TECHNISCHE UNIVERSITÄT MÜNCHEN
Fakultät für Informatik
Lehrstuhl für Medizinische Informatik

Helmut Toni Spengler

Agile and Privacy Preserving Data Warehousing for Biomedical Research

TECHNISCHE UNIVERSITÄT MÜNCHEN

Fakultät für Informatik

Agile and Privacy Preserving Data Warehousing for Biomedical Research

Diplom-Informatiker Univ.

Helmut Toni Spengler

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender: Prof. Dr. Daniel Rückert

Prüfer der Dissertation:

1. Prof. Dr. Klaus A. Kuhn
2. Prof. Dr. Oliver Kohlbacher
   Universität Tübingen
3. Prof. Dr. Fabian Prasser
   Charité – Universitätsmedizin Berlin

Die Dissertation wurde am 13.04.2021 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 30.07.2021 angenommen.

Für Gaby und Daniel

Abstract

Introduction: Modern, data-driven biomedical research requires integrating a broad range of data types from various sources, across and within institutions. Clinical and translational data warehouses, which store transformed replicates of data from heterogeneous sources in a central database optimized for efficient ad-hoc analyses, are essential building blocks of the IT infrastructures that make this possible. However, provisioning clinical and translational data warehouses entails several challenges: (1) the complexity of these systems' management and ETL (extract-transform-load) processes impede agile provisioning processes; (2) existing formal methods for privacy protection are difficult to combine with the data processing models of modern ETL environments; (3) data collected in clinical care are often incomplete, inaccurate, or miscoded, raising concerns about the level of evidence generated from these data.

Objectives: The work presented in this cumulative dissertation addresses these challenges by (1) providing novel methods for facilitating agile provisioning of clinical and translational data warehouse platforms; (2) integrating formal anonymization and privacy risk evaluation methods into ETL pipelines; (3) offering a flexible and comprehensive architecture for data quality assessment and monitoring.

Methods: To reach these objectives, we (1) developed software container images, bundled with a clear and comprehensive configuration and management façade, allowing administrators to orchestrate multiple warehouse instances of different types efficiently; furthermore, we created an ETL pipeline implementing a declarative configuration paradigm and providing a high automation degree of data preprocessing and cleansing tasks; (2) developed a novel cell suppression algorithm that allows for combining different threat scenarios in one ETL workflow; (3) developed a data quality monitoring architecture which comprises an API for capturing data quality issues, a multidimensional data store and an interface for modern monitoring systems to provide alerting mechanisms and configurable dashboards.

Results: The solutions presented in this thesis include (1) a warehouse management and data loading platform for clinical and translational data warehouses; (2) a plugin for a widely used ETL environment that enables seamless integration of formal anonymization methods directly into the ETL processes; (3) an implementation of the developed data quality monitoring architecture. The methods developed to achieve this have been evaluated analytically and experimentally or are successfully used in large, national research projects, for instance, in the DIFUTURE consortium of the German Medical Informatics Initiative.

Discussion: Experimental evaluations, comparisons with existing solutions, and our experiences in the projects mentioned above have shown that (1) the developed warehouse management and ETL platform can significantly reduce the complexity of the provisioning process and is the only solution fulfilling a wide range of requirements necessary for agile data warehousing provisioning processes in high-security environments like hospital information systems; (2) the developed plugin is the only solution that integrates formal anonymization methods directly into ETL processes while outperforming comparable solutions in terms of scalability and risk-utility trade-offs provided; (3) our monitoring architecture is the only existing approach that meets a wide range of requirements for comprehensive data quality monitoring in clinical and translational data warehouses.

Danksagung

An erster Stelle möchte ich mich herzlich bei Prof. Kuhn für die kontinuierliche Unterstützung sowohl bei meiner Promotion als auch im wissenschaftlichen Alltag bedanken. Das von ihm geschaffene Arbeitsumfeld, viele anregende und erhellende Diskussionen, sowie seine Anmerkungen und kritischen Fragen haben mich nachhaltig geprägt. Ebenso danke ich Prof. Prasser für seine Unterstützung, seine Geduld, für unzählige wertvolle Hinweise, sowie für seine Ermutigung in der Endphase dieser Arbeit, wo ich sie am meisten gebraucht habe. Prof. Kuhn als mein Doktorvater und Prof. Prasser als mein Mentor haben maßgeblich zur Entstehung dieser Arbeit beigetragen.

Ohne die unermüdliche Unterstützung, Liebe und Neugier meiner Familie – meiner Frau Gaby, meines Sohnes Daniel und meiner Schwiegermutter Hedi – wäre all dies jedoch nie möglich gewesen.

Ein ganz besonderer Dank gilt meinen Eltern, die immer an mich geglaubt, mich gefördert und unterstützt haben. Ihr seid stets bei mir.

Mein Dank gilt auch Florian Kohlmayer für seine Kollegialität, Geduld, zahlreiche anregende Diskussionen und sein unerschütterlich positives Wesen.

Nicht zuletzt gilt mein Dank auch meinen Kollegen am IMedIS für zahlreiche inspirierende Diskussionen und ein unvergleichliches Teamwork.

Contents

List of Figures

CHAPTER 1

Introduction

Modern, data-driven biomedical research promises new insights into the development of diseases and to enable preventive, predictive, and personalized medicine (Hood and Friend, 2011). A learning health system, in which data initially collected for facilitating individual patient care can be re-used (in a secured and trusted manner) for generating new knowledge and which (in reverse) accelerates "the progression of knowledge from the laboratory bench to the patient's bedside", plays an essential role in this development (Friedman et al., 2010). By utilizing patient data at comprehensive breadth and depth, modern data analytics methods can help identify unknown correlations between biomedical variables and develop decision support systems to infer diagnoses, recommend treatments, or predict outcomes (Schneeweiss, 2014; Esteva et al., 2019). The acquisition and provision of the large and high-quality datasets needed to realize this vision require comprehensive information integration across and within institutions. Data warehouses, which store transformed replicates of various types of data (e.g., clinical or omics data) from heterogeneous sources in a central database optimized for efficient ad-hoc analyses, have become an essential building block of data integration concepts in data-driven clinical and translational research projects (Canuel et al., 2015). Important representatives of these so-called *clinical and translational data warehouse* platforms include Informatics for Integrating Biology and the Bedside (i2b2) (Murphy et al., 2010) and tranSMART (Scheufele et al., 2014), which is in significant parts based on i2b2. The former is well suited for the representation and analysis of longitudinal clinical data; the latter is designed for processing high-throughput data as well as structured research data and offers comprehensive functionalities for cohort comparison and ad hoc graphical data analysis, supporting numerous omics data types such as gene expression and protein arrays and various genomic variant types.

Background

Providing unified views on information originating from multiple, distributed, autonomous sources entails overcoming heterogeneity on multiple levels, including differences in the source systems' query languages and communication protocols (*technical heterogeneity*); character encodings and binary number formats (*syntactic heterogeneity*); *which* data models (e.g., XML, relational, object-oriented) are used for representing information (*data model heterogeneity*); *how* the elements of these data models are used to represent specific subject matters (*schematic heterogeneity*); and differences in meanings and interpretations of terms, concepts, and values (*semantic heterogeneity*) (Leser and Naumann, 2007).

Approaches to information integration can be distinguished between *virtual* and *materialized* integration. In virtual integration, the data remain distributed in the source systems until a query gets executed, and only the data necessary to answer the query are accessed and integrated on-demand. Important representatives of this type of integration are federated database systems (Sheth and Larson, 1990). In materialized integration, transformed replicates of data from multiple heterogeneous sources get materialized at regular intervals in a central database optimized for complex ad-hoc queries on large data volumes. Heterogeneity is resolved at the time of the database's population, and queries are executed directly on the replicated data, so the source systems are no longer involved at the time of the query. Such systems can be summarized under the term *data warehouses*. They have been used for many years in strategic corporate planning and decision support and are, in recent years, increasingly being used in biomedical research.

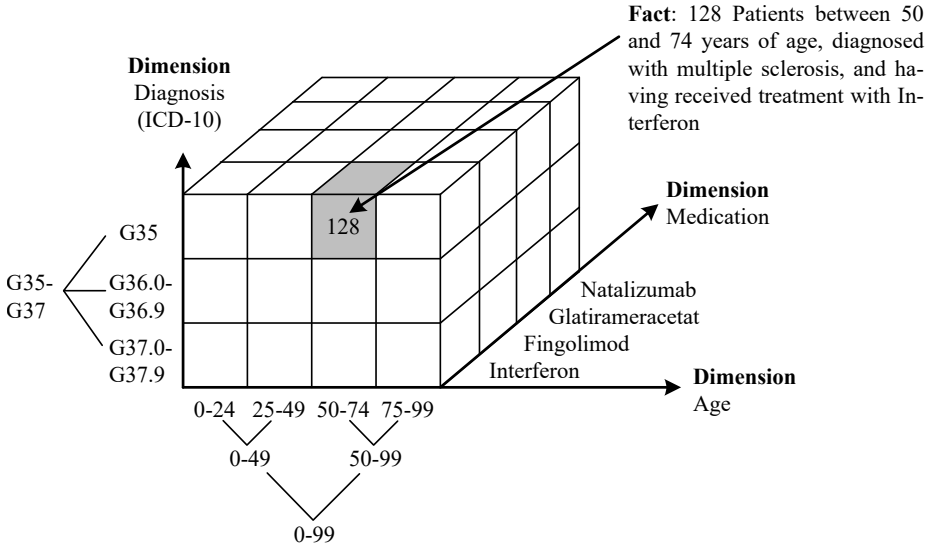


Figure 1.1: Schematic illustration of a data cube.

Data warehouses typically implement multidimensional data models, which represent data as multidimensional data cubes, as illustrated in Figure 1.1. Here, data are categorized either as *facts* or *dimensions*. Facts are associated with numerical *measures*; dimensions characterize and provide the necessary context for the facts. Examples for data that can be organized into dimensions include patients, visits, medication, or diagnoses. Dimensions are also used to select and aggregate data at the required level of detail. To this end, they can be organized into hierarchies composed of specific levels, each representing a certain level of detail for analyses. An example of a hierarchy that can be used for organizing a dimension is the *International Classification of Diseases* (ICD). The combinations of dimension elements define the different cells of a cube. Each cell that contains an associated numerical value (e.g., a count, average, minimum, or maximum) represents a fact. Facts represent the subject to be analyzed and are often implicitly defined by combinations of dimension elements. An example of a fact (illustrated in Figure 1.1) is *the number of patients between 50 and 74 years of age that have been diagnosed with multiple sclerosis and received treatment with Interferon.*

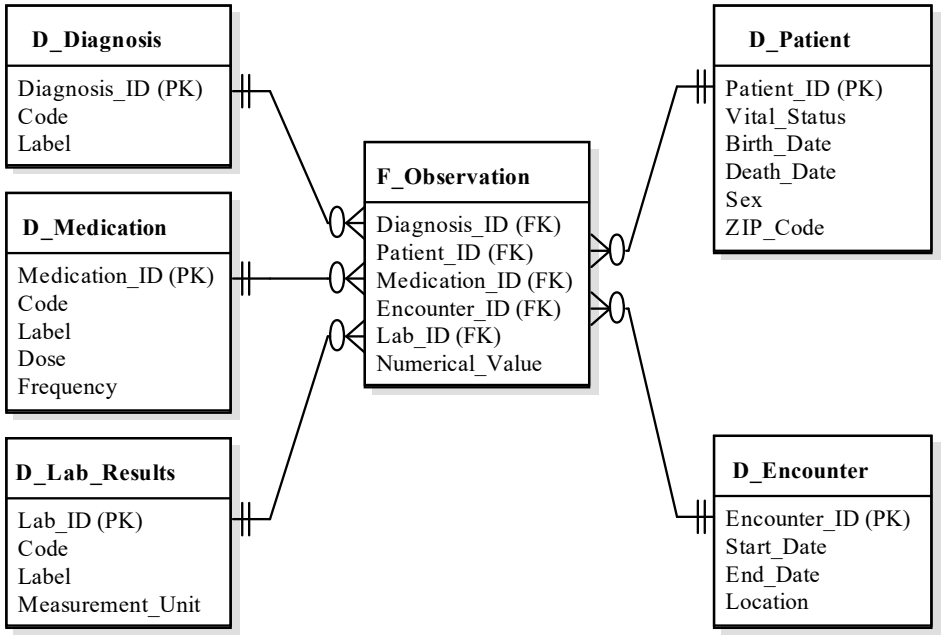


Figure 1.2: Simplified star schema.

Multidimensional database models can be implemented as a (relational) *star schema*, in which data are stored in one fact table and several dimension tables. Figure 1.2 illustrates a simplified star schema. Each row in the fact table represents a fact in the cube. The fact table contains a column for each measure, holding the value for the corresponding fact. Furthermore, each dimension is represented by a column that

contains a foreign key to the corresponding dimension element in the related dimension table (Pedersen and Jensen, 2001). Thus, the patient counts illustrated in Figure 1.1 can be obtained with a simple count query on the fact table. Note that this simplified example does not cover the representation of hierarchies mentioned before. There are various ways for expressing dimensional hierarchies in a relational schema. A representation used in several biomedical data warehouse platforms is to code each dimension element's full path within the hierarchy into its primary key. For instance, the dimension element representing 'Multiple Sclerosis' in an ICD hierarchy would have the primary key '/Diagnoses/Diseases of the nervous system (G00-G99)/Demyelinating diseases of the central nervous system (G35-G37)/Multiple sclerosis'.

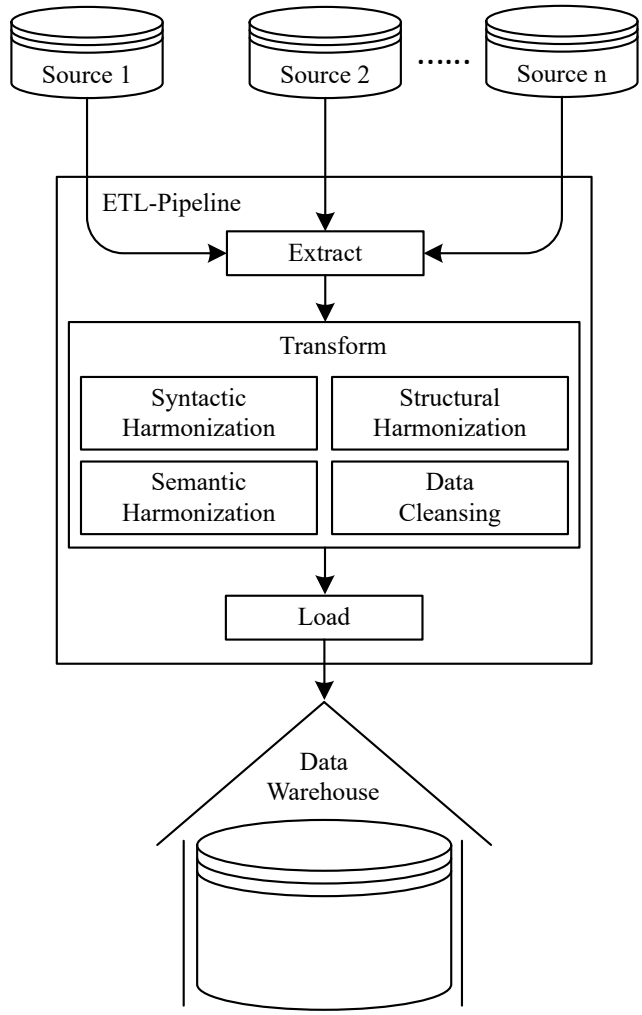


Figure 1.3: Dataflow in a typical data warehouse architecture.

Data warehouses are typically populated through pipelines of procedural operations. These operations are grouped into the three phases *Extract-Transform-Load (ETL)*. In the extract phase, data are replicated from the original data sources into a staging area, which typically involves overcoming technical heterogeneity. In the transform phase,

data are converted into a form optimized for later analysis, which typically involves syntactic harmonization, structural harmonization (i.e., overcoming data model and schematic heterogeneity), semantic harmonization, and data cleansing. In the load phase, the data are transferred into the central database. Figure 1.3 shows a typical data warehouse architecture with the data flowing from the source systems at the top into the data warehouse at the bottom. Because designing, implementing, and managing these complex processes is not trivial, several platforms have been developed for coping with this complexity. Examples include Pentaho Data Integration (Casters et al., 2010) and Talend Open Studio (Bowen, 2012). These platforms typically provide graphical user interfaces, comprehensive libraries of connectors (to source and target systems) and transformation methods for designing and implementing these processes, and run-time environments for initiating and monitoring their execution.

CHAPTER 2

Objectives

Integrating clinical and translational data using data warehouse platforms entails numerous challenges. The work described in this dissertation aims to overcome some of these challenges located in three diverse (albeit intersecting) subject areas: the **agility** of these systems' provisioning process (Killcoyne and Boyle, 2009); **data quality** (Verheij et al., 2018); and **privacy** (Malin et al., 2010). These areas and their intersections are illustrated in Figure 2.1.

For each of these subject areas, this chapter describes the specific challenges, outlines the state of the art, points out remaining gaps, and derives the respective objectives of this thesis.

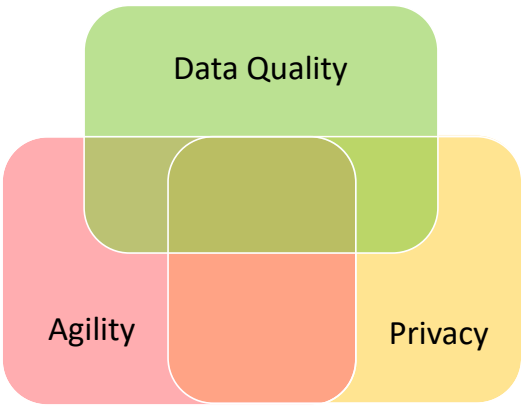


Figure 2.1: Subject areas covered by this thesis.

Agility

Medical information processing takes place in a highly complex system in which dynamic interactions between technology, people in very different roles, and complex organizational structures occur, as new diagnostic and therapeutic procedures and the general conditions of financing in the health care system are constantly chang-

ing (Wurst et al., 2009). In order to facilitate optimal adaptability to these changes, adequate software development methods need to be applied (Lenz and Kuhn, 2004).

Agile software development methods acknowledge Boehm's life cycle cost differentials theory (the cost of change increases during the project lifecycle) (Boehm, 1976) not by *avoiding* change, but by *embracing* it as an inherent aspect of software development and thus reducing the cost of change throughout the project (Highsmith and Cockburn, 2001). This is achieved by applying adaptive planning techniques and practices to identify and incrementally implement requirements in a highly participatory process involving interdisciplinary and self-organizing teams composed of highly qualified software developers and (representatives of) project stakeholders. Agile software development methodologies have gained increased popularity by the publication of the *Manifesto for Agile Software Development* (Beck et al., 2001). It promotes a set of values and principles that constitute the foundation of a variety of software development methodologies already in place at that time, including eXtreme Programming (Beck, 1999) and Scrum (Schwaber and Beedle, 2002). Each methodology can be characterized by a specific combination and emphasis on several practices. These practices include iterative and incremental development (Larman and Basili, 2003), continuous integration (Meyer, 2014), and test-driven development (Janzen and Saiedian, 2005).

For determining whether project management should follow an agile or a plan-driven approach or when to mix agile and plan-driven methods, Boehm and Turner identify several discriminating factors: first, agile methods match well to small products and teams, while plan-driven methods have evolved to handle large products and teams. Second, agile methods have not been tested much on safety-critical products, whereas plan-driven methods have proven to handle highly critical products. Third, simple design and continuous refactoring are ideal practices in highly dynamic environments; the detailed plans and sophisticated preliminary design of plan-driven methods are best used in highly stable environments. Finally, the successful use of agile methods requires the continuous presence of a critical mass of highly qualified experts. Plan-driven methods also need a minimum number of highly qualified team members during project definition but can work with fewer later in the project (Boehm and Turner, 2003b).

Data warehouse development projects in biomedical research exhibit various characteristics that suggest the use of agile software development methods (Boehm and Turner, 2003a; Killcoyne and Boyle, 2009). However, existing environments and tools lack essential features for provisioning and managing these platforms, resulting in substantial manual effort and consequently impeding an agile software development approach. The key challenges here are the complexity of (1) the analytics components' deployment and (2) the development of the ETL process. Simultaneously, both pro-

cesses are characterized by a high degree of repetitiveness in manual activity, which offers enormous optimization potential. This potential could be leveraged by providing a solution consisting of two components:

A data warehouse management platform which

- facilitates the deployment and management of multiple instances of the well-known analytics platforms *Informatics for Integrating Biology & the Bedside (i2b2)* and tranSMART with a compact but comprehensive set of commands and configuration options;
- provides essential security features such as secure default passwords, password management, host-based access control, and transport-layer-encryption by default;
- is built upon authenticatable and verifiable software to support deployments in high-security environments of hospital information systems;

a versatile data loading pipeline which

- is capable of loading data into i2b2 and tranSMART with a single and concise set configuration directives;
- supports processing of data from heterogeneous sources with varying degrees of cleanliness and structure, including data from the research context, complex longitudinal data from the health care context, and highly structured accounting data;
- can perform automatic preprocessing and cleansing tasks, such as automatic detection of encoding schemes, format, and syntax of input data, and can cope with missing and duplicate data.

Data Quality

Large biomedical research networks such as PCORnet (Qualls et al., 2018), OHDSI (Hripcsak et al., 2015), and the German Medical Informatics Initiative (Semler et al., 2018) rely on the re-use of data from electronic health records (EHR) that were collected in the course of clinical care and whose later use was unknown at the time of data entry. The circumstances under which these data were entered into the EHR system are often not clear. These data can be incomplete, miscoded, or inaccurate. Reasons include but are not limited to time pressure, lack of training on the use of the EHR system, lack of IT skills, lack of support (from the EHR system's vendor or within the workplace), or lack of automated system checks during data entry. Therefore, it

is vital to understand the quality characteristics of the data from the various sources before they can be considered "fit for use" (Juran, 1989), integrated, and provisioned to data consumers. "Unless data quality issues are better understood and unless adequate controls are embedded throughout the data lifecycle, data-driven health care will not live up to its expectations" (Verheij et al., 2018).

In order to enable consistency of terms in discussions about data quality, to facilitate systematic approaches to measuring data quality, and to foster the development and sharing of best practices (Weiskopf and Weng, 2013), several conceptual frameworks have been proposed. These include the hierarchical data quality framework by Wang and Strong (1996), which is based on a survey with data consumers. In this framework, 15 data quality *dimensions* (e.g., accuracy, completeness, interpretability, access security) are grouped into the four quality *categories* intrinsic, contextual, representational, and accessibility quality. Weiskopf and Weng (2013) also grouped data quality terms into several categories (completeness, correctness, concordance, plausibility, currency) but based their work on a literature review in which methodologies for assessing data quality of electronic health records are discussed. Similarly, Johnson et al. (2015) based their work on a literature review but used an ontological approach to harmonize and redefine existing data quality concepts using constraints and relationships between concepts. Also, Kahn et al. (2016) proposed a harmonized terminology and conceptual framework based on existing data quality publications. However, the authors also considered operation manuals of mature research networks utilizing EHR data as well as input collected from interviews and workshops with stakeholders of established data quality programs, analytics, and informatics experts. As a conceptual extension to previous frameworks, the data quality categories and subcategories were defined depending on the context of use: *verification* (e.g., with organizational data) and *validation* (e.g., against accepted gold standards). Today, Kahn et al.'s approach is used in the EHR-based research networks mentioned at the beginning of this section (Qualls et al., 2018; Hripcsak et al., 2015; Semler et al., 2018). The latest notable refinement of existing conceptual frameworks is derived from Kahn et al.'s approach: Henley-Smith et al. (2019) introduce an additional axis concerning different contexts of use: regarding the data's initial purpose; regarding the transformations in the context of the ETL process; and regarding the subsequent intentions for the data's secondary uses.

While various implementations exist to assess the quality of data provided by data warehouses, these implementations do not acknowledge the role of the ETL process in the data quality management process or exhibit substantial limitations for practical use. However, comprehensive data quality management for data warehouses requires a software architecture which

- considers the complete data lifecycle, in particular (1) the source data, (2) the ETL process, and (3) the data in the warehouse itself;
- is based on well known and established data quality measures and is highly customizable to enable integration into existing data quality management processes;
- can monitor multiple warehouse instances–which can be based on different platforms–simultaneously;
- supports capturing and analyzing the temporal evolution of data quality measures to be able to follow up on corrective measures and to help discover data anomalies that otherwise would not be visible;
- stores information about reported quality issues in a form and at a level of detail that enables tracing them back to their origin.

Privacy

The increasing amount and complexity of data collected for biomedical research not only holds enormous potential for improving health care outcomes. It also raises concerns about possible violations of patients' and probands' privacy due to the misuse of their data. These concerns are addressed by several national and international laws and regulations, e.g., by the Privacy Rule of the *Health Insurance Portability and Accountability Act* (HIPAA) in the United States (US Department of Health and Human Services, 2002) or–in the European Union–the *Charter of Fundamental Rights* (European Convention, 2012) and the *General Data Protection Regulation* (GDPR) (European Parliament and Council of the European Union, 2016). Under Article 9 of the GDPR, genetic data and data concerning health belong to the data categories subject to exceptional protection.

Following privacy risk types that are recognized as essential for anonymization (Article 29 Data Protection Working Party, 2014): *Singling Out* corresponds to the ability of an adversary to isolate a subset or all records of an individual whose data is represented in a dataset; *Linkability* refers to the possibility to link two or more records concerning an individual or a group of individuals, whereas these records can be stored in the same or different datasets; *Attribute Inference* denotes the ability to infer (with sufficient probability) the value of a specific attribute from the values of one or several other attributes.

In order to address these risks, organizational and contractual measures must be combined with technical solutions for enhancing data security and privacy. An important representative of these technical solutions is data anonymization, which means that data transformation methods are applied to input data to reduce privacy risks

to an acceptable degree (ISO/IEC 20889:2018). Since clinical and translational data warehouse platforms are typically based on relational database management systems, the focus of this dissertation lies on methods suitable for protecting tabular data. When transforming biomedical data, it is essential to retain the input data's truthfulness and plausibility. Therefore, this work focuses on non-perturbative transformation methods. These methods include attribute generalization, which replaces attribute values of the input data with less specific but semantically consistent values and which typically involves the definition of so-called generalization hierarchies. Further truthful transformation methods include attribute-suppression, in which all values of a particular variable are suppressed; cell-suppression, which refers to the suppression of specific data points; character-suppression, which typically replaces parts of a data point with predefined characters; and record-suppression, which denotes the suppression of complete records of data. Non-truthful transformation methods include noise addition, random shuffling, microaggregation, and random data generation. Transformation methods can be used on their own or in combination to achieve a better balance between analytic data utility and residual privacy risks, for instance, when combining random sampling (a special case of record-suppression) with generalization (Bild et al., 2018).

Transforming data using the techniques mentioned above inevitably reduces analytic data utility. Therefore, a significant challenge in data anonymization is finding an optimal balance between data privacy and data utility. Various models exist for quantifying these antagonizing demands. Hence, finding this balance can be considered an optimization problem, in which data utility is to be maximized while predefined risk thresholds have to be met. Privacy **risk measures** help to quantify the susceptibility of datasets to different privacy threats. A well-known measure is k-anonymity, which calculates the maximum size of sets of indistinguishable records concerning specific attributes (so-called quasi-identifiers) and helps to estimate the probability that the identity of an individual represented in the dataset can be inferred by linking it with external data sources (Samarati and Sweeney, 1998). Another well-known approach to quantifying privacy risks is differential privacy. Differentially private anonymization algorithms guarantee that their output changes only to a negligible degree when information about an individual is added or removed (Dwork, 2006). Data **utility measures** typically either calculate statistical properties of the output data to estimate their amount of information (e.g., the average size of sets of indistinguishable records (LeFevre et al., 2006)), or capture differences between statistical properties of the input and output data (e.g., by determining the relative sizes of attribute domains (Iyengar, 2002)).

Integrating privacy protection methods into ETL processes is cumbersome, as existing anonymization tools require the anonymization process to be conducted within their own specific working environments, and existing ETL tools do not support formal anonymization methods. To facilitate the integration of privacy preserving methods into ETL processes, a solution is required which

- uses formal anonymization methodologies that retain truthfulness of the input data, require minimal configuration, are widely accepted in the expert community and by data protection officers;
- supports the processing of very large datasets and data streams;
- enables to combine different threat scenarios in one ETL pipeline;
- directly integrates these methods into common development and execution environments for ETL processes.

Furthermore, although there is a considerable body of research concerning the implications of anonymization on the usability of output data when protecting biomedical data against singling out and linkage, guidance regarding the use of methods for protecting data against attribute disclosure is missing. For being able to use existing methods for protecting biomedical data against attribute inference, it needs to be investigated,

- when it is feasible at all to apply these methods;
- how to select appropriate methods;
- how to parameterize them.

CHAPTER 3

Methods and Results

The work presented in this cumulative thesis addresses fundamental challenges in the three described subject areas by **(1)** presenting a versatile infrastructure that enables the provisioning of biomedical data warehouses in an agile manner; **(2)** developing a flexible architecture that that facilitates comprehensive data quality monitoring for data warehouses; **(3)** examining techniques for protecting biomedical data against attribute disclosure and developing novel methods for integrating such techniques directly into ETL processes.

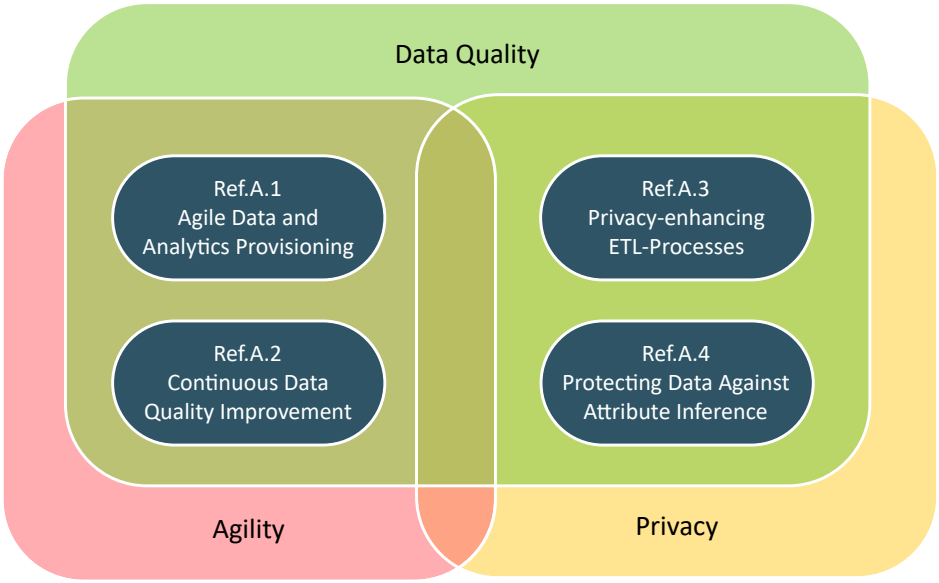


Figure 3.1: Assignment of this thesis' contributions to the subject areas.

In all three areas, innovations were achieved through the concepts and methods presented in these contributions. They have been published as full papers in international, peer-reviewed journals and conference proceedings and are referred to as **Ref.A.1** through **Ref.A.4**. They describe conceptual solution approaches and present methods

and implementations which have been evaluated analytically and experimentally or are in productive use in various large, national research projects. Figure 3.1 shows how these contributions are assigned to the three subject areas agility, data quality, and privacy. This chapter summarizes each of these publications–focussing on the methods used and the results achieved–and lists this thesis's author's individual contributions.

3.1 Agile Data Warehousing

Modern biomedical data warehouse platforms offer support for a broad range of use cases by integrating numerous third-party software solutions based on complex multi-tier architectures. As a result, setting up such systems for professional use requires significant *technical* expertise. Concurrently, significant *medical* expertise is required to make sure that the data are represented in an appropriate structure with adequate semantics. Following agile methods in software engineering, the approach described in **Ref.A.1** aims at bridging the resulting interdisciplinary gaps by enabling to conduct data warehouse provisioning using an agile development process.

To meet this goal, we identified system requirements necessary for achieving agility, focusing on platform management and data loading, and implemented a platform fulfilling these requirements. It consists of a container-based warehouse development and provisioning infrastructure offering a comprehensive set of tools for managing multiple instances i2b2 and tranSMART warehouses and a versatile and easy-to-configure ETL pipeline implementing a declarative configuration paradigm.

The warehouse management infrastructure utilizes the OS-level virtualization platform Docker and supports integrated management of an arbitrary number of instances of i2b2 and tranSMART by providing a compact configuration façade. The solution stack's full source code is publicly available, allowing verification for use in high-security environments. The loading pipeline supports populating both tranSMART and i2b2 instances with a single, concise configuration and can automatically handle heterogeneous data with varying levels of cleanliness and structure.

Based on an experimental evaluation, we could show that our solution is able to process data that cannot be processed with other tools unless significant manual data cleansing and preprocessing is performed. We could also show that the declarative approach enables a reduction of configuration efforts (measured by lines of configuration) by orders of magnitude. A comparison with related tools and environments showed that our warehouse management infrastructure and loading pipeline is the only solution that supports all requirements we have identified for agile data warehouse provisioning.

