popular music genres is staggering, reaching easily over 600. Many genres however, can either be seen as a subgenre of a larger genre or as very similar to another one (sibling genres). Metaphorically speaking, these genres are either smaller nodes connected to bigger network nodes or they are nodes located in high proximity to other nodes. There is no objective way to determine the true amount of genres. Musicmap is not the result of a parametrically based algorithm. The amount and selection of mentioned genres are determined on their importance and uniqueness by academic research. They are carefully selected by comparing as many genealogies as possible and examining their context and relevance in literature. There is no mathematical threshold involved to identify a genre, certainly not in the amount of practicing artists (which would also be a flawed approach for aforementioned reasons). Some genres can be extremely small (e.g. Old Skool Rap, Gothic Rock, Grunge, Musique Concrète) yet form an important link and source of inspiration for other genres. Other genres can be huge, yet their offshoots or subgenres vary too little to become a separately mentioned genre (e.g. Disco, Singer/Songwriter, Death Metal, Psytrance).*

*The name for a genre is often coined by a music journalist as an attempt to summarize underlying currents in past events, sometimes many years after the date of origin when the boundaries of the common denominator characteristics have finally become clear. The urge to do so in a concentrative manner (a noun or genre name) can lead to unfortunate results, i.e. something "new" or "progressive" in the eighties can hardly sustain this feat in present times. In other occasions, artists themselves come up with a name to describe their style, often compared to already existing genres. Another possible origin is by mixing records from two different genres, either consecutively or simultaneously. When the crowd starts to welcome these mixes, new records get made to approximate this combination and a new genre becomes born. More often than not, genre names are poorly chosen and reveal little information about the true nature of the genre. There is however no debate about which genre is which: once denounced, the genre name stays infinitely.*

*Because of the large focus on retrospective analysis, it is important to take all music genres from 2010 until now with a grain of salt, because their definitions are still forming. The ink is still wet; there is no telling where it might run.*

*Musicmap does not display any end time of music genres, only the (approximate) year of origin. A music genre after all, cannot die. It is never impossible to recreate a specific genre, even though the sociologic circumstances have* *been changed. Sometimes a return to one particular genre of the past happens on a larger scale, with updated equipment, sound and small alterations: a revival. Revivals form a big part of popular music in the last two decades, marking a possible saturation point of styles, and the perfect starting point for a summary of popular music genres thus far.*

In a first phase, Tier 4 (super-genres) (and Tier 2 as well) can be omitted to test the minimum required network, consisting of only Nexi within a singular marketplace.


Contributor
New Nexus
Marketplace
Curator
Nexus
www.spotify.com/zqjdjqzUHdqso544
www.applemusic.com/qzodk871LL
www.youtube.com/dqjoETVsdçuzadj
www.deezer.com/zaodkk75632
www.youtube.com/dqjopjsdçu871czadj
www.spotify.com/zqpLJopdj99
www.youtube.com/d90000qjopjsdçuzadj
www.soundcloud.com/875DIkcc
www.youtube.com/dqjoPME57zadj
www.spotify.com/ztcVUHs87dso544
www.youtube.com/dqj32pjsdçuzaDF
www.bandcamp.com/zdq89456qd
www.tidal.com/zdkqs56545
...
Curator
New Data Entry
Contributor
Consumer

*Highly simplified graph of the network, showing only the minimum governance components.*

## Creating a Super-Genre  (Proposal within the network)

Contributors can also propose a new Super-Genre of course, which is a top level proposal that should be rigorously controlled with strict rules. The lower the tier in the network, the more widespread is allowed.