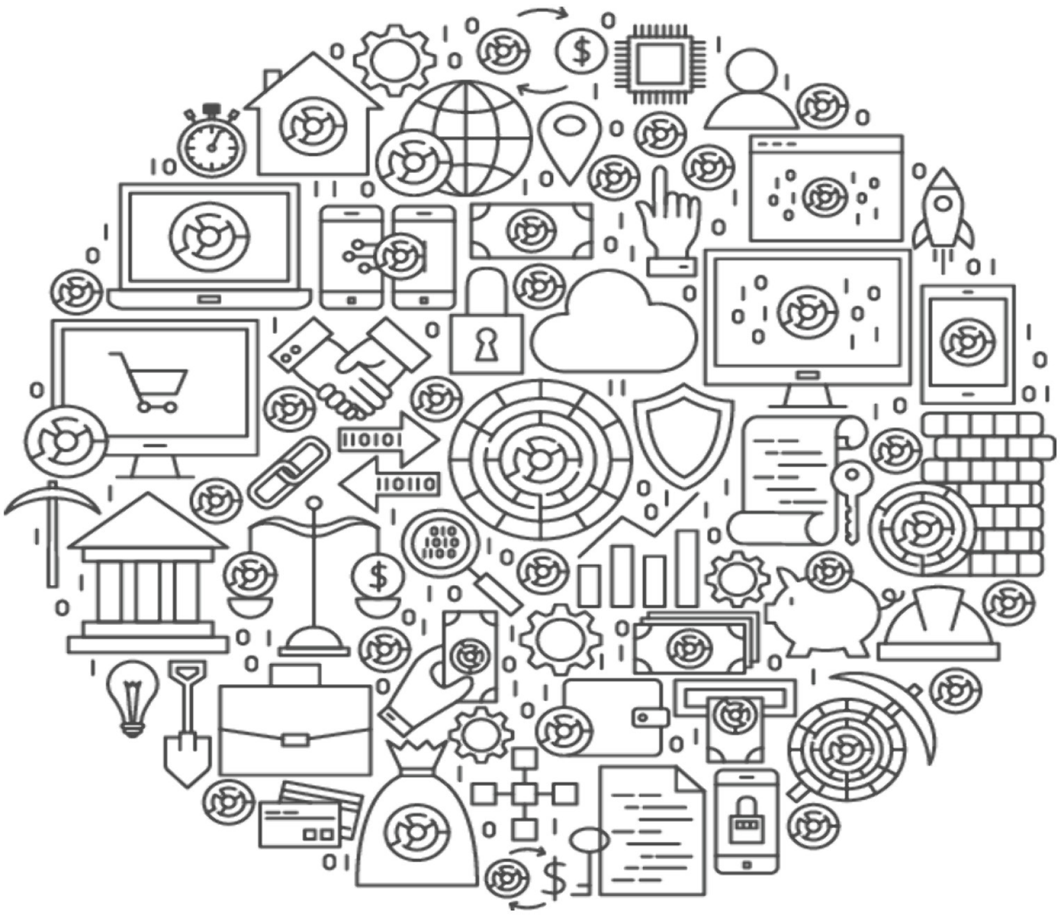
Determine the potential for digitization and harmonisation of administrative process

Deliverable 6: To-Be Situation Analysis

Technical Support Instrument

Supporting reforms in 27 Member States



Funded by the European Union

This document was produced with the financial assistance of the European Union. Its content is the sole responsibility of the author(s).

The views expressed herein can in no way be taken to reflect the official opinion of the European Union.

The project is funded by the European Union via the Technical Support Instrument, managed by the European Commission Directorate-General for Structural Reform Support (DG REFORM).

This report has been delivered in January 2025 un-der the EC Contract No. SRSS/SC2021/052. It has been produced as part of the project "Deter-mine the potential for digitization and harmoniza-tion of administrative processes".

Directorate-General for Structural Reform Support
REFORM@ec.europa.eu
+32 2 299 11 11 (Commission switchboard)
European Commission
Rue de la Loi 170 / Wetstraat 170
1049 Brussels, Belgium

Table of Contents

Table of Figures

Table of Tables

List of abbreviations

Abbreviation	
ADFS	Active Directory Federation Services
AI	Artificial intelligence
API	Application Programming Interface
AWS	Amazon Webservices
BERT	Bidirectional Encoder Representation of Transformers
BID	Drilling Identification
BPMN	Business Process Modeling Notation
BSW	*Behörde für Stadtentwicklung und Wohnen*
BUKEA	*Behörde für Umwelt, Klima, Energie und Agrarwirtschaft*
CaaS	Container as a Service
CI	Continuous Integration
DeBERTA	*Decoding-enhanced BERT with Disentangled Attention*
DEV / OPS	A compound of development (Dev) and operations (Ops), DevOps is the union of people, process, and technology to continually provide value to customers.
DVC	Data Version Control
EDA	Exporatory Data Analysis
GLA	*Geologisches Landesamt* (Geological State Office)
HmbDSG	Hamburgisches Datenschutzgesetz
IaaS	Infrastructure as a Service
IDM	Intelligent Dialogue Management
MICE	Multivariate Imputation by Chained Equations
ML	Machine Learning
MLOps	Machine learning operations (MLOps) is the use of machine learning models by DevOps Teams
NOBO	Norddeutsche Bohranzeige
RAM	Random Access Memory
SDK	Software Development Kit
SKA	Brief written inquiries (*Schriftliche Kleine Anfragen*)
SOTA	State of the art
SQL	Structured Query Language
SSO	Single Sign On
SSOT	Single Source Of Truth
vRAM	Video RAM

ZSL	Zero Shot Learning

To-be situation analysis

This Deliverable presents the to-be situation analysis of the five selected to-be processes, thereby focussing on the legal, technical and organisational implications. It further entails the IT architecture for each to-be process including a description of the respective modules the IT architecture contains.

Executive Summary

This deliverable outlines the **to-be situation analysis for the five selected to-be processes** brief written inquiries (Schriftliche Kleine Anfragen), senate printed matter coordination (Senatsdrucksachenabstimmung), the imputing procedure (BohrIS process), info boxes (Infoboxen) and the knowledge management process (Wissensmanagement) for the respective selected alternatives as of deliverable 5 (Business Case).

This entails an overarching description of the technical, legal, organisational and operational implications of the different technologies that would be implemented. Furthermore, the overarching conceptual considerations for the technological IT architecture will be outlined.

Thereafter, for each to-be process, the foreseen IT infrastructure and its components will be described and analysed in detail.

The detailed analysis regarding specific legal, organisational and technical implications for the five selected to-be processes shows that the **four processes** brief written inquiries (Schriftliche Kleine Anfragen), senate printed matter coordination (Senatsdrucksachenabstimmung), the imputing procedure (BohrIS process) and the knowledge management process (Wissensmanagement) **have good preconditions to implement the respective to-be models**. This entails the implementation of the IDM workflow management tool (or, in the case of the imputing procedure BohrIS, the "Modul F") with several core functions and, depending on the process, several microservices, e.g., interfaces to existing databases and/or solutions and features such as the completeness check, plausibility check, intelligent search and more.

Given the fact that **for the process info boxes**, no data is currently stored, the foreseen assignment of responsibilities with a supporting AI forwarding assistant cannot be trained with state of the art (SOTA) models yet. At the model and data level, this function, could be trained with traditional natural language processing algorithms or few shot learning[1] algorithms like SetFit[2] with multiclass or multilabel predictions. The **data necessary to create the assignment of responsibilities (assisted)** is, however, **not sufficient to train the SOTA models with a fine-tuning approach like DeBERTa**[3]. Hence, in this case, incoming data (emails from external stakeholders) must first be collected and stored before conducting the data analysis and the deployment and training of the foreseen AI forwarding assistant. A phased approach could be taken, implementing the IDM tool first and after sufficient data has been collected, the assignment of responsibilities (assisted).

[1] https://www.analyticsvidhya.com/blog/2021/05/an-introduction-to-few-shot-learning/
[2] https://arxiv.org/abs/2209.11055 - Efficient Few-Shot Learning Without Prompts
[3] https://arxiv.org/abs/2006.03654 - DeBERTa: Decoding-enhanced BERT with Disentangled Attention

Introduction

This deliverable outlines the to-be situation analysis for the five selected to-be processes brief written inquiries (Schriftliche Kleine Anfragen), senate printed matter coordination (Senatsdrucksachenabstimmung), the imputing procedure (BohrIS process), info boxes (Infoboxen) and the knowledge management process (Wissensmanagement) for the respective selected alternatives as of deliverable 5 (Business Case).

This entails an overarching description of the technical, legal, organisational and operational implications of the different technologies that would be implemented. Furthermore, the overarching conceptual considerations for the technological IT architecture will be outlined.

Thereafter, for each to-be process, the foreseen IT infrastructure and its components will be described and analysed in detail.

Please note that all described to-be models in this deliverable are considered to be suggestions. Before the implementation of the five to-be process models, further alignments and consultations should be made with the process owners, although they are not authorized to make the final decisions regarding the to-be processes.

To-be situation: overarching implications and aspects

The to-be situation that would foresee an implementation of the five to-be processes entail different implications at a technical, legal, organisational and operational level.

Technical implications refer to those aspects that need to be changed/adapted or introduced from a technical point of view to the existing processes, **legal implications** outline the consequences that need to be addressed from a legal point of view. Regarding legal aspects, the Hamburg Data Protection Act ("Hamburgisches Datenschutzgesetz (HmbDSG) of May 18, 2018, in particular the second section thereof (§§ 4 – 8) on the principles of the processing of personal data must be considered. It is further recommended to involve the Office of the Hamburg Commissioner for Data Protection and Freedom of Information (HmbBfDI) when implementing the to-be processes. This office, including the data protection officers at BSW and BUKEA, advises citizens, authorities and companies in the city on all questions on data protection, support them in exercising their rights to access information and helps those affected to enforce their rights and monitors the administrations in Hamburg. **Operational implications** directly result from the implementation of the respective to-be process, i.e., new/adapted process steps, additional efforts etc. at the process level. These implications per to-be process will be further detailed in Deliverable 7 (To-be process models in BPMN format). **Organisational implications** refer to those implications that result from the operational to-be situation at the organisational level.

Overview of overarching implications of the to-be technologies

These overarching implications are summarised in the table below, thereby structured along the technologies to be introduced. The use of the respective technologies in the individual to-be processes is described in the following chapters.

