Cloud Computing and Data Science
https://ojs.wiserpub.com/index.php/CCDS/

UNIVERSAL WISER PUBLISHER

Research Article

Reverse-Engineering the Design Rules for Cloud-Based Big Data Platforms

Ravi S. Sharma[1,3*], **Purna N. Mannava**[2,3], **Stephen C. Wingreen**[2,3]

[1]College of Technological Innovation, Zayed University, Abu Dhabi City, United Arab Emirates
[2]School of Business, University of Canterbury, Christchurch, New Zealand
[3]Center for Inclusive Digital Enterprise, New Zealand
E-mail: rsharma@ceide.org

Received: 20 October 2021; **Revised:** 7 January 2022; **Accepted:** 18 January 2022

Abstract: Big Data's 5 V complexities are making it increasingly difficult to develop an understanding of the end to end process. Big Data platforms play a crucial role in many critical systems, combining Internet-of-Things, Artificial Intelligence and Business Analytics. It is both relevant and important to understand Big Data systems to identify the best tools that fit the requirements of heterogeneous platforms. The objective of this paper is to "discover" a set of design principles and rules for Cloud-based Big Data platforms for complex, heterogeneous environments. The design scope comprises Big Data's significance, challenges and architectural impacts. Using a methodology called Reverse Engineered Design Science Research (REDSR), artifacts from leading vendors are used to elicit the design principles and rules with relevant details of Big Data components. We conclude that the findings are relevant and useful for DevOps architects and practitioners in operating complex, heterogeneous Cloud-based Big Data platforms.

***Keywords*:** big data inter-operability, design specifications, heterogeneous cloud computing

1. Introduction

Digitalisation transformation is not only the evolution of electronic devices but also the integration of intelligent data into every aspect of digital live-styles [1]. Today Data is being generated through web applications, mobiles, sensors, Global Positioning System (GPS), social networks. According to Forbes there are 2.5 quintillion bytes of data that is getting generated every day [2]. Report from International Data Corporation (IDC) shows that trends in Data being created in Zeta bytes as we can see in Figure 1 [1]. From this report it unveils the sheer size and pace at which the data is being created around us.

Big Data can be referred as data characterized by huge volume, variety, veracity and velocity which makes difficult for traditional systems to store and process [3]. Big Data consists of different types of data like structural data (Relational databases), semi-structured data (XML, JASON files) and unstructured data (Images, Audio and Video files). Not all data generated is valuable, however, analysing Big Data reveals valuable insights and helps business to gain competitive advantage. Big Data is being recognised in range of industries for its ability to provide information from large data sets. Big Data revolutionised the way of using traditional analytical platforms [4]. Big Data value had already proved its efficiency in ability to cutting costs, improving operational efficiency from retail to medical fields.

DOI: https://doi.org/10.37256/ccds.3220221213

Undoubtedly Big Data need a massive infrastructure and computational platforms based on its significance and ability to blend with the latest emerging technologies like Internet of Things (IoT), Artificial Intelligence (AI) and machine learning etc. Cloud computing perfectly complements Big Data systems by cutting costs and providing required infrastructure. According to a Gartner press release cloud computing will be touching $300 billion business by year 2021 [5]. The rise in the adoption of cloud services by businesses, is because cloud systems provide speed and agility of digital business requirements along with significant cost savings and creation of new revenue sources [5]. Also, a report from International Data Corporation (IDC) identifies that there is in increase demand for Data storage in Cloud as we see in Figure 2 [1]. Data is key in any analytics.

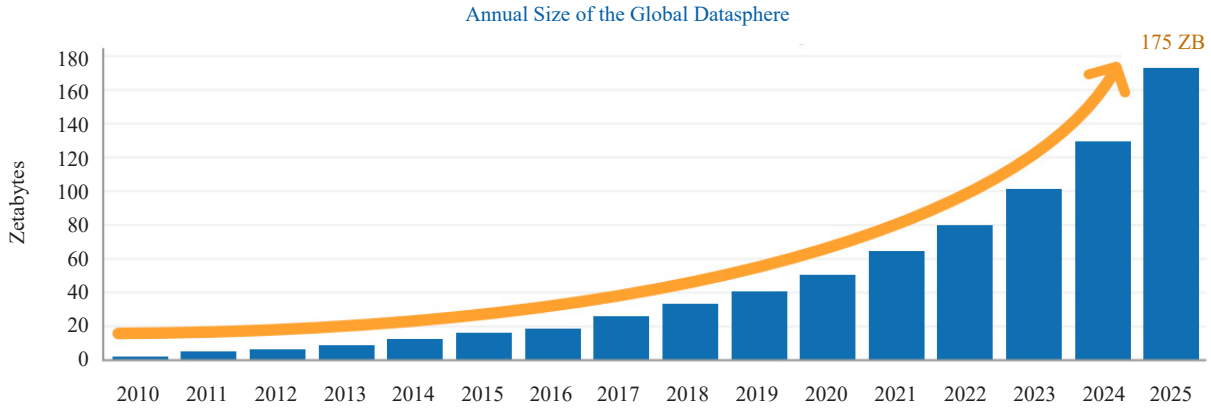

Figure 1. Data growth projections from International Data Corporation [1]

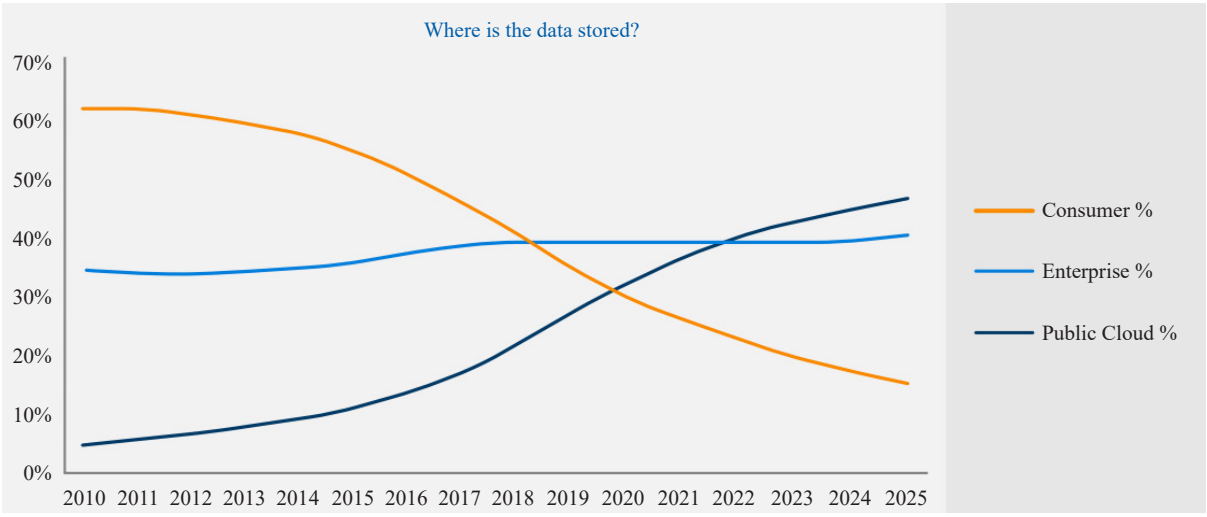

Figure 2. Data storage projections from International Data Corporation [1]

In an earlier work, Hashem et al. [6] investigate some significant open research issues such as scalability, availability, data integrity, data transformation, data quality, data heterogeneity, privacy, legal and regulatory issues, and governance. After such an extensive review of prior research, they conclude that the most important research issue facing the staging of Big Data is the heterogeneous nature of data. Bloomberg further remarks: Cloud-native computing is a paradigm shift in enterprise technology that extends the hard-fought best practices of the cloud to all of Information Technology (IT). In particular, this approach centers on a comprehensive abstraction that hides the complexity of hybrid

multi-cloud environments from the workloads and applications that run on them. It is therefore critical to identify a set of design specifications that will hide the complexity of heterogeneous cloud environments while revealing any potential inter-operability problems to be faced by their workloads and applications. The current research is to identify design principles and rules by using reverse engineering design process on Big Data platforms in cloud. Several Big Data analytical platforms are considered for this research (please refer to Appendix). The Design Science Research (DSR) paradigm in Information Systems (IS) helps to create IT artifacts, solve problems, make research contributions, and validate designs [7]. In the Preface to their classic text on the subject, Vaishnavi and Kuechler [8] declare that DSR's focus on the creation and/or improvement of IS artifacts could be further guided by the use of patterns to discover knowledge about design artifacts. Hence DSR may be effectively used to develop a deep understanding about Big Data platforms as well as for the discovery of design principles and rules from patterns. In present work and a companion paper [9], design science and reverse engineering techniques are fused together for the purpose of discovering design specifications of big data platforms.

The remainder of this paper is structured as follows. In section 2 we will review Big Data Definition, Analytics, Architecture and Use-Cases of Cloud-based Big Data Platforms. The research methodology defined as "Reverse Engineering Design Science Research" will be detailed in Section 3. In section 4 we discuss the findings of applying such a methodology to several Cloud-based Big Data Platforms. Finally, in section 5 we present some implications, limitations and future research.

2. Background review

The term Big Data originated from the need of large corporations such as Google, Facebook and Yahoo [10]. Big Data can be referred as an abstract concept because a part of its massive data sets which cannot be handled traditional IT hardware Infrastructure or hardware tools [8]. Initially Big Data was defined by 3 V's characteristics namely volume, velocity and Variety but later evolved to 4 V's addition of veracity [9]. But the research by Yaqoob I, et al. [10] specified that Big Data is a characterized by 5 V's by addition of 'Value' to the exiting characters catalogue. It is important to understand the definition of Big Data in order to distinguish it from existing Data systems. Most of the literature emphases the importance of understanding the characteristics of Big Data first, as these characteristics pose problems in extracting insight from the Big Data during analytics.

