IMPLEMENTING CLOUD BASED BIG DATA PLATFORMS–A CASE USING MICROSOFT AZURE

Conference Paper · October 2018

CITATIONS
0

READS
1,077

2 authors:

Soffi Westin
Agder Energi, Kristiansand, Norway
6 PUBLICATIONS **33** CITATIONS
SEE PROFILE

Karen Stendal
University of South-Eastern Norway
28 PUBLICATIONS **286** CITATIONS
SEE PROFILE

IMPLEMENTING CLOUD BASED BIG DATA PLATFORMS – A CASE USING MICROSOFT AZURE

Karen Stendal
University of South-Eastern Norway
karen.stendal@usn.no

Soffi Westin
University of Agder
soffi.westin@uia.no

ABSTRACT

Digital transformation and implementation of big data platforms are inevitable in any industry. Big data constitutes an important area of research, however, implementation of platforms like Microsoft Azure have yet to be explored. Through a narrative case study, we aim to explore the implementation of such big data platforms in the power industry. Our case is based in a Norwegian power company who are early movers in implementing Microsoft's Azure platform across multiple units in the organization. With the support of top management and eager business units one would expect this process to be fairly straight forward. Our findings show that the maturity of the technology, in addition to challenges of being an early mover, create an unexpected path to success.

1. INTRODUCTION

There is a Norwegian mountain advice that say to listen to experienced hikers when heading out for a hike. This is good advice independent of context, listen to those who have the experience when venturing into new areas. But what do you do when the experience is lacking or not existing?

Digitalization and digital transformation are topics highly covered through research (Bouwman, Nikou, Molina-Castillo and de Reuver, 2018; Kane, Palmer, Phillips, Kiron and Buckley, 2015; Timonen and Vuori, 2018; Weichenrieder, 2018). We are the population of the information age, where the collection of data are never ending and one of the major topics are big data (Hashem, Yaqoob, Anuar, Mokhtar, Gani and Khan, 2015).

Big data as a term is a fairly new term, but the challenges connected to big data is not new (Hashem et al., 2015). Early challenges imply that the amount of data is too large for the technology to store, manage, and process efficiently (Hashem et al., 2015). Big data provides us with a stream of new and digital data on how people, organizations and society interact (Blazquez and Domenech, 2018), which organizations can use to a deeper understanding of their customers, suppliers and collaborators.

Due to the vast amount of data cloud based big data platforms are adopted and implemented into organizations worldwide (Hashem et al., 2015). Platforms, such as Microsoft Azure, are continuously develop to facilitate organizational needs to store and analyze the data (Copeland, Soh, Puca, Manning and Gollob, 2015).

Implementation of technology into an organization is traditionally seen as being dependent on factors like top management support, system champion, good project management, and a clear goal (Jiang, Klein and Balloun, 1996; Reitsma and Hilletofth, 2017). Through this research we aim to explore how organizations who are ready and prepared to implement big data platforms experience the process.

Therefor our research question is:

What are the challenges of implementing cloud based big data platforms?

To answer this question, we have conducted a qualitative exploratory research within a Norwegian power company. Through observation, interviews and process documents we have gained a unique view into how the implementation process has been conducted and how the organization have had to create their own path towards successful implementation of a cloud based big data platform.

The remainder of this paper is structured as follows: First we present related literature on cloud computing, big data, implementation of cloud based big data platforms and DevOps. Second, our research method and empirical case are presented. Further, we present the findings from our study, followed by a discussion, conclusion, and implications of this study.

2. RELATED LITERATURE

Through this section we present the key concepts relevant to this research. Further, we are presenting how these concepts relate and increase the complexity of the field.

2.1 Big Data

The term "big data" has grown to be on everybody's lips, it has been called the hottest trend in technology and promise to have great impact on society in various areas (John Walker, 2014). But what does it imply?

Big data has been defined as:

"Big data is eliciting attention from the academia, government, and industry. Big data are characterized by three aspects: (a) data are numerous, (b) data cannot be categorized into regular relational databases, and (c) data are generated, captured, and processed rapidly. Moreover, big data is transforming healthcare, science, engineering, finance, business, and eventually, the society" (Hashem et al., 2015, p. 99).

This shows that big data is more than just the vast amount of data, the four Vs of big data is widely recognized: Volume, Variety, Velocity and Value (Hashem et al., 2015). These four Vs imply the complexity of big data and how difficult it is and will be to deal with this in the future. We are dealing with data sets of a massive scale, which is complex, collected from various sources, transferred at an increasing speed, and contains a hidden value for the data owners (Hashem et al., 2015).

In the power industry, big data has been an important factor for years (Kezunovic, Xie and Grijalva, 2013). Implementation of real-time monitoring of power usage is one of the sources for data collection, and are contributing to the big data challenges in this industry.

2.2 Cloud Computing

Cloud Computing was defined in 2011 by NIST, the US National Institute of Standards and Technology as *"a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction"* (Marinescu, 2017, p. 2).

Cloud computing have received increasing interest from enterprises to support critical business functions, and are considered among the five most influential technologies globally (El-Gazzar, Hustad and Olsen, 2016). Further, cloud computing allows organizations to explore a unique capabilities to move into a competitive position when utilizing IT-solutions at a low cost (El-Gazzar et al., 2016).

Through their study El-Gazzar et al. (2016) indicate there is issues and challenges for organizations when adopting cloud computing. These challenges are categorized as: Security, Availability, Migration, Business, Legal and Ethical, Culture, Awareness, Impact, IT-governance, and Strategy. Such issues make adoption of cloud computing a challenge to any organization, and does not take into account the vast amount of data collected and challenges of big data.

2.3 Cloud Based Big Data Platforms

Knowing the challenges from both big data and cloud computing, why would anybody attempt the put the two together? An important issue to think about is the relationship between big data and cloud computing. Figure 1 presents how this relationship can be seen. Big data and cloud computing are tightly connected (Hashem et al., 2015). Cloud computing provides the underlying engine that are used for analyzing and store large dataset, collected from various sources. Big data utilize distributed storage technology through cloud computing, instead of local computer power. Hashem et al., (2015) indicates that cloud computing is not only a facilitator for big data, but also can be seen as a service model.

