UiO : University of Oslo

IS artifact embedded in an enterprise application to facilitate data collection adapted for machine learning and span software developers and data scientists expertise

Viorica Fluer

Informatics: digital economy and leadership

Master Thesis 60 points

Department of informatics

Faculty of mathematics and natural sciences

UNIVERSITY OF OSLO

Date: 16.05.2022

Abstract

Software companies aspire to deliver the best software solutions and leverage the latest technologies to satisfy their customers needs. As big data and machine learning are known to bring value and propel businesses to flourish, companies are trying to remodel and become Data Driven Organizations (DDO). But companies face multiple challenges on their path to become DDOs and not all organizations manage to fully benefit from being DDOs. One of the challenges is not having relevant data which can help data scientists to explore practical enquiries and create accurate models. At the core of data collection are software applications storing data. Software developers are continuously developing and improving the existing software applications, on the other side data scientists struggle to analyze the data and seek to derive meaningful insights from it.

This design research proposes to use an innovative artifact as a tool to investigate how software developers and data scientists interact and revise the lack of coordination between them. Changing the relation between data scientists and software developers has the potential to improve data collection and ensure access to relevant data. Moreover, exploring the relationship between data scientists and software developers through an artifact becomes more practical and can drive new ideas for improvement. These innovative ideas have the potential to redefine the organizations' processes and support companies to gain from being DDOs.

The approach used to conduct this study was to frame the paper as a Design Science Research (DSR). DSR provides mechanisms which help create knowledge through designing and building of the innovative artifact. The artifact proposed for this research was designed and built following the DSR method. This research proposes a set of principles to be adopted when creating a similar artifact. The following principles were established: principle of availability, principle of following IS standards, principle of using the artifact as a boundary resource.

Contents

Acronyms

ADR	Action Design Research
AI	Artificial Intelligence
API	Application Programming Interface
DDO	Data Driven Organization
DSR	Design Science Research
DSRC	Design Science Research Cycle
ERP	Enterprise Resource Planning
ETL	Extract, Transform, and Load
FEDS	Framework for Evaluation in Design Science
GDPR	General Data Protection Regulation
GUI	Graphical User Interface
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
II	Information Infrastructure
IS	Information System
IT	Information technology
JSON	JavaScript Object Notation
KPI	Key performance indicator

ML	Machine Learning
REST	Representational state transfer
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
XML	Extensible Markup Language

List of Figures

List of Tables

Glossary

alpha version - a version of a software application which is released to be tested

beta version - an early version of a software application that contains major changes which is released for testing

components' robustness - a software component's ability to face perturbations and ensure a well functioning

data driven - a business activity's progress is managed by data, and not by human instincts

debugging - software development activity to identify and remove the error from a software application

graphical user interface (GUI) - application front end/ interface which facilitates the interaction with the users

metadata - a group of data which provides information about another data

practitioner - a person employed to work in a discipline/ field/ profession

software architecture - a high level pattern utilized to build and develop a software system

third party code - represents the code written by external parties

third-party developer - is a developer which has not written code directly for the primary system

toolkit - an assembly / set of tools; foundation set to create an user interface

use case - a specific scenario in which an application, product can be utilized

web application - a software application which runs in a web browser, as a request

web service endpoints - a resource / entity which handle different web service messages

Acknowledgments

First, I would like to express my gratitude to my supervisor, Alexander Moltubakk Kempton for guiding me throughout this research and inspiring me to write this master thesis. I am thankful for your feedback and knowledge you shared with me. I appreciate your effort, it has helped me to develop and finish this research.

Second, I would like to thank my dear friend Behrouz, for his patience to listen to all my ideas. Many times I have drawn inspiration from our long discussions. I would like to add my appreciation to all my family members, especially my mother Maria and sister Ludmila who have supported me in this journey. Your discussions and multiple questions helped me to delve deeper into some of the research topics.

Introduction

Recently data collection has become a point of interest for most Information Technology (IT) companies (Kim et al., 2016). To become data driven is part of the companies' strategic portfolio (Mohanty & Vyas, 2018). Developing a data driven culture is one of the aims companies set on their path to become Data Driven Organizations (DDO) (ibid.). The reason why companies aspire to become data driven, is the benefits big data and Artificial Intelligence (AI) can bring (ibid.). Among the benefits are the potential to make business operations efficient and lower the costs (ibid.). The ability to improve business operations and reduce costs, give companies competitive leverage (Werder et al., 2020). Therefore, organizations are so eager to adopt a data driven approach (ibid.).

Although, becoming a data driven company is generally recognized as beneficial for the business, multiple studies have also shown that companies face various challenges when it comes to benefiting from being a DDO. One of the challenges noted by Muller et al. (2019) is lack of relevant data. In order to derive value from data, companies have to focus on what data they have, how to best use this data and generate more relevant data (Muller et al., 2019). This observation has also been noted in Kim et al. (2016) paper. Kim et al. (2016) assert that DDOs face multiple challenges when it comes to data related tasks (Kim et al., 2016). To understand the problem of missing relevant data at its root, I propose to examine data collection tasks' struggles.

Data collection plays an important role for Machine Learning (ML) training sets (Mohanty & Vyas, 2018). Machine learning models are developed to improve predictions through experience by employing data (Arankalle, 2020). For instance, companies utilize ML model's predictions to make strategic decisions for business operations (Werder et al., 2020). To develop reliable ML models it is necessary to have good training sets (Arankalle, 2020). In order to obtain a training set, vast amounts of data have to be stored, cleaned and adapted for the Machine Learning (ML) model (ibid.). Bean (2022) asserts that in data science having good data is vital to make adequate predictions. This study explores a strategy to acquire relevant data for ML purposes by employing an artifact. Morabito (2015) explains how expanding the products by embedding tools, can help generate big data. I have followed the same approach to design the IT artifact to be embedded in software applications to collect data.

Data is stored and collected from different software products (Kim et al., 2016). The software products are developed by software development teams (ibid.). Kim et al. (2016) remark that software companies are in need of “data scientists with analytical and software engineering skills” (Kim et al., 2016, p. 96). This role, described as a professional with both analytical skills and software engineering skills, is not yet defined in companies (Kim et al., 2016). Kim et al. (2016) describe this new role as an emerging role of a data scientist working in a software development team. One can come to the conclusion that a new role, in software development teams, is emerging. This role requires some knowledge from both data science field and software development. Lindgren et al. (2008) point out that necessity to combine the expertise from different fields is common in organizations (Lindgren et al., 2008). This expertise can be acquired by a practitioner who works closely with experts from these fields, as a result competence in boundary spanning develops (ibid.). Usually, the interaction between experts leads to creation of new knowledge at cross of all involved experts’ fields (ibid.). From Kim et al. (2016), one can deduce that a new role emerges from combination of two different fields of expertise: data science and software development. And, Lindgren et al. (2008) point out that merging two different competences leads to knowledge creation and boosts innovation. This acknowledgement raised my interest to explore how to induct collaboration between data scientists and software developers in an organization. Levina & Vaast (2005) underline that IT based artifacts are able to manage the process of boundary spanning competence. Data scientists and software developers professionals use a wide range of IT systems (Kim et al., 2016). In light of these factors, I have decided to employ the IT artifact to address lack of collaboration between data scientists and software developers. The IT artifact aims to act as a communication tool between data scientists and software developers.

As described above, there are two major challenges which will be addressed in this study by employing the artifact: missing relevant data for ML purposes and lack of collaboration between software developers and data scientists. Thus, the IT artifact’s two main functions are:

(1) to support the communication between data scientists and software developers
(2) to collect relevant data from software products as an embedded IS component

This research has elaborated on the design of IT artifact and evaluation of the artifact in an organizational setup, by employing design science research as a framework. The IT artifact has originated in this research. The innovative aspect of artifact lies into the functions that tool provides.

The artifact's design is an important part of this study. In order to design and develop the artifact, literature resources and experts were consulted. DSR methodology offers guidelines on how to design and develop the artifact. During design and development process, design principles have been established. The design principles are refined to help others to reconstruct this type of artifact.

The evaluation part has been guided by the DSR paradigm. The DSR cycle has been followed to organize and structure the evaluation part of the study. Strategies to conduct the evaluation efficiently have been forethought. Valuable input for this research was received from Tietoevry's practitioners. To evaluate the proposed IS artifact for this study in the context of an organization, it was appropriate to select a company which has both data scientists and software developers experts. Therefore, Tietoevry company was suitable for this study.

Artifact description

The IT artifact addressed in this study has been designed as a solution for research problems. The IT artifact was developed to take on the following challenges:

(1) Amend the relation between data scientists and software developers

(2) Improve data collection for ML

The artifact serves as a communication tool to build a bridge between data scientists and software developers. In this context a data expert will take on the responsibility to analyze and determine which data are to be extracted from an application and how should the data be structured. While the software developers will be notified what data and which format is to be collected from a software application. Software developers will also have the possibility to come with suggestions regarding the proposed data structures. This activity will enact a regular collaboration between software developers and data scientists. Hence, the possibility to fuel boundary spanning competence as a result of frequent communication.

The artifact's second function is related to being embedded into software applications and collecting data. Software developers will use the artifact as a boundary resource for software applications. This feature enables data collection from software applications. Which data to be stored and how to structure data is priorly established by joint work of data scientists and software developers. Collecting data based on explicit specification and in a specific format, potentially can ensure data quality for ML.

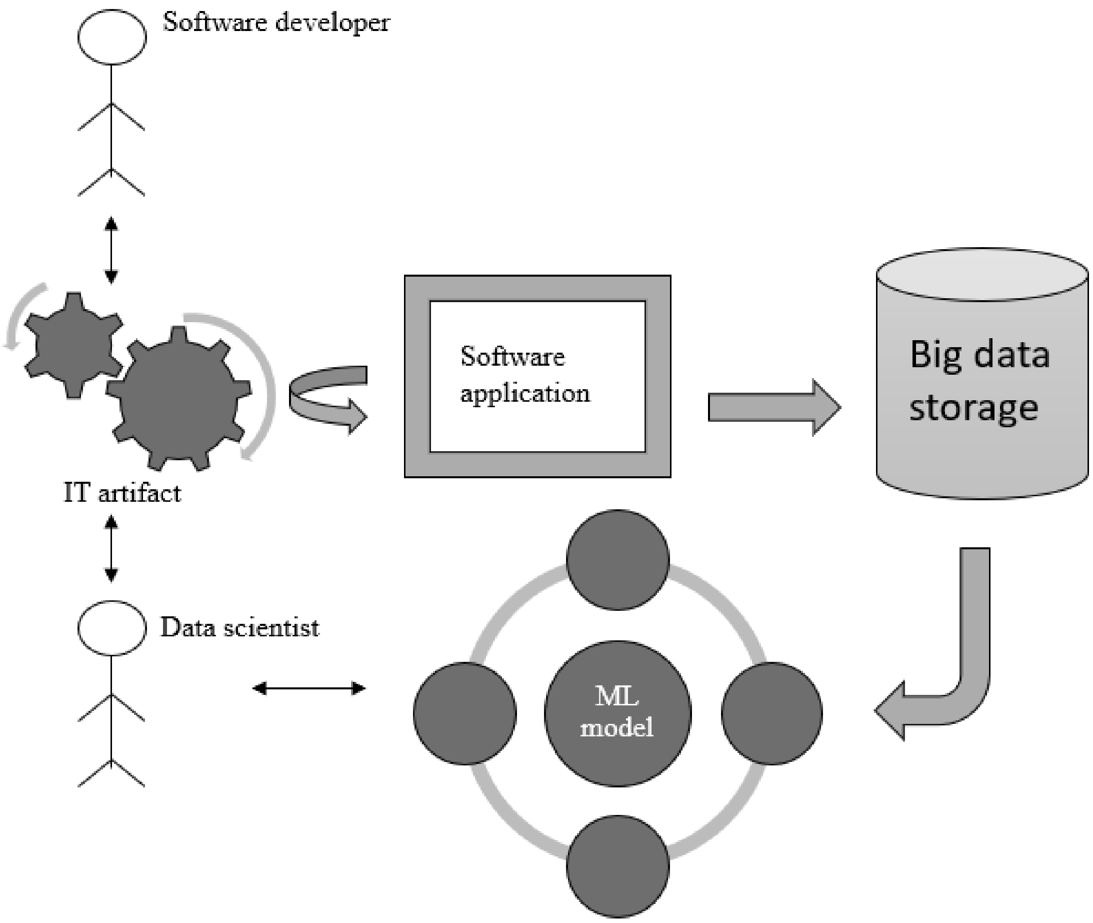


Figure 1.1 IS artifact's mode of operation overview

Figure 1.1 illustrates an overview of artifact's operations. The collaboration between software developers and data experts is managed by IS artifact. The artifact is to be embedded in a software application. Data collected by the artifact from the application is to be stored in a big data storage system. Later this data is to be used to feed ML models and soothe data scientists' work related to data cleaning tasks.

Research question

This master thesis addresses the engagement of software developers and data scientists in boundary spanning by using an IT artifact. The IT artifact can be used as: (1) a communication tool between software developers and data scientists; and (2) a tool which assists data collection for ML purposes.

The accosted research question is following:

"Can an IT artifact facilitate the collaboration between software developers and data scientists to support boundary spanning competence, and improve data collection for ML purposes?"

Research aims

By answering the main research question I have also underlined several aims to be attained during the research. I have divided my case study's aims into two sections, in view of two different perspectives. First form, the artifact's usability perspective, and second from the artifact's design process.

(A) The following are my study's aims from the artifact's usability perspective:

(1) Explore the utility of the artifact from the practitioners' perspective with regard to boundary spanning

(2) Examine the IT artifact's usefulness in collecting data for ML models, by employing practitioner's feedback

(B) The following are my study's aims regarding the artifact's design process:

(1) Accumulate knowledge throughout the process of designing and developing the artifact

(2) Establish design principles for development and usage of the artifact

The generic aim is to formulate and gain a deeper understanding of the addressed problems, based on research conducted in an organizational setting, and have a practice and theoretical inspired analysis.

Research methodology

The research framework employed in this case study is Design Science Research approach. This research methodology was chosen due to DSR's ability to guide design and develop an artifact. Moreover, DSR explores how to solve organizational problems by introducing innovative artifacts (Hevner et al., 2004).

Design science research paradigm helps researchers to derive knowledge and comprehension of a set of organizational problems by designing an IT artifact (Hevner et al., 2004). DSR offers guidelines for artifact's design process and artifact's evaluation (ibid.).

The DSR paradigm is popular in IS discipline, DSR gives researchers the possibility to combine organizational elements, human aspect and technology (ibid.).

Master thesis structure

Chapter 1: Introduction

Introduction chapter offers a generic overview of the study and helps a reader to easily dive into the research.

Chapter 2: Literature

Literature chapter offers an overview of the literature resources used to build this case study

Chapter 3: Methodology

Chapter 3 describes the research methodology used for this case study. The applicability of the research framework is presented. And research elements followed throughout the research.

Chapter 4: Context, artifact and evaluation

Context, artifact and evaluation present firstly, an overview of the organizational setup in which the study took place. Secondly, this section provides a description of the artifact.

Chapter 5: Discussion

In the discussion chapter, a summary of this study is provided. Author's contributions are also introduced, and a review of main points of the study is presented.

Chapter 6: Conclusion

Conclusion chapter offers a reflection about the research process. And highlights main findings and future recommendations.

Literature

This chapter helps to build a foundation for my research, providing the perspectives which have been developed in existing scientific resources related to this case study. Literature review presents scientific articles, books and theories related to the research problem (USC Libraries, 2022). The literature review is used in studies to give an analysis of all the sources accessed and investigated during the research (ibid.).

The Literature chapter is divided into three subchapters: Background literature, Generic theories and Kernel theories. Background literature offers an overview of the research context and explains concepts related to the organizational setup. Generic theories help assess the base knowledge for the study and gain an understanding of ML field and boundary spanning practice. Kernel theories are focused on the scientific theories used to build and design the artifact.

Background literature

This chapter offers an apprehension of the context in which the research has been conducted and an overview of main contextual concepts. First, a description of a data-driven organization is presented, what does it mean to be a DDO. Understanding of DDOs is important for this study as the case has been conducted in the context of Tietoevry. Tietoevry is a company which is considered to be a data driven organization. Second, literature dives into the comprehension of software developers and data scientists terms. Moreover, it portrays an overview over roles that software developers and data scientists have in an organization. And, lastly, scientific resources draw attention to an emerging role explored by researchers: the data scientists' position in a software development team.

Considering that the study was carried out in Tietoevry company, which is a DDO, and conducted with one data science and one software development department. I regard its importance to understand the organizational context from DDO perspective, as it is to comprehend the roles of data scientists and software developers in an organization. The background literature takes on the responsibility to set up a foundation on which the research context will be easier to grasp.

Data-Driven organization

Data Drivenness denotes the ability to develop a system and a culture which is data oriented (Anderson, 2015). One of a DDO company's goals is to make the most out of data potential (ibid.). Leveraging the big-data has become a strategy that companies use to compete and add value to their business (McCarthy et al., 2019). Organizations use historical data to predict the future trends and use these insights to grow and increase their profits (ibid.). To better understand DDOs, Anderson (2015) has identified the types of activities conducted by organizations with a data driven path (Anderson, 2015). To have a neat overview of activities, they are tabulated in Table 2.1. Anderson (2015) indicates that a data-driven organization will pursue at least one of the activities explained in Table 2.1.

DDO activity	Characteristics
Constant development philosophy	A data oriented organization has to be open to reorganize itself and regularly improve. Generally DDOs are looking for ways to automate: inside operations and increase performance.
Use of predictive analytics	Analyzing data and creating machine learning models to predict future trends is an activity which is often adopted by data driven organizations. The machine learning models have to be improved and perpetually fed with data, in order to enhance the results and acquire new observations
Make decisions related to future proceedings based on a collection of weighted variables	There are different paths a company can take and certain decisions to be made, therefore it is important for an organization to consider all the indicators available. Usually a model uses a set of variables. The preponderance of these variables can be adjusted based on their importance to the final resolution.
Constant testing and data observations as part of daily activities	Some tests are required to be continuously performed. Some of these tests ensure gathering feedback from customers. The data from the feedback is used to decide new functionalities to be introduced into system.

Table 2.1 DDO activities according to Anderson (2015)

Nowadays it is attractive to become a data driven company, therefore this is a goal most companies set for themselves (Mohanty & Vyas, 2018). Mohanty & Vyas (2018) emphasize on the competitive advantage a company gains from becoming a data driven enterprise (ibid.). By leveraging the data and creating a data driven strategy a company can enhance its capabilities to deliver excellent services and products while improving and optimizing internal processes (ibid.). Although a data driven enterprise is capable of improving business processes, there are some challenges to fully benefit from being data driven (Mohanty & Vyas, 2018). Some of the challenges Mohanty & Vyas (2018) exemplify are following: (1) handling vast amounts of data, (2) acquiring relevant data, (3) adopting proper tools, (3) developing knowledge (4) attracting experts and (5) balancing between digital transformation and current business operations. To address these challenges companies are trying to find solutions by employing Information System (IS) tools (Mohanty & Vyas, 2018).

Software developers

Ozkaya (2021) describes software developers as professionals working with tools to implement software features by using one or more programming languages (Ozkaya, 2021). Software developers have experience in information technology and can be mentioned also as: “programmers, developers and coders” (Ozkaya, 2021, p.3). Although software developers are expected to code, not all software developers are qualified for software engineering work (Ozkaya, 2021). Software engineers undergo training in the software development process in addition to the acquired knowledge related to software engineering activities (ibid.). Software engineering work requires “to understand the requirements of the software to be developed, formulate the design and architecture of the software by understanding its tradeoffs, and understand the many activities that need to be executed for the successful delivery of the software ” (Ozkaya, 2021, p.3).

Techopedia (2017) defines developer as “an individual that builds and creates software and applications” (Techopedia, 2017). The tasks of a software developer include writing code, debugging and executing a software application’s source code (ibid.). The most important contributors to development of a software application are software developers (ibid.). Software developers have expertise in at least one programming language and have skillset to structure and develop the source code for a software application (ibid.). Software developers’ main activity is to write code, nonetheless a software developer might also participate in other tasks related to the process of software development like collecting requirements, architectural design, document design and specifications (ibid.).

Data scientists

DJ Patil and Jeff Hammerbacher came up with the term data scientist in 2008 and used it to describe their job roles at Facebook and LinkedIn, the work implies collecting and analyzing data (Kim et al., 2016).

Kim et al. (2016) studied the role data scientists have in an organization. In Kim et al. (2016) study 16 data practitioners from Microsoft were interviewed, each participant with a role related to data science work. Although participants' work was linked to data science tasks, their skill set and work tasks were different from one another (Kim et al., 2016). Among tasks, following activities were noted: collect data from software applications, obtain insights from users' interaction with GUI, analyze and understand the results captured (ibid.). In the light of these observations one may conclude that data science work contains a large set of different competences. van der Aalst (2014) lists the following competences related to data science: data mining, machine learning, analytics, statistics, behavioral/ social sciences , industrial engineering, system design, distributed computing, domain knowledge and visualization (van der Aalst, 2014). van der Aalst (2014) defines data scientist role as a combination of various sub disciplines which portrays an engineer who has technical skills, with a statistical/ analytical mindset, communicative and creative skills and is able to deliver complete solutions (van der Aalst, 2014).

Data scientists' role in software development

Data scientist roles are becoming popular among software development teams, many companies choose to hire data science practitioners to work together with software developers (Kim et al., 2016). The following tasks are to be realized when having a data scientist in the software development team: (1) collecting data from software applications related to user behavior and application's execution, (2) analyzing the data and (3) making predictions based on data (ibid.).

The challenges data scientists experience are mostly related to data, analysis and human aspect (Kim et al., 2018). When it comes to data related challenges, the following have to be addressed: (a) poor data quality, (b) data availability, and (c) data preparation. Lack of good data is due to issues in the process of collecting the data and acquiring relevant data (ibid.) Data availability problems come from following cases: missing data, shortage of relevant features and too much disorganized data (ibid). These issues hinder the capability to properly understand and structure data (ibid.). Often data preparation faces an inconsistency of what data is about due to limited documentation and irregular schemas (Kim et al., 2018). Usually data comes from different sources and has different structures, which makes it difficult for data scientists to merge and extract consistent data (ibid.). Data analysis challenges are related to scalability and machine learning (Kim et al., 2018). In the Kim et al. (2018) study, the issues related to scalability were connected to the size of data, process time and generic tools which do not handle particular cases. The machine learning's challenges appear when it comes to feature engineering and performance issues in some given circumstances (ibid.). The challenges involving human aspect have to do with system updates which are hard to keep up with and responsibilities outside of data analysis work (Kim et al., 2018).

