University of Stuttgart - Institute for Parallel and Distributed Systems / AS

A Zone Reference Model for Enterprise-Grade Data Lake Management

Corinna Giebler, Christoph Gröger, Eva Hoos, Holger Schwarz, Bernhard Mitschang

In: Proceedings of the 24th IEEE Enterprise Computing Conference (EDOC 2020)

BIBTEX:

@inproceedings{zoneReferenceModel_EDOC_2020,

author = {Giebler, Corinna and Gr{\"{o}}ger, Christoph and Hoos, Eva and Schwarz, Holger and Mitschang, Bernhard},

booktitle = {Proceedings of the 24th IEEE Enterprise Computing Conference (EDOC 2020)},

title = {{A Zone Reference Model for Enterprise-Grade Data Lake Management}},

year = {2020},

doi = {https://doi.org/10.1109/EDOC49727.2020.00017}

}

A Zone Reference Model for Enterprise-Grade Data Lake Management

Corinna Giebler
University of Stuttgart
Stuttgart, Germany
Corinna.Giebler@ipvs.uni-suttgart.de

Christoph Gröger
Robert Bosch GmbH
Stuttgart, Germany
Christoph.Groeger@de.bosch.com

Eva Hoos
Robert Bosch GmbH
Stuttgart, Germany
Eva.Hoos@de.bosch.com

Holger Schwarz
University of Stuttgart
Stuttgart, Germany
Holger.Schwarz@ipvs.uni-stuttgart.de

Bernhard Mitschang
University of Stuttgart
Stuttgart, Germany
Bernhard.Mitschang@ipvs.uni-stuttgart.de

***Abstract*— Data lakes are on the rise as data platforms for any kind of analytics, from data exploration to machine learning. They achieve the required flexibility by storing heterogeneous data in their raw format, and by avoiding the need for pre-defined use cases. However, storing only raw data is inefficient, as for many applications, the same data processing has to be applied repeatedly. To foster the reuse of processing steps, literature proposes to store data in different degrees of processing in addition to their raw format. To this end, data lakes are typically structured in zones. There exists various zone models, but they are varied, vague, and no assessments are given. It is unclear which of these zone models is applicable in a practical data lake implementation in enterprises. In this work, we assess existing zone models using requirements derived from multiple representative data analytics use cases of a real-world industry case. We identify the shortcomings of existing work and develop a zone reference model for enterprise-grade data lake management in a detailed manner. We assess the reference model's applicability through a prototypical implementation for a real-world enterprise data lake use case. This assessment shows that the zone reference model meets the requirements relevant in practice and is ready for industry use.**

Keywords*— *Data Lake, Zones, Reference Model, Industry Case, Industry Experience

I. Introduction

In recent years, data lakes gained popularity as they not only allow reporting but also flexible and advanced analytics on heterogeneous data for both batch and real-time processing [1]. Work on data lake management, i.e., the management of data within a data lake, is however premature and inconsistent [2]. In particular, practical experience shows that the initial idea of deferring any kind of data transformation and data processing until data are retrieved for analysis (as seen e.g., in [3]) is not viable. Especially when data are reused for at least similar purposes multiple times, starting with raw data and performing the same processing steps each time is inefficient [4].

A solution to this problem is to store not only raw, but also pre-processed data in the data lake [4]. To manage these diversely processed data, literature frequently suggests *zone models* (e.g., in [4–6]). These zone models define in which processing degrees (e.g., raw, cleansed, aggregated) data are available in the data lake, and how they are governed (e.g., regarding access rights, data quality, and responsibilities). For different use cases, data in the most fitting processing degree can then be accessed. Zone models thus allow to share and reuse pre-processed data between use cases. Zones are similar to the layers in data warehousing (e.g., in [7]), but data may not move through all zones or even move back.

Literature describing these zone models is varied, vague, and inconsistent. There neither exists a uniform concept for zone-based data lake management, nor any form of systematic assessment of available concepts. When building data lakes in practice, this diversity poses a challenge, as it remains unclear which zone model to use and how to implement it.

We address this problem in this work. As a basis, we use the following data lake definition based on [3]: the data lake serves as a data management platform for all kinds of analytics, from reporting and OLAP (Online Analytical Processing) to advanced analytics. Data of any format can be stored and used for any analytical use case without the need to define all of the data's future use upon ingestion. To achieve this flexibility, data are stored in their raw format. Various user groups can access and make use of these data in their everyday work life.

Based on industry experience with an enterprise-wide data lake, we assess existing zone models and develop a zone reference model for enterprise-grade data lake management. The term "enterprise-grade" means that the model can support use cases typical for enterprises. To this end, we make following contributions:

- We investigate representative real-world data analytics use cases for data lakes from multiple business domains and derive a set of requirements from practice.
- We use these requirements to assess existing data lake management concepts, in particular data ponds [8] and zone models [4–6, 9–11].
- We introduce a meta-model for zones that defines a zone's attributes and interactions within and outside of the zone model.
- We develop a zone reference model that addresses the identified requirements as an instantiation of the meta-model. This zone reference model provides guidance for the realization of zone-based data lake management.
- We assess this reference model in two ways: 1) we provide a prototypical implementation for an additional data analytics use case not covered during the requirement analysis to assess its adaptability, 2) we evaluate its suitability with regard to the derived requirements.

The remainder of this paper is structured as follows: Section II gives an overview of the underlying industry case

TABLE I. REQUIREMENTS OF THE INVESTIGATED DATA ANALYTICS USE CASES

Requirement	Finance	Quality Management	Manufacturing	End Customer Services
Pre-Processed	✓	✓	✓	✓
Cleansed	✓	X	✓	✓
Integrated	✓	✓	✓	✓
Governed	✓	✓	✓	✓
Reporting and OLAP	✓	✓	✓	✓
Advanced Analytics	X	✓	✓	✓
Writing back	X	✓	✓	✓

and derives a set of requirements from typical use cases. Section III presents related work in data lake management and assesses existing zone models using the derived requirements. Section IV introduces the meta- model for zones, which serves as a basis to develop and detail the zone reference model in Section V. Section VI assesses the developed reference model. Section VII concludes the paper.

II. USE CASES AND REQUIREMENTS DERIVATION

We use multiple real-world use cases from a global manufacturer as a basis for our assessment of existing zone models and the development of the zone reference model. According to our experience, the observations made here also apply to other large enterprises. The business of the considered manufacturer is very diverse ranging from mass production to individual production. To enhance its business and increase competitiveness [12], the manufacturer implements methodologies from industry 4.0 [13] by integrating data analytics in the entire industrial value chain. Various data analytics use cases from varied contexts manage their data in an enterprise-wide data lake.

