Research and Data Centre

Cahier 2024-21

Data lineage for the justice system

Scope, potentials, and directions

Dutch institute for Justice and Security

Cahier 2024-21

Data lineage for the justice system

Scope, potentials, and directions

Mortaza S. Bargh

Cahier
This series comprises overviews of studies carried out by or for the WODC Research and Data Centre. Inclusion in the series does not mean that the sheet's contents reflect the viewpoint of the Minister of Justice and Security.

Table of contents

Abbreviations

AI:	Artificial Intelligence
ChatGPT:	Chat Generative Pre-Trained Transformer
DAMA:	Data Management (community)
DBMS:	DataBase Management System
DL:	Data Lineage
DJS:	Dutch Justice System
DQM:	Data Quadrant Model
DQSS:	Data Query Support Solution
ETL:	Extraction, Transformation and Load
FAIR:	Findable, Accessible, Interoperable and Reusable
GDPT:	General Data Protection Regulation
GPAI:	General-Purpose AI
IdP:	Identity Provider
IS:	Information System
LLM:	Large Language Model
MA:	Metadata Aggregator
ML:	Machine Learning
NLIDBS:	Natural Language Interfaces for DataBase
NoSQL:	Not only SQL
PASS:	Provenance-Aware Storage System
PDP:	Policy Decision Point
PEP:	Policy Enforcement Point
PPS:	Public Prosecution Service
PS:	Probation Service
R&D:	Research & Development
SP:	Service Provider
SPADE:	Support for Provenance Auditing in Distributed Environments
SQL:	Structural Query Language
SSBI:	Self-Service Business Intelligence
UDF:	User Defined Function
VQL:	Visual Query Language
WFMS:	WorkFlow Management System
WODC:	Wetenschappelijk Onderzoek- en Datacentrum

Summary

Problem statement

Problem (context)

Data is currently being generated, collected, shared, analyzed, and distributed at a fast-growing pace. As a result of this growth, there is a rising interest (and demand) to harvest the available data by using (advanced) algorithms to analyze data and develop data-driven systems for easing our daily lives, creating additional value for businesses, providing insight into societal phenomena, and guiding policymaking processes. The way that data is collected, which is often blended with biased, partial, faulty, sensitive, and stigmatizing information about individuals, groups, and organizations; and the way that algorithms and data-driven systems are designed, implemented, interpreted, and (mis)used (are going to) impact us deeply at individual, group, and societal levels. Therefore, to capitalize on data one should be attentive of the risks of the data used for delivering personal, social, or organizational benefits. As such, gaining trust in data is a perquisite for respecting the basic human rights like privacy, liberty, autonomy, and dignity.

Also in the justice domain, particularly in the Dutch Justice System (DJS), we witness a trend of applying digital technology and data-driven systems. The Information Systems (ISs) in the justice domain, which collect, store, share and processes data, are often physically distributed, have many loosely coupled subsystems, and are administrated by various organizations (i.e., spreading across many administrative domains). Utilizing data in this setting requires interconnecting various information sources and integrating their information in a trustful and responsible way. Those who share data (like judicial service providers) should entrust data consumers in using the data responsibly and those who consume data (like policymakers) should trust data sources in collecting and sharing data responsibly.

Establishing trust in data production and consumption requires, among others, mitigating the security issues of largely distributed data-reliant systems, managing the data quality issues of loosely coupled data sources, and dealing with wrongful and/or malicious use of data and algorithm outcomes. Given, on the one hand, the importance of data sharing and, on the other hand, the increased complexity involved in data sharing among (many) stakeholders and ISs, there is a need for establishing an appropriate data ecosystem. This data ecosystem establishment requires solid and effective data governance and data management to ensure the quality of data, secure the storage and exchange of data, optimize the tradeoff among contending values (like data utility and data privacy), and operationalize the Findable, Accessible, Interoperable and Reusable (FAIR) principles for the data.

An efficient and effective data governance/management requires, among others, data lineage and data provenance. Data lineage refers to the process of tracking the flow of data over time, i.e., during the data lifecycle/journey. It uses metadata to provide a clear description of the data origin(s), data changes in its journey, and data destination(s). As a similar case, data provenance (sometimes) refers to the sources (i.e., the origins) of the data and its historical changes.

Data lineage is seen as an essential instrument for enhancing data trustworthiness. It enables, for example, efficient data quality management, data change management, and data lifecycle management. A famous metaphor used to illustrate the role of data lineage for gaining trust in data is the case of gaining trust in the nutritiousness and healthiness of an apple by knowing how it is cultivated, harvested, transported, stored, distributed, and retailed. To gain this trust, one could keep track of the relevant information (lineage information) in every stage of the apple's lifecycle (i.e., the apple's journey) in the supply chain. In this metaphor, the object of interest is a physical object (commodity). Having the same type of assurance is also relevant for any digital object (e.g., a dataset, a digital picture, and a digital document). For example, citizens are increasingly subject to a growing quantity of multimedia content (i.e., data) of various types like images, videos, audio recordings, and documents. This content is often blended with misinformation (e.g., photos generated by Generative AI) or manipulated information (e.g., photos processed by the Photoshop tool), which makes it difficult for ordinary people (or even professionals) to distinguish between real and fake/manipulated content. In the case of photo sharing, data lineage can help photo makers, publishers, and consumers the ability to learn about how and by whom the photo is created and to learn about every edit made to the photo throughout its lifecycle/journey. The lineage information, which can trustfully be appended to the photo and accompany the photo through its entire journey, can easily be inspected by downstream consumers to get ensured about the origin and any edition made to the photo. This knowledge can help downstream consumers gain trust in the media content they encounter on, for example, social media and news feeds. Within the DJS, data lineage can similarly contribute to gaining trust in, for example, a standing policy by answering questions like *which datasets or documents are used as evidence for grounding the policy*.

In this contribution, we describe the results of our explorative study about data lineage (and provenance) technology, particularly about the concepts behind data lineage and the methods/tools used for data lineage. The study context relates to the data collected, shared, stored, and processed by the ISs within the DJS.

Research objective and research questions

The research objective can be specified as investigating how data lineage technology can contribute to data governance and data management within the DJS. Achieving this objective requires investigating the benefits of data lineage technology and the directions (and challenges) that one might explore (and expect) in deploying it within the DJS.

This study is a preliminary study towards the abovementioned objective. The research approach can be characterized as explorative, where we seek answers to the following research questions:

1 *What is data lineage?* For answering this question, we will also describe the context (or the data ecosystem) in which data lineage is used.
2 *Which objectives can data lineage contribute to?* For answering this research question, we will give an insight in the potential advantages of data lineage.
3 *How can data lineage tools be deployed?* For answering this question, we will elaborate on typical approaches for and challenges of data lineage deployment.
4 *What are the capabilities (and limitations) of existing data lineage tools?* For answering this question, we will sketch a framework for specifying the relevant data lineage capabilities. Further, as an example and for a limited number of existing

data lineage tools, we will give an insight in two capabilities that are of interest for this study.

Scope

These research questions will be addressed within the context of DJS, which consists of many semi-autonomous organizations, collectively implementing the rule of law within the Dutch society. The relations between these organizations are often characterized as a linear chain (where a stage must be concluded before the next stage may begin), having sometimes loops and parallel relations. The term justice system is used to refer to the bodies in the apparatus of law, which are involved in creating data, ranging from legislative texts to judicial decisions. As such, the scope of the justice system is broader than courts and court procedures.

In this contribution we intend to provide an overview of some relevant aspects that can be considered for deploying data lineage in the organizational setting of the DJS. As such, we do not intend to design or prescribe a solution for data lineage deployment in this contribution. The target audience of this report is system designers and architects as well as data officers and engineers. The report aims at informing these groups about the design space within which they can design or choose a data lineage solution.

Methodology

For this study we have conducted a critical literature review, where several selected information sources are analyzed, and a reflection is done on the existing concepts, methods, and approaches. The selected information sources are not only from scholarly literature, but also from gray literature like commercial websites, whitepapers, and weblogs. The latter is because many vendors and system developers are dominantly active in the field of data lineage, who introduce many innovative concepts, features, and tools to the domain.

In addition to literature study, we have conducted four semi-structured interviews with experts from different organizations within the DJS to gain insight in ongoing data lineage related (R&D) activities within the DJS as well as in eliciting the needs and visions of those experts involved in data governance/management within their organizations. Further, we organized two expert focus groups with data stewards and data management experts to present our intermediary results and get early feedback. The four interviewees and the two focus groups were chosen based on the expertise and availability of the participants rather than their representativeness. This choice is motivated by the nature of the study in being preliminary and explorative.

Main results and contributions

In this report we provide an overview of several aspects of data lineage technology and its deployment in cross organizational settings. The report can serve as a knowledge base for informing the design and deployment processes of the data lineage in the DJS setting. In the following, we summarize the answers given to the posed research questions.

What is data lineage?

Based on some existing definitions and the insights gained during the study on data lineage, we define data lineage as *the description of data movements and transformations at various abstraction levels along data journey paths. The description encompasses those aspects that are of interest in an application context like how (i.e., by whom, when, where, which, etc.) data objects are processed (i.e., created, collected, stored, accessed, transformed, transmitted, etc.) and how they are related to high-level data concepts.* This definition conceptualizes not only the physical distribution of data objects (like data origins, flows and transformations), but also the semantical distribution of the related concepts (like the business, legal and organizational terms that relate or apply to those data objects). Further, the definition offers a means to limit the scope of data lineage to those aspects that are of interest in each context.

Data lineage contributes to gaining trust in data and in responsible data transformation and sharing. Data lineage relies on deriving and managing metadata that is relevant for the aimed data lineage usage objective(s). Data lineage can be characterized from various aspects, which are not necessarily independent. These aspects include (a) data origin vs data flow lineage, (b) where vs how data lineage, (c) data transformation types, (d) coarse-grained vs fine-grained data lineage, (e) lazy vs eager data lineage, (f) backwards vs forward data lineage, (g) tracing vs tracking data lineage, (h) technical vs business data lineage, and (i) horizontal vs vertical data lineage. These data lineage characteristics specify the technical space in which a data lineage solution can be designed, chosen, and/or deployed.

Which objectives can data lineage contribute to?

The concept of lineage has been applied to a wide range of cases, ranging from tracking/tracing the computation flows in a single software program to tracing/tracing the data flows in distributed ISs. Data lineage can contribute to many objectives, each of which, in turn, plays a role in enhancing trust in data, data sharing, and data-driven applications and policymaking. These objectives include (a) data governance, (b) privacy protection, (c) trusting AI models, (d) data and AI explainability, interpretability and fairness, (e) data quality management, (f) data change management, (g) data ownership, (h) regulatory compliance, audit and accountability, (i) data security, (j) data modeling, and (k) data discovery. These objectives capture the usage areas of data lineage and, as such, they specify the *societal relevancy of data lineage* at large.

It would be intriguing to aim at all mentioned objectives when deploying a data lineage solution. Such a versatile data lineage solution could immediately become too complex and costly, thus might become impossible to realize especially in distributed settings (e.g., among the semi-autonomous organizations of the DJS). Knowing the relevant data lineage objectives, one can determine which characteristics of data lineage are relevant in an operational setting. Based on the required data lineage characteristics one can determine (the type of) data lineage metadata to be collected, and accordingly design data lineage (deployment) architecture to store, query, process, retrieve data lineage metadata (i.e., to decide on the architecture of data lineage metadata management).

How can data lineage tools be deployed?

Based on our literature study and expert interviews, we elucidated that data lineage within the DJS can (or is required to) contribute to data governance, data discovery, data quality management, data change management, and privacy and security mainly. Further, we draw the following conclusions for data lineage in the DJS setting.

- It is necessary to trustfully share data within and across organizational boundaries in the DJS while allowing participating organizations maintain their autonomy and have own business, conceptual and logical data models and ISs. As such, data lineage within the DJS should account for diversity at all levels, namely at technical, logical, conceptual, and business levels.
- Within the DJS, data is shared and processed for not only research and strategic purposes, but also for operational purposes. Thus, the shared data could be at various aggregation levels, i.e., at group and individual levels, which requires having data lineage at coarse-grained and fine-grained levels.
- The end-users of data lineage within the DJS can have different backgrounds with a varying set of data (science) skills and may reside at the beginning, middle or end of data pipeline. Therefore, there should be enough flexibility to serve a wide range of users (so-called, the democratization of data lineage).
- We define five abstraction levels for data lineage namely (a) physical, (b) logical, (c) conceptual, (d) business, and (e) legal and ethical levels. The legal and ethical level is particularly important in the DJS, as the DJS is responsible for overseeing and safeguarding the rule of law in the society.
- In cross organizational settings (like that of the DJS), we foresee that horizontal data lineage can be applied at not only physical level but also at business and conceptual levels. Further, we may need adopting a combined vertical and horizontal data lineage for enabling a horizontal data lineage at the physical level to deal with interoperability issues of data lineage (i.e., having an intertwined horizontal and vertical lineage configuration).

Metadata management in the DJS setting should be scalable and distributed as well as should fit the organizational structure of the DJS. Considering the cross organizational structure of the DJS, we propose considering a federated data lineage metadata management architecture for deploying data lineage. A federated architecture relies on the existing organizational structure, which, therefore, can scale up organically as seen in the case of federated identity management among European universities.

For managing data lineage metadata, automizing the metadata collection process is necessary, considering the high speed and large volume at which data is collected and shared nowadays. For storing data lineage metadata, the type of data repository should be chosen according to the data lineage characteristics needed and the context in which data lineage operates. According to the type of the data repository chosen, one can opt for an appropriate query processing language. System-user interactions can be facilitated in different ways, depending on the technical skills of the users.

Being business driven is necessary as data-driven working has become a common practice nowadays. This requires involving non-technical end-users in the design process as well as the possibility of accommodating new queries by design in data lineage systems. Nevertheless, it is not feasible and efficient to collect and manage a huge amount of data lineage related metadata exhaustively in anticipation that one day some part of it would be useful for answering new business driven queries. To be cost effective, one may choose for collecting a reasonable amount of data lineage

metadata (i.e., collecting metadata coarsely) and should a new query rise, go for either a targeted search (i.e., to carry out a zoom-in search on a need-to-know basis) an/or for an approximate reply if having certain amount of uncertainty in replies is acceptable. It may be needed to deploy natural-language-based user interfaces which are capable of handling both predefined and not predefined queries. Hereby one can make data lineage become business-driven and usable for users with limited technical backgrounds (like business consultants and policymakers).

What are the capabilities (and limitations) of existing data lineage tools?

Determining the relevant capabilities of data lineage tools in each usage context is important as they can be used as criteria for evaluating the existing tools and choosing the one that fits the context the best. Based on the results of the study, we foresee several relevant capabilities for data lineage software tools in the DJS setting. The main capabilities are to have flexibility in being business driven (replying to new and unforeseen data lineage related queries), to have more granular lineage than attribute level (like being cell level), to have interoperability with other tools (from other organizations), and to offer good user experience and useability.

As part of the answer to this research question, we sketch a framework for evaluating data lineage tools. Based on this framework, one should define a set of evaluation criteria by, for example, elucidating the objectives that DJS organizations seek in deploying data lineage, elucidating the desired data lineage characteristics, and defining evaluation criteria accordingly. Subsequently, one can specify several granularity levels per each criterion. These levels can be defined qualitatively, i.e., be assigned a (meaningful) numerical value or a score. Finally, one can merge quantified criteria using numerical methods like averaging and thresholding.

Commercial software tools generally provide a wide range of data lineage functionalities together with other functionalities (like data management and data catalog). As such, they are suitable for large organizations which can afford paying the expenses of such tools and which need using a wide range of functions these tools provide. Open-source tools generally offer a limited subset of data lineage functionalities, are cost free, and are integrate-able with other (open-source) applications. As such, they are suitable for low-budget, small enterprises to use or customize these tools to their data lineage needs. Note that an extensive integration of open-source tools requires inhouse technical skills or extra budget that might not be available in small organizations. Further, the open-source tools might be useful for conducting small-scale experimentations within the DJS setting to gain some hands-on experience about data lineage technology.

Recommendations for follow-up research

Several directions are identified for future research during project execution. In this section we group them in three categories, organized from more practice-oriented research one to more applied research one.

Need for a requirement elicitation study

For choosing a (set of) data lineage tool(s) that can be experimented with (or deployed) in the DJS setting we recognize the need for eliciting the requirements with which the tool(s) should comply. To this end, we foresee the following action points:

- Identifying the typical end-users and their data lineage queries,
- Identifying whether there is a need for adjusting the end-users' queries in the future,
- Defining the desired data lineage objectives/characteristics, and
- Defining the data lineage tool evaluation method by determining the evaluation criteria and how to measure and merge them.

When the intention is to deploy data lineage in the DJS setting, there is a need for a further study of a suitable structural and functional architecture for data lineage deployment in DJS setting.

Another direction for research is to investigate the requirements and ways for mixing vertical and horizontal data lineages at the boundaries of organizations. This requires mapping between data semantics at the borders of collaborating organizations and dealing with uncertainties that may be caused due to this mapping. To this end, gaining hands on experience with data lineage tools in real world cases can be useful.

Need for democratization of data lineage

There is a need for further research on making data lineage become business-driven and usable for users with limited technical backgrounds (like business consultants and policymakers). To this end, investigating methods and tools for natural-language-based user interfaces is a promising direction. Large Language Model (LLMs) can be considered for mapping between natural language texts to formal database queries (e.g., SQL). This direction may require investigating ways to deal with uncertainty in the mapping between natural and formal languages.

Need for effective and efficient data lineage

Reducing the complexity of data lineage and the associated costs is a crucial factor in successful adoption of data lineage technology.

For data lineage related metadata collection, developing automated methods is necessity. For example, the use of LLMs can be investigated for (semi)automatically creating business level metadata (like a report in natural language that describes the technical analyses conducted on the data for business-level end-users) from technical level metadata (e.g., from data query scripts). In this way, the burden of business-level data lineage metadata creation on technical experts can be alleviated.

In conventional lineage it is assumed that users can understand how an output is created by observing the source data and knowing that the data transformation is a sequence of simple operations like filter, join, and aggregation. However, in complex data analysis, like using AI/ML algorithms, more information about data transformation is needed. A question that may arise is which data lineage information should be provided to explain and/or influence the outcomes of very complex data transformations (like LLMs) and how this data lineage information should be managed in a (cost) effective way.

An issue in data origin lineage is how to determine data origins in each setting. We suspect that there might be multiple views with different data origins (specially in cross organizational settings). If this conjecture holds, then lineaging data origin boils down to or, better said, requires lineaging data flows. It is for future research to investigate how to determine data origins (or data destinations) in each operation setting.

1 Introduction

In this section we present the problem statement. We start with describing the problem context in Section 1.1 and (the research objective and) research questions in Section 1.2. Subsequently, we explain the setting of the study, the research methodology, and the scope of the study in Sections 1.3, 1.4, and 1.5, respectively. Finally, we provide the outline of the report in Section 1.6.

1.1 Problem context

Data are currently being generated, collected, analyzed, and distributed at a fast-growing pace. Public organizations, among others, collect a vast amount of data directly as a necessary input for provisioning their services or as a byproduct of their services. Often, organizations share the collected data with partner organizations as needed for accomplishing their public mission and serving citizens. Further, due to availability of data, there is a rising interest and demand to harvest the available data by using statistical analysis, Artificial Intelligence (AI) and Machine Learning (ML) algorithms. These efforts aim at developing advanced data-driven systems and applications to ease the daily lives of citizens, create strategical insights for organizations, provide insight into societal phenomena, and guide policymaking processes.

Also in the justice domain, we witness a trend of applying digital technology and data-driven systems, resulting in various forms of smart justice. Adopting digital technology in these settings can be characterized as digitization (a technical process which aims at automizing existing analogous objects and services), digitalization (a sociotechnical process which aims at integrating digital technology with the social context the technology is used), and digital transformation (a socio-cultural process which aims at transforming the mindset and culture of an organization to ensure that the technology can be deployed as a multiplier of impact).

In the justice domain, Information Systems (ISs) which collect, store, share, and processes data, are often physically distributed, have many loosely coupled subsystems, and are administrated by various organizations (i.e., spreading across many administrative domains). Utilizing data in such settings requires interconnecting various information sources and integrating their information in a trustful and responsible way. Those who share data (like judicial service providers) should entrust data consumers in using the data responsibly and those who use data (like policymakers) should trust data sources in collecting and sharing data responsibly. Establishing these trusts requires, among others, mitigating the security issues of largely distributed ISs, managing the data quality issues of loosely coupled data sources, and dealing with wrongful and/or malicious use of data and algorithms (via, e.g., delivering algorithmic fairness). For example, lack of algorithmic fairness, which leads to various forms of injustice like biased treatment and wrongful discrimination of individuals and groups, can be attributed to the bias embedded in collected data or induced due to inattentive design and use of algorithms. Not handling both so-called hard challenges (like which data a policymaking process is based upon) and soft challenges (like how data is shared and used in a policymaking process) appropriately

and adequately would harm individuals, groups, and society; adversely affecting democracy and basic human rights (like privacy, liberty, autonomy, and dignity).

Given, on the one hand, the importance of data sharing among ISs and, on the other hand, the increased complexity involved in data sharing among (many) organizations, there is a need for establishing an appropriate data ecosystem. This data ecosystem establishment requires solid and effective *data governance* and *data management*. Data governance is defined as "the exercise of authority and control (planning, monitoring and enforcement) over the management of data assets" (Brous, 2020, p. 432). Data governance aims at defining the organizational structures, data owners, policies, rules, processes, business terms, and metrics for the whole lifecycle of data, i.e., collection, storage, use, protection, archiving, and deletion (Everett, 2023). Data management can be seen as the technical implementation of data governance (Everett, 2023). Data governance/management responsibilities include ensuring the quality of data, securing the storage and exchange of data, optimizing the tradeoff among contending values (like data utility and data privacy), and operationalizing the Findable, Accessible, Interoperable and Reusable (FAIR) data principles.

An efficient and effective data governance/management requires, among others, *data lineage* and *data provenance*. Data lineage refers to the process of tracking the flow of data over time, i.e., during the data lifecycle/journey. It uses metadata to maintain a clear understanding of the data origin(s), data changes in its journey, and ultimate data destination(s). As a similar case, data provenance sometimes refers to the sources (i.e., the origins) of the data and its historical changes. Data lineage is seen as an essential instrument for enhancing data trustworthy. It enables, for example, efficient data quality management, data change management, and data lifecycle management. In Box 1.1 we provide two examples to intuitively convey the concept behind (data) lineage.

Box 1.1 A metaphor for and an example of data lineage

A metaphor for data lineage
A famous metaphor used to illustrate the role of data lineage for gaining trust in data is the case of gaining trust in the nutritiousness and healthiness of an apple by knowing how it is cultivated, harvested, transported, stored, distributed, and retailed. To this end, one could keep track of the relevant information (lineage information) in every stage of the apple's lifecycle (i.e., the apple's journey) in the supply chain.

An example of data lineage usage
In the metaphor mentioned above the object of interest is a piece of fruit. Having the same type of assurance is also relevant for any data object. For example, citizens are increasingly subject to a growing quantity of multimedia content (i.e., data) of various types like images, videos, audio recording, and documents. This content is often blended with misinformation (e.g., images generated by Generative AI) or manipulated information (e.g., images processed by Photoshop tool), which makes it difficult for ordinary people (or even professionals) to distinguish between real and fake/manipulated images.

In this case, data lineage can help content consumers to gain trust in media content they encounter on social media, news feeds, etc. The initiative Coalition for Content Provenance and Authenticity (C2PA) – formed by various companies and organizations such as Adobe, BBC, Google, Intel, Publicis Groupe, Microsoft, OpenAI, Sony and Truepic – is developing an open technical standard for data lineage that offers the publishers, creators, and consumers of media content the ability to trace the creation and every edit made to a piece of media throughout its lifecycle.

The C2PA accomplishes its objective by trustfully (i.e., by using cryptographic methods) binding lineage information to any piece of media content. The lineage information, which accompanies the media content through its entire lifecycle – or so-called its journey, can easily be inspected by downstream consumers to get ensured about the origin and any edition made to the media content. For example, when a photographer makes a picture, a C2PA enabled camera records also the lineage information (like the date, the location, and the photographer) and trustfully appends the lineage information to the picture in a temper evident way (i.e., in a way that everyone can notice if someone changes the picture or its lineage information). A website editor who receives the picture can easily inspect the lineage information (by clicking on a small icon on the picture) to see the lineage information and learn about who has made it and when/where. Hereby the editor can gain trust in or infer the level of the originality of the picture. If the editor wants to edit the picture with the Photoshop tool and publish it on his website, the description of the modifications made is appended as new lineage information to the picture as before. In this way, all modifications made to the picture are recorded and appended to the picture as lineage information, accompanying the picture wherever it goes. Downstream consumers (e.g., citizens) can always inspect the lineage information on the picture to see all its history, and thereby (dis)trust the picture at any point in its journey. For more information about the C2PA initiate and approach see (C2PA, 2024) and the explanatory video therein.

1.2 Research objective and questions

The research objective of the study can be specified as investigating how data lineage technology can contribute to data governance and data management within the Dutch justice system. Achieving this objective requires investigating the benefits of data lineage technology and the directions (and challenges) that one might explore (and expect) in deploying it within the Dutch justice system.

This study is a preliminary study towards the abovementioned objective. The research approach can be characterized as explorative, where we seek answers to the following research questions.

1 *What is data lineage?* For answering this question, we will also describe the context (or the data ecosystem) in which data lineage is used.
2 *Which objectives can data lineage contribute to?* For answering this research question, we will give an insight in the potential advantages of data lineage.
3 *How can data lineage tools be deployed?* For answering this question, we will elaborate on typical approaches for and challenges of data lineage deployment.
4 *What are the capabilities (and limitations) of existing data lineage tools?* For answering this question, we will sketch a framework for specifying the relevant data lineage capabilities. Further, as an example and for a limited number of existing data lineage tools, we will give an insight in two capabilities that are of interest for this study.

These research questions will be addressed within the context of Dutch justice system as explained in the following section.