It is essential to understand a data scientist's role in a software development team and challenges practitioners face, in order to explore new activities, and further research opportunities emerging from this setup (Kim et al., 2018).

Generic theories

Generic theories have the role to build knowledge related to the area of study explored in this research. This chapter introduces literature related to machine learning and boundary spanning theory.

After being acquainted with literature related to the context in which the study is conducted in the previous chapter: Background literature. The apprehension of why a company is a DDO and what are the roles of software developers and data scientists, helps to shape the setting of the research and carry on with the study. An examination of the IT artifact's main features is necessary to proceed with the study. The researched artifact provides the following services: (1) facilitate communication between data scientists and software developers, and (2) collect relevant data for machine learning models.

To further dive into the study, first, one has to get familiar with the machine learning term and additionally be introduced into the data collection workloads specific to data science work. Second, an apprehension of the boundary spanning concept is needed to unfold why collaboration of data scientists and software developers is important to be explored.

Machine learning

Machine learning is a process in which computers are creating modifications and adjustments to their actions (Marsland, 2015). Subsequently, the ML process develops competence in solving a task autonomously (ibid.). To assess the efficiency of a machine learning outcome, accuracy is used as a measurement (ibid.). Accuracy indicates if selected actions are correct (ibid.).

Nowadays vast amounts of data are stored across the world through existing applications and devices used every day in different domains (Marsland, 2015). Related to that, some questions arise: (a) how to use data properly in order to create value? (b) how to build something useful? and (c) how to improve certain operations? (ibid.). ML has capability to address all these questions, therefore ML is so widespread at present (ibid). ML models require complex computational tasks which can not be handled by the human brain (ibid.). Due to technological advancements which enabled computers to process and map large amounts of data, lately ML field has gained even more attention and become more popular (ibid.).

The ML field has the fundamental mission to learn from data, the essence is to learn from experience by recognising old patterns which had a specific output (Marsland, 2015). After identifying the patterns, this knowledge is used to generalize the ML model in order to be able to predict outputs for new elements (ibid.).

Data in Machine Learning

Data is a prerequisite for the machine learning process (Arankalle, 2020). The machine's autonomous learning depends on structured, organized and relevant data (ibid.). Storing data for machine learning purposes versus general purposes is different (ibid.). The data structures used in classical computer science like trees, arrays, and hash tables are less used for machine learning (Marsland, 2015). Predominantly ML uses data structures such as vectors, matrices and tensors (ibid.).

There are various data storage layers which are used in designing an AI system (Arankalle, 2020). The first step is to feed the data storage layers with data from the source (ibid.). There are a series of prerequisites to be met for each storage layer prior to storing the data, therefore the source data has to undergo an ETL process to match the AI system's storage model (ibid.). The step in which data is cleaned and prepared has an important role for the AI system, as success and accuracy of the machine learning models depend on training data sets (ibid.). This requirement implies some tasks to be performed: analyzing data, creating some routines to adjust data to the target model, enforcing data validations etc. (ibid.). There are also tools helping data engineers to handle the ETL process, but usually to use these tools efficiently engineers have to spend time on training courses and testing (ibid.). Usually, it is difficult to get an in-depth understanding on how to configure and work with the ETL applications (ibid.). Another disadvantage is high cost of such tools, for instance the ETL tools DataStage and Informatica have expensive licenses (ibid.). Furthermore, the ETL tools are not always reliable (Petrova-Antonova & Tancheva, 2020). Petrova-Antonova & Tancheva (2020) case study has shown that when using OpenRefine and Trifacta tools, to clean the data, different outputs were given for the same dataset, although similar techniques for data cleaning were applied.

A known impediment in achieving a satisfying machine learning model is data quality (Gudivada et al., 2017). Many companies have been struggling to implement advanced prescriptive and predictive analytics due to lack of relevant data (ibid.). Frequently raw-data has to go through a cleaning process (ibid.). To use data for ML models is often needed to eliminate duplicate data, address the missing data and determine relevant features (ibid.). The consequences of poor data quality are underestimated, costs related to data transformations and preparations are quite high in organizations (ibid.). Most of the research has been focussed on improving data quality in relational databases and data warehouses, and less attention has been paid to NoSQL big data which have strived with data integrity (ibid.).

Workloads related to data collection in ML

Data collection is considered to be one of the impediments in ML, as most of the time is spent on adapting the data (Roh et al., 2021). In the Kim et al. (2016) study, data scientists have acknowledged the vast amount of time that they spend on preparing data. The task of preparing data has been estimated to take 80% of the data science job (ibid.).

Data management affiliation has researched techniques used to collect data in the context of data analytics and data science (Roh et al., 2021). Data adjustment process consists of: “collecting, cleaning, analyzing, visualizing and feature engineering” (Roh et al., 2021, p.

1328). Roh et al. (2021) examine the synthesis of data management and ML for data collection, and outline flow in the data collection landscape, from ML and data management perspective. Figure 2.1 illustrates data collection activity, followed by techniques applied to collect data for ML (Roh et al., 2021). The blue text color highlights data management contributions to the ML data collection field (ibid.). Figure 2.1 gives an overview of the operations related to the data collection part and answers the question "*why data collection is becoming one of the critical bottlenecks in machine learning*" (Roh et al., 2021, p. 1328).

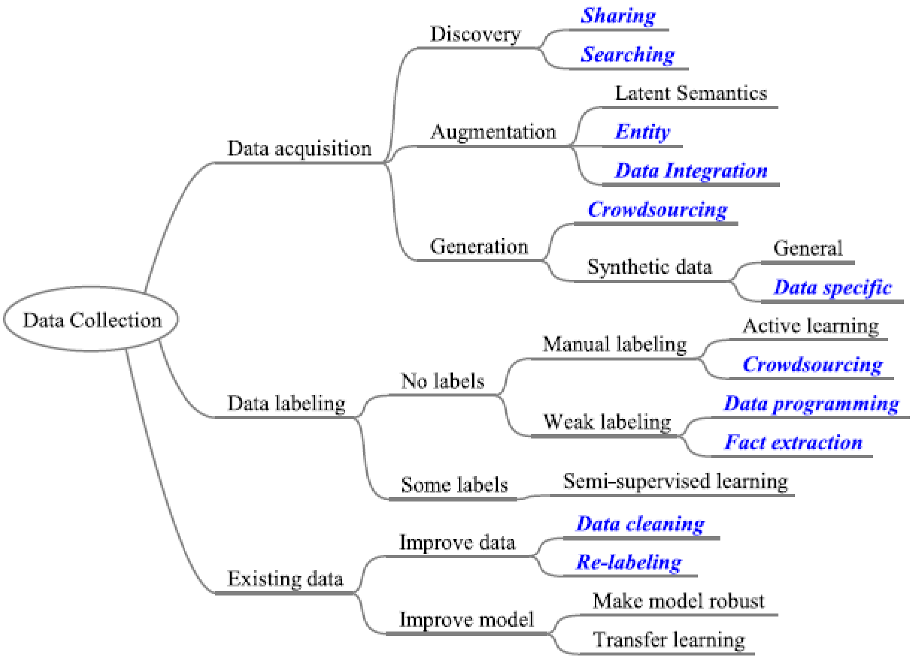


Figure 2.1 "A high level research landscape of data collection for machine learning. The topics that are at least partially contributed by the data management community are highlighted using blue italic text. Hence, to fully understand the research landscape, one needs to look at the literature from the viewpoints of both the machine learning and data management communities." (Roh et al., 2021, p. 1329)

Roh et al. (2021) argue that in order to grasp data collection outlook, one needs to first understand research related to the data management community and machine learning field (Roh et al., 2021). The practitioners which begin to employ data collection techniques for machine learning need to understand when and which techniques to use, from this perspective data management has an important role (ibid.).

As we see in Figure 2.1, Roh et al. (2021) divide data collection techniques into three main approaches: data acquisition, data labeling and improving existing data. Data acquisition has the main mission to find sets of data suitable for machine learning models (Roh et al., 2021). In literature, three methods of data acquisition are reviewed: data discovery, data generation and data augmentation (ibid.). Data discovery uses the approach of searching or sharing data (Roh et al., 2021). Data scientists have different sources to find and share datasets appropriate for a machine learning model, such as corporate data lakes, web services and DataHub (ibid.). Data generation is applicable for scenarios where there are no relevant datasets, one can use crowdsourcing or synthetic data generation (Roh et al., 2021). Crowdsourcing is the method in which human resources are used to gather necessary data manually (ibid.). Synthetic data generation method is based on a process which automatically reproduces artificial data sets which can be general or specific for a given model (ibid.). And, data augmentation is the method to increase the data sets by using external data to expand existing data sets (Roh et al., 2021). Augmentation techniques have been used by the data management community to refine current information (ibid.). The data augmentation methods are latent semantics, entity and data integration (Roh et al., 2021). Following are latent semantics techniques: (1) produce and utilize representation for words, knowledge or items, and (2) group certain parts of data which share similarities (ibid.). Entity augmentation relies on Web search to fill in the absent information in order to complete the data sets (Roh et al., 2021). Data integration method collects and joins information from different data sources (ibid.). Usually mixing data with different provenience is beneficial for training models (ibid.). Data labeling is the next phase after acquiring data (Roh et al., 2021). Data labeling consists in establishing informative characteristics to tag or label raw data (ibid.).

Machine learning relevance to the research

One of the goals of the IS artifact addressed in this research is to improve data collection for ML models. Therefore understanding some conceptual elements related to machine learning and data collection adapted for the ML purposes is essential for this case study. The necessity of the IS artifact comes from challenges faced in the ML field. The idea is to create a tool for improving data collection for ML models through the active and effective communication between data scientists and software developers.

One of the most important aspects of ML is acquiring relevant data. The data are used to produce meaningful and accurate predictions for different prospects, such as business decisions and further developing system’s enhancements. One of the IS artifact aims is to provide a technique to collect data from an application in a defined structure. Basically, define data structures which will be suitable for the ML objectives.

Although some insights related to usability of the application can be standardized and defined beforehand; rest of data will have to be identified by a data analyst with prior knowledge of business functions that the system performs. In this scenario the IS artifact will play the role of a communication tool which can be used to define data and the structure of data which is to be collected from the application. This tactic is intended to enhance the ETL process by making collected data more structured and organized in a NoSQL database and reduce time spent on workloads related to data.

Data is a key driver for ML models accuracy and value creation. It is imperative for a data scientist to have access to large data. The data is used to train and create machine learning models. The IS artifact takes on the task to facilitate the data collection directly from a software application in a priorly defined structure. As big data usually comes from a software application as a result of the system's use, it becomes valuable to explore the benefits of structuring the data before storing the data. Furthermore, the opportunities related to how clean and relevant data can be produced should be investigated by a joint work of data scientists and software developers.

The IS artifact proposed for this research aims to introduce new perspectives on how to improve the data collection for ML. One of artifact’s functions is to be embedded in a software application and structure data prior to storing it. It is assumed that this approach can improve data quality and reduce the time spent on data cleaning.

Boundary spanning

The organization’s ability to foster expertise sharing and knowledge fusion is an important element for its prosperity (Levina & Vaast, 2005). An environment, where practitioners can combine their know-how, contributes to the success of the company (ibid.). Additionally, by managing expertise exchange more effectively than the competition, can help position a company as a market leader (Levina & Vaast, 2005).

Boundary spanning term is used to describe emergence of a new joint field as a result of extending and merging knowledge between different practitioners in a company (Levina & Vaast, 2005). Extensive literature has drawn attention to the importance of spanning boundaries of different specialities and systems in an organization (ibid.). These practices “can become key organizational competence” (Levina & Vaast, 2005, p.336). Development of boundary spanning in an organization takes place, when agents affiliate together to attain a common goal and build a new shared field (Levina & Vaast, 2005). Moreover, by expanding boundary spanning competence in a company, the organizational capital is generated by employing capital obtained from other fields (Levina & Vaast, 2005).

When creating a new product, practitioners with different backgrounds have to cooperate and share their knowledge (Levina & Vaast, 2005). In order to absorb the know-how from other domains, practitioners have to meet on a common ground and work together. This process unites agents and creates new roles which are crossing different fields, in practice it unveils the boundary spanning competence (ibid.). Another aspect of boundary spanning is related to the work being distributed across different departments (Lindgren et al., 2008). And the need of creating processes to interchange data and work between departments arises (ibid.). In this circumstance an exchange takes place between professionals that acquire knowledge at the periphery of distinct fields, creating boundary-spanning practices (ibid.). “Information technology (IT) systems have been hailed as a critical enabler of boundary spanning” (Lindgren et al., 2008, p.641).

Boundary objects

The boundary objects concept has arised from the need to mitigate struggles boundary spanners encounter in certain situations, such as: time, physical distance and lack of social engagement (Levina & Vaast, 2005). In practice boundary objects can be represented by: “physical prototypes, design drawings, use scenarios, engineering sketches, accounting ledgers and standardized reporting forms” (Levina & Vaast, 2005, p. 339).

In the IS field, there are numerous examples, such as archives, ERP applications depicted as boundary objects (Levina & Vaast, 2005).In addition IT artifacts play an important role in boundary spanning (ibid.). The characteristics of boundary objects are: abstraction, standardization, modularity and adaptability (Levina & Vaast, 2005). Boundary objects and their characteristics have been addressed in different studies (Levina & Vaast, 2005) Studies advise to employ boundary objects that are reachable, specific, fresh and tangible in order to achieve satisfying results (ibid.).

In order to determine how boundary objects emerge, it is practical to classify them as “designated boundary objects and boundary objects in use” (Levina & Vaast, 2005, p. 340). Designated boundary objects are those determined as being important for the process of boundary spanning by qualified agents in specific fields (Levina & Vaast, 2005). Although these designated boundary objects contribute to competence sharing between distinct fields, not always these objects turn out to be objects in use (ibid.). The boundary objects-in-use term has been initially defined as “artifacts to be locally useful (i.e., be meaningfully and usefully incorporated into practices of diverse fields) and must have a common identity across fields” (Levina & Vaast, 2005, p. 341).

Boundary spanning relevance for the research

Researches have shown that boundaries between different practitioners can be mitigated by using objects. From this perspective, the IT artifact addressed in this research aims to play the role of a boundary object and help initiate boundary spanning. The artifact takes on the responsibility to create a bridge between data scientists and software developers. The IT artifact will be used as a communication tool between two groups of professionals: data scientists and software developers. My assumption is that by employing the tool practitioners will initiate boundary spanning.

Lindgren et al. (2008) underlined that “IT artifacts have been recognized as having the potential to be adapted to local needs, while at the same time providing a source of common identity across boundaries.” (Lindgren et al., 2008, p. 642). Considering that the artifact is to be used by distinct departments and/ or across practitioners with different backgrounds, the artifact has potential to support boundary spanning.

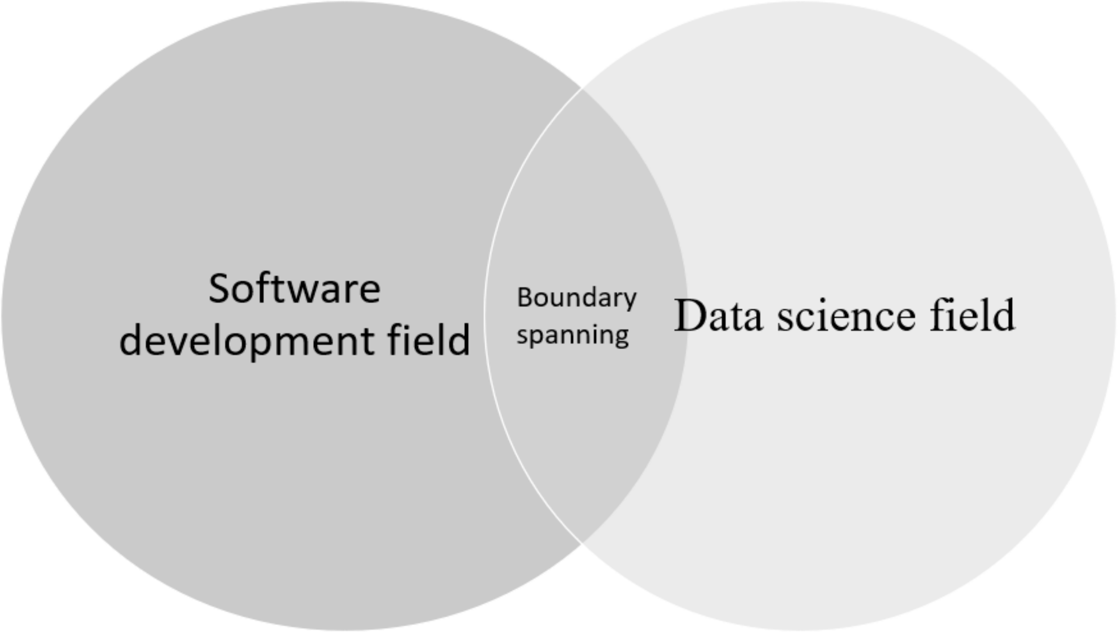


Figure 2.2 Boundary spanning between data science and software development field

As Figure 2.2 illustrates, in this case study, boundary spanning is examined between data scientists and software developers. From this perspective, the artifact will be subjected to evaluation and analysis. This case study will examine the potential IT artifact has to initiate boundary spanning. And, the artifact's role as a boundary object will be evaluated.

Kernel theories

Kernel theory, also referred to as IS design theory, can emerge from academic literature or practice-based theory (Markus et al., 2002). The kernel theory term was initially used in the context of natural sciences, social sciences and mathematics (Gregor & Hevner, 2013). Markus et al., 2002 indicate the following components of IS design theory:

”

(1) A set of user requirements derived from kernel theory

(2) Principles governing the development process

(3) Principles governing the design of a system

“ (Markus et al., 2002, p. 182).

Kernel theory structures empirical hypotheses which can be subjected to testing, or formulates a theory that results from mandatory IS requirements (Markus et al., 2002). IS design theories have the scope to help developers build and test the system subjected to research (ibid.). Gregor & Hevner, 2013 presents kernel theory as a “descriptive theory that informs artifact construction” (Gregor & Hevner, 2013, p.340). Furthermore, kernel theory is used to clarify why a certain design can accomplish a function and operate appropriately (ibid.). Kernel theories can also handle certain pieces of design, by providing knowledge related to some of the design parts (Gregor & Hevner, 2013).

Considering the purpose of kernel theories, this chapter gives a comprehensive assessment of theories used to design and develop the IT artifact. As mentioned earlier in the study, there are two main utility purposes of the IT artifact: (1) facilitate communication between data scientists and software developers and (2) collect relevant data for ML models. From the perspective of the (1) first utility purpose, following technical aspects are to be taken into consideration and described: systems availability and following IS standards. From the perspective of the (2) second utility purpose, following technical aspects have been theorized and explained: API architecture and boundary resource.

During research, kernel theories can also contribute as a knowledge which is expanded through evaluations of the artifact that can lead to progress of behavioral theories based on the artifact’s usage (Gregor & Hevner, 2013). Although some improvements come from practitioners evaluations, the explanation of why enhancements are adopted into the artifact have to be based on the kernel theory and have a knowledge foundation (ibid.). In design science research, kernel theory justifies design principles based on how the artifact is developed; the knowledge supports the artifact's development relative to established design principles (Gregor et al., 2020).

System availability

Availability in the realm of information systems represents the capability of an application to operate and execute the built-in functions (Atchison, 2016). Availability also refers to the system's accessibility for the users (ibid.). Atchison (2016) views availability as a vital aspect for scalable systems.

Piedad & Hawkins (2001) recommend to establish users' requirements for availability. The following availability levels based on users’ interaction with applications are identified: high availability, continuous operations and continuous availability (Piedad & Hawkins, 2001). High availability level is represented by a system which is accessible for users based on a defined schedule (Piedad & Hawkins, 2001). The schedule usually is established based on the system’s utility operation, such as working hours in a company (ibid.). This schedule is usually set to minimize disrupting the users’ interaction with the application (ibid.). During the established schedule the system should be up and running (ibid.). For the high availability level, system’s downtime has to be planned in advance and users’ must be informed beforehand (ibid). Continuous operations represent a system’s capability to run constantly and be continuously available (Piedad & Hawkins, 2001). Regarding the downtime of the system’s, continuous operations level accepts only scheduled downtime, unscheduled interruptions are not accepted (ibid.). To obtain this level for a system, it requires designing a reliable and low maintenance system (ibid.). Continuous availability level implies that the system has to be operational and accessible all the time (Piedad & Hawkins, 2001). For this level of availability no planned downtime is expected, the system has to operate with no scheduled service (ibid.). This level is the hardest to achieve, because it demands many resources in order to support uninterrupted operations (ibid.). Usually, only systems with a critical importance for vital services require continuous availability level (ibid.).

Availability has to always be decided according to users’ needs and systems’ role in IS infrastructure (Piedad & Hawkins, 2001). It is important to establish the availability level for a system, as this affects application’s design (ibid.). Moreover, system’s availability has a direct impact on users' needs and influences the costs related to the development and operation (ibid.).

IS standards

“It has been widely accepted almost from the advent of digital communication technology that its dissemination depends on shared international standards” (Hanseth et al., 1996, p.410) . Systems communicate with each other and use the same infrastructure in an organizational setup (Monteiro et al., 2013). Therefore, the foundation of standards is crucial for the well functioning of all systems within the organization (ibid.).

From the information infrastructure (II) view, adopting standards is essential in order to create common practices and interoperability across all the systems in an organization (Monteiro et al., 2013). Hanseth et al. (1996) discuss the need to standardize some technical elements. This standardization comes as a requirement for the interconnection and interoperability between systems (Hanseth et al., 1996). Although it is important to establish a standardization in an institution, the need for a system to change and adapt over time to new requirements also have to be considered when setting up standards (ibid.).

IS standardization refers to how a technology is designed and implemented with respect to the local infrastructure and the integration with other technologies (Monteiro et al., 2013). There are widely known technology standards used internationally such as: Internet Protocol (IP), Transmission Control Protocol (TCP) etc. (Hanseth et al., 1996). Different technologies interact with each other and are built on top of each other; this has been possible due to standards defined in the IS infrastructure (Monteiro et al., 2013) . The development of standards has led to a faster growth of different services around the world (Hanseth et al., 1996). Therefore the importance of standards is undisputed in II (ibid.).