The investigated use cases originate from four different but frequently represented business domains, namely finance, quality management, manufacturing, and end costumer services. The use cases cover a wide variety of analytics (traditional reporting to advanced analytics using machine learning) and data (structured to unstructured), and thus can be considered representative for data lake applications. We have already used three of these use cases as a representative basis for previous work [14] , where we evaluated the Data Vault modeling technique for the usage in data lakes. In the following, we examine these use cases and their requirements.

The finance analytics use case aims at realizing *reporting and OLAP* [15] on the data lake. Structured batch data from multiple different sources have to be *integrated* with each other, e.g., from multiple ERP (Enterprise Resource Planning) systems. As the results of financial analyses are of high relevance to the enterprise, the data used should be *cleansed* and carefully *governed*. Certain analyses like the calculation of KPIs (Key Performance Indicators), such as the operating cash flow, are executed regularly and would thus benefit from *pre-processed* data.

The quality management analytics use case uses data from a wide variety of sources to investigate quality defects in manufactured products. To this end, root-cause analyses are executed using defect reports, together with other *advanced analytics*. Additionally, batch data are used for traditional *reporting and OLAP*, e.g., measuring the number of defects for a certain product line. Since data are acquired from different source systems, they have to be *integrated*. The quality of data is assured in the source systems, which makes *cleansing* in the data lake redundant. Some of the data used in this use case are personal, e.g., customer data from defect reports. Thus, they need to be *governed* accordingly, since legal regulations, e.g., GDPR, apply to personal data. Again, this use case comprises analyses that are executed regularly and thus benefit from *pre-processed* data. An additional requirement is that data scientists should be able to share the results of their analyses with other users by *writing* them *back* into the data lake. These results, such as transformations or data mining models, then can be reused for other use cases.

The goal of the manufacturing analytics use case is to gain insights into the manufacturing process of car parts, where various parts from different suppliers are used in assembly. Batch data on the supplied parts, and real-time sensor data from machines and measuring stations are used to support various analyses, from *reporting and OLAP* on manufacturing data (see for example [16]), to *advanced analytics* [17], e.g., machine learning on structured and unstructured data [18]. A large number of source systems (over 600) are involved in this use case. Thus, the data available have to be *integrated*. Some data are captured manually by workers, e.g., defect descriptions, while others are sensitive, e.g., workers' personal data. Thus, *cleansing* and *governing* data are of high importance. So is *pre-processing* a subset of the data, as certain use cases are executed periodically (e.g., KPI calculation). Data scientists should be able to *write* results *back* in the data lake for future use.

The End Customer Services analytics use case is part of the mobility sector. Real-time field data, e.g., GPS data, are collected at the customer's site and used to offer services to the customer via an app (e.g., route planning services or dashboards). Additionally, analyses are executed on batch data to improve the product. As data are collected about certain customers, thorough data analytics can affect the customers' privacy. Thus, data need to be *governed* and protected. To improve the product, data scientists analyze the data in an *advanced* manner (e.g., using machine learning). *Reporting and OLAP* are also performed. Results acquired by the data scientists should be available for further use in the data lake and thus be *written back*. Again, this use case involves various data sources that have to be *integrated* with each other. In addition, various analyses, such as the creation of dashboards, are executed periodically on the data and thus benefit from *pre-cleansed* and *pre-processed* data.

From investigating these representative data analytics use cases, we derive seven feature requirements (in addition to storing raw data) a data lake management has to meet.

- Data should be available in a *pre-processed* state, i.e., they are no longer in their raw format, but, e.g., aggregated or filtered to support several use cases.
- Data should be available in a *cleansed* format, meaning that syntactical errors are erased.
- Data from different sources should be available in an *integrated* format, where they are consolidated and connected.
- Sensitive and critical data should be *governed* accordingly.
- *Reporting and OLAP* [15] for, e.g., KPI calculation, should be supported.
- *Advanced analytics* [17] using, e.g., machine learning should be supported.
- To make results available to other users of the data lake, *writing back* data into the data lake should be supported.

Table I depicts all these requirements and how the different data analytics use cases contribute to them. We use these general requirements to assess existing data lake management models in Section III and to development a zone reference model in Section V.

III. Related Work

Literature provides different concepts for data management in data lakes. This section describes the existing concepts (Section A) and assesses available zone models using the requirements from Section II (Section B).

A. Overview of Existing Data Lake Management Concepts

Two major approaches exist to manage pre-processed data in data lakes [19]: The *data pond architecture* and the *zone architecture*. The following paragraphs give a short overview over the basics of these approaches.

The *data pond architecture* [8] by Inmon separates the data lake into five disjoint ponds: Raw Data Pond, Analog Data Pond, Application Data Pond, Textual Data Pond, and Archival Data Pond. Data are always only available in one of the mentioned ponds at any given time, and they are processed as they make their way through the ponds. Thus, the original data are lost. This contradicts the concept of data lakes [19] and prevents advanced analytics. Regarding the other requirements in Section II, the data pond architecture does separate data into disjoint data ponds, which prevents comprehensive integration. Governance is not discussed in the data pond architecture and there is no possibility to write data back into the data lake. Due to these restrictions, the data pond architecture is not considered in the remainder of this work.

There exists a multitude of different variants of the *zone architecture* [4–6, 9–11]. These variants differ greatly not only in the zones they include (see Fig. 1), but also in the number of zones, the supported user groups (data scientists only vs. data scientists and business users alike), and their focus (processing [4] vs. governance [9]). The fundamental idea, however, remains the same: Different zones contain data in different degrees of processing, for example raw or processed. To this end, each zone defines certain characteristics data in it must have (e.g., data are cleansed and in a common format). In contrast to the pond architecture, zones are not disjoint. Data might be copied from zone to zone, or a zone might contain views on data from a different zone. That way, raw data always remain available in what is mostly called the "Raw Zone". The characteristics of this zone remain the same across all zone models: data are stored permanently in their raw format. Regarding the number and characteristics of other zones, however, existing concepts are varied and inconsistent.

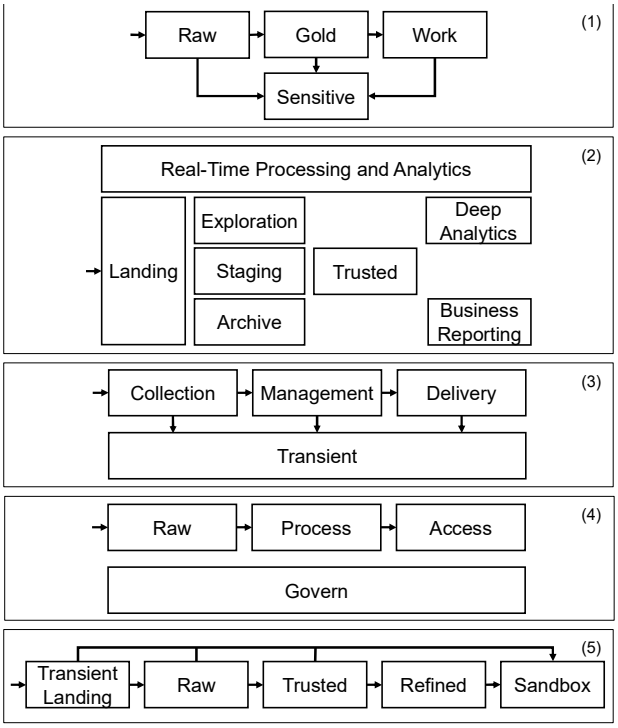

Fig. 1. Overview of the five different zone models: (1) Gorelik [9], (2) IBM [11], (3) Madsen [4], (4) Ravat [10], and (5) Zaloni [5, 6].