These results are consistent with our experiences in several data warehousing projects, where the platform helped to significantly reduce the efforts needed for administrating warehouse instances and loading data; for instance, in two large biomedical research projects that use a broad spectrum of different types of data (Prasser et al., 2018; Haller, 2020).

Individual Contributions of Thesis Author: Problem definition; literature survey; analysis, design, prototyping, and implementation; composition, revision, and editing of the manuscript.

3.2 Data Quality Monitoring

It is well-known that agile software development processes rely on early detection and correction of (software) quality issues. In data warehousing projects, the need for tight quality monitoring also applies to the *data* provisioned. In biomedical research, this need is emphasized by the fact that, particularly when re-using health-care data, controlled data collection procedures and clear data definitions are often lacking, making it even more challenging to ensure the level of data quality required for research.

The work described in **Ref.A.2** addresses this problem by presenting a flexible architecture that helps capture, report, and monitor the results of data quality screenings performed during the execution of ETL processes. The architecture comprises (1) an application programming interface (API) which enables developers of ETL processes to log results of data quality screenings, (2) a multidimensional data store as the foundation for efficient analyses, (3) an audit-service, which calculates quality metrics based on the content of the data store, and (4) a monitoring component which provides configurable dashboards and alerting functionalities. The API and the data store support capturing and analyzing data quality issues occurring at different stages of the data lifecycle. Data about these issues are stored at a level of detail that allows for tracing them back to their origin. This architecture enables monitoring of the temporal evolution of data quality metrics and supports various important biomedical data warehouse platforms. Moreover, it facilitates the integrated monitoring of multiple instances of these warehouses and can be used with many ETL environments. As proof of concept, we provide a turnkey implementation in the form of a self-contained stack of containers with (1) template screen types based on Kahn et al.'s well-known data quality assessment framework (Kahn et al., 2016), (2) a set of pre-configured quality metrics to be calculated by the audit-service, and (3) a pre-configured dashboard for presentation and analysis of these metrics.

Implementations of this architecture are used in three large biomedical research projects for analyzing the data quality of warehouses containing up to 100k patients, 300k visits, and 6m facts (Prasser et al., 2018; Haller, 2020; Kamdje-Wabo et al., 2019). The data are loaded with the ETL pipeline described in **Ref.A.1**, which we have integrated with the quality monitoring architecture. This solution enables a quantification of the degree to which the data has been cleansed. The degree of cleansing matches our expectations, taking into account the data's various origins. The software is publicly available under an open-source license (Spengler, 2020b).

Individual Contributions of Thesis Author: Problem definition; literature survey; analysis, design, and implementation; composition, revision, and editing of the manuscript.

3.3 Privacy-Enhancing ETL-Processes

In **Ref.A.3**, we present an integration of well-known and flexible risk assessment and anonymization methods into ETL processes by providing a plugin for a widely used ETL platform. It is based on a novel cell suppression algorithm that enforces risk thresholds by suppressing individual attribute values and thus preserves the input data's schematic properties. Combined with the use of different, context-dependent ways of interpreting suppressed values, this makes it possible to protect data against multiple threats by assembling different instances of the plugin. Furthermore, this method reduces the required configuration to the definition of the risk thresholds and the declaration of quasi-identifiers. Specifically, no generalization hierarchies need to be defined. Moreover, the algorithm retains the plausibility and correctness of the data, which is particularly important in biomedical research. The definable risk thresholds are derived from El Emam's widely used methodology which makes use of the *prosecutor*, *journalist*, and *marketer* attack models. In order to support the processing of very large datasets, the plugin leverages the *row-oriented* processing pipeline of the underlying ETL platform. However, this row-oriented processing paradigm conflicts with the fact that the methods we use for risk assessment and anonymization require a holistic view of the dataset. Therefore, we implemented a technique termed row-blocking, which means that incoming sets (or *blocks*) of records of a defined size are materialized and then anonymized as a whole before the processed block is fed back into the processing pipeline. Our work shows that it is possible to integrate expert-level anonymization methods into ETL workflows for biomedical data warehouses. By combining different anonymization operations in one ETL process, it is possible to protect data from several threats simultaneously. An experimental evaluation shows that our implementation can process very large datasets and that it outperforms existing, generalization-based approaches in terms of scalability and information loss. Furthermore, the software enables to conduct quantitative assessments of re-identification risks and thus document privacy threats, which is an essential aspect regarding compliance with current privacy laws, e.g., the European General Data Protection Regulation (European Parliament and Council of the European Union, 2016; Prasser et al., 2019). The software is publicly available under a non-restrictive open-source license (Spengler and Prasser, 2020).

Individual Contributions of Thesis Author: Problem definition; literature survey; analysis, design, prototyping, and implementation; formal proof of algorithm's correctness; composition, revision, and editing of the manuscript.

3.4 Protecting Against Attribute Inference

Anonymization methods reducing re-identification-risks are widely used but are often not sufficient to protect against the inference of sensitive information from published data (also termed *attribute inference*). Although various privacy models exist that can be used to quantify the risk of a successful inference attack and transform the data so that risks are below a defined threshold, the relevance of such models has been doubted. It has been argued that applying these models involves substantial data transformations to such an extent that the data's usability is severely limited.

In **Ref.A.4**, we studied the ability of different well-known privacy models to help find an optimal trade-off between utility and privacy risks and identify essential factors in this process. To this end, we conducted an experimental evaluation in which we used these models to anonymize different real-world datasets with comparable schemata but different statistical properties using different parameterizations and measured data utility and residual privacy risks of the output data. We normalized the results with respect to two very simple protection methods: (1) to only protect the datasets against re-identification, but *not* against attribute inference, and (2) to completely remove all values of the sensitive attribute. Based on these normalized data, we used risk-utility curves to visualize the effects of different parameterizations on risk and utility when using different privacy models for protecting the datasets. By analyzing these curves, it was possible to estimate and compare how well a particular privacy model is suited for protecting a dataset with specific statistical properties.

As expected, for datasets with high skewness in the distribution of the sensitive attribute (measured by the index of dispersion), it was challenging to find a reasonable balance between risk and utility. However, for datasets with low skewness, it could be shown that–in contrast to popular opinion–it is feasible to improve the balance between risk and utility using truthful and well-known anonymization methods. Interestingly enough, the most intuitive and simple model, distinct ℓ-diversity, often yielded better trade-offs than sophisticated models like β-likeness, even when protecting highly skewed data, for which the latter has specifically been developed.

However, when selecting the appropriate protection method, the dataset's specific properties and the circumstances of the data's use must be considered, which requires in-depth analysis. The approach described in **Ref.A.4** uses the dispersion index of the sensitive attribute and normalized risk-utility curves for selecting appropriate methods and parameterizations. It can be used as a blueprint for further analyses. The corresponding source code is available online (Spengler, 2020a).

Individual Contributions of Thesis Author: Literature survey; analysis and implementation; experimental evaluation; manuscript composition, revision, and editing.

CHAPTER 4

Discussion

4.1 Assessment and Prior Work

Agility

A variety of implementations exist for streamlining the warehouse deployment and management process, most notably, the i2b2 Wizard of the Integrated Data Repository Toolkit (IDRT), a shell-script based tool with a simple graphical user interface for installing and administrating a single i2b2 warehouse instance (Bauer et al., 2016) and i2b2 Quickstart, a shell-script based command-line tool for installing i2b2, which has been published by core members of the i2b2 development community (Wagholikar et al., 2018). Furthermore, several data loading tools exist to simplify the population of the complex database schemata of i2b2 and tranSMART. These include tranSMART-ETL, which is shipped together with tranSMART and utilizes the Pentaho Data Integration platform (i2b2 tranSMART Foundation, 2018), tMData-loader, which uses stored procedures (Clarivate Analytics Life Sciences, 2020), and transmart-batch, which leverages the Spring Batch framework (i2b2 tranSMART Foundation, 2016).

Ref.A.1 contains structured comparisons of these implementations regarding their suitability for facilitating agile warehouse provisioning processes. The comparisons' results can be summarized as follows: all existing **warehouse management infrastructures** either support i2b2 or tranSMART, not both. None of them supports managing multiple warehouse instances of the same type. None of them provides transport layer encryption, integrated password management, and secure default passwords in their standard setups. The existing solutions based on OS-level virtualization cannot be used within high-security perimeters of hospital information systems because their authenticity cannot be verified. Most of the existing implementations are actively maintained. IDRT's codebase has not been updated for over three years. Regard-

ing **data loading pipelines**, only transmart-batch can load data both into i2b2 and tranSMART. However, even for transmart-batch, the target platforms' formats differ significantly, resulting in a duplicated configuration effort. None of the existing tools can handle different data types, encodings, and syntaxes in the same ETL process, resulting in significant data preprocessing required before the data can be processed by these tools. Only one existing tool can handle missing or invalid data. IDRT and tranSMART-ETL are the only tools that can handle duplicate data. IDRT does not provide EAV support at all, while all other existing tools do not support more than one attribute column. Most of the existing implementations are actively maintained.

In summary, both comparisons have shown that none of the existing approaches fulfill all requirements necessary for supporting agile warehouse deployment and management processes.

Data Quality

The most notable practical implementations for assessing data quality for biomedical data warehouses include Achilles Heel (Schuemie et al., 2020) and Data Quality Dashboard (Blacketer et al., 2020). Both employ Kahn et al.'s conceptual framework (Kahn et al., 2016) and are utilized in the Observational Health Data Sciences and Informatics (OHDSI) Initiative (Hripcsak et al., 2015). The Sentinel Initiative (Ball et al., 2016) follows a holistic approach with *Sentinel Operations Centers* supporting technical as well as processual aspects of data quality management in the context of comprehensive *Data Quality Review and Characterization Process and Programs* (Maro, 2019).

Ref.A.2 contains a structured comparison of existing approaches to measuring or monitoring data quality in clinical and translational data warehouses, which can be summarized as follows: the only approaches that consider the complete data lifecycle (in particular the ETL processes) either require membership in specific research networks (i.e., Sentinel), require the data to be transformed to a specific data model (i.e., OMOP Common Data Model or Sentinel Common Data Model), or do not provide a publicly available solution; furthermore, only two of the existing approaches allow to capture, analyze and monitor the evolution of data quality measures over time. However, also their use is tied to the application of a specific data model or even membership in a specific research network; furthermore, none of the research network's independent solutions is known to support simultaneous monitoring of multiple warehouses.

In summary, none of the existing approaches fulfills all criteria necessary for flexible and comprehensive data quality monitoring of clinical and translational data warehouses.

Privacy

Privacy-enhancing ETL Processes

Existing work in the area of privacy-enhancing ETL processes concerns data masking, synthetic data generation, and anonymization. Like anonymization, data masking aims at reducing privacy risks by applying transformations to input data. While the transformations used are similar, data masking methods do not consider the analytic utility of the output data.

Simple masking operations, including different forms of suppression and random data generation, are available as standard transformations in common platforms for developing and executing ETL processes as well as in several standalone solutions. The latter provide slightly more complex masking and data generation operations. More sophisticated masking operations and methods for synthetic data generation are implemented by sdcMicro (Templ et al., 2015) and are, for instance, able to preserve specific statistical properties of the original data. However, this tool is very hard to integrate into typical ETL processing pipelines, as it is designed to be used from within the *R* statistical computing environment. The only solution known to integrate formal anonymization methods directly into ETL processes is a commercial software called Eclipse Risk (Privacy Analytics, Inc., 2020). However, little is known about the methodologies used, as publicly available information about this solution is scarce.

Regarding the privacy risk types singling out, linkability, and attribute inference, most research focuses on protection against singling out and linkability. Research considering protection against attribute inference mostly presents single methods for measuring related disclosure risks (e.g., ℓ-diversity (Machanavajjhala et al., 2007) and t-closeness (Li et al., 2007)). However, except for the work authored by Brickell and Shmatikov (2008), a comprehensive overview of available methods is missing. The authors compared anonymization based on several privacy measures with trivial anonymization methods regarding output data utility in this work. They argue that applying these privacy measures provides no advantage over trivial anonymization like attribute suppression without severely compromising privacy. However, their work has several weaknesses: they measure the relative data utility using statistical classification by merely comparing the k-fold cross-validation results on the input data with k-fold cross-validation on the anonymized data. To get a more realistic estimation of relative classification performance, the model evaluation should rather be performed on (transformed) input data, not on the already anonymized data (Inan et al., 2009); furthermore, they consider statistical classification only for evaluation of data utility and not as a potential tool of an attacker; finally, Cao and Karras (2012)

have shown that applying the privacy model recommended in this work (δ-disclosure privacy) overprotects the data, as it imposes a constraint on *negative* information gain.

4.2 Future Work

Agility

While the infrastructure for enabling agile data warehouse provisioning processes presented in **Ref.A.1** resulted in a significant decrease in efforts required for providing initial and subsequent releases of warehouse instances, the next step will be the development of a continuous integration and deployment (CI/CD) infrastructure to shorten the release intervals further. Moreover, the support for different types of warehouses is currently limited to i2b2 and tranSMART. Supporting cBioPortal and the analytics tools provided by OHDSI would further increase the range of use cases covered by our infrastructure. As these tools are already available as Docker containers, their integration will not be too complex.

However, due to the high degree of normalization of the underlying database models, the ETL pipeline's extension in this direction will require significantly more work. Here, in the nearer future, we plan to provide support for incremental data loading, the possibility to load additional high-dimensional data types, and to implement interfaces to ontology and terminology services. Finally, for deployments with large numbers of users, support for single sign-on technologies, such as OAuth2 (IETF, 2020), is planned to be implemented.

Data Quality

The methods for data quality monitoring presented in **Ref.A.2** were developed and refined in the context of several successful data warehouse projects, where they have already proven to be effective (e.g., (Kuhn et al., 2017; Prasser et al., 2018; German Biobank Alliance (GBA), 2020)). However, to show the generalizability of these methods, a detailed evaluation has yet to be performed. To this end, feedback loops with the data providers of ongoing warehousing projects are currently being established. Furthermore, as will be outlined below, there is significant potential in integrating the monitoring architecture with the contributions presented in **Ref.A.3**.

Privacy

The methods and implementations for providing privacy-enhancing ETL processes presented in **Ref.A.3** are particularly well suited for the protection of data that is rather infrequently collected or relatively stable over time (Malin et al., 2010, 2011). For protecting frequently changing or longitudinal data, specific measures have to be imple-

mented (Terrovitis et al., 2008). Furthermore, we plan to implement additional transformation techniques like data swapping, data generalization, and micro-aggregation into our plugin. Moreover, the methods presented in **Ref.A.3** could easily be integrated into the ETL pipeline presented in **Ref.A.1**. Based on this integration, the monitoring architecture presented in **Ref.A.2** can also be used to report and monitor the results of privacy risk and data utility analyses. Finally, the integration of further risk models like differential privacy or the privacy measures for quantifying risks of attribute inference presented in **Ref.A.4** into the ETL pipeline represents interesting avenues for future research.

While we could show the feasibility of the methods for protecting biomedical data against attribute inference in **Ref.A.4**, using them is still cumbersome and requires significant technical expertise. To increase usability and support the described processes more directly, we plan to integrate these methods into the data anonymization tool ARX (Prasser and Kohlmayer, 2015), for instance, through its graphical user interface (Spengler and Prasser, 2019).

CHAPTER 5

References

Article 29 Data Protection Working Party. Opinion 05/2014 on Anonymisation Techniques. `https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2014/wp216_en.pdf`, 2014. [Online; accessed 29-Aug-2020].

Robert Ball, Melissa Robb, Steven A Anderson, and Gerald Dal Pan. The FDA's sentinel initiative–a comprehensive approach to medical product surveillance. *Clinical Pharmacology & Therapeutics*, 99(3):265–268, 2016.

Cristian Bauer, Thomas Ganslandt, Benjamin Baum, J Christoph, I Engel, Matthias Löbe, Sebastian Mate, S Stäubert, J Drepper, Hans-Ulrich Prokosch, et al. Integrated Data Repository Toolkit (IDRT). *Methods of Information in Medicine*, 55 (02):125–135, 2016.

Kent Beck. Embracing change with extreme programming. *Computer*, 32(10):70–77, 1999.

Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, and Dave Thomas. Manifesto for Agile Software Development. `http://agilemanifesto.org/`, 2001. [Online; accessed 3-Oct-2020].

Raffael Bild, Klaus A Kuhn, and Fabian Prasser. SafePub: A Truthful Data AnonymizationAlgorithm With Strong Privacy Guarantees. *Proceedings on Privacy Enhancing Technologies*, 2018(1):67–87, 2018.

Clair Blacketer, Frank DeFalco, et al. OHDSI/DataQualityDashboard. `https://github.com/OHDSI/DataQualityDashboard`, 2020. [Online; accessed 06-Sep-2020].

Barry Boehm. Software engineering. *IEEE Transactions on Computers*, C-25(12): 1226–1241, 1976.

Barry Boehm and Richard Turner. *Balancing Agility and Discipline: A Guide for the Perplexed*. Addison-Wesley Professional, 2003a.

Barry Boehm and Richard Turner. Using Risk to Balance Agile and Plan-Driven Methods. *Computer*, 36(6):57–66, 2003b.

Jonathan Bowen. *Getting Started with Talend Open Studio for Data Integration*. Packt Publishing Ltd, 2012.

Justin Brickell and Vitaly Shmatikov. The cost of privacy: Destruction of data-mining utility in anonymized data publishing. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 70–78, 2008. doi: 10.1145/1401890.1401904.

Vincent Canuel, Bastien Rance, Paul Avillach, Patrice Degoulet, and Anita Burgun. Translational research platforms integrating clinical and omics data: a review of publicly available solutions. *Briefings in Bioinformatics*, 16(2):280–290, 2015.

Jianneng Cao and Panagiotis Karras. Publishing microdata with a robust privacy guarantee. *Proceedings of the VLDB Endowment*, 5(11):1388–1399, 2012.

Matt Casters, Roland Bouman, and Jos Van Dongen. *Pentaho Kettle solutions: building open source ETL solutions with Pentaho Data Integration*. John Wiley & Sons, 2010.

Clarivate Analytics Life Sciences. tMDataLoader. `https://github.com/Clarivate-LSPS/tMDataLoader`, 2020. [Online; accessed 06-Sep-2020].

Cynthia Dwork. Differential Privacy. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming. ICALP 2006*, volume 4052 of *Lecture Notes in Computer Science*. Springer, 2006.

Andre Esteva, Alexandre Robicquet, Bharath Ramsundar, Volodymyr Kuleshov, Mark DePristo, Katherine Chou, Claire Cui, Greg Corrado, Sebastian Thrun, and Jeff Dean. A guide to deep learning in healthcare. *Nature Medicine*, 25(1):24–29, 2019. doi: 10.1038/s41591-018-0316-z.

European Convention. Charter of Fundamental Rights of the European Union. *Official Journal of the European Union*, 55, 2012. Notice No 2012/C 326/02. `http://data.europa.eu/eli/treaty/char_2012/oj` [Online; accessed 14-Oct-2020].

European Parliament and Council of the European Union. Regulation (EU) 2016/679 of the Eur. 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). *Official Journal of the European Union*, May 2016. L119/59. `http://data.europa.eu/eli/reg/2016/679/oj` [Online; accessed 15-Oct-2020].

Charles P Friedman, Adam K Wong, and David Blumenthal. Achieving a nationwide learning health system. *Science Translational Medicine*, 2(57):57cm29, 2010. doi: 10.1126/scitranslmed.3001456.

German Biobank Alliance (GBA). German Biobank Node (GBN). `http://www.bbmri.de/biobanking/it/infrastruktur/`, 2020. [Online; accessed 05-Sep-2020].

Dirk Haller. CRC 1371 - Microbiome Signatures. `https://www.sfb1371.tum.de/`, 2020. [Online; accessed 12-Jun-2020].

Sandra Henley-Smith, Douglas Boyle, and Kathleen Gray. Improving a secondary use health data warehouse: Proposing a multi-level data quality framework. *eGEMs*, 7 (1):38, 2019.

Jim Highsmith and Alistair Cockburn. Agile software development: The business of innovation. *Computer*, 34(9):120–127, 2001.

Leroy Hood and Stephen H Friend. Predictive, personalized, preventive, participatory (p4) cancer medicine. *Nature Reviews Clinical Oncology*, 8(3):184–187, 2011. doi: 10.1038/nrclinonc.2010.227.

George Hripcsak, Jon D Duke, Nigam H Shah, Christian G Reich, Vojtech Huser, Martijn J Schuemie, Marc A Suchard, Rae Woong Park, Ian Chi Kei Wong, Peter R Rijnbeek, et al. Observational health data sciences and informatics (OHDSI): opportunities for observational researchers. *Studies in Health Technology and Informatics*, 216:574–578, 2015.

i2b2 tranSMART Foundation. tranSMART Batch. `https://github.com/tranSMART-Foundation/transmart-batch`, 2016. [Online; accessed 06-Sep-2020].

i2b2 tranSMART Foundation. transmart-ETL. `https://github.com/transmart/tranSMART-ETL`, 2018. [Online; accessed 06-Sep-2020].

IETF. The OAuth 2.0 Authorization Framework. `https://tools.ietf.org/html/rfc6749`, 2020. [Online; accessed 11-Sep-2020].

Ali Inan, Murat Kantarcioglu, and Elisa Bertino. Using anonymized data for classification. In *Proceedings of the IEEE International Conference on Data Engineering*, pages 429–440, 2009.

ISO/IEC 20889:2018. Privacy enhancing data de-identification terminology and classification of techniques. Standard, International Organization for Standardization, Geneva, CH, 2018.

Vijay S Iyengar. Transforming data to satisfy privacy constraints. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining*, pages 279–288, 2002.

David Janzen and Hossein Saiedian. Test-Driven Development: Concepts, Taxonomy, and Future Direction. *Computer*, 38(9):43–50, 2005.

Steven G Johnson, Stuart Speedie, Gyorgy Simon, Vipin Kumar, and Bonnie L Westra. A data quality ontology for the secondary use of EHR data. In *AMIA Annual Symposium Proceedings*, volume 2015, page 1937, 2015.

Joseph M Juran. *Juran on Leadership for Quality: An Executive Handbook*. The Free Press, 1989.

Michael G Kahn, Tiffany J Callahan, Juliana Barnard, Alan E Bauck, Jeff Brown, Bruce N Davidson, Hossein Estiri, Carsten Goerg, Erin Holve, Steven G Johnson, et al. A harmonized data quality assessment terminology and framework for the secondary use of electronic health record data. *eGEMs*, 4(1):1244, 2016.

Gaetan Kamdje-Wabo, Tobias Gradinger, Matthias Löbe, Robert Lodahl, Susanne Andrea Seuchter, Ulrich Sax, and Thomas Ganslandt. Towards Structured Data Quality Assessment in the German Medical Informatics Initiative: Initial Approach in the MII Demonstrator Study. *Studies in Health Technology and Informatics*, 264: 1508–1509, 2019.

Sarah Killcoyne and John Boyle. Managing chaos: lessons learned developing software in the life sciences. *Computing in Science & Engineering*, 11(6):20–29, 2009.

Klaus A Kuhn, Helmut Spengler, and Rainer Blaser. *Schlussbericht Verbundprojekt e:AtheroSysMed: Systemmedizin der koronaren Herzkrankheit und des Schlaganfalls, Teilvorhaben Klinikum rechts der Isar*. Klinikum rechts der Isar, Technische Universität München, 2017.

Craig Larman and Victor R Basili. Iterative and incremental development: A brief history. *Computer*, 36(6):47–56, 2003.

Kristen LeFevre, David J DeWitt, and Raghu Ramakrishnan. Mondrian multidimensional k-anonymity. In *Proceedings of the International Conference on Data Engineering (ICDE)*, pages 25–25, 2006.

Richard Lenz and Klaus A Kuhn. Towards a continuous evolution and adaptation of information systems in healthcare. *International Journal of Medical Informatics*, 73 (1):75–89, 2004.

Ulf Leser and Felix Naumann. *Informationsintegration*, chapter 9. dpunkt, 1st edition, 2007. ISBN 978-3-89-864400-6.

Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and ℓ-diversity. In *Proceedings of the IEEE International Conference on Data Engineering*, pages 106–115, 2007.

Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. ℓ-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data*, 1(1):3, 2007.

Bradley A Malin, David Karp, and Richard H Scheuermann. Technical and policy approaches to balancing patient privacy and data sharing in clinical and translational research. *Journal of Investigative Medicine*, 58(1):11–18, 2010. doi: 0.2310/JIM.0b013e3181c9b2ea.

Bradley A Malin, Grigorios Loukides, Kathleen Benitez, and Ellen Wright Clayton. Identifiability in biobanks: models, measures, and mitigation strategies. *Human Genetics*, 130(3):383, 2011.

Judith C Maro. Medical Product Safety Surveillance: Data Quality in the Sentinel Initiative, 2019. Presentation at the Canadian Society for Pharmaceutical Sciences, Available from https://www.sentinelinitiative.org/communications/publications/medical-product-safety-surveillance-data-quality-sentinel-initiative. Accessed 9 Mar 2020.

Mathias Meyer. Continuous integration and its tools. *IEEE Software*, 31(3):14–16, 2014.

Shawn N Murphy, Griffin Weber, Michael Mendis, Vivian Gainer, Henry C Chueh, Susanne Churchill, and Isaac Kohane. Serving the enterprise and beyond with informatics for integrating biology and the bedside (i2b2). *Journal of the American Medical Informatics Association*, 17(2):124–130, 2010. doi: 10.1136/jamia.2009.000893.

T. B. Pedersen and C. S. Jensen. Multidimensional database technology. *Computer*, 34(12):40–46, 2001.

Fabian Prasser and Florian Kohlmayer. Putting statistical disclosure control into practice: The ARX data anonymization tool. In *Medical Data Privacy Handbook*, pages 111–148. Springer, 2015.

Fabian Prasser, Oliver Kohlbacher, Ulrich Mansmann, Bernhard Bauer, and Klaus A Kuhn. Data integration for future medicine (DIFUTURE). *Methods of Information in Medicine*, 57(S 01):e57–e65, 2018.

Fabian Prasser, Helmut Spengler, Raffael Bild, Johanna Eicher, and Klaus A Kuhn. Privacy-enhancing ETL-processes for biomedical data. *International Journal of Medical Informatics*, 126:72–81, 2019.

Privacy Analytics, Inc. Privacy Analytics Eclipse. `https://privacy-analytics.com/software/privacy-analytics-eclipse/`, 2020. [Online; accessed 08-Sep-2020].

Laura Goettinger Qualls, Thomas A Phillips, Bradley G Hammill, James Topping, Darcy M Louzao, Jeffrey S Brown, Lesley H Curtis, and Keith Marsolo. Evaluating foundational data quality in the national patient-centered clinical research network (PCORnet®). *eGEMs*, 6(1):3, 2018.

Pierangela Samarati and Latanya Sweeney. Generalizing Data to Provide Anonymity when Disclosing Information (Abstract). In *Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, volume 98, page 188, 1998.

Elisabeth Scheufele, Dina Aronzon, Robert Coopersmith, Michael T McDuffie, Manish Kapoor, Christopher A Uhrich, Jean E Avitabile, Jinlei Liu, Dan Housman, and Matvey B Palchuk. tranSMART: an open source knowledge management and high content data analytics platform. In *Proceedings of the AMIA Joint Summits on Translational Science*, pages 96–101, 2014.

Sebastian Schneeweiss. Learning from big health care data. *The New England Journal of Medicine*, 370(23):2161–2163, 2014. doi: 10.1056/NEJMp1401111.

Martijn Schuemie, Chris Knoll, et al. OHDSI/Achilles. `https://github.com/OHDSI/Achilles`, 2020. [Online; accessed 06-Sep-2020].

Ken Schwaber and Mike Beedle. *Agile software development with Scrum*, volume 1. Prentice Hall Upper Saddle River, 2002.

Sebastian C Semler, Frank Wissing, and Ralf Heyder. German Medical Informatics Initiative. *Methods of Information in Medicine*, 57(S 01):e50–e56, 2018.

Amit P Sheth and James A Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys (CSUR)*, 22(3):183–236, 1990.

Helmut Spengler. Attribute Disclosure Benchmark. `https://github.com/arx-deidentifier/attribute-disclosure-benchmark`, 2020a. [Online; accessed 11-Sep-2020].

Helmut Spengler. Data Quality Monitor. `https://gitlab.com/DIFUTURE/data-quality-monitor`, 2020b. [Online; accessed 11-Sep-2020].

Helmut Spengler and Fabian Prasser. Protecting biomedical data against attribute disclosure. *Studies in Health Technology and Informatics*, 267:207–214, 2019.

Helmut Spengler and Fabian Prasser. Plugins for the Pentaho Data Integration platform. `https://github.com/arx-deidentifier/arx-pdi-plugins`, 2020. [Online; accessed 11-Sep-2020].

Matthias Templ, Alexander Kowarik, and Bernhard Meindl. Statistical disclosure control for microdata using the R-package sdcMicro. *Journal of Statistical Software*, 67(1):1–36, 2015. doi: 10.18637/jss.v067.i04.

Manolis Terrovitis, Nikos Mamoulis, and Panos Kalnis. Privacy-preserving anonymization of set-valued data. *Proceedings of the VLDB Endowment*, 1(1):115–125, 2008.

US Department of Health and Human Services. Standards for privacy of individually identifiable health information, Final Rule. 45 CFR, Parts 160-164. *Federal Register*, 67(157):53182–53273, 2002.

Robert A Verheij, Vasa Curcin, Brendan C Delaney, and Mark M McGilchrist. Possible sources of bias in primary care electronic health record data use and reuse. *Journal of Medical Internet Research*, 20(5):e185, 2018.

Kavishwar B Wagholikar, Michael Mendis, Pralav Dessai, Javier Sanz, Sindy Law, Micheal Gilson, Stephan Sanders, Mahesh Vangala, Douglas S Bell, and Shawn N Murphy. Automating installation of the integrating biology and the bedside (i2b2) platform. *Biomedical Informatics Insights*, 10, 2018.

Richard Y Wang and Diane M Strong. Beyond accuracy: What data quality means to data consumers. *Journal of management information systems*, 12(4):5–33, 1996.

Nicole Gray Weiskopf and Chunhua Weng. Methods and dimensions of electronic health record data quality assessment: enabling reuse for clinical research. *Journal of the American Medical Informatics Association*, 20(1):144–151, 2013.

Sebastian HR Wurst, Gregor Lamla, Fabian Prasser, Alfons Kemper, and Klaus A Kuhn. Einsatz von Dataspaces für die inkrementelle Informationsintegration in der Medizin. In *Informatik 2009 – Im Focus das Leben*, pages 41–41. Gesellschaft für Informatik e. V., 2009.

APPENDIX A

Original Contributions

Contents

A.1 Enabling Agile Clinical and Translational Data Warehousing

Original Paper

Enabling Agile Clinical and Translational Data Warehousing: Platform Development and Evaluation

Helmut Spengler[1], Dipl Inf; Claudia Lang[1], MSc; Tanmaya Mahapatra[1], PhD; Ingrid Gatz[1], MSc; Klaus A Kuhn[1], MD, PhD; Fabian Prasser[2,3], PhD

[1]Institute of Medical Informatics, Statistics and Epidemiology, University Medical Center rechts der Isar, School of Medicine, Technical University of Munich, Munich, Germany

[2]Charité - Universitätsmedizin Berlin, Berlin, Germany

[3]Berlin Institute of Health, Berlin, Germany

Corresponding Author:
Fabian Prasser, PhD
Charité - Universitätsmedizin Berlin
Charitéplatz 1
Berlin
Germany
Phone: 49 30450 ext 528781
Email: fabian.prasser@charite.de

Abstract

Background: Modern data-driven medical research provides new insights into the development and course of diseases and enables novel methods of clinical decision support. Clinical and translational data warehouses, such as Informatics for Integrating Biology and the Bedside (i2b2) and tranSMART, are important infrastructure components that provide users with unified access to the large heterogeneous data sets needed to realize this and support use cases such as cohort selection, hypothesis generation, and ad hoc data analysis.

Objective: Often, different warehousing platforms are needed to support different use cases and different types of data. Moreover, to achieve an optimal data representation within the target systems, specific domain knowledge is needed when designing data-loading processes. Consequently, informaticians need to work closely with clinicians and researchers in short iterations. This is a challenging task as installing and maintaining warehousing platforms can be complex and time consuming. Furthermore, data loading typically requires significant effort in terms of data preprocessing, cleansing, and restructuring. The platform described in this study aims to address these challenges.

Methods: We formulated system requirements to achieve agility in terms of platform management and data loading. The derived system architecture includes a cloud infrastructure with unified management interfaces for multiple warehouse platforms and a data-loading pipeline with a declarative configuration paradigm and meta-loading approach. The latter compiles data and configuration files into forms required by existing loading tools, thereby automating a wide range of data restructuring and cleansing tasks. We demonstrated the fulfillment of the requirements and the originality of our approach by an experimental evaluation and a comparison with previous work.

Results: The platform supports both i2b2 and tranSMART with built-in security. Our experiments showed that the loading pipeline accepts input data that cannot be loaded with existing tools without preprocessing. Moreover, it lowered efforts significantly, reducing the size of configuration files required by factors of up to 22 for tranSMART and 1135 for i2b2. The time required to perform the compilation process was roughly equivalent to the time required for actual data loading. Comparison with other tools showed that our solution was the only tool fulfilling all requirements.

