Value-Based Security Requirements in a Highly Decentralized Digital Ecosystem: the MUSIC360 Case

Arthur Mclaren[1,*,†], Yulu Wang[1,†], Charlotte van de Velde[1,†], Ed Green[2,†], Roel Wieringa[2,†] and Jaap Gordijn[1,†]

[1] *Vrije Universiteit Amsterdam, The Netherlands*
[2] *The Value Engineers*

Abstract
Security requirements in digital ecosystems should be first and foremost driven by the notion of economic value. Attackers are interested in objects of value, which are exchanged between actors in a network of organizations and end-users. We explain how we derive data-oriented security requirements by means of high level goals, a value model representing the ecosystem, and a data model detailing the value model. The approach is illustrated and validated in a complex ecosystem in the music sector.

Keywords
ecosystem, security requirements, music, conceptual modeling

1. Introduction

Many innovative services are offered by digital business ecosystems (DBEs). We define a DBE as a system of economic actors that depend on each other for their survival and well-being [1]. A DBE has an associated value network which explains what actors trade with each other.

One such innovative ecosystem is the EU Horizon Europe MUSIC360 ecosystem[1], which aims at providing insights into the value of music to (1) creatives (performers and authors of music), venues (restaurants, retail shops, offices), and policymakers (EU officials, national authorities and lobbyists). The MUSIC360 ecosystem provides data about the value of music, where 'value' can have an economical, societal, or cultural connotation. To do so, MUSIC360 collects data about music used in venues (e.g. by installing music finger-printing devices in venues to discover what is actually played), effects of music played (for example increased revenue), and meta data of music such as rightholders (performers and authors) and earnings by rightholders. Important players are the collective management organizations (CMOs). They collect money from venues in return for a license to play music and pay the money subsequently (minus an administrative) fee to the right holders). This data should be protected for reasons of privacy, but also misuse of data by hyperscalars such as Google, Amazon, Meta, and Alibaba should be discouraged.

We will protect privacy-sensitive data by means of access controls. More difficult is the prevention of misuse by hyperscalars. The risk is not so much unauthorized access but taking over the full ecosystem by one of these hyperscalars. We mitigate this risk by constructing a very decentralized way ecosystem and also employ decentralized governance in which power and decision taking is well balanced. This makes it more difficult to buy the complete ecosystem,

All this implies that we have to design a highly decentralized platform that still should behave in an integrated way, where parties remain in control of their own data, but nevertheless can together present a coherent view on the value of music, and privacy is respected. This paper presents the

Companion Proceedings of the 17th IFIP WG 8.1 Working Conference on the Practice of Enterprise Modeling Forum, M4S, FACETE, AEM, Tools and Demos co-located with PoEM 2024, Stockholm, Sweden, December 3-5, 2024
*Corresponding author.
†These authors contributed equally.
a.a.mclaren@student.vu.nl (A. Mclaren); y.wang4@vu.nl (Y. Wang); c.s.d.vander.velden@student.vu.nl (C. v. d. Velde); ed@thevalueengineers.nl (E. Green); roel@thevalueengineers.nl (R. Wieringa); j.gordijn@vu.nl (J. Gordijn)

[1] see https://www.music-360.eu

requirements engineering process to achieve such a highly decentralized, privacy preserving ecosystem, and evaluates it. The focus is on security requirements, which are about access controls for data stored. The contribution of this paper is that we show how to base security requirements and policies on the value produced, distributed and consumed in a digital ecosystem.

1.1. Contributions

This research aims to contribute twofolds:

- **Methodology** This paper introduces a value-driven approach to security requirements engineering, where security measures are based on the economic value exchanged in a decentralized ecosystem. By combining the e^3*value* model with UML class models, the methodology ensures that security is aligned with stakeholders' valuable assets and interests.
- **Case Study** The MUSIC360 ecosystem is used as a case study to demonstrate and validate this methodology. It shows how this approach works in practice within a multi-actor environment, highlighting the benefits and challenges of applying value-based security in real-world settings.

1.2. Research problem

The research problem addressed in this paper is the challenge of establishing security requirements and policies within a digital ecosystem based on the value that is produced, distributed, and consumed. This approach shifts the focus from traditional security models to one that is more aligned with the economic dynamics of digital ecosystems, aiming to enhance the relevance and effectiveness of security measures by tying them directly to the flow of value within these environments.

This paper is structured as follows. In Sec. 2 we discuss related security requirements engineering approaches. Then, in Sec. 3 we discuss our value-based security requirements engineering approach, followed by the goals of the ecosystem (Sec. 4), the value model (Sec. 5), the IT architecture (Sec. 6), and the structure of the data to be secured (Sec. 7). Finally in Sec. 8, we evaluate our approach and present conclusions (Sec. 9).

2. Related Work: Security Requirement Engineering

Security Requirement Engineering (SRE) is a critical domain in the development of secure software systems/platforms. It involves the identification, analysis, specification, and validation of security requirements to ensure that software systems are robust against threats and vulnerabilities [2]. Methodologies include Goal-Oriented Requirement Engineering (GORE) [3] frameworks such as KAOS and i*. According to the comparative work done by [4], the KAOS framework systematically derives security goals and refines them into operational requirements and constraints, while the i* framework models organizational environments and relationships to identify and mitigate threats through goal refinement and actor dependency analysis.

Misuse and abuse cases are also a way to conduct a security requirement analysis which focuses on capturing and modeling systems' unauthorized use, introduced by [5] and further developed by [6]. This method develops negative scenarios by modeling potential threats and attacks, aiding in the identification of security requirements. Risk-based approaches like the CORAS methodology and OCTAVE provide model-based and comprehensive risk assessment frameworks, respectively, to align security requirements with identified risks [7].

These frameworks can be integrated into the systematic processes provided by tools like Secure Tropos and SQUARE (Security Quality Requirements Engineering), which offer structured processes for integrating security requirements into the development lifecycle. Secure Tropos incorporates security concerns into early development stages using actor-based models [8], while SQUARE involves a nine-step process for eliciting, categorizing, and prioritizing security requirements [9]. Ansari et al. present the STORE methodology for security requirement elicitation, which emphasizes early identification of security threats during the software development process [10]. Additionally, Maskani et al. advocate for a comprehensive approach to SRE that reconciles different perspectives on security requirements, such as user, threat, and goal perspectives [11].