McKinsey Global Institute report states that Data and Analytics is enabling corporations to develop new business models and effectively run the core operations [11]. Analytics at a granular level of data is helping to personalize products and services. Most significantly in Health care Industry is a classic example for importance of analytics in an organization [11]. Big Data analytics reveals unknown stories of business performance for an organization. Deep analysis of huge volumes of data from different sources can only be accomplished through using Big Data tools. According to research by Jin X, et al. [12] the overall process of extracting insights from Big Data is broken into two sub processes data management and analytics. The process of researching massive data to identify hidden patterns and correlation is called Big Data Analytics [4]. The hidden patterns often show the performance of organization, customer relationship etc. which play critical role in developing business strategies.

Big Data in a vacuum is worthless and its potential can only be unlocked when it is leveraged for decision making in an organization [12]. Generally speaking, any knowledge discovery process will have stages such as input, analysis and output [13]. When observing any Big Data analytical platform, we may break down the component architecture into different layers such as ingestion, storage, processing, analytics and visualization. With these layer forms the building blocks of architecture for Big Data analytics.

There is considerable agreement in the literature that Big Data architectures use powerful distributed systems for performing storage and computational tasks on massive data sets. They use algorithms like Map Reduce for parallel processing of data which have different formats with high performance. Research by Begoli and Horey [14] noted that distributed analytical databases are preferred for storing highly structured relational data for their high optimization and distribution of data for scalability. Databases like NoSQL offers scalable databases in Big Data architecture without any adjustment to schema reducing deployment time [15]. We may conjecture that Big Data systems with distributed architecture are very effective compared to traditional systems in performing data analytics in terms of speed and time.

Big Data platforms use different architectures to achieve the goal of knowledge discovery. For the ease of understanding: current research on Big Data platforms are classified into two categories, being either Hadoop based Big Data platforms or Cloud based Big Data platforms.

Cloud computing is the fastest growing field in Information technology by promising reliable hardware and software and Infrastructure services over the internet using remote data center. Cloud systems have a powerful architecture that helps to execute complex, large-scale computations and also perform IT management functions ranging from storage to application and database service [10]. Cloud is cost effective when handling massive data sets or computational tasks for an organization because of its ability to scale up and scale back resources based on demand [16].

Current research literature discusses the topics of Big Data's significance, emerging trends in Big Data and the implementation of Big Data with upcoming technologies. But there is a distinct research gap in the aspect of design research of heterogeneous platforms since mostly research is confined to Big Data, but not Big Data platforms. Despite the significant role of Big Data platforms in extracting the value from Big Data, the understanding about these systems is poor. A Gartner survey in 2015 mentions that 60% of Big Data projects fail [17] but other sources mention that this could be even higher. Big Data is vast field with lot of challenges, design science research helps to create as well evaluate the IT artifacts with an intended solution to a observed problem [7]. Thus, it is important to implement of design science research for developing deep understanding about Big Data.

3. Research methodology

3.1 *Combining design science research with reverse engineering*

Research in Information systems is performed with two paradigms: design science and behavioural science [18]. Behavioural science helps in identifying organisational context, whereas design science research is used in information systems fields for creating new innovative ideas and artefacts which help broaden organisations abilities and capacities with respect to Information technology [18]. IT artifacts in Information systems research are classified as constructs, models and initiations which help in resolving problems through developing a successful implementation of IS using executive information systems and systems to support knowledge process [18]. In current research, as mentioned previously it is focused in develop an understanding of Big Data platforms with help of design principles and rules for organizations and researchers. Hence design science research for Big Data platforms is the best suited approach.

Design science has a two parts process of (activities) and products (artifact). Design process is a sequence of activities to create the product and evaluation, which helps to develop better understanding of the quality of product which is developed on a purpose to resolve an organizational problem [18]. In short, this seminal set of guidelines state that DSR: 1) must produce a viable artifact; 2) must develop a technology solution to a relevant problem; 3) must demonstrate validation; 4) must provide clear, verifiable contributions; 5) must apply rigour to the construction and evaluation of artifacts; 6) must search available means to reach desired objective; and 7) must be effectively presented to both technology and management audiences.

The structured way of performing design science research will help in doing effective research in order to create IT artifacts that are accepted legitimately. In the process model proposed by Peffers K, et al. [7], six processes and four research objectives are given. The design science research framework hence helps with recognition of research objectives, process activities and outcomes. Without the use of this framework this research paradigm can only justified on ad-hoc basis manner [7]. The seven "Design Science Research Guidelines" by Tsai CW, et al. [18] and Design Science Research Methodology (DSRM) Process model by Peffers K, et al. [7] have laid the foundation for synthesis of the 'Reverse engineering Design research methodology'.

Big Data systems are component-based platforms. In overview of the reverse engineering process, this helps to study any product by fragmenting it into its components and understanding them to reveal the hidden design insights of information about the product. A classic paper by Ajila states: The aim of reverse engineering is to provide comprehensive information about the design concepts included in a program. The idea is to provide abstraction mechanisms to allow an easy understanding of the structure of the program to people without or with minimal prior knowledge of it. In more recent research by Gartner [19] the reverse engineering process helps with the understanding of an existing implemented system and represents it at higher level of abstraction. Use of Model Driver Reverse

Engineering (MDRE) helps to identify higher view of systems example design models. MDRE comprises of model discovery and model understanding [19] and these processes helps to represent artifacts.

The objective of current research work is to understand the design details of cloud based Big Data platforms. To reach the benefits of this research, there is a necessity to use both the reverse engineering process and design research methodology. After thorough analysis of both the processes a new methodology is synthesized which is called as Reverse Engineering Design Science Research Methodology which is illustrated in Figure 3. The motive behind this methodology is to provide higher level of understanding of Big Data platforms with help design details after exploring the deeper details about the systems.

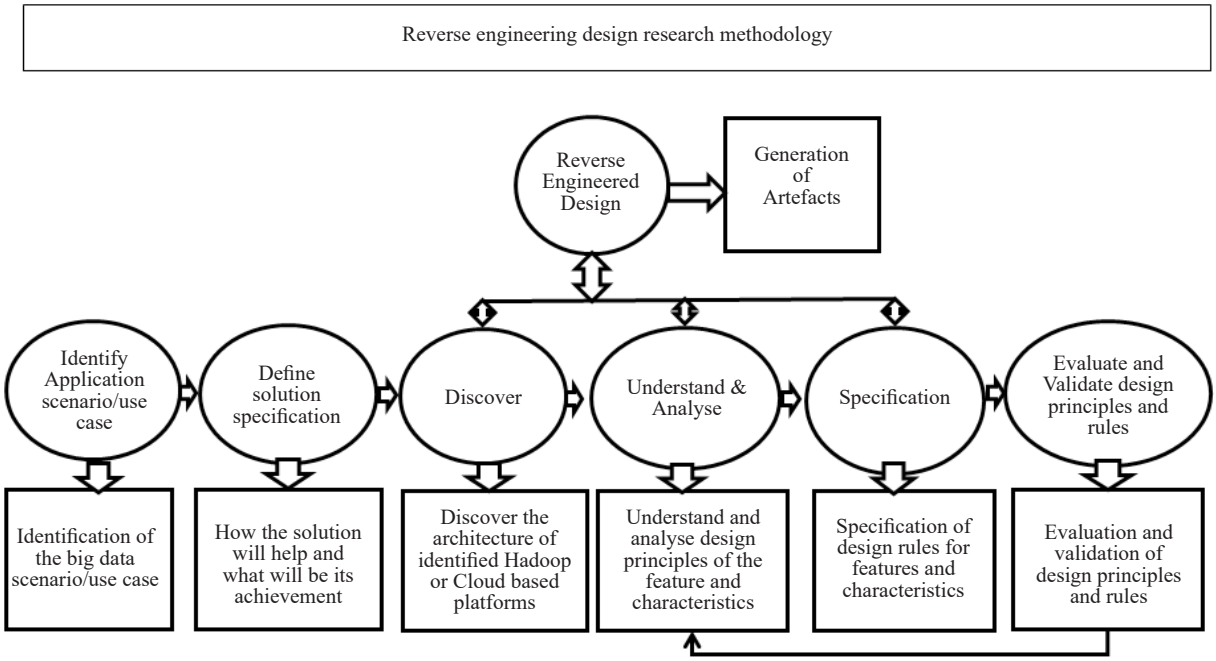

Figure 3. Reverse engineering design research methodology

The step by step processes of the hybrid Reverse Engineered-Design Science Research (RE-DSR) approach are described in a companion article [9] since it is not the purpose of this article to focus on methodology. Suffice to state, the steps shown in Figure 3 (guided by the above cited classic papers) were iteratively applied to the problem domain at hand; namely the extraction of design principles and rules for from design artefacts created from Big Data cloud platforms.

Step 1-Identification

The objective of this step is to understand the design aspects from end to end for Cloud based Big Data applications which are very complex and lack visibility for details at higher level. The design aspects include design principles and rules. A design principle can be regarded as a law or concept that is to be accepted for the creation of the product. On other hand a design rule is a bottom-line element that helps in accomplishing design principle.

Step 2-Definition

Recall that the core objective of this research is to generate design guidelines that can explain the design principles and rules of cloud based Big Data platforms. As illustrated in Figure 4, Cloud-based Big Data Analytics Platforms handle a variety of high velocity data which are parsed through collection, storage and processing steps in order to provide descriptive visualizations and analytics which shed insights. Therefore, in order to help organizations adopt this technology, as well as assist practitioners in the design of Big Data systems, we attempt to “discover” design principles

and rules from exemplar artefacts.

Step 3-Discovery

In this phase we have selected two cloud based Big Data platforms to examine, which are a Google cloud platform and an Amazon Big Data platform. For the current research, Big Data platforms which use cloud service models like IaaS, PaaS, SaaS are out of the scope. According to research by Gartner [19] the discovery phase in MDRE helps to obtain raw models of selected systems. Analysed overall design detail of Big Data platforms are identified in general as Big Data platform basic architectural layers, such are Data Ingestion/Collection, Data Storage, Data computing and processing layer along with Analytics and visualisations which are illustrated in Figure 4.

