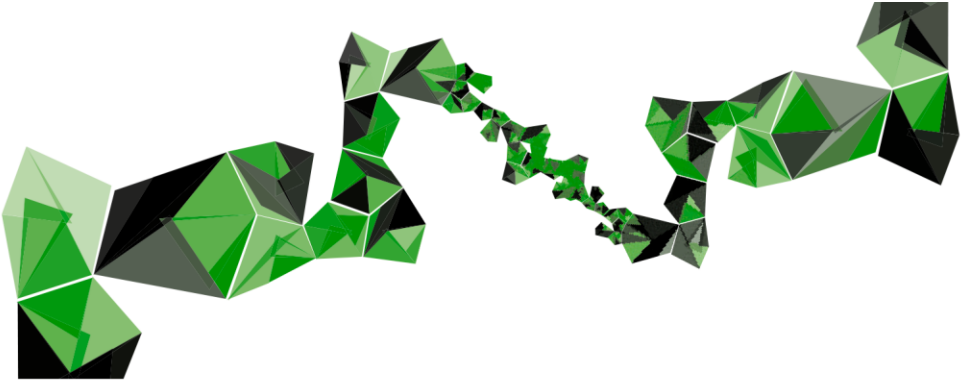

UNIVERSITY OF TWENTE.

Faculty of Electrical Engineering,
Mathematics & Computer Science

Improving an iPaaS through the addition of enterprise data catalog features

Jelle Johan Smits

Master thesis

Supervisors:
Dr. ir. Marten van Sinderen
Dr. Lucas Meertens
Dr. João Rebelo Moreira

External supervisor:
Samet Kaya

Executive summary

This research evaluates two software solutions for organizations and possible opportunities in combining them. These are the integration Platform as a Service (iPaaS) and the enterprise data catalog. An iPaaS provides a platform for organizations to create integrations between different systems, whereas an enterprise data catalog focuses on making data of an organization visible and usable to all employees who need it. This research is focused on applying the features of an enterprise data catalog to an iPaaS. This direction is chosen since an iPaaS aims at reducing the complexity of creating integrations between software applications, which traditionally is a more complex task. Adding features of an enterprise data catalog to an iPaaS is hypothesized to improve the workflow of users of an iPaaS. This research identifies which features of enterprise data catalogs are relevant to add to an iPaaS, and provides an example application for one iPaaS vendor.

The inspiration for this research originates from the increased dependence of organizations on data, and the increase in the sources that produce this data. This increase in data sources is enabled by developments such as the Internet of Things (IoT), which enables all kinds of sensors to produce high amounts of data, as well as the increase in the number of applications that are used within an organization. With this increase, difficulty in interpreting the data due to its vast amount can occur. Especially taking into account that the raw data often needs to be transformed into another format before it can be used can impose difficulties. Both an iPaaS and an enterprise data catalog provide a solution to this.

In addition, the position of enterprise data catalogs and iPaaS platforms are both at a central position within an organization's enterprise architecture, although they are focused on different groups of users. An enterprise data catalog focuses on providing an inventory of data assets to all business users who might need data, whereas an iPaaS is used to register applications and ensure that these applications are integrated. Although the functionality is fundamentally different, both applications include data and operations on this data.

As approach, this research starts by identifying features of an enterprise data catalog from the literature. Since the literature on this topic is limited, the literature on open data catalogs is also evaluated. This is combined with a market analysis of enterprise data catalogs to obtain an inventory of the features offered by the data catalogs of major vendors at the time of this research. In order to design a solution to include enterprise data catalog features into an iPaaS, the iPaaS of one vendor is used. From this vendor, a panel of experts is consulted to identify which of the enterprise data catalog features would be relevant to include in their iPaaS.

Based on the findings of these experts, a prototype is created using the iPaaS of this vendor to demonstrate the validity of the design. Users and developers of the iPaaS are interviewed to identify to which extent these features are relevant for their iPaaS if it solves problems that they experience, and whether the design of the proposed features is relevant for the use case of the iPaaS. It was found that the features as shown in the prototype are relevant additions to the analyzed iPaaS. Users find them to be relevant, and from the vendors' perspective, the new features solve issues that customers currently experience when using the platform. The steps taken in this research are summarized as a framework. This framework describes how the findings of this research can be used by iPaaS vendors to identify which features of enterprise data catalogs are relevant to add to their iPaaS.

Acknowledgments

On the paper or screen in front of you is the thesis which is the result of the research which concludes my Masters' in Business Information Technology. This thesis marks the end of my study and the start of my professional career. I am very grateful to everybody who supported me in producing this thesis, either directly or indirectly.

Throughout this research, my university supervisors were Marten van Sinderen, Lucas Meertens, and João Rebelo Moreira. I am grateful for their trust in my progress, flexibility in planning meetings, and constructive feedback to improve my work. They have challenged me to get the most out of this research. My thanks go out for their guidance, time, and effort they spend supervising me throughout this research.

I had the privilege of conducting my research at a company. I would like to thank them for providing me with the inspiration for this topic, for the opportunity to work at their office when possible, and for the involvement of the staff. Special thanks go out to Samet Kaya, who triggered me through discussions, as well as providing me with literature and documentation, helped me with my scope, and involved his colleagues to provide me with valuable input.

Throughout this research, I have depended on the help of numerous experts who participated in the expert panels. Their input was key to completing this research. I am grateful for the time they were able to make in their busy schedules.

When conducting this research, we were unfortunate enough to go through yet another coronavirus lockdown. Therefore, I am grateful for my friends, my parents, and family who provided the much-needed distractions and support. I would like to especially thank my girlfriend Elise for her ongoing support, and distractions and for motivating me when it was really needed.

I hope you enjoy reading this research.

Jelle Smits

Enschede, May 2022

Table of contents

1. Introduction

1.1. Background

Whereas data storage used to be expensive, this cost has decreased significantly. The adoption of cloud services reduced this cost even further [1]. This trend of reduced cost of data storage enables the production of data by an increasing number of sources and makes the analysis of this data more obtainable. Organizations of all sorts may benefit from using data in their business decisions. An example of this is a retailer which combines data from their cash registers with weather information and can now predict which products are in higher demand during extreme weather [1].

Apart from enabling data-driven decision-making, organizations can also benefit from improved workflows by connecting their applications. Creating these integrations ensures that different departments within an organization can work together to ensure the overall process goes as smoothly as possible. An example of how such integrations can work, an example of a general wholesaler is given below. In general, their sales process consists of four steps:

1. An order of a customer is received and confirmed
2. The ordered products are collected in the warehouse
3. The picked orders are shipped to the customer
4. An invoice for the order amount is sent

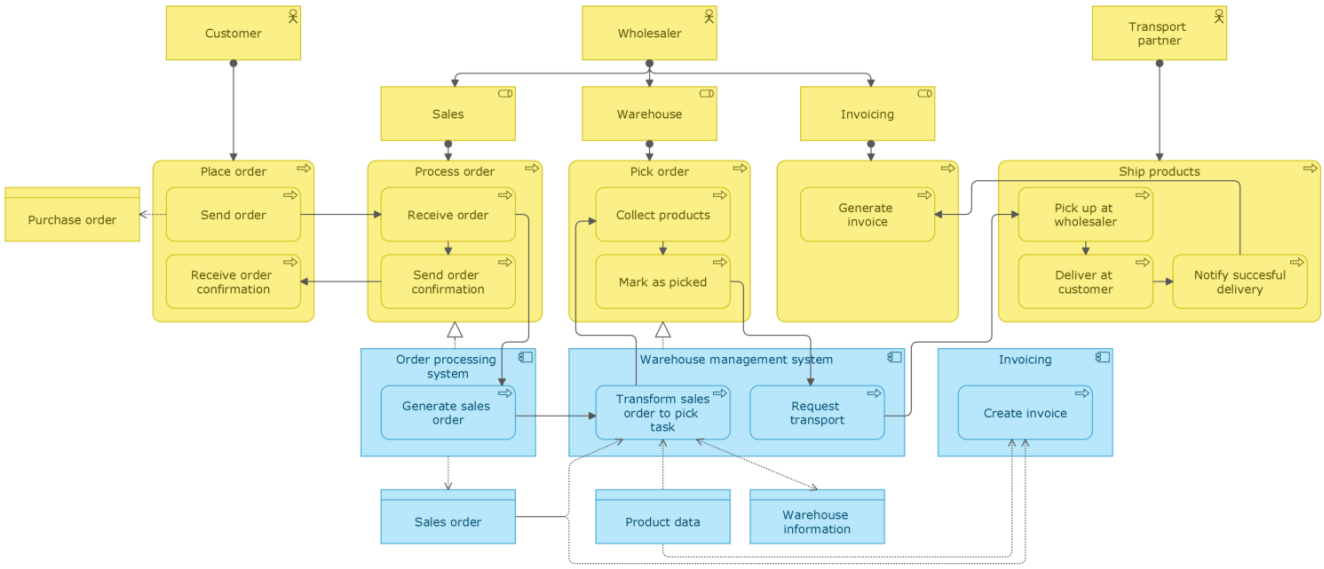

Figure 1: Receiving and processing of an order at a wholesaler in ArchiMate. Yellow elements indicate business elements, blue elements application elements

To illustrate how integrations work, each of the steps mentioned above is addressed. Figure 1 shows an example process in ArchiMate, a modeling language for business processes and their interactions with IT applications and hardware, this process focuses on the steps of the *Wholesaler* and their *Transport partner*.

- After receiving an order from the customer, a *Sales* employee loads the order into their *Order processing system* and sends an order confirmation to the customer. When the order is loaded into the system, the system makes a digital object which contains the most important information: product number and desired quantity for each product, delivery date, and delivery address.

- When the *order processing system* notices it is time to start collecting and shipping the products of the *order*, it sends a message to the *Warehouse management system* with a copy of the sales order. For the warehouse staff to be able to collect the products, just product numbers and quantities are not enough information. To create a viable *picking task* for an employee, the *warehouse management system* needs more information. It obtains this from the internal datasets of *Product data* and *Warehouse information*. From the *Product data*, the *warehouse management system* can enrich the *sales order* with a product description and the number of units in a carton, so that the employees can collect a certain number of boxes rather than individual products. Finally, from the *Warehouse information*, the stock locations of each product are added, so that the employee knows where he can go to collect the products.
- As soon as the employee collected all products, they can mark their task as *done*. This triggers the *warehouse management system* to send out a transport request to the transport partner, which picks up the shipment from the wholesaler and delivers it to the customer. As soon as it is delivered, a *delivery notification* is sent to the wholesaler, which triggers the invoicing department to create an invoice. For this, the system once again uses the initial sales order, in combination with product data to know the unit prices of each product.

Note that this simplified example does not take work processes into account which may make the overall process more complex. Examples of such processes, which are likely to occur in reality, are special agreements regarding discounts for certain products for a certain customer or shipment cost agreements per customer.

Just going through this example of how an order is received and processed by a wholesaler, illustrates the numerous times a data object is accessed or altered to suit the objective of the application. It can be imagined that the process gets more complex when there are more departments involved, or if the customer and transport partner would like to integrate their systems with the wholesaler to improve their workflows. In this example, only three applications are used. In practice, the number of applications used by an organization can range from about a handful to several hundreds of applications, depending on multiple factors such as company size, dependency on data and presence of legacy systems. To make sure all these applications interact with each other in the way they are expected to, integrations between these applications are needed.

Apart from facilitating business processes, as illustrated in the example above, data can also be used to support business decisions. This is often done through dashboards, which combine data from multiple sources. For larger organizations with large amounts of applications, it can be imagined that it is difficult to retain an overview of which system originally produced the data and where this data has been altered throughout its lifecycle. As a result, a third of managers do not fully trust the data they use to make a decision [1]. To improve trust in data, the origins of data should be known. In addition, even though an increased number of organizations consider themselves to be data-driven, it is found that organizations do the majority of the data they have available [2]. This provides a strong need for software that can help in finding data sources and how data is used throughout the organization. Such software can also help combine fragmented data sources and integrations into a single overview. This, in turn, can reduce redundancy and system complexity [3] since there is a clear overview of what already exists. A reduction of redundancies, as well as this decreased system complexity, can help teams who need to work with data do their work more efficiently, since they have to spend less time on finding data, and do not need to spend time creating an insight or integration which already exists.

One such software product which creates insights into the data portfolio[1] of an organization is an enterprise data catalog [4]. These data catalogs provide extensive features to collect and visualize the data portfolio of a company, which helps the user in finding the data they need and ensuring that this data is reliable. In this research, the features a data catalog can offer are analyzed extensively.

Whereas a data catalog is a solution for unifying the data portfolio of an organization by making data sources visible, searchable, and accessible, being able to use the data requires the different sources of data to communicate with each other. These sources can be of various natures, which might make integration difficult. It is no exception that multiple sources of data have to be combined. Examples of such combinations of data were previously illustrated by the retailer who combined their sales data with weather information or the wholesaler who needs multiple applications to work together. When data needs to be processed by multiple applications, sharing the data between these applications is generally done through developing an integration. This way, application A can access application B and vice versa. Since connecting an application needs to be tailored to both application A as well as application B, these integrations are traditionally built point-to-point. This means that each integration is built specifically tailored to both systems. In the case of connecting two applications, one integration is needed. However, as the number of applications grows, the number of integrations that are needed grows exponentially. Adding one more application, making the total three, means that two more integrations are needed. Another two applications, bringing the total number of applications to five, further increases the total number of integrations to ten. Now that there are numerous integrations in place, an application update might change how connections to their applications are made. Now, each integration to this system needs to be changed. This illustrates that it is not feasible to keep adding integrations, as this would significantly impact the maintainability and cost of adding a new application.

A solution for decreasing the number of needed integrations is the usage of an integration platform. Such a platform would need only one integration to each application. This means that in order to develop integrations between *n* applications, the number would be much closer to *n* rather than a multitude of *n* as illustrated previously. Therefore, there are significantly fewer integrations that need to be maintained. In addition, the cost of adding a new application is lower since it only needs to be connected to the integration platform rather than to every other application in the organization. The difference between creating point-to-point integrations compared to using an integration platform is illustrated in Figure 2.

[1] A data portfolio can be described as all data which is produced and consumed within an organization

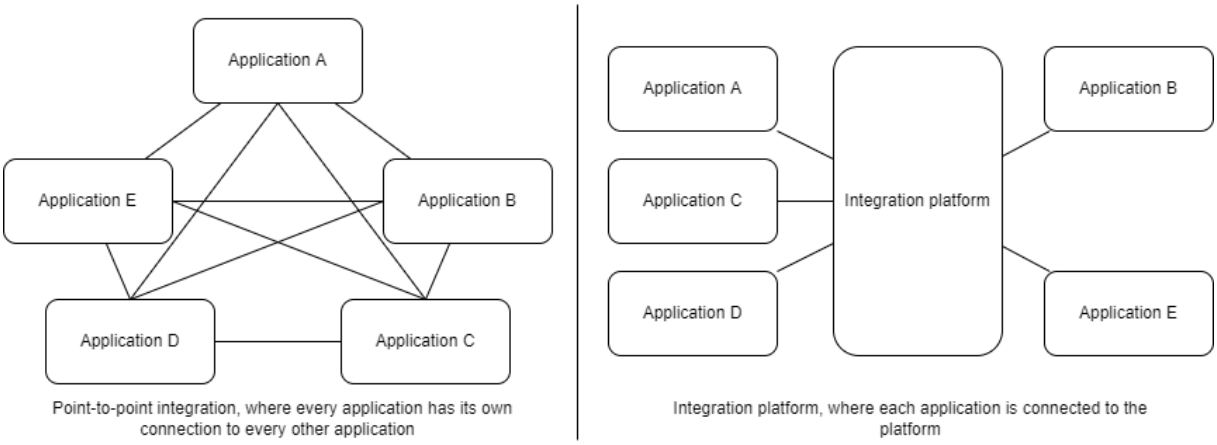

Figure 2: A visual representation of the difference between point-to-point integration and integration using an integration platform

1.2. Objective

This research is focused on the application of features of an enterprise data catalog into an integration platform. In preparatory research, which is included in this research, a gap was identified on the topic of the enterprise data catalog. More specifically, on the application within an integration platform. This research aims to identify whether the features of an enterprise data catalog, which are focused on providing context to data for all (business) users within an organization, are also applicable for an integration platform. For the integration platform, this research focuses on an integration Platform as a Service (iPaaS). Very briefly, an iPaaS is an integration platform that provides all systems needed to facilitate integrations, offered through a cloud infrastructure. A more extensive explanation of what an iPaaS is and does is evaluated in section 2.2. The enterprise data catalog is also evaluated in more detail later in section 2.1.2.2. Both technologies are seeing increased usage with the transition to cloud usage by organizations. This is not reflected, however, in academic research, since no research addressing a combination of these features was found. It is hypothesized that the features of a data catalog can be a valuable addition to an iPaaS environment. Although an iPaaS does already have an inventory of systems, integrations, and data objects which are processed, it lacks core features a data catalog does offer, such as advanced search, data discovery, and data lineage. There is a significant difference in the user focus of iPaaS environments and data catalogs. Enterprise data catalogs are most often rolled out organization-wide, whereas building integrations using an iPaaS platform is focused on certain teams within an organization. Yet there is still overlap in the objective of an iPaaS platform and data catalogs, where both aim at increasing the accessibility for business users.

The overall objective of this research is to increase the usability of iPaaS platforms for the end-user, by developing a framework for adding data catalog functionality into an iPaaS environment, and by identifying the differences in objectives and features of enterprise data catalogs and iPaaS platforms.

1.3. Structure

With the introduction of the research topic already given, the research questions are introduced next, including the methodology for the overall research. After this, the research problem is evaluated so that there is a clear understanding of which improvements are needed, and what the two technologies of enterprise data catalogs and iPaaS can offer to an organization. After this, a possible solution to the identified problem is designed, which is validated through the development of a prototype of the design. To ensure that this prototype offers a solution to the problem, it is validated using the findings of experts on iPaaS. Based on their findings, the fit of features from

enterprise data catalogs into an iPaaS can be assessed in the conclusion, and a discussion is created outlining the possible improvements of the research. Since this report presents academic research, it provides an overview of implications for practice as well as research at the end of this report.

1.4. Research Questions

This research follows the principles of design science methodology, as described by Wieringa [5]. Based on the objective described above, there is a design problem to be answered. Wieringa [5] proposes formulating technical research problems in the style of *How to <(re)design an artifact> that satisfies <requirements> so that <stakeholder goals can be achieved> in <problem context>?*

Applying this format to the research objective described in the previous section, the main research question of this research can be defined as follows:

How to design a solution for improving the usability of iPaaS platforms by adding features of enterprise data catalogs into these iPaaS platforms that enables an improved workflow for its users?

To answer this main research question, several sub-questions have been formulated.

1. **Is it useful to extend an iPaaS with functionalities of a data catalog and why?**
   Many iPaaS platforms use a data model which already gives an overview of the data assets within the platform. In contrast to data catalogs, which discover the data automatically, the available data within iPaaS platforms is modeled. Since all data has to be modeled, it can be seen as an overview of all available data and can be used by users of the platform to identify the data they need. This question focuses on identifying the added value of data catalog features versus the single source of truth the central data model is currently used as.
   - ***1.1. Which overlap already exists between features of a data catalog and an iPaaS?***
     Through legal demands, such as the GDPR[2], or customer requirements, an iPaaS platform might already have parts of the identified data catalog features. This question aims at identifying which features are already common in iPaaS platforms, and possibly only need some extension. This question also helps in answering questions one and three.
2. **What is a suitable position within an enterprise architecture for a tool to create an overview of fragmented data sources?**
   For various reasons, larger organizations can split their integration landscape into different sub-models. Whereas this may decrease the complexity of each individual model, it makes it more difficult to compare which data is used in which instance, especially as the number of models or integrations grows. It is expected that the primary benefit of data catalog features can be obtained for these organizations with multiple models.
   - ***2.1. How can features of enterprise data catalogs be included in an iPaaS?***
     The scenario described in question two focuses on finding the suitable position of a tool providing an overview of all data sources. It is hypothesized that this position would need to be quite central in the overall enterprise architecture. This means that an enterprise data catalog would overlap in position with an iPaaS. Therefore, this question investigates whether these features could also be added to the iPaaS.
3. **Which data catalog features are relevant to add to an iPaaS?**
   It is hypothesized that not all features of a data catalog are relevant to apply within an iPaaS environment because of the difference in the objective of the applications. Whereas a data

[2] General Data Protection Regulation, a regulation imposed by the European Union to harmonize the privacy legislation of all member states

catalog in an enterprise application aims at making the data inventory of an organization accessible to all users, the application within an iPaaS environment helps in providing an overview and thereby increasing the productivity of the users of the platform.

4. **How does the proposed design fulfill stakeholder objectives?**
   The main research problem leads to a prototype, based on the results of the first three research questions. This prototype is evaluated with existing customers of the iPaaS tool used within the prototype, to ensure that it fulfills the goals of the stakeholders identified in later parts of this research.

1.5. Methodology

This research is conducted following the design cycle of the Design Science Research Methodology (DSRM) [5]. This methodology is a proven way of developing a framework with an accompanying prototype. Its formal processes help in ensuring that a valid result comes from the research. The following steps are part of this design cycle:

- **Problem investigation**: The first step in this research is to investigate the problem, to provide a full picture of what the problem consists of. This step helps in identifying the stakeholders and the design objectives.
- **Treatment design**: In this phase, the detailed requirements are identified and the artifact, fulfilling the goals described in the research problem, is designed. This step concludes with a prototype, proposing an artifact.
- **Treatment validation**: this step aims at validating the prototype delivered in the previous phase. This phase shows that the prototype fulfills the requirements, based on expert interviews.

In Table 1, an overview is shown of the research questions that are answered in each of the phases of the design cycle, and which methodologies are used in order to answer these questions.

Table 1: Overview of methods used and research questions answered in each of the phases of this research

Phase	Method used	Research questions answered	Deliverable
Problem investigation	Systematic literature review, Semi-structured interviews	1.1, 1, 2	Interview results and literature, included in chapter 2
Treatment design	Semi-structured interviews, TOGAF	3, 2.1	ArchiMate diagrams, solution design (internal)
Treatment validation	Single case mechanism experiments, Expert opinion	4	Prototype, Validation results

2. Problem investigation

Wieringa describes that the first step before attempting to create a solution to a problem is to fully understand the problem and its context [5]. This chapter focuses on providing an understanding of the problem. This is done by first analyzing enterprise data catalogs and identifying their main features. Afterward, an interview is conducted with stakeholders to get better insights into the problems they experience. The chapter concludes with a summary of the problem, which provides the full context before starting the design of the artifact in the next chapter.

2.1. Literature review: Data catalogs

2.1.1. Methodology

The literature review aims to provide extensive knowledge of what a data catalog is and what its main features are. To this extent, this research investigates the state of the art in data catalogs and the current offering of the commercial vendors in the market of data catalogs.

The topics which are needed in order to provide a proper background into data catalogs are primarily related to its features, as well as ways to show or hide sensitive data as these are important in finding if it is relevant to apply a data catalog within a domain-specific environment. Therefore, this literature review answers the following research questions:

1. What information is shown in a data catalog and how is this data collected?
2. What are the most important features of a data catalog?
    a. Which features does a data catalog have according to the literature?
    b. Which features are offered in data catalog products of current commercial vendors?
3. What prevents unauthorized access to sensitive data in a data catalog?
4. What problems does a data catalog solve?

For question 2b, the literature study is not expected to give these results as a representation of the current market. In order to obtain the answer to this question, the data catalog products of large commercial vendors are evaluated. This is done through their company website and documentation of their product.

2.1.1.1. Literature review methodology

The literature research is done through the methodology of a systematic literature review [6]. This literature review uses Scopus in order to obtain high-value literature. Technology is developing quickly, and the amount of data created and consumed worldwide is following an exponential pattern [7]. For this reason, research which has been conducted a long time ago is likely to have less relevant results for this literature review, since with the change in data volume the demands for data cataloging have changed as well. For determining the relevant time period, the number of results from Scopus as well as Google Scholar has been plotted into a graph displaying the results of querying these databases for 'Data Catalog' per year of publication, which is shown in Figure 3. This shows that from 2005 onwards, a tipping point has been reached, where the number of articles published each year increased significantly. This sets our first exclusion criteria to *the article must be published in 2005 or later*.

Because of differences in the way the Scopus search engine works compared to Google Scholars', two different queries have been used to get an overview of the number of articles on this topic over the years. The main difference is within the way differences in spelling, as *catalog* is mostly used in American English, but *catalogue* is used in British English. This research uses *catalog* consistently since this term is used mostly in literature, and by commercial vendors. Within Scopus, this difference in spelling can simply be included by using the * to stem the word, which makes it include

both *catalog*, *catalogue*, *catalogs*, and other word variants of the *catalog* word stem. Within Google Scholar, the * is used as a wildcard for full words rather than stemming a word, therefore, the other spelling terms have been manually included.

A second difference in the query of Google Scholar is the '-*VizieR*' operation at the end of the query. This has been added since there is significant noise in the number of results from research about the VizieR data catalog. This data catalog is not relevant to the objective of this research, since it regards a catalog of astronomical data, which is not within the scope of this research. Therefore, this term is excluded which reduces the Google Scholar results from 13.900 to 264, yielding a comparable number of results for both Scopus and Google Scholar. The used queries are:

1. Scopus: TITLE-ABS-KEY ("data catalog*") AND (LIMIT-TO (SUBJAREA, "COMP"))
2. Google Scholar: allintitle: "data catalog" OR "data catalogue" OR "data cataloguing" OR "data cataloging" -VizieR

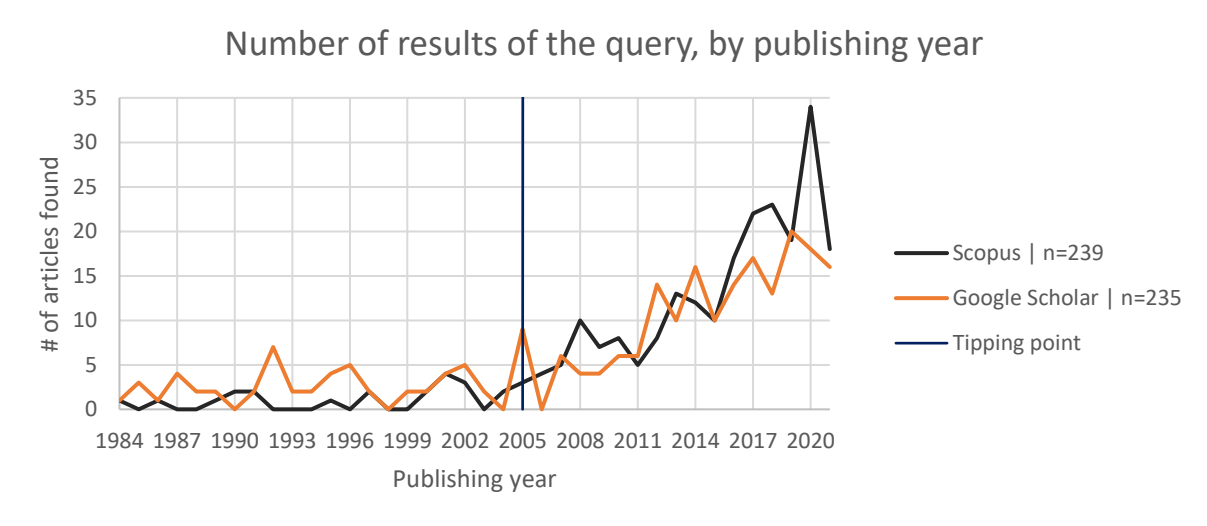

Figure 3: Number of results of the query in the different databases

As the initial results provide too many results to analyze in detail, exclusion criteria are set to narrow down the number of results and increase the relevancy of the results found. The following exclusion criteria are set:

1. The article is written in English: relevant literature is expected to be in English, as this language is the global standard for research in the field of computer science.
2. The article is written in the domain of computer science: the term *data catalogs* has a wide representation in other applications, such as libraries and biological studies, which hold no relevance for this research.
3. The article is published in 2005 or later: as indicated above and shown in Figure 3, 2005 can be identified as the tipping point where an increased academic interest in the topic was identified based on the number of articles that are published.
4. The article is published in a scientific journal, magazine, or conference proceedings: this research aims to derive relevant solutions through the literature study. This is only possible when high-value research is used as input.
5. The article is accessible: full text of the articles is needed to use them in this research. Accessible can be through open access, or through the University of Twente repository.
6. The article has data catalogs as its main topic: the main topic of this research is that of data catalogs. There is abundant literature available with some mention of a data catalog, but no

significant contribution on this topic, making them not suitable to be used. This is assessed based on title and abstract and can also be decided based on full text in a later stage.

7. The article is not about a domain-specific application: similar to the widespread use of data catalogs outside of the domain of computer science, there is a large part of the articles which only apply to a specific application of data catalogs, such as a data catalog application within smart home environments. These are not relevant for the broader view of data catalogs and are often not relevant in enterprise settings.

Before applying any of the selection criteria, the search query of "Data catalog*" in Scopus yields 583 results in October 2021. In Figure 4, the results of applying the exclusion rules upon the initial query are shown. Each exclusion rule is abbreviated as 'Ex', followed by the number of the rule listed above. Behind this, the total number of results in the remaining result set, after applying the exception rule is shown, and below this, is the decrease in results this exclusion rule caused.

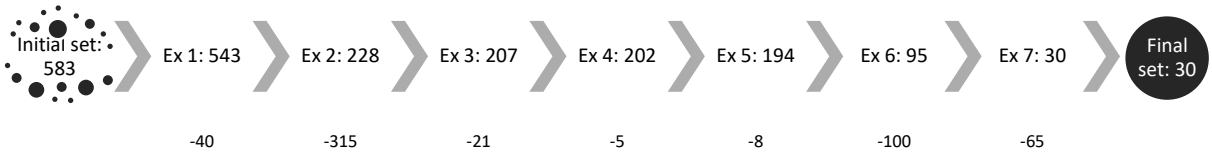

Figure 4: Results of applying the exclusion rules to the found data catalog literature

2.1.1.2. Market analysis methodology

As indicated previously, a market analysis is needed to answer research question 2b. This analysis is done in a structured way with regard to the selection of vendors and the mapping of the features. For the readability of this research, the methodology of the market analysis is introduced in section 2.1.2.3, where the results are also immediately given.

2.1.2. Results

In this section, the findings from the literature are evaluated to answer the research questions introduced earlier. The results of the literature are structured to start by outlining what a data catalog is and how it works. This outline starts by introducing the basis of a data catalog: metadata. After this, the concept of a data catalog is explained in more detail. In this part, the feature overview of commercial data catalogs for enterprise settings is introduced, after which they are compared to open data catalogs. These open data catalogs are extensively covered in the academic literature, and therefore also included in this research. In the final section, the differences identified in the features of an open data catalog and an internal enterprise data catalog are evaluated.

2.1.2.1. Metadata

According to the ISO 11179 standard, metadata is data that defines and describes other data. Every user of a computer uses metadata: saving a document in a folder and giving it a certain name means actively assigning metadata attributes such as *file name* and *path* to the file. This metadata can later be used to trace back the file when it is needed again. There are different kinds of metadata families [8], [9]:

- **descriptive metadata**: aimed at improving findability and understanding of the contents
- **administrative metadata**: can be split into multiple subcategories such as technical, preservation, and authorizations. This metadata gives context regarding technical contexts such as filetype and decoding, sensitivity and accessibility attributes

- **structural metadata**: data system relationship context, data linkage, and relationship context, and business context

For example, *duration* and *author* are descriptive metadata attributes of an audio file, and *height*, *width*, and *camera model* are descriptive metadata of an image file. Examples of administrative metadata could be the *file type*, *file size*, and *path*. Structural metadata aims at improving navigation. An example of this type would be the chapters within a text file.

Although the examples above, on an individual file scale, are most familiar to consumers, businesses often use datasets rather than individual files. Datasets are essentially files containing data in some structured format. Datasets can have many data entries and a data environment can in turn consist of many datasets. Because looking for a dataset should be a quick action, information is needed from a dataset regarding its contents. This information can be provided in the form of metadata. Take for example Table 2, which shows a part of an unknown dataset. Without further information, it is difficult to identify what this dataset describes. It could be the primary external contact persons of clients, but it might also be a dataset of suppliers and the internal account manager. In this case, some context, such as a better name of the table, or a summary of its contents could have the data consumer help in knowing if this is the data they are looking for. For example, if it was known that this dataset is created by the software of the sales department, a better direction would be available regarding whom to ask for details on what this dataset is. An even more useful metadata field, in this case, could be a description such as *primary contacts per client*. The more metadata is available, the better can be identified if this is indeed the data that is needed.

Table 2: An extract of a fictional unidentified dataset called 'company-contact'

id	person_name	company_name
324	John Doe	Corporation A
533	Jane Doe	B Inc

It has been shown that it is not feasible to manually look through an entire dataset to identify what it is about. It is relevant to note that data production and consumption are large, and increasing [7]. As searching for data must be a quick action, ideally as quick as using a search engine, searching on document contents, rather than its metadata is not feasible. Within data-intensive enterprises, even searching for metadata may even pose problems. For example, the data catalog of Google indexes 26 billion datasets, and this number only includes datasets that are accessible by all Google employees [10]. On this scale, even gathering metadata of all datasets, let alone looking into the data itself, takes too much time and requires special approaches [10].

2.1.2.2. Data catalogs

Companies and enterprises are harboring increasingly more data, either from their own production processes or through their software. With the growing number of sources of data, the variety of the data also increases. Companies want to be able to use this data in many different ways. One example of using data to support business decisions is through dashboards, for which they generally use business intelligence software. This is confirmed by the continuous growth in the revenue generated by business intelligence software and by the increase in the amount companies spent on software per employee [11]. As also stated in the introduction, it gets more difficult to find a specific resource when multiple teams create resources, and there are large numbers of data sets. This is confirmed by a market analysis by Seagate in 2020, where they found that up to 44% of data that is available for organizations is not captured, and an additional 43% of the data which is captured is barely used [2]. Furthermore, one of the main challenges experienced in leveraging the collected data is ensuring

that the correct data is collected [2]. In order to create a better overview and increased insights into the datasets, as well as any data integrations and interactions, a data catalog can be used. Within academic literature, a singular definition of a data catalog is not given. Within the commercial market, the definition Gartner gives seems to be the standard many data cataloging software packages adhere to. The interest in data cataloging tools is also increasing, based on the increase in the number of inquiries Gartner received over the last years [12]. Gartner defines a data catalog as:

"*A data catalog maintains an inventory of data assets through the discovery, description, and organization of datasets. The catalog provides context to enable data analysts, data scientists, data stewards, and other data consumers to find and understand a relevant dataset for the purpose of extracting business value.* [12]"

Based on this definition, general categories of data catalog features can be derived based on the definitions' keywords *discovery*, *description,* and *organization*. These would correspond to, at minimum, features such as *search, metadata management*, and *tagging*, respectively.

A data catalog can generate short- and long-term benefits for the enterprise. Some of these benefits include [12]:

- Regulatory compliance as the catalog provides context to data and ensures its traceability
- A live data asset inventory
- Monitoring, auditing, and traceability supporting governance
- Providing context to data in the organization
- Clarifying accountable persons for data

Data catalogs are often implemented on an enterprise-wide scale to increase transparency and data usage as well as to facilitate better access to data [4]. Compliance and risk management are also motivations, although less often mentioned [4]. One of the ways in which better access is realized is by offering an advanced search function. In a survey of 11 large enterprises who were in the progress of adopting a data catalog, they all confirmed that search functionality is a must-have feature of a data catalog [4]. Apart from the search functionality, other features identified as the most important features of a data catalog include data registration, metadata management, a business glossary, role management, tagging, and sharing [4]. This overlaps with the initial categories of features that were expected based on the Gartner definition. It is important to note that the conclusions of the research of [4] are based on the input of only 11 enterprises. Although this number is limited, these enterprises were of different industries, and the results were not based on a one-time survey but concluded from an extensive analysis over the span of 11 months. It focused on the implementation and selection process of a suitable data catalog.