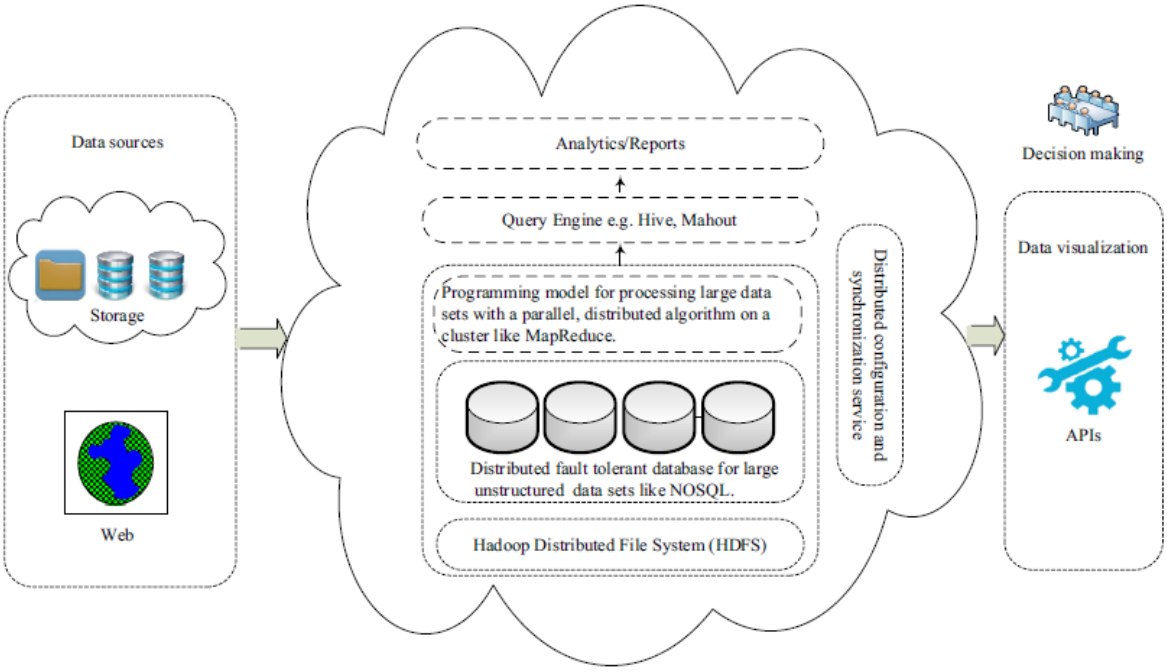

Figure 1 Cloud computing usage in big data (Hashem et al., 2015, p. 103)

The main vendors of cloud based big data platforms is Google, Microsoft Amazon, and Cloudera (Hashem et al., 2015). In the context of this research we focus on Microsoft Azure. Microsoft Azure is the brand name for Microsoft's cloud computing services and was introduced in 2008 (Copeland et al., 2015).

To our knowledge, the implementation process of cloud based big data platform has received little focus from the research community.

2.4 DevOps

DevOps is a combination of Development and Operations (Jabbari et al, 2016) and aims at bridging the gap between the two. Jabbari, Ali, Petersen and Tanveer (2016) defines DevOps in Chapter 5 as: *"a development methodology [] aimed at bridging the gap [] between Development [] and Operations [], emphasizing communication and collaboration [], continuous integration [], quality assurance [] and delivery [] with automated deployment [] utilizing a set of development practices."*

By utilizing DevOps one example shows how infrastructure can be maintained just in form of code (Virmani, 2015). Another example describes how you integrate both the development and operation environment seamlessly through utilizing the client-side integrated development environment (IDE) on the one side, and the server-side service portfolio and cloud controller on the other. This way the developer

may test their applications in large scale on their own computers, deploying it to the cloud where it will be automatically monitored and provide feedbacks for the developers (Hosono, 2012).

This is different from the "normal" handover of software process where developers made the applications or the integrations between various objects in an infrastructure, and then handed the solutions over to operators who performed maintenance and support (Khan, 2013). However, Qazi and Bashir (2018, p. 1) point out the dilemma of handing over software to stakeholders "who was not a part of the process before".

3. RESEARCH METHOD

To answer our research question we have conducted an exploratory qualitative case study. Qualitative research is appropriate when we need a complex and detailed understanding of an issue that is new or poorly understood (Creswell, 2007). A case study is applied because the issues under study are processes very much linked to their contexts (Hartley, 2004). The case we studied was purposely and theoretically sampled (Eisenhardt, 1989). One of the authors are employed by Agder Energi (AE).

3.1 Case – Agder Energi

Agder Energi (AE) is Norway's third largest energy group in terms of hydroelectric production. The group's 49 wholly and partly-owned power stations produce around 8.1 TWh of renewable energy on an annual basis.

Energy management and trading is AE's core activity and for that there are three critical conditions related to data consumption:

1. The need for a vast amount of data collected from a diversity of internal and external data sources, and a possibility to add new sources in order to perform more and improved analyzes.
2. Short time from making change orders to the ICT-department to operation
3. High uptime is required for the applications

The key issue is to be in the forefront of, and better than, your competitors. In trading there are small time windows to operate within, hence any solution must be up and running at the moment it is needed.

From early 2015 digitalization, in the sense of exploiting existing and new technology in more efficient ways, became an agenda for AE. This resulted in several initiatives in the effort to start a digitalization journey: a digitalization strategy was decided by corporate management; Microsoft Azure was chosen as the platform for advanced analytics and later also for new applications development. Several pilot projects were executed in three different business areas: "Energy Management & Trading" (EMT), LOS who delivers energy to private customers, and, Agder Energi Nett (AEN) who operates the electric power grid.

To ensure a common understanding and direction for advanced data analysis in AE a strategy project covering both new and existing uses of data was launched, the project got the name Saturn. Aligned with the new corporate strategy, which focuses on digitalization, the Saturn project conducted several workshops in each of the five main business areas, the three areas mentioned above and in addition Agder Energi Konsern and Agder Energi Vannkraft (AEVK) were included. For each business area, several use cases utilizing data and advanced analytics were identified and prioritized. In December 2017, the Saturn project was completed, and the identified cases were handed over to each business area to scope in more detail and decide on a roadmap and timeline.