Table 1: Overarching technical, legal, organizational and operational implications of the to-be technologies

Technology	Technical implications	Legal implications	Organisational implications	Operational implications
Workflow management (IDM tool) and correspondin g features/tools	• The workflow of each new case within every to-be process is controlled by an frontend that can be accessed from the browser, it is no longer necessary to use the Office Suite, but you can use Office applications.	• Restricted access to several functions of the dashboard, depending on the process and person, must be considered. • Data protection regulations must be complied with, in particular when dealing with information/data	• Given the experience with IDM and the initial development of the solution, it can be expected that the introduction of this IDM tool has a low threshold and can easily be implemented in other departments/authorities and expanded to other use cases	• For users/process owners, a central platform where they can manage the steps of the process enhances transparency and communication and ensures compactness • The platform facilitates/supports various process steps with its features (e.g.,

Technology	Technical implications	Legal implications	Organisational implications	Operational implications
		from external stakeholders • When archiving data and requests, data protection regulations must be complied with, e.g. duration of archiving or blackening of personal information.	• Communication improvements are expected given the use of a central platform • Reduction of media breaks given the bundling of communication/information and data exchange in one workflow management • Positive synergy effects are expected to increase the more interfaces are connected and the more authorities/departments/units use this tool.	deadline tracking, control/monitoring dashboard) and thereby ensures a clear overview.
Assisted assignment of responsibilities	• The assignment of responsibilities is supported by suggestions from an artificial intelligence (AI) or machine learning (ML) model. The final assignment is still in the power of a human. The AI only supports the assignment with proposals. • The utilized AI or ML models will improve in performance over time based on the increasing number data.	• When forwarding data/information, data protection regulation must be complied with, in particular concerning sensitive and personal information.	• Given the experience in the assignment of responsibilities within the process 'citizen letters', synergies regarding the implementation and adaption to other processes as well as the mitigation of technological challenges can be generated. • Depending on the availability and quality of data, further assessments regarding the applicability of this AI solution to other processes must be determined. This also includes the adaption and training of the AI model.	• For process owners, an assisted assignment of responsibilities can reduce time efforts needed to allocate an email/request/… to the responsible person. • In time-critical situations, a traceable and quick allocation of assignment can be ensured.
Labelling	• Consistent and correct labelling of the training data is key for yielding performant models, therefore the labelling takes place in a prepared environment by experts of the department. • A labelling tool will be implemented for labelling data for all use-cases.	• Procedure regarding labelling of sensitive information/data must be established in accordance with data protection regulation. • Restricted access procedure as well as storage of documents to train AI model must be agreed on with involved departments/units and the data protection.	• Cross-authority and cross-department use is possible once the labelling process is implemented given the low thresholds to introduce the labelling to other departments/units. • Even higher positive synergies can be generated if the labelled data is stored centrally (complying with certain security and data protection restrictions accordingly) and the more data will be labelled. • Labelling also improves intelligent searches and assignment of responsibilities.	• Initial time efforts at the time of implementation are expected given the need to label existing data – depending on the amount of data to be labelled. • Additional recurring efforts when storing new data/information in the databases occur (employees must label documents accordingly). • Additional one-off coordination efforts occur as departments/units must agree on the procedure

Technology	Technical implications	Legal implications	Organisational implications	Operational implications
				of how to label (e.g. terminology).
Intelligent search	• The Intelligent search will serve as an individual search engine not mandatory coupled to the process. Staff is free to use the search to fulfil a task. The intelligent search would allow for facilitated searches over the historic database, and thus simplify and accelerate the process, which would be advantageous in time efficiency and output consistency. • The intelligent search will be executed on a specialized database of all historic data.	• Restricted access to protected documents and information must be ensured. • Compliance with data protection regulation must be ensured, in particular concerning sensitive and personal information. • The need-to-know principle should be followed, i.e. sensitive information should only be accessed by those who have been authorized accordingly.	• Given the existence of pre-trained models, comparably large and quick positive synergy effects can be expected. • Once the model is trained, it can be expanded to other departments/authorities, thereby also benefitting from the fact that the respective departments within BUKEA and BSW work with similar topics which facilitates a genre-specific training of the AI model. • As a consequence, and given the fact that searches are a part of almost every employee at BUKEA and BSW, a reduction in workload and an improvement of quality can be expected.	• Given a more precise search due to a better coverage of available information, improved and faster search results can be expected. • This, in turn, could significantly reduce the workload of those employees using an intelligent search function.
Imputing	• Some data fields can be imputed by calculating statistical figures, thereby selecting the most probable category. Other data fields only allow this deduction by applying more sophisticated machine learning imputation algorithms.	• Accuracy of data must be verified/checked manually. • Sensitive information and personal data cannot be imputed.	• First experiences of the imputing process can be gained and can be transferred to other tabular data and processes outside the BohrIS process.	• Reduces time efforts resulting from tracing loops for units that need certain data e.g. from external stakeholders or other internal units. • In turn, employees can allocate their resources to content-related (instead of administrative) activities.
Completeness Check	• Instead of manually tracking missing fields of information, the completeness check automatically checks for missing relevant fields.	• The accuracy of the check itself must be verified and checked manually.	• Could contribute to yield a reduction in administrative workload and thereby disburden employees from time-consuming manual data reconciliations. • This, in turn, frees up capacity for technical reviews and content-related work.	• Reduces the manual time that was required to double check for missing fields. • The completeness check must be reviewed manually to ensure that the completeness checks are current and accurate.
Plausibility Check	• Instead of checking the values manually, the plausibility check reads the data and ensures that the	• The accuracy of the completeness check needs to be manually checked and inspected.	• Could contribute to yield a reduction in administrative workload and thereby disburden employees from	• Minimizes the time spent checking the plausibility of the data entered. • Checks for plausibility must be checked to

Technology	Technical implications	Legal implications	Organisational implications	Operational implications
	values entered are the expected ones. • Ranging from simple type checks to statistically.		time-consuming manual data reconciliations. • This, in turn, frees up capacity for technical reviews and content-related work.	ensure that they are up to date and correct.

Source: Deloitte (2022)

Overarching conceptual considerations

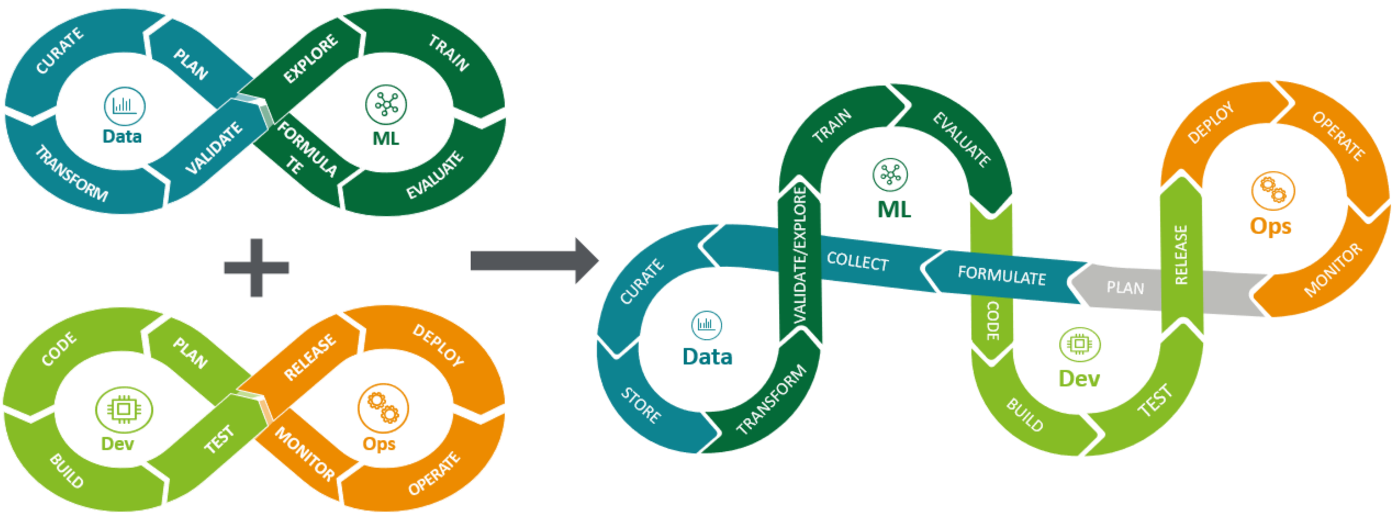
This subchapter outlines the overarching conceptual considerations that are relevant in the context of the implementation of the to-be processes.

The double infinite loop and the concepts of model-centric vs. data-centric

In particular, the **concepts of a model-centric vs a data-centric approach** will be explained and how these approaches relate to alternative 2 and 3 (see Deliverable 5 – Business Case).

The following figure provides an overview of the double infinite loop concept that combines the DevOps loop and the DataML loop. This concept is preferred for all software projects with a machine learning or deep learning component because it allows partitioners to extract the most value from the available data and has the highest probability of success.

Figure 1: Double infinite loop Machine Learning Operations (MLOps), a combination of Data and Machine Learning (ML) and Development and Operations



Source: https://www.ml4devs.com/articles/mlops-machine-learning-life-cycle/

Project implementations with machine learning components **should be implemented with iterative cycles**, although traditional software projects also support this approach.

Since the to-be processes brief written inquiries (Schriftliche Kleine Anfragen), imputing procedure (BohrIS process), info boxes (Infoboxen) and knowledge management process (Wissensmanagement) (**processes** for which **alternative 3** was determined as preferred alternative – see Deliverable 5 Business Case) include

either a machine learning or a deep learning component, the **approach using an iterative life cycle should be followed** for the overarching process architecture of these to-be processes. Hence, the DEV / OPS[4] infinite loop should be used for those shortlist processes where alternative 2 is selected (without machine learning or deep learning components) – the senate printed matter coordination (Senatsdrucksachenabstimmung). For all other processes with alternative 3 as preferred alternative, the process architecture should be extended to the double infinite loop with DEV / OPS & DATA / ML (MLOPs)[5].

This **iterative life cycle approach is referred to as data-centric approach**, as great importance is attached to the collection and quality of the data to improve the machine learning models. In contrast, the **model-centric approach** tries to improve the machine learning models by changing the model itself, for example by hyperparameter tuning. Deeplearning.ai[6], an initiative started by Stanford professor Andrew NG[7], shows in different conferences indications for the **superiority of the data-centric approach over the model-centric approach in terms of performance improvement**.

The table below provides a summary of the performance change over specific tasks. The Zero-Shot-Learning (ZSL) Text-Classifier[8], matches in large parts our assignment of responsibilities (assisted) functionality, integrated in brief written inquiries and info boxes to-be processes within alternative 3 and shows the advantage over a model-centric approach.

Table 2: Overview of the performance change for specific tasks with various approaches

Approach	Steal defect detection	Solar Panel	ZSL Text-Classifier
Base Model	76,2%*	75,68%*	67%*
Model-centric approach	+0% (76,2%)	+0,04% (75,72%)	+0% (67%)
Data-centric approach	+16,9% (93,1%)	+3,06% (78,74%)	+15% (82%)

Source: Data-centric AI conference hosted by Deeplearning.ai[9], own illustration
*Base Performance Metric e.g. Accuracy[10]

Process steps of the double infinite loop in detail

The individual process architecture steps are explained below using the double infinite loop.

This Data / ML infinite loop is not linear. At every stage, you don't always move forward to the next stage. Upon discovering problems, you go back to the relevant previous stage to fit them. Hence, there are implicit edges from each stage to previous stages.