Since data catalogs rely on metadata, it is also dependent on the quality of the metadata. Multiple studies have focused on how metadata quality can be assessed [13]–[17], and emphasize that a lack of metadata quality can result in decreased searchability and accessibility [13]. Since the metadata proves to be such a key element of the usability of the data catalog, all data cataloging tools are expected to have some kind of metadata management module in their software, as confirmed by [8].

Within data catalogs, different groups of users can be identified [18]. These consist of *data providers*, who ensure that the data is put into the system, taking into account applicable (internal) regulations, and applying the proper structure. The second group are the *data custodians*. These users maintain the data and ensure their quality. Finally, there are the *data consumers*. This is the biggest group that can be further divided into roles [18]. As this research does not discuss the implementation of the

data catalog at an enterprise in such a level of detail at this point, the role division of the categories of enterprise users is not addressed.

As a response to business needs, numerous commercial software providers offer data catalog software. In the following section, the features offered by these commercial software products are listed and compared to the previously given definition. Many of the commercial vendors use the Gartner definition above on their webpages, although the interpretation of the definition might differ between vendors.

2.1.2.2.1. Data catalog features

As previously mentioned, there are already a significant number of commercial data catalog software providers. In this chapter, an overview of their features is given. First, a description of the features is given, followed by a side-by-side comparison of the data catalog features. This gives insight into the limitations of current offerings as well as the opportunity to compare the definition of a data catalog as given previously to the features the industry translated this into.

Gartner provides a platform on which enterprise software users can review this software [19]. Based on this, they have a list of features they judge data catalogs upon. Additional research by Gartner also outlines some core features of data catalogs. Overlapping the features identified by Gartner in [12], [19] with features identified in the academic literature by [4], [8], [20] combined with other features found to be prominently represented in data cataloging software generates the data catalog feature overview as shown in Table 3. At this point, no assumptions are made about their importance to the data catalog. Therefore, the features in the table are sorted alphabetically.

Table 3: Features of a data catalog, sorted alphabetically

Feature	Description	Source
Access management	Allows configuration of user roles and enables request-based access to data. Helps with governance	[4], [20]
Business glossary	Provides an overview of business terms and how they are used in the specific business context. Used to provide context to data, and mitigate risk caused by differences in vocabulary interpretation. Can include a description of the meaning of metadata.	[4], [8], [12], [19], [20]
Collaboration	Offers functionality that enables the users to communicate, comment, tag, and share data	[4], [8], [12]
Connectors	Offers out-of-the-box options to connect with industry defaults, such as (No)SQL servers and large software applications. Ensures that the data search and data discovery can work	[4], [19]
Data lineage	Traces back data to its sources. Shows dependencies and helps assess data quality by identifying origin data. Also provides an overview of changes made to the data	[4], [12], [19], [20]
Data quality profiling	Supports measurement of user-defined metrics to profile, monitor, and improve data quality	[4], [8], [20]
Data search & discovery	Enables automated discovery of new data sources, and provides a search option to find relevant data	[4], [12], [19]
Governance	Utilizes the overview offered by the catalog to aid the enterprise in monitoring and enforcing compliance with legislation and regulations. Also aids during audits of regulations. Can also be named *rule management*	[4], [8], [12], [19], [20]
Machine Learning (ML)[3]	Overlaps all other features, offering means to automate tasks	[4]

[3] Although ML is only mentioned by one source as a useful feature of a data catalog, development in ML is fast and the topic is intensively being researched in both academic as well as non-academic settings, and is therefore included in the list of features

Impact analysis[4]	Provides information on the effects of changing a data source, to other data sources which depend on it	[19]
Metadata management	Documents and manages metadata as the core of the system. Enables browsing of metadata. Includes means for interacting and harvesting metadata	[4], [8], [12], [19], [20]

Figure 5: Gartner's Magic Quadrant for Metadata Management Solutions [19]

2.1.2.3. Market analysis: Comparing feature list to features of commercial vendors

Based on the features of a data catalog as identified in Table 3, the offering of commercial products is compared against these products. In order to select vendors whose data catalog products to analyze, the Gartner Magic Quadrant on the topic of metadata management solutions is used. Although these companies pay to be listed in the report, they provide a good representation of the market offerings, based on the criteria Gartner set in order for a vendor to be included in the report. These criteria consist of, among others: revenue, the continuous growth of product usage, usage of the product in multiple global regions, customers in multiple sectors, and the need to have none of the core capabilities of their product outsourced to other vendors. Especially the criteria of revenue, growth of the product, and customers in multiple sections ensures that the selected vendors for the analysis are key vendors who have mature products. It is important to note that this also means that some features are automatically present in all vendors' products since these are a requirement to be in Gartner's Magic Quadrant.

The Magic Quadrant of November 2020 shown in Figure 5 is divided into four named parts:

- *Niche players*: provide a small focus to support specific use cases
- *Visionaries*: have a strong understanding of upcoming technology and trends and have tailored their offering to this expected demand. Are not yet competitive beyond their standard user base

[4] Although impact analysis is only mentioned by one source, it is a *must-have* feature of Gartner, meaning that it must be present in each of the analyzed vendors' products

- *Challengers*: have the key trends within their offering, but are limited by the support and functionality for a wide range of use cases
- *Leaders*: provide offerings supporting all capabilities and have a clear understanding of where the market is headed, tailoring their offering to these future demands

Within the division, the companies in the *leaders* segment are most relevant to look into, since they offer the most complete offering. This brings the list of companies to be investigated for their feature offerings to Alation, Alex Solutions, ASG, Collibra, Erwin, IBM, Informatica, Oracle, SAP, and Smartlogic. It is important to note that the analysis was done based on the widest range of features the vendor offers with their product, not taking into account possible subscription tiers which might exclude some of the features.

This research does not provide judgment of any kind on the quality or performance of the selected vendors' products. Vendors have been evaluated based on their websites and the documentation of their products.

During the analysis, it is found that there are certain categories to which all selected vendors conform. These include the features listed in Table 3 which were confirmed by [19], as all vendors had to abide by a number of core features in order to be considered to be included. However, other features have also been found, which were not mentioned in [19], which all selected vendors have. Apart from these common features, also often included features can be found, and additional features, which are included in only some of the vendors' products but do add value to be within a data catalog. Therefore, the feature list of Table 3 can be extended, and divided into three categories. Note that the names of the feature groups are not fixed in time. Features that are identified as advanced or unique to just some vendors might become standard as technology progresses. The naming is relevant at the time period of this research.

- **Core features:** these features are represented in all of the analyzed vendors' data catalog products, and therefore considered to be a must-have feature for any enterprise data catalog
- **Advanced features**: these features are present in the data catalogs of most of the analyzed vendors, and provide immediate and broad benefits to the enterprise as well as being relevant for a data catalog
- **Unique features**: offered only in some of the vendors' products. Consists of specialistic add-ons, which are relevant to be included in a data catalog product, and cannot be grouped as an extension of an already identified *core* or *advanced* feature.

It is important to note that the *advanced* and *unique* feature categories do not include all features of all products. Features in this category can be part of one of the core features but are considered a unique selling point, rather than a way of describing features within one of the main features already described. For example, a vendor can advertise with a feature such as *Stewardship* within their governance. This can also be seen as the core of governance since the administrator has to configure which roles can access which dashboards and sources. Similarly, *Artificial Intelligence (AI)* and *Machine Learning* are grouped together. Despite their differences, the benefits they offer to a platform are similar.

Table 4: All data catalog features, grouped into classifications, features listed alphabetically.

Classification	Feature
Core	Business glossary
	Connectors
	Data lineage
	Data search & discovery

	Governance
	Impact analysis
	Metadata management
Advanced	Collaboration
	Data quality profiling
Unique	Access management
	Machine Learning (ML)
	Tribal knowledge sharing

In Table 4, all found features are grouped into the feature classifications introduced previously. There is one new feature that has not been defined in Table 3 before. It can be defined as:

- **Tribal knowledge sharing**: offers the possibility to document *tacit* knowledge, knowledge which cannot be obtained directly through data, but resides within employees based on their experience. Tribal knowledge sharing aims at providing the means to store this tacit knowledge so that more people in the organization can benefit from it, and it is not lost when an employee leaves the organization.

Although Table 3 mentions that access management and connectors are part of another feature (governance and data search & discovery, respectively), they are included as a separate feature. This is because of the differences in occurrence within the data catalog tools for access management, which cannot be seen as a core feature at this point. Although connectors are a must-have feature

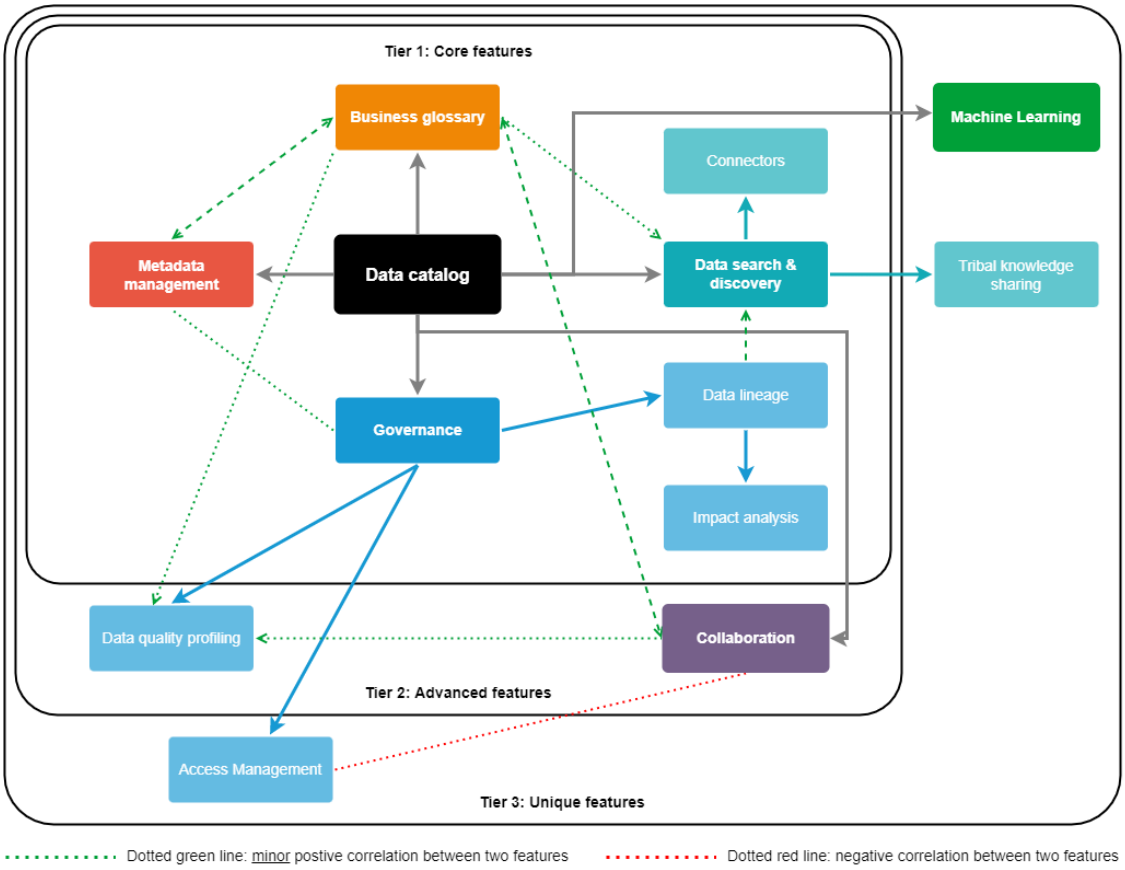

Figure 6: An overview of the features of an enterprise data catalog and the correlations between features

for a data catalog in order to be able to conduct searching and data discovery, it is important to mention this feature separately to have a complete view of the features of a data catalog.

To clarify the relatedness of the identified features, Figure 6 shows each of the features as outlined in Table 4 and their position in the hierarchy. This clarifies that, for example, data lineage is part of governance features, and that the impact analysis is an extension of the data lineage. The division into *core*, *advanced* and *unique* is also shown using the three tier boxes. Each independent feature group has a different color. This shows that some features, such as the connectors, are a supporting feature for data search and discovery, although they are of such significance important that they need to be mentioned separately.

The green lines indicate a supporting relationship between the two features. The level of coarseness of the line indicates how strong the relationship is: a finer line means a weaker relationship, whereas a coarse line indicates a strong relationship. For example, a business glossary strongly depends on metadata, as it can provide meaning to metadata. Data quality profiling, on the other hand, provides some support for collaboration, as a higher data quality ensures that other users can interact with decreased need of validating the data, smoothening the overall process.

Machine Learning, which also includes Artificial Intelligence, is marked as a green box since it can provide support to any of the other features. It could, for example, be used to provide automatic recognition and enrichment of metadata, recognize definitions in documents to add to the glossary or help discover new data sources.

Using the final list of features, the offerings of each of the vendor's products can be compared to the identified features. The result of this is shown in Table 5. For readability, the core features are not listed one-by-one but are grouped since each of the data catalogs, by definition, offers all of the core features. In addition, vendors are sorted alphabetically.

Table 5: Feature overview of the ten 'leader' vendors' data catalog solutions, features, and vendors sorted alphabetically. A checkmark means that a vendor offers the feature or feature group

Vendor	**Alation**	**Alex Solutions**	**ASG Technologies**	**Collibra**	**erwin**
Product name	Alation Data Catalog	*Multiple products*	ASG Data Intelligence	Collibra Data Catalog	erwin Data Intelligence
		Features			
Core (grouped)	✓	✓	✓	✓	✓
		Advanced			
Collaboration	✓	✓		✓	
Data quality profiling	✓	✓			✓
		Unique			
Access management	✓			✓	
Machine learning (ML)	✓	✓			
Tribal knowledge sharing	✓				

Vendor	**IBM**	**Informatica**	**Oracle**	**SAP**	**Smartlogic**
Product name	IBM Watson Knowledge Catalog	Enterprise Data Catalog	OCI Data Catalog	SAP Data Intelligence Cloud	Semaphore
		Features			
Core (grouped)	✓	✓	✓	✓	✓
		Advanced			
Collaboration	✓	✓	✓		
Data quality profiling	✓	✓		✓	
		Unique			

Access management	
Machine learning (ML)	✔
Tribal knowledge sharing	

2.1.2.4. Open data catalogs

Data catalogs can be applied to data that resides within enterprises but also to data generated by governmental bodies. Since the literature on the topic of the application of data catalogs within enterprise settings is limited, the more extensive literature on government open data catalogs is a partial representation of the problems that may arise when rolling out a data catalog in enterprises. Governments offer significant parts of their data on publicly accessible platforms, a trend that got attention as it was one of the first acts of a new US presidential administration in 2009 [21], [22], followed by the G8 Open Data Charter in 2013[5], and again with the launch of the European Union Open Data Portal in 2015.

Open data can be loosely defined as data which free to be used and redistributed by anyone, and which is machine-readable [15]. Governments are motivated to offer open data by their desire of being transparent [17], as well as legal obligations towards publishing their data. For example, in the EU, the 2003 Public Sector Information (PSI) directive enforces its member states to ensure that published data is re-usable and motivates all member states to publish data [23]. Also, practicing open licensing of the data is strongly encouraged [23]. Other than abiding by regulations, governmental motivations for sharing data also include increasing collaboration, participation, and transparency [22]. By providing data to the public, governments offer a higher degree of transparency. The public can, in turn, participate by using this data. By offering the available open data of a government through a data catalog, governments can significantly improve the discoverability of their datasets [15].

There are different standards that aim at making open data more usable, such as the Findable, Accessible, Interoperable and Reusable (FAIR) standard [24] which was developed for scientific data, and the 5-star open data classification[6], which promotes open formats and interconnectivity for open data. These aim to increase the usefulness of the data, and the way in which the data can be interacted with in an automated fashion. The need for these standards is presented through research of the open data sets offered by US governmental bodies in 2017, which found around 17% of the data sets do not contain enough metadata to determine the format of the data set, and another 56% of datasets to be unstructured or in a proprietary format [22]. Some governments show initiatives to improve their use of standards to increase their data quality [25].

Issues reported in governmental open data catalogs include inconvenient formats, a lack of consistency between data sets, and poor documentation [26]. In addition, the search functionality is argued to only work for people who know what they are searching for and therefore does not work for the majority of the users. In order to bring out the full potential of a data catalog, simply searching the title and description is not enough [27]. This is especially the case when datasets with different original languages, such as is the case in the EU open data portal. The EU wants to support all of its member states by offering the metadata of the datasets on the European Data Portal in all of the EU's 24 official languages since every EU citizen has the right to communicate in any of these 24 languages [28]. The portal currently supports this by translating the keywords of each dataset, either by the author or through automatic translation, into all of the 24 languages.

[5] https://www.gov.uk/government/publications/open-data-charter/g8-open-data-charter-and-technical-annex
[6] https://5stardata.info/en/

Issues that might be caused through interpretation and translation of natural language are illustrated by a study that compares the open data categories offered by the Italian (Lombardy province) and Swiss open data portals. This study intended to obtain the most optimal data for local tourism [29]. Since Switzerland has four official languages, one of them being Italian, there was no need to conduct any translation since the Swiss data was already offered in Italian. An immediate issue that arises with this cross-border cooperation is the difference in categorization. For example, in one language a category might be named *politics* whereas the other one chose *government.* Also, differences in grouping categories can occur, such as *culture and sport* in one data catalog, grouped as *culture, media and sport* in the other catalog. Since these are differences on the tag of `dct:theme`, of the Data Catalog Vocabulary (DCAT) standard, as is addressed further in section 2.1.2.5, it is difficult to link these misaligned themes, even though both of the open data catalogs use a specialization of the European Union DCAT-AP standard: DCAT-AP-IT and DCAT-AP-CH, respectively. Another example of the language problem is the need to combine data from different sources in order to obtain the data needed from, for example, a European perspective rather than a national perspective. Here, issues arise with trying to combine datasets, as the language and terminology differ, and normal translation might not yield the expected results due to limited search flexibility, as well as problems with differences in data attributes which make it hard to combine data [27].

2.1.2.5. Standards: Data Catalog Vocabulary (DCAT)

DCAT is a Resource Description Framework (RDF) vocabulary with the intention of offering interoperability between data sets and data services. Both the DCAT and the RDF standards are standardized and managed by the World Wide Web Consortium (W3C). Within DCAT, a dataset is defined as a "*collection of data, published or curated by a single agent, and available for access or download in one or more serializations or formats*" [30].

DCAT helps in achieving this by describing dataset metadata in order to be able to catalog and search them. It does so by having a standardized set of classes and properties which are mandatory, recommended, and optional. An overview of the vocabulary is shown in Figure 7. Any data catalog which conforms to DCAT would have its data organized into datasets, distributions, and data services, provide an RDF description of the catalog, and all classes and properties of DCAT are used consistently [30]. This ensures that a data catalog can be built which shows at least the most basic attributes of a data set since the mandatory attributes are always visible in data sets that conform to the standard. A DCAT conform catalog would therefore be able to process all DCAT classes and attributes, and may also include non-DCAT fields. An important benefit of applying DCAT is that it enables machine-processable classification means, which can improve upon dataset discovery [31].

DCAT is primarily applied in governmental data catalogs. For example, the European Union has its own DCAT application profile DCAT-AP. All data within the EU open data portal has to conform to this standard. Many countries have further specializations of the European application profile, such as DCAT-AP-DONL for the Netherlands. In the case of the Netherlands, the specialization was created because it wants to offer fewer free metadata fields than possible in DCAT-AP, in order to enable easier checking of metadata quality [32].

In Figure 8, a partial example of an application of the DCAT-AP standard is shown. This example is of a dataset, as stated in the second line, and includes all mandatory attributes (title and description), most of the recommended attributes (keyword, publisher, and distribution) and some optional properties (issued, modified and language). The dates are mentioned as the type `xsd:date`, which clarifies that the date is listed as year-month-day.

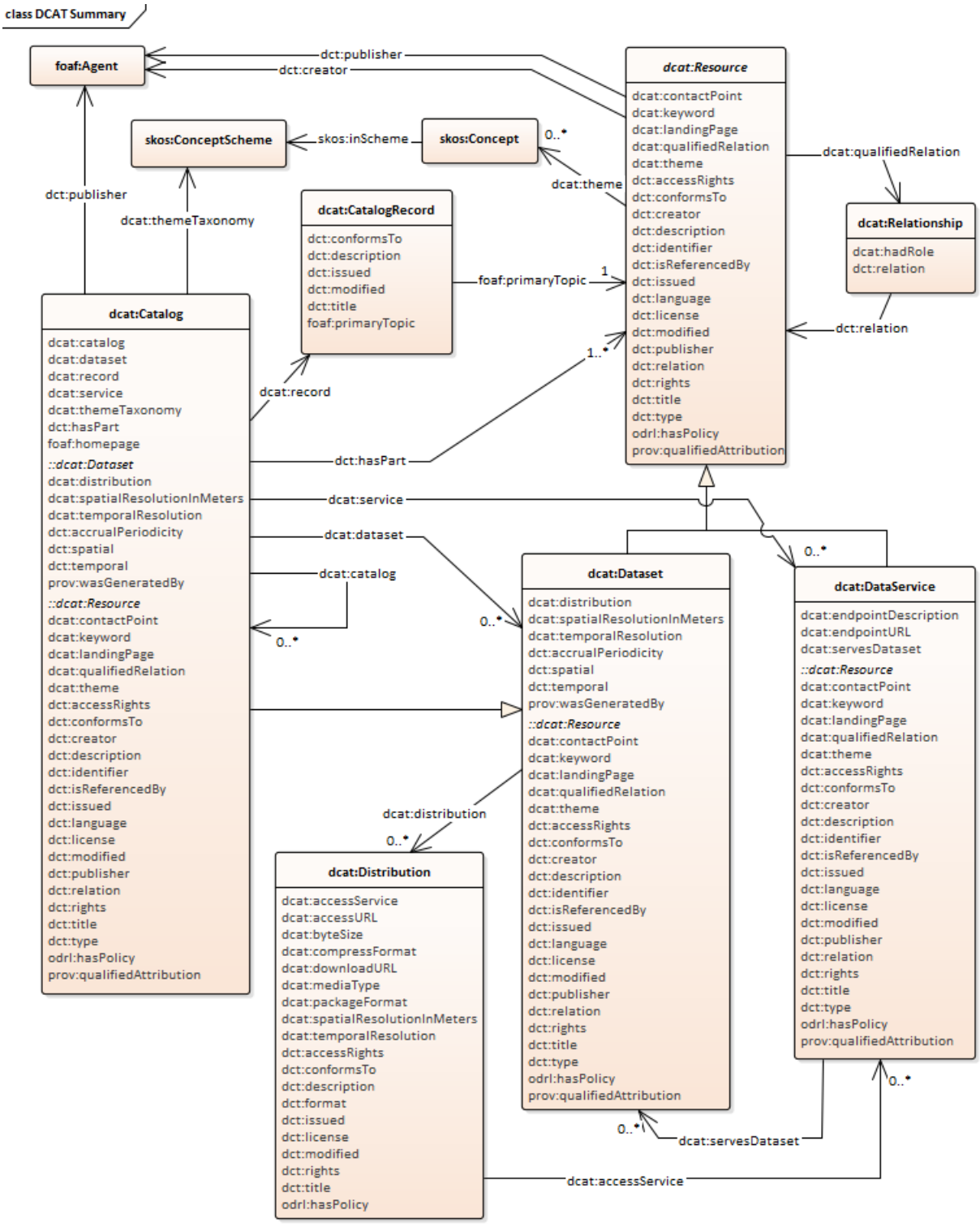

Figure 7: An overview of the DCAT language, used to describe data catalogs uniformly

```
:dataset-example
    a dcat:Dataset ;
    dct:title "Example dataset"@en ;
    dct:description "This is an example dataset for an example"@en ;
    dcat:keyword "example"@en, "dataset"@en ;
    dct:issued "2021-06-21"^^xsd:date ;
    dct:modified "2021-09-29"^^xsd:date ;
    dct:publisher :example-ministry ;
    dct:language <http://publications.europa.eu/resource/authority/language/ENG> ;
    dcat:distribution  :dataset-example.csv .
```

Figure 8: An example of a dataset specification in DCAT-AP

2.1.2.6. Enterprise and open data catalogs: similarities and differences

This section compares the features that were identified as part of an enterprise data catalog to the features in open data catalogs. In Table 6, the data catalog features as identified are shown, as well as the relevance for applying these in open data catalogs. In the following subsections, it is shown why these are relevant or irrelevant to use in an open data catalog.

Table 6: Differences in features desirable in enterprise and open data catalogs. An extension of Table 3.

Classification	Feature	Enterprise relevance	Open data relevance
Core	Business glossary	✔	
	Connectors	✔	
	Data lineage	✔	✓
	Data search & discovery	✔	✔
	Governance	✔	
	Impact analysis	✔	
	Metadata management	✔	✔
Advanced	Collaboration	✔	✔
	Data quality profiling	✔	✔
Unique	Access management	✔	
	Machine Learning (ML)	✔	✔
	Tribal knowledge sharing	✔	
Legend	✔ *relevant*	✓ *somewhat relevant*	*blank: not relevant*

2.1.2.6.1. Differences and similarities on feature-level

In general, there is a fundamental difference in the objective of an open data catalog compared to a data catalog in enterprise settings. Enterprises can have strong motivations to share their data similar to how open data catalogs do. These motivations include monetization of data, marketplaces, industrial data platforms, technical enablers, and open data [33]. Primarily, however, the first intention of building an enterprise data catalog is to use the catalog for internal benefits, which has been addressed at the beginning of this chapter. Based on these internal objectives, the feature comparison of Table 6 was built. This gives an overview of the features which are applicable either in both kinds of data catalogs or only one of them. This section focuses on the differences identified and explains in more detail what the difference or similarity in relevance is, or why a given feature does not have relevancy for open data catalogs.

Business glossary

This feature is not identified in any of the open data catalogs which were analyzed during this research, and it has not been mentioned in the literature on open data catalogs. It would also be significantly more difficult to apply a glossary to an open data catalog since this data consists of datasets from many different disciplines which makes it likely that homonyms exist, which makes it more difficult to agree on a single truth. This is a significant difference from the enterprise data

catalogs, where the business glossary is one of the core features. This difference can be explained by the differences in the objectives of open data catalogs and enterprise data catalogs since enterprise users are expected to use the data for the organization's benefit, whereas there are no limitations on how consumers of open data use it. This means that organizations can have a single definition for a term, whereas the objective of open data is that anybody is free to use it to their liking. This more heterogeneous target audience can result in multiple definitions. A glossary could still have added value for open data catalogs as an extension to the metadata which is normally registered, to provide an improved meaning to the data as it is intended by the producer to avoid ambiguities.

Connectors

Connectors are important to enable automated data discovery, as well as for other features such as access management. Within open data catalogs, the data which is shown often consists of exports of the internal systems of governmental organizations. Whereas this limits how interested parties can interact with the data, and ensure that they have the most recent data, it is understandable that the governmental organizations do not want to open up direct access to their systems because of the security risks involved. Within enterprise settings, this is completely different. Enterprises often have many systems within their organizations that are required to interact with one another, and often also have external partners which whom they need to connect with. Without being able to have an enterprise data catalog connected to data sources, it cannot add other benefits such as data discovery.

Data lineage

Open data catalogs most often contain data from various publishers and in different formats. This makes it more difficult to provide detailed data lineage [34]. A limited version of data lineage is often offered in open data catalogs, showing the publisher of the data or, if standards such as DCAT are used, differences in the distributions can be indicated which helps keep track of the most recent data, and changes over time. Other than differences in publishers, there is also less interactivity between datasets, making data lineage less relevant. Within enterprise data catalogs, the data lineage has been identified as a core feature as it is important to ensure the quality of data, as business decisions are made based on the data. In addition, as also illustrated in the example in the introduction, enterprise applications are often built in a way in which they are dependent on each other. This means that data lineage is not only relevant for knowing from which application data originates and where edits have been made but also in ensuring governance.

Data search & discovery

Enterprise data catalogs and open data catalogs are fundamentally different in how they collect their data. Where enterprise data catalogs are expected to have some automated discovery of data sources that are to be shown in the catalog [4], open data catalogs of governments contain datasets that are most often uploaded to the catalog directly. This makes the focus of an open data catalog more on the presentation of the datasets rather than the collection. For both kinds of catalogs, it is important to have good search functionality. Since datasets of both enterprises and governments can get extremely large - Google has tens of billions of datasets [10], and the European Data Portal[7] catalogs 1,1 million datasets as of October 2021 – a catalog would not be usable if no proper search tools would be available. In addition, data catalogs can use data discovery to find similar datasets to

[7] https://data.europa.eu/en

connect to a certain dataset. By linking related datasets together, the search experience is simplified for the end-user.

Data discovery within enterprises can be impeded when a company has to take additional measures regarding privacy regulations or security certifications, which is addressed under *Governance and access management*. When data gets stored within a database this implies that it is subject to compliance protocols and storing has implications for maintenance. These consequences might not always be desired for a temporary dataset or solution which might lead to an architect creating a temporary solution for creating a data snapshot. As this lacks a formal definition, this does have implications for discovering these data fragments. Within governments, this is less of an issue, since the data owner publishes the data and therefore the catalog does not have to find the datasets but just present them. Governmental bodies might also benefit from some form of data discovery, since many countries have an implementation of a freedom of information act, requiring governments to provide data to citizens upon their request. These kinds of data catalogs within governmental organizations, which would be more similar to enterprise data catalogs, are not addressed in the literature and therefore no conclusions can be drawn from this.

Governance and access management

Since enterprise data that needs to be shown in the catalog is inevitably sensitive, strict access policies must be applied. This is further complicated by the need for compliance with regulations. Some examples of regulations companies have to adhere to are the European Union's *GDPR* which involves personal data to a broad extent for any company operating in the EU, the global *BCBS239* regarding the compliance of banks, and the USA's *Health Insurance Portability and Accountability Act (HIPAA)*. These all affect the access an individual employee may have to certain data, and which data needs additional security measures. From this, we can derive a strong demand for access control in enterprise data catalogs. This is in line with the features that have been identified in Table 4 [4]. For open data catalogs, access control is not needed as the data is, by definition, freely available on the internet for anybody to use [15].

Impact analysis

The feature of impact analysis is used to see what the effects of changing data would be on other data. As open data catalogs work with often isolated data, and different distributions, in combination with limited to no interaction opportunity with the data, they do not benefit from impact analysis. Within enterprise settings, data is often more interactive, with, for example, dashboards that are built upon data from different sources, or a dataset that aggregates data from different sources. As long as open data catalogs aim to offer an overview of available data rather than an interaction with the data, this feature is only important for enterprise data catalogs.

Metadata management

Both of the data catalogs rely heavily on metadata as the heart of the catalog. Both of the applications of a data catalog can run into issues with a lack of metadata, which inevitably results in a worse user experience. Examples of problems with metadata include false labeling or ambiguity [22]. Within open data catalogs, it is noticeable that metadata availability varies based on the organization which offers it [16]. It is assumed that businesses can run into the same issue, as the quality of the metadata is only as good as the level of attention the data providers put into it.

Collaboration

From the literature as well as from the study of the market it was found that collaboration is an important feature for a data catalog in order to let its users extract more value from it. Within the literature on open data catalogs, the value of collaboration is barely addressed. It is argued, however, that there are not enough assessment means, in combination with the number of visitors to catalogs and the number of datasets offered, to understand the potential collaboration can bring [22]. In addition, demand is identified [35] for users to collaborate by e.g. a discussion on a dataset. Although useful, collaboration is often a time-intensive process and does not necessarily benefit the objective of the data catalog to make data findable and searchable. From this, it can be derived that whereas it was known in literature and by the market that collaboration is a useful feature for enterprise data catalogs, it also adds value to open data catalogs, although it is not yet implemented in any of the examples of open data catalogs evaluated in this research.

Data quality profiling

Although this feature is marked as an advanced feature based on what the data cataloging market offers, data quality is key in the relevance of decisions made based on it. This is applicable for both business decisions as well as governmental decisions. As previously shown in Table 3, data quality profiling is found to be important in enterprise data catalogs. Data quality is also significant in open data portals. There has been numerous research and initiatives on assessing and measuring (metadata) quality, as outlined by [14]. In addition, also other quality metrics such as the usage of *truly* open data formats have been indicated as a problem in open data [22].

Machine Learning

Machine Learning, as well as Artificial Intelligence (AI), can help in automating certain tasks or processes within any of the features presented. Although the techniques can be highly beneficial to improving efficiency for both enterprise data catalogs as well as open data catalogs, this feature is not often addressed in the literature on the open data catalogs. Some literature does confirm its relevance in open data catalogs, primarily in automating the classification of datasets [15], [27]. ML could also aid in better consistency with multi-lingual datasets and catalog languages, a problem that is introduced later in this section.

Tribal knowledge sharing

This feature is not prominently represented in enterprise data catalogs and is currently not seen in open data catalogs. Tribal knowledge sharing was addressed in the literature of neither enterprise data catalogs nor open data catalogs. Since it relates to collaboration, with the opportunity to share tacit knowledge, there is some overlap and relevancy for both enterprise and open data catalogs.

2.1.2.6.2. Differences and similarities beyond feature-level

Language

Differences in language or even inconsistency in the language in free text attributes are problematic for searching through datasets [13], [27], [29], [36]–[38]. Generally, a user interacting with datasets of different languages either needs knowledge of all languages they interact with or risks losing linguistic relevance through translation. Although this may be a more significant problem for governments, who might have multiple official languages which their data has to be accessible for, or for cross-border cooperation, such as the European Data Portal mentioned previously. This problem is also relevant for enterprises, however, especially for those operating in different countries. Although they might agree on a single language for communication, it is difficult to enforce a default

language in the data that is collected, as employees typically interact with a system in their own language, and therefore data in the catalog might still be 'contaminated' with language inconsistencies.

Data format

In the previous sections, open data standards such as FAIR, the 5-star open data principles, and the DCAT vocabulary were introduced. These standards have been created to try and resolve issues that arise when interacting with different data sources, which inherently have their differences. These differences can be relatively small, such as different naming for the same fields, but can also involve the file type, making it difficult to combine sources. Even the European Data Portal, which encourages using standardized open-source file types, still has 50 data format options listed in its search option. This catalog also contains some duplicate filetypes, such as 'CSV' and 'text/csv' and proprietary formats such as Microsoft's .doc, .docx, and .xls and .xlsx. The issue of using proprietary data formats in open data catalogs is also addressed by [22]. This gives a clear indication that even strongly encouraging people to use certain formats does not give any guarantee that uniform formats are provided. Within enterprise data catalogs, there might be differences in the implementation of the FAIR principles, with for example the *Interoperability* focusing on achieving high data quality rather than using standardized open formats [4].

Data ownership

Within enterprises, data that is protected under some regulation such as the GDPR or other privacy regulations are often used, and desires or obligations to abide by data security guidelines such as ISO 27001 standards may have consequences. This gives certain obligations towards how long the data can be kept, and which security measures need to be taken in storing them. These do limit data consumers' access and interaction opportunities with data and make architects give a second thought to which data they store.

Industry standards

In sections 3.3 and 3.4, the application of data catalogs for open data was introduced. In these sections, a number of standards have been mentioned, such as DCAT as vocabulary, Comprehensive Knowledge Archive Network (CKAN) as a cataloging tool [31], [39]–[42], FAIR, and 5-star open data. All of these standards have been created by academics or the industry to optimize data usage and formats for optimal interoperability and data discovery [24]. These standards are widely used and enforced in open data catalogs [43], but lack adoption in enterprise settings. For example, enterprise data catalogs are found to primarily use proprietary metadata schemas [8]. Enterprise data catalogs can also benefit from the usage of existing open standards such as FAIR, albeit sometimes in altered implementations as shown by [4] who found that the focus differs for *Findable, Accessible,* and *Interoperable* when applying it in an enterprise context rather than in research settings. It is also important to note that using a standard does not guarantee that users will use it, making incentivization important as well as the application of standards [4].