A big data team was put together with people from the ICT department and hired consultants. Several issues concerning the big data platform, such as governance, security and data management were discussed and initial guidelines were made.

For this study we want to focus on one of the identified cases from a development point of view, to provide insight into how the development process progressed, challenges we stumbled in to and eureka moments, without getting to technical about it. Rather, we want to point out a few issues that might lead to a change

in how organizations like AE, where IT is not their core business, may need to do major or even radical changes to their IT development and operation processes.

In November 2017 we chose a case identified by AE as a relevant big data case in the business area Energy Management & Trading (EMT). The case involves re-creating an existing report "The Key Figures Report" (KF-report), using the Microsoft Azure analytics platform and the tools it provides. The authors chose this case because it also was the first case chosen by the big data team and hence, at the time of writing this paper, has come further than any other cases even though it has yet to be completed.

For the participants of the big data team the advantages of starting with solving the KF-report-case was that the scope was well defined, and at the same time it covered many of the aspects the developers needed for exploring the possibilities in Azure and building experience.

The underlying data of the KF-report is mainly time series from external power related sources such as exchanges (Nord pool, EEX, Nasdaq OMX), data aggregators and analysis firms (Reuters, Wattsight) and internal production data. These data sources deliver market data such as prices or power production related to various types of power generating sources (e.g. coal, gas, oil, water, wind, etc.).

Data has been collected through documentation produced throughout the implementations process, such as meeting minutes and presentations made for the management team. In addition, we have conducted in-depth interviews with two central members of the implementation process. Three additional members of the implementation process have answered questions to validate and verify our understanding of the process.

Next we will describe the components we have used so far in Azure, what we used them for, and which challenges we met.

4. Implementation of Azure in Agder Energi

4.1 Abbreviations

Table 1 displays the abbreviations we have used together with a description of each of them.

ADLS – Azure Data Lake Store, Microsoft describes ADLS as "*Azure Data Lake Store is an enterprise-wide hyper-scale repository for big data analytic workloads. Azure Data Lake enables you to capture data of any size, type, and ingestion speed in one single place for operational and exploratory analytics*" (Microsoft, 2018b).

ADF – Azure Data Factory, Microsoft describes ADF like this: "*Big data requires service that can orchestrate and operationalize processes to refine these enormous stores of raw data into actionable business insights. Azure Data Factory is a managed cloud service that's built for these complex hybrid extract-transform-load (ETL), extract-load-transform (ELT), and data integration projects.*" (Microsoft, 2018a).

ALA – Azure Logic Apps, Microsoft describes ALA like this: "*Logic Apps helps you build, schedule, and automate processes as workflows so you can integrate apps, data, systems, and services across enterprises or organizations*" (Microsoft, 2018c).

ADLA – Azure Data Lake Analytics, Microsoft describes ADLA like this: Azure Data Lake Analytics is an on-demand analytics job service that simplifies big data. (Microsoft, 2018b).

AF - Azure Function is a solution for easily running small pieces of code, or "functions," in the cloud.

U-SQL – a query language from Microsoft; a combination of SQL and C#

Table 1 Abbreviations and descriptions used in this case

4.2 Azure Data Lake Store in Agder Energi

AE started by building a folder structure in ADLS. Based on suggestions described by Brandt-Kjelsen (2017) and later adapted to their needs by their big data team, AE defined 5 main stages to make the basic structure for data ingest and further processing of data (Figure 2). The brown marking in Figure 2 shows which stages and types of data sources that are used in the example pipeline for data ingest described in section 4.3.

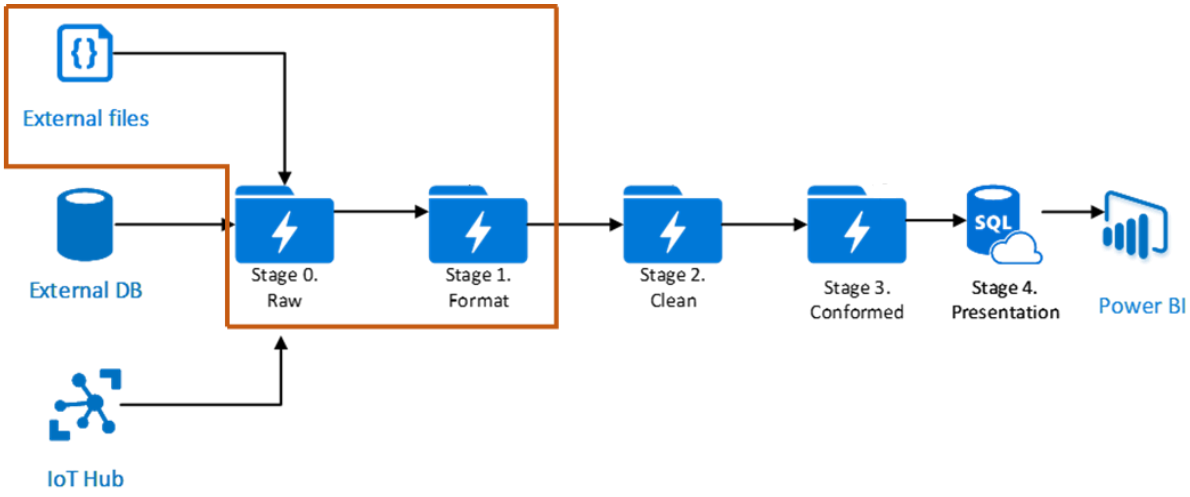

Figure 2 Agder Energi's stages in Azure Data Lake Store (adapted from (Brandt-Kjelsen, 2017))

The following description of the stages are following the descriptions from Brandt-Kjelsen (2017), although adjusted to AE's own definitions and stage numbering.

Stage 0 contains raw data ingested from various internal and external data sources. All data, regardless of source and structure remains here for all time, with the idea that one may go back to any point in time and perform analysis.

Stage 1 contains single format, but otherwise untreated data. The single format ease querying data across different data sources.