This is similar to the DEV / OPS loop which developers follow. Not every code that goes to the test stage progresses to the release stage. If the tests fail, it goes back to the code (sometimes even to plan) stage for

[4] https://en.wikipedia.org/wiki/DevOps - DevOps is a set of practices that combines software development and IT operations. It aims to shorten the systems development life cycle and provide continuous delivery with high software quality.
[5] https://en.wikipedia.org/wiki/MLOps - MLOps or ML Ops is a set of practices that aims to deploy and maintain machine learning models in production reliably and efficiently. The word is a compound of "machine learning" and the continuous development practice of DevOps in the software field.
[6] https://www.deeplearning.ai/
[7] https://hai.stanford.edu/people/andrew-ng
[8] https://arxiv.org/abs/1707.00600 - Zero-Shot Learning -- A Comprehensive Evaluation of the Good, the Bad and the Ugly
[9] See the following videos of the conference for more information: https://www.youtube.com/watch?v=06-AZXmwHjo&t=400s, https://www.youtube.com/watch?v=Yqj7Kyjznh4&t=3340s
[10] https://en.wikipedia.org/wiki/Accuracy_and_precision

problems to be rectified. Therefore, **the whole planning** cannot be finalized upfront in a separate digitization project but **must be adapted in an agile manner within an implementation project.**

Planning

The first phase for each to-be process is **planning**. Discussing the business goals and key business metrics, as well as the product features that can help achieve those goals stands in the center of this stage. Drilling down the end-user problems and debate about user journeys to address those problems and collect required data to assess how a new digital process is changing and improving things. This phase is already covered to a large extent, but not in full, by the deliverables 6 (to-be-analysis), deliverables 7 (to-be process models in BPMN) and deliverable 8 (roadmap) of the Deloitte project REFORM SC2021/064.

Formulate

The second phase is called **formulate**, which is also already covered to a large extent by the project REFORM SC2021/064 with City of Hamburg and Deloitte. This phase starts the Data / ML loop of the to-be process. Data scientists translate a business objective into a machine learning problem. There are several factors that you may need to consider:

- **Business Objective (covered by the project REFORM SC2021/064):** Narrow it down to a small set of machine learning problems that can serve the business objective
- **Cost of Mistakes (to be covered upon implementation of the to-be processes)**: No machine learning model will be 100% accurate. What is the cost of false negatives and false positives? For example, if an image classification model wrongly predicts a lethal disease in a healthy person, further tests will rectify it. But if the model fails to diagnose this disease in a patient, then it can turn out to be fatal due to late detection.
- **Data Availability (partly covered by the Deloitte project REFORM SC2021/064):** It may come as a surprise, but you may start with no or very few data. As the data becomes richer (into later cycles), it may make more types of models viable. For example, if you were to do anomaly detection[11] with no labeled data, you may start with various kinds of unsupervised clustering algorithms[12] and mark points that are not in any cluster as anomalies. But as you collect user reactions to your model, you will have a labeled dataset. Then you may want to try if a supervised classification[13] model will perform better. Another example, with respect to our functionality assignment of responsibilities (assisted), would be to first use few-shot-learning algorithms (will be discussed in more detail in the individual chapters of the shortlist processes) and after collecting enough data in the following cycles switching to a traditional fine-tuning[14] method.
- **Evaluation Metrics (covered by the Deloitte project REFORM SC2021/064):** Depending upon problem formulation, specifying a model performance metric to optimize for, which should align with the business metric for your business objective and with the data distribution of the data is important. For example, evaluating the assignment of responsibilities (assisted) with an accuracy[15] metric, will not lead to satisfactory models, if the data distribution contains many underrepresented district offices and departments. This is called imbalance datasets[16] in machine learning and is conquered by evaluation metrics like F1-Score[17] or Matthews Correlation Coefficient[18].

[11] http://scikit-learn.org/stable/modules/outlier_detection.html
[12] https://scikit-learn.org/stable/modules/clustering.html#overview-of-clustering-method
[13] https://en.wikipedia.org/wiki/Statistical_classification
[14] https://arxiv.org/abs/1801.06146 - Universal Language Model Fine-tuning for Text Classification
[15] https://en.wikipedia.org/wiki/Accuracy_and_precision#In_classification
[16] https://www.analyticsvidhya.com/blog/2021/06/5-techniques-to-handle-imbalanced-data-for-a-classification-problem/
[17] https://en.wikipedia.org/wiki/F-score
[18] https://en.wikipedia.org/wiki/Phi_coefficient

For both phases planning and formulate full coverage will only be achieved upon the first implementation cycle, not within this project REFORM SC2021/064, which should give a decision template which short list process should be implemented first.

Collect

The **collect** phase within the Data / ML subloop covers the necessary data collection from internal applications as well as external sources. It may be by scraping[19] the web, capturing event streams from mobile apps or web service, or simply collecting data from windows folder structures. This stage invokes data versioning tools like DVC[20].

Curate / Store

After collecting the data, we enter the **curate** stage. The data collected is almost never pristine. You need clean it, fill in missing values, remove duplicates, and **store** it in a data warehouse, data lake or other suitable places. If a supervised learning task is contemplated for the to be model the data has to be annotated and in machine learning language this is called labelling. Some tasks like simple text classification tasks, e.g. assignment of responsibilities (assisted), can be labelled with Excel, but it is beneficial to implement a central labelling and annotation environment, which can be used across all projects where machine learning or deep learning is involved and leverage synergies. With professional labelling environments like Label Studio[21] and Kerni.ai[22] the workflow of labelling gets more efficient, and the data is more reliable and of higher quality because they enable control mechanisms within the labelling process like review loops or annotating one example by more than one labeler at the same time to measure inter rater reliability[23]. Also, cataloging the data, so that it can be easily discovered and correctly understood is advantageous.

Transform

The curate stage is followed by the **transform** phase. Once data has been cleaned, and labelled a transformation is needed to suit the analytics and machine learning modeling. It may require changing the structure, joining with other tables, aggregating or summarizing along important dimensions (dimensionality reduction[24]), computing additional features[25], etc. Data Engineers will automate all of it in the data pipeline with tools like pandas [26] and scikit-learn[27] preferred in a python[28] programming environment.

Validation / Explore

The **validation** stage implements quality checks, maintain logs of statistical distributions from the data over time, and create triggers to alert when any of the checks fail or the distribution sways beyond expected limits. Data Engineers in consultation with data scientists implement these validations in the data pipeline with tools like Pytest[29] and Pandera[30]. The downstream step **Explore** is an important step in machine learning pipelines. Especially in cases without deep learning algorithms involved like in our Imputing shortlist process. Data scientists perform Exploratory Data Analysis (EDA)[31] to understand the relationships between various features and the target value they want the model to predict. They also do Feature Engineering[32], which is likely to lead to adding more curation and transformation (the previous two stages). This stage invokes the use of various python libraries like Pandas Profiling [33]or Sweetviz[34] for running a

[19] Web scraping is the process of using computer programs to extract content and data from a website.
[20] https://dvc.org/
[21] https://labelstud.io/
[22] https://www.kern.ai/
[23] https://en.wikipedia.org/wiki/Inter-rater_reliability
[24] https://en.wikipedia.org/wiki/Dimensionality_reduction
[25] https://en.wikipedia.org/wiki/Feature_engineering
[26] https://pandas.pydata.org/
[27] https://scikit-learn.org/stable/
[28] https://www.python.org/
[29] https://docs.pytest.org/en/
[30] https://pandera.readthedocs.io/en/stable/
[31] https://en.wikipedia.org/wiki/Exploratory_data_analysis
[32] https://en.wikipedia.org/wiki/Feature_engineering
[33] https://pandas-profiling.ydata.ai/docs/master/index.html
[34] https://github.com/fbdesignpro/sweetviz

comprehensive analysis of every variable in the data. There are also special toolkits for feature selection and feature engineering. The library getML[35] offers the functionality of an automated feature selection as well as feature engineering, which considerably reduces the effort of the datascientist for this part of the analysis.

Train

The **train** stage is straightforward. Data scientists run multiple experiments (i.e. train the models), compare model performance, tune hyper-parameters, and select a couple of best-performing models.

Evaluate

Afterwards we enter the final DATA / ML infinite loop stage called **evaluate**. Which is evaluating the model characteristics against business objectives and metrics. Some feedback may result in even tweaking and formulating the ML problem differently and repeating the whole subprocess loop all over again.

Code

The **code** stage is the first DEV / OPS subloop part in the process architecture. This phase is for designing and developing the software or application. Especially frontend and backend functions are implemented. User reactions feedback and insights of the evaluation phase will be integrated. It is very important to get developers, data engineers, and data scientists on the same page regarding the coding. This stage invokes software version control systems like git[36] as well as agile workflow tools like Jira[37] or Gitlab[38] to control the workflow in combination of agile management like scrum[39].

Build

The **build** stage is the next downstream step. This stage fuels the continuous integration (CI)[40] of various parts as they evolve and package into a form that will be released. It can be a library or a software development kit (SDK)[41], a docker image[42], a web service, an API[43] or an application binary. In our case it will most likely be a docker image, a web service and an API.

Test

In the **test** phase unit tests, integration tests, coverage tests, performance tests, load tests, privacy tests, security tests, and bias tests are taking place. Testing should be done on a staging environment[44] that is similar to the targeted production environment but not designed for a similar scale. It may have dummy, artificial, or anonymized data to test the software system end-to-end. For general tests the Python internal test library Unittest[45] can be used. Alternatively, there are also external libraries such as Pytest[46], which can also be used to perform comparable tests with much less code. It is also possible to have these tests automated by tools. For example, with Smartbear[47] it is possible to thoroughly test scripts of different

[35] https://getml.com/
[36] https://git-scm.com/
[37] https://www.atlassian.com/software/jira
[38] https://about.gitlab.com/
[39] https://www.scrum.org/resources/what-is-scrum
[40] https://en.wikipedia.org/wiki/Continuous_integration
[41] https://en.wikipedia.org/wiki/Software_development_kit
[42] https://ieeexplore.ieee.org/document/7883438 - A Docker image is a file used to execute code in a Docker container. Docker images act as a set of instructions to build a Docker container, like a template. Docker images also act as the starting point when using Docker. An image is comparable to a snapshot in virtual machine (VM) environments. A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.
[43] https://en.wikipedia.org/wiki/API
[44] https://www.techtarget.com/searchsoftwarequality/definition/staging-environment
[45] https://docs.python.org/3/library/unittest.html
[46] https://docs.pytest.org/en/7.1.x/
[47] https://smartbear.com/

programming languages without manual coding. In addition, automated repository tests are feasible with the software docker[48], which can be connected to the continuous integration pipeline.