1.3 Organizational setting

The justice domain in The Netherlands includes three legal branches pertaining to criminal law, civil law, and administrative law. Within the Dutch justice domain there are many semi-autonomous organizations that collectively serve implementing the rule of law within the Dutch society. For example, the independent organizations and agencies involved in Dutch criminal justice system, which corresponds to the criminal law branch of the Dutch justice system, include the Police, the Public Prosecution Service (PPS), the courts, the Central Fine Collection Agency, the Custodial Institutions Agency (i.e., prisons), and the Probation Service (PS), see (Netten et al., 2016). The relations between these organizations are often characterized as a linear chain that consists of investigation, prosecution, judgment, and enforcement stages (Tak, 2008). It is linear in the sense that a stage must be concluded before the next stage may begin. However, we note that these relationships are not strictly linear. There are sometimes parallel relations, for example the PS or Dutch Institute for Forensic Psychiatry can work on a criminal case in parallel with the Police and the PPS. Furthermore, the relations among Dutch criminal law organizations might be in loops, for example, in wrong convictions the supreme court can decide a case to be heard again by a court of appeal (or even by a new court). Consequently, such cases go through the entire process again. Note that the number of organizations in the Dutch administrative and civil justice systems, which correspond to the administrative law and civil law branches of the Dutch justice system, respectively, is smaller than that in the Dutch criminal justice system. Thus, the networking aspect of Dutch administrative and civil justice systems can be simpler than that of the Dutch criminal justice system.

Lampoltshammer et al. (2017) use the term *justice system* to refer to the bodies in the apparatus of law, involved in creating data, ranging from legislative texts to judicial decisions. As such, the scope of the justice system is broader than courts and court procedures. Within the *Dutch Justice System* (DJS), we conclude, the data is generally gathered by various organizations forming a *network* of semi-autonomous organizations that collectively implement the rule of law within the Dutch society.

1.4 Methodology

For this study we have conducted a critical literature review (Paré et al., 2015), where several selected information sources are analyzed, and a reflection is done on the existing concepts, models, and approaches. The information sources used are not only from scholarly literature, but also from gray literature like commercial websites, whitepapers, and weblogs. The latter is because many vendors and system developers are dominantly active in the field of data lineage, who introduce many innovative concepts, features, and tools to the domain.

In addition to literature study, we have conducted four semi-structured interviews with experts from different organizations within the ministry to gain insight in ongoing data lineage related (R&D) activities within the ministry as well as in eliciting the needs and visions of those experts involved in data governance/management within their organizations. Further, we organized two expert focus groups with data stewards and data management experts to present our intermediary results and get early feedback. The four interviewees and the two focus groups were chosen based on the expertise and availability of the participants rather than their representativeness. This choice is motivated by the nature of the study in being preliminary and explorative.

1.5 Study scope

In this contribution we intend to provide an overview of some relevant aspects that can be considered for deploying data lineage technology in organizational settings such as that of the DJS. As such, we do not intend to design or prescribe a solution for data lineage deployment in this contribution.

1.6 Outline

The organization of the report is as follows.

- In Section 2 we provide some background information about data lineage, its related concepts, and its main characteristics.
- In Section 3 we present a model for specifying the scope of data lineage, as definition of data lineage, and the main objectives for deploying data lineage.
- In Section 4 we introduce a functional architecture for data lineage to base the follow up discussions on.
- In Section 5 we elaborate on the characteristics of two boundary strategies for deploying data lineage like systems within and across organizations.
- In Section 6 we provide a high-level evaluation of several software tools offering data lineage functionality.
- In Section 7 we draw our conclusions and describe directions for future research and developments.

2 Data lineage

In this section we aim at answering the first research question: "What is data lineage?" For answering this question, in Sections 2.1 we explain the evolutionary path of data documentation in which data lineage fits nowadays. Subsequently, we provide a visual view on what data lineage delivers and a definition of data lineage in Sections 2.2 and 2.3, respectively. We list the main characteristics of data lineage in Section 2.4. Finally, we recapture the main outcomes of the whole section in Section 2.5.

2.1 Metadata for data utilization

As mentioned in the introduction section, data is currently being generated, collected, analyzed, and distributed at a fast-growing pace. Further, in the justice domain, the ISs which collect, store, share and processes data, are often physically distributed, have many loosely coupled subsystems, are administrated by various organizations. As such, the data is often hidden in silos in this setting, which makes it difficult for data users to access and utilize the existing data trustfully. Consequently, users (or researchers) need to invest a substantial amount of time in gathering data they need for answering a research, business, or data management related query.

Utilizing the existing data in such an ecosystem requires interconnecting various data sources and integrating their data in a responsible way. This, in turn, asks for making data available for a legitimate use, for everybody or every party interested/authorized, and at an appropriate level of detail; to name a few. Nowadays it is desired and/or necessary to make data available not only to data specialists, but also to so-called data citizens. The term data citizens may refer to individuals who are not data specialists but use data on a regular basis to fulfill their daily jobs or may refer to the public being interested in overseeing government agencies and their policies/actions. This type of involvement can be associated with the trending topic of *data democratization*, which has gained in popularity among practitioners but has very limited scientific conceptualizations (Lefebvre et al., 2021). Data democratization refers to the process that removes obstacles to data exploration and data sharing to empower a group of users spanning beyond the established data experts to access and use data (Labadie et al. 2020).

To encourage data usage (for, for instance, government transparency purposes) and/or to enhance the reusability of data (for, for example, saving data gathering costs), many scholars have proposed the so-called FAIR Guiding Principles (Wilkinson et al., 2016). These principles encourage enhancing the Findability, Accessibility, Interoperability, and Reuse (FAIR) of data assets to ease the efforts and alleviate the costs of data gathering processes. According to the FAIR principles, one should be able to figure out what data exists and where (i.e., data being findable), how the found data can be made available (i.e., data being accessible), how the available data can be formatted appropriately (i.e., data being interoperable), and how the commonly formatted data can be enriched with contextual information in order to be understandable (i.e., data being reusable), see (Labadie et al., 2020). Although realizing the FAIR principles is an enormous challenge in practice – think of, for example, making data automatically interoperable (Choenni et al., 2022) – they guide

the initiatives in enterprises and organizations to (re)use and share data within and cross organizational boundaries.

Making data reusable, in general, and realizing the FAIR principles, in particular, ask for, among others, a better *data documentation*. Technically, data documentation is possible via metadata (management). *Metadata*, which is often defined as data about data in its simplest form (Roszkiewicz, 2010), aims "at facilitating access, management and sharing of large sets of structured and/or unstructured data" (Kerhervé & Gerbé, 1997). This objective aligns well with the objectives of the FAIR principles.

Labadie et al. (2020) sketch the evolution of different concepts related to data documentation, spanning from concepts *field names* in 1960s, to *table definitions* in 1970s, and to *technical data dictionaries* in 1980s. The latter refer to system-specific data dictionaries, providing a technical documentation of database tables. With the emergence of enterprise resource planning systems, business process integration was added to data documentation. It became necessary to link data architecture and data integration with business needs so that via analyzing business data, which was stored in data warehouses, one could facilitate business decision-making processes. Linking technical aspects with business needs increased the complexity of data documentation, leading to creation of:

- *Business data dictionaries*, which extended technical data dictionaries with documentation of business data in the form of metadata.
- *Business glossaries*, which were introduced to define the semantics of business terms for a correct interpretation of data for different use cases.
- *Data catalogs*, which can be seen as the next step in the evolution of data documentation concepts and data provisioning.

As the newest trend of data documentation, despite gaining popularity in practice-oriented community, the concept of data catalog has not been defined universally. Nevertheless, there are several definitions of data catalog mentioned in the literature. For example, Zaidi et al. (2017) define a data catalog as maintaining "an inventory of data assets through the discovery, description, and organization of datasets. The catalog provides context to enable … data consumers to find and understand a relevant dataset for the purpose of extracting business value". This definition, however, does not elaborate upon how data access is controlled (Labadie et al., 2020). Generally, data catalog can contribute to the transparency of data, trustworthiness of data, readiness of data for use by various data consumer types (which are specified based on the level of data and data science literacy of data consumers), compliance of data with the FAIR principles, and democratization of data (Labadie et al., 2020).

Data catalog, in practice, can be seen as a set of functionalities offered in a metadata management platform, being "the single place for all users to find, understand, and govern data" (QLIK-2, n.d.). We think, it can best be described by the functionalities that data catalog tools may or are supposed to provide typically. The key functional components of data catalog tools, see (Labadie et al. 2020; QLIK-n.d.), are:

- *Data onboarding*: To document and profile the exact content, structure, and quality of data of a source by, for example, generating rich metadata about (new) datasets and enabling automated discovery of those datasets.
- *Data cataloging*: To enrich the catalog by (automatically) identifying and describing every relevant aspect of the data and data management process. To this end, AI

based metadata management (collection, tagging, and semantic inference) can be exploited.
- *Data searching*: To provide powerful and multifaceted search capabilities by keywords, facets, and business terms as well as by natural language search inquiries for business users.
- *Data glossary*: To develop and share a data glossary (or dictionary) which defines the business terms and concepts used in the organization, thus giving consistent business context across multiple tools.
- *Data consumption*: To enable easy and secure consumption of data by all types of users by supporting, for example, one-time data exports, recurring and automated data publishing, simple integration with workflow schedulers, built-in event logging and notifications, and automatic obfuscation of sensitive fields.
- *Data lineage*: To enhance trust in data by giving full visibility into the origin of data, how it is transformed, and where it has moved over time.

Note that data catalogs are commonly used together with data marketplaces, but they have several differences (Informatica, 2024b). The main difference is that the former focuses more on inventory of data assets and information about those assets, and the latter focuses more on sharing and promoting curated data enhanced with contextual information.

Nowadays data lineage is seen as part of data catalog to simplify end-users' access to lineage information, it's often incorporated into data catalogs (Stedman & Loshin, 2022), but there is a separate historical path for its development. See (Bose & Frew, 2005) for a view on the historical development of data lineage concept in business, scientific, and organizational settings. From this point on, we will focus on the last functionality of data catalogs, i.e., data lineage, and will describe how it is shaped and could be supported in practice.

2.2 A visual view on data lineage

As the dependency of business on data increases, we need to know about the *data journey* from its origins, which can be where the data is created (e.g., by users of social media platform) or first stored (e.g., a database, file directory, cloud data storage, and big data platform of an organization), to its consumers (e.g., a data scientist, data analyst, and data auditor) to trust the data. A way to provide visibility into how data moves throughout an organization (Segment, 2023) or a set of organizations is to use a transformation graph (Ikeda & Widom, 2009). As shown in Figure 2.1, a transformation graph consists of some input datasets (denoted by I_1, I_2, …, I_K), which are fed into some transformations (denoted by T_1, T_2, …, T_N) according to a graph model, to yield some output datasets or documents (denoted by O_1, O_2, …, O_M). Also shown in Figure 2.1 there are intermediary outputs, each of which can be seen as input datasets to their downstream data transformations and outputs.

Figure 2.1 An illustrative transformation graph, for visualizing data lineage

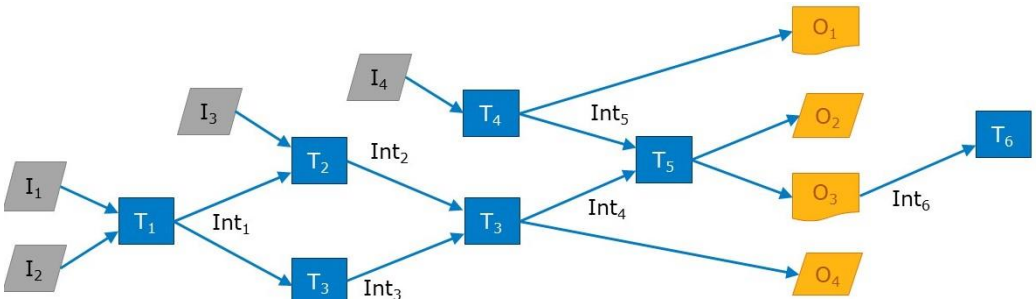
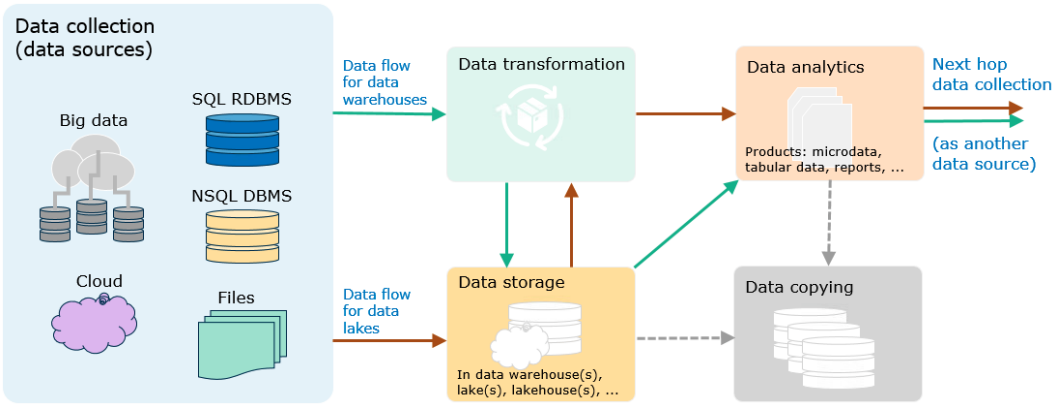


Figure 2.2 illustrates a typical step of a data journey, where two typical data flows, pertained to data storage flows in data warehouses and data lakes, are shown. In a data warehouse, as shown in Figure 2.2, the data is transformed to a structured format with enough quality before being stored in a data warehouse. Data consumers can query the neatly stored data for analytical purposes, often using relational query languages like the Structural Query Language (SQL). In a data lake, as also shown in Figure 2.2, the data is stored in large volumes in its native format (e.g., in structured and unstructured formats) in a data lake storage. Data consumers need to transform the raw data stored in a data lake to a desired format before using it for their analytical purposes. Shown in Figure 2.2 is also a data copying process that may (often) remain hidden for data management and data maintenance, resulting in stalled and outdated data which suffer from serious data quality issues. Data lineage mainly contributes to establishing trust in data by maintaining an oversight of such data journeys.

Figure 2.2 Example data flows between various system components

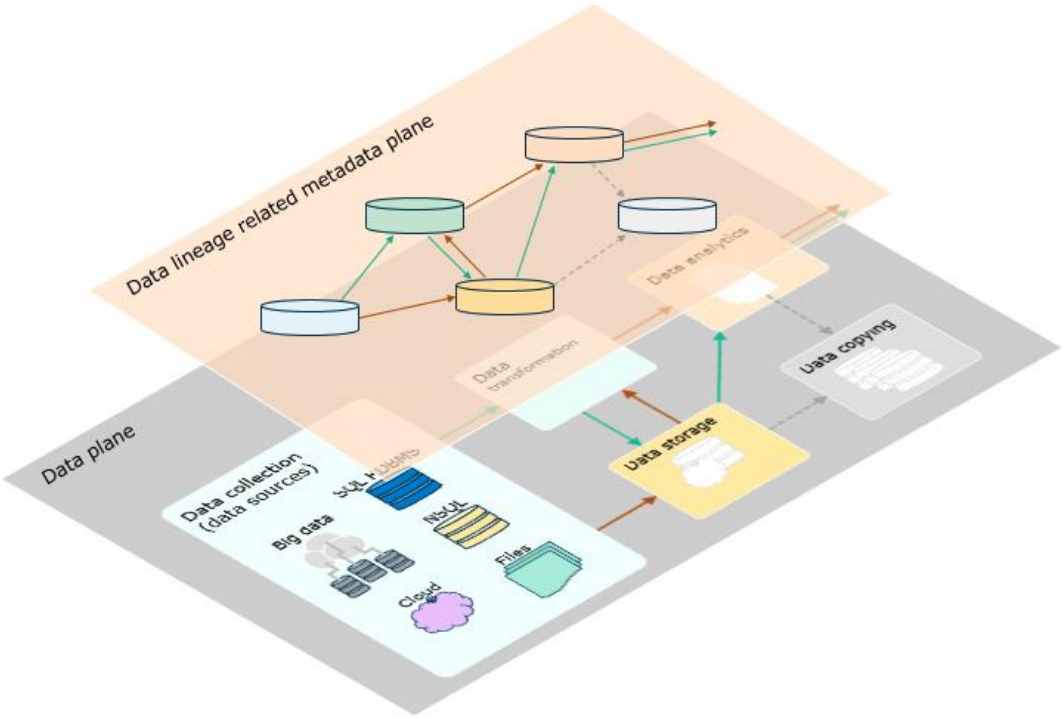


2.3 Data lineage definition

Data lineage is based on and relies on creating and managing (i.e., collecting, sharing, storing, and transforming) the metadata that is suitable/relevant for the type of the oversight sought (or to be provided) by data lineage. The types of the oversight sought by data lineage, in turn, determine the data linage characteristics (to be described in Section 2.4). In its basic form, the data lineage metadata management concerns managing the metadata about data flows and its transformations as well as processing the metadata for answering queries made about data lineage. As such, the

data lineage related metadata is the enabling component of data lineage. Figure 2.3 illustrates data lineage related metadata by a plane on top of the data plane that captures and manages the metadata about how the data stored and exchanged in the ISs shown in Figure 2.2.

Figure 2.3 An illustration of the metadata plane for data lineage



Although there is no universal definition for data lineage, there are many attempts to conceptualize it. A common definition of data lineage, which stems from the academic community, is the following. "Lineage, or provenance, in its most general form describes where data came from, how it was derived, and how it was updated over time" (Ikeda & Widom, 2009). A few examples of data lineage from gray literature are given in Appendix 1. A scan of these definitions reveals that nowadays, especially among practitioners, data lineage includes some other aspects than those given in the definition of (Ikeda & Widom, 2009). Indicated by bold letters in the table in Appendix 1; these other aspects include data lineage being a process, expressing when, by whom, why data transformations are done, indicating where data being stored (which is relevant for stalled data), recording who has accessed data, and linking business concepts and technical objects.

Linking business concepts and technical objects has become prominent in the last decade as organizations and enterprises seek to monetize data for shaping their strategies/services and/or business benefits. This linking within the data lineage field is referred to as *vertical data lineage* as it aims at establishing relationships between objects at multiple abstraction levels, ranging from business concepts to technical objects (i.e., data items). On the contrary, the traditional definitions of data lineage, e.g., that of (Ikeda & Widom, 2009), emphasize the spatial characteristics of data flows and transformations of data objects along the technical system components on the data journey paths. This traditional perspective on data lineage is referred to as

horizontal data lineage. We will elaborate more on vertical data lineage and horizonal data lineage, and their relations to technical and business lineage concepts in Sections 2.4.8, 2.4.9 and 4.3.2.

Based on the discussions above, we adopt the following definition of data lineage.

Box 2.1 Definition of data lineage

Data lineage is the description of data movements and transformations at various abstraction levels along data journey paths. The description encompasses those aspects that are of interest in an application context like how (i.e., by whom, when, where, which, etc.) data objects are processed (i.e., created, collected, stored, accessed, transformed, transmitted, etc.) and how they are related to high-level data concepts.

In the definition given above the term "at various abstraction levels" characterizes the business and technical aspects of data lineage, the term "how … data objects … are related to high-level concepts" captures the vertical aspect of data lineage, and the term "along data journey paths" captures the horizontal aspect of data lineage. The convolution among vertical vs horizontal data lineage and business vs technical data lineage is described in more detail in Sections 2.4.8 and 2.4.9.

2.4 Data lineage characteristics

Data lineage is characterized from various aspects both in the practice and the literature. In Sections 2.4.1 through 2.4.9 we will present several characteristics of data lineage and describe the types associated with each of them.

2.4.1 Data origin vs data flow data lineage

A key feature of data lineage is to lineage (i.e., to track and/or trace) data flows, which includes tracking/tracing the data paths from its origin(s) to its destination(s). As a special case, the scope of data lineage can be limited to following only the data origin(s). Some call this special case (i.e., tracking/tracing the origin datasets of data and its historical record like who created it and who/how changed it) as data provenance (Atlan, 2024). This notion of data provenance does not include tracking/tracing the intermediary points on the paths of data flows nor transformations made along these paths. As such, data provenance and data lineage differ in depth and focus (Atlan, 2024). Like in conventional lineage, see (Yamada, et al., 2023), this view on data provenance assumes that users can understand how an output is created only by observing the origins of the data. This assumption holds when data transformations are consequences of simple operations like filter, join, and aggregation.

Note that there are some sources that consider data provenance synonymous to data lineage like (Bertino et al., 2014; Ikeda & Widom, 2009). In this case, they define the scope of data provenance widely as to follow data flows (including data origins) as well as data transformations along data path(s).

Specifying data origins might not be straightforward always, especially when data is propagated in a chain or network of systems/organizations. In such cases, depending

on the data abstraction level and the operational context, a dataset might be considered as both an input dataset (i.e., as data origin) or an intermediary dataset for the downstream datasets.

2.4.2 *Where vs how data lineage*

The data lineage definition given in (Ikeda & Widom, 2009) encompasses two key characteristics of data lineage namely to specify data flows and to specify data transformations. To illustrate these characteristics, let's consider the *transformation graph* (Ikeda & Widom, 2009) shown in Figure 2.1, where output O_m can be either an output dataset or a tuple of an output dataset. For output O_m, Ikeda and Widom (2009) specify data lineage further into two types:

- *Where lineage* to specify which inputs from the set of inputs {In} have contributed to the output Om, and
- *How lineage* to specify the data transformations made on those inputs.

The distinction described above is from an output view. We suspect a similar distinction can be made from an input view, leading to where-to lineage and how-to lineage types. Suggesting this input view can be justified when considering the backward vs forward data lineages (see Section 2.4.6) and tracing vs tracking data lineages (see Section 2.4.7).

2.4.3 *Data transformation types*

Which data transformations data lineage needs to keep record of depends on the type of the queries that are made to the data lineage (Ikeda & Widom, 2009). The types of data transformations can vary from conventional ones, where for instance a sequence of simple operations is carried out on relational datasets (like performing filter, join and aggregation functions), to advanced ones (like applying User Defined Functions, UDFs), and to highly advanced ones (like training Large Language Models, LLMs). In conventional operations, users can understand how an output is created by observing the source data and the transformations made on the data, but this is not possible in (highly) advanced transformations.

The (way that the) data-lineage-related-metadata that should be collected depends on the computational model (i.e., transformations) applied to data. Bose and Frew (2005) recognize five computational models (or data processing types), as listed below. For each of these, we provide a short description of example metadata that can be collected without going into details, to indicate (the way of collecting) the metadata differs for these computational models.

- *Program-based processing*, which refers to programming languages such as C/C++, Fortran, and Java that rely on statically typed and compiled instructions. In this category, a retrospective view on data transformations can be provided by registering user-supplied program instructions.
- *Script-based processing*, which refers to programming languages such as Python, Perl, and MATLAB that rely on dynamically typed, general-purpose interpreted scripts. In this category, a retrospective view on data transformations can be provided by registering user-supplied scripts.

Other script-like processing types with being subject to more constraints are:

- *Query-based processing*, which is based on users submitting queries to a DataBase Management System (DBMS). In this category, a retrospective view on data

transformations can be provided by using a DBMS that registers all user-defined processing algorithms and the corresponding additional functions that deliver weak inversion and verifications for resolving data lineage queries on the fly.
- *Service-based processing*, which is based on a network of web servers or grid nodes that are invocated for transforming data in a distributed multi-user and multi-institutional environment. In this category, a retrospective view on data transformations can be provided by, for example, querying the service invocation system.
- *WorkFlow Management System (WFMS)-based*, which relies on instructions expressed in a specific process definition language and the registration or wrapping of external code in so-called task objects. In this category, a retrospective view on data transformations can be provided by registering the control flow between data and the task objects which are abstractions of data transformations.

We note that in the DJS setting we are often concerned with script-based, query-based, service-based, and WFMS-based models.

2.4.4 *Coarse-grained vs fine-grained data lineage*

Data lineage can be carried out at a coarse-grained level or a fine-grained level (Ikeda and Widem, 2009). These options specify the so-called *what* aspect of data lineage. Coarse-grained, also called *schema-level*, lineage aims at answering data lineage queries at dataset level, like which input datasets produced (or contributed to producing) a given output dataset. Fine-grained, also called *instance-level,* lineage aims at answering data lineage queries at a tuple level, i.e., treating individual items within datasets individually. For example, which tuples in inputs datasets were responsible for producing a tuple in an output dataset.

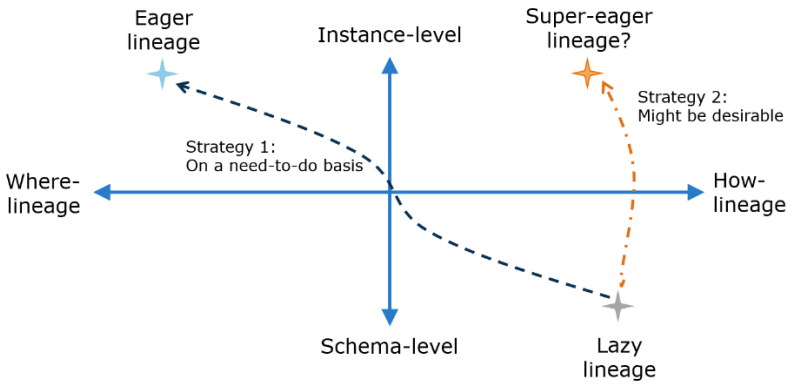
It is worthwhile to note that fine-grained lineage is much more costly than coarse-grained lineage. To alleviate these costs, one can use coarse-grained lineage information for supporting fine-grained lineage in some cases (Cui & Widom, 2003). For more information about this strategy, see also strategy 1 explained in Section 2.4.5. Further, for some transformations, one might be able to automatically derive the lineage for every item in the output dataset by inspecting the whole input dataset. Thus, for such transformations, just a coarse-grained lineage needs to be stored (i.e., a coarse-grained lineage being sufficient to deliver a fine-grained data lineage).

2.4.5 *Lazy vs eager data lineage*

In terms of how to register and retrieve data-lineage related metadata, Ikeda and Widom (2009) identify two strategies namely eager lineage and lazy lineage for instance-level lineage. For the eager lineage one registers instance-level where-lineage immediately after every change (i.e., after conducting transformations). This operation mode is illustrated by the operation point on the top-left corner in Figure 2.4.

Often, an eager lineage can be a heavy operation in practice. Instead, one can deploy a lazy lineage by (a) storing schema-level lineage in sufficient detail after having any change or transformation (see the operation point on the bottom-right corner in Figure 2.4), and (b) using this information to derive instance-level lineage on-the-fly whenever needed (as denoted by dotted arrow labeled by strategy 1 in Figure 2.4).