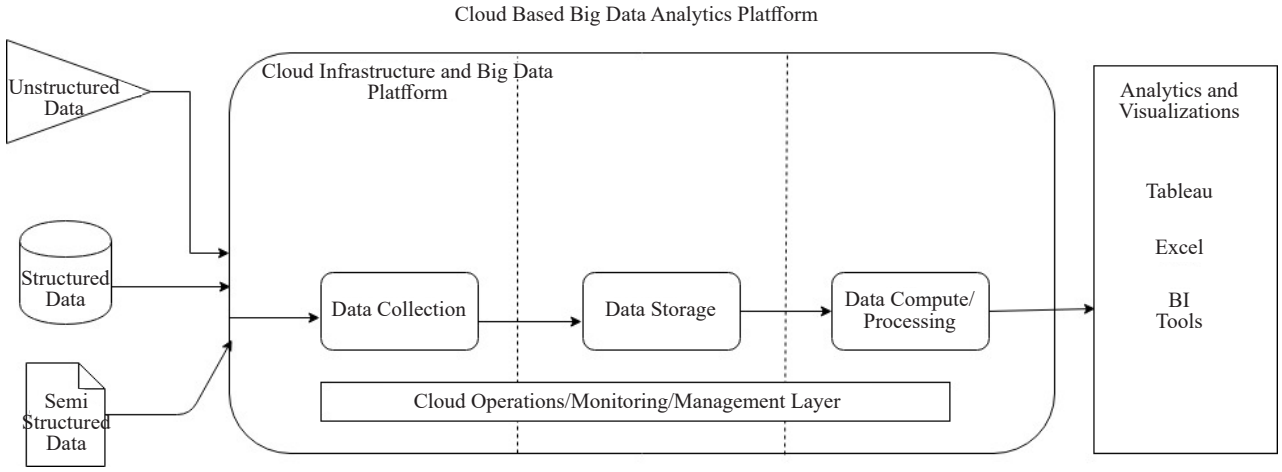

Figure 4. Context diagram of Cloud-based Big-Data platform

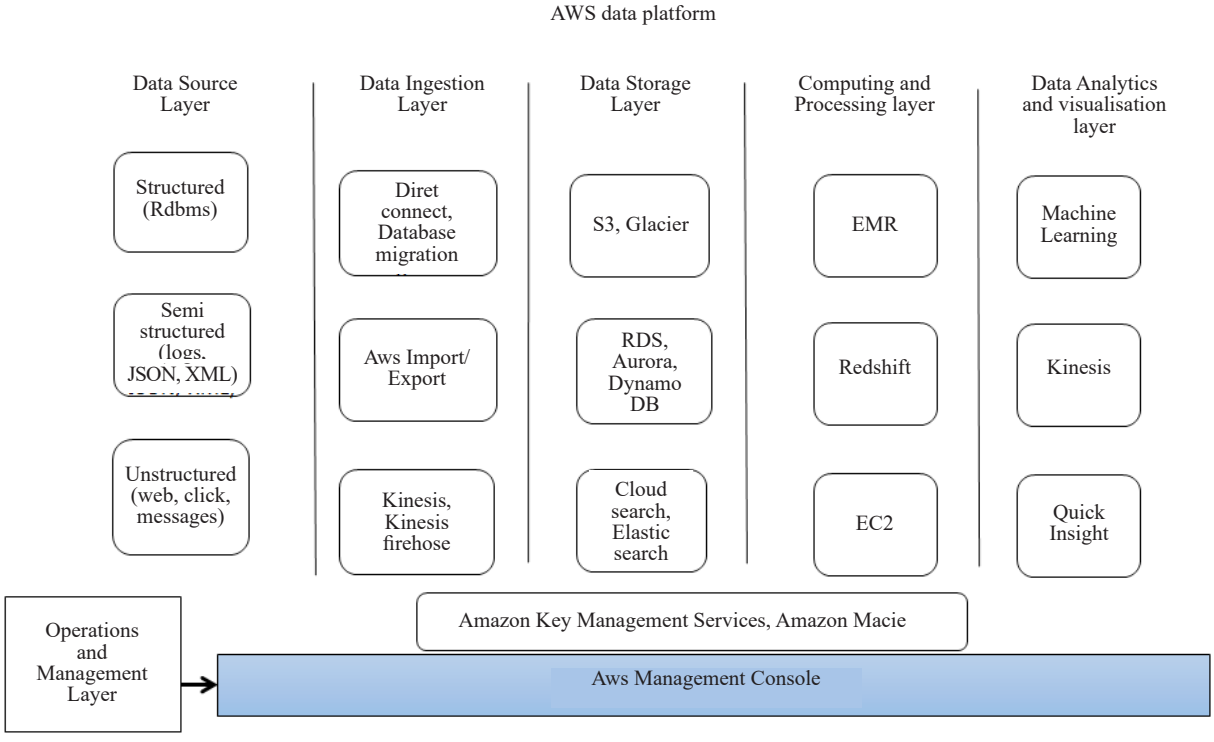

Figure 5. AWS' Big Data platform component diagram

In the next step-Discover phase-the architectural layers are broken down into detail layers to understand such components which fits into different layers.

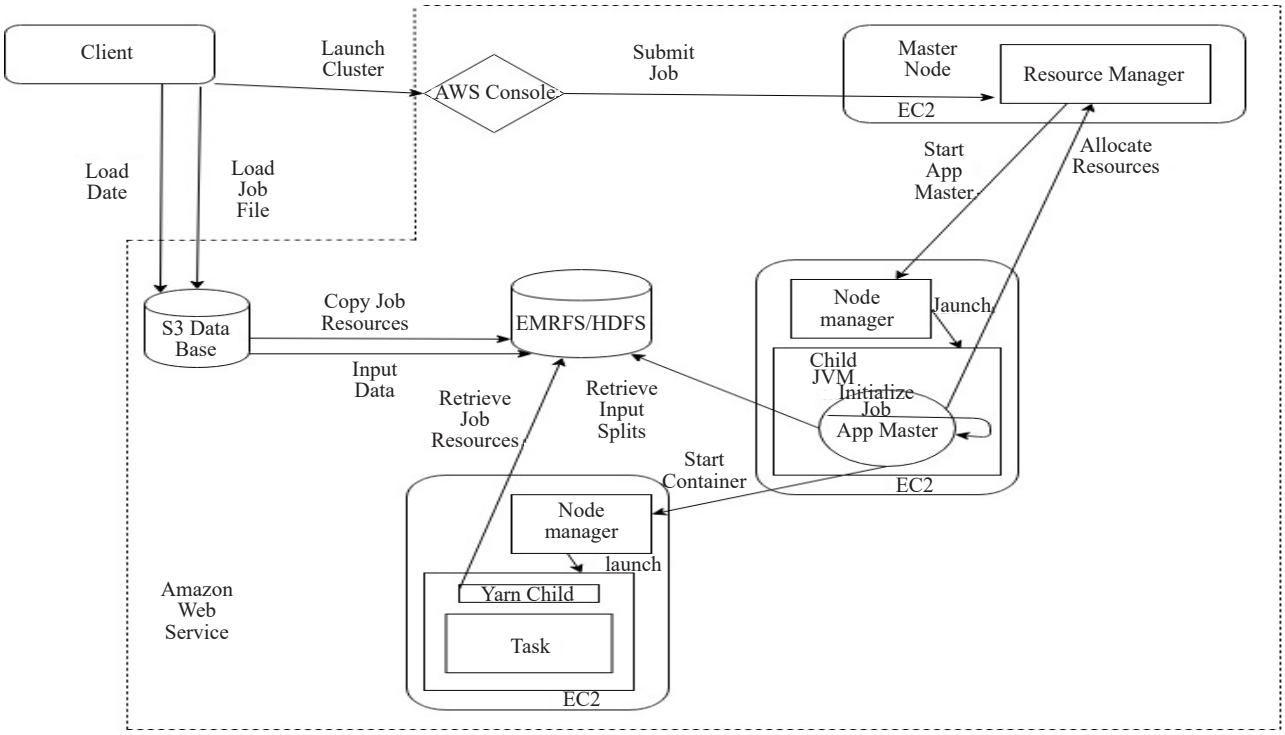

Figure 6. Design principle 6 AWS, EWR (HDFS) architecture illustration

3.2 *Amazon Web Services and Google Cloud artifacts*

Amazon Web Services (AWS) is a cloud-based platform of Amazon Inc. which can deliver on-demand computation power, data storage and other IT infrastructure through cloud services on internet by using pay-as-you-go pricing model [20]. AWS cloud models include Infrastructure as service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) [20]. AWS offers wide range of components for Ingestion, storage, computation, Databases and Analytics as we can see in Figure 5. During research all the available components related to Big Data analytics where studied. Components which are have more significance were considered and classified into layers based on their functionality in Figure 6. For example, the computing and processing layer also comprises components such as Amazon lambda, Elastic Search service etc. Being a cloud-based environment AWS has AWS console manager for User Interface. AWS even deploys security components for Data security and Access management.

Google Cloud Platform (GCP) generally comprises resources such as physical assets from example computer hardware and hard disks. Virtual machines which are spread across Google data centers across the work and connected in a high bandwidth network [21]. Google document mention that this distribution of resources across is vital for clients as it benefits redundancy during system failures. Like AWS, GCP also provides services for computation (Compute Engine, App Engine), Storage (Cloud Storage, Cloud FireStore), Databases (Cloud SQL, Cloud Big Table), Networking and Management tools which we can see in Figure 7. It has components that support IoT, Machine learning etc. For example, Cloud Machine Learning (ML) in the computing and processing layer as illustrated in Figure 8. Similar to above AWS, the components of GCP related to Big Data analytics are studied based on their feature and segregated under different layers of component-based architecture. Google also provide console manager for User interaction along tools for data security and access management.