Release

The Test stage is followed by the **release** phase. Once all automated tests pass and, in some cases, test results are manually inspected, the software code or models are approved for release. Just like code, models should also be versioned and necessary metadata automatically captured. Just as the docker images are versioned in a docker repository, the model should also be persisted in a model repository. If models are packaged along with the code for the microservice that serves the model, then the docker image has the model image too. This is where Continuous Integration ends, and Continuous Deployment takes over[49]. Continuous deployment can be integrated using various tools, based on the decision of whether or not to self-host the pipeline. For example, the previously mentioned docker framework is well suited for outsourced hosting of the pipeline. For self-hosted solutions, Rancher[50] or Kubernetes[51] can be a suitable alternative.

Deploy

The next phase called **deploy** is picking the released artifacts from the docker repository or model store and deploying it on production infrastructure. Depending on the need, different infrastructures e.g. infrastructure as a service (IaaS)[52], container as a service (CaaS)[53], platform as a service (PaaS)[54] or on-premise[55] can be chosen.

The shortlist processes in our case would encourage a container as a service (CaaS) or On-Premise structure. Whereby the CaaS implementation brings some advantages with it, such as scalability, faster provision of the infrastructure and high transparency of the costs through a utilization-based billing model. This stage invokes software for managing containers such as docker. Choosing CaaS implies the need of a service provider, of which there are many. One possibility is amazon EC2 container service[56], which is particularly suitable if other AWS services are also used. Comparable services are also available from all common competitors of AWS, with Google Kubernetes Engine[57] or Azure Kubernetes Service[58]. Docker itself also offers CaaS and furthermore can also be used with on-premise solutions.

Operate

After deployment we enter the **operate** stage. Once the services are deployed, one could decide to send a small percentage of the traffic first. Canary deployment[59] is common tactic to update in phases (e.g. only 25% of the processes will be handled with the new digitized form, in the next step 50% and so on). In case of a problem, unexpected behavior, or a drop in metrics, you can roll back to the old process. The canary deployment is not mandatory.

---

48 https://docs.docker.com/
49 https://www.xenonstack.com/insights/continuous-integration-vs-continuous-deployment
50 https://www.rancher.com/ - Rancher is an open source software platform that enables organizations to run containers in production. With Rancher, organizations no longer have to build a container services platform from scratch using a distinct set of open source technologies.
51 https://kubernetes.io/ - Kubernetes is an open-source container orchestration system for automating software deployment, scaling, and management. Google originally designed Kubernetes, but the Cloud Native Computing Foundation now maintains the project.
52 https://en.wikipedia.org/wiki/Infrastructure_as_a_service
53 https://en.wikipedia.org/wiki/Content_as_a_service
54 https://en.wikipedia.org/wiki/Platform_as_a_service
55 https://en.wikipedia.org/wiki/On_Premises
56 https://aws.amazon.com/blogs/aws/cloud-container-management/
57 https://cloud.google.com/kubernetes-engine/
58 https://learn.microsoft.com/en-us/azure/aks/intro-kubernetes
59 https://semaphoreci.com/blog/what-is-canary-deployment

Monitor

The last stage and circular connection to start a new cycle with a new planning phase is the Monitor stage. This involves an automatic and constant monitoring of the health of services, errors, latencies, model predictions, outliers and distribution of input model features, collect user feedback etc. In case a problem arises, depending upon the severity and diagnosis, you may roll back the system to an older version, release a hotfix, trigger model re-training, or carry out any other actions that may be required. User feedback and other monitoring insights should be used in the next cycle planning phase to improve the model. Monitoring container-based infrastructures is most suitable in the environment of the service provider of the container service. For example, when using the AWS EC2 container service, the application of AWS container monitoring[60] is best suited. Other alternatives may be Google cloud monitoring[61] or Azure monitor log analytics[62], for instance.

Prevailing potential for synergies

We expect synergy effects to be particularly high in the following three areas:

- On the one hand, the implementation of the IDM tool becomes significantly faster due to high similarity in follow-up implementation projects, in which the IDM tool is also set up.
- Furthermore, after legal examination, data can be shared among the individual projects for the pre-training of models. This approach improves the foundation of the Deep Learning models used, whereby anonymization of personal identifiable information (PII)[63] can be applied using tools such as presidio[64] to comply with data protection guidelines. The merging of data would be applied, for example, to the functionalities assisted assignment of responsibilities for the processes brief written inquiries, senate printed matter coordination and info boxes. For the functionalities intelligent search, the data of the processes brief written inquiries and knowledge management could share their data.
- The usage of an annotation or labelling environment e.g., Label Studio[65] offers the most synergies, because a large part of machine-learning and deep-learning applications fall into the group of supervised learning[66] and require labels or annotations. Whether a computer vision[67], natural language processing (NLP)[68], or tabular problem is to be solved using artificial intelligence, such an annotation environment will always be used when the annotation complexity exceeds simple classifiers[69] or regressions[70].

[60] https://aws.amazon.com/cloudwatch/container-monitoring/
[61] https://cloud.google.com/monitoring/
[62] https://learn.microsoft.com/en-us/azure/container-apps/log-monitoring
[63] https://gdpr.eu/eu-gdpr-personal-data/
[64] https://github.com/microsoft/presidio
[65] https://labelstud.io/
[66] https://en.wikipedia.org/wiki/Supervised_learning
[67] https://en.wikipedia.org/wiki/Computer_vision
[68] https://en.wikipedia.org/wiki/Natural_language_processing
[69] https://en.wikipedia.org/wiki/Statistical_classification
[70] https://en.wikipedia.org/wiki/Regression_analysis

Brief written inquiries (Schriftliche Kleine Anfragen, SKA)

This chapter describes the technical to-be situation and its underlying components as well as their technical implications for the to-be process brief written inquiries (Schriftliche Kleine Anfragen).

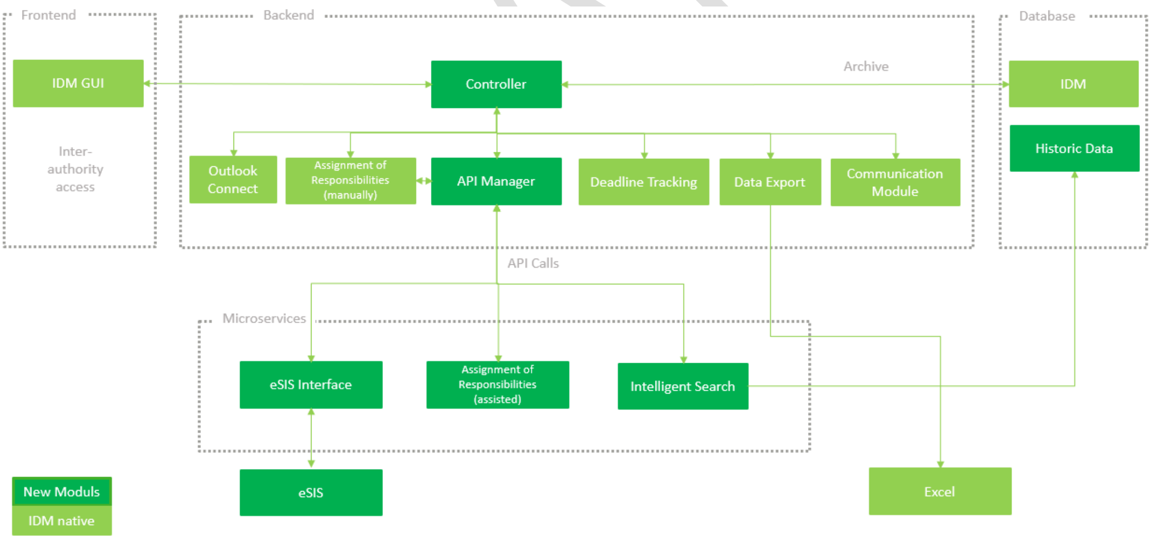
For the brief written inquiries, the preferred alternative that resulted from the Business Case analysis (see Deliverable 5) is alternative 3. This alternative foresees that the process is restructured with the help of an already existing process management tool, the IDM tool (Intelligent Dialogue Management), whereby the tool is equipped with further functionalities, including artificial intelligence. The following functionalities have been agreed upon:

- A **central access channel (dashboard)**
- **Assignment of responsibilities (manually)** and
- **Assignment of responsibilities (assisted**; supported by a AI forwarding assistant)
- **Archiving** of the inquiries and the process and response histories in a central database
- **Monitoring** and
- **Deadline tracking**
- **Comment and communication** functions
- **Inter-authority access**
- **Data export**
- **Intelligent search**
- **eSIS Interface**

To-be infrastructure

The following figure provides an overview of the to-be technical infrastructure for the process brief written inquiries.

Figure 2: To-be infrastructure for the process brief written inquiries



Source: Deloitte (2022)

The existing IDM tool is primarily extended by a controller, which manages the various existing and new functionalities. The database of the IDM tool, an adapted version of the IDM frontend, the manual assignment of responsibility module, the deadline tracking module, the data export module, and the communication module will remain and are called main functions.

IDM main functions

There are no **model and data levels** and no **dataset** layer is involved for the IDM main functionalities because the main functions do not include artificial intelligence.

On the **hardware layer** side the IDM backend and frontend can be served with a 2 Core CPU with 32 GB of RAM and a SSD hard drive with 8 TB of space and a redundant array of independent disks RAID level of 1.

The **software layer** includes a Windows or Ubuntu operating system with a Nginx[71] web server to server the IDM tool. The Single-Sign-On (SSO)[72] functionality is severed with the Active Directory Federation Services (ADFS)[73] with additional active directory processes for nightly updates regarding extended user data of the employees. The frontend is developed with Vue.js[74].

At the **staffing level,** the following profiles are needed (they are not disjunct, so one person can unite different profiles):

- Special Matter Expert: Is a person who has accumulated knowledge in a particular fields or topics of interest (e.g. brief written inquiries) and is familiar with the process and information processed and are responsible for labeling the data and giving business feedback
- Frontend/backend Developer: Is usually responsible for writing APIs and the necessary extensions in the IDM front- and backend. The frontend developer needs to have JavaScript[75] especially Vue.js skills.

At **data architecture level** a PostgreSQL 13[76] database is used and will be further extended.

Newly provided functionalities will be the **inter-authority access** for the IDM frontend, which takes over user authorizations and access rights management. Furthermore, a newly introduced **API manager** is controlled by the **backend controller**, which ensures that all AI applications and interfaces to external tools such as eSIS can be provided. This backend controller should preferably be implemented in the backend language of choice, for example in Python. The API Manager can either be developed in-house to be perfectly integrated into the environment, or a commercial API management tools can be used. Among others, there is the AWS internal Amazon API Gateway[77], or alternatively Microsoft's Azure API Management[78]. IBM API Connect [79]is also an API management service that can manage APIs across the entire business ecosystem.

The interfaces and AI applications are integrated via **microservice** approach[80], which means they are standalone services and receive a request for example per POST[81] or GET[82] method with a corresponding payload from the API manager, triggered by a user action in the frontend and returns a structured predefined data format such as JSON[83] or XML[84], which can be further processed by the receiver. Preferred here would be microservices with python based on FastAPI[85], which fulfills the REST architecture[86].