Figure 2.4 An illustration of eager vs lazy lineage



We suspect that, the lazy lineage can sometimes also deliver instance-level how lineage whenever a need for it arises as denoted by dotted arrow labeled by strategy 2 in Figure 2.4. Due to fine-grained characteristic of Strategy 2, we call it a *super-eager lineage*. An example of this strategy that results in instance-level how lineage is reported in (Yamada et al., 2023). The authors apply this example strategy to classification-based decision-support systems, which can be seen as user defined functions. The strategy can help explainability and interpretability of machine learning based decision-support systems and facilitate algorithmic recourse (Karimi et al., 2022) and algorithmic contestability (Alfrink et al., 2023).

2.4.6 Backward vs forward data lineage

Investigating the application areas of data lineage closely, summarized in Section 3.2, reveals that data lineage related queries can be done by various stakeholders along the data pipeline. Some stakeholders reside dominantly at the beginning of the pipeline (like data subjects), some at the end of the pipeline (like data scientists), and some in between (like data engineers and stewards). As such, location-wise, various views on data lineage are possible like:

- *Output viewpoint:* Given an output point at the data pipeline, one may query: Which inputs did contribute to the output point? How were the inputs manipulated to produce the output point?
- *Input viewpoint:* Given an input point at the data pipeline, one may query: Which outputs did the input point contribute to? How was the input point manipulated to result in those outputs?
- *Intermediary viewpoint:* Given an intermediary point at the data pipeline, one may query: Which inputs did contribute to the intermediary point? How were the inputs manipulated to produce the intermediary point? Which outputs did the intermediary point contribute to? How was the intermediary point manipulated to result in those outputs?

According to (Qlik, 2024), backward data lineage looks at the data journey from its downstream end-users' perspective and back-dates it to its source (thus, being related to the output viewpoint). Forward data lineage begins at the source of the data journey and follows it to the end (thus, being related to the input viewpoint).

2.4.7 *Tracing vs tracking data lineage*

In data lineage, tracking is associated with the collection of the lineage-related metadata and tracing is the investigative process that provides replies to data lineage related queries (for tracking see Cui & Widom, 2003). This view looks like the view in supply chain which distinguishes tracking and tracing based on the direction and the point in time that the journey of an object is observed. "To track an object, you follow the path forwards from the starting point to wherever the object currently is, whereas, to trace an object, you follow the path backwards from its current point to where it began" (Odette, 2024).

In (Xie, 2022) data lineage is classified as a tracer or a tracker, depending on how much is known about data transformations, i.e., in being black-boxes or white-boxes. In *lineage tracking* one already knows the applied transformations fully (being a white box). This is the case where, for example, we already have the source code that generated the output data, and we typically focus on efficiently representing and storing the lineage information in metadata. This can be done by, for example, augmenting (big) data computation platforms via recording the lineage information during data processing. Such an augmentation should be with little overhead, like the approach adopted in (Tang et al., 2019). In *lineage tracing* one does not know data transformations fully (being a black box). Therefore, data lineage is inferred from the inputs and outputs. In practice, however, one applies both tracking and tracing approaches, depending on how much is known about the transformations, focusing on both doing inference and managing metadata.

2.4.8 *Technical vs business data lineage*

Both terms of technical data lineage and business data lineage are used to refer to overseeing and monitoring data flows and transformations between IT systems. The technical data lineage provides a physical view on stored data, data transformations, and data flows. In other words, it provides an oversight about which ISs within an organization or across organizations hold (a copy of) the data, lie on the path(s) of the data, and transform the data in certain ways.

The business data lineage offers a logical view on how data flows and data transformations are connected to business representation of data, see (Karkošková & Novotný, 2021) and the references therein. This implies that, as we understand, business data lineage links higher level concepts (like legal and business concepts) to lower-level data objects and transformations. Further, we note that, like data objects, business concepts may change across domains as data flows between the ISs of those domains. Thus, in principle, both technical and business lineage may spread spatially along the routes of data flows.

2.4.9 *Horizontal vs vertical data lineage*

Technical and business data lineage are related to *vertical* and *horizontal* data lineage. The way that business data lineage is defined in (Karkošková & Novotný, 2021), i.e., linking business concepts with technical objects, is referred to as *vertical data lineage* as it aims at establishing relationships between objects at multiple abstraction levels, ranging from business concepts to technical objects (i.e., data items).

The traditional definitions of data lineage, e.g., that of (Ikeda & Widom, 2009), emphasize the spatial characteristics of data flows and transformations of data objects along the ISs on the data journey paths. This traditional perspective on data lineage is referred to as horizontal data lineage. As noted at the end of Section 2.4.8, a horizontal lineage can occur also at a business level by linking business level data related concepts along the paths of data flows and transformation. As such horizontal data lineage can be relevant to both business data lineage and technical data lineage.

The convolution among vertical vs horizontal data lineage and technical vs business data lineage are illustrated in Table 2.1. In Section 4.3.2, we will elaborate more on vertical vs horizontal data lineage (and their relations to business vs technical lineage) from an architectural perspective.

Table 2.1 Convolution of vertical/horizontal and technical/business lineages

Explanation	Horizontal lineage	Vertical lineage
Business lineage	Lineage among business/legal concepts across administrative domains (e.g., among organizations)	Lineage among business/legal concepts and data objects/transformations within an administrative domain (an organization)
Technical lineage	Lineage among data objects/ transformations across administrative domains (among organizations), as defined in (Ikeda & Widom, 2009)	

2.5 Concluding remarks

Data lineage is a key functionality of data cataloging that contributes to gaining trust in data and in responsible data sharing. Based on some existing definitions and the body of knowledge on data lineage, we coined a definition of data lineage that conceptualizes not only the physical distribution of data related objects (like data origins, flows and transformations), but also the semantical distribution of the related concepts (like the legal, business, and organizational terms) that relate or apply to those data objects.

Data lineage relies on deriving and managing metadata that is relevant for the aimed data lineage purpose. Data lineage can be characterized from various aspects, which we have elaborated upon many of them in this section. Some of these characteristics are independent from each other like where/how data lineage vs coarse-grained/fine-grained lineage, while some of them are (closely) related like technical/business data lineage vs horizontal/vertical data lineage. Nevertheless, these data lineage characteristics specify the space in which a data lineage solution can be designed, chosen, and/or deployed. In the following section, we will present various objectives that one may seek when deploying data lineage in an organizational setting. Realizing any subset of those objectives requires having an oversight of typical data lineage characteristics and specifying those characteristics that are applicable to the aimed objective(s).

Concerning the data lineage characteristics presented in this section, we provide a few comments in the following, which were raised during various stages of our study.

- An issue in data origin lineage is to determine data origins in each setting. We suspect that there might exist multiple views on data origins which may differ

(specially in cross organizational settings). If this conjecture holds, then lineaging data origin boils down to or, better said, requires lineaging data flows.

- The type of the queries made to data lineage (Ikeda & Widom, 2009) as well as the stakeholders that pose these queries determine which characteristics of data lineage are relevant in an operational setting. For example, assume that we are interested in the how-lineage for an output tuple, which is affected by a subset of data items in the input dataset(s) for the data transformation in mind. In these cases, an output-view, instance-level how-lineage is relevant. In this case, however, an output-view instance-level where-lineage would be sufficient if simple relational transformations are carried out (i.e.., those transformation that how-lineage can be derived from where-lineage).
- To determine which characteristics of data lineage are relevant in a setting, determining the (types of) data lineage queries is the first step, followed by mapping these to the type of data lineage metadata needed, and the strategy for data-lineage metadata collection and retrieval.
- The cost and complexity of metadata management for data lineage can be high, especially if one needs a fine-grained data lineage and/or if one is concerned with (highly) advanced data transformations (like UDFs or training LLMs). To alleviate the cost and complexity burden, one may opt for (developing) hybrid solutions where proactive data lineage related metadata management is carried out coarsely and whenever a fine-grained data lineage is needed, an on-the-fly targeted query is activated reactively (i.e., on a need-to-know basis), as sketched in Section 2.4.5.

3 Application areas of data lineage

In this section, we aim at answering the second research question: "Which objectives can data lineage contribute to?" For answering this question, we start with some use-cases of data lineage in Section 3.1. Subsequently, in Section 3.2 we state the contributions of data lineage to (various aspects of) gaining trust in data. In Section 3.3 we provide several typical queries to which data lineage can answer. Finally, we recapture the main outcomes of the whole section in Section 3.4.

3.1 Data lineage use-cases

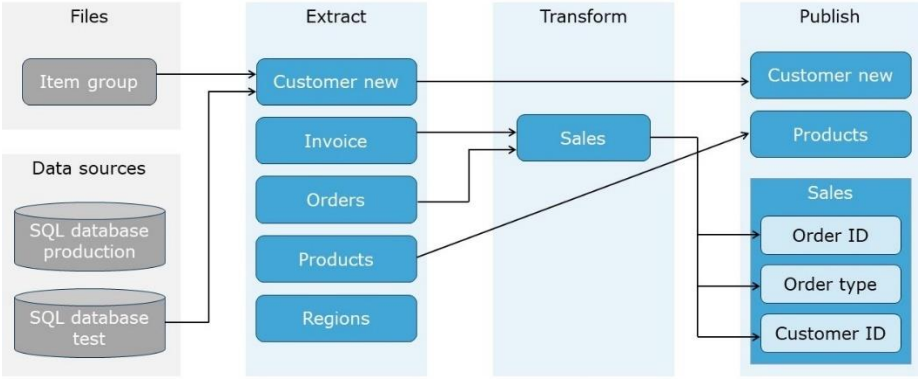
In order to get some feeling about how data lineage can be used in practice, we present a few examples from the research and practice communities in Sections 3.1.1 and 3.1.2, respectively.

3.1.1 From the literature

In this section we present three different use-cases from (gray) literature for data lineage to illustrate the wide range of its applicability.

Figure3.1, which is adopted with adaption from (Qlik, 2024), shows a view of a modern data lineage tool that provides a visualization of the sources and the journey routes of data (Qlik, 2024). The figure visualizes a combination of a conceptual data model and a logical data model, which together establish a view on the (data processing) entities, their attributes, and their relationships. Shown entities in Figure3.1 are files and data sources (marked as "SQL database production" and "SQL database test" in the figure), and their transformations during three data processing steps of Extract, Transform and Publish. Shown attributes in Figure3.1 are Order ID, Order Type, Customer ID, etc. The relationships between these entities are shown by directed edges in Figure3.1. Such a conceptual data model, which is a technology-agonistic and high-level representation of data journey, aims at creating a shared understanding of data lineage needed within an organizational or business setting.

Figure3.1 An illustration of data lineage at a conceptual data model level

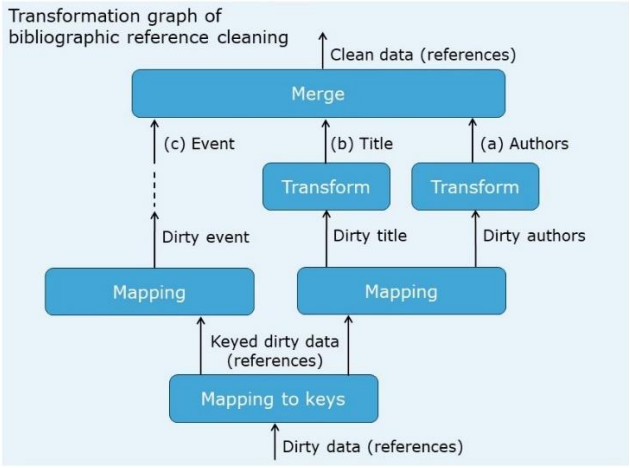


Another example of data lineage usage is described in (Galhardas et al., 2001), where it is used to facilitate data cleaning. This example shows that data lineage can be of

use within one single program to support end-users with exception handling. Data cleaning, a well-known data preprocessing activity in decision support systems and data warehouses, deals with resolving inconsistencies and errors from original datasets. The authors aim at applying data cleaning to unstructured data. As an example, they consider transforming unstructured references to articles in online documents (e.g., reports, whitepapers, and weblogs) to well-structured and searchable bibliographic references (i.e., structured with respect to, for example, authors' names, paper title, publication date and place).

Cleaning very dirty data can be difficult to automate with a fixed set of transformations. In the abovementioned example, the publication year of a paper might be different in two documents and there might be no obvious criteria to decide which date to use. Therefore, the user must be consulted explicitly to resolve non-automize-able inconsistencies and exceptions. Galhardas et al. (2001) specify a data cleaning application as a graph of high-level transformations. For example, Figure 3.2 shows a simplified transformation graph of that in (Galhardas et al., 2001) for dirty bibliographical reference cleaning. The transformation graph consists of data flows for structuring dirty data for (a) authors' names, (b) publication title, and (c) publication event (like its type, year, URL). The data flow graph representing the cleaning strategy contains various nodes as logical operators for mapping, transforming (via mapping, matching, and clustering), and merging. During data cleaning operations, various exceptions can arise, for example, violation of data output constraints. Data lineage can be used as a feature for addressing these exceptions by allowing a user to "(i) inspect the set of exceptional tuples, (ii) backtrack in the data flow graph and discover how the exceptional tuples were generated, and (iii) modify attribute values of tuples, insert or delete a tuple of any relation of the data flow graph in order to remedy the exceptions" (Galhardas et al., 2001).

Figure 3.2 An example of using data lineage within a data cleaning program



The last example of data lineage usage described in this section is from (Yamada, et al., 2023) that considers data lineage for enhancing the explainability and interpretability of the outputs of complex decision support systems. The authors argue that conventional lineage is suitable for a sequence of simple data operators (e.g., filter, join, and aggregation) where one can understand why an output was derived only by observing the source data. However, for complex data analysis (like those based on machine learning), one needs some information about the reasoning basis (for AI/ML processing) in addition to source data. For such cases, the authors propose,

as they call, an augmented lineage which expresses complex data analysis flows by using relational operators combined with UDFs. UDFs, in this case, represent the invocations of AI/ML models within the data analysis. Their method takes UDFs into account to derive the augmented lineage for arbitrarily chosen tuples among the analysis results. The information about the reasoning basis, which helps users understand the computation, can for example be "which region of the content data (e.g., images or videos) the analytical model emphasizes" and "which attributes much affect the AI/ML model decision" (Yamada, et al., 2023).

3.1.2 From the practice

Data lineage is gaining popularity in practice. According to (Foote, 2023), data lineage is used in large organizations and businesses (like Airbnb, Netflix, UBS, Slack, and Postman) mainly due to being rather new, regardless of being at the expensive side. These large businesses, which need reliable data for good decision-making, see benefits of using data lineage such as providing visibility needed to effectively deal with data migrations, making system updates, and correcting errors. Two real-world examples mentioned in (Foote, 2023) are mentioned below.

- In its response to a data breach in September 2018 which affected 380,000 customers' credit cards and personal information, British Airways used data lineage to trace the breach to a malicious script on their website. This enabled them to identify and repair the issue quickly.
- Air France faced problems with data processing and data segregation as, due to their business growth, they were processing over 2.5 million new visitors on their website. This growth made keeping track of all data from various databases very difficult. Using a new data lineage system enabled Air France to deliver personalized advertising and real-time updates without breaching GDPR regulations.

Some other applications of data lineage by companies and organizations are mentioned in (KnowledgeNile, n.d.).

Disclaimer: Note that the material in this section stem from gray literature (personal and/or commercial websites) that are not peer reviewed. As such, we cannot verify or ensure the validity of the claims made about data lineage benefits in these websites.

3.2 On relevancy of data lineage

In this section we present those desired objectives (or system functions) to which data lineage can *contribute*[1]. In other words, this section explains the usage areas of data lineage and, as such, the section sheds light on the societal relevancy of data lineage at large.

3.2.1 To gain trust in data

Primarily, data lineage enhances the trust in data in various ways (see the metaphor and example use-case in Box 1.1). The main contributions of data lineage that act as factors of enhancing the trust in data and its handling are:

- knowing who has produced data and how it is changed, which constitute the original motivations behind data lineage adoption,

[1] Note that we use the term "contribute" to indicate that data lineage is not the only element for realizing the functionalities mentioned in this section.

- showing regulatory compliance through enabling the data audit,
- enabling data governance,
- facilitating data migration,
- discovering and mitigating privacy and security risks,
- enabling algorithmic explainability, algorithmic interpretability, algorithmic recourse, and algorithmic contestability, and
- assuring how ML and data analytics models are trained and constructed.

In the following we explain each of these data lineage contributions in more detail.

3.2.2 To data (analytics) governance

Knowing data history (e.g., data origin, when and where data is transformed) contributes to data transparency. This can be helpful for various aspects of data governance like enabling compliance audit, risk management improvement, accountability assurance, and compliance (i.e., ensuring data being processed in line with organizational policies, community policies, legal rules and regulations, and regulatory standards). Further, it can boost various aspects of data management such as data quality management, data migration management, data silos integration, and data gap detection and mitigation (Stedman & Loshin, 2022). Each of these areas to which data lineage contributes, is explained in more details in the following sections.

Nowadays, the scope of the governance of smart environments is broaden in literature (Choenni et al, 2022). In addition to the substance of rules around the data collection, use, sharing, retention, and disposal, contemporary data governance includes specifying the ways that these rules are made, disputed, and changed (e.g., by whom and how the rules are made), see (Chyi & Panfil, 2020). Knowing about the ways that governance policies are made can enhance trust in data governance as well as in the data being governed. To contribute to such advanced data governance, data lineage should collect metadata on how and by whom data governance policies are made.

3.2.3 To personal data protection

Data lineage can contribute to various aspects of personal data protection. In this section we review some of these aspects as specified in the EU General Data Protection Regulation (GDPR, 2016) articles and recitals. Conforming with these articles and recitals requires the ability to identify which data is personal data as well as the entities from/through/to which the personal data is originated, transformed, and sent.

From the viewpoint of *data subjects*, data lineage can contribute to realization of, among others, the following articles of the GDPR.

- *Art. 15:* Right of access by data subjects to know about the purpose of personal data being processed, the categories of personal data being used, the (categories of) data recipients (especially those in third countries or international organizations), and the period of data being stored. Further, data subjects have right to get meaningful information about automated decision-making taken place based on their data.
- *Art. 16:* Right to rectification of inaccurate (and possibly incomplete) personal data.
- *Art. 17:* Right to erasure existing personal data, which is related to the right to be forgotten.

- *Art. 20:* Right to data portability, which enables data subjects to have the personal data transmitted directly from one data controller to another, where technically feasible.

From the viewpoint of *data consumers*, data lineage can inform them about a dataset having privacy issues like not being pseudonymized, anonymized, fair, or without users' consent.

3.2.4 To entrust AI models

A precondition for entrusting the models derived from data by various sorts of algorithms (like UDFs, AI/ML algorithms, and statistical algorithms) is to know which and how datasets are used for training these models. For example, users of such models should know about whether and to what extent the training data has data quality, Copy Right (CP), and bias issues. Heaven (2021) and Roberts et al. (2021) mention examples where ill trained AI models are developed due to not knowing about or not paying attention to the datasets used for training those models. As an example, they report about a case where researchers used datasets of adult patients with COVID-19 and of, as a control group, very-young patients without COVID-19 to train a model for detecting COVID-19. The trained model showed a strong performance as tested on those datasets but, in fact, the model "learned to identify kids, not covid" (Heaven, 2021), "merely detecting children versus adults" (Roberts et al., 2021). Therefore, data models obtained from statistics, AI/ML and UDFs should be traceable to the data used for their training. Data lineage can contribute to this traceability.

Data lineage can contribute to establishing trust in AI models in various ways. In the following we review two of these contributions as specified in recently ratified EU AI Act (2024). These contributions relate to two areas of the so-called high-risk AI systems and General-Purpose AI (GPAI) models.

The EU AI Act defines high-risk AI systems as those that are used as a safety component, used in by EU-laws specified products, or used in certain usage areas such as critical infrastructure, employment, law enforcement, migration, asylum, border control management, and administration of justice and democratic processes. One of the requirements for providers of high-risk AI systems is that those AI providers must provide instructions for use of these systems (which includes AI models) to downstream deployers to enable the latter's compliance (Art. 8-17). In providing AI systems with enough instructions, we think, one should include enough information about the data used for training the corresponding AI models. This is an interesting case of data lineage as the metadata of the training data accompany a product (i.e., the AI model) obtained from the training data. Thus, in such cases the data flow, i.e., the flow of the training data of the AI models, is extended with data-product flow, i.e., the flow of the trained model.

The AI Act defines a GPAI model as a "model, including where such an AI model is trained with a large amount of data using self-supervision at scale, that displays significant generality and is capable of competently performing a wide range of distinct tasks regardless of the way the model is placed on the market and that can be integrated into a variety of downstream systems or applications" (Art. 3-63). Providers of GPAI should provide, among others, a sufficiently detailed summary about the content used for training the GPAI model (Recital 107). Further, the providers of GPAI models that present a systemic risk should also track serious incidents caused by their

models and report them together with corrective measures to the commission (Recital 115). The providers of GPAI models shall draw and update technical documentation, including training and testing process and evaluation results, and supply them to downstream providers who integrate the GPAI model into their own AI system (Art. 53-1-b). This reportage can help downstream providers to understand the capabilities and limitations of the models and enables them to comply with their obligations (idem). Further, for governance purposes, downstream providers can lodge a complaint regarding upstream providers' infringement of the regulation (Art. 89). All these, we think, asks for monitoring incidents in downstream consumers of such models by GPAI model providers.

Another area of data lineage applicability within the EU AI Act can be attributed to the cybersecurity issue. According to the act, the GPAI providers must ensure an adequate level of cybersecurity protection by also conducting model evaluations and adversarial testing, as well as tracking and reporting serious incidents and mitigating cybersecurity protections (Article 55). As articulated in the following, data lineage can contribute to realization of such features that ensure cybersecurity of GPAI models.

3.2.5 To data (and model) explainability, interpretability and fairness

The increasing usage of AI systems for automated decision making or for supporting decision making (either at a personal or policy level) increases concerns over their lack of fairness, legitimacy, and accountability. Data lineage helps know how output data are derived from some input data. As such, it enables *algorithmic recourse* and *algorithmic contestability* for/by data consumers (like citizens, data journalists and civic organizations) in all stages of data lifecycle.

Algorithmic recourse is concerned with providing explanations and recommendations to individuals who have received unfavorable outcomes from automatic decision-making systems (Karimi et al., 2022). The explanations and recommendations should provide end-users with actionable measures whereby the outcome of an AI system can be changed to the favorable one. Making AI systems contestable by design is another way to mitigate these concerns (Alfrink et al., 2023). Via algorithmic contestability one aims at making AI systems responsive to human intervention throughout the system lifecycle. Such a human intervention can ask an AI system to explain and interpret how outcomes are derived from which input and training data. Unlike in algorithmic recourse, in algorithmic contestability the objective is not to change the algorithm outcome, but to change the outcome of the whole process from which the algorithm is part of. Answering these questions in both algorithmic recourse and algorithmic contestability can be facilitated by having data lineage in place.

3.2.6 To data quality management

Data lineage can improve data quality by contributing to the following aspects.

- Analyzing the root causes of data quality issues observed at the data consumer side. Such a reactive detection and improvement of data quality issues can be done by back tracing the observed errors/exceptions at downstream nodes of data journey paths (Stedman et al., 2022; Foote, 2023).
- Rectifying the data quality issues from the data origin side. Such a proactive detection and correction of data quality issues can be done by forward tracing the observed errors/exceptions at upstream nodes of data journey paths.

- Exception handling by validating the accuracy and consistency of data for accurate data analytics, BI and data science usages.
- Identifying incorrect assumptions about data due to, for example, change of semantics along data journey.
- Data cleaning via archiving data and/or deleting data whenever necessary. These may arise due to, for example, data being old or expired or data subjects requesting to delete their personal data for privacy reasons.

3.2.7 To data change management

Data lineage allows a proactive approach for data change management (Stedman et al., 2022; Foote, 2023), through analyzing the downstream impacts of envisioned changes made to datasets at upstream nodes. To this end, data lineage enables understanding the location of data destinations, the lifecycle of data, and the downstream IT operations. Hereby, upstream nodes get some ideas about the impacts on people and systems before propagating envisioned corrections and changes.

Example usage scenarios of data lineage for data change management are to ease large data migrations (e.g., moving to clouds, implementing upgrades, and performing consolidations), and to reduce risks when implementing data process changes by preparing for mitigations based on possible impacts on downstream data, processes, and systems.

3.2.8 To data ownership

Data lineage can inform data consumers about data flows and data sources, which may include some information about the data owners. Data consumers can hereby know who is responsible for and who owns every dataset or process in the data pipeline. This knowledge can ensure data consumers about who to contact for issues or changes. Data owners, in turn, can use data lineage to know where, by whom and how their data is used in downstream applications. As such, data lineage provides answers to the questions of data owners about their data (Informatica, 2024).

3.2.9 To regulatory compliance, compliance audit, and accountability

Data lineage may prevent getting fines for data processors on the data pipeline through indicating whether data transformations are done according to regulatory rules and guidelines, and whether appropriate controls and policies for containing possible threats (like security, privacy, safety, and fairness threats) are in place. Two of these regulatory regimes are EU GDPR and EU AI Act, which pertain to privacy and responsible AI usage, respectively. Some of GDPR and AI Act principles are outlined in Sections 3.2.3 and 3.2.4, respectively.

Data lineage, on the one hand, can assist data controllers/processors in knowing about having or not having compliance with policies, laws, and regulations for their data, data flows, and data transformations. On the other hand, it can provide a means for authorities to audit the compliance of data controllers/processors with policies, laws, and regulations.

Further, data lineage can contribute to data accountability as it can show who is/are responsible for data and its transformations along its journey path(s). To this end, data lineage should keep track of the roles and responsibilities at every stage of data

journey (Verma et al., 2024). When data lineage is used in the context of accountability it is critically important that data lineage metadata is managed securely sufficiently. Tan et al. (2012) mention a minimum set of security requirements for data lineage namely confidentiality, integrity, authenticity, and reliable collection. The latter means having trustworthy and accurate data lineage related metadata collection mechanisms. We think that for accountability non-repudiation should also be added to the minimum list of security requirements mentioned above.

3.2.10 To data security/privacy

Data lineage can increase security/privacy posture by enabling the search of data upstream and downstream to discover anomalies, and by tracking, identifying, and correcting potential risks associated with data (flows).