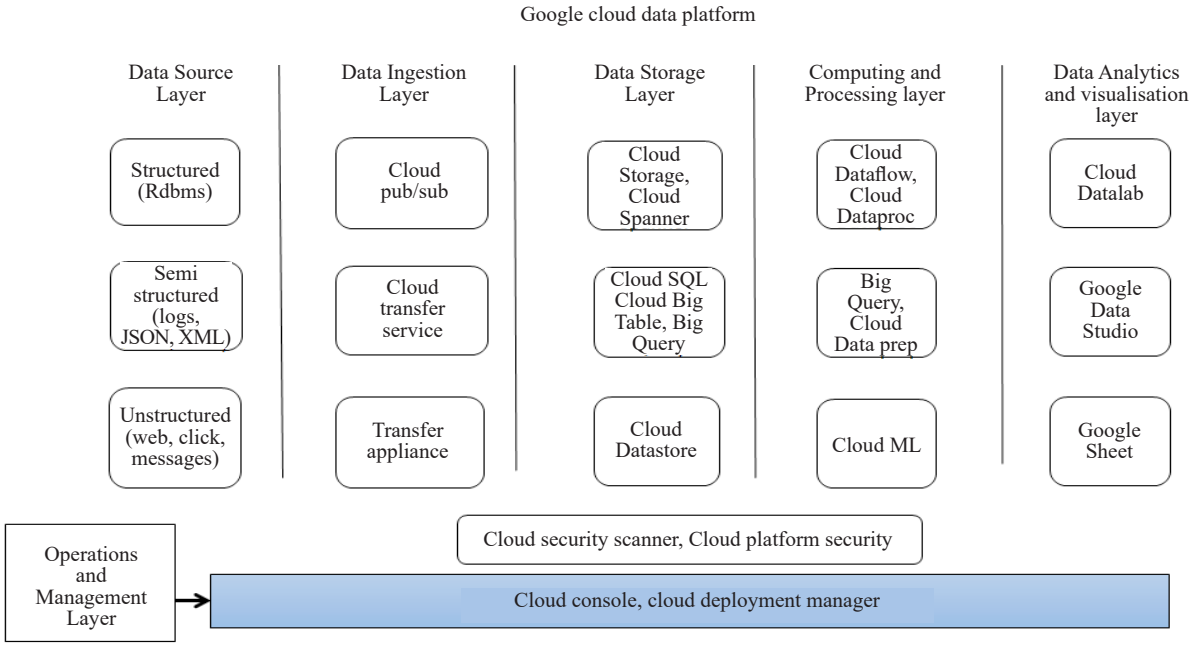

Figure 7. GCP Big Data platform component diagram

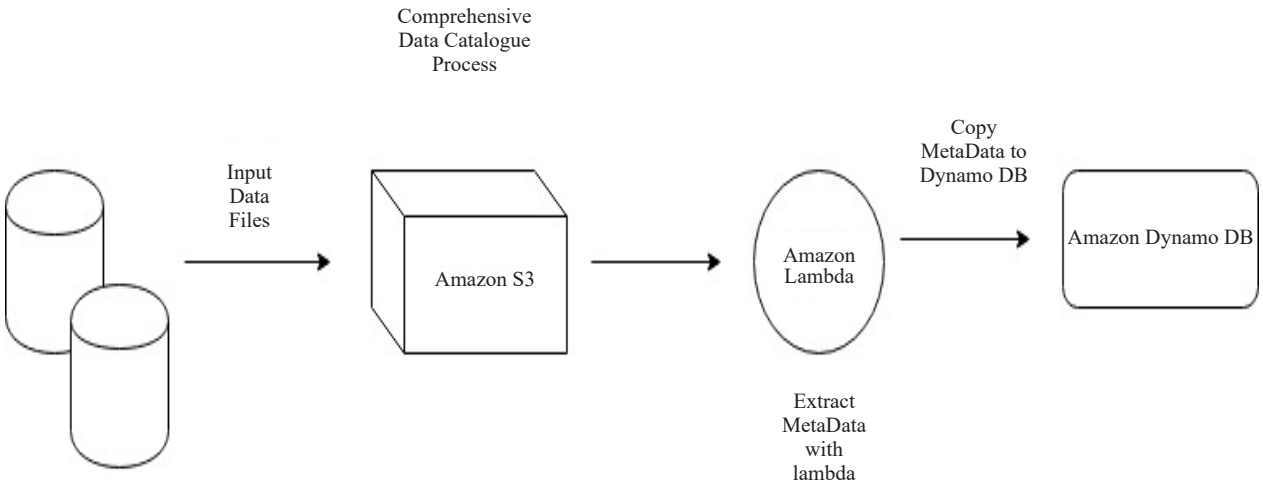

Figure 8. Design principle 10 metadata illustration

Step 4-Analytic understanding

Components are studied at deeper levels in terms of inputs, features, functionality and outputs. This helps in generation of higher level of design concepts which are design principles of the Big Data platforms.

Step 5-Specification

When design principles are extracted, the next phase components are studies with respect to internal architectural level to extract the design rules. As with the design of the Big Data solutions, the extraction and specification of rules using architectural diagrams is more an art than hard-nosed engineering.

Step 6-Evaluation and Validation

As Vaishnavi and Kuechler remark [8], with DSR it is necessary to validate and evaluate a design solution and its hypothesized claims about correctness and acceptability. Design outcomes hence need to be evaluated to identify the quality of solution to address the defined problem. Design principles and rules which fall out of scope are removed. Filtered design principles and rules are evaluated with help of use cases and architectural patterns. Design principles and rules which are not satisfactory are again re-iterated from the 'Analytic Understanding' phase. The reverse-engineered design principles and rules are derived at the end of this process. These principles and associated rules are presented as findings in the next section.

4. Applying the RE-DSR approach

In the RE-DSR procedure, Big Data architectures such as Google Big Data analytics and Amazon web services analytics may be fed in as artifacts that represent industry best practices. The full list of Big Data solutions that were used in the RE-DSR procedure are given in the Annex to this article. For each of these, every component in the architecture was studied with respect to its functionality, architectural design, inputs etc. It is worth noting that both Amazon Web Services and Google Cloud Platform are capable of creating an end-to-end data pipeline for Big Data projects from Ingestion to visualization. Both Amazon Web Services and Google Cloud Platform also include tools for performing Machine Learning and AI.

4.1 *Generating design principles for Cloud based Big Data platforms*

Examining the artifacts of the component-based architectures of Big Data platforms has revealed the following Design principles and rules. Design principles refer to the "what" of effective solution design, whereas design rules address the "how". The following design Principles and rules were discovered by observing the component features and functionality, referring to different use cases and architectural blogs of the platforms.

Design Principle 1-Flexibility and robustness of Big Data platforms

Based on observation it is identified that Robust and Flexible architecture of the platform is essential. Key rules that underlie in this principle are ability to integrate with multiple operating systems and easy integration between the Big Data layers of architecture. Business environments are dynamic in nature. Based on business requirement IT infrastructure like architecture, applications keep changing on the client side. Also new requirements may need new computational tools during analytics phase. Research by Chen C, et al. [22] identifies that in a Big Data system good architecture and frameworks should be given top priority. This principle also emphasis need for interoperability of architectural layers.

Design Rule 1.1-Ability to handle different Operating System (OS) platforms during integration based on customer needs

There are different ways operating system integration comes into the picture. Either users are required to use a different operating system in the Big Data environment. For example, user want to install Microsoft Structured Query Language (SQL) server into Big Data platform or a user want to access Cloud platform through On-premise machines with different operating system. Big Data platforms should able to support these different kinds of operating systems. For example, an Elastic Compute Cloud (EC2) instance which is meant for computation purpose can run with different operating systems, such as Linux and Windows [23]. Similarly, GCP also enables its virtual machines which are computing instances to integrate with different operating systems based on requirement [24]. In discussing the other case where client wants to connect with cloud platforms from on premises both AWS [25] and GCP [22] provide SDK's compatible for different operating systems like Linux and Windows.

Design Rule 1.2-Smooth flow of integrated processes among different layers of architecture in Big Data platform

Big Data platforms need to have ease of integration between different architectural layers. It is observed in the study in the process of data pipeline a component from any layer like storage, computing and processing may be required to be used more than once. For example, Amazon S3 is used to store all types of data. When a user finishes processing data in Elastic MapReduce (EMR), they might need to re-integrate the computation layer back with storage layer for storing the structure of the format data file. Similarly, when using GCP, a user processes the data in flow and pushes it to the storage layer. For example, cloud Pub/Sub and again needs to re-integrate with Computation layer for further processing. So, when these kinds of requirements are coming from users, it is important for a platform to have ease in integrating the architectural layers. Any deviation will cause loss of time and value from the data.

Design Principle 2-Process of Ingestion to handle different input data formats

Based on our study it is evident that ingestion layer acts as building channel between data source and Cloud based Big Data Platforms. As it is already discussed Big Data comprises of different types of data, it is important to ensure all types of data can be ingested into Big Data platforms. User requirements may vary for different reasons. Key rules under this principle include ability to ingest batch /real time data and minimum latency. The data may be fed into the such in way such as a raw data dump or it even may include batch, streaming and real time data which generated from different systems and on the fly needs to be ingested into the system. Data types vary from structured to unstructured data. Both AWS and GCP provide tools like AWS Import/Export which is physical data transfer [26], Cloud transfer service for movement of raw data [27] and Amazon kinesis and Cloud pub/Sub for real time data processing.

Design Rule 2.1-Ability of Ingestion process to support batch and real-time data

Users may need Big Data platform to support to real time and batch data. Batch data may be the case where data will be moved in batches at regular intervals of time for analytics. But real time data can be mentioned as live data or latency between data generation and needs for this type of data processing is very minimal. Real time data analytics is strong emerging requirement from users to understand more about customers, quick decision-making purposes. For example, Financial times which is the leading organisation in Business news uses AWS platform for accessing real time data for better decision while making request for proposal issues [28].