In contrast to traditional goal-oriented approaches, we propose deriving security requirements based on the concept of economic value. This approach directly links security needs to the valuable assets within the ecosystem, addressing the challenges of securing decentralized systems where multiple stakeholders interact. By focusing on economic value, our method ensures that security is implemented in areas where stakeholders have the most at risk, such as sensitive data, intellectual property, and financial transactions.

3. Towards a secure and decentralized MUSIC360 ecosystem

We propose in Fig. 1 a step-wise approach to find security requirements in ecosystems, to protect data, and to keep data ownership at the data owners in order to prevent misuse of the data by for example hyperscalars.

Our approach begins by identifying the high-level goals of the ecosystem, which are then linked to the economic value exchanged among stakeholders. Using the e^3*value* model, we map out the flow of value and the actors involved, ensuring that each step of the process accounts for the security needs of these stakeholders. This value model is then integrated with UML class models to detail the data structures involved, while security requirements are systematically elicited and documented based on these models. This process is applicable both within the MUSIC360 ecosystem and more broadly to other decentralized systems, allowing for flexibility and scalability.

An essential element of our approach is that we elicit a value model of the ecosystem, cf. the e^3*value* [12] method. The reason for this is the following: an e^3*value* model focuses on what actors exchange between each other, in terms of value objects. These objects may represent money, but also other valuable things, such as data or mandates to exploit music. The actors in an e^3*value* model show the affected parties, e.g. if data is stolen, and the value objects represent what can be stolen. Usually, attackers are only interested in things of value (e.g. kinds of data they can resell). Hence, we argue that each security requirements engineering process should elicit a value model in an early stage, to identify what is of value to whom.

The high-level security requirements, the e^3*value* model, and requirements of other categories (out-of-scope for this paper) are the input for the design of the IT architecture.

The MUSIC360 ecosystem is specifically concerned with protection of the data in the system, e.g. to prevent theft of data or abuse of data by hyper-scalars. To this end, we have defined a specific sub-process for data security requirements. Using the e^3*value* model as input, data requirements are elicited (e.g. what kinds of data will be stored by the system) and represented as a UML class model. This class model is the primary source for finding data security requirements. These are used to define data access controls, which are effectively access policies per data element.

The process ends with a (prototype) implementation that can be validated by the MUSIC360 actors.

Each activity requires two or more workshops with the relevant stakeholders to find the requirements and create the design. In the remainder of this paper, we illustrate each step in the process further.

4. MUSIC360 Goals

The MUSIC360 project goals are already stated in the project proposal to the EU, and was the result of a number of workshops with the partners involved. To make the paper self-contained, we briefly review the requirements below:

- **G1:** A fair distribution of neighboring and author rights based on the actual use of music. For the MUSIC360 project, two intellectual property rights are relevant: (1) the neighboring right which

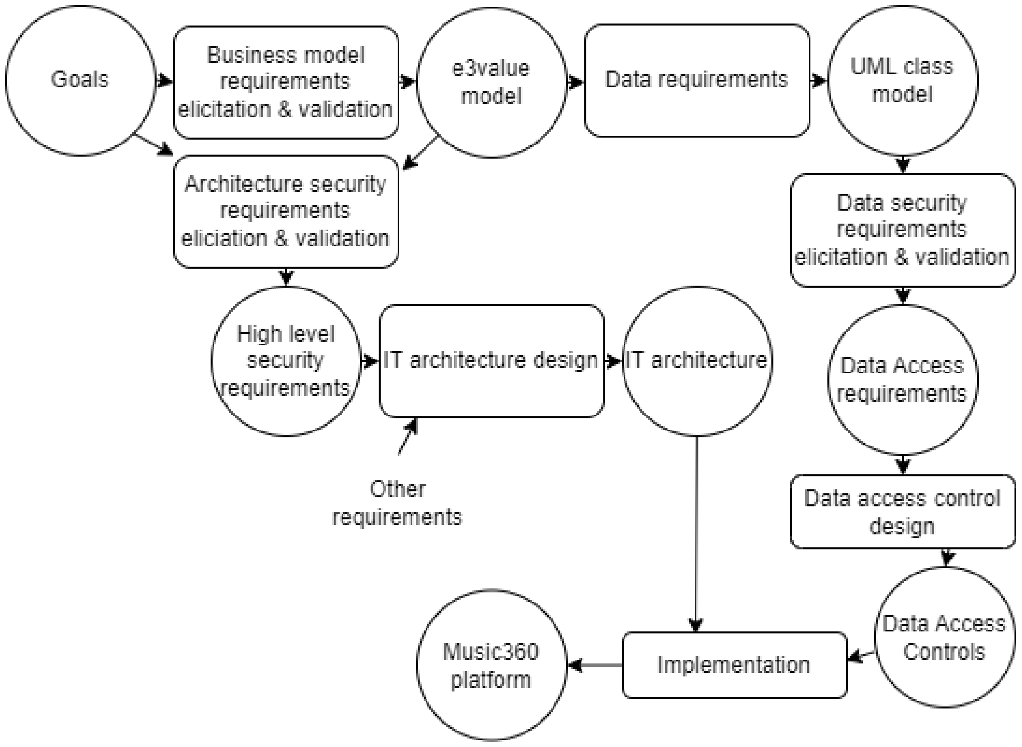
Goals
Business model requirements elicitation & validation
e3value model
Data requirements
UML class model
Architecture security requirements eliciation & validation
Data security requirements elicitation & validation
High level security requirements
IT architecture design
IT architecture
Data Access requirements
Other requirements
Data access control design
Data Access Controls
Music360 platform
Implementation

Figure 1: MUSIC360 security requirements and design process. Circles represent artifacts cf. Design Science [13], bounded rectangles represent activities, and arrows indicate the control flow. We report on the (data) security requirements and design.

results from playing music in public, e.g. a restaurant or a shop, and (2) the author right which is the result of written lyrics and scores by an author.