Conclusions: Our platform significantly reduces the efforts required for managing clinical and translational warehouses and for loading data in various formats and structures, such as complex entity-attribute-value structures often found in laboratory data. Moreover, it facilitates the iterative refinement of data representations in the target platforms, as the required configuration files are very compact. The quantitative measurements presented are consistent with our experiences of significantly reduced efforts for building warehousing platforms in close cooperation with medical researchers. Both the cloud-based hosting infrastructure and the data-loading pipeline are available to the community as open source software with comprehensive documentation.

(JMIR Med Inform 2020;8(7):e15918) doi: 10.2196/15918

KEYWORDS

cohort selection; hypothesis generation; data warehouse; translational research; hosting; Docker; extract-transform-load; i2b2; tranSMART

Introduction

Background

Digitalization of health care promises to enable personalized and predictive medicine [1]. On the basis of digital data that characterize patients and probands at comprehensive depth and breadth [2], modern methods of data analytics can be used to detect unknown relationships between biomedical parameters, discover new patterns, and enable decision support systems by using this knowledge to infer or predict parameters, for example, diagnoses or outcomes [3,4]. A *learning health system* [5], which makes health care data available for secondary research purposes, is an important building block of this development. By comprehensive data integration within and across sites, a massive change in clinical and research processes is envisioned, which will accelerate translation and lead to measurable benefits for patients [6]. In this study, we focus on the integration of structured, that is, typically tabular, clinical and research data.

Multiple technical challenges must be addressed to provide the large, high-quality data sets needed for such purposes. Data from distributed and heterogeneous sources must be integrated at the technical, structural, and semantic levels [7]. To this end, a 3-step extraction-transformation-loading (ETL) process is often implemented:

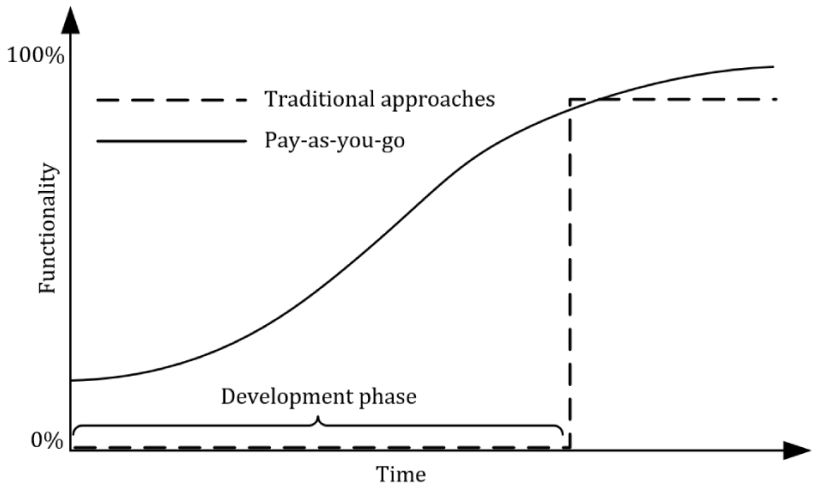
1. Data from research and health care systems are transferred into a staging area in the form of nearly exact copies of data extracted from the source systems [8].
2. Within the staging area, the structure, syntax, and semantics of these data extracts are then normalized into a common data model (CDM) using standard terminologies. These common data representations typically implement a specific database schema, which efficiently and effectively supports complex analytical query processing.
3. Finally, the data are loaded into the target system.

Important examples include clinical and translational data warehousing platforms, such as Informatics for Integrating Biology and the Bedside (i2b2) [9], tranSMART [10], and the Observational Medical Outcomes Partnership (OMOP) CDM [11]; federated and distributed solutions, such as the Shared Health Research Information Network [12]; and the tools provided by Observational Health Data Sciences and Informatics (OHDSI) [11], which can be deployed on top of these analytical databases.

These existing biomedical data analytics platforms offer a wide range of functionalities and integrate different software solutions for data storage, workflow orchestration, and data analysis using multi-tier architectures. As a result of this complexity, considerable technical expertise is required to set them up in a secure manner. These challenges increase even further when organizations run several data-driven research projects and hence need to set up, configure, and maintain multiple warehouse instances. Moreover, ensuring that input data are represented in the analytics platforms in a sound structure with reasonable semantics requires significant medical expertise. It is well known that bridging the interdisciplinary gap between these two worlds requires iterative development processes, in which different solutions are evaluated in short feedback cycles [13]. As we will show later, existing data-loading tools for the aforementioned platforms, however, typically require complex configuration files and input data that adhere to specific formats and structures. Consequently, substantial data restructuring and cleansing is required before data loading can be started and initial feedback can be collected.

In an ideal world, upfront efforts for project-specific technical setup, data cleansing, and data structuring can be avoided, and development starts rapidly, while repeated discussions with clinicians and medical researchers are carried out in parallel [14]. Technical solutions that facilitate this approach have been called *dataspace management systems* [15]. The key idea is to implement a *pay-as-you-go* approach to data integration. A comparison with traditional approaches is presented in Figure 1. It illustrates how the traditional approaches are characterized by an initial development phase in which the data are being integrated on a syntactic, structural, and semantic level, and no service is provided to the users. In contrast, the pay-as-you-go approach provides some initial functionality from the beginning, which is then incrementally extended to better meet the requirements [15,16]. This means that the associated development process can be carried out in an agile manner, involving close cooperation and short feedback cycles with end users. This comes with multiple benefits for the parties involved: clinicians or medical researchers are provided with initial functionalities much more quickly, and feedback can be provided to the development team more often. This is particularly important for data loading because it has been estimated that the development of ETL processes accounts for up to 70% of the total effort required to set up data warehouses [7,17]. For both end users and developers, this can also lead to the reduction of duplicate and redundant work, thus significantly reducing the efforts required. The approach is related to agile methods of software engineering, in which software evolves through continuous collaboration between developers and users. It is well known that this can also help to better bridge the interdisciplinary gaps [18].

Figure 1. Schematic comparison of traditional approaches to data integration and the pay-as-you-go approach.



Objectives

The aim of this study was to implement a platform that enables the deployment and customization of well-known clinical and translational data warehousing solutions in close cooperation with end users in an agile approach. Our solution consists of 2 parts with the following unique features:

1. A cloud-based warehouse management infrastructure, which supports the installation and maintenance of i2b2 and tranSMART in an integrated manner by providing a common set of commands; implements security-by-default features, including transport layer encryption, host-based access control, and password management; and is based on verifiable and authenticatable software to enable installations within high-security perimeters of hospital information technology (IT) environments.
2. A flexible data-loading pipeline, which supports loading data into both i2b2 and tranSMART; is able to process heterogeneous data with different degrees of structure and cleanliness; and performs automated data cleansing and preprocessing, including automatic detection of the syntax and format of input data, and has the ability to handle different encodings as well as missing and duplicate data.

The complete software stack is available to the community as open source software [19,20]. In this study, we provide readers with an overview of the most important system requirements and design decisions. To demonstrate that our solution enables an agile approach to be implemented in a professional context, we present the results of a structured comparison with existing management infrastructures and data-loading pipelines as well as an experimental evaluation of data-loading processes. Our results show that our management infrastructure is the only publicly available open source implementation that supports all the abovementioned features, which is essential for secure deployments in professional IT environments. Moreover, the experimental evaluation showed that no other open source data-loading pipeline was able to process 3 different benchmark data sets, including structured research data, complex longitudinal clinical data, and highly structured billing data, in their raw form. The experiments also showed that our solution is feasible from a computational perspective. We believe that the software presented in this study can be an important tool to support medical informaticians with realizing data warehousing projects and that the methods implemented can provide system developers with novel ideas for the development of future platforms.

Methods

Selection of Target Systems

Clinical and translational data warehouses provide users with efficient analytical access to integrated data sets [21,22]. As an initial step, we decided to utilize an infrastructure supporting i2b2 and tranSMART as both of these have a broad installed base and strong community support. For example, the integrated solution of Hôpital Européen Georges-Pompidou [23] uses i2b2 and tranSMART, integrating data from electronic patient records, including aggregated, anonymized, and *deanonymized* patient data. The tranSMART platform [10] is based on the i2b2 framework, and its suitability for data from clinical studies has already been demonstrated in various projects [24]. In combination, they can be used to support a wide range of use cases.

The i2b2 platform is very well suited for representing longitudinal and often semistructured clinical data, and it supports complex features such as temporal queries against time series data [9]. TranSMART was built over the i2b2 data model to provide improved support for high-dimensional data. The system is well suited for integrating structured research data as well as high-throughput data, and it provides comprehensive support for ad hoc graphical data analysis and cohort comparison [10]. TranSMART offers built-in support for various types of omics data, such as protein and gene expression arrays, single-nucleotide polymorphism data, and certain types of genomic variants. With the recent merger of the i2b2 Foundation and the tranSMART Foundation, a process has been started to

unify both platforms. Until a combined solution becomes available, installations of both systems are needed to support different use cases and to handle different types of data.

The 2 systems offer web-based graphical user interfaces. TranSMART employs a classical three-tier information system architecture, whereas i2b2 consists of an extendable framework consisting of several *cells*. Both platforms can be installed on top of different database management systems. As we focus on open source software, we decided to use PostgreSQL, an open source relational database management system.

Cloud Infrastructure for Managing i2b2 and tranSMART

Rationale and Requirements

Both i2b2 and tranSMART offer a wide range of functionalities, and they are based on a software architecture that integrates components for data storage, workflow orchestration, and data analysis. Consequently, installation, configuration, and maintenance procedures are complex and require solid technical expertise. Concurrently, documentation is often lacking. As an example, the number of tranSMART software dependencies is very large, which regularly leads to some dependencies not being up to date or having become incompatible with the underlying (operating) system infrastructure, requiring manual changes to installation scripts. In contrast, the i2b2 installation process is fairly robust, well documented, and up to date [25]. However, it can be quite challenging to debug configuration errors of i2b2 owing to its highly modular architecture, which involves exchange of complex data via web services. These challenges increase significantly when a larger number of instances need to be set up, configured, and maintained. Furthermore, when deploying such systems in production environments, additional aspects such as transport encryption and password management need to be considered. These and further functionalities are not supported by existing cloud-based deployment solutions for i2b2 and tranSMART, such as the Integrated Data Repository Toolkit (IDRT) [26], i2b2 Quickstart [27], or the prebuilt images available on Docker Hub [28] (see the *Discussion* section for an in-depth comparison).

We, therefore, decided to employ clean virtual containers, ideally together with associated maintenance scripts to quickly boot up, configure, and shut down instances of i2b2 and tranSMART in a uniform manner. The most important requirements were as follows:

1. *Robust installation of a trusted runtime environment:* The solution developed shall streamline the complex installation process of tranSMART and enable rapid instantiation of new instances of tranSMART and i2b2.
2. *Unified installation and maintenance:* The solution shall provide a façade encapsulating important configuration options and make the effective management of multiple instances of i2b2 and tranSMART straightforward by providing easy-to-use common commands for both platforms.
3. *Built-in security:* The solution shall significantly improve the security of i2b2 and tranSMART by enabling transport encryption and host-based access control by default as well as by automatically setting nontrivial passwords.

Technical Design

The cloud infrastructure has been designed to run on a physical or virtual machine with a standard Linux operating system. In this system, Docker needs to be installed as a virtualization platform that enables the provisioning of software in deployment units called containers. Each container encapsulates a complete software stack together with all required dependencies, such as libraries and configuration files. Docker employs OS-level virtualization, which means that in contrast to full virtualization, where each virtual machine contains and runs its own operation system, Docker containers can share one single operating system instance and are thus more lightweight than virtual machines. Although containers are isolated from each other, they can be enabled to communicate through definable channels (eg, Transmission Control Protocol ports). Containers can quickly be instantiated and customized via runtime parameters in this process.

We chose Docker for the following reasons: (1) it enables describing and documenting installation processes in a machine and human-readable format, thus fulfilling our requirement for robust installation and quick instantiation; (2) it allows customizing running containers by means of runtime parameters (eg, access permissions, passwords, and instance names), thus fulfilling our requirement to provide uniform configuration and maintenance scripts for both platforms; (3) its efficient use of resources allows rapid booting up and shutting down instances; and (4) it integrates well with common software development infrastructures, such as GitLab.

As a gateway component to provide transport encryption, host-based access control, and data routing for the particular warehouse instances, we decided to include the Apache HTTP Server into the host environment utilizing its proxy and virtual host modules.

Meeting Requirement 1: Robust Installation of a Trusted Runtime Environment

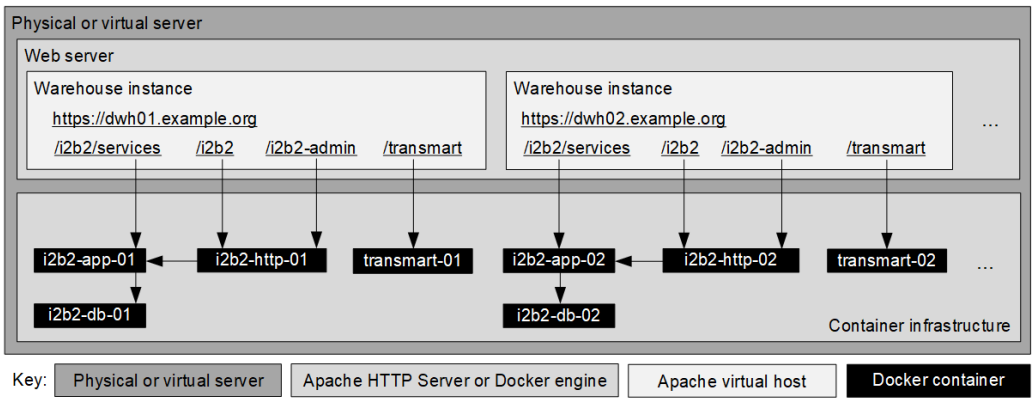
The solution can be used to host an arbitrary number of i2b2 and tranSMART instances. Each host system includes the following containers per instance: (1) a database server for i2b2, (2) an application server for i2b2, (3) a web server for i2b2, and (4) a complete tranSMART software stack. It can be accessed via specific URLs. The subdomain in this URL denotes the warehouse instance, for example, *dwh01* or *dwh02*. Each subdomain is represented by a dedicated Apache virtual host and provides one instance of i2b2 and one instance of tranSMART. As an example, the URL-pattern *[http|https]://dwh02.example.org/i2b2/* denotes the web front-end of i2b2 instance *02*, which is exposed by the Apache virtual host *dwh02.example.org*.

Both tranSMART and i2b2 expose specific ports to provide specific services. These include their web front-ends and various web services. To avoid port clashes when running multiple warehouse instances and their respective containers, the ports used by each container are mapped to corresponding ports on

the host system using specific offsets such that a certain set of ports uniquely identifies each service of each container.

Figure 2 illustrates the components used by the environment and their interactions. The actual instances of i2b2 and tranSMART are implemented as (stacks of) Docker containers (black boxes). Access to these containers is relayed by an Apache web server, which acts as a gateway. Each warehouse instance is represented by a virtual host of the gateway and is identified by the first part of the hostname contained in the URL of the request. Detailed installation instructions along with well-documented configuration files are available on the web [19].

Figure 2. Schematic overview of the components for the provisioning of multiple warehouse instances and their interaction.



Meeting Requirement 2: Unified Installation and Maintenance

To support unified management for instances of both types of systems, we have developed 2 configuration scripts that can be parameterized. Target instances are identified by their type and consecutive numbers (eg, i2b2-04). The first script can be used to set up new warehouse instances and to reset existing instances. It does so by creating configurations for Apache's proxy and virtual host modules and environment files for the Docker compose scripts. If needed, the resulting files can be edited by the administrator (eg, to replace randomly generated passwords) before the new instances are created. The second script can be used for starting, stopping, and deleting warehouse instances as well as associated disk volumes. It has been implemented as a wrapper for Docker compose commands that access the environment variables defined in the associated environment files.

Meeting Requirement 3: Built-In Security

The setup process implements several crucial security measures, including transport layer encryption, server authentication, restricted access paths, and nontrivial default passwords.

Access to the services running on each server is only permitted indirectly via the Apache HTTP Server, which acts as a central gateway. This component takes care of the transport encryption and server authentication mentioned above as well as address-based access control. The only service that can be reached without having to pass the gateway is the database system to enable efficient data loading. Here, access control is implemented at the database level. Permission to access the database has to be granted explicitly, which includes the declaration of address ranges with specific access rights. To simplify the Transport Layer Security configuration, we make use of the *subject alternative name* extension to the X.509 server certificates [29], which our platform uses for authenticating the data warehouses and for transport layer encryption. Embedded plain text secrets and the fact that the source and content of many images cannot be verified have been identified as major risks for system components based on container technologies [30]. This impedes the use of prebuilt images in high-security IT environments. Our infrastructure does not suffer from these shortcomings as we employ Docker Content Trust [31] to verify the authenticity of all base images used. As the current images for i2b2 and tranSMART do not support this authentication mechanism, we decided to build our own images based on authenticated sources (by verifying Pretty Good Privacy signatures of binaries used and/or building them from source). Secure default passwords are automatically created via a random password generator [32] with a default length of 10 characters and injected into the containers at runtime.

Generic and Agile Data-Loading Pipeline for i2b2 and tranSMART

Rationale and Requirements

Populating i2b2 and tranSMART with data is cumbersome and requires significant expertise regarding the underlying database schema and how both systems use it. For this reason, several tools have been developed to simplify this process, including tranSMART-ETL [33], tMDataLoader [34], transmart-batch [35], Integrated Curation Environment (ICE) [36], IDRT [26], transmart-copy [37], and TranSMART data curation toolkit (tmtk) [38]. However, none of these tools fulfill the requirements needed to implement agility (see the *Discussion* section for an in-depth comparison).

First, all available data loaders except transmart-batch are strongly tied to 1 of the 2 target systems. As both are often needed in parallel, this introduces additional preprocessing and configuration efforts. The main reason is that loaders for different systems make different assumptions about the degree of structure and cleanliness of import data. In addition, different loaders use different configuration mechanisms. Moreover, existing tools follow imperative configuration paradigms, where it must be specified how the loading process should be executed, making this process complex and requiring substantial technical expertise as well as domain knowledge. Finally, to support agile and fast loading, tools should be able to automatically handle heterogeneity and errors in input data, such as differences in data encoding and syntax as well as missing and duplicate data. To address these challenges, we needed a data-loading pipeline fulfilling the following requirements:

1. *Platform independence:* The data-loading pipeline shall be designed independent of a specific target system, enabling the loading of data into both i2b2 and tranSMART with the same pipeline using the same configuration files.
2. *Support for different types of data:* The pipeline shall support heterogeneous data with different degrees of structure and cleanliness, such as structured research data, complex longitudinal clinical data, and highly structured billing data, without requiring complex preprocessing or configuration efforts.
3. *Automated data cleansing and preprocessing:* The pipeline shall automatically detect the syntax and format of input data and handle different encodings as well as missing and duplicate data. This significantly reduces efforts and improves agility when providing warehousing solutions.

Technical Design

The most important design decision made to fulfill the requirements listed above was to center the tool around a declarative and model-driven way of configuring the import process. The basic idea was to enable users to match data to an entity-relationship (ER) model that describes the desired target representation of the data. The tool then automatically determines how the input data must be interpreted, transformed, and loaded to reflect this model in the target database. This includes the automatic creation of the ontologies required by i2b2 based on this model. This is in stark contrast to the imperative configuration paradigm found in most ETL tools for i2b2 and tranSMART and significantly reduces the complexity of configuration files and hence efforts (see the *Results* and *Discussion* sections). Moreover, the approach enables our tool to automatically perform a wide range of data transformation and cleansing tasks, thus fulfilling our requirements to support different types of data and automate data cleansing. To fulfill the requirement of platform independence, our tool acts as a *compiler* for configuration files to be used for different ETL tools for i2b2 and tranSMART.

The data-loading tool has been developed in Java using the Spring Batch framework for robust, maintainable, and extensible orchestration of the individual steps of the ETL process; the Univocity parser for reading and writing comma-separated values (CSV) files; and juniversalchardet, a Java port of Mozilla's library, for the automatic detection of file encodings. Access to the target relational database systems has been implemented using Java Database Connectivity.

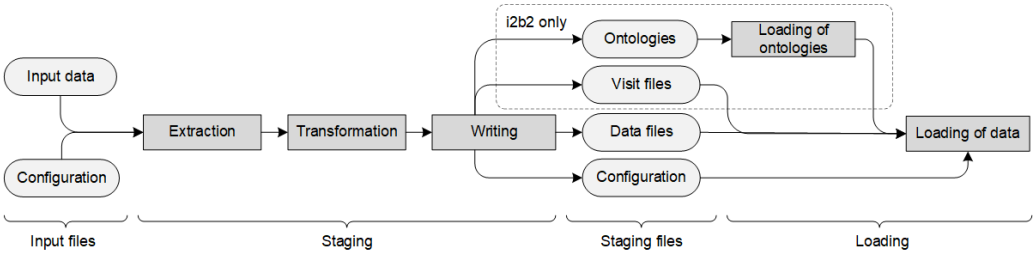
Meeting Requirement 1: Platform Independence

As some powerful loading tools for the different target platforms have already been developed, we decided to implement a meta-loading process consisting of 2 phases: the first is the *staging* phase, in which data are transformed into an intermediate staging representation and configuration files are compiled into the target configuration language for the respective loading tool, which we term *back-end* loader in the context of our meta-loading process. We refer to the transformed data and the configuration files created in this phase as *staging files*. The second is the *loading* phase, in which the staging files are used to execute the respective back-end loader for the chosen target platform.

Figure 3 illustrates a typical staging and loading process. The *staging phase* is divided into 3 subphases: data extraction, data transformation, and data writing. In the data extraction subphase, our tool reads the declarative configuration, which describes the structure of data to be represented in the target system. On the basis of this configuration, it reads and parses the input data. Details are presented in the 2 subsequent sections. In the data transformation subphase, different data cleansing steps are performed, which are also be presented in the 2 subsequent sections. The last subphase involves writing the transformed data into intermediate files, which are consumed by the back-end data loaders in the loading phase. In the case of i2b2, visit data are written separately. This is followed by writing the associated configuration files, describing how the staging data are to be loaded. In the case of i2b2, this (pre-)final step is concluded by writing data describing the underlying ontologies into separate files. In the *loading phase*, the actual data loading is performed by executing the user-defined back-end loaders. If i2b2 has been selected as the target system, this step is preceded by loading the ontology trees into the target system. Currently, our tool supports the following 2 back-ends for data loading:

- *tMDataLoader*, which has been implemented in Groovy and in stored procedures of the underlying database system to automate data loading for tranSMART [34]. The tool relies on a specific directory structure, containing the data sets and configurations, thus following the convention over configuration approach. It supports the full spectrum of features provided by tranSMART, including the annotation of selected values with timestamps.
- *transmart-batch*, which is implemented in Groovy using Spring Batch and which has been designed to support both tranSMART and i2b2. It requires a specific set of files to be provided about subjects and visits as well as further files containing the actual payload data. It supports fewer features of tranSMART than tMDataLoader and requires significant data cleansing to be performed upfront to provide data in the syntax and structure required.

Figure 3. Overview of data staging and loading with the tool developed. i2b2: Informatics for Integrating Biology and the Bedside.



Meeting Requirement 2: Support for Different Types of Data

As mentioned before, the configuration is performed using a *declarative* approach [39]. This means that users do not need to specify how data should be loaded, but instead map an ER model to the data files to describe the relationship between input and output data. Consequently, the tool can perform a wide range of data transformations automatically without prior normalization, including the automatic creation of the target ontology. Although users are less flexible in defining how data should be represented in the target system, a decent representation can typically be achieved for almost all of the data items, as we will show later, with just a fraction of the effort required to use a more versatile loader. If needed, users can still modify and fine-tune the intermediate staging files to achieve an optimal representation.

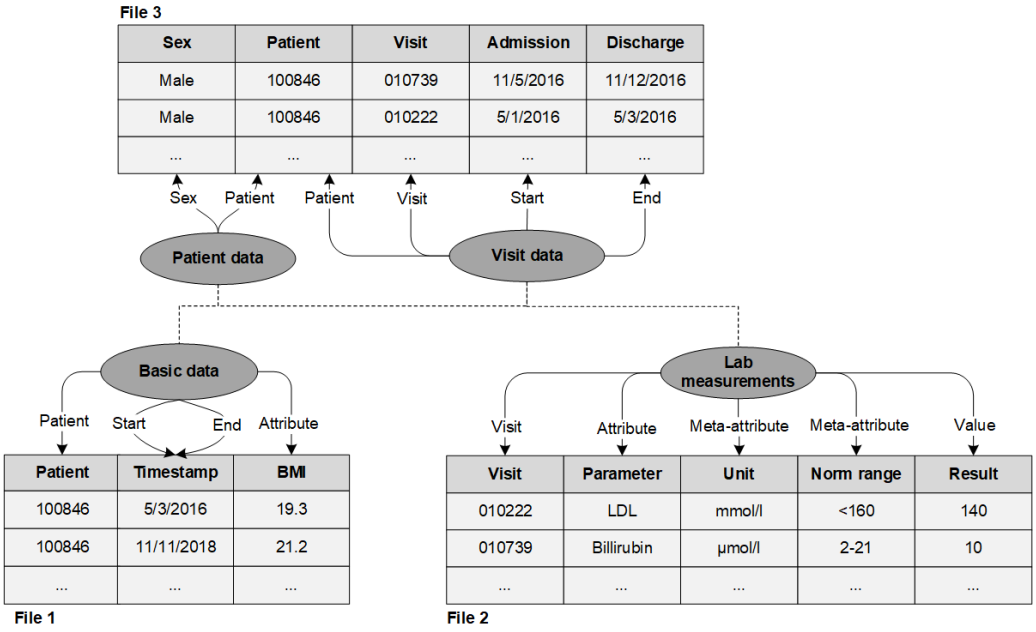
The tool developed was designed in such a way that the maximum degree of the work that needs to be done for successful loading is automated. There are just a few assumptions that are made about input data: (1) data must be tabular, as this is in our experience the most typical format in which clinical and research data can be provided; (2) every line within a file must contain data for a specific patient, visit, or encounter; (3) patients, visits, or encounters must be identified by (composite) keys or timestamps; (4) one file must contain information about the patients or probands—a file describing visits or encounters is optional; and (5) entities may be related to patients, visits, or encounters. Providing information on time points is optional but recommended.

Figure 4 provides an example of how the tool is configured. As can be seen, users are able to specify *entities* that are related to a certain patient or visit and that have *attributes*. Attributes can be mapped to specific columns in the input files. Attributes can be annotated with *meta-attributes*, which are attributes that further specify a specific value for an attribute of a specific entity. In i2b2, these are mapped to *modifiers*. Although there is no direct support for meta-attributes in tranSMART, they can in some cases be represented by creating multiple variants of an attribute that encodes the values of the associated meta-attributes. In addition, there are specific attributes for specifying timestamps and patient or visit identifiers.

The figure also shows an example of how data stored in an *entity-attribute-value* (EAV) model can automatically be denormalized. The EAV model is often used in data collection systems when a large number of different observations are recorded but only a few of them typically apply to a specific patient or proband (eg, lab values). To support this, an additional property *value* is introduced, which can be used to specify how data in EAV form should be denormalized. In the example, one entity will be created in the target systems for each instance of the column *Parameter* having the value from the column *Result* and being annotated with meta-attributes *Unit* and *Norm range*. This is implemented by parsing the input files and populating the configuration with automatically generated parameters for each EAV-encoded data item.

By specifying basic patient, visit, and observational data, the specified EAV entities, the patient data, the observations, an internal model of the ontology, and optionally the associated visits are automatically created. Furthermore, by mapping patients to visits and by relating entities to visits or patients, implicit relationships between the different types of data are constructed. These will also be reflected within the target systems.

Figure 4. Simplified example of an annotation of input files with entities, attributes and relationships. LDL: low density lipoprotein.



Meeting Requirement 3: Automated Data Cleansing and Preprocessing

There are multiple additional features that have been added to the tool based on our experiences with loading a wide range of real-world data sets, which help enforce the syntactic and structural integrity of the input data and which are particularly important due to the heterogeneity of the data sources with respect to these parameters. Important examples include the automated detection of charsets and syntax of input data as well as the automated detection of data types of variables. Features that help enforce semantic integrity include the detection and handling of duplicate data, inconsistent timestamps, and missing values. Finally, support for data filtering and methods for handling uncertainty in timestamps are provided. On a technical level, these tasks are executed as part of either the data loading or the data transformation subphase.

Experimental Design

We evaluated our solution by performing an experimental evaluation of our data-loading approach using different real-world data sets. In the experimental evaluation, we focused on 3 different aspects:

1. *Flexibility:* To demonstrate that our loading tool is able to perform automated data cleansing and restructuring, we used it to load three different types of data sets with varying degrees of structure and cleanliness. Moreover, we also tried to load these data sets using existing data-loading tools to demonstrate that they are not able to process them without prior data cleansing.
2. *Reduced efforts:* To demonstrate that the declarative configuration paradigm of our loading tool significantly reduces the effort required, we compared the number of lines in the configuration files for our tool with the number of lines of the configuration files generated for and needed by existing data-loading tools.
3. *Scalability:* To demonstrate that our approach is computationally feasible, we compared the time needed for automated data cleansing and preprocessing with the time required for actual data loading.

In the experiments, we used real-world data sets from 3 different previous projects: (1) a research data set including *microbiome profiles*, (2) clinical data on *multiple sclerosis*, and (3) *billing data*.

The microbiome profile data set was collected in a study context by our internal medicine department in 2019 and included general information about the probands, lifestyle information obtained through questionnaires, and microbiome profiles (species identified by 16S rRNA gene sequencing) generated from sampled stool, feces, and esophagus tissue. The multiple sclerosis data set was collected by our neurology department since 2010 in the health care context and consisted of longitudinal clinical data, including diagnoses, procedures, clinical scores, medication, lab values, references to biosamples, and metadata of imaging tests. The billing data set consisted of discharge data collected in our hospital in the years 2015-2017 containing demographics and visit data including ventilation time, diagnoses, and procedures. Further details on the projects and use cases supported by these data sets are presented in the *Discussion* section.

For loading data into i2b2, we used the transmart-batch backend, and for loading data into tranSMART, we used the tMDataLoader backend of the pipeline. The experiments were performed with the warehouse instances hosted on a server with Intel Xeon central processing units (CPUs) running at 2.4 GHz with 80 cores, along with 512 GB RAM and 16 TB hard-drives using kernel-based virtual machines provided by Quick EMUlator 2.5.0 running on Ubuntu 18.04. The ETL processes were executed on a desktop machine equipped with a quad-core 3.2 GHz Intel Core i5 CPU running a 64-bit Windows NT kernel, with a 32-bit Java Virtual Machine (1.8.0_202_x86), and with the data input files located on the local file system.

Results

Experiment 1: Flexibility of the Loading Process

In this section, we present results on the flexibility of the loading process for our evaluation data sets and both i2b2 and tranSMART as target systems. The basic properties of the data sets and their representations in the target systems are shown in Table 1.

The microbiome data set originates from a study context and is highly structured. For this reason, and as can be seen in Table 1, i2b2 and tranSMART were both fully able to represent the data set as is. The multiple sclerosis data set, in contrast, was collected in the health care context and consisted of longitudinal clinical data with less structure and a multitude of detailed measurements, such as laboratory values. As can be seen in Table 1, tranSMART could only capture parts of these data (fewer facts by a factor of 6 compared with i2b2) because of missing support for complex time series data and meta-attributes. The billing data set was also highly structured and contained dates of admission and discharge as well as coded diagnoses and procedures. In general, these data could be represented well in i2b2 as well as tranSMART, but the latter system was not able to capture meta-attributes, for example, of diagnoses, resulting in some loss of information.

We emphasize that loading into the different target systems was achieved using the same configuration files. We conclude that our tool provides a high degree of flexibility but that the different target systems are not able to capture all aspects of input data. In general, i2b2 is more suited for representing longitudinal clinical data, and tranSMART is better suited for analyzing highly structured research data.

We further emphasize that our loading pipeline was the only tool with which we were able to load all the data sets described in their raw form without prior transformations or preprocessing. In the remainder of this section, we will briefly cover the issues encountered when using existing open source loading software. We present a detailed comparison with our approach in the *Discussion* section.

When loading the data sets into i2b2, we encountered the following issues: transmart-batch for i2b2 requires the extraction and loading of concept trees into i2b2 before the import of the actual facts. This process is not supported by the tool, and import files also need to be annotated with codes associated with the ontology nodes in the database in an additional preprocessing step. The loading pipeline of IDRT is no longer maintained (over 2.5 years old) and is not compatible with i2b2 1.7.09c and higher, resulting in various errors during data loading. When loading the data sets into tranSMART, we noticed the following problems: tMDataLoader, tmtk, transmart-batch, and ICE could not load the clinical data set where multiple values were provided for the same variable and subject in the same visit. Furthermore, values are required to conform to predefined formats (eg, “yyyy-mm-dd hh:mm:ss” for dates), requiring preprocessing. Transmart-copy could not load any of the data sets used in our experiments without significant preprocessing at the structural and syntactical level, as it required input data to precisely conform to the target schema. TranSMART-ETL could also not load the clinical data set as it was not able to handle missing values. Moreover, it required specific column separators and number formats to be used, requiring input files to be preprocessed accordingly.