The microservice approach is preferred for machine learning and deep learning modules because of several advantages:

➔ Scalable — Each microservice is an independent component that runs its own process and is deployed independently. As each service is deployed independently, a particular microservice can be scaled independently of the entire application.

[71] https://www.nginx.com/ - Nginx is a web server that can also be used as a reverse proxy, load balancer, mail proxy and HTTP cache.
[72] https://en.wikipedia.org/wiki/Single_sign-on - Single sign-on is an authentication scheme that allows a user to log in with a single ID to any of several related, yet independent, software systems. True single sign-on allows the user to log in once and access services without re-entering authentication factors.
[73] https://en.wikipedia.org/wiki/Active_Directory_Federation_Services - Active Directory Federation Services, a software component developed by Microsoft, can run on Windows Server operating systems to provide users with single sign-on access to systems and applications located across organizational boundaries.
[74] https://vuejs.org/ - Vue.js is an open-source model–view–viewmodel front end JavaScript framework for building user interfaces and single-page applications. It was created by Evan You, and is maintained by him and the rest of the active core team members.
[75] https://www.javascript.com/ - JavaScript, often abbreviated as JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS.
[76] https://www.postgresql.org/ - PostgreSQL, also known as Postgres, is a free and open-source relational database management system emphasizing extensibility and SQL compliance. It was originally named POSTGRES, referring to its origins as a successor to the Ingres database developed at the University of California, Berkeley.
[77] https://aws.amazon.com/api-gateway/api-management/
[78] https://learn.microsoft.com/en-us/azure/api-management/
[79] https://www.ibm.com/products/api-connect
[80] https://en.wikipedia.org/wiki/Microservices
[81] https://en.wikipedia.org/wiki/POST_(HTTP)
[82] https://en.wikipedia.org/wiki/GET_(HTTP)
[83] https://en.wikipedia.org/wiki/JSON
[84] https://en.wikipedia.org/wiki/XML
[85] https://fastapi.tiangolo.com/
[86] https://en.wikipedia.org/wiki/Representational_state_transfer

- ➔ Agility — Failure in a microservice application only affects a particular service instead of the entire application. Therefore, fixing and debugging will be done on that particular microservice instead of pausing and fix the entire application.
- ➔ Flexibility — Team members are not limited by the programming languages or tools used to create and deploy the microservice and can have different frameworks for each microservice. In addition, if there are codes developed previously, team members can leverage those codes instead of re-building them again.
- ➔ Autonomy — Developing using a microservice architecture approach allows more team autonomy as each member can focus on developing a specific microservice that focuses on a particular functionality for example each member build a microservice that focuses on a particular task in the machine learning deployment process such as — data ingestion, feature engineering, data validation, model scoring, etc.
- ➔ Easy to Understand — As things are being split up into smaller components, a single microservice application is easier to understand and managed as usually, one microservice focuses on a particular task.

Microservice - Assignment of responsibilities (assisted)

At the **model and data level**, the function assignment of responsibilities (assisted) is a classical natural language processing[87] text classification[88] algorithm with multiclass[89] or multi label[90] predictions. The current state of the art models in this segment have a transformer architecture[91]. If only few labeled data is available (e.g. from 8 up to 64 samples) the setfit methodology is preferred over fine-tuning. As soon as more labeled data becomes available (>100 labels per class) traditional fine-tuning methodologies, for example the DeBERTa[92] model, can easily take over and replace setfit. These models are trained with python[93] using the packages Huggingface[94] and Pytorch[95].

In our interviews we discovered that the **dataset** for BUKEA and BSW holds more than 700 brief written inquiries each. The data has different formats. The brief written inquiries are stored in PDF, Word and Excel format whereas the answers are saved in PDF format. This implies a certain amount of data preparation before labeling can be started and will most likely not be automated, instead it will only be possible to process the data manually or semi-automatically. To train the classifier the extracted texts have to be labeled in a labeling iteration by subject matter experts. To conclude, the fundamental dataset is available to train an assisted assignment of responsibilities model.

At the **hardware level**, we need to distinguish between two environments. On the one hand, the training environment in which the transformer models are trained, and on the other hand, the productive environment that is connected to the API Manager for classification tasks as a microservice. The training environment needs a graphics card with at least 24 GB vRam from nvidia[96] (e.g. K80, RTX 3090 TI, etc…), since the models are trained on these graphics cards. Furthermore, at least 128 Gb Ram and 8 cores are seen as minimum requirements. The productive environment should have the same configuration, but can work with smaller graphics cards if necessary, because the models can be optimized beforehand with quantization[97] or pruning[98], to run on smaller computers. As an alternative to on-prem procurement of hardware, it is also possible to rent this hardware as required. This is possible with various service providers such as AWS, Azure or Google. But there is also a wide range of other service providers in addition to these.

[87] https://en.wikipedia.org/wiki/Natural_language_processing
[88] https://paperswithcode.com/task/text-classification
[89] https://en.wikipedia.org/wiki/Multiclass_classification
[90] https://en.wikipedia.org/wiki/Multi-label_classification
[91] https://en.wikipedia.org/wiki/Transformer_(machine_learning_model)
[92] https://arxiv.org/abs/2006.03654
[93] https://www.python.org/
[94] https://huggingface.co/
[95] https://pytorch.org/
[96] https://www.nvidia.com/en-us/
[97] https://arxiv.org/abs/2106.08295
[98] Pruning is a technique in deep learning that aids in the development of smaller and more efficient neural networks. It's a model optimization technique that involves eliminating unnecessary values in the weight tensor.

At the **software level** the training environment should be setup with an Ubuntu 20.04 LTS[99] or Ubuntu 18.04 LTS [100] operating system with anaconda[101] environment installed and with internet access. The microservice should run with a docker[102] container.

At the **staffing level** the following profiles are needed (they are not disjunct, so one person can unite different profiles):

- Data Scientist: Main area of expertise is the training of the models, with a focus on developing appropriate statistical models and algorithms
- Data Analyst: Data analysts specialize in the pre-processing and collection of data. They usually have a good understanding of databases and data visualization
- Special Matter Expert: Is a person who has accumulated knowledge in a particular field or topic and is familiar with the process and information processed and is responsible for labeling the data and giving business feedback
- Machine Learning Engineer: Focuses on deploying machine learning products, test set-up, releases and monitoring
- Frontend/backend Developer: Is usually responsible for writing APIs and the necessary extensions in the IDM front- and backend

Microservice - Intelligent search

At the **model and data level** the state of the art AI driven intelligent search is based on vector search algorithms. To describe it a bit more detailed, one uses SentenceTransformer[103] deep learning models to generate so called text embeddings[104] from documents and texts. These embeddings can then be compared similarity or distance metrics like cosine-similarity[105] to find texts with a similar meaning. This is the state of the art technique[106] for semantic textual similarity[107] (find similar documents from other documents) and semantic search[108] (find interesting documents from a search query). The framework SBERT[109], who can handle this, is based on PyTorch and Huggingface Transformers and offers a large collection of pre-trained models tuned for various tasks. Further, it is easy to fine-tune own models. Different wrapper libraries can be chosen to leverage SBERT like FAISS[110], Pinecone[111] and Haystack[112]. The tuning is done in two steps per adaptive pretraining[113]. First, the pre-trained models, such as "paraphrase-multilingual-mpnet-base-v2"[114], are pre-trained again with an unsupervised learning algorithm[115] like TSDAE[116]. This is done purely with the texts of the brief written inquiries with no labels being added. Then the so-called fine-tuning follows with the help of labeled data, wherein there are different types of labeling. The currently preferred way is with the help of a Contrastive Loss[117]. For a Contrastive Loss two texts and a label of either 0 or 1 is given. 1 if the two texts belong to the same cluster (same brief written inquiry) and 0 otherwise.

[99] https://releases.ubuntu.com/20.04/
[100] https://releases.ubuntu.com/18.04/
[101] https://www.anaconda.com/products/distribution
[102] https://www.docker.com/
[103] https://arxiv.org/abs/1908.10084
[104] https://en.wikipedia.org/wiki/Word_embedding
[105] https://en.wikipedia.org/wiki/Cosine_similarity
[106] https://haystackconf.com/files/slides/haystack2022/Scalable-Semantic-Search-at-Course-Hero-Kazem-Jahanbakhsh.pdf
[107] https://www.sbert.net/docs/usage/semantic_textual_similarity.html
[108] Semantic search seeks to improve search accuracy by understanding the content of the search query. In contrast to traditional search engines which only find documents based on lexical matches, semantic search can also find synonyms. https://www.sbert.net/examples/applications/semantic-search/README.html
[109] https://www.sbert.net/
[110] https://github.com/facebookresearch/faiss
[111] https://www.pinecone.io/
[112] https://haystack.deepset.ai/overview/intro
[113] https://www.sbert.net/examples/domain_adaptation/README.html#adaptive-pre-training
[114] https://huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2
[115] https://en.wikipedia.org/wiki/Unsupervised_learning
[116] https://arxiv.org/abs/2104.06979
[117] http://yann.lecun.com/exdb/publis/pdf/hadsell-chopra-lecun-06.pdf

The **dataset** for BUKEA and BSW which holds more than 700 brief written inquiries each is considered to be sufficiently large to follow an adaptive pretraining approach and come up with an operational intelligent search.

Hardware, Software and Staffing level are not subject to fundamental changes compared to the microservice assisted assignment of responsibilities and can be implemented in the same way.

At the **data architecture level**, the objective is to support the data-centric approach with the help of the architecture. This means, for example, that there is only one single source of truth (SSOT) in which all data is always up to date. The SSOT involves the creation of a single data model that is used by all users and associated information systems. Using a single project-wide data source eliminates isolated data repositories created by information systems and their associated data sources and structures, thus avoiding multiple data instances. An appropriate infrastructure for realizing the data-architecture is provided by the hosting platforms mentioned above, such as Amazon Webservices, Microsoft Azure or Google. With the flexible solutions of these service providers, all different data requirements can be handled individually. Structured data that changes little has different requirements than frequently changing data with many accesses. All of these service providers have a repertoire of extensive database systems, to handle these different requirements[118]. If the demand for on-premise solutions is desired, the free PostgreSQL[119] management system, which is a community-based open source database management system, should be considered. With PostgreSQL all SQL related requirements can be realized without leaving the in-house hardware.

Microservice - eSIS Interface

The eSIS Interface is a simple interface to provide the necessary data from the presidential department to the senate chancellery and vice versa. This includes notifications from the presidential department to the senate chancellery and in return as well as submitting the final answer and additional information to the senate chancellery.

Because three are no models backing up this microservice, we don't need to define a **model and data level** for this microservice.

At **hardware level** this microservice is satisfied with low requirements, since only database queries and forwarding take place. The productive environment should be useable with a two Core CPU and 32 GB of RAM with no graphic card.

At the **software level** the microservice should run with a docker container and python installed.