- **G2:** Establish a method to value music, such that is usable to motivate a 'fair compensation' for the use of music. The law prescribes that if music is played in public, rightholders are entitled to a fair compensation, without actually stating what 'fair' actually is. MUSIC360 argues that fair compensation should be based on the value of music for its users.
- **G3:** Wide and transparent availability of information concerning the value of music respecting confidentiality requirements. Currently, information about the (value of) music is available in a fragmented way, e.g. at record labels, collective management organizations (CMOs), companies measuring the actual use of music, and the rightsholders themselves. MUSIC360 wants to make this information available in a consistent and integrated way, which is given the state-of-the-art in the music industry a problem by itself. However, quite some information is privacy sensitive, e.g. the rightsholders of a particular recording, what their earnings are, and also the price of the licenses venues have to pay to CMOs. It is clear that this last goal embodies an important security requirement, namely that information about music should only be available to the right stakeholders. Consequently, in this paper, we continue with this last requirement.

5. Ecosystem value model

After stating the high-level goals, we elicit and design a value model cf. the e^3*value* modeling language [12]. This value model not only highlights the flow of assets but also serves as the foundation for deriving security requirements. By focusing on valuable assets like rightsholder earnings and license data, the value model ensures that security measures are prioritized for areas of high economic risk. The security requirements in Sec.6 are directly tied to the value exchanges illustrated in the model, ensuring that each requirement addresses the protection of economically significant assets. For this case, we follow a normal elicitation process for e^3*value* models, as explained in [14]. For reasons of brevity, we immediately present the resulting e^3*value* model in Fig. 2.

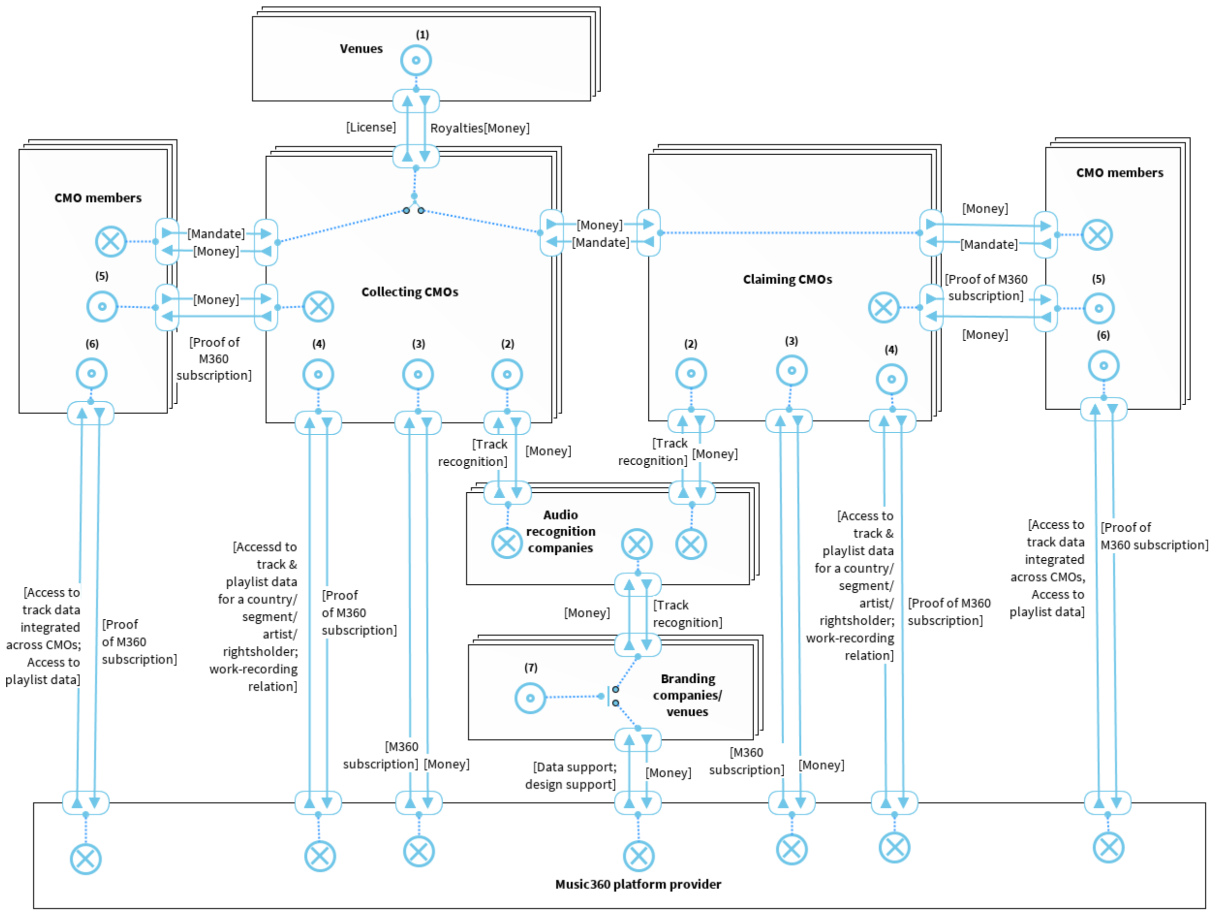
Venues
(1)
[License]
Royalties[Money]
CMO members
[Mandate]
[Money]
(5)
[Money]
[Proof of M360 subscription]
(6)
Collecting CMOs
(4)
(3)
(2)
[Money]
[Mandate]
Claiming CMOs
(2)
(3)
(4)
[Money]
[Mandate]
[Proof of M360 subscription]
[Money]
CMO members
(5)
(6)
[Track recognition]
[Money]
[Track recognition]
[Money]
Audio recognition companies
[Money]
[Track recognition]
(7)
Branding companies/ venues
[Access to track data integrated across CMOs; Access to playlist data]
[Proof of M360 subscription]
[Accessd to track & playlist data for a country/ segment/ artist/ rightsholder; work-recording relation]
[Proof of M360 subscription]
[M360 subscription]
[Money]
[Data support; design support]
[Money]
[M360 subscription]
[Money]
[Access to track & playlist data for a country/ segment/ artist/ rightsholder; work-recording relation]
[Proof of M360 subscription]
[Access to track data integrated across CMOs, Access to playlist data]
[Proof of M360 subscription]
Music360 platform provider

Figure 2: E^{3}value model for the MUSIC360 platform

At the top, venues such as shops and restaurants pay for their license to play music in public. They always pay for this license to the collective management organization (CMO) in their country. Then two things can happen: (1) if the CMO member (performer or author) is in the same country as the collecting CMO, money is paid by that CMO to the CMO member, or (2) if the CMO member is in another country, the money is transferred to the claiming CMO of that country, who subsequently pays the CMO member. This captures the essential operations of CMOs.