Table 1. Overview of the properties of the data sets used in the projects.

Data set	Microbiome profiles	Multiple sclerosis	Billing data
Number of input files	15	19	11
Size of input files in MB	1	497	252
Patients	~50	~7000	~100,000
Visits	~100	~40,000	~300,000
Facts in i2b2[a]	~90,000	~4,600,000	~6,200,000
Facts in tranSMART	~90,000	~750,000	~3,800,000

[a]i2b2: Informatics for Integrating Biology and the Bedside.

Experiment 2: Reduction of Efforts

In this section, we present the results of the reduction of efforts that can be achieved by using our loading tool. We captured this aspect by analyzing the size of files used for actual data loading, which are shown in Table 2. It shows the complexity of configuration files required for data loading with our tool compared with the complexity of the configuration files generated for the backing data loaders. As can be seen, the tool presented in this study generated a large number of files for the different specified entities. Moreover, as a result of the automated denormalization of EAV data and the automated detection of data types, configuring data loading with our tool required significantly fewer lines of configuration parameters

than what would have been required using transmart-batch or tMDataLoader. The configuration files for tranSMART for the multiple sclerosis and the billing data sets were much smaller than the corresponding files for i2b2, as they did not include specifications for meta-attributes.

For the microbiome data set, configuration files for our tool were smaller by factors of between 17.7 (i2b2) and 22.1 (tranSMART). For the multiple sclerosis data set, configuration files for our tool were smaller by factors of between 3.9 (tranSMART) and 216.1 (i2b2). For the billing data set, configuration files where smaller by factors of between 1.2 (tranSMART) and 1135.0 (i2b2). We note that the sizes were (roughly) equal only for the billing data set and tranSMART, which is because this data set is highly structured and because this type of data is well supported by tranSMART. We conclude that our tool can significantly reduce the efforts required for configuring the loading process.

Table 2. Comparison of input required for data loading.

Data set	Microbiome profiles	Multiple sclerosis	Billing data
LOC[a] input	496	1090	83
LOC staging, i2b2[b]	8772	235,582	94,213
LOC staging, tranSMART	10,976	4272	99
Input files	15	19	11
Staging files, i2b2	2207	1034	31
Staging files, tranSMART	2194	854	18

[a]LOC: lines of configuration.

[b]i2b2: Informatics for Integrating Biology and the Bedside.

Experiment 3: Scalability

In this section, we present the results on the scalability of our tools with respect to increasing volumes of data. The execution times measured in the experiments are provided in Table 3.

The table shows the time needed for staging and loading the data from the 3 evaluation data sets for i2b2 and tranSMART. As can be seen, the execution times scaled roughly linearly with the number of facts loaded into the target systems. Moreover, the relative time needed for data staging was the highest for the multiple sclerosis data set, which is also the data set with the highest complexity, thus requiring the most preprocessing.

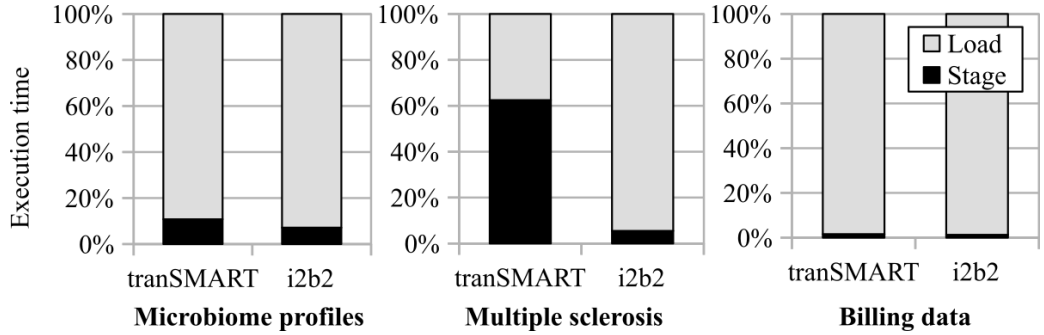
Figure 5 provides an overview of the relationship between the times needed for staging and loading. As can be seen, the (relative) staging times for tranSMART were generally higher than those for i2b2. This can be explained by the fact that more data normalization and restructuring were needed to be performed by the tool to ensure that the data could be loaded into the target system. In addition, more complicated procedures for duplicate detection were needed, as there is little support for the time axis in tranSMART. In summary, we conclude that our approach is scalable and can be used to process large data sets.

Table 3. Execution times of data-loading processes in seconds.

Data set	Microbiome profiles	Multiple sclerosis	Billing data
tranSMART			
Staging time	13	687	91
Loading time	109	413	5687
Total time	122	1100	5778
i2b2[a]			
Staging time	11	804	790
Loading time	144	13,895	61,417
Total time	155	14,699	62,208

[a]i2b2: Informatics for Integrating Biology and the Bedside.

Figure 5. Percentage of loading and staging times regarding the complete process. i2b2: Informatics for Integrating Biology and the Bedside.



Discussion

Principal Findings

We have presented a comprehensive cloud-based platform and a flexible data-loading pipeline to enable the agile provisioning of clinical and translational data warehousing solutions. We have presented an extensive experimental evaluation, dealing with different types of data and targeting platforms with different data analytics capabilities. The results of our analysis show that the presented platform significantly simplifies the management of the supported data warehousing solutions and enables quick loading of data in various representations. This enables the development of such platforms in close cooperation with users based on short feedback cycles. The cloud-based hosting infrastructure and the data-loading pipeline are available as open source software.

The infrastructure and tools presented in this study and the data sets used in our experimental evaluation have been used to support a variety of real-world projects. In particular, the infrastructure is being used to support a large clinical research center [40] that studies shifts in the composition and activity of the microbial ecosystem focusing on clinical endpoints that are associated with well-documented changes in the gut microbiome (inflammation and cancer). For this purpose, a platform is being set up to provide researchers with integrated access to different types of data generated within the consortium. Moreover, our platform is being used within the DIFUTURE (Data Integration for Future Medicine) project to improve data availability and accessibility through an integrated view on health care and research resources, such as biobanks [6]. An important example of one of the use cases of the project is the development of an infrastructure for personalized optimal treatment of multiple sclerosis combined with efforts to better understand the disease in general. Finally, the billing data set has been used in a nationwide cross-site analysis aiming at the reproduction of published comorbidity scores and the descriptive analysis and visualization of the distribution of comorbidity scores as well as the distribution of rare diseases in Germany [41].

Comparison With Prior Work

Analytics Platforms

Currently, our solutions support i2b2 version 1.7.09c and tranSMART version 16.3. In future work, we plan to add support for further warehousing platforms and further versions to support further use cases. An important system of interest is i2b2-tranSMART, which is the result of an initiative to integrate tranSMART with the i2b2 cohort selection services and improved support for managing time series data [42]. In theory, this would obviate the need to support 2 different systems (i2b2 and tranSMART) with a similar technological basis. However, i2b2-tranSMART is still under active development and is not yet suitable for deployment in production environments. It is planned to release this software directly as a Docker container; therefore, we expect little effort to integrate it into the presented environment.

The OMOP CDM and OHDSI toolset also provide an interesting target platform [11]. OHDSI is an international collaborative initiative aimed at making clinical data accessible to analytics efforts, also in distributed settings, to generate actionable insights for improving health care. The OMOP CDM is a CDM for consistently representing health care data from diverse sources by making the relationships between different concepts explicit [11]. The OHDSI project provides a wide range of analytics front-ends, such as ACHILLES (Automated Characterization of Health Information at Large-scale Longitudinal Evidence Systems) or Atlas, an open source application developed as a part of OHDSI intended to provide a unified interface to patient level data and analytics. Both are aimed at end users and can be deployed over the OMOP CDM. Supporting OMOP/OHDSI within the described cloud-based hosting infrastructure will not be too complex. Implementing an agile loading process, however, will be challenging as the OMOP CDM requires a significant amount of data normalization and encoding with standard terminologies. Finally, cBioPortal would be an important additional system to support as it provides a platform for interactive exploration of multi-dimensional genomics data sets, intending to also support rapid, intuitive, and high-quality access to molecular data and clinical data [43]. A dockerized version for the presented cloud environment has already been implemented, but integrating the software with our data-loading pipeline requires more work.

Cloud-Based Infrastructures

Regarding cloud-based management infrastructures for clinical and translational data warehousing, most studies focus on i2b2 only. The *i2b2 Wizard*, which is part of the IDRT, as well as i2b2 Quickstart aims to simplify installation, setup, and

administration of single i2b2 instances. There are also images available on Docker Hub. However, as neither the source code of these images is publicly available in full nor can their authenticity be verified (eg, using Docker Content Trust [DCT]), we could not use them as a base for further development because of security considerations. For tranSMART, a large number of images are available on Docker Hub. However, they have not been maintained for some time, contain artifacts with unclear provenance, or their documentation leaves out important aspects.

We compared these alternative solutions with our approach with respect to the following criteria:

1. *Supported target platforms* indicates whether a solution can be used for the current major version of i2b2 (ie, 1.7.x) and/or tranSMART (ie, 16.3).
2. *Container-based* denotes whether the solution is encapsulated using container virtualization, which significantly increases the ease and robustness of the installation procedures.
3. *Security by default* covers 3 subcriteria—whether *transport encryption* is part of the default deployment, whether the solution automatically provides strong default passwords and whether these can be changed in an integrated way, that is, without risking to break the application (*password management*), and whether the solution uses or provides means to verify the trustworthiness of the installation package, for example, by using digital signatures or by providing the source code (*trusted runtime environment*).
4. *Unified interface* shows whether the solution helps manage multiple warehouse instances of different types.
5. *Sustainability* covers 2 subcriteria—*full availability of source code* is important for customizing the solution to local requirements and the *last update* of the installation package is an indicator of whether the solution is actively maintained by the provider of the solution or by the community.

The results of the comparison are presented in Tables 4-5.

As can be seen, our infrastructure is the only off-the-shelf solution supporting both i2b2 and tranSMART. Moreover, our software, the IDRT i2b2 Wizard, and i2b2 Quickstart are the only solutions that fulfill requirement 1 (robust installation of a trusted runtime environment), as the other (cloud-based) solutions are not capable of providing a trusted runtime environment due to the reasons explained above. However, i2b2 Wizard and i2b2 Quickstart are not container-based solutions but rather script-based solutions and thus are significantly less flexible than our tool, which is based on container virtualization. Furthermore, our tool is the only solution that fulfills requirement 2 (unified installation and maintenance) because it provides integrated support for both i2b2 and tranSMART through common commands. Finally, our tool is the only solution that fulfills requirement 3 (built-in security) as it is the only solution that provides out-of-the-box support for multiple important security features, such as transport encryption and strong passwords. The IDRT i2b2 Wizard is quite outdated and has not received updates in more than 2 years.

Table 4. Comparison of provisioning infrastructures: Our solution, IDRT[a] and i2b2[b] Quickstart.

Feature	Our solution	IDRT[a] [26]	i2b2[b] Quickstart [27]
Supported target platforms			
i2b2 (current major version)	Yes	No	Yes
tranSMART (current major version)	Yes	No	No
Container based	Yes	No	No
Security by default			
Transport encryption	Yes	No	No
Password management	Yes	No	No
Trusted runtime environment	Yes	Yes	Yes
Unified interface			
Central multi-instance management	Yes	No	No
Sustainability			
Full availability of source code	Yes	Yes	Yes
Last update	March 2020	August 2017	February 2020

[a]IDRT: Integrated Data Repository Toolkit.

[b]i2b2: Informatics for Integrating Biology and the Bedside.

Table 5. Comparison of provisioning infrastructures: i2b2[a] on Dockerhub, tranSMART on Dockerhub, and manual installation.

Feature	i2b2[a] on Dockerhub [28]	tranSMART on Dockerhub	Manual installation
Supported target platforms			
i2b2 (current major version)	Yes	No	Yes
tranSMART (current major version)	No	Yes	Yes
Container based	Yes	Yes	No
Security by default			
Transport encryption	No	No	Yes
Password management	No	No	Yes
Trusted runtime environment	No	No	Yes
Unified interface			
Central multi-instance management	No	No	No
Sustainability			
Full availability of source code	No	No	Yes
Last update	February 2020	October 2019	April 2020

[a]i2b2: Informatics for Integrating Biology and the Bedside.

Data-Loading Tools

In addition to transmart-batch and tMDataLoader, which are both used by our solution, there are further data loaders for tranSMART and i2b2. First, transmart-ETL is the standard loading tool for tranSMART. It is included in the standard software installation of tranSMART and is based on the Pentaho Data Integration platform. Second, ICE is a data loading and curation tool supporting a graphical user interface [36]. Third, transmart-copy is a very lightweight loading tool that copies data provided in a tabular form into the tables of the tranSMART database. tmtk is the solution most similar to our approach. It is a Python-based solution that enables the integration of data via a high-level language and several classes. It is typically used in Jupyter notebooks. Analogous to our solution, it uses transmart-batch as a loading tool. It also supports flexible means for organizing data into entities and attributes through an additional graphical tool called the Arborist. Moreover, for i2b2 only, there are other loading tools available. The most comprehensive is the IDRT Import and Mapping Tool [26]. The tool supports various import formats, such as CSV files; provides access to structured query language databases, such as Clinical Data Interchange Standards Consortium (CDISC) Operational Data Model (ODM) [44,45]; and provides direct support for CDMs, that are, for example, used for billing purposes. Talend Open Studio is used for all ETL processes.

We compared these tools with our approach with respect to the following criteria:

1. As in the previous section, the criterion *supported target platforms* shows whether a solution can be used for the current major version of i2b2 (ie, 1.7.x) and/or tranSMART (ie, 16.3).
2. The criterion *EAV schema support* indicates whether the tool supports EAV input data with multiple attribute columns (*multi-column*) or with only one attribute column (*basic*).
3. *Automated data cleansing and preprocessing* covers subcriteria indicating whether the tool can handle *different encodings*, *data types*, and *syntaxes* for different data sources or if the tool requires all incoming data to conform to a single, predefined specification, and the subsequent subcriteria show whether the tool can handle *missing or invalid data* and *duplicate data* or whether the ETL process is aborted if it encounters one of these anomalies.
4. The criterion *loading strategy* indicates whether the tool employs other data-loading tools (*meta*) or whether the tool implements its own loading procedures (*direct*).
5. *Configuration paradigm* indicates whether the tool configuration follows a declarative approach or an *imperative* approach.
6. The criterion *sustainability*, as in the previous section, covers 2 subcriteria with the same semantics—*full availability of source code* and the *last update*.

The results of the comparison are provided in Tables 6-7.

Table 6. Comparison of extraction-transformation-loading tools: Our solution, tranSMART-ETL[a], tMData-loader, and transmart-batch.

Feature	Our solution	tranSMART-ETL[a] [33]	tMData-loader [34]	transmart-batch [35]
Supported target platforms				
i2b2[b] (current major version)	Yes	No	No	Yes
tranSMART (current major version)	Yes	Yes	Yes	Yes
EAV[c] schema support	Multi-column	Basic	Basic	Basic
Automated data cleansing and preprocessing				
Different encodings, data types, and syntaxes	Yes	No	No	No
Missing or invalid data	Yes	No	No	No
Duplicate data	Yes	Yes	No	No
Loading strategy	Meta	Direct	Direct	Direct
Configuration paradigm	Declarative	Imperative	Imperative	Imperative
Sustainability				
Source code fully available	Yes	Yes	Yes	Yes
Last update	March 2020	March 2018	December 2017	June 2016

[a]ETL: extraction-transformation-loading.
[b]i2b2: Informatics for Integrating Biology and the Bedside.
[c]EAV: entity-attribute-value.

Table 7. Comparison of extraction-transformation-loading tools: Integrated Curation Environment, Integrated Data Repository Toolkit, transmart-copy, and tmtk[a].

Feature	ICE[b] [36]	IDRT[c] [26]	tranSMART-copy [37]	tmtk[a] [38]
Supported target platforms				
i2b2[d] (current major version)	No	No	No	No
tranSMART (current major version)	Yes	No	Yes	Yes
EAV[e] schema support	Basic	No	No	Basic
Automated data cleansing and preprocessing				
Different encodings, data types, and syntaxes	No	No	No	No
Missing or invalid data	No	No	No	Yes
Duplicate data	No	Yes	No	No
Loading strategy	Meta	Direct	Direct	Meta
Configuration paradigm	Imperative	Imperative	Imperative	Imperative
Sustainability				
Source code fully available	No	Yes	Yes	Yes
Last update	July 2016	August 2017	December 2019	February 2020

[a]tmtk: TranSMART data curation toolkit.
[b]ICE: Integrated Curation Environment.
[c]IDRT: Integrated Data Repository Toolkit.
[d]i2b2: Informatics for Integrating Biology and the Bedside.
[e]EAV: entity-attribute-value.

As can be seen, our solution and transmart-batch are the only tools to support both i2b2 and tranSMART and thus to fulfill requirement 1 (p*latform independence*). Requirement 2 (*support for different types of data*) is strongly connected to requirement 3 (*automated data cleansing and preprocessing)*. At the structural level, our tool is the only tool to support EAV schema resolution in which multiple columns can be combined (eg, *lab analytes* together with *units of measurement*) and thus is the

only one to fulfill requirement 2 (*support for different types of data*). Moreover, our tool is also the only one that is capable of automatically detecting and handling multiple input data properties, such as encodings, syntaxes, and data types, and thus to ingest heterogeneous data often encountered in the clinical context. Our tool and tranSMART-ETL are both capable of automatically handling duplicate data. In addition to our tool, tmtk and ICE are also meta-loading tools; however, they have fewer data cleansing functionalities. tMDataLoader, ICE, and IDRT are quite outdated and have not received updates in more than 1.5 years.

We conclude that our set of tools is the only solution that supports all requirements outlined in the *Methods* section. Moreover, our solutions are fully open source software, allowing users to maintain their own version if needed, thus decreasing the risks of adoption and improving sustainability.

Limitations and Future Work

In future work, we plan to address the limitations of the current version of the infrastructure. First, the current implementation does not scale to huge data volumes. At the infrastructure level, this would require support for shared databases. On the data-loading layer, support for processing data in the form of smaller blocks or chunks is needed. Extending the data-loading pipeline with this feature will be relatively straightforward. However, the loading tools used as backends need to support incremental loading, which is currently only supported for i2b2 with the transmart-batch backend. In general, the pipeline would benefit significantly from incremental loading capabilities; therefore, we are exploring options to integrate an incremental loading procedure directly into the software.

An additional area of future improvements is authentication and authorization management. For deployments with a large user base, the use of single sign-on concepts, such as OAuth2 [46], will become relevant. As tranSMART uses Spring Security [47], which supports OAuth2, this should be straightforward to accomplish. However, the software stack used by i2b2 does not support OAuth2 natively. Therefore, we plan to evaluate the approach described by Wagholikar et al [48]. Another limitation in terms of information security is that our use of DCT [31] is currently restricted to checking the authenticity and integrity of the base images when building the images. In future versions, we plan to use DCT to sign images as well, which is particularly important when publishing them on the internet.

The current version of the infrastructure focuses on clinical data or selected genomic variants. TranSMART, however, has built-in support for a wide range of high-dimensional data types (see the *Selection of Target Systems* section). In future work, we plan to add support for loading these types of data as well. Although this will require some effort, such data are typically much more structured and represented in standardized formats than the data considered in this study.

Currently, our loading pipeline focuses on automated structural and syntactic harmonization. Automated mapping procedures to standard terminologies are not yet implemented, mainly because in a first step, we have developed the pipeline following our project-specific requirements. Here, all data sets integrated until now have mostly either been (1) collected in a structured form, using standard terminologies as they were captured; (2) mapped to standard terminologies before they were fed into our pipeline; or (3) loaded for use cases that did not require mapping to semantic standards. However, semantic harmonization is a very important process, and the implementation of interfaces to terminology and ontology services directly into our pipeline is part of our development roadmap.

Finally, we also plan to integrate a wide range of privacy-enhancing technologies into the pipeline. In previous work, we have already integrated a flexible method for data anonymization into an earlier version of our software [49]. Currently, we are working on integrating the pipeline with a HL7 FHIR (Health Level Seven Fast Healthcare Interoperability Resources)–based pseudonymization component.

Summary and Conclusions

In this paper, we have presented a flexible infrastructure that supports the agile development and provisioning of translational data analytics platforms to researchers. Our solution helps to bridge the interdisciplinary gap between clinicians and informaticians by enabling the creation of data warehousing solutions in an iterative process involving short feedback cycles following a pay-as-you-go approach [15]. We have achieved this by combining a Docker-based (private) cloud infrastructure for managing warehouse instances with a flexible and easy-to-use loading pipeline based on a declarative configuration paradigm. We have used the platform successfully to support a wide range of projects that used different types of data, which we used in our experiments. The solutions described in this paper are available to the community as open source software [19,20].

Acknowledgments

The authors wish to thank the reviewers for their insightful comments, which helped to significantly improve the earlier versions of this manuscript. The work was, in parts, funded by the German Federal Ministry of Education and Research within the *Medical Informatics Funding Scheme* under reference number 01ZZ1804A (Data Integration for Future Medicine).

Conflicts of Interest

None declared.

References

1. Hood L, Friend SH. Predictive, personalized, preventive, participatory (P4) cancer medicine. Nat Rev Clin Oncol 2011 Mar;8(3):184-187. [doi: 10.1038/nrclinonc.2010.227] [Medline: 21364692]

2. Schneeweiss S. Learning from big health care data. N Engl J Med 2014 Jul 5;370(23):2161-2163. [doi: 10.1056/NEJMp1401111] [Medline: 24897079]
3. Esteva A, Robicquet A, Ramsundar B, Kuleshov V, DePristo M, Chou K, et al. A guide to deep learning in healthcare. Nat Med 2019 Jan;25(1):24-29. [doi: 10.1038/s41591-018-0316-z] [Medline: 30617335]
4. Tran BX, McIntyre RS, Latkin CA, Phan HT, Vu GT, Nguyen HL, et al. The current research landscape on the artificial intelligence application in the management of depressive disorders: a bibliometric analysis. Int J Environ Res Public Health 2019 Jun 18;16(12):- [FREE Full text] [doi: 10.3390/ijerph16122150] [Medline: 31216619]
5. Friedman CP, Wong AK, Blumenthal D. Achieving a nationwide learning health system. Sci Transl Med 2010 Dec 10;2(57):57cm29. [doi: 10.1126/scitranslmed.3001456] [Medline: 21068440]
6. Prasser F, Kohlbacher O, Mansmann U, Bauer B, Kuhn KA. Data integration for future medicine (DIFUTURE). Methods Inf Med 2018 Jul;57(S 01):e57-e65 [FREE Full text] [doi: 10.3414/ME17-02-0022] [Medline: 30016812]
7. Kimball R, Caserta J. Surronding the requirements. In: The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data. Hoboken, New Jersey, USA: John Wiley & Sons; 2011:3-28.
8. Halevy A, Korn F, Noy N, Olston C, Polyzotis N, Roy S, et al. Goods: Organizing Google's Datasets. In: Proceedings of the 2016 International Conference on Management of Data. 2016 Presented at: SIGMOD'16; June 26-July 1, 2016; San Francisco, CA, USA. [doi: 10.1145/2882903.2903730]
9. Murphy SN, Weber G, Mendis M, Gainer V, Chueh HC, Churchill S, et al. Serving the enterprise and beyond with informatics for integrating biology and the bedside (i2b2). J Am Med Inform Assoc 2010;17(2):124-130 [FREE Full text] [doi: 10.1136/jamia.2009.000893] [Medline: 20190053]
10. Scheufele E, Aronzon D, Coopersmith R, McDuffie MT, Kapoor M, Uhrich CA, et al. tranSMART: an open source knowledge management and high content data analytics platform. AMIA Jt Summits Transl Sci Proc 2014;2014:96-101 [FREE Full text] [Medline: 25717408]
11. Hripcsak G, Duke JD, Shah NH, Reich CG, Huser V, Schuemie MJ, et al. Observational health data sciences and informatics (OHDSI): opportunities for observational researchers. Stud Health Technol Inform 2015;216:574-578 [FREE Full text] [Medline: 26262116]
12. McMurry AJ, Murphy SN, MacFadden D, Weber G, Simons WW, Orechia J, et al. SHRINE: enabling nationally scalable multi-site disease studies. PLoS One 2013;8(3):e55811 [FREE Full text] [doi: 10.1371/journal.pone.0055811] [Medline: 23533569]
13. Killcoyne S, Boyle J. Managing chaos: lessons learned developing software in the life sciences. Comput Sci Eng 2009 Dec;11(6):20-29 [FREE Full text] [doi: 10.1109/MCSE.2009.198] [Medline: 20700479]
14. Kannan V, Basit MA, Youngblood JE, Bryson TD, Toomay SM, Fish JS, et al. Agile co-development for clinical adoption and adaptation of innovative technologies. Health Innov Point Care Conf 2017 Nov;2018:56-59 [FREE Full text] [doi: 10.1109/HIC.2017.8227583] [Medline: 30364762]
15. Franklin M, Halevy A, Maier D. From databases to dataspaces. SIGMOD Rec 2005 Dec;34(4):27-33. [doi: 10.1145/1107499.1107502]
16. Prasser P. Incremental Ontology-Based Integration for Translational Medical Research. Technical University of Munich. 2013. URL: https://mediatum.ub.tum.de/doc/1119200/document.pdf [accessed 2020-05-31]
17. Petrović M, Vučković M, Turajlić N, Babarogić S, Aničić N, Marjanović Z. Automating ETL processes using the domain-specific modeling approach. Inf Syst E-Bus Manage 2016 Jul 9;15(2):425-460. [doi: 10.1007/s10257-016-0325-8]
18. Dingsøyr T, Nerur S, Balijepally V, Moe NB. A decade of agile methodologies: towards explaining agile software development. J Syst Software 2012 Jun;85(6):1213-1221. [doi: 10.1016/j.jss.2012.02.033]
19. Spengler H. Analytics Environment. DIFUTURE. 2020. URL: https://gitlab.com/DIFUTURE/analytics-environment [accessed 2020-05-31]
20. Spengler H, Lang C, Mahapatra T, Gatz I, Prasser F. ETL Pipeline. DIFUTURE. 2020. URL: https://gitlab.com/DIFUTURE/etl-pipeline [accessed 2020-05-31]
21. Canuel V, Rance B, Avillach P, Degoulet P, Burgun A. Translational research platforms integrating clinical and omics data: a review of publicly available solutions. Brief Bioinform 2015 Mar;16(2):280-290 [FREE Full text] [doi: 10.1093/bib/bbu006] [Medline: 24608524]
22. Tatonetti NP, Denny JC, Murphy SN, Fernald GH, Krishnan G, Castro V, et al. Detecting drug interactions from adverse-event reports: interaction between paroxetine and pravastatin increases blood glucose levels. Clin Pharmacol Ther 2011 Jul;90(1):133-142 [FREE Full text] [doi: 10.1038/clpt.2011.83] [Medline: 21613990]
23. Jannot A, Zapletal E, Avillach P, Mamzer M, Burgun A, Degoulet P. The Georges Pompidou University hospital clinical data warehouse: a 8-years follow-up experience. Int J Med Inform 2017 Jun;102:21-28. [doi: 10.1016/j.ijmedinf.2017.02.006] [Medline: 28495345]
24. Geerts H, Dacks PA, Devanarayan V, Haas M, Khachaturian ZS, Gordon MF, Brain Health Modeling Initiative (BHMI). Big data to smart data in Alzheimer's disease: the brain health modeling initiative to foster actionable knowledge. Alzheimers Dement 2016 Sep;12(9):1014-1021 [FREE Full text] [doi: 10.1016/j.jalz.2016.04.008] [Medline: 27238630]
25. i2b2 Installation Guide. i2b2 Community Wiki. 2020. URL: https://community.i2b2.org/wiki/display/getstarted/i2b2+Installation+Guide [accessed 2020-05-31]

26. Bauer CR, Ganslandt T, Baum B, Christoph J, Engel I, Löbe M, et al. Integrated data repository toolkit (IDRT). A suite of programs to facilitate health analytics on heterogeneous medical data. Methods Inf Med 2016;55(2):125-135. [doi: 10.3414/ME15-01-0082] [Medline: 26534843]
27. Wagholikar KB, Mendis M, Dessai P, Sanz J, Law S, Gilson M, et al. Automating installation of the integrating biology and the bedside (i2b2) platform. Biomed Inform Insights 2018;10:1178222618777749 [FREE Full text] [doi: 10.1177/1178222618777749] [Medline: 29887730]
28. Wagholikar KB, Dessai P, Sanz J, Mendis ME, Bell DS, Murphy SN. Implementation of informatics for integrating biology and the bedside (i2b2) platform as Docker containers. BMC Med Inform Decis Mak 2018 Jul 16;18(1):66 [FREE Full text] [doi: 10.1186/s12911-018-0646-2] [Medline: 30012140]
29. Internet Engineering Task Force. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. Request for Comments. 2008. URL: https://tools.ietf.org/html/rfc5280#section-4.2.1.6 [accessed 2020-05-31]
30. Souppaya M, Morello J, Scarfone K. Application Container Security Guide. NIST Special Publication 800-190. 2017. URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-190.pdf [accessed 2020-05-31]
31. Content Trust in Docker. Docker Documentation. 2020. URL: https://docs.docker.com/engine/security/trust/content_trust/ [accessed 2020-05-31]
32. T'so T. pwgen. GitHub. 2018. URL: https://github.com/tytso/pwgen [accessed 2020-05-31]
33. transmart-ETL. GitHub. 2018. URL: https://github.com/transmart/tranSMART-ETL [accessed 2020-05-31]
34. tMDataLoader. GitHub. 2020. URL: https://github.com/Clarivate-LSPS/tMDataLoader [accessed 2020-05-31]
35. tranSMART Batch. GitHub. 2016. URL: https://github.com/tranSMART-Foundation/transmart-batch [accessed 2020-05-31]
36. transmart-ICE. GitHub. 2016. URL: https://github.com/transmart/transmart-ICE [accessed 2020-05-31]
37. The Hyve. transmart-copy. GitHub. 2019. URL: https://github.com/thehyve/transmart-core/tree/dev/transmart-copy [accessed 2020-05-31]
38. The Hyve. tmtk. GitHub. 2020. URL: https://github.com/thehyve/tmtk/ [accessed 2020-05-31]
39. Lloyd J. Practical Advantages of Declarative Programming. In: Joint Conference on Declarative Programming. 1994 Presented at: GULP-PRODE'94; September 19-22, 1994; Peñiscola, Spain.
40. Haller D. Microbiome Signatures. CRC 1371. 2019. URL: https://www.sfb1371.tum.de/ [accessed 2020-05-31]
41. Kamdje-Wabo G, Gradinger T, Löbe M, Lodahl R, Seuchter SA, Sax U, et al. Towards structured data quality assessment in the german medical informatics initiative: initial approach in the MII demonstrator study. Stud Health Technol Inform 2019 Aug 21;264:1508-1509. [doi: 10.3233/SHTI190508] [Medline: 31438205]
42. I2b2 tranSMART Foundation. tranSMART PMC Roadmap. Google Docs. 2018. URL: http://roadmap-i2b2-transmart-pmc.hms.harvard.edu [accessed 2020-05-31]
43. Gao J, Aksoy BA, Dogrusoz U, Dresdner G, Gross B, Sumer SO, et al. Integrative analysis of complex cancer genomics and clinical profiles using the cBioPortal. Sci Signal 2013 May 2;6(269):pl1 [FREE Full text] [doi: 10.1126/scisignal.2004088] [Medline: 23550210]
44. Kubick WR, Ruberg S, Helton E. Toward a comprehensive CDISC submission data standard. Drug Inf J 2016 Aug 28;41(3):373-382. [doi: 10.1177/009286150704100311]
45. Hume S, Aerts J, Sarnikar S, Huser V. Current applications and future directions for the CDISC operational data model standard: a methodological review. J Biomed Inform 2016 May;60:352-362 [FREE Full text] [doi: 10.1016/j.jbi.2016.02.016] [Medline: 26944737]
46. The OAuth 2.0 Authorization Framework. IETF Tools. 2012. URL: https://tools.ietf.org/html/rfc6749 [accessed 2020-05-31]
47. Nachimuthu N. Spring Security OAuth. Spring Projects. 2016. URL: https://spring.io/projects/spring-security-oauth [accessed 2020-05-31]
48. Wagholikar KB, Mandel JC, Klann JG, Wattanasin N, Mendis M, Chute CG, et al. SMART-on-FHIR implemented over i2b2. J Am Med Inform Assoc 2017 Mar 1;24(2):398-402 [FREE Full text] [doi: 10.1093/jamia/ocw079] [Medline: 27274012]
49. Prasser F, Spengler H, Bild R, Eicher J, Kuhn KA. Privacy-enhancing ETL-processes for biomedical data. Int J Med Inform 2019 Jun;126:72-81 [FREE Full text] [doi: 10.1016/j.ijmedinf.2019.03.006] [Medline: 31029266]

Abbreviations

ACHILLES: Automated Characterization of Health Information at Large-scale Longitudinal Evidence Systems
CDISC: Clinical Data Interchange Standards Consortium
CPU: central processing unit
CSV: comma-separated values
CDM: common data model
DCT: Docker Content Trust
DIFUTURE: Data Integration for Future Medicine
EAV: entity-attribute-value
ER: entity-relationship

ETL: extraction-transformation-loading
HL7 FHIR: Health Level Seven Fast Healthcare Interoperability Resources
i2b2: Informatics for Integrating Biology and the Bedside
ICE: Integrated Curation Environment
IDRT: Integrated Data Repository Toolkit
IT: information technology
ODM: Operational Data Model
OHDSI: Observational Health Data Sciences and Informatics
OMOP: Observational Medical Outcomes Partnership
tmtk: TranSMART data curation toolkit

Edited by C Lovis; submitted 17.09.19; peer-reviewed by E Frontoni, R Ho, E Andrikopoulou, Z He; comments to author 05.12.19; revised version received 16.02.20; accepted 06.05.20; published 21.07.20

Please cite as:
Spengler H, Lang C, Mahapatra T, Gatz I, Kuhn KA, Prasser F
Enabling Agile Clinical and Translational Data Warehousing: Platform Development and Evaluation
JMIR Med Inform 2020;8(7):e15918
URL: https://medinform.jmir.org/2020/7/e15918
doi: 10.2196/15918
PMID:

A.2 Improving Data Quality in Medical Research

Improving Data Quality in Medical Research: A Monitoring Architecture for Clinical and Translational Data Warehouses

Helmut Spengler*, Ingrid Gatz*, Florian Kohlmayer†, Klaus A. Kuhn*, Fabian Prasser‡§
*Institute of Medical Informatics, Statistics and Epidemiology, Technical University of Munich, Munich, Germany
{helmut.spengler, ingrid.gatz, klaus.kuhn}@tum.de
†Bitcare GmbH, Munich, Germany
florian.kohlmayer@bitcare.de
‡Berlin Institute of Health, Berlin, Germany
§Charité - Universitätsmedizin Berlin, Berlin, Germany
fabian.prasser@charite.de

***Abstract*—Clinical and translational data warehouses are important infrastructure building blocks for modern data-driven approaches in medical research. These analytics-oriented databases have been designed to integrate heterogeneous biomedical datasets from different sources and to support use cases such as cohort selection and ad-hoc data analyses. However, the lack of clear definitions of source data and controlled data collection procedures often raises concerns about the quality of data provided in such environments and, consequently, about the evidence level of related findings.**

To address these problems, we present an architecture that helps to monitor data quality issues when importing data into warehousing solutions using ETL (Extraction, Transformation, Load) processes. Our approach provides software developers with an API (Application Programming Interface) for logging detailed and structured information about data quality issues encountered. This information can then be displayed in dynamic dashboards, the evolution of data quality can be monitored over time, and quality issues can be traced back to their source. Our architecture supports several well-known data quality dimensions, addressing conformance, completeness, and plausibility.