Stage 2 contains cleaned data sets which reduces time spent on preparation of data for advanced analytics.

Stage 3 contains data that is conformed in the sense that it will be possible to combine or integrate data across sources in a more structured way, using common references and keys.

Stage 4 is the presentation stage. Here, data may need to be aggregated and prepared in different ways depending on purpose of use.

4.3 Building a pipeline for ingest and further processing of data

In the Azure Data Lake Store section, AE explained the stages and their intended use. In this section, we will describe the pipeline of ingest and further processing of data through Stage 0 and Stage 1 in more detail. These stages are more or less completed for the KF-report, while the remaining stages do not contain any data yet (April, 2018). Still, questions and challenges that arose through these two stages should be of interest for similar organizations wanting to take on the same journey of exploring and exploiting a big data platform.

The basics of the case described is an already existing report; "The Key Figures Report" (KF-report), which AE will re-create in Azure. The report mainly consists of time series data. Although the report depends on data from several external sources, AE will concentrate the case around two of these sources when explaining the pipeline for ingesting data from the sources to Stage 0 and further transformation and transfer to Stage 1. For each step in the pipeline we will comment on various challenges experienced by AE.

Figure 3 illustrates one actual pipeline produced, from a Logic Apps Designer environment within Azure. It displays the pipeline steps from collecting data from an external source to "Stage 0. Raw" and further to "Stage 1. Format".

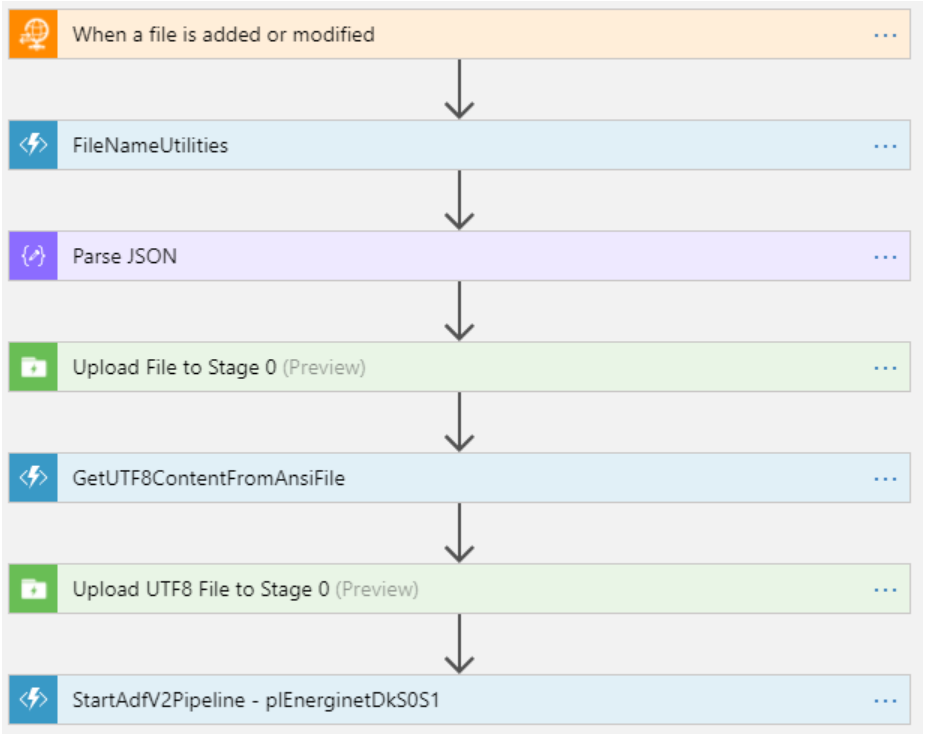

Figure 3 Pipeline for ingesting and processing files displayed from Logic Apps Designer

Step 1: "When a file is added or modified", is a trigger that checks the ftp site containing files from EnerginetDK (https://energinet.dk/). The check is performed every 15 minutes. If a file is changed or added the next step (Step 2) will be triggered.

First step challenge: Originally, the developers wanted to use Data Factory to orchestrate the whole pipeline (ref. Microsoft's description of Data Factory mentioned earlier in this section). However, Data Factory v1 could only be triggered by a schedule, which in this case meant that it would have started the whole pipeline every 15 minutes whether a file was changed/added or not. While the trigger used by Azure Logic Apps (ALA) only triggers the next step in the pipeline *if* a file is changed/added.

Data Factory v2 (preview) existed at this point in time, but in preview version only. The developers hesitated to take it into use since it was going to run in a production environment. Up until now, the policy was to use only well tested software versions from vendors before they put anything in to production. Also, the improvements in v2 concerning trigger functions would not help much as it would still run the whole pipeline for each check, whether it was necessary or not.

Step 2: "FileNameUtilities", is an Azure Function (AF) containing in-house coding (C#) to collect file - and folder name, both individually and concatenated. The output from this second step is in JSON format and is used as input to the next step.

Second step challenges: This could have been done later in the process, during the last step "StartAdfV2Pipeline – plEnerginetDkS0S1", but the developers ran in to some challenges. First, the string functions in Logic Apps is slightly different from what the developers are used to, and the UI is a bit "cumbersome to work with" as one of the developers put it, and he continued "amongst other issues cut and paste is a pain; on ctrl+v often the UI simply disappears and that is not a very practical behavior when you have to perform this operation several times". Instead the operation is performed one time in this second step and then it is not required to be done again.

Step 3: "Parse JSON", is a data operation where the output from Step 2 is parsed, so that the individual elements from the input becomes available for later steps. The file – and folder names are now already on the wanted format and it can simply be pushed through during the pipeline steps and be used where needed.

Third step challenges: None, this data operation component is a built-in component in Logic Apps that parses JSON formatted text against a schema and makes JSON formatted data available for the rest of the pipeline.

Step 4: "Upload File to Stage 0" is an Azure Data Lake action where the file is uploaded to Stage 0 on its original format, in this case a .txt file in ANSI.