API Architecture

The Application Programming Interface (API) term is defined as: "sets of standardized requests that allow different computer programs to communicate with each other" (Encyclopædia Britannica Inc, 2020). The connection between systems can be easily established via APIs provided by the programs (ibid.). APIs make the access to the data and systems' functionalities easy to establish and integrate into external programs (ibid.).

API architecture is a vital part in the future evolution of the APIs (Auburn et al., 2022). The decisions related to the design of the API, shape their value and development (ibid.). The Representational State Transfer (REST) is an API architectural style which provides guidelines related to the architectural elements part of a scattered system (Auburn et al., 2022). Auburn et al. (2022) give a practical example of an API, which provides data about "attendees resource" with "resource identifier: *http://mastering-api.com/attendees* ". In Figure 2.3 we have an overview of architectural elements as part of a request and a response: method, headers, body (Auburn et al., 2022). When designing a RESTful API the following are to be considered: request method, the body representation, and metadata of the header representation (Auburn et al., 2022). In the given example in Figure 2.3, for the Request part, the following is presented (Auburn et al., 2022,):

(1) The request method is "*Get*" which is described by a verb and determines the type of the operation.

(2) The body representation is defined in the header's "*Accept*" element as "application/json", indicating the body format to expect.

(3) The header's metadata contains the "*Accept*" element.

The request response usually comprises the status code and the feedback message incorporated in the body (Auburn et al., 2022). In the example illustrated in Figure 2.3, the status code is "*200 ok*" and the response message is to be expected as a JavaScript Object Notation (JSON) (ibid.). In the REST requirements the HyperText Transfer Protocol (HTTP) is not specified as a must, however the defined architectural elements are modeled considering HTTP protocol (ibid.).

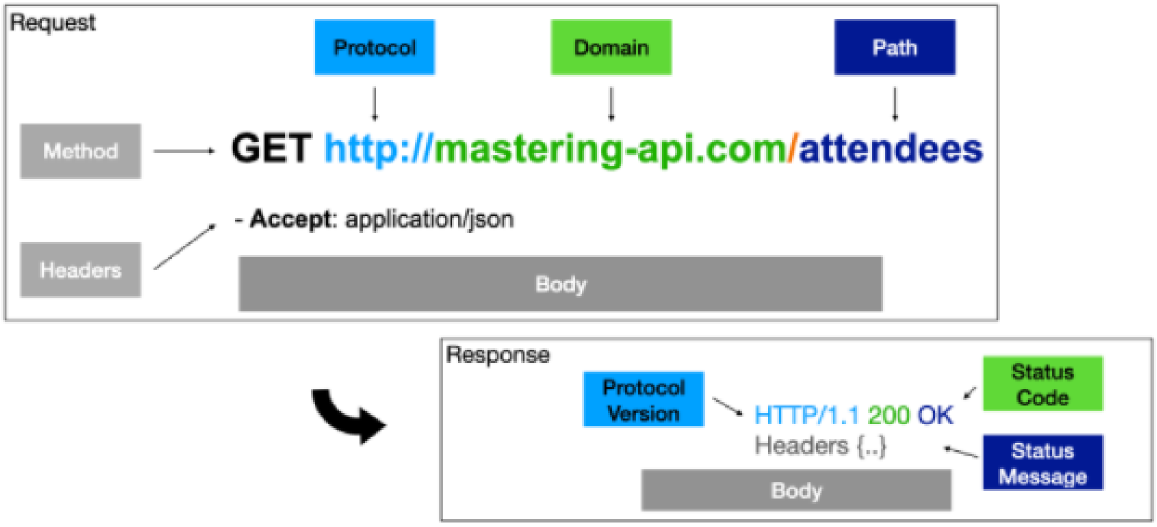


Figure 2.3 "Anatomy of a RESTful Request and Response over HTTP" (Auburn et al., 2022)

The REST style pattern does not impose considerable limitations when it comes to the guidelines offered to build and develop REST APIs (Auburn et al., 2022). As illustrated in Figure 2.4, there are four levels of adoption, which can be applicable in building APIs according to "Richardson Maturity Model Levels" (Auburn et al., 2022).

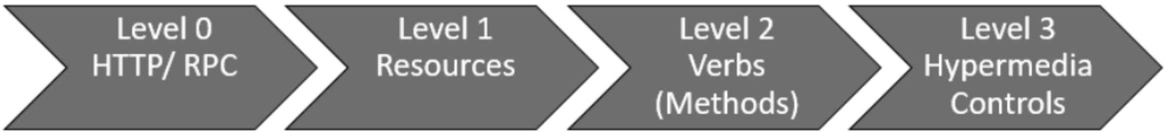


Figure 2.4 Richardson Maturity Model Levels

For “Level 0” the main aspect is the use of HTTP (Auburn et al., 2022). In this context the use of Uniform Resource Identifier (URI) and HTTP method, highlights the fact that API design matches Level 0 (ibid.). The API “attendees resource” example is fitting “Level 0”, as it uses an URI having the protocol: “*HTTP*” and the method: “*GET*” (ibid.).

“Level 1” targets the resources format, the aim is to model the way resources are presented in the URI (Auburn et al., 2022). At this level the objective is to structure the URI so that the identities follow the main resource, having the“attendees resource” example adding “1” at the end of the URI “/attendees/1” to extract the specific identity 1 would make the API match the “Level 1” structure (Auburn et al., 2022).

“Level 2” puts the accent on the verbs (methods) used in the APIs design based on the operation to be executed (Auburn et al., 2022). The API request verbs have to be representative to the operation (ibid.). For instance when requesting the data, “GET” verb is to be used (ibid.). While when the data is subjected to modifications, the verb “PUT” is recommended (ibid.).

At “Level 3” the requirement for REST architecture includes APIs usability and accessibility to resources ensured by the Hypertext As The Engine Of Application State (HATEOAS) (Auburn et al., 2022). This constraint has the overview of how a resource can be modified (ibid.). The response of a request should comprise information about the available operations of a resource (ibid.). In the “attendees resource” example the response of the request would hold “update”, “delete” options allowed to be performed on the attendee, plus information related to what is needed in order to initiate the call (ibid.).

According to Preibisch (2018), designing an API system offers following:

1. More opportunities for the company and its’ market
2. Recognition related to users' significance for the systems life cycle
3. The possibility for developers to explore and contribute to the solutions

An API system opens up for different opportunities for an organization (Preibisch, 2018).. And even though APIs have been in the picture for a while, recently they gained even more attention and have become the preferable architectural approach within IT organizations (ibid.)

Boundary resource

The rapid technological changes occurring nowadays, have made it difficult for the companies to make up-to-date decisions when it comes to which software applications, solutions or tools to implement and/ or adopt (Ghazawneh & Henfridsson, 2012).

The fast development of the technologies and innovative toolkits compel software platform owners to include third-party code which can be easily and quickly integrated into the existing software applications (Ghazawneh & Henfridsson, 2012). Third-party code is created by third-party developers (ibid.). Third-party developers develop small services, functions publicly accessible or contribute to already existing open source applications (ibid.). These resources, platforms' owners receive from third-party developers, are defined as platform boundary resources (ibid.).

As an example of boundary resource Ghazawneh & Henfridsson (2012) give the following: "the software tools and regulations that serve as an interface for the arm's length relationship between the platform owner and application developer" (Ghazawneh & Henfridsson, 2012, p. 174). Bondel et al. (2021) note that APIs are defined in Information Systems literature as boundary resources. APIs are at the cross between third-party software developers and platform owners (Bondel et al., 2021). The API software artifacts have diverse applicability such as supporting features extensions of an application, distributing and transferring data, and acting as an integration between two or more systems (ibid.). This versatility provided by internet standards such as HTTP protocol, makes APIs strategic resources to be introduced in a platform infrastructure (ibid.).

Literature has classified platform boundary resources as application boundary resources, development boundary resources and social boundary resources (Engert et al., 2022). Application boundary resources give access to third-party services to interact with the core platform's features (ibid.). Development boundary resources grant developers the possibility to use tools and libraries to develop their own services and applications (ibid.). Social boundary resources represent the documentation which assist software developers on how to use the platform's services and features (ibid.).

The advantage of resource boundaries is the capability of freely ensuring access to core functions of a platform (Ghazawneh & Henfridsson, 2012). This aspect facilitates the platform's expansion (ibid.). Boundary resources allow different communities to contribute with updates simultaneously, without the need for cooperation (ibid.). The prospect of developing new boundary resources can be proposed by the platform owner and third party developers (ibid.). On one hand the platform owners have the motivation to explore external input to augment platform's possibilities, and on the other hand the third-party developers aspire to benefit from boundary resources and create their products (ibid.).

Ghazawneh & Henfridsson (2012) point out potential boundary resources have to derive innovation. Given the fact that boundary resources can be accessed and utilized by various actors, innovative solutions have the likelihood of emerging as a result (Ghazawneh & Henfridsson, 2012).

Methodology

The methodology I have chosen to follow for my study is design science research. The chapter Design science research describes DSR methodology. The section gives an overview of how DSR methodology is developed in the following subchapters: Design science research cycle and Design science research theory. The Design science research applied, connects the paradigm theory to the current case study.

Data collection chapter describes the data collection methods used for this study. Followed by the Data collection process section, which offers an overview of how data collection has taken place in the study. Moving forward the Evaluation framework in design science gives a theoretical background on how to adequately evaluate an artifact. The Limitations chapter provides an explanation of the constraints observed during this case study. Last but not least, the Ethical considerations part has an important duty to inform the readers about the ethical dilemmas of this case study.

Design science research

This study has been framed as a Design Science Research. Throughout the study the DSR approach has been followed to structure the investigation of an innovative IT artifact in the context of TietoEvry company. The IT artifact has two main goals: (1) to facilitate the communication of software developers and data scientists and (2) to improve data collection for machine learning purposes.

Design science research methodology centers an innovative artifact as the main tool (Hevner et al., 2004). The IT artifact is designed to broaden and improve organizational and human capabilities (ibid). This artifact is built with the aim to solve a problem or class of problems in a specific context (ibid.). A deeper know-how about the problem and the key to solve this problem, comes from the process of designing and creating the artifact (ibid.).
The design science research "is fundamentally a problem solving paradigm" (Hevner et al., 2004, p.76). One of the important characteristics of the artifact is its novelty (ibid.). The IT artifact helps design-science researchers with the following (Hevner et al., 2004):

(1) to comprehend the artifact's ability to solve the problems

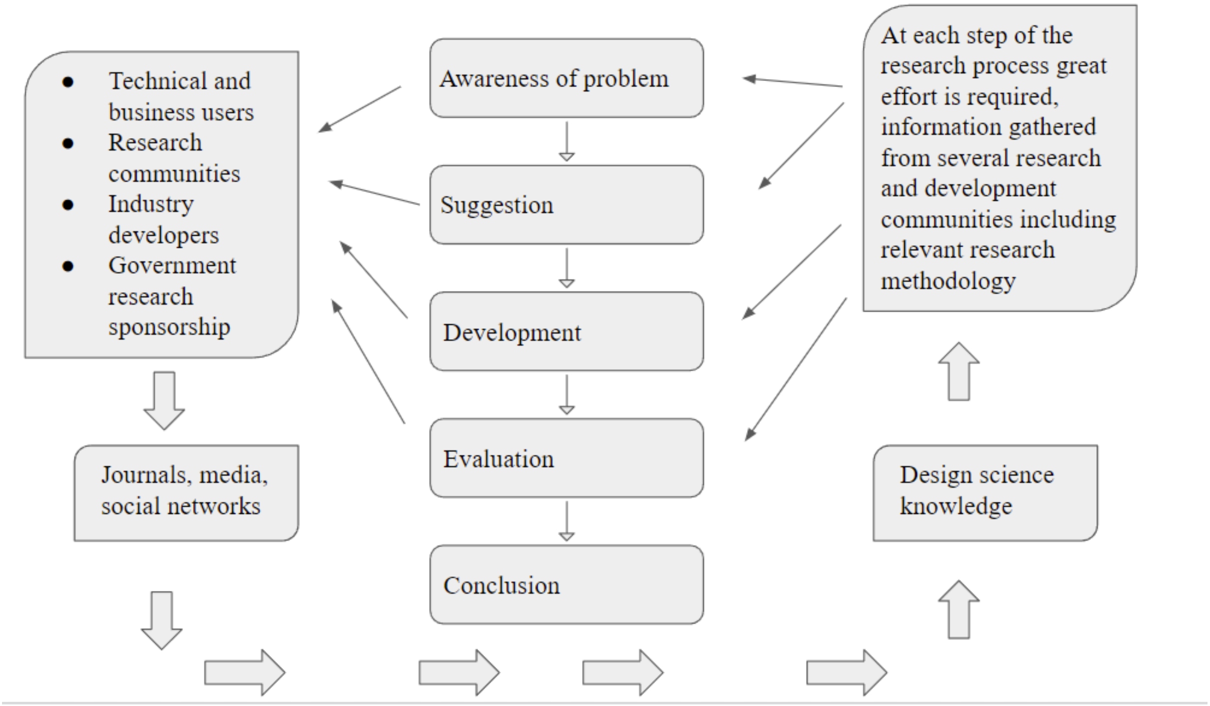
(2) to analyze the artifact's effectiveness in practice

The research in the IS field has considerably applied behavioral science and design science paradigms (Hevner et al., 2004). Behavioral science paradigm approach is to develop and justify hypotheses, which clarify or predict aspects related to phenomena occurring in the context of a business (Hevner et al., 2004). While, the design science paradigm accosts the study by creating and evaluating an artifact which is built to address a business challenge (ibid.).

Design science research cycle

The design science research cycle (DSRC) is an effective process to comprehend practitioners' activity (Kuechler & Vaishnavi, 2015). This process offers an overview of intellectual development at different stages across different domains (ibid.).

As one can see in Figure 3.1, Kuechler & Vaishnavi (2015) have modeled a cycle to be applicable to the design science research. This workflow helps researchers to gain knowledge at each phase of the cycle and improve the artifact's design (Kuechler & Vaishnavi, 2015). Considering this aspect, design science research studies are not only adopting the DSRC, but also can collaborate with groups of intellectual communities and practitioners (ibid.). This collaboration is designed to be a cumulative contribution to the DSR (ibid.).



Technical and business users
Research communities
Industry developers
Government research sponsorship
Journals, media, social networks
Awareness of problem
Suggestion
Development
Evaluation
Conclusion
At each step of the research process great effort is required, information gathered from several research and development communities including relevant research methodology
Design science knowledge

Figure 3.1 "Aggregate design science research cycle" (Kuechler & Vaishnavi, 2015)

The design science research cycle, depicted in Figure 3.1, helps researchers to accumulate insights (Kuechler & Vaishnavi, 2015). The information acquired during the process assists researchers with expanding creativity and improving artifact's design (ibid.). Each phase of the process plays an important role in the study's progress (ibid.). Every step helps to deepen the understanding of the researched phenomena and build the case study (ibid.).

This process starts with *awareness of the problem* phase (ibid.). In *awareness of the problem* part "the problem is not only identified, but also defined" (Kuechler & Vaishnavi, 2015, p. 52). The design science research cycle presumes that researchers take necessary time to formulate and determine the problem before starting to build the tool (Kuechler & Vaishnavi, 2015).

In order to define the problem the researcher should attend to the following (Kuechler & Vaishnavi, 2015):

(1) Study the existing literature in the field and ascertain that the problem has not been already resolved.

(2) Verify that the problem is known and that the proposed solution will be valuable both for practitioners and for research communities.

(3) "Define and scope the problem" according to the existing resources (Kuechler & Vaishnavi, 2015, p.52).

The second phase *suggestion* refers to discovering propositions to resolve the problem (Kuechler & Vaishnavi, 2015). The research framework offers guidelines on how to discover the propositions to solve the problem (ibid.). This discovery can be accomplished by using an abductive approach to deduce from the available sources and current theory, and /or by building up suggestions for the solution in a creative manner.

The effort to create, or/and design the artifact based on the proposed solution comes as a next step in the DSRC, named *development* (ibid.). In the *development* phase most of the work is related to the design and implementation of the artifact. At this step all the collected knowledge is combined and refined to be adapted into the development of the artifact (ibid.). After the *development* phase, the artifact which might be fully or partially developed is subjected to evaluations (Kuechler & Vaishnavi, 2015). This part of the process called *evaluation,* has the main goal to test "how well an artifact works" (Hevner et al., 2004, p.88). Hevner et al. (2004) suggest going through different stages iteratively, in order to improve the design and usefulness of the artifact . Figure 3.2 illustrates the generate/test cycle which pushes the progress of the solution with each iteration (Hevner et al., 2004).

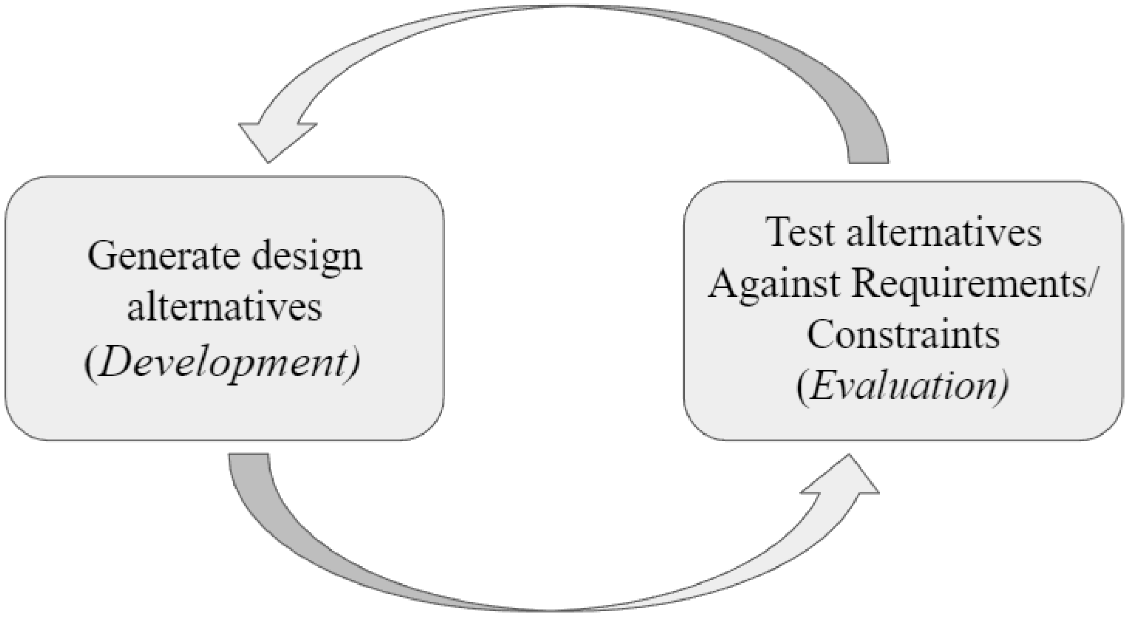

Generate design alternatives (Development)
Test alternatives Against Requirements/ Constraints (Evaluation)

Figure 3.2 "*The Generate/ Test Cycle*" (Hevner et al., 2004, p.89)

Hevner et al. (2004) specified the following activities for "The Generate/ Test cycle":

(1) Develop design options

(2) Test the options considering specifications/ constraints

The last activity "Test the options considering specifications/ constraints", corresponds to: (1) *development* and (2) *evaluation* phases from the DSRC. Hevner et al. (2004) suggest to iteratively work with the two activities in order to achieve a better outcome for the study. I decided to adopt this strategy in the DSRC cycle followed in my research.

The last phase of the DSRC *conclusion* "indicates termination of a specific design science research project" (Kuechler & Vaishnavi, 2015, p. 53).

Design science research theory

In DSR artifacts are "valid and useful for a specific context, the one characterized by the problem space and the group of professionals (practitioners) outlined by the researcher" (De Sordi, 2021, p.9). The notion design refers to outlining a model designated to solve a problem and meet some of practitioners' specific needs (De Sordi, 2021).

De Sordi (2021) analyzed the theory development from artifacts phenomenon, this occurs in the DSR projects (De Sordi, 2021). The theory development from an artifact, requires long-term dedication from the researchers. Moreover, the artifact development demands a massive involvement in constantly improving and evaluating the artifact (ibid.). Hevner et al. (2004) affirm that design theory is assembled to uphold the knowledge derived from the implementation and evaluation of the IS artifact (Hevner et al., 2004). The IS design theory "can be thought of as a package of guidance for designers facing a particular set of circumstances" (Markus et al., 2002, p.181). The IS design theory has two elements to consider (Markus et al., 2002):

(1) That it has a theory foundation

(2) That it gives a clear direction for practitioners

The theory foundation for IS design theory, also noted as "kernel theory", can be an academic rationale or a theory used by proactionner (ibid.). Kernel theory "enables formulation of empirically testable predictions relating the design theory to outcomes like system-requirements fit" (Markus et al., 2002, p. 181).

Design science research applied

In the current paragraph the applicability of DSR to this case study will be explained. The DSR cycle was followed in this study. Below I extend on how the DSR was employed and the DSR cycle pursued throughout this case study. Each phase of the DSR cycle is described in this section. And how each phase took place in the context of this research is unfolded.