We present an open-source implementation, which is compatible with common clinical and translational data warehousing platforms, such as i2b2 and tranSMART, and which can be used in conjunction with many ETL environments.

***Index Terms*—data warehouses, data quality, monitoring, i2b2, tranSMART, architecture**

I. Introduction

A. Background

Modern data-driven approaches in medical research promise new advances in the prevention, diagnosis, and treatment of diseases [1]. Clinical and translational data warehousing platforms are important infrastructure building blocks for providing medical researchers with unified access to the large datasets needed to realize this. These analytics-oriented databases have been designed to integrate heterogeneous biomedical datasets from different sources and to support use cases such as cohort selection and ad-hoc data analyses [2], [3]. Important platforms include *Informatics for Integrating Biology and the Bedside* (i2b2) [4] and its derivative *tranSMART* [5], the Observational Medical Outcomes Partnership (OMOP) Common Data Model (CDM) [6] and the tools provided by OHDSI [6], which can be deployed on top of this data model, as well as federated and distributed solutions, such as the Shared Health Research Information Network (SHRINE) [7].

Data are typically replicated into these warehouses using Extract-Transform-Load (ETL) processes [8], [9]: (1) they are extracted from source systems, (2) cleansed, harmonized and transformed into a form suitable for analyses, and (3) loaded into the target platform. Since source data have often been collected for other purposes initially, e.g., for clinical care or quality improvement [10], [11], the problem of overcoming heterogeneity at the technical, structural, and semantic level is often aggravated by a lack of uniform and controlled data collection procedures and clear data definitions. Therefore, *data cleansing*, i.e., the process of detecting and correcting – as far as possible – data quality issues is an important part of ETL processes.

However, this also raises concerns about data quality in warehousing platforms and, as a consequence, about the level of evidence of the findings generated from these data [12], [13]. There is a considerable body of work on data quality in medical research [13]–[16], which has been well characterized by the statement "unless adequate controls are embedded throughout the data lifecycle, data-driven health care will not live up to its expectations" [17]. To solve the problem, a number of data quality frameworks have been proposed [14], [18]–[20]. Kahn et al. have summarized these approaches into a common framework [15], which has for example been used by projects from the OHDSI program [6] or PCORnet [21]. We use this framework as a theoretical foundation for our approach and describe it in more detail in Section III-A. In the next section, we provide an overview of existing architectures

The work was, in parts, funded by the German Federal Ministry of Education and Research (BMBF) within the "Medical Informatics Funding Scheme" under reference number 01ZZ1804A (DIFUTURE).

and implementations, together with their limitations.

B. Related work

Achilles Heel [22] and the *Data Quality Dashboard* [23] have both been developed in the context of OHDSI [24], provide comprehensive tools for data quality checks, but are strongly tied to the OMOP CDM and cannot be used for monitoring i2b2 or tranSMART or for integrated monitoring of multiple warehouse instances. Furthermore, they do not provide mechanisms to capture quality problems encountered during the ETL process. Of the two, only Achilles Heel is able to analyze the evolution of data quality metrics over time. The latter two limitations also apply to the approaches by Bialke et al. [25] and Juárez et al. [26].

The Sentinel Initiative [27] has established a comprehensive *data quality review and characterization* process [28], which also includes tools for data quality assessment and which covers all important aspects of data quality management. However, usage of this process and its tools is strongly coupled to the membership in the according network. For instance, it depends on services provided by the Sentinel Operations Center (SOC) and the tools require data to be modeled according to the Sentinel Common Data Model (SCDM). Furthermore, licenses for the SAS statistical software suite are required. Altogether, this strongly limits the possible applications outside the initiative.

The MIRACUM approach [29] considers the ETL processes as an important factor for data quality analysis and thus quality improvement. The authors describe an implementation that supports different clinical and translational warehousing platforms, such as i2b2, tranSMART, or the OMOP CDM. However, it is left open, if the approach allows analyzing the temporal evolution of data quality metrics, how exactly data quality issues found during the ETL process are registered and stored, and their software has not been published.

We conclude that although practical approaches to improving data quality in medical research exist, many approaches (e.g., Data Quality Dashboard and the approaches by Bialke et al. and Juárez et al.) focus on covering the measurement of quality parameters at a certain stage of the data lifecycle – they do not provide facilities for fine-grained logging of data quality issues encountered during the ETL process, nor the monitoring of the effectiveness of correctional activities. Some are only available for certain data warehousing platforms (e.g., Achilles Heel, Data Quality Dashboard, Sentinel). The only alternative approach that does not exhibit these limitations (MIRACUM), leaves open, whether the evolution of data quality over time can be monitored and how flexible the logging capabilities are. Furthermore, no publicly available implementation has been provided so far.

II. Contribution and outline

The aim of the work described in this article was to close these gaps by developing a software system which integrates (1) well-known methods for data cleansing for data warehouses with (2) methodologies of measuring data quality in medical research and (3) modern methods of monitoring software systems and thus facilitates an iterative and continuous process of improving data quality in medical research. To this end, we provide ETL developers with an error event logging facility to store detailed and structured information about data quality issues encountered during the loading process for later analysis. An auditing service analyzes these events and displays the results together with well-known metrics reflecting the quality of data loaded into the target warehouse.

We present an implementation, which fulfills three major goals. Firstly, we wanted to be able to display and monitor the evolution of the results of data quality audits over time and thus facilitate the discovery of anomalies in loading processes that would otherwise not be visible. Secondly, our solution should be able to provide an integrated view of the results of several data marts, supporting different warehousing platforms. Thirdly, our solution should be able to ingest and store coordinates (e.g. source file and attribute names, row-Ids) of noteworthy data points of the source data in a structured form in order to facilitate tracing back data quality issues to their source.

The remainder of this paper is structured as follows: in Section III we describe the general design of our solution, including the data quality metrics used, the architecture and its different components. Next, we present our implementation in Section IV. Finally, we conclude our article and give an outlook on future work in Section V.

III. Overall design

A. Data quality metrics

We decided to use the widely accepted data quality framework proposed by Kahn et al., because it well reflects our needs for capturing data quality in our integration projects, e.g., in the context of the German Medical Informatics Initiative, which aims at providing cross-site access for researchers to integrate data on a national level [30]. The second reason is to maintain compatibility with other initiatives in the field, e.g., OHDSI [6] or PCORnet [21]. The data quality framework defines the following categories for quality measures [15]:

Conformance: Data quality measures in this category quantify compliance with internal or external formatting, relational, or computational rules. They only assess whether present values meet structural or syntactic constraints, not their completeness or plausibility. This assessment is often done on the basis of a data dictionary or metadata repository, which provides information about the required format and the allowed values for each data element. This category is divided into three subcategories: Value conformance, Relational conformance, and Computational conformance. **Value conformance** measures adherence to the specifications regarding data format, allowed values, data domains, and data types. **Relational conformance** quantifies compliance with structural constraints defined by the database schema like primary and foreign key relationships. **Computational conformance** denotes the accordance between derived values from existing variables and the intended results.

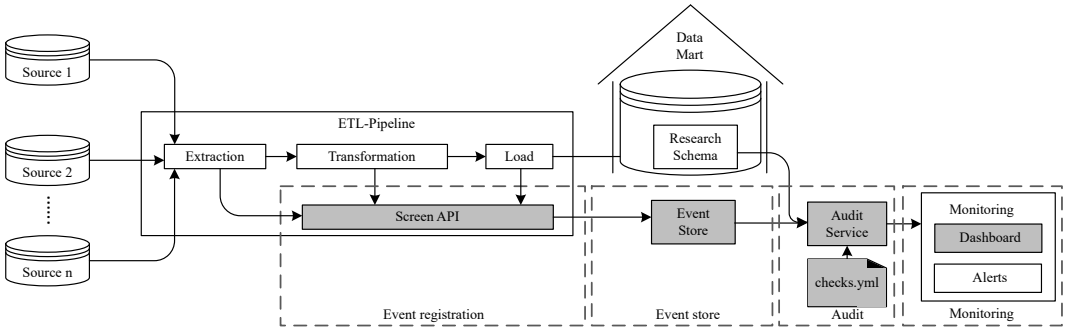


Fig. 1. Overall architecture of our solution for a single data mart in the context of a typical data warehouse architecture. Components marked in gray represent artifacts of our implementation.

Completeness: This category subsumes data quality measures quantifying the frequencies of data attributes without reference to a value. This includes the absence of data at a single moment or at multiple moments in time. The underlying concept from the statistics literature [31] is termed *missingness*. This category does not include measures capturing structure or plausibility aspects. These are considered in the data quality categories Conformance and Plausibility.

Plausibility: Data quality measures in this category capture the truthfulness or believability of values in a dataset. This can be done on the basis of the value itself, within the context of the value of other variables or regarding temporal sequences or state transitions. While the categories Conformance and Completeness measure the compliance with respect to syntactical and structural constraints, Plausibility captures the adherence to *semantic* rules. It comprises the subcategories Uniqueness plausibility, Atemporal plausibility, and Temporal plausibility. **Uniqueness plausibility** quantifies the frequencies of undesired duplications of objects, e.g. different records of person master data referring to the same person. **Atemporal plausibility** measures the adherence of data values to common knowledge or to definitions provided by an external source. **Temporal plausibility** measures the compliance with expectations regarding time-related variables (e.g., patient admission date must be *before* patient discharge date).

B. High-level architecture

Our general architecture is inspired by the approach proposed by Kimball [32], in which data quality is measured during the ETL workflow by applying diagnostic filters on data flows (so-called *quality screens*), which generate error events that are stored in the fact table of a star schema [8]. The fact table is linked to an *audit dimension*, which facilitates analysis of the stored events.

In our approach, data quality auditing is more decoupled from this event store in that, we provide a dedicated auditing sub-system, which assesses the set of events generated and additionally provides methods for *monitoring* data quality issues. The overall architecture of our approach is shown in Fig. 1. It is divided into four components for (a) registration of fine granular data quality issues in the form of events – since we refer to the tests, that generate these events as *screens*, we term it *screen API*, (b) a database based event-store; to avoid disturbances of the operation of the data warehouse system (e.g., due to high-frequency event registrations), it should be decoupled from the warehouse – these components aim at fulfilling our third design goal, (c) the *audit* service, which calculates quality measures based on the linkable event and final warehouse data, and (d) a monitoring component, which comprises a visualization server that provides a configurable user interface for interactively analyzing the data quality metrics – the latter two components aim at fulfilling our first and our second design goal.

C. Error event registration

The ETL component is typically responsible for screening data quality issues in the context of data cleansing [33]. We therefore provide an API, which can be used by the developers of the ETL process to register identified issues in the event store (see the next section).

In contrast to a typical logging API, the registered information is way more structured. Parameters include, e.g., a predefined screen type, which is uniquely assigned to one of the quality categories defined in [15] together with the coordinates of the data point (specification of the data object, identifying attributes, etc.) causing the issue. Furthermore, the API offers a means for registering baseline values (such as the number of input data objects for specified object types), which later can be used for calculating error rates, etc. for the dashboard.

D. Event store

At the core of our architecture is the event store, which is designed to contain detailed information for monitoring, analyzing and correcting quality issues. Following the approach described in [32], it implements a star schema. A simplified version of the star schema of the event store is illustrated in Fig. 2. Each registered quality issue is represented by an (event) fact (table `event_fact`). Dimensions include the

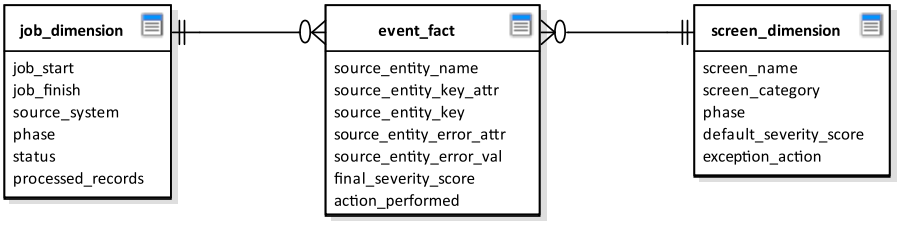


Fig. 2. Illustration of the details of the star schema of the event store. Some entities have been omitted due to space limitations.

screen type (which is assigned to exactly one quality category as defined in [15], table `screen_dimension`), the ETL-job providing the context (table `job_dimension`), baseline information about record counts in the source entity (table `baseline_dimension` – not contained in the figure due to space limitations) and the coordinates of the data point causing the issue (columns `source_entity_name`, etc.). Furthermore, the action performed (e.g., IGNORE, CORRECT) and a severity score is stored in the fact table. Based on the data contained in the event store, detailed reports can be generated that provide information, in which input data source and where in this data source, data quality issues have originated.

E. Audit service

The audit service is responsible for conducting quality audits by integrating and aggregating the data residing in the research schema and the event store. It is also able to provide an integrated view of more general parameters of the data mart, such as the number of contained data items or execution times of the ETL processes.

The audits can be defined by the system operator in the form of SQL queries against the research schema and the event store (`checks.yml`). For each check, different queries for different research schemata (e.g., i2b2, tranSMART, OMOP CDM) can be defined. Depending on the configured target database, the service determines the proper SQL to execute. The result is transformed into a format suitable for the monitoring system.

F. Monitoring system

By leveraging intrinsic functions of modern monitoring systems, our architecture facilitates observing how the different quality dimensions evolve over time and supports monitoring and comparing different data marts in an integrated manner. Further advantages include a flexible alerting mechanism and easily configurable dashboards. These dashboards typically offer web-based, interactive graphical analytics features.

IV. Implementation

As proof of concept, we provide an open-source implementation, which includes (1) the API for capturing data quality events from within the ETL processes, (2) a relational star schema representing the event store, (3) an HTTP based audit service, which contains an initial set of quality check templates, and (4) configuration templates for the monitoring component. As one of our goals is to make the results of our work available to software developers and informatics researchers, all of the components of our reference architecture are based on open-source software, which consequently also applies to our implementation [34].

In this section, we describe the technology stack used for this implementation and the supported target environments as well as the different templates that can be used as a starting point when setting up a data quality monitoring infrastructure. Finally, we describe how we use this implementation in our medical research projects.

A. Base technology stack

We chose to base the screen API and the audit server on the Java ecosystem, as it offers a widely used cross-platform development environment. The event store is based on the PostgreSQL relational database management system. The monitoring component is based (1) on the Prometheus systems monitoring and alerting toolkit, which provides a multidimensional database model, a flexible query language, and an HTTP API and (2) on Grafana, a feature-rich and interactive web-based dashboard software.

To facilitate the adoption of our implementation, we chose Maven as a build system for the Java-based artifacts, which makes it easy to include the API in other projects. The audit server and the monitoring component are prepared for use as Docker containers, including a `docker-compose.yml` with a definition of a self-contained service stack.

B. Supported target environments

Our implementation supports the data models of important data warehouse platforms (i.e., i2b2, tranSMART, and OMOP CDM) and can be used from within the majority of the currently available ETL platforms used in this context (i.e., Talend Open Studio, Pentaho Data Integration, Spring Batch). Current loading tools for these platforms (e.g., transmart-batch, etc.) can be easily enhanced to leverage our architecture.

The monitoring system integrates well with modern IT infrastructures and can, therefore, be used for integrated monitoring of other system parameters (e.g., system load, number of currently logged in users, etc.). Furthermore, it allows for integrated monitoring of different data marts.

C. Provided templates

1) Screen types: As mentioned in Section III-C, developers of the ETL system are responsible for implementing the logic

of the screens to be performed during the execution of the ETL process. The design of the API forces them to assign generated events to one of the predefined screen types. This restriction allows for structured analysis and aggregation of the data quality issues found.

We provide a set of template screen types (e.g., for value range violations or invalid data types), which can easily be extended, and which are mapped to the quality categories defined in [15] for further processing by the audit service.

2) Audit checks: In our implementation, we distinguish between three types of audit checks: (a) generic audit checks, which can be applied independently of the underlying warehouse architecture and concrete data mart instance (e.g., calculating measures capturing the frequency of patients for which observations exist, but no entry in the patient table), (b) system-specific audit checks, which can be applied only within the context of a specific warehouse platform but are still invariant with respect to different instances (e.g., calculating measures capturing the frequencies of encounters without patient information – the former is typically maintained in i2b2, but not in tranSMART), (c) instance-specific audit checks (e.g., calculating measures capturing data quality aspects that are specific to the research question underlying the instantiation of a particular data mart). We provide templates for generic and system- (i.e., i2b2/tranSMART/OMOP CDM) specific audit checks, which can be rolled out together with the docker-compose service stack (e.g., count queries for assessing the number of violations of integrity constraints). The latter are in small parts based on [24] and [35]. Instance-specific audit checks can be created and deployed individually based on the generic and system-specific templates.

3) Dashboard: Our implementation also includes a basic dashboard definition for Grafana. Fig. 3 shows a screenshot of a dashboard, which displays different system parameters of an i2b2 warehouse. The smaller panels defining the upper right angle of the dashboard display general system parameters: the upper three panels display the number of different types of objects stored in the data mart, the panel titled *Num. patients by sex* displays a patient distribution, and the panel titled *ETL job duration* displays the execution time of the last successful ETL process. The panel *Num. data quality issues in final data* provides an overview of *uncategorized* data quality issues found in the data mart. The panel *Num. data quality issues in source data* provides an aggregation of the numbers of data quality issues encountered during the last successful ETL process, grouped with respect to the data quality categories defined in [15]. The bottom panel shows the development of these data over time.

D. Real-world applications

We used the described implementation for monitoring the quality of data marts created for three different medical research projects. These data marts contain up to 6m facts about up to 100k patients, and 300k visits. The data originated from different sources: (1) a research dataset including lifestyle information and microbiome profiles, (2) clinical data about

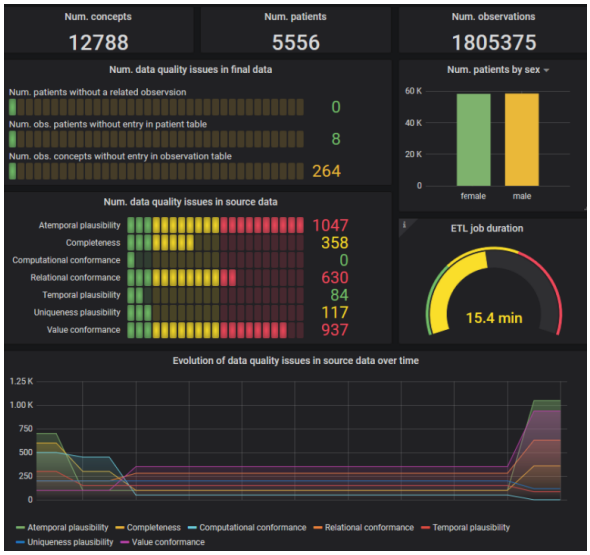


Fig. 3. Basic dashboard template of our implementation monitoring an i2b2 data source.

health care for patients with multiple sclerosis, (3) hospital billing data. The datasets were loaded using an ETL pipeline that supports automated data cleansing [36]. We integrated that pipeline with the quality monitoring component described in this article to analyze the amount of data cleansing performed. The number of data quality issues identified corresponded with our expectations considering the different origins of the datasets. We identified issues for 0.1% of the facts in the billing dataset, 0.05% of the facts in the research dataset, and 6% of the facts in the dataset containing clinical documentation. The most frequent issues identified were missing values and invalid categories. We are currently establishing feedback loops to implement an iterative data quality improvement process.

V. Conclusions and Future Work

We have described an architecture for improving data quality in clinical and translational data warehouse infrastructures. Our approach enables monitoring the evolution of data quality over time using configurable dashboards and alerting mechanisms for important medical research platforms such as i2b2, tranSMART, and warehouses based on the OMOP CDM. Furthermore, this architecture enables integrated monitoring of multiple data marts. Developers of ETL processes are provided with an event logging API that helps to store detailed and structured information on data quality issues for later analysis.

As proof of concept, we provide an implementation, which we use in current medical research projects, and which is available to the public as open-source software [34]. While we are currently working on a detailed evaluation, our experiences to date indicate that our implementation can be used as a starting point for implementing infrastructures for continuous

and iterative data quality improvement in further data-driven medical research projects.

While the event store provides detailed and structured information for generating reports which facilitate root cause analyses of data quality issues, this reporting capability is not yet integrated into the monitoring component. We plan to provide this integration in a future release. Furthermore, we plan to enhance the screen API (1) to be able to interact with a metadata repository for semantic integrity checks and (2) to further assist the ETL developers' workflow regarding identification and tracking of quality issues.

ACKNOWLEDGMENT

We thank the members of the ETL working group of the i2b2 tranSMART foundation for the insightful discussions.

REFERENCES

[1] S. Schneeweiss, "Learning from big health care data," *N Engl J Med*, vol. 370, no. 23, pp. 2161–2163, 2014.

[2] J. C. Prather, D. F. Lobach, L. K. Goodwin, J. W. Hales, M. L. Hage, and W. E. Hammond, "Medical data mining: knowledge discovery in a clinical data warehouse," in *Proc AMIA Annu Fall Symp*. American Medical Informatics Association, 1997, p. 101.

[3] A.-S. Jannot, E. Zapletal, P. Avillach, M.-F. Mamzer, A. Burgun, and P. Degoulet, "The Georges Pompidou University Hospital Clinical Data Warehouse: A 8-years follow-up experience," *Int J Med Inform*, vol. 102, pp. 21–28, 2017.

[4] S. N. Murphy, G. Weber, M. Mendis, V. Gainer, H. C. Chueh, S. Churchill, and I. Kohane, "Serving the enterprise and beyond with informatics for integrating biology and the bedside (i2b2)," *J Am Med Inform Assoc*, vol. 17, no. 2, pp. 124–130, 2010.

[5] E. Scheufele, D. Aronzon, R. Coopersmith, M. T. McDuffie, M. Kapoor, C. A. Uhrich, J. E. Avitabile, J. Liu, D. Housman, and M. B. Palchuk, "tranSMART: an open source knowledge management and high content data analytics platform," in *AMIA Jt Summits Transl Sci Proc*, 2014, pp. 96–101.

[6] G. Hripcsak, J. D. Duke, N. H. Shah, C. G. Reich, V. Huser, M. J. Schuemie, M. A. Suchard, R. W. Park, I. C. K. Wong, P. R. Rijnbeek *et al.*, "Observational health data sciences and informatics (OHDSI): opportunities for observational researchers," *Stud Health Technol Inform*, vol. 216, p. 574, 2015.

[7] A. J. McMurry, S. N. Murphy, D. MacFadden, G. Weber, W. W. Simons, J. Orechia, J. Bickel, N. Wattanasin, C. Gilbert, P. Trevvett *et al.*, "SHRINE: enabling nationally scalable multi-site disease studies," *PloS one*, vol. 8, no. 3, 2013.

[8] W. H. Inmon, *Building the data warehouse*. John Wiley & Sons, 2005.

[9] M. J. Denney, D. M. Long, M. G. Armistead, J. L. Anderson, and B. N. Conway, "Validating the extract, transform, load process used to populate a large clinical research database," *Int J Med Inform*, vol. 94, pp. 271–274, 2016.

[10] C. P. Friedman, A. K. Wong, and D. Blumenthal, "Achieving a nationwide learning health system," *Sci Transl Med*, vol. 2, no. 57, p. 57cm29, 2010.

[11] W. R. Hersh, "Adding value to the electronic health record through secondary use of data for quality assurance, research, and surveillance," *Clin Pharmacol Ther*, vol. 81, pp. 126–128, 2007.

[12] W. R. Hogan and M. M. Wagner, "Accuracy of Data in Computer-based Patient Records," *J Am Med Inform Assoc*, vol. 4, no. 5, pp. 342–355, 1997.

[13] T. Botsis, G. Hartvigsen, F. Chen, and C. Weng, "Secondary use of EHR: data quality issues and informatics opportunities," *Summit Transl Bioinform*, vol. 2010, p. 1, 2010.

[14] N. G. Weiskopf and C. Weng, "Methods and dimensions of electronic health record data quality assessment: enabling reuse for clinical research," *J Am Med Inform Assoc*, vol. 20, no. 1, pp. 144–151, 2013.

[15] M. G. Kahn, T. J. Callahan, J. Barnard, A. E. Bauck, J. Brown, B. N. Davidson, H. Estiri, C. Goerg, E. Holve, S. G. Johnson *et al.*, "A harmonized data quality assessment terminology and framework for the secondary use of electronic health record data," *eGEMs*, vol. 4, no. 1, 2016.

[16] S. Henley-Smith, D. Boyle, and K. Gray, "Improving a secondary use health data warehouse: Proposing a multi-level data quality framework," *eGEMs*, vol. 7, no. 1, 2019.

[17] R. A. Verheij, V. Curcin, B. C. Delaney, and M. M. McGilchrist, "Possible sources of bias in primary care electronic health record data use and reuse," *J Med Internet Res*, vol. 20, no. 5, p. e185, 2018.

[18] M. G. Kahn, M. A. Raebel, J. M. Glanz, K. Riedlinger, and J. F. Steiner, "A pragmatic framework for single-site and multisite data quality assessment in electronic health record-based clinical research," *Med care*, vol. 50, 2012.

[19] M. Nahm, "Data quality in clinical research," in *Clinical research informatics*. Springer, 2012, pp. 175–201.

[20] D. McGilvray, *Executing data quality projects: Ten steps to quality data and trusted information*. Elsevier, 2008.

[21] L. G. Qualls, T. A. Phillips, B. G. Hammill, J. Topping, D. M. Louzao, J. S. Brown, L. H. Curtis, and K. Marsolo, "Evaluating foundational data quality in the national patient-centered clinical research network (PCORnet®)," *eGEMs*, vol. 6, no. 1, 2018.

[22] M. Schuemie, C. Knoll *et al.*, "OHDSI/Achilles," 2020, Available from https://github.com/OHDSI/Achilles. Accessed 7 Mar 2020.

[23] C. Blacketer, F. DeFalco *et al.*, "OHDSI/DataQualityDashboard," 2020, Available from https://github.com/OHDSI/DataQualityDashboard. Accessed 7 Mar 2020.

[24] M. Schuemie, V. Huser, and C. Blacketer, "The Book of OHDSI – Chapter 15 – Data Quality," 2019, Available from http://book.ohdsi.org/. Accessed 9 Mar 2020.

[25] M. Bialke, H. Rau, T. Schwaneberg, R. Walk, T. Bahls, and W. Hoffmann, "mosaicQA – A General Approach to Facilitate Basic Data Quality Assurance for Epidemiological Research," *Methods Inf Med*, vol. 56, no. S 01, pp. e67–e73, 2017.

[26] D. Juárez, E. Schmidt, S. Stahl-Toyota, F. Ückert, and M. Lablans, "A Generic Method and Implementation to Evaluate and Improve Data Quality in Distributed Research Networks," *Methods Inf Med*, vol. 58, no. 2–03, pp. 86–93, 2019.

[27] R. Ball, M. Robb, S. Anderson, and G. Dal Pan, "The FDA's sentinel initiative–a comprehensive approach to medical product surveillance," *Clin Pharmacol Ther*, vol. 99, no. 3, pp. 265–268, 2016.

[28] J. C. Maro, "Medical Product Safety Surveillance: Data Quality in the Sentinel Initiative," 2019, Presentation at the Canadian Society for Pharmaceutical Sciences, Available from https://www.sentinelinitiative.org/communications/publications/medical-product-safety-surveillance-data-quality-sentinel-initiative. Accessed 9 Mar 2020.

[29] L. A. Kapsner, M. O. Kampf, S. A. Seuchter, G. Kamdje-Wabo, T. Gradinger, T. Ganslandt, S. Mate, J. Gruendner, D. Kraska, and H.-U. Prokosch, "Moving towards an EHR data quality framework: The MIRACUM approach." *Stud Health Technol Inform*, vol. 267, pp. 247–253, 2019.

[30] F. Prasser, O. Kohlbacher, U. Mansmann, B. Bauer, and K. A. Kuhn, "Data integration for future medicine (DIFUTURE)," *Methods Inf Med*, vol. 57, no. S 01, pp. e57–e65, 2018.

[31] R. J. Little and D. B. Rubin, *Statistical Analysis with Missing Data, Second Edition*. John Wiley & Sons, 2019, vol. 793.

[32] R. Kimball, "An Architecture for Data Quality," 2007, Available from http://www.kimballgroup.com/wp-content/uploads/2007/10/An-Architecture-for-Data-Quality1.pdf. Accessed 7 Mar 2020.

[33] P. Vassiliadis, "A survey of extract–transform–load technology," *Int J Data Warehousing and Mining*, vol. 5, no. 3, pp. 1–27, 2009.

[34] H. Spengler, "Data Quality Monitor," 2020, Available from https://gitlab.com/DIFUTURE/data-quality-monitor. Accessed 9 Mar 2020.

[35] M. Mendis, "i2b2/i2b2-transmart-etl," 2020, Available from https://github.com/i2b2/i2b2-transmart-etl. Accessed 7 Mar 2020.

[36] H. Spengler, C. Lang, T. Mahapatra, I. Gatz, K. A. Kuhn, and F. Prasser, "Enabling Agile Clinical and Translational Data Warehousing: Platform Development and Evaluation," *JMIR Med Inform*, vol. 8, no. 7, p. e15918, 2020.

A.3 Privacy-Enhancing ETL-Processes for Biomedical Data

International Journal of Medical Informatics 126 (2019) 72–81

Contents lists available at ScienceDirect

International Journal of Medical Informatics

journal homepage: www.elsevier.com/locate/ijmedinf

Privacy-enhancing ETL-processes for biomedical data

Fabian Prasser[1,*], Helmut Spengler[1], Raffael Bild, Johanna Eicher, Klaus A. Kuhn

Institute of Medical Informatics, Statistics and Epidemiology, University Hospital rechts der Isar, Technical University of Munich, Ismaninger Str. 22, 81675 Munich, Germany

ARTICLE INFO

Keywords:
Clinical data warehousing
Extract Transform Load
Privacy
Anonymization

ABSTRACT

Background: Modern data-driven approaches to medical research require patient-level information at comprehensive depth and breadth. To create the required big datasets, information from disparate sources can be integrated into clinical and translational warehouses. This is typically implemented with Extract, Transform, Load (ETL) processes, which access, harmonize and upload data into the analytics platform.
Objective: Privacy-protection needs careful consideration when data is pooled or re-used for secondary purposes, and data anonymization is an important protection mechanism. However, common ETL environments do not support anonymization, and common anonymization tools cannot easily be integrated into ETL workflows. The objective of the work described in this article was to bridge this gap.
Methods: Our main design goals were (1) to base the anonymization process on expert-level risk assessment methodologies, (2) to use transformation methods which preserve both the truthfulness of data and its schematic properties (e.g. data types), (3) to implement a method which is easy to understand and intuitive to configure, and (4) to provide high scalability.
Results: We designed a novel and efficient anonymization process and implemented a plugin for the Pentaho Data Integration (PDI) platform, which enables integrating data anonymization and re-identification risk analyses directly into ETL workflows. By combining different instances into a single ETL process, data can be protected from multiple threats. The plugin supports very large datasets by leveraging the streaming-based processing model of the underlying platform. We present results of an extensive experimental evaluation and discuss successful applications.
Conclusions: Our work shows that expert-level anonymization methodologies can be integrated into ETL workflows. Our implementation is available under a non-restrictive open source license and it overcomes several limitations of other data anonymization tools.