CMO members can use the MUSIC360 platform to access data they are interested in: e.g. data about tracks played (containing both recordings and music works), playlist data, and also data about their revenue. CMOs also have access to similar data about their members, and about usage data in other countries than the one they represent. Finally, there are value-adding companies, such as music branding companies. They curate playlists with the intention of creating brand awareness through music in a venue. Similarly, there are market research companies who, on behalf of venues, do experiments to find out whether music indeed adds value (e.g. increased revenue) for the venue. These parties need data support for experiments, e.g. monitoring of what is played in a venue. Additionally, they need metadata on the music, e.g. its genre, tempo, language used, etc.

Services are provided by the MUSIC360 platform. This is modeled as a single actor, but in reality reflects the coalition of its members, including the required governance. The precise structure is under design, but we foresee that the platform at least consists of CMOs, and other data providers, such as audio recognition companies.

The e^3*value* model is useful for eliciting security requirements in a number of ways. First, it shows the actors (profit-and-loss responsible parties). Since actors have an economic value perspective by definition, for security considerations it is important to distinguish them. If a security breach occurs, or data is misused by hyper scalars, the actors in the e^3*value* model are the victims. Second, the e^3*value* model gives a first glance on the valuable assets, because value objects such as 'access to track data' and 'playlist data' are valuable objects by definition and hence subject to attacks.

6. IT Architecture

We perform a normal requirements elicitation process for architectural requirements using several workshops with all stakeholders involved. This results in a list of requirements, which we summarize below:

- **R1: The MUSIC360 ecosystem should support multiple parties of different stakeholder types**, namely creators (performers and authors of music), collective management organizations (CMOs) who collect money for the public use of music on behalf of the creators, venues (the users of music), policy makers, and commercial entities who make music available (e.g. Spotify or other streaming parties). Most of these parties come from the e^3*value* model. Others (e.g. streaming parties) have been added later in the process and should be fed back into the e^3*value* model. **Solution:** Different parties, such as creators, venues and policymakers can access the MUSIC360 ecosystem via a browser, which serves as client. The browser communicates via the OAuth 2.0 / OpenId Connect protocol [15] and REST services [16] with the rest of the components. Each of these backend components can be provided by a single party.
- **R2: The data owner remains in control with respect to its own data**. One of the critical success factors for the MUSIC360 platform is that all parties, specifically those who own data, remain in control with respect to their data. Most parties have data that is private and competitively sensitive.
  - **R2.1: Data is available to a user if they possess clearly defined access rights that align with their role in the ecosystem.** Data is stored in databases which are under control of the data owner (e.g., physically hosted at the premises of the data provider trusted to the owner, or in a data centre, still under control of the data owner). Access to data should be restricted at least at the instance level (e.g., a recording, work, rightsholder), and preferably at the property level (earnings of the recording, use of the recording). **Solution:** Access to data is controlled by two complementary approaches. First, data access is restricted by using other data in the databases. For example, performer X may only access its own recordings and music works. This often referred to as Attribute Based Access Control (ABAC) [17]. Second, data can have Access Control Lists (ACLs). These lists relate data to a user or group identifier, which is compared to the authenticated user. This is called Discretionary Access Control (DAC) [18]. Together, these mechanisms are highly flexible access control solutions and are implemented as policies in the database management system.
  - **R2.2: Data owners may delegate access control management to others**. An example of this scenario is a creative entity which allows their representing CMO to make its data available under certain conditions. Often this data is clustered and enriched with additional metadata. In this scenario, the CMO is responsible for data storage and access control on behalf of the creative entity. **Solution:** The architectural solutions allows data owners such as CMOs to stay in control of their own data, but they may also outsource parts (e.g. the authentication server) to a trusted party with an appropriate legal contract.
  - **R2.3: Data owners can make data available in an inaccessible, containerized way** (that is: someone who tries to access the data cannot read the data, but with a set of allowed predefined operations users can do calculations on the data (e.g. arithmetic operations)). The result of the operations can be inspectable or not, depending on the use case. It allows to make data available to untrusted parties, without disclosing the data, but allowing them to do some controlled operations on the data. **Solution:** We do not solve this requirement in this paper but come back to it in Sec. 9 concerning Future Work.

- **R3: Authentication should be done by entities owning the data**. As data owners should remain in control, they should also authenticate users of the system. Data owners may decide to trust each other authentication services. Also, they may decide to trust an external authentication party such as Google or GitHub. As data owners trust the other authentication providers, misbehaving authentication providers are not considered as an attack vector. **Solution:** Each data owner operates an authentication server cf. OAuth 2.0 / OpenID Connect to authenticate users to its own data.
- **R4: Data of different stakeholders should be properly isolated**. This means that parties (specifically data providers), cannot see the data of the other MUSIC360 participants. **Solution:** Depending on the data storage solution chosen, this requires specific measures. In others words, all data will not be stored in one database, where all the data owners have full access to.
- **R5: The MUSIC360 system should be as open as possible**. **Solution:** From a technical perspective, this implies the use of (de-facto) open standards.
- **R6: The MUSIC360 system should be scalable**, with respect to the number of users and the amount of data. **Solution:** The vast majority of data operations are READ operations, meaning that data enters the system once, and is queried many times. This allows for replication and/or partitioning of the very large data set, hence distributing data over a large number of decentralised databases and hardware.

The most important security requirements for this paper are based on R1, R2, R3 and R4. We have also discussed the other requirements because it explains some choices made for the architecture.

7. Data security requirements and access controls

Data model. To understand the security requirements with respect to stored data, we first have to know the structure of the data itself (for MUSIC360, we do not use unstructured data). The e^3*value* model provides hints concerning the data elements, such as track data, country/geographical segment, rightsholder, work, recording, playlist, license and mandate. However, an e^3*value* model is meant to understand the actors involved in an ecosystem, and what they exchange of value with each other. Therefore, for a detailed analysis of the data required, we executed a data modeling task using standard UML class modeling. This effort takes several workshops, and results in a data model in Fig. 3.