2.1.3. Summary

Overall, a gap has been identified in the literature on the topic of the application of enterprise data catalogs, in general. Additionally, although academics and business researchers have developed various standards and guidelines to aid in sharing data between systems, systems of the analyzed commercial vendors use proprietary standards. A standard in data cataloging for enterprises has not been identified.

The primary benefit of a data catalog can be obtained for organizations that are either large in size, and therefore have a high need for collaboration, those who are smaller but do have large data repositories, or a combination of these two. Whereas most tools for data repositories are aimed at aiding the data providers, a data catalog offers significant benefits to the data consumers as well. Users, especially business users, are assisted in finding desired data by presenting the organizations' repository, as various literature confirms [4], [8], [18].

Three categories of features have been identified as the features currently offered in the market. At the moment of this research, core features of data catalogs included a business glossary, data lineage, search and automated discovery, governance, impact analysis, and metadata management. Through automated data discovery, the data shown in a data catalog is an accurate view of the available data resources [4], [12]. Additionally, the catalog helps the (business) user in providing the data they need, while keeping the information security policies of the enterprise through the governance features in a data catalog.

The impact analysis functionality of the data catalog helps data providers in assessing the effects of changing an entity on other data which depends on it. This reduces the time needed to analyze the data chain manually and shows which changes are needed, or if the proposed change is safe to be made.

Through the governance features of a data catalog, organizations are aided in retaining compliance with regulations and standards. Governance features, sometimes in combination with access control, consist of a rule management tool, which classifies certain data and only allows the appropriate Create, Read, Update or Delete (CRUD) rights when this is allowed for their role. If this is not allowed, they either have no access or have the possibility to request the needed access. This helps in ensuring that confidential and sensitive information is not available to everybody in the enterprise, although the data catalog might advertise the presence of this information.

It remains important to note that at the time of this research, there is very limited literature on the topic of data catalogs for businesses. A vast majority of the available literature focuses on open data catalogs. It was shown that there is some overlap between these two kinds of data catalogs, but there are also fundamental differences. In addition, the analyzed commercial data catalogs in this research are general data catalogs, which aim at being applicable in as many organizations as possible. Research on the application of enterprise data catalogs into specific environments, such as an iPaaS environment, was not available at the time of this research.

2.2. Integration Platform-as-a-Service (iPaaS)

Similar to the approach described in 2.1.1 for the literature on the topic of data catalogs, a similar approach was taken towards collecting literature on iPaaS platforms. The literature on this topic is even more limited. At the time of writing, a Scopus query of *'iPaaS' OR 'integration Platform as a Service'* yields 43 results. In order to filter these for literature relevant to this research, the same exclusion criteria as used for the literature on data catalogs are applied. One exception is criterium 3, which excluded literature written before 2005. No literature written before this year was found, as iPaaS are relatively new:

1. The article is written in English
2. The article is written in the domain of computer science
3. The article is published in a scientific journal, magazine, or conference proceedings
4. The article is accessible
5. The article has iPaaS as its main topic

Applying these criteria results in a total of 7[8] results.

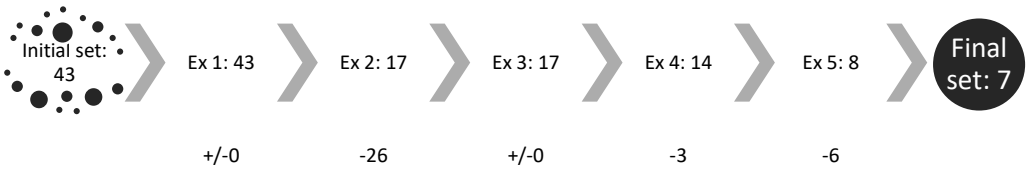

Figure 9: Results of applying the exclusion rules to the found iPaaS literature

The search query used can be regarded as a quite concise query. To illustrate that the number of results in the final literature, which really focuses on iPaaS is a relevant result, one of the papers in the final set did a literature review on a broader query: *("Integration frameworks" OR "system integration" OR "integration tool") AND "application integration".* This provided a total number of 108 articles, of which 15 were used [44].

2.2.1. Results

An iPaaS platform is a modern, cloud-based, approach to facilitating integrations between enterprise applications [45]. It helps in reducing the complexity of enterprise architecture by eliminating the need for point-to-point, also called application-to-application interfaces between applications. This used to be done through Enterprise Application Integration (EAI) middleware, of which an iPaaS is its cloud-based equivalent [46]. Its main benefits are that it combines mature enterprise application integration functionalities with the benefits of Software-as-a-Service (SaaS) applications, such as predictable costs, and that iPaaS are significantly less complex than traditional EAI middleware. This makes the development with an iPaaS significantly quicker and easier compared to EAI. The need for this quicker implementation is further demonstrated by the increase in the volumes of data, the demand to have real-time data leverage opportunities, and a trend in moving towards cloud storage solutions. iPaaS platforms are better tailored towards future changes in the enterprise application landscape, as they are ready to integrate with other cloud platforms. Since an iPaaS is hosted and managed in a cloud and offered as a service, the scalability of an iPaaS is a significant advantage. Similarly, the cost associated is more attractive than traditional EAI middleware, cloud services can be up or downgraded at a moment's notice, and therefore no overhead which might be rarely used needs to be taken into account.

Critical notes towards these iPaaS platforms include data security since all data flows through the internet in order to interact with the cloud-based iPaaS platform. Another note is that possibly sensitive data, such as metadata and application data is not only shared with the iPaaS platform provider but also indirectly with the cloud provider which the iPaaS provider uses to offer its services. The literature on the topic of iPaaS platforms is also very limited, failing to address, for example, a critical comparison of the benefits and disadvantages of iPaaS platforms compared to traditional enterprise application integration tools. Overall, the interest in adopting an iPaaS solution for integrations is on the rise, and therefore the market for iPaaS solutions is growing rapidly, with a growth in revenue of 38,7% from 2019 to 2020 [47].

Since an iPaaS platform can take a potentially central position within the enterprise architecture of an organization, the comparison to a data catalog, which is also a central overview of all data, can quickly be made. However, iPaaS platforms are not the same as a data catalog. The position of an iPaaS platform in the data hierarchy, however, means that the platform's central position enables it

[8] Applying all the exclusion criteria yielded 8 results, of which one paper was shown twice. Therefore, the final number of results is 7

to be aware of which data exists and how it is interacting with different applications. This offers opportunities to extend an iPaaS functionality with a data catalog. It is also important to note the benefits to the target user groups. Where a data catalog is typically rolled out for the entire organization [4], an iPaaS platform is often used only by select people in the organization.

2.2.2. iPaaS terms and definitions

Starting in section 2.3 and onwards, iPaaS terminology is regularly used. In order to have a good understanding of what is intended with each term, this section gives an overview of the meaning of the different terms.

Table 7: Definitions relating to iPaaS platforms, which are used throughout this research

Term	Definition
Client	A single organization, that has a subscription to the iPaaS. Every client has at least one model
Model	A collection of different systems that have interactions with each other
Environment	A cloud environment containing the interactions of a certain data exchange protocol
Data exchange protocol	A method of exchanging data between two systems. Can be synchronous and asynchronous. Examples of data exchange protocols include messaging and API
Integration	A connection between two different systems, exchanging a message using a certain format
Flow	The most detailed element in the hierarchy, enables the sending, receiving or sending and receiving of one message type from the system to the iPaaS platform
Message	A message consists of (parts of) one or more entities, translated to a message in a standardized format, such as JSON or XML, which can be exchanged between two systems
Entity	A data element, as would be stored in a database. Consists of one or more attributes
Attribute	A property of an entity, describing it. Has a type, such as *date* or *text*

To help give context to the position of each term within an iPaaS platform hierarchy as they are used throughout this research, Figure 10 provides an overview of this.

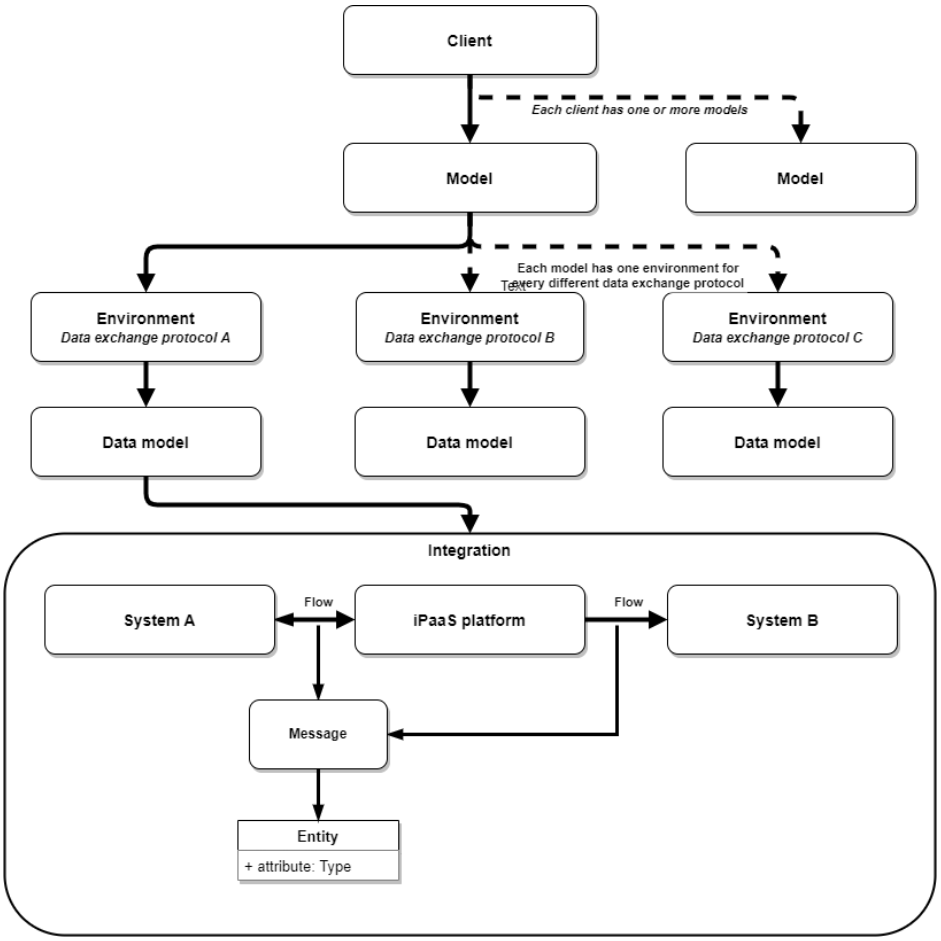

Figure 10: An overview of the iPaaS terminology and their hierarchy

2.2.3. iPaaS target audience

iPaaS platforms are created to integrate multiple applications. Therefore, not every organization might need such an iPaaS platform. Since the objective of an iPaaS is to connect systems with each other in a hybrid way, organizations would only obtain a need for such a solution when they have a certain number of applications that need to be integrated. This cannot be translated directly to a number of employees, as organizations in different sectors can have different data demands. For example, a construction company with 50 employees, of which 5 are in their office, might not have a strong demand to integrate their systems since these are supporting rather than operational: their business is producing physical buildings, and invoicing and paying their employees are supporting tasks. On the other hand, an e-commerce business with the same number of employees is more dependent on data for its core business, and would therefore have a need to integrate its systems.

Another difference between iPaaS and other software solutions are the users of the system. In an iPaaS, the users can create integrations between applications. Although iPaaS platforms are set up in such a way that the platform facilitates the creation of integrations between applications without the need for a programmer, this still requires a user with a certain skill set and knowledge of the system landscape at the client. Take, for example, the e-commerce vendor again. To be able to have their customer service department help customers quickly, the customer service agents need to be able to obtain relevant information about a customer, such as their personal data and order history, at the moment they call. This could be enabled by creating an integration between the customer service application, which contains email and a digital phone connection, to the customer relationship management software. This integration would then be used by every customer service agent who

uses the application. None of these agents might, however, be aware that the data comes from a different application, and none of them might even have access to the iPaaS facilitating this connection. Therefore, the actual group of users of the iPaaS is generally a small subset of the number of employees of the organization.

2.3. Expert consultation

This section describes how an interview is used to get an overview of the relevance of each of the data catalog features, as identified in section 2.1.2.2.1, for application within an iPaaS.

2.3.1. Methodology

Because of the differences in the user base of an iPaaS platform and an enterprise data catalog, the features identified in Table 4 may not all be relevant to be applied to an iPaaS platform. Since there is limited literature on both the enterprise data catalogs as well as the iPaaS platforms, literature is not a suitable method of selecting the features of data catalogs that might have relevance in an iPaaS. For this reason, seven interviews have been conducted with stakeholders from two companies. The first company is a provider of an iPaaS platform, and the stakeholders interviewed are active in the development of the platform. Secondly, stakeholders of an IT implementation company, a partner of the iPaaS provider, who use the iPaaS in client solutions have been interviewed.

Having both the view of developers and users of the platform ensures that a wide scope of potential features is obtained. Rather than evaluating each of their results individually, the following sections provide a summary of the findings.

The interviews are conducted in a semi-structured fashion to maximize the exploratory nature of this interview. The baseline structure of the questions is included in Appendix A. Since the objective of this interview is to find the enterprise data catalog features which can also be of added value in an iPaaS, each of the questions is assigned to one of the categories of the enterprise data catalogs as shown in Table 3. Some findings in the results section do not directly correlate to one of the questions because of the semi-structured nature. Similarly, some of the questions as outlined in the structure have a very wide scope, to maximize the input from the interviewees. If the interpretation of the interviewee did not match the question in the way intended, follow-up questions were asked to narrow the scope and ensure that each interviewee got to share their view on each of the topics.

Within the iPaaS provider, interviewees were stakeholders in management, development, marketing and design have been interviewed.

At the implementation partner, consultants and architects have been interviewed.

2.3.2. Results

This section discusses the answers to the interview questions. Conform the structure of the conducted interviews, the results are listed per category of data catalog features. The interview started with the interviewees indicating which companies would need an iPaaS for their organization. The general consensus on this question is that every company with five or more applications that need to communicate with each other would benefit from adopting an iPaaS. With regards to the specific iPaaS platform the interviewees used or developed, some of the interviewees indicated that it is focused on an enterprise level, as the platform is able to handle complex integration patterns, which is not the focus of all iPaaS providers.

Searching

The iPaaS currently does not provide a free text search function within its application. Instead, searching is possible through the browser default option (accessible through the shortcut ‘control’ +

f) and the possibility to change the view by filtering on previously, manually assigned tags. Most of the interviewees indicate that the tagging functionality is not used and maintained to its full extent and therefore filtering on them is not always of added value.

The participants have different opinions on whether a free text search function would be of added value. Whereas a small majority of the participants indicate that they find the free text search which is currently offered in the documentation of the tool very useful, other participants indicate that a free text search giving access to a certain system or integration is not always useful, since the platform is created to maintain a visual overview, and it would be hard to show the visual context in a search menu. The participants indicate that adding a glossary function, as previously described as a data catalog feature, would be very useful.

Finding

As mentioned in the literature on iPaaS solutions, they offer support for different data exchange protocols (such as messaging or API data exchange). The analyzed platform currently offers separate configurations for each of these separate data exchange protocols. All participants indicated that there are disadvantages to the current approach where the configurations are fully separated. Some interviewees think that some functionalities, such as configuring a system, are not always put into a logical place. This can cause issues, especially for less experienced users of the platform. Other interviewees indicate important benefits to the split, as they think it provides more clarity than combining the views. They indicate that it would be difficult to indicate differences between the different data exchange protocols used for a certain flow, and there would be a need for significantly more information on the screen, which would negatively affect the visual overview aims of the platform.

The platform currently aims at providing clarity through splitting views based on functionality. When evaluating an environment based on the data that it transfers, this significantly complicates finding the data, since this would need to be evaluated manually and in different views. The interviewees' responses indicate that the current split in the landscape should stay possible for development, but would like to have a new view that combines these different data exchange protocols into a single view to get a better insight into which data is used where. Such a top-level overview is currently not present within the platform and most interviewees indicate that this overview would indeed be useful. In the case of large clients, who have multiple models as part of their iPaaS adoption, they indicated that these clients carefully evaluated which systems and integrations they placed in each model. A minority of the employees believed that due to this careful division into different models, these customers would not benefit from a top-level view of systems, integrations and data exchanged. This is in contrast to what the majority of the interviewees believe. They expected these larger clients would especially benefit from a top-level overview, since the division into separate models is not always based on business units, for example, but also on technical reasons. These technical reasons might include risk reduction and performance motivations. Also, if the division into a different model was indeed based on a logical division such as per business unit, an architect would still need to have an overall overview. They currently maintain this overview through documents and diagrams outside of the platform, but it would be more desirable to have this option within the platform.

Discovery

Systems included in an iPaaS are usually added manually. The main profit for data discovery would be by discovering deviations from the recorded patterns, for example detecting when messages constructively arrive with more attributes than recorded in the platforms' predefined message

definition. Currently, the interviewees indicate that messages would be rejected if there would be any sort of validation over them. If messages would not have a validation action, they would most likely go through without error. Since the platform does not store any data, it would not be known that these messages have ever arrived.

Some of the interviewees confirmed that message definitions do change regularly, although some message definitions rarely ever change. Overall, users are positive about adopting data discovery through received messages, and multiple interviewees indicate that this would become especially useful when the platform would be used more for sensor data such as IoT devices.

Discovery (maintenance and resource control)

The participants were asked whether the platform provided the user with information on the usage of integrations in a production environment. The interviewees indicated that the platform does offer a way to view usage per integration at a glance, but it would be possible to derive this information by manually analyzing the statistics and logs of a certain integration. Almost all interviewees indicated that the user would benefit from having a feature showing the integrations that are almost never used. They indicated that there are numerous customer environments that have unused integrations somewhere in their lifecycle. This creates a distorted overview, as it may contain numerous integrations that are not active. Consultants currently do not have their main focus on maintaining a 'clean' landscape, since properly removing the unused applications is indicated to be cumbersome work, although this removal process has improved compared to earlier versions of the platform.

Governance

Regarding the question of which governance tooling the platform already offers at this point, the interpretations were different. Because of the exploratory nature of this interview session, these different answers were included and are listed below. The interviewer ensured that every interviewee was also asked about the features the platform includes in terms of governance as it is defined within enterprise data catalogs. This was previously defined in Table 3 as *monitoring and enforcing compliance to legislation and regulations and aiding during audits of regulations*.

Features the platform currently offers with regards to governance, according to the interviewees, include:

- Dashboards showing incidents
- Log entries
- Edit history per flow
- User (role) management
    - Conform procedure, user rights of each environment are re-evaluated monthly
- Connection to external dashboards

Most of the interviewees indicated that they would find it desirable to have more governance features within the platform, although they did not have specific features which they were missing. Three of the interviewees at the implementation partner indicated that they currently use a third-party tool to obtain important insights regarding performance insights and filtering error messages on time. They would prefer to have these features built in into the platform rather than relying on the external tool.

Another interviewee indicated that the requirements for any new governance features would need to be properly evaluated with customers to ensure that this conforms to their expectations and is something that they would need and use. In addition, some governance features might also need

legal insights. The interviewees did not have knowledge of any of the end customers demanding a feature currently not offered by the platform regarding governance.

All interviewees indicated that the position of an iPaaS is a suitable position within an organization to offer governance features. Some indicated that since the target market segment of this iPaaS are enterprises, a certain level of governance features within the tool are a must-have rather than a nice-to-have feature. The platform therefore currently fulfills these must-have features.

The main target audience of a top-level overview of system usage and data operations is expected to be the architects.

Data lineage and impact analysis

The interpretation of the interviewees of the questions regarding data lineage and impact analysis lead to conflicting results. Some of the interviewees indicated that data lineage is currently already indicated clearly in the system, by marking the dependency of systems on each other when one is selected. Although this is correct, interviewees who focused this question more on the data transfer that the integrations facilitate would like to have the opportunity to see where data is viewed or mutated, since the platform does not currently offer a single overview of this. For data that is protected under privacy regulations such as the GDPR, such as personal data of employees and customers, interviewees at the implementation partner notice customer demand to have indications of which systems access and edit data. This currently has to be evaluated manually. With increases in the number of customers, the interviewees indicating this demand expect this to happen often enough to include such a detailed data lineage as a feature within the platform. Other benefits regarding more advanced data lineage include the opportunity to provide impact analysis, showing clearly which flows are affected when data is changed or when a message definition changes. This simplifies work for the developer, as they do not need to manually evaluate this.

Access control

Currently, the platform does not offer access control on integration level. Instead, a division is made based on the features of the system the user would need. Within each of the feature groups of the platform (e.g. designing the system, technical specification of systems and integration, deployment, and monitoring), read and edit rights can be given. This means that a user who should not be able to deploy new features to production can be excluded, but once a user has edit access to the technical details of one system, they are able to edit all systems within that feature group. All interviewees indicate that the current level of access control is adequate. This is mainly because developers generally work on a team basis and need to have an overview of what is already available in order to do their work, and larger clients, with different teams working on separate focus points, have a division of models, to which they can assign different team members. The interviewees indicated that there is currently no demand to further narrow down the access control to a per-system or per-integration basis. This would require significant change within the platforms' landscape, by adding ownership and approval flows, for example.

Some of the interviewees indicated that if this demand were to arise from a (future) customer, a more suitable method rather than providing access per system would be to do this per group of systems. This could be implemented by using the previously mentioned tagging method, where a user could, for example, be given access to all systems tagged as *finance*. This kind of specifying would, however, go against the objective of a data catalog, which is to give the user a full overview of the data and systems within the organization.

Data quality

Interviewees indicated that the platform does not enforce its users to take any actions regarding data quality. Important to note is that the platform itself does not store data, and thus any quality improvements would be through how the architecture and structure are visualized or how the data is processed and modified. Since the platform is set up to provide clarity by using visual representations, these visuals can already help the user to get a clear view of what is within the organization's data portfolio. On a more data-focused level, the platform offers means to convert data entries to other data types, add filters to ensure correct input and output, and enrich data.

None of the interviewees see a demand to add more tools to improve data quality. Some indicate that some kind of advice on, for example, attribute naming could be helpful, but these kinds of nice-to-have features are not seen as a feature that should get priority.

Collaboration

The platform is accessible through an internet browser and is accessible by multiple users simultaneously. It offers version management, a central model which provides a *single truth* and has been designed to use on a team basis since different expert viewpoints are needed to capture the organization's environment, configure integrations and deploy them.

Collaboration is often done through the usage of external tooling, e.g. for assigning tasks to a user and communicating with colleagues. The interviewees generally feel that there should not be a goal to add features these external tools already offer to the platform since this would limit the opportunity to work with external parties who might not have access to the platform, but do need to be involved.

Some interviewees indicated that the integrations, which cannot be edited simultaneously, do not clearly indicate who is currently editing them, which might result in two users doing a similar task and therefore not spending their time as productive as it could be. Another indication was that it could be helpful to see which other users currently had a certain environment open.

A limiting factor of collaboration is licensing. The platform currently does not make a difference between a technical user of the platform, who needs to configure integrations, and users who would only need viewing rights. This limits the number of users within an organization who have access to the platform. Although most users might not need to view a data landscape on a level at which they cannot see the actual data, only their origins, this does conform to the data lineage feature of a data catalog, which enables (business) users to view the origins of their data and where it was edited, so they can ensure that the data they use for their decision making comes from the source they expect it to come from.

Top-level data model

Although this feature is not mentioned specifically in the previous tables which outline the features of an enterprise data catalog, this feature is added to the list of features based on the findings of the interviews. When explaining the concept of the enterprise data catalog to the users who are not yet familiar with this product, one of the appealing concepts of such a catalog is the visual overview of all the data in the organization. This can be seen as the core function of the enterprise *data catalog*, as it gives a *catalog* of the available data. All the other features around this are supporting this main objective. As to prevent unclarities by using the term *data catalog* for both the product of enterprise data catalogs as well as one of its features, the term *top-level data model* will be used as the feature name throughout this research.

2.3.3. Conclusions

In their answering, many of the interviewees advised on the technical feasibility of some of the features and some of their suggestions. Although this does not impact the findings of the interview, this might affect the choices made for the prototype that this research produces. Based on the seven interviews, the relevance of each of the identified data catalog features to an iPaaS, as previously identified in the literature in chapter one is shown in Table 8. The features are sorted alphabetically. To see how these features might be related, refer to Figure 6.

Table 8: Relevance of data catalog features in an iPaaS in general, based on findings of the interviews and literature

Feature	Relevant	Somewhat relevant	Not relevant	Expected in an iPaaS
Access management	✓			✓
Business glossary	✓			
Collaboration	✓			✓
Data discovery		✓		
Data lineage	✓			✓
Data search	✓			✓
Data quality		✓		✓
Governance	✓			✓
Impact analysis		✓		✓
Metadata management		✓		✓
Top-level data model	✓			
Legend (last column)	✓ *expected*		✓ *expected to some extent*	

Apart from the relevance of the data catalog features to a general iPaaS, the interviews can also be used to provide an overview of the features worth putting further development into. In other words, the features are not offered to the desired extent at the current moment in the analyzed iPaaS. Note that these findings might not apply to all iPaaS platforms, but are based on the findings of the interviews which all used a single iPaaS platform. The findings can however be generalized to some extent, as is explained further in this section. In the rest of this section, the motivations for rating each of the features as (not) relevant in Table 8 and Table 9 are given.

Table 9: Data catalog features worth developing for the analyzed iPaaS, based on the interviews

Feature	Development relevant	Relevant but no priority	Development not relevant	Already in the analyzed platform
Access management			✗	✓
Business glossary	✓			
Collaboration			✗	✓
Data discovery	✓			
Data lineage	✓			✓
Data search	✓			
Data quality			✗	✓
Governance		✓		✓
Impact analysis		✓		✓
Metadata management		✓		✓
Top-level data model	✓			
Legend (last column)	✓ *included*		✓ *partially included*	

The remainder of this section provides the motivation for assigning each of the features as *Relevant*, *Somewhat relevant,* or *Not relevant* in both Table 8 and Table 9.

Access management

- *General*
  Access management can be seen as a specialization feature of the group of governance features. Because of the difference in the intended userbase of an iPaaS and a data catalog, an iPaaS is a less relevant position for access management because its userbase is limited to more specialized users rather than all users that exist in a company. This would create a need to add an external directory of users. An iPaaS can however clearly manage which systems can interact with certain systems, and through which channels they can access information. This does offer some kind of control over where data can flow, although not on an individual user level
- *Platform-specific*
  The platform currently has coarse access management. Users' read and edit rights can be assigned separately for each of the phases, to ensure that only users with the proper expertise have access to a certain part of the integration development. Access control on system or integration basis is not possible, but there is no demand for this at this point since this would limit users of the platform rather than help them, as well as impede collaboration and independence.

Business glossary

- *General*
  An iPaaS is a suitable place to provide a business glossary. Although it is not used by as many people as might depend on a data catalog, the users of an iPaaS have different levels of domain expertise. In order to facilitate each of them to use the platform without depending on team members for an explanation of some integrations, they might be in need of some business or technical context. This can be provided by a glossary.
- *Platform-specific*
  The platform does currently provide the option to provide a description for each entity and its attributes, but this option is not used extensively. According to the interviewees, this is caused by the lack of usage of such a definition at later stages of the platform. Interviewees indicate that these would only be useful if they were to be used more extensively throughout the platform and could be accessed in every place. In addition, this kind of glossary is stated to be relevant for stakeholders with lower domain knowledge, such as support staff as well as the consultants of the implementation partner who are not (yet) familiar with all definitions used within the clients' domain.

Collaboration

- *General*
  iPaaS platforms are built to be used by multiple people since knowledge of different areas of expertise is needed to successfully develop integrations.
- *Platform-specific*
  The platform is already set up in a way to be used on a team basis, and different user perspectives are needed to successfully build integrations. There are some features indicated that might be able to further improve collaboration options, but most of the interviewees did not see any of these features as a feature that should be given priority at this point and think that the dependence on external tools is normal in teamwork, and the aim should not be on

integrating features currently given by external tools into the platform since this would impede collaboration with external actors. Therefore, some relevance to the development of collaboration features was found, but these were not considered to be of priority. Since most of these features, such as assigning tasks to users, are currently already done through external tools, development into collaboration aspects is not seen as relevant.

Data discovery

- *General*
  Within an iPaaS, there would be no need to develop a data discovery algorithm such as is present within a data catalog. iPaaS solutions are built to specifically connect applications that need information exchange, rather than mapping every data which might be available within the organization. It does benefit from data discovery but on a different level. For example, an iPaaS could do more with rejected messages to see if these deviations are a trend or coincidence, and therefore discover differences between reality and system definitions.
- *Platform-specific*
  The platform currently does not offer (active) data discovery. The systems shown in the platform are added manually by a domain expert on the client side. Since the system does not store any data and an iPaaS is not connected to the entire company, but only the systems the client chooses that need data exchange, automated data discovery in a way a data catalog offers is not possible and not desired. On the other hand, interviewees indicate that more could be done with messages that are currently rejected by the system. This would be relevant for the data exchange types of messaging and API. Currently, a rejected message could be found manually, but there is no clear dashboard and the abilities to sort error messages, for example, timeslot or error type, are too limited. Benefits could be obtained by providing the user with a clear dashboard where they can see whether a rejected message was an incident or a trend.

Data lineage

- *General*
  Providing data lineage gives significant benefits to the users of an iPaaS platform. It shows the developers how data entities or attributes are interacted with throughout their lifecycle. This gives significant benefits, as it can show a developer that the integration he wants to create already exists, helps in showing conformance to data protection regulations, and help a developer assess what he needs to change when a flow is changed.
- *Platform-specific*
  Currently, the platform offers some data lineage overview. It is built in a way to dynamically show which systems and integrations are affected when a system or integration is selected from the visual overview. Interviewees indicate that this overview is currently already clear and often used, but an overview of interactions on the data level is currently lacking. In order to see what would happen with the data, the integrations should be analyzed on a more detailed level on a per integration basis. This takes considerable time for the user, and there is a demand from the client to have this kind of knowledge. This feature is therefore listed as very relevant.

Data search

- *General*

For larger clients who have large models and numerous integrations, it can be difficult to provide the user with a full overview of their model. In this case, a search function might be relevant for the developer to quickly find and interact with entities and flows.

- *Platform-specific*
  Methods for searching are not included in the platform, other than using the browser built-in search function, which can only be used for exact text matches which are currently on screen, and filtering on previously, manually assigned tasks. Interviewees indicate that it would be useful, especially for users with lower levels of domain knowledge, to find integrations and systems based on their name or description rather than only their technical name. This search function should clearly show in which phase the element is located, as the platform has a strict division into different phases. Interviewees indicate that they are already content with the search functionality offered by the documentation of the platform.

Data quality

- *General*
  iPaaS platforms are not expected to have a data quality dashboard, similar to a data catalog. They should be able to have data validation and enrichment features, but since they depend on how systems supply the data, the platform cannot fully control how data comes out, since this depends on the input. Data quality profiling is therefore not a focal point of an iPaaS, although some quality features relating to input and output should be in the platform.
- *Platform-specific*
  Data quality is not profiled in a dashboard and statistics as done in some data catalogs, but the platform does provide means of ensuring data quality. By using formal processes and motivating the users to map their landscape, the origins of data are more clear. In addition, data input can be validated, transformed, and enriched prior to outputting it to the target system. This position as middleware ensures that the data input into depending systems is of high quality, and therefore does help data quality in the overall organization.
  The platform does not, however, force any quality decisions onto the users. Since the platform is used to connect applications running in production environments of large companies, the data is already expected to be of some level of quality, but the platform only enables its users to improve their data quality if they are actively using the features the platform offers. Interviewees indicated that the user should never be forced to abide by a certain principle.

Governance

- *General*
  Because of the position of an iPaaS, it must have certain governance aspects, to ensure proper user control, aid in showing conformance to regulations, and transferring data securely. Governance is an area that has ongoing development. For example, the European Union's privacy regulation GDPR was introduced recently, in 2018, but was of significant impact. In the upcoming years, more regulations might be imposed in the future. Organizations expect software they pay for to at least adhere to privacy regulations. Enterprises might expect even more governance features by default in enterprise-oriented applications.
- *Platform-specific*
  As some of the interviewees stated: the platform is designed for enterprise usage, and therefore governance features are a must-have rather than a nice-to-have. The platform therefore already offers governance features, some of which are indicated in the previous

section. A majority of the users indicates that the platform could do more on the level of governance, but no suggestions for missing features were given, as the interviewees did not experience this feature demand from the clients.

Impact analysis

- *General*
  Impact analysis can be seen as an extension of data lineage. It helps in assessing what other elements in the landscape are affected when a certain element is changed. Although this might not be a default yet in iPaaS applications, developers significantly benefit from this clarity, and would therefore be very relevant to include within an iPaaS.
- *Platform-specific*
  Similar to the data lineage, impact analysis is currently offered on integration-level, but not on the data level. Since users indicated that they experience a cumbersome process to remove redundant integrations from the platform, and they are often kept in, polluting the overview, impact analysis could benefit the platform. This would ensure a cleaner and better maintainable landscape and solve a user problem. Since the interviewees indicate this as more as a nice-to-have feature than a must-have feature, it has been rated as somewhat relevant.

Metadata management

- *General*
  Every iPaaS has some form of metadata features since the core entities within an organization need to be known in order to build integrations around these.
- *Platform-specific*
  The platform does offer an overview of the available data, and ways to describe the data. Therefore, it does offer metadata management to some degree. The main profit to be obtained here would be connecting the data models in the application with a glossary, so the field names can be understood by anybody with access to the platform. Therefore, this feature was listed as somewhat relevant, but it can be improved by focusing on the business glossary.

Top-level data model

- *General*
  For an iPaaS, the top-level data model offers a means to show all the entities and attributes which are present in the numerous messages which exchange between the different systems through a single overview. This aids the user to retain a clear overview of the messages which are sent between the different systems and enables the user to make an abstraction to map the data model of the platform to the data model of the organization.
- *Platform-specific*
  The platform currently has data models which are always shown at full level. For customers who have large numbers of integrations, these data models can become large and do not give an overview at a single glance at the data model. By enabling *zooming in* and *out* of the model, an abstraction of the detailed model can be used to provide an overview, and selecting a certain object can help the user *zoom in* on this object to see its more detailed usage.

3. Treatment design

This chapter focuses on developing a design for a solution to the research problem as was identified in chapter one. Before focusing on a potential design, the stakeholders are evaluated, followed by a list of requirements the design should fulfill.

3.1. Stakeholders

Stakeholders can be identified as the person, persons, or organization that is affected by treating the problem [48]. All of the requirements which are identified further in this chapter relate to one or more of the stakeholders.

As this research is being conducted at a provider of an iPaaS solution, some clarity is first needed in the definitions used in this chapter. The provider of the iPaaS solution is addressed as *the provider* throughout this chapter. The implementation partner, which is using the iPaaS solution offered by the provider for client solutions, is addressed as the *partner,* and the company paying for using the providers' platform (either directly or through the partner) is addressed as the *client.*

With a clear definition of the various stakeholder origins, the stakeholders can now be defined. First, the different stakeholders are outlined, then they are connected to the stakeholder roles as indicated by Wieringa in chapter 4.1 of his book [48]. Since the intended solution is intended to solve a problem of the client, the stakeholder list is ordered in such a way that first the client stakeholders are mentioned, followed by the partners' and finally the stakeholders of the provider.

- **Architect:** work at the client and are responsible for maintaining the overall architecture of the organization. Have high domain knowledge of the clients' company and IT landscape.
- **Developer:** are active users of the iPaaS platform of the provider, using it to develop and maintain integrations for the client. This group consists of consultants working at the partner, as well as employees of the client. Have moderate domain knowledge and high platform knowledge.
- **Business user:** This role consists of users working at the client that are currently not using the iPaaS platform of the provider. They do not have extensive technical knowledge of the different systems the client has, or in which ways the systems might interact with each other and instead are working with the data that the platform transports between the different systems of the client. They can use the system to track the origins of their data to ensure their quality and validity. Examples of this stakeholder include executives who want to ensure the validity of their data before basing a decision upon them, and a security officer who needs to know which applications access, modify, and store data that is protected by the GDPR.
- **Support:** employees in the support department of the provider and the partner, who need to ensure all integrations and environments of the client stay operational. Also need to respond in cases of incidents.
- **Product owner:** work at or for the provider and need to implement new features. Have little knowledge about the domains of specific customers, and high knowledge of the platform.

For the scope of the requirements to be analyzed in the next section, this research does not consider stakeholders in the wider environment, such as financial and political beneficiaries or threat agents. The list of requirements is kept concise and focused on the features. Requirements that are considered trivial for an enterprise-oriented application such as an iPaaS are not mentioned. An example of such a requirement could be that the data processing has to be in line with the requirements as imposed by the GDPR.

Table 10: Stakeholders and their stakeholder roles

Stakeholder	**Stakeholder role according to** [49]	**Stakeholder role description adapted from Alexander** [49]
Architect	Functional beneficiary	Benefit from the output of the system by using it.
Developer	Normal operator	Has routine interaction with the system can be seen as the end-users
Business user	Interfacing systems	Primarily use systems whose behavior is affected by the system, but have an interest in the scope of the system under design
	Functional beneficiary	Benefit from the output of the system by using it.
Support	Operational support	Support the normal operator and ensure that the system stays operational
Product owner	Developer	Consists of designers, programmers, and testers who build the system. Do not benefit from system output during its runtime
	Maintenance operator	Interact with the system to keep it running

Each of the stakeholders identified in Table 10 is briefly described below with a shortlist of desires with regards to the system in design, which helps in getting a better understanding of their desires and relates the requirements to each of them.

Architect

Key focus areas for the architect include an overview of the entire organization's IT landscape. Since the architect has to collaborate with people who do not have access to the iPaaS, they benefit from overviews that are understandable by external actors without platform knowledge. Because of their advanced domain knowledge, the architect can be used to provide entries to the business glossary.

Developer

The developer obtains benefits from most of the potential features to be developed. They would benefit from the business glossary since this enables them to work more independently without extensive domain knowledge. They can combine the business glossary with data search to quickly find the systems they need and see how data is edited in each flow through the data lineage tooling. The data lineage, as well as the impact analysis, helps the developer in maintaining an overview of the integrations that already exist, which helps to eliminate duplicate integrations, and therefore enables the developers to prevent unused integrations within an environment, which benefits the overview of all users. The impact analysis also makes the removal of flows easier.

Business user

Did not have previous access to the platform, and has very limited platform knowledge, and some domain knowledge. Gains benefit from being able to track data origins. Within a data catalog, this would be the main target group of users, which would interact with the catalog to find the data they need. For the application within the iPaaS platform, this user could benefit from viewing data lineage and therefore knowing which systems they are interacting with. This could also help this user in obtaining data they need which is currently not within their systems.

Support

Previously, the support staff was identified as a group of individuals with platform knowledge, but due to their workload consisting of various clients, they cannot have domain expertise for each client. To better understand a runtime issue that is reported to the support actors, they would

benefit from a business glossary that describes a system and therefore helps the support actors in determining how vital this system is within the architecture. Additionally, the impact analysis would help them in finding other systems, flows, and integration that are affected by a problem.

Product owner

The product owner needs to decide whether and how to implement certain features and make sure they are maintainable. Their main focus is on the technical feasibility of proposed solutions, as well as the fit into the platform.

3.2. Goals

In the previous section, the stakeholders and the ways in which they can benefit from some of the proposed features, based on the literature on data catalogs as well as the interview with some of the stakeholders, have been identified. This section states clear goals, and to which stakeholders they relate. This helps in understanding the stakeholders' motivations and the level of involvement needed to successfully develop features that satisfy their stakeholder goals.

Table 11: Stakeholder goals

#	Goal	Stakeholder
1	To manage the entire IT landscape of the organization through a single overview within the platform	Architect
2	To have the overview understandable to external actors to enable collaboration	Architect
3	To reduce the number of redundancies in the platform	Architect
4	To understand the business context of a certain system or integration	Developer Support
5	To simplify the removal of redundancy in the system by knowing which dependencies a flow has	Developer
6	To search for systems, integrations, and flows by their name or description	Developer
7	To know the origins of the data used to make a business decision	Business user
8	To see dependencies of systems, integrations, and flows	Support
9	To ensure the platform offers relevant and expected features	Product owner
10	To develop and maintain the platform	Product owner

3.3. Requirements

This section identifies the requirements for the system under development. These requirements are set up as suggested by Wieringa [48]; split into functional and non-functional requirements and related to each of the goals and therefore to a stakeholder.

3.3.1. Functional requirements

The functional requirements describe the desired function of the system under design.

Requirement 1: The platform should offer a total overview of systems and system integrations within the organization

This requirement contributes to goal 1. The overview is mainly used by the architect, and the main group of clients who benefit from the fulfillment of this requirement is the clients who have multiple models, which can currently not be combined into a single view.

Requirement 2: The overview the platform provides should be maximized by grouping similar entities to minimize the visual load

As an environment gets used for an increased number of integrations, it is likely that the data model starts containing a large number of entities. Although these entities combined offer a full overview, they also negatively impact the overview the data model offers, since it takes considerable time to take in the full picture. By reducing the data model to the core entities, a better overview is provided. By excluding the secondary and tertiary entities which are connected to a 'parent' entity, fewer elements need to be shown on screen and therefore only the most relevant entities are shown on screen, which helps with making the overview useful for external parties. This requirement contributes to goal 2.

Requirement 3: The platform should offer a business glossary throughout the entire platform, which dynamically shows descriptions for both business definitions and technical definitions

A business glossary provides business context to terms to ensure that terminology can be understood by business actors. For an iPaaS that has both users with domain knowledge but no technical knowledge as well as users without domain knowledge but with technical knowledge, this glossary should work in two directions: technical to business and business to technical. This aids the business users to understand their data origins and helps developers who do not work directly for the client as well as support staff to understand the business context. Contributes to goal 4.

Requirement 4: The platform shall offer data lineage as an extension to the current lineage on an integration level

The platform currently shows which integrations depend on each other, rather than showing this information on data level. To contribute to stakeholder goals 5 and 7, this should be extended to a data level. This can currently be viewed manually, by accessing each flow that might access an entity. In order to be able to do this significantly faster, the platform should extract this information to a higher level, showing the interactions of each system based on a selected entity or field.

Requirement 5: The platform should offer search functionality that is connected to the glossary, which enables searching for systems, integrations, and flows

Search functionality would contribute to goal 6 and would help the overall goal of helping a developer to work more efficiently on the platform. The search functionality would ensure that the developer does not only rely on visual cues and knowledge of the specific integration names in order to find them but can also find them based on their name or description. This searching by description can be facilitated by using the glossary as previously defined in requirement 4.

Requirement 6: The platform should show statistics that can be used to derive the usage of each integration

This requirement focuses on giving the developers and architects the means to identify applications that are rarely or never used, so they can decide based on this whether these integrations need to be removed from the environment, whether there is another integration doing the same or whether there is an issue with the integration. Contributes to goal 3.

3.3.2. Non-functional requirements

These requirements relate to those that cannot directly be translated onto a function of the system under development but instead relate to quality properties as defined in the ISO 25010 standard, section 4.2 [50].

Requirement 7: The overviews provided by the platform shall be understandable to people who do not have knowledge of the platform

Ensuring that the overviews are understandable by people without technical knowledge of the platform ensures that the overviews can be used to their full extent. This can be obtained by ensuring the overviews present the data in a visual way, which follows the principles of industry standards, such as BPMN[9] or ArchiMate[10]. Satisfies goal 2, and relates to recognizability.

Requirement 8: Ensure that features of the platform are according to client and market expectations

In order to stay competitive, the platform should offer services that are expected of an iPaaS platform, and it should use modern techniques in order to keep up with market demand. Relates to goal 9. The focus definitions of the ISO 25010 standard are *functional completeness* and *functional appropriateness*.

Requirement 9: Ensure maintainability of the platform

Any new feature included in the platform should not affect the maintainability of other components negatively. This can be ensured by developing in a modular fashion.

3.3.3. Requirement and stakeholder overview

In the previous three sections, the stakeholders, their goals, and the requirements for the system have been identified and motivated. Although Table 10 and Table 11 already help in identifying their position, an ArchiMate motivation view has also been created, to relate each stakeholder to their goal(s) and corresponding requirements.

This ArchiMate motivation view consists of some additional attributes. From top to bottom, the stakeholders, drivers, goals, outcomes, and requirements are shown. They can be recognized by their symbols. This visualization clearly shows who the primary beneficiary of a requirement is, and which goals they fulfill. The numbers in curly brackets indicate the goal or requirement they are connected to. This ArchiMate motivation view is shown in Figure 11.

[9] Such as BPMN 2.0: https://www.omg.org/spec/BPMN/2.0/PDF

[10] Maintained by The Open Group. Examples of visuals used by the ArchiMate 3.1 specifications can be found here: https://pubs.opengroup.org/architecture/archimate3-doc/

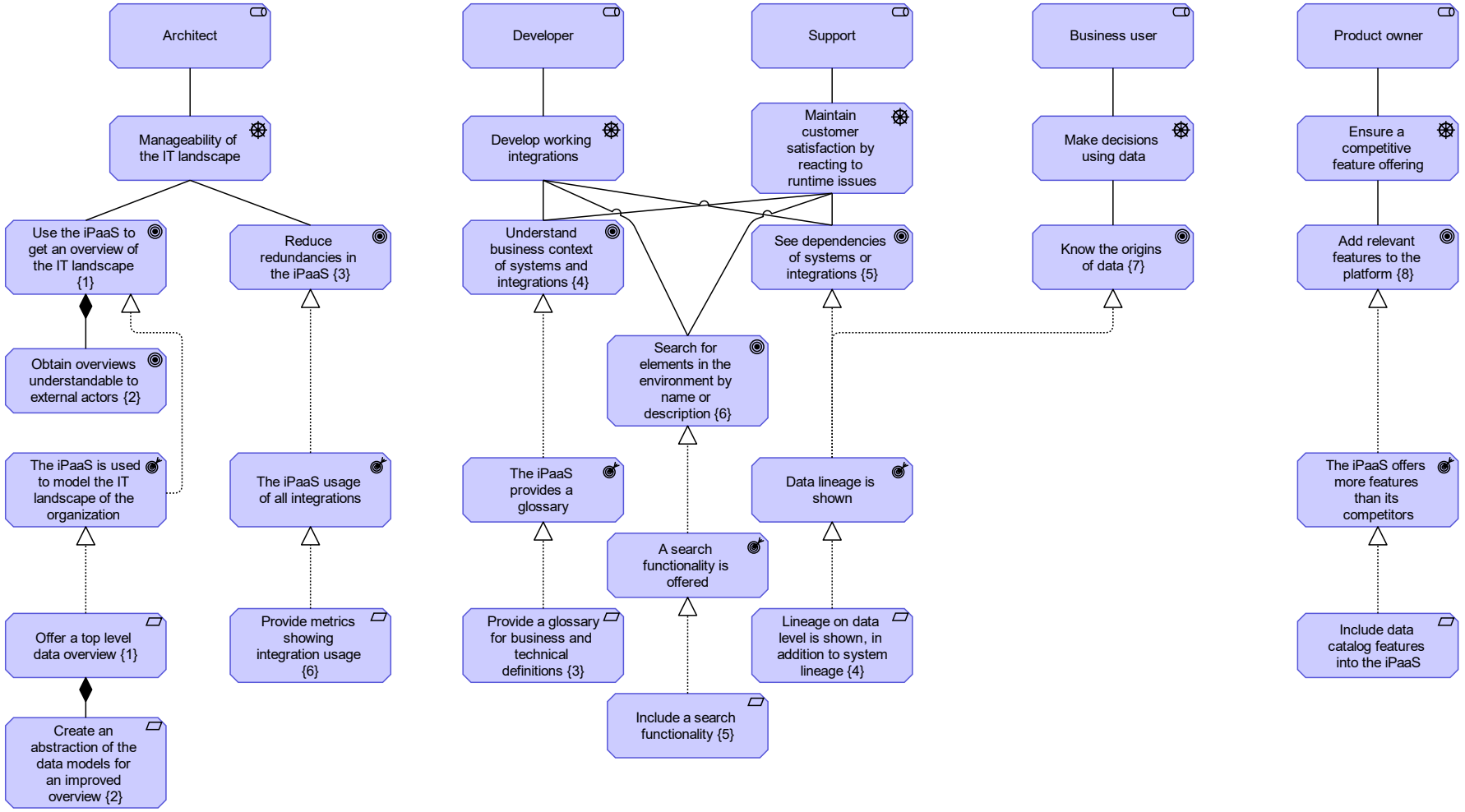

Figure 11: ArchiMate motivation view of the identified stakeholders, goals, and requirements

3.4. Existing solutions

In section 2.1.2.3, the existing solutions with regard to enterprise data catalogs have been identified and analyzed. A similar approach is taken to analyze existing iPaaS platforms. This analysis is used for other iPaaS platforms than the one evaluated in the interview in section 2.3. Similar to the approach in the previous market analysis, the Gartner Magic Quadrant for Enterprise Integration Platform as a Service [47] is used to determine the *market leaders* to analyze. The Gartner report includes only those companies who have a significant number of clients, in this case at least 900 clients, of which 200 should be direct. The inclusion criteria also included support for the features which Gartner identifies as the base characteristics. These features are summarized in Table 12.

Because of the similarities in the setup of the Gartner Magic Quadrants, the *leaders* are, again, included in this analysis. For an explanation of the meaning of each of the four quadrants *niche players*, *visionaries*, *challengers*, and *leaders*, see Section 2.1.2.3.

The companies whose iPaaS solutions are evaluated in this section are Informatica, Boomi, Workato, SAP, MuleSoft, Oracle, Microsoft, and TIBCO Software.

Similar to the previous market analysis, this research does not aim to provide a judgment about the offerings of any of the companies listed above. The objective is to obtain an objective overview of the features these iPaaS solutions offer, and how these relate to features also offered in the data catalog, as previously identified in Table 3. All vendors have been evaluated based on their websites and the documentation of their products.

Table 12: A summary of the features of iPaaS platforms, based on the inclusion criteria of the Gartner Magic Quadrant for Enterprise Integration Platform as a Service report [47] *and the commonly found features of the analyzed vendors*

Feature	Description
	Inclusion criteria of Gartner[11]
Cloud service	Offered as a cloud service, accessible through the internet. Offers scaling without interruption of client activities and some degree of resource tracking
Full cloud management	The platform should handle procurement and management regarding virtual and physical machines without the client having to be aware of them. The platform also ensures patching and the health of their platform
Application integration features	The platform is able to enable different applications to exchange messages at the transaction level. It enables data consistency and the composition of new services using existing applications or services
Supports industry-standard data exchange formats and protocols	The platform supports key standards for adopting Business to Business (B2B) integrations, such as EDIFACT
Multiple integration pattern support	Supports multiple integration patterns (such as messaging and API, among others)
Out-of-the-box connectors	Offers readily-configured connectors for many widely adopted software programs and technologies. Include applications such as Salesforce and SAP, database connectors for (No)SQL databases, and technology connectors such as FTP and HTTP
Data quality and modification tools	Offers validation, mapping, and transformation options for messages. Can be extended by a data quality dashboard
SLA & disaster recovery	Ensures a very high uptime to prevent disruption of critical business processes. Offers a backup to ensure uptime in case of issues with virtual or physical machines
	Other common features

[11] These features are found in all of the iPaaS vendors since it is a criterium to be included in the Magic Quadrant. These features are therefore not be evaluated individually in the comparison table.

Low-code User Interface (UI)	Features its platform in a visual, drag & drop UI, to improve the accessibility for non-IT people
Machine Learning	Features automated guidance, enrichment of data, or other features by using ML technologies
Robotic Process Automation (RPA)	Includes tools to automate processes to prevent repetitive manual work
Store with prebuilt templates	A way of accessing often-used integrations from widely adopted software applications. Reduces the need for every client to build all of their integrations and automation from scratch

In Figure 12, the Magic Quadrant for the Enterprise iPaaS solutions is shown. At first glance, a few of the names can be recognized from the market analysis of data catalog vendors in section 2.1.2.3. Vendors which are in both the Magic Quadrant of Figure 5 as well as Figure 12 are Informatica, SAP, Oracle, and IBM, although IBM is not a leader in the iPaaS space, according to Gartner's figure. Boomi, which is a leader in the Magic Quadrant for iPaaS solutions, also offers a data catalog as a separate part of their offering, which they acquired through acquisition at the beginning of 2020. A similar offering is made by TIBCO Software, which also has a data catalog in its offering. It is important to note that one of the requirements of Gartner state that "*vendors must offer stand-alone packaged software tools or cloud-based software (that is, not only embedded in, or dependent on, other products and services)"* [19]. Since the products of these vendors can be purchased separately, this indicates that there are differences in the products. The number of vendors offering both an iPaaS as well as a data catalog is expected to indicate the relevance of both products for the market, and that there is demand from users to use both. Therefore, a combination of the data catalog features into an iPaaS could potentially be confirmed to be useful.

In the market analysis is found that some of the features of iPaaS platforms are also in the data catalog package of some of the vendors, as identified by the market analysis of the data catalog vendors previously. If this is the case, it is indicated clearly. It is important to note that there are significant reasons to choose one vendor over another, to prevent vendor lock-in or due to licensing costs. Unfortunately, these factors cannot be taken into account as pricing is generally very untransparent if it can be found at all, and no trustworthy independent sources upon vendor lock-in can be found.

Figure 12: Gartner's Magic Quadrant for Enterprise Integration Platform as a Service [47]

Boomi

Boomi acquired its data cataloging capabilities through the acquisition of Unifi at the beginning of 2020. They included these functionalities within their 'Boomi platform' as the *Boomi Data Catalog and Preparation*. Since Boomi was not included in the Gartner Magic Quadrant for Metadata Management Solutions, it is not a guarantee that their data cataloging capabilities are offered as a separate service. This could be caused by the acquisition of Unifi being in the same year as the Magic Quadrant, since 2020 was the last year that a Magic Quadrant for Metadata Management Solutions was released, and it is currently being transitioned to *Active* Metadata Management.

Table 13: An overview of the features and USPs for each of the 'leader' iPaaS vendors

Vendor	**Boomi**	**Informatica**	**Microsoft**	**MuleSoft**
Product name	Atomsphere Platform	Intelligent Cloud Integration Services	Azure Logic Apps	AnyPoint Platform
	Features			
Low-code User Interface (UI)	✓	✓	✓	✓
Machine Learning	✓	✓		
Robotic Process Automation (RPA)	✓		✓	
Store with prebuilt templates			✓	✓
Platform based on open-source software				✓
Vendor also offers a data catalog[12]	✓	✓	✓	

Vendor	**Oracle**	**SAP**	**TIBCO**	**Workato**
Product name	Oracle Integration	SAP Integration Suite	Cloud Integration	Workato
	Features			
Low-code User Interface (UI)	✓	✓	✓	✓
Machine Learning	✓	✓	✓	
Robotic Process Automation (RPA)	✓		✓	✓

[12] Refers to a data catalog providing at least the *core* functionality as identified in Table 4

Store with prebuilt templates	✓	✓	✓	✓
Platform based on open-source software			✓	
Vendor also offers a data catalog[12]	✓	✓	✓	

3.4.1. Overlap in vendors' offering of data catalog and iPaaS solutions

In Table 13, it stands out that six of the eight vendors who are a leader in the iPaaS space, also offer a data catalog feature. When the two Magic Quadrants (Figure 5 and Figure 12) are compared, 4 vendors are included in both Gartner quadrants, indicating that they offer products with significant usage and corresponding to most of the features identified for both data catalogs and iPaaS solutions in this research. Focusing on all seventeen vendors in the Magic Quadrant of Figure 12, regardless of the *Quadrant* they are placed within, there are 17 vendors in the quadrant of whom seven offer both an iPaaS and a data catalog[12]. On the other hand, of the seventeen vendors in the Magic Quadrant of Figure 5, five of them offer an iPaaS next to their data catalog solutions. Some of the vendors in both quadrants who do not offer both products do have connectors or collaborations with other vendors to offer both products. A full overview of each of the offerings of each of the vendors, and the product names of their products, is shown in Appendix B.

The overlap in vendors offering both a data catalog as well as an iPaaS indicates a significant market demand for them. However, since only very large vendors offer both products, this might indicate that either significant resources are needed to develop and maintain such platforms, or that specific expertise is needed.

Even focusing on the vendors that offer both solutions, they do not have any focus on the research objective of this research, which is to use the data catalog features to help the iPaaS users. Whereas this could demonstrate the relevance of including a *business user* who might use the iPaaS offering for some of the features that they would normally find in a data catalog, it should still be investigated whether offering data catalog features in an iPaaS would be of similar benefit to a business user as having a dedicated data catalog. For the vendors which offered both, the features of data catalogs focused on aiding the business user, and have a different pricing model than the iPaaS. Although prices are not presented transparently, it was generally found that iPaaS solutions are considerably more expensive than data catalogs, but need to be rolled out to fewer users. Licensing methods of iPaaS platforms may vary whether they are on a per-user basis or in user tiers (for example 1 to 10 users) or uncapped, whereas data catalogs are mostly offered on basis of the number of users.

Since the marketing of both products focuses on two separate user groups, and benefits that are unique to each of these groups, it can be argued that all of these vendors do not offer an integrated solution between the data catalog and the iPaaS, where data catalog features are integrated into the iPaaS to combine the benefits of both. The definition of *integration* here has to be clarified further: it is likely that extending an existing iPaaS, of a vendor who offers both products, with a data catalog would be easier than adding a data catalog of another vendor to the landscape in which an iPaaS already exist. Therefore, if a single vendor offers both products, it might be easier to extend one product with another in the same software suite. However, adding a data catalog to an iPaaS within the same suite does not provide the iPaaS users with benefits within their iPaaS.

From a commercial perspective, being able to upsell a second product such as a data catalog to a client who already uses the iPaaS is an easy way of gaining revenue by adding a large number of business user subscriptions and committing a client to your applications. On the other hand, the fact that only very large vendors have been found to offer both products indicates that detailed knowledge and substantial effort are needed to successfully develop and market both products.

Therefore, this is not expected to be viable for smaller companies offering either an iPaaS or a data catalog.

3.4.2. Feature overlap between data catalog and iPaaS

Based on the combinations of the features identified for data catalogs in Table 4 and iPaaS platforms in Table 12, it can be investigated whether the two products have some overlap within their common features. At first glance, there is overlap in three features: connectors, data quality, and Machine Learning. There are some significant differences between these features, even though their names are similar. For connectors, these are mostly the same. One of the differences is that the connectors for a data catalog mostly rely on having to read the (meta)data in the application, whereas the connector for an iPaaS has to enable two-way communication (send and receive data). For data quality, the data catalog mainly aims at giving the users guides towards ensuring that the data they would like to use is of proper quality, whereas within an iPaaS this is a toolkit for ensuring data validation and transformation, and in some cases, this feature is extended with a dashboard in a style similar to what one might expect within a data catalog. Finally, as mentioned previously, Machine Learning is a supporting feature, which does not contribute as a single feature but as a set of supporting features. It can therefore not be compared between data catalogs and an iPaaS.

To find possible overlap in other features, all features of both data catalogs, as well as iPaaS, are combined into one table overview, and their names are generalized. It is then evaluated whether each feature is in the products. The result of this is shown in Appendix C. This illustrates that there is overlap in many of the features of both products when comparing them on basis of how different vendors name their features. When going into more detail with regard to each feature and what is expected of them in an enterprise data catalog and an iPaaS, more differences are found.

3.5. Architecture

In this section, The Open Group Architecture Framework (TOGAF) is used to develop baseline architectures of both a data catalog as well as an iPaaS platform. These provide a high-level comparison of the architecture of both products so possible similarities can be discovered. For the development of the architecture, the ArchiMate language is used. This language is closely related to the TOGAF framework, as they are both managed by the Open Group. Not all phases of the TOGAF framework are needed, as the focus for this section is primarily on the architecture of the business actors and processes, their interactions with the application, and the technology needed to support this. This corresponds to TOGAF phases B, C, and D, corresponding with the business, application, and technology layer as outlined in Figure 13. Using ArchiMate, these focus areas can be modeled clearly. ArchiMate is intended to give a high-level overview of an IT landscape and therefore does not necessarily include a high level of detail such as is given in other languages, such as BPMN.

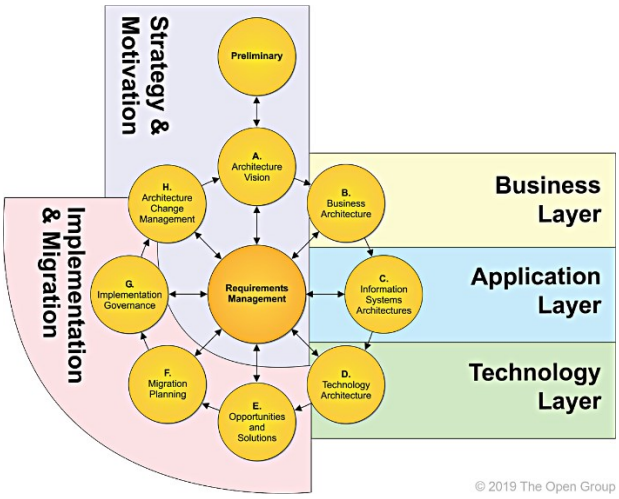

Figure 13: Overlap between TOGAF and ArchiMate [51]

3.5.1. ArchiMate

This section gives a brief summary of what the ArchiMate framework does and does not cover, and how it is used in this section. Although it is not mandatory within the modeling language, ArchiMate is split into elements that belong to different layers. The core layers consist of the business, application, and technical layers. These layers are often shown in a bottom-down visual, where, similarly to how it is shown in Figure 13, each of the layers is shown with business elements on the top, followed by application elements, and technology elements on the bottom. For even more clarity, colors fixed colors are used for each of the processes: yellow for business elements, blue for application elements, and green for technology elements. The diagrams created in this section aim to abide by these principles.

3.5.1.1. Business layer

The business layer supports the modeling of processes, persons, roles, and objects which are relevant on a business level, in a technology-neutral fashion. This helps in modeling the behavior of actors. For the models which are created for this research, the level of detail in the business layer is limited to those roles and processes which have immediate interaction with either the iPaaS or the data catalog. Since these products can be rolled out in a wide array of organizations, and it is not possible to generalize all possible scenarios, the business layer could be extended considerably given organizational context.

3.5.1.2. Application layer

This layer can be used to model the features of the system, as well as how it interacts with business processes and on which technology it depends. This layer gets the primary focus on both the architecture of a generalized data catalog as well as the iPaaS. It can show the individual processes and functions within an application, how different applications relate to each other, and which data objects are produced and consumed.

3.5.1.3. Technology layer

The technology layer refers to virtual or physical devices that are used to run the application. This can be used to show which (cloud) dependencies are needed in order for an application or for a multitude of applications to run. This helps in determining which technological demands a certain integration has, and which in technology are needed for adding or changing a feature and therefore giving a clearer view of the effects of up or downscaling on the needed architecture. Technology layer elements are generally connected to the application layer.

3.5.2. Baseline architectures

In Figure 14 and Figure 15, the baseline architectures of a data catalog and an iPaaS solution are shown. Since this research focuses on the features of the applications, and not on the ways in which the different connectors may interact with the different devices or software on the technology layer, this is not modeled in the iPaaS diagram. In addition, the generalized approach of the architecture does not allow for too much detailing of different processes, since each vendor might have their own unique touches on this topic. This also holds for the level of detail of the technology layer.

For the data catalog diagram in Figure 14, the technology has been simplified so the cataloging features are not placed within a specific organizational environment, with regards to its technology layer or connections to other applications, but provide a generalized overview of the data catalog as an isolated application, and the technology elements it can connect to. The application features are included as they have previously been identified in Table 4.

As for the iPaaS reference architecture, a similar approach is taken. In addition, only primary functional requirements have been included. Requirements such as support for a certain standard, and ways in which security measures are included, are not shown in the architecture. Also, the base technology elements needed are shown, based on a cloud application of the platform, which does include some security elements such as the isolation of a client's environment through the usage of a Virtual Private Cloud (VPC).

Both of the architectures can have significantly more business layer elements, including processes and benefits of both products that are not included now. The architectures do not include these at this point, since their aim is to provide an overview of the main functions of both products and not a full overview of their benefits.

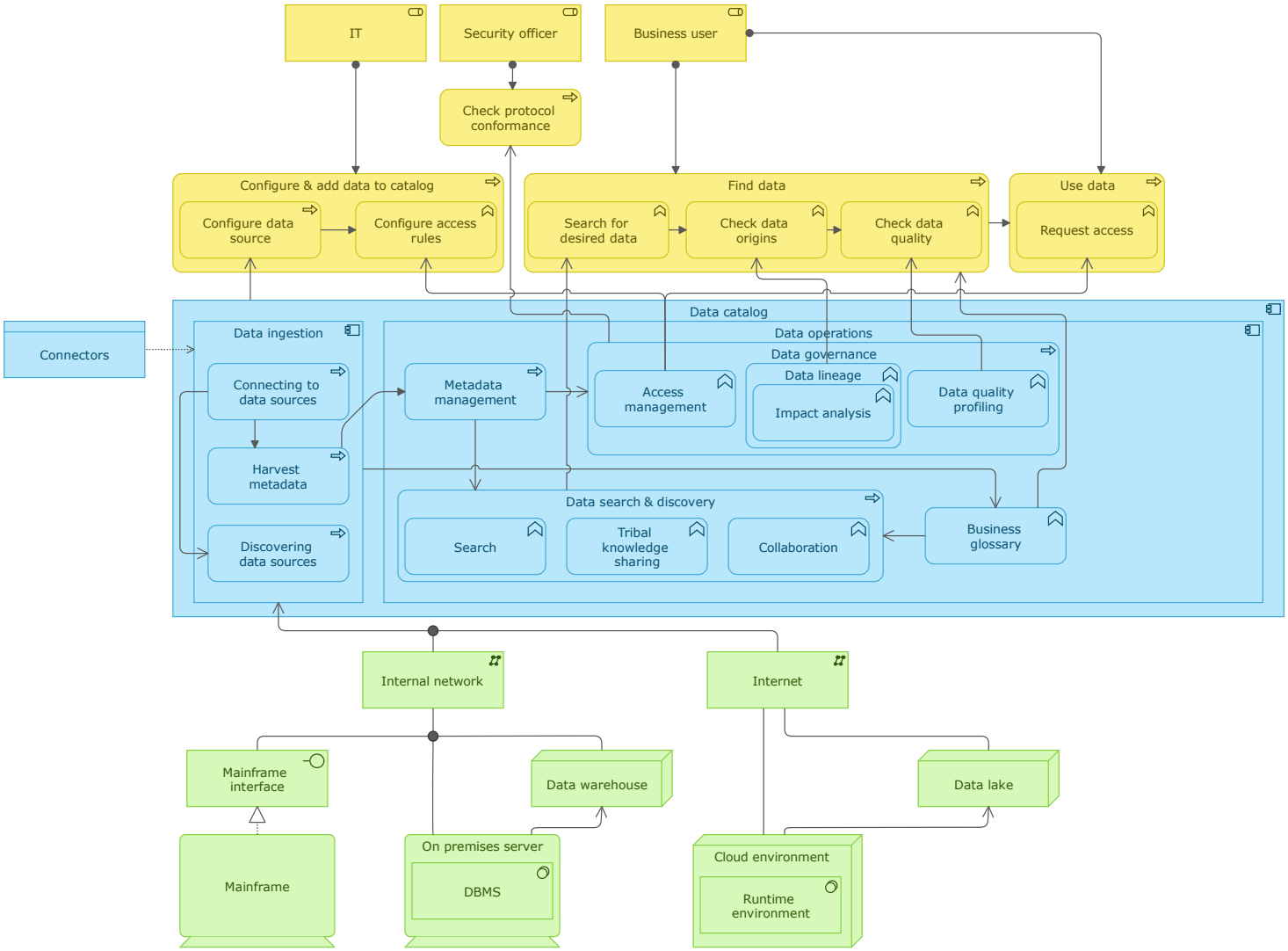

Figure 14: Simplified general architecture of a data catalog

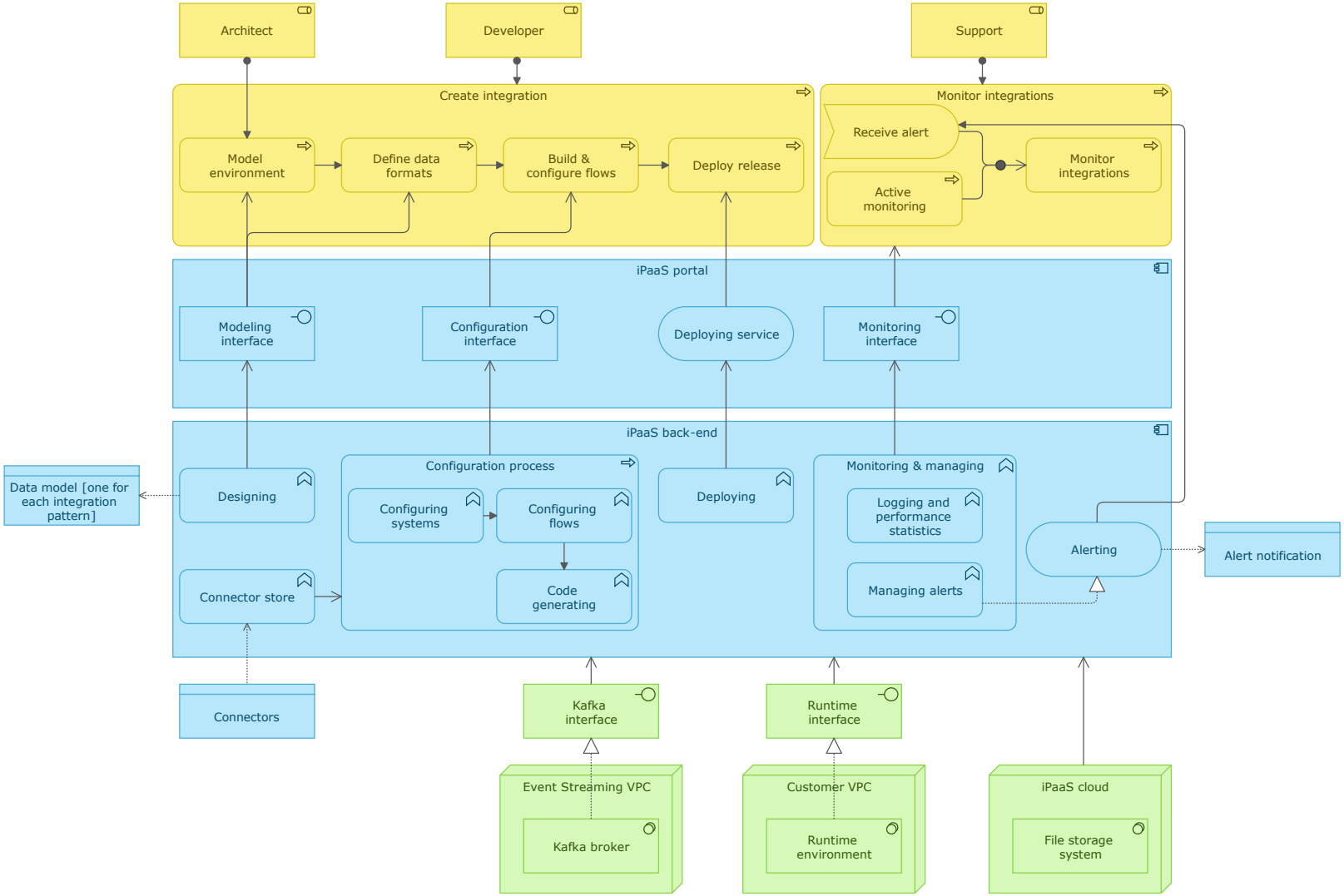

Figure 15: Simplified architecture of an iPaaS platform

3.5.2.1. Overlap between the architectures

At first glance, the two architectures belong to two separate products. The only identical component standing out is the connectors. Analyzing the architectures in more detail, more similarities can be discovered.