1. Introduction

Modern medical research requires data of comprehensive depth and breadth to improve our understanding of the development and course of diseases and to ultimately develop methods for prevention, targeted diagnosis and therapy. In a learning health system "every clinical encounter contributes to research and research is being applied in real time to clinical care" [1]. To implement this on a large scale, data must be made accessible, harmonized and integrated [2,3]. This also requires using data for secondary applications that go beyond the initial purpose of collection [4,5].

Data integration and in particular data warehouses are central to these efforts. In this context, database systems are set up that integrate disparate data into a common layout which efficiently supports complex analyses. The i2b2 platform [6] is a well-known example of a system that focuses on data generated by clinical and health services and by epidemiological studies [7]. A related platform is tranSMART, which has been developed for the analysis of integrated clinical and 'omics' data for translational research [8]. Some institutions, such as the Vanderbilt University Medical Center [5], have also developed custom solutions.

Data is typically replicated from routine systems into warehouses using ETL processes [9,10]: (1) data is *extracted* from source systems, (2) cleansed, harmonized and *transformed* into a form suitable for analyses, and (3) *loaded* into the analytics solution. To manage the complexity of such processes, they are often implemented using specific environments, which offer libraries of connectors to different types of sources, transformation operators and a graphical workbench for

* Corresponding author.
E-mail address: fabian.prasser@tum.de (F. Prasser).
[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.ijmedinf.2019.03.006
Received 22 June 2018; Received in revised form 6 November 2018; Accepted 6 March 2019

combining them into complex workflows. Well-known solutions are Pentaho Data Integration (PDI, also known as Kettle) [11], which is the standard tool for loading data into tranSMART, and Talend Open Studio (TOS) [12], which is a central component of the Integrated Data Repository Toolkit (IDRT) [13] for creating i2b2-based warehouses.

When pooling medical data or when re-using it for secondary purposes, privacy concerns and legal requirements need careful consideration. Privacy protection involves ethical, legal and societal issues (ELSI) and several layers of technical and non-technical measures are typically required to implement it [14]. On the technical side, the privacy of patients and probands is often protected by data anonymization, which means that datasets are altered in a way that prevents successful re-identification. National and international privacy regulations address data anonymization. In the United States, the *Safe Harbor* method of the Privacy Rule of the Health Insurance Portability and Accountability Act (HIPAA) provides a catalog of attributes for which values need to be removed or modified [15]. In addition, the *Expert Determination* method permits the use of formal and statistical methods for assessing and managing re-identification risks, which is similar to the way in which data anonymization needs to be implemented in the European Union [16].

Data anonymization is a complex process in which the resulting reduction of re-identification risks needs to be balanced against a reduction of data utility [14,17]. A wide variety of different models and methods for data transformation, risk assessment and utility estimation have been proposed to address this trade-off. To manage this complex process, a number of tools have been developed, including sdcMicro [18], which focuses on official statistics, and ARX [19], which has specifically been designed for applications to biomedical data by implementing methods which have been recommended in the field [19–21]. Both tools offer a high level of maturity and they have been included into official guidelines, e.g. from the European Union Agency for Network and Information Security (ENISA) [22] and the European Medicines Agency (EMA) [23].

1.1. Objectives and outline

Performing data anonymization and re-identification risk analyses as part of ETL workflows is a common requirement (see Section 4.1). Typical application scenarios include the loading of data into clinical and translational warehouses, the extraction of data from cross-institutional research registries, and the sharing of data with external research groups. However, ETL platforms such as TOS or PDI do not provide modules which support formal methods of data anonymization and re-identification risk analysis. Although anonymization tools such as sdcMicro or ARX can be used for these purposes, they are based on their own working environments which cannot easily be integrated into ETL platforms (see Section 2).

To bridge this gap, we have developed a plugin for an ETL platform, which supports data anonymization and re-identification risk assessment. The most important design goals were (1) to utilize expert-level risk assessment methodologies, (2) to implement a data transformation method which preserves both the truthfulness of input data and its schematic properties (e.g. data types), (3) to utilize an anonymization process which is easy to understand and intuitive to configure, and (4) to achieve high scalability.

To meet these design goals we had to overcome various challenges. First, we needed to decide on a suitable design and execution environment for ETL processes. Second, we needed to select and integrate methodologies for risk assessment and anonymization which are well-known, flexible and easy to understand. This involved managing the complex interplay of methods for measuring and reducing privacy risks. Finally, we had to develop an efficient implementation.

The remainder of this paper is structured as follows: in Section 2 we describe the methods for risk assessment that we build upon, present a novel anonymization method, and describe how we have implemented and integrated it into an existing ETL platform. In Section 3 we describe how we designed our experiments and present the results. In Section 4 we discuss the principal results, applications in practice, and perform a conceptual comparison with prior work. In Section 5 we conclude and point out directions for future work.

2. Materials and methods

Our method for integrating data anonymization and risk assessment into ETL processes is based on established methods for estimating re-identification risks of medical data, which we present in the first part of this section. In the second part we present a novel anonymization algorithm which we have developed in order to facilitate an effective integration of these methods into ETL platforms. The last part of this section focuses on how we implemented these methods and how we integrated them into a concrete ETL platform.

2.1. Common models for risk assessment

Re-identification is the primary threat addressed by laws and regulations [15,16] and models for quantifying related risks are therefore central to data anonymization and privacy risk management. Re-identification can be understood as a *linkage process* [24]: the uniqueness of (combinations of) attributes is exploited to link records of datasets with additional data or background knowledge of the adversary. Attributes that can be used for establishing a link are termed *quasi-identifiers* [25]. Typical examples include demographic data and other information that is likely to be known to adversaries, such as educational or employment status [21]. Implementing protection requires to consider various factors, e.g. the objectives of likely attackers, the replicability and distinguishability of the data to be protected, and the availability of background knowledge [26,27].

Three different threat scenarios can be distinguished [28]. Under the *prosecutor* model, the adversary is assumed to target a specific individual and to know that data about this individual is contained in the dataset. The risk of a successful attack can be calculated, based on the distinguishability of records in the dataset regarding the quasi-identifiers [26]. It has been shown, however, that this method significantly over-estimates risks in most cases [29]. Under the *journalist* model, the adversary is assumed to target an arbitrary individual without prior knowledge about membership. Often, this background knowledge is much more realistic than in the prosecutor model, as the set of individuals represented in a dataset is just a sample of a larger population. However, the fact that knowledge about the population is typically not available makes it also difficult to reliably determine and manage the risk of successful journalist attacks. Finally, under the *marketer* model, the adversary is assumed to aim at re-identifying as many individuals as possible. Thus the risk of a successful attack can be expressed as the expected average number of re-identified individuals.

El Emam has proposed a methodology that combines estimates of risks under these established models [28]. As journalist risk cannot be quantified in most cases, the methodology makes use of the fact that prosecutor risk is always an upper bound for journalist and marketer risk. Prosecutor risk is quantified for all records and aggregated into three global measures. The first measure is the *Highest Risk* (R_h). It quantifies risks in the worst case scenario, i.e. a prosecutor attack against the record with the highest re-identification risk in the whole dataset. For each record r, the re-identification risk is calculated as $\frac{1}{f_r}$, where f_r is the number of records in the dataset that are indistinguishable from r regarding the quasi-identifiers (including r itself). As noted before, this is also an upper bound for risks in the other scenarios, i.e. for journalist or prosecutor attacks. Even when this risk is bound by a threshold, an attacker can expect to re-identify a certain number of individuals by random linkage to matching records. This is captured by the second measure, *Average Risk* (R_a), which provides a

more tight bound for marketer risks. To account for the fact that the prosecutor model is based on worst-case assumptions, a third measure, called *Records at risk* (R_r) can be used to slightly relax the protection requirements. It expresses the frequency of records that are associated with a re-identification risk higher than a given threshold θ. Formal definitions of these three risk measures are provided in Section A of the supplementary file.

With this methodology and just a single user-specified parameter (θ), three intuitive risk measures can be derived that quantify the susceptibility of data to all types of attacks considered. At the same time, the model facilitates a balancing of privacy protection and the usefulness of data, as it enables the user to permit that a fraction of records has a risk that is higher than the threshold θ. Given a sufficiently small θ and a sufficiently small fraction of records at risk, a high degree of protection can be assumed, as it is very unlikely that the record targeted in a (prosecutor or journalist) attack is one of the records that exceeds the threshold [28]. Thresholds τ_a for the average risk R_a and τ_h for the highest risk R_h can be introduced in addition to θ to specify protection levels which must be satisfied by a data anonymization procedure.

2.2. A novel anonymization method

Automatically altering data such that it meets user-specified risk thresholds is complex and requires integrating risk models with data transformation techniques and methods for measuring data utility. Producing truthful output data implies that input data is not perturbed and that no synthetic data is generated, which is particularly important in medical research where plausibility and correctness are central [30]. Therefore, we decided against transformation schemes which employ noise addition [31] or aggregation of data [32]. Moreover, we wanted to ensure that our method can be integrated into existing ETL workflows without the need to modify intermediate or target data representations. This implies that schematic properties of input data must be preserved, which means that data types must not be altered and that no additional attributes must be introduced into the tables and rows processed. Thus we could not use data generalization [25] or bucketization [33].

Based on these considerations, we decided to implement a cell suppression algorithm. With this model, risk thresholds are enforced by removing individual attribute values from individual records. The method requires zero configuration (apart from specifying risk thresholds), output data is truthful and schematic properties are being preserved. Moreover, the results are well suited for performing common statistical analyses, provided that the effect of cell suppression is considered (e.g. by imputation) [28,30,34].

Fig. 1 shows how cell suppression can be used to protect a dataset from two different threat scenarios. In this simplified example, a clinical dataset is protected from marketer attacks by *external attackers* using the demographic attributes {*Age*, *Sex*, *Region*} and from prosecutor attacks by *internal attackers* using the clinical attributes {*Weight*, *ICD-10*}. Suppressed values (which are denoted by *) are treated as an own category, which means that suppressed values are only considered to be equal to other suppressed values. Under this assumption all sets of rows containing the same quasi-identifying attribute values are pairwise disjoint and form so-called *equivalence classes*. Each equivalence class describes a set of records which are indistinguishable to the attacker and hence its size determines the risk of successful re-identification. In the example, equivalence classes are illustrated by dotted lines. By suppressing 20 of the 50 attribute values in the dataset (40%), the risk of a successful external attack dropped from 60% ($R_a = \frac{6}{10}$) to 30% ($R_a = \frac{3}{10}$) and the risk of a successful internal attack dropped from 100% ($R_h = \frac{1}{1}$) to 33% ($R_h = \frac{1}{3}$). The example also shows that cell suppression is challenging to implement efficiently, as the space of potential solutions for a given dataset consists of $O(2^{n \cdot m})$ transformations where n is the number of records and m is the number of attributes that could be used for linkage. This equates to 2^{50} potential solutions already in our simple example. Thus cell suppression is typically performed using heuristic algorithms.

Our implementation follows this approach by recursively enforcing the user-defined thresholds τ_a and τ_h for subsets of the input dataset. This is implemented with ARX, which is able to compute an optimal solution to a data anonymization problem that is specified as follows [35]: (1) all risk thresholds must be met, (2) each column that contains quasi-identifying values may either be kept as-is or suppressed entirely (attribute suppression), (3) a specified number of records may be entirely suppressed (called the *suppression limit*), (4) the overall number of suppressed cells must be minimal. Our method executes this process recursively for the records that have been suppressed, as is illustrated in Fig. 2. In each iteration, τ_h and τ_a are enforced on a set of the records; the others are suppressed. We use the *k*-anonymity privacy model to enforce τ_h [25] and enforce τ_a by specifying an upper bound on the arithmetic mean of the records' re-identification risks. An additional parameter l_s specifies the maximum number of recursive calls by defining a suppression limit for each iteration. Pseudocode illustrating the anonymization method in more detail and a discussion of implications for data quality is provided in Section B of the supplementary file. While this process is very efficient and effective, as we will show in the next section, it remains necessary to show that it is actually correct. It is easy to see that enforcing an overall threshold on the highest re-identification risk R_h can be performed by enforcing the same threshold on disjoint subsets of records. However, it is not trivial to see that this process can be used to implement a global threshold on the average re-identification risk R_a. A proof is provided in Section C of the supplementary file.

2.3. Implementation and integration

To make our solution accessible to a broad spectrum of users we

Age	Sex	Region	Weight	ICD-10
53	F	North	73	C18.7
68	F	North	73	C18.7
68	M	North	82	C18.7
68	M	North	77	C18.7
71	M	North	73	C18.2
71	M	North	67	C18.2
68	M	South	67	C18.2
68	F	South	67	C18.7
68	F	South	67	C18.7
68	F	South	67	C18.7

(a) Input dataset

Age	Sex	Region	Weight	ICD-10
*	*	North	*	C18.7
*	*	North	*	C18.7
*	*	North	*	C18.7
*	M	North	*	C18.7
*	M	North	*	C18.2
*	M	North	*	C18.2
68	*	South	*	C18.2
68	*	South	67	C18.7
68	*	South	67	C18.7
68	*	South	67	C18.7

(b) Output dataset

Fig. 1. Example dataset before (a) and after (b) it has been transformed using cell suppression. Dotted lines illustrate equivalence classes with respect to two different sets of quasi-identifiers: {*Age*, *Sex*, *Region*} and {*Weight*, *ICD-10*}.

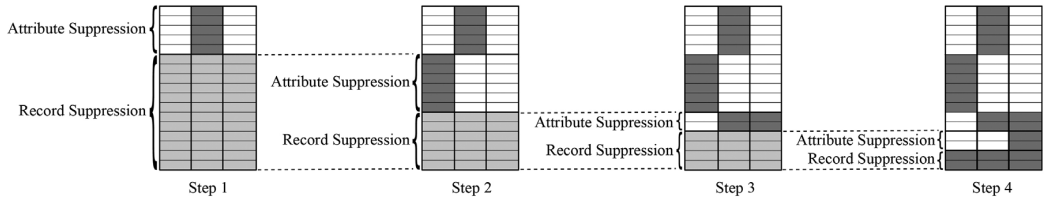


Fig. 2. Illustration of the recursive cell suppression algorithm. In each of the recursion steps the algorithm determines the optimal balance between attribute and record suppression.

decided on a two-step implementation strategy. In the first step, the described anonymization and risk assessment methodology was implemented into ARX. This allowed us to leverage its highly scalable anonymization framework [19] to create a risk assessment and anonymization operator which can then be integrated into ETL environments in the second step. In this context, we decided to develop a plugin for the PDI platform for several reasons. First, we frequently use PDI for loading data into tranSMART. Second, the interface provided by PDI is quite intuitive while the learning curve for TOS can be considered to be somewhat steeper. Third, PDI offers a broad set of features in its community edition (e.g. deployment to clusters) while most advanced features of TOS are only available through a commercial license. Moreover, with the recent release (version 8.0), the programming interfaces of the PDI platform have received significant modernization.

In the PDI workbench, ETL processes can be modeled as directed graphs, where data sources, transformations, and data sinks are represented as nodes called “steps”. Data flow between nodes is represented by edges. Data that could not be processed can be annotated with additional information and routed to a dedicated error output. By combining multiple steps, complex ETL processes integrating heterogeneous sources can be designed, executed and monitored. Fig. 3 shows a screenshot of an ETL process in which data from three different data sources (a CSV file, a relational database, and a HL7 message stream) are joined, validated, transformed, and finally loaded into a target database.

Data processing in PDI is stream-oriented with single rows of data constituting atomic and isolated units of a data stream. This means that data is passed through the ETL pipeline row by row. This enables *pipeline parallelism* across a chain of steps. However, it also implies that plugins that require a holistic view on the overall dataset, such as our plugin for assessing risks or anonymizing data, need to buffer the incoming rows. There are trade-offs involved in implementing this, as the latter breaks pipeline parallelism and high volume datasets can be too large to completely materialize them in main memory.

To solve this issue, we implemented a technique called *row blocking*. This means that our plugin materializes sets of records (i.e. blocks) of a user-defined size, which are then analyzed or anonymized. As soon as each block has been processed, the contained rows are passed on to the next plugin in the workflow. As a consequence, parallelism can be maintained and very large datasets can be processed. In terms of privacy protection, the approach is guaranteed to be correct (see Section C of the supplementary file).

We implemented all methods into a plugin for the PDI platform. Our implementation is available as open source software [36,37] which is compatible with the latest version 8.0 of PDI. The plugin provides methods for re-identification risk analyses and data anonymization. It is compatible with all other functionalities and plugins of PDI.

The tab *Risk thresholds*, which is shown in Fig. 4(a), enables users to specify quasi-identifiers and the thresholds described previously. Compatible to the relational model underlying the ETL environment, values that are suppressed are replaced with *NULL*. Thus the schema and data types of input data are preserved. When risks are assessed and any of them exceeds a user-defined threshold, the incoming data will not be transferred to the subsequent step and, if desired, it can be routed to an error exit. Risk measures are printed to the console for logging purposes. The tab *Runtime settings*, which is shown in Fig. 4(b), can be used to specify parameters affecting the runtime behavior of the anonymization algorithm.

To address multiple threat scenarios, data can be passed through different instances of the plugin configured to address different threat scenarios (cf. example in Fig. 1). This is possible because the plugin preserves the schematic properties of input data and because it makes

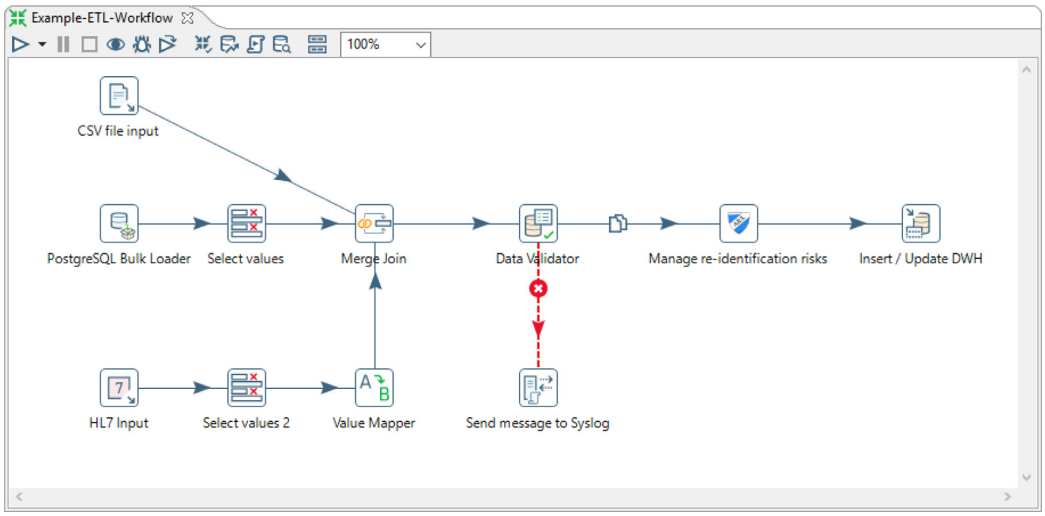
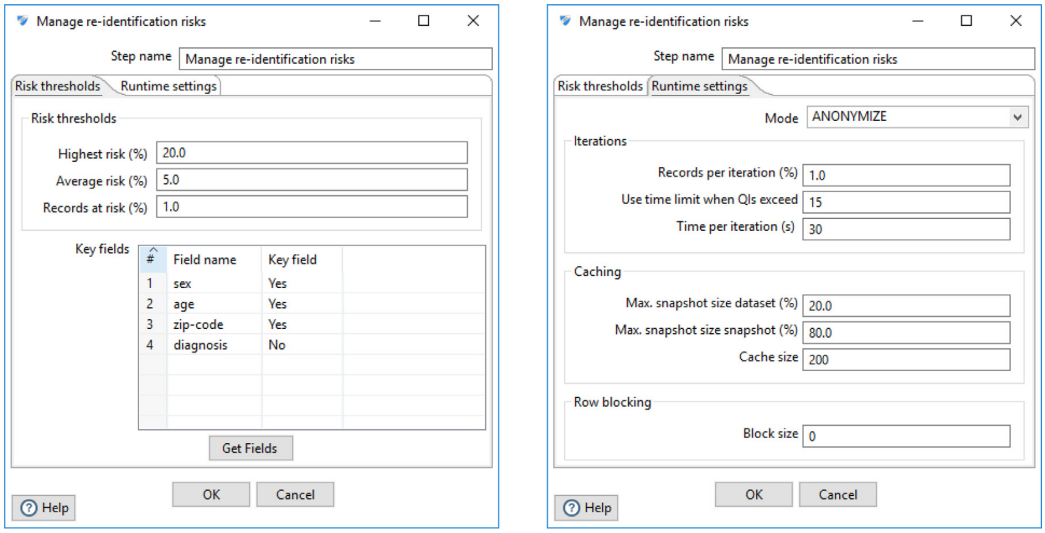


Fig. 3. A typical ETL process in PDI's design environment Spoon.



(a) Privacy related parameters. (b) Parameters determining runtime behavior.

Fig. 4. Screenshots of the plugin's configuration dialogs.

use of different ways of interpreting suppressed values. During anonymization, suppressed values are treated as an own category, meaning that *NULL* only matches *NULL* when calculating the distinguishability of records. However, in a chain of anonymization steps with *overlapping* quasi-identifiers, this can lead to situations, in which one anonymization operation invalidates the privacy guarantees that have been enforced in previous steps because new categories are introduced into quasi-identifying variables addressed previously (an example can be found in Section D of the supplementary file). For this reason, when *assessing* risks, our plugin interprets suppressed values as wild cards. This means that they can match any other (suppressed *or* unmodified) value, which avoids this problem. While it has been shown that this interpretation can provide adversaries with attack vectors under rare circumstances [38], we point out that this is the standard interpretation in the field of statistical disclosure control and also the default in sdcMicro.

3. Results

3.1. Experimental setup

In this section, we present results of evaluating the scalability of our solution as well as the quality of output data, including comparisons with prior work. We point out that a theoretical bound on the data quality provided by our approach cannot easily be obtained (for a discussion of optimality aspects we refer to Section B of the supplementary file). Hence, we focus on an experimental evaluation with real-world datasets to analyze how the method performs in practice. We performed four different sets of experiments:

- **Comparison with prior work**: We first compared the performance of our plugin to sdcMicro (version 5.0.3) [18], which features a cell suppression algorithm that has been implemented in C++ and linked into the software. Next, we studied the utility of output data produced by our cell suppression method in comparison to other data transformation methods using the concept of privacy-preserving data cubes proposed by Kim et al. [39].
- **Comparison using different threat scenarios**: sdcMicro and the work by Kim et al. focus on simple threat scenarios, while our approach supports combinations of several different risk thresholds. We performed additional experiments using various parameterizations and measured output data quality to study their effects.
- **Analysis of risk-utility trade-offs**: In the third set of experiments we constructed risk-utility frontiers, which are plots visualizing the trade-offs that an anonymization method provides between privacy protection and data quality [40].
- **Analysis of the effect of row blocking**: The parameter that specifies the block size has various influences on the quality of output data and the execution time of the anonymization process. In a final set of experiments we studied these effects to determine whether row blocking is an effective mechanism for processing large datasets with our plugin.

We used two datasets, which differ in scope and size and which have already been utilized for evaluating previous work on data anonymization: (1) *US Census*, an excerpt of 30,162 records from the 1994 census database, which serves as the de-facto standard for the evaluation of anonymization algorithms, and (2) *Health Interviews*, a set of 1,193,504 responses to a large health survey. For a detailed description we refer to [41]. For each dataset we selected up to nine quasi-identifiers, consisting of demographic data and further attributes, which are often considered to be associated with a high risk of re-identification [21]. All experiments were performed on a desktop machine equipped with a quad-core 3.2 GHz Intel Core i5 CPU running a 64-bit Windows 7 operating system. The PDI platform (version 8.0) was executed using a 64-bit Oracle JVM (1.8). The number of iterations performed by our algorithm (parameter l_s) was set to 100 in all experiments.

3.2. Experimental comparison with prior work

We first compared our plugin to sdcMicro [18]. The cell suppression algorithm of sdcMicro has been implemented in C++ and linked into the package to improve scalability. The software only supports cell suppression for enforcing a threshold on the highest risk. Hence we set τ_r (records at risk) to zero and used a threshold on the prosecutor re-identification risk (τ_h) of 20%, which is a common parameterization [21].

Fig. 5(a) shows the execution times measured while increasing the number of quasi-identifying attributes. It can be observed that our implementation is significantly more scalable than sdcMicro. While our method was able to easily handle the US Census dataset regardless of

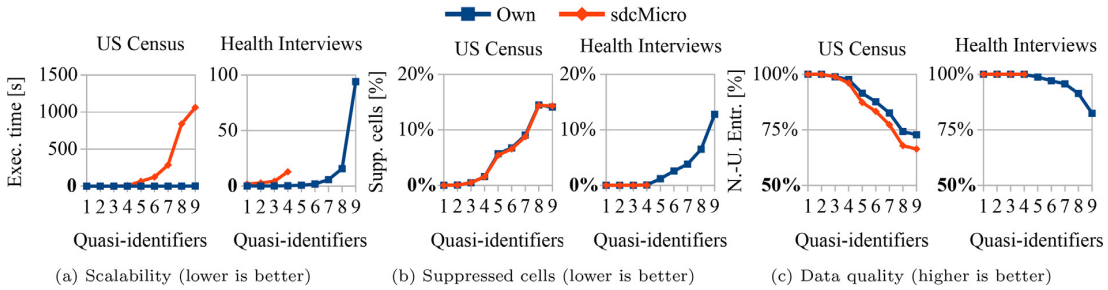


Fig. 5. Comparison of the results obtained with our plugin and the results obtained using sdcMicro. We report average execution times, the number of suppressed cells and data quality quantified with the Non-Uniform Entropy model.

the number of quasi-identifiers selected (≤ 2 s in all configurations), sdcMicro already needed more than 1000 s to process the dataset with nine quasi-identifiers configured. Furthermore, sdcMicro was not able to handle the Health Interviews dataset within 1800 s when more than four quasi-identifiers were specified. For practical reasons, we cancelled all experiments using sdcMicro that did not complete within this time frame. Our plugin generally processed this dataset in not more than 94 s. It can be seen that both implementations were affected by the exponential increase in the size of the solution space with an increasing number of quasi-identifiers [35]. However, our plugin can be configured to use an effective heuristic algorithm when the solution space becomes too large [42].

Regarding data quality, we measured comparable numbers of cells suppressed by our method and by sdcMicro (Fig. 5(b)). Finally, Fig. 5(c) shows how anonymization has impacted the distributions of attribute values. To measure this, we used the Non-Uniform Entropy model [43] which is often used to assess the quality of de-identified data and is based on the concept of mutual information [28]. We normalized the results obtained by this model in such a way that 100% represents the original input dataset while 0% represents a dataset from which all values have been removed. It can be seen that data quality decreased when the number of quasi-identifiers increased, especially for the smaller dataset US Census. It can also be observed that our method had less impact on the distribution of attribute values, implying a more balanced application of value suppression.

Recently, Kim et al. performed an experimental evaluation of the effects of different data anonymization methods when implementing privacy-preserving warehouses for medical data [39]. In their study, data was anonymized and then aggregated into data cubes, which is a model used in warehousing applications. The authors then measured the information loss induced by the anonymization methods and the precision of the results of two types of queries issued against the data cubes: *point queries*, which count the number of records matching a specific combination of attribute values and *range queries*, which count the number of records matching a combination of ranges over the domain of attribute values. They studied two generalization-based approaches and one bucketization algorithm.

We exactly reproduced their experimental setup, which also used the US Census dataset, and compared results obtained using our method with the results presented in [39]. For an exact specification of the algorithms and an in-depth discussion of the results we refer to Section E of the supplementary file. As can be seen in Table 1, our method outperformed both generalization-based approaches in terms of information loss, performed very well on point queries and provided reasonable performance on range queries. At the same time, our method is the only approach considered in the experiments that preserves the schematic properties of input data, and it is much easier to configure than generalization-based algorithms.

3.3. Experimental analysis using different threat scenarios

Our plugin supports thresholds on prosecutor re-identification risk (τ_h) and marketer re-identification risk (τ_a). *Strict-average risk* [21] is a common privacy model combining both risk thresholds. To analyze the improvements in data utility that can be obtained by using this model, we have performed a comparison of both approaches. As a risk threshold, we also used 20%. We used the same threshold once for controlling prosecutor risk and once for controlling marketer re-identification risk but combined the latter with a threshold of 50% on prosecutor risk, which ensures that no record is uniquely identifiable. We note that this comparison focused on our plugin only, as strict-average risk is to our knowledge not supported by any other tool.

We measured no significant differences in execution times when using the two models. We did, however, observe notable improvements in data quality when using strict-average risk. Fig. 6(a) shows the number of suppressed cells when enforcing the thresholds on strict-average risk relative to the number of suppressed cells when enforcing the threshold on prosecutor risk. It can be seen that using strict-average risk resulted in significantly less suppressed cells, especially when configurations with fewer quasi-identifiers were being used. Effects on the distribution of attribute values are presented in Fig. 6(b). In contrast to the effect on the number of suppressed cells, the improvements obtained in terms of Non-Uniform Entropy increased with the number of quasi-identifiers. This implies that data quality can be more effectively increased by using less strict privacy models when it must be assumed that the adversary possess a lot of background knowledge.

3.4. Experimental analysis of the risk-utility trade-off provided

Our plugin provides a broad spectrum of anonymization options, ranging from very strict to very relaxed parameterizations. To analyze these different options in more detail, we constructed risk-utility

Table 1
Comparison of methods for creating privacy-preserving data cubes as proposed by Kim et al. [39].

	Global generalization	Local generalization	Bucketization	Cell suppression
Information loss	0.41	0.13	Not applicable	0.10
Median relative error for point queries (%)	18.3	9.79	0.02	0.00
Median relative error for range queries (%)	10.16	0.81	0.02	41.33

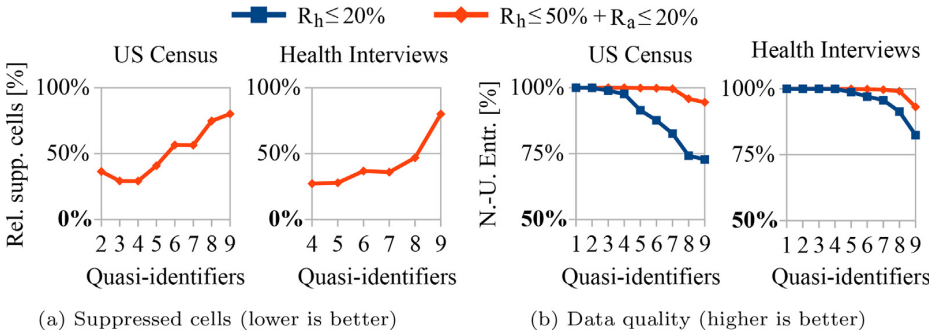


Fig. 6. Comparison of the results obtained when only enforcing a threshold on prosecutor risk with the results obtained enforcing a threshold on strict-average risk. We report the number of suppressed cells relative to the numbers obtained using the prosecutor model for cases in which at least one cell was suppressed. Data quality was quantified using the Non-Uniform Entropy model.

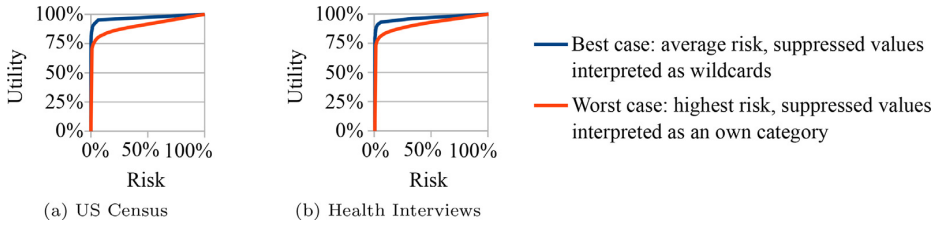


Fig. 7. Risk-utility frontiers for different risk models and different interpretations of missing values.

frontiers, which are plots visualizing the trade-offs that an anonymization method provides between privacy protection and data quality [40]. Each point in these plots represents a transformed dataset offering an optimal privacy/utility trade-off, which means that risk cannot be reduced further without reducing quality and vice versa. Fig. 7 shows the results of our method for both datasets using two extreme configurations addressing all quasi-identifiers. In the best case scenario, thresholds on the average risk R_a have been enforced while interpreting missing values as wild cards. In the worst case scenario, thresholds on the highest risk R_h have been enforced while treating missing values as an own category. Data utility was estimated with the relative number of cells that have *not* been suppressed.