The initial phase *awareness of the problem* started in May 2021 with a discussion with one of the project managers (**participant 1**). Project manager works with several data science projects conducted by TietoEvry. During an unstructured interview the project manager has reflected on the challenges related to data science work. From the noted observations, the following provocations related to data science projects were brought out:

(1) Difficulty to find relevant data to feed machine learning models,

(2) Time-consuming tasks related to preparing the data

(3) Vast effort to coordinate between practitioners in order to find AI applicability opportunities, in the context of the existing resources and data in TietoEvry

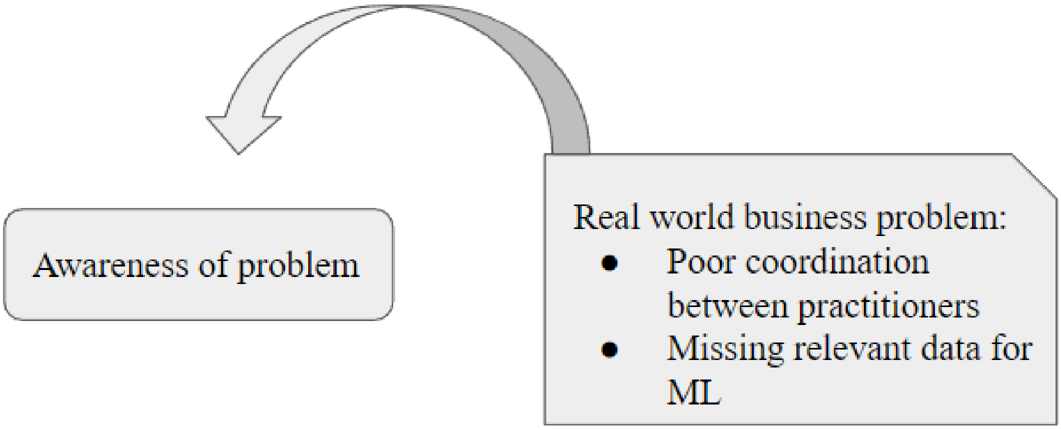

Awareness of problem
Real world business problem:
Poor coordination between practitioners
Missing relevant data for ML

Figure 3.3 Awareness of the problem

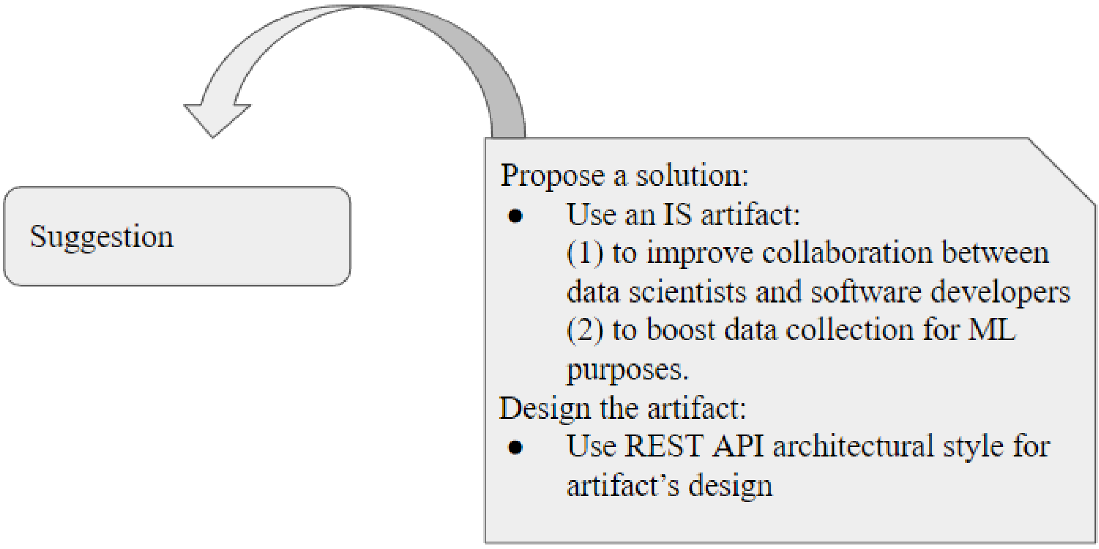
For the first activity of the DSR cycle *awareness of the problem*, researchers are expected to find an existing problem which is met in the real world business practice (Kuechler & Vaishnavi, 2008). As from this interpretation, after the first unstructured interview the research began with formulating and exploring a real world problem mentioned by **participant 1**. The problem that caught my attention for further investigation was related to the poor coordination between practitioners and missing relevant data for ML models. As presented in Figure 2.3 the *awareness of the problem* was shaped starting from a real-world problem. Due to my background as a software developer, I have decided to focus exclusively on software developers and data scientists practitioners collaboration.

From literature, Kim et al. (2016) argue that as more companies seek to use big data, it has become relevant to research software-oriented data scientists (ibid.). I argue that in order for a software-oriented data scientists competency to emerge, a close collaboration between data scientists and software developers has to be fostered. To the best of my knowledge and research, the collaboration between data scientists and software developers has not been considered in literature so far. The most relevant studies have been conducted by Kim et al. (2016) and Kim et al. (2018). Kim et al. (2016) studied the role data scientists have in software development teams. And Kim et al. (2018) address the challenges data scientists face in software development teams.

Regarding the problem related to lack of relevant data for ML models, Roh et al. (2021) point out the existing data collection struggles in machine learning. Roh et al. (2021) affirm that when it comes to the work related to machine learning tasks "data collection is becoming one of the critical bottlenecks" (Roh et al., 2021, p. 1328). Same observation comes from Gudivada et al. (2017), that one of the drawbacks in ML fields is lack of clean and good data.

Kuechler & Vaishnavi (2015) highlight the knowledge contribution which may come from the researched field, as an input to formulate and understand the problems. Moreover, getting acquainted with the material from the studied field, can lead to new findings and help shape the problem formulation (Kuechler & Vaishnavi, 2015). Therefore, it is useful to research the existing literature related to the topic, in the first phase of the DSR cycle (ibid.).

The *suggestion* phase implies work on a proposal for the problem of interest and an initial design of the solution (Kuechler & Vaishnavi, 2015).



Propose a solution:

- Use an IS artifact:
  (1) to improve collaboration between data scientists and software developers
  (2) to boost data collection for ML purposes.

Design the artifact:

- Use REST API architectural style for artifact's design

Figure 3.4 Suggestion

Figure 3.4 depicts the *suggestion* phase. In the figure the following main activities of the *suggestion* phase are outlined:

(1) Propose a solution

(2) Design the artifact

The solution proposed for my research problem is to use an IT artifact.The problems were underlined in the *awareness of the problem* phase. The IT artifact takes on the following tasks: (1) to facilitate the communication between data scientists and software developers and (2) to improve data collection for ML purposes. For the *suggestion* phase I worked on an initial design of the artifact. After an investigation of the software architectural styles, I have decided to use the REST API architectural approach. Chapter API architecture outlines the characteristics of REST API architectural style. Kuechler & Vaishnavi (2008) describe the *suggestion* phase as being also the creative part of the research, where a new artifact is created. Usually, the artifact brings an innovative aspect either regarding configuration or a mix of existing and new components (Kuechler & Vaishnavi, 2008). The artifact proposed in this study has the newness element in its purpose and the manner in which the artifact will be utilized. The artifact has been thought and designed as a communication tool between data scientists and software developers, and as an embedded component to collect data for ML purposes.

The *suggestion* phase is followed by the *development* phase. Kuechler & Vaishnavi (2015) describe the development phase as the activity in which the artifact is created. For the *development* phase, I have started to implement the artifact in July 2021 following an API architectural approach. The development phase took place in two rounds, the first development cycle for the alpha version of the prototype took place in summer 2021 and the second development for the beta version of the prototype took place in autumn 2021.

The *evaluation* phase requires the involvement of practitioners to revise the artifact and provide feedback (Kuechler & Vaishnavi, 2015). As the *development* phase took place in two rounds, likewise the evaluation of the IS artifact.

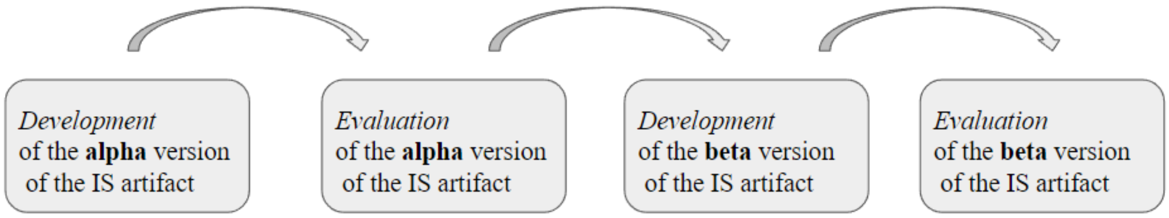

Development of the **alpha** version of the IS artifact
Evaluation of the **alpha** version of the IS artifact
Development of the **beta** version of the IS artifact
Evaluation of the **beta** version of the IS artifact

Figure 3.5 Development/ Evaluation cycle

As illustrated in Figure 3.5 the development started with the implementation of the alpha version of the artifact. The alpha prototype was evaluated by two practitioners in Tietoevry company. The first participant has a data scientist lead role. And the second participant has a data scientist role. The evaluation took place using individual semi structured interviews where the participants were introduced to the artifact and its functionality and purpose. After the first evaluation the received feedback has been used to further develop the artifact into the beta version. The second *evaluation* was conducted with the participation of two software developers; both participants have senior software developers roles in TietoEvry. Hevner et al. 2004 express the importance of the Generate /Test Cycle depicted in Figure 3.2 for development of the artifact. “Progress is made iteratively as the scope of the design problem is expanded” (Hevner et al., 2004, p.89). In my study “Generate / Test cycle” corresponds to “Development/ Evaluation cycle” illustrated in Figure 3.5. The process of repeating *evaluation* and *development* phases successively has been fruitful for the development of the IS prototype.

The *conclusion* phase represents the end of a study cycle, or the end of a particular research resolution (Kuechler & Vaishnavi, 2015). This phase wrapps up the result of the research, the hypotheses are reviewed and the outcome is reported (ibid.). The results can be considered ”good enough” even though the artifact is not perfect and doesn’t fit all the requirements (ibid.). For the *conclusion* activity the researcher has to write the insights gained during the case study, outline the findings and suggest the atypical elements which need further investigations(ibid.). For the *conclusion* phase, two entire sections have been dedicated in this study: discussion and conclusion chapters.

Data collection

This chapter starts with the description of research participants’ role, in the first subchapter research participants. Afterwards, definitions and explanations about the methods employed in this study for data collection are presented in the following subchapters: unstructured interview, semi structured interview and structured interview.

Data collection process section has the goal to offer an overview on how the data collection took place in this study. Moreover, it provides a clarification on why certain data collection methods were preferred at a particular given moment during the research.

Research participants

DSR is one of the few research methods which sustains close collaboration with practitioners and experts outside academia (De Sordi, 2021). An important facet of the DSR study is to identify the context in which the artifact will be used (ibid.). Furthermore, it is essential to understand the environment and indicate who the artifact is addressed to as in end-users (ibid.)

The IS artifact presented in this study has as end-users on one side data scientists and on the other side software developers. The participants involved in this study were part of a data science unit and a software development unit in the TietoEvry organization. In my case I had an understanding of the research context beforehand, as I am working for Tietoevry. Tietoevry has several data science teams and software development sections responsible for various products and services. For this case study two departments have volunteered to participate.

Unstructured interview

Unstructured interviews are helpful to find out what the participants are reflecting on and create space for them to freely express their point of view (Pickard & Childs, 2013). For the unstructured interview the questions have to be formulated in an open-ended manner to give the interviewees the liberty to expand on their ideas (ibid.).

This type of interview is employed frequently at the beginning of the case study in order to discover the challenges coming from the participants' thoughts and reflections (Pickard & Childs, 2013). I have chosen the same strategy to start the research process with an unstructured interview. My goal was to obtain some raw information about the challenges related to the data science world without inflicting my opinion. The unstructured interview for my case study was focused only on one subject: "challenges in data science work". Likewise Pickard & Childs, 2013 remark that the unstructured interview "is useful for eliciting information about a specific topic" (Pickard & Childs, 2013, p 200.).

Semi Structured interview

Semi structured interviews are frequently used as a data collection method due to flexibility and adjustability (Kallio et al., 2016). Semi structured interviews demand that the interviewer is first acquainted with the studied topic and the questions are formulated based on the theory (ibid.). Prior to my semi-structured interviews, I had already formulated the research question. Therefore, I had the opportunity to consult the available literature related to my area of study in advance. The questions were constructed with respect to theory. For my research most of the interviews were semi structured, as it gave me the possibility to not impose strictness and let the conversation flow more naturally. But I had still taken into account the subject of the research and followed a pre-established structure.

Structured interview

For the structured interviews the researcher follows a set of questions arranged and formulated beforehand (Pickard & Childs, 2013). The questions have to be formulated in such a matter that the respondents have a defined set of answers (ibid.). Structured interviews impose a restrictive framework which prohibits the interviewer to deviate from the pre decided questions (ibid). Moreover the interviewer has to restrain oneself from contributing to the conversation, extend the theme or add more information (ibid.).

I have used a structured interview at the end of my case study to verify a couple of conclusions drawn in the previous interviews. The goal was to ascertain my interpretation related to some of the topics priorly discussed. As the literature advises, the questions used for this interview were structured in a straightforward manner.

Data collection process

De Sordi (2021) affirms that closer the researcher is with the practitioners participating in the study “the greater the probability of perception and natural manifestation of a field study” (De Sordi, 2021, p.60). The close proximity of the researcher with the professionals has certain leverages such as (De Sordi, 2021):

(1) The privilege to receive direct approval from the practitioners
(2) The researcher has the knowledge related to the field’s jargons
(3) Better comprehension of the challenges professionals face
(4) The know how of the domain data
(5) The familiarity with the existing tools practitioners use on daily basis
(6) Knowledge of the environment

During the research I have also experienced the advantages De Sordi (2021) describes. Working as a software developer for TietoEvry gave me a head start. Since, I was acquainted with the environment and some of the interviewed practitioners. Additionally, I had a more in depth understanding of the challenges practitioners face. This made it also easier to communicate and to not struggle with the domain's jargon.

For this research study I used the following data collection methods: unstructured, semi structured and structured interviews. Cairns & Cox (2016) state the following "researchers make use of interviews when they wish to obtain more detailed and thorough information on a topic" (Cairns & Cox, 2016, p.21). The same experience I had during the interviews, the different types of interviews gave me the opportunity to develop my research and focus on the researched question.

Pickard & Childs (2013) recommend to consider first what is the goal of the interview, and answer the following (Pickard & Childs, 2013, p. 196):

a. "What can an interview contribute to your research?
b. What can an interview do that no other technique can do, or at least not do as well?
c. Is the interview 'fit for purpose'? "

Data collection was conducted by interviewing practitioners in the TietoEvry company. The first interview was face to face, taking notes during the interview. The next six interviews were held online via Microsoft Teams communication platform. All the online interviews were recorded and transcribed. As illustrated in Table 3.2 the interviews have taken place between 23.05.2021 and 15.03.2022. Each interview gave me the possibility to explore and develop my study. Moreover, the interviews helped me to gain a deeper understanding of the practionners' point of view and evaluate my case.

Participant	**Scheduled date**	**Participant role**	**Method**
Participant 1	23.05.2021	Project manager data science unit	Unstructured interview
Participant 2	01.10.2021	Data scientist lead	Semi structured interview
Participant 3	12.11.2021	Data scientist	Semi structured interview
Participant 4	09.12.2021	Project manager software development unit	Semi structured interview
Participant 5	10.12.2021	Senior software developer	Semi structured interview
Participant 6	17.12.2021	Senior Software developer	Semi structured

			interview
Participant 5	15.03.2022	Senior Software developer	Structured interview

Table 3.2 Data collection schedule and methods

The first interview started in May 2021 using the unstructured interview method. As it was the first interview my aim was to give the practitioner the freedom to reflect on the organizational challenges, without my interference. The next five interviews were semi structured. The intention during the semi structured interviews was to effectively gather as much input as possible in a limited amount of time. Having an interview guide during semi structured interviews, helped me to approach my investigation's objectives. The interview guide also helped me not to deviate too much from the initial arrangement. During this research for semi structured interviews I have used four different sets of interview guides, presented in the appendix A. The questions prepared for each interview had the aim to address the practitioner's expertise and background. For instance the data scientist lead has a better understanding about the work processes in the data science department. A data scientist has the ability to evaluate the artifact from a practical angle. While a software developer has the expertise to evaluate the technical design of the artifact. The last interview was performed following a strict set of questions, this was chosen because of my intention to clarify a set of impressions. These impressions were sketched from the previous interviews.

Evaluation framework in design science

One of the most important part of Design Science Research is the evaluation of the artifact (Venable et al., 2016). Evaluation in DSR covers both the theories applied in the research and the designed artifact (ibid.). This section will be dedicated to the evaluation of the artifact and theories related to this research.

Venable et al. (2016) propose a Framework for Evaluation in Design Science (FEDS) as a practical tool to help researchers with the evaluation process. The FEDS approach lays out the following two range elements: (Venable et al., 2016, p.80)

(1) "Functional purpose of the evaluation (formative or summative)

(2) The paradigm of evaluation (artificial or naturalistic)"

Formative and summative aspects are differently assessed based on the accentuated facet of functional utility (Venable et al., 2016). The formative evaluation is more oriented on the process during assessment (ibid.). Additionally, formative evaluation is focused on enhancing the results (ibid.). On the other hand, the summative evaluation is giving prominence to specifications and standards (ibid.). As one can see in Figure. 3.6: At the left end point is the formative evaluation which requires a clear evaluation of the process activities. And, towards the right end point is the summative evaluation, where the assessment requires a thorough analysis of the technical requirements (ibid.).

Artificial and naturalistic evaluations are distinguished based on the intensity of practicality and philosophical manner (Venable et al., 2016). The artificial aspect is characterized by the evaluations conducted in testing rooms, based more on theoretical analysis and explained by phenomenons proved in existing studies (ibid.). While the naturalistic aspect stresses more on the artifacts' examination in a real context, usually conducted inside an organization (ibid.). "The FEDS evaluation design process is comprised of four steps:

(1) Explicate the goals of the evaluation,
(2) Choose the evaluation strategy or strategies,
(3) Determine the properties to evaluate
(4) Design the individual evaluation episode(s)

" (Venable et al., 2016, p.77).

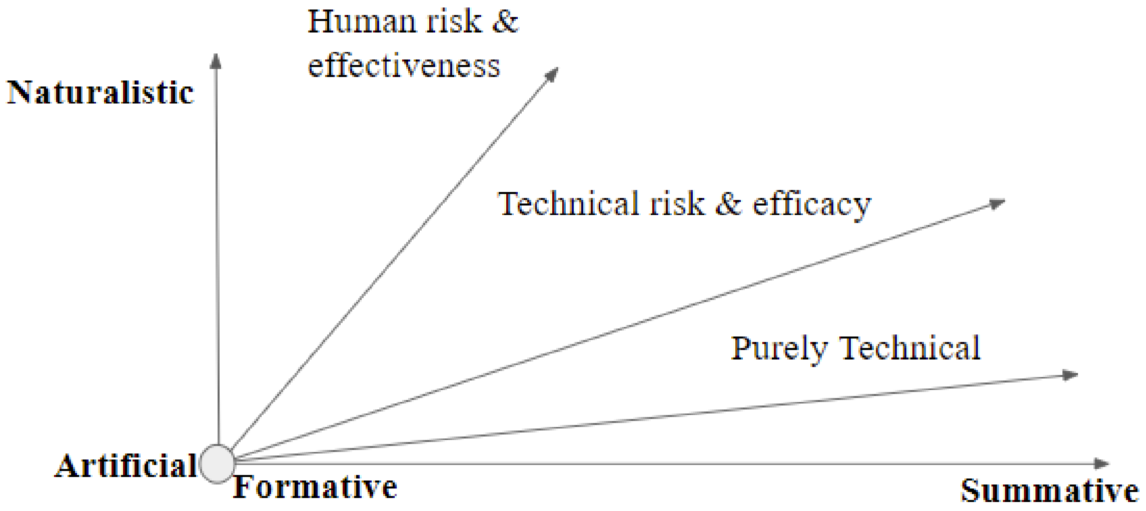


Figure. 3.6 "Framework for Evaluation in Design Science" (Venable et al., 2016)

Explicate the goals

For the first step “explicate the goals” in the FEDS, Venable et al. (2016) identify four goals to be applicable during the DSR process. Table 3.3 gives an overview with explanations of the four goals.

Rigour	The results should be relevant for the artifact and have no interference from other variables. Determine the effectiveness of the artifact and its use in real set up.
Uncertainty and risk reduction	At this stage the formative evaluation is essential. Firstly to identify risks and ambiguities related to the case study. And secondly to establish a strategy and develop a process to address these challenges.
Ethics	When conducting assessments the following have to be taken into consideration: safety of the vital applications and software infrastructure, the responsibility for the welfare of animals, the impact on the future generations, people and institutions.
Efficiency	To efficiently assess all the targets, the researcher has to find a balance between all the objectives and prioritize based on feasible resources and accommodate the requirements in a reasonable manner.

Table 3.3 The four goals applicable in DSR as part of “step 1 explicate the goals” Venable et al. (2016)

Choose a strategy or strategies for evaluation

At this step the researcher has to define the strategies and choose which strategies to adopt for the study (Venable et al., 2016). Having as a foundation the evaluation goals the following has been suggested (Venable et al., 2016):

(1) Analyze the design risks and order them based on priority. Having this overview helps researchers to create a test environment to experiment and discover technical limitations.

(2) Calculate project’s expenditure and evaluate resources availability based on the context

(3) Establish if the artifact is utterly technical or more a design framework solving a problem. This helps researchers to understand how the artifact will be tested and assessed. And, establish if there is a need for human resources to test, or it can be tested by using simulation tests.

(4) Evaluate if the design of the artifact is complex or simple and how big the system is. Based on this analysis it is possible to set up the course of the artifact's development.

Determine the properties to evaluate, design principles

For this step, Venable et al. (2016) suggest determining a set of design characteristics to be examined during the evaluation process. A set of prerequisites related to the artifact's design and properties have to be established taking into consideration the general design theories (Venable et al., 2016). When designing the artifact it is necessary to carefully select a set of principles which uniquely contribute to the artifact's development (ibid.). Gregor et al. (2020) highlight the importance of drawing up a set of design principles to be used in developing the artifact. One can distinguish between three categories of design principles with regard to user activity:

"

(1) Design principles about user activity

(2) Design principles about an artifact and

(3) Design principle about user activity and an artifact

" (Gregor et al., 2020, p. 1628). The **design principles about user activity** cover the interaction of the user with an IT artifact, these principles depict how a software tool assists users' activities to accomplish a defined goal in an organizational setup (Gregor et al., 2020). **Design principles about an artifact** target the artifact's functionalities and technical aspects on how an artifact has been developed such as architecture, design and services (ibid.). The **design principles about user activity and an artifact** are a mix between the first two categories described above. These principles focus on including both the artifact technical requirements and users' activities related to artifact's usability (ibid.). The design principles have the ability to present how users' activities can be reshaped as goals and mechanisms to fulfill these goals (Gregor et al., 2020). For my research design one design principle has been formed as a **design principle about user activity**. And two design principles were formulated as **design principles about an artifact**. A detailed overview on how the principles were established, is given in the artifact design principles section.

Design the individual evaluation episode

After goals, strategies and design principles have been selected the evaluations have to be designed (Gregor et al., 2020). This step aims to settle the episodes required for a specific DSR project's strategy of evaluation (ibid.).
Herein the following should be taken into considerations:

"

(1) Identify and analyze the constraints in the environment. What resources are available - time, people budget, research site, ets.? What resources are in short supply and must be used sparingly?

(2) Prioritize the above contextual factors to determine which aspects are essential, more important, less important, nice to have, and irrelevant.