Some of the feature groups of the data catalog, such as metadata management, and elements of the governance, reside within one of the feature groups of the data catalog. Since these features are more of a supporting feature rather than a feature that can be used as a selling point of the platform, they are not included in both architectures. Overlap exists within:

- **Metadata management:** Although this is the main feature of a data catalog, is one of the features of the designing feature of an iPaaS. Within the design phase of an iPaaS, a system can be added and information is left behind regarding the definition and the usage of this system. Similar information is given about flows from system to iPaaS or from iPaaS to system. Currently, the degree to which this information is used in later stages of the iPaaS platform varies, whereas a data catalog relies upon this core information to build its catalog
- **Data governance – Access management**: Access management is a feature within a data catalog that can be highly detailed, with workflows through which a user can request access to a specific resource, which results in the data owner getting a request to assess the request. For an iPaaS, this level of detail is different. Within the *iPaaS backend*, each user is assigned access to which services and interfaces of the front end can be used. Therefore, access control is included, but it is not nearly as fine-grained as it is for a catalog. This is because of one of the core differences in the target user group, and the type of data that is processed in both applications. Most iPaaS platforms are a processor of data that flows over them, not a way to access data that exists in the organization. This gives a lesser need to restrict access to resources, as the resources are only accessible on a high level: the source and target system can be identified, and the way the message looks, with perhaps an example message. Additionally, restricting users of an iPaaS also restricts what applications they can develop, increases the likelihood of duplicate applications, and decreases productivity since there is more dependency on other team members.
- **Collaboration:** this feature group is embedded in an iPaaS since it is developed as a tool for teams to work in. Although they do not include all imaginable collaboration tools, and as with any development team, other tools are used to aid in prioritizing what to work on next, this is not necessarily something that is missing. For a data catalog, adding collaboration features is a way to improve data quality and contribute to glossary entries.

3.5.2.2. Differences between the architectures

This section addresses the differences which can be recognized in both the architectures. It also illustrates whether this is something that would be beneficial to include in the iPaaS or whether this is a feature that should not be included because of the differences in the expected behavior of the products.

- **Data ingestion as a separate application process:** Within a data catalog, one of its core features, isolated as a sub-application in Figure 14, is data ingestion. The data catalog has to know all data that resides in the organization in order to enable it to find more data and catalog it. This is a fundamental difference with iPaaS platforms, which only need to integrate the systems that need integration, and only extract data from these systems and provide it to these systems when it is needed. An iPaaS does and should not extract all

possible data from a system, since it is not intended to store metadata about this data, but it is designed to exchange data when needed. With the focus of an iPaaS often being on applications rather than databases and data warehouses, it is generally known which information can be extracted from the system since this is provided in the documentation of the applications. Additionally, the organization also knows the information they enter into the system. For this reason, other features within the data ingestion application, such as data harvesting and data source discovery are not needed within a data catalog either.

- **Data lineage:** all data catalogs feature lineage features within their application. Looking at the previous difference regarding data ingestion, and the data catalog including data discovery features, which might mean that data is shown in the catalog which not all employees would be able to find otherwise, makes this lineage feature especially useful. Even though the origins of data within an iPaaS are better known, and there is no automated discovery, an iPaaS can still lack an overview of where data is used. Although iPaaS platforms do clearly illustrate the data origins, a more advanced data lineage, which shows how data flows through a system and where it is edited, is lacking. Since iPaaS platforms are often used at an enterprise level, where large numbers of systems need integrations, a lack of overview can start occurring. Therefore, iPaaS platforms are expected to benefit from data lineage on entity level. This also helps in fulfilling enterprise expectations, such as being able to identify which system uses certain sensitive data.
- **Glossary:** as previously discussed, a glossary is currently not included in iPaaS platforms. This is expected to provide benefit to an iPaaS, in combination with the metadata management features to which it is highly related.

3.5.3. Target architecture

As discussed in the previous sections, the target architecture is that of an iPaaS as is presented in Figure 15 with certain functions of a data catalog within it. The current functions that are included are based on the findings of the relevant features to apply in an iPaaS, as outlined previously in Table 8. It also shows the overlap of features in the iPaaS architecture which were not recognizable from the baseline architecture yet. The target architecture provides a reference as to what implementing the suggested features into an iPaaS would look like. In Figure 16 and Figure 17 the ArchiMate diagrams of the target architecture are shown. This illustrates that the features would reside within the iPaaS backend and would offer support for multiple features. Figure 17 shows the features that are built in the prototype which is introduced in more detail in the next chapter. In this figure, the additions which are included in the prototype are highlighted.

As illustrated, the discovery from the log messages is excluded for the prototype. This is chosen since the initial aim of this research is to improve the usability of the platform for the end-user. These are users of the customers of the iPaaS. It is identified that the support staff, which is part of the vendor, can also benefit from some of these features, but this is more suitable to investigate in further research. In addition, excluding this feature, for now, improves the validation of the prototype with experts. The concise steps for the validation are shown in Chapter 5 and are not discussed in detail in this chapter. Since the *log and error discovery* is the only feature that relates to the *support* actor, however, including this feature in the prototype would mean that experts in the role of support are also needed to be taken into account for the validation. This would essentially mean a third set of experts.

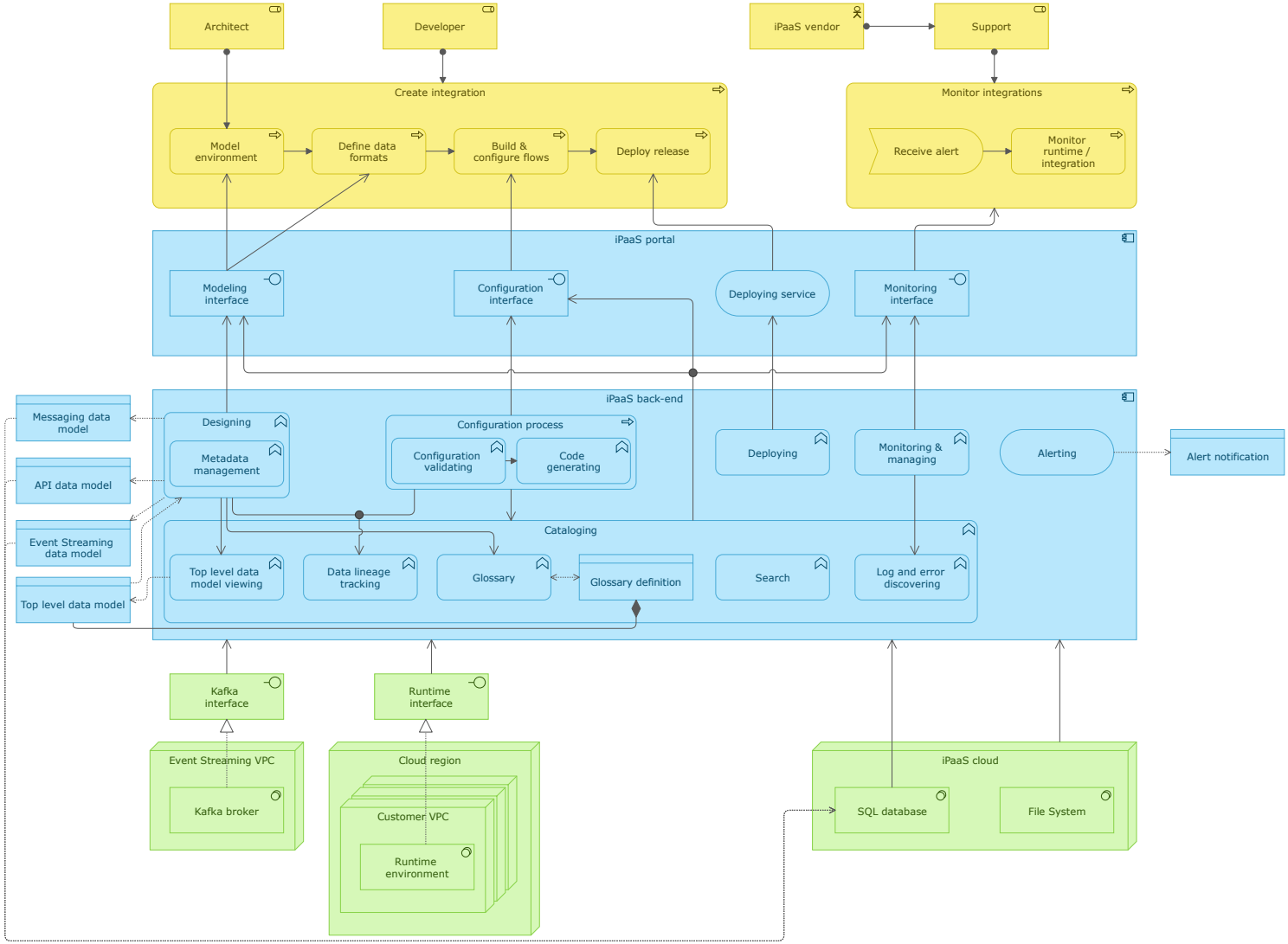

Figure 16: Target architecture of an iPaaS with enterprise data catalog features

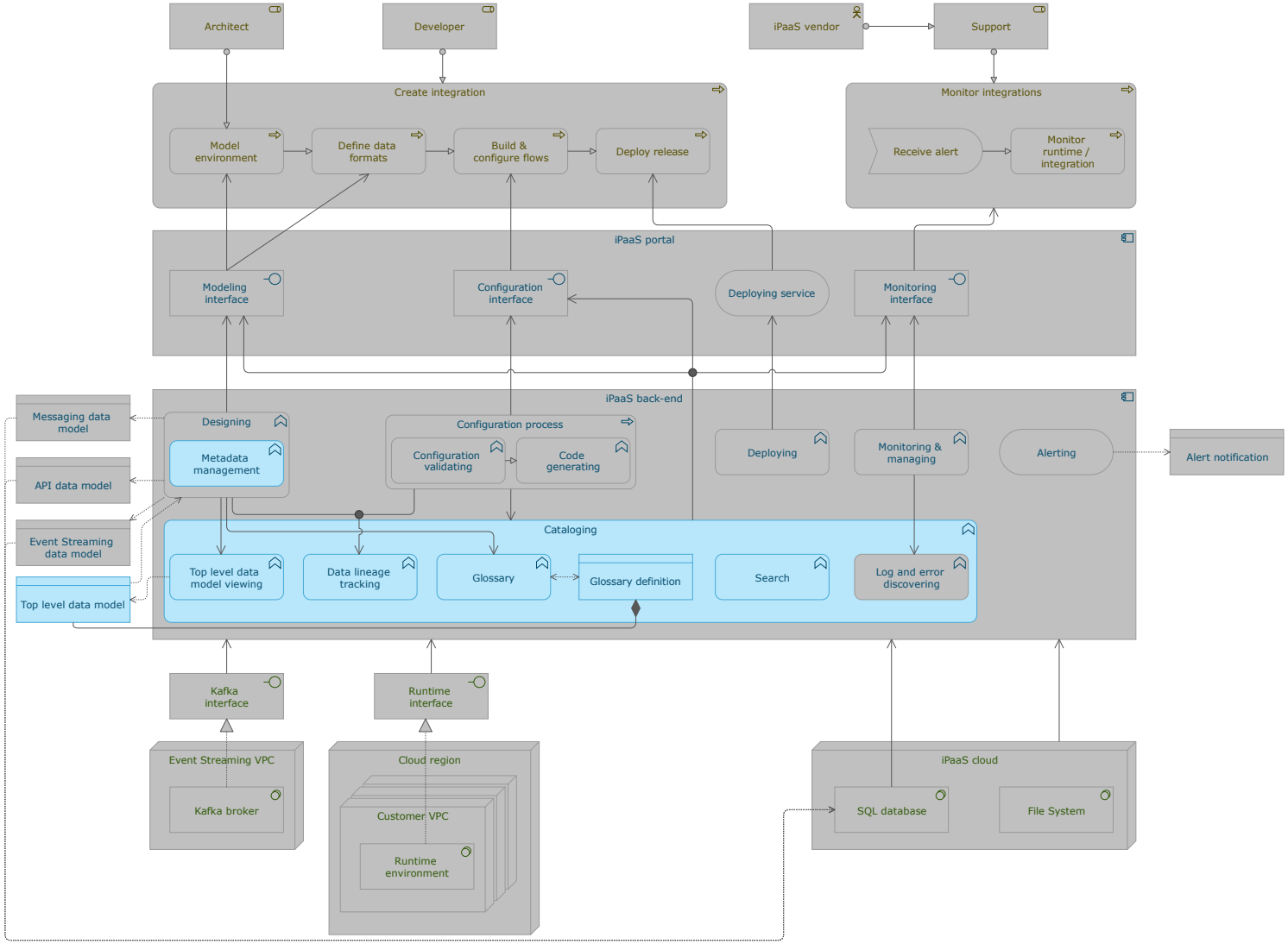

Figure 17: Target architecture, showing the features which are evaluated in the prototype

3.6. iPaaS used throughout this research

This research focuses on the combination of features of a data catalog into an iPaaS. As this research produces a design of features and a validation of their fit into an iPaaS, it is not realistic to evaluate this for the iPaaS products of a large number of vendors, due to differences in the feature offering as well as differences in scope and focus of different vendors. Therefore, the iPaaS product of one vendor is picked, and the prototype is built-in and tailored to, their platform. The findings of the implementation of these features into this iPaaS provide valuable information regarding the fit of the selected features of the prototype. Thus, although the features are developed in a single iPaaS, the findings are generalized for the application of these features into any iPaaS.

In previous sections, it was shown that although there are vendors which offer both a data catalog as well as an iPaaS, this does not mean that the data catalog features that the experts which are interviewed in section 2.3 found to be relevant for an iPaaS (Table 8) are also already included in the iPaaS of that same vendor (section 3.4.2).

The iPaaS which is used throughout this research is the same as the one used by the interviewees in section 2.3.

3.7. Framework

The contents of sections 3.1 through 3.3 and 3.5 provide the steps needed to identify the features which are relevant to evaluate for the specific iPaaS used in this research, which is introduced in the section above. These steps can be repeated for any other given iPaaS, to derive from which features that iPaaS benefits. This section provides a framework that can be used by any iPaaS vendor to identify from which enterprise data catalog features their platform would benefit.

In the previous chapter, the literature review resulted in a list of features as expected in enterprise data catalogs. This list was then cross-referenced in an expert consultation with an expert panel consisting of stakeholders from an iPaaS vendor as well as experienced users of that iPaaS. The features of the enterprise data catalogs are described in Table 3. Their relevance to an iPaaS is given in Table 8.

Each of the enterprise data catalog features can affect all of the stakeholders of an iPaaS in different ways. For example, impact analysis would be most helpful for the integration developer, as it helps this actor in identifying other changes that are needed as a result of the change they intend to do. A business glossary would be helpful for the support staff and the integration developer, to ensure that they understand the business context of what each integration does. Governance features would primarily be used by the business user, such as an information security officer of an enterprise, and the enterprise architect who needs to apply these standards to the IT systems. In this way, each feature holds relevance for most of the stakeholders but to different extent.

3.7.1. Applying the framework

It is difficult to know which features are needed for a given iPaaS before first conducting a proper analysis. This framework is intended for vendors of an iPaaS, and to give them a means to find the most relevant features for their platform.

Before starting the analysis, it is important to first have a clear view of what each of the features of an enterprise data catalog is, and how these can be implemented into an iPaaS. Where Table 3 gives an overview of each of the data catalog features with a description, Table 14 describes each of the features with their relevance to be applied within an iPaaS.

Table 14: All features of enterprise data catalogs which are relevant for an iPaaS, described for their relevance within an iPaaS

Feature	Description
Access management	Ensures that the users of the platform can only access and change what they are allowed to change based on their role. Helps enforcing organizational governance.
Business glossary	Since not all users of an iPaaS necessarily have domain knowledge of the customer of the iPaaS, a glossary is a central place within the platform which acts as a dictionary for all users, to align terminology throughout the platform.
Collaboration	Functionality that enables the users to communicate with other users, by, for example, leaving comments and placing tags on areas in the platform where the user feels extra clarification is needed for future reference.
Data lineage	Ensures that the origins of data is known, and producing and consuming systems can be easily separated. Aids in governance, by providing a means to show where data flows, and provides a foundation for impact analysis by showing systems and integrations which might be impacted by a change.
Data quality profiling	Shows metrics to help determine the quality of (meta)data.
Data search & discovery	Search: Enables the user to find any system, integration, or feature of the platform from a single place. Discovery: Helps the user in identifying data flows which contain more or less information than recorded in the system. Can be useful to identify where to find currently missing data, or to extend functionality to other systems.
Governance	Provides means for the organization to ensure that information sharing adheres to the organizational policies. 2-factor authentication, access rules and obfuscating sensitive data fields in logs are examples of governance features.
Impact analysis	Provides information on the effects of changing a data source, on other data sources which depend on it
Metadata management	Documents and manages metadata as the core of the system. Enables browsing of metadata. Includes means for interacting and harvesting metadata
Top-level data model	The top-level data model is an aggregation of all the data objects that can be found within the platform for a single customer. This helps the user in getting an overview of the data objects used, which can be cross-referenced with the organizational structure, which helps to recognize inconsistencies in the model and which enables better alignment of the naming of objects

Now that a clear definition of each of the features which might be relevant to an iPaaS, some steps are needed to apply it. To do so, the following plan can be followed. This is the same structure as how this research has been done.

1. **Analyze:** it is important to know the current position of the iPaaS. Use Table 14 above to discuss internally which of these features might already be covered by the platform, as to rate each of the features which are worth investigating developing into the platform.
2. **Refine**: make a selection of features to evaluate with users. Evaluating all features at the same time is time-consuming and lacks detail which can make it difficult to find users willing to participate in an interview session. To identify how this shortlist of features can solve the problems of the customer, create exploratory questions to identify how each of these features might impact the end-user. Select relevant users of the platform who are targeted by potentially adding these features. For example, if the objective is to better serve clients with complex data models, make a selection of users of clients serving this criterium.
3. **Discuss:** Set up interview sessions with the selected users to identify which features would provide the most added value. It is recommended to use a semi-structured interview format to obtain the most information out of the interview sessions. An example of such a format as used in this research is provided in Appendix A.

4. **Tweak:** Process the findings of each interview to be able to score each feature for the most relevance. Use this step to provide a set of requirements for each of the features.
5. **Prototype**: Create a prototype that contains the findings of the result. This prototype can be of high fidelity, to enable a more swift interaction with the stakeholders. After this step, circle back to step 3 to validate the prototype.

After iterating through steps 3-5 at least once, a prototype with a feature set that fulfills most user requirements can be the result. At this stage, moving further to stage 5 is more user testing and tweaking than prototyping.

4. Prototype

In the previous chapter, requirements and goals have been identified for which a proper solution to the main research question should solve. These requirements are implemented into a prototype, which aims to provide a solution to the stakeholder's goals. As a methodology, the prototype is a part of the single case mechanism experience for the validation of this research [48]. This chapter focuses on the prototype which is developed to provide an artifact to provide a solution to the research problem. Because of time limitations, the prototype is a minimum viable product: it includes a selection of the features which were identified to be relevant to develop.

4.1. Method

Up until this point, this research has been following the Design Science Research Methodology [48]. This methodology does not include a method for developing a prototype. Therefore, a different methodology is used. Within software engineering, there are two main methodologies used to develop software. These are *Waterfall* and *Agile* (or *Incremental*) methodologies. Waterfall methodologies are a more traditional method of development, where each step in the process such as *Requirements*, *Design*, and *Implementation* are done in isolation from each other, and once a process is finished the next process is started. Waterfall-based software development requires a full understanding of the problem and releases the full solution after running through it once [48], [52]. Agile methodologies are more similar to the way in which humans solve problems in their day-to-day life. Rather than ensuring that the whole problem is understood in advance, smaller increments are made, and corrections can be made during the development [52]. Agile methods generally work with short, two-week cycles to solve one problem at a time [48]. This allows for easier accommodation of changes, more involvement of the client, and a faster delivery process [52].

The objective of this prototype is to build the prototype as part of a potential solution to the problem users of an iPaaS platform might experience. This prototype is developed as an extension of the product of an existing iPaaS vendor. To ensure the relevance of their platform as well as the delivery of a prototype that contains a relevant representation of the features, it is important to involve this vendor and to have an opportunity to implement their feedback. This inclusion of feedback is possible by using an agile development approach. Therefore, this is the approach that is used for the development of the prototype. More specifically, the scrum methodology is used [53]. This means that the requirements, as previously identified in chapter 3.3, are divided over *sprints*, which are short development periods, which aim at delivering a functioning subset of the prototype at the end of each *sprint*. Because of the limited time available for this research, the sprints are one week each. After each sprint, the deliverable is presented to stakeholders of the iPaaS vendor so their feedback can be taken into account. The first sprint is sprint 0, used by the researcher to familiarize themselves with the iPaaS which is used and to prepare it for the first sprint by adding basic data to a new project. A more detailed sprint planning, along with the expected feature-focus of the sprints is included in section 4.5.

4.2. Features

Based on the findings of the interviews among stakeholders of the selected iPaaS, four main features have been identified which are relevant to work on compared to the current state of this iPaaS. These features have previously been outlined in Table 9, and consist of *data lineage*, *glossary*, *search,* and a *top-level data model*. This section evaluates what each of these features does, and which functionalities they add to the platform. This is done in each of the subsections, in alphabetical order. These sections extend on the definitions previously given to these features in the data catalog context in Table 3 and tailor their description to the application within an iPaaS

4.2.1. Data lineage

Data lineage within the application of an iPaaS would mean that a certain data entity that flows over a (number of) system(s) can be traced back to its source system, and the changes which have been made to the data can be seen. Within the scope of an iPaaS, this provides two kinds of iPaaS users with insights:

- The developer can use this information to see which integrations are impacted when the data model is changed. This benefit overlaps with the impact analysis since data lineage is an enabler for impact analysis.
- A business user, in the role of the security officer, can monitor which applications use sensitive data, and which applications edit this.
- An architect and developer can recognize entities that are not used in any integration

For the prototype, the objective of the data lineage feature is that the user first selects an entity they are interested in evaluating. The platform then evaluates all integrations and checks where this entity is used. Based on what it finds, it shows all integrations which use this entity. In addition, this feature shows whether the system provides or receives the message. The overview, therefore, provides an overview of all systems with the kind of usage as well as all integrations for the selected entity. Figure 18 shows an example of how this data lineage could be shown. In this case, it shows that the SAP system produces *employee* entities, and the integration types of *Employee and Job Data* and *Holiday Info* are transferred to the Payroll and the Planning systems. The direction of the flow is not necessarily left to right but is indicated by the arrowheads. Everything within the light blue box are processes happening within the iPaaS. This means that the message is sent to the iPaaS by the SAP system, which then matches the message to be either of the type *Holiday Info* or *Employee and Job Data*. When this is done, it is sent to the appropriate target system. In this case, an additional transformation might be needed to tailor the output to the expected input of the target system. This is shown by the fact that a second dark blue block is shown.

Based on the information provided in Figure 18, a data security officer can see that there are 3 systems processing data about *Employees*: SAP, Payroll, and Planning system. They can also see that the data originates from SAP.

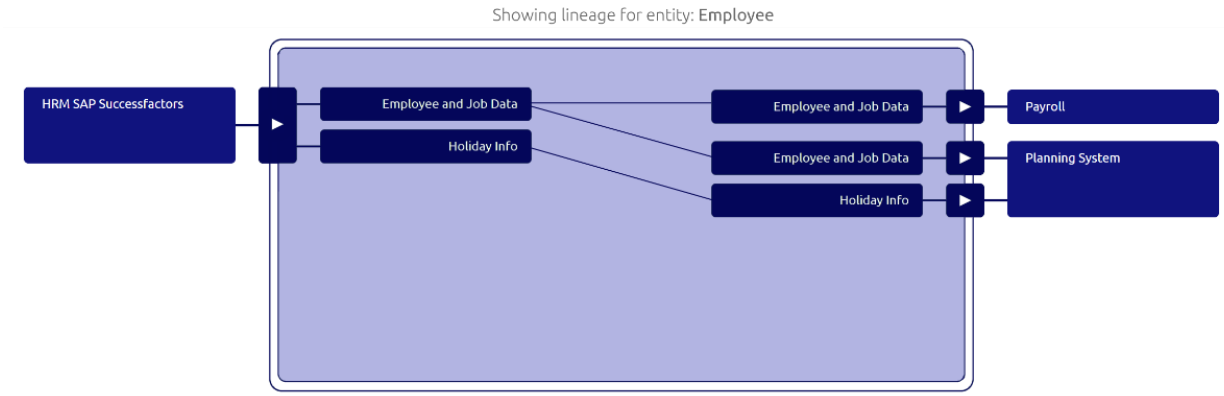

Figure 18: An example of how data lineage can be shown

In the shown example, it is quite straightforward that these integrations messages and systems process data about an employee because of the names given to each. A security officer would know that this data originates within the SAP system, and it can logically be derived that the payrolling system and planning system would need this information. For a larger landscape, however, some

choices which might not become obvious could occur. Take for example a boarding computer of a truck, which might be configured to require logging in with certain credentials. These credentials could be compared to the current active directory, which in turn might have built-in security which validates whether the employee connected to the account still has an active contract, to prevent employees without a contract from using a truck. Such a relationship between the active directory and SAP system might not be deferred from the context immediately. In large models, this information is even more likely to be overlooked.

4.2.2. Glossary

In contrast to the role of the *business glossary* in data catalogs, the focus of the glossary within an iPaaS is not only on mapping the system or integration to the business context, but also to ensure that this information is accessible to the user from any location in the platform.

Within the prototype, this is done by firstly formalizing the way in which information about a system or integration is documented. Currently, the iPaaS chose to do this primarily through the usage of open questions. The objective would be to provide more closed questions so that the responses can automatically be linked to certain properties. In addition, this helps the users of multiple models to have a more uniform way of recognizing the properties of a system or integration.

Based on this entered information, the platform can then show this entered information as read-only in any place in the platform where the system or integration is shown. This helps ensure that the information is always accessible to the user, rather than having it hidden away.

In addition, formalizing the information input facilitates the creation of new features, such as automated *tagging* of a system or integration based on their documentation. For example, if an integration is labeled as ‘processes personal data (as per GDPR)’, it could automatically be assigned a *sensitive* tag, which the user can then use to apply to only show those systems and integrations which process privacy-sensitive data.

4.2.3. Search

The general principle of the search functionality is clear: it should provide the user with a means to find any information entered into the platform and navigate to the position this information is shown when a result is clicked.

For the chosen iPaaS, there is currently no search functionality. In addition, it has a clear division in the lifecycle, which means that an integration is built by going through multiple steps. This means that one system can potentially be shown on multiple screens within the platform. Therefore, a search result such as ‘CRM [system]’ does not give a clear point to where the user wishes to navigate to. Is it the CRM system within the requirements capture phase, or the technical configuration of this system? Since the platform currently does not have a page on which both of these example data is shown, it should be made clear where the search result was found. An example of how this could be done is shown in the mockup in Figure 19.

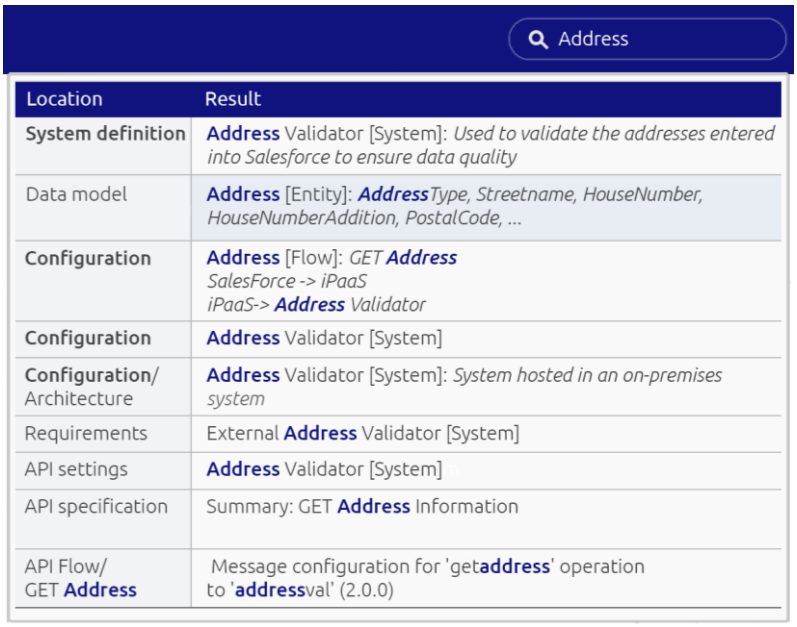

Figure 19: An example search and search result for the query 'Address'

4.2.4. Top-level data model

The last feature, but essential for the core objective of including data cataloging features within an iPaaS would be that of the top-level data model. This feature is an extension and abstraction of the data model(s) that the iPaaS already has. Its main objective is to provide a better overview of the data within the platform. It does so by enabling *zooming out* of the general data model(s). For example, a developer who wants to identify the core information which is processed by the platform does not need to see a full database-schema-like overview of all the data entities as they can pass through the platform. These models are generally quite large, and it might therefore be difficult for a user to find the most relevant information at a glance. An example of such a partial data model is shown in Figure 20. Each of the abstraction possibilities which is offered by the top-level data model feature is explained and mapped onto this figure. Finally, Figure 21 shows what this partial data model looks like after applying the abstractions.

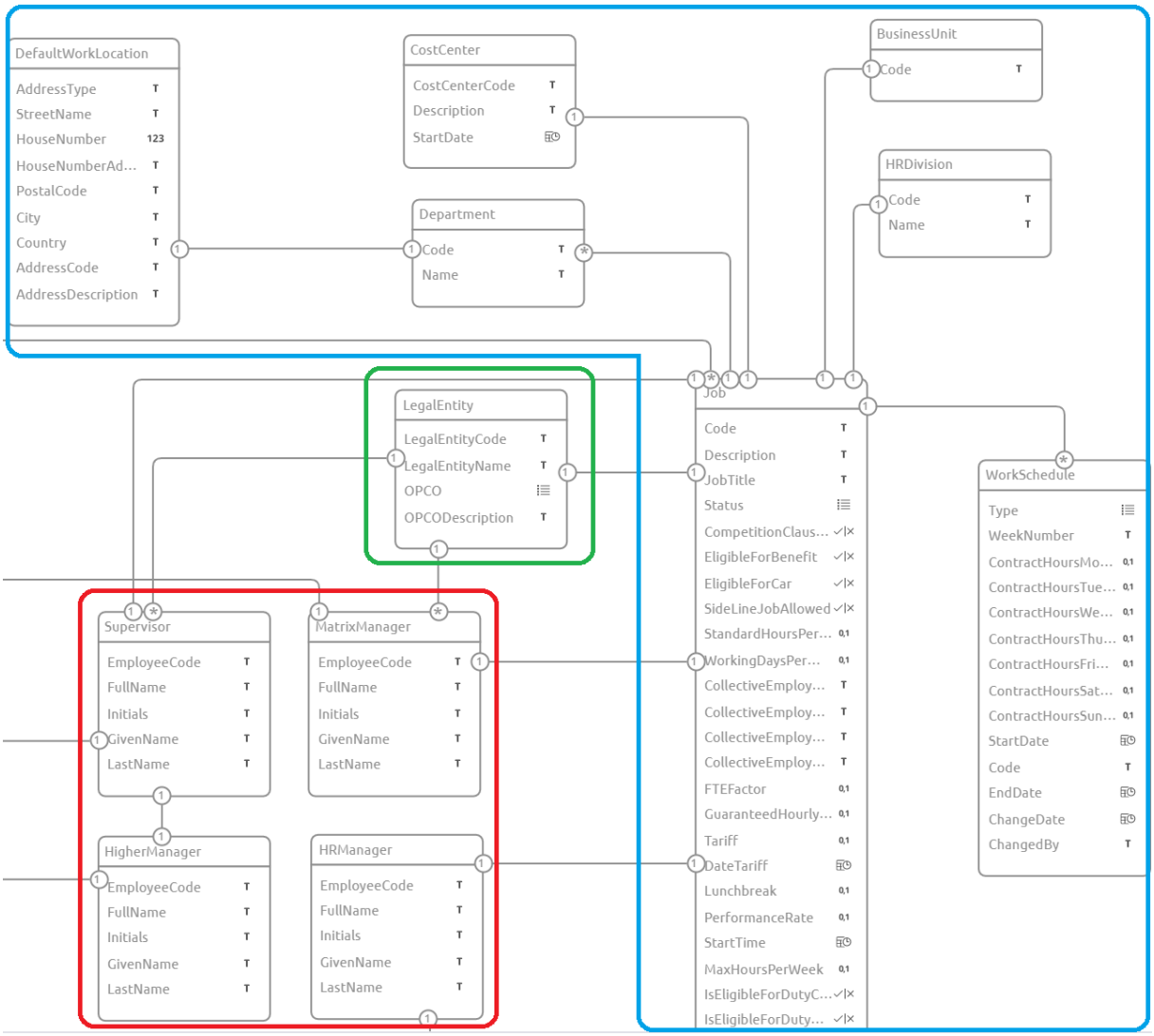

Figure 20: An extract of a data model, complete with entities, their attributes, and relationships, with entity groups color-coded for clarity

A method to make these large data models more readable is to remove redundant information. This can be done in a number of ways:

- **Remove detail**: As a first step, the attributes and their data types can be removed from the data model. This strongly impacts the screen space needed to show an entity, since an entity such as an employee with potentially dozen(s) of attributes only needs to show the name of the entity *Employee*, and the developer inherently understands that such an object would contain fields such as 'first name', etcetera. Also, this reduces the number of text lines needed to display the entity with its number of attributes.
- **Group entities based on relationship**: Within relational database models, which a data model can be compared with, relations between entities can exist. Generally, relationships such as a one-to-many relationship (e.g. one *Employee* can have multiple phone numbers) could be grouped together. At first glance, it might potentially not be beneficial to show these sub-entities belonging to one 'parent' entity. Therefore, within the top-level data model, the option to group these entities should be shown. This abstraction is applied to the blue entity group.

- **Group entities based on similarity**: Another abstraction could be made based on the similarity of the attributes of two entities. For data models which are not properly normalized, or which have other reasons to include multiple entities with almost exactly the same attributes, it might be beneficial to group the entities with the same attributes together. There are multiple ways to apply this. For the prototype, it only takes into account entities with a 100% match between their attribute names and types. For a final implementation, the user should be enabled to tailor the parameters to for example group entities as soon as their attributes overlap for a custom percentage. This abstraction is applied to the red entity group.
- **Adjust the shown entities**: The user should always have the opportunity to create a custom view. For example, this should enable them to show or hide an individual entity, or to choose to not group entities. This view should be remembered for every user, and they should be given the same view as they left it the previous time. This feature is not illustrated in Figure 21, but applying this for e.g. the entity 'WorkSchedule' would result in showing a fourth entity, with a relationship to job.