As can be seen, we could not measure any significant differences between the results for the two datasets. In both cases, we observed that high data quality can be maintained at very low risk levels. The frontiers for the best case scenarios were almost optimal. Here, we measured an area under the curve (AUC, 1 optimal, 0 worst) of 0.971 for the US Census dataset and 0.966 for the Health Interviews dataset. In the worst case scenarios we measured AUCs of 0.901 and 0.912, respectively.

3.5. Experimental analysis of the effect of row blocking

Next, we investigated the effect of row blocking on execution times and on output data quality. The experiments were performed with nine quasi-identifying attributes and the same risk models and thresholds as in the previous experiments while varying the *block size*. Previously, we did not use row blocking and were thus able to only report the time needed to anonymize the data. In the results presented here, execution times include the time needed to read the data from disk, anonymize it and persist the results on disk.

As can be seen in Fig. 8(a), execution times decreased with increasing block sizes up to a block size of roughly 10^5, from where on they slowly increased again. This increase can be explained by the fact that much larger data volumes needed to be processed in each anonymization operation. For strict-average risk and block sizes between 10^2 and 10^3, we also observed an increase of execution times. This can be explained by the fact that this setup significantly increased the number of invocations of the underlying anonymization algorithm. Although each invocation had to handle a smaller number of records, the complexity of the anonymization problem with respect to the number of quasi-identifiers remained constant. Moreover, anonymizing fewer records tends to be more computationally expensive, as good solutions are harder to find [35]. Regarding the number of suppressed cells and effects on data quality, when increasing block sizes, we measured a logarithmic decrease (Fig. 8(b)) and increase (Fig. 8(c)), with values converging towards the baselines (dotted) obtained without row blocking. With block sizes of about 10^4 (US Census) and 10^5 (Health interviews) or bigger, the effects of row blocking on output data quality were almost negligible compared to anonymization without row blocking. This indicates that row blocking can be used to effectively balance data quality and execution times when processing large datasets.

4. Discussion

4.1. Principal results and applications in practice

In this article, we have presented a plugin supporting integrated data anonymization and re-identification risk analysis during ETL processes. Our implementation is based on the PDI platform, which is in widespread use within the biomedical field. The methods presented in this paper have also been implemented directly into ARX [19]. The risk assessment methodology described is robust, easy to configure and it provides a good balance between simple but strict approaches such as *k*-anonymity [25] and more flexible but complex models that provide higher degrees of output data quality (e.g. super-population models [44] or game theoretic approaches [45,46]). The proposed transformation method produces truthful datasets which are well suited for performing common statistical analyses [21,28,47,34]. Finally, the software overcomes several limitations of previous data anonymization solutions: data can easily be protected from multiple threats by combining different anonymization operations within a single ETL workflow and very large datasets can be processed by leveraging the streaming-based processing model of the underlying platform. Due to the fact that our approach can be used to process data which has been partitioned into independent subsets (see Section 2.3 and Section C of the supplementary file) it can also be used to incrementally add data to

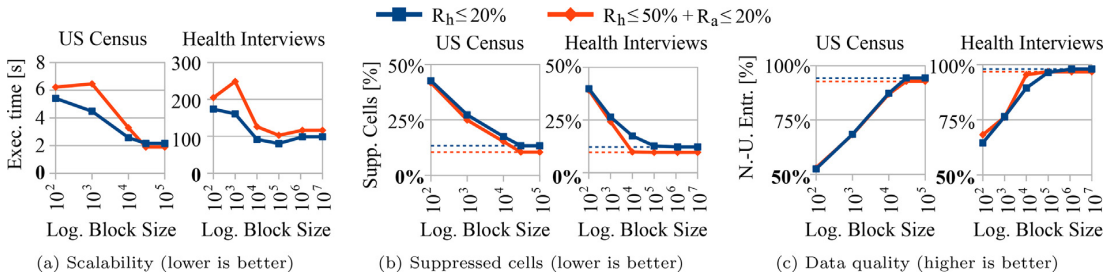


Fig. 8. Semi-log plots visualizing the results of row blocking experiments. We report average execution times, the number of suppressed cells and data quality as reported by the Non-Uniform Entropy model. Dotted lines represent baseline values obtained without row-blocking.

existing databases without violating the privacy guarantees provided.

The software described in this article has already been used in various projects. For example, it was used to anonymize demographic data for a research data warehouse at the Department of Cardiovascular Diseases of the German Heart Centre Munich. The warehouse integrated phenotypic and genotypic data of more than 70,000 patients with coronary artery disease to support data visualization, cohort discovery and hypothesis generation. We have also frequently used the methodology described here when protecting data extracts before sharing them with external partners, for example in the context of research registries for mitochondrial disorders [48] and for neurodegenerative diseases [49]. Finally, the described methods have also been used through ARX by other research groups, for example to create an open dataset for studies of learning behaviour [50] and for anonymizing data from a cancer screening program [51].

4.2. Conceptual comparison with prior work

On the conceptual level, prior work can be found in many areas, including data anonymization, synthetic data generation and data masking. We have already covered related environments for implementing ETL processes and other open source data anonymization solutions in the previous sections. Another software worth mentioning is Privacy Analytics Eclipse [52], which is a commercial data anonymization platform built on Apache Spark [53]. While the software implements formal methods that are quite similar to the ones implemented by our plugin, little has been published about the exact methodology and its implementation.

In the remainder of this section, we focus on further solutions that integrate data protection features into ETL processes. *Data masking* is a technique which has also been integrated into ETL platforms. Methods from this field are not based on formal risk assessment and data anonymization, but they implement simple rule-based transformation processes, e.g. for the removal of data. They are typically used to create data for software development and testing purposes. Examples of relevant implementations include Informatica's *Data Masking* [54], IBM's InfoSphere Optim Data Privacy [55], Oracle's Data Masking and Subsetting Pack [56], ProxySQL [57], and Hush Hush's Data Masking Components [58]. Also, TOS and PDI both offer modules providing basic data masking functionalities.

Synthetic data generation is also supported by the masking solutions presented in the previous paragraph. Most implementations are rather simple, but there are also sophisticated approaches, such as the algorithms supported by sdcMicro [18] which are able to preserve uni- and multivariate statistical properties of input data. Random data generation is also supported by plugins for TOS and PDI. Bijoux is another well-known example [59]. However, data generation plugins for ETL processes are typically too simple to be useful for more than test data generation.

5. Conclusion and future work

In this article, we have described a plugin for a common ETL platform which supports robust anonymization and risk assessment functionalities. The software is available under a non-restrictive open source license. Our method can be integrated into existing ETL workflows, and it supports typical warehousing solutions for biomedical data, such as i2b2 and tranSMART. Even in cases where it is not possible to significantly reduce risks without considerable impacts on data utility, our software can be used to perform quantitative re-identification risk assessments for documenting privacy threats. This is an important aspect of modern privacy laws, such as the European General Data Protection Regulation [16].

The methods and implementations presented in this article are particularly well suited for protecting data that is collected infrequently (e.g. demographics) or which remains rather stable over time (e.g. diagnoses or lab values of particular interest for a specific study) [14,27]. If longitudinal or frequently changing data needs to be protected from linkage attacks, specific measures must be implemented that can cope with higher dimensionality and changes to data [60]. While we plan to extend our software to cover such use cases in future work, we also emphasize that such data often poses much less risk, as it is unstable, difficult to replicate and it is therefore less likely that adequate background knowledge is available to adversaries [14,27]. An additional area of future work is improved support for incrementally adding new data. While this is supported already by the current version of our plugin, we plan to add functionalities for considering data that already exists within the database when measuring and reducing risks during the process of loading new data. This could help to further reduce the amount of suppression needed.

Cell suppression enables the anonymization of datasets with minimal configuration efforts, but further transformation methods can also be useful in certain scenarios. Data generalization and microaggregation are two techniques of specific interest. We plan to add support in future versions of the plugin. However, as these methods may have impacts on the schematic properties of data (e.g. changes in data types and scales of measure) integrating them with the processing environments of ETL solutions is challenging. An alternative anonymization approach to cell suppression is cell swapping (or data swapping) [61] which essentially works by exchanging attribute values between records. Analogously to our work, it preserves the schematic properties of data. In contrast to our approach, data swapping does not remove attribute values and hence it preserves statistical aggregates such as counts of attribute values. However, unlike cell suppression, data swapping is inherently perturbative. Hence, it does not satisfy truthfulness, which is an important requirement in our context (cf. Section 2.2). Moreover, data swapping is typically implemented based on simple risk models, which offer much lower degrees of protection than the methods used in our work. A potential direction for future work would be to investigate how data swapping could be integrated into the proposed anonymization framework, including the strong

protection models used, and to examine potential resulting increases of data utility. One possible approach for this would be to firstly perform anonymization using cell suppression, including risk assessment as described in this article. This step could then be followed by a post-processing step in which the original values of suppressed cells are being swapped and then re-inserted into the output dataset.

While our plugin supports one of the most widely adopted environments for implementing ETL processes, TOS is also frequently used in biomedical data warehousing projects. We have already started to port our plugin to this platform but, due to the differences between development environments and concepts for managing data and control flows, a complete integration will require more work.

Summary points

What was already known on the topic?

- Anonymization is important in biomedical research, especially when data is pooled or re-used for secondary purposes.
- Common ETL (Extract-Transform-Load) tools for integrating data into clinical and translational warehouses do not support anonymization. Moreover, common anonymization tools cannot easily be integrated into ETL workflows.
- Anonymization tools can be difficult to configure and they have scalability issues when processing very large datasets.

What has this study added to the body of knowledge?

- Expert-level anonymization methodologies can be integrated as intuitive plugins into ETL platforms.
- With these plugins, data can be protected from multiple threats within a single ETL workflow.
- Very large datasets can be anonymized efficiently by leveraging the streaming-based processing model of ETL platforms.
- High data utility and compatibility with existing databases and platforms can be achieved by using transformation methods that preserve both the truthfulness of data and its schematic properties.

;1;

Authors' contributions

FP, HS, and RB developed the algorithms. HS and FP designed and implemented the plugins. HS, JE, RB and FP designed, implemented and performed the experiments. RB, HS and FP developed the formal proofs of correctness of the approach. HS, RB, JE, KK and FP discussed the conception and design of the work as well as the manuscript at all stages. All authors have contributed to the manuscript. All authors have read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

The work was, in parts, funded by the German Federal Ministry of Education and Research (BMBF) within the "Medical Informatics Funding Scheme" under reference number 01ZZ1804A (DIFUTURE).

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.ijmedinf.2019.03.006.

References

[1] S.O. Dyke, A.A. Philippakis, J.R. De Argila, D.N. Paltoo, E.S. Luetkemeier, B.M. Knoppers, A.J. Brookes, J.D. Spalding, M. Thompson, M. Roos, et al., Consent codes: upholding standard data use conditions, PLoS Genet. 12 (1) (2016) e1005772, https://doi.org/10.1371/journal.pgen.1005772.
[2] S. Schneeweiss, Learning from big health care data, N. Engl. J. Med. 370 (23) (2014) 2161–2163, https://doi.org/10.1056/NEJMp1401111.
[3] A.J. McMurry, S.N. Murphy, D. MacFadden, G. Weber, W.W. Simons, J. Orechia, J. Bickel, N. Wattanasin, C. Gilbert, P. Trevvett, et al., SHRINE: enabling nationally scalable multi-site disease studies, PLoS One 8 (3) (2013) e55811, https://doi.org/10.1371/journal.pone.0055811.
[4] K. Shameer, M.A. Badgeley, R. Miotto, B.S. Glicksberg, J.W. Morgan, J.T. Dudley, Translational bioinformatics in the era of real-time biomedical, health care and wellness data streams, Brief. Bioinform. 18 (1) (2017) 105–124, https://doi.org/10.1093/bib/bbv118.
[5] I. Danciu, J.D. Cowan, M. Basford, X. Wang, A. Saip, S. Osgood, J. Shirey-Rice, J. Kirby, P.A. Harris, Secondary use of clinical data: the Vanderbilt approach, J. Biomed. Inform. 52 (2014) 28–35, https://doi.org/10.1016/j.jbi.2014.02.003.
[6] A.-S. Jannot, E. Zapletal, P. Avillach, M.-F. Mamzer, A. Burgun, P. Degoulet, The Georges Pompidou University Hospital Clinical Data Warehouse: a 8-years follow-up experience, Int. J. Med. Inform. 102 (2017) 21–28, https://doi.org/10.1016/j.ijmedinf.2017.02.006.
[7] S.N. Murphy, G. Weber, M. Mendis, V. Gainer, H.C. Chueh, S. Churchill, I. Kohane, Serving the enterprise and beyond with informatics for integrating biology and the bedside (i2b2), J. Am. Med. Inform. Assoc. 17 (2) (2010) 124–130, https://doi.org/10.1136/jamia.2009.000893.
[8] E. Scheufele, D. Aronzon, R. Coopersmith, M.T. McDuffie, M. Kapoor, C.A. Uhrich, J.E. Avitabile, J. Liu, D. Housman, M.B. Palchuk, tranSMART: an open source knowledge management and high content data analytics platform, AMIA Jt. Summits Transl. Sci. Proc. (2014) 96–101.
[9] W. Inmon, Building the Data Warehouse, John Wiley & Sons, 2005.
[10] M.J. Denney, D.M. Long, M.G. Armistead, J.L. Anderson, B.N. Conway, Validating the extract, transform, load process used to populate a large clinical research database, Int. J. Med. Inform. 94 (2016) 271–274.
[11] M. Casters, R. Bouman, J. Van Dongen, Pentaho Kettle Solutions: Building Open Source ETL Solutions with Pentaho Data Integration, John Wiley & Sons, 2010.
[12] J. Bowen, Getting Started with Talend Open Studio for Data Integration, Packt Publishing Ltd, 2012.
[13] C. Bauer, T. Ganslandt, B. Baum, J. Christoph, I. Engel, M. Löbe, S. Mate, S. Stäubert, J. Drepper, H.-U. Prokosch, U. Sax, Integrated Data Repository Toolkit (IDRT). A suite of programs to facilitate health analytics on heterogeneous medical data, Methods Inf. Med. 55 (2) (2016) 125–153, https://doi.org/10.3414/ME15-01-0082.
[14] B.A. Malin, D. Karp, R.H. Scheuermann, Technical and policy approaches to balancing patient privacy and data sharing in clinical and translational research, J. Investig. Med. 58 (1) (2010) 11–18, https://doi.org/10.2310/JIM.0b013e3181c9b2ea.
[15] US Department of Health and Human Services, Standards for privacy of individually identifiable health information, Final Rule. 45 CFR, Parts 160–164, Federal Register 67 (157) (2002) 53182–53273.
[16] Regulation (EU) 2016/679 of the Eur. 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), Off. J. Eur. Union (May 2016) L119/59.
[17] F. Kohlmayer, F. Prasser, K.A. Kuhn, The cost of quality: implementing generalization and suppression for anonymizing biomedical data with minimal information loss, J. Biomed. Inform. 58 (2015) 37–48, https://doi.org/10.1016/j.jbi.2015.09.007.
[18] M. Templ, A. Kowarik, B. Meindl, Statistical disclosure control for microdata using the R-package sdcMicro, J. Stat. Softw. 67 (1) (2015) 1–36, https://doi.org/10.18637/jss.v067.i04.
[19] F. Prasser, F. Kohlmayer, Putting statistical disclosure control into practice: the ARX data anonymization tool, Medical Data Privacy Handbook, Springer, 2015, pp. 111–148.
[20] K. El Emam, L. Arbuckle, Anonymizing Health Data: Case Studies and Methods to Get You Started, 1st ed., O'Reilly, 2013.
[21] K. El Emam, B.A. Malin, Appendix B: Concepts and methods for de-identifying clinical trial data, in: Committee on Strategies for Responsible Sharing of Clinical Trial Data, Board on Health Sciences Policy, Institute of Medicine (Eds.), Sharing Clinical Trial Data: Maximizing Benefits, Minimizing Risk, National Academies Press (US), Washington (DC), 2015, pp. 1–290.
[22] European Union Agency for Network and Information Security (ENISA), Privacy and Data Protection by Design – from policy to engineering (2014), 1–79.
[23] European Medicines Agency (EMA), EMA/90915/2016 – External guidance on the implementation of the European Medicines Agency policy on the publication of clinical data for medicinal products for human use (2016), 1–99.
[24] B. Fung, K. Wang, R. Chen, P.S. Yu, Privacy-preserving data publishing: a survey of recent developments, ACM Comput. Surv. (CSUR) 42 (4) (2010) 14.
[25] L. Sweeney, k-anonymity: a model for protecting privacy, Int. J. Uncertainty Fuzziness Knowl.-Based Syst. 10 (05) (2002) 557–570.
[26] F. Prasser, F. Kohlmayer, K.A. Kuhn, et al., The importance of context: risk-based de-identification of biomedical data, Methods Inf. Med. 55 (4) (2016) 347–355, https://doi.org/10.3414/ME16-01-0012.
[27] B. Malin, G. Loukides, K. Benitez, E.W. Clayton, Identifiability in biobanks: models,

measures, and mitigation strategies, Hum. Genet. 130 (3) (2011) 383.
[28] K. El Emam, Guide to the De-Identification of Personal Health Information, CRC Press, 2013.
[29] D.C. Barth-Jones, The 'Re-Identification' of Governor William Weld's Medical Information: a critical re-examination of health data identification risks and privacy protections, then and now, Available from SSRN: http://ssrn.com/abstract=2076397. Accessed 5 January 2018 (2012). doi:10.2139/ssrn.2076397.
[30] F.K. Dankar, K. El Emam, Practicing differential privacy in health care: a review, Trans. Data Privacy 6 (1) (2013) 35–67.
[31] C. Dwork, Differential privacy: a survey of results, International Conference on Theory and Applications of Models of Computation Springer (2008) 1–19.
[32] J. Domingo-Ferrer, J.M. Mateo-Sanz, Practical data-oriented microaggregation for statistical disclosure control, IEEE Trans. Knowl. Data Eng. 14 (1) (2002) 189–201.
[33] X. Xiao, Y. Tao, Anatomy: simple and effective privacy preservation, Proceedings of the 32nd International Conference on Very Large Data Bases, VLDB Endowment, 2006, pp. 139–150.
[34] L. Ohno-Machado, S. Vinterbo, S. Dreiseitl, Effects of data anonymization by cell suppression on descriptive statistics and predictive modeling performance, J. Am. Med. Inform. Assoc. 9 (Supplement_6) (2002) S115–S119.
[35] F. Prasser, F. Kohlmayer, K.A. Kuhn, Efficient and effective pruning strategies for health data de-identification, BMC Med. Inform. Decis. Mak. 16 (1) (2016) 49, https://doi.org/10.1186/s12911-016-0287-2.
[36] arx-deidentifier/arx-pdi-plugins, Plugins for the Pentaho Data Integration platform. Available from https://github.com/arx-deidentifier/arx-pdi-plugins. Accessed 23 March 2018.
[37] arx-deidentifier/cell-suppression-benchmark, Benchmark of cell-suppression methods in ARX. Available from https://github.com/arx-deidentifier/cell-suppression-benchmark. Accessed 23 March 2018.
[38] M. Ciglic, J. Eder, C. Koncilia, k-Anonymity of microdata with NULL values, Int. Conf. Database Exp. Sys. Appl. Springer (2014) 328–342, https://doi.org/10.1007/978-3-319-10073-9_27.
[39] S. Kim, H. Lee, Y.D. Chung, Privacy-preserving data cube for electronic medical records: an experimental evaluation, Int. J. Med. Inform. 97 (2017) 33–42, https://doi.org/10.1016/j.ijmedinf.2016.09.008.
[40] L.H. Cox, A.F. Karr, S.K. Kinney, Risk-utility paradigms for statistical disclosure limitation: how to think, but not how to act, Int. Stat. Rev. 79 (2) (2011) 160–183, https://doi.org/10.1111/j.17515823.2011.00140.x.
[41] F. Prasser, F. Kohlmayer, K.A. Kuhn, Efficient and effective pruning strategies for health data de-identification, BMC Med. Inform. Decis. Mak. 16 (1) (2016) 49, https://doi.org/10.1186/s12911-016-0287-2.
[42] F. Prasser, R. Bild, J. Eicher, H. Spengler, F. Kohlmayer, K.A. Kuhn, Lightning: utility-driven anonymization of high-dimensional data, Trans. Data Privacy 9 (2) (2016) 161–185.
[43] A. De Waal, L. Willenborg, Information loss through global recoding and local suppression, Netherlands Off. Stat. 14 (1999) 17–20.
[44] F.K. Dankar, K. El Emam, A. Neisa, T. Roffey, Estimating the re-identification risk of clinical data sets, BMC Med. Inform. Decis. Mak. 12 (1) (2012) 66, https://doi.org/10.1186/1472-6947-12-66.
[45] Z. Wan, Y. Vorobeychik, W. Xia, E.W. Clayton, M. Kantarcioglu, R. Ganta, R. Heatherly, B.A. Malin, A game theoretic framework for analyzing re-identification risk, PLoS One 10 (3) (2015) e0120592, https://doi.org/10.1371/journal.pone.0120592.
[46] F. Prasser, J. Gaupp, Z. Wan, W. Xia, Y. Vorobeychik, M. Kantarcioglu, K.A. Kuhn, B.A. Malin, An open source tool for game theoretic health data de-identification, AMIA Annu. Symp. Proc. (2017).
[47] K. El Emam, F.K. Dankar, R. Issa, E. Jonker, D. Amyot, E. Cogo, J.-P. Corriveau, M. Walker, S. Chowdhury, R. Vaillancourt, et al., A globally optimal k-anonymity method for the de-identification of health data, J. Am. Med. Inform. Assoc. 16 (5) (2009) 670–682, https://doi.org/10.1197/jamia.M3144.
[48] B. Büchner, C. Gallenmüller, R. Lautenschläger, K. Kuhn, I. Wittig, L. Schöls, D. Rapaport, D. Seelow, P. Freisinger, H. Prokisch, et al., The German Network for Mitochondrial Disorders (mitoNET), Med. Genet. 24 (3) (2012) 193–199, https://doi.org/10.1007/s11825-012-0338-8.
[49] B. Kalman, B. Büchner, F. Kohlmayer, K.A. Kuhn, R. Lautenschlaeger, T. Klopstock, T. Kmiec, An international registry for neurodegeneration with brain iron accumulation, Orphanet J. Rare Dis. 7 (2012) 66, https://doi.org/10.1186/1750-1172-7-66.
[50] J. Kuzilek, M. Hlosta, Z. Zdrahal, Open University Learning Analytics dataset, Sci. Data 4 (2017) 170171, https://doi.org/10.1038/sdata.2017.171.
[51] G. Ursin, S. Sen, J.-M. Mottu, M. Nygård, Protecting privacy in large datasets—first we assess the risk; then we fuzzy the data, Cancer Epidemiol. Biomarkers Prev. 26 (8) (2017) 1219–1224, https://doi.org/10.1158/1055-9965.EPI-17-0172.
[52] Privacy Analytics, Inc., Privacy Analytics Eclipse. Available from https://privacy-analytics.com/software/privacy-analytics-eclipse/. Accessed 5 January 2018.
[53] Apache Spark. Available from https://spark.apache.org/. Accessed 12 January 2018.
[54] Informatica Corporation, Data Masking. Available from https://www.informatica.com/gb/products/data-security/data-masking.html. Accessed 5 January 2018.
[55] IBM Corporation, IBM InfoSphere Optim Data Privacy. Available from https://www.ibm.com/ms-en/marketplace/infosphere-optim-data-privacy/details#product-header-top. Accessed 5 January 2018.
[56] Oracle Corporation, Oracle Data Masking and Subsetting Pack. Available from http://www.oracle.com/technetwork/database/options/data-masking-subsetting/overview/ds-security-dms-2245926.pdf. Accessed 12 January 2018 (2016).
[57] R. Cannaò, ProxySQL. Available from http://proxysql.com. Accessed 5 January 2018 (2018).
[58] Hush, Hush Information Technology and Services, Data Masking Components. Available from http://mask-me.net/. Accessed 5 January 2018 (2017).
[59] V. Theodorou, P. Jovanovic, A. Abelló, E. Nakuçi, Data generator for evaluating ETL process quality, Inform. Sys. 63 (Supplement C) (2017) 80–100, https://doi.org/10.1016/j.is.2016.04.005.
[60] M. Terrovitis, N. Mamoulis, P. Kalnis, Privacy-preserving anonymization of set-valued data, Proceedings of the VLDB Endowment 1 (1) (2008) 115–125.
[61] S.E. Fienberg, J. McIntyre, Data swapping: variations on a theme by Dalenius and Reiss, J. Off. Stat. 21 (2) (2005) 309.

Privacy-Enhancing ETL-Processes for Biomedical Data

Supplementary Material

Fabian Prasser, Helmut Spengler, Raffael Bild, Johanna Eicher, Klaus A. Kuhn

A. Formulas for the risk measures used in this article

El Emam has proposed a methodology that combines estimates of risks according to three risk models: *prosecutor* attacks, *journalist* attacks, and *marketer* attacks [1]. As journalist risk and marketer risk can often not be measured exactly, because exact information about the characteristics of the underlying population would be required, prosecutor risk is measured for all records and compiled into three global measures, two of which provide upper bounds for the risk of successful journalist and marketer attacks [1].

Let n be the number of records of a dataset. The probability for a correct re-identification of each record r_i $(1 \leq i \leq n)$ under the prosecutor model is given by $\frac{1}{f_i}$, where f_i is the number (or frequency) of records in the dataset sharing the same combination of quasi-identifiers as r_i.

The first measure is *Highest Risk (R_h)*. It quantifies risks in the worst case scenario, i.e. a prosecutor attack against the record with the highest re-identification risk in the whole dataset. As noted before, this is also an upper bound for risks in other scenarios, i.e. for journalist or marketer attacks. It is defined as

$$R_h = \max_{1 \leq i \leq n} \left(\frac{1}{f_i} \right) = \frac{1}{\min_{1 \leq i \leq n} (f_i)}. \tag{1}$$

Even when this risk is bound by a threshold, an attacker can expect to re-identify a certain number of individuals by random linkage to matching records. This is captured by the second measure, *Average Risk (R_a)*, which provides a more tight bound for marketer risk. It is given by

$$R_a = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{f_i}. \tag{2}$$

To account for the fact that prosecutor risks are typically significant overestimates of real risks, a third measure, called *Records at risk (R_r)*, expresses the frequency of records that are associated with a re-identification risk higher than a given threshold θ. It is defined as

$$R_r = \frac{1}{n} \sum_{i=1}^{n} I\left(\frac{1}{f_i} > \theta\right). \tag{3}$$

Here, the function $I(\cdot)$ returns 1 if its argument is true and otherwise 0.

B. Pseudocode illustrating the anonymization method

```
1 Dataset suppressAttributesAndRecords(Dataset d, MaxRisks r, Integer l)
2 {
3    Arx arx = new Arx();
4    arx.addPrivacyModel(new KAnonymity(ceil(1 / r.highestRisk)));
5    arx.addPrivacyModel(new AverageClassSize(1 / r.averageRisk));
6    arx.setSuppressionLimit(l);
7    return arx.process(d);
8 }
```

Figure 1: Pseudocode illustrating the method `suppressAttributesAndRecords`.

The core of the proposed anonymization method is a routine which performs attribute and record suppression using ARX as is sketched in Figure 1. The suppression limit `l` is used to enforce that no more than `l` records may be suppressed.

Figure 2 illustrates how the the method `suppressAttributesAndRecords` is being applied to subsets of the input dataset. The pseudocode is formulated iteratively rather than recursively for ease of understanding. In line 8, the method `extractUntransformedSuppressedRecords` returns the original, i.e. untransformed version of all records which have been subject to record suppression in `t`. In line 13, the method `extractNonSuppressedRecords` returns the transformed version of all other records, i.e. the records which have been subject to attribute suppression.

```
1  Dataset suppressCells(Dataset d, MaxRisks r, Integer maxIterations)
2  {
3     Dataset result = emptyDataset();
4     for (int remaining = maxIterations; remaining > 0; remaining--)
5     {
6        Integer l = |d| - |d| / remaining;
7        Dataset t = suppressAttributesAndRecords(d, r, l);
8        d = extractUntransformedSuppressedRecords(t);
9        if (|d| == |t|) { // If all records have been suppressed
10          result = union(result,t);
11          return result;
12       }
13       Dataset a = extractNonSuppressedRecords(t);
14       result = union(result,a);
15    }
16 }
```

Figure 2: Pseudocode illustrating the anonymization method.

The parameter `maxIterations` determines the maximal number of iterations which may be performed. Within each execution of the for loop, the suppression limit used for each invocation of `suppressAttributesAndRecords` is calculated appropriately in line 6 to guarantee that the condition in line 9 is satisfied within at most `maxIterations` iterations. The choice of `maxIterations` balances execution times against data quality.

Internally, ARX selects a transformation from a solution space which is known as generalization lattice [2]. It comprises attribute generalization schemes which replace all values of each attribute with more general, but semantically consistent values. Additionally, a number of records (which is limited by the suppression limit) may be completely suppressed. In the context of the proposed anonymization method, we have configured ARX in such a way that only two generalization strategies are applicable for each attribute: Either all values of the attribute are kept as-is, or they are replaced with a semantic-free placeholder so that the attribute is effectively suppressed. Consequently, in each iteration of the proposed method, a combination of attribute and record suppression is applied to the respective subset of data. This is performed using the algorithm proposed in [2] which selects a transformation which is optimal in the sense that the total number of suppressed cells is minimal. However, we emphasize that this optimality holds only within each of the respective subsets, and only with respect to combinations of attribute and record suppression. After applying different such transformations to different subsets, the

resulting overall dataset will effectively have been transformed via cell suppression in a heuristic manner. Formally arguing about the quality which can be achieved using such heuristics is generally difficult and rarely performed in the literature. Many papers hence provide only empirical but not analytical quality evaluations, and it has been argued that approximation algorithms for which guaranteed approximation factors can be proven usually perform worse in practice than heuristic algorithms for which no such guarantees can be provided [3].

In our experience, the proposed cell suppression algorithm is efficient for a number of quasi-identifiers in the order of up to 15. When the number is higher, execution times can be traded off against data quality by utilizing the heuristic algorithm proposed in [4] rather than the optimal algorithm mentioned above within each subset.

C. Proof of correctness of the cell suppression algorithm

The cell-suppression algorithm presented in this work enforces risk thresholds by removing individual attribute values from individual records. Our implementation solves this problem efficiently by recursively enforcing the user-defined thresholds τ_a and τ_h on R_a and R_h for subsets of the given dataset. The principle of our algorithm is illustrated in Figure 3.

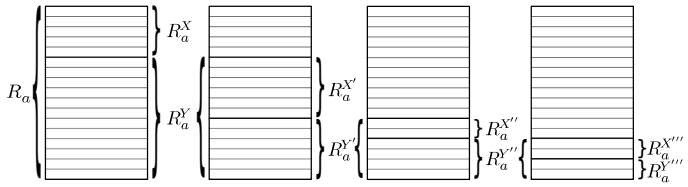


Figure 3: *Subset division principle for efficiently implementing cell suppression.* Each iteration divides the dataset into two disjoint subsets: The subsets $R_a^{X^{(*..)}}$ fulfill the given threshold ($R_a^{1^{(*..)}} \leq \tau_a$); the complementary subsets $R_a^{Y^{(*..)}}$, which don't, are processed recursively in the next step.

It is easy to see that enforcing an overall threshold τ_h on the highest re-identi- fication risk can be performed by enforcing the same threshold on disjoint subsets of records. However, it is not trivial to see that this process can be used to implement a threshold τ_a on the *average* re-identification risk.

Proposition. For any dataset Z with an average re-identification risk R_a^Z, any two disjoint subsets X and Y with $X \cup Y = Z$ and risks R_a^X and R_a^Y, respectively, and any given risk threshold τ_a, the following holds:

$$R_a^X \leq \tau_a \wedge R_a^Y \leq \tau_a \implies R_a^Z \leq \tau_a. \tag{4}$$

Proof. Let X, Y, and Z contain n_X, n_Y, and n_Z records, respectively (as $X \cup Y = Z$ and $X \cap Y = \emptyset$, it follows that $n_X + n_Y = n_Z$). Depending on whether a certain row is referenced from Z or from the subset that it belongs to (either X or Y), its index can have different values. In order to indicate the set that a certain row index i refers to, we term the different row indices r_i^X, r_i^Y, r_i^Z and the related frequency indices f_i^X, f_i^Y, f_i^Z. The indices are related to each other as follows:

$$\begin{aligned} \forall i = 1, \ldots, n_X: &\quad r_i^Z = r_i^X, \\ \forall i = 1, \ldots, n_Y: &\quad r_{n_X+i}^Z = r_i^Y. \end{aligned} \tag{5}$$

Using the definition for the average risk (cf. Formula (2))

$$R_a = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{f_i}, \tag{6}$$

we can express R_a^X, R_a^Y, and R_a^Z as

$$R_a^X = \frac{1}{n_X} \sum_{i=1}^{n_X} \frac{1}{f_i^X}, \quad R_a^Y = \frac{1}{n_Y} \sum_{i=1}^{n_Y} \frac{1}{f_i^Y}, \text{ and } \quad R_a^Z = \frac{1}{n_Z} \sum_{i=1}^{n_Z} \frac{1}{f_i^Z}.$$

Let us assume that

$$R_a^X \leq \tau_a \wedge R_a^Y \leq \tau_a. \tag{7}$$

For better readability, we substitute $\sum_{i=1}^{n_X} \frac{1}{f_i^X} = \Sigma_X$ and $\sum_{i=1}^{n_Y} \frac{1}{f_i^Y} = \Sigma_Y$, which yields

$$\frac{\Sigma_X}{n_X} \leq \tau_a \wedge \frac{\Sigma_Y}{n_Y} \leq \tau_a. \tag{8}$$

Without loss of generality, we further assume that $\frac{\Sigma_Y}{n_Y} \leq \frac{\Sigma_X}{n_X}$ is satisfied. We can conclude:

$$\begin{aligned} \frac{\Sigma_Y}{n_Y} \leq \frac{\Sigma_X}{n_X} &\implies \Sigma_Y \leq \frac{\Sigma_X \cdot n_Y}{n_X} \\ &\implies \Sigma_X + \Sigma_Y \leq \Sigma_X + \frac{\Sigma_X \cdot n_Y}{n_X} \\ &\implies \Sigma_X + \Sigma_Y \leq \frac{\Sigma_X}{n_X}(n_X + n_Y) \\ &\implies \frac{\Sigma_X + \Sigma_Y}{n_X + n_Y} \leq \frac{\Sigma_X}{n_X} \underset{(8)}{\leq} \tau_a. \end{aligned} \tag{9}$$

We now resubstitute Σ_X and Σ_Y, which yields

$$\frac{1}{n_X + n_Y}\left(\sum_{i=1}^{n_X} \frac{1}{f_i^X} + \sum_{i=1}^{n_Y} \frac{1}{f_i^Y}\right) \leq \frac{1}{n_X}\sum_{i=1}^{n_X} \frac{1}{f_i^X} \leq \tau_a. \tag{10}$$

While we have to take into account the possibility that $f_i^Z \neq f_i^X$ and $f_{n_X+i}^Z \neq f_i^Y$ holds, we know that the frequency of a record w.r.t. the whole dataset can never be smaller than its frequency w.r.t. the subset it belongs to. This implies that

$$\begin{aligned} \forall i = 1, \ldots, n_X : \frac{1}{f_i^Z} \leq \frac{1}{f_i^X}, \\ \forall i = 1, \ldots, n_Y : \frac{1}{f_{n_X+i}^Z} \leq \frac{1}{f_i^Y}, \end{aligned} \tag{11}$$

and has the consequence that

$$\frac{1}{n_Z}\sum_{i=1}^{n_Z} \frac{1}{f_i^Z} \leq \frac{1}{n_X + n_Y}\left(\sum_{i=1}^{n_X} \frac{1}{f_i^X} + \sum_{i=1}^{n_Y} \frac{1}{f_i^Y}\right). \tag{12}$$

Finally, we can conclude

$$\frac{1}{n_Z}\sum_{i=1}^{n_Z} \frac{1}{f_i^Z} \underset{(12)}{\leq} \frac{1}{n_X + n_Y}\left(\sum_{i=1}^{n_X} \frac{1}{f_i^X} + \sum_{i=1}^{n_Y} \frac{1}{f_i^Y}\right) \underset{(10)}{\leq} \tau_a.$$

□

This proof also applies to our implementation of row blocking, where privacy guarantees are enforced on subsets (i.e blocks) of the incoming data. As we have shown, privacy guarantees that apply to each of these subsets also apply to the dataset as a whole. Therefore, using row

blocking does not negatively affect the privacy guarantees provided by the implementation.

D. Example for invalidation of privacy guarantees

In a chain of anonymization steps with overlapping quasi-identifiers, treating *NULL* as an own category can lead to situations, in which one anonymization operation invalidates the privacy guarantees that have been enforced in previous steps, because new *NULL*-values are introduced into quasi-identifying variables addressed previously.

Tables 1 (a) and (b) show an example dataset which has been transformed to control the prosecutor risk with respect to the quasi-identifiers *Sex* and *Age*. The dotted lines delimit the equivalence classes concerning different sets of quasi-identifiers. It can be seen that the prosecutor risk with respect to {*Sex*, *Age*} is $\frac{1}{3}$ (see (a)). The prosecutor risk with respect to {*Age*, *Region*} is 1 (see (b)).

Table 1: *Invalidation of privacy guarantees in a chain of anonymization steps*

(a)

Sex	Age	Region
M	*	North
M	*	North
M	*	South
F	67	South
F	67	South
F	67	South

(b)

Sex	Age	Region
M	*	North
M	*	North
M	*	South
F	67	South
F	67	South
F	67	South

(c)

Sex	Age	Region
M	*	North
M	*	North
M	*	South
F	*	South
F	67	South
F	67	South

(d)

Sex	Age	Region
M	*	North
M	*	North
M	*	South
F	*	South
F	67	South
F	67	South

Suppose that in the next step, this dataset is transformed in order to control prosecutor risk with respect to {*Age*, *Region*}. The result of an according transformation is depicted in Tables 1 (c) and (d). It can be seen that by suppressing one more cell, prosecutor risk with respect to {*Age*, *Region*} is reduced to $\frac{1}{2}$ (see (c)). Now, however, the privacy guarantee relating to {*Sex*, *Age*} is violated as prosecutor risk for this set of quasi-identifiers is raised from $\frac{1}{3}$ to 1 (see (d)).

E. Evaluation using privacy-preserving data cubes

Kim et al. recently performed an experimental evaluation of the effects of different data anonymization methods on the querying accuracy provided by privacy-preserving warehouses containing medical data [5]. In their study, data was anonymized using three different transformation methods: global generalization, local generalization (using a clustering algorithm), and bucketization.

Generalization-based algorithms utilize user-provided domain-generalization hierarchies to reduce the fidelity of attribute values. With global generalization, all values of an attribute are transformed to the same level of the associated hierarchy [6]. With local generalization, different values can be generalized to different levels of the according hierarchy [7]. Bucketization works by splitting a table into different disjoint tables and then introducing foreign-key relationships in a way that satisfies the privacy model provided [8]. We emphasize that, in contrast to cell suppression, none of these transformation models is schema-preserving: bucketization is implemented by splitting up the input relation and generalization of values does not preserve the attributes' data types.

After anonymization, Kim et al. aggregated the data into OLAP cubes, which is a model used in warehousing applications. The authors then measured the information loss induced by the anonymization methods using the model by Iyengar [9], which quantifies the degree of generalization of attribute values. A value of 0 represents the unmodified input dataset, while a value of 1 indicates that the dataset has been completely generalized (i.e. all values have been suppressed). Moreover, they measured the precision of the results of two types of queries issued against the data: *point queries*, which count the number of records matching a specific combination of attribute values and *range queries*, which count the number of records matching a combination of ranges over the domains of a subset of the attributes. Results were reported as the median relative error of the counts returned for 1000 (point or range) queries over each possible combination of attributes.

One of the two datasets used by Kim et al., i.e. the US Census dataset, is publicly available. We exactly replicated their setup and used this dataset to compare our method to the methods studied by the authors. The information loss model by Iyengar is not applicable to data which has been transformed using bucketization. Kim et al. measured a value of 0.41 for global generalization and a value of 0.13 for the local generalization method [5]. We measured a value of 0.10 for our cell-suppression algorithm. We employed the following simple probabilistic querying mechanism to execute point and range queries over data with missing values (i.e. values suppressed by our algorithm).

Let $r \in D$ be a record of the dataset D, which has n attributes. r_i describes the value of the record r for the i-th attribute. Let further $Q = \{V_1, ...V_n\}$ be a query against m attributes,

Table 2: *Experimental results obtained by Kim et al. compared with the experimental results obtained for our cell suppression algorithm.*

	Global generalization	Local generalization	Bucketization	Cell suppression
Median relative error for point queries (%)	18.3	9.79	0.02	0.00
Median relative error for range queries (%)	10.16	0.81	0.02	41.33

where each V_i is a set of attribute values for the i-th attribute. When $m < n$, we can simply assume that V_i contains all attribute values when the i-th attribute is not part of the query.

For a given record r the probability of matching a query Q is:
$\Pr[r_1 \text{ matches } Q] \times ... \times \Pr[r_n \text{ matches } Q]$.

For the methods studied by Kim et al. $\Pr[r_i \text{ matches } Q]$ is 1 if $r_i \in V_i$ or any value in V_i is a generalization of r_i. The count returned is increased by 1 when the result of the product is also 1. This is consistent with standard query processing in relational database systems.

When querying data returned by our cell-suppression algorithm we followed the same approach when r_i was not a suppressed value. When r_i was a wildcard, however, we estimated $\Pr[r_i \text{ matches } Q]$ with the sum of the frequencies of all values in V_i in the output dataset. We increased the count by 1 when the result of the product was $\geq \frac{1}{m}$. The results are shown in Table 2.

As can be seen, our approach performed very well on point queries and provided reasonable performance on range queries. At the same time, our method is the only approach considered in the experiments that preserves the schematic properties of the data and it is much easier to configure than the generalization-based approaches, as no hierarchies must be specified. Moreover, existing data warehousing platforms and ETL workflows can be used and no specific privacy-preserving implementation of data cubes has to be developed.

References

[1] K. El Emam, Guide to the De-Identification of Personal Health Information, CRC Press, 2013.

[2] F. Kohlmayer, F. Prasser, C. Eckert, A. Kemper, K. A. Kuhn, Flash: Efficient, Stable and Optimal k-Anonymity, in: Proc Int Conf on Inf Priv Secur Risk Trust, 2012.

[3] J. Goldberger, T. Tassa, Efficient anonymizations with enhanced utility, in: Data Mining Workshops, 2009. ICDMW'09. IEEE International Conference on, IEEE, 2009, pp. 106–113.

[4] F. Prasser, R. Bild, J. Eicher, H. Spengler, F. Kohlmayer, K. A. Kuhn, Lightning: Utility-Driven Anonymization of High-Dimensional Data, Trans Data Priv 9 (2) (2016) 161–185.

[5] S. Kim, H. Lee, Y. D. Chung, Privacy-preserving data cube for electronic medical records: An experimental evaluation, Int J Med Inform 97 (2017) 33–42. doi:10.1016/j.ijmedinf.2016.09.008.

[6] F. Kohlmayer, F. Prasser, C. Eckert, A. Kemper, K. A. Kuhn, Highly efficient optimal k-anonymity for biomedical datasets, in: Computer-Based Medical Systems (CBMS), 2012 25th International Symposium on, IEEE, 2012, pp. 1–6.

[7] J.-W. Byun, A. Kamra, E. Bertino, N. Li, Efficient k-anonymization using clustering techniques, in: International Conference on Database Systems for Advanced Applications, Springer, 2007, pp. 188–200.

[8] X. Xiao, Y. Tao, Anatomy: Simple and effective privacy preservation, in: Proceedings of the 32nd international conference on Very large data bases, VLDB Endowment, 2006, pp. 139–150.

[9] V. S. Iyengar, Transforming data to satisfy privacy constraints, in: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2002, pp. 279–288.

A.4 Protecting Biomedical Data Against Attribute Disclosure

German Medical Data Sciences: Shaping Change – Creative Solutions for Innovative Medicine
R. Röhrig et al. (Eds.)

doi:10.3233/SHTI190829

Protecting Biomedical Data Against Attribute Disclosure

Helmut SPENGLER[a,1] and Fabian PRASSER[a]
[a] *Technical University of Munich, School of Medicine, Institute of Medical Informatics, Statistics and Epidemiology, Munich, Germany*

Abstract. Modern medical research requires access to patient-level data of significant detail and volume. In this context, privacy concerns and legal requirements demand careful consideration. Data anonymization, which means that data is transformed to reduce privacy risks, is an important building block of data protection concepts. However, common methods of data anonymization often fail to protect data against inference of sensitive attribute values (also called attribute disclosure). Measures against such attacks have been developed, but it has been argued that they are of little practical relevance, as they involve significant data transformations which reduce output data utility to an unacceptable degree. In this article, we present an experimental study of the degree of protection and impact on data utility provided by different approaches for protecting biomedical data from attribute disclosure. We quantified the utility and privacy risks of datasets that have been protected using different anonymization methods and parameterizations. We put the results into relation with trivial baseline approaches, visualized them in the form of risk-utility curves and analyzed basic statistical properties of the sensitive attributes (e.g. the skewness of their distribution). Our results confirm that it is difficult to protect data from attribute disclosure, but they also indicate that it can be possible to achieve reasonable degrees of protection when appropriate methods are chosen based on data characteristics. While it is hard to give general recommendations, the approach presented in this article and the tools that we have used can be helpful for deciding how a given dataset can best be protected in a specific usage scenario.

Keywords. data protection, data anonymization, inference attacks

1. Introduction

To be able to develop methods for individualized prevention, diagnosis and therapy, modern medical research requires data of comprehensive breadth and depth [1]. In order to create the required big datasets, patient-level data must be re-used for secondary purposes and shared across institutional boundaries [2,3]. In this context, an important recent project is the German Medical Informatics Initiative (MII) in which four consortia work together on creating a nation-wide infrastructure for data integration and sharing [4–7]. In such projects, data protection aspects and legal requirements, e.g. specified by the European General Data Protection Regulation (GDPR) [8] or the US Health Insurance Portability and Accountability Act

[1] Corresponding Author, Helmut Spengler, Technical University of Munich, School of Medicine, Klinikum rechts der Isar, Institute of Medical Informatics, Statistics and Epidemiology, Grillparzerstr. 18, 81675 Munich, Germany; E-mail: helmut.spengler@tum.de.

(HIPAA) [9], need careful consideration. An important building block of data protection concepts is anonymization, which means that data is transformed to reduce privacy risks. This approach has, for example, been used to perform the first large-scale cross-site analysis within the context of the MII (the so called "demonstrator study", which focused on multimorbidity and rare diseases).

In the area of data anonymization, bridging the gap between legal requirements and technical solutions is challenging and subject of ongoing research [10]. One important issue is that, technically, the risk of re-identification is a continuum, while the legal perspective is dichotomous. In this article, we refer to the threats outlined in the *Opinion 05/2014 on Anonymisation Techniques* by the Article 29 Working Party [11], an independent advisory body on data protection and privacy in the European Union, which has now been replaced by the European Data Protection Board. These threats are: (1) singling out, (2) linkage and (3) inference attacks.

Anonymization methods which reduce the risk of successful linkage or singling out are widely applied in the field, but these methods are often not sufficient to prevent attackers from inferring sensitive personal information (also called *attribute disclosure*). Various models can be used to quantify the risk of attribute disclosure and to transform data to make sure that risks fall below a specified threshold [11]. However, it has been argued that such models are of little practical relevance, because implementing them requires significant data transformations which may remove an unacceptable amount of information [12].

Although the influence of anonymization on data utility has been studied extensively (see e.g. [13–17]), the literature lacks guidance on the strengths and weaknesses of different approaches for protecting biomedical data from attribute disclosure and insights into factors influencing their performance. The objective of the work described in this article was to study indicators and tools that can help to decide when and how biomedical data can be protected from sensitive attribute disclosure without compromising its usefulness too much. We focused on truthful transformation methods, which maintain the plausibility of data, as this is an important requirement in the biomedical domain [18].

2. Methods

2.1. Background

Singling out means that an attacker is able to isolate some or all records which identify an individual in a dataset [11]. This threat is also mentioned as an example of re-identification in the GDPR [8]. *Linkability* denotes the ability to link two (or more) records relating to the same individual or a group of individuals, either within the same dataset or in different datasets. Technically, the attributes that can be used for such attacks are called *quasi-identifiers*. *Attribute inference* (or disclosure) [19] occurs when specific individuals can be associated with attribute values representing sensitive information (e.g. a diagnosis indicating an HIV infection). An attribute which may take sensitive values is called a *sensitive attribute*. Table 1 shows an example dataset in which the risk of singling out and successful linkage has been reduced by generalizing the attribute "Age" and removing the values of the attribute "Sex".

Table 1. Example dataset which has been protected against re-identification only.

Quasi-identifiers			Sensitive attribute
Age	*Sex*	*State*	*Diagnosis*
[60,80[	*	NY	Colon cancer
[60,80[	*	NY	Colon cancer
[60,80[	*	NY	Colon cancer
[60,80[	*	NY	Breast cancer
[20,50[	*	NY	Hodgkin disease
[20,50[	*	NY	Breast cancer
[20,50[	*	NY	Colon cancer

This transformation increases the indistinguishability of the quasi-identifiers while at the same time reducing data utility. Despite the resulting reduction of risks of successful singling out or linkage, the first group of records is susceptible to attribute disclosure: if an adversary targets the record about a 65-year-old male living in the state of New York – which matches the first set of four records with indistinguishable quasi-identifiers – it can be inferred with high probability that the diagnosis is "Colon cancer".

Privacy models can be used to estimate the degree of protection of a dataset against such attacks and they can hence be used to control the risks involved with sharing or re-using data by transforming the data in such a way that a given risk threshold is met. The best-known privacy model is *k*-anonymity [20], which guarantees indistinguishability regarding the quasi-identifiers and hence prevents singling out and reduces the risk of correct linkage. Privacy models for protecting data from attribute disclosure measure risks either based on a quantification of the *diversity* of sensitive data or on a quantification of the *distances* between the distributions of sensitive data in certain groups of records and in the overall dataset. Distance-based models have been developed more recently, with the aim to overcome limitations of diversity-based models, for example with respect to skewed data [21]. Well-known examples are ℓ-diversity [22], which is diversity-based, and *t*-closeness [21] as well as β-likeness [23], which are distance-based.

2.2. Experimental Design

The aim of our work was to gain insights into how well different privacy models for protecting data against attribute disclosure are suited to balance risks against output data utility and what factors need to be considered to achieve an optimal trade-off. To this, we studied the results obtained with different privacy models in relation to trivial baseline solutions. First, we generally protected the data from singling out and linkage using the *k*-anonymity privacy model. Next, we constructed baselines for risk and utility using the following trivial protection measures:

1. Protecting the datasets against attribute disclosure by completely removing all sensitive attribute values (*Full Protection*)

2. Not protecting the dataset from attribute disclosure by keeping the sensitive attribute values as-is (*No Protection*)

Within this context, we compared the trade-off between risks and utility of datasets that have been protected against attribute disclosure using different privacy models and parameterizations.

Technically, the risk of attribute disclosure corresponds to the ability of an attacker to infer values of the sensitive attribute from values of the quasi-identifiers. We modeled this by using logistic regression classifiers that have been trained on anonymized data to predict the values of the sensitive attribute using the quasi-identifiers as features [24]. We quantified the risk by determining the prediction accuracy using 3-fold cross-validation. For measuring the utility of output data, we used a general-purpose model that captures the granularity of output data [25].

All experiments were performed using the open source data anonymization tool ARX, which we configured to use local generalization [26], which is a truthful transformation method. Protection against singling out and linkage has been implemented using *5*-anonymity [20]. For additionally protecting the datasets against attribute disclosure, we used the following privacy models: distinct-ℓ-diversity [22], t-closeness using the earth-movers distance based on value generalization hierarchies [21] and enhanced β-likeness [23]. For each model, we selected a representative set of parameterizations covering the complete spectrum of reasonable values. The source code of our experiments is available online [27].

Table 2. Statistical properties of the sensitive attributes in the evaluation datasets.

Dataset	Sensitive attribute	Domain size	Min. frequency	Max. frequency	Dispersion index
Census	Marital status	5	0.014	0.446	0.777
Health interviews	Marital status	10	$1.3 \cdot 10^{-5}$	0.236	0.899
Census	Education	25	0.010	0.176	0.943
Health interviews	Education	26	0.001	0.192	0.952

To be able to study influencing factors with respect to data characteristics, we selected datasets with a similar schema but different statistical properties. The *Health interviews*[2] dataset consisted of 1,193,645 records from the U.S. National Health Interview Survey (NHIS). The *Census*[3] dataset contained 68,725 responses to the American Community Survey (ACS) from randomly selected people living in the state of Massachusetts in the U.S. We selected *Sex*, *Age* and *Race* as quasi-identifiers, since demographic parameters are typically considered to be associated with a high risk of re-identification [28] and to provide comparability with prior studies [16,20,25]. As examples for sensitive attributes we selected *Marital status* as well as *Education* and protected them from inference attacks.

Table 2 provides an overview of the statistical properties of the sensitive attributes in the evaluation datasets. We used the *dispersion index* [29], as an indicator for the

[2] https://nhis.ipums.org/nhis/
[3] http://www.census.gov/programs-surveys/acs/data/pums.html

skewness of the distribution of attribute values (*skewness of the sensitive data*, henceforth). A value of 1 represents minimum skewness with a uniform distribution of two or more distinct attribute values. A value of 0 represents maximum skewness where there is no distribution of values but only *one* uniform value.

3. Results

To visualize the results of our experiments, we computed risk-utility curves that summarize the results obtained for the different privacy models and parameterizations into one diagram per dataset and sensitive attribute. Each data point represents an anonymized output dataset for a specific model and parameter value. The x- and y-coordinates of the data points represent the according risk and utility values, which have been normalized in relation to the baseline approaches *"Full Protection"* (lowest risk, lowest utility) and "*No Protection"* (highest risk, highest utility). Table 3 contains an overview of the values obtained for the baseline approaches.

Table 3. Baseline values for the evaluation datasets.

Dataset	Sensitive attribute	*Full Protection*		*No Protection*	
		Risk	Utility	Risk	Utility
Census	Marital status	0.0	0.7481	0.7495	0.9981
Health interviews	Marital status	0.0	0.7498	0.6108	0.9998
Census	Education	0.0	0.7481	0.3609	0.9981
Health interviews	Education	0.0	0.7498	0.3954	0.9998

The risk-utility curves are shown in Figure 1. The results have been sorted such that the experiment with the highest skewness of sensitive data (*Census / Marital status*) is on the left and the experiment with the lowest skewness of sensitive data (*Health interviews /Education*) is on the right.

A perfect risk utility curve would contain a data point with zero risk and 100% utility. The line between (0,0) and (1,1) represents all solutions where reductions in risk are directly proportional to reductions in utility. The further a solution is above this line, the higher the increase in protection relative to the reduction in utility. Solutions below the line can be considered suboptimal, as they represent trade-offs that are worse than those provided by the baseline approaches. However, it must be noted that these solutions are not necessarily worthless, particularly in scenarios where neither of the two baseline approaches can be used.

As can be seen, the privacy models studied in this article enable a balancing of risks and utility within the spectrum of removing all sensitive data (0,0) and not protecting a dataset at all (1,1). However, when protecting datasets with skewed sensitive attributes (*Census / Marital status* and *Health interviews / Marital status*) no solutions could be found that provide a better risk-utility trade-off than the baseline approaches.

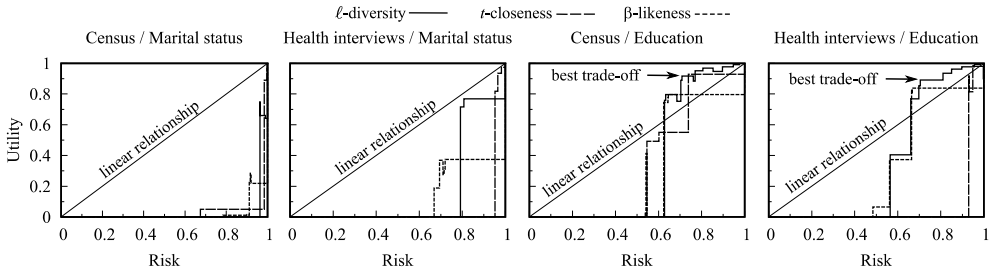


Figure 1. Risk-utility curves for the privacy models ℓ-diversity, t-closeness, and β-likeness. The solution with the best risk-utility trade-off above the linear relationship is marked with an arrow.

For datasets with low skewness of the sensitive data (*Census / Education* and *Health interviews / Education*), particularly when using ℓ-diversity and β-likeness, there were quite a few solutions that provided significant reductions in risk without affecting data utility too much. Using more sophisticated distance-based models did not lead to better results compared to outputs produced using the very basic ℓ-diversity model, which simply restricts the number of distinct sensitive attribute values per group of indistinguishable records [22]. This was true even for data sets with highly skewed sensitive data. In both experiments with the *Health interviews* datasets, t-closeness provided the worst trade-offs of all models.

4. Discussion

Our work is not the first to study the risk-utility trade-off provided by methods for protecting data against attribute disclosure. Notably, Brickell and Shmatikov thoroughly studied the effects of implementing common privacy models on the performance of statistical classification models trained on output data [16]. However, as Li and Li have pointed out, the methodology used in [16] is unsound [30]. Moreover, when studying the effect of methods for protecting data from attribute disclosure, statistical classification is typically used for measuring data utility while we used it to estimate the risk of sensitive attribute inference. Furthermore, we also considered the properties of the distribution of sensitive attribute values. Finally, we have put a specific emphasis on truthful data transformation methods which have been recommended for biomedical data [31,32].

Our results indicate that it is indeed hard to achieve a reasonable trade-off for skewed data, even with distance-based models that have been developed specifically for this purpose [21,23]. When data is only moderately skewed, both β-likeness and ℓ-diversity can yield significantly better risk-utility trade-offs than the baseline approaches. Interestingly, using ℓ-diversity, which is the simplest model with the most intuitive semantics, often provided better results than more recent and more sophisticated models like β-likeness.

In summary we conclude that – contrary to popular opinion – it can be possible to significantly reduce the risk of successful inference attacks using well-known and truthful anonymization methods. However, it is important to consider the specifics of the dataset that is to be protected and the context of data usage. For this purpose, an in-

depth analysis must be performed, and the approach used in this paper can serve as a blueprint. To calculate the necessary statistics and analyze risk-utility trade-offs, common anonymization tools such as ARX [26] and sdcMicro [33] can be used. The source code of our analysis is available online [27]. However, in future work it should be considered to extend the tools mentioned with additional methods to more directly support the processes described (e.g. through the graphical user interface). Our results indicate that particular attention should be directed towards simple methods with intuitive semantics, such as ℓ-diversity, and weak parameterizations, such as $\ell \leq 5$.

One of the main limitations of the present work is the use of specific methods for quantifying risks and data utility. In fact, different measurement methods may be relevant in different data use scenarios. This is particularly true for methods that quantify data utility.

References

[1] T.B. Murdoch, and A.S. Detsky, The Inevitable Application of Big Data to Health Care, *J. Am. Med. Assoc.* **309** (2013) 1351. doi:10.1001/jama.2013.393.

[2] J. Christoph, L. Griebel, I. Leb, I. Engel, F. Köpcke, D. Toddenroth, H.-U. Prokosch, J. Laufer, K. Marquardt, and M. Sedlmayr, Secure Secondary Use of Clinical Data with Cloud-based NLP Services, *Methods Inf. Med.* **54** (2015) 276–282. doi:10.3414/ME13-01-0133.

[3] S. Schneeweiss, Learning from Big Health Care Data, *N Engl J Med*. **370** (2014) 2161–2163. doi:10.1056/NEJMp1401111.

[4] F. Prasser, O. Kohlbacher, U. Mansmann, B. Bauer, and K. Kuhn, Data Integration for Future Medicine (DIFUTURE), *Methods Inf. Med.* **57** (2018) e57–e65. doi:10.3414/ME17-02-0022.

[5] B. Haarbrandt, B. Schreiweis, S. Rey, U. Sax, S. Scheithauer, O. Rienhoff, P. Knaup-Gregori, U. Bavendiek, C. Dieterich, B. Brors, and others, HiGHmed – An Open Platform Approach to Enhance Care and Research across Institutional Boundaries, *Methods Inf. Med.* **57** (2018) e66–e81. doi:10.3414/ME18-02-0002.

[6] H.-U. Prokosch, T. Acker, J. Bernarding, H. Binder, M. Boeker, M. Boerries, P. Daumke, T. Ganslandt, J. Hesser, G. Höning, and others, MIRACUM: Medical Informatics in Research and Care in University Medicine, *Methods Inf. Med.* **57** (2018) e82–e91. doi:10.3414/ME17-02-0025.

[7] A. Winter, S. Stäubert, D. Ammon, S. Aiche, O. Beyan, V. Bischoff, P. Daumke, S. Decker, G. Funkat, J.E. Gewehr, and others, Smart Medical Information Technology for Healthcare (SMITH), *Methods Inf. Med.* **57** (2018) e92–e105. doi:10.3414/ME18-02-0004.

[8] The European Parliament and the Council of the European Union. Regulation 2016/679 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation), *Off J Eur Commun*. **L 119** (2016) 1–88.

[9] U.S. Department of Health and Human Services, Office for Civil Rights. HIPAA administrative simplification regulation, *45 CFR Parts 160, 162, 164*. (2013).

[10] M. Karg, Anonymität, Pseudonyme und Personenbezug revisited?, *Datenschutz Und Datensicherheit - DuD*. (2015). doi:10.1007/s11623-015-0463-z.

[11] Article 29 Data Protection Working Party, Opinion 05/2014 on Anonymisation Techniques, (2014). http://ec.europa.eu/justice/data-protection/article-29/documentation/opinion-recommendation/files/2014/wp216_en.pdf (accessed May 13, 2018).

[12] K. El Emam, and C. Álvarez, A critical appraisal of the Article 29 Working Party Opinion 05/2014 on data anonymization techniques, *Int. Data Priv. Law*. **5** (2015) 73–87. doi:10.1093/idpl/ipu033.

[13] K. El Emam, E. Jonker, L. Arbuckle, and B. Malin, A systematic review of re-identification attacks on health data, *PLoS One*. **6** (2011).

[14] A. Gionis, and T. Tassa, Using k-anonymization with minimal loss of information, *IEEE Trans. Knowl. Data Eng.* **21** (2009) 206–219. doi:10.1109/TKDE.2008.129.

[15] B.C.M. Fung, K. Wang, A.W.-C. Fu, and P.S. Yu, Introduction to Privacy-Preserving Data Publishing: Concepts and Techniques, 2010.

[16] J. Brickell, and V. Shmatikov, The cost of privacy: destruction of data-mining utility in anonymized data publishing, in: 14th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., 2008: pp. 70–78. doi:http://doi.acm.org/10.1145/1401890.1401904.

[17] F. Prasser, F. Kohlmayer, and K.A. Kuhn, Efficient and effective pruning strategies for health data de-

identification, *BMC Med. Inform. Decis. Mak.* (2016). doi:10.1186/s12911-016-0287-2.
[18] H. Lee, S. Kim, J.W. Kim, and Y.D. Chung, Utility-preserving anonymization for health data publishing, *BMC Med. Inform. Decis. Mak.* **17** (2017) 104.
[19] D. Lambert, Measures of disclosure risk and harm, *J. Off. Stat.* **9** (1993) 313–331.
[20] L. Sweeney, Achieving k-anonymity privacy protection using generalization and suppression, *Int. J. Uncertainty, Fuzziness Knowledge-Based Syst.* **10** (2002) 571–588. doi:10.1142/S021848850200165X.
[21] N. Li, T. Li, and S. Venkatasubramanian, t-Closeness: Privacy beyond k-anonymity and l-diversity, in: 23rd IEEE Int. Conf. Data Eng., 2007: pp. 106–115. doi:10.1109/ICDE.2007.367856.
[22] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, l-diversity: Privacy beyond k-anonymity, *ACM Trans. Knowl. Discov. Data*. **1** (2007) 3. doi:10.1145/1217299.1217302.
[23] J. Cao, and P. Karras, Publishing Microdata with a Robust Privacy Guarantee, *Proc. VLDB Endow.* **5** (2012) 1388–1399. doi:10.14778/2350229.2350255.
[24] F. Prasser, J. Eicher, R. Bild, H. Spengler, and K.A. Kuhn, A tool for optimizing de-identified health data for use in statistical classification, in: 2017 IEEE 30th Int. Symp. Comput. Med. Syst., 2017: pp. 169–174.
[25] V.S. Iyengar, Transforming data to satisfy privacy constraints, in: 8th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., 2002: pp. 279–288. doi:10.1145/775047.775089.
[26] F. Prasser, F. Kohlmayer, R. Lautenschläger, and K.A. Kuhn, ARX - A Comprehensive Tool for Anonymizing Biomedical Data, in: AMIA Annu. Symp. Proc., 2014: pp. 984–993.
[27] arx-deidentifier/attribute-disclosure-benchmark, (2019). https://github.com/arx-deidentifier/attribute-disclosure-benchmark (accessed February 25, 2019).
[28] L. Sweeney, Simple demographics often identify people uniquely, *Carnegie Mellon Univ. Data Priv. Work. Pap. 3*. **671** (2000) 1–34.
[29] W.D. Schafer, Assessment of dispersion in categorical data, *Educ. Psychol. Meas.* **40** (1980) 879–883.
[30] T. Li, and N. Li, On the tradeoff between privacy and utility in data publishing, in: 15th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., 2009: pp. 517–526. doi:10.1145/1557019.1557079.
[31] K. El Emam, and L. Arbuckle, Anonymizing health data: case studies and methods to get you started, O'Reilly, 2013.
[32] K. El Emam, and B. Malin, Appendix B: Concepts and methods for de-identifying clinical trial data, in: Comm. Strateg. Responsible Shar. Clin. Trial Data; Board Heal. Sci. Policy; Inst. Med. Ed. Shar. Clin. Trial Data Maximizing Benefits, Minimizing Risk, 2015: pp. 1–290.
[33] M. Templ, A. Kowarik, and B. Meindl, Statistical disclosure control for microdata using the R-package sdcMicro, *Trans. Data Priv.* **1** (2008) 67–85. doi:10.18637/jss.v067.i04.