Design Rule 2.2-Latency in data ingestion should be minimal

Latency is a critical factor that can influence the decision-making process. Minimal latency needs to be assured by the cloud based Big Data platforms for the users. For example, in GCP most of the streaming data generated by users and systems is distributed, Cloud pub/sub leverages Google's front-end load balancer support for data ingestion across all the regions of GCP for minimal latency [29].

Design Principle 3-Big Data analytics platform must handle heterogeneous cloud storage platforms

Based on the design artifacts it is found that Big Data platforms should support different storage file systems (e.g. Google, Amazon, Azure) and should be able to provide parallel access from multiple third-party applications. Key rules underlie for this principle are ability to support on-premise/External hosted file systems and providing parallel access for storage. It is interesting to notice that both AWS and GCP provide flexibility to use multiple file systems on their respective Big Data platforms and also enabled support external file systems.

Design Rule 3.1-Big Data analytics platform must handle external hosted and on-premise storage

The Hadoop Distributed File System (HDFS) is prominent file system in Big Data space. Many organisations use this file system for their Big Data analytics. Likewise, the needs of user will change based on their requirements. Big Data platform should support multiple file systems meet end user needs. For example, AWS uses S3FS for storage and GCP uses Colossus as file system. However, AWS facilitates implementation of HDFS in EMR [30] and GCP enable the same with help of Cloud data proc [31]. Moreover, Big Data platforms should able to accept files transfer from external

file systems to Cloud storage File system. Both AWS and GCP provide this facility. User with help of AWS command line interface and GCP G-suit command line can initiate data file transfer into cloud storage from external environment. Also, it is interesting to note that it is possible to integrate AWS with GCP [14].

Design Rule 3.2-Big Data platform storage could be accessed in parallel by different user applications

It is equally important to provide access to storage through incorporating parallel access for different enterprise applications of storage. With the ability to support multiple file systems, these systems unable to provide access to storage for various needs, which would not serve the complete purpose. For example, Amazon gate way service enables access for enterprise application from on premises with AWS Cloud [32].

Design Principle 4-Big Data storage in both open file format and specific file formats

Based on the research it is found that Big Data systems should support different file formats. Data is being generated in different formats across industries. Systems generate log files, Text files, Excel data, Web application create Jason and XML formats. Consequently, storage should able to load all these different file formats. A key rule under this principle is that computation components also should able to process these file formats. AWS and GCP platforms are capable of storing and computing wide variety of file formats.

Design Rule 4.1-The Compute/Process components and analytical components must able to process files of different formats

This design rule is centred around the ability to store different formats, as any failure to compute these files will go against the purpose of Big Data platforms. So, it is important to ensure that computation, processing and analytical components can process all relevant file formats. For example, in AWS, EMR can work with files in S3FS, EC2 with local disk for HDFS files [33]. Default Hadoop takes '.TXT' file format but Hadoop interface inputFormat can process other custom formats. This is similar to GCP cloud DataProc because it runs on HDFS. On other hand GCP Cloud Dataflow which uses Apache Beam as an underlying technology uses PCollection to custom file formats for processing [16].

Design Principle 5-Assuring data redundancy in Big Data platform

Research identifies that redundancy of data needs to be ensured in cloud-based platforms. Any Big Data platform should ensure the redundancy of data. Unlike transactional systems where redundancy of data is a problem in storage, analytical systems in distributed environment place an emphasis on the redundancy of data. Because in distributed environment data files are split into blocks and stored across the machines in network. In case of failure with one machine, there need to be back up data/redundancy to ensure that this data is not lost. Any loss of data is a loss of value and veracity in Big Data. By default, AWS and GCP ensure replication of data stored in their cloud platforms.

AWS kinesis is good example when discussing redundancy. Amazon kinesis is a distributed messaging queue in which data is injected [34]. When data is ingested, streams are recorded in shards which is uniquely identified sequence of data [34]. AWS Kinesis does not loose data because it is replicated over the nodes automatically ensuring redundancy.

Design Principle 6-Data computation needs to be executed on local storage

Computation is the most important phase in Big Data analytics. Based on our research it was identified that computation should be performed on the local disk of machine, or computation instances of cloud should able to read data from storage with fault tolerance. In distributed processing there is chance for failure of one or more jobs which could result in an incomplete output. As illustrated in Figure 6, fault tolerance in computation is critical. According to research by Vaishnavi V, et al. [8] fault tolerance is one of the highest prioritised area in Big Data. So Big Data platforms should able to perform computation with fault tolerance. Hence a key rule that is associated with this principle is coordination between data units, processing and task failure prevention.

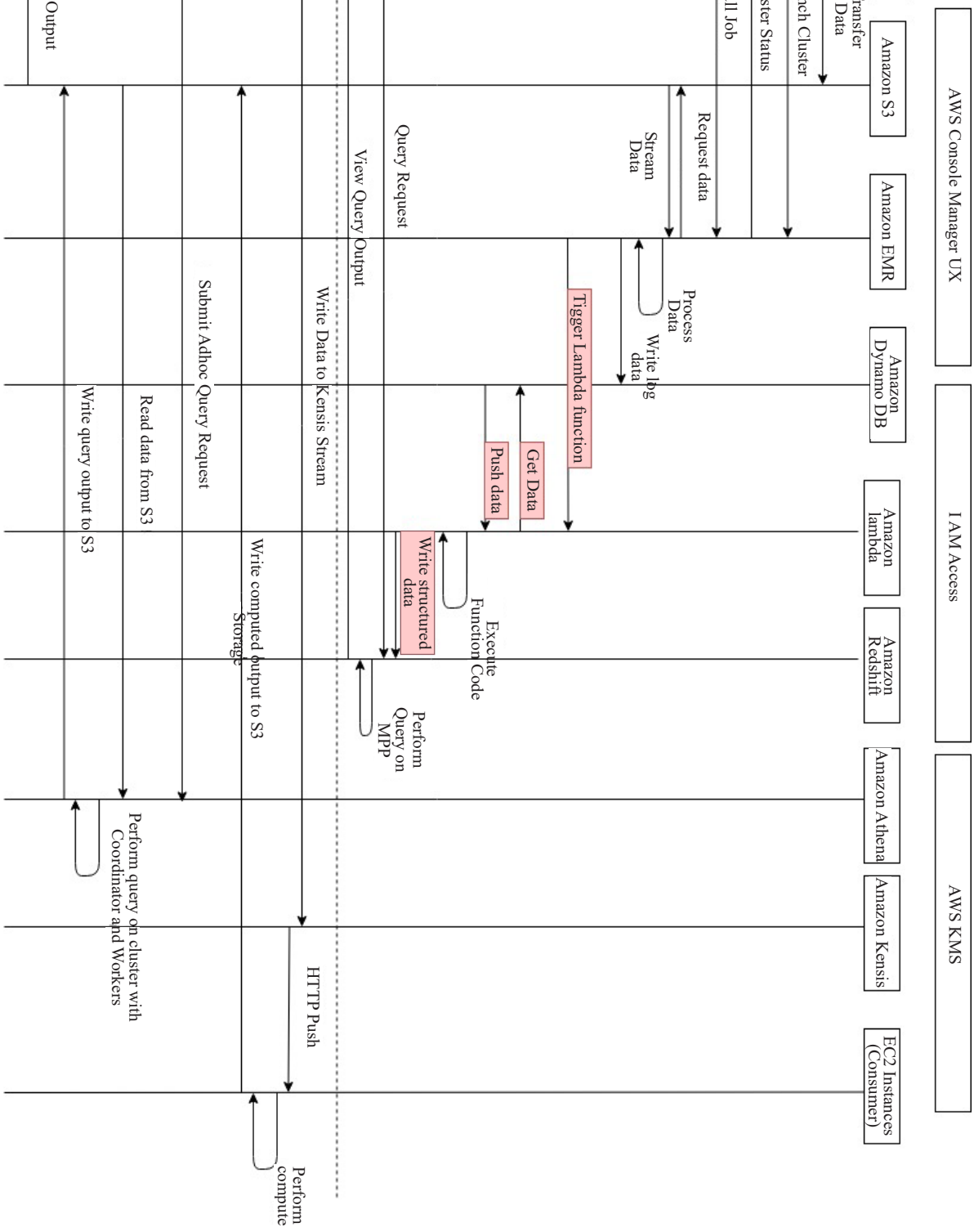

Figure 9. AWS data analytics process illustration

Design Rule 6.1-Coordination between the data blocks and job processing must be clearly established

Coordination between processing and data units is critical. There needs to be a monitoring method in place to ensure this. Good examples for this are AWS, EMR and GCP Cloud data proc which works with Hadoop and its system. Figure 9 illustrates Hadoop YARN when used in AWS, it also details the YARN process. In AWS as illustrated in Figure 9, when the Hadoop cluster is created, nodes are created from preconfigured EC2 instances with 'instance store' which serves as local storage [33]. A disadvantage here is the data life is limited to the computation instance lifetime. Users need to ensure that the program used to write the data to other location, is done before the computation process completes. However, by using EMRFS which an implementation of HDFS, EMR can directly read or write data from Amazon S3 storage [33] As shown in Figure 6, EMRFS/HDFS are depicted separately however in real time they are available along with EC2 instances. Coordination between data storage and processing needs to ensured. After ensuring this, when a user runs Hadoop in EMR, it uses YARN (Yet Another Resource Negotiator). YARN aids the resource manager, to ensure the coordination between processing, and data units [35]. According to research done by Apache [36] in a cluster, coordination between different processing units and data blocks is highly essential, in order to gain fault tolerance in Bigdata platforms.

Design Rule 6.2-Resilient computation to provide mitigation against job failures

When jobs are running in a cluster across thousands of nodes, it is expected that there will be failures. Computation processing should ensure to re-run failed jobs, in order to mitigate this. For example, in GCP cloud data proc which runs on the Hadoop system uses YARN. The application manager in YARN receives the status from job containers about each task's execution. In case of a failure, the application manager will use alternate computing resources and re-execute the job [35].

Design Principle 7-The Compute and Process layer must handle the distributed-parallel processing on the Big Data platform

Based on the research, it was identified that computation and processing layer should support parallel processing. The processing of data can also be performed using a queue using one machine processing these data units. Implementing a similar way of processing would be time consuming and expensive. However Big Data platforms are able to perform computation parallel across distributed machine. Key rules under this principle are that the data needs to be stored in distributed manner and concurrency in processing data units. In both AWS and GCP all the computational components perform parallel and distributed processing. To provide specific examples, AWS redshift and GCP BigQuery are good examples of doing processing data in distributed platforms parallel. At an abstract level both tools work in similar ways.

Design Rule 7.1-Loading data in distributed manner in order to balance workloads

In Big Data analytics the volume of data will be in petabytes with billions of records. So even if the data is stored in one machine with high configuration I/O processes this will require a vast amount of processing power which can potentially hang the system. Hence work load balancing of data requires storage in a distributed environment.

For example, AWS Redshift is a powerful data warehousing platform which uses a massive parallel processing technology for performing analytics for fast execution of query input [15]. When data is moved to Redshift, data is split and records are return in columnar way. Data is stored across compute nodes in distributed manner [37]. The advantage of storing in columnar way in distributed manner, is that faster I/O processing can be achieved.

Design Rule 7.2-Many parallel computation instances could process the data concurrently

In the processing layer, it is important to run computation concurrently. This helps with performance and the speed of processing. For example, AWS redshift delivers quick query output virtually on any size of data with help from column storage and the parallel processing of query's through executing it across distributed data which is stored in multiple nodes [37]. Users submits queries through the Leader node which parses and develops an execution plan

[38]. This is then pushed as compiled query code to compute nodes where the data resides along with dedicated Central Processing Unit (CPU) and memory. Compute nodes have node slices which are allocated with a portion of node memory and disk space. Query are performed in these compute nodes concurrently and results are shared back with leader node and then to user (cf. Amazon Redshift Overview Internal Architecture and System Operation).

Design Principle 8-Memory based compute process for higher efficiency in performance

Based on our research work it was identified that Big Data platforms should support 'in memory computation'. Saving and retrieving data from memory instead of writing and reading it from disk will improve the performance of processing time during computation. We observed both AWS and GCP platforms, they allow creation instances with different types like memory based (High Memory) and compute based (High CPU). Analytical tools such as AWS Athena are interactive analytical tools which uses Presto, a distributed SQL engine [39]. Presto solely uses in memory processing which is pipelined across the network. The key rule under this principle is the ability of the platform to store intermediate data in memory. By logical reasoning, it is evident that since AWS, EMR and GCP cloud data proc are able to run on an Apache Spark server (a memory-based processing engine, they would be inter-operable. If the cloud Big Data platform cannot support memory-based computation by intermediate data capturing in memory it cannot be compatible with these sophisticated tools.

Design Rule 8.1-Transitional data should be made available from the cache (memory) for analysis

Allowing data to be saved and accessed from memory helps to lower latency. For example, when we observed map reduce jobs, results are written to HDFS file system and retrieved again. A large quantity of disk read and writes will increase latency and performance. Research by Apache [36] mentioned memory-based data processing as a design principle in their work, because it uses Random Access Memory (RAM) for data storage rather than local disk. When Apache spark is installed in AWS or GCP, memory-based computing capacity in these solutions are also scalable. When Apache spark is installed in AWS or GCP computing instances will be created. When user run the program, this performs a driver/master process that converts programming code into Directed Acyclic Graph (DAG). After series of steps DAG is converted into an execution plan, which creates execution units that are tasks at each stage. From this driver processes request resources from cluster manager (YARN, Mesos). The cluster manager then launches executions on behalf of the driver to finish the task. Spark processing collects data items called 'Resilient Distributed data sets' which are split into partitions and stored on memory of worker nodes which are AWS EC2 or GCP VM (Virtual Machines) which are computing instances.

Design Principle 9-Data Security assurance is necessary to avoid breaches in system and unauthorised access

Research work has identified that security of the data must be ensured by Big Data platforms. Security is very sensitive issue. The public impression is still that cloud systems are not secure for the data, as third-party environment or systems which is not under direct control of organisation. According to research by Amazon Web Services [40] challenges of organisational management includes security of data, privacy, governance and ethical aspects. The key rule underlies this principle is data encryption, authorisation and account management.

Design Rule 9.1-Authorisation, Data Encryption, and User Account Management are essential features of Big Data platforms

Even though authorisation and access management are key security rules, data encryption can be rated as top priority among all cloud-based platforms, as cloud systems are off premises and completely under control of the vendor. In cloud computing it is agreed that there will be a large amount data transfer which occurs over the intranet which coordinates the computation process. Any breach of network which exposes data is potential threat. Cloud platforms also need to ensure end to end security of data throughout the data pipeline. For example, AWS provides two level security features such as server-side data encryption and client-side data encryption [37]. Under server-side encryption, it allows users to request encryption of data in Amazon S3 storage before being saved on the disks of data centres. Client-side encryption allows users to load the encrypted data in to Amazon S3 however user has to be responsible for

managing encryption keys and security tools [37]. GCP also provides extensive data security features, to protect data at three states like encryption at rest encryption, in transit and encryption in use [41]. Both AWS and GCP provide a 'Key Management' service as well Identity and Access Management tools like (IAM).

Design Principle 10-Effective User Experience (UX) to configure, install and manage Big Data systems is needed

Research has found that management of Big Data platforms requires a user friendly UX interface. Big Data platforms have different layers of architecture and comprise of different tools. The management of Big Data platforms in terms of installation, configuration and data management is not an easy job. Key rules that associate with this principle are user friendly management tools and Meta data management.

Design Rule 10.1-Customer-centric Platform Operations Management functionalities like monitoring, cloud management, performance metrics

Since cloud vendors charge users based on pay for use, it is important to have monitoring, performance management and cloud management tools. AWS achieves this with help of AWS console manager and GCP by providing Google Cloud Console. With the help of these tools user can access computation tools and be able to manage them.

Design Rule 10.2-Metadata management like capturing and governance should be available

Metadata in simple terms, is the data that defines data. Metadata management is very crucial in data management, as it helps in determining the source and data types. When observed in AWS and GCP there is no direct tool that enables metadata management. However, metadata management is emphasised and made available through different ways in both Big Data platforms. For example, In GCP every computing instance captures metadata on metadata server that can be accessed programmatically with the help of the compute engine Application Programming Interface (API) [42]. Likewise, in AWS metadata management is possible through data cataloguing process as illustrated in Figure 10. In AWS with help of comprehensive data catalogue process metadata can be extracted from Amazon S3 with help of Amazon Lambda and queried, as shown in Figure 8 [43]. Also, AWS uses AWS Glue for creating Hive compatible meta-store for the data in Amazon S3 (See Appendix, Data Cataloging).

With reference to multiple use cases and architectural patters the overall analytics processes of AWS and GCP are illustrated below in Figures 9 and 10 respectively. Sample architectural patterns can be identified from Table 1.

Table 1. Architectural patterns and Platforms

Architectural pattern	Platform
Build a Healthcare Data Warehouse Using Amazon EMR, Amazon Redshift, AWS Lambda, and OMOP	AWS [44]
Create a Healthcare Data Hub with AWS and Mirth Connect	AWS [45]
Architecture: Real-Time Stream Processing for IoT	GCP [46]

4.2 *AWS Analytics process scenario*

A sample scenario when using AWS with raw data with the objective of making it ready for analytics through converting this data to a tabular form or any other custom data formats, this data can be transferred to Amazon S3 storage. It is important to remember the user interaction with AWS happens through AWS console Manager. Data that has been stored in Amazon S3 will be read by Amazon EMR (Cluster of virtual EC2 compute instances) during the computation process. The output of this process can be moved to Amazon DynamoDB which is NoSQL database. With help of Amazon lambda function, it can be transformed into desired custom format and copied to Amazon Redshift

database which acts like a data warehouse.

Another sample scenario is where the objective is to perform real time analytics in AWS. Data can be feed into Amazon Kinesis which is meant for streaming analytics. EC2 compute instances will perform the required action to save the data into a file which is then uploaded to Amazon S3. From this Amazon Athena can be used to perform query analytics on the data files stored in Amazon S3 in previous step and write outputs at this stage again back to Amazon S3 from where users can read the final results.

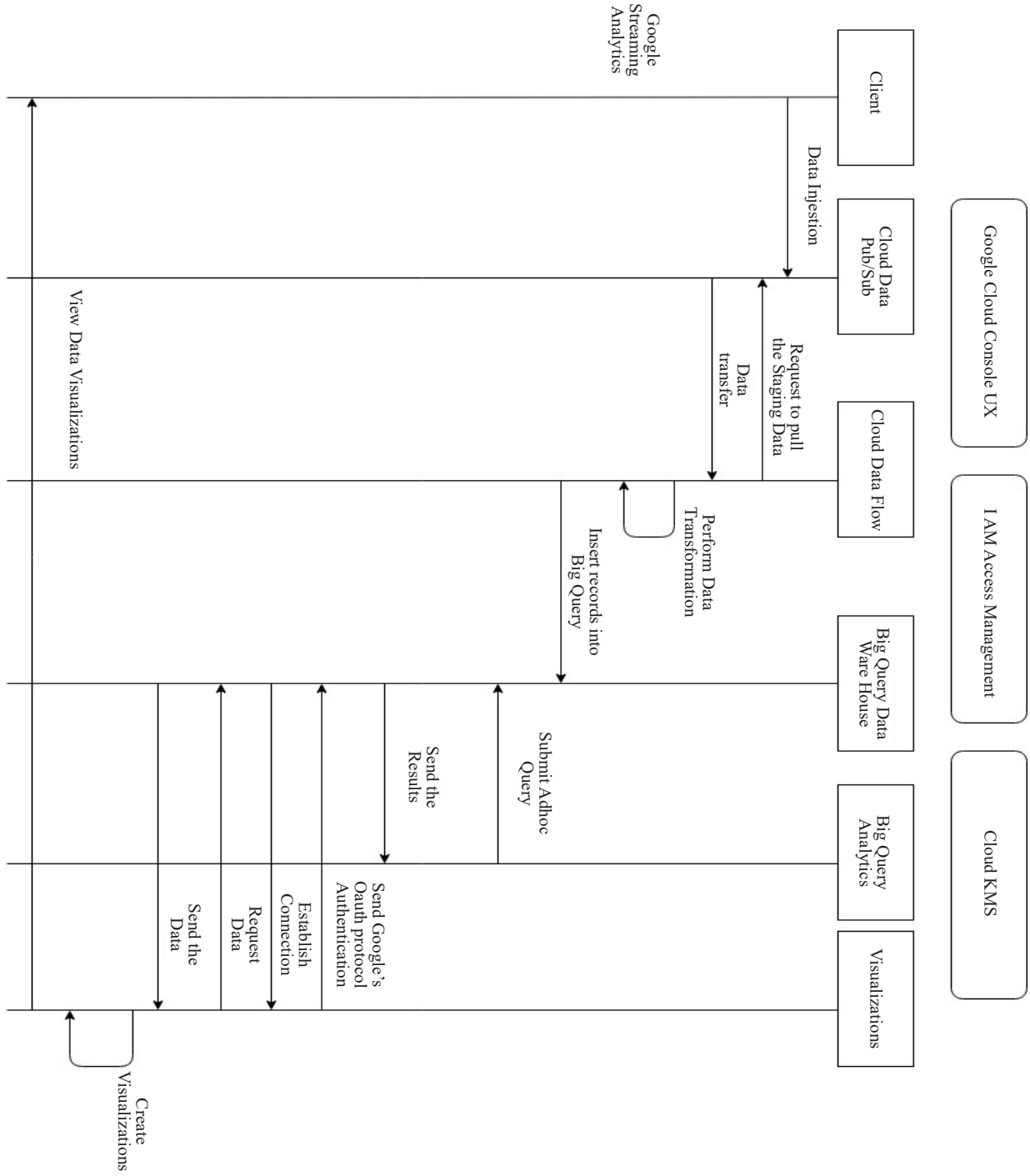

Figure 10. GCP data analytics process illustration

4.3 *GCP Big Data analytics process scenario*

Let's take for example, the task of performing streaming data analytics using GCP. Data may be fed into Cloud pub/sub which works on messaging architecture. This data will then be pushed to Cloud Dataflow which works on underlying technology from Apache Beam (Programming model of Batch and Streaming data processing) to perform the computation to transform the data into structured format and writes output to Big Query. The Big Query database can be viewed as a data warehouse for Google, which allows users to submit queries and retrieve results.

5. Concluding remarks

Big Data's prominence has been realized since the early classic papers by Hashem and his co-workers [6] and Chen and Zhang [22]. It has become part of the digital life-style and is gaining traction with powerful cloud, IoT, AI, and context-aware technologies. To recap, the RE-DSR approach used in this paper was effective in extracting ten design principles and nested rules from leading Cloud-based Big Data artifacts. It is analogous but not similar to the Knowledge Discovery from Databases (KDD) methodology used in data mining where patterns and rules are extracted from input datasets. In this paper and the parallel project described previously [9], we have used design specifications from artifacts in production (cloud-based big data platforms) as input to abductively reason how they address an IS problem (inter-operability and heterogeneity). Using the categorization from Vaishnavi and Kuechler in their classic text [8], such experimentation by observing current cloud-based platforms allows us to reverse design the platform specifications that are improvements over existing systems [39]. Following from this, logical reasoning, while weaker than mathematical proofs or experimentation, nevertheless supplements the discovery of design knowledge by constantly revisiting the context of cloud-based Big Data platforms and deducing rules that support inter-operability. On the basis of replication, we claim validity of process (RE-DSR) as well as construct (design principles and rules) because similar but not identical rules were derived from Hadoop-based systems [9]. This theoretical work will be helpful in developing a quick understanding of the ideas surrounding cloud-based Big Data platforms. The design rules provided can aid cloud practitioners as well as researchers in understanding and addressing the heterogeneity challenge on such platforms.

This paper seeks to support both client and service-provider decision making process for the adaption or later modification of existing Big Data systems which is a typical DevOps function. With help of the design principles presented, practitioners and managers can incorporate systems flexibility, interoperability, agility and compatibility. Design research in Big Data systems is currently limited; we suggest that this paper contributes to the knowledge base for fruitful research on the design of Big Data platforms.

As a practical contribution, the synthesis of RE-DSR approach has shown to be applicable and useful when presented to an audience of cloud architects as part of a seminar. This framework may be used by Cloud research or DevOps teams when analyzing complex or extensive IT systems.

The validation of the design principles and rules presented in this research work is largely thought based, through the use of logical reasoning [8]. The design artifacts were published, non-proprietary material based on secondary data such as project documentation, training tutorials, white paper documents from vendors and other web-based information. They were used to extract the higher level of understanding about the platforms. We believe much more materials exist in the hidden intranets of the vendors which we could not access. As well, certified cloud architects would serve as the ideal participants in a validation workshop that could further improve the set of principles and rules. This is suggested as further research.

In closing, the significance of Big Data research is increasing rapidly with the growth of cloud, edge and P2P computing. With such evolving technologies, challenges are also presented, which have scope for conducting RE-DSR studies. The current research work may be extended by practically implementing the design rules in real time environments and assessing the performance of platforms with real-time dashboards.

Conflict of interest

The authors declare no conflict of interest.

References

[1] Reinsel D, Gantz J, Rydning J. *The Digitization of the World-From Edge to Core*. Seagate, International Data Corporation (IDC). Report number: #US44413318, 2018.

[2] Marr B. *How Much Data Do We Create Every Day? The Mind-Blowing Stats Everyone Should Read.* Available from: https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/#605d605e60ba [Accessed 14th September 2019].

[3] Statistical Analysis System (SAS). *Big Data What it is and why it matters*. Available from: https://www.sas.com/en_nz/insights/big-data/what-is-big-data.html [Accessed 14th September 2019].

[4] Sagiroglu S, Sinanc D. Big Data: A review. *2013 International Conference on Collaboration Technologies and Systems (CTS)*. Piscataway, NJ, USA: IEEE; 2013. p. 43-47.

[5] Gartner. *Gartner Survey Says Cloud Computing Remains Top Emerging Business Risk*. Available from: https://www.gartner.com/en/newsroom/press-releases/2018-08-15-gartner-says-cloud-computing-remains-top-emerging-business-risk [Accessed 16th November 2019].

[6] Hashem IAT, Yaqoob I, Anuar NB, Mokhtar S, Gani A, Khan SU. The rise of "big data" on cloudcomputing: Review and open research issues. *Information Systems*. 2015; 47: 98-115.

[7] Peffers K, Tuunanen T, Rothenberger MA, Chatterjee S. A design science research methodology for information systems research. *Journal of Management Information Systems*. 2007; 24(3): 45-78.

[8] Vaishnavi V, Kuechler W. *Design Science Research Methods and Patterns: Innovating Information and Communication Technology*. New York, US: CRC Press; 2015.

[9] Sharma RS, Wingreen SC, Janarthanan SBT. Design principles for hadoop-based platforms: A reverse-engineered design-science approach. *Journal of Management Information and Decision Sciences.* 2022; 25(2): 1-19. Available from: https://www.abacademies.org/articles/design-principles-for-hadoopbased-platforms-a-reverseengineered-designscience-approach.pdf [Accessed 14th January 2022].

[10] Yaqoob I, Hashem IAT, Gani A, Mokhtar S, Ahmed E, Anuar NB, et al. Big data: From beginning to future. *International Journal of Information Management.* 2016; 36(6): 1231-1247. Available from: doi: 10.1016/j.ijinfomgt.2016.07.009.

[11] Chen M, Mao S, Liu Y. Big Data: A survey. *Mobile Networks and Applications*. 2014; 19: 171-209. Available from: doi: 10.1007/s11036-013-0489-0.

[12] Jin X, Wah BW, Cheng X, Wang Y. Significance and challenges of Big Data research. *Big Data Research.* 2015; 2(2): 59-64. Available from: doi: 10.1016/j.bdr.2015.01.006.

[13] McKinsey Global Institute. *The age of analytics: competing in a data-driven world*. Available from: https://www.sipotra.it/wp-content/uploads/2017/01/THE-AGE-OF-ANALYTICS.pdf [Accessed 14th September 2018].

[14] Begoli E, Horey J. Design principles for effective knowledge discovery from Big Data. *2012 Joint Working IEEE/IFIP Conference on Software Architecture and European Conference on Software Architecture*. Piscataway, NJ, USA: IEEE; 2012. p. 215-218.

[15] Hu H, Wen Y, Chual TS, Li X. Toward scalable systems for Big Data analytics: A technology tutorial. *IEEE Access*. 2014; 2: 652-687. Available from: doi: 10.1109/ACCESS.2014.2332453.

[16] Assunção MD, Calheiros RN, Bianchi S, Netto MA, Buyyab R. Big Data computing and clouds: Trends and future directions. *Journal of Parallel and Distributed Computing*. 2014; 79-80: 3-15. Available from: doi: 10.1016/j.jpdc.2014.08.003.

[17] Gandomi M, Haider T. Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management*. 2015; 35(2): 137-144. Available from: doi: 10.1016/j.ijinfomgt.2014.10.007.

[18] Tsai CW, Lai CF, Chao HC, Vasilakos A. Big data analytics: A survey. *Journal of Big Data*. 2015; 2(21): 1-32. Available from: doi: 10.1186/s40537-015-0030-3.