At the **staffing level** the following profiles are needed (they are not disjunct, so one person can unite different profiles):

- Special Matter Expert: Is a person who has accumulated knowledge in a particular field or topic and is familiar with the process and information processed and is responsible for labeling the data and giving business feedback
- Backend Developer: To extend IDM backend. PostgreSQL and Python skills are required

At the **data architecture level,** the SSOT would be the IDM database and IDM archive database. The eSIS Interface will pull and push information from and to the SSOT.

[118] https://aws.amazon.com/de/products/databases/
[119] https://www.postgresql.org

Senate printed matter coordination (Senatsdrucksachenabstimmung)

This chapter describes the technical to-be situation and its underlying components as well as their technical implications for the to-be process senate printed matter coordination (Senatsdrucksachenabstimmung).

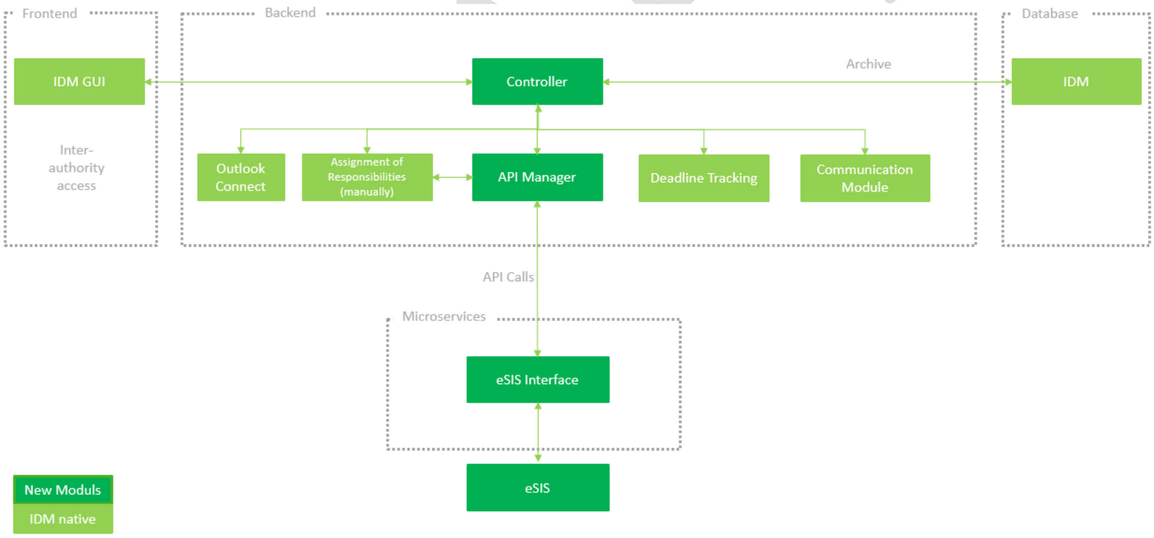
For the brief written inquiries, the preferred alternative that resulted from the Business Case analysis (see Deliverable 5) is alternative 2. This alternative foresees the implementation of an already existing process management tool, the IDM tool (Intelligent Dialogue Management), whereby the tool is equipped with further functionalities. Hamburg's internal steering group decided against the use of artificial intelligence in this use case. The following functionalities have been agreed upon:

- A **central access channel (dashboard)**
- **Assignment of responsibilities (manually)**
- **Archiving functionality** of information that arise within the process (e.g. statements of the involved authorities, finalized printed matter and process meta information)
- **Monitoring** and
- **Deadline tracking**
- **Comment and communication** functions
- **Inter-authority access**
- **eSIS Interface**

To-be infrastructure

The following figure provides an overview of the to-be technical infrastructure for the process brief written inquiries.

Figure 3: To-be infrastructure for the process senate printed matter coordination



Source: Deloitte (2022)

Like in the brief written inquiries use case the senate printed matter coordination use case also build on the existing IDM tool. This is extended in particular by a controller that manages the various existing and new functionalities. The IDM tool database, an adapted version of the IDM frontend, the manual assignment of responsibilities module, the deadline tracking module and the communication module will remain and are called main functions.

IDM main functions

The IDM main functions are the same as those of the process brief written inquiries. A detailed list of the **hardware, software, data architecture and staffing layers** can be found in the respective chapterIDM main functions.

Analogous to the brief written inquiries, **inter-authority access** for the IDM frontend will be added as a new functionality, as well as the previously elaborated **API manager** and a **backend controller**, which in the case of the senate print matter coordination will serve the e**SIS interface** instead of multiple microservices. The backend controller should preferably be implemented in the backend language of choice, for example in Python. The API Manager can either be developed in-house to be perfectly integrated into the environment, or a commercial API management tool can be used. For further technical details and advantages of the microservice approach please look at the previous chapter brief written inquiries in which this is explained in more detail and analog in this use case.

Microservice - eSIS Interface

The eSIS Interface is a simple interface to provide the necessary data from the presidential department to the senate chancellery and vice versa. This includes the report of the printed matter from the presidential department to the senate chancellery as well as the other way around . E.g. as soon as the printed matter is placed on the agenda of the Senate Chancellery (Senatskanzlei), the initiated authority is informed about this via the IDM tool interface.

Because there are no models backing up this microservice, we don't need to define a **model and data level** for this microservice.

The **hardware level** this microservice is satisfied with low requirements, since only database queries and forwarding take place. The productive environment should be useable with a two Core CPU and 32 GB of RAM with no graphic card.

At the **software level** the microservice should run with a docker container and python installed.

At the **staffing level** the following profiles are needed (they are not disjunct, so one person can unite different profiles):

- Special Matter Expert: Is a person who has accumulated knowledge in a particular field or topic and is familiar with the process and information processed and is responsible for labeling the data and giving business feedback
- Backend Developer: To extend IDM backend. PostgreSQL and Python skills are required

At the **data architecture level** the SSOT would be the IDM database. The eSIS Interface will pull and push information from and to the SSOT.

Imputing procedure (Imputing-Verfahren) - BohrIS

This chapter describes the technical to-be situation and its underlying components as well as their technical implications for the to-be process BohrIS.

For the process BohrIS, the preferred alternative that resulted from the Business Case analysis (see Deliverable 5) is alternative 3. This alternative foresees a workflow management tool such as the DM tool (Intelligent Dialogue Management) or "Modul F", whereby the tool is equipped with further functionalities, including imputing, completeness, and plausibility checks with artificial intelligence support. The following functionalities have been agreed upon:

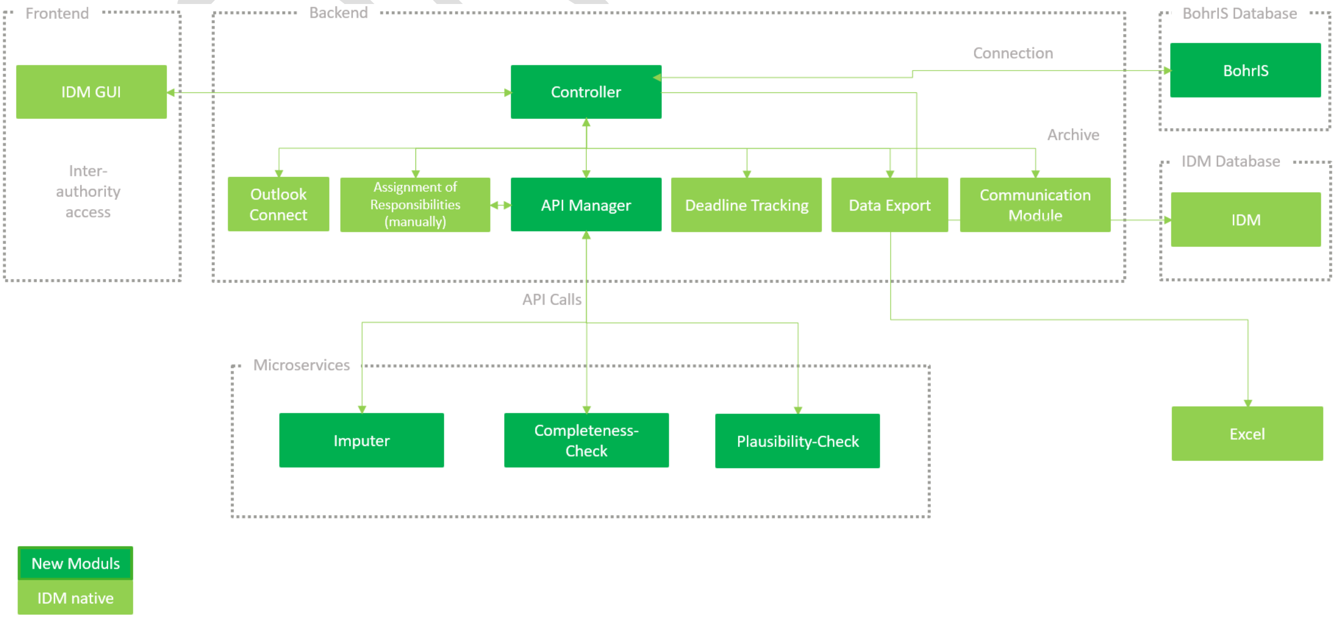
- A central access channel (dashboard)
- Assignment of responsibilities (manually)
- Archiving functionality of information that arise within the process (e.g., Drilling notification according to §§50, 127 Federal Mining Act, §8 Geological Data Act and §49 Federal Water Act; Drilling log, drilling profile and bore log for the technical data as well as location plan as well as process meta information like deadlines etc.)
- Monitoring
- Deadline tracking
- Comment and communication functions
- Inter-authority access
- Imputer
- Completeness check
- Plausibility check

To-be infrastructure

The following figure provides an overview of the to-be technical infrastructure for the BohrIS process.

Like in the two use cases before the imputing procedure use case also build upon the existing IDM tool or "Modul F". The workflow management tool will then be extended by a controller that manages the various existing and new functionalities. The IDM tool database, an adapted version of the IDM frontend, the manual assignment of responsibilities module, the deadline tracking module and the communication module.

Figure 4: To-be infrastructure for BohrIS (imputing procedure)



Source: Deloitte (2022)

IDM main functions

The IDM main functions are the same as those of the previous two procedures process brief written inquiries and senate printed matter coordination. A detailed list of the **hardware, software, data architecture and staffing layers** can be found in the chapter brief written inquiries IDM main functions.

Analogously to the brief written inquiries, **inter-authority access** for the IDM frontend will be added as a new functionality. Furthermore, a newly introduced **API manager** is controlled by the **backend controller**, which ensures that all interfaces (the imputer, completeness check and the plausibility check) can be provided. The backend controller should preferably be implemented in the backend language of choice, for example in Python. The API Manager can either be developed in-house to be perfectly integrated into the environment, or a commercial API management tools can be used.

For further technical details and advantages of the microservice approach please look at the previous chapter brief written inquiries in which this is explained in more detail and analog to this use case.