For reasons of brevity, we explain the most important elements. Rightsholders (e.g. performers or authors) have claims on a creation (either a recording or a work). Rightsholders can transfer their claim to a beneficiary and can use an agent to represent them. Collective management organizations (CMOs) collect compensations for usage of claimed intellectual property rights by venues. CMOs have a mandate for creations (often recordings) which are played, usually as part of a performed playlists by venues (shops, restaurants, etc.). These venues have a license to do this, which they obtain from a CMO. Plays are monitored by a company, called the monitor, doing identification of the played recording of work. One or more plays of a recording or work result in earnings, via a claim, for the rightsholder.

Data security requirements. We elicit the data security requirements in several workshops using the following steps:

- **Stakeholder identification**. The stakeholders are based on the actors in the e^3*value* model and further detailing in the UML class model. There are CMO members (rightsholders, beneficiaries, and their agents), CMOs themselves, and venues.
- **Asset identification**. Also, assets are based on the e^3*value* model and the UML class model (see Table 1). Some classes in the UML are so-called non-assets meaning that there are no access restrictions, mostly because it concerns publicly available data (see Table 2).
- **Determination of access**. Finally, using the identified stakeholders, and the identified assets, we systematically elicit the required and allowed access rights using a series of workshops. In

Table 3, we exemplify the result for the asset ‘Rightsholder‘. We concentrate here on READ access. Only data owners have the right to CREATE, UPDATE, and DELETE data.

Table 1
Selection of assets

Assets	
Class	**Motivation**
Claimant (and subclasses)	Contains personal data
Agent	Contains personal data (in the case of a non-company agent), and has possible competition sensitivity
Claim	Could link to earning or rightsholder. In some cases, it is also not public knowledge that a rightsholder has a claim on a creation
Earning	Contains personal financial data
Play	Indirectly infers personal financial data
License	Sensitive information regarding fees/amount paid for license
Representation	Sensitive due to link to Claimant, Agent, and Claim
Performed playlist	Playlists curated by venue or playlist provider could be sensitive (IP issue)
Mandate	What mandates a CMO has for which artist can be competition sensitive between CMOs
Venue (and related classes)	In the initial phase of the project, the participating venue(s) have expressed the wish for their participation to remain private

Table 2
Selection of non-assets

Non-assets	
Class	**Motivation**
CMO	Data stored is only the name and identifier
Monitor	Only used to identify certain (physical) monitors
Playlist provider	In the current state of just storing the name, it is not sensitive
Creation, Work, Recording	All data is publicly available

Access Controls. Access controls are expressions that determine who can access which data. They are directly based on Table 3 and similar tables for the other assets. Access controls are part of the database. More precisely, we use Row Level Security (RLS) policies which determine, based on attributes such as the user identifier, the data elements someone is allowed to read. The design of the access controls therefore concerns the transformation of all asset access conditions as defined in Table 3 and results in RLS policies. Given that there is a one-to-one correspondence between the assets and the relational model in the database, this is a straightforward task.

8. Evaluation

We evaluate our approach designing a data-secure architect using the following criteria:

- Connection Between Value Model and Security Requirements

Table 3
Asset access: Rightsholder

Rightsholder	
Can be accessed by	**Under the condition**
Rightsholder	owns a Claim connected to a Creation connected to a second Claim that is owned by the Rightsholder (that is being accessed)
Beneficiary	1. operates a Claim owned by the Rightsholder, or 2. operates a Claim connected to a Creation connected to a second Claim that is owned by the Rightsholder
Agent	1. represents a Representation that represents the Rightsholder, or 2. represents a Representation that represents a Rightsholder who owns a Claim connected to a Creation connected to a Claim that is owned by the Rightsholder
CMO	1. represents a Mandate that justifies a Claim owned by the Rightsholder, or 2. represents a Mandate that justifies a Claim owned by a Rightsholder who owns a Claim connected to a Creation connected to a second Claim that is owned by the Rightsholder
Venue	no access

- Correctness and completeness
- Effectiveness and efficiency
- Ease of implementation.

They have been validated with the stakeholders, through internal review, and with the developers respectively.

8.1. Connection Between Value Model and Security Requirements

The value model introduced in Sec. 5 (Fig. 2) is central to understanding how security requirements are derived in the MUSIC360 ecosystem. The value model outlines the economic exchanges between key actors, such as venues, CMOs, rightsholders, and other participants, highlighting the valuable assets at stake, including data, money, and rights. These economic flows inform the security requirements, ensuring that they protect valuable assets from potential threats.

Rightsholder Earnings Protection. One of the key assets identified in the value model is the rightsholder earnings (Sec. 5). Rightsholders receive compensation for the public use of their music, with the flow of funds moving from venues through CMOs to the rightsholders. The security requirement in Sec. 7 ensures that only relevant stakeholders, such as rightsholders and their authorized agents, can access this sensitive earnings data.

The value model illustrates the economic flow, showing how earnings travel through the ecosystem, which directly informs the access control policies in R2.1 and R2.2. For instance, only rightsholders or their representatives can view their earnings, while CMOs are limited to viewing data relevant to their members. This structure reflects the economic relationships shown in the value model and ensures that financial data is protected according to the stakeholders' interests.

License and Usage Data Security. In the value model, venues (e.g., shops and restaurants) pay for licenses to publicly play music, with those payments flowing to CMOs. The license data and the usage data collected by monitors are essential for determining rightsholders' compensation. The value model illustrates the data exchange between monitors, CMOs, and venues, highlighting these as critical assets.

As a result, the security requirements ensure that only venues with valid licenses can access their specific license fee and usage data, safeguarding commercially sensitive information. Additionally, monitors have restricted access to this raw data, preventing them from unauthorized use or resale, which aligns with the concern of protecting valuable data from hyperscalars, as noted in R2.3.

Multi-Actor Governance and Data Control. The multi-actor nature of the MUSIC360 ecosystem, highlighted in the value model, means that different stakeholders, including CMOs, rightsholders, and venues—have distinct roles and exchange various forms of value. Each stakeholder maintains control over their own data, requiring security requirements that reflect these relationships.