[19] Gartner. *Gartner Says Business Intelligence and Analytics Leaders Must Focus on Mindsets and Culture to Kick Start Advanced Analytics*. Available from: https://www.gartner.com/en/newsroom/press-releases/2015-09-15-gartner-says-business-intelligence-and-analytics-leaders-must-focus-on-mindsets-and-culture-to-kick-start-advanced-analytics [Accessed 14th September 2019].

[20] Hevner R, Ram S, March ST, Park J. Design science in information systems research. *MIS Quarterly*. 2004; 28(1): 75-105. Available from: doi: 10.2307/25148625.
[21] Brunelière H, Cabot J, Dupé G, Madiot F. MoDisco: A model driven reverse engineering framework. *Information and Software Technology*. 2014; 56(8): 1012-1032. Available from: doi: 10.1016/j.infsof.2014.04.007.
[22] Chen C, Zhang CY. Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Information Sciences*. 2014; 275: 314-347. Available from: doi: 10.1016/j.ins.2014.01.015.
[23] AWS Whitepaper. *Overview of Amazon Web Services*. Available from: https://docs.aws.amazon.com/whitepapers/latest/aws-overview/amazon-web-services-cloud-platform.html [Accessed 19th November 2019].
[24] Google Cloud. G*oogle Cloud Overview*. Available from: https://cloud.google.com/docs/overview/ [Accessed 14th September 2019].
[25] Hadoop. *Apache Hadoop YARN*. Available from: https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html [Accessed 13th November 2018].
[26] Sivarajah U, Kamal MM, Irani Z, Weerakkody V. Critical analysis of Big Data challenges and analytical methods. *Journal of Business Research*. 2017; 70: 263-286. Available from: doi: 10.1016/j.jbusres.2016.08.001.
[27] Provost F, Fawcett T. Data science and its relationship to Big Data and Data-Driven decision making. *Big Data*. 2013; 1(1): 51-67. Available from: doi: 10.1089/big.2013.1508.
[28] Wamba SF, Akter S, Edwards A, Chopin G, Gnanzou D. How 'big data' can make big impact: Findings from a systematic review and a longitudinal case study. *International Journal of Production Economics*. 2015; 165: 234-246. Available from: doi: 10.1016/j.ijpe.2014.12.031.
[29] Demchenko Y, Laat CD, Membrey P. Defining architecture components of the Big Data Ecosystem. *2014 International Conference on Collaboration Technologies and Systems (CTS)*. Piscataway, NJ, USA: IEEE; 2014. p. 104-112.
[30] Jagadish H, Gehrke J, Labrinidis A, Papakonstantinou Y, Patel JM, Ramakrishnan R, et al. Big Data and its technical challenges. *Communications of the ACM*. 2014; 57(7): 86-94. Available from: doi: 10.1145/2611567.
[31] Saggi MK, Jain S. A survey towards an integration of big data analytics to big insights for value creation. *Information Processing and Management*. 2018; 54(5): 758-790. Available from: doi: 10.1016/j.ipm.2018.01.010.
[32] Belangour EA, Tragha A. Digging into Hadoop-based Big Data Architectures. *International Journal of Computer Science Issues*. 2017; 14(6): 52-59. Available from: doi: 10.20943/01201706.5259.
[33] Kambatlaa K, Kollias G, Kumar V, Gramaa A. Trends in big data analytics. *Journal of Parallel and Distributed Computing*. 2014; 74(7): 2561-2573. Available from: doi: 10.1016/j.jpdc.2014.01.003.
[34] Talia D. Clouds for scalable Big Data analytics. *Computer*. 2013; 46(5): 98-101. Available from: doi: 10.1109/MC.2013.162.
[35] Ji CQ, Li Y, Qiu WM, Awada U, Li K. Big Data Processing in Cloud Computing Environment. *2012 International Symposium on Pervasive Systems, Algorithms and Networks*. Piscataway, NJ, USA: IEEE; 2012. p. 17-23.
[36] Apache. *Apache Hadoop*. Available from: https://hadoop.apache.org/ [Accessed 13th November 2018].
[37] Google Cloud. *Data Lifecycle*. Available from: https://cloud.google.com/solutions/data-lifecycle-cloud-platform#ingesting_streaming_data [Accessed 14th September 2019].
[38] Gartner. *Gartner Survey Shows More Than 75 Percent of Companies Are Investing or Planning to Invest in Big Data in the Next Two Years*. Available from: https://www.gartner.com/en/newsroom/press-releases/2015-09-16-gartner-survey-shows-more-than-75-percent-of-companies-are-investing-or-planning-to-invest-in-big-data-in-the-next-two-years [Accessed 13th November 2018].
[39] Garg M, Jindal MK. Reverse engineering-roadmap to effective software design. *International Journal of Recent Trends in Engineering*. 2009; 1(2): 186-188.
[40] Amazon Web Services (AWS). *What is Amazon Athena?* Available from: https://docs.aws.amazon.com/athena/latest/ug/what-is.html [Accessed 14th September 2019].
[41] Google Cloud. *Cloud Data Transfer*. Available from: https://cloud.google.com/products/data-transfer/ [Accessed 14th September 2019].
[42] Google Cloud. *Cloud SDK*. Available from: https://cloud.google.com/sdk/ [Accessed 14th September 2019].
[43] Google Cloud. *Data Lifecycle*. Available from: https://cloud.google.com/solutions/data-lifecycle-cloud-platform#ingesting_streaming_data [Accessed 14th September 2019].
[44] Amazon Web Services (AWS). *AWS Big Data Blog*. Available from: https://aws.amazon.com/blogs/big-data/build-a-healthcare-data-warehouse-using-amazon-emr-amazon-redshift-aws-lambda-and-omop/ [Accessed 14th September 2019].
[45] Amazon Web Services (AWS). *Create a Healthcare Data Hub with AWS and Mirth Connect*. Available from:

https://aws.amazon.com/blogs/big-data/create-a-healthcare-data-hub-with-aws-and-mirth-connect/ [Accessed 14th September 2019].
[46] Google Cloud. *Cloud DATAPROC*. Available from: https://cloud.google.com/dataproc/ [Accessed 14th September 2019].

Appendix

Table A1. Platform specifications documents for reverse engineered design research

Amazon Web Services	Google Cloud Platform
Overview of Amazon Web Services AWS Whitepaper. Amazon web services. https://docs.aws.amazon.com/whitepapers/latest/aws-overview/amazon-web-services-cloud-platform.html [Accessed 19th November 2019].	Google Cloud Platform Overview. https://cloud.google.com/docs/overview/ [Accessed 14th September 2019].
Amazon EC2 for Microsoft Windows Server. https://aws.amazon.com/windows/products/ec2/ [Accessed 19th November 2019].	Installing the Linux Guest Environment. https://cloud.google.com/compute/docs/instances/linux-guest-environment [Accessed 14th September 2019].
Tools for Amazon Web Services. https://aws.amazon.com/tools/ [Accessed 19th November 2019].	Cloud Data Transfer. https://cloud.google.com/products/data-transfer/ [Accessed 14th September 2019].
Introducing AWS Import/Export for Physical Data Transfer. https://aws.amazon.com/about-aws/whats-new/2009/05/20/AWS-Import-Export/ [Accessed 19th November 2019].	Cloud SDK. https://cloud.google.com/sdk/ [Accessed 14th September 2019].
AWS Case Study: Financial Times. https://aws.amazon.com/solutions/case-studies/financial-times/ [Accessed 19th November 2019].	Data Lifecycle. https://cloud.google.com/solutions/data-lifecycle-cloud-platform#ingesting_streaming_data [Accessed 14th September 2019].
Work with Storage and File Systems. https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-plan-file-systems.html [Accessed 19th November 2019].	Cloud DATAPROC. https://cloud.google.com/dataproc/ [Accessed 14th September 2019].
Introducing Storage Gateway. https://aws.amazon.com/storagegateway/ [Accessed 19th November 2019].	Map AWS services to Google Cloud Platform products. https://cloud.google.com/free/docs/map-aws-google-cloud-platform [Accessed 14th September 2019].
Work with Storage and File Systems. https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-plan-file-systems.html [Accessed 19th November 2019].	Programming Model for Cloud Dataflow SDK 2.x. https://cloud.google.com/dataflow/docs/concepts/beam-programming-model [Accessed 14th September 2019].
Kinesis Data Streams Concepts. https://docs.aws.amazon.com/streams/latest/dev/key-concepts.html [Accessed 19th November 2019].	Architecture: Real-Time Stream Processing for IoT. https://cloud.google.com/solutions/architecture/real-time-stream-processing-iot [Accessed 14th September 2019].
Amazon Redshift System Overview Performance. https://docs.aws.amazon.com/redshift/latest/dg/c_high_level_system_architecture.html [Accessed 19th November 2019].	Encryption in Transit in Google Cloud.
Amazon Redshift Overview Internal Architecture and System Operation. https://docs.aws.amazon.com/redshift/latest/dg/c_internal_arch_system_operation.html [Accessed 19th November 2019].	Storing and Retrieving Instance Metadata. https://cloud.google.com/compute/docs/storing-retrieving-metadata [Accessed 14th September 2019].
Protecting Data Using Encryption. https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingEncryption.html [Accessed 19th November 2019].	AirAsia: Turning to Google Cloud to refine pricing, increase revenue, and improve customer experience. https://cloud.google.com/customers/airasia/ [Accessed 14th September 2019].
What is Amazon Athena? https://docs.aws.amazon.com/athena/latest/ug/what-is.html [Accessed 19th November 2019].	
Data Cataloging. https://docs.aws.amazon.com/aws-technical-content/latest/building-data-lakes/data-cataloging.html [Accessed 19th November 2019].	
AWS Big Data Blog. https://aws.amazon.com/blogs/big-data/build-a-healthcare-data-warehouse-using-amazon-emr-amazon-redshift-aws-lambda-and-omop/ [Accessed 19th November 2019].	
Create a Healthcare Data Hub with AWS and Mirth Connect. https://aws.amazon.com/blogs/big-data/create-a-healthcare-data-hub-with-aws-and-mirth-connect/ [Accessed 19th November 2019].	