When data lineage is used in the context of data security and privacy protection, like in case of accountability, it is critically important that data lineage metadata is managed securely, where confidentiality, integrity, authenticity, and reliable collection are provided adequately (Tan et al., 2012). For example, Bertino et al. (2014) discuss information leakage concerns for data provenance due to sharing lineage metadata (especially in cross organizational settings).

3.2.11 To data modeling

Data lineage can provide the information needed to present visual representations of differing data components and their connections. The connections between data components can be shown in a model to show the *dependencies* present throughout a data ecosystem (Foote, 2023).

3.2.12 To data discovery

Partly, data lineage is about discovering the whereabouts and usages of already processed/shared (personal) data. As such, it contributes to and is relevant for discovering such data. To our understanding, data lineage is not about discoverability of new data which has not been published or processed. In the sense of finding data, the scope of data discovery is wider than that of data lineage. Note that data lineage is concerned with more functionalities than just data discoverability, as listed above.

3.3 Typical queries to data lineage

A practical approach for data lineage deployment starts with specifying the (types of) queries that the users of a data lineage system need to get answer for. Ram and Liu (2007) introduce a so-called W7 model for capturing the semantics or meaning of data lineage. Their generic model comprises seven related question words namely *what, when, where, how, who, which, and why*. Although the authors use these question words to specify the functionality of data lineage, we think they can be used to inspire posing typical queries to a data lineage system.

To gain more insight in what data lineage can mean in practice, we provide a list of questions that are typically posed to data lineage from different sources. Example data lineage related questions are:

- Where does this data come from?

- Where does this data go to?
- Do we have sensitive data that propagates unsafely?
- Is my database still in use?
- Can I delete my dataset?
- What systems and reports would be impacted by a change in a particular dataset?
- What systems and reports would be impacted by a change in a business operation process?

Data lineage related queries can be formulated based on the needs of organizations and their stakeholders within the context that they operate. Consequently, these queries could be business driven, serving the (business) needs of the stakeholders involved. In Section 4, we will categorize these business needs and stakeholders. For now, in addition to questions about the whereabouts of data, we elaborate on questions/queries at higher levels of abstraction that can be queried from data lineage. For example, from (Informatica, 2024) we list the following questions that one may ask from data lineage about data governance and data privacy.

- "What data in my enterprise needs to be governed for compliance with local, national and industrial regulations?"
- "What data is appropriate to migrate to the cloud and how will this affect users?"
- "Where do we have data flowing into locations that violate data governance policies?"
- "How does data quality change across multiple lineage hops?"
- "How can data scientists improve confidence and trust in the data needed for advanced analytics?"
- "What data sources have the personal information needed to develop new customer experience initiatives? And how is this data distributed across the organization?"

We note that the last query seems more like a data discovery type than a data lineage type. For a more data lineage type, one could reformulate it as: What data sources, which are used for this data product, have personal information? And how is this data distributed across the organization?

3.4 Concluding remarks

The concept of lineage has traditionally been applied to a wide range of cases, ranging from tracking/tracing the computation flows in a single software program to tracing/tracing the data flows in distributed ISs. Data lineage can contribute to many objectives, each of which, in turn, plays a role in enhancing trust in data and data driven applications and policymaking. We have enumerated and explained 12 (related) objectives to which data lineage can contribute.

It would be intriguing to aim at all mentioned objectives when opting for a data lineage solution. Such a versatile data lineage solution could immediately become too complex and costly, thus impossible to realize especially in distributed settings (e.g., among the semi-autonomous organizations of the DJS). It is therefore imperative to start with identifying the potential users of the aimed data lineage system, and their data lineage needs (i.e., to be business driven). A way to elucidate these needs is to identify the typical quarries that the potential users (would) want to ask from the data lineage system. One can subsequently analyze these typical queries and determine whether they are fixed (i.e., they can be predetermined and remain the same in the future) or should be flexible. For the latter, it is necessary to estimate how much flexibility is

needed. The analysis results can subsequently be mapped to the objectives that a desired data lineage system must fulfill. Having an insight in all these aspects will influence the characteristics and architecture of, and the tools used for data lineage.

Reducing the complexity of data lineage and the associated costs is a crucial factor in successful adoption of data lineage. In conventional lineage it is assumed that users can understand how an output is created by observing the source data and knowing that the data transformation is a sequence of simple operations like filter, join, and aggregation (Yamada, et al., 2023). However, in complex data analysis flows, like UDFs and AI/ML algorithms, more information about data transformation is needed. For example, for a case of UDFs Yamada, et al. (2023) argues that more information about the reasons/features affecting the outcome of a classifier's decision can be provided. A question that may arise is which data lineage information should be provided to explain and/or influence the outcomes of very complex data transformations (like LLMs) and how this data lineage information should be managed in a (cost) effective way.

4 Data lineage architecture

In this section we aim at making preparation for answering the third research question: "How can data lineage tools be deployed?" Specifically, considering the study context and scope, we derive a generic architecture of data lineage in this section as an introduction to answering the third requestion in the following section. To this end, we start with eliciting some requirements from different viewpoints and identifying possible stakeholders in Section 4.1. Subsequently, based on these requirements, we revisit the definition of data lineage in Section 4.3, present a structural architecture in Section 4.3.2, and propose a functional architecture in Section 5.1. Finally, we recapture the main outcomes of the whole section in Section 4.5.

4.1 Requirements

In this section we review the general requirements that might influence and be relevant for deploying data lineage within the DJS. Using the term "general" here stems from the fact that the work presented in this report is the result of a preliminary study to explore the field and does not aim at proposing a customized data lineage solution for the DJS. As such, there is no intention here to be exhaustive and elicit all data lineage requirements in the operational context of the DJS. Note that the requirements mentioned in this section are derived based on our four expert interviews, the discussions within two expert focus groups and with the members of the project advisory board, and self-reflection.

4.1.1 From the viewpoint of the organization

The main driving force for using data lineage within the DJS is to gain trust in data and the outcomes of algorithms such as the models and policies, predictions, and inferences made based on those models. Particularly, within public organizations (e.g., the DJS), there is a need to know about and investigate the data (i.e., the evidence) used for making a policy. Considering the contributions areas of data lineage mentioned in Section 3.2, we distill that data lineage within the DJS can (or is required to) contribute to the following functionalities: data governance, data discovery[2], data quality management, data change management, and privacy and security mainly.

As mentioned in Section 1.3, there are many semi-autonomous organizations within the DJS that collectively form a linear chain of relations with parallel and loop links (thus, forming a *network* of semi-autonomous organizations). Every organization within the DJS operates independently largely as far as data governance and data management are concerned. It is, however, necessary to share data within and across organizational boundaries and to have trust in the data shared. Therefore, data lineage should be possible within and across these organizations. Further, the autonomy of the organizations involved results in having diverse business, conceptual and logical data models across these organizations. Even within one organization, different data management systems might be in use. As such, data lineage within DJS should

[2] This concerns discovering the data shared already or used as evidence for existing policies, prediction/inference models. Similarly, one can search for relevant data for making future policies and models. This sort of data discovery for new usages, we suspect, is not part of data lineage.

account for diversity at all levels, namely at technical, logical, conceptual, and business levels.

Within the context of the DJS, data is shared and processed for not only research and strategic purposes, but also for operational purposes. Both types of purposes may require the shared data to be at various aggregation levels, i.e., at group and individual levels (i.e., aggregated data and microdata, respectively). As such, data lineage can be needed at both coarse-grained and fine-grained levels.

4.1.2 From the viewpoint of the study inquirer

The inquiring party for this study was interested in investigating data lineage technology along two dimensions as follows.

- *Technical vs business data lineage* (see Section 2.4.8): On the one hand, it is desired to track/trace data spatially by capturing data journey among the ISs spread across organizations (i.e., to investigate technical data lineage methods and tools). On the other hand, it is desired to follow the logical path of data along various levels, consisting of legal, business, conceptual, logical, and physical levels (i.e., to investigate business data lineage methods and tools). Having a legal perspective – or better said, starting from a legal perspective – is particularly important within the DJS because laws and legal frameworks define the substance and form of all activities within an organization that is responsible for guarding the rule of law in the society.
- *Data origin vs data flow lineage* (see Section 2.4.1): On the one hand, it is desired to know which input data source(s) has(have) contributed to an output, without knowing/tracking the intermediary nodes involved. This is sometimes called data provenance. On the other hand, there is a desire to track and trace data flows from origin(s) to destination(s), including intermediary nodes.

We note that considering the granularity of the above-mentioned types of data lineage (i.e., coarse-grained vs fine-grained, see Section 2.4.4) was not required at this phase.

4.1.3 From the viewpoint of the end-users of data lineage

The end-users of data lineage are of various types – like data subjects, data stewards, data engineers, data auditors, data analysts, business analysts and data scientists. Some of them reside at the beginning of the data pipeline (like data subjects), some at the end (like data scientists), and some in between (like data engineers). Further, data lineage end-users often have different backgrounds with a varying set of data (science) skills. This may range from data science professionals to data-driven business analysts and citizens. Each category of end-users may query some aspects of data lineage and should be provided with replies that are appropriate for their needs and backgrounds. Such flexibility can serve the purpose of the democratization of a data-driven practice.

4.2 Revisiting the data lineage definition

In this section we revisit the definition of data lineage given in Section 2.3, stating:

> "Data lineage is the description of data movements and transformations at various abstraction levels along data journey paths. The description encompasses those aspects that are of interest in an application context like how (i.e., by whom, when, where, which, etc.) data objects are processed (i.e., created, collected, stored, accessed, transformed, transmitted, etc.) and how they are related to high-level data concepts."

This revisiting will be through the lens of the requirements raised in Section 4.1, result of which is summarized in Table 4.1.

Table 4.1 Revisiting the data lineage definition

Which requirements (with a link to data lineage characteristics, see Section 2.4)	Coverage by the data lineage definition
Knowing about the data origins vs data flow (horizontal lineage)	"… along data journey paths …"
The shared data within the DJS can be at various aggregation levels for strategic and operational purposes (coarse- vs fine-grained)	"… various abstraction levels …" "… those aspects of interest … data objects"
The complexity and cost of a data lineage tool depend on the subset of the purpose of data lineage in mind (see Section 3.2). In practice, one might not need to fulfill all these purposes.	"… those aspects that are of interest in an application context …"
Following the logical path of data along legal, business, conceptual, logical, and physical levels (vertical lineage)	"… at various abstraction levels …"

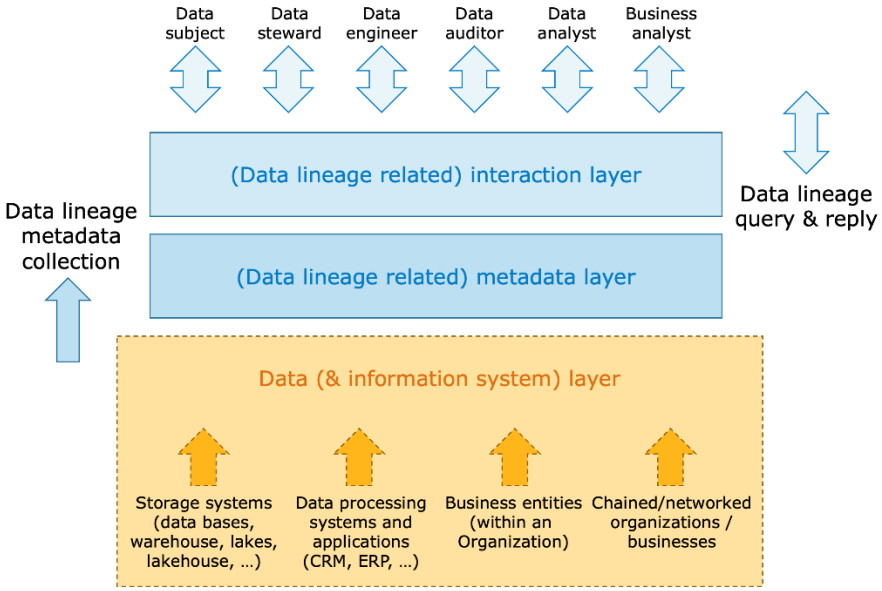
4.3 Data lineage context

In Section 4.3.1, we provide a layered model that gives insight in data lineage component of ISs and in Section 4.3.2 we elaborate on two categories of metadata that are pivotal for this report.

4.3.1 A layered model

To highlight the data lineage functionality within a typical data driven system, as shown in Figure 2.2, we convert the schematic view shown in Figure 2.3 to a layered model in Figure 4.1. The data layer, which is not part of data lineage, interfaces with various (types of) ISs distributed over the whole network of organizations. The middle layer represents the data lineage related metadata, which is collected from various ISs (see the left-most arrow), processed, stored, and shared to provide data lineage information to the data lineage related interaction layer. The latter receives the data lineage related queries of various types of end-users; and composes and provides the replies to the data lineage related queries in appropriate formats.

Figure 4.1 A layered IS model with a focus on its data lineage component



4.3.2 *Revisiting vertical and horizontal data lineages*

In Section 2.4.9, we briefly explained vertical and horizontal data lineage and their relation to business and technical data lineage (see Table 2.1). Although there are no universal and standardized definitions of these terms (Karkošková & Novotný, 2021), we find it worthful to make the distinction between horizontal and vertical lineage more explicit. Hereby we can characterize the typical types of metadata that should be collected for data lineage.

The definition of data lineage touches upon the concept of "abstraction levels", which may relate to the concept of vertical data lineage (see Table 4.1). *Vertical lineage* focuses on the design of a database and relates abstract objects at different layers, objects such as a business partner (e.g., a product supplier) and physical implementation (e.g., a part of a star schema), see (Freche, Heijer and Wormuth, 2021). For the layers involved in vertical lineage Steenbeek (2022) adopts the following four levels.

- *Physical level*: To depict the physical artifacts on the database (like physical data models).
- *Logical level*: To describe data entities and data transformation rules (like logical data models).
- *Conceptual level*: To describe entities and business restriction rules (like conceptual data models).
- *Business level*: To describe business processes and roles.

During project execution, we identified and added another level that is particularly important for the organizations that oversee and safeguard the rule of law in the society (like the DJS).

- *Legal and ethical level*: To specify the legal and ethical grounds that allow using data for a specific purpose. To this end, various national laws (like Wpg[3] and

[3] The Police Act, in Dutch "Wet politiegegevens" (Wpg).

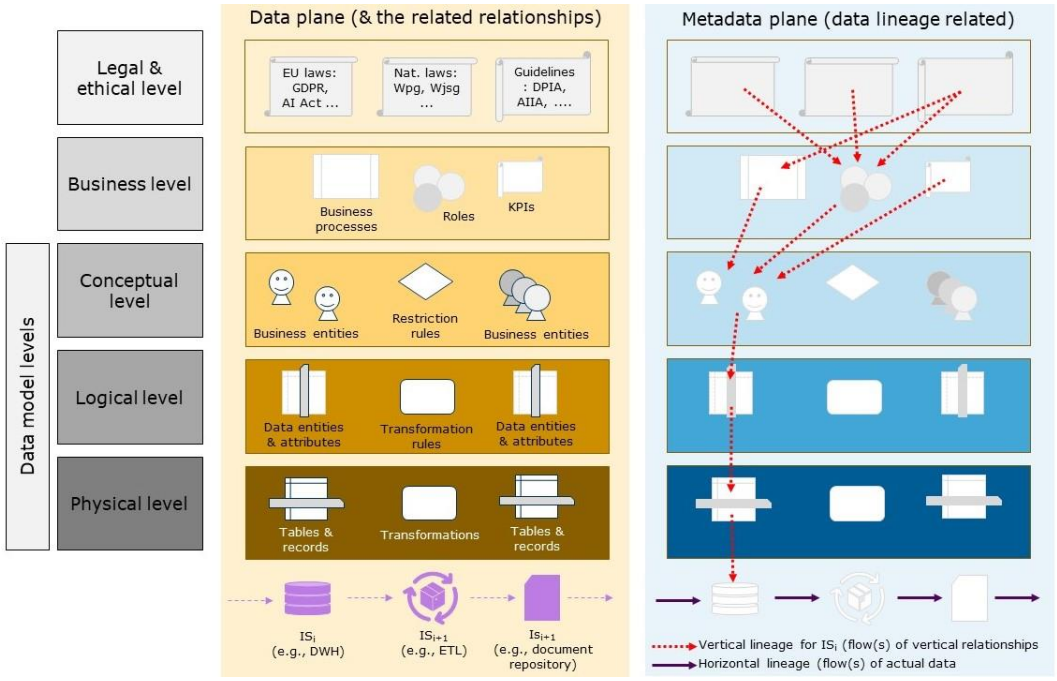
Wjsg4), international laws (like GDPR and AI Act), and guidelines and standards (like DPIA5 and AIIA6) may apply.

The definition of data lineage touches upon the concept of "along data journey paths", which relates to the concept of horizontal data lineage (see Table 4.1). *Horizontal lineage* describes a physical lineage (Freche et al., 2021) through describing how data items (ranging from datasets to data tuples) flow between ISs, applications and platforms (like databases, data warehouse, data lakes, data lakehouses, data transformers) and get transformed along their paths. In short, a horizontal lineage shows how actual data flows and gets transformed along its paths between ISs. These data flows and transformation can occur within an organization or across organizations. "Horizontal lineage enables understanding the dependencies and relationships between data sources, data transformers and data consumers, and to identify potential data quality issues, risks, and gaps" (Freche et al., 2021). According to these definitions, horizontal data lineage occurs at the physical level, as such it is called technical data lineage. In Section 5.1.2 we will show that horizontal lineage can also be applicable to higher abstraction levels (like business level) especially in cross organizational settings (as also indicated in Table 2.1).

Figure 4.2 illustrates the concepts of horizontal and vertical lineage, which is inspired by the model in (Steenbeek, 2022). As indicated in Figure 4.2, the last three levels represent the typical data model levels of detail (DAMA International, 2017: Ch5). As mentioned before, there isn't any established norm for defining the layers of vertical lineage. Nevertheless, it is essential to identify these abstractions levels as we foresee the possibility of replying some data lineage queries may require lineaging among or at higher level of abstraction like: Which business processes do use this dataset? What are the legal grounds for sharing this dataset?

4 The Judicial and Criminal Data Act, in Dutch: "Wet justitiële en strafvorderlijke gegevens"(Wjsg).
5 Data Protection Impact Assessment (DPIA)
6 AI Impact Assessment (AIIA)

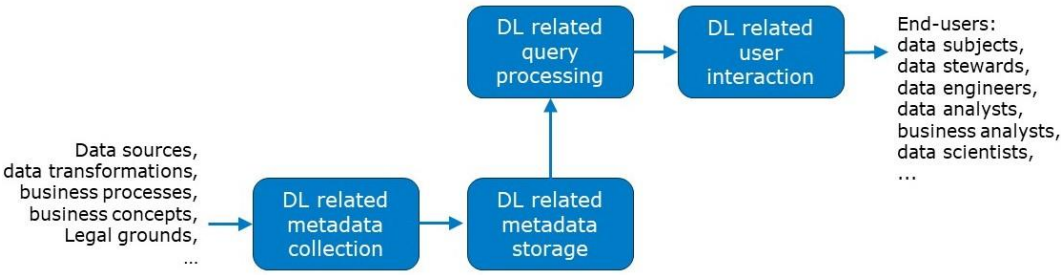
Figure 4.2 Illustration of horizontal vs vertical lineage (in metadata plane)



4.4 A functional architecture

From the layered Data Linage (DL) model shown in Figure 4.1 we distil four functions, as shown in Figure 4.3, for a typical data lineage system, namely DL related metadata collection, DL related metadata storage, DL related query processing, and DL related user interaction. Note that the arrows in Figure 4.3 show the main information flows between the identified functions. In the following subsections we use this model to describe the functional components of data lineage in more detail. Further, note that we do not aim at capturing all aspects of metadata management in the model in Figure 4.3 (i.e., data collection and data storage are part of data management functionality).

Figure 4.3 A functional architecture for data lineage



4.4.1 *Metadata collection*

Metadata collection is concerned with the updates, i.e., the writes, of data lineage related metadata. It can be done in various ways. Common techniques for collecting

data lineage related metadata are listed below from (Stedman & Loshin, 2022; Imperva, 2024; Foote, 2023).

Pattern-based data lineage: In this approach one looks for patterns in datasets such as similar data elements, rows, and columns. In case that such similar elements exist, one may conclude that the datasets are related and may be part of a data flow. For example, if there is a column in two datasets with a similar name and almost the same data values, it is very likely to consider them as the same data being in two stages of its lifecycle. If one finds out that some data values or attributes are different in those flows, (s)he may conclude that some transformations are taken place on those data flows. The data transformations and data flows can then be documented as data lineage metadata. The advantage of pattern-based data lineage is that one does not need to know the code used to generate/transform the data.

Data lineage by data tagging: In this approach one assumes that somehow a transformation engine tags data along its journey. The tagging can be done by, for example, examining some metadata manually (by data stewards or end-users) or automatically (by, e.g., built in data governance software tools). Subsequently, the tags are tracked from start to finish to create data lineage related metadata. This data lineage by data tagging approach is effective only if there is a "consistent transformation tool" (Foote, 2023) that consistently controls all data movements and has a consistent view on the tagging structure along data journey. As such, it is suitable for data lineage in controlled/closed data systems.

Self-contained data lineage: In this approach one possesses a data storage and processing environment with a master data management unit that has a central control over metadata. This type of self-contained system can provide data lineage inherently, without needing external tools. Like the data tagging approach, self-contained data lineage is applicable within controlled/closed environments.

Data lineage by parsing: In this approach one uses advanced tools to identify and extract data lineage metadata from data transformation logic, runtime log files, data integration workflows and other data processing codes. This approach applies reverse engineering to data transformation logic throughout data journey (i.e., delivering end-to-end tracing across various ISs). As such, it is the most advanced form of data lineage, which requires a good understanding of the tools and programming languages used throughout data journey. Therefore, data lineage by parsing is complex to deploy but can be more accurate than pattern-based lineage.

Manual data linkage: In this approach, as the name suggests, one interviews various actors – like business users, BI analysts, data scientists, data stewards, data integration developers – about how data flows and how data gets transformed when going through ISs. The resulting data flow and transformation information can be used as a starting point for a more automated data lineage metadata collection. Manual metadata collection may require a significant resource investment and be prone to error.

Real-world based data lineage: In this approach one uses (physical, chemical, etc.) relations within the real-world to establish lineage between data objects in the cyberspace. For example, the discovery of a specific DNA evidence may link two data objects in the cyberspace that had no relation otherwise.

Note that abovementioned methods for data lineage related metadata collection aren't necessarily mutually exclusive (Stedman & Loshin, 2022). Further, an organization might deploy more than one data lineage approach considering its needs and data environment.

4.4.2 Metadata storage

The metadata about an object (e.g., a file or dataset) can be maintained in a fixed volume (e.g., when one is interested in knowing the current state of the object), in a linearly increasing volume (e.g., when one is interested in knowing when the object is operated upon), or in an exponentially increasing volume (e.g., when one is interested in knowing how processes utilize multiple objects), see (Gehani et al., 2009). Data lineage in its broad sense is of the last type, in that data lineage metadata grows exponentially, inflicting significant costs for data lineage related metadata storage and data lineage related metadata retrieval, especially in distributed environments (Gehani et al., 2009). To deal with these costs, data lineage related metadata should be stored effectively depending on the data lineage objectives sought and the operation context.

For storing lineage metadata for files in filesystems, several strategies are mentioned in (Gehani et al., 2009). We list these strategies for storing file (or data) related metadata in the following as they can be insightful for storing data lineage related metadata in general.

- *In auxiliary files*, where the metadata is stored in dedicated files. The drawback of this strategy is that the link between the metadata and the file (or data) can be lost. For example, if an operation that is not lineage aware is performed on the file (or data) the metadata may stay behind and can be lost for the downstream operations.
- *In a filesystem layer*, where the metadata is stored in a central place at the file system. This strategy retains the connection between a file (or data) and the metadata within the filesystem but does not retain it when the file (or data) is moved out of the filesystem.
- *In a local database*, where the metadata is retained in the node where the data is generated, i.e., it is not propagated along the file (or data) journey path. This strategy retains the connection between a file (or data) and the metadata within the node but does not retain it when the file (or data) is moved out of the node.
- *In a file server*, where the metadata is retained in a central server within the domain where the data is generated, i.e., it not propagated along the file (or data) journey path. This strategy retains the connection between a file (or data) and the metadata within the domain (not outside of the domain). This strategy may face degraded storage and retrieval performance if the metadata is barely or partly used/retrieved.
- *As in-band encoding*, where the metadata is encoded to the data (like watermarking in multimedia content) and can accompany the data along its journey. Such hard-coded metadata may not be noticed by legacy applications that are not lineage aware. Consequently, new operations by such applications may not be added to lineage metadata and the encoded metadata may degrade data quality (thus, harm the user experience). For example, if an AI training algorithm uses a dataset consisting of X-ray images and their data lineage metadata, then the resulting model may rely on patterns in the metadata rather than in the X-ray images. The model may show a good performance on the same test dataset for detecting fractures because those test X-ray images were made with an X-ray machine from the emergency room department, rather than detecting the fractures

in the X-ray images. For lineage-aware applications, adopting this strategy may cause exponential growth of metadata along the data journey path.
- *In headers and footers*, where unlike in-band encoding, the lineage metadata is not hard encoded to the data but is added as the header or footer to the file (or data). The C2PA approach for trustfully binding lineage information to any piece of media content, see Box 1.1, appears to follow this strategy. This strategy mitigates the user experience problem, but still may cause exponential growth of metadata along the data journey path.

Based on the illustrative example of metadata storage strategies for filesystems, we distill the following three models (or architectures) for data lineage storage.
- *Piggybacking*[7] model: Here data lineage metadata accompanies the data along its journey (like in-headers/footers and in-band encoding strategies mentioned above, each having a varying degree of data and metadata stickiness).
- *Centralized* model: Here data lineage metadata is stored locally in a central repository (like in a fileserver, in a local database, in a file system layer or in auxiliary files repository strategies mentioned above, each having a varying size of locality).
- *Distributed* model: Here data lineage metadata is stored locally in a central repository within a domain (e.g., an organization). Moreover, there is an efficient built-in mechanism in place to establish links between repositories of different domains whenever a data lineage query requires for that (i.e., on a need-to-know basis).

The piggybacking model adds an overhead of a fixed or a linearly increasing volume if data lineage is about the original state of the data or about when the data is operated upon, respectively. When data lineage is about retaining the information about how data is combined with multiple data objects, then the piggybacking model cannot be scalable due to its exponential volume growth.

The advantages of the centralized model include the ease of maintenance and curation, storage efficiency, and access control (Groth, P.T. (2008). On the other hand, the disadvantages include introducing significant network overhead due to data lineage updates (i.e., transferring data lineage records to the central data store) in cases where the metadata is heavily accessed which leads to increased latency of remote lookups (Gehani, 2009) or there is barely request for this information (Malik et al., 2013). Further, sometimes it is useful to have oversight at the application level on the location where data lineage metadata is collected, processed, stored, and consumed (Malik et al., 2013).

A key issue in realizing the distributed model is to semantically describe and optimally reference and discover data lineage objects (metadata items) across domains. The identification of objects uniquely across different administrative boundaries can be done in various ways (Malik et al., 2013). For example, according to (Malik et al., 2013), global naming, indexing, and querying are used in the PASS: Provenance-Aware Storage System (Muniswamy-Reddy et al., 2009) in the context of sensor data. The SPADE: Support for Provenance Auditing in Distributed Environments system (Gehani & Tariq, 2012) uses data item identifiers that are unique to a host and the distributed queries refer to these data items unambiguously by using the host on

[7] Like sticky policies: ”Machine-readable policies are sticked to data to define allowed usage and obligations as it travels across multiple parties, enabling users to improve control over their personal information” from https://privacypatterns.cs.ru.nl/patterns/Sticky-policy

which the data items are generated and the local identifiers on that host (Malik et al., 2013).

The metadata can be stored in data lineage systems in various ways. Example repository types are (Abiodun et al., 2022): distributed file systems (Mothukuri et al., 2021), graph databases (Vicknair et al., 2010; Woodman et al., 2017), relational databases (Vicknair et al., 2010), triple storage[8] (Wylot et al., 2017) and NoSQL (Kashliev, 2020). Each of these storage models is efficient for specific query type, storage size, and inference support (Abiodun et al., 2022). For example, let's consider graph and relational database types, see (Memgraph, 2023; AWS, 2024). Graph databases are suitable for highly connected datasets, where there is a need for analyses that require searching for hidden and apparent relationships. Further, graph databases are optimized to retrieve data rather than to store data. Finally, graph databases are flexible and thus they are suitable for inconsistent data structures that require frequent change to data models (like adding new attributes, adding missing attributes for some entities, and not having strictly defined attribute types). If the data is transactional and data queries are used for lookup information stored in key-value pairs, then a graph database is likely not suitable. For example, when we need to record and retrieve personnel information (like individuals' names, job-functions, and dates of employment). As there is no need to retain additional information (i.e., the data structure is static and the columns on the table will not change), a relational database is more suitable for such unrelated and tabular data.

In conclusion, the type of data repository should be chosen according to the data lineage type needed and the context in which it operates.

4.4.3 Query processing

The methods used for querying data lineage databases are determined by the storage method chosen for lineage metadata (Bose, 2002). For managing and manipulation of relational databases SQL is mostly used as the programming language (Memgraph, 2023). SQL is used not only for querying but also for creating, modifying, deleting tables and the data in the tables as well as to insert, update, and delete data in the tables. Most relational database management systems (e.g., MySQL, Oracle, Microsoft SQL Server, PostgreSQL, and SQLite) use SQL as their standard language.

A query language used for graph databases is Cypher. It can be seen "as mapping English language sentence structure to patterns in a graph" (Memgraph, 2023). It is a powerful and flexible query language well-suited for complex queries about highly connected data with deep hierarchical relationships (e.g., parent-child relationships) or many-to-many relationships between different tables. Note that having *flexibility* in terms of the types of queries that can be posed, has a downside in that replies might be uncertain to some degree (like in the case of information retrieval systems).

[8] "A triplestore or RDF store is a purpose-built database for the storage and retrieval of triples [1] through semantic queries. A triple is a data entity composed of subject–predicate–object, like "Bob is 35" (i.e., Bob's age measured in years is 35) or "Bob knows Fred". Much like a relational database, information in a triplestore is stored and retrieved via a query language. Unlike a relational database, a triplestore is optimized for the storage and retrieval of triples" Wikkipedia.

4.4.4 *User interaction*

Currently data querying is mostly performed by data science experts using formal and technical languages such as SQL (for relational databases) and Cypher (for graph databases). Such query languages are not suitable for non-technical business experts (Pinon et al., 2023).

To achieve Self-Service Business Intelligence (SSBI), where one aims at producing timely, factual and contextualized information for decision makers, democratizing Data Query Support Solutions (DQSSs) are needed (Pinon et al., 2023). This has been widely demonstrated in the literature through the development of two categories of solutions, namely: *Visual Query Languages (VQLs)* and *Natural Language Interfaces for DataBases (NLIDBs)* (Pinon et al., 2023). An introduction to these solution directions is summarized from (Pinon et al., 2023) and is presented in three paragraphs below.

Unlike SQL that relies on textual representation, VQLs rely on visual representations to enable end-users to construct data queries. Visual representations used in VQLs include tables, diagrams, or icons to represent concepts and relationships (Catarci et al., 2018). End-users can focus on the meaning of their queries, not much on the syntactical aspects of queries. This approach, however, might not be suitable for complex query types (Silva et al., 2019) and has not been successfully implemented in real-world environments.

NLIDBs, developed in the 70's after VQLs, are increasingly robust and commercially available. They allow end-users to write data queries in natural languages (Androutsopoulos et al., 1995) and eliminate the need for using a formal query language (e.g., DAX, SQL and SPARQL) or visual representations (i.e., VQLs). These are like probing and prompting, used in interacting with LLMs such as the way done in ChatGPT. Ambiguity inherent in natural languages and the increasing complexity of data queries are the main challenges for these tools (Pinon et al., 2023; Őzcan et al., 2020).

Along reducing the complexity of technical data query languages for non-IT experts, Pinon et al. (2023) see the necessity of enriching the NLIDB to deal with (a) the issue of the semantic gap existing between database jargon (of technical minded system developers) and business jargon (of business minded end-users), and (b) the problem of finding information in fast growing data nowadays.

In Section 4.1.3, we mentioned that data lineage end-users may reside at the beginning of data pipeline or at the end of data pipeline (and some in between). Further, data lineage end-users often have different backgrounds with a varying set of data (science) skills. Each category of end-users may query some aspects of data lineage and should be provided with replies that are appropriate for their needs and backgrounds. Such flexibility can serve the purpose of the democratization of data-driven practice. To reply to all queries of such a wide range of end-users, we recognize the following types of data queries, which can be related to the vertical aspect of data lineage:

- *Business driven data lineage*, where data lineage is set up and steered by the business needs of the end-users (who may reside at the beginning, middle or end of data pipeline), see a similar work in (Pinon et al., 2023). This data lineage approach should be able to reply to ad-hoc and previously undefined queries.

- *Data-driven data lineage*, where data lineage is set up and steered by data providers (who often reside at the beginning of data pipeline). This approach requires a standardized and/or centrally coordinated data lineage design. As such, it may not be responsive to ad-hoc queries, i.e., data lineage can reply to limited and pre-defined queries like the WODC mashup demo (Choenni & Leertouwer, 2010).
- *A combination*: Like strategies 1 and 2 in Figure 2.4, some aspects of data lineage are pushed from data origin, the rest are the aspects that are not predefined and should be realized at run-time (like the way that prompt engineering perform in LLMs/GenAI).

The abovementioned options specify the way that end-user may interact with a data lineage system. One offers a predetermined set of queries and the other offers a flexible (not predefined) set of queries to end-users. Adopting each of these options has its own practical implications.

4.5 Concluding remarks

Gaining trust in data, e.g., for trusting the data used for making a policy, is the main driving force behind deploying data lineage. Based on our interviews, we distilled that data lineage within the DJS can (or is required to) contribute to data governance, data discovery, data quality management, data change management, and privacy and security mainly.

It is necessary to trustfully share data within and across organizational boundaries in the DJS while allowing participating organizations maintain their autonomy and have own business, conceptual and logical data models and ISs. As such, data lineage within the DJS should account for diversity at all levels, namely at technical, logical, conceptual, and business levels. Further, data is shared and processed for not only research and strategic purposes, but also for operational purposes. Thus, the shared data could be at various aggregation levels, i.e., at group and individual levels, which results in having data lineage at both coarse-grained and fine-grained levels. The inquirer of the study was interested in technical and business lineages as well as data origin and data flow lineages.

The end-users of data lineage within the MJS can be of different backgrounds with a varying set of data (science) skills and may reside at the beginning, middle or end of data pipeline. Therefore, there should be enough flexibility to serve a wide range of users and aim for data lineage democratization.

The data lineage definition we provided covers various aspects that were raised in the expert interviews, expert focus groups and the (gray) literature. Particularly, it covers the data lineage scope at various abstraction and granularity levels, along data journey paths, and being limited to those aspects of data lineage that are of interest in each context.

Based on the gained insights, we defined five abstraction levels for data lineage namely three from the traditional data model (i.e., physical, logical, and conceptual levels), one from the literature (i.e., business level), and one introduced by us that is inspired by the common practice within the DJS (i.e., the legal and ethical level). The latter is particularly important in the DJS, as it is responsible for overseeing and

safeguarding the rule of law in the society. The defined levels constitute the levels of business data lineage, which coincide with vertical data lineage. In the following section we will argue that the business data lineage has also a horizontal dimension to it, particularly in cross organizational settings.

Data lineage is functionally divided in four areas in this section namely data collection, data storage, query processing, and user interaction. There are various ways for data lineage metadata collection. Nevertheless, automizing the data collection process is necessary considering the speed and volume at which data is collected and shared nowadays. Metadata storage models that suit the DJS are piggybacking if data lineage is about the original state of the data. A centralized model cannot be suitable for managing data lineage related metadata across autonomous organizations like in the DJS. Therefore, there is a need for a scalable distributed model that well fits the organizational structure of the DJS. In the following section, we will suggest considering a federated data lineage architect for the DJS to establish links between data lineage metadata repositories of different domains.

There are various repository types for storing data lineage metadata. The type of data repository should be chosen according to the data lineage type needed and the context in which it operates. According to the type of the data repository chosen, one can opt for an appropriate query processing language. The user interaction can be facilitated in different ways, depending on the technical skills of the users. For example, a business-driven data lineage should be usable for users with less technical backgrounds and ad-hoc (i.e., not pre-defined) data lineage queries. For these users, it may be needed to deploy natural-language-based user interfaces which are capable of handling predefined and new (i.e., not predefined) queries.

5 Data lineage deployment

In this section we aim at answering the third research question: "How can data lineage tools be deployed?" For answering this question, we will sketch two directions for deploying data lineage within the context of the DJS. Note that via this sketch we intend to illustrate two boundary states in the solution space and give insight in their potentials and challenges. As such, it is not our intention here to prescribe a specific deployment solution. The deployment directions sketched will be based on the generic architecture and principles presented in Section 4.

We start this section with laying down an envisioned architecture for data lineage in the cross organization setting of the DJS in Section 5.1. Subsequently, we present a model for explaining deployment scenarios for data-driven systems in Section 5.2, and explain two strategies delineating the deployment of data lineage systems in Section 5.3. Finally, we recapture the main outcomes of the whole section in Section 5.4.

5.1 Data lineage in cross organizational settings

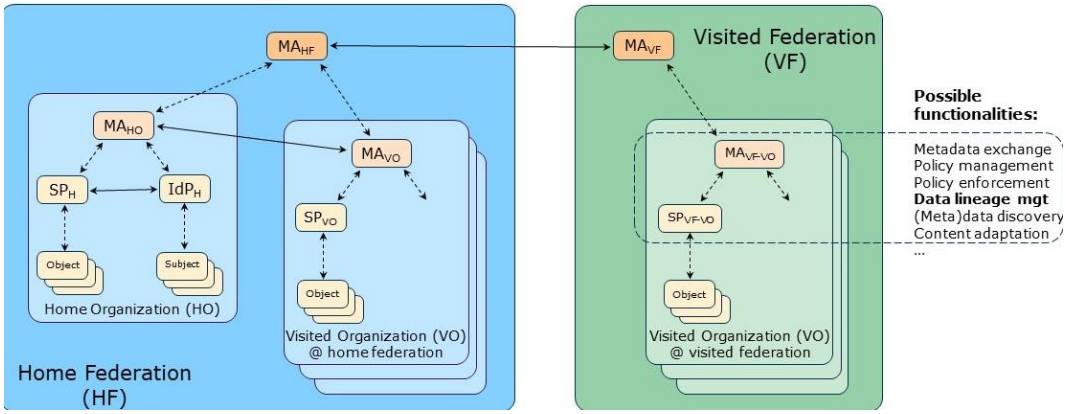
In this section we describe an IS architecture that can inspire the deployment architecture of data lineage in the DJS in Section 5.1.1. Subsequently in Section 5.1.2 we elaborate on how vertical and horizontal lineage can be intertwined in cross organizational settings such as that of the DJS.

5.1.1 A model for data lineage metadata management

Out of the three models described in Section 4.4.2, we consider piggybacking and distributed models as candidate models for deploying data lineage in cross organizational setting of the DJS. The piggybacking model can be a valid option if one opts for data source tracking and the data is not combined with other data streams. It may not be scalable if one is interested in tracking all data transformations and the data is expected to be combined with multiple data objects. Further, the piggybacking model alone is not helpful for the data lineage needs of the end-users who reside at the beginning of the data pipeline/journey.

The alternative option is to deploy data lineage across the DJS organizations according to the distributed model. As mentioned in 4.4.2, a key issue in realizing the distributed model is to semantically describe and optimally reference and discover data lineage objects (data lineage metadata items) across domains. A relevant architecture for this purpose is that of the federated identity management used within the federation of Dutch universities, called SURFconext (2024), and across the university federations of mainly European countries, called eduGAIN (eduGAIN, 2024). The architecture of these federated identity management systems is illustrated in Figure 5.1, together with components Metadata Aggregator (MA), Service Provider (SP) and IdP (Identity Provider). The architecture is distributed and federated.

Figure 5.1 A federated model for identity management



The federated model relies on a hierarchical structure within an organizational unit (i.e., universities in the example above) and a peer-to-peer structure across organizational units (i.e., between universities in a federation/country and across university federations/countries in the example above). The model works very well in practice, considering the success of SURFconext and eduGAIN identity federations. This success of the model can be attributed to its reliance on the existing organizational structures (Bargh et al., 2024) where:

- every organization manages its own (meta)data locally, based on its internal policies and systems,
- partner organizations join forces and form an alliance (i.e., a federation) to collaborate on matters of common interest (in our case, data lineage metadata exchange), and
- alliances (i.e., federations) of organizations may also collaborate on matters of common interest (i.e., forming federation among federations or confederations).

Such a reliance on the existing organizational structure works well and can scale up organically. Inspired by these success stories, we propose considering a similar architecture – let's call it *federated data lineage metadata management architecture* – for deploying data lineage in cross organizational settings such as that of the DJS. This model can be deployed in a centralized way per organization (or per department within an organization) and in a federated way across organizations (or across departments within an organization).

The proposed architecture, shown in Figure 5.1, can be used for trustful metadata exchange across collaborating organizations. The proposed architecture can be related to that of edge computing (Qiu et al., 2020), in the sense that (semi)autonomous domains are connected via gateways at the boundaries of those domains. A gateway is an entry/exit point to/from a domain, which is responsible for, among others, access control to the domain, content adaption towards outside the domain, and service and data discovery within and across domains. The Metadata Aggregators (MAs) shown in Figure 5.1 are gateways that can store data lineage related metadata locally and enable exchanging (a digest of) the locally collected data lineage metadata with other peer organizations in the (con)federation. The MAs can also host other functionalities and roles like Policy Enforcement Point (PEP), Policy Decision Point (PDP), context transfer, content adaption; should the underlying components/parts of an organization or federation be unable to do so. As indicated in Figure 5.1, the MAs can also host and exchange other types of metadata, like for trust establishment (Bargh et al., 2024),

data management, and data governance. These functionalities include per domain Service Discovery, Data Discovery, Data Catalog, and Data Lineage.

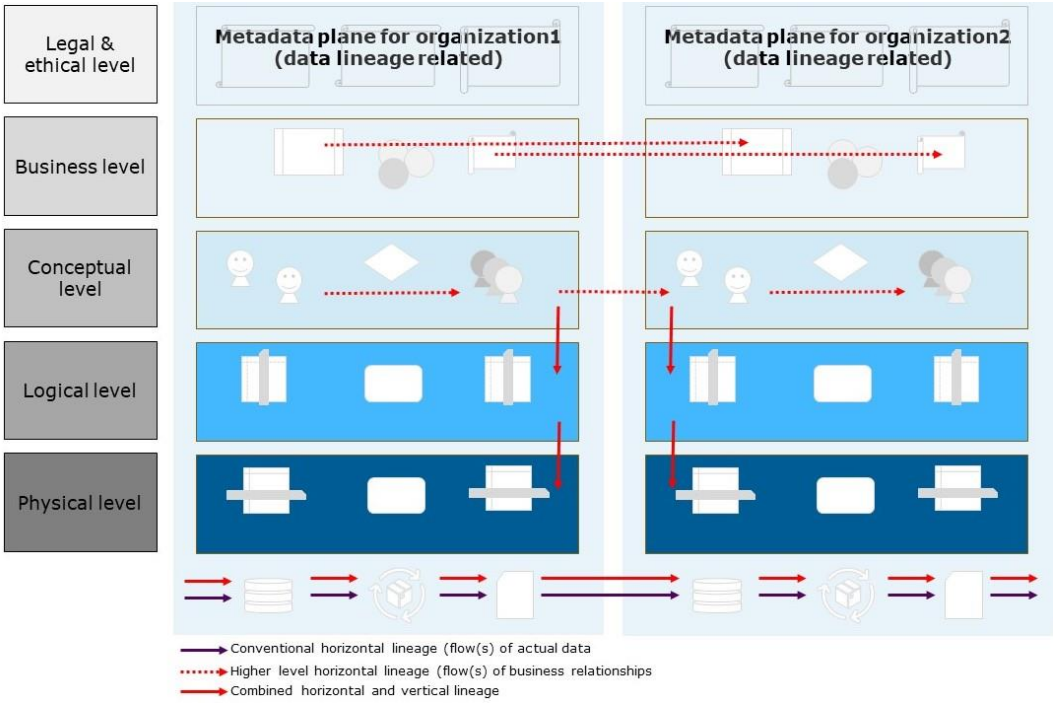
Note that the realization of the higher levels peer-to-peer relations in Figure 5.1 (e.g., that of the confederation) can be centralized (e.g., via a trusted third party) or be made peer-to-peer (e.g., by using a blockchain). As an example of the latter, Matsubara et al. (2020), propose an architecture for data lineage among collaborating organizations. The proposed architecture manages metadata related to data lineage hierarchically within organizations and uses a blockchain for exchanging metadata between organizations.

5.1.2 Intertwined vertical and horizonal lineage

In cross organizational settings we foresee an intertwined relationship between vertical and horizontal data lineage, which can be related to technical and business data lineage.

As illustratively shown in Figure 5.2, although horizontal data lineage is often prescribed for physical level objects, as indicated by continuous blue arrows in Figure 5.2, it might also be relevant for concepts at the higher abstraction levels, as indicated by dashed red arrows in Figure 5.2. For example, in the DJS setting, some business concepts do not have one-to-one relationships across the chain of organizations. This inconsistency requires defining a horizontal lineage for these concepts at the business and/or conceptual level across the chain of organizations. We expect less dynamicity and variation at the legal and ethical level within the DJS compared to those at the other levels, therefore this level is shown not predominantly in Figure 5.2. Based on this observation, we conclude that horizontal data lineage can be applicable to physical level objects (which corresponds to the technical lineage) as well as to business and conceptual level concepts in DJS settings (as also reflected in Table 2.1). Based on the abovementioned observation, we also conclude that in cross organizational settings, for a horizontal data lineage at the physical level we may need adopting a combined vertical and horizontal data lineage, as conceptually indicated by continuous red arrows in Figure 5.2. This is the reason to call this configuration as intertwined horizontal and vertical lineage.

Figure 5.2 Illustrating a mix of data lineage types across organizations



5.2 A development and deployment model of data-driven systems

There are several questions related to deploying data lineage tools and platforms in (cross) organizational settings as follows.

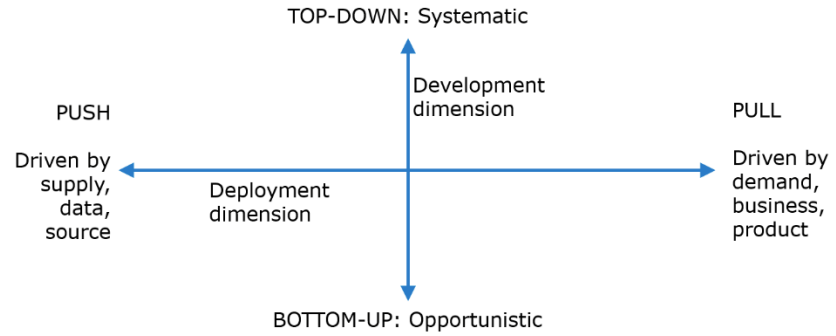
- Is the deployment of data lineage tools necessary, relevant, and cost effective?
- Should data lineage tools be deployed as standalone tools next to other data management tools such as data catalogs?
- Should data lineage be part of other data management tools such as data catalogs?
- How can these tools be introduced to and deployed in organizations? What are possible strategies and/or best practices?

Although all these questions are relevant, we intend to shed light on some aspects of the last question in this section. Specifically, we sketch two strategies that illustrate two boundary states in the solution space for deploying data lineage tools in organizations. We sketch these strategies to give insight in their potentials and challenges, which can be used by system designers to choose a solution in the spectrum delineated by these (and similar) strategies.

The discussion presented in this section is based on a so-called Data Quadrant Model (DQM) that is introduced in (Damhof, 2013) for *deploying data (driven systems)* in organizations. A similar model is also used in (Labadie et al., 2020), which reviews three use-cases (and strategies) from practice for deploying data catalog tools. This section presents the DQM.

The DQM, shown in Figure 5.3, has two dimensions: one capturing data deployment drivers and the other capturing data development styles.

Figure 5.3 The DQM for deploying data driven systems from (Damhof, 2013)



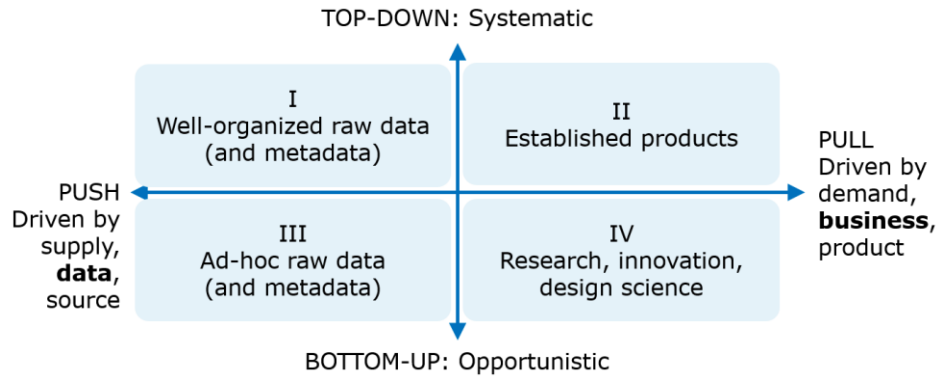
Drivers[9] *of data deployment* (the horizontal axe in Figure 5.3): Deployment of data-driven systems can start from resources and be based on existing raw data, thus being supply-driven (see the push system deployment, shown on the left side in Figure 5.3). Alternatively, deployment of data-driven systems can start from businesses and be based on desired data products, thus being demand-driven (see the pull system deployments shown on the right side in Figure 5.3). Push data deployments, which are supply/source/data-driven, work on a standardized way and offer low demand variability, low product personalization, better economics of scale, low manufacturing variability, low setup change costs, and low lead times (Damhof, 2013). Pull data deployments, which are demand/business/data-product driven, offer or possess the opposite properties mentioned above for push data deployments. As a coarse distinction, a push data deployment offers reliability, while a pull data deployment provides flexibility. We note that, in practice, there is a need for making a trade-off between reliability and flexibility.

Development[10] *styles of data deployment (*the vertical axe in Figure 5.3): Deployment of data-driven systems can be developed in systematic style or in an opportunistic style (Damhof, 2013). In the systematic style (the upper side in Figure 5.3), end-users and developers are separated, there is a tendency for centralized and controlled way of data deployment using IT services, processes, and products. In the opportunistic style (the lower side in Figure 5.3), there is no separation between end-users and developers, the tendency is to have a decentralized and collective control of data deployment, and deployment takes place in/during production.

The DQM specifies four operation regions, as shown in Figure 5.4, namely:
- *Region I:* Representing a well-organized raw data ecosystem, which is ready for making those data products that comply with the way that the data is organized.
- *Region II:* Representing well-defined data products that rely on well-organized raw data.
- *Region III:* Representing innovation and research space with known business needs but without well-organized and well available data, and
- *Region IV:* Representing unclear business needs/problems with ill-organized data.

[9] According to (Labadie et al., 2020), this aspect captures the preferred user type to implement the system.
[10] According to (Labadie et al., 2020), this aspect captures the order in which the tasks are performed.