This leads to the development of security policies in R2.1 and R2.2, which ensure that data owners (e.g., rightsholders, CMOs) control access to their data. For instance, a rightsholder controls who can access their earnings and usage data, while a CMO can delegate data control to trusted agents, reflecting the decentralized structure of the ecosystem.

The value model helps ensure that security requirements consider these multi-actor relationships, allowing for both flexibility and strong protections.

Data Security for Cross-Border Transactions. The value model shows that cross-border transactions, such as when rightsholders reside in different countries, involve the transfer of money between CMOs. This introduces risks related to data privacy and security during these international exchanges.

The security requirements in Sec. 7 address these risks by enforcing encryption and anonymization during the transfer of personal and financial data between CMOs. The value model illustrates where data crosses national borders, ensuring that the necessary protections are in place to comply with regulations like the General Data Protection Regulation (GDPR)[2].

Intellectual Property (IP) Protection and Mandates. The value model captures the process by which rightsholders grant mandates to CMOs, allowing them to collect license fees on their behalf. These mandates represent valuable intellectual property (IP) rights, which must be protected against unauthorized access.

The security requirements in Sec. 7, supported by R2.1, focus on protecting IP data. Unauthorized access to this data could lead to exploitation of rightsholders' works without proper compensation. Therefore, security measures such as discretionary access control (DAC) and attribute-based access control (ABAC) ensure that only authorized users can access the IP-related data.

Limitations. While the value model provides a structured way to derive security requirements, it still has some limitations as follows.

- The value model focus on economic exchanges may overlook non-monetary assets or intangible values, such as reputational risks or stakeholder trust, which could also influence security needs.
- This method assumes that all actors are clearly defined and that their economic roles and exchanges remain stable, which may not hold true in dynamic ecosystems where new stakeholders or value flows could emerge over time.
- For now, the approach relies on workshops and stakeholders' input to validate security requirements. If certain stakeholder groups are underrepresented or not fully engaged, there is a risk that their security concerns may not be adequately addressed, potentially leading to gaps in the overall security framework.

8.2. Evaluation: Correctness and Completeness

A validation workshop with stakeholders was held to validate the proposed access control rules. During this session, we examined and discussed each asset and its corresponding rules in order. This was done

[2]https://gdpr-info.eu/

while presenting a copy of the data model to clarify and refer to the specific assets and relations we were discussing. As expected, there was feedback on our initial proposal which has been processed. Aside from this feedback, however, we did not receive a further indication that specific (general) rules were missing. Also from a reasoning standpoint, the collection of access control rules contained for *each* asset and *each* main user/stakeholder under which general condition they would need to have access. The combination of this reasoning and the stakeholder validation leads us to conclude that the methods used led to a sufficient degree of correctness and completeness.

Since development is an iterative process, wishes and insights will likely change throughout the project's lifetime. This highlights the importance of keeping the stakeholders involved and iteratively validating whether the access control rules still match their needs and expectations, even after initial deployment. Among others, [19] also highlights this as an important aspect of security requirement engineering in general. Also specifically for access control policy in particular, the iterative nature of the process is highlighted in research [20].

Limitations. We were not able to directly contact all stakeholder groups. Instead, we were able to extensively elicit from a stakeholder group, namely the CMOs, that is (in theory) aware of the wishes of most other stakeholder groups. However, this is still an important limitation to the assessment of whether the elicited access control rules are complete and correct. This is because we could not directly involve all the most important intended users of the system. This goes against the principle of eliciting directly from the user, which is commonly accepted to yield better requirement results [21].

8.3. Evaluation: Effectiveness and Efficiency

In general, we evaluate the process to be sufficiently effective and efficient. A (sufficiently) complete and correct list of general access control rules has been produced within the allotted time, with only 3 sessions with external stakeholders. During the process we experimented with different methods of formatting the proposed controls/results, so in future iterations, this would likely be even more effective.

We were not able to speak to all stakeholder types directly. In the case of MUSIC360 the stakeholder group we were able to elicit from extensively (CMOs) is in contact with all other main stakeholders. This allowed them to provide substantiated reasoning for their wishes. Especially during the initial phase, this had a positive effect on efficiency since we only had to set up limited interviews and/or workshops before reaching an initial proposed set of rules. This indicates that in time- or resource-constrained projects eliciting only from a select group of stakeholders that is knowledgeable about other groups can at least provide the basis for the initial proposal. This does not take away that to improve validity the results would need to be validated with the other stakeholder groups themselves. However, to reach the initial proposal, sessions with a stakeholder group that is (at least to an extent) knowledgeable about the wishes of the other stakeholder groups, could be sufficient.

Before the final validation workshop, the assets had not been validated for completeness and correctness. Instead, we used the final validation workshop both for validating the proposed list of assets as well as their corresponding access control rules. This led to a situation where we had missed some resources that could be considered as assets, and in turn had not prepared the proposed access controls for these assets. For future implementations of our methods, we would suggest validating the list of potential assets separately from the final access control validation workshop.

Limitations. Due to logistical issues, not all the researchers were able to attend the second exploratory workshop in person. In our case, the semi-hybrid format made it harder for the online participants to interact. While research has been done on hybrid formats for education and courses, we find a gap regarding its effects on the requirement elicitation process in particular, which could be an angle for future research.

The workshops we held were with participants who had a relatively technical background. They were not all computer scientists or developers but had generally worked (extensively) with technical projects during their careers. This potentially allowed the currently chosen method of presenting the access controls to be more well-understood by these stakeholders, than by completely non-technical stakeholders. During this research process, we used the same tabular format for presenting the controls to these stakeholders (with explanation) as to the developers. In the case of less technical stakeholders, we expect that a more visual presentation method may be more suitable. Specific research into formats and language for presenting access control rules for stakeholder validation could be an angle for future research.

8.4. Evaluation: Ease of implementation

There are several aspects necessary to implement access control policies regardless of the access control type being implemented.

- Which user (types) need to access a resource?
- What access type should they have? (CRUD)
- Under which condition?