Microservice – completeness check

The completeness check verifies that no entries in the submission process steps are missing. An example is the automated standard mail sent to the mailbox bohranzeigen@bukea.hamburg.de from Norddeutsche Bohranzeige (NOBO). Here, basic information about new drilling projects is transmitted using PDF, XML and XSD format. Typical data fields would be the BID (drilling identification), the location of the drill, the district, the parcel, drilling start date as well as coordinates, etc. In the picture below an example PDF of a NOBO transmission is shown. In the first drilling project the parcel (Flurstück) is not specified and in the second drilling project the parcel is specified. Such missing values would be targeted by the completeness check and raise a notification.

Figure 5: Screenshot 1 of a data extract from the BohrIS database

BID (Bohrungsidentifikation):	2325H10050
Blattnummer der TK25:	2325
Standort - Straße, Hausnr.:	Brückwiesenstraße 8
Standort - PLZ, Ort:	22453 Hamburg
Gemarkung:	Groß-Borstel
Flurstück:	
Gemeinde:	Hamburg, Freie und Hansestadt
Landkreis:	Hamburg, Freie und Hansestadt
Koordinaten (EPSG: 4647):	32564048,34 / 5940238,82
Bohrungsname:	Groß Borstel-1
Bohrstrecke [m]:	100
Bohrbeginn:	04.11.2020
Art des Vorhabens:	Erdwärmesonde
Bohrverfahren:	Drehspülbohrung [Rotationsspülbohrung]
Bohrzweck:	Erdwärmegewinnung
Bohrdurchmesser [mm]:	160
Bemerkung:	

Vertraulichkeit:	Die Bohrergebnisse sind für Dritte unmittelbar freigegeben.

BID (Bohrungsidentifikation):	2325H10051
Blattnummer der TK25:	2325
Standort - Straße, Hausnr.:	Brückwiesenstraße 8
Standort - PLZ, Ort:	22453 Hamburg
Gemarkung:	Groß-Borstel
Flurstück:	1073
Gemeinde:	Hamburg, Freie und Hansestadt
Landkreis:	Hamburg, Freie und Hansestadt
Koordinaten (EPSG: 4647):	32564044,46 / 5940234,26
Bohrungsname:	Groß Borstel-2
Bohrstrecke [m]:	100
Bohrbeginn:	05.11.2020
Art des Vorhabens:	Erdwärmesonde
Bohrverfahren:	Drehspülbohrung [Rotationsspülbohrung]
Bohrzweck:	Erdwärmegewinnung
Bohrdurchmesser [mm]:	160
Bemerkung:	

Source: Geological State Office, City of Hamburg (2022)

At the **model level**, we are in a rule-based system. If a previously defined field is missing, a definition would be made as to whether it is optional or mandatory. If optional fields are missing, a yellow notification would follow, if mandatory fields are missing, a red notification would occur. The **data** necessary to create the completeness checks are more than adequately fulfilled by the **dataset**. The database contains several thousand entries of wells and about 5000 new records are added annually.

Microservice – plausibility check

The plausibility check module provides the possibility to check the delivered data for plausibility. This can be done either based on a rule-based approach or an AI module (**model level**). Which approach is more suitable must be checked in the context of the supplied data. For a first implementation rule-based models will do fine and can be extended by AI modules. Both methodologies can also be used simultaneously side by side for different checks. The **data** will also be checked manually for correctness afterwards in terms of content. The data necessary to create the plausibility-checks is more than adequately fulfilled by the **dataset** with several thousand records in the database.

To give an example, the automated submission to the mailbox bohranzeigen@bukea.hamburg.de from Norddeutsche Bohranzeige (NOBO) contains a PDF and an XML file that has the following information, shown in the figure below, in it. One can see that the E-mail address for the “Auftraggeber”, highlighted in yellow, is a dummy E-mail. The completeness check will pass with no notification, but the plausibility check can raise a notification based on rules detecting dummy E-mail addresses. Other examples could be, that the address is no valid address because the zip code doesn’t align with the city name or the “Bohrdurchmesser” of 160 millimeters and the “Bohrstrecke” of 100 meters in figure 6 doesn’t lie in the expected range.

Figure 6: Screenshot 2 of a data extract from the BohrIS database

Anzeige eines Bohrvorhabens
- Pdf-Ausdruck für die Unterlagen des Anzeigenden -

erstellt durch
Norddeutsche Bohranzeige Online

07.10.2020

Bohrfirma:

Name:	CST-Erdenergie, Carsten Stawaritsch
Straße:	Steinweg 5
PLZ, Ort:	27801 Dötlingen
Telefon:	04432 / 918630
E-Mail:	info@cst-erdenergie.de

Auftraggeber:

Name:	Markwardt, Ulrike und Kaufholt, Jochen
Straße:	Straßenbahnring 37
PLZ, Ort:	20251 Hamburg
Telefon:	04261 848801
E-Mail:	unbekannt@unbekannt.de

Beratende Firma:

Name:	CST-Erdenergie, Carsten Stawaritsch
Straße:	Steinweg 5
PLZ, Ort:	27801 Dötlingen

Source: Geological State Office, City of Hamburg (2022)

Microservice – Imputer

The imputer module provides the possibility to impute missing values recognized from the completeness check or annotated as missing manually by hand. To give an example, the missing value for “Flurstück” in the first drilling project in figure 6 can be filled / imputed by a rule-based imputer because the address information in conjunction with a cadaster can automatically derive this information. Another example would be to impute a missing “Bohrdurchmesser” based on other information given like “Bohrstrecke” or “Bohrverfahren” with a machine learning based imputing method.

Within the **model layer** of the imputer we can use rule based imputing and imputing based on machine learning. The methods used to impute via machine learning are the multivariate imputation by chained equations (MICE) [120]imputer and the KNN Imputer[121], both implementable in python with the package scikit-learn. The **data** necessary to create the machine learning based imputation is more than adequately fulfilled by the **dataset** with several thousand records in the database.

The **data architecture layer** as well as **the hardware layer, the software layer and the staffing** can be described aggregated for all microservices.

The **data architecture layer** from BohrIS is driven by an Oracle database[122]. Also, a PostGIS[123] implementation is added to support geographic objects to the object-relational database. The BohrIS database can be connected to the IDM tool via interface to pull and push information. The main IDM Tool database, like in previous procedures, can be a SQL based database and powers all microservices.

The **hardware layer** for all three microservices can be handled with one machine with 6 CPU Cores (~3.3 GHz), 64 GB RAM with no graphic card.

At the **software layer a** setup with an Ubuntu 20.04 LTS or Ubuntu 18.04 LTS as operating system with anaconda environment, docker and python installed and with internet access is required. The microservice should run on docker containers on one machine.

At the **staffing level** the following profiles are needed (they are not disjunct, so one person can unite different profiles):

- Data Scientist: Main area of expertise is the training of the models, with a focus on developing appropriate statistical models and algorithms
- Data Analyst: Data analysts specialize in the pre-processing and collection of data. They usually have a good understanding of databases and data visualization
- Special Matter Expert: Is a person who has accumulated knowledge in a particular field or topic and is familiar with the process and information processed and is responsible for labeling the data and giving business feedback
- Machine Learning engineer focuses on deploying machine learning products, test set-up, releases and monitoring
- Frontend/backend Developer: Is responsible for writing APIs and the necessary extensions in the IDM front- and backend

[120] https://arxiv.org/abs/2203.00087 - Using Multivariate Imputation by Chained Equations to Predict Redshifts of Active Galactic Nuclei

[121] Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein and Russ B. Altman, Missing value estimation methods for DNA microarrays, BIOINFORMATICS Vol. 17 no. 6, 2001 Pages 520-525.

[122] Oracle Database is a multi-model database management system produced and marketed by Oracle Corporation. It is a database commonly used for running online transaction processing, data warehousing and mixed database workloads.

[123] PostGIS is an open source software program that adds support for geographic objects to the PostgreSQL object-relational database. PostGIS follows the simple features for SQL specification from the Open Geospatial Consortium. Technically PostGIS was implemented as a PostgreSQL external extension.

Info boxes (Infoboxen)

This chapter describes the technical to-be situation and its underlying components as well as their technical implications for the to-be process info boxes (Infoboxen).

For the process info boxes, the preferred alternative that resulted from the Business Case analysis (see Deliverable 5) is alternative 3. This alternative foresees the IDM workflow management tool, whereby the tool is equipped with further functionalities, including artificial intelligence. The following functionalities have been agreed upon:

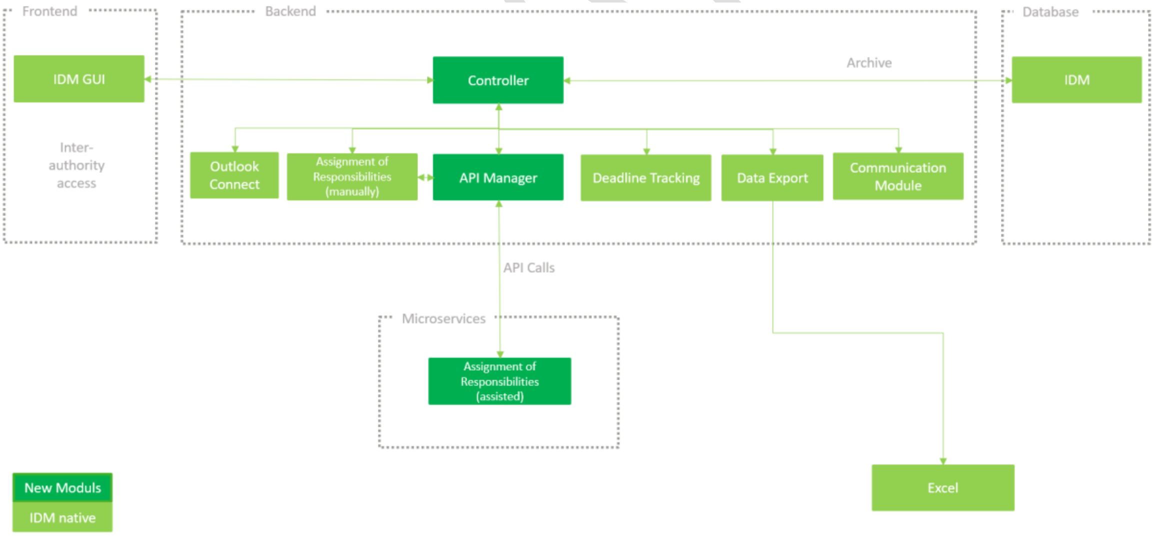
- A **central access channel (dashboard)**
- **Assignment of responsibilities (manually)** and
- **Assignment of responsibilities (assisted**; supported by a AI forwarding assistant**)**
- **Archiving** the received messages
- **Monitoring** and
- **Deadline tracking**
- **Comment and communication** functions
- **Inter-authority access**
- **Data export**

To-be infrastructure

The following figure provides an overview of the to-be technical infrastructure for the info boxen.