Fourth step challenges: This is a built-in component in Logic Apps, but maximum file size for the "Azure Data Lake action"-component is only 30MB and that was a challenge. Another limitation was that, Logic Apps itself has a limitation of 100 MB files, *and* a 2 minutes limitation before it goes to time-out when calling functions. This was a challenge as well since some functions had 10 minutes limitations before they went to time-out. This again led to situations where Logic Apps went to time out after 2 minutes, while the function was still running. Also, the files grow extensively when transformed from original format to JSON. Hence, runtime errors were frequent before the developers identified the limitations. From these experiences they knew they would have to use Data Factory to transform and transfer files from one stage to another.

Step 5: "GetUTF8ContentFromAnsiFile" is an Azure Function containing in-house coding (C#) to get the file from Stage 0, and change it from ANSI to UTF-8 format.

Fifth step challenges: Azure Data Lake supports only UTF-8 (8-bit Unicode Transformation Format) and expects that everything stored is in UTF-8. Hence, the developers needed to do a transformation from ANSI to UTF-8 in this case, otherwise they would experience difficulties with processing files using special characters (such as Scandinavian letters like "æ", "ø" and "å"). One of the C# developers suggested there should be a built-in converter in the Azure connector that triggers file fetch from external sources (in Step1), or some other Azure component that converts the file to the wanted format right before the file is loaded to Stage0 in ADLS.

Step 6: "Upload UTF8 File to Stage 0" is an Azure Data Lake action where the file again is uploaded to Stage 0, this time overwriting the original file.

Sixth step challenges: Same as in fourth step; size and timeouts challenges

Step 7: "StartAdfV2Pipeline – plEnerginetDkS0S1", is an Azure Function where Azure Data Factory is used to move the file from Stage 0 to Stage 1 and transform the format to JSON.

Seventh step challenges: The transformation to a common format is done to make it easier to analyse and compare data sets. But this gave rise to a challenge when it came to the handling of metadata. From the original data source, files are often given a set of metadata placed at the start of the file content (e.g. source name and update frequency) and then the business data (could be prices, temperatures etc.) are placed after that. Metadata are imperative for the interpretation of the data set. But, metadata could also create some difficulties if both the metadata part and the business data part were transformed into the same JSON object. This is due to Azure Data Lake Analytics (ADLA) using U-SQL: since U-SQL primarily reads lists it is difficult to both read and write files containing both business data and metadata in the same file. Hence, in this step the raw data files are split into one meta data file and one business data file, and both files are named in a specific way to know what meta data belongs to what business data file. This solution has another advantage as well; if the originator of the file changes the metadata in the future (could be another update frequency or another unit), users will always know which metadata belongs to which business data file.

In Appendix A, we present an example of how file content could look like, when the original file is a .csv file and the outcome after transformation to JSON is one metadata file and one business data file.

There are one such pipeline made for each source data are collected from, and the sources may deliver data on different formats.

4.4 DevOps

In addition, during the time the big data team has been working with the KF-report and other use-cases, DevOps (the combination of Development and Operations) has become a more and more frequently discussed theme. Through the experiences Agder Energi have had with the use-cases so far; they see an emerging picture of the need for a change in AE's existing IT development and operations processes. This is mainly because they do not find it suitable to hand over applications and integrations to traditional operations personnel who has not been a part of the development as well.

One of the developers put it this way when one of the authors asked him to define DevOps: "*You know, in earlier days it was developer teams who developed applications, and then there was an operational team with application administrators and system administrators or, well, all these names related to operations. But, a DevOps team not only develop the solutions, they carry the operational responsibilities as well, 100%. Hence, they have the responsibility not only for the development, but also for continuous improvement, and fixing issues that might turn up, and they should do that for the whole lifecycle of the application*".

This is a challenge in an organization which up until now has used the traditional handover to operations when developers complete their tasks.

4.5 Summary of challenges

A) Technical limitations in current version of available integration tools
   - Logic Apps has limitations related to both file size and run-time when transferring data and calling functions. In addition, some of the components you may use together with Logic Apps have even more limitations on file size than Logic Apps. Also, the files grow extensively when transformed from original format to JSON. The main challenge to these limitations is that even if AE perceived Logic Apps as a nice tool for what it was meant for (small amounts of data, with several built-in flexible trigger objects), it was not very well documented. Hence, the result is a lot of testing, failing and testing again.

B) Limited possibilities for efficient and consistent orchestration of data flow
   - Data Factory is the orchestration component according to Microsoft. Still, there was no way to start out the pipeline with this component since it lacked a possibility to listen on external sites for additions/changes of files. Neither was it possible to listen for the same issues in the Data Lake. Data Factory v1 was lacking features to a degree that made it not so useful.

C) Development of tools happens fast
   - Data Factory v2 was launched in preview (similar to a beta-version) and at this point in time (April 1st, 2018) it is still in preview. Agder Energi developers perceived version 2 as a huge improvement from version 1. As one of the developers described it: "Data Factory v2 is a completely different tool than v1, you cannot even compare them, which is my opinion". But, using a preview version in production felt like exceeding a threshold, and the developers were wondering what the consequences might be. Nevertheless, Data Factory v2 was used for Step 7 in the pipeline for ingesting and processing files (see Figure 3) in the production environment.

D) Experience is lacking also outside the organization

E) More in-house coding was needed than expected.
   - U-SQL (a combination of SQL and C#) was very useful for developers coming from a data warehouse environment. Even so, the developers experienced that you really needed expert C# programmers as well

The developers experienced several challenges, hence they were searching for answers to these challenges in many places, such as web forums related to Azure; a network of developers who they knew from

international conferences and seminars; people from Microsoft; etc. The developers' opinion is that there are few experts in this area, although all asked are kindly willing to suggest other places to look for an answer. On one occasion, the developers got a very nice work around for one of their problems when raising a question in an Azure forum, responded by an expert from Microsoft. But the latter is not the normal response, usually either no one knows, or they reply with another question of challenges of their own, happy to meet someone in the same situation. Mostly the developers ended up using a lot of time on trying, failing, and trying again.