Figure 5.4 Regions of the data quadrant model from (Damhof, 2013)

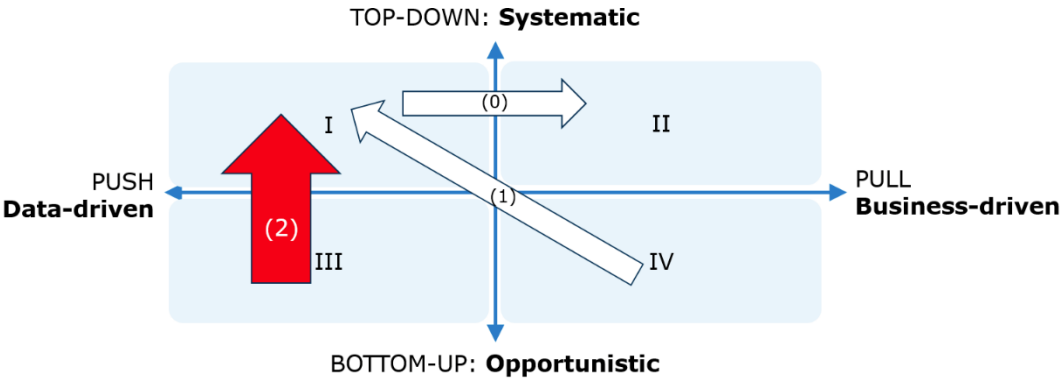


5.3 Two boundary deployment strategies

In this section we sketch two strategies for deployment of data driven systems (in our case, implementing the required data lineage functionality). These strategies are inspired by the work done in (Labadie et al., 2020) which investigates three strategies for deploying data catalog tools at three organizations, namely: Albaco enterprise, Strychem enterprise, Mom-and-Pop enterprise.

The first strategy, illustrated in Figure 5.5, assumes that we have well-organized raw data from which one can derive well-define products (like monitoring dashboards), as shown by arrow (0) in Figure schematically. Such systems have a drawback that they cannot address new (i.e., not previously thought of) business driven queries based on the existing well-organized raw data, see arrow marked by arrow (1) in Figure schematically. To be able to address such on-demand queries, one way is to organize a large amount of raw data proactively, hoping that they would become useful at some point in the future. This would impose a lot of burden for collecting and managing redundant data for just-in-case situations, as illustrated by arrow (2) in Figure schematically.

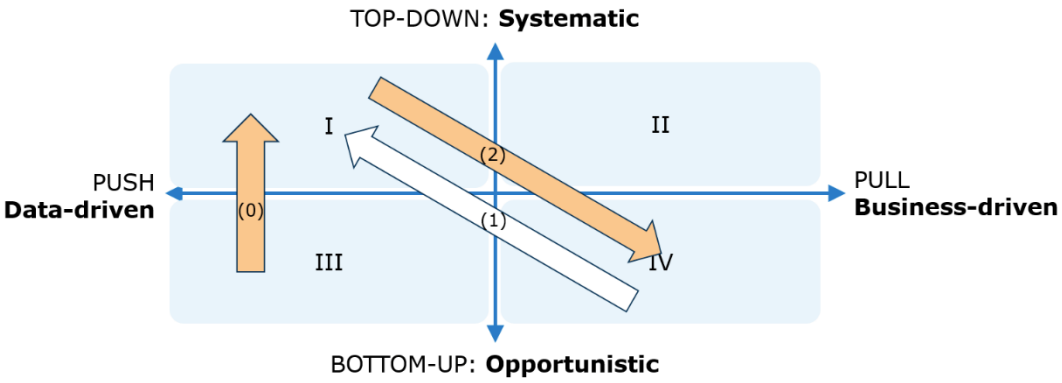
Figure 5.5 Strategy 1 with a sound-proactive-organization of raw data



The second strategy, illustrated in Figure 5.6, assumes that we have an affordable amount of well-organized data available, as indicated by arrow (0) in Figure 5.6 schematically. This available data is enough to provide good enough replies (with an

acceptable level of uncertainty) to unforeseen queries, as indicated by arrows (1) and (2) in Figure 5.6 schematically.

Figure 5.6 Strategy 2 with an opportunistic-proactive-organization of raw data



As evidence for the second data deployment strategy, we recall the lazy and eager lineage discussion in Section 2.4.5, where one strategy starts from lazy lineage to yield eager lineage, and another strategy starts from lazy lineage to yield super-eager lineage. Other examples of the second data deployment strategy are information search-engine based information retrieval (like Google) and LLM based information probing (like ChatGPT). A search engine or an LLM does not store all data it learns from. However, both search engine and LLM learn an abstract form of the data and use that to provide links to the original data (in the case of a search engine) or to compose a plausible reply (in the case of an LLM).

The abovementioned strategies capture the two extremes for making trade-offs between:

- Having certainty, while investing a lot in in-advance data organization (strategy 1).
- Having uncertainty, while investing an acceptable amount in in-advance data organization (strategy 2).

A data lineage system that should reply with a high certainty to queries of users (like per individual queries in operational situations within the DJS) should incline towards strategy 1. In situations where less certain replies to data lineage queries of users are acceptable (like when one is interested to know which data sources/documents are related to an existing policy), one may incline more towards strategy 2.

5.4 Concluding remarks

For the DJS, the piggybacking model can be a valid option if one opts for data source tracking and the data is not combined with other data streams. It may not be scalable if one is interested in tracking all data transformations and the data is expected to be combined with multiple data objects. Further, the piggybacking model alone is not helpful for the data lineage needs of the end-users who reside at the beginning of the data pipeline/journey.

In this section, we proposed considering a federated data lineage metadata management architecture for deploying data lineage in cross organizational settings such as that of the DJS. This model can be deployed in a centralized way per organization (or per department within an organization) and in a federated way across organizations (or across departments within an organization). Such a reliance on the existing organizational structure works well and can scale up organically as seen in the case of federated identity management among European universities.

So far horizontal data lineage has been prescribed for physical level data objects, often coined as technical data lineage. In cross organizational settings, however, we foresee that horizontal data lineage can also be applied at higher abstraction levels to, for example, business and/or conceptual level concepts. Further, we concluded that, in cross organizational settings, we may need adopting a combined vertical and horizontal data lineage for a horizontal data lineage at the physical level. This so-called intertwined horizontal and vertical lineage configuration is necessary to deal with interoperability issues of data lineage in cross organizational settings. A combination of vertical and horizontal lineage can be needed for some real-world cases in the DJS setting.

Automating data lineage related metadata collection is a must due to highly increasing amount of data being shared and transformed. Further, being business driven is necessary as data-driven work has become a common practice. This requires involving the end-users as well as the possibility of accommodating new queries by design in data lineage systems. Nevertheless, it is not feasible and efficient to collect and manage a huge amount of data lineage related metadata exhaustively in anticipation that one day some part of it would be useful for answering new business driven queries. To be cost effective, one may choose for collecting a reasonable amount of data lineage metadata and should a new query rise, go for either a targeted search (i.e., to carry out a zoom in search on a need-to-know basis) and/or for an approximate reply if having certain amount of uncertainty in replies is acceptable.

For some products one may use strategy 1 (making all metadata ready), while for some others strategy 2 is better (making a reasonable amount of metadata ready). Note that we do not claim that the two strategies mentioned cover the whole space of all possible data lineage deployment strategies.

6 Data lineage tools

In this section we aim at answering the fourth research question: "What are the capabilities (and limitations) of existing data lineage tools?" In Section 6.1, we explain a framework for evaluating data lineage tools based on specifying the data lineage capabilities (or criteria) that are relevant for an operation context. Subsequently, we present the result of our preliminary evaluation of a limited number of existing data lineage tools based on two chosen capabilities (or criteria) in Section 6.2. Finally, we recapture the main outcomes of the whole section in Section 6.3.

6.1 Evaluation framework

To evaluate data lineage software tools an organization must define the evaluation criteria that are relevant for its operational context. For this report, we rely on general criteria due to the nature of this study in being explorative and preliminary. As an example, Balm and Bakker (2024) define 14 criteria for evaluating data lineage tools, which are inspired by (Sankar, 2022; Steenbeek 2023) and their own experience with the practice. These criteria are: (1) openness: being open-source or commercial, (2) monetary costs, (3) standalone-ness: just for data lineage or more (data management) functionalities, (4) supported data lineage functions, (5) interfacing capability with software systems: how many connections being possible and whether being customizable, (6) granularity level: being how far from the column/code level, (7) performance and scalability, (8) visualization capability: how much being customizable, outcomes being exportable, etc., (9) user experience level, (10) security level, (11) support for compliance with regulations, (13) collaboration support: enabling multiple users to use the tool, and (14) documentation and troubleshooting support. Some other sets of evaluation criteria for data lineage tools are listed in (Atlan, 2023; Qlik, 2024; Stedman & Loshin, 2022).

The set of criteria for evaluating data lineage tools can be defined based on the needs and interests of the organizations and stakeholders involved. Specifically, the data lineage criteria that might be of interest can be determined by the type of the queries posed (see Section 3.3), the sought objectives in deploying data lineage (see Section 3.2), and the desired data lineage characteristics (see Section 2.4). The chosen criteria may be grouped in different categories like being data lineage functionality related, performance related, and monetary cost related.

After determining which criteria are relevant in a setting, one can specify several granularity levels per each criterion. These levels can be defined qualitatively, possibly based on some insights from the operation context and setting. Each qualitatively defined level can be assigned a (meaningful) numerical value (or a score) to make the outcome of each criterion quantitative. For those quantified criteria, one can use numerical methods to relate/merge the criterion scores (like averaging and thresholding). For example, the monetary costs can be assigned two outcomes: being commercial and being open source (thus, needing financial resources and being free of charge, respectively). Considering monetary costs as a thresholding type, one can assign binary values 0 and 1 to the defined levels of commercial and open source, respectively. For an evaluation instance, one may opt for open-source tools, thus consider only those tools for which the assigned score of the monetary cost criterion is

1. For another subset of the criteria that are relevant for evaluating and deciding on a tool, one may apply the averaging operator to quantify the scores of the corresponding aspects in a numerical value (a total score). For example, Balm and Bakker (2024) apply averaging to the scores of their criteria numbered 4, 5, 6, 8, 13 and 14, where the score per criterion can take one of values 1, 2, 3, 4, and 5. A concern in scoring and merging the evaluation criteria is the possibility of having dependency or correlation between some of them, which should be accounted for appropriately. See (Bargh et al., 2022) for applying a similar approach for measuring the degree of data openness.

The ways to measure the evaluation criteria can differ per criterion and/or per tool (Balm & Bakker, 2024). For some, like being open-source or commercial, one can analyze publicly available documentation of a tool to infer the score. For some, like user experience, one should employ empirical studies with the tool. Conducting such investigations requires having access to (a beta version of) the tool or to obtain a research license for experimenting with the tool. One might use also publicly available reports and papers, like (Balm & Bakker, 2024), keeping in mind that their evaluation might not fit the operation setting in mind.

6.2 An example evaluation

A comprehensive evaluation of data lineage tools and technology has not been within the scope of this project. However, in this preliminary phase, as mentioned in Section 4.1.2, it was suggested to conduct a preliminary evaluation of some important (popular) data lineage tools based on the following two criteria.

- *Business lineage vs technical lineage* (see Section 2.4.8), which relates to horizontal (at the physical level) vs vertical lineage (see Section 2.4.9), and
- *Data origin vs data flow lineage*[11] (see Section 2.4.1).

In Section 6.2.1, we explain how the data lineage tools are selected and in Section 6.2.2 we report on a preliminary evaluation of the above-mentioned criteria.

6.2.1 Selected tools

In this study we consider two categories of commercial and open-source data lineage tools separately. The study of (Balm & Bakker, 2024) shows that commercial tools generally provide a wide range of data lineage functionalities together with other functionalities (like data management and data catalog). As such, they are suitable for large organizations which can afford paying the expenses of such tools and need to use a wide range of functions that these tools provide. Open-source tools generally offer a limited subset of data lineage functionalities, are cost free, and integrate-able with other (open-source) applications. As such, they are suitable for low-budget, small enterprises to use these tools or customize them to their data lineage needs (Balm & Bakker, 2024). Note that an extensive integration of open-source tools requires inhouse technical skills or extra budget that might not be available in small organizations (idem). Further, open-source tools might be useful for conducting small scale experimentations within the DJS to gain some hands-on experience about data lineage technology.

[11] The project requester mentioned the terms provenance vs lineage. Data origin lineage is the adopted definition for data provenance in this report.

In (Stedman & Loshin, 2022) the vendors of commercial data lineage tools are divided in the following types (non-exhaustive):

- Large IT vendors that are providers of data management platforms (e.g., IBM, Informatica, Microsoft, Oracle, SAP and SAS) and cloud platform providers (e.g., AWS and Google Cloud),
- software vendors with broad product portfolios that also offer data management and governance tools (e.g., Hitachi Vantara, OneTrust, Precisely and Quest Software),
- vendors with a focus on offering data management and governance (e.g., ASG Technologies, Ataccama, Boomi, Collibra, Semarchy, Syniti and Talend),
- metadata management and data lineage specialists (e.g., Alex Solutions, Manta and Octopai),
- vendors of data catalog tools (e.g., Alation, Atlan, Data.world and OvalEdge), and
- vendors of self-service data preparation software for data engineers and analytics teams that also have data lineage capabilities (e.g., DataRobot and Alteryx's Trifacta unit).

Within the DJS, the interviewees mentioned a few tools that they use in a small scale or experimentally for fulfilling some aspects of data lineage. The tools mentioned are PowerBI (a tool for data visualization), Snowflake (a platform for data management – data engineering, data analytics, machine learning, data sharing among federation of organizations – in cloud), Informatica (a platform for processing and managing large amounts of data), and Collibra (a platform for metadata management for data cataloging, data management and data governance).

In their evaluation of data lineage tools, Balm and Bakker (2024) consider 30 commercial data lineage tools and choose the following 7 tools based on the volume of available online resources about them (i.e., their "perceived popularity"). These 7 tools are Alation, Atlan, Colibra, Informatica, Manta, Microsoft Purview, and Zeenea. For this preliminary study we choose these 7 tools as well, as two of them (Informatica and Colibra) are also mentioned by the interviewees at the DJS. Although, as mentioned above, open-source tools might not be suitable for deployment in large organizations like the DJS, we list 9 of them in the following from (Balm & Bakker, 2024) as they might be used for small scale experimental studies about data lineage. These tools are Amundsen, Apache Atlas, DataHub, Egeria, Kylo, Marquez, OpenMetaData, Spline, and Tokern. One can find other sets of data lineage related tools from the literature and websites, see for example (King, 2022).

6.2.2 Evaluation results

To evaluate data lineage tools based on the two desired criteria of (a) data origin vs data flow and (b) business lineage vs technical lineage, we also provide the results of the evaluation in (Balm & Bakker, 2024) for their two criteria of (a') granularity and (b') types of lineage because they are closely related to criteria (a, b). Note that criteria (a', b') differ from criteria (a, b) slightly.

In (Balm & Bakker, 2024) criterion (a') granularity, is described as follows. "From extracted lineage, one would want to see as much information as possible. However, not all tools can extract lineage down to the level of columns. This criterion answers questions such as: Is lineage possible to the column level? Is it possible to see code of ETL transformations?" Subsequently, the whitepaper defines the following 5 scores for the granularity of data lineage:

- *Poor (score 1)* means no support for column level lineage and not being able to see the ETL source codes.
- *Below average (score 2)* means support for column level lineage for up to 5 sources.
- *Average (score 3)* means support for column level lineage for up to 20 sources or being able to see the ETL source codes.
- *Good (score 4)* means column level lineage support for up to 35 sources or being able to see the source code of ETL jobs.
- *Excellent (score 5)* means column level lineage is support for at least 35 sources and it is possible to see the source code of ETL jobs.

In (Balm & Bakker, 2024) criterion (b') types of lineage is described as follows. "Does the software platform provide information on both business lineage and technical lineage?" Subsequently, the whitepaper defines the following scores for the types of data lineage:

- *Below average (score 2)* means no native support for data lineage.
- *Average (score 3)* means only one type of lineage is supported, which is always technical data lineage, considering the tools investigated.
- *Good (score 4)* means that in addition to technical lineage there is also support for business lineage in some way (like support for the creation of a business glossary).
- *Excellent (score 5)* means visualization of both technical and business lineage is supported.

Table 6.1 presents the evaluation of (Balm & Bakker, 2024) for the chosen commercial tools based on criteria (a', b'), i.e., criteria granularity and types of lineage (for the latter see the column named as "Business-technical *"). The desired criteria (a, b) are given in columns "data flow (& origine)" and "more information about the business lineage aspect", respectively. Column (b) can be seen as the clarification of criterion (b').

Table 6.1 An evaluation of 7 Commercial data lineage tools

Commercial tools	Granularity *	Data flow (& origin)	Business-technical *	More information about the business lineage aspect *
Alation	**4**	√	4	Technical assets can be given a business meaning
Atlan	**5**	√	4	A business glossary can be created
Colibra	4	√	**5**	Supports both technical and business lineage types
Informatica	**5**	√	4	Technical assets can be associated with business terms
Manta	**5**	√	**4.5**	Abstract some factors to make them more understandable
Microsoft Purview	4	√	4	A business glossary can be created
Zeenea	4	√	4	A business glossary can be created

* From (Balm & Bakker, 2024)

Similarly, Table 6.2 presents the evaluation of (Balm & Bakker, 2024) and ours for the chosen open-source tools.

Table 6.2 An evaluation of 9 open-source data lineage tools

Open-source tools	Granularity *	Data flow (& origin)	Business-technical *	More information about the business lineage aspect *
Amundsen	4	√	2	No support for business lineage
Apache Atlas	**2**	√	4	A business glossary can be created and linked to technical lineage, but not visualized
DataHub	3	√	**4**	A business glossary can be created and linked to technical lineage
Egeria	**5**	√	5	Supports both technical and business lineage types
Kylo	**2.5**	√	**4**	Business lineage looks like that can be defined manually
Marquez	4	√	3	No support for business lineage
OpenMetaData	4	√	4	A business glossary can be created
Spline	3	√ (single input & output)	3	No support for business lineage
Tokern	2	√ (suspect)	3	No support for business lineage

* From (Balm & Bakker, 2024)

The results of Table 6.1 and Table 6.2 show that all software tools investigated support both data origin and data flow lineage reasonably.

6.3 Concluding remarks

The set of criteria for evaluating data lineage tools can be defined based on the needs and interests of the organizations and stakeholders involved. To determine the lineage criteria, one should elucidate the data lineage queries (see Section 3.3) and the data lineage objectives (see Section 3.2) that DJS organizations seek in deploying data lineage and, based on those objectives, elucidate the desired data lineage characteristics (see Section 2.4). Knowing the data lineage objectives and characteristics can contribute to determining the criteria for evaluating data lineage software tools.

The chosen criteria may be grouped in different categories like being data lineage functionality related, performance related, and monetary cost related. After determining which criteria are relevant in a deployment setting, one can specify several granularity levels per each criterion. These levels can be defined qualitatively, possibly based on some insights from the operation context and setting. Each qualitatively defined level can be assigned a (meaningful) numerical value (or a score) to make the outcome of each criterion quantitative. For those quantified criteria, one can use numerical methods to relate/merge the criterion scores (like averaging and thresholding). A concern in scoring and merging the evaluation criteria is the possibility of having dependency or correlation between some of them, which should be accounted for appropriately.

The ways to measure the evaluation criteria can differ per criterion and/or per tool. For some, like being open-source or commercial, one can analyze publicly available documentation of a tool to infer the score. For some, like user experience, one should employ empirical studies with the tool (Balm & Bakker, 2024). Conducting such investigations requires having access to (a beta version of) the tool.

Based on the results of the previous sections, we foresee considering the following criteria for evaluating data lineage software tools for the DJS setting. These criteria include having flexibility in being business driven (replying to new and unforeseen data lineage related queries), having more granular lineage than attribute level (like being cell level), having interoperability with other tools (from other organizations), and offering good user experience and useability. The latter criterion captures the ease of use which can be realized by, e.g., anonymization of the metadata to be shared to deal with business sensitivity and privacy issues, and using visualization based and/or natural-language based querying (thus, facilitating the democratization of data lineage). “Remember, the best tool for you depends on your specific needs, the complexity of your data environment, and the skill level of your users” (Atlan, 2023).

Commercial software tools generally provide a wide range of data lineage functionalities together with other functionalities (like data management and data catalog). As such, they are suitable for large organizations which can afford paying the expenses of such tools and need to use a wide range of functions that these tools provide. Open-source tools generally offer a limited subset of data lineage functionalities, are cost free, and are integrate-able with other (open-source) applications. As such, they are suitable for low-budget, small enterprises to use or customize these tools to their data lineage needs. Note that an extensive integration of open-source tools requires inhouse technical skills or extra budget that might not be available in small organizations (Balm & Bakker, 2024). Further, the open-source tools

might be useful for conducting small scale experimentations within the DJS to gain some hands-on experience about data lineage technology.

All evaluated software tools, either being commercial or open-source, support both types of data flow lineage and data origin lineage. As such, the dimension data origin/flow does not make any distinction and should not be considered as a criterion for choosing among these tools.

7 Conclusion

In this section, we draw our conclusions from the discussions made about the research questions in the previous sections in Section 7.1. Subsequently, we provide three avenues for future research and development in Section 7.1.

7.1 Reflection on the research questions

To investigate how data lineage technology can contribute to data governance and data management within the DJS (Dutch Justice System) we have conducted a preliminary study by posing four research questions. The study outcomes can be a knowledge base for informing the design and deployment process of the technology in the DJS setting. In this section we reflect on the discussion results in the previous sections aimed at answering these research questions.

7.1.1 What is data lineage?

Based on some existing definitions and the insights gained during the study on data lineage, we defined data lineage as *the description of data movements and transformations at various abstraction levels along data journey paths. The description encompasses those aspects that are of interest in an application context like how (i.e., by whom, when, where, which, etc.) data objects are processed (i.e., created, collected, stored, accessed, transformed, transmitted, etc.) and how they are related to high-level data concepts.* This definition conceptualizes not only the physical distribution of data objects (like data origins, flows and transformations), but also the semantical distribution of the related concepts (like the business, legal and organizational terms that relate or apply to those data objects). Further, the definition offers a means to limit the scope of data lineage to those aspects that are of interest in each context.

Data lineage contributes to gaining trust in data and in responsible data transformation and sharing. Data lineage relies on deriving and managing metadata that is relevant for the aimed data lineage usage objective(s), which are listed in Section 7.1.2. Data lineage can be characterized from various aspects, which are not necessarily independent. These aspects include (a) data origin vs data flow lineage, (b) where vs how data lineage, (c) data transformation types, (d) coarse-grained vs fine-grained data lineage, (e) lazy vs eager data lineage, (f) backwards vs forward data lineage, (g) tracing vs tracking data lineage, (h) technical vs business data lineage, and (i) horizontal vs vertical data lineage. These data lineage characteristics specify the technical space in which a data lineage solution can be designed, chosen, and/or deployed.

7.1.2 Which objectives can data lineage contribute to?

The concept of lineage has been applied to a wide range of cases, ranging from tracking/tracing the computation flows in a single software program to tracing/tracing the data flows in distributed ISs. Data lineage can contribute to many objectives, each of which, in turn, plays a role in enhancing trust in data, data sharing, and data-driven applications and policymaking. These objectives include (a) data governance,

(b) privacy protection, (c) trusting AI models, (d) data and AI explainability, interpretability and fairness, (e) data quality management, (f) data change management, (g) data ownership, (h) regulatory compliance, audit and accountability, (i) data security, (j) data modeling, and (k) data discovery. These objectives capture the usage areas of data lineage and, as such, they specify the *societal relevancy of data lineage* at large.

It would be intriguing to aim at all mentioned objectives when deploying a data lineage solution. Such a versatile data lineage solution could immediately become too complex and costly, thus might become impossible to realize especially in distributed settings (e.g., among the semi-autonomous organizations of the DJS). Knowing the relevant data lineage objectives, one can determine which characteristics of data lineage are relevant in an operational setting. For example, for some data transformations (like in image processing) the how-lineage for an output tuple may be affected by a subset of data items in the input dataset(s). In these cases, an output-view, instance-level how-lineage is relevant. Based on the required data lineage characteristics, one can determine (the type of) data lineage metadata to be collected. Subsequently, for storing and retrieving data lineage metadata, the deployment environment and its organizational structure are determining factors to decide on the architecture of data lineage metadata management for collecting, storing, querying, processing, and retrieving data lineage data/information.

To determine the data lineage objectives relevant in each operation context, it is imperative to start with identifying the potential users of such a data lineage system, and their data lineage needs (i.e., to be business driven). A way to elucidate these needs is to identify the typical queries that the potential users (would) want to be answered with the data lineage system. One can subsequently analyze these typical queries and determine whether they are fixed/predetermined or should be flexible. For the latter, one needs to get insight in how much flexibility is needed. The analysis result can subsequently be mapped to the objectives that a desired data lineage system must fulfill. As mentioned above, having an insight in all these objectives will impact the architecture of and the tools used for data lineage.

7.1.3 How can data lineage tools be deployed?

Based on our literature study and expert interviews, we elucidated that data lineage within the DJS can (or is required to) contribute to data governance, data discovery, data quality management, data change management, and privacy and security mainly. Further, we draw the following conclusions for data lineage in the DJS setting.

- It is necessary to trustfully share data within and across organizational boundaries in the DJS while allowing participating organizations maintain their autonomy and have own business, conceptual and logical data models and ISs. As such, data lineage within the DJS should account for diversity at all levels, namely at technical, logical, conceptual, and business levels.
- Within the DJS, data is shared and processed for not only research and strategic purposes, but also for operational purposes. Thus, the shared data could be at various aggregation levels, i.e., at group and individual levels, which requires having data lineage at coarse-grained and fine-grained levels.
- The end-users of data lineage within the DJS can have different backgrounds with a varying set of data (science) skills and may reside at the beginning, middle or end of data pipeline. Therefore, there should be enough flexibility to serve a wide range of users and aim for the democratization of data lineage.

- We defined five abstraction levels for data lineage namely (a) physical, (b) logical, (c) conceptual, (d) business, and (e) legal and ethical levels. The legal and ethical level is particularly important in the DJS, as the DJS is responsible for overseeing and safeguarding the rule of law in the society.
- In cross organizational settings (like that of the DJS), we foresaw that horizontal data lineage can be applied at not only physical level but also at business and conceptual levels. Further, we may need adopting a combined vertical and horizontal data lineage for enabling a horizontal data lineage at the physical level to deal with interoperability issues of data lineage (i.e., having an intertwined horizontal and vertical lineage configuration).

Metadata management in the DJS setting should be scalable and distributed as well as should fit the organizational structure of the DJS. Considering the cross organizational structure of the DJS, we proposed considering a federated data lineage metadata management architecture for deploying data lineage. A federated architecture relies on the existing organizational structure, which, therefore, can scale up organically as seen in the case of federated identity management among European universities.

For managing data lineage metadata, automizing the metadata collection process is necessary, considering the high speed and large volume at which data is collected and shared nowadays. For storing data lineage metadata, the type of data repository should be chosen according to the data lineage characteristics needed and the context in which data lineage operates. According to the type of the data repository chosen, one can opt for an appropriate query processing language. System-user interactions can be facilitated in different ways, depending on the technical skills of the users.