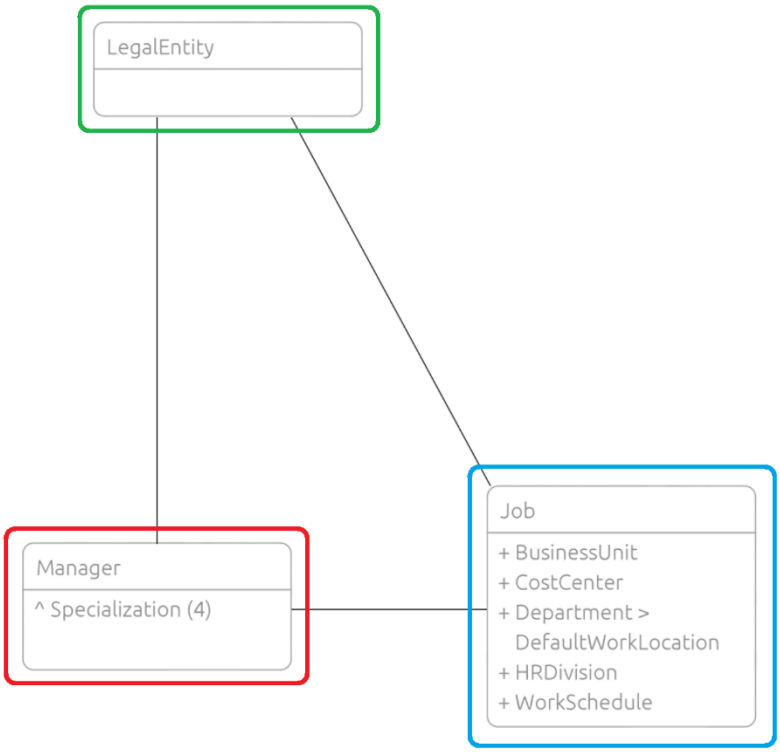

Figure 21: The same data model as shown before, with the described abstractions applied, changes color-coded for clarity

4.3. Focus

As indicated previously, not all identified features can be developed to their full extent cause of the time constraints. Therefore, a selection of the features is made. Although each of the features could be implemented as a stand-alone feature, the features combined are what makes a data catalog useful. Therefore, the benefit of adding data cataloging features is greatest when all features are added to the iPaaS. This can be recognized from Figure 6, which was shown earlier, where the relations of data catalog features to each other are shown. This figure shows that many of the features are interconnected. From this, it can be concluded that each feature by itself offers some benefits, but multiple features combined offer more benefits. A basis for selecting the features to be included in the prototype could be made on basis of the number of connections it has to other

features. One of these features with the largest number of outgoing connections in Figure 6 is the glossary. Therefore, a considerable benefit is expected to be gained by adding this feature. Similarly, the order of priority of the other features can be determined. These are the data lineage and the data search features. In addition, the data cataloging feature, which shows the data within the iPaaS is included in the prototype, as this is the core selling point of a data catalog even though it was not identified as a separate feature.

The order of priority for implementing the features is the following:

1. Glossary
2. Top-level data cataloging
3. Data lineage
4. Data search

The data search feature is given the lowest priority because the principle of search functionality is a feature that is already present in many applications, and even if the users are not familiar with those, they are familiar with searching using a search engine. Therefore, a fully functioning search algorithm is not needed for the user to have a good understanding of how such a search function should work. To still retain a complete overview, the prototype can show a search functionality as a visual feature, which is not built to show a working search but shows an example of how a search query could be shown.

In the architecture shown in Figure 17 of the previous section, these four features have also been highlighted before to show what their position within an iPaaS would be. This also shows one of the features which has been identified as a relevant feature, based on the interviews, but which is not included in the prototype. This is the data discovery, which the interviews indicated could work on basis of log and error entries, and could help the monitoring and management of the platform. Although this would indeed add new functionality to the iPaaS, this feature would be a stand-alone feature and its perceived use would be for a smaller group of users than the other 4 features, which would be in a more prominent position within the platform.

4.4. Platform

The prototype is built as an extension of the iPaaS product of the vendor used throughout this research. This has multiple benefits. Firstly, for the validation of the prototype, as is discussed in more detail in the next chapter, customers and employees of this iPaaS vendor are interviewed. Showing these stakeholders features within a platform that is already familiar to them, ensures that they can form their opinion based on the perceived use of the prototype, and reduces the need to imagine how the features could look like when included within their iPaaS.

In addition, developing the prototype on top of an existing iPaaS ensures that these features can actually be implemented within the chosen iPaaS, using the data which is available. Next to that, it also gives the opportunity to validate whether the features are placed in a suitable position within the platform, which can further confirm the relatedness of the features of the data catalog, as they were identified in Figure 6.

4.5. Sprint planning

As indicated previously, there are 6 sprints, including a preparatory ‘sprint 0’. Each of these sprints focuses on some of the features and has certain deliverables. The deliverables and focus points of each sprint are summarized in Table 15.

Table 15: Overview of the objectives of each sprint

Sprint	Objective	Deliverable
0	Prepare a project in the vendors' iPaaS, and install software and dependencies. Design and add a search bar to the platform	Locally running instance of the iPaaS, search bar and results box visual design, visible throughout the platform
1	Adjust the platform to provide a glossary, offering definitions and summaries to derive the criticality, data volume, and veracity of a system or a flow	Glossary feature applied throughout the platform
2	The first step of implementing top-level data view, by combining all data models into one page, and offering a top-level data model showing all entities of all data models	A single page for selecting data models, a first draft of the top-level data model
3	Completion of the top-level data view, creating an abstraction of all entities, to only show those with the most relevance	A fully working top-level data model, showing an abstracted view of all data models
4	Start with the lineage feature, focusing on selecting only a subset of systems based on the entity selected	Data lineage showing system usage
5	Continue lineage feature, focusing on showing symbols to illustrate what is being done with the data in each flow	Data lineage showing access/edit actions

This planning shows only those features that have been selected as features that is evaluated for the prototype. This is in line with what was shown previously in the target architecture of the prototype in Figure 17.

4.6. Sprint review and planning

Within the scrum software development methodology, each sprint ends with a *sprint review*, during which the deliverable of that sprint is discussed, and the *backlog*, which are the user stories that need to be developed, can be revised.

For this prototype, the reviews are conducted with two stakeholders of the vendors' iPaaS platform. During these reviews, the backlog can be changed based on progressive insight.

Each sprint review is followed by a sprint planning, in which the user stories for the next sprint are presented and adjusted where needed. During the review and planning, deviations from the planning as presented in Table 15 can be made.

5. Validation

The previous chapter introduced the prototype which contains the most important elements integrated into an existing iPaaS solution. This chapter provides a plan to validate the created prototype, to validate that the effects of the prototype when applied in practice are indeed those that are intended in the design.

5.1. Methodology

Most important for validating the artifact is determining on which dimensions validation is needed. The ISO 25010 standard, which was previously used in section 3.3.2 to determine the non-functional requirements for the prototype provides quality metrics for software and systems. It does so on two levels: *product quality*, focusing on design attributes, and *quality in use*, focusing on performance and usage attributes. Since the prototype is created in a stage prior to the implementation, the *quality in use* metrics are not relevant to evaluate for the validation of the prototype. Of the eight categories of quality metrics the ISO standard defines for product quality, this validation focuses on two of these categories: *functional suitability* and *usability*. The validation is conducted using the validation models of expert opinion.

Expert opinion validation is a qualitative validation model. This means that few experts are needed to obtain relevant responses. This does, however, make it important that the selected experts have a good understanding of the problem context, so they can predict the effects of the prototype when put into the intended context [48]. For each of the two aspects, a different group of experts is interviewed, with each group focusing on either the functional suitability aspects or the usability quality metrics. Functionalities of iPaaS offerings of two vendors as well as locations of certain functionalities in general affect which features are relevant for a certain iPaaS. The prototype has been developed on the basis of features that are relevant for this selected iPaaS, as shown in Table 9. In this way, the prototype provides answers to whether these four features of a data catalog are relevant for application in an iPaaS. To prevent *noise* in the validation results because of users expecting a different way in which core functionalities of the platform work, the experts of both expert opinion panels should already be familiar with the iPaaS which was used to develop the prototype.

A choice can be made upon whether to interview each of the experts individually or to interview them in small panels, with more than one expert in one interview session. Because of the qualitative aspect of the expert interview validation method, only a relatively small number of experts is needed in order to derive relevant conclusions. Therefore, each of the experts has been interviewed individually, to ensure that they can provide all input they deem relevant from their perspective.

As indicated previously, two panels of experts are used for the validation. These are the following:

- Functional aspects: internal stakeholders of the iPaaS, with a role in which they can affect which features are developed and released. These experts are working on a daily basis to develop or assess features and look into operational issues. Therefore, they are an expert on the iPaaS used to build the prototype and have significant insights into the strengths and weaknesses of their platform. Since these experts have experience in building, testing, and releasing features, they can give viable input on the relevance of the features developed, as well as how well it fits into their iPaaS application. To ensure a wide view, a broad array of experts was selected, including a product manager, CTO, and product owner.
- Usability aspects: users of the iPaaS. For this panel, a mixture of users who use the iPaaS on a daily basis and users who use it less often was chosen. In addition, this panel consists of

multiple user roles as they have been identified previously, such as that of developer and architect. Since an iPaaS may also be used by an integration consultant to (co-)build integrations for a client, some of the interviewees are developing integrations as an integration consultant. This expert panel is expected to give insights into the relevance of each of the features, and to be able to confirm the found relevance of each of the features.

Both expert interview panels follow a similar structured approach. With each of the experts, a one-hour session was scheduled with the following schedule:

- **Introduction** (10 minutes). The concepts of an enterprise data catalog are introduced to the expert since they might not have extensive knowledge of this yet. In addition, this interview presents which features are included in the prototype and why this selection was made.
- **Design** (10 minutes). During this part of the session, the experts are shown the prototype. If needed, the changes made compared to the unmodified iPaaS are highlighted. The visual changes, and their intended usage, other than those included in the minimum viable prototype, are verbally explained. The prototype is filled with example data, to mimic what a small customer environment may look like.
- **Feedback** (30 minutes). This final part of the session uses the interview protocols of Appendix D (functional expert panel) and Appendix E (usability expert panel) to obtain insights from the experts based on the prototype as it was shown to them. Both of the protocols follow a structured interview approach, to ensure that each of the experts is asked the same questions and in the same order. As shown in the interview protocols, the first questions consist of some questions to investigate the role and seniority of the experts, after which more feature-specific questions are asked.

During the interviews, notes are made of the answers of the expert. Immediately after the interviews, these notes are written out with a summary of the answers to each question. After all expert interviews of a panel, its results are compared to see overlap and differences in the answers of the experts. These are then summarized in the next section.

5.2. Results

This results section first addresses the expert opinion panel on functional aspects, followed by the expert opinion panel on usability aspects. The full results of each individual interview can be found in Appendix D.1 to Appendix D.4 (functionality) and Appendix E.1 to Appendix E.4 (usability).

5.2.1. Functional expert interview findings

The findings of the functional expert interview were based on stakeholders on the side of the iPaaS which was used throughout this research. Their roles included those of CTO, product manager, software delivery manager, and product owner. These roles all have a mature view of the current state of the product, as well as the direction they want to move in. The point of view of the CTO ensures that the proposed features are also technically feasible, the product manager is highly aware of the demands of the customers regarding desired functionality of the iPaaS, and finally, the software delivery manager and product owner are very well aware of the current roadmap and how well the features with in the platform. In addition, each of the experts can help in obtaining input regarding the fit with the platforms' vision. A summary of the findings of this expert interview is shown in Table 16.

Table 16: Overview of the findings of the expert interviews on functionality. Scores of 1-5, a higher score is better. Question number included in brackets

Role (1)	Software Delivery Manager	CTO	Product Manager	Product Owner
Solves customer issues (3)	4/5: Top-level data model & data lineage 3/5: Search & glossary	5/5	5/5	4/5
Company size (5)	All sizes	All sizes	Medium and large	Medium and large
User seniority (6)	Junior	Medium	Medium & Expert	Medium
User role (7)	-	Architect & Developer	Architect & Developer	Architect & Developer
Impact on complexity (8)[13]	2/5	2/5: search & glossary 4/5: top-level data model & data lineage	1/5	2/5
Completeness (11)	5/5	4/5	4/5	4/5

After the initial questions to determine the role and experience of the expert, the questions first focused on functional appropriateness. This started with a question in which the experts were asked to which extent the shown features would solve issues customers experience in the usage of the iPaaS. Although one of the experts indicated a difference between the features of the top-level data & data lineage and the search & glossary, the other two experts rated this with a 5/5. This indicates that these features add features to the platform which are expected to provide a solution to problems customers currently experience within the platform.

Regarding the design, all of the experts indicated a good fit with the platform, taking into account that they would do more detailed user testing prior to implementing a new design into the platform. From the input of the experts, it can be derived that the features which might need the most attention prior to developing them into a feature of the iPaaS are the search and the data lineage features. For the search feature, it is indicated that it would be able to make a clear distinction between the results which can be found in different views of the iPaaS, as the iPaaS might display certain elements on multiple positions in the platform, which might make it hard to assess the location the users wants to go to. In addition, some focus might still be needed on the data lineage feature, which was indicated to have been placed at a potentially unsuitable position within the platform.

5.2.1.1. Target audience

Within the scope of functional appropriateness, the experts were asked which of their clients, which user roles, and which user seniority levels would benefit most from the features. For the selection of which client would benefit most, the experts were presented with three options:

- Small clients: clients who have one model, with a data model with a small number of entities
- Medium clients: clients who have one model, with a data model with a medium to large number of entities
- Large clients: have multiple models, each of which with a medium to large number of entities in each data model

[13] The scale used for this question deviates from the other scales. In this case, a 1 means that the usage gets easier, whereas a 5 means an increased complexity

Finally, there was also the option to choose all clients. There was no clear consensus between the experts on which client would benefit the most. Whereas the experts agreed that medium and large clients would strongly benefit from the addition of the proposed features, since retaining a clear overview of the data model is more difficult for them, and the search would bring more added value, two of the experts strongly indicated that the small customers also benefitted. One expert indicated that this was primarily since they see a trend where clients of their iPaaS gradually grow in the number of integrations they want the iPaaS to facilitate, and therefore will eventually have a larger model and therefore become a medium client. The second expert argued that the small clients would especially benefit from features such as the search and glossary functionality, which guide these clients through the platform. Since the small clients have fewer integrations, there is also less work for them to do in the platform. Because of this, they might not be using the platform on a daily basis, and therefore would not be able to *blindly* navigate to everything they need. This would make the glossary and the search feature especially useful to them as well. Therefore, it can be argued that the features are beneficial to all clients, although the top-level data model and data lineage features might not be needed by clients with a small data model.

Regarding the user roles, four stakeholders have previously been identified in Chapter 3.1: the architect, developer, business user, and support. All experts agree that the features in the design are focused on the architect and the developer. The business user does not currently have access to the iPaaS and does therefore not benefit from the features. Support does not currently benefit since the features at not focused on pages that are used by support. The experts do indicate that the features, perhaps in altered form, could also be applied in the management parts of the iPaaS, which would also make them relevant for support.

Finally, a selection of the level of expertise of a user is made. Users are divided into three groups which illustrate the different levels of experience users may have with the platform, and be able to tailor the fit of the features that are designed to these user groups:

- Junior users: with little to no experience in working with the iPaaS;
- Medium users: have experience in building (parts of) integrations in the iPaaS;
- Expert users: highly experienced with all aspects of the iPaaS, including building integrations, deploying them, and monitoring them.

The opinions of the experts on which user would obtain the most benefits from the addition of the features were various. The medium user was mentioned the most, but individual experts also indicated the junior user and the expert user as benefitting from the features. The counter-arguments to these were relevant, however. For example, two of the experts argued that the junior users would not benefit from the features other than search since they were not yet experienced enough with the base functionality of the platform to be in a position where they would need advanced features such as the top-level data model or data lineage. In addition, expert users were argued to already have their methods to obtain the overview created by data lineage, for example, and might therefore be difficult to convince to obtain this information through a new feature. The features are therefore most relevant for medium users.

Regarding the overall target audience, the features are focused on clients of the iPaaS of all sizes, where they help the developer and architect with medium platform knowledge.

5.2.1.2. Learning curve and effects of implementing

For the next questions, the experts were asked to think of the effects of implementing these features on the platform as a whole, with the benefits and disadvantages this might bring, as well as the effect on the learning curve for (new) users.

As for the learning curve, a difference in rating was again given for the different groups of functionalities. As for the search and glossary functions, all experts found these to reduce the complexity of the platform. The data lineage and the top-level data model were regarded to be an advanced feature by all experts, and their opinion on the effect of the learning curve was different. One expert argued that these features add complexity (score of 4/5), by adding functionality to the platform which is new, and might need some explanation, also for the medium- and expert-level users. On the other hand, the other experts argued that by adding these features the platform facilitates obtaining insights that were previously significantly more difficult to obtain. These experts rated the question on complexity with a 1/5 or 2/5, indicating a (significant) reduction in the complexity of the platform.

On a more general level, the experts have been asked about the benefits and disadvantages they see regarding the addition of these features. As for the benefits, all experts indicated that they see the top-level data model, which provides the opportunity to combine data models of different integration patterns and have the possibility of 'zooming in' or 'zooming out' of the overview depending on the view needed, as a major benefit. The data lineage was also seen as a clear benefit, but it was expected that users would not be able to assess the relevance of this feature at first glance. The search was found to add value, although it was argued by one of the experts that this feature needs to be researched and user-tested extensively prior to release since there is only one chance to launch it during which the feature can be either widely adopted or disregarded if not developed properly. A benefit indicated for the glossary was that it would provide motivation for the users to fill out the questions at the requirements capture part of the iPaaS since this information would come back throughout the development lifecycle.

As for disadvantages, multiple points were also provided. Firstly, since there are new views added, users have another way of retrieving information from the platform. This has to be clearly mentioned to the users, and the training has to be adjusted for this. As for the glossary, the questions provided are currently not mandatory. This might not be enough motivation for users to fill them out. On the other hand, making them mandatory fields can negatively impact the speed of building an integration. Also related to the glossary, is the effect on existing customers' models if this feature were to be released. Since the questions would be changing, the fields might not be able to be updated dynamically, and therefore the existing clients would need to fill out the information again. They might not want to put time into doing this, which would also mean that this feature would only be adopted by new clients, making its effect more limited.

5.2.1.3. Completeness and suggestions

The final part of the interviews focused on a rating of the completeness of the features as they are shown in the prototype and explained to the expert, as well as open questions regarding suggestions the experts might still have. As for the completeness as the features were presented and discussed, the experts all rated this with a 5/5 or a 4/5.

As for the suggestions, several were given. Firstly, the prototype did not show a working implementation of providing a *tag* to a system or integration, based on the input in the requirements capture phase. This is one of the feedback points which originated from the sprint reviews while developing the prototype. It would be a useful addition that would use the data from the glossary in a more interactive way, which could then also be applied throughout the portal, and based on these automated tags, also different layouts could be applied to other features, such as the data lineage where a high-confidentiality integration, as documented in the requirements phase, can be marked with a red icon, for example. Secondly, multiple experts indicated that the data lineage feature was

not placed appropriately. This was already addressed previously. The most suitable position should be looked into.

5.2.2. Usability expert interview findings

The second set of expert interviews was focused on the usability of the proposed features and has been conducted with users of the iPaaS. These users were all working for a client of the iPaaS, either directly or indirectly in the role of, for example, an integration consultant. They each had multiple years of experience with the iPaaS, but might not all be using the platform on a daily basis anymore. An important basis for selecting the interviewees was to have them be of various roles. In addition, the organization they conduct their work for had to be different, to ensure that the input of multiple organizations is taken into account. This criterium was set to ensure that the results were not affected by differences in the extent to which a client uses the iPaaS. The overview of the ratings for each of the *scale* questions is shown in Table 17. This table also shows the role of each of the interviewees and the average of the scores.

Table 17: Overview of the findings of the expert interviews on usability. Scores of 1-5, a higher score is better. Question number included in brackets

Role (1)	**Architect**	**Architect**	**Developer**	**Developer**	**Average**
Type of user	Regular	Regular	Daily	Daily	
Feature positioning (3)	4	4	5	4	**4,25**
Glossary usefulness (5)	3	4	4	4	**3,75**
Search usefulness (6)	5	4	5	3	**4,25**
Top level data model need (7)	4	3	3	5	**3,75**
Data lineage need (11)	5	4	3	4	**4**
Recommend implementation (13)	4.5	4	4	4	**4,13**
Overlap with external tools (14)	3	3	2	3	**2,75**

The interview was opened with a question regarding the suitability of the placement of each of the features within the platform. All of the interviewees agreed that the location of each of the features as they were included in the prototype was in a logical place, with all of the interviewees giving a rating of 4/5 or 5/5.

Standing out from this list of ratings is the rating of the overlap with external tools, of question 14. This has an average score of only 2,75, indicating (very) little overlap with external tools, and certainly not enough to replace them. With these external tools, the experts regarded a wide array of tools, such as other (enterprise architecture) modeling tools, communication tools and data tools. The positioning of the platform, which is focused on a relatively narrow user group within an organization currently limits the central position the platform might be placed at if access would be rolled out to more users. The lower score of this criterium does not affect the usefulness of the addition of the features.

This followed with the initial idea of the relevance of each feature. The overview of the rating of the features is shown in Table 18. This question was asked as one of the first questions of the interview, to capture the initial thoughts of the interviewee regarding their opinion on each of the features, before possibly influencing them with follow-up questions on more specific features. The feature which clearly came out on top with regards to perceived usefulness was the search functionality. This is an interesting finding since the search functionality was not shown as a functioning feature in the prototype, only as a feature with static in and output. This illustrates a clear need for the search functionality in an iPaaS. In addition, several users indicated that a connection between the search functionality with other features of the prototype, such as the glossary, would be useful.

Three out of the four experts rated the search functionality as their number one feature in terms of most perceived usefulness. The fourth person, however, rated it as the lowest. The consistency of the placement of the perceived usefulness of other features was even more diverse. In these features, a clear distinction between the user preference and their role can be seen, as well as a distinction between organizations. As mentioned previously, the experts consisted of people working at (integration landscapes of) different organizations. The way in which these organizations have configured their iPaaS models is, at the core, very different. For example, these organizations have different methods of building integrations, for example, a structure-based more on team autonomy or a structure based more on a top-down approach. This resulted in, for example, one architect rating the lineage function as the number two and the glossary as the last place, another rated the lineage function as the last place, and the glossary as number 2. It is important to note here that although place four, or the last place, was the lowest, none of the interviewees indicated that they found this feature to not be useful. Their opinion was mostly based on the other feature being *even* more useful.

Table 18: Overview of the relevance assessment of each of the four features. #1 is the highest relevance, #4 the lowest. Answered from the perspective of the expert

Role	**Architect**	**Architect**	**Developer**	**Developer**	**Average**[14]
Feature #1	Search	Search	Search	Data lineage	**Search**
Feature #2	Data lineage	Glossary	Top level data model	Top level data model	**Top level data model**
Feature #3	Top-level data model	Top-level data model	Glossary	Glossary	**Data lineage**
Feature #4	Glossary	Data lineage	Data lineage	Search	**Glossary**

After these general questions, the interview went more into detail for each of the four functionalities. These topics were addressed in the same way in which they were also shown in the prototype demonstration: starting with the search glossary, followed by the search functionality, the top-level data model, and finally the data lineage. This has to do with the way in which the iPaaS is set up, and how these features correlate. The order in which they were shown is a logical feature in terms of which path a user follows when creating an integration, for example.
To illustrate this further the order in which an integration is built is briefly explained: firstly, the systems and integrations are drawn and their base information, such as names, descriptions, and information flows are addressed. This can be described as the requirements capture phase. In this phase, the glossary is filled out. After this, the search functionality is able to show its first results: systems and integrations and their definitions. From this, the user proceeds to define more technical details upon which messages are transferred in integrations, and how these messages look. At this stage, the data model is built or extended. This is also the location where the top-level data model comes into play. Especially for existing landscapes, this could help a new user to get an overview of what already exists. After this, integrations can be developed to actually be able to connect between systems. At this stage, the data lineage also comes in.

5.2.2.1. Glossary

The opinions regarding the glossary varied. This primarily had to do with the individual use of the requirements capture phase of the interviewees. It received a 3/5 rating from one interviewee who

[14] These averages are calculated by taking the average of the place of each of the features. This gives search: 1,75 | Top level data model 2,5 | Data lineage 2,75 | Glossary 3.

rarely used this part of the iPaaS themselves, and a 4/5 rating from other interviewees, who found this feature to be either an improvement of their own workflow or considered it a feature that would encourage users to fill out the requirements information.

As previously mentioned, this feature was also seen as an enabler of the search functionality, which makes it possible to display definitions and details even when a given data object is not currently shown on the page.

An important remark made by one of the interviewees was that the glossary should not enable the editing of requirements in other positions than this is currently possible. They strongly encouraged the visualization of this data in other phases than currently, but expect that changing the portal to allow for the editing of the requirements at any point in the platform renders the requirements capture phase mostly redundant, and reduces the perceived need for filling out this information since it can be done at any point later on in the development process.

5.2.2.2. Search

All interviewees were enthusiastic about the search functionality, as already indicated by all of them rating it as the number one in perceived usefulness. Because of this enthusiasm, considerable feedback was given upon its implementation.

In order to provide background for these additional suggested features, the initially intended functionality it is good to note the initial intention of the search functionality: it should search through all data, including entities, attributes, systems, integrations, and definitions, as they are recorded within the opened model.

Additions that were suggested were to not only search through everything in the currently opened model but to search through everything which can be found in the client belonging to the currently opened model. For the feature of the top-level data model and lineage, this was part of the interview protocol, but for this feature, the interviewees suggested this addition by themselves, prior to asking the interviewees their opinion regarding the extension to multiple models for the other features.

Other than the multi-model search, another interviewee suggested that they would also like to be able to search through the documentation through the same search. Other interviewees have not been asked about their opinion on this, as this was not part of the interview protocol.

By design, the search functionality should be able to navigate the user to the position the search result was found. This was one of the main features which made the users enthusiastic about this feature.

5.2.2.3. Top-level data model

The top-level data model was found to be in the middle of the perceived usefulness. The perceived need for the top-level data model was also different. Two of the interviewees had a neutral (3/5) opinion towards the need they experience at their client for such a model. The enterprise architect was more enthusiastic about this feature, giving it a 4 out of 5. This was mostly caused by their data models being quite extensive, giving them a stronger need to get this overview. In addition, their organization also worked in more of a top-down approach, where the team of enterprise architects had a strong vision and impact on the development of the integrations. Since this provides the need to maintain the overview of multiple models, a clearer overview per model would aid them in their work. Interviewees who were active at organizations that were more focused on team autonomy did not experience the same need for this top-level data model but did illustrate that they would be likely to use this feature if it were available.

As addressed previously, this model had a question in the interview list regarding the perceived usefulness of extending this functionality across models rather than being focused on one model. Similar to the suggestion of one of the interviewees while addressing the search functionality, all of the interviewees were in favor of extending the current functionality to include all models of a client rather than concentrating on a single model. This is in contrast with the opinion of some of the interviewees in the interview of 2.3, where some interviewees indicated that clients do not have a need for this, since the split in their models is chosen carefully. Using the visualization as shown in the prototype, this minority which initially indicated that they did not see perceived use for this feature in the previous interview disappeared.

Several improvements have been described for the top-level data model. Currently, the top-level data model does not show cardinalities of relationships, in contrast to other data models which are already offered through the iPaaS. Multiple interviewees indicated that they would find it valuable to be able to see the cardinalities. Other interviewees suggest features such as color-coding entities to show that they belong to a similar group and being able to drag the entities around to the desired position. There are all mostly minor differences which further increase the visual overview. Suggestions for additional abstraction levels, other than the ones introduced in 4.2.4 were not given.

5.2.2.4. Data lineage

The fourth feature evaluated during the validation interview was the data lineage feature. Depending on the role of the interviewee, the added value of this feature was not immediately understood. This illustrates that this feature would have a relatively narrow focus on a certain user group. After some further explanation about what the data lineage feature does, and how it obtains its information from the platform to the interviewees who requested more information, they got a clearer understanding of the perceived usefulness of this feature. Although this might have impacted the overall rating of which feature they would find to be most and least relevant, the rating given for the perceived usefulness is now more reliable: a score of 4/5 on average, although the lowest score was a 3/5 because of one individual expecting less usage of this feature for their role.

The interviewee that was most enthusiastic about this feature worked in an organization that also has a considerable number of applications that are not running within the iPaaS but are integrated using different methods. This interviewee indicated that they would greatly benefit from including these external message exchanges within the iPaaS, even if this meant that they would have to fully model these integrations manually within the iPaaS although the iPaaS does not process the messages. It has the potential of giving them a full overview in a way they do not have available yet.

An improvement that was remarked by one interviewee directly, and another one indirectly was to extend the functionality of the data lineage to processes rather than only on an entity basis. By including processes, the lineage feature has the potential of showing how data flows through the system and how data originates, on a full data lifecycle rather than focusing on only data entity objects. Another suggestion made was to integrate this feature more with the glossary. This would be helpful by adding more questions at the requirements capture phase of the iPaaS where for each flow information is given upon each of the CIA-triad areas, which can then be shown in the lineage to get a quick overview of which processes are especially critical or sensitive. From an enterprise architect's point of view, this was found to be very useful.

5.2.2.5. General findings

At the end of the interview, all interviewees were asked if they would like to see the shown and discussed features be implemented into the iPaaS they used. All of the interviewees would recommend the implementation of these features, indicating that they would be able to perform

their work better and faster when they would have access to these features. This confirms the user demand for the features and confirms the findings of these four features to be relevant as outlined in Table 9.

In addition, the experts found there was some overlap between the features added in the prototype and features obtained through external tools within the organizations. All of the interviewees indicated that they did not expect the features as shown in the prototype to be of effect upon the usage of these external tools: these remain in use. On the other hand, adding the features of the prototype was indicated by multiple interviewees to improve the information which can be obtained from the platform. It was emphasized that the views which can be generated by the iPaaS should be able to be exported to a platform-independent format, such as an image or PDF file.

5.3. Conclusions

In both of the expert interview panels, the relevance of the features included in the prototype for the iPaaS was confirmed. All users indicated that they would like to see these features implemented into the platform. This was primarily caused by the placement of these features into the position where it solves problems of the iPaaS, as was confirmed by the expert panel on the functional aspects. The proposed features do not necessarily add new functionality to the iPaaS with regard to the current focus: creating system integrations. Instead, it is a set of features that supports the users in making decisions and assessing the need for changing the existing model. These are all choices that need to be made when creating an integration, but for which the platform does not currently provide extensive support.

The feature of the top-level data model helps the user in assessing the data which is already in the environment and enables the user to assess which data is needed for an integration. In the actual development of the integration, the data lineage feature can help in assessing the impact of a chance, prior to making the change and possibly impacting the integration landscape of their organization. The glossary and search functionalities apply to all users, not only those who need to do more advanced work within the platform.

With the exception of the glossary, each of the features that are introduced in the prototype provides a new means by which the platform provides the information to its users based on the information that is already stored in the platform. This implies that for the case of this iPaaS vendor, changes are primarily needed in the visualization of data rather than collecting additional information. In turn, this means that the users do not need to enter any additional data into the platform to achieve an improvement in the insights the platform can help them obtain. The exception of the glossary is caused by the change in the format of the metadata that the platform collects. Although the glossary aims at providing better insights to the user with less input from them, the transitioning for existing customers of the iPaaS would need substantial user input.

All of the interviewed expert users indicate that the inclusion of these features into the platform would improve the pace at which they can do their work in the platform. This input was obtained even though this was not a question that was directly asked to them. This is in line with the objective of the main research question, which is to improve the efficiency with which the developers of the platform do their work. A clear feature that provides most of the benefits for the user cannot be identified. This is primarily caused by the different roles that benefit from different features and the fact that the features are interdependent.

The set of features as it has been produced in the prototype is focused on the user roles of the architect and the developer. These roles obtain benefit from these features primarily clients of the

iPaaS who have large models with more complex data models and a high number of systems and integrations. The features are expected to be used primarily by the users who already have experience within the iPaaS. New users would not need these features to start off getting to know the iPaaS.

It has been found that for the implementation of the features into this iPaaS some additional research might be needed in order to find the most suitable way for this iPaaS to implement the features. This would need to be done on a per-feature basis. As for the data lineage, a suitable position for this feature in the platform needs to be found and tested. The other features need less testing with regards to their position, and more on design level, and what the exact expectations of the users are.

6. Conclusions

This chapter starts by providing answers to the research questions as they were previously identified. After this, it discusses opportunities for further research as well as the implications of these results for practice and academic research.

6.1. Conclusion

This section provides answers to each of the research questions formulated in Section 1.4. For more details on the answer and the method used to conclude it, the corresponding sections of this research are mentioned. This overall research focused on identifying the relevance of applying features of an enterprise data catalog into an iPaaS, to improve the efficiency of the users of the iPaaS. The main research question of this research is:

How to design a solution for improving the usability of iPaaS platforms by adding features of enterprise data catalogs into these iPaaS platforms that enables an improved workflow for its users?

To design a solution that improves the usability of iPaaS platforms, features of enterprise data catalogs should be added. This research provides a framework to identify which features of enterprise data catalogs would be most beneficial to their users. This is a collaborative process that involves both the iPaaS vendor and users of their product. The validity of the framework is shown by applying it to the platform of one single iPaaS provider. After following the steps of the framework, a prototype is developed based on this iPaaS. After validating this prototype with two sets of expert users, one consisting of users of the platform, and the other of internal stakeholders of the vendor, it is confirmed that the additional features improve the usability of the platform. From the vendor's perspective, the additional features are confirmed to solve several issues which are indicated by some of their customers. Through the application of the framework, it is shown that 4 out of the 6 identified features which were considered relevant for the iPaaS evaluated in this research are a valuable addition.

The framework provided in this research enables vendors of iPaaS platforms to improve their platforms. Through the application of the framework weaknesses in the current platform can be identified as well as opportunities for improvements. This helps vendors of an iPaaS to improve the maturity of their platform by providing their users with features that are needed for optimal usability of their platform.

This conclusion is the result of answering the sub-research questions. Each of these individual questions is answered below.

1. **Is it useful to extend an iPaaS with functionalities of a data catalog and why?**
   Functionalities of enterprise data catalogs are of high relevance to iPaaS platforms. Whereas the objective of these two products is different, they both involve enabling the user to work with data sources. The focus of an iPaaS is narrower than an enterprise data catalog, as it focuses primarily on transferring data between applications, whereas the enterprise data catalog focuses on making it accessible and understandable. Although the users of an iPaaS do not need the accessibility aspects of data in their platform, they do benefit from making it more understandable.
   In order to arrive at this conclusion, the first step is conducting literature and market analysis to identify the features of enterprise data catalogs. These features are then presented to a panel of experts who provided arguments as to why each of these features is or is not useful in an iPaaS.

More details on this research question are found in Chapter 2, and the expert interviews are found in Chapter 2.3.

1.1. Which overlap already exists between features of a data catalog and an iPaaS

Several features of enterprise data catalogs and iPaaS platforms are overlapping. This is caused by both of the platforms being developed for enterprise-level customers. These customers demand a certain level of features with regard to, for example, governance. In addition, both platforms enable connections with multiple systems and different data formats. This inevitably leads to overlap in features, such as similarities in offering system connectors. On the other hand, because of the different objectives of these two technologies, features that are only present in data catalogs, such as a top-level data model, a business glossary, data lineage & impact analysis are found. Since the literature on enterprise data catalogs is limited, more substantial literature on open data catalogs is also used. This results in a comparison of open data catalogs and enterprise data catalogs, which helps narrow down which features are more relevant to enterprise settings and which features are more general.

The overlap is studied through a systematic literature review of enterprise data catalogs in Section 2.1.2, which provides an overview of the features of the enterprise data catalog in Figure 6. The features of iPaaS platforms are identified through market analysis and literature. This process is described in section 3.4

2. **What is a suitable position within the enterprise architecture of an organization for a tool to create an overview of fragmented data sources?**

Both an iPaaS and an enterprise data catalog can reside at a central position within an enterprise's IT architecture. This is due to the large number of connections both of these technologies make to applications and data sources within the organization. Because of the nature of both technologies, it is best to retain this central position, as they are likely to be integrated with numerous other applications. This is especially the case for iPaaS platforms, which aim at reducing the overall number of integrations and improving maintainability. Its optimal benefits can only be obtained when it gets direct interactions with the applications. This research question is addressed in Section 3.5, where the architectures of the enterprise data catalog and the iPaaS have been compared through visualization in ArchiMate diagrams.

2.1: How can features of enterprise data catalogs be included in an iPaaS?

Because of the similarity of the position within an enterprise's IT architecture of iPaaS and enterprise data catalogs, the features depend on similar information. Therefore, a majority of the information needed to make it possible to include features of enterprise data catalogs into an iPaaS can already be derived based on the information which is already available to an iPaaS. The glossary, for example, can be seen as a specialization in metadata management. The top-level data model requires visualization of the relations which are already created within the iPaaS. The same holds for the other features which have been modeled in the ArchiMate diagram of the target architecture in Figure 16.

This question is addressed in the interview of Section 2.3 and in the creation of the enterprise architecture diagrams in section 3.5.

3. **Which data catalog features are relevant to add to an iPaaS?**

The list of features which resulted from studying the literature and through conducting a market analysis of enterprise data catalogs was shown to experts who were consulted for their input on which features they consider relevant for an iPaaS. This results in a list of features that might be

relevant to include in an iPaaS. This list is found in Table 8. Since each iPaaS is different, and a clear consensus on the features that are expected within an iPaaS could not be identified within this research, not each of the proposed features might hold the same relevance for every iPaaS. This is shown in Table 9, which shows the features which are relevant for the iPaaS of the vendor used throughout this research. Some of the features are already (partially) offered, and some require no further deepening. This is the case for every product, as there is no open standard on which the iPaaS platforms of different vendors are based, which is available for open data catalogs.
This research question is addressed in section 2.3.

4. **How does the proposed design fulfill stakeholder objectives?**
   The stakeholder objectives are defined in Chapter 3, which gives a full overview of all stakeholders. Based on the stakeholders and the objectives which are derived from the interviews of Section 2.3, a prototype is developed which validates the developed framework by applying it to the product of a single iPaaS vendor. The exact process of developing this prototype and the requirements evaluated for this can be found in Chapter 4.

   The validation of this prototype found that all interviewed stakeholders are convinced that the features are of added value to an iPaaS and will improve usability. Some of the stakeholders who participated from a user perspective have indicated that they expect the addition of these features to the platform will increase their efficiency of working since the additional features enable them to make decisions faster and that their workflow is made quicker through the addition of these features. At this point, no quantifiable results of this can be shown. A more extensive user study would be needed to confirm these claims. Some of the experts indicate that the features included in the prototype contribute to the maturity of the platform, which provides a convincing argument for a vendor to put in the time and effort to develop these new features as it can potentially increase their sales positioning.
   The validation of the fulfillment of the requirements is addressed in Chapter 5.

6.2. Discussion and limitations

This chapter outlines challenges experienced throughout this research, from the problem investigation through the development of the prototype and validation of the design.

6.2.1. Available literature

Sections 2.1.1 and 2.2 provide a literature analysis on the topics of *data catalogs* and *iPaaS*. For both of these topics, the number of academic sources that have been found is limited. The literature review of (enterprise) data catalogs provided a final set of 30 articles, of which about 2/3rd of the result set focused on open data catalogs. These open data catalogs do have some overlap with enterprise data catalogs, which is why they have been included in the literature review. On the other hand, the literature on open data catalogs fails to address topics that are especially relevant in enterprise applications, such as data access and security, and features such as a business glossary are not found in open data catalogs.

On the topic of the iPaaS, a result set of only 7 academic articles remains. iPaaS is barely addressed in literature as the platforms provide a specialistic solution, and are often tailored to specific organizational sectors, yet do provide the main subject in this research.

Because of the limited availability of scientific literature on both of these topics, this research had to rely on *grey* literature where academic literature was not available. Grey literature is not always

produced by academic standards, and it is, therefore, more difficult to assess its quality. Especially reports which are produced by organizations or commissioned by them are more prone to be biased.

This research has previously addressed that the number of organizations that are interested in leveraging their data is increasing. This can result in increased interest in both enterprise data catalogs as well as iPaaS applications. Therefore, if the literature reviews as outlined in this research were to be applied in some years, more literature might be available.

6.2.2. Lack of transparency in the offering of vendors

In this research, two market analyses are conducted. These market analyses are done based on public information about the features offered by different vendors' enterprise data catalogs and iPaaS products. Independent external analysis of multiple products was not found, these market analyses depended on information as the vendors displayed them on their website and documentation and tutorials which some vendors offered with public access. For this reason, both the market analysis of enterprise data catalogs as well as iPaaS solutions has been done not only on basis of webpages and documentation but also of external analysis of Gartner. Whereas Gartner does not offer independent research, it is considered a valuable addition to the findings which could not be extracted based on the products' website and documentation alone.

A disadvantage of this approach to collecting the information is that the websites which showcase the enterprise data catalog and iPaaS products are often focused on spiking interest for a product, and show very limited technical details. Therefore, the level of detail that a vendor has included in each of the features could not be evaluated, and the market analysis instead focused primarily on whether the previously identified features are present in each of the vendors' products. This was, where possible, extended with information that could be extracted through a vendors' documentation portal. It is important to note that not all vendors had their documentation portals publicly accessible. Therefore, this approach could not be taken for each of the vendors' products.

An improved method of conducting this analysis would be through the usage of a trial of the products. During the market analysis, it was found that a majority of the vendors offer either a trial of the products or a demo session. Since the number of vendors evaluated in this research was quite large, with a total of 30 different vendors for the enterprise data catalog and iPaaS products combined, the analysis of each of the different products through requesting a trial or demo could not be done in this research. In addition, the academic relevance of such a comparison is questionable, since the features of each of the products are under constant development, which implies that such time-intensive research could produce results that are already incorrect at the time when the research is completed.

6.2.3. Time sensitivity

The previous section concluded that time-intensive research on certain commercial products could produce conclusions that are already irrelevant at the time when the research is completed. This research also included a market analysis, which is done at one point in time, several months before the completion of this research. Although the author tried to ensure that this information is still accurate at the time of publishing this work, changes in the products of these vendors, which will inevitably happen over time, can decrease the accuracy of the findings.

Although the market analysis as outlined in this research would not be accurate when the products of the vendors change, a trend where the vendors adopt features as proposed in this research would confirm the findings of this research. In addition, it would confirm the business case of adding these features to the platform, since the products of all analyzed vendors are commercial.

6.2.4. Stakeholder inclusion

In Chapter 3.1, the stakeholders are identified. Two of the identified stakeholders are the *Business user* and *Support*. Whereas the proposed design does propose the addition of features that are beneficial to these stakeholders, these features were not included in the prototype and therefore also not part of the validation. As found in the validation, these stakeholders might gain some benefit from the features which were included in the prototype in a slightly altered form, but no validation has been done with any business user or support actors.

Although these stakeholders are still considered to be relevant, as confirmed by remarks made on the roles of these stakeholders by a number of the experts in the validation interviews, the expert panels did not have a representative of these roles in them. Since no stakeholders of either the business user or support actor roles are present in the expert interview panels, the extent to which the proposed features are relevant to them is not clear from this research alone. In order to get an academically sound conclusion on the relevance of these actors, additional research would need to put focus on those features which are considered to be relevant to these actors. An approach similar to the one taken in this research, where a prototype of these features is created and shown to experts of these stakeholder roles, can then provide guidelines on how to apply the features to be relevant for these stakeholders.

Although both the stakeholders of support actor and business user are limited in the extent they have been taken into account in the validation of this research, the reasons for doing so are different. The support actor is seen as a user which can benefit from the proposed features, but they would need the features to be placed in other places in the platform. This could not be taken into account for the prototype, because of time limitations. Focusing on this feature would reduce the level of detail on the places where these features were relevant for the architect and developer, which belong to the group of users which get the most focus for development. The business user, on the other hand, is a group of users that does currently not have access to the platform. Although it is still relevant to investigate the extent to which this group could benefit from having access to the platform, it is worthy to devote a research of itself to that topic. Such a change to iPaaS platforms, which are focused on more technical users such as the developer and the architect who are previously introduced, would potentially affect the pricing and marketing of iPaaS products and could therefore not be taken into account for the extent of this research.

6.2.5. Generalizability

Whereas this research focuses on finding the relevance for applying features of enterprise data catalogs into iPaaS solutions, the findings and evaluation of this research are done based on a single iPaaS product. Although Chapter 6 provides a generalization for the proposed design, this is no guarantee that the design would be relevant to *any* iPaaS solution. Based on the analysis Gartner made on the offering of various iPaaS providers [47], existing iPaaS products can be divided into different maturity levels. For example, Gartner defines *leaders*, which they consider to have a full understanding and implementation of the features needed to be called an iPaaS. On the other side of Gartners' spectrum, *niche players* are defined that focus on certain customer segments and might not be applicable to all customer segments of iPaaS products. Therefore, the full offerings of iPaaS providers which are on opposite sides of the *Magic Quadrant* can be very different. This in turn makes it complex to provide a generalization. This research would therefore need to be applied on a per-case basis to find the suitable aspects for each iPaaS.

Aside from the potential difference in features already offered by each iPaaS, the lack of transparency that is identified in 6.2.2, further illustrates the complexity in generalizing the findings of applying the prototype created as an artifact of this research to iPaaS products of other vendors.

This can also be recognized from the differences in relevant features from enterprise data catalogs for iPaaS products, compared to the iPaaS product used in this research. This difference is shown in Table 8, which shows the relevant features of enterprise data catalogs for iPaaS products, and Table 9, which shows the relevant features for the iPaaS used in this research. Whereas these features overlap, applying the same solution to that currently created as a prototype might produce similar results in a different iPaaS. It might, however, also result in different results.

6.2.6. Design Science Research Methodology validity

Although this research uses the DSRM, the iterative nature of the DSRM has not been fully applied. This can mean that the most optimal solution is not achieved yet. The author believes that this is partially covered by other parts of the overall methodology which have been applied iteratively. For example, the development of the prototype using 6 sprints uses an iterative approach, where feedback from the stakeholders can be taken into account for the next development cycle. To ensure that the adoption of the proposed features into an iPaaS does indeed deliver the intended results, it is advisable for the platform to conduct proper user testing. This helps ensure that the features indeed perform as well as they potentially can, which this design currently does not take into account.

6.3. Future research

Given the limited time available for conducting the research in the scope of this Master's thesis, the results of this research provide opportunities for further topics of research. This section gives an overview of topics that can use further research based on the findings of this research.

Because of the closed nature of the commercial offerings of both enterprise data catalogs as well as iPaaS solutions, this research uses one iPaaS throughout the research. The results produced and discussed in the previous section are expected to be similar to other iPaaS platforms. However, this could be investigated further by applying the solution to an iPaaS of a different vendor.

Further research could focus on providing quantifiable results for the claims of some of the users who indicated that their productivity would increase when the iPaaS were to be extended with the features of enterprise data catalogs. In order to produce quantifiable metrics, it would be required to implement a high-fidelity prototype of an iPaaS with and without the features of enterprise data catalogs. A substantial number of participants with various levels of experience in using an iPaaS are needed to confirm this claim.

Another topic where more research can be put into is the addition of features that are focused at support users and the business user. This research identified that support and the business user are relevant actors who can benefit from the addition of features of enterprise data catalogs but focused on the added value for the enterprise architect and developer.

In addition, this research did not take into account the possibility of combining the data catalog with an iPaaS. It was discovered that there is a very limited number of vendors who offer both an enterprise data catalog as well as an iPaaS, which might illustrate a different set of expertise for a vendor to be able to build software for the different applications. Yet, it would still be an opportunity to investigate whether the extension of an iPaaS with business users and functionality tailored to them could eliminate the need for an enterprise to also have an enterprise data catalog. As also mentioned in the previous section, further research could look into obtaining quantifiable results regarding the efficiency improvements of applying the proposed features into an iPaaS.

Finally, a research opportunity for research that has considerably more time than the research conducted as a Master's thesis could be to look into the future of system integration. With an

increasing trend of adopting cloud applications rather than self-hosted software, creating an integration between two applications might become easier than the extensive connection opportunities currently offered by integration platforms such as iPaaS. Potentially, programming languages could create standards throughout languages that enable the exchange of information between them without the need for middleware. Research could look into the development of such standard exchange of data between applications.

6.4. Implications

This section addresses the implications of this research results both for practice, including iPaaS and enterprise data catalog vendors as well as organizations using either of these software products, as well as for academic research.

6.4.1. For practice

This research provides a market analysis of enterprise data catalogs and iPaaS platforms as they are offered through various large, commercial vendors during the time of this research.

This research provides a framework that can be used by vendors of iPaaS products to improve their products through the addition of features of enterprise data catalogs. This research concludes that these additions improve the usability of iPaaS platforms, and users claim that adding these features can help them to conduct their work quicker. In addition, providing more detailed and clearer information to the users ensures that integrations are created *first time right*, saving considerable development hours for a developer to develop integrations.

As for organizations looking to adopt an iPaaS platform, this research provides a compelling argument for validating whether enterprise data catalog features, as identified in this research, are available in the iPaaS they are considering. An iPaaS that offers there features has better usability and is likely to have higher productivity among its users.

Similarly, vendors of iPaaS platforms can ensure to include the proposed features and clearly advertise them. This, in turn, can justify charging a higher fee per user, as the savings of the organizations adopting this platform are higher.

Ultimately, widespread adoption of the proposed features into iPaaS platforms can bring the market segment to a higher level of maturity, and make them relevant for other types of organizations.

6.4.2. For academic research

During the time of this research, no substantial amount of academic literature on the topics of both enterprise data catalogs and iPaaS products was available, although there are numerous commercial vendors offering such a product. This research adds academic literature on both of these topics and provides new academic insights into a gap that has not been previously addressed in academic research: the combination of these two products.

It also compares identifies the differences between open data catalogs, which are actively studied in academic settings, and identifies a gap between the needs for open data cataloging and enterprise data cataloging.

In addition, design theory is successfully applied to develop a framework that enables a validated way of improving iPaaS products.

References

[1] N. Neuteboom, C. Burgering, and S. Duijn, “Van data naar daadkracht,” 2018.

[2] IDC, “The Seagate Rethink Data Survey,” 2020.

[3] A. Shahrokni and J. Söderberg, “Beyond information silos challenges in integrating industrial model-based data,” *CEUR Workshop Proc.*, vol. 1406, pp. 63–72, 2015.

[4] C. Labadie, C. Legner, M. Eurich, and M. Fadler, “FAIR Enough? Enhancing the Usage of Enterprise Data with Data Catalogs,” *Proc. - 2020 IEEE 22nd Conf. Bus. Informatics, CBI 2020*, vol. 1, pp. 201–210, 2020, doi: 10.1109/CBI49978.2020.00029.

[5] R. J. Wieringa, *Design science methodology: For information systems and software engineering*. 2014.

[6] B. Kitchenham and S. Charters, “Guidelines for performing Systematic Literature Reviews in Software Engineering,” vol. 2, 2007.

[7] Statista, “Volume of data/information created, captured, copied and consumed worldwide from 2010 to 2015 (in zettabytes) [Graph].” 2021, Accessed: Oct. 15, 2021. [Online]. Available: https://www.statista.com/statistics/871513/worldwide-data-created/.

[8] L. Ehrlinger, J. Schrott, M. Melichar, N. Kirchmayr, and W. Wöß, “Data Catalogs: A Systematic Literature Review and Guidelines to Implementation,” in *Database and Expert Systems Applications - DEXA 2021 Workshops*, 2021, vol. 2, pp. 148–158, doi: 10.1007/978-3-030-87101-7_15.

[9] J. Riley, *Understanding Metadata*. Baltimore: National Information Standards Organization (NISO), 2017.

[10] A. Halevy *et al.*, “Goods: Organizing Google’s Datasets,” in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2016, pp. 795–806, doi: 10.1145/2882903.2903730.

[11] Statista, “Business Intelligence Software - Worldwide,” 2021. .

[12] E. Zaidi, G. De Simoni, R. Edjlali, and A. D. Duncan, “Data Catalogs Are the New Black in Data Management and Analytics,” *Gartner*, no. December, pp. 1–16, 2017.

[13] J. Nogueras-Iso, J. Lacasta, M. A. Urena-Camara, and F. J. Ariza-Lopez, “Quality of Metadata in Open Data Portals,” *IEEE Access*, vol. 9, pp. 60364–60382, 2021, doi: 10.1109/ACCESS.2021.3073455.

[14] S. Neumaier, J. Umbrich, and A. Polleres, “Automated Quality Assessment of Metadata across Open Data Portals,” *J. Data Inf. Qual.*, vol. 8, no. 1, pp. 1–29, Nov. 2016, doi: 10.1145/2964909.

[15] J. Kučera, D. Chlapek, and M. Nečaský, “Open Government Data Catalogs: Current Approaches and Quality Perspective,” in *Technology-Enabled Innovation for Democracy, Government and Governance*, 2013, pp. 152–166, doi: 10.1007/978-3-642-40160-2_13.

[16] F. Maali, R. Cyganiak, and V. Peristeras, “Enabling Interoperability of Government Data Catalogues,” in *IFIP International Federation for Information Processing*, 2010, pp. 339–350, doi: 10.1007/978-3-642-14799-9_29.

[17] J. Klímek, “Reflections on: DCAT-AP representation of Czech national open data catalog and its impact,” *CEUR Workshop Proc.*, vol. 2576, no. 19, pp. 1–9, 2019.

[18] C. Labadie, M. Eurich, and C. Legner, “Data democratization in practice: fostering data usage with data catalogs,” 2020.

[19] G. De Simoni and M. Beyer, “Magic Quadrant for Metadata Management Solutions,” *Gartner*, no. November, 2020.

[20] G. Seshadri and S. Shanmugam, “Aspects of Data Cataloguing for Enterprise Data Platforms,” *Proc. - 2nd IEEE Int. Conf. Big Data Secur. Cloud, IEEE BigDataSecurity 2016, 2nd IEEE Int. Conf. High Perform. Smart Comput. IEEE HPSC 2016 IEEE Int. Conf. Intell. Data S*, pp. 134–139, 2016, doi: 10.1109/BigDataSecurity-HPSC-IDS.2016.52.

[21] M. Lee, E. Almirall, and J. Wareham, “Open Data and Civic Apps: First-Generation failures, second-generation improvements,” *Commun. ACM*, vol. 59, no. 1, 2016.

[22] A. L. Washington and D. Morar, “Open government data and file formats: Constraints on collaboration,” in *ACM International Conference Proceeding Series*, 2017, vol. Part F1282, pp. 155–159, doi: 10.1145/3085228.3085232.

[23] European Commission, *Directive (EU) 2019/1024 of the European Parliament and of the Council of 20 June 2019 on open data and the re-use of public sector information*, no. L172. European Parliament, 2019, pp. 56–83.

[24] M. D. Wilkinson *et al.*, “The FAIR Guiding Principles for scientific data management and stewardship,” *Sci. Data*, vol. 3, no. 1, p. 160018, Dec. 2016, doi: 10.1038/sdata.2016.18.

[25] Y. Asano *et al.*, “Constructing a Site for Publishing Open Data of the Ministry of Economy, Trade, and Industry,” *New Gener. Comput.*, vol. 34, no. 4, pp. 341–366, Oct. 2016, doi: 10.1007/s00354-016-0403-y.

[26] R. Cyganiak, F. Maali, and V. Peristeras, “Self-service linked government data with dcat and gridworks,” *ACM Int. Conf. Proceeding Ser.*, 2010, doi: 10.1145/1839707.1839754.

[27] T. Skopal, J. Klímek, and M. Nečaský, “Improving findability of open data beyond data catalogs,” *ACM Int. Conf. Proceeding Ser.*, pp. 2–6, 2019, doi: 10.1145/3366030.3366095.

[28] European Union, “EU languages,” *28-07-2020*, 2020. https://europa.eu/european-union/about-eu/eu-languages_en (accessed Oct. 26, 2021).

[29] J. F. Toro, D. Carrion, A. Albertella, and M. A. Brovelli, “Cross-border open data sharing: GIOCOnDA project,” *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. XLII-4/W14, no. August, pp. 233–238, Aug. 2019, doi: 10.5194/isprs-archives-XLII-4-W14-233-2019.

[30] R. Albertoni, D. Browning, S. Cox, A. G. Beltran, A. Perego, and P. Winstanley, “Data Catalog Vocabulary (DCAT) - Version 2,” *W3C*, 2020. https://www.w3.org/TR/vocab-dcat-2/ (accessed Oct. 28, 2021).

[31] X. Wang, T. Tiropanis, and R. Tinati, *Metadata and Semantics Research*, vol. 672. Cham: Springer International Publishing, 2016.

[32] Kennis- en Exploitatiecentrum Officiële Overheidspublicaties, “DCAT-AP-DONL,” *docs.datacommunities.nl*, 2021. https://docs.datacommunities.nl/data-overheid-nl-documentatie/dcat/dcat-ap-donl (accessed Oct. 27, 2021).

[33] C. Arnaut, M. Pont, E. Scaria, A. Berghmans, and S. Leconte, “Study on data sharing between companies in Europe,” European Commission, 2018. doi: 10.2759/354943.

[34] P. Holl and K. Gossling, “Midas: Towards an Interactive Data Catalog,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11721 LNCS, pp.

128–138, 2019, doi: 10.1007/978-3-030-33752-0_9.

[35] S. Neumaier, L. Thurnay, T. J. Lampoltshammer, and T. Knap, “Search, Filter, Fork, and Link Open Data: The ADEQUATe platform: Data- and community-driven quality improvements,” in *The Web Conference 2018 - Companion of the World Wide Web Conference, WWW 2018*, 2018, pp. 1523–1526, doi: 10.1145/3184558.3191602.

[36] P. Škoda, D. Bernhauer, M. Nečaský, J. Klímek, and T. Skopal, “Evaluation Framework for Search Methods Focused on Dataset Findability in Open Data Catalogs,” in *Proceedings of the 22nd International Conference on Information Integration and Web-Based Applications & Services*, 2020, pp. 200–209, doi: 10.1145/3428757.3429973.

[37] M. I. S. Oliveira, L. E. R. De Alencar Oliveira, A. G. De Fátima Barros Lima, and B. F. Lóscio, “Enabling a unified view of open data catalogs,” in *ICEIS 2016 - Proceedings of the 18th International Conference on Enterprise Information Systems*, 2016, vol. 2, no. Iceis, pp. 230–239, doi: 10.5220/0005835202300239.

[38] S. R. Ojha, M. Jovanovic, and F. Giunchiglia, “Entity-Centric Visualization of Open Data,” no. November 2014, pp. 149–166, 2015, doi: 10.1007/978-3-319-22698-9.

[39] E. Quimbert, K. Jeffery, C. Martens, P. Martin, and Z. Zhao, “Data Cataloguing,” in *Towards Interoperable Research Infrastructures for Environmental and Earth Sciences: A Reference Model Guided Approach for Common Challenges*, vol. 12003, Z. Zhao and M. Hellström, Eds. Cham: Springer International Publishing, 2020, pp. 140–161.

[40] F. Kirstein, B. Dittwald, S. Dutkowski, Y. Glikman, S. Schimmler, and M. Hauswirth, “Linked Data in the European Data Portal: A Comprehensive Platform for Applying DCAT-AP,” 2019, vol. 11685, pp. 192–204, doi: 10.1007/978-3-030-27325-5.

[41] M. Y. Choi, C. J. Moon, and S. J. Jung, “Building methods of intelligent data catalog based on graph database for data sharing platform,” *ICIC Express Lett. Part B Appl.*, vol. 11, no. 10, pp. 953–959, 2020, doi: 10.24507/icicelb.11.10.953.

[42] J. Klímek and P. Škoda, “Linkedpipes DCAT-AP viewer: A native DCAT-AP data catalog,” *CEUR Workshop Proc.*, vol. 2180, no. 16, pp. 1–4, 2018.

[43] PwC EU Services, “DCAT Application Profile for data portals in Europe Document Metadata,” pp. 0–39, 2015, [Online]. Available: https://joinup.ec.europa.eu/asset/dcat_application_profile/asset_release/dcat-application-profile-data-portals-europe-draft-2.

[44] R. Z. Frantz, R. Corchuelo, V. Basto-Fernandes, F. Rosa-Sequeira, F. Roos-Frantz, and J. L. Arjona, “A cloud-based integration platform for enterprise application integration: A Model-Driven Engineering approach,” *J. Softw.*, pp. 824–847, 2021, doi: 10.1002/spe.2916.

[45] S. M. Hyrynsalmi, K. M. Koskinen, M. Rossi, and K. Smolander, “Towards the utilization of cloud-based integration platforms,” *2021 IEEE Int. Conf. Eng. Technol. Innov. ICE/ITMC 2021 - Proc.*, 2021, doi: 10.1109/ICE/ITMC52061.2021.9570235.

[46] N. Ebert, K. Weber, and S. Koruna, “Integration Platform as a Service,” *Bus. Inf. Syst. Eng.*, vol. 59, no. 5, pp. 375–379, 2017, doi: 10.1007/s12599-017-0486-0.

[47] E. Thoo and K. Guttridge, “Magic Quadrant for Enterprise Integration Platform as a Service,” *Gartner*, no. September, pp. 1–17, 2021.

[48] R. J. Wieringa, *Design science methodology: For information systems and software engineering*. Springer Berlin Heidelberg, 2014.

[49] I. F. Alexander, “A Taxonomy of Stakeholders: Human Roles in System Development,” *Int. J. Technol. Hum. Interact.*, vol. 1, no. 1, pp. 23–59, 2005, doi: 10.4018/jthi.2005010102.

[50] ISO and IEC, “ISO/IEC 25010:2011(en) Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE),” 2011. https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en.

[51] The Open Group, “ArchiMate® 3.1 Specification: Relationship to Other Standards, Specifications, and Guidance Documents,” 2019. https://pubs.opengroup.org/architecture/archimate3-doc/apdxd.html (accessed Jan. 26, 2022).

[52] I. Sommerville, *Software Engineering*, 9th ed. Boston: Pearson Education, 2011.

[53] K. Schwaber and J. Sutherland, “Scrum Guide V7,” no. November, pp. 133–152, 2015.

Appendix A. Exploratory interview structure

The interview is opened with the interviewer introducing themselves and stating the objective of the interview. The interviewees are reminded at this point that there is a difference in the standard userbase of an enterprise data catalog and an iPaaS, so they that it is possible that some features are not useful from this perspective. After ensuring that there are no further questions, the interview is started with the following structure.

Firstly, the role of the interviewee in their organization is asked to provide more suitable context for comparing potential conflicting answers at a later stage. Then, in order to confirm findings of the literature, the interviewee is asked for which type and size of company the interviewee thinks an iPaaS is necessary.

After this, the main interview is conducted. Because of the semi-structured nature, the questions may not always be asked in the order below, and new questions can be added. If these yield relevant results, they are included in the results part of the report.

The questions are sorted based on the primary categories identified as features of enterprise data catalogs. This gives the following list of questions per category.

- **Searching, finding and discovering data**
    - How can a user search for something in the platform? This can be either data, a system, or an integration?
        - *If there is a limited number of search options:* Is this a design choice, or desirable functionality?
        - What is the effect of the presence or lack of search functionality on the learning curve of new users?
    - Your iPaaS has different models for different integration patterns. Does this lead to more clarity, or to more confusion for the end user?
    - Do larger customers, who have more than one iPaaS model, have a way to get an overview of the systems and integrations throughout their environments?
        - What is the reason for (not) providing this overview?
        - *If the overview is not currently offered*: Would this be a desirable feature?
    - What happens when the platform receives a message containing more fields than are recorded in the data model?
- **Governance, data lineage and impact analysis**
    - How does the platform currently provide the user with insights regarding governance?
        - Do you think the current governance features are enough or do these need to be extended?
    - Is the position of an iPaaS suitable to include governance features?
    - If an overview of data usage and interaction per environment were to be built, what would be the target audience of this overview?
- **Access control**
    - Does the platform offer the possibility to give rights on integration level?
        - Why (only) at the level described?
    - What is the meaning of the ‘confidential’ market for an attribute? Would you say you would need any other measures on attribute level?
- **Data quality**
    - Does the platform provide means to increase the clients’ data quality?

- **Collaboration**
    - Which collaboration functionalities are offered?
    - Are these functionalities often used?
    - Which features are you missing?
    - Collaboration often also involves the usage of external tools. Would you find it desirable to include these functionalities within the platform?
- **Metadata management**
    - In which manners would you say the platform facilitates metadata management?

Appendix B. Overview of vendors offering iPaaS and data catalog products

Vendor	In Data Catalog Magic Quadrant	In iPaaS Magic Quadrant	Offers data catalog	Offers iPaaS
Adaptive	✓		✓ Adaptive Metadata Manager	
Alation	✓		✓ Alation Data Catalog	
Alex Solutions	✓		✓ Alex Data Marketplace & Alex Scanner Marketplace	
ASG	✓		✓ Enterprise Data Intelligence	
Boomi		✓	✓ Data Catalog and Preparation	✓ AtomSphere Platform
Celigo		✓		
Collibra	✓		✓ Collibra Data Catalog	
Data Advantage Group	✓		✓ MetaCenter	
data.world	✓		✓ data.world	
Erwin	✓		✓ erwin Data Intelligence Suite	
Huawei		✓		✓ ROMA Connect
IBM	✓	✓	✓ Watson Knowledge Catalog	✓ Cloud Pak for Integration[15]
Infogix/Precisely[16]	✓		✓ Data360	✓ Precisely Integrate
Informatica	✓	✓	✓ *multiple*[17]	✓ Informatica Intelligent Cloud Services
Integromat		✓		✓ Integromat
Jitterbit		✓		✓ Jitterbit Harmony
Microsoft		✓	✓ Azure Data Catalog	✓ Azure Integration Services[18]
MuleSoft		✓		✓ Anypoint Platform
Oracle	✓	✓	✓ Oracle Enterprise Metadata Management	✓ *multiple*[19]
SAP	✓	✓	✓ *multiple*[20]	✓ SAP Integration Suite
Semantic Web Company	✓		✓ PoolParty Semantic Suite	
Smartlogic	✓		✓ Semaphore	
SnapLogic		✓		✓ Intelligent Integration Platform
Software AG		✓		✓ webMethods.io
Solidatus	✓		✓ Solidatus platform	
Syniti	✓		✓ Syniti Knowledge Platform	
Talend		✓		✓ Talend Data Fabric
TIBCO Software		✓	✓ TIBCO Cloud Metadata	✓ TIBCO Cloud Integration
Tray.io		✓		✓ Tray Platform
Workato		✓		✓ Workato Workspace
Legend (last 2 columns)	✓ *offered by vendor*		✓ *offered, but does not include all core features identified in this research*	

[15] Includes each iPaaS feature as a module
[16] Infogix was acquired by Precisely in 2021. Since the name Infogix is used in the Gartner Magic Quadrant, both names are included here
[17] Enterprise Data Catalog, Axon Data Governance, Data Privacy Management, Metadata Manager, Business Glossary
[18] Includes each iPaaS feature as a module
[19] Oracle Integration, Oracle GoldenGate, Oracle SOA Suite on Marketplace, Oracle IoT Cloud Service, OCI API Gateway, OCI Streaming, OCI Data Integration
[20] SAP PowerDesigner, SAP Information Steward, SAP Data Intelligence

Appendix C. Overlap of data catalog and iPaaS features

Generalized feature	In catalog	In iPaaS	Motivation
Access management	✓	✓	Both platforms need some kind of access management, for an iPaaS this can be in the form of user management
Application integration features		✓	Is not needed in a data catalog, as those use connectors to connect to data sources
Business glossary	✓		Was not found in the iPaaS platforms
Cloud service		✓	A catalog is not nessesarily needed to be offered as a cloud service, and might also be offered on-premises
Collaboration	✓	✓	Both products benefit from enabling collaboration between users. For an iPaaS, this is generally on smaller scale, without the need for approval flows
Connectors	✓	✓	Both products must have connectors to able to connect to data sources and applications
Data lineage	✓		Extensive data lineage was not found in an iPaaS, but is a must-have feature of a catalog
Data quality and modification tools	✓	✓	These are named similarly, but are different in how they are needed for each product. An iPaaS needs to be able to validate and modify data, whereas a catalog does not need to modify data but merely view it and show this into statistics
Data search & discovery	✓	◌	Data search is a must-have for a catalog, and a helpful feature for an iPaaS. Discovery is also a must-have for a catalog, but not for an iPaaS in the same fashion
Full cloud management		✓	Cloud management is not needed for a catalog
Governance	✓	✓	Governance features help fulfill the same objectives in both products, and since the products are focussed on enterprises, these features must be in the platform. Since the products are different, there are also some differences in the exact features that are needed
Impact analysis	✓		Impact analysis is a nice to have in an iPaaS, but was not seen as a feature in any of the analyzed iPaaS platforms.
Low-code UI	◌	✓	Whereas a data catalog does not generally need coding, it is important that its UI is user friendly and understandable
Machine Learning	✓	✓	Machine Learning is relevant as a supporting feature in both products
Metadata management	✓	✓	Metadata management is handled by both tools, iPaaS create metadata for the intergrations they built and the systems that are included, catalogs need to read metadata and show this to the user
Multiple integration pattern support		✓	This feature is only relevant for iPaaS
Robotic Process Automation	✓	✓	Both platforms may offer opportunites to automate workflows by using RPA
SLA & disaster recovery		✓	An iPaaS is a critical application which needs its uptime to be ensured, which is not needed for a data catalog
Store with prebuilt templates		✓	Data catalogs are tailored to an organization and the setup of a data catalog does not require advanced standardizable integrations
Supports industry-standard data exchange protocols		✓	This kind of data exchange is not needed for a data catalog
Tribal knowledge sharing	✓		iPaaS do not exchange tribal knowledge within their platform
Legend	✓ *present in the product*		◌ *present in the product to some degree*

Appendix D. Functional expert interview protocol

This interview protocol was used for the expert interviews for the validation of the functional aspects of the developed prototype. This protocol was part of the expert interview sessions, which have been conducted with 4 experts of the selected vendors' iPaaS product. The sessions were structured to start with an introduction of the interviewer and the topic, followed by a demonstration of the prototype, during which the current features are demonstrated, and the feature as it is intended to use as part of the design is explained to the expert. After this, the protocol is started.

	Background & expertise	
1	Can you describe your current function in the organization and which activities it entails?	Open
2	How many years of experience do you have in your function and in these activities?	Open
	Functional appropriateness	
3	To what extent would implementation of all of the shown features[21] solve issues your end-user currently experiences with the platform?	Scale with motivation
4	To what extent does the design of the functionalities fit within the platform?	Open
5	Which type of customer would benefit most from the addition of these features to the platform?	Categorical + open
	Categories *Small: 1 model with some integrations*	
	Medium: 1 model with numerous integrations	
	Large: multiple models with large numbers of integrations	
6	Which seniority level of users would have the most significant increase in perceived user experience when these features are added?	Categorical + open
	Categories *Junior*	
	Medium	
	Senior	
7	Which user role would benefit most from the addition of these features to the platform?	Categorical + open
	Categories *Architect*	
	Developer	
	Architect	
8	To what extent is the complexity of the platform impacted by adding these functionalities? *Scale:* *1: A lot easier/shallower learning curve* *3: Neutral* *5: A lot more difficult/steeper learning curve*	Scale
9	What would you say are the primary advantages of adding these features to the platform?	Open
10	What would you say are the primary disadvantages of adding these features to the platform?	Open
	Functional completeness	
11	To what extent would you say the functionalities, as they are proposed, are complete? And which features are you still missing per functionality?	Scale with motivation
12	Which changes or additions would make the functionalities more complete?	Open
13	How would you order each of the four features on a balance of most to least added value?	Open
14	Do you have any other remarks?	Open

Every scale is a scale of 1 to 5. Unless a different classification is given, a 1 is the lowest grade, comparable to an answer such as 'None' or 'Very low', a 5 is the highest grade, comparable to 'Very high'. A 3 is a neutral response.

[21] For every instance where this interview protocol mentions *all features* or *the features*, the following 4 features are meant, as previously introduced in section 4.2: Data lineage, Glossary, Search, and Top level data model

Appendix D.1. Functional expert interview 1

Question 1: Within my organization, I am the software delivery manager. My focus is to ensure that all features that are planned are also delivered.

Question 2: I have been working in this position for 3 years, and have been working with this iPaaS for the past 10 years. Both in a role as integration developer, i.e. active user of the platform as well as overseeing the development.

Question 3: This is different for each of the functionalities. I would scale each of the functionalities as following:

Data lineage: 4/5
Glossary: 3/5
Search: 3/5
Top-level data model: 4/5 | I would say that this is one of the features which would have more use and therefore more benefit than how a user the initial perceived usefulness a user might have.

Question 4: The design of the features fits well within the design of the platform. Some refinements could be made, but the prototype is not expected to yield the results exactly as we would implement them into our product.

Question 5: There is no one group of customers who benefits most. Even the customers who only have a handful of integrations in the platform are already of a certain size, otherwise they would not need an enterprise tooling such as our iPaaS. There would be differences in the feature which would be most useful for a certain group of users. For example, I would expect that the data lineage and top-level data model would mostly be useful for the largest customers, those with multiple models and a lot of integrations in each model. On the other hand, the very small organizations might not need an abstraction of their data model when it is still quite small, but a glossary might be helpful for them, especially when they are not working with the iPaaS on a daily basis and therefore do not have all information on top of mind.

Question 6: Junior users. These features, e.g. the glossary, affect users on the most basic level. Whereas some features are too advanced for these users, they can take this with them in their learning curve and fully adopt these features into their work with the platform over time. The experienced users might already have their own ways and methods to find data, and it might be difficult to convince them of a better way.

Question 7: This would be the architect and the developer. They obtain the most value, since they are the actual users of the platform. Support works on the management of the deployments, which was not addressed in the prototype.

Question 8: 2/5 | Adding these features ensures that the user does not have to go deep into settings and configurations, but has the means to view the information they need in a quick and easy way. Of course, these would need to be included in the training as well, but altogether they would decrease the complexity.

Question 9: The opportunity to search is a main benefit. In addition, the top level data model which provides a way to ‘zoom in’ and ‘zoom out’ helps in retaining overview of the data landscape.

Question 10: It adds another feature to the platform, and an additional way of viewing information. Users have to be able to differentiate between the different ways in which they can view the data, and which is the most appropriate for their specific use case. This might require some training.

Question 11: 5/5 | Taking into account the full scope of the features as they were shown and additions which are not yet included in the prototype, they are very complete.

Question 12: An interesting addition would be the usage of an automated tagging feature, which would provide a system, integration or entity with a tag based on how they are configured. The iPaaS currently already offers such tags, but by making them be tagged automatically rather than manually, this could further increase the overview within the platform.

Question 13: Ordered from highest to lowest:

1: Top level data model
2: Data lineage
3: Glossary
4: Search | searching is already possible by using the browser built-in 'control + f', whereas all the other features add something new to the platform.

Question 14: - No further remarks -

Appendix D.2. Functional expert interview 2

Question 1: My function is that of CTO, although that does not describe clearly what I do. I am responsible for the tools and technologies used to run our iPaaS. Within this role, I do not build integrations using our iPaaS, although I used to build integrations without an iPaaS before I started my work here.

Question 2: I have been active in my position for about 10 years.

Question 3: 5/5 | If the features would all be developed into our iPaaS this would definitely provide solutions to problems of customers. Both from directions which are on our roadmap as well as based on requests our clients provided to us. As for the search functionality, I never heard this request from a user, although I can see the added benefit of it for users who are less familiar with a model.

Question 4: This differs a bit as your developed four features. For the glossary, the information can be put in on the location where I would expect it, and it is shown at relevant location. The search bar is placed at a sensible location. Since it is located in the menu, it insinuates that it searches the entire model rather than the currently open page, which is indeed the intended functionality. The top level data model could use some polishing with regards to how you switch between the data models of the top level and for those of each integration pattern. The buttons are currently quite hidden. Finally, the lineage is currently too hidden. This would deserve a more prominent location. It is not a niche feature, but rather a main feature. It should have a place in the top menu. In addition, additional visualizations could be added.

Question 5: All clients | Although the medium and large clients would benefit most from the top level data model, since they have significantly larger data models than the smaller clients, features such as the glossary are more useful to small customers, since I know most of them do not work with our iPaaS on a regular basis, which might they need more referencing to recorded information.

Question 6: Medium-level users | New users still have a lot to learn about the platform. These new features would be even more for them to learn, and these features would not be needed for them to start off in the iPaaS. On the other hand, the expert users are likely to already have their own means of obtaining the information these new features provide, or have them on top of mind. Therefore, the user group which would benefit most would be the medium users. They are ready for using some more features than the new users, and do not have the model-level expertise of the expert users. At the same time, there is also a difference in the adoption of each of the features. The search feature would likely be the first feature to be used by all user groups, but features such as the top level data model might have a steeper adoption curve.

Question 7: Again, I expect the different users to have different features which they would primarily use.

> Architect | Would benefit most from data lineage and the top level data view, as this fits within their function description
> Developer | Would primarily use the search and glossary, as they are actively working with the iPaaS and might have to reference back to how a certain integration was designed.
> Support | Does not benefit from the features as they are currently designed. This would require extending the features further into the dashboard and log entries they use, which the prototype does not currently include.

Question 8: Once again I would have to give a different answer for the different features. Both the search and the glossary facilitate a decrease in the complexity, so I would give a 2/5 for these two functionalities. The data lineage and the top level data model, on the other hand, add new

functionality to the platform and therefore give something more to learn. So I would say that these make the iPaaS a bit more complex, and would therefore rate these 4/5.

Question 9: As for the glossary, it simplifies recording information of systems and integrations and therefore also motivates the usage of this option. Currently, we see that some customers choose to not record any information at all for their systems and integrations. As a USP, the top level data model is a strong potential selling point. This can help architects in monitoring and safeguarding data structure within a model or within an organization. To me, data lineage is an extension to this, which can also be used to get answers on details of the integrations.

Question 10: Especially the glossary feature really build upon existing information, but the user is not required to fill in any of this data. Therefore, a client is not guaranteed to obtain benefit from adding these features to the platform; if they did not provide any information, this can also not be visualized. I do think that it is important, however, that the speed of developing is not impacted by providing the developer with a lot of required fields which need to be filled in before they can proceed.

Question 11: It is more appropriate to rate each feature separately:

Data lineage: 4/5 | The current feature shows a good start, but immediately gives a lot of inspiration for potential additions based on this feature.
Glossary: 4/5 | More could be done, for example with different questions and removing some of the free text fields, since these are not always needed. I would also like to see this feature extended to the management parts of the platform, so they can also be used by support.
Search: 5/5 | Although the feature in the prototype does not show a fully functioning search functionality, the mock-up and the way in which the results are ordered and located is perfect.
Top level data model: 4/5 | This is quite complete, but could use some further extension, by for example showing more of the cohesion throughout the data models for different integration patterns. In addition, switching between the models should be done through a clearer switch button.

Question 12: I would integrate data lineage more prominently into the platform, with a dedicated menu item for example. It is currently too hidden. I have no direct comments for the other features.

Question 13: This strongly depends on the type of user, as I also outlined in some of the other questions. For me personally, I do not make use of the information, so the glossary would not add too much benefit for me. Therefore, my personal list would be:

1: Search
2: Data lineage
3: Top level data model
4: Glossary

Question 14: The features are currently very concentrated on the developer point of view. This could be extended to the management pages of our iPaaS, to also benefit the support staff of our platform. This would not even necessarily need a change in the features as they are designed now, but rather making them accessible in other pages.

Appendix D.3. Functional expert interview 3

Question 1: I am the Product Manager of this iPaaS, which means that I am responsible for the full lifecycle, including development, sales and quality assurance.

Question 2: I have been working in this role for the last 2.5 years, with numerous years in other roles prior to this.

Question 3: 5/5 | This is without a doubt. These features are features I have received numerous customer requests for.

Question 4: Most of the features, such as the top level data model, are placed in such a way that they seem to be integrated, and taking the perspective of a user which is new to the platform, the feature is placed where I would expect it. Regarding the enablement of the 'tags' based on the input in the glossary, a clearer division would be needed between the manual tags and the tags which would be automatically assigned.

The search functionality is incredibly useful and I can map this one to one with customer demand. It would be topic for debate what the search functionality should entail, only the information within a model, or also the documentation. This would have to be researched carefully.

Finally, the data lineage feature is incredibly useful and understandable from the platforms design point of view. I just have my questions regarding the position of the feature. This is currently put on a page where I would not directly expect it. Some further discovery might be appropriate as to what the most suitable position for this feature would be.

Question 5: considering that you describe *small* customers as those with small models, they would benefit the least from the features, given that their landscape does not need abstraction of their data model, and is still quite comprehensible because of the compact size. Medium and large customers would definitely benefit from the features a lot. It is a bit debatable how little the small sized customers would benefit, since they would benefit from the search functionality, and in the trend of our customers we see that they generally do get more and more integrations, and therefore might grow to become medium customers, at which point they would obtain more benefit because of their increased platform complexity. I think therefore all customers would ultimately benefit from the features.

Question 6: The new users are not yet at the stage where they would need these features, maybe with the exception of the search functionality. The features are mostly useful for customers with already existing, more complex, environments. Therefore, the medium and expert users would benefit the most.

Question 7: Two of these groups would benefit, obviously these are the architect and the developer. Since the current prototype does not show the features applied in any of the monitoring pages, support does not gain benefit from the features, although if the shown features would be included into those monitoring parts of the platform with some small adjustments, support could also greatly benefit of them.

Question 8: 1/5 | Even though you are actively adding features to the platform, I would say it gets a lot easier to use. Especially the search function and the data lineage are strong aids for the customer in building an integration and finding what they need, so that the platform guides them through what they need.

Question 9: The main advantage would be the possibility to combine all data entities, regardless of their origins, into a single data model. In addition, it ensures that all sub models can be found at a place where they would be expected. The data lineage, in addition, ensures that changes can be made and incorporated easier. By being able to deliver integrations faster, which these features clearly enable, and needing less rework before approval increases productivity. The features support the user in getting their work *first time right*.

Question 10: If I would have to point some things out it would be on clarifying the search functionality, which might be worthy of a research of its own, since there is one chance to really release the feature and get the level of adoption it deserves.

Question 11: 4/5 | Apart from the placing of the lineage feature, everything is clear and looks like it belongs to be in the platform like this.

Question 12: It would be good to clarify that data lineage shows all integration types side by side, and does not only show integrations of an entity per integration type. Just by including this in the prototype would already make it more convincing to me with regards to the lineage functionality.

Question 13: I would rank them in the following order:

1: Data lineage | I rate this functionality the highest since I have numerous customers ask me for such a feature in the past period. In addition, this helps the developer by providing tools from the platform side in doing an operation which usually is quite risky.
2: Top level data model | For the architect stakeholders, this would be the number one functionality, so this one is also very high on the list.
3: Search
4: Glossary

Question 14: 4/5 | If you were to change the position of the data lineage feature, this would be a 5/5.

Question 15: I found it remarkable that in your initial analysis the impact analysis came out as *relevant but no priority*. It overlaps with questions of customers who are now performing manual impact analyses.

Altogether, these features would be a good way to pull our platform to a higher level.

Appendix D.4. Functional expert interview 4

Question 1: My role is that of Product Owner of multiple teams. That means that I ensure focus of the teams and manage stakeholders, but do not do any development myself.

Question 2: I have been working in this role for a little over a year now. Before that, I worked on platform development.

Question 3: 4/5 | Out of the four features you proposed, I would say that both the search and data lineage functionalities are not requested by users, but there is a strong need for them. Regarding the top level data model, I would say that there is a strong need for this feature, and it is also requested by users. The glossary is a bit harder for me to estimate. There is no active demand from the users, and I am not sure that the glossary as it is designed currently would be used thoroughly.

Question 4: The search is extremely well integrated. I have my doubts about the placement of the glossary, it currently repeats the information entered previously, it might benefit from a more central position in the platform. The top level data model is placed adequately, but the data lineage feature might be more appropriate as extension of existing features in the platform rather than including it as a new feature.

Question 5: I would say that medium to large clients would benefit most, although small customers might also benefit from the search functionality and the opportunity to zoom in and out using the top level data model, since they might not be working with the platform as often as the medium and large clients. Although this feature might not include all segments of customers, this is not too big of an issue, as the features we develop are also focused more on medium to large clients as they generally have a larger need for additional features.

Question 6: I would say that the average user would use these features the most. For the new users, the concept of the data model might already be too much, and therefore the top level data model would bring them more confusion than clarification. They might benefit from the search and glossary, though. As for the expert users, they already know their way through the landscape and would have a decreased need for the glossary and the search functionality. Data models are more relevant for them. Generally, I would say that quite some users of our platform would be expert users.

Question 7: This is difficult to say. Support is left a bit out of the picture since the features are not placed in positions where they would look at most often, but with some slight changes I could see the glossary, search and data lineage features as quite useful to them. Between the architect and developer, the glossary and lineage features are be useful to both of them. Regarding the top level data model, I expect the architect to have more need for this. The developer, on the other hand, would use the search more extensively to also navigate through the platform.

Question 8: I would say the overall platform would be less complex, but the onboarding process might be impacted a bit by the addition of these features, since this is quite a lot of change at once. On a feature level, I would say that the data lineage and glossary features as they are now would add some complexity, the search functionality would significantly simplify the platform and the top level data model would make it a bit easier. Therefore, I would give this a 2/5.

Question 9: Personally, I would really like to see the search functionality. Regarding expectations of our customers I would say that the top level data model adds the most significant benefit. It removes a big pain point in the management of large data models. I would say that data lineage would be a part of this top level data model, with a more integrated experience.

Question 10: The glossary as it is designed now still contains quite some questions for every integration and system. Therefore, I do not see the added benefit of it at this point. This might need some further extension. A side effect of the search functionality could be that people would be using the search as a way for quicker navigation, which might make them forget how to navigate to the feature using the normal navigation options.

Question 11: 4/5 | Given that this is a prototype, I would say it is very complete. Combined with the story about how the features would each be supposed to work if implemented completely, there would be some points which might need further research, especially user testing, but the basis is really solid.

Question 12: Regarding the top level data models, some further visual clarifications might be needed to display in which integration pattern the entity is used. In addition, it is a bit vague to me how it would work if there are different relationships in different data models.

Question 13: I would rank each of the features in the following way:

1: Top level data model | this feature is requested a lot and solves a real issue customers experience, and is therefore the most important
2: Search | I would say it facilitates the lineage, and it is a nice feature for the platform overall
3: Data lineage
4: Glossary

Question 14: 4/5 | Only the glossary I would really change, for the rest it matches with what I would expect.

Appendix E. Usability expert interview protocol

This interview protocol was used for the expert interviews for the validation of the usability aspects of the developed prototype. This protocol was part of the expert interview sessions, which have been conducted with 4 experienced end-users of the vendors' iPaaS product. The sessions were structured to start with an introduction of the interviewer and the topic, followed by a demonstration of the prototype, during which the current features are demonstrated, and the feature as it is intended to use as part of the design is explained to the expert. After this, the protocol is started.

	Background & expertise	
1	Can you describe your current function in the organization and which activities it entails?	Open
2	How many years of experience do you have in your function and in these activities?	Open
	Usability	
3	Do you think the position of each of the shown functionalities[22] is logical?	Scale with motivation
4	If you were to rank each of the functionalities, from most value added to least value added, how would you rank the four features?	Open
5	**Glossary**: To which extent do you see added value in displaying previously recorded system, integration and entity details at each occurrence of the system, integration or entity?	Scale with motivation
6	**Search**: To which extent do you find the search functionality a useful addition to this iPaaS?	Scale with motivation
7	**Top level data model**: To which extent does your organization have a need to be able to view data entities on a more abstract level, such as shown through the top level data model?	Scale with motivation
8	**Top level data model**: The current focus of the top level data model is to visualize the data entities within a model. Would you also like to see this extended to all models of your organization? *[only for interviewees who work with multiple models]*	Closed with motivation
9	**Top level data model**: How does your organization currently retain an overview of their data portfolio?	Open
10	**Top level data model**: Which additional features would you suggest for the top level data model, other than the features already shown?	Open
11	**Data lineage**: To what extent does your organization have a need to obtain an overview of which integrations and systems which process and modify an entity, as shown?	Scale with motivation
12	**Data lineage:** Which additional features would you suggest for the data lineage features, other than those already shown?	Open
	General	
13	To what extent would you recommend the vendor of this iPaaS to implement the shown features?	Scale with motivation
14	To which extent would the implementation of these features within the iPaaS make other tools your organization uses to visualize the application and data landscape less needed? *Scale used:* *1: There is no overlap at all between the proposed features and currently used external tools* *3: There is some overlap between the proposed features and external tools, but not enough to replace external tools* *5: Implementing these features into the iPaaS would replace some currently used external tools*	Scale with motivation
15	Do you have any other remarks?	Open

Every scale is a scale of 1 to 5. Unless a different classification is given, a 1 is the lowest grade, comparable to an answer such as 'None' or 'Very low', a 5 is the highest grade, comparable to 'Very high'. A 3 is a neutral response.

[22] For every instance where this interview protocol mentions *all features* or *the functionalities*, the following 4 features are meant, as previously introduced in section 4.2: Data lineage, Glossary, Search, and Top level data model

Appendix E.1. Usability expert interview 1

Question 1: My function title is that of enterprise architect. I do this in for an construction enterprise, which has multiple subsidiaries for each of their disciplines. Within my role, I am responsible for the information and data architecture for all of the subsidiaries. In work in a team with other enterprise architects.

Question 2: I have been working as an enterprise architect for about four years. Before that, I worked on the other side of the scope where I developed integrations.

Question 3: 4/5 | The position of each of the functionalities is logical. For example, you placed the search bar all the way at the top. This is where I would expect to find it, and its position also indicates that it searches not only in the current page, but throughout the model.

Question 4: Ranked with one as the most added value and 4 as the least, I would rank them in the following way:

1: Search | This would be a feature which I would use the most. Since the search feature is currently not included in the platform, I use the browser built-in 'control + f' to find what I want, but since our models are quite extensive, this does not always give me the results I want. In addition, this only lets me search on the currently opened page. The search functionality, as it is shown in your prototype shows what I want to find regardless of whether I search for abbreviation or full name, and shows where it is found before I have to go to it.

2: Data lineage | This would be a feature which has a lot of potential. Currently, this information is not shown explicitly anywhere within the iPaaS. You would have to rely on your knowledge of the model. Generally, I am quite aware of in which integration certain data objects are used, but some of my colleagues might now have this on top of mind, especially when a certain model has not been their focus for a while. This feature ensures that all relevant integrations and systems are displayed, and you do not need to rely on your own memory with the risk of missing information.

3: Top level data model | The top level model as it was shown is very useful, but for me personally it would come after the lineage feature. It has a lot of potential to display our data models in an overview, which ensures that information is more findable, without losing the level of detail currently offered by the iPaaS.

4: Glossary | For my work, I do not generally depend on the information recorded about systems or integrations. Since I have some seniority in my function I am quite aware of the meaning of systems and integrations, and of their usage. I expect that this feature would be more relevant to people actually building the integrations.

Question 5: 3/5 | I generally do not record or use any of the information regarding systems or integrations. It is therefore hard for me to say how useful I would perceive this functionality. I can see the potential benefit, but it would be better to ask an integration developer for a better opinion on this feature.

Question 6: 5/5 | As I motivated previously, the search functionality would be used in my day-to-day work and has the potential to really improve how I work with the iPaaS.

Question 7: 4/5 | Our models are quite extensive, which means that in the current general data model, a lot of the relationship lines overlap with the entities shown in the data model. This requires a lot of zooming in and out to be able to read the model properly. Currently, we use ArchiMate to map data entities in each of our models to business entities as we recorded them. The top level data

model could have a similar function for us, where it enables us to zoom in from the business entity level to data level.

Question 8: Yes | Our organization also has integrations which are not within this iPaaS platform, but which might be relevant to connect to systems who do reside within the iPaaS. It would be great to be able to connect this information to the iPaaS. In addition, our organization aims to retain a similar core entity set on which each model builds. Having the top level model integrate data models from our various models, this would help in ensuring that our data models conform to our general design.

Question 9: Our team of enterprise architects uses separate tooling to model the entire application and data landscape of our organization. Each of the models we create has references to other models, or to our iPaaS to refer to a more detailed view.

Question 10: The data model currently does not show cardinality[23]. This is critical information when I assess a data model. As a more general remark, I would like to be able to assign a key identifier for an entity, but this is something that the platform in general does not support at the moment, not so much regarding the features shown.

Question 11: 5/5 | We especially care about where certain data originates from. On organizational level, we would benefit from this since we do not always have a clear data owner because of the way in which our organization is set up. Therefore, we would greatly benefit from seeing data origin and where the data flows. This is especially the case for data which flows to external systems which are not under control of our organization. This information is currently not very visible in the iPaaS because of the large size of our models. Therefore, we have to rely on the information from the memory of our architects, and that is not the most desirable situation.

Question 12: It would be great if the lineage feature could show integrations or systems who have a high CIA-score[24] in a different color. This would also be an addition to the glossary, which would in turn need to ask questions to assess the CIA-score.

Question 13: 4.5/5 | These features make my work a lot easier, for example by preventing that each individual integration has to be investigated in order to see data usage. This would also help my colleagues who did not create a certain integration. Therefore, these features help in making the iPaaS work more transparently, and make it less of a *black box*. It would also motivate us to match the iPaaS more with our external tooling.

Question 14: 3/5 | The proposed features have overlap, but the current tools we use remain important to use and cannot be replace with these. The external tools are also used to collaborate with other actors who do not have access to the iPaaS, such as application administrators. In addition, the external tooling is used for communication with business stakeholders for information such as contract period, lifecycle, and other information which we do not want to record in the iPaaS. It absolutely makes my work easier, but does not replace my external tools.

Question 15: I would be willing to use the iPaaS to model integrations which do not use the iPaaS for their integration if the iPaaS would be able to show me the same information as you showed in your prototype about these non-iPaaS integrations. It would be important to me that I would be able to

[23] Within data modelling, *cardinality* describes the relationship between two entities. For example, an employee has one address (one-to-one), but can have multiple cars (one-to-many).
[24] CIA in data security stands for the three aspects of securing data: Confidentiality, Integrity and Availability

export this information in a non-proprietary form, such as an image or PDF file so I can share this data also with users who do not work in the iPaaS.

Appendix E.2. Usability expert interview 2

Question 1: I am a lead consultant and solution architect for larger clients of my organization. I work together with the architects of the client to maintain the overview of multiple iPaaS models.

Question 2: I have been working as a consultant for 6 years, and the last two years I have been active as a solution architect as well.

Question 3: 4/5 | Each of the features by themselves is positioned on a logical place. I am a bit confused about how the search would work precisely, as to where it navigates you when you click a result.

Question 4: The top 3 I give are the features whose usefulness is immediately clear to me based on your prototype.

1: Search

2: Glossary. I would expect this feature to be linked with the search functionality. It can provide a lot of information regarding a search result, for example providing the description of a system when you search for it

3: Top level data model. It is very important to be able to visualize the connection between different data models and to improve the overview

4: Data lineage. I primarily give this the 'lowest' ranking because it is not quite clear for me yet how this feature would be used. This could perhaps be cause by me not working with the iPaaS on a daily basis, I expect that these users would give it a higher rating.

Question 5: 4/5 | It improves the workflow of the developer by removing the need to keep a second browser tab open to access this information whilst simultaneously working on an integration. This is a feature which should have been available already.

Question 6: 4/5 | This is a very powerful tool, especially to be used while in conversation with the business stakeholders and you need to make a quick validation upon whether the data they are mentioning is already being used or mentioned within the platform. I would give this a 5/5 if the search would not only work within a model but for all information of a client, regardless of the model it resides within.

Question 7: 3/5 | It is difficult to estimate this for my client. Currently, they are mostly using one type of integration pattern, which this iPaaS displays relatively clearly. In addition, the data models are quite normalized and therefore not too extensive. I do know that there are some clients of my organization which would have a higher need for this functionality.

Question 8: Yes | This would be a strong yes. The number of environments our large clients have is increasing, and therefore there is inevitably overlap in entities used throughout these models. It would be very useful to be able to combine these and see in which model they reside.

Question 9: There are a limited number of external tools used to provide an overview, but most information is within the iPaaS.

Question 10: I would like to be able to drag the entities around, just as is currently possible in the other data models, and to give them a color-code to group them together. Maybe the tagging as the iPaaS already offers in some other places can also be applied to entities. It would be even better if these tags would be automatically assigned.

Question 11: 4/5 | I can certainly see the benefit of this feature now. I would like to see the opportunity to add context to this. As I indicated previously, my client has a normalized data model,

which means that a data object can have different meanings, and also different systems it flows to based on its context.

[interviewer notes: prior to this question the objective of the data lineage feature has been explained in some more detail, since the interviewee did not yet have a clear understanding and was therefore unable to answer the question, but from their role their input was expected to be valuable. The additional explanation was done using the feature as it was shown in the prototype and focused primarily on the way in which the feature obtained its data.]

Question 12: As indicated, I would add context awareness, even the option to do so manually. In addition, the combination with data previously entered in the requirements capture section would be useful to show here. A connection to the search, and the option to navigate directly to a lineage page for a certain entity would also be valuable.

Question 13: 4/5 | I can see benefit of these features for a lot of projects. Currently, many clients' models require a lot of scrolling and zooming to view the information. In addition, adding these features reduces the need of exclusive knowledge to understand how data flows through the iPaaS. I would see these features as some kind of process explanation, which helps users gain insights into how the iPaaS works. I would also expect these features to prevent the development of duplications throughout clients' models.

Question 14: 3/5 | I would say that some overlap exists, but not enough to replace the external tooling. As I mentioned previously, there is not a lot of external tooling being used by my client currently. The tooling they are using for the simplified landscape architecture overviews cannot be replaced one to one by these features. Some aspects are just not visible when exporting a view from the iPaaS, which can be visible in an external representation.

Question 15: For further improvement of the features, perhaps user feedback can be used for preventing the need for shadowing with diagrams in external tools. Also, ensure that the search functionality is thought through extensively. It might be quite complex to get to work, and should not provide over-information. The best method to develop this search method alone might be worth a research of its own.

Appendix E.3. Usability expert interview 3

Question 1: I am an integration manager of one of our clients. This means that I develop integrations within the iPaaS on a daily basis.

Question 2: I have been working in this position for 3 years now.

Question 3: 5/5 | The placing of the features is exactly where I would expect to find them. Especially the change as compared to the current situation, where the data models of different integrations are accessible through a single page rather than being 'hidden' in different section of the iPaaS. The position of the search bar at top level also clearly illustrates that this is indeed the functionality it is intended to have: search through all information within the model.

Question 4: I rank each of the features based on the relevance for my role as integration manager.

1: Search | This is by far the most relevant feature for me. It can help me find what I need faster.

2: Top level data model | I often use the data models. It is very useful to have the different models accessible through a single page, and also have a top level one which provides me with a better overview.

3: Glossary | Generally, I am the one either entering this information or checking it. Therefore, I do not often need this info as a reference, since I have it on top of mind.

4: Data lineage | I do not expect to need this functionality a lot in my function. This would perhaps be used more by architects.

Question 5: 4/5 | The glossary as you integrated it into the platform encourages the users to fill out the integration and system details in a better way. Currently, I notice that quite a number of users do not fill out this information. By ensuring that this information might become beneficial by showing it in other pages as well, this gives more motivation. In addition, showing this information on other pages prevents the need of having to switch between pages and it would therefore improve the speed at which I can do my work. I would like to emphasize that the information should be displayed as read-only fields in the phases other than the requirements capture, since making them modifiable outside of this phase would disturb the development flow and I would expect that to reduce the number of users filling out this information.

Question 6: 5/5 | A proper search functionality is very beneficial to the platform. Even though it is not a fully functional feature in your prototype, the way in which you display the results is exactly what I would expect from my search results. Maybe a suggestion to make it even more useful would be to also be able to find the portals documentation, as well as the objects and definitions within my model. Of course, it might be difficult to display this clearly, but however difficult this might be, a good search functionality should be able to do so. But even in its current form I would already be able to use the search bar as a *hack* to navigate myself through the portal more quickly. Rather than going to a feature which is hidden behind multiple clicks, I can just search for it and instantly open it.

Question 7: 3/5 | I do not notice an immediate demand from my client, but I do regularly get a question regarding the available integrations. I know that there are other clients at my organization who do have a significant need for this feature. As for my client, it would be more useful when their integrations are using more different integration patterns. They are currently in the process of increasing the usage of other integrations patterns. So the need is going to be there, but the question is when they notice this need.

Question 8: Yes. My client has multiple models and it is not ruled out that they will have more models in the near future. This also means that there will inevitably be some intra-model

communication, which would require overlap in the data models of the different models. For every client with multiple models, it is going to be useful at some point to align the different data models to be more similar at core level. This would also affect the data lineage, which would also be a great feature to apply to multiple levels.

Question 9: The usage of external tools is limited. We receive drawings which are made outside of our iPaaS for new integrations or features our client would like us to develop, but as for the general landscape, the iPaaS is mostly used. We provide or clients with exports and screenshots from the iPaaS from time to time, to give insights into the data flow. I am not aware of other more extensive usage of tools outside of the iPaaS to retain an overview.

Question 10: I am already quite enthusiastic about the features shown. Perhaps a small addition such as the cardinality would be nice to show as well. But I can also expect situations where I would like to hide these as well, as you currently have. Therefore, I think it might be best to use a toggle option for this, to turn it on or off.

Question 11: 3/5 | For my client, I would not say there is a direct need for this. It does have the potential to get more important. This is similar to what I previously answered regarding the top level data model. The data lineage, in contrast, also has the potential to be included in the risk inventory. Also, I see myself using this feature to check where certain entities are used and how they flow through the system.

Question 12: I would like to see how data flows through the iPaaS, for example that an order is produced by application A, goes to application B, which creates a form output in application C and triggers application D to create an invoice. This would pretty much describe data lineage as you currently have it on process level, i.e. going over multiple entities. That would really help new users of the system to understand what is going on inside the iPaaS. In addition, our client has a team of people with iPaaS access who act as a *fixing crew* for bugs. Such a process overview would also be very valuable for them

Question 13: 4/5 | The proposed features would be very beneficial for our users and provide maturity for the platform. It would facilitate the growth to prospects who have even more integrations than our current large customers.

Question 14: 2/5 | I am not fully aware of the tools that are currently used by my clients' architects to maintain an overview of the landscape. Generally, I think there is a low overlap of the external tools that they are using and the additional insights these features give.

Appendix E.4. Usability expert interview 4

Question 1: My role is that of integration consultant. For this, I am working with one of our large customer to develop and maintain integrations using the iPaaS, with an occasional other project.

Question 2: For three years I have been active as an consultant, of which the last two years I have had this role with the daily usage of the iPaaS.

Question 3: 4/5 | All functionalities are where I would expect them. I just have my questions regarding the position of the search bar. To me, its current position implies that it also searches through the documentation. I do not know to which extent this should be wanted.

Question 4: Based on my role, I would rate the functionalities as following:

1: Data lineage | Apart from the fact that the data this functionality shows is currently very hidden in the platform, it would also help me to assess the impact of a change I would make. I expect to use this feature on a daily basis.
2: Top level data model | This is a feature which is really new, and provides an insight which was not available previously.
3: Glossary
4: Search

Question 5: 4/5 | I would say this highly depends on how well the requirements are filled out. Currently, there are numerous integrations without the information properly filled out. On the other hand, having the information be displayed throughout more places in the iPaaS would certainly help motivate users to fill out the information better. This would be even better if this information was used throughout even more features, or if automated recommendations could be made. In addition, but maybe not within the scope of what you are trying to research, you might want to look at how this information could be monitored and adjusted automatically. For example, you fill out that you expect an integration to have 500 uses per hour, but if you see that this integration has an actual usage on production with a mean usage of 1000 uses per hour over 2 weeks, you might want to show this automatic measure as well so you can ensure that the information displayed remains relevant.

Question 6: 3/5 | As indicated previously, I currently find it difficult to assess. My first association with the position the search bar is currently displayed is that I can use it to search through documentation, which is just what it is not intended to do. I would be curious how the search results could be shown in a meaningful way on a production environment. For example, I know that a model could have several hundreds of hits on a search query such as ‘employee’. How would the search functionality be able to differentiate between these, and give me useful results rather than a list of some hundreds of hits?

Question 7: 5/5 | The top level data modelling functionality would be very useful, especially for less experienced users of the iPaaS. This would help them in better deciding whether a new entity is needed, or if there already exists such an entity. Although there are checks and balances in place, I still notice that this does happen from time to time. Having such an overview would significantly improve the speed at which a user can find existing entities and therefore be able to develop their integrations quicker as they need to spend less time on checking the availability of a data object.

Question 8: Yes, the top level data model is very helpful for architects as well as developers. Developers are able to quickly check the availability of a certain entity and which information might be connected to this which might also be useful to them. Architects have an important objective in keeping models aligned with their design. Especially in the case of clients who have multiple iPaaS models, being able to overlay the data objects of different models is very useful in checking whether

the models are developed conform the requirements the architects set, and upon creating alignment throughout different models.

Question 9: There are some Excel sheets being used which document attributes and everything which is shown in the data models. This includes a description of each attribute and entity. Enterprise architects of my client also apply this vice-versa: make recommendations and decisions based on the information which is registered in the data model of the iPaaS. There are not too many external tools being used as far as I know.

Question 10: I would recommend to take into account how you can make the difference between the origin of the data objects. The iPaaS supports up to three different data models, I am not sure as to how the top level data model currently retains the connection to the originating data model other than combining them into a single view.

Question 11: 4/5 | With my client, I have had conversations about being able to do just this, but then manually, multiple times. This feature would really help me in doing my work. It would be a 5 out of 5 when you would also be able to see which attributes of an entity are accessed or modified.

Question 12: Other than the visibility of actions on attribute level rather than entity-only as mentioned in your previous question, it would be even more useful when rather than (only) displaying based on entity level, it could be applied to process level. In this case, I imagine that I would be able to trace the flow and lifecycle of a message, seeing precisely where it is produced, transformed, triggering another system to produce a message, etcetera. This would really add value to the platform for users of all levels and all roles.

Question 13: 4/5 | Especially the top level data model is the feature which gets me enthusiastic. But also the lineage and the other features would be useful. As indicated, I would do a proper in-depth investigation of how clear the search would be prior to building it.

Question 14: 3/5 | As I mentioned previously, there are not too many external applications being used to my knowledge. The changes you propose are somewhat overlapping, but I do not expect these external means to disappear any time soon.