5. DISCUSSION

Through this research we have presented the implementation process of the cloud based big data platform Microsoft Azure in AE. We have focused on the implementation of the platform into one business unit, Energy Management and Trading. Through this implementation process we have observed that AE has followed the guidelines for implementation of enterprise wide applications. The decision was well anchored in the management group, and communicated as a part of the cooperate strategy. Further, the big data team put together by the organization to ensure the Saturn project the best working conditions for their work. The corporate strategy gave the project a clear goal, where the scope of the big data team was decided. In addition, AE had business units ready to start the implementation, in that way functioning as champions and involved end users in the process. All of these steps is in alignment with previous research on critical success factors for implementation of enterprise wide applications (Jiang et al., 1996; Reitsma and Hilletofth, 2017).

Even with these steps, AE experienced challenges in the implementation process. Some of these challenges were seen as surprising and may warrant a closer look into processes like this. In the further we will discuss what implications may be derived from the challenges experienced when building the "pipeline of ingest and further processing of data" described in section 4.3.

Although Microsoft Azure has been around since 2008 (Copeland et al., 2015), the big data team experienced the technology to be immature in terms of technical limitations in current version of available integration tools. These limitations included file size limitations, run-time limitations, and limited possibilities to orchestrate data flows.

When Azure Data Factory v2 was released the developers realized that this version was very different from v1, and that file size and run-time limitations would not be a problem if they could use the latest version to move data from one stage to another. However, there were two problems related to v2.

First, ADFv2 could still not "listen" to changes in external data sources in order to start the ingest pipeline when such changes occurred. For this problem Logic Apps was the solution. But, this solution meant that Data Factory could not be used to orchestrate the whole pipeline, it could be started only when the data was already in Stage 0 and prepared for moving to Stage 1. Hence, since Microsoft advertised Data Factory as the orchestration tool, it is fair to expect new version(s) of Data Factory to be released in near future. Second, ADFv2 is so far released only in preview. The normal procedure for Agder Energi would be to wait for well tested version of vendor software. Hence, at first the developers hesitated to put the preview version to use in production, but realized that with the history of continuously updates and rapid changes in Azure (Copeland et al., 2015) they had to. Not just now, but preview versions in production may well be the norm also in the future.

The issues discussed above also lead to another implication; if continuously updates and rapid changes in Azure also is the norm for the future, it is hard to see how applications developed using the Azure stack of tools can be handed over to operations personnel that have not taken part in the development. This implies that developers in Agder Energi also have to function as operations personnel since the developers will be the only ones capable of coping with the speed in changes affecting the applications they developed.

Further, if the organization wants to stay in the forefront of competitors, there are two other challenges to discuss. First; experience is lacking also outside our case organization. As mentioned in Chapter 4.5, the developers were searching for answers to their challenges in many places, such as web forums related to Azure; in networks of developers who they knew from international conferences and seminars; Microsoft personnel; etc. It appears that experts are scare in this field. The lack of answers implies that rapid learning

capabilities among developers will be a must since there are few know-how people to ask. Second (and this may be limited to our case organization); more in-house coding than expected were needed (especially in C#) and this underline the need for rapid learning capabilities among developers. Figure 4 summarizes the challenges experienced by our case organization, and what the implications for each of them are.

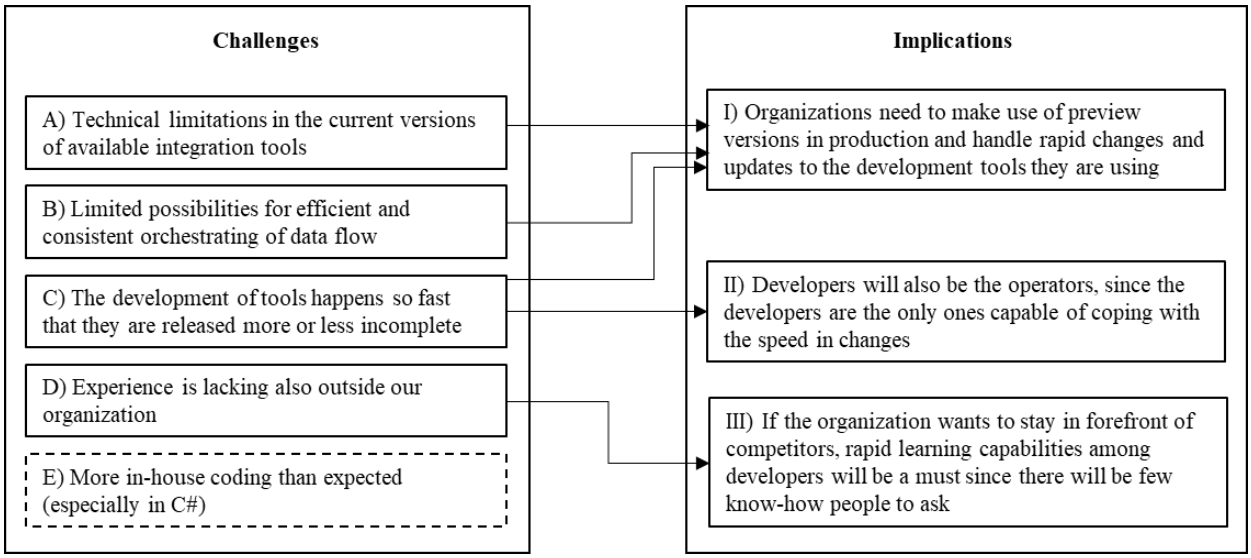

Figure 4 Challenges and Implications

The capital letters on the left side under Challenges correspond with the letters presented in chapter "4.5 Summary of challenges". Challenge E) is dotted because this challenge might be limited to the case organization, while the other challenges could be more general since they mostly reflect issues concerning various tools in Azure.

6. CONCLUSION AND IMPLICATIONS

The aim of this research was to answer the question: What are the challenges of implementing cloud based big data platforms?

We have described the process of such implementation in AE, and identified the challenges experienced when implementing the cloud based big data platform Microsoft Azure. In Figure 4 we provide an overview of the connections between the challenges and implications, and how they related to the critical conditions of data needs in our case.

It is interesting to observe the challenges identified are not the traditional known challenges for implementation of enterprise wide information systems. There are indications that the cloud based big data platforms are not mature and organizations will need additional programming resources to ensure being able to utilize the platform in an optimal way.

The big data team experienced gradually a need for utilizing DevOps. DevOps was by the team understood as an agile development and operations lifecycle of every in-house developed cloud application where the developers also perform the operation management. This means there will be no "hand-over" to the "Operation Management department", rather the applications will go through a test regime, and if it succeeds, the application will be put in production by the developer. There will be automatic monitoring of the application and belonging processes, with automatic error fixes when available and alerts to developer if not. This is similar to the definition of DevOps presented by Jabbari et al, (2016). Through the use of DevOps, the organization can ensure full control and maintenance for the entire lifecycle of the application without handing it over to stakeholders "who was not a part of the process before (Qazi and Bashir, 2018, p.1).

From the Implications in Figure 4, it seems that utilizing DevOps could be a natural follow-up on the situation. Also, looking back at the critical conditions required for data consumption (the need for a vast

amount of data collected from a diversity of internal and external data sources; short time from order to operation; and high uptime for required applications (chapter 3.1)) it will obviously be required that operations personnel know the various solutions very well. Which again support the idea that the developers should also act as operations personnel.

However, before taking on a full-blown DevOps strategy there is a need for looking into the impact the use of DevOps will have on AE's existing software handover to operation processes, and how it will affect other processes and people. This has not been looked in to yet.

There is a clear path towards a paradigm shift from traditional "software handover to operations"- processes to DevOps processes where developers also are the operators. The transformation to DevOps processes might imply high impact in various ways on the organization. Several questions arise and may be worth looking into:

What does a shift to DevOps mean for the organizations need for future resources? Does it mean the organization need more developers and fewer "regular" operators? Does it mean we will have more organizations outsourcing software development and hence increase danger of losing ownership to the solutions? If so, is there a danger that they might lose control over their own data?

Future research should explore how organizations that do not have IT as their core business, can cope with such a transformation. Our research warrants continuous work to understand how to deal with this new landscape.

7. REFERENCES

Blazquez, D. and Domenech, J. (2018). Big Data sources and methods for social and economic analyses. Technological Forecasting and Social Change, 130, 99-113.

Bouwman, H., Nikou, S., Molina-Castillo, F. J. and de Reuver, M. (2018). The Impact of Digitalization on Business Models. Digital Policy, Regulation and Governance, just-accepted, 00-00.

Brandt-Kjelsen, E. (2017). *Data driven development and operations to support digitalization.* Master's, Westerdals – Oslo School of Arts, Communication and Technology, Oslo.

Copeland, M., Soh, J., Puca, A., Manning, M. and Gollob, D. (2015). Microsoft Azure and Cloud Computing *Microsoft Azure: Planning, Deploying, and Managing Your Data center in the Cloud* (pp. 3-26). Berkeley, CA: Apress.

Creswell, J. W. (2007). Qualitative inquiry & research design: choosing among five approaches: Sage Publications.

Eisenhardt, K. M. (1989). Building Theories from Case Study Research. Academy of Management Review, 14, 4, 532-550.

El-Gazzar, R., Hustad, E. and Olsen, D. H. (2016). Understanding cloud computing adoption issues: A Delphi study approach. Journal of Systems and Software, 118, 64-84.

Hartley, J. (2004). Case Study Research. In C. Cassel & G. Symon (Eds.), *Qualitative methods in organizational research. A practical guide*. London: Sage.

Hashem, I. A. T., Yaqoob, I., Anuar, N. B., Mokhtar, S., Gani, A. and Khan, S. U. (2015). The rise of "big data" on cloud computing: Review and open research issues. Information Systems, 47, 98-115.

Hosono, S. (2012). A DevOps framework to shorten delivery time for cloud applications. International Journal of Computational Science and Engineering, 7, 4, 329-344.

Jabbari, R., Ali, N. b., Petersen, K. and Tanveer, B. (2016). What is DevOps?: A Systematic Mapping Study on Definitions and Practices. Paper presented at the Proceedings of the Scientific Workshop Proceedings of XP2016, Edinburgh, Scotland, UK.

Jiang, J. J., Klein, G. and Balloun, J. (1996). Ranking of system implementation success factors. Project Management Journal, 27, 49-53.

John Walker, S. (2014). Big data: A revolution that will transform how we live, work, and think: Taylor & Francis.

Kane, G. C., Palmer, D., Phillips, A. N., Kiron, D. and Buckley, N. (2015). Strategy, not technology, drives digital transformation. MIT Sloan Management Review and Deloitte University Press, 14.

Kezunovic, M., Xie, L. and Grijalva, S. (2013). The role of big data in improving power system operation and protection. Paper presented at the Bulk Power System Dynamics and Control-IX Optimization, Security and Control of the Emerging Power Grid (IREP), 2013 IREP Symposium.

Khan, A. S. (2013). *A Framework for Software System Handover*. KTH Royal Institute of Technology.

Marinescu, D. C. (2017). Cloud Computing: Theory and Practice: Elsevier Science.

Microsoft. (2018a). Introduction to Azure Data Factory Retrieved 07. April, 2018, from https://docs.microsoft.com/en-us/azure/data-factory/introduction

Microsoft. (2018b). Overview of Azure Data Lake Store Retrieved 7. April, 2018, from https://docs.microsoft.com/en-us/azure/data-lake-store/data-lake-store-overview

Microsoft. (2018c). What is Azure Logic Apps? Retrieved 7. April 2018, from https://docs.microsoft.com/en-us/azure/logic-apps/logic-apps-overview

Qazi, A. M. and Bashir, M. (2018). A Service Oriented Handover Process for Software Development Projects. International Journal of Advances in Computer and Electronics Engineering, 3, 5, 1-8.

Reitsma, E. and Hilletofth, P. (2017). Critical success factors for ERP system implementation: A user perspective. European Business Review, just-accepted, 00-00.

Timonen, H. and Vuori, J. (2018). Visibility of Work: How Digitalization Changes the Workplace. Paper presented at the Proceedings of the 51st Hawaii International Conference on System Sciences.

Virmani, M. (2015). Understanding DevOps & bridging the gap from continuous integration to continuous delivery. Paper presented at the Innovative Computing Technology (INTECH), 2015 Fifth International Conference on.

Weichenrieder, A. (2018). Digitalization and taxation: Beware ad hoc measures: SAFE Policy Letter.