The produced access control rules contain all three of these components. This in turn means that the rules are implementable. In this case, the access rules were all Read access (except for CMO admin/fingerprint devices/music providers), but for cases where this would differ per line, an extra column stating the type of access could easily be added.

Regarding ease of implementation in terms of clarity, we tried multiple ways of presenting the controls. The presented format was decided upon through discussion within the team on the level of clarity per presentation approach. The methods tried were:

- A tabular overview from the perspective of the user, for purely the identified assets. In other words "User X can access, Resource Y under condition Z, can access A under condition B, etc".
- The same as above, for non-assets as well
- A tabular overview from the perspective of the asset (as was the final chosen presentation type), however, presented within a single table
- A visual diagram style representation from the perspective of the asset.

Based on an evaluation by the developer and team members on clarity, we decided that an overview per asset was the preferred method.

Another option would have been to provide the access control rules in code format. The benefit of this would have been an even higher level of realizability for the eventual developer. We expect that this could be preferred for some projects where:

- The eventually chosen access control type is (as good as) set in stone
- The language/framework that the rules will be implemented in is certain
- The requirement engineer is adept in the language/framework that the rules are to be implemented in

When these are not the case, we expect it to be more effective to provide the logical reasoning behind access control rules as a separate document, as has been done in our results. In other words to stay away from particular code formats. These criteria would have to be tested over a larger number of projects to assess whether they constitute an effective rule of thumb regarding the formatting of access control rule deliverables from the requirement engineer(s) to the development team.

Limitations. The validation for ease of implementation was limited to other people working on the MUSIC360 project. To make truly grounded claims regarding ease of implementation for this specific format, a higher sample size and possibly implementations over different projects would be needed. In our case, the tabular format for presenting access control rules was found to be effective. However, which exact wording patterns or structure to use has not been particularly researched. Nor can our research provide grounded claims on whether the type of wording we used is best for enhancing ease of implementation. We propose future research to focus specifically on best practices and conventions regarding framing access control conditions for improving ease of implementation. Including for example further support for which situations allow handover in code to be effective.

9. Conclusion

Value-based Security Requirements. In this paper, we have proposed an approach to elicit security requirements in digital ecosystems, starting with identifying high-level goals, understanding what is at stake by means of an e^3*value* model, deriving a UML class model, and finally identifying asset access rights and related access controls. We evaluated our approach in terms of correctness, completeness, efficiency, effectiveness, and ease of implementation.

Future Work. We have not covered **R3** 'Data owners can make data available in an inaccessible, containerized way'. We plan to address this requirement with Multi-Party (MPC) computing [22] and homomorphic computing [23]. These are techniques to do calculations without knowing the complete data, or knowing the data at all. Also, it can be controlled by which parties can read the outcome. An example use case is that a CMO is interested in the effect of a playlist on revenue of retail shops in The Netherlands. However, the revenue is only known to the shop itself, and can not be shared in plain text. However, the *aggregated* revenue number for all relevant shops is visible to the CMOs. There is a need for computation of the total revenue in a sort of secure container where no one knows the inputs (the revenue numbers of the individual shops) but where the CMOs know the aggregated results. It is clear that it is a different class of security requirements than access controls only. We plan to address this with MPC computing, most notably secret sharing [24] and homomorphic encryption.

Acknowledgments

Funded by the European Union under project no. 101094872. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

References

[1] J. Gordijn, R. Wieringa, The business model of digital ecosystems: Why and how you should do it, in: C. G. M. J. S. Guerreiro (Ed.), Advances in Enterprise Engineering XVI, Lecture Notes in Business Information Processing, Springer-Verlag, Germany, 2023. URL: https://dise-lab.nl/eewc__2022_invited_talk-24-03-2023/.

[2] P. Salini, S. Kanmani, Survey and analysis on security requirements engineering, Computers Electrical Engineering 38 (2012) 1785–1797. URL: https://www.sciencedirect.com/science/article/pii/S0045790612001644. doi:https://doi.org/10.1016/j.compeleceng.2012.08.008.

[3] A. van Lamsweerde, Goal-oriented requirements engineering: a guided tour, in: Proceedings Fifth IEEE International Symposium on Requirements Engineering, 2001, pp. 249–262. doi:10.1109/ISRE.2001.948567.

[4] V. Werneck, A. d. P. Oliveira, J. Leite, Comparing gore frameworks: I-star and kaos, 2009.

[5] J. P. McDermott, C. Fox, Using abuse case models for security requirements analysis, Proceedings 15th Annual Computer Security Applications Conference (ACSAC'99) (1999) 55–64. URL: https://api.semanticscholar.org/CorpusID:206580824.

[6] J. Heikka, M. Siponen, Abuse cases revised: An action research experience, 2006, p. 46.

[7] N. Mayer, A. Rifaut, E. Dubois, Towards a risk-based security requirements engineering framework, in: Proceedings of REFSQ, volume 5, 2005. URL: https://api.semanticscholar.org/CorpusID:8837909.

[8] H. Mouratidis, P. Giorgini, Secure tropos: A security-oriented extension of the tropos methodology, International Journal of Software Engineering and Knowledge Engineering 17 (2007). doi:10.1142/S0218194007003240.

[9] N. Mead, T. Stehney, Security quality requirements engineering (square) methodology, ACM SIGSOFT Software Engineering Notes 30 (2005) 1–7. doi:10.1145/1082983.1083214.

[10] M. T. J. Ansari, D. Pandey, M. Alenezi, Store: Security threat oriented requirements engineering methodology, Journal of King Saud University-Computer and Information Sciences 34 (2022) 191–203.

[11] I. Maskani, J. Boutahar, S. E. G. El Houssaïni, Analysis of security requirements engineering: towards a comprehensive approach, International Journal of Advanced Computer Science and Applications 7 (2016).

[12] J. Gordijn, R. Wieringa, E3value User Guide - Designing Your Ecosystem in a Digital World, 1st ed., The Value Engineers, 2021.

[13] A. R. Hevner, S. T. March, J. Park, S. Ram, Design science in information systems research, MIS quarterly (2004) 75–105.

[14] J. Gordijn, H. Akkermans, Value Webs - Understanding e-Business Innovation, The Value Engineers, 2018. URL: https://www.thevalueengineers.nl/product-category/publications/e3value-ref/.

[15] D. Hardt, The OAuth 2.0 Authorization Framework, RFC 6749, 2012. URL: https://www.rfc-editor.org/info/rfc6749. doi:10.17487/RFC6749.

[16] L. Richardson, S. Ruby, RESTful Web Services, O'Reilly, Beijing, 2007. URL: https://www.safaribooksonline.com/library/view/restful-web-services/9780596529260/.

[17] A. Singhal, T. Winograd, K. Scarfone, Guide to secure web services, National Insitute of Standards and Technology (2007).

[18] V. Hu, D. Ferraiolo, D. Kuhn, Assessment of access control systems, National Insitute of Standards and Technology (2007).

[19] B. Fabian, S. Gurses, M. Heisel, T. Santen, H. Schmidt, A comparison of security requirements engineering methods, Requir. Eng. 15 (2010) 7–40. doi:10.1007/s00766-009-0092-x.

[20] Q. He, A. I. Antón, Requirements-based access control analysis and policy specification (recaps), Information and Software Technology 51 (2009) 993–1009. URL: https://www.sciencedirect.com/science/article/pii/S0950584908001699. doi:https://doi.org/10.1016/j.infsof.2008.11.005.

[21] S. Kujala1, Effective user involvement in product development by improving the analysis of user needs, Behaviour & Information Technology 27 (2008) 457–473. doi:10.1080/01449290601111051.

[22] C. Zhao, S. Zhao, M. Zhao, Z. Chen, C.-Z. Gao, H. Li, Y. an Tan, Secure multi-party computation: Theory, practice and applications, Information Sciences 476 (2019) 357–372. URL: https://www.sciencedirect.com/science/article/pii/S0020025518308338. doi:https://doi.org/10.1016/j.ins.2018.10.024.

[23] A. Acar, H. Aksu, A. S. Uluagac, M. Conti, A survey on homomorphic encryption schemes: Theory and implementation, ACM Comput. Surv. 51 (2018). URL: https://doi-org.vu-nl.idm.oclc.org/10.1145/3214303. doi:10.1145/3214303.

[24] A. C.-C. Yao, How to generate and exchange secrets, in: 27th Annual Symposium on Foundations of Computer Science (sfcs 1986), 1986, pp. 162–167. doi:10.1109/SFCS.1986.25.

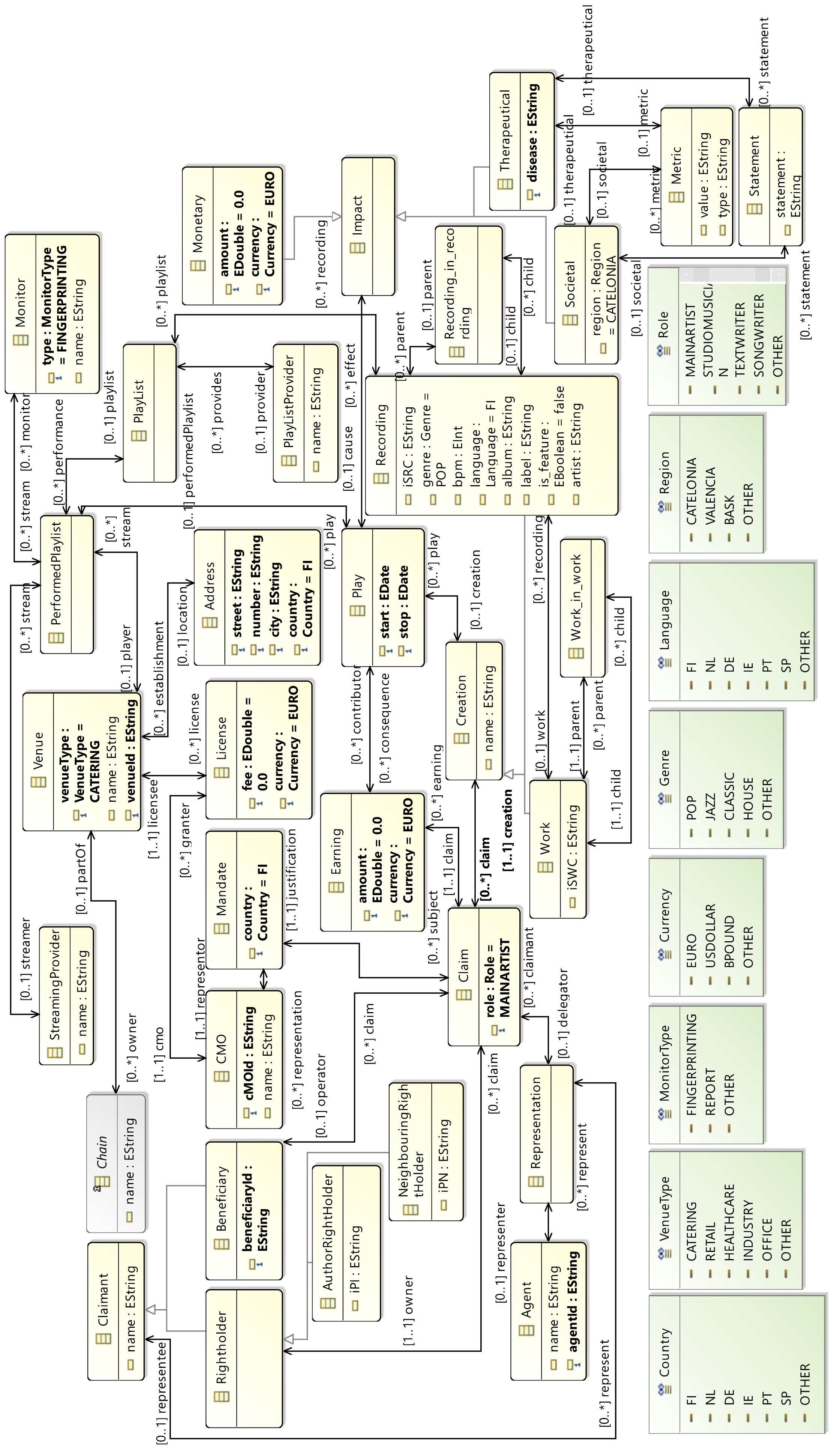

Figure 3: The MUSIC360 data model