(3) Decide a plan including determination of how many evaluation episodes there will be as well as when particular evaluation episodes will be conducted and in what way. Hence the outcome is: Who? Is doing what? When?

" (Venable et al., 2016, p.84).

Limitations

Although I found studies related to the role of data scientists in the software development teams and the challenges related to the data science work. I didn't find studies about challenges regarding the collaboration between data scientists and software developers. Moreover, in TietoEvry context it was also hard to understand the challenges related to the cooperation between data scientists and software developers due to the fact that they seldom interact. Therefore, I have focused on the lack of cooperation between the data scientists and software developers. And together with the participants we have created some examples to envision the advantages of the collaboration. Practitioners' previous work experience was also discussed when relevant. However, it was difficult to objectively put it in the current context.

One important element which was not discussed as part of this research, but definitely has to be considered when introducing the artifact in a real world scenario is the IS security. Due to the artifact's openness and necessity to be accessible by several actors, proper security measures have to be considered. But due to the limited time of the project, the security aspect was not prioritized to be discussed in the interview sessions. Another critical element that has not been considered is the data protection. How will the data gathered by the IS artifact be stored in a manner that will respect the data protection regulations was not addressed.

Inability to adopt the artifact in an organization, has also limited my findings and conclusions. The artifact's functions and design were only discussed from a theoretical perspective. Although, a test case scenario was implemented and a test application developed for the discussions. It was still difficult to understand how the artifact will affect data collection. As remarked by participants the artifact's effects on data collection can be only tested, after using the artifact for a significant amount of time embedded in an application.

Ethical considerations

Ethics can be explained as a set of moral principles which influence or guide a behavior (Myers & Venable, 2014). Ethics can also be referred to as the field of knowledge about moral principles (ibid.). Researchers have the moral obligation to follow ethical guidelines according to the code of research conduct (ibid.) Ethical principles can be slightly different depending on the discipline and object of study (ibid.).

DSR studies are addressing the design of IT artifacts, herein the researchers have the responsibility to get acquainted with the ethical principles when creating IT artifacts (ibid.). Myers & Venable (2014) suggest a set of ethical principles to be applied for DSR studies, synthesized in Table 3.4 .

Ethical principle	Description
The public interest	Design science researchers should explicitly identify all stakeholders who may be affected by the artifacts once placed into use and critically consider what benefit or harm may result for/to such stakeholders. Generally, principles of safety, health, democracy, empowerment, and emancipation for all, particularly for the public, should predominate in choices of features and capabilities that an artifact should or should not have.

Informed consent	All design science researchers in IS should obtain informed consent from any person who is in some way involved with the research project.
Privacy	All design science researchers in IS should ensure that there are adequate safeguards in place to protect privacy, not just of those people directly involved with the current project (as with any behavioral research project), but those who might use or be affected by any developed software, IS, or IS development method artifact in the future.
Honesty and Accuracy	Design science researchers should not plagiarize ideas but should acknowledge inspiration from other sources. They should also honestly report their research findings about the new artifact.
Property	All design science researchers in IS should ensure that there is an agreement about ownership of the Information Property (IP) at the beginning of the project. There should also be an agreement about the ownership of any information that is collected during the project and what rights the researcher has to publish findings.
Quality of the artifact	Every attempt should be made to ensure the quality of the artifact(s). Where risks are potentially high, for example in safety-critical situations, design should account for and address such risks and evaluation and testing should be sufficiently rigorous to ensure safety in use.

Table 3.4 "A proposed set of ethical principles for design science research in IS" (Myers & Venable, 2014, p.806)

For this research I have taken into account ethical principles proposed by Myers & Venable (2014). For the first principle "the public interest", stakeholders who will use the artifact are data scientists and software developers. The benefits of adopting the artifact by stakeholders are knowledge sharing and effective communication. The artifact itself can not bring any harm to the stakeholders, unless users engage in toxic communication. This has to be addressed by the company as part of organization culture rules.

"Informed consent principle" has been addressed in this research by following the recommendations from Norwegian Center for Research Data (NSD). A consent form was created for each participant interviewed for the research project. The consent form was signed by all participants, the form is attached in the Appendix C. The recordings, transcriptions and other data related to the participants will be deleted at the end of the project.

When it comes to the “Privacy” principle, the artifact's function to be embedded in an application and collect data has the potential to break data protection regulations. In this regard the general recommendation is to follow the European General Data Protection Regulation (GDPR) rules for data collection. When adopting the artifact, another “Privacy” aspect to be considered is the Information Infrastructure security of the organization, as the artifact will be used in the organization II.

For the “Honesty and Accuracy” principle, during this research my goal was to be truthful and considerate to the findings gathered from literature and practitioners. I tried my best to follow recommendations related to the design science research process and evaluation, in order to build the study and report the results in a good scientific practice. “Property” principle refers to establishing the ownership of intellectual property (Myers & Venable, 2014). This research was conducted by one researcher, an agreement to establish the ownership was not arranged.

The last principle “Quality of the artifact” advises researchers to ensure artifact integrity (Myers & Venable, 2014). For this case study I have established design principles which are to be followed when building the artifact. These principles will assist those who attempt to develop the artifact. I have also noted some aspects to be considered in order to safely adopt the artifact, such as: organization II security and respect GDPR regulations.

Context, artifact and evaluation

This chapter begins with an overview of the context in which the study took place, giving a preface of how the project started and unfolded. Followed by the artifact implementation structure and design principles. Lastly, the Evaluation section provides a presentation of how the assessment occurred for this study.

Tietoevry

Tietoevry corporation’s name appeared as a result of merger between Tieto corporation and Evry ASA company (TietoEVRY Corporation, 2019). The new joined corporation was registered on 5th of december 2019 at the Finish Trade Register (ibid.). The company's main business area is to develop IT solutions for different organizations (ibid.). Tietoevry operates in industries such as: automotive, banking and financial services, construction, education, healthcare, energy, forest, pulp, paper and fiber (TietoEVRY corporation, n.d.). The company seeks to explore opportunities available in the information technology field (ibid.). Tietoevry develops services which propel innovations and ensure clients with new available technologies (TietoEVRY corporation, 2018).

TietoEvry company’s headquarter is based in Finland and currently has approximately 24 000 employees worldwide (TietoEVRY corporation, n.d.). The corporation is delivering digital services and solutions for thousands of organizations in more than 90 countries (ibid.). TietoEvry is listed on the stock market NASDAQ in Stockholm and Helsinki, likewise on the Oslo Børs having a turnover of around three billion euros annually (ibid.).

Tietoevry history

Tietoevry began its activity as Tietotehdas and was founded by Union Bank of Finland in 1968 (TietoEVRY corporation, 2018). The same year in 1968 Enator, Tieto’s Swedish branch was started (ibid.). The company’s first activities were to implement and support information systems used internally, by some forest industry companies, by Union Bank of Finland and by their clients (ibid.). The union of Tietotehdas and Enator has occurred in 1999 under a new name TietoEnator, following some other small acquisition (TietoEVRY corporation, 2018). In 2009 the company decided to change the name to Tieto (ibid.).

Company's contribution in the ‘90s has shaped the banking industry (TietoEVRY corporation, 2018). Swish and Sitro were the first internet banks which have been supported by Tieto’s IT systems, at the present both banks are innovating the payment sector (ibid.). Currently TietoEvry continues to work on the mission of delivering IT systems which drives innovation in both the public and private sector (TietoEVRY corporation, 2018).

Data, AI and Analytics in Tietoevry

Tietoevry has embraced AI, data and analytics as part of the services organization delivers (TietoEVRY, 2022). By employing AI, Tietoevry aims to explore the digital potential for customers (ibid.). The company's vision regarding AI is to make intelligent decisions based on the insights provided by real-time data (ibid.).

Tietoevry aspires to help customers to become data-driven by administering data and analytics and managing data governance (TietoEVRY, 2022). In a constantly changing world, companies can leverage the use of data and AI to be more competitive on the market (ibid.). Tietoevry's goal is to innovate the businesses and develop new solutions which would meet clients' needs (ibid.). The company plans to include automation to most of the delivered solutions (ibid). Among the products and solutions with AI capabilities Tietoevry lists the following services: data management and analytics, artificial intelligence accelerators, intelligent automation and robotics, content services integration and API management (ibid).

Data management and analytics service is designed to help organizations' to exploit and administer their data. The service takes on the responsibility to transform data into valuable decision-making insights and modernize the data ecosystem according to the latest technologies (TietoEVRY, 2022). Artificial intelligence accelerators deliver solutions which help AI scaling within an organization, having the goal to release the capabilities related to the digital aspect across a company (ibid.). Intelligent automation and robotics helps businesses to find operations which can be automated, and deliver solutions which will handle the process automatically (TietoEVRY, 2022). The intelligent solutions will support automated operations, hence reducing costs and increasing KPIs (ibid.). Content services offers a management tool to organize the content moving within the organization and provide a way to handle these massive amounts of content (TietoEVRY, 2022). The target is to create an intelligent method to automatically structure all the existing and future content (ibid.). Integration and API management service help businesses to improve the data exchange between different systems proposing an architecture adapted for the organization's integration needs (TietoEVRY, 2022).

Tietoevry's participating departments

The study was conducted in the Tietoevry organization. Two departments have volunteered to participate in this study. First department's work is mainly focused on data science. And the second department has as its main activity software development. Considering the nature of the IS artifact and the problems addressed, it was important to involve in the study, an organization with access to data scientist practitioners and software developers. Another important factor referring to why Tietoevry was a suitable organization for this study case is due to the background the practitioners have as employees of a multinational company. Herein, the practitioners are accustomed to using different types of systems part of the company's ecosystems. Moreover, practitioners often collaborate across departments and are well known with the challenges of crossing knowledge between experts.

TietoEvry has taken a path towards a data-driven future, this approach makes the practitioners well known with the challenges and strategies related to a data oriented organization. Both the incumbent companies and startups face new challenges in the data-oriented world (Kim et al., 2016). According to Kim et al. (2016), following are two of the most common challenges:

(1) How to efficiently integrate data scientists in the organization structure?

(2) How to utmost benefit from the available data?

Organizations which take on the path of a data-driven approach have to assess how to structure their teams and what are the roles of the people working directly and indirectly with data (Kim et al., 2016). Professionals have different responsibilities in a data oriented company, for instance managers are focussed on adopting data findings in the decision making, software developers are interested in how to improve application's performance, while testers seek to know which vulnerabilities to target (ibid.). Therefore companies have to structure their departments with respect to these nuances and find a way to consolidate the processes related to data science work (ibid.).

In Tietoevry there are several data science departments handling different tasks. Data scientist lead (**participant 2)** has described how the data science section, he is part of, is organized. The data science section which participated in this research has the following structure: a project management team, head of product, data scientist lead, development director, chief architect, data engineers and data scientists (**participant 2**).

According to **participant 2:**

- The project management team includes product managers together with the head of product which "are responsible for development of the product and collecting feedback and customer requirements".
- Data scientist lead is responsible for "all indications related to data and artificial intelligence".
- Development director works with different product units to designate "what projects are prioritized", "other strategic priorities", which project should be started.
- Chief architect "is responsible for a platform which will be technically scalable" and make decisions for which architectural patterns to be followed and technical stack.
- Data engineers establish and develop pipelines to prepare data for data science work
- Data scientists work with processed data and develop machine learning models

The data science department doesn't collaborate with software development teams, as **participant 2** mentions "we have not had the opportunity to work directly with software developers".

Project manager from software development unit (**participant 4)** has provided a general view of the software development team's work and structure. The software development department which participated in this research develops a suite of software products which cover a range of functionalities that efficiently manage the entire process of lending (**participant 4**). The lending operations cover the following: administration of mortgages, secured and unsecured loans, corporate credits and private credits (Tetoevry, n.d.). Regarding the lending software suite: "a powerful business and credit rules engine is built into the solution ensuring secure and precise decisions and pricing in origination" (Tetoevry, n.d.).

The lending software development department has the following structure: project management team, resource management, product owner, software architects, support and test team, software developers (**participant 4**). Although the department does not work directly with a data science department, according to the project manager (**participant 4**), there are two software developers that take on the tasks of data engineers.

Explore Tietoevry as DDO, IT artifact debut

A data driven organization is represented by its ability to make business decisions and take actions based on the data (Hagen & Hess, 2020). To keep the position on the market and aspire for growth in a digitalized world can be challenging (ibid.). Therefore most of the organizations seek to innovate and leverage data in order to succeed (ibid.).

Tietoevry is also positioning itself as a data driven organization and aims to be successful in deriving business value from data. Tietoevry has established data science departments to handle different data driven organiztaion's tasks. DDO tasks have been described in the literature by Anderson (2015), these activities are illustrated in Table 1.1. According to Anderson (2015) a company is considered to be a DDO, if the company adopts at least one of the activities presented in Table 1.1. From the subchapter Data, AI and Analytics in Tietoevry, one learns the DDO activities Tietoevry has adopted. After a synthesis, the activities presented by Anderson (2015) are pursued by Tietoevry as follows:

- Constant development philosophy can be derived from Tietoevry's goals: (1) to research the digital potential for Tietoevry's customers, (2) to innovate the businesses and develop new solutions and (3) to discover operations which can be automated.
- Use of predictive analytics is part of data management and analytics service. This service is developed by Titoevry to help other organizations to exploit and leverage data.
- Make decisions related to future proceedings based on a collection of weighted variables. Derives from the services developed by Tietoevry to transform data into valuable decision-making insights.
- Constant testing and data observations as part of daily activities. This activity was described by **participant 4**, the software development department engages every day in continuous testing activity and data analysis.

Hence, one can strongly state that Tietoevry is a data driven organization. One of the impressions I got from approaching the managers with my project was the managers' open mindset. I noticed the receptivity they had towards new possibilities and innovative solutions. Hagen & Hess (2020) have also noted that in order to create business value and take advantage of the existing data, business actors of an organization should be interested in "performance improvements, process optimization, product and service innovation and customer experience enrichment" (Hagen & Hess, 2020, p.3). From my experience when interacting with Tietoevry's practitioners, the observation was that they were eager to explore new opportunities which can generate value and be more effective in a data oriented world. Practitioners were keen to share their knowledge, and give thorough feedback related to the IT artifact. Moreover, practitioners were open to review and contribute with expert knowledge to advance the research.

Participant 1 has contributed with the challenges related to the data science field, which gave me the confidence to start the research. Afterwards, a comprehensive literature study was consulted to formulate research problems and work on the solution. The solution was designed as an IT artifact. A non-formal discussion with **participant 1** followed to view the possibility to evaluate the artifact in Tietoevry organization. Although initially was discussed the prospect of using the artifact as part of a small group of people, later on was agreed that this would require too much time and resources, which at the time were not available. The agreed approach was to schedule interviews with the participants and theoretically evaluate the IT artifact.

Artifact

This section starts with the Artifact architecture part. Software architecture reveals the structure of a system without going into implementation details. Decisions taken for the software architecture approach are important, as they guide the development of the system. The architectural style selected for IT artifact is discussed in the literature, API architecture section. The Artifact architecture explains how architectural style was adopted for IT artifact. Followed by Artifact interface part, which provides an overview of the GUI approach applied for IT artifact. Artifact design principles section aims to expound on how design principles were adapted to the IT artifact.

Artifact architecture

The IT artifact used in this research has been designed as a set of APIs. The REST architectural style was adapted for the APIs design. As described in literature API approach is favored in organizations. Moreover, APIs are used commonly as boundary resources and can be easily integrated into other systems. This section describes how the designed artifact followed the REST architectural style described in literature, chapter API architecture.

First, APIs' architectural elements used for the IT artifact are inspected. As an example to examine API elements, "data entities" resource is used. Figure 4.1 depicts the request structure:

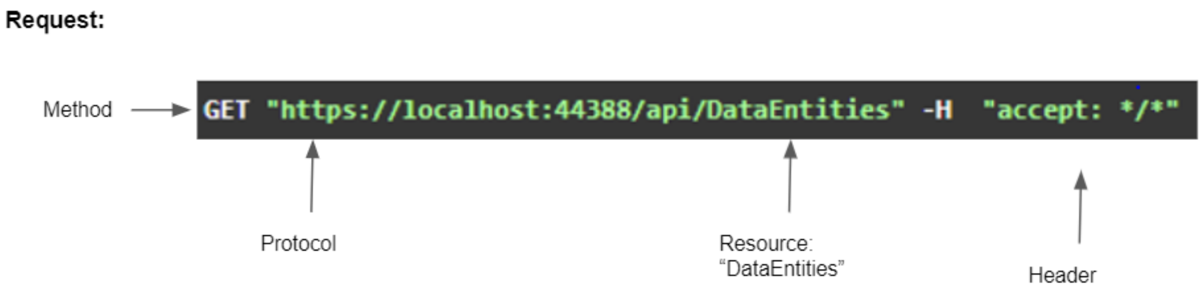
1. Request protocol: **"HTTPS"**

The transfer of the API resources is done over HTTPS protocol

2. Request method represented by the verb "**GET**"

The request verb used to retrieve resources is "GET"

3. Request headers: "accept: */*"



```
GET "https://localhost:44388/api/DataEntities" -H  "accept: */*"
```

Figure 4.1 Request URL "data entities" resource

The response answers the inquiry initiated by the request, based on the URL structure and the parameters used initially in the request. Figure 4.2 illustrates the response of the request initiated for "data entities" resource, the following elements are used in the response:

1. Response status

In the example provided the status code: 200, which is a successful response code.

2. Response body

The response body is presented in a json format and lists the data corresponding to "data entities resource". In the example "data entities" resource contains one object. The object is composed of five fields: "dataEntityName", "jsonEntityObject", "createDateInt", "createDate", "modifiedDateInt" and "modifiedDate".

3. Response header

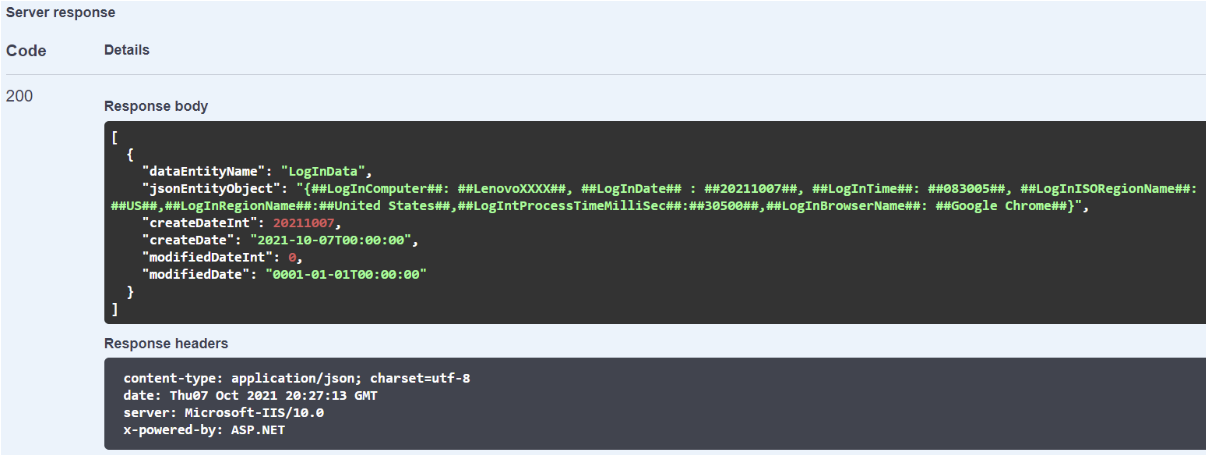
The response header contains the following information:

a. content-type: specifies the representation in which the requested object has been formatted, in the “data entities resource” example the expected representation is json: “application/json; charset”.

b. date: the response date stamp

c. server: specifies the software which managed the request on the server, from the example the software employed on the server is “Microsoft IIS/10.0”

d. x-powered-by: indicates the technology supported by the application, in the example is indicated “ASP.NET” technology.



Server response

Code Details

200

Response body

```
[
  {
    "dataEntityName": "LogInData",
    "jsonEntityObject": "{##LogInComputer##: ##LenovoXXXX##, ##LogInDate## : ##20211007##, ##LogInTime##: ##083005##, ##LogInISORegionName##:
##US##,##LogInRegionName##:##United States##,##LogIntProcessTimeMilliSec##:##30500##,##LogInBrowserName##: ##Google Chrome##}",
    "createDateInt": 20211007,
    "createDate": "2021-10-07T00:00:00",
    "modifiedDateInt": 0,
    "modifiedDate": "0001-01-01T00:00:00"
  }
]
```

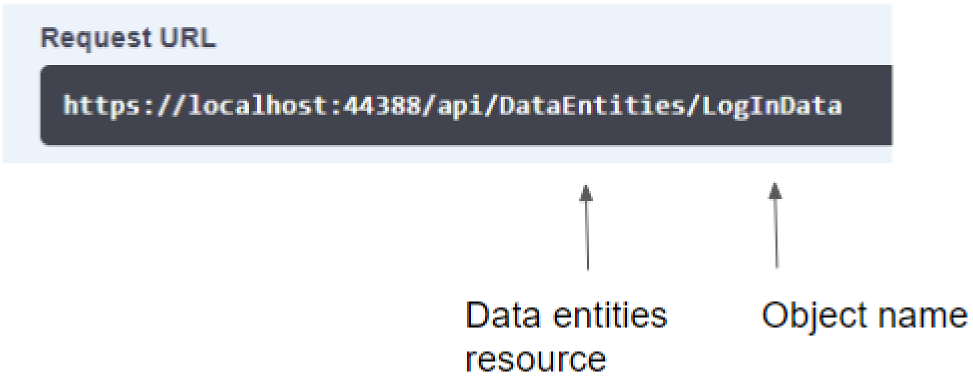
Response headers

```
content-type: application/json; charset=utf-8
date: Thu07 Oct 2021 20:27:13 GMT
server: Microsoft-IIS/10.0
x-powered-by: ASP.NET
```

Figure 4.2 Server response data entities resource

The APIs resources in the current IT artifact match the first 3 levels of REST API according to “Richardson Maturity Model Levels”.

- Level 0
  The IS artifact’s APIs match Level 0, as one can see from the “data entities” resource example the use of HTTP protocol and the method “GET”
- Level 1
  The requirement to have the resource name as part of the URI and extend it with an identity to request a specific object is covered, as we see in Figure 4.3 the “data resources” URL is extended with the object name “LogInData” used as an identifier to request a specific object from “data entities resource”.



Request URL

```
https://localhost:44388/api/DataEntities/LogInData
```

Figure 4.3 Server request data entities resource specify identity

- Level 2

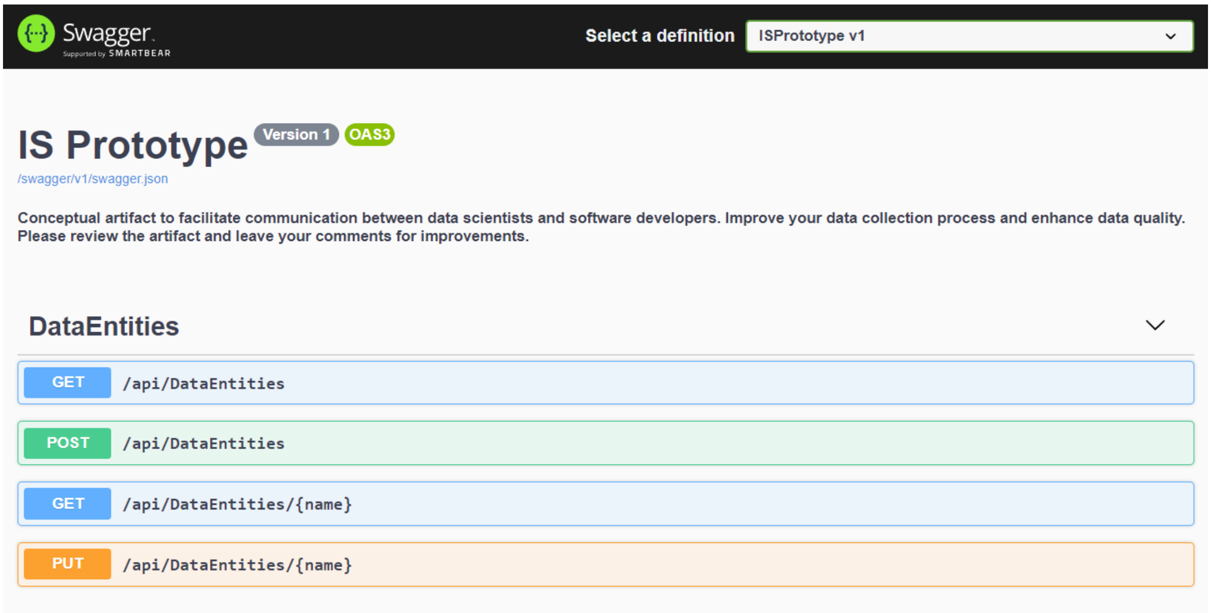
  For Level 2 the prerequisite is to use verbs(methods) in accordance with the type of the operation, additionally the verbs must be part of the representation. As one can see in Figure 4.1 the verb “Get” is part of the representation and characterizes the operation which is supposed to be requested.

Artifact interface

The visualization and interaction with artifact’s functionalities is carried through Swagger UI. Swagger UI is part of the Swagger suite used by software developers to create APIs (2021 SmartBear Software, n.d.). Swagger assists the API creation process and provides powerful tools supporting the API lifecycle (ibid.). Swagger supports developers with the following activities: design process, deployment, test and documentation (2021 SmartBear Software, n.d.). Swagger is composed of “a mix of open source, free and commercially available tools that allow anyone, from technical engineers to street smart product managers to build amazing APIs that everyone loves” (2021 SmartBear Software, n.d.).

This tool helps developers and users to have an overview over API’s resources (2021 SmartBear Software, n.d.). Swagger UI helps practitioners to collaborate and share resources without having distinct logic implementation for each new end point (ibid.). Swagger offers automatic generation of a standard GUI to visualize the API resources. There is documentation available on how to introduce swagger in a project (ibid.). This makes it easy to incorporate the tool into the source code and covers both the implementation of the back end solution and client-side utilization (ibid).

Swagger provides a user-friendly interface for the IS prototype and also allows developers to add customizations to better suit the needs of the users (2021 SmartBear Software, n.d.). Figure 4.4 offers an overview of the IS prototype used in demonstration. A custom description was added to understand the artifact's purpose: "Conceptual artifact to facilitate communication between data scientists and software developers. Improve your data collection process and enhance data quality. Please review the artifact and leave your comments for improvements".



Select a definition: ISPrototype v1

IS Prototype Version 1 OAS3

/swagger/v1/swagger.json

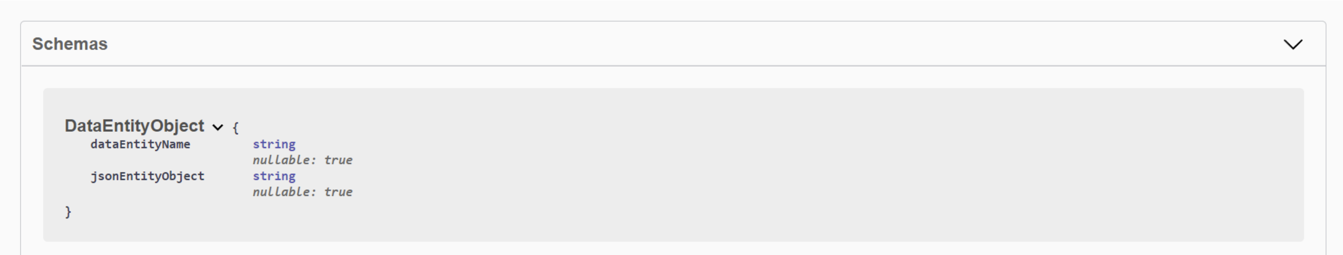
Conceptual artifact to facilitate communication between data scientists and software developers. Improve your data collection process and enhance data quality. Please review the artifact and leave your comments for improvements.

DataEntities

```
GET   /api/DataEntities
POST  /api/DataEntities
GET   /api/DataEntities/{name}
PUT   /api/DataEntities/{name}
```

Figure. 4.4 Is prototype Swagger overview

Swagger GUI displays additional information regarding data models used in the IS prototype, as we see in Figure 4.5 example "DataEntityObject " is represented by a JSON schema where fields' data types are specified.



Schemas

```
DataEntityObject ⌄ {
    dataEntityName        string
                          nullable: true
    jsonEntityObject      string
                          nullable: true
}
```

Figure 4.5 Swagger UI Data models (Schemas)

Preibisch (2018) suggests using Swagger as an API tool to help document the functionality and usability in a coherent way (Preibisch, 2018). Using the swagger approach dismisses the need to spend time on training the practitioners, as most of the IT professionals are well known with the interface (ibid.).

Artifact design principles

It is important to establish solid design principles prior to implementing the artifact(Gregor et al., 2020).. "Clearly formulated design principles will support the process of developing and implementing the IS artifacts and, thus, improve practice in digital innovation " (Gregor et al., 2020, p. 1639). A design principle refers to an abstracted detailed aspect used to build and design artifacts (ibid.). The observers participating in the artifact's analysis must understand the level of abstraction used to design the artifact (ibid.). IT artifacts' complexity requires a fragmentation in order to institute design principles at a lower level of abstractization (ibid.).

The following principles have been considered when creating the IS artifact for this study: principle of availability, principle of following standards and principle of using the artifact as a boundary resource. A classification of the design principles is presented in the section: Determine the properties to evaluate, design principles. As we can see in Table 4.1, the design principles proposed for the researched artifact fall into the following categories: **design principles about an artifact** and **design principles about user activity** . The first two principles suggested: principle of availability and principle of following standards are technical specifications recommended to be followed when building the artifact. These two principles are categorized as **design principles about an artifact.**

The principle of using the artifact as a boundary resource is indicating the way users will utilize the artifact and the operations to be carried out to adopt the artifact. Therefore, the principle of using the artifact as a boundary resource falls into the category of design principles about an artifact.

Design principle	Design principles about an artifact	Design principles about user activity	Design principle characteristics
Principle of system availability	**X**		Technical requirements: (1) Artifact has to respect continuous operation level of availability (2) Artifact has to support proper execution of its function
Principle of following IS standards evaluation	**X**		Technical requirements: (1) Follow IS international standards (2) Adopt the IS standards which are part of company's infrastructure
Principle of using the artifact as a boundary resource evaluation		**X**	Users' activity: (1) Grant collaboration between data scientists and software developers (2) Artifact to be used as boundary resource, to be embedded in other software applications

Table 4.1 Artifact's design principles

Table 4.1 offers an overview of the artifact's design principles. Table 4.1 illustrates the classification artifact's established design principles and a generic characteristic of each principle.

Principle of system availability

In kernel theories, the system availability section illustrates the availability concept from the perspective of application users' having access to the application and system's capability to execute its function. IS artifact is to be utilized by software developers and data scientists, and also used in different software applications as an embedded component. Therefore, from the availability point of view the artifact has to be:

(1) available to software developers and data scientists

(2) as a service, available to be integrated into software applications

Considering the two availability scenarios the artifact must respect, the recommendation is to design the artifact as a set of web service endpoints and have a GUI to interact with these web service endpoints. The IS artifact designed for this research will grow and develop over the time, therefore system's availability have to be forethought. Furthermore, considering that the availability has an impact on the system's development, it is vital to take into account the system's availability prior to starting implementation.

For building the IT artifact as web service endpoints, Artifact architecture chapter describes API Rest architectural style adopted by the artifact. The artifact's GUI has to ensure the interaction between data scientists and software developers. Swagger UI tool is used to support the collaboration. Chapter Artifact interface explains Swagger UI tool applicability.

In the System availability chapter, three availability levels are defined, based on availability levels characteristics presented by Pickard & Hawkins (2001). A software tool which is supposed to be available 24 hours a day, 7 days a week with only scheduled disruption, falls under the category of availability called "*continuous operations*" (Piedad & Hawkins, 2001).

The artifact has to respect the continuous operations level, the API architecture recommended for the artifact has the potential to support the level of continuous operations. My suggestion for the development of the artifact is to thoroughly consider the core components which have the responsibility to keep the artifact running, and ensure components' robustness. Herein the artifact has to be available 24/7 considering that the artifact is built to be integrated in different systems that have different operations schedules.

Principle of following IS standards

The artifact approached in this research has to be integrated in an organization's infrastructure and incorporated into software applications. As discussed in chapter IS standards, following technology standards is beneficial and mandatory in an organization. For the IS artifact to be successfully integrated in the organization a standardization approach of the artifact is imperative.

A REST architectural style approach for building APIs is endorsed as a generally known and understood practice in developers' world (Preibisch, 2018). The standards-based interface imposed by the REST architectural style helps practitioners to easily dive into the operations available through APIs (ibid.).

My recommendation for the development of the artifact is to follow a REST API architecture, in order to ensure a common usability and apprehension of the system across practitioners. Moreover, applying a Rest API architecture a set of standards are included by default, such as: HTTP/ HTTPS protocols, possibility to choose a standard data structure JSON/ XML. How these standards are incorporated into the Rest API architectural style is described in the API architecture literature. And the artifact's design following the REST API architectural style is presented in the Artifact architecture chapter.

Using standards is very important for the building of artifacts in today's world. Systems are integrated into an organizational II and are expected to be further integrated with other systems (Monteiro et al., 2013). The artifact presented in this research has been designed to incorporate a set of standards. These sets of standards will make the artifact to be easily integrated in an organizational setting and customly adapted to be utilized by the practitioners.

Principle of using the artifact as a boundary resource

Boundary resources like APIs are largely used to refine and extend platform ecosystems with new technological extensions (Ghazawneh & Henfridsson, 2012). APIs are defined as boundary resources in the context in which platform owners and third-party developers are drawn together (Bondel et al., 2021).

My recommendation is to use the artifact as a resource boundary. The decision was to design the artifact as a set of APIs. "An API initiative cannot exist in isolation, but ongoing collaboration with various stakeholders inside and outside the organization is required" (Bondel et al., 2021 p. 2). Boundary resources have the potential to stimulate innovation and increase the platform's value by expanding it (Ghazawneh & Henfridsson, 2012).

Taking into account that the artifact will be embedded in various software applications, and that the artifact is to be integrated in the organization's ecosystem; the artifact can be defined as a boundary resource. The idea is to use the artifact within the organization's software applications to generate big data. Therefore the artifact's ability to be embedded into the software applications was considered from the beginning of the design.

Evaluation

The evaluation is an important part of this case study. In this section, first, FEDS was applied as a framework for conducting evaluations. FEDS is described in chapter Evaluation framework in design science. Second, I have elaborated on the evaluation of artifact's usability. Third, I have assessed all the design principles exclusively. This arrangement has guided me in developing a solid structure for the case study's evaluation part.

Evaluation applying FEDS

The FEDS has a clear and effective approach to conduct the evaluation in the DSR, therefore I have chosen to follow it for this section. The FEDS recommends first to identify the research goals (Venable et al., 2016). Related to my research the following goals of the artifact's design have been taken into account:

(1) Assess artifact's usability
(2) Discuss the impact on data collection
(3) Practical interaction between practitioners
(4) Evaluate artifact's design principles

In the Evaluation framework in design science section, the definitions of formative evaluation and summative evaluations are provided. The evaluation of the artifact can be considered as a balanced approach between the formative and summative evaluation. Literature presents formative evaluation as being focused on the process, while summative evaluation more focussed on technical requirements. For the artifact's evaluation I have taken into consideration both the process of artifact's development and technical requirements. Regarding the use of formative evaluation, the process followed during the artifact's development was "Generate/ Test cycle", presented in Figure 3.2. The process of following the "Generate/ Test cycle" for this case study is illustrated in Figure 3.5.

Regarding the use of summative evaluation, the artfact's design principles as in technical requirements were thoroughly evaluated. The artifact was implemented and developed according to the design principles. The design principles were evaluated as part of the process "Generate/ Test cycle". As illustrated in the Figure 3.5 the IS artifact has undergone the following:

(1) Development of the alpha version was succeeded by the Evaluation of the alpha version.

(2) The Evaluation of the alpha version was followed by the Development of beta version

(3) The Development of beta version was followed by Evaluation of beta version

During the evaluation's episodes, the design principles and other usability aspects were discussed in the context of a developed version of the artifact. Considering the noted goals and the evaluation approach, a detailed pathway on how these aims were to be achieved was underlined for this research as follows:

- Create a use case example which helps practitioners to test and evaluate the artifact. This use case example was chosen with regard to the practitioners' background in the TietoEvry context. A real example was incorporated in the artifact: the "Login" data. Based on this example the evaluators could provide an explicit feedback and effortlessly assess the artifact. In order to achieve that the artifact had to be designed in a manner that respects the principle of availability, having the artifact's design approachable for the practitioners.

- Understanding the risks and vagueness of the research at an early stage, have helped to envisage and prepare a plan on how to address them. First, was taken into consideration the fact that the artifact cannot be integrated directly in the Tietoevry company for testing. Therefore, the plan was to develop a real test case scenario managed by the artifact. And also provide examples on further utilities and features which can be applied. Second, the DSR cycle proposed by Kuechler & Vaishnavi (2015) was followed as a strategy to attain pre-established research aims for this case study.

- The implications of the artifact on the existing systems in the organization was discussed and reflected upon. Even though the introduction of errors in the existing company's software solutions is limited due to the architectural approach of the artifact, still there might be some challenges. The challenges can be related to the resources necessary in order to incorporate the artifact and the performance which might be affected. Regarding the ethical concerns referring to the welfare of animals and impact on future generations, I couldn't find any direct implications to be considered.

- During the research process one of my objectives was to wisely distribute the available resources. Moreover, I have addressed the challenges on the way based on the context.

Venable et al. (2016) recommend researchers to establish strategies to be selected for the study. Research strategies are presented in the subchapter Choose a strategy or strategies for evaluation. Venable et al. (2016) suggests a plan of action, in the context of my research analysis the following have been considered:

(1) The challenge the design initially faced was to find a way to make the artifact usability straightforward. This provocation was managed by following the principle of standards. This principle is described in the subchapter Principle of following IS standards.

(2) The need to calculate the amount of costs was not relevant for this research. The tools used for development were open source and the evaluators, as in practitioners working for TietoEvry have volunteered for the project. But a strategy on how to efficiently use the time allocated for the interviews was considered. The strategy was to have most of the interviews as semi-structured interviews. This type of interview forces the researcher to prepare questions for the interview. The interview becomes organized and efficient, due to having a guideline for the interview’s flow.

(3) The artifact was a mix between a conceptual design and a technical tool. Therefore the strategy during the evaluations was to receive feedback from practitioners about the conceptual idea, the artifact’s utility and the design principles.

(4) Even though the artifact purpose was quite complex, the conceptual design was simple. The complexity of the artifact is more connected to the various scenarios in which the artifact can be employed. Therefore the strategy to create a software application to have a foundation for the discussions was employed. As mentioned earlier a case scenario of “LogIn” data was used for the evaluations. This use case scenario viewed as part of a web application was employed to simplify the evaluations and have a clear examination. Illustrations of the application are presented in Appendix B.

Taking into account the goals of the evaluation and strategies prepared for this evaluation, the evaluation process has been well structured and straightforward. The literature recommendations for artifact's evaluation assisted significantly in this study.

Evaluate artifact's usability

For the artifact's usability evaluation an application named "Custom workflow" was used. The application was developed to base the discussions with the practitioners on a software application example. The discussion ground was user's "LogIn" data. This example was reflected on for the evaluation of both artifact's usability and design principles. The "Custom workflow" web application's examples of user registration and login account is presented in the Appendix B, Custom workflow application use case example. Followed by the "LogIn" data artifact's case scenario, Artifact's "LogIn" data case scenario.

The strategy to use a specific example was adopted to mainly assess the artifact's usability and design principles. The practitioners have pointed out that this has made it easier to understand the artifact's functions. When reflecting about the artifact and the way it will be accessed in the context of the organization, one of the senior software developers (**participant 6**) pointed out the importance to know the application and its features and comprehend "what application is about and understand what data do we need to gather from that system." (**participant 6**).

Regarding the use of the artifact as a communication tool between data scientists and software developers. Data scientist lead (**participant 2)** mentioned the following "I see that this would be valuable in some cases, let's say like, sometimes you have data and neither you nor customer understand how exactly it was generated and then, maybe software developers best know how and what they have generated in their software".

Participant 3, who works as a data scientist, said "I am able to imagine how this would help". This remark was in the context of artifact's function as a communication tool between practitioners. Additionally, **participant 3** has described a role from a previous workplace called service manager. The service manager "acted between data scientists, data analysts and business side" (**participant 3**). This was brought up as an example that the artifact would have a similar role of acting between two qualifications: software developers on one side and data scientists on the other side.

One of the senior software developers (**participant 5**) has pointed out that the artifact would be a good solution to engage the practitioners in collaboration, especially if the practitioners are from different sections and do not interact on a daily basis. When discussing the artifact's use as a tool to exchange information and data between data scientists and software developers, **participant 3** remarked that "it seems as a good communication improver, so something that could potentially improve the understanding from both sides and the transfer of information".

Related to artifact's function to improve data collection, practitioners withhold from giving elaborate answers. The comments were that from a theoretical perspective this looks promising, but will have to be tested. The reason for that was that in order to see if the artifact improves data collection, the artifact has to be used for some time to collect data. And afterwards, having data collected, an assessment of ML models results could be the resolution of the artifact's ability to improve data quality.

Design principles evaluation

When designing the artifact, three main principles are recommended to be followed. The principles are described in the Artifact design principles section. Each principle has been evaluated with Tietoevry's practitioners and described in the sections below.

Principle of system availability evaluation

Principle of availability can be seen and analyzed from two perspectives:

(1) System to be available for the users

(2) The ability of the application to function as expected

From the interview with the data science lead (**participant 2**) from the beginning the importance of availability was specified, referring to different actors such as sales personnel and product managers having to "know that we have this product". This emphasizes the need of a system to be known and have visibility in an organization. Therefore, having a system which is available for the practitioners is vital in an organization in order to be integrated and used accordingly.

Related to the discussion of having access to the artifact's features via GUI such as Swagger UI, explained in chapter Artifact interface. **Participant 6** had a positive attitude toward the artifcat's access to the functionality saying "a very good way to understand it, I think".

Another remark coming from **participant 6** related to the GUI setup was that the development of the artifact requires "some work to be done, in order to really support the understandability". From this remark one can conclude that for the artifact's GUI has to be put some thought into the GUI's content and visual design implementation.

Principle of following IS standards evaluation

Among the practitioners the importance of following standards in the information systems realm is well accepted and acknowledged.

When inspecting the artifact functionalities the data scientist (**participant 3**) was familiar with the standards used such as JSON format of the objects and the elements of the APIs structure. As a data scientist, **participant 3** has mentioned that JSON structures are often used in the data science fields when it comes to the data format. Following standards also helps the application to be easier integrated into the company's current IS infrastructure, as many of the practitioners are well known with the IS standards.

During the structured interview related to the architectural style and the standards used for the development of the artifact, **participant 5** agreed that the artifact is using the standards such as HTTP protocol, JSON data structures. **Participant 5** added that by following the recommended standards in the IS field makes the communication and integration of the artifact easier for the practitioners with a technical background.

Using the Principle of using the artifact as a boundary resource evaluation

The way the artifact is recommended to be used has to be considered. The manner in which the artifact is to be used has an impact on the process which has to be followed in order to achieve consistency.

The idea of the artifact is to be integrated into other software applications in order to gather necessary data, but also to facilitate the communication between data scientists and software developers regarding which data to be stored and what format to be used. Related to the artifact's function to gather necessary data from the application, the recommendation is to use the artifact as a boundary resource.

Related to the use of the artifact as a boundary resource **participant 3** remarked that "basically, the intention is to save some time", which is one of the boundary resource aspects discussed in the Boundary resource section related to adopting an external component rapidly. Regarding artifact's design and use as a boundary resource, **participant 4** mentioned that "you have to do some programming for each type of system you plan to gather data from" .Which is an accurate description of the fact that the design of the artifact makes the artifact a boundary resource. The artifact is practically a tool to be embedded into the existing applications in order to collect data. From the software development team's perspective it will require time and resources to be spent in order to adopt the artifact, and it might affect the application's performance.

Discussion

This case study has addressed the following research question:

"Can an IT artifact facilitate the collaboration between software developers and data scientists, support boundary spanning competence and data collection for ML purpose?"

I have followed the DSR methodology to answer this research question. Design science research offers a framework to study IT artifacts employed to solve relevant problems (Aier & Gleichauf, 2010). By following the DSR research framework the process of designing, developing and evaluating the artifact was systematic. The DSR paradigm has guided the case study and was an effective framework for this research. Additionally, the DSR paradigm has helped me as a researcher to build knowledge and wrap up all the aspects of this research.

At the center of this research is an IT artifact which aims to solve two real-world problems. Hevner et al. (2004) assert that design science's concern is an innovative IT artifact (Hevner et al., 2004). This artifact is built to solve existing problems in an organization (ibid.). In this study, the IT artifact attempts to address the following real-world problems discovered in Tietoevry company:

1) Lack of collaboration between data scientists and software developers
2) Poor data quality for ML purposes

The propositions developed by IS researchers enriches IS design science, "and hence knowledge, about all facets of design and design thinking" (Mckay et al., 2012, p.135).

The following are my main contributions for this research:

1) Establishing design principles of how the artifact is designed, to help others to build a similar artifact
2) Giving an explanation of how the IS artifact acts as a boundary spanning tool

Empirical work has been conducted: (1) to design an artifact which addresses the identified organizational problems, (2) to evaluate the artifact in the organizational context and (3) to understand its role in the boundary spanning. The design of the artifact was tailored to solve research problems. The problems presented in Figure 3.3, were identified during the first unstructured interview with one of the project managers (**participant 1**).

An in depth understanding of the design process is valuable for the design and development of the artifact (Zhao et al., 2010). A clear overview of the design process improves the artifact's features and assists the artifact's quality (ibid). But, also this design activity process gives researchers the possibility to build knowledge (ibid). Therefore for this research special attention was paid to the design process. In order to structure the design process DSR recommendations have been followed. The following concepts were employed: DSR cycle and FEDS framework.

The DSR cycle has helped to create an "itinerary" for the artifact's design process. Figure 3.1 depicts the phases of the DSRC. Hevner et al. (2004) define the design process as a series of knowledge based tasks which create an innovative artifact (Hevner et al., 2004) . Considering that the artifact is at the core of this research, the design process played a fundamental role for this study. "Design science addresses research through the building and evaluation of artifacts designed to meet the identified business need" (Hevner et al., 2004, p.79-80). Evaluation process is elaborated in the Evaluation chapter and it is an important part of the study's findings. The FEDS framework employed for this research has guided the evaluation process. Moreover, the FEDS framework assures an appropriate evaluation of the artifact and has a structured appraisal. The artifact's evaluation was focused on the artifact's usability and on the artifact's design principles.

Review IS artifact's usability

Following are research aims related to artifact's usability:

(1) To explore the utility of the artifact from the practitioners' perspective with regard to boundary spanning

(2) To examine the IT artifact's usefulness in collecting data for ML models, by employing practitioner's feedback

As the evaluation revealed, practitioners have helped this case study with straight forward feedback related to the artifact's utility as a communication tool. Regarding the artifact's effectiveness on improving data collection for ML purposes, practitioners could not objectively analyze the artifact's benefit.

The know-how exchange in an organization often emerges in competence boundary spanning (Levina & Vaast, 2005). Practitioners in this research have also indicated the knowledge exchange when the IS artifact is employed as a communication tool. When building new products, experts with different backgrounds have to communicate (Levina & Vaast, 2005). Same insight came from **participant 2**, related to the new data science projects, coming with new sets of data. To be able to understand the data sometimes it requires knowing how the data was generated. This information can be acquired from software developers. This communication can be more effective by employing the researched IT artifact. Hence, the conclusion that in such a case the IT artifact would play an important role in boundary spanning between data scientists and software developers.

Levina & Vaast (2005) explain the boundary object concept (Levina & Vaast, 2005). The boundary object can be a prototype, characterized by such aspects as abstraction, adaptability and modularization (ibid.). Moreover, the boundary object is used in the middle of different expert fields (ibid.). The IT artifact researched for this case study can be defined as a boundary object. Boundary object is a tool used to reduce the challenges practitioners face during the process of exchanging expertise (Levina & Vaast, 2005). A similar observation was provided by **participant 5.** Regarding the distance between practitioners, which can be a challenge for open communication. The practitioner has noted that the artifact would bring the practitioners closer and make practitioners engage more in the collaboration.

Although, the artifact's ability to improve the collaboration of data scientists and software developers was only discussed from a theoretical point of view. Participants understood quite well the artifact's purpose. And agreed that a closer cooperation between data scientists and software developers could bring new knowledge.

Review artifact's design process

Research aims related to the artifact's design:

(1) Accumulate knowledge throughout the process of designing and developing the artifact

(2) Establish design principles for the development and usage of the artifact

A set of design principles have been established to help others to build a similar artifact and further explore its usability. “Design principles are an important part of design theory” (Gregor et al., 2020). Design principles were formulated based on the existing literature, presented in chapter Kernel theories. The knowledge related to design principles comes from the design theory (Gregor et al., 2020). An important part of artifact’s development is to formulate design principles (ibid). In order to establish design principles DSR recommendations were followed. Three design principles were established for this research: principle of availability, principle of following IS standards and principle of using the artifact as a boundary resource. These design principles have been discussed and evaluated together with the practitioners.

From the literature system availability represents a system’s capability to perform the inbuilt operations and ensure access to these operations for the users. Pickard & Childs (2013) suggest to select a system availability level based on users’ requirements. The availability level of the artifact was established during the conversations with the participants. The most appropriate level for the artifact was accepted to be “*continuous operation*”. This level means that the artifact will have to run 24/7, and the maintenance related to the artifact’s updates will have to be scheduled.

Following IS standards when building IT systems is widely accepted in the IT community (Monteiro et al., 2013). Adopting IS standards when designing the IT artifact for this research was one of the requirements all practitioners strongly agreed with. The REST architectural style employed for artifact’s development cemented the use of IS standards. As Preibisch (2018) highlights the fact that the REST API approach is well-known for practitioners in the IT field. Moreover, the guidelines for API development are standardized and follow the well known IS standards such as: HTTP/ HTTPS protocol, JSON/XML data formats etc ((Preibisch, 2018). Similar reviews about the REST API approach, using IS standards as an accepted actuality came from Tietoevry’s practitioners. **Participant 3** felt comfortable to see data formatted as JSON. **Participant 5** confirmed that APIs are preferred among the software developers for different integrations.

Using the artifact as a boundary resource is a design principle focused on the way the artifact is to be used. The artifact's function, to collect data from existing software applications implies that the artifact will have to be embedded into those software applications. Ghazawneh & Henfridsson, 2012 define boundary resource as a software component used to extend the services of an infrastructure by enabling software developers to freely use this software component. One of the characteristics of the boundary resources is the embeddedness into other services and applications. The principle of using the artifact as a boundary resource was discussed with participants as a matter of course. The artifact function to collect data from software applications in a specific format, requires the IS artifact to be integrated into the software applications.

Conclusion

The latest technological advancements in the ML field and the capability data has to transform and improve business operations, has become attractive for the companies to adopt a data driven path (Mohanty & Vyas, 2018). The benefits of being a data driven company have been widely researched. Mohanty & Vyas (2018) assert that taking advantage of a data driven approach gives companies a competitive advantage.

Despite the fact that companies can benefit from a data driven approach, there are still challenges companies face in order to fully leverage their data driven potential. I have discussed these challenges with one of Tietoevry's project managers (**participant 1**). The problem formulation has been cemented based on this conversation with **participant 1**. The problem formulation started with the assumption that there is a lack of coordination between software developers and data collection for ML purposes has to be improved.

Software companies are most likely to embrace data-driven culture easier given the technological background. This also means that software companies will be the first to discover new needs regarding data driven organizational setup. According to Kim et al (2016) a necessity of a new role emerges from the need software companies have to search for data scientists with software engineering knowledge (Kim et al., 2016, p.104). The practice of combining different sets of skills across distinct fields is common in companies (Lindgren et al., 2008). As literature shows, from the work between different experts, new knowledge can be created (Levina & Vaast, 2005). Moreover, this new knowledge has the potential to propel innovation and improve businesses (ibid.). Figure 2.2 illustrates how boundary spanning applies for this study, at the intersection of the software development field and data science field. The researched IT artifact has the potential to drive knowledge exchange between data scientists and software developers, and as a result create value for an organization. Almost all practitioners agreed that the artifact can be a good communication tool. From the discussions with participants the artifact potential was seen as a helpful boundary object which will help practitioners to share their knowledge.

Kim et al. (2016) acknowledged the need of a professional with data analyst and software development knowledge. The argument presented in this research is that for a professional to build the competency at the cross of data science and software development fields, boundary spanning between these expertise is needed. The IT artifact is designed to take the role of enacting boundary spanning competence. Participants in this study have admitted that the use of an IT system such as the IT artifact presented, could facilitate the communication. Moreover, the IT artifact would also structure the knowledge sharing and document this process. Hence, the first finding of this case study is that a collaboration between data scientists and software developers is necessary in software companies. To the extent of my knowledge, the cooperation between data scientists and software developers was not yet studied.

The main key finding of this research is that IS artifact has the potential to improve collaboration between practitioners and act as a boundary spanning tool. Although, the investigation about the challenges related to data collection for ML purposes, has not proven to be fruitful. One can say that the design principle established for the IT artifact, can encourage other researchers to develop such an artifact. And further research the effect the artifact has on the data collection for ML purposes.

References

Aier, S., & Gleichauf, B. (2010). Applying Design Research Artifacts for Building Design Research Artifacts: A Process Model for Enterprise Architecture Planning. *International Conference on Design Science Research in Information Systems. Springer, Berlin, Heidelberg*, 333-348.

Anderson, C. (2015). *Creating a Data-driven Organization: Practical Advice from the Trenches*. O'Reilly Media Incorporated.

Arankalle, C. (2020). *The artificial intelligence infrastructure workshop : build your own highly scalable and robust data storage systems that can support a variety of cutting-edge AI applications* (1st ed.). Packt.

Atchison, L. (2016). *Architecting for Scale: High Availability for Your Growing Applications*. O'Reilly.

Auburn, M., Bryant, D., & Cough, J. (2022). *Mastering API Architecture*. O'Reilly Media, Inc.

Bean, R. (2022, 02 24). Why Becoming a Data-Driven Organization Is So Hard. *Harvard Business Review*. https://hbr.org/2022/02/why-becoming-a-data-driven-organization-is-so-hard

Bondel, G., Landgraf, A., & Mathes, F. (2021, July). API Management Patterns for Public, Partner, and Group Web API Initiatives with a Focus on Collaboration. *European Conference on Pattern Languages of Programs (EuroPLoP'21)*. https://dl.acm.org/doi/10.1145/3489449.3490012

Brewer, E. (2000). Towards robust distributed systems. *Proceedings of the Nineteenth Annual ACM Symposium on Principles of Distributed Computing*. 10.1145/343477.343502

Cairns, P., & Cox, A. L. (Eds.). (2016). *Research Methods for Human-Computer Interaction*. Cambridge University Press. https://doi-org.ezproxy.uio.no/10.1017/CBO9780511814570

De Sordi, J. O. (2021). *Design Science Research Methodology: Theory Development from Artifacts*. Springer International Publishing. https://doi.org/10.1007/978-3-030-82156-2

Encyclopædia Britannica Inc. (2020, 07 02). API. *Encyclopædia Britannica Online*. https://academic-eb-com.ezproxy.uio.no/levels/collegiate/article/API/443751

Engert, M., Evers, J., Hein, A., & Krcmar, H. (2022). The Engagement of Complementors and the Role of Platform Boundary Resources in e-Commerce Platform Ecosystems. *Information Systems Frontiers*. https://doi.org/10.1007/s10796-021-10236-3

Ghazawneh, A., & Henfridsson, O. (2012). Balancing platform control and external contribution in third-party development: the boundary resources model. *Information Systems Journal*, *23*, 173-192. 10.1111/j.1365-2575.2012.00406.x

Gregor, S., Chandra Kruse, L., & Seidel, S. (2020). Research Perspectives: The Anatomy of a Design Principle. *Journal of the Association for Information Systems*, *21*(6), 1622-1652. 10.17705/1jais.00649

Gregor, S., & Hevner, A. R. (2013, 06). Positioning and Presenting Design Science Research for Maximum Impact. *MIS Quarterly*, *37*(2), 337-355.

Gudivada, V. N., Apon, A., & Ding, J. (2017, July). Data Quality Considerations for Big Data and Machine Learning: Going Beyond Data Cleaning and Transformations. *International Journal on Advances in Software*, *10*.

Hagen, J. A., & Hess, T. (2020). Linking Big Data and Business: Design Parameters of Data-Driven Organizations. *AMCIS 2020 Proceedings*, *5*. https://aisel.aisnet.org/amcis2020/data_science_analytics_for_decision_support/data_science_analytics_for_decision_support/5

Hanseth, O., Monteiro, E., & Hatling, M. (1996). Developing Information Infrastructure: The Tension Between Standardization and Flexibility. *Science, Technology, & Human Values,*, *21*(4), 407-426.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004, 03). Design Science In Information Systems Research. *MIS Quarterly*, *28*(1), 75-105.

Kallio, H., Pietila, A.-M., Johnson, M., & Kangasniemi, M. (2016). Systematic methodological review: developing a framework for a qualitative semi-structured interview guide. *Journal of Advanced Nursing*, *72*(12), 2954-2965. 10.1111/jan.13031

Kim, M., Zimmermann, T., DeLine, R., & Begel, A. (2016). The Emerging Role of Data Scientists on Software Development Teams. *2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE)*, 96-107. https://doi.org/10.1145/2884781.2884783

Kim, M., Zimmermann, T., DeLine, R., & Begel, A. (2018, 11). Data Scientists in Software Teams: State of the Art and Challenges. *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, *44*(11), 1024-1038.

Kuechler, B., & Vaishnavi, V. (2008). On theory development in design science research: anatomy of a research project. *European Journal of Information Systems*, *17*(5), 489-504. https://doi.org/10.1057/ejis.2008.40

Kuechler, W., & Vaishnavi, V. K. (2015). *Design Science Research Methods and Patterns: Innovating Information and Communication Technology, 2nd Edition*. CRC Press.

Levina, N., & Vaast, E. (2005, 06). The Emergence of Boundary Spanning Competence in Practice: Implications for Implementation and Use of Information Systems. *MIS Quarterly*, *29*(2), 335-363.

Lindgren, R., Andersson, M., & Henfridsson, O. (2008, 11). Multi-contextuality in boundary-spanning practices. *Information systems journal (Oxford, England)*, *18*(6), 641-661.

Markus, M. L., Majchrzak, A., & Gasser, L. (2002, 09). A Design Theory for Systems That Support Emergent Knowledge Processes. *MIS Quarterly*, *26*(3), 179-212.

Markus, M. L., Majchrzak, A., & Gasser, L. (2002, 09). A DESIGN THEORY FOR SYSTEMS THAT SUPPORT EMERGENT KNOWLEDGE PROCESSES'. *MIS Quarterly*, *26*(3), 179-212.

Marsland, S. (2015). *Machine Learning (2nd ed.)*. CRC Press. https://doi.org/10.1201/b17476

Marsland, S. (2015). *Machine Learning: An Algorithmic Perspective* (2nd ed.). CRC Press. 10.1201/b17476

McCarthy, R. V., McCarthy, M. M., Halawi, L., & Ceccucci, W. (2019). *Applying Predictive Analytics: Finding Value in Data*. Springer International Publishing.

Mckay, J., Marshall, P., & Hirschheim, R. (2012, 06). The Design Construct in Information Systems Design Science. *Journal of Information Technology*, *27*(2).

Mohanty, S., & Vyas, S. (2018). *How to Compete in the Age of Artificial Intelligence: Implementing a Collaborative Human-Machine Strategy for Your Business*. Apress.

Monteiro, E., Pollock, N., Hanseth, O., & Williams, R. (2013). From artefacts to infrastructure. *Springer-Verlag*. https://doi.org/https://doi.org/10.1007/s10606-012-9167-1

Morabito, V. (2015). *Big Data and Analytics: Strategic and Organizational Impacts*. Springer International Publishing.

Muller, M., Lange, I., Lang, D., Piorkowski, D., Tsay, J., Liao, Q., Dugan, C., & Erickson, T. (2019). How Data Science Workers Work with Data. *Proceedings of the 2019 CHI Conference on Human Factors in Computing System*, 1-15. https://doi.org/10.1145/3290605.3300356

Myers, M. D., & Venable, J. R. (2014). A set of ethical principles for design science research in information systems. *Information & Management*, *51*(6), 801-809. https://doi.org/10.1016/j.im.2014.01.002

Ozkaya, I. (2021). The Developer Nation. *IEEE Software*, *39*(1), 3-6. https://doi.org/10.1109/MS.2021.3118481

Petrova-Antonova, D., & Tancheva, R. (2020). Data Cleaning: A Case Study with OpenRefine and Trifacta Wrangle. *Quality of Information and Communications Technology*, 32-40. https://doi.org/10.1007/978-3-030-58793-2_3

Pickard, A. J., & Childs, S. (2013). *Research Methods in Information*. Facet Publishing.

Piedad, F., & Hawkins, M. (2001). *High Availability: Design, Techniques, and Processes*. Prentice Hall PTR.

Preibisch, S. (2018). *API Development: A Practical Guide for Business Implementation Success*. Apress.

Roh, Y., Heo, G., & Euijong Whang, S. (2021). A Survey on Data Collection for Machine Learning: A Big Data - AI Integration Perspective. *TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, *33*(4), 1328-1347.

Roh, Y., Heo, G., & Whang, S. E. (2021). A Survey on Data Collection for Machine Learning: A Big Data - AI Integration Perspective. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, *33*(4), 1328-1347. 10.1109/TKDE.2019.2946162

Techopedia. (2017, December 12). *What is a Developer? - Definition from Techopedia*. Techopedia. Retrieved January 22, 2022, from https://www.techopedia.com/definition/17095/developer

Techopedia. (2021). *Enterprise Application (EA)*. Techopedia. https://www.techopedia.com/definition/24804/enterprise-application-ea

Tetoevry. (n.d.). *Retail lending Enhance your retail lending process, from first customer contact all the way to final accounting.* Tietoevry. https://www.tietoevry.com/en/industries/financial-services/credit/retail-lending/

TietoEVRY. (2022). *Data, AI, and Analytics*. TietoEVRY. Retrieved January 20, 2022, from https://www.tietoevry.com/en/services/data-ai-and-analytics/

TietoEVRY corporation. (n.d.). *Who we serve*. TietoEVRY. Retrieved December 7, 2021, from https://www.tietoevry.com/en/industries/

TietoEVRY corporation. (2018, May 8). *50 years of shaping the future*. TietoEVRY. Retrieved December 7, 2021, from https://www.tietoevry.com/en/blog/2018/05/50-years-of-shaping-the-future/

TietoEVRY Corporation. (2019, 12 05). *Merger between Tieto and EVRY completed – TietoEVRY established*. TietoEVRY. https://www.tietoevry.com/en/newsroom/all-news-and-releases/stock-exchange-releases/2019/12/merger-between-tieto-and-evry-completed--tietoevry-established/

2021 SmartBear Software. (n.d.). *Swagger UI*. Swagger Supported by SMARTBEAR. https://swagger.io/tools/swagger-ui/

USC Libraries. (2022, 05 09). *Research Guides: Organizing Your Social Sciences Research Paper: 5. The Literature Review*. Research Guides. Retrieved May 12, 2022, from https://libguides.usc.edu/writingguide/literaturereview

van der Aalst, W. M. P. (2014). Data Scientist: The Engineer of the Future. In: Mertins K., Bénaben F., Poler R., Bourrières JP. In K. Mertins, F. Bénaben, J.-P. Bourrières, & R. Poler (Eds.), *Enterprise Interoperability VI: Interoperability for Agility, Resilience and Plasticity of Collaborations*. Springer International Publishing.

Venable, J., Pries-Heje, J., & Baskerville, R. (2016). FEDS: a Framework for Evaluation in Design Science Research. *European Journal of Information Systems*, *25*(1), 77-89. https://doi.org/10.1057/ejis.2014.36

Werder, K., Seidel, S., Recker, J., Berente, N., Gibbs, J., Abboud, N., & Benzeghadi, Y. (2020, 04). Data-Driven, Data-Informed, Data-Augmented: How Ubisoft's Ghost Recon Wildlands Live Unit Uses Data for Continuous Product Innovation. *Sage*, *62*(3), 86-102. 10.1177/0008125620915290

Zhao, J. L., Winter, R., & Aier, S. (2010). *Global Perspectives on Design Science Research* (J. L. Zhao, S. Aier, & R. Winter, Eds.). Springer.

Zhu, X., Niu, B., Whitehead Jr., E. J., & Sun, Z. (2018, 11). An empirical study of software change classification with imbalance data-handling methods. *Wiley*, *48*(11), 1968-1999.

Appendices

Appendix A

Interview guide designed for data scientist lead practitioner

1) What is the structure of the AI department? What roles do people have?
2) What is the current approach for data collection? Can you name some of the challenges related to data collection you have experienced?
3) Are software developers interacting with data scientists?
(If not, do you think the cooperation between software developers and data scientists can bring some value)
4) Do you have any projects regarding collecting data on how users interact with the application?
5) Do you believe that machine learning could benefit from this data, for instance providing some insights about how the application can be improved based on users' behavior and further automate some steps in the current features of the application?
6) What is your view on expertise sharing and knowledge sharing between software developers and data scientists?

Interview guide designed for data scientist

1) Do you see this artifact as a tool to help you better interact with software developers?
2) Would it be easier to ask for a specific set of data by using the artifact?
3) Do you believe that the artifact has the potential to improve data collection?
4) Do you think about using this tool to improve communication with software developers? In what sense is that
5) Did you have any experience working with software developers or other specialists like business analysts in your work,
6) Would you say that it can be a gap between the knowledge practitioners have and it is hard to share

7) I see this tool more as a, IT artifact that can skip a few steps in the communication between software developers and data scientists,
8) What is your view related to the artifact?

Interview guide designed for software development team lead

1) Does your department collaborate with data scientists?
2) Do you have any form of data related to how the users interact with the application?
3) Do you think the cooperation between software developers and data scientists can bring some value?
4) Do you believe that a software application could benefit from data related to how users interact with the application, for instance providing some insights about how the application can be improved based on users' behavior and further automate some steps in the current features of the application?
5) What is your view on expertise sharing and knowledge sharing between software developers and data scientists?

Interview guide designed for software developers

1) Do you see this artifact as a tool to help you better interact with data scientists?
2) Would it be easy to embed the artifact in a software application?
3) Do you believe that the artifact has the potential to improve data collection?
4) What is your view related to the artifact?

Structured interview guide designed for software developers

1. When thinking about embedding an information system would you prefer an IS with API architecture or another one?
2. Do you think an API architecture for the artifact is appropriate for this context?

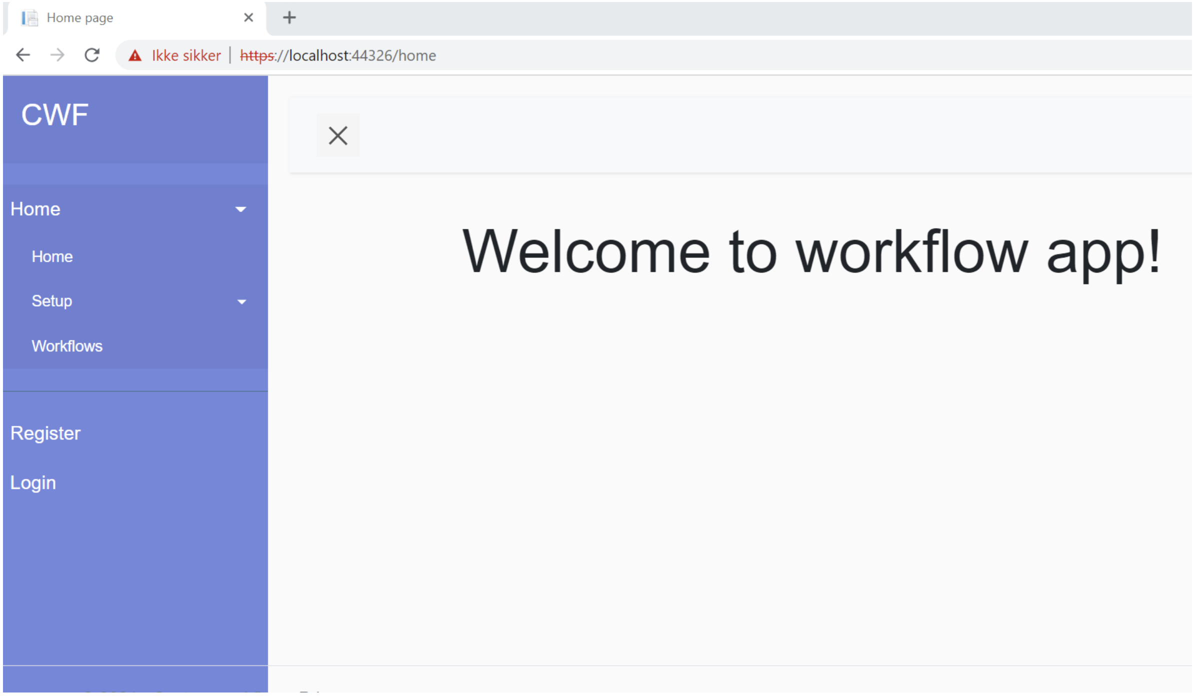
3. Do you see Swagger UI for an API based artifact as a good tool to visualize the data and understand the endpoints?
4. Can you see the API approach as a possibility to communicate with other practitioners with a tech background?

5. Do you think the artifact respects the IS standards, considering the RESTful architectural approach followed?
6. What would be the effects for a software team in adopting the artifact?
7. Do you believe that the artifact can be efficiently used to communicate with data scientists?
8. Do you think the artifact's architecture makes it highly available to other IT practitioners?
9. What is your opinion on using a highly distributed database system for storing the data?

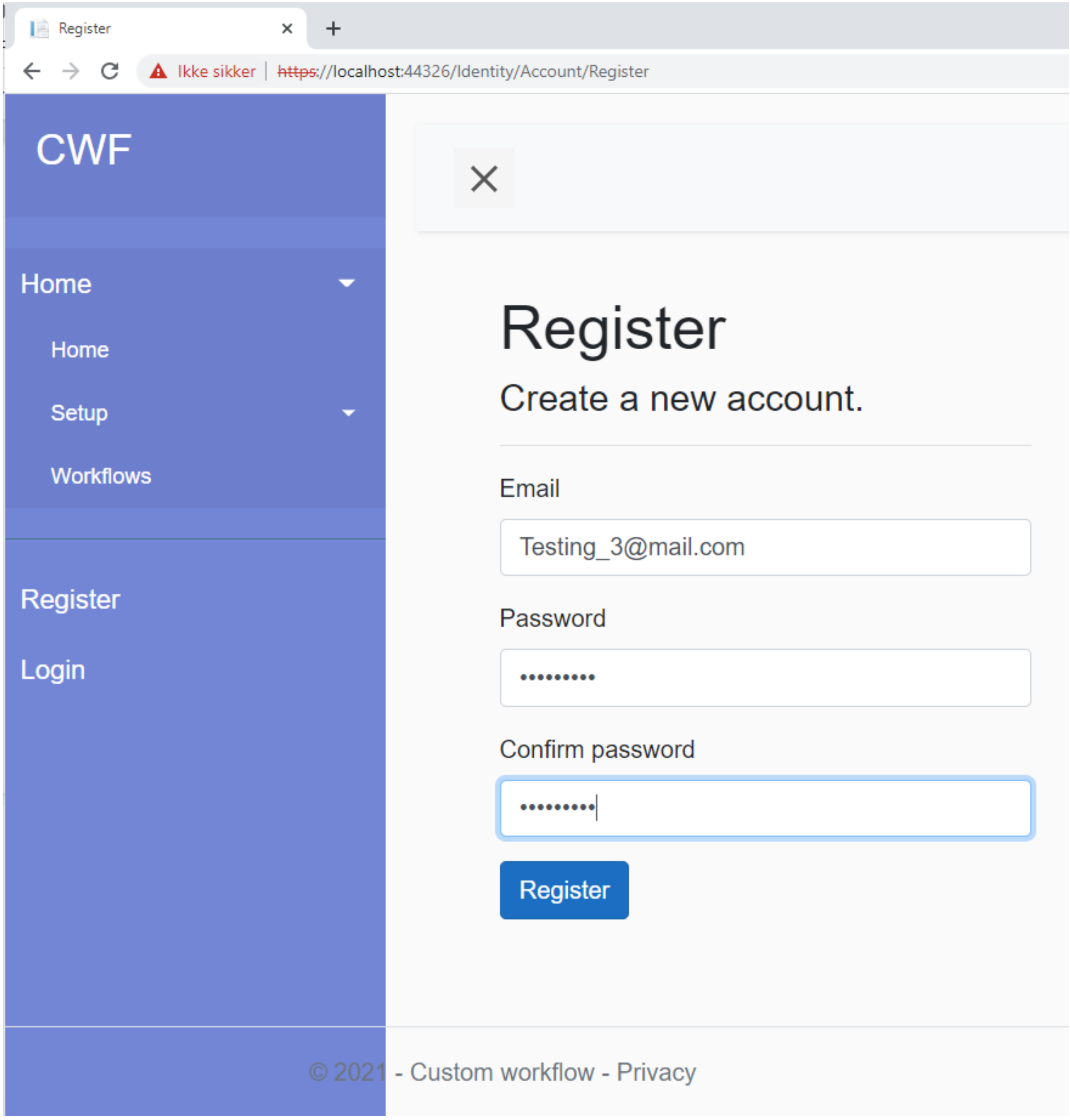
Appendix B

Custom workflow application use case example

Home page web application



Register user



https://localhost:44326/Identity/Account/Register

Register

Create a new account.

Email

Testing_3@mail.com

Password

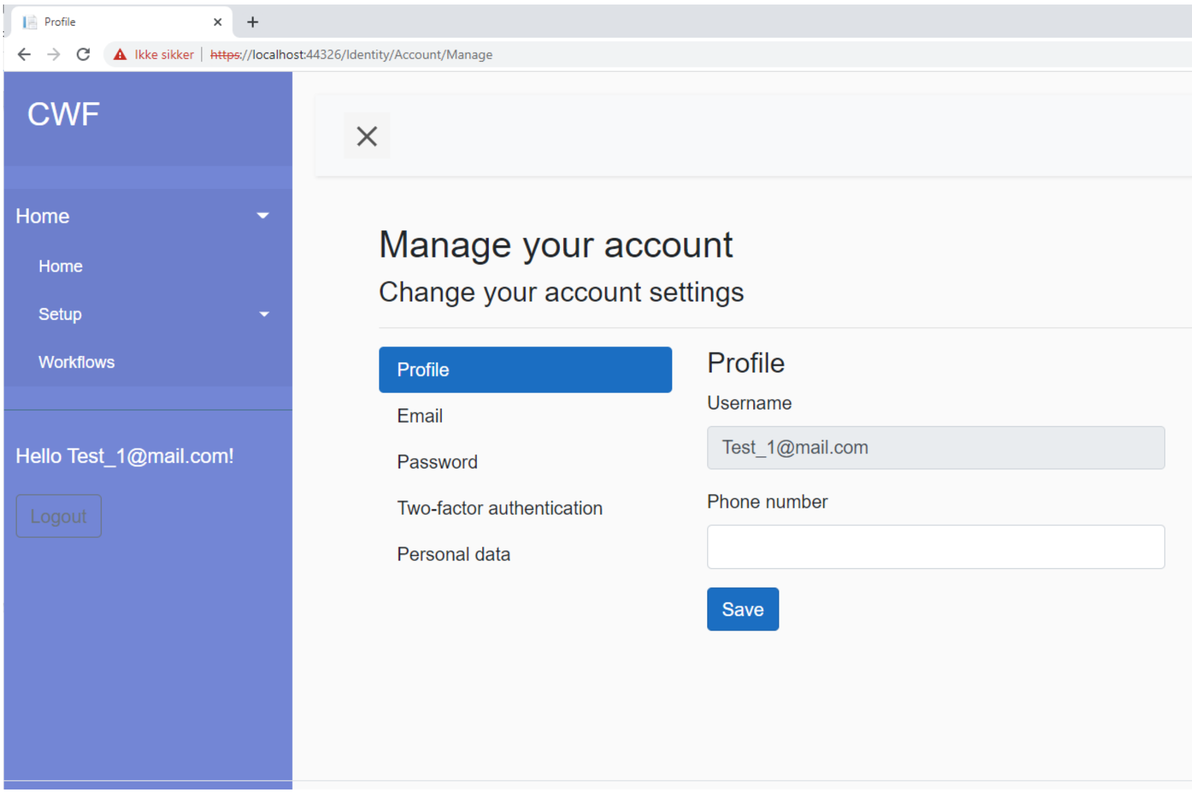
•••••••••

Confirm password

•••••••••

© 2021 - Custom workflow - Privacy

Logged in user


Profile
Ikke sikker | https://localhost:44326/Identity/Account/Manage
CWF
Home
Home
Setup
Workflows
Hello Test_1@mail.com!
Logout

Manage your account

Change your account settings

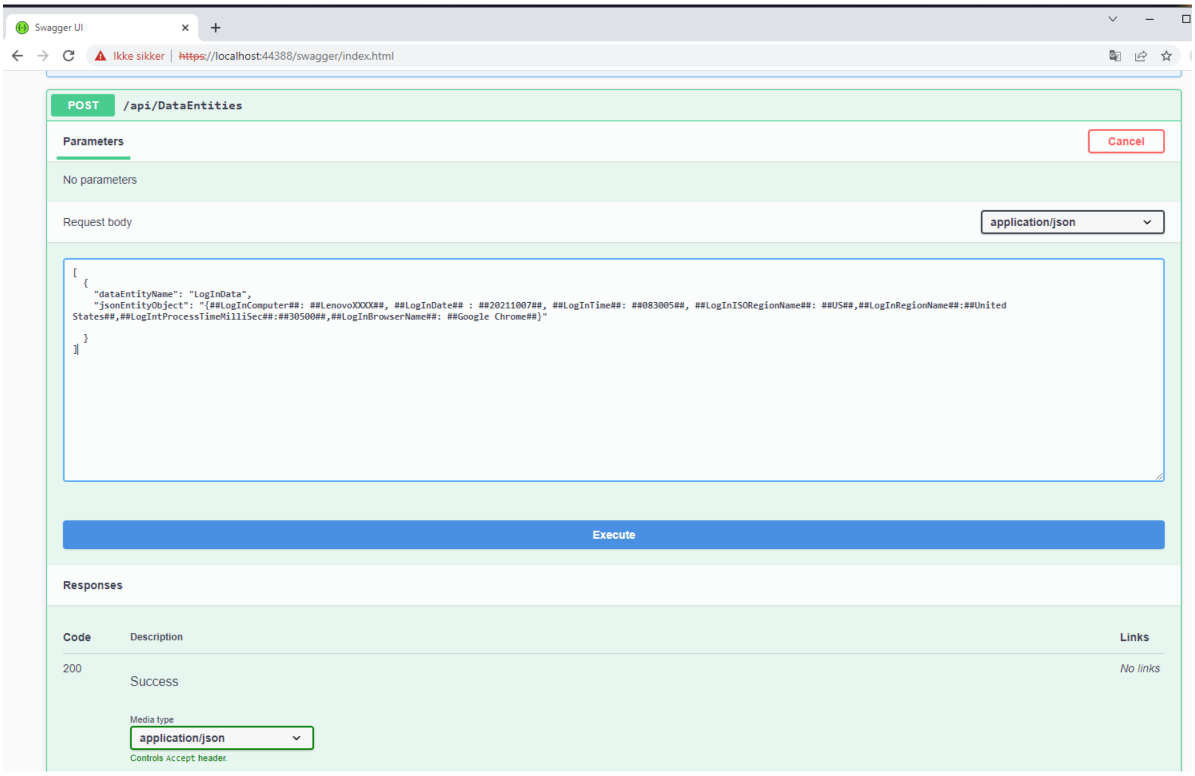
Profile

Username: Test_1@mail.com

Phone number:

Artifact's "LogIn" data case scenario

Create "LogIn" data entity



No parameters

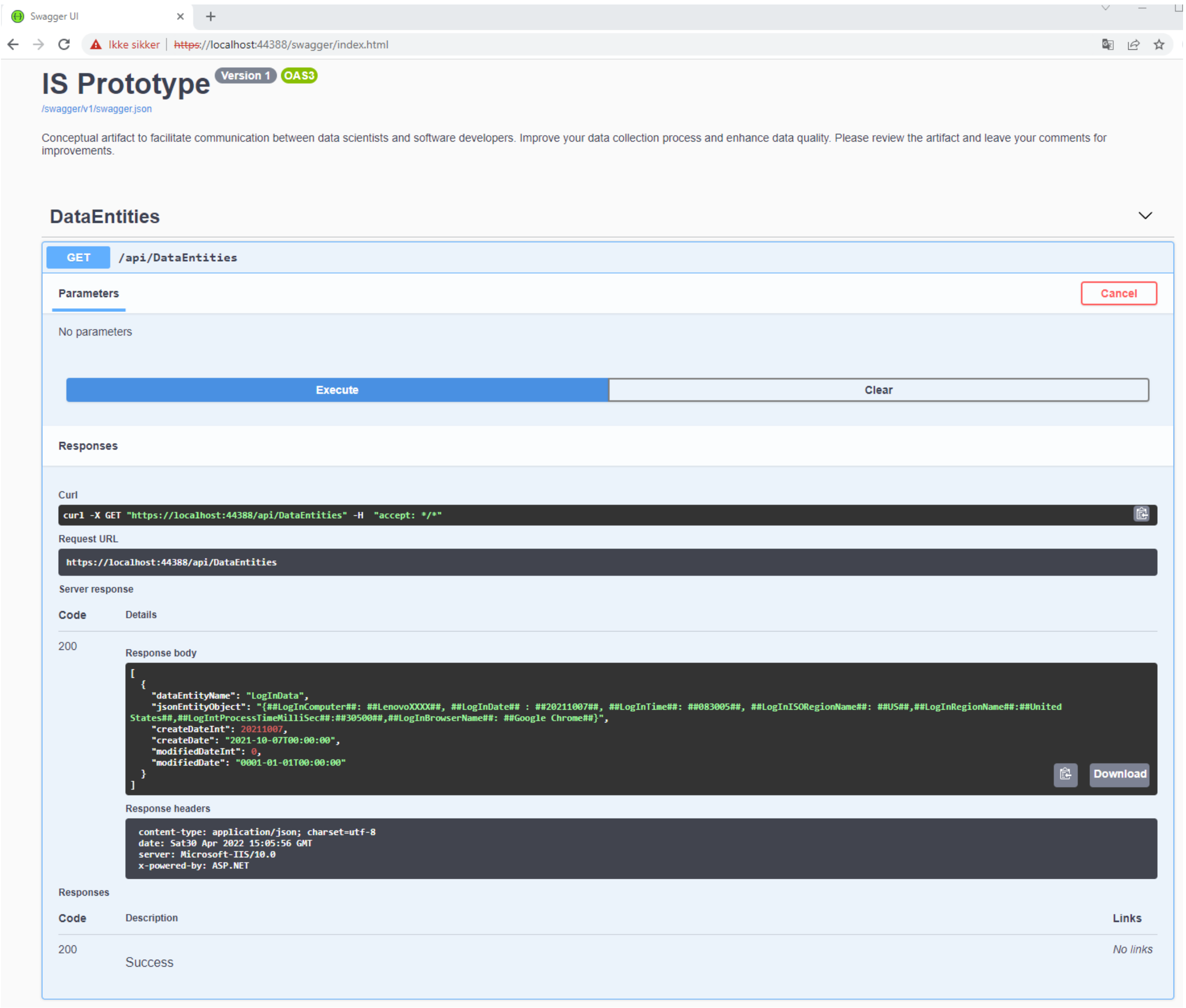
Request body: application/json

```
[
  {
    "dataEntityName": "LogInData",
    "jsonEntityObject": "{##LogInComputer##: ##LenovoXXXX##, ##LogInDate## : ##20211007##, ##LogInTime##: ##083005##, ##LogInISORegionName##: ##US##,##LogInRegionName##:##United States##,##LogIntProcessTimeMilliSec##:##30500##,##LogInBrowserName##: ##Google Chrome##}"

  }
]
```

Code	Description	Links
200	Success Media type: application/json Controls Accept header.	No links

View created “LogIn” data entity



IS Prototype Version 1 OAS3

/swagger/v1/swagger.json

Conceptual artifact to facilitate communication between data scientists and software developers. Improve your data collection process and enhance data quality. Please review the artifact and leave your comments for improvements.

DataEntities

GET /api/DataEntities

No parameters

Responses

Curl

```
curl -X GET "https://localhost:44388/api/DataEntities" -H  "accept: */*"
```

Request URL

```
https://localhost:44388/api/DataEntities
```

Server response

Code: 200

Response body

```
[
  {
    "dataEntityName": "LogInData",
    "jsonEntityObject": "{##LogInComputer##: ##LenovoXXXX##, ##LogInDate## : ##20211007##, ##LogInTime##: ##083005##, ##LogInISORegionName##: ##US##,##LogInRegionName##:##United
States##,##LogIntProcessTimeMilliSec##:##30500##,##LogInBrowserName##: ##Google Chrome##}",
    "createDateInt": 20211007,
    "createDate": "2021-10-07T00:00:00",
    "modifiedDateInt": 0,
    "modifiedDate": "0001-01-01T00:00:00"
  }
]
```

Response headers

```
content-type: application/json; charset=utf-8
date: Sat30 Apr 2022 15:05:56 GMT
server: Microsoft-IIS/10.0
x-powered-by: ASP.NET
```

Responses

Code	Description	Links
200	Success	No links

Appendix C

Consent form

Are you interested in taking part in the design research project:

"IS artifact embedded in an enterprise application to facilitate data collection adapted for machine learning and span software developers and data scientists expertise"

This is an inquiry to participate in a design research project where the main purpose is to gather information and feedback about a functional prototype embedded in an application to standardize data storage for a supervised machine learning model without affecting application's performance and study it's the potential impact on the organizational structure.

Purpose of the project

The aims of this project are:

1) To build a functional prototype embedded in an application for data collection

2) Create prototype presentations for AI professionals and software developers, receive feedback and evaluate the prototype

3) Conduct interviews with professionals researching the impact the prototype would have on the collaboration between software developers and data scientists

Who is responsible for the research project?

University of Oslo is the institution responsible for the project. The present study is part of Department of Informatics at University of Oslo.

Why are you being asked to participate?

TietoEvry company has data scientists and software developers which can evaluate the potential the prototype might have. Dhivya Gopalakrishnan is asked to participate as project manager.

What does participation involve for you?

Participation involves to engage data scientists and software developers to be interviewed and give feedback regarding the IS artifact.

Participation is voluntary

Participation in the project is voluntary. If you chose to participate, you can withdraw your consent at any time without giving a reason. All information about you will then be made anonymous. There will be no negative consequences for you if you chose not to participate or later decide to withdraw.

Your personal privacy – how we will store and use your personal data

We will only use your personal data for the purpose(s) specified in this information letter. We will process your personal data confidentially and in accordance with data protection legislation (the General Data Protection Regulation and Personal Data Act). To ensure confidentially and anonymity your name or identification will not be included in the interviews and presentations. University of Oslo (Norway) is responsible for the project, and only the project leader and the master student collecting data (University of Oslo) will have access to the personal data. This letter of informed consent will only be kept for as long as the study is in progress. All information data will only be analyzed for research purposes; all information will be anonymous and analyzed without any reference to your name. The results will be used in the students' project.

What will happen to your personal data at the end of the research project?

The project is scheduled to start in September 2021 and end in May 2022. Indirectly identifiable information will be stored by UiO to 2022 and be available for analysis and research purposes.

Your rights

So long as you can be identified in the collected data, you have the right to:

- access the personal data that is being processed about you
- request that your personal data is deleted
- request that incorrect personal data about you is corrected/rectified
- receive a copy of your personal data (data portability), and
- send a complaint to the Data Protection Officer or The Norwegian Data Protection Authority regarding the processing of your personal data

What gives us the right to process your personal data?

We will process your personal data based on your consent.

Based on an agreement with University of Oslo, NSD – The Norwegian Centre for Research Data AS has assessed that the processing of personal data in this project is in accordance with data protection legislation.

Where can I find out more?

If you have questions about the project, or want to exercise your rights, contact:

- Postdoctoral Fellow [redacted], University of Oslo, [redacted]

- Master student Viorica Fluer, University of Oslo, [redacted]
- NSD – The Norwegian Centre for Research Data AS, by email: (personverntjenester@nsd.no) or by telephone: +47 55 58 21 17.

Yours sincerely,

Viorica Fluer

Consent form

I have received and understood information about the project "*IS artifact embedded in an enterprise application to facilitate data collection adapted for machine learning and span software developers and data scientists expertise*" and have been given the opportunity to ask questions. I give consent:

□ to participate in interviews and feedback sessions

I give consent for my personal data to be processed until the end date of the project, approx. 15.08.2022

(Signed by participant, date)