Like in the previous use cases the info boxes use case also builds on the existing IDM tool. This is extended by a controller that manages the various existing and new functionalities. The IDM tool database, an adapted version of the IDM frontend, the manual as well as the assisted assignment of responsibilities module, the deadline tracking module and the communication module.

Figure 7: To-be infrastructure for the process info boxes (Infoboxen)



Source: Deloitte (2022)

IDM main functions

The IDM main functions are similar to the previously described use cases. A detailed list of the **hardware, software, data architecture and staffing layers** can be found in the respective chapters.

Analogous to the previously described use cases**, inter-authority access** for the IDM frontend is added as a new functionality, as well as the already elaborated **API manager** and a **backend controller** serving the **assisted assignment module** instead of multiple microservices in the case of info boxes. The backend control should preferably be implemented in the backend language of choice, e.g. Python. The API manager can either be developed in-house to integrate perfectly into the environment, or a commercial API management tools can be used. More technical details and benefits of the microservice approach can be found in the previous chapter brief written inquiries, where this is further explained and analogous to this use case.

Microservice – Assignment of responsibilities (assisted)

At the **model and data level**, function assignment of responsibilities (assisted) is a traditional algorithm for classifying text in natural language processing with multiclass predictions, where the current state of the art is models with transformer architecture that can yield great results with limited labeled data. The **data** available to create the assignment of responsibilities (assisted) is not sufficient to train the models. Due to General Data Protection Regulation (GDPR)[124] right now there is no permanent storage and no archive of the data in place. All data is stored only temporarily for one month. This leads to the conclusion, that traditional state of the art models like DeBERTa can only be trained after collecting more data. The preferred solution could be to train few shot learning models like setfit, which achieve better results with less labeled data and start collecting more data under GDPR conform circumstances.

Further elaboration of the respective **software level**, **hardware level**, as well as the **staffing label** can be found in the preceding chapter of the brief written inquiries use case.

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

Knowledge management (Wissensmanagement)

This chapter describes the technical to-be situation and its underlying components as well as their technical implications for the to-be process knowledge management (Wissensmanagement).

For the knowledge management process, the preferred alternative that resulted from the Business Case analysis (see Deliverable 5) is alternative 3. This alternative foresees the IDM workflow management tool, whereby the tool is equipped with further functionalities, including artificial intelligence. The following functionalities have been agreed upon:

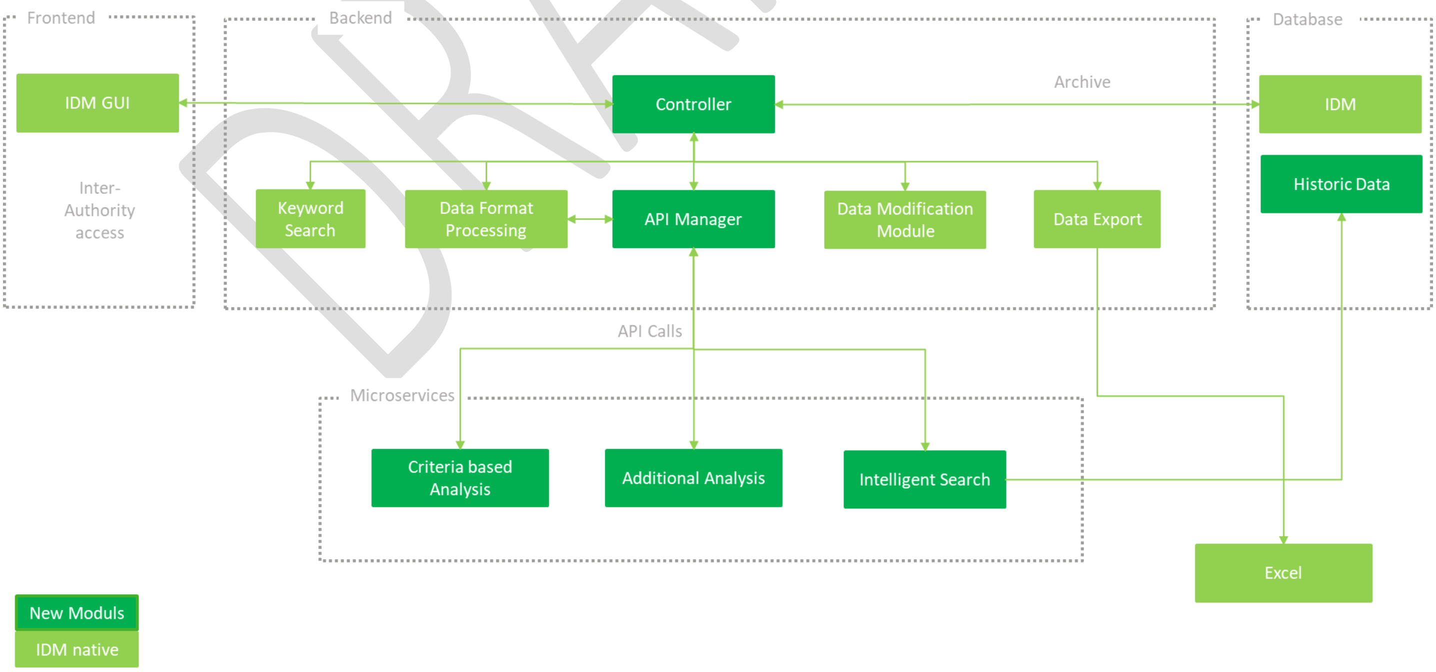
- A central access channel (dashboard)
- Data format processing
- A data modification module
- Archiving of all data provided by any department
- Inter-authority access
- Data export
- Intelligent search
- Criteria based analysis and additional analysis

To-be infrastructure

The following figure provides an overview of the to-be technical infrastructure for the knowledge management process.

Like in all previous use cases the knowledge management use case also builds on the existing IDM tool. This is extended by a controller that manages the various existing and new functionalities. The expanded functionalities are a customized version of the IDM frontend, keyword search, the data format processing module, the data modification module, and the microservice-based modules for criteria-based analysis, additional analysis, and intelligent similarity search.

Figure 8: To-be infrastructure for the knowledge management process



Source: Deloitte (2022)

IDM main functions

The IDM main functions are similar to the previously described use cases. A detailed list of the **hardware, software, data architecture and staffing layers** can be found in the respective chapters.

Analogous to the previously described use cases**, inter-authority access** for the IDM frontend is added as a new functionality, as well as the already elaborated **API manager** and a **backend controller** serving the **assisted assignment module** instead of multiple microservices in the case of info boxes. Similar to the previously described use cases, the backend controller should preferably be implemented in the chosen backend language, for example, python. The detailed preference of the API manager is described in more detail in the previous chapters.

Microservice – Intelligent Search

Intelligent search has also been elaborated in considerable depth in the use cases described previously and should be implemented accordingly when realizing the knowledge management use case.

Microservice – Criteria Based Analysis & Additional Analysis

The **criteria-based and additional analysis** module provides the possibility to create further analyses based on the historical database of the knowledge management process. These analyses can contain, for example, clustering of text documents based on topics via topic modelling[125] (models would be BERTopic[126] and HDBSCAN[127]) and exploratory data analysis (EDA)[128] about the size of the stored documents, the extensions (*.pdf, *.xlsx, *.docx, *.jpeg etc.) or meta data of the files. Through these analyses, an overview of the use and data basis of the knowledge management process can be obtained and are presented in a graphical or a tabular form. Illustrative analyses could be a distribution analysis via histograms[129] of the extensions or a time series analysis of the added documents over time.

These additional analyses will not be deployed directly for the time being but can be added to the dashboard afterwards at any time.

The **data architecture layer** as well as **the hardware layer, the software layer and the staffing** can be described aggregated for all microservices.

The **data architecture layer** from the knowledge management process is driven by an Oracle database. Also, a PostGIS implementation is added to support geographic objects to the object-relational database. The database can be connected to the IDM tool via an interface to pull and push information. The main IDM Tool database, like in previous procedures, could be an SQL based database and powers all microservices.

The **hardware layer** for all microservices can be handled with one machine with 6 CPU Cores (~3.3 GHz), 64 GB RAM with no graphic card.

At the **software layer** a setup with an Ubuntu 20.04 LTS or Ubuntu 18.04 LTS as operating system with anaconda environment, docker and python installed and with internet access is required. The microservice should run on docker containers on one machine.

At the **staffing level** the following profiles are needed (they are not disjunct, so one person can unite different profiles):

- Data Scientist: Main area of expertise is the training of the models, with a focus on developing appropriate statistical models and algorithms
- Data Analyst: Data analysts specialize in the pre-processing and collection of data. They usually have a good understanding of databases and data visualization
- Special Matter Expert: Is a person who has accumulated knowledge in a particular field or topic and is familiar with the process and information processed and is responsible for labeling the data and giving business feedback
- Machine Learning Engineer: Focuses on deploying machine learning products, test set-up, releases and monitoring

[125] In statistics and natural language processing, a topic model is a type of statistical model for discovering the abstract "topics" that occur in a collection of documents. Topic modeling is a frequently used text-mining tool for discovery of hidden semantic structures in a text body.
[126] https://github.com/MaartenGr/BERTopic
[127] https://arxiv.org/abs/1911.02282 - A Hybrid Approach To Hierarchical Density-based Cluster Selection
[128] Exploratory Data Analysis (EDA) is an approach to analyze the data using visual techniques. It is used to discover trends, patterns, or to check assumptions with the help of statistical summary and graphical representations.
[129] https://en.wikipedia.org/wiki/Histogram - A histogram is an approximate representation of the distribution of numerical data.

- Frontend/backend Developer: Is responsible for writing APIs and the necessary extensions in the IDM front- and backend

Conclusion

The detailed analysis regarding specific legal, organisational and technical implications for the five selected to-be processes shows that the **four processes** brief written inquiries (Schriftliche Kleine Anfragen), senate printed matter coordination (Senatsdrucksachenabstimmung), the imputing procedure (BohrIS process) and the knowledge management process (Wissensmanagement) **have good preconditions to implement the respective to-be models**. This entails the implementation of the IDM workflow management tool (or, in the case of the imputing procedure BohrIS, the "Modul F") with several core functions and, depending on the process, several microservices, e.g. interfaces to existing databases and/or solutions and features such as the completeness check.

Given the fact that **for the process info boxes**, no or only a few data samples are currently stored, the foreseen AI forwarding assistant (assignment of responsibilities (assisted)) cannot be trained with traditional methods yet, but the function could be trained with a few shot learning algorithm like setfit. The **data necessary to create the assignment of responsibilities (assisted)** is, however, **not sufficient to train a state of the art model like DeBERTa**. Hence, in this case, incoming data (emails from external stakeholders) must in the first place be collected and stored. A phased approach could be taken, implementing the IDM tool first and after sufficient data has been collected, the assignment of responsibilities (assisted) can be trained and implemented. This is under the assumption, that all incoming data is saved in accordance with GDPR compliance.

Visit our website:

Deloitte.

Funded by
the European Union

Find out more
about the Technical
Support Instrument: