Jonas Kauko

Development of Event Management process (ITSM) for a cloud-based SaaS company

Metropolia University of Applied Sciences

Bachelor of Engineering

Industrial Management

Bachelor's Thesis

7 December 2021

Abstract

Author:	Jonas Kauko
Title:	Development of Event Management process (ITSM) for a cloud-based SaaS company
Number of Pages:	67 pages + 2 appendices
Date:	7 December 2021
Degree:	Bachelor of Engineering
Degree Programme:	Industrial engineering and management
Professional Major:	ICT-business management
Supervisors:	Anna Sperryn, Senior Lecturer Employee of Case Company, Senior Architect & Product Manager

---

The purpose of this study was to create a proposal of a developed Event Management process for the case company. The main objective was to create a process including steps for technical implementation. This was an independent project for the case company, related to several internal improvement efforts.

The research is based on interviews of the case company's employees, discussions, and inspection of technical implementation and documentation. Two interviews were conducted with the case company's employees in related roles to event management and other related areas, such as cloud, integrations, and service desk.

The results of the current state analysis including interviews, indicated that the current process is not sufficient and does not meet internal standards or possible customer or prospect requirements. The case company's event management is built on the incident management process and needs separation to its own process.

After the current state analysis, existing knowledge and best practices was researched, including topics like ITIL, event management, incident management, service desk practices, integrations, and event correlation. These topics provided a deep understanding of different areas needed to develop an event management process and the steps to implement it.

The final outcome of this thesis is a holistic event management process including best practices and technical specifications needed to successfully implement said process. This will help the case company to be more efficient.

Keywords: ITSM, Event Management, EAI

Contents

List of Abbreviations

ITSM	IT service management
ITIL	IT Infrastructure library
SaaS	Software as a service
ESM	Enterprise service management
MVP	Minimum viable product
EM	Event Management
IM	Incident Management
SM	Service Management
IT	Information technology
PM	Product Manager
MSP	Managed Service Provider
QA	Quality Assurance
BP	Best Practice
CI	Configuration Item
EC	Event Component
EAI	Enterprise Application Integration
API	Application Programming Interface

List of Figures

List of Tables

1 Introduction

This study examines the case company's Event Management (EM) process and proposes a plan on how to improve the process to match industry standards, to serve the internal stakeholders, and how the process should be technically developed to fit in with the current processes and systems. The case company is a small to medium sized cloud-based service provider, which operates with a Software as a Service (SaaS) business model.

The primary focus of this thesis is to propose a developed EM process for the case company that meets the requirements of the case company and solves any issues identified in the current solution.

1.1 Business Context

The case company for which this project was conducted, is a relatively new cloud-based service provider with a SaaS business model. The company operates across Europe.

1.2 Business Challenge, Objective and Outcome

An EM process is crucial for companies that have any IT-services and especially, companies that provide cloud-based solutions. The Event Management process monitors events through the internal IT-infrastructure, or any provided services, like cloud-servers or customer environments. Event Management allows for services to run normally, without interference, and allows for escalation of any exceptional conditions. (Axelos, 2019)

The case company for who this study was created for, has a basic Event Management process implemented which is in use. However, the current EM process does not meet the case company's internal requirements or some

expectations of the case company's customers or prospects. The current Event Management process is excessively heavy and creates a lot of manual work, by creating unnecessary tickets to the current system, of which many needs to be deleted or ignored since not being relevant. The data created from the monitoring is not rich and structured enough to be used properly to create meaningful information that can be used automatically. Also, the current process does not provide sufficient reporting capabilities that could be offered to customers. Better reporting capabilities could also help understand the root causes of different events and problems and can be used to develop the internal business activities.

The above-mentioned reasons make it clear that there is a need to develop the current event management process to match industry standards, to improve the service quality and to increase the efficiency and ease the service desk agent's day to day work.

The objective of this study was to develop an improved EM process proposal to match industry standards and to meet the case company's requirements, resolving issues that the current solution causes.

The outcome is a proposal of a developed Event Management solution that meets the case company's requirements. The outcome should improve the service quality and ease the service desk agents work by creating a structural and coherent EM process, enriching, and structuring the event data. The technical implementation to the case company's system is outside the scope of this thesis due to time restrictions.

1.3 Thesis Outline

The study was conducted by using qualitative research methods such as interviews, discussions, existing documentation, existing technical implementations, as well as workshops. The area of research was limited to the case company's internal Event Management process and the project that was conducted to implement this.

This study contains sections, where section 1 introduces the case company briefly, its business context and challenge, as well as the objective and outcome of the study. Section 2 elaborates the method and materials used in this thesis, by describing the research design, project schedule and the data used in this study. Section 3 analyses the current state of the case company's Event Management process and level of automation. Section 4 covers the available knowledge and best practices in ITIL, Event Management, Incident Management, Service Desk Practices, Enterprise Application Integration, Application Programming Interface, and lastly Event Correlation. Section 5 describes grounding of the project and answers the "why" and "how". This consists of the proposal for the developed event management. Section 6 discusses validation, pain points and how the proposal brings benefits. The last section evaluates and summarizes the thesis, recognizes the benefits, and addresses what could have done differently.

2 Methods and Material

The objective of this study was to create and implement an Event Management process for the case company. This section describes how the study was carried out to achieve this objective, by inspecting at the study's research design, how the project was planned to be executed and its schedule and lastly, presentation of the data collected in this study and analysis methods described.

2.1 Research Design

This study was conducted in 5 stages as shown in Figure 1 below.

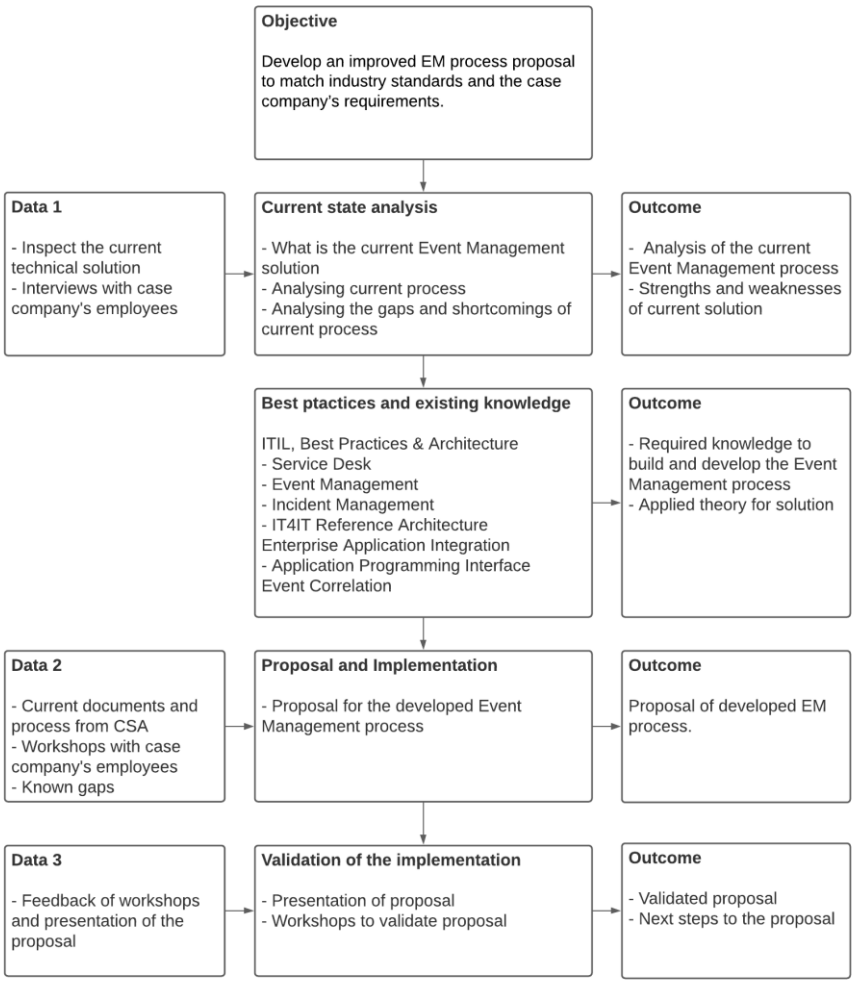

Figure 1. Research Design

The study started by defining the outcome of the project, which was to develop the existing Event Management process to enhance the service desk employees work, and to enhance service quality.

Next section analysed the current state (CSA) of the Event Management process and how it is related to other relevant processes and activities. Here, interviews with the case company's employees and inspection of existing process and solution, served as the data source. The objective of the CSA was to offer required knowledge to develop the Event Management process and to apply the best practices and theory to full extent.

After the CSA, a proposal of the EM process was built and proposed to fit into the case company's existing Service Management system. The outcome is a proposed first version, which will be implemented later, outside of this study.

Lastly, validation of the proposal was done including suggestions and next steps. Outcome of the validation is a final version of the proposal.

2.2 Project Plan and Schedule

The project plan and schedule was made to start in week 40 and continue until week 46. The schedule was rather tight, to cover analysis of the current state and proposal of an Event Management process.

Week 40 started with planning the study and the kick-off. After this, Gates 1 & 2 were started and completed until the middle of week 41. This is when the Gate 3 started, including the Current State Analysis, which continued until week 43 and was done parallel with Gate 4, which is the part where relevant theory and best practices were researched. In the end of week 43, proposal was started to be built. This continued until the end of week 45. Parallel to this, Gate 6, where validation of the proposal, was done. Finally, in the last week, Gate 7 was done, which summarizes the study and reviews it critically. Also, a steering meeting was

held weekly, starting from week 41, with the study's supervisor from the case company. Figure 2 below depicts the study schedule for this study.

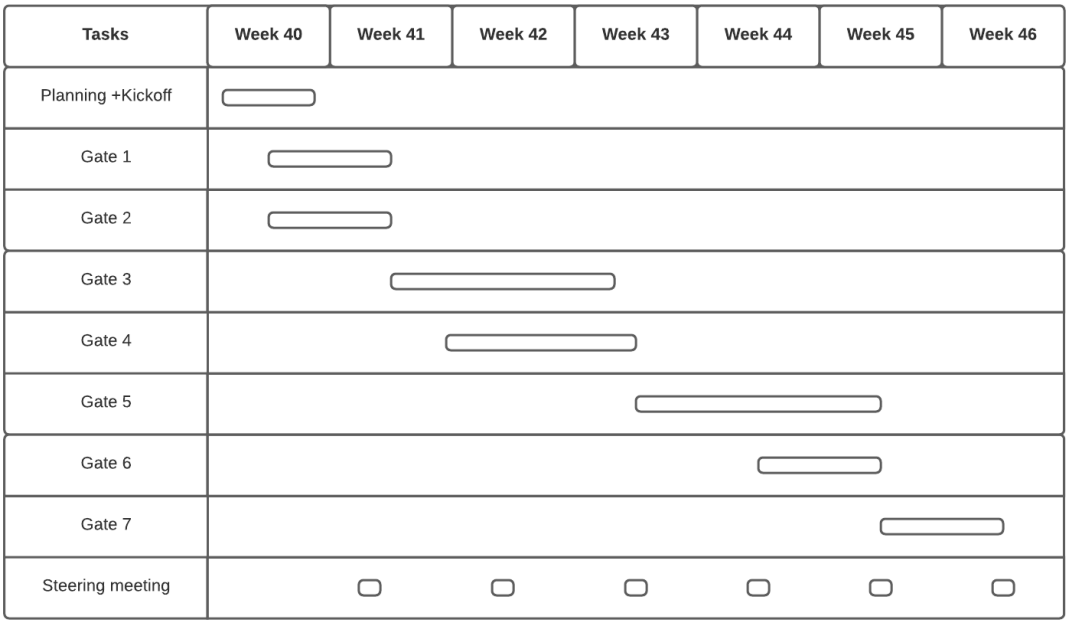

Figure 2. Gantt-chart of the study schedule

This simple Gantt-chart above in figure 2, shows visually how the schedule of the study was planned.

2.3 Data Collection and Analysis

The data collected for this study is divided in 3 different sections, where the first section gathers the data for conducting the CSA, second, of interviews and workshops with relevant personnel of the case company, and third, feedback on the proposal via discussion and workshops conjugated with the presentations of the proposal.

For the first data collection round, Table 1. down below, presents the interviews and discussions with different employees of the case company. Persons were selected to give a comprehensive understanding of the current state from different positions and views in the case company.

Table 1. Data collection table for first round.

Data 1 - For Current State Analysis					
	Participants / Role	**Data type**	**Topic**	**Date**	**Documented**
1	Senior Architect & Product Manager	Discussion	Current Event Management process	15.10.2021	Field notes
2	Service Desk Lead	Interview	Service desk process and tickets related to event management	18.10.2021	Interview transcription
3	Cloud Product Manager	Interview	Cloud and server events and monitoring	19.10.2021	Interview transcription
4	Senior Consultant	Discussion	Current Technical Solution / Process	28.10.2021	Field notes

Seen in the table 1 above, the first collected information for the CSA was a discussion held with the case company's Senior Architect & Product Manager. Here the overview of the current EM process was discussed. The first interview was held with the Service Desk Team Lead of which a transcript of the interview can be found in Appendix 1. The second interview was to get a perspective from the Cloud Ops. This interview can be found in Appendix 2. Lastly, discussion with a Senior Consultant about the process and technical implementation was held, to help gain an understanding of how to document it.

Table 2. Data collection table for round 2.

Data 2 – For proposal building and technical implementation					
	Participants / Role	**Data type**	**Topic**	**Date**	**Documented**
1	Senior Architect / Product Manager and SD Team Lead	Workshop	Specifications and requirements for event management proposal	1.11	Field notes
2	Senior Integration Consultant	Workshop		8.11	Field notes

For collection of data 2, the first workshop was held with the case company's employees with the title Senior Architect & Product Manager and Service Desk Team Lead. Here specifications and requirements for the new and developed EM process were discussed. Topics such as what does the SD need from this and what requirements does customers have for reporting were addressed. These requirements will be presented later on in the study, in section *5.2 Findings of data 2.*

The second workshop held in regard to data 2 collection, was a workshop with the case company's Senior Integration Consultant. In this workshop, the technical integration process was discussed and planned. The integration process is described in detail and visualized later on in section *5.2.2 Integration process* of the study.

Table 3. Data collection table for round 3.

Data 3 – For validation and testing					
	Participants / Role	**Data type**	**Topic**	**Date**	**Documented**
1	Senior Architect / PM	Workshop	Validation of the proposed EM process.	17.11.2021	Field notes

The last data collection round included a workshop for validating the proposal. Here discussion about the proposed EM process was held and any improvement and comments were brought up.

3 Current State Analysis of the EM Process in the case company

In this section of the study, current state of the case company's EM process is analysed, by conducting interviews with key employees, as well as inspecting documents and technical implementation. The goal for this section is to find the strengths and weaknesses of the current process and technical implementation. This is done to have a starting point for building the proposal and implementation, later in the study.

First, the purpose of conducting the CSA is discussed.
the service desk's role in the case company's EM process is described. The data for this section, was collected via an interview of the case company's employee with the title Service Desk Lead. In this interview, relevant questions about the current EM process were asked, which can be found in the Appendix 1. Also, in this section, the data structure is described, as well as the event management process in its entirety. Secondly, integration events from the case company's customer cloud environments are inspected. After inspecting the integration events, cloud monitoring is described. In the last two sections, strengths and weaknesses of the current EM process are analysed, and finally the CSA is summarised.

3.1 The purpose for conducting the CSA

The purpose of the current state analysis was to gain an understanding and to collect relevant information on how the EM process is carried out in the case company. For being able to develop an EM process, the current state needed to be defined and analysed in different aspects of event management.

By having a clear understanding of the CSA, the plan for a developed event management could be created and followed for the technical implementation. Also, by analysing the strengths and weaknesses of the current EM process, a

deeper understanding was formed of the problematic aspects to be addressed in the proposal of a developed process.

3.2 The current state analysis

As the case company is a cloud-based software provider, which operates with a Software as a service (SaaS) business model, EM is an important part of maintaining the provided services. Since the software is sold as a service, the case company as the managed service provider (MSP), is responsible of maintaining the servers and other relevant operations to keep everything running.

Many different operations that the case company is conducting to run its business, generates data that different internal stakeholders use to perform daily operations and to develop internal activities. However, all data is not relevant, thus identifying what data is important and what to do with it is crucial. Since the case company is an MSP, different data is generated throughout the company, and used in different manners in operations related to e.g., the Service Desk, Cloud Operations and Quality Assurance (QA) etc. Without a centralized EM process, managing this data and using it to its fullest potential is difficult.

3.2.1 Service Desk operations

The Service Desk plays a key role in EM, by being a single point of contact for customers and end users. All tickets, incidents or service requests, goes through the SD from where they might be escalated forward if required.

In the interview conducted for this study with the case company's employee as the SD Team Lead, service desks relation to EM was described as "Analyzing and handling event related tickets. Escalating events that require more investigation to consultants/other specialists. Handling all event related incidents and customer communication." (SD Team Lead, 2021).

Since the case company provides most customers with their own cloud-environments where they host different applications which are then monitored by built-in monitoring features, or external monitoring software. When an error for example happens in one of the applications, cloud servers or elsewhere, the monitoring creates an event of it. This event includes relevant information about the error, which then gets sent to the case company's service desk.

Currently, all events generated are sent via email to the ticketing tool that the case company uses. When these emails are read into the system, they always generate an incident. This incident is then manually categorized and prioritized. Depending on the event, customer information needs to be manually added. If many events are generated from the monitored system, server, or application, they all form an incident which are not automatically linked to each other. This mean that this must be manually checked if there are multiple incidents generated from the same event, and then linked together. The linking happens by selecting all related tickets to one main ticket of the event. The tickets have a many-to-many relation to each other. Lastly, guidance of how the incident should be handled is often added to the ticket. For example, for integration related events, guidance in form of integration error cards is linked to the incident. Also, KB-articles can be linked. All this is done by the SD agents manually, which consumes a significant amount of time.

When the incident is categorized, the metadata is corrected, relevant linking is done, and it gets handled by the agent to best ability. If more information is needed from the customer, the agent contacts directly them by email from the ticket. However, this is quite rare, since most customers does not know how the configuration works in detail. If the event is more complex or major, the incident gets escalated to relevant internal specialist who could be a consultant that has built the configuration. The ticket ownership remains with the SD which then closes the ticket when it's completed, if not otherwise agreed. If the event requires a change request, it is transformed by the SD agent, and given to a consultant or other specialist to fulfill. In this case, the ownership of the ticket is transferred to the consultant in charge. The described process can be seen in Figure 3 below.

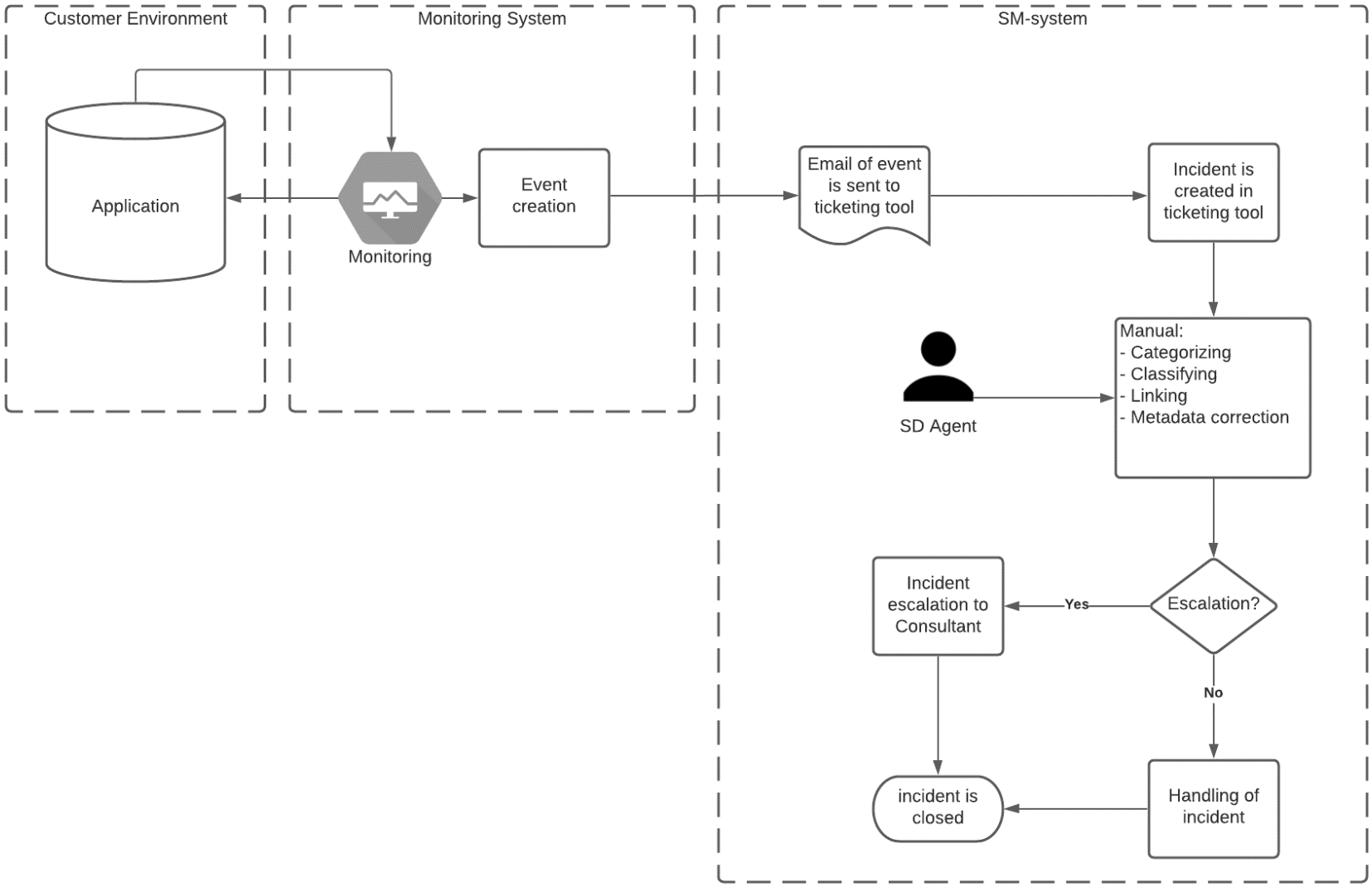

Figure 3: EM process overview based on interviews and discussions.

The process described and illustrated above in Figure 3 is based on the Interviews, discussions, and technical inspection that were conducted in the CSA. Transcripts of the interviews can be seen in Appendix 1 & 2 of this study.

The manual work needed to categorize, link, and correct any metadata is time-consuming and laborious for the SD agent. The interviewee estimated that all EM related tickets consume 0,5h – 1h of workload per SD agent per day for 5 agents. This calculates to 2,5h – 5h per week, which is 12,5 hours and up to 25 hours per week of work. This is comparable to one person working only with this monthly doing half-time.

The technical process could be visualized as seen below in figure 4, where an integration runs in the case company's integration platform. When an error occurs, it gets sent to the ticketing tool via email and an incident is created to existing IM process.

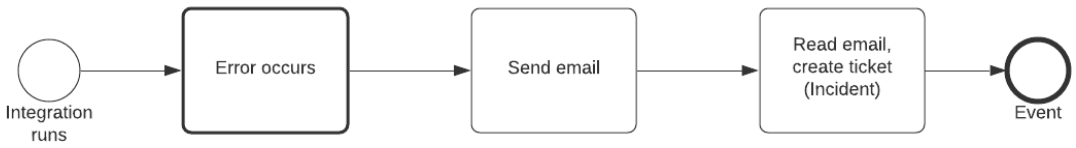

Figure 4: Current EM process from technical aspect

3.2.2 Integration events and monitoring

Integrations are an important part of the case company's add-ons to the provided applications. Different integrations allow for expansion and larger scale usage of the system. Integrations can be created to almost any other application or software, and thus the use cases vary a lot. This also makes systems more error-prone since there are more variables and systems that generate data and errors.

These integrations cause a large amount of the total events generated from monitoring. This can also be seen in the statistics of created and opened tickets in the service desk.

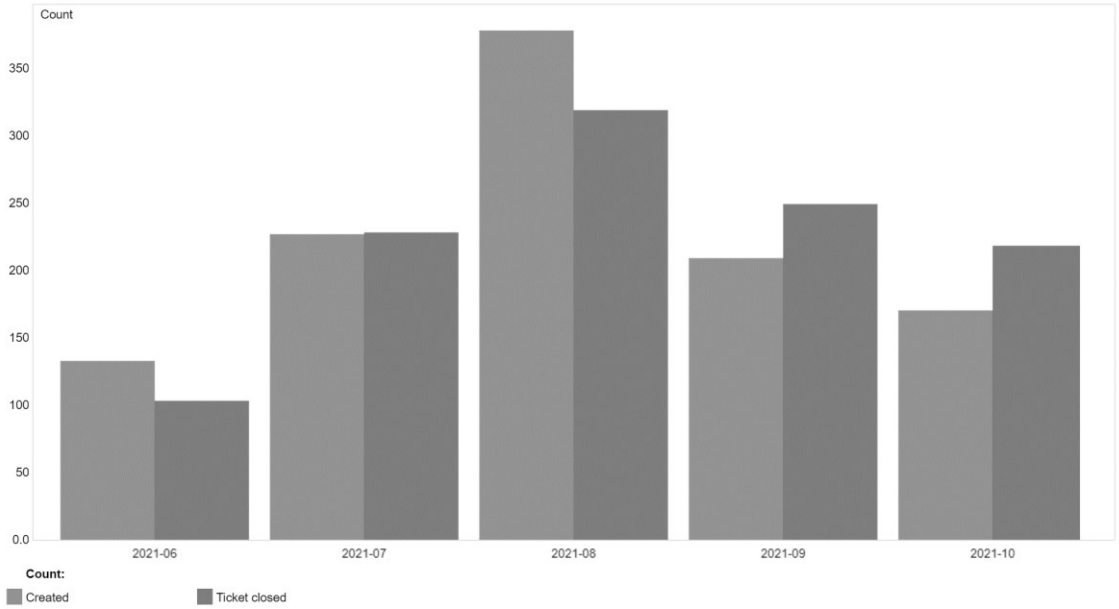

Figure 5: Opened and closed integration event related tickets monthly from the last 4 months.

Figure 5 above shows how many integrations event tickets have been opened and closed in the SD as incidents from the last 4 months. The quantity of opened and closed tickets differs to some extent, depending on the nature of the event. The average amount of opened incidents related to integration errors is about 200.

Current events generated from integration monitoring, includes data such as information about what integration or process is in question, a link to the integration platform and the error message itself. More data is collected in the integration platform, mut not generated to the event. This data is then sent as an event to the service desk.

3.2.3 Cloud events and monitoring

Cloud Ops (Cloud Operations) is a core business function in the case company. They manage all cloud-related operations, such as new customer onboarding, platform and capacity management, release deployment, and second level support for cloud related incidents. An interview with the case company's employee who oversees daily operations in the cloud-team with the title Team Lead & Product Manager was conducted to gain an understanding of how the Cloud Ops is related to event management.

In the interview, the current EM process in terms of Cloud Ops was described shortly as following; "Monitoring data is collected and utilized to applied extent for event management at this moment." and "There are many different processes and operations that uses event management, but they are on different maturity levels." (Cloud Ops Team Lead, 2021).

3.2.3.1 Cloud server storage monitoring

Cloud server storage monitoring is a commonly monitored event for the case company. All cloud servers provided, have a specific amount of storage on which the applications are running on. This storage depends on the customer needs including environment size, amount of stored data, number of logs generated, configuration complexity etc. A threshold is set to the monitoring system to monitor the server storage. When this threshold is crossed, which in this case is gigabytes (GB), the monitoring system creates an event of it, and sends it to the case company's ticketing tool, which creates an incident. The incident created, goes through the same process described in the section "3.2.1 Service Desk Operations", with some divergence in the included data. If the monitored event is considered critical, the monitoring system directly sends an alert to a third-party call-service, which informs the on-duty personnel of the cloud team.

3.2.3.2 Certificate monitoring

Certificate monitoring is another commonly related practice in the current EM process. Cloud server certificates is a way to authenticate server identity to the client. In this case, the threshold is a date instead of GB limit. Same process applies here as in storage monitoring, which is described above. In addition, an integration has been built between the monitoring system and the case company's communication tool, which allows for notifications of these events straight to the correct team and channel or person.

3.2.3.3 Environment updates

Environment updates is an important part of Cloud Ops. Updates are done to applications on a regular basis, following a release schedule and roadmap of features, bug-fixes, and other updates. When updates are done to customer environments, a lot of data is collected in forms of logs and events. The case company uses two different monitoring tools, one which monitors events, and a second which visualizes logs. By visualizing logs a great deal if information can be seen automatically, which helps the cloud- and QA-team to find any problems.

Events are generated from servers, server platforms, applications, and logs currently with the monitoring software. However, these events are not utilized automatically. Only manual checks are performed at the time of updates and in some cases, logs have been visualized to show errors in the logs.

3.2.3.4 Capacity management

The purpose of capacity and performance management is to ensure that services achieve agreed and expected performance, satisfying current and future demand in a cost-effective way. Capacity management related events are generated from operations, such as, Mass migrations in blade servers. As in the environment

updates, these events are not utilized in a common EM process in the case company.

3.3 Strengths and Weaknesses of current EM process

This section discusses the strengths and weaknesses found in the current EM process. These findings are a result of interviews with key employees of the case company, discussions, and internal investigation. First, the strengths are described which can be found in Table 4 below.

Table 4. Strengths found in EM process

#	**Strengths**
1	Possibility to react to events proactively
2	Automatic generation of incidents from events
3	Solving event related incidents before they become problems.

The listed strengths of the current event management process are not too long. Both the SD Team Lead and the Cloud Ops Team Lead commented the same strength and the importance of being able to proactively react to different events before they become issues or problems. Also, by this process being automatic, even its shortcomings, the current EM process saves a great deal of time if it was done manually or in retrospect.

Table 5. Weaknesses found in EM process

#	**Weaknesses**
1	Unnecessary incidents are created from events, making the EM model too heavy.
2	Lack of data for event related incidents.
3	Automatic linking missing between event related incidents to each other and to KB-articles.
4	Reporting of event related incidents is not truthful because manual work affects the integrity of the data.
5	Negative affect to efficiency regarding lack of standardized and centralized EM process.
6	Lack of forecasting possibilities. For example, upgrades of customer environments.
7	Too much manual work in EM process for SD

One of the main shortcomings with the current EM process were that in many cases, too many incidents are created, many of which, are duplicates or otherwise not relevant or only informative events. This has mostly to do with integration related events which were described in section 3.2.2 *Integration events and monitoring.*

in contrast to too many incidents created from events, not all events are utilized at all in the current EM process. This leads to less efficient operations of cloud related processes, as for example updating of environments and creating hotfixes.

3.4 Summary of CSA

To summarize the CSA of the case company's EM process, description of the overall process on high-level, generated workload, strengths, and weaknesses and lastly the gaps found are described.

The service desk is the centre of event management process in the case company. A lot of events are generated from different operations and goes through the SD. Incident management practice is used in combination with event monitoring. All events created from different monitoring tools, are sent via email to a ticketing tool where they are generated as incidents. Different events contain different data, and newly created incidents needs to be categorized, linked, and corrected before the handling. When this is done, the incident gets either escalated or solved directed by the SD agent.

The total amount of events generated was difficult to estimate, since not all events are generated as incidents and some unnecessary or duplicate events are deleted when they come in. However, integration related events are the majority of created incidents. An estimation made in the section "3.2.2 Integration events and monitoring" shows that approximately 200 incidents are created monthly only from integration related events. In addition, cloud related events were estimated to be around 40 incidents monthly from the interview with Cloud Ops Team Lead. These numbers translate to one person working partly or full-time only handling EM related incidents.

Both strengths and weaknesses were found in the case company's EM process. Biggest weaknesses were that too many incidents are created, the data is not sufficient for automation or reporting and they require a lot of manual work to be handled. In other words, gaps were found in those areas which can be filled with a developed EM process.

These shortcomings shows that there is a need for development of the EM process in many areas. The goal of the study is to make the SD agents workload

easier by implementing a developed EM process. This can be done by reducing unnecessary creation of incidents, enrichening the event data, and automating the workflows in the EM process. The proposal will be presented in the 5th chapter “5 Proposal and implementation” of this study, in addition with the technical implementation.

4 Literature review

This section discusses relevant literature, available knowledge, and best practices, used in this study. In the first chapter, ITIL, best practices and Architecture is discussed from different angles. The service desk function is described to provide a clear understanding of how it ties everything together and is the centre point of event- and incident management in the ITIL framework. Next event management is discussed, including ITIL best practices and architecture from the IT4IT reference architecture 2.1. Lastly incident management is described briefly. The second chapter takes a look at Enterprise Application Integration, going through different integration styles. In the third chapter application programming interfaces are briefly discussed. Finally, a conceptual framework is presented, which captures the study's key elements from the theory and the relationship between them.

4.1 ITIL, Best Practices and Architecture

The Information Technology Infrastructure Library, abbreviated ITIL, is an adaptable framework for managing services within the digital era, as Axelos (2021) describes it. Continuing, ITIL v4, which is the latest evolution of the framework, helps to optimize digital technologies to co-create value with consumers, drive business strategy, and embrace digital transformation, through different best practice modules. The latest version builds on the former versions, providing comprehensive, practical, and proven guidance. The support of traditional SM practices such as Incident Management is a key part, but also technologies such as Cloud, Automation, and AI can be aligned. (Axelos 2021)

According to the Cambridge Dictionary, a Best Practice (BP) is described as "A working method or set of working methods that is officially accepted as being the best to use in a particular business or industry." ITIL 4 has its own best practices, which are described in detail in the ITIL 4 Publication from Axelos (2019).
Taking a look at the ITIL framework, which can be displayed in a

4.1.1 ITIL service lifecycle

Event tough ITIL v4 is the newest evolution of the framework, older versions are still valid. The newer versions build up and complements the older versions, such as ITIL v3, or 2011 edition.

The ITIL service lifecycle, introduced in ITIL 2011 edition, displays the lifecycle of the service as seen in figure 6 below. The service lifecycle has been carried over and evolved since the 2011 edition but serves the same fundamentals in newer versions such as ITIL v3 or V4. The ITIL service strategy can be divided into 3 areas: service operation, service design, and service transition. These three areas include processes or practices, such as incident management under service operation, service level management under service design, and knowledge management under service transition only to name a few. The lifecycle adapts a continual improvement process, to improve the provided services in an iterative manner. (ITIL service operation, 2011)

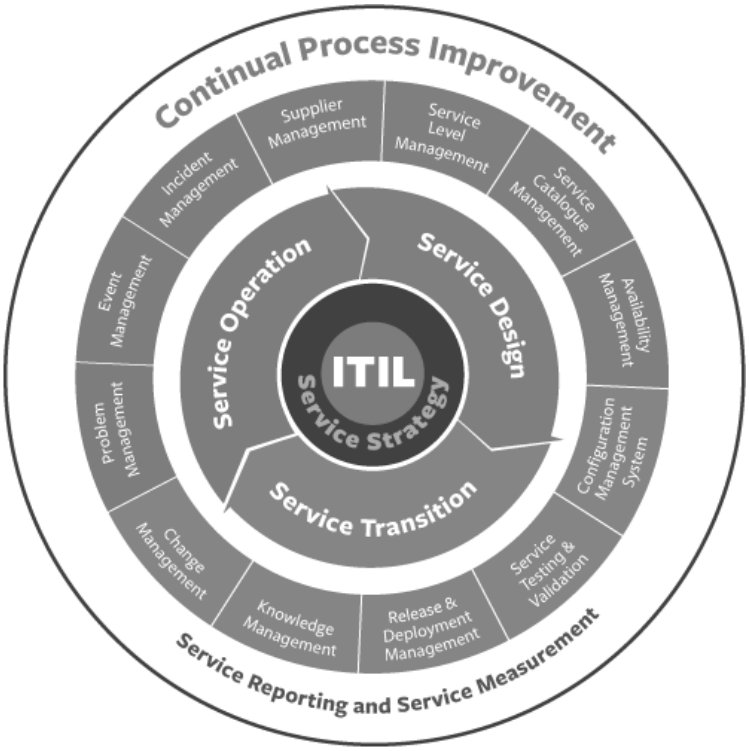

Figure 6: ITIL service lifecycle (BMC, 2020)

4.1.2 Service Desk

ITIL 4 Foundation, describes the service desk as following; “The purpose of the service desk practice is to capture demand for incident resolution and service requests. It should also be the entry point and single point of contact for the service provider with all of its users” (SERVIEW GmbH and AXELOS 2019).

The ITIL 4 Foundation (2019) continues to describe that service desks provide a clear path for users to report issues, queries, and requests, and have them acknowledged, classified, owned, and actioned. The service desk function can be a mix of different working methods, combining for example different skillsets, people, bots, working virtually or locally. Regardless of the model or the combination, the function and value shall remain the same.

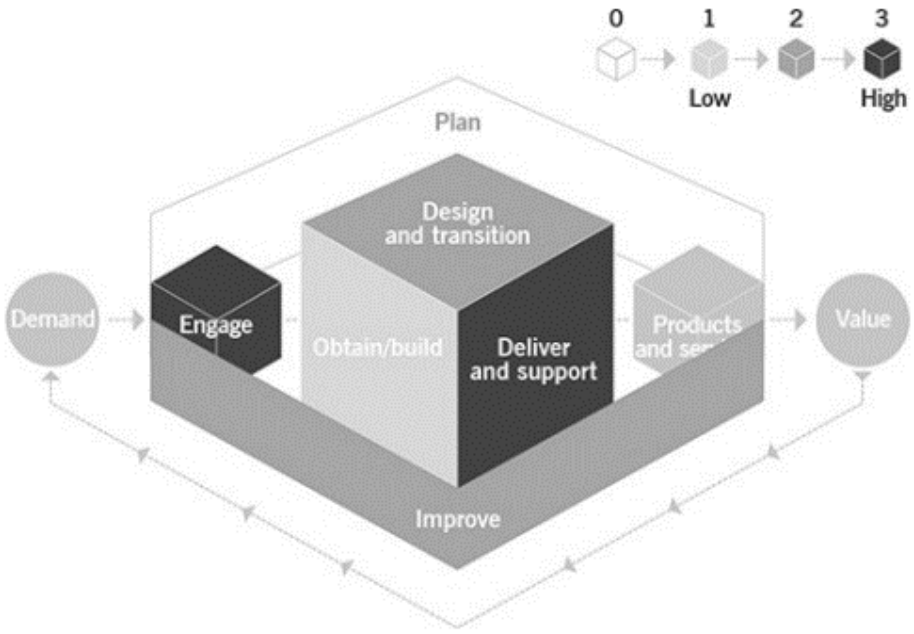

Figure 7: Heat map of the contribution of the service desk to value chain activities (Axelos 2019)

Figure 7 demonstrates the SD's contribution in the different ITIL value chain activities. Here the level 3 activities, are the key areas to deliver value in the service desk. For the SD agents, the key skill is to be able to fully understand and diagnose the specific incident in terms of business priority, and to take appropriate action to get this resolved, using available skills, knowledge, people, and processes. The service desk is the coordination point for managing incidents and service requests (Axelos 2019).

According to Bayes, S (2019) in the paper published for the Service Desk Institute (SDI), key trends since 2017 has been automation and self-service capabilities. The top key innovations or improvements that the members have wanted in 2019, has been automation/AI with 43%, in contrast to 2017, when the percentage was only 6%. This demonstrates a huge change in the demand of service desk features and expectations.

The effect of digitalization and customer demands with increased automation and bots contributing to the service desk function and its workload, has led to gradual removal of technical debt, which in contrast has changed the SD to provide more support oriented towards people and business, rather than technical issues according to Serview and Axelos, in their joint ITIL 4 publication (2019).

4.1.3 Event Management

An Event is defined as "Any change of state that has significance for the management of a service or other configuration item (CI). Events are typically recognized through notifications created by an IT service, CI,
or monitoring tool." according to Serview and Axelos (2019).

The purpose of EM practice, is to systematically observe services and service components, and to record and report selected changes of state identified as events. Infrastructure, services, business processes, and information security

events are identified and prioritized by the practice after which an appropriate response is established. This includes responding to different conditions that could lead to potential faults or incidents. (Serview and Axelos 2019)

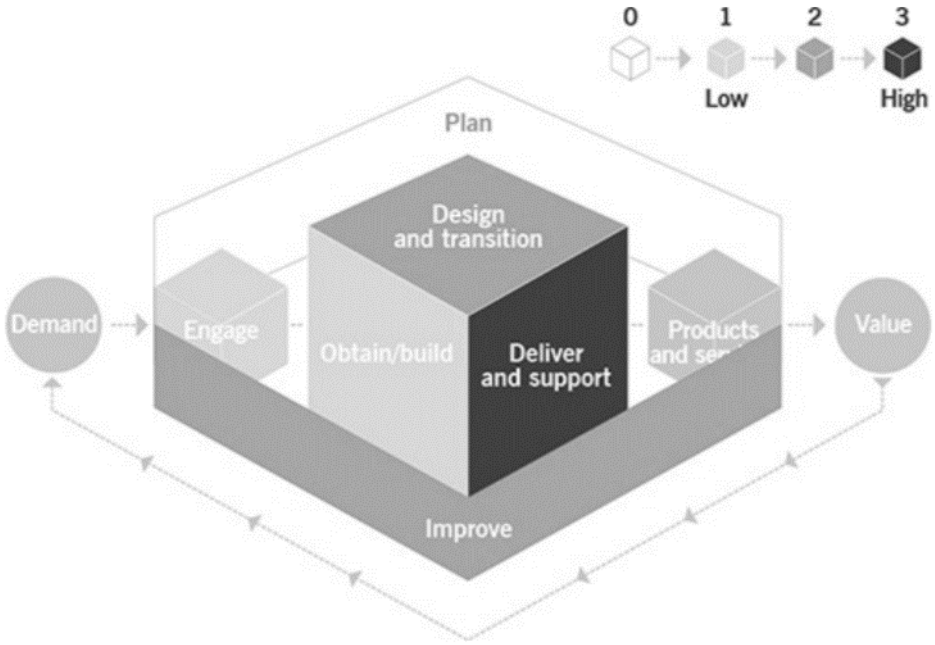

Figure 8: Heat map of the contribution of monitoring and event management to value chain activities. (Axelos 2019)

Axelos (2019) displays EM in the value chain heat map as seen in the figure 8 above. In contrast to the heat map for SD, only Deliver and Support is marked as high. The practice guides how the organization manages internal support of identified events, initiating other practices as appropriate in the deliver and support value chain activity. (Axelos 2019)

Monitoring is systematic observation of services and CIs, detecting conditions with potential significance to create an event. According to Serview and Axelos (2019), Monitoring should be performed in a highly automated manner, and can be done actively or passively. The organization defines what is considered and event and how it is handled, depending on its significance and impact. This could be a threshold value that is crossed, such as a date or an indicator of storage or capacity left. EM focuses on recording the monitored changes in the system,

which the organization has defined. In some cases, no other action is needed than continuing to monitor the situation.

Events have different level of significance, which require specific actions. Serview and Axelos (2019) classifies events as informational, warning, and exceptions. The 3 different types of events require their own actions. Informational events do not immediately require any actions but should be observed and analysed when more data is collected from them. This can uncover useful information that can help to better the service for example. In contrast, warning events requires action before any negative impact on the service or business is experienced. The third type of event, exception events, indicate that a breach has been identified in the monitored service or CI, and requires immediate action. (Serview and Axelos 2019)

The key activities that should be addressed according to Serview and Axelos (2019) are:

- Establishing the monitoring strategy
- Identifying the services, Cis or other components that should be monitored
- Implementing and maintaining monitoring
- Establishing and monitoring thresholds and other criteria
- Establishing and maintaining policies for how each type of detected event should be handled
- Implementing processes and automations required to operationalize the defined thresholds, criteria, and policies

The Serview and Axelos ITIL publication (2019) emphasizes automation and how it is the key to successful monitoring and event management. Here, more than in other practices automation is crucial. Automation should also be used for correlation of events.

4.1.3.1 IT4IT Reference Architecture for EM

The Open Group IT4IT Reference Architecture version 2.1 is and reference architecture that is describes as following by the Open Group (2017); “The Open Group IT4IT Reference Architecture is a standard reference architecture for managing the business of IT. It uses a value chain approach to create a model of the functions that IT performs to help organizations identify the activities that contribute to business competitiveness.”

Event Functional Components (EFC) purpose is described in the reference architecture 2.1 (2017) as managing events through the event lifecycle for events that occur on any IT service. Included in the event lifecycle are detecting, categorizing, filtering, analysing, correlating, logging, prioritizing, and closing events. However, the event lifecycle is not limited to those activities. Events can serve as initiators of incidents.

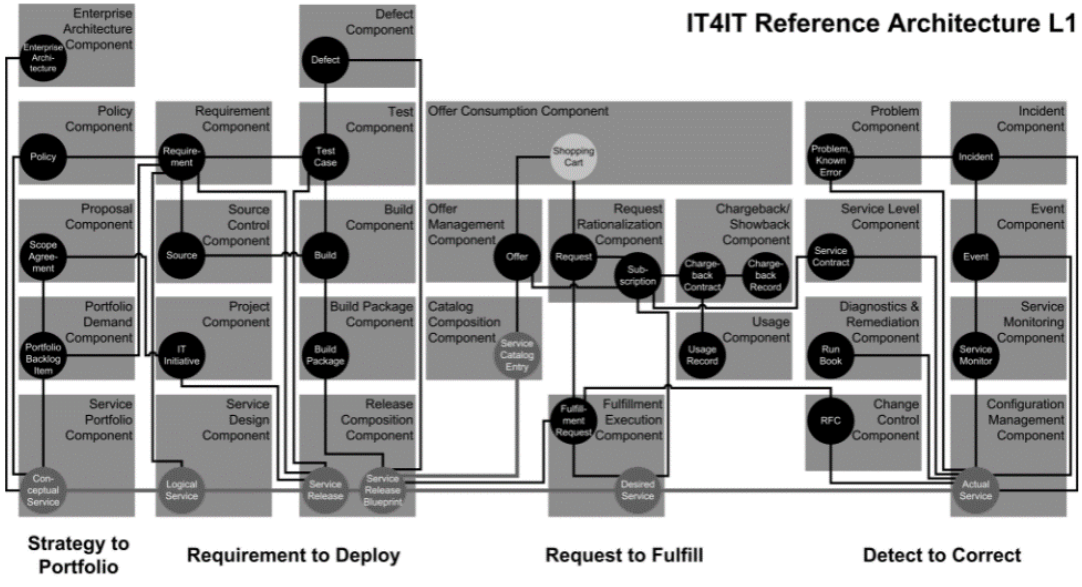

Figure 9: IT4IT Reference Architecture version 2.1 L1 diagram (Open Group 2017).

The Open Group IT4IT Reference Architecture version 2.1, shows the Event Components (EC) relation in the architecture L1, as seen in the Figure 9 above. Taking a closer look, the relation between an EC and Incident Component (IC)

can be seen in figure 10 below. EC and IC have a N:M relation between them, which means that they both might have many relationships between each other.

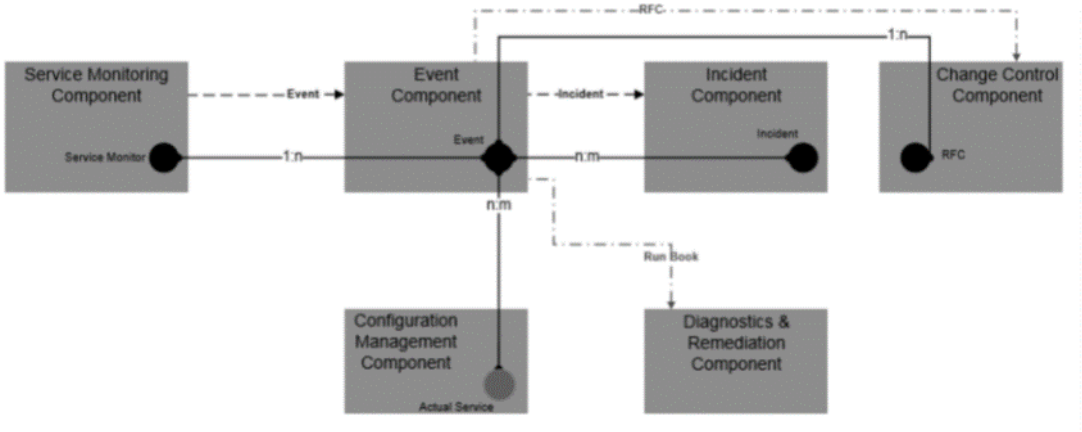

Figure 10: Event Functional Component Level 2 Model (Open Group 2017)

The architecture model of the Event Functional Component is illustrated above in figure 10. From left to right, The Service Monitoring Component (SMC) generates and event, which has a relationship with IC, Configuration Management Component (CMC) and Change Control Component. An incident can be created of the event which is then handled with appropriate manner. The CMC is the actual service that the event is linked to.

4.1.4 Incident Management

Another key ITIL practice that the service desk operates, is Incident Management (IM). Axelos (2019) defines an incident as; “An unplanned interruption to a service or reduction in the quality of a service.”. The purpose of IM practice according to Axelos (2019) is “to minimize the negative impact of incidents by restoring normal service operation as quickly as possible.”. Every incident should be recorded, logged, and handled according to the service level agreement between the service provider and customer. Also, classifications should be agreed on how to prioritize and classify, depending on the business impact in realistic expectations. Highest prioritized incidents with the largest business impact should be resolved first. (Axelos 2019)

The appropriate management and resources should be allocated to resolving different types of incidents. Low impact incidents should be resolved efficiently to not consume too many resources and in contrast, high impact incidents may require more resources and management. More complex, major or information security incidents have usually their own process on how to manage them. Incident should be or are usually stored in an IT service management (ITSM) tool. The tool should provide linking between CIs, changes, problems, know errors and knowledge base articles. This helps to manage the complete lifecycle of the incident, providing efficient diagnosis and fast recovery. Automation and Artificial Intelligence (AI) can be used in modern ITSM tools to help solve the incident. (Axelos 2019)

Effective IM requires often high level of collaboration between teams, groups, and individuals, depending on the complexity and type of incident. All related persons need to understand the IM process clearly. This can involve teams such as the service desk, including first- and second level support, technical support, application support, and third-party vendors. Sometimes, the incidents are resolved by the users themselves using self-help. However, this varies a lot on the maturity level of the user or organization. Escalation is a commonly used way of handling incidents. This means that the SD for example, escalates the issue to another team or group, that could be second level support, a specialist in the field that the incident is related to, or a vendor or partner that supplies the service.

To summarize, best practices in IM can be following:

- A formal process for logging and managing incidents should be in place.
- Incidents should be logged into a ITSM system.
- Incidents should always be classified and categorized.
- High level of collaboration between teams is important for efficient IM.
- Automation or scripts that collect initial information to resolve the incident in the initial contact can lead to directly solving simple incidents.

4.2 Enterprise Application Integration

Linthicum, D. (2003) describes Enterprise Application Integration (EAI) as follows: “At its foundation, a response to decades of creating distributed monolithic, single-purpose applications leveraging a hodgepodge of platforms and development approaches”. In other words, unrestricted sharing of data and business processes among any connected applications and data sources in the enterprise. (Linthicum, 2003)

All companies have a magnitude of different systems and applications that are used for various purposes. All these applications, being enterprise or other applications, produces data in masses. Some of the data is needed in other applications to enable specific processes and workflows. This is where EAI comes into question. By integration of different applications data transfer from one system to another is possible.

In *Enterprise Integration Patterns*, Hohpe and Woolf (2003) provide different patterns for EAI, which has been widely used. These patterns combined could be called a pattern language even. Hohpe and Woolf (2003) distinguish four main integration styles. These are File Transfer, Shared Database, Remote Procedure Invocation and Messaging, which is the most powerful, while the others are also regularly used. We’ll take a brief look at each pattern.

4.2.1 File Transfer

File transfer is the first of the four different patterns where integrators take the responsibility of transforming files that the other applications consume, to different formats. A diagram of the File Transfer integration style is shown below in Figure 11. Modern applications tend to use XML or JSON formatted files.

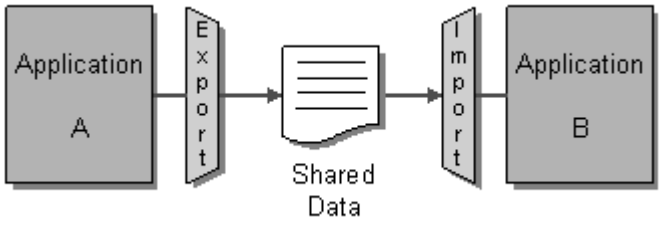

Figure 11: File Transfer integration style (Hohpe & Woolf 2003)

4.2.2 Shared Database

The second integration style is called Shared Database. This style uses a shared database between the applications where integration is wanted. Consistency between the data is guaranteed since its stored in the same location and does not need to be transformed. This style ensures that the applications are always consistent by utilizing the same shared database. Usually, large ERP or CRM systems use this kind of configuration. A transaction management system usually facilitates simultaneous updates to the data, where the time between any errors is small and easier to find and fix. The shared database integration style can be seen in the figure 12 below.

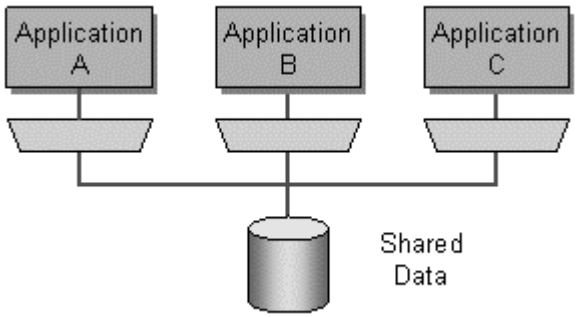

Figure 12: Shared Database integration style (Hohpe & Woolf 2003)

4.2.3 Remote Procedure Invocation

The third integration style by Hohphe and Woolf (2003) is Remote Procedure Invocation. Here, for each application, a large-scale object or component should be developed with encapsulated data. Furthermore, an interface that allows the

applications to interact with the running applications should be provided. This integration style applies the principle of encapsulation to integrating applications. In other words, if an application needs some information that is owned by another application, it asks for it directly and if the data needs to be modified, it does so to make a call to another application. This way each application can maintain the integrity of its own data and alter it without affecting other applications. Figure 13 below illustrates the Remote Procedure Invocation.

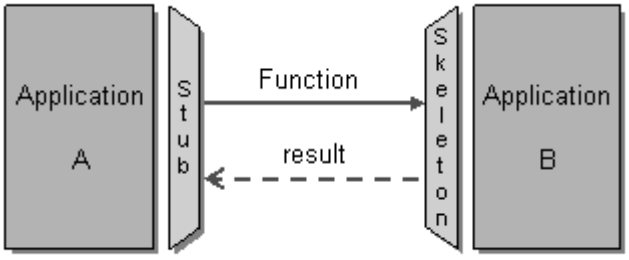

Figure 13: Remote Procedure Invocation integration style diagram (Hohpe & Woolf 2003)

4.2.4 Messaging

The fourth and last integration style in EAI described by Hohpe & Woolf (2003) is Messaging, where packets of data is frequently, immediately, reliably, and asynchronously transferred, using customizable formats. This style does not require all systems to be up and running at the same time, or the fact that working with a remote application is slower. Each of the applications is connected to a common messaging system where messages are exchanged. For example, if application A needs to send application B data, application A sends a message to the common messaging system or the Message Bus, from where then application B polls it.

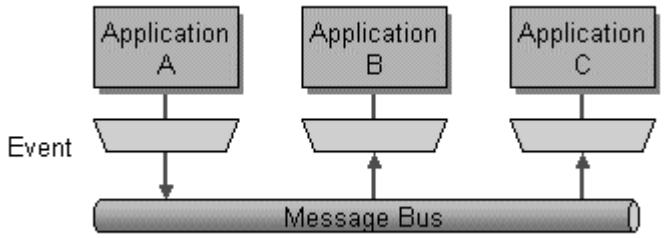

Figure 14: Messaging integration style diagram (Hohpe & Woolf 2003)

These four different integration styles cover the different integrations in EAI. However, these styles are mixed and matched depending on the systems and wanted architecture.

4.3 Application Programming Interface

In this section, we'll take a brief look at Application Programming Interfaces or API for short. De, B. (2017) describes an API in *API Management* as an "software-to-software interface that defines the contract for applications to talk to each other over a network without user interaction.". With APIs applications can be installed and accessed from different devices and locations. There are different types of APIs, including Private/Internal, Private/Partner and Public APIs. In this section of the study, we'll only examine private and partner APIs.

A private API is behind closed doors and can only be accessed by an organization itself. These are mostly used for application integrations or B2B applications with partners for example. These are made private by providing security and access control, and by restricting the APIs use to a limited number of developers or partners. (De 2017)

Below in figure 15, a visualization on how APIs work in a general manner. A web application is connected to the internet and sends a request to the API, which is connected to the webserver. A response is sent back to the application from the API.