B. Assessment of Existing Zone Models

As part of our research, we have discovered five major zone models (i.e., variants of the zone architecture): Gorelik [9], IBM [11], Madsen [4], Ravat [10], and Zaloni, whose model exists in multiple versions by different authors [5, 6]. Fig. 1 shows an overview of all five zone models in a systematic format to visualize their key characteristics.

In this section, we assess these zone models with respect to the feature requirements derived in Section II. We also investigate two methodological requirements: 1) the level of detail in which the zone model is described, including implementation details, and 2) whether it provides a derivation methodology (i.e., a description of the process by which the models were created) and an assessment. Table II shows the assessment results, i.e., to what extent the zone models fulfill the requirements. A bracketed checkmark indicates that while the zone model provides some of the requested characteristics, we do not consider the requirement as sufficiently met. The following paragraphs detail on our assessment.

In Gorelik's zone model [9] (Fig. 1 (1)), the Gold Zone allows to manage *pre-processed* data. However, there is no zone for explicitly *cleansed* or *integrated* data. The Sensitive Zone allows to *govern* especially sensitive data and ensures

TABLE II. ASSESSMENT OF THE EXISTING ZONE MODELS WITH RESPECTIVE TO THE REQUIREMENTS FROM SECTION II. ✓– REQUIREMENT MET, X – REQUIREMENT NOT MET, (✓) – REQUIREMENT INSUFFICIENTLY MET.

	Requirement	Gorelik [9]	IBM [11]	Madsen [4]	Ravat [10]	Zaloni [5, 6]
Feature	Pre-Processed	✓	✓	✓	✓	✓
	Cleansed	X	✓	✓	X	✓
	Integrated	X	✓	✓	X	✓
	Governed	✓	(✓)	X	(✓)	✓
	Reporting and OLAP	✓	✓	✓	✓	X
	Advanced Analytics	✓	✓	✓	✓	✓
	Writing back	✓	X	X	X	X
Method-ological	Description Detail	(✓)	X	X	X	(✓)
	Derivation and Assessment	X	X	X	X	X

that legal regulations are complied with. *Reporting and OLAP* can be done using the Gold Zone. The Work Zone provides data for *advanced analytics* for data scientists. Results from the Work Zone can be *written back* into the Gold Zone for further use. While for each zone, a *detailed description* is given, the model is lacking implementation details. Thus, we only consider this requirement partially met. There exists no *derivation methodology* or *assessment* of the model.

IBM's zone model [11] (Fig.1 (2)) provides zones for *pre-processed*, *cleansed*, and *integrated* data . However, there is little information on *governed* data. Especially legal regulations are not addressed, which is why this requirement is not fully met. The zone model provides both places for *reporting and OLAP*, and *advanced analytics* with the Exploration Zone and the Business Reporting Zone. However, the model provides no possibility for *writing back* into the data lake. The zone descriptions in this model mix logical and physical aspects and do not explicitly state the interactions between zones. We thus rate the *description detail* as insufficient. Again, no *derivation methodology* and *assessment* are provided.

In Madsen's zone model [4] (Fig.1 (3)), the Management Zone provides storage for *pre-processed*, *cleansed*, and *integrated* data. A place for *governed* data is not part of the model. Both *reporting and OLAP*, and *advanced analytics* are supported by two different zones. Again, there is no possibility of *writing* results *back*. The zones are only briefly described with low *description detail*. There is neither a *derivation methodology* nor an *assessment*.

Ravat's zone model [10] (Fig. 1 (4)) only provides a place for *pre-processed* data. Neither *cleansed* nor *integrated* data are part of the model. However, it is empathized that all data is *governed.* Since compliance with legal regulations is not at all discussed in the model, this requirement is only partially met. The model provides both *reporting and OLAP*, and *advanced analytics* in the Access Zone. *Writing back* data and results is not part of the model. Only very little *description detail* is provided, and there is no *derivation methodology* or *assessment*. Thus, both requirements are unmet.

Finally, Zaloni's zone model [5, 6] (Fig.1 (5)) provides a possibility to manage *pre-processed*, *cleansed*, *integrated*, and *governed* data. It even includes legal regulations in the management of data by allowing masking and anonymization. While there is a zone for *advanced analytics*, there is none for *reporting and OLAP*. In this model, *writing back* data is available only for systems, not for human users. We thus consider this requirement not met. While a *detailed description* is given in [6], it lacks implementation details. There is no *derivation methodology* or *assessment*.

This assessment shows that while all feature requirements are addressed by at least one existing zone model, none of the related work could meet all requirements. In particular, the available description details these zone models are insufficient for a practical implementation. In many cases, the zones are only described vaguely. Only IBM's zone model provides any hints for its implementation, but they are heavily mixed with the conceptual model. It remains unclear how the zones can be realized, what standardized data format should be used, etc. In addition, none of the investigated models were derived in a systematic way, and no assessment or discussion of related work is given. In the following sections, we thus develop and describe a zone reference model systematically, based on the requirements derived in Section II.

IV. META-MODEL FOR ZONES

When investigating different zone models it became apparent that even across models, the general concept of a zone remains the same. Each zone can be described by a limited set of attributes (e.g., processing degree of the contained data) and its interactions with other zones and the outside world (e.g., where it imports data from). In this section, we develop a meta-model for zones from both literature and the industry case. This meta-model enables a standardized and generic, and thus comparable, description for zones. It is the basis of a systematic approach to defining a zone reference model in Section V as an instance of this meta-model.

Fig. 2 depicts the meta-model for zones as an entity-relationship-diagram. Zones are modeled as entities. We introduce the meta-model for zones in two parts: 1) the attributes of a zone (left in Fig. 2) and 2) a zone's interaction within and outside of the zone model (right in Fig. 2). Each zone is identified by a unique *name* within the zone model.

A zone defines multiple *data characteristics* that data in this zone have. Each characteristic refers to one of four aspects: *granularity*, *schema*, *syntax*, and *semantic*. *Granularity* describes whether data are raw or aggregated,

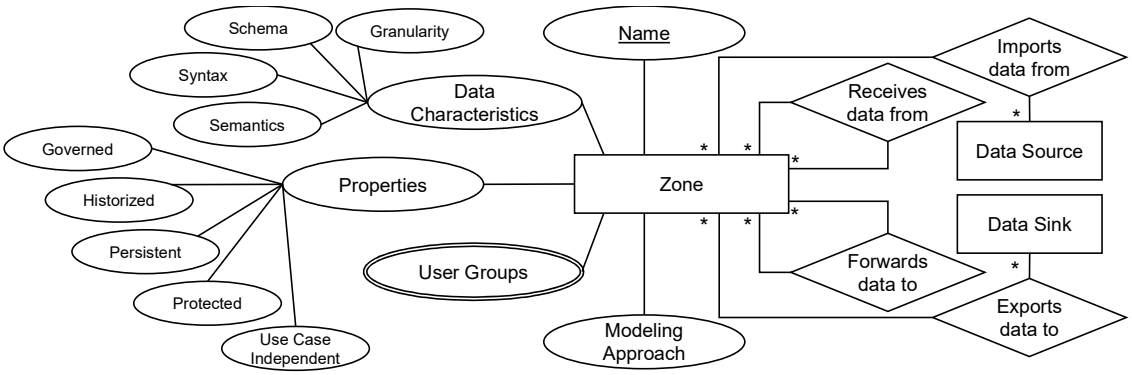

Fig. 2. The meta-model for zones describes a zone as an ER diagram. The left side of the meta-model contains the attributes of a zone. The right side describes how a zone interacts with other zones and the outside world.

e.g., through KPI calculation. *Schema* refers to the data's structure, which might change through adding new fields or relationships. *Syntax* refers to whether the data are changed syntactically, e.g., through data type conversions. *Semantic* refers to the data's meaning, which can be changed by, e.g., removing semantical flaws, such as outliers.

In addition to its data characteristics, each zone has *properties* that describe the zone's nature. From literature (e.g.,), we identified *governed*, *historized*, *persistent*, *protected*, and *use case independent* as important properties for zones. Governed refers to whether the zone is managed by IT and thus has to comply with corporate rules. Historized means that changes in the source data are traceable in the respective zone. Persistent describes whether data in this zone are stored for a prolonged time in contrast to temporary storage. Protected describes whether data in a zone are protected beyond the default, e.g., through encryption or stricter access controls. Use case independent refers to whether data in a zone were processed according to a specific use case or use case groups, or can be used flexibly. Madsen's zone model [4] also names *immutable* as a zone property. However, practice shows that storing all data immutably, and not archiving or deleting them, leads to storage and management issues. Additionally, some legal regulations, e.g., the GDPR, explicitly demand data mutability. Thus, we decided to not include this property.

One or more *user groups* interact with the zone. Possible are human users (data scientist, domain expert, business user, see [19]) and non-human users (systems, processes). Each zone has at least one user group, namely the processes that enter data into the zone.

Finally, each zone has a *modeling approach* associated with it, i.e., a description of how to achieve a specific schema, such as dimensional tables. Note that we also consider "no pre-defined schema" a schema. Even within a data lake, data modeling should not be neglected, as this may lead to issues with data quality, data comprehensibility, or data integration [20]. Examples for possible modeling approaches are copying source system formats, flat files, or Data Vault [7], which is suitable for data lakes [14].

Zones have interactions with both other zones and systems that are beyond the data lake. A zone *receives data from* zero or multiple other zones, and can *forward data to* zero or multiple zones. This allows data transfer between zones. Similarly, a zone can *import data from* and *export data to* external data sources and data sinks not part of the zone model, e.g., operational systems, data streams, or file systems.

This meta-model for zones provides a systematic and generic description of the concept of zones. Using this description, we can develop a zone reference model.

V. Zone Reference Model

The meta-model for zones developed in Section IV allows a wide variety of possible instantiations. However, not all of the zone models that are valid according to the meta-model are also reasonable. Our investigation of related work showed that certain zones and data characteristics reappear, such as zones for cleansed or integrated data. We thus combine the frequently encountered concepts from literature with the general requirements derived in Section II to develop a zone reference model that meets the requirements posed in practice. By systematically defining zones and their characteristics, the zone reference model provides guidance and a scope towards the implementation of zone-based data lake management. Within this scope, the zone reference model can be adapted to the specific needs of the application scenario, e.g., by omitting certain zones that are not needed in a specific implementation.

Our reference model can be applied to both batch and real-time processing for data of any structure. For batch processing, the zones store data in different processing degrees. For real-time processing, zones define processing steps for the passing data stream.

In the following subsections, we detail on the different zones in the developed zone reference model. While the Landing Zone and the Raw Zone are named in accordance with existing literature (e.g., [6]), the remaining zones received new names. We use the meta-model as a basis to describe each of the zones. In these descriptions, we provide insights for the zone implementation using the End Customer Service analytics use case. Table III summarizes the attributes of the zones. A description of a prototypical implementation can be found in Section VI. Fig. 3 depicts the zone reference model and the interactions between the zones. The model consists of a use case independent and a use case dependent part. Zones in the use case independent part preserve all of the original information that is in the data, while zones in the use case dependent part accept information loss to achieve a better support of certain uses. According to the zone reference model, any zone aside from the Raw Zone can be omitted.

As depicted in Table III and Fig. 3, all zones contain a *protected part*. This part is encrypted and secured, and stores data that need extensive protection (e.g., personal data). Data wander from the protected part of one zone to the protected part of the next zone. They may only leave the protected part

TABLE III. OVERVIEW OVER THE ZONES' ATTRIBUTES IN THE ZONE REFERENCE MODEL.

	Landing	**Raw**	**Harmonized**	**Distilled**	**Explorative**	**Delivery**
Granularity (Raw – Aggregated)	Raw	Raw	Raw	Aggregated	Any	Any
Schema (Any – Consolidated)	Any	Any	Consolidated	Consolidated, enriched	Any	Any
Syntax (Unchanged – Consolidated)	Basic transformations	Basic transformations	Consolidated	Consolidated	Any	Any
Semantics (Unchanged – Processed)	Mostly unchanged, unless needed for compliance	Mostly unchanged, unless needed for compliance	Mostly unchanged, unless needed for compliance	Complex processing	Any	Any
Properties	Governed, non-historized, non-persistent, protected part, use case independent	Governed, historized, persistent, protected part, use case independent	Governed, historized, persistent, protected part, use case independent	Governed, historized, persistent, protected part, use case dependent	Not governed, non-persistent, protected part, use case dependent	Governed, persistent, protected part, use case dependent
User Groups	Systems, processes	Data scientists, systems, processes	Data scientists, systems, processes	Data scientists, domain experts, systems, processes	Data scientists	Any human users, systems, processes
Modeling Approach	Any	Any	Standardized	Standardized	Any	Any

after being desensitized (e.g., by anonymization). Data in this part are subject to strict access controls and governance. The protected part shares all other characteristics with the rest of the zone it is in.

A. Landing Zone

The *Landing Zone* is the first zone of the data lake. Data are ingested as batch or as data stream from the sources. The Landing Zone is beneficial when the requirements of the ingested data and those of the Raw Zone diverge. For example, data might need to be ingested at a vast rate due to its volume and velocity. If the technical implementation of the Raw Zone cannot provide this high ingestion rate, a Landing Zone can function as a mediator in between: data are ingested at a high rate into the Landing Zone, and then are forwarded to the Raw Zone as batches.

For the *data characteristics*, data ingested into the Landing Zone remains mostly raw. Their *granularity* remains raw, just like in the source systems. The *schema* of the data is not changed; they can simply be copied in their source system format. However, their *syntax* might be changed. Basic transformations are allowed upon ingestion into the Landing Zone, such as adjusting the character set of strings or transforming timestamps into a common format. In addition, data may be masked or anonymized to comply with legal regulations. Aside from these changes, the *semantic* of the data remains the same as in the source systems.

As shown in Table III, the *properties* of the Landing Zone are: *governed*, *non-historized*, *non-persistent*, i.e., data are removed when moving to the Raw Zone; and *use case independent*, as data remain mostly raw. With regards to *user groups,* the Landing Zone is not intended to be used by end users. Only systems and processes may enter data into or retrieve it from the Landing Zone. Finally, no *modeling approach* is defined as data may be ingested in any format.

In the End Customer Service analytics use case, both batch data from, e.g., ERP systems, and streaming data from the field are ingested into the data lake. The Landing Zone forwards streaming data to both a batch Raw Zone for permanent storage and batch processing, and to a real-time Raw Zone, based on hybrid processing architectures (e.g., lambda architecture [21]). The Landing Zone is implemented using Kafka[1], forwarding the data to diverse storage systems (e.g., HDFS[2], relational databases), and stream processing engines (e.g., Spark Streaming[3]). As the use case uses customer data, a lot of the data captured are personal. They are stored in the protected part of the Landing Zone that is realized through separated databases. In addition, anonymized versions of these data are available in the normal Landing Zone for arbitrary use.

B. Raw Zone

All data in the data lake is available in mostly raw format in the *Raw Zone*. Only basic transformations (see Landing Zone) are applied on the data. If the Landing Zone is omitted, these transformations are performed in the Raw Zone.

The attributes of the Landing and the Raw Zone only differ in two aspects (see Table III): the *properties* and the *user groups*. For properties, the Raw Zone stores data *persistently*. In general, data should neither be manipulated nor deleted from the Raw Zone. However, according to our experiences, such an approach is not feasible in practice, as the amount of data to be stored grows rapidly (e.g., sensor measurements) and some data are subject to legal regulations that demand deletability (e.g., GDPR). Thus, data may be manipulated or deleted, resulting in a trade-off between storage space reduction and compliance, and completeness of data.

[1] https://kafka.apache.org/
[2] https://hadoop.apache.org/
[3] https://spark.apache.org/streaming/

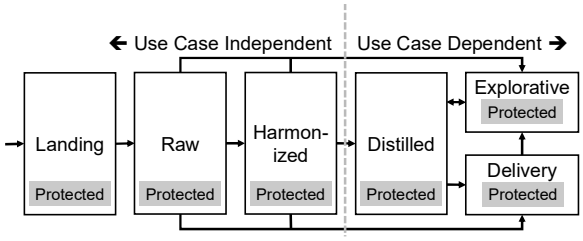

Fig. 3. The developed zone reference model comprises six zones. From the left to the right, data are processed more and more for specific usage.

Additionally, data stored in the Raw Zone are *historized*, i.e., changes in the data are traceable. As for the *user groups*, data scientists may access the Raw Zone. This user group has a deep understanding for data analytics. They can copy data to e.g., the Explorative Zone for analytics. However, use of data from the protected part is heavily restricted.

All data gathered in the End Customer Service analytics use case are permanently stored in the Raw Zone, batch and streaming data alike. The Raw Zone consists of various storage systems (e.g., HDFS, relational databases, cf. [22]), storing data where it fits best. For example, streaming data are typically stored as JSON files in HDFS, while data from ERP systems are stored in relational databases. To historize data changes, updates are added as new, timestamped records. All personal data in the Raw Zone are anonymized, rendering the protected part irrelevant in this use case.

C. Harmonized Zone

A subset of the data stored in the Raw Zone is passed to the *Harmonized Zone* in a demand-based manner. It is important to note that these data are not deleted from the Raw Zone. Instead, the Harmonized Zone contains a copy of or a view on the data in the Raw Zone. The Harmonized Zone is also the place where master data [23] are accessible for analyses. As these data are crucial for enterprises, master data management is of high importance in the data lake. Thus, they should exclusively be accessed after being cleansed.

The *data characteristics* in this zone differ greatly from those in the Raw Zone (see Table III). Data *schema* and *syntax* change when compared to the source data. Data from different source systems are integrated into a consolidated schema, regardless of their structure (e.g., by link-based integration [24]). The data syntax is also consolidated in the Harmonized Zone: when data from multiple source systems are merged (e.g., multiple tables into one), data types have to be adapted. The *properties* and *user groups* do not change compared to the Raw Zone.

The aim of the Harmonized Zone is to provide a harmonized and consolidated view on data. To this end, the Harmonized Zone uses a standardized *modeling approach* (e.g., dimensional modeling or Data Vault [14]) that all of the enterprise's data are modeled in. This does not mean all data are part of one overarching schema. Rather, multiple partial schemata exist that cover different data sources and contexts. Each of these partial schemata is growing incrementally when new data are added to the Harmonized Zone in a demand-based manner. It might happen that multiple of these partial schemata are connected to one bigger partial schemata. However, it is not the goal of the Harmonized Zone to provide one single schema for all data in the enterprise. In the partial schemata, a high level of data integrity (e.g., primary key and foreign key constraints) should be satisfied.

For the End Customer Service analytics use case, data in the Harmonized Zone are modeled using Data Vault, connecting data across storage systems. Heterogeneous data from various sources are integrated, e.g., structured data on customers are connected to pictures of travel routes the customers took using link-based integration [24]. Consistency and correctness are ensured.

D. Distilled Zone

In contrast to the Raw and Harmonized Zone, where the focus is to quickly make data available for use, the *Distilled Zone* focuses on increasing the efficiency of following analyses by preparing the data accordingly. To this end, the *data characteristics* differ with regards to *granularity* and *semantics* (see Table III). The granularity of the data may be changed, e.g., data may be aggregated for the calculation of KPIs. Complex processing is applied that change the data's semantics but are too extensive for the Landing Zone, Raw Zone, and Harmonized Zone. However, the *schema* might also change slightly, depending on the supported use case. For example, fields to enrich the data could be added.

Regarding the *properties*, the Distilled Zone is the first zone that is *use case dependent*. Data are processed to fit a certain group of use cases. This applies to both batch and streaming data. For batch data, multiple different transformations of the same data might be available in the Distilled Zone. The *user groups* of the Harmonized Zone also have access to the Distilled Zone. In addition, domain experts with less experience in data analytics are allowed to access data in the Distilled Zone. To access the data, easy-to-use interfaces should be provided that allow to query the data using known query languages (e.g., SQL).

In the End Customer Service analytics use case, the Distilled Zone is modeled using Data Vault. Additional fields are added to the data, such as pre-defined KPIs that are of interest for multiple business divisions. Data are transformed using business logic and the results are stored as views, e.g., point in time views providing the most recent data values instead of the full change history.

E. Explorative Zone

The *Explorative Zone* is the place where data scientists can play with and flexibly use the data. There are no common *data characteristics* that apply to all data in the Explorative Zone (see Table III). Data scientists can use and explore data in the data lake in any way they desire, except for sensitive data. These data are only usable according to strict rules. *Granularity*, *schema*, *syntax*, and *semantic* may be changed in any way necessary for analyses. The *properties* differ from those of the Distilled Zone. The Explorative Zone is *not governed*, allowing a user to process data in any way beneficial for their intent. Depending on the use case, data are not necessarily *historized* anymore, which is why the historized property is not set for this zone. In addition, the Explorative Zone is *non-persistent*. However, as results of an analysis may be useful for data analytics in general (e.g., a transformation beneficial for analysis or a mining model that should be used operationally), these results can be forwarded to the Distilled Zone before being deleted from the Explorative Zone. There are few rules and restrictions for the use of data, which makes the Explorative Zone the most flexible zone. Sometimes, it is necessary to perform advanced analytics on sensitive data. These analyses have to be performed in the protected part of the Exploration Zone,

which is encrypted and secured. For the *user groups*, only data scientists access data in this zone. They are allowed to write into the Explorative Zone and thus can store transformations or analysis results. Also, they can pull data from any other zone into the Explorative Zone. Any *modeling approach* is allowed, as data scientists process the data as needed.

In the End Customer Service analytics use case, data scientists use the Explorative Zone to discover, e.g., novel KPIs that might be of interest for different lines of business. They do so in a Sandbox on a system based on, e.g., Hadoop[4] and use tools, such as Python[5] or various visualization tools.

F. Delivery Zone

In the *Delivery Zone*, small subsets of data are tailored to specific usage and applications. This does not only include analytical use cases, such as reporting and OLAP, but also operational use cases, e.g., providing data relevant for certain use as context-aware decision information packages [25]. This zone thus provides functionality similar to data marts and operational data stores in data warehousing. Data from this zone may be forwarded to external data sinks. Similarly to the Explorative Zone, the *data characteristics* depend on specific use cases in contrast to use case groups in the Distilled Zone (e.g., data are prepared for specific tools). To this end, multiple different transformations of the same data might be available. As seen in Table III, however, the Delivery Zone differs from the Explorative Zone in its *properties* and *user groups*. Data in the Delivery Zone is *governed* and stored *persistently*, unless the use case it was processed for is no longer of interest. Data in this zone can be accessed by a large number and variety of users. Human users (data scientists, domain experts, business users) as well as systems and processes read data from the Delivery Zone. The Delivery Zone especially supports users with little knowledge on data analytics. Data have to be easily findable and importable into various analytics tools. As for the *modeling approach*, data are available in whatever format supports the intended use case best, e.g., dimensional modeling for OLAP, or flat tables for operational use.

For the End Customer Service analytics use case, the Delivery Zone is realized using relational databases. Data are prepared for traditional reporting and OLAP use cases, and modeled using dimensional modeling, such as the star schema. Managers access this zone for their sales and revenue reports.

VI. Prototypical Implementation and Assessment

In this section, we describe our prototypical implementation of the zone reference model for an analytic use case (Section A) and assess it in two steps: first, we assess our model's applicability and feasibility based on the prototypical implementation. Second, we show that the zone reference model meets the requirements derived in Section II (Section B).

A. Prototype

We illustrate the feasibility, applicability, and the benefits of the zone reference model by prototypically implementing it for a use case from *product lifecycle management*. This data analytics use case was not yet considered in the development of the zone reference model. This subsection first details on the use case itself, before discussing the zones needed and providing a detailed description of the zones' implementations. In particular, we highlight implementation decisions that were guided by the zone reference model.

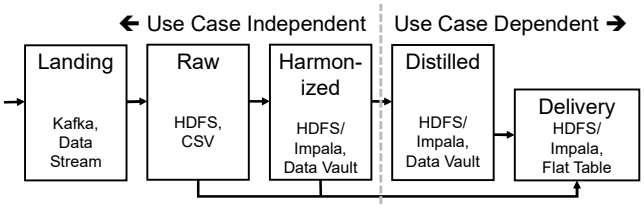

Fig. 4. The prototypical implementation of the zone reference model uses five of the six zones. Data are stored in different systems and schemata to realize the defined characteristics of the zones.

The used analytics use case is well suited for assessing the generality of the zone reference model, as it covers multiple business domains (e.g., product design, engineering, customer service), as well as a wide variety of data sources (e.g., sensors, ERP systems) and data formats (structured, semi-structured, unstructured). The aim is to use field data collected at the customer's site to discover defects in the product and to enhance future product generations. To this end, semi-structured sensor data are combined with structured product and customer information, and unstructured data from the engineering process (e.g., CAD files). In particular, domain experts investigate the history of field data to identify trends and unexpected behavior. A visualization tool is used to depict these trends. Semi-structured sensor data are combined with structured information on the product, such as technical specifications. These data are used to discover irrational behavior. Insights are forwarded to product design and engineering to prevent such behavior in future product iterations. While this is an enterprise use case, similar use cases are conceivable in other areas, such as healthcare (e.g. monitoring patient heart rates).

Fig. 4 depicts our implementation of the zone reference model. As the data used in this use case are partially streaming data, we decided to add a Landing Zone to our implementation. This way, the rapidly arriving data can be buffered and forwarded as batches. The Raw Zone is mandatory and part of every zone model implementation. Connecting the streaming data and the technical data used in this use case is beneficial for other applications as well, e.g., root cause analysis on a product failure. We thus decided to include a Harmonized Zone, where data from different source systems are consolidated and connected. This connected schema then can be reused by other use cases as well. We also included a Distilled Zone, as certain processing steps are often re-executed on the same data. Such processing steps are, for example, the aggregation of sensor measurements. While the data lose their use case independence through the aggregation, they can still be reused in other applications that require aggregated measurements. For this use case, a certain visualization tool is used to investigate the data. Thus, we also include the Delivery Zone in the prototype to hold data in the right format for the import into the tool. Since the intended use of the data in this use case is pre-defined, we did not implement an Explorative Zone. The use of the zone reference model here was beneficial, as it defined which zones to include to store the desired data in.

The *Landing Zone* was implemented using Kafka. Field data are collected as data stream from the product in intervals

[4] https://hadoop.apache.org/

[5] https://www.python.org/

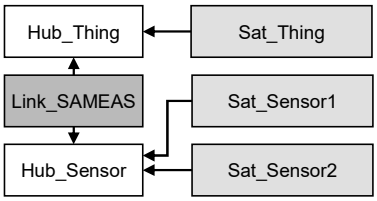

Fig. 5. In the Harmonized Zone, we modeled the data using Data Vault. The collected field data are linked to a sensor hub, which connects to a thing hub holding additional information from different source systems.

of around 200 milliseconds. Kafka buffers these data before storing them in the Raw Zone. In this use case, no transformations are applied to the data in the Landing Zone.

We used HDFS to implement the *Raw Zone*. The streaming data arriving from Kafka are stored as csv files organized by a timestamp. These files contain measurements from various sensors, e.g., temperature or the speed of moving parts, each associated with a timestamp and the id of the smart thing (e.g., a sensor) that captured the data. This id refers to a table in the Raw Zone that originates from an ERP system and contains further data on the smart thing itself, such as its name and description. If data change in the source systems, they are historized by adding a new record containing the updated values with the change timestamp to the Raw Zone. This way, changes are available quickly in the data lake. Further tables are available in the Raw Zone, containing information on, e.g., the bill of material. This information originates from Excel files. The table containing the bill of material holds the ids of all sub-components of a stored thing, their description, and their used quantity. However, these data are not of interest for the current use case. For the Raw Zone, the zone reference model specifies no standardized modeling approach, which is why the data remain in the format used in the source systems, i.e., tables and csv files.

The *Harmonized Zone* was also realized using HDFS. Data are processed and forwarded to this zone by ETL processes. For the standardized modeling approach, we decided on Data Vault, as it supports flexible modeling for data lakes [14]. Fig. 5 depicts the tables of the Data Vault model for our use case implementation. They were implemented using Impala[6]. From the ERP table on the smart thing, we created a hub table Hub_Thing and a respective satellite table Sat_Thing according to the Data Vault methodology. These two tables contain all information that originates from ERP systems, such as the thing's name. If the thing is also a sensor, the same-as link connects it to the respective entry in the sensor hub. Attached to the sensor hub are various satellites, one for each observed sensor. These satellites do not only contain the measured values, but also their timestamps and their unit. Data Vault comprises a historization methodology, which makes use of the change timestamps from the Raw Zone. The zone reference model here guides the integration of data sources by specifying a standardized modeling approach. Data available in the Harmonized Zone can be used whenever it is necessary to link sensor measurements to the sensor or thing that captured them.

In the *Distilled Zone*, we prepared the measurements of one sensor for visualization. To this end, we aggregated the measurements to minutes using the average. In addition, we reduced the number of decimal places from 15 to two. This processing was performed according to the requirements of the use case. We added these processed values as a new aggregated satellite to the hub table Hub_Sensor using Impala. The guidance the zone reference model provides here is to separate use case independent but cleansed data (in the Harmonized Zone) from use case dependent data (in the Distilled Zone). While data in the Harmonized Zone can be used in any analysis that requires integrated data, data in the Distilled Zone has been changed semantically, and is thus tailored to use cases where one value per minute is sufficient.

We also realized the *Delivery Zone* with HDFS and Impala. We extracted one day of aggregated sensor measurements from the Distilled Zone. As the visualization tool used in this use case requires a flat table, we combined these selected data with the thing id from the hub table Hub_Thing of the Harmonized Zone. The resulting table contains the thing id, the timestamp of the measurement, and the aggregated and rounded measurement itself. The zone reference model provides the benefit of supporting pre-defined use cases in the Delivery Zone, while still maintaining the data lake's flexibility in preceding zones. Data available in the Delivery Zone can only be reused for very few use cases. However, should a use case have the same requirements as the implemented one (one day of data, flat table, same data and aggregation degree), the data can still be reused.

B. *Applicability and Requirements Assessment*

The discussion above shows that the zone reference model is applicable for a use case that was not considered during the development of this reference model. This use case is representative, as it combines various kinds of data (structured batch data and semi-structured streaming data from sensors), and performs a typical analysis in the form of visualization. Overall, this implementation shows that the zone reference model is applicable and can be tailored to the use case at hand, as zones can be omitted for certain uses (e.g., the Exploration Zone in this case). The zone reference model provided guidance by defining each zones characteristics in a systematic manner. All zones are realizable by choosing appropriate realization techniques. For each zone, we provided examples of how the data in this zone can be reused for other use cases to reduce the number of processing steps.

In Section II, we derived seven general requirements a zone model must meet to support a wide variety of data lake use cases relevant in practice. These requirements served as a basis for the development of the zone reference model. The following paragraphs detail on how the requirements are met.

The reference model considers *pre-processed* data within the Distilled Zone, as data are prepared for specified use cases. For example, data might be aggregated, filtered, or otherwise enhanced to improve analyses. The Harmonized Zone holds data in a syntactically cleansed format, and semantical errors in the data can be corrected in the Distilled Zone. Hence, syntactically and semantically *cleansed* data are considered as well. The standardized modeling approach in the Harmonized Zone is used to provide *integrated* data from different sources. Sensitive data are *governed* in different parts of the zone reference model: data are available with ensured quality in the Harmonized Zone. In addition, the protected part of each zone contains critical data and secures them using appropriate access controls. Outside of these protected parts, these data are

[6] https://impala.apache.org/

only available in an anonymized and insensitive format if at all. The Delivery Zone of the zone reference model provides functionality similar to data marts. Data in this zone are specifically prepared for *reporting and OLAP*, and operational use cases. Data scientists can access the Explorative Zone for *advanced analytics*. Data from any zone, except the landing zone, can be used to uncover new insights. While the Exploration Zone itself is non-persistent, results that provide a benefit when reused can be *written back* into the Distilled Zone. It follows that the zone reference model provides the features that we derived in Section II as mandatory for practical use.

Overall, our assessment underlines that the zone reference model is applicable for real-world data lake use cases. Additionally, it can be adapted and tailored to a specific use case as shown in the implementation and thus streamline the realization of data lake use cases. Data can be reused between use cases as detailed in Section VI.A. Yet, the zone reference model focuses on a conceptual view on data lake management. That is, the efficiency and reusability of a specific zone model instance still depend on its technical implementation.

VII. Conclusion and Future Work

Data lakes promise the flexible and comprehensive analyses of data. To increase the efficiency of data analyses on data lakes and exploit synergies between use cases by reusing processed data, different processing degrees of data are often managed in zones. Literature introduces a multitude of different zone models, but there exists no consensus and no assessment. It remains unclear which zones should be included in a zone model in practice to support the multitude of use cases that are implemented on a data lake.

In this work, we addressed this gap. From multiple data analytics use cases at a large, globally active manufacturer, we derived requirements towards an enterprise-grade zone-based data lake management. It showed that existing zone models could not meet all requirements. In addition, they significantly lacked description detail as well as a derivation methodology and assessment. We thus developed a meta-model for zones that allows to describe zones in a zone model in a generic scheme. Based on this meta-model for zones and the derived requirements, we developed the zone reference model to streamline the realization of use cases in an enterprise-wide data lake. This reference model specifies six zones (Landing Zone, Raw Zone, Harmonized Zone, Distilled Zone, Explorative Zone, and Delivery Zone) and details their characteristics. Finally, we prototypically implemented the zone reference model as a proof of concept for a data analytics use case from a real-world enterprise. In doing so, we evaluated our concept's practicability in a realistic scenario. We also assessed the reference model's suitability with regard to the derived requirements. Overall, our assessment shows that the reference model was generally applicable.

Our current implementation of the zone reference model served just as a proof of concept. Future work thus has to investigate possible implementations for zones in a data lake, identify challenges, and derive implementation patterns for the zones of the reference model. These patterns consider the dependencies between zones, data modeling, storage architecture, and other aspects of the data lake. In doing so, they provide guidance towards the definition of a data lake architecture, allowing for standardization and interoperability among data lakes and other systems.

References

[1] P. Russom, "Data Lakes - Purposes, Practices, Patterns, and Platforms," *TDWI*, vol. Q1, 2017.

[2] P. Tyagi and H. Demirkan, "Data Lakes: The biggest big data challenges," *Analytics*, vol. 9, no. 6, pp. 56–63, 2016.

[3] C. Mathis, "Data Lakes," *Datenbank-Spektrum*, vol. 17, no. 3, pp. 289–293, 2017.

[4] M. Madsen, "How to Build an Enterprise Data Lake: Important Considerations before Jumping In," *Third Nature Inc.*, 2015.

[5] P. Patel, G. Wood, and A. Diaz, "Data Lake Governance Best Practices," *The DZone Guide to Big Data - Data Science & Advanced Analytics*, vol. 4, pp. 6–7, 2017.

[6] B. Sharma, *Architecting Data Lakes - Data Management Architectures for Advanced Business Use Cases*, 2nd ed. O'Reilly Media, 2018.

[7] D. Linstedt and M. Olschimke, *Building a Scalable Data Warehouse with Data Vault 2.0*. Elsevier, 2015.

[8] B. Inmon, *Data Lake Architecture - Designing the Data Lake and avoiding the Garbage Dump*. Technics Publications, 2016.

[9] A. Gorelik, *The Enterprise Big Data Lake*. O'Reilly Media, 2016.

[10] F. Ravat and Y. Zhao, "Data Lakes: Trends and Perspectives," in *Proceedings of the 30th International Conference on Database and Expert Systems Applications (DEXA 2019)*, 2019.

[11] P. Zikopoulos, D. DeRoos, C. Bienko, R. Buglio, and M. Andrews, *Big Data Beyond the Hype*, 1st ed. McGraw-Hill Education, 2015.

[12] V. Morabito, "Big Data and Analytics for Competitive Advantage," in *Big Data and Analytics*, Cham: Springer International Publishing, 2015, pp. 3–22.

[13] J. Lee, H.-A. Kao, and S. Yang, "Service Innovation and Smart Analytics for Industry 4.0 and Big Data Environment," *Procedia CIRP*, vol. 16, pp. 3–8, 2014.

[14] C. Giebler, C. Gröger, E. Hoos, H. Schwarz, and B. Mitschang, "Modeling Data Lakes with Data Vault: Practical Experiences, Assessment, and Lessons Learned," in *Proceedings of the 38th Conference on Conceptual Modeling (ER 2019)*, 2019.

[15] H. Chen, R. H. L. Chiang, and V. C. Storey, "Business Intelligence and Analytics: From Big Data to Big Impact," *MIS Quarterly*, vol. 36, no. 4, pp. 1165–1188, 2012.

[16] M. T. Koch, H. Baars, H. Lasi, and H.-G. Kemper, "Manufacturing Execution Systems and Business Intelligence for Production Environments," in *Proceedings of the 16th Americas Conference on Information Systems (AMCIS 2010)*, 2010.

[17] R. Bose, "Advanced analytics: opportunities and challenges," *Industrial Management & Data Systems (IDMS)*, vol. 109, no. 2, pp. 155–172, 2009.

[18] A. Ismail, H.-L. Truong, and W. Kastner, "Manufacturing process data analysis pipelines: a requirements analysis and survey," *Journal of Big Data*, vol. 6, no. 1, p. 1, 2019.

[19] C. Giebler, C. Gröger, E. Hoos, H. Schwarz, and B. Mitschang, "Leveraging the Data Lake - Current State and Challenges," in *Proceedings of the 21st International Conference on Big Data Analytics and Knowledge Discovery (DaWaK 2019)*, 2019.

[20] P. Stiglich, "Data Modeling in the Age of Big Data," *Business Intelligence Journal*, vol. 19, no. 4, pp. 17–22, 2014.

[21] N. Marz and J. Warren, *Big Data - Principles and best practices of scalable real-time data systems*. Manning Publications, 2015.

[22] F. Nargesian, E. Zhu, R. J. Miller, and K. Q. Pu, "Data Lake Management: Challenges and Opportunities," in *Proceedings of the 45th International Conference on Very Large Data Bases (VLDB'19)*, 2019.

[23] B. Otto, "How to design the master data architecture: Findings from a case study at Bosch," *International Journal of Information Management*, vol. 32, no. 4, pp. 337–346, 2012.

[24] C. Gröger, H. Schwarz, and B. Mitschang, "The Deep Data Warehouse: Link-Based Integration and Enrichment of Warehouse Data and Unstructured Content," in *Proceedings of the 2014 IEEE 18th International Enterprise Distributed Object Computing Conference (EDOC 2014)*, 2014.

[25] [25] E. Hoos, P. Hirmer, and B. Mitschang, "Context-Aware Decision Information Packages: An Approach to Human-Centric Smart Factories," in Proceedings of the 21st European Conference on Advances in Databases and Information Systems (ADBIS 2017), 2017.