Appendix A

Below we present an example of how file content could look like, when the original file is a .csv file and the outcome after transformation to .json is one metadata file and one business data file. In this example we have used a file collected from Wattsight (https://www.wattsight.com/).

Example of a raw data .csv file transformed to .json:

The raw data file stored in Stage 0 is displayed with 12 rows, while in reality it includes several more rows. This is done due to space limitations of this paper. This .csv file typically includes 23-25 business data rows since the data frequency is set to "hourly", but other files could include several more rows depending on data frequency which could be set to a more rapid paste (f.ex minutes).

Raw data file (.csv, row 1-6 is metadata, row 7 is column headings, row 8-12 business data):

1. Data from Wattsight.com
2. Updated: ;"2018-03-25 00:26";
3. Descriptive name: ;"Hydrology Hourly (np)";
4. Update frequency ;"Daily";
5. Data Frequency ;" Hourly";Timezone:;CET;
6. Comment: ;"Actual hydrological data backwards and forecasted hydrological data from our short and medium term models. Forecasts are continuously updated with the latest weather forecasts.";
7. Date;RRE NPA HY-TOT MK02 GWH H AF;INF NPA HY-TOT MK02 GWH H AF;INF NPA HY-TOT GROSS MK02 GWH H AF;PRO NPA HY-ROR MK01 MWH H AF;PRO NPA HY-TOT MK01 MWH H AF;RES NPA HY-SGW MK04 GWH H AF;RES NPA HY-WTR MK04 GWH H AF;RES NPA HY-BAL MK04 GWH H AF;RES NPA HY-WTR CAP MK03 GWH D A;

8. 2017-12-15 00:00;27,344499401632728;8,352733355605457;8,565382750681962;7561, 967135656299;27828,152861409682;24386,450760987813;90005,17762465394;11662,730869181898;120294,0;
9. 2017-12-15 01:00;22,99402642079624;8,35807263990231;8,570722034978814;7450, 175148673449;26420,52821706903;24400,87406537363;89988,48029686748;11658,134584760668;120294,0;
10. 2017-12-15 02:00;26,542248901196352;8,347857503260164;8,56050689833667;7496, 599095259022;26121,525585075244;24418,85580737649;89972,13094991026;11657,439871670595;120294,0;
11. 2017-12-15 03:00;26,11903961053204;8,34133229730737;8,553981692383875;7558, 343041844596;26342,200402125087;24436,42086529464;89955,57248605024;11656,114524357154;120294,0;
12. 2017-12-15 04:00;23,933559048209446;8,289525355220267;8,50217475029677;7588, 00698843017;27482,68623076906;24451,85224959255;89937,9431254624;11651,579787339268;120294,0;

After storing the .csv file in Stage 0, the file is split into two JSON files on its way to Stage 1, one for metadata and one for business data:

Metadata file (.json, represent rows 1-7 from the .csv file)

[

{"MetadataFile":"Updated: ;2018-03-25 00:26;"},

{"MetadataFile":"Update frequency ;Daily;"},

{"MetadataFile":"Descriptive name: ;Hydrology Hourly (np);"},

{"MetadataFile":"Date;RRE NPA HY-TOT MK02 GWH H AF;INF NPA HY-TOT MK02 GWH H AF;INF NPA HY-TOT GROSS MK02 GWH H AF;PRO NPA HY-ROR MK01 MWH H AF;PRO NPA HY-TOT MK01 MWH H AF;RES NPA HY-SGW MK04 GWH H AF;RES NPA HY-WTR MK04 GWH H AF;RES NPA HY-BAL MK04 GWH H AF;RES NPA HY-WTR CAP MK03 GWH D A;"},

{"MetadataFile":"Data from Wattsight.com"},

{"MetadataFile":"Data Frequency ; Hourly;Timezone:;CET;"},

{"MetadataFile":"Comment: ;Actual hydrological data backwards and forecasted hydrological data from our short and medium term models. Forecasts are continuously updated with the latest weather forecasts.;"}

]

Business data file (.json, represent rows 8-12 from the .csv file, pluss row 7 to provide each row with column headings and business data)

[

{"TimeStamp":"2017-12-15T00:00:00","RES NPA HY SGW MK04 GWH H AF":24386.450760987813,"RES NPA HY WTR MK04 GWH H AF":90005.177624653938,"RES NPA HY BAL MK04 GWH H AF":11662.730869181898,"RES NPA HY WTR CAP MK03 GWH D A":120294.0},{"TimeStamp":"2017-12-15T01:00:00","RES NPA HY SGW MK04 GWH H AF":24400.874065373631,"RES NPA HY WTR MK04 GWH H AF":89988.480296867478,"RES NPA HY BAL MK04 GWH H AF":11658.134584760668,"RES NPA HY WTR CAP MK03 GWH D A":120294.0},{"TimeStamp":"2017-12-15T02:00:00","RES NPA HY SGW MK04 GWH H AF":24418.855807376491,"RES NPA HY WTR MK04 GWH H AF":89972.130949910264,"RES NPA HY BAL MK04 GWH H AF":11657.439871670595,"RES NPA HY WTR CAP MK03 GWH D A":120294.0},{"TimeStamp":"2017-12-15T03:00:00","RES NPA HY SGW MK04 GWH H AF":24436.42086529464,"RES NPA HY WTR MK04 GWH H AF":89955.572486050238,"RES NPA HY BAL MK04 GWH H AF":11656.114524357154,"RES NPA HY WTR CAP MK03 GWH D A":120294.0},{"TimeStamp":"2017-12-15T04:00:00","RES NPA HY SGW MK04 GWH H AF":24451.85224959255,"RES NPA HY WTR MK04 GWH H AF":89937.9431254624,"RES NPA HY BAL MK04 GWH H AF":11651.579787339268,"RES NPA HY WTR CAP MK03 GWH D A":120294.0}

]

From the example above, it is visible that row 7 from the raw data file contains column headings. These column headings are transferred both to the Metadata file and to the Business data file. This is a choice made because it was believed this will be more user friendly, but it could be needed remove also these headings from the business data file, if they for example slows down machine reading of some sort.