Being business driven is necessary as data-driven working has become a common practice nowadays. This requires involving non-technical end-users in the design process as well as the possibility of accommodating new queries by design in data lineage systems. Nevertheless, it is not feasible and efficient to collect and manage a huge amount of data lineage related metadata exhaustively in anticipation that one day some part of it would be useful for answering new business driven queries. To be cost effective, one may choose for collecting a reasonable amount of data lineage metadata (i.e., collecting metadata coarsely) and should a new query rise, go for either a targeted search (i.e., to carry out a zoom-in search on a need-to-know basis) an/or for an approximate reply if having certain amount of uncertainty in replies is acceptable. It may be needed to deploy natural-language-based user interfaces which are capable of handling both predefined and not predefined queries. Hereby one can make data lineage become business-driven and usable for users with limited technical backgrounds (like business consultants and policymakers).

7.1.4 What are the capabilities (and limitations) of existing data lineage tools?

Determining the relevant capabilities of data lineage tools in each usage context is important as they can be used as criteria for evaluating the existing tools and choosing the one that fits the context the best. Based on the results of the study, we foresaw several relevant capabilities for data lineage software tools in the DJS setting. The main capabilities are to have flexibility in being business driven (replying to new and unforeseen data lineage related queries), to have more granular lineage than attribute level (like being cell level), to have interoperability with other tools (from other organizations), and to offer good user experience and useability. The latter criterion captures the ease of use which can be realized by, for instance, anonymization of to be

shared metadata to deal with business sensitivity and privacy issues, and use of visualization based and/or natural-language based querying for collecting metadata (thus, facilitating the democratization of data lineage).

As part of the answer to this research question, we sketched a framework for evaluating data lineage tools. Based on this framework, one should define a set of evaluation criteria by, for example, elucidating the objectives that DJS organizations seek in deploying data lineage, elucidating the desired data lineage characteristics, and defining evaluation criteria accordingly. Subsequently, one can specify several granularity levels per each criterion. These levels can be defined qualitatively, i.e., be assigned a (meaningful) numerical value or a score. Finally, one can merge quantified criteria using numerical methods like averaging and thresholding.

Commercial software tools generally provide a wide range of data lineage functionalities together with other functionalities (like data management and data catalog). As such, they are suitable for large organizations which can afford paying the expenses of such tools and which need using a wide range of functions these tools provide. Open-source tools generally offer a limited subset of data lineage functionalities, are cost free, and are integrate-able with other (open-source) applications. As such, they are suitable for low-budget, small enterprises to use or customize these tools to their data lineage needs. Note that an extensive integration of open-source tools requires inhouse technical skills or extra budget that might not be available in small organizations. Further, the open-source tools might be useful for conducting small-scale experimentations within the DJS setting to gain some hands-on experience about data lineage technology.

We reported on an evaluation of 7 commercial tools and 11 open-source software tools on two criteria: data origine vs data flow lineage and technical vs business lineage. All commercial and open-source tools considered, support both types of data flow lineage and data origin lineage. As such, the dimension data origin/flow does not make any distinction and should not be considered as a criterion for choosing among these tools.

7.2 Recommendations for future research

Several directions are identified for future research during project execution. In this section we group them in three categories, organized from more practice-oriented research one to more applied research one.

7.2.1 Need for requirement elicitation study

For choosing a (set of) data lineage tool(s) that can be experimented with (or deployed) in the DJS setting we recognize the need for eliciting the requirements with which the tool(s) should comply. To this end, we foresee the following action points:

- Identifying the typical end-users and their data lineage queries (see Section 3.3),
- identifying whether there is a need for adjusting the end-users' queries in the future,
- defining the desired data lineage objectives/characteristics (see Section 2.4), and
- defining the data lineage tool evaluation method by determining the evaluation criteria and how to measure and merge them (see Section 6.1).

When the intention is to deploy data lineage in the DJS setting, there is a need for a further study of a suitable structural and functional architecture for data lineage deployment in DJS setting (see Sections 5.1 and 5.2).

Another direction for research is to investigate the requirements and ways for mixing vertical and horizontal data lineages at the boundaries of organizations. This requires mapping between data semantics at the borders of collaborating organizations and dealing with uncertainties that may be caused due to this mapping. To this end, gaining hands on experience with data lineage tools in real world cases can be useful.

7.2.2 Need for democratization of data lineage

There is a need for further research on making data lineage become business-driven and usable for users with limited technical backgrounds (like business consultants and policymakers). To this end, investigating methods and tools for natural-language-based user interfaces is a promising direction. Large Language Model (LLMs) can be considered for mapping between natural language texts to formal database queries (e.g., SQL). This direction may require investigating ways to deal with uncertainty in the mapping between natural and formal languages.

7.2.3 Need for effective and efficient data lineage

Reducing the complexity of data lineage and the associated costs is a crucial factor in successful adoption of data lineage technology.

For data lineage related metadata collection, developing automated methods is necessity. For example, the use of LLMs can be investigated for (semi)automatically creating business level metadata (like a report in natural language that describes the technical analyses conducted on the data for business-level end-users) from technical level metadata (e.g., from data query scripts). In this way, the burden of business-level data lineage metadata creation on technical experts can be alleviated.

In conventional lineage it is assumed that users can understand how an output is created by observing the source data and knowing that the data transformation is a sequence of simple operations like filter, join, and aggregation. However, in complex data analysis, like using AI/ML algorithms, more information about data transformation is needed. A question that may arise is which data lineage information should be provided to explain and/or influence the outcomes of very complex data transformations (like LLMs) and how this data lineage information should be managed in a (cost) effective way.

An issue in data origin lineage is how to determine data origins in each setting. We suspect that there might be multiple views with different data origins (specially in cross organizational settings). If this conjecture holds, then lineaging data origin boils down to or, better said, requires lineaging data flows. It is for future research to investigate how to determine data origins (or data destinations) in each operation setting.

Samenvatting

Data lineage voor het rechtsstelsel

Toepassingsgebied, potentieel en aanbevelingen

Probleemstelling

Probleem (context)

Data worden momenteel in een steeds sneller tempo gegenereerd, verzameld, gedeeld, geanalyseerd en verspreid. Als gevolg van deze ontwikkeling is er een toenemende belangstelling voor (en vraag naar) methoden om de beschikbare data bij elkaar te brengen, met behulp van (geavanceerde) algoritmen te analyseren en datagestuurde systemen te ontwikkelen om ons dagelijks leven te verlichten, toegevoegde waarde te creëren voor bedrijven, inzicht te geven in maatschappelijke fenomenen, en beleidsvormingsprocessen te sturen. De wijze waarop data worden verzameld, wat vaak gepaard gaat met subjectieve, partijdige, foutieve, gevoelige en stigmatiserende informatie over individuen, groepen en organisaties, en de wijze waarop algoritmen en datagestuurde systemen worden ontworpen, geïmplementeerd, geïnterpreteerd en (verkeerd) gebruikt, hebben grote invloed (of gaan dat hebben) op ons als individu, als groep en als maatschappij. Als een organisatie profijt wil hebben van data, moet zij dus alert zijn op de risico's van de data die worden gebruikt om persoonlijke, sociale of organisatorische voordelen te bieden. Vertrouwen scheppen in data is dan ook een absolute voorwaarde voor het respecteren van fundamentele mensenrechten zoals privacy, vrijheid, autonomie en waardigheid.

Ook op justitieel gebied, met name in het Nederlandse rechtsstelsel, worden in toenemende mate digitale technologieën en datagestuurde systemen toegepast. De informatiesystemen in het justitieel apparaat, die gegevens verzamelen, opslaan, delen en verwerken, zijn vaak fysiek verspreid, hebben veel losjes gekoppelde subsystemen en worden beheerd door verschillende organisaties (d.w.z. verspreid over veel administratieve domeinen). Om in deze setting data te gebruiken, moeten verschillende informatiebronnen met elkaar worden verbonden en moeten de data op een betrouwbare en verantwoorde manier worden geïntegreerd. Degenen die data delen (zoals gerechtelijke dienstverleners) moeten op de datagebruikers vertrouwen dat zij de data op verantwoorde wijze gebruiken, en degenen die data consumeren (zoals beleidsmakers) moeten er vertrouwen in hebben dat gegevensbronnen op verantwoorde wijze gegevens verzamelen en delen.

Vertrouwen in dataproductie en -gebruik vereist onder meer het terugdringen van de beveiligingsproblemen van grotendeels verspreide gegevensafhankelijke systemen, het managen van datakwaliteitsproblemen van losjes gekoppelde databronnen en het aanpakken van onrechtmatig en/of kwaadwillig gebruik van data en algoritmeresultaten. Gezien het belang van gegevensuitwisseling enerzijds en de toegenomen complexiteit van gegevensuitwisseling tussen (de vele) belanghebbenden en informatiesystemen anderzijds, is het noodzakelijk een passend data-ecosysteem tot stand te brengen. Een dergelijk data-ecosysteem vereist solide en effectieve data-

governance en effectief databeheer om de kwaliteit van data te waarborgen, de opslag en uitwisseling van data te beveiligen, de wisselwerking tussen concurrerende waarden (zoals datagebruik en dataprivacy) te optimaliseren, en de beginselen van vindbaarheid, toegankelijkheid, interoperabiliteit en herbruikbaarheid voor de data te operationaliseren.

Een efficiënte en effectieve data-governance/-management vereist onder andere kennis over de geschiedenis van data (hierna *data lineage*) en de herkomst van data (hierna *data provenance*). Data lineage verwijst naar het proces van traceren van de datastroom in de tijd, d.w.z. tijdens de levenscyclus/het traject van de data. Met metadata wordt een duidelijke beschrijving gegeven van de herkomst van de data, de datatransformaties langs het datatraject en de bestemming(en) van de data. Een soortgelijk geval van data lineage is data provenance, d.w.z. het bijhouden van de herkomst (de oorsprong) van data en de historische veranderingen ervan.

Data lineage wordt gezien als een essentieel instrument om de betrouwbaarheid van data te verbeteren. Het maakt bijvoorbeeld efficiënt beheer van datakwaliteit, datawijzigingen en datalevenscycli mogelijk. Een bekende metafoor die gebruikt wordt om de rol van data lineage bij het scheppen van vertrouwen in data te illustreren is het scheppen van vertrouwen in de voedzaamheid en gezondheid van een appel door kennis over hoe een appel wordt gekweekt, geoogst, vervoerd, opgeslagen, gedistribueerd en verkocht. Om vertrouwen te scheppen, zou men de relevante informatie (informatie over de ontstaansgeschiedenis) in elke fase van de levenscyclus van de appel (d.w.z. de route die de appel aflegt) in de toeleveringsketen kunnen bijhouden. In deze metafoor is het object in kwestie een fysiek object (product). Eenzelfde vorm van vertrouwen is ook belangrijk voor een digitaal object (bijv. een dataset, een digitale afbeelding, of een digitaal document). Zo worden burgers geconfronteerd met een steeds grotere hoeveelheid multimedia-content (oftewel data) in verschillende formaten, zoals afbeeldingen, video's, audio-opnamen en documenten. Deze content wordt vaak vermengd met desinformatie (bijv. foto's gegenereerd door generatieve AI) of gemanipuleerde informatie (bijv. foto's bewerkt met behulp van Photoshop), waardoor het voor gewone mensen (en zelfs voor professionals) moeilijk is om onderscheid te maken tussen echte en nep-/gemanipuleerde content. Bij het delen van foto's kunnen fotomakers, uitgevers en consumenten met behulp van data lineage erachter komen hoe en door wie een foto is gemaakt en welke bewerking(en) de foto heeft ondergaan gedurende de hele levenscyclus/ontwikkelingstraject. De herkomstinformatie, die betrouwbaar aan de foto kan worden toegevoegd en de foto tijdens zijn hele reis vergezelt, kan eenvoudig worden ingezien door consumenten stroomafwaarts in de pijplijn, zodat ze op de hoogte zijn van de herkomst en bewerkingsgeschiedenis van de foto. Deze kennis kan consumenten stroomafwaarts in de pijplijn helpen vertrouwen te krijgen in de content die ze tegenkomen op bijvoorbeeld sociale media en nieuwsfeeds. Binnen het Nederlandse rechtsstelsel kan data lineage op soortgelijke wijze bijdragen tot groter vertrouwen in bestaand beleid, bijvoorbeeld door vragen te beantwoorden als *welke datasets of documenten worden gebruikt als bewijs om het beleid te staven*.

In dit verslag beschrijven we de resultaten van ons verkennend onderzoek naar data lineage (en data provenance)-technologie, met name over de gedachte achter data lineage en de methoden/tools die gebruikt worden voor data lineage. De onderzoekscontext heeft betrekking op de data die worden verzameld, gedeeld, opgeslagen en verwerkt door informatiesystemen binnen het Nederlandse rechtsstelsel.

Doel van het onderzoek en onderzoeksvragen

Het doel van het onderzoek is erachter te komen hoe data lineage-technologie kan bijdragen aan data-governance en databeheer binnen het Nederlandse rechtsstelsel. Om dit doel te bereiken, moeten de voordelen van data lineage-technologie en de toepassingsmogelijkheden (en uitdagingen) binnen het Nederlandse rechtsstelsel worden onderzocht.

Dit is een vooronderzoek naar bovengenoemde doelstelling. De onderzoeksaanpak kan worden gekarakteriseerd als verkennend, waarbij we antwoorden zoeken op de volgende onderzoeksvragen:

1 *Wat is data lineage?* Om deze vraag te beantwoorden, zullen we ook de context (of het data-ecosysteem) waarin data lineage wordt gebruikt, beschrijven.
2 *Aan welke doelstellingen kan data lineage bijdragen?* Bij het beantwoorden van deze onderzoeksvraag gaan we in op de potentiële voordelen van data lineage.
3 *Hoe kunnen data lineage-tools worden ingezet?* Om antwoord te vinden op deze vraag, gaan we dieper in op de gebruikelijke manieren waarop data lineage wordt ingezet, en de uitdagingen die daarbij spelen.
4 *Wat zijn de mogelijkheden (en beperkingen) van bestaande data lineage-tools?* Om deze vraag te beantwoorden, schetsen we een kader voor het specificeren van de relevante data lineage-mogelijkheden. Voor een beperkt aantal bestaande data lineage-tools schetsen we twee toepassingsmogelijkheden die van belang zijn voor dit onderzoek.

Toepassingsgebied

Deze onderzoeksvragen zullen worden behandeld in het kader van het Nederlandse rechtsstelsel, dat bestaat uit veel semi-autonome organisaties die gezamenlijk de beginselen van de rechtsstaat in de Nederlandse samenleving waarborgen. De relaties tussen deze organisaties worden vaak gekenmerkt als een lineaire keten (waarbij een fase moet worden afgesloten voordat de volgende fase kan beginnen), met soms lussen en parallelle relaties. De term 'rechtsstelsel' verwijst naar de organisaties in het rechtsapparaat die betrokken zijn bij het produceren van data, variërend van wetteksten tot rechterlijke uitspraken. De werkingssfeer van het rechtsstelsel is ruimer dan dat van rechtbanken en gerechtelijke procedures.

In deze bijdrage willen we een overzicht geven van enkele belangrijke aspecten die kunnen worden overwogen bij het inzetten van data lineage in de organisatorische setting van het Nederlandse rechtsstelsel. Ons doel is niet om in deze bijdrage een oplossing te formuleren of voor te schrijven voor de implementatie van data lineage. De doelgroep van dit onderzoeksverslag zijn systeemontwerpers en -architecten, datafunctionarissen en dataspecialisten. Dit rapport heeft tot doel deze groepen te informeren over de ontwerpruimte waarbinnen zij een data lineage-oplossing kunnen ontwerpen of kiezen.

Methodologie

Voor dit onderzoek hebben we de literatuur kritisch bestudeerd, waarbij verschillende geselecteerde informatiebronnen zijn geanalyseerd, en hebben we nagedacht over de bestaande concepten, methoden en benaderingen. De geselecteerde informatiebronnen zijn niet alleen afkomstig uit wetenschappelijke literatuur, maar ook uit 'grijze' literatuur zoals commerciële websites, whitepapers en weblogs. Dit laatste

komt doordat veel leveranciers en systeemontwikkelaars voornamelijk actief zijn op het gebied van data lineage en veel innovatieve concepten, functies en tools in het domein introduceren.

Naast de literatuurstudie hebben we vier semi-gestructureerde interviews gehouden met deskundigen van verschillende organisaties binnen het Nederlandse rechtsstelsel om inzicht te krijgen in lopende data lineage-gerelateerde (R&D) activiteiten binnen het rechtsstelsel, en om de behoeften en opvattingen te peilen van de deskundigen die betrokken zijn bij data-governance/-management binnen hun organisatie. Verder hebben we twee focusgroepgesprekken georganiseerd met data-stewards en databeheer-experts om onze tussentijdse resultaten te presenteren en vroegtijdig feedback te krijgen. De vier geïnterviewden en de twee focusgroepen werden geselecteerd op basis van de deskundigheid en beschikbaarheid van de deelnemers en niet zozeer vanwege hun representativiteit. Deze keuze werd ingegeven door de voorlopige en verkennende aard van het onderzoek.

Belangrijkste resultaten en bijdragen

In dit verslag geven we een overzicht van verschillende aspecten van data lineage-technologie en hoe dit wordt ingezet in verschillende organisatorische omgevingen. Het verslag kan dienen als kennisbasis bij het ontwerpen en toepassen van data lineage in het Nederlandse rechtsstelsel. Hieronder volgt een samenvatting van de antwoorden op de gestelde onderzoeksvragen.

Wat is data lineage?

Op basis van enkele bestaande definities en de inzichten die tijdens het onderzoek naar data lineage zijn verkregen, definiëren we *data lineage* als volgt: *de beschrijving van databewegingen en -transformaties op verschillende abstractieniveaus langs datatrajecten*. *De beschrijving omvat de aspecten die van belang zijn in een toepassingscontext, zoals hoe (d.w.z. door wie, wanneer, waar, welke, enz.) dataobjecten worden verwerkt (d.w.z. gemaakt, verzameld, opgeslagen, geopend, getransformeerd, verzonden, enz.) en hoe deze gerelateerd zijn aan dataconcepten van hoog niveau.* Deze definitie conceptualiseert niet alleen de fysieke distributie van dataobjecten (zoals de oorsprong, stromen en transformaties van data), maar ook de semantische distributie van de gerelateerde concepten (zoals de business, juridische en organisatorische termen die betrekking hebben op, of van toepassing zijn op deze dataobjecten). Voorts biedt de definitie een middel om het toepassingsgebied van data lineage te beperken tot de aspecten die in elke context van belang zijn.

Data lineage draagt bij aan het vergroten van het vertrouwen in data en in verantwoorde datatransformatie en data-uitwisseling. Data lineage is afhankelijk van het afleiden en beheren van metadata die relevant zijn voor de beoogde gebruiksdoelstellling(en) van data lineage. Data lineage kent verschillende kenmerken, die niet noodzakelijkerwijs onafhankelijk van elkaar zijn. Deze kenmerken omvatten (a) oorsprong van data versus *lineage* van datastromen, (b) het 'waar' versus het 'hoe' van data lineage, (c) typen datatransformatie, (d) grofkorrelige versus fijnkorrelige data lineage, (e) luie versus gretige data lineage, (f) achterwaartse versus voorwaartse data lineage, (g) volgen versus traceren van data lineage, (h) technische versus business data lineage, en (i) horizontale versus verticale data lineage. Deze kenmerken van data lineage markeren de technische ruimte waarbinnen data lineage-oplossingen kunnen worden ontworpen, gekozen en/of ingezet.

Aan welke doelstellingen kan data lineage bijdragen?

Het *lineage*-concept is toegepast op een groot aantal gevallen, variërend van het volgen/traceren van de datastroomberekeningen in één softwareprogramma tot het traceren/traceren van datastromen in verspreide informatiesystemen. Data lineage kan bijdragen aan vele doelstellingen, die elk op hun beurt een rol spelen bij het vergroten van het vertrouwen in data, het delen van data, en datagestuurde applicaties en beleidsvorming. Deze doelstellingen omvatten a) data-governance, b) privacybescherming, c) vertrouwen in AI-modellen, d) data en AI uitlegbaarheid, interpreteerbaarheid en eerlijkheid, e) kwaliteitsbeheer van data, f) beheer van datawijzigingen, g) eigendom van data, h) naleving van regelgeving, controle en verantwoordingsplicht, i) databeveiliging, j) datamodellering en k) datadetectie. Deze doelstellingen beslaan de gebruiksgebieden van data lineage en specificeren als zodanig de *maatschappelijke relevantie van data lineage* in het algemeen.

De uitdaging is om bij de toepassing van een data lineage-oplossing alle genoemde doelstellingen in het vizier te hebben. Een dergelijke alomvattende data lineage-oplossing zou meteen te complex en kostbaar worden en dus niet realiseerbaar, met name in federatieve settings (bijv. in de semi-autonome organisaties van het Nederlandse rechtsstelsel). Als de doelstellingen voor data lineage duidelijk zijn, kan worden bepaald welke kenmerken van data lineage relevant zijn in een operationele setting. Op basis van de vereiste kenmerken van data lineage kan worden bepaald welk (type) metadata moet worden verzameld, en kan er vervolgens data lineage (implementatie)-architectuur worden ontworpen om metadata van data lineage op te slaan, te doorzoeken, te verwerken en op te halen (d.w.z. kan er een beslissing worden genomen over de architectuur voor het beheren van data lineage-metadata).

Hoe kunnen data lineage-tools worden ingezet?

Op basis van onze literatuurstudie en interviews met deskundigen concluderen we dat data lineage binnen het Nederlandse rechtsstelsel vooral kan (of moet) bijdragen aan data-governance, data discovery, kwailteitsbeheer van data, wijzigingsbeheer van data, en privacy en beveiliging. Verder zijn we tot de volgende conclusies gekomen over data lineage in het Nederlandse rechtsstelsel.

- Het is van belang dat gegevens binnen en tussen organisaties in het Nederlandse rechtsstelsel op betrouwbare wijze kunnen worden gedeeld, en dat deelnemende organisaties hun autonomie behouden en over eigen business, conceptuele en logische datamodellen en informatiesystemen beschikken. Data lineage binnen het Nederlandse rechtsstelsel moet rekening houden met diversiteit op alle niveaus, d.w.z. op technisch, logisch, conceptueel en business niveau.
- Binnen het Nederlandse rechtsstelsel worden data niet alleen voor onderzoek en strategische doeleinden gedeeld en verwerkt, maar ook voor operationele doeleinden. De gedeelde data kunnen zich dus op verschillende aggregatieniveaus bevinden, d.w.z. op groeps- en individueel niveau, wat betekent dat data lineage op grof- en fijnkorrelig niveau nodig is.
- De eindgebruikers van data lineage binnen het Nederlandse rechtsstelsel kunnen verschillende achtergronden en verschillende niveaus van datakennis- en -vaardigheden hebben, en bevinden zich mogelijk aan het begin, in het midden of aan het einde van de datapijplijn. Daarom moet er voldoende flexibiliteit zijn om een breed scala van gebruikers te bedienen en moet worden gestreefd naar het bredere bruikbaar maken van data lineage (zogenoemd de democratisering van data lineage).

- We definiëren vijf abstractieniveaus voor data lineage, te weten (a) fysiek, (b) logisch, (c) conceptueel, (d) business en (e) juridisch/ethisch. Het juridische en ethische aspect is bijzonder belangrijk in het Nederlandse rechtsstelsel, aangezien het stelsel verantwoordelijk is voor het toezicht op en de bescherming van de rechtsstaat.
- In organisatieoverschrijdende settings (zoals die van het Nederlandse rechtsstelsel) voorzien we dat horizontale data lineage niet alleen op fysiek niveau kan worden toegepast, maar ook op business en conceptueel niveau. Verder hebben we mogelijk een gecombineerde verticale en horizontale data lineage nodig om een horizontale data lineage op fysiek niveau te realiseren en zo interoperabiliteitsproblemen met betrekking tot data lineage aan te pakken (d.w.z. een verweven horizontale en verticale lineage-configuratie).

Het beheer van metadata in het Nederlandse rechtsstelsel moet schaalbaar en verspreid zijn en moet passen bij de organisatiestructuur van het stelsel. Gezien de organisatieoverschrijdende structuur van het rechtsstelsel, stellen we voor om een federatieve architectuur voor het beheer van data lineage-metadata te overwegen bij de inzet van data lineage. Een federatieve architectuur is gebaseerd op de bestaande organisatiestructuur, die daardoor organisch kan worden uitgebreid, zoals het geval is bij federatieve identiteitsbeheer tussen Europese universiteiten.

Voor het metadatabeheer van data lineage moet het proces voor het verzamelen van metadata worden geautomatiseerd, gezien de hoge snelheid waarmee data tegenwoordig worden verzameld en gedeeld, en het grote volume ervan. De keuze voor het type opslag van data lineage-metadata moet zijn gebaseerd op de vereiste data lineage-kenmerken en de context waarin data lineage wordt toegepast. Het type dataopslag bepaalt de keuze voor een geschikte verwerkingstaal voor zoekopdrachten. De interactie tussen systeem en gebruiker kan op verschillende manieren worden bevorderd, afhankelijk van de technische vaardigheden van de gebruikers.

Businessgerichtheid is een noodzaak, aangezien datagestuurd werken tegenwoordig de gangbare praktijk is geworden. Dit vereist dat niet-technische eindgebruikers bij het ontwerpproces moeten worden betrokken, en dat nieuwe zoekopdrachten automatisch kunnen worden opgenomen in data lineage-systemen. Het is echter niet haalbaar en efficiënt om uitgebreid een enorme hoeveelheid data lineage-metadata te verzamelen en te beheren, en dan af te wachten tot een deel van deze gegevens op een toekomstig moment bruikbaar is voor het beantwoorden van nieuwe business gestuurde zoekopdrachten. Om kostenefficiënter te zijn, kan ervoor worden gekozen een redelijke hoeveelheid data lineage-metadata te verzamelen (d.w.z. het 'grofkorrelig' verzamelen van metadata) en als er een nieuwe opzoekopdracht komt, kan men ofwel een gerichte zoekopdracht doen (d.w.z. een gefocuste zoekactie op need-to-know-basis) of een om-en-nabij antwoord geven als een zekere mate van onzekerheid in de antwoorden aanvaardbaar is. Wellicht is het nodig om op natuurlijke taal gebaseerde gebruikersinterfaces in te voeren die zowel vooraf gedefinieerde als niet vooraf gedefinieerde zoekopdrachten kunnen verwerken. Hiermee maak je data lineage business gestuurd en bruikbaar voor gebruikers met een beperkte technische achtergrond (zoals bedrijfsadviseurs en beleidsmakers).

Wat zijn de mogelijkheden (en beperkingen) van bestaande data lineage-tools?

Het is belangrijk de toepassingsmogelijkheden van data lineage-tools te bepalen in elke gebruikscontext. Ze kunnen namelijk worden gebruikt als criteria voor het

evalueren van de bestaande tools en bij de keuze welke tool het beste past bij de context. Op basis van de resultaten van het onderzoek voorzien we verschillende toepassingsmogelijkheden voor data lineage softwaretools in het Nederlandse rechtsstelsel. De belangrijkste mogelijkheden zijn: flexibiliteit om businessgericht te zijn (reageren op nieuwe en onvoorziene zoekopdrachten met betrekking tot data lineage), meer gedetailleerde lineage dan op attribuutniveau (zoals celniveau), interoperabiliteit met andere tools (van andere organisaties), en een goede gebruikerservaring kunnen bieden.

Als onderdeel van het antwoord op deze onderzoeksvraag schetsen we een kader voor het evalueren van data lineage-tools. Op basis van dit kader moet een reeks evaluatiecriteria worden gedefinieerd door bijvoorbeeld de doelstellingen te verduidelijken die justitiële organisaties nastreven bij het implementeren van data lineage, de gewenste data lineage-kenmerken te verduidelijken, en de evaluatiecriteria dienovereenkomstig te definiëren. Vervolgens kunnen per criterium verschillende niveaus van gedetailleerdheid worden gespecificeerd. Deze niveaus kunnen kwalitatief worden gedefinieerd, d.w.z. dat ze een (betekenisvolle) numerieke waarde of score toegekend krijgen. Ten slotte kan men gekwantificeerde criteria samenvoegen met behulp van numerieke methoden zoals middelen en het vaststellen van drempelwaarden.

Commerciële softwaretools bieden over het algemeen een breed scala aan data lineage-functionaliteiten, in combinatie met andere functionaliteiten (zoals databeheer en datacatalogus). Ze zijn geschikt voor grote organisaties die zich de kosten van dergelijke tools kunnen veroorloven en die gebruik moeten kunnen maken van het brede scala aan functies die deze tools bieden. Open-source- softwaretools bieden over het algemeen een beperkte subset aan data lineage-functionaliteiten, zijn gratis en kunnen worden geïntegreerd met andere (open-source-)applicaties. Daarom zijn ze geschikt voor kleine bedrijven met een gering budget. Zij kunnen deze tools gebruiken of ze aanpassen aan hun behoeften op het gebied van data lineage. Een uitgebreide integratie van open-source-tools vereist interne technische vaardigheden of extra middelen, die mogelijk niet beschikbaar zijn in kleine organisaties. Verder kunnen de open-source-tools nuttig zijn voor het uitvoeren van kleinschalige experimenten binnen de justitiële instelling om praktische ervaring op te doen met data lineage-technologie.

Aanbevelingen voor vervolgonderzoek

Tijdens het uitvoeren van projecten worden diverse toekomstige onderzoekspaden vastgesteld. In deze paragraaf groeperen we ze in drie categorieën, variërend van praktijkgericht onderzoek tot toegepast onderzoek.

Behoefte aan een onderzoek naar bestek van eisen

Bij het kiezen van een (set van) data lineage-tool(s) waarmee kan worden geëxperimenteerd (of die kunnen worden ingezet) in de justitiële instelling, erkennen we de noodzaak om de vereisten te verduidelijken waaraan de tool(s) moeten voldoen. Daarbij horen de volgende actiepunten:

- Bepalen van de typische eindgebruikers en hun data lineage-zoekopdrachten;
- nagaan of het nodig is de zoekopdrachten van de eindgebruikers in de toekomst aan te passen;
- definiëren van de gewenste doelstellingen/kenmerken van data lineage; en

- vaststellen van de evaluatiemethode voor de data lineage-tool door de evaluatiecriteria te bepalen, en hoe deze moeten worden gemeten en samengevoegd.

Als het doel is om data lineage in het Nederlandse rechtsstelsel te implementeren, moet verder onderzoek worden gedaan naar een geschikte structurele en functionele architectuur voor het inzetten van data lineage in het rechtsstelsel.

Ook zou onderzoek moeten worden gedaan naar de voorwaarden en manieren voor het mengen van verticale en horizontale datalijnen op de grenzen tussen organisaties. Daarvoor is het nodig dat de datasemantiek op de grenzen tussen samenwerkende organisaties in kaart wordt gebracht. Verder is de vraag hoe om te gaan met onzekerheden die kunnen worden veroorzaakt door deze inventarisatie van datasemantiek. Hiertoe kan het nuttig zijn om ervaring op te doen met data lineage-tools in de dagelijkse praktijk.

Behoefte aan het bredere bruikbaar maken van data lineage

Er is behoefte aan verder onderzoek naar het businessgericht en bruikbaar maken van data lineage voor gebruikers met een beperkte technische achtergrond (zoals bedrijfsadviseurs en beleidsmakers). Een veelbelovend onderzoeksgebied in dit verband zijn de methoden en tools voor op natuurlijke taal gebaseerde gebruikersinterfaces. Large Language Models (LLM's) kunnen worden ingezet voor het koppelen van teksten in natuurlijke taal aan formele database-zoekopdrachten (bijvoorbeeld SQL). Wellicht is het dan nodig te onderzoeken hoe met onzekerheid kan worden omgegaan bij de afstemming tussen natuurlijke en formele talen.

Behoefte aan effectieve en efficiënte data lineage

Voor een succesvolle implementatie van data lineage-technologie is het cruciaal om de complexiteit van data lineage en de bijbehorende kosten te verminderen.

Voor het verzamelen van data lineage-metadata is het noodzakelijk om geautomatiseerde methoden te ontwikkelen. Zo kan het gebruik van LLM's worden onderzocht voor het (semi-)automatisch genereren van metadata op business niveau (zoals een rapport in natuurlijke taal waarin de technische analyses worden beschreven die worden uitgevoerd op de gegevens voor eindgebruikers op business niveau) op basis van metadata op technisch niveau (bijvoorbeeld van dataquery-scripts). Op deze manier kan de belasting van het genereren van metadata op business niveau, voor technische deskundigen worden verlicht.

In conventionele data lineage wordt aangenomen dat gebruikers begrijpen dat een output wordt gegenereerd door het observeren van de brongegevens en dat zij weten dat de datatransformatie een reeks eenvoudige bewerkingen inhoudt, zoals filteren, samenvoegen en aggregeren. Bij complexe data-analyse, zoals het gebruik van AI/ML-algoritmen, is echter meer informatie over datatransformatie nodig. Een vraag die zich kan voordoen is welke data lineage-informatie moet worden verstrekt om de resultaten van zeer complexe datatransformaties (zoals LLM's) te verklaren en/of te beïnvloeden, en hoe deze data lineage-informatie op een (kostenefficiënte) manier kan worden beheerd.

Een probleem bij het bepalen van de herkomst van data is hoe de herkomst van data in elke setting moet worden bepaald. We vermoeden dat er meerdere opvattingen zijn met verschillende herkomsten van data (met name in organisatieoverschrijdende settings). Als deze hypothese standhoudt, dan vereist het bepalen van de herkomst van data het traceren van gegevensstromen. In toekomstig onderzoek moet worden gekeken naar hoe de herkomst van data (of bestemming van data) in operationele settings kan worden bepaald.

References

Abiodun, O. I., Alawida, M., Omolara, A. E., & Alabdulatif, A. (2022). Data provenance for cloud forensic investigations, security, challenges, solutions and future perspectives: A survey. *Journal of King Saud University-Computer and Information Sciences, 34*(10), 10217-10245.

Alfrink, K., Keller, I., Kortuem, G., & Doorn, N. (2022). Contestable AI by Design: Towards a Framework. *Minds And Machines*, *33*(4), 613–639. https://doi.org/10.1007/s11023-022-09611-z

Androutsopoulos, I., Ritchie, G. D., & Thanisch, P. (1995). Natural language interfaces to databases–an introduction. *Natural language engineering, 1*(1), 29-81.

Atlan, T. (2023, july 19). *What is Metadata Lineage & Why You Should Care About It?* Retrieved November 7, 2024, from https://atlan.com/metadata-lineage/#how-to-evaluate-a-metadata-lineage-tool

Atlan, T. (2024, 26 oktober). *Data Lineage vs Data Provenance: Nah, They Aren't Same!* Retrieved January 28, 2025, from https://atlan.com/data-lineage-vs-data-provenance/

AWS. (n.d.). What Is a Graph Database? - Graph DB Explained. *Amazon Web Services, Inc*. Retrieved November 4, 2024, from https://aws.amazon.com/nosql/graph/

Balm, K., & Bakker, S. (2024). Data lineage platforms a comparative analysis. *Clever Republic*. Retrieved November 7, 2024, from https://www.cleverrepublic.com/blog/data-lineage-platforms-a-comparative-analysis/

Bargh, M. S., Choenni, S., Meijer, R., & Choenni, S. (2022). A method for assessing the degree of openness of semi-open data initiatives: Applied to the justice domain. *International Journal of Electronic Governance, 14*(1-2), 207-235.

Bargh, M.S., Omar, A., & Choenni, S. (2024). Zero-trust security model applied to smart shipping. *Advances in Knowledge-Based Systems, Data Science, and Cybersecurity; Research,* 1(1), 5.

Bertino, E., Ghinita, G., Kantarcioglu, M., Nguyen, D., Park, J., Sandhu, R., Sultana, S., Thuraisingham, B., & Xu, S. (2014). A roadmap for privacy-enhanced secure data provenance. *Journal of Intelligent Information Systems, 43*, 481-501.

Bose, R. (2002, July). A conceptual framework for composing and managing scientific data lineage. In *Proceedings 14th International Conference on Scientific and Statistical Database Management* (pp. 15-19). IEEE.

Bose, R., & Frew, J. (2005). Lineage retrieval for scientific data processing: A survey. *ACM Computing Surveys, 37*(1), 1-28.

Brous, P., Janssen, M., & Krans, R. (2020). Data governance as success factor for data science. In *Lecture notes in computer science* (pp. 431–442). https://doi.org/10.1007/978-3-030-44999-5_36

C2PA (2024). *Coalition for Content Provenance and Authenticity (C2PA) overview.* Retrieved January 7, 2025, from https://c2pa.org

Catarci, T., Mecella, M., Kimani, S., & Santucci, G. (2018). Visual query interfaces. *The Wiley Handbook of Human Computer Interaction, 2,* 561-577.

Choenni, S., Bargh, M. S., Busker, T., & Netten, N. (2022). Data governance in smart cities: Challenges and solution directions. *Journal of Smart Cities and Society*, *1*(1), 31–51. https://doi.org/10.3233/scs-210119

Choenni, S., & Leertouwer, E. (2010). Public safety mashups to support policy makers. In *Electronic Government and the Information Systems Perspective: First*

International Conference, EGOVIS 2010, Bilbao, Spain, August 31–September 2 (pp. 234-248). Springer.
Chyi, N., & Panfil, Y. (2020). *A commons approach to smart city data governance: How Elinor Ostrom can make cities smarter*. New America. Retrieved December 12, 2024, from https://www.newamerica.org/future-land-housing/reports/can-elinor-ostrom-make-cities-smarter/
Cui, Y., & Widom, J. (2003). Lineage tracing for general data warehouse transformations. *The VLDB Journal, 12*(1), 41-58.
Dama International (2017). DAMA-DMBOK: *Data management body of knowledge* (2nd ed.). Technics Publications, LLC.
Damhof, Ronald (2013). *Data management & decision support.* Retrieved August 2, 2024, from https://prudenza.typepad.com/dwh/2013/08/4-quadrant-model-for-data-deployment.html
eduGAIN (2024). *eduGAIN's technical site*. Retrieved November 5, 2024, from https://technical.edugain.org.
EU AI Act (2024). *Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024, laying down harmonised rules on artificial intelligence and amending Regulations* (EC) No 300/2008, (EU) No 167/2013, (EU) No 168/2013, (EU) 2018/858, (EU) 2018/1139 and (EU) 2019/2144 and Directives 2014/90/EU, (EU) 2016/797 and (EU) 2020/1828 (Artificial Intelligence Act)
Everett, D. (2023, June 29). *Data governance vs. data management: What's the difference?* Retrieved October 11, 2024, from https://blogs.informatica.com/2019/08/07/data-governance-vs-data-management-whats-the-difference/
Foote, K. D. (2023, June 13). *Data lineage use cases.* Retrieved October 28, 2024, from https://www.dataversity.net/data-lineage-use-cases/
Freche, J., Heijer, M.D., & Wormuth, B. (2021). Data lineage. *The Digital Journey of Banking and Insurance, 3*(5-19).
Galhardas, H., Florescu, D., Shasha, D. E., Simon, E., & Saita, C. A. (2001). Improving data cleaning quality using a data lineage facility. In *Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW)*, 3.
Gehani, A., & Tariq, D. (2012). SPADE: Support for provenance auditing in distributed environments. In *ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing* (pp. 101-120). Springer.
Gehani, A., Kim, M., & Zhang, J. (2009). Steps toward managing lineage metadata in grid clusters. In *1st workshop on Theory and Practice of Provenance* (pp. 1-9).
GDPR (2016). *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)*. Retrieved November 12, 2024, from https://eur-lex.europa.eu/eli/reg/2016/679/oj
Groth, P.T. (2008). A distributed algorithm for determining the provenance of data. In *2008 IEEE Fourth International Conference on eScienc*, (pp. 166-173). IEEE.
Heaven, W.D. (2021). Hundreds of AI tools have been built to catch covid. None of them helped. *MIT Technology Review*. Retrieved January 9, 2025 from https://www.technologyreview.com/2021/07/30/1030329/machine-learning-ai-failed-covid-hospital-diagnosis-pandemic/
Ikeda, R., & Widom, J. (2009). Data Lineage: A survey. *Stanford University.* Retrieved November 12, 2024, from http://adrem.uantwerpen.be/sites/default/files/lin_final.pdf

Imperva (2024). *Data Lineage*. Retrieved October 16, 2024, from https://www.imperva.com/learn/data-security/data-lineage/#:~:text=Data%20lineage%20is%20the%20process,Data%20lineage%20process

Informatica (2024a). *What is data lineage and why is it important?* Retrieved October 16, 2024, from https://www.informatica.com/nl/resources/articles/what-is-data-lineage.html

Informatica (2024b). *Data marketplace vs. data catalog: Benefits and key differences*. Retrieved December 12, 2024, from https://www.informatica.com/resources/articles/data-marketplace-vs-data-catalog.html

Karimi, A. H., Barthe, G., Schölkopf, B., & Valera, I. (2022). A survey of algorithmic recourse: contrastive explanations and consequential recommendations. *ACM Computing Surveys, 55*(5), 1-29.

Kashliev, A. (2020). Storage and Querying of Large Provenance Graphs Using NoSQL DSE. In *2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC)* and *IEEE Intl Conference on Intelligent Data and Security (IDS)* (pp. 260–262). IEEE.

Karkošková, S., & Novotný, O. (2021). Design and Application on Business Data Lineage as a part of Metadata Management. In *2021 International Conference on Computers and Automation (CompAuto)* (pp. 34-39). IEEE.

Kerhervé, B., & Gerbé, O. (1997). Models for metadata or metamodels for data? In *Proceedings of 2nd IEEE Metadata Conference*, (Silver Spring, Ma, USA).

King, T. (2022). The 8 best open-source data lineage tools to consider. *Solutions Review*. Retrieved November 9, 2024, from https://solutionsreview.com/data-management/the-best-open-source-data-lineage-tools-to-consider/

KnowledgeNile (n.d.) *Top 10 examples of data lineage*. Retrieved October 30, 2024, from https://www.knowledgenile.com/blogs/data-lineage-examples

Lampoltshammer, T.J., Guadamuz, A., Wass, C., & Heistracher, T. (2017). Openlaws.eu: open justice in Europe through open access to legal information. In *Achieving Open Justice through Citizen Participation and Transparency* (pp. 173-190). IGI Global.

Labadie, C., Legner, C., Eurich, M., & Fadler, M. (2020). FAIR enough? Enhancing the usage of enterprise data with data catalogs. In *2020 IEEE 22nd Conference on Business Informatics (CBI)*, *1*, 201-210. IEEE.

Lefebvre, H., Legner, C., & Fadler, M. (2021). Data democratization: toward a deeper understanding. In *Proceedings of the 42nd International Conference on Information Systems (ICIS)*, Austin, USA.

Malik, T., Gehani, A., Tariq, D., & Zaffar, F. (2013). Sketching distributed data provenance. In *Data Provenance and Data Management in eScience* (pp. 85-107).

Matsubara M., Miyamae T., Ito A., & Kamakura K. (2020). Improving reliability of data distribution across categories of business and industries with chain data lineage. *Fujitsu Scientific & Technical Journal, 56*, 52-59.

Memgraph (2023). *Graph database vs relational database*. Retrieved November 4, 2024, from https://memgraph.com/blog/graph-database-vs-relational-database

Mothukuri, V., Cheerla, S.S., Parizi, R.M., Zhang, Q., & Choo, K.K.R. (2021). BlockHDFS: Blockchain-integrated Hadoop distributed file system for secure provenance traceability. In *Blockchain: Research and Applications 2*(4), 100032.

Muniswamy-Reddy, K., Macko, P. & Seltzer, M. (2009). Making a Cloud provenance-aware. *In 1st USENIX Workshop on the Theory and Practice of Provenance.*

Netten, N., Bargh, M.S., van den Braak, S., Choenni, S., & F. Leeuw (2016). On enabling smart government: A legal logistics framework for future criminal justice systems. In *Proceedings of the 17th Annual International Conference on Digital Government Research (dg.o)*, Fudan University, Shanghai, China, June 8-10, pp. 293-302.

Odette International (2024, September 19). *Track & Trace*. Retrieved Oct 18, 2024, from https://www.odette.org/process/track-and-trace#:~:text=The%20difference%20is%20in%20direction,point%20to%20where%20it%20began

Őzcan, F., Quamar, A., Sen, J., Lei, C., & Efthymiou, V. (2020). State of the art and open challenges in natural language interfaces to data. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data* (pp. 2629-2636).

Paré, G., Trudel, M. C., Jaana, M., & Kitsiou, S. (2015). Synthesizing information systems knowledge: A typology of literature reviews. *Information & Management, 52*(2), 183-199.

Pinon, S., Burnay, C., & Linden, I. (2023). Business-driven data recommender system: design and implementation. *Journal of Computer Information Systems*, 1-14.

Qlik (2024). Data Lineage. Retrieved October 16, 2024, from https://www.qlik.com/us/data-management/data-lineage

Qiu T., Chi J., Zhou X., Ning Z., Atiquzzaman M., & Wu, D.O. (2020). Edge computing in industrial internet of things: architecture, advances and challenges. *IEEE Communications Survey Tutorials, 22*, 2462-2488.

Ram, S., & Liu, J. (2007). Understanding the semantics of data provenance to support active conceptual modeling. In *Active Conceptual Modeling of Learning: Next Generation Learning-Base System Development 1,* 17-29. Springer.

Roberts, M., Driggs, D., Thorpe, M., Gilbey, J., Yeung, M., Ursprung, S., ... & Schönlieb, C.B. (2021). Common pitfalls and recommendations for using machine learning to detect and prognosticate for COVID-19 using chest radiographs and CT scans. *Nature Machine Intelligence, 3*, 199–217. https://doi.org/10.1038/s42256-021-00307-0

Roszkiewicz, R. (2010). Enterprise metadata management: How consolidation simplifies control, *Journal of Digital Asset Management, 6*(5), 291–297. https://doi.org/10.1057/dam.2010.32

Sankar, P. (2022). *14 questions to ask when evaluating data lineage.* Atlan. Retrieved November 7, 2024, from https://humansofdata.atlan.com/2022/11/14-questions-evaluating-data-lineage/

Segment (2023). What is data lineage? Complete guide + tools, tips, & examples. *A guide to data lineage best practices and processes.* Retrieved October 16, 2024, from https://segment.com/blog/data-lineage/

Silva, E., Franco, N., Ferro, M., & Fidalgo, R. (2019). Mental workload impact of a visual language on understanding SQL queries. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)* (vol. 30, No. 1, p 239).

Stedman, C., & Loshin, D. (2022). *What is data lineage? Techniques, best practices and tools*. Retrieved October 16, 2024, from https://www.techtarget.com/searchdatamanagement/tip/How-data-lineage-tools-boost-data-governance-policies

Steenbeek, Irina (2023). *Choosing data management IT tools: Data lineage solutions*. Retrieved November 7, 2024, from https://datacrossroads.nl/2023/06/16/choosing-data-management-it-tools-data-lineage-solutions/

Steenbeek, I. (2022). *Data lineage: the needs of and benefits to various stakeholders.* Retrieved August 1, 2024, from https://www.irmconnects.com/data-lineage-the-needs-of-and-benefits-to-various-stakeholders/

Surfconext (2024). *Secure access everywhere with one set of credentials.* Retrieved November 5, 2024, from https://www.surf.nl/en/services/surfconext

Tak, P.J. (2008). *The Dutch criminal justice system.* http://hdl.handle.net/20.500.12832/2945

Tan, Y. S., Ko, R. K., & Holmes, G. (2013). Security and data accountability in distributed systems: A provenance survey. In *2013 IEEE 10th International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing* (pp. 1571-1578). IEEE. Retrieved December 12, 2024, from https://tarjomefa.com/wp-content/uploads/2016/09/5415-English.pdf

Tang, M., Shao, S., Yang, W., Liang, Y., Yu, Y., Saha, B., & Hyun, D. (2019, April). SAC: A system for big data lineage tracking. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)* (pp. 1964-1967). IEEE.

Verma, R., Shrivastava, P., & Merla, N. (2024). Tracing the path: Data lineage and its impact on data governance. In *International Journal of Global Innovations and Solutions (IJGIS)*. Retrieved December 12, 2024, from https://ijgis.pubpub.org/pub/d6k8bzn0/release/1

Vicknair, C., Macias, M., Zhao, Z., Nan, X., Chen, Y., & Wilkins, D. (2010). A comparison of a graph database and a relational database: A data provenance perspective. In *Proceedings of the 48th annual Southeast regional conference* (pp. 1-6).

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., … & Mons, B. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, *3*(1). https://doi.org/10.1038/sdata.2016.18

Woodman, S., Hiden, H., & Watson, P. (2017). Applications of provenance in performance prediction and data storage optimisation. *Future Generation Computer Systems 75*, 299–309.

Wylot, M., Cudré-Mauroux, P., Hauswirth, M., & Groth, P. (2017). Storing, tracking, and querying provenance in linked data. In *IEEE Transactions on Knowledge and Data Engineering 29*(8), 1751–1764.

Xie, Z. (2022). Tracer: A machine learning based data lineage solver with visualized metadata management (Doctoral dissertation, Massachusetts Institute of Technology).

Yamada, M., Kitagawa, H., Amagasa, T., & Matono, A. (2023). Augmented lineage: Traceability of data analysis including complex UDF processing. *The VLDB Journal, 32*(5), 963-983.

Appendix 1 Definitions of data lineage by practitioners

Although there is no universal definition for data lineage, there are many attempts to conceptualize it. A few examples of data lineage from gray literature are given in Table B1.1.

Table B1.1: Data lineage definitions from various websites

Definition	Source
"Data lineage essentially helps to determine the data provenance for your organization. It can provide an ongoing and continuously updated record of where a data asset originates, how it moves through the organization, how it gets transformed, **where it's stored**, **who accesses it** and other critical metadata."	(Informatica, 2024)
"Data lineage is the **process** of understanding, recording, and visualizing data as it flows from data sources to consumption. This includes all transformations the data underwent along the way – **how** the data was transformed, **what** changed, and **why**."	(Imperva, 2024)
"Data lineage is the **process** of recording and tracking the flow of data **throughout its lifecycle**. It enables businesses to visualize and understand where data comes from, how it transforms over time, and **where it's ultimately stored**."	(Segment, 2023)
"Data lineage refers to the **process** of understanding and visualizing data flows from source to current location and tracking any alterations made to the data on its journey. This lets you know where any specific piece of data comes from, **when** and where it separated and merged with other data, and what transformations that have been applied to the field, from initial input to final application."	(Qlik, 2024)
"Data lineage documents the journey that data takes through an organization's IT systems, showing **how it flows** between them and gets transformed for different uses along the way. It **uses metadata** – data about the data – to enable both end users (e.g., data analysts, scientists, etc.) and data management professionals to track the history of data assets and get information about **their business meaning or technical attributes.**"	(Stedman & Loshin, 2022)

Appendix 2 List of the persons involved in the study

Project advisory board

Members of the project advisory committee (in alphabetical order):

- drs. **Koen Balm**, Clever Republic (private company),
- dr.ir. **Sunil Choenni**, Research and Data Centre (in Dutch: "Wetenschappelijk Onderzoek- en Datacentrum", WODC), Dutch Ministry of Justice and Security,
- MSc. **Joost de Haan**, Information Provisioning and Procurement department (in Dutch: "Directie Informatievoorziening en Inkoop", DI&I), Dutch Ministry of Justice and Security, and
- prof.dr.ir. **Maurice van Keulen** (chair of the advisory committee), Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS), University of Twente.

Research advisor

Contributing to project proposal as well as reviewing the manuscript:

- dr. **Debora Moolenaar**, Research and Data Centre (in Dutch: "Wetenschappelijk Onderzoek- en Datacentrum", WODC), Dutch Ministry of Justice and Security.

The WODC (Research and Data Centre), a Dutch agency in the field of Justice and Security, is an independent knowledge institute that falls under the Dutch Ministry of Justice and Security. The WODC contributes to upholding and improving the rule of law by carrying out high-quality scientific research (or commissioning others to do so on its behalf), as well as by presenting solicited and unsolicited knowledge, points for improvement and (where possible) thinking strategies.

More information:

www.wodc.nl