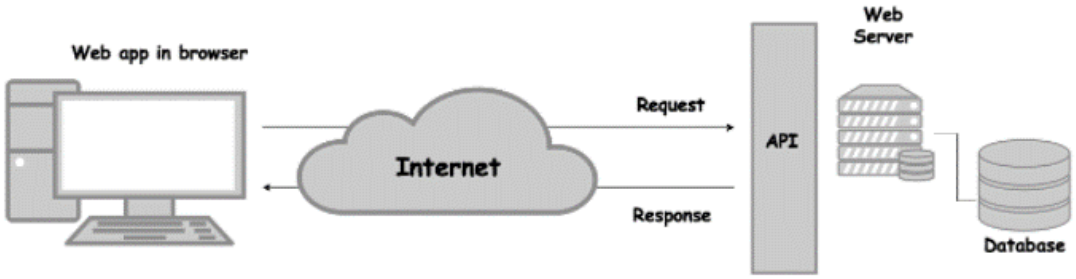

Figure 15: Visualization of API (Ahamed 2020)

4.3.1 Web API

Web API is a type of API, however, differing from a web service, by being a subset of a web service according to De, B. (2017). Defined by W3C Web Service Architecture Working Group, that web service architecture requires specific implementation of web service, in which an interface is described in a machine-processable format, specifically WSDL. Other systems interact with the web service using SOAP (Simple Object Access Protocol) messages that are typically transported using HTTP with an XML serialization and other web-related standards (De 2017). Web APIs has recently started to be replaced with REST-based communications, or REST API, which stands for representational state transfer. REST API architecture differs and does not need XML-based web service protocols like SOAP and WSDL to support their interfaces.

4.4 Event correlation

As events are and can be generated from various sources and systems in masses, a technique is needed for making sense of these events, so the actual issue is easily and quickly detected. If a so-called event burst happens, meaning that if many monitored objects reports a fault or error at the same time, systems or administrators gets flooded by incoming events, even if the actual cause is only one object that is down, which makes it look that all other related objects are also down. Gruschke, B (1998) describes this well in the *paper Integrated event management: event correlation using dependency graphs.* He takes an example,

describing a scenario where four hosts are connected to a router. Two of these hosts are sending an alert about lost connection to their server respective server. By knowing the network topology, the event correlator can point to a faulty router. Example visualized in figure 16 below.

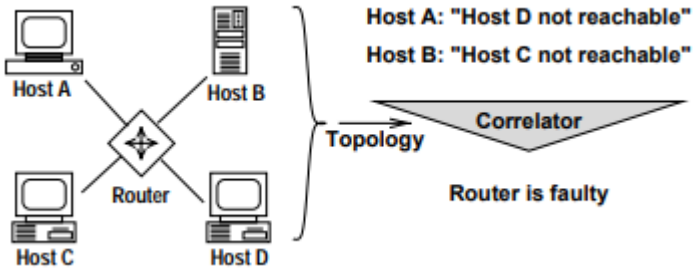

Figure 16: Example of event correlation. (Gruschke 1998)

However, event corelation is not simple. Gruschke, B (1998) identifies four different open issues related to event correlation:

- The number of managed systems and their complexity makes it hard to develop a holistic event correlation solution.
- High change rate between the managed systems, meaning new technologies and configuration changes makes events hard to interpret.
- Knowledge needed for correlation is usually highly distributed and the knowledge needed for resolving the events is between many units and persons.
- Inconsistencies between the event correlator and monitored system can mislead.

These above-mentioned open issues make event correlation very hard from a software developer's perspective – maintaining a complex program without making mistakes.

Integrated event management is the solution to tackle these issues, according to Gruschke, B (1998), where an event correlator is us used in conjunction with existing event management system. Below is an illustration in figure X of an event

correlator connected to a management system. This way, the correlator can retrieve needed data and knowledge from the management system.

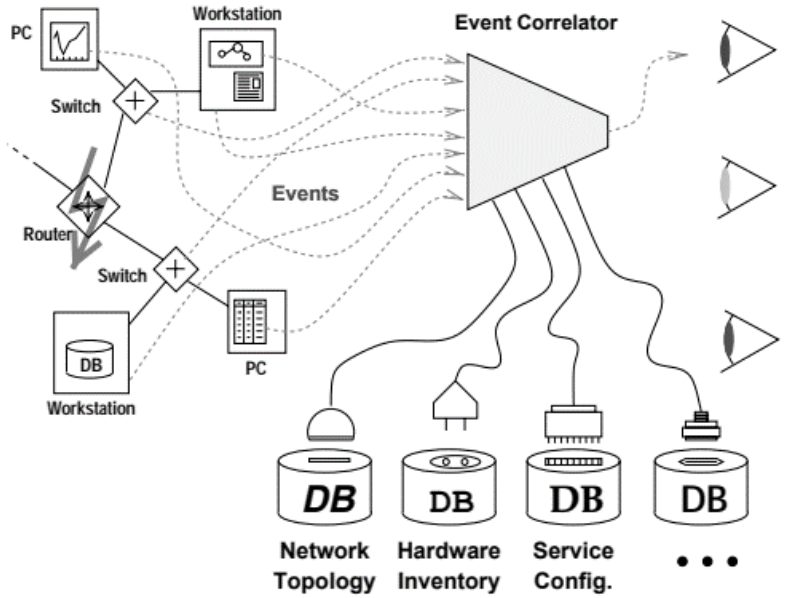

Figure 17: Event Correlation connected to a Management System. (Gruschke 1998)

To summarize, event correlation is a way to bundle large amounts of different events caused from the same fault. This helps to find the root cause of the issue, and to solve it quickly, without having to go through all the generated events which most of are not relevant to the causing issue.

4.5 Conceptual framework

A conceptual framework was created to illustrate the theory used in this study, which is shown in Figure 18 below followed by a description of it.

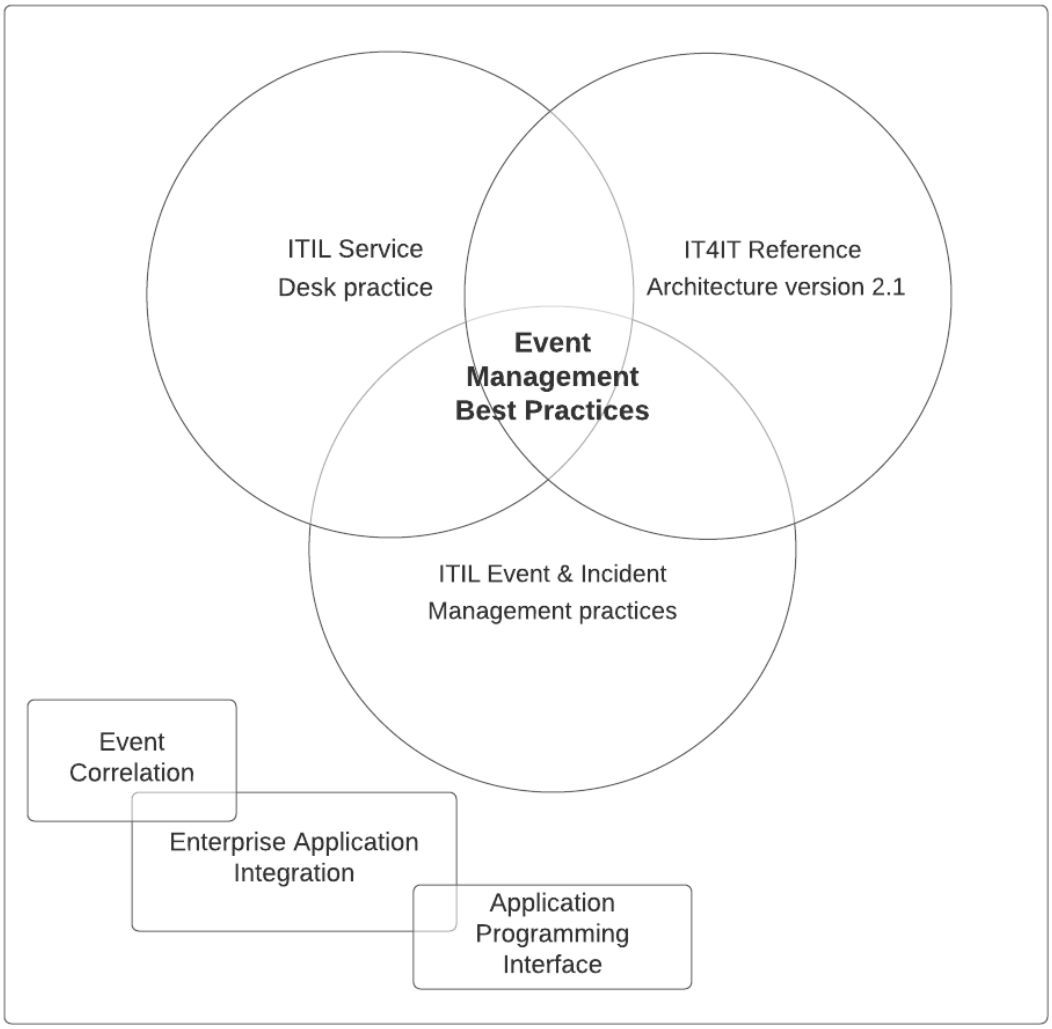

Figure 18: Conceptual framework of the study.

The theory of this study is based on 6 elements, of which 3 main elements create the core, and 3 complements it to create a holistic framework that leads to the best practices used for this study's proposal of an EM process. The first element, seen in the upper left-hand corner of figure 14 above, represents the theory about the service desk practice from Axelos and Serview (2019). This is essential, since as described in chapter 4.1.1, the SD is a single point of contact for customers and end users. All incidents and service requests, including events, goes through

the SD. The SD practice includes incident- and event management (Axelos 2019). The second element, IT4IT Reference Architecture version 2.1, seen in the figure 14 above in the upper right-hand corner, describes the architecture of event management in the whole reference architecture. Understanding this is important to gain a perspective on how it relates to other practices, such as incident management. The third core element, takes a deeper look at event management and incident management practices in the ITIL framework by Axelos and Serview (2019). Understanding EM and IM practices are crucial to create an EM process. These 3 core elements create the Event Management Best Practices.

Enterprise Application Integration and Application Programming Interface theory is examined in the literature study, which creates the 2 elements that can be seen in the Figure 18 above. These are not directly attached to the EM best practices, but is necessary to understand, since integrations will be used in the developed EM process. Lastly, linked also to integrations, Event Correlation is discussed, to understand the subject, and how it should be used. These three elements complement the whole framework to create a holistic proposal.

All of the above-described elements of the conceptual framework are important fields for this study and will be utilized in the following chapter where the proposal of a developed EM process will be built and implemented to the case company.

5 Proposal and implementation

In this chapter of the study, a proposal of the developed EM process is built. The proposal will be based on the data collected from the CSA, acknowledging the gaps found in the current EM process. The best practices to build the EM process will be applied from the chapter *4 Literature Study.*

5.1 Overview of Proposal Building

This section describes how the proposal will be built by providing an overview of the steps required to create a proposal of a developed EM process for the case company. This includes what theory is used from the literature review, how the theory is used, and finally, how it becomes the proposal.

Below, in figure 19 is illustrated the linkage between the areas related to the proposal building. Based on the CSA, the first area to be developed, is the EM process itself, which needs to be redesigned to accommodate the requirements of the case company, and also to improve the weaknesses found in the CSA. The requirements will be displayed in the next chapter *Findings of Data 2.* Selected theory for this area includes chapters *4.1.1 Service Desk, 4.1.3 Incident Management, and 4.1.2 Event Management.* The theory about service desk will be used firstly, to understand how the SD works, and furthermore, what's the relation to EM & IM in the ITIL framework. Theory related to IM is also used to understand the process in more detail, and to suggest best practices that will be used in the proposal.

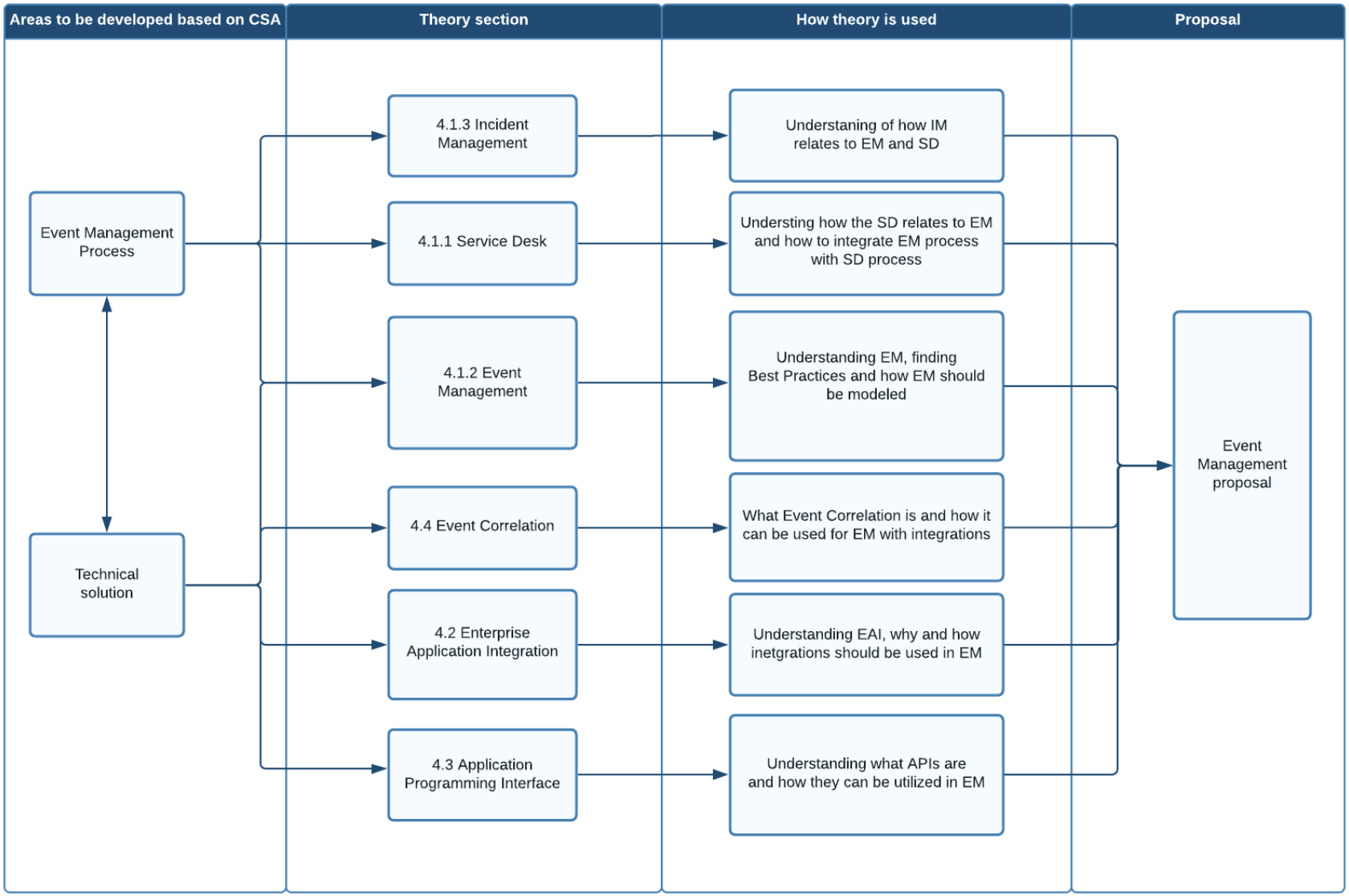

Figure 19: Areas to be developed based on CSA and linkage between them.

Lastly, theory about EM will be used to understand the practice in general, and to apply best practices to the developed proposal. These will form a developed process.

The second area to be developed is the technical solution for the EM process. To complement the developed EM process, the technical solution needs to be considered, for a complete EM process that can be implemented and duplicated easily. In this area, theory about EAI and APIs will be used to understand the role of integrations between applications to be able to plan the needed integration for the EM process, together with key stakeholders from the case company.

Both areas, the EM process and the technical solution need to be developed, based on findings from the CSA. By applying theory in the above-mentioned ways visualized in figure 19, a developed EM process will be built.

5.2 Findings of Data 2

This section will showcase and discuss the finding from data 2. Two workshops were held with key personnel of the case company related to event management, service desk, and Integrations.

The first workshop was held with the SD Team Lead and Senior Architect & PM of the case company. In this workshop, the requirements for the developed EM process were discussed. Also, general planning and discussion about the developed EM process occurred. The results of the first workshop will be displayed in the next chapter 5.2.1 *Requirements for Event Management Process.* The second workshop for Data 2, discussed the integration process required for the developed EM process. Together, with a Senior Integration Consultant from the case company, the integration process was planned and documented as a flow chart. The results from this workshop will be displayed in the chapter *5.2.2 Integration process.*

5.2.1 Requirements for Event Management Process

In this chapter, the results from the first workshop regarding the requirements for the developed EM process will be displayed. Four main requirements were found by the case company that should be applied to the developed EM process. These requirements are displayed in the table 6 below.

Table 6: Results from workshop related to EM requirements.

#	Requirement
1	The EM process should be separated from the current IM process.
2	The solution should not be as demanding and should facilitate the SD agents work.
3	The developed EM process should enable accurate reporting capabilities.
4	An event correlation logic should be considered in the process.

The first, and foremost requirement were that the EM process should be separated from the current IM process. As seen from the CSA, an event always generates an incident in the case company's SM tool. An event can create an incident but should not always create one. Only if the event requires customer communication, escalation, configuration changes, or any other work, the event should be transformed into an incident. The second requirement is related to the current EM process being laborious for the SD agents. A lot of manual work is required in the current EM process and the developed process should try to minimize the amount of manual work with automation. In this way, the developed EM process should facilitate the SD agents work. The third requirement, is that the developed EM process should provide accurate data, to allow up to date reports for internal use, as well for customers. Finally, the fourth requirement were that the developed EM process should include event correlation logic, to detect the really important events from a large number of events generated.

These requirements will be considered in the proposed developed EM process in conjunction with best practices from the theory section of this study.

5.2.2 Integration process

Based on the requirements, best practices and workshops, the developed EM process will utilize an integration layer, to enable data transfer from the monitoring tool to the case company's SM system. A workshop was held with the case company's Senior Integration Consultant, where this integration process was discussed and planned. This integration process will be described and displayed in this section.

To enable data transfer and automatic event creation based on the monitoring from the monitoring system used by the case company, an integration layer is required between the monitoring system and SM system. This integration will be created using the case company's own integration platform, and by utilizing Web API to transport the data from the monitoring system to the SM-system.

The below displayed process displays the initial version of the integration process for the developed EM process. This process starts when an error occurs in the monitored system or application, where the integration is running. When the error occurs, the data from that error is transformed into an event. This data includes Integration ID, error message, a unique ID, and a description. The integration then continues to lookup for an existing, open event in the system. If no matches are found, the event gets imported to the case company's SM system via Web API. If a matching event is found, the integration updates the found event with the new information. This information will be set into a comment-type of attribute in the system. After either a match or no matching event is found, a check if the import was successful is done by the integration. When the import is successful, the event gets logged as an event data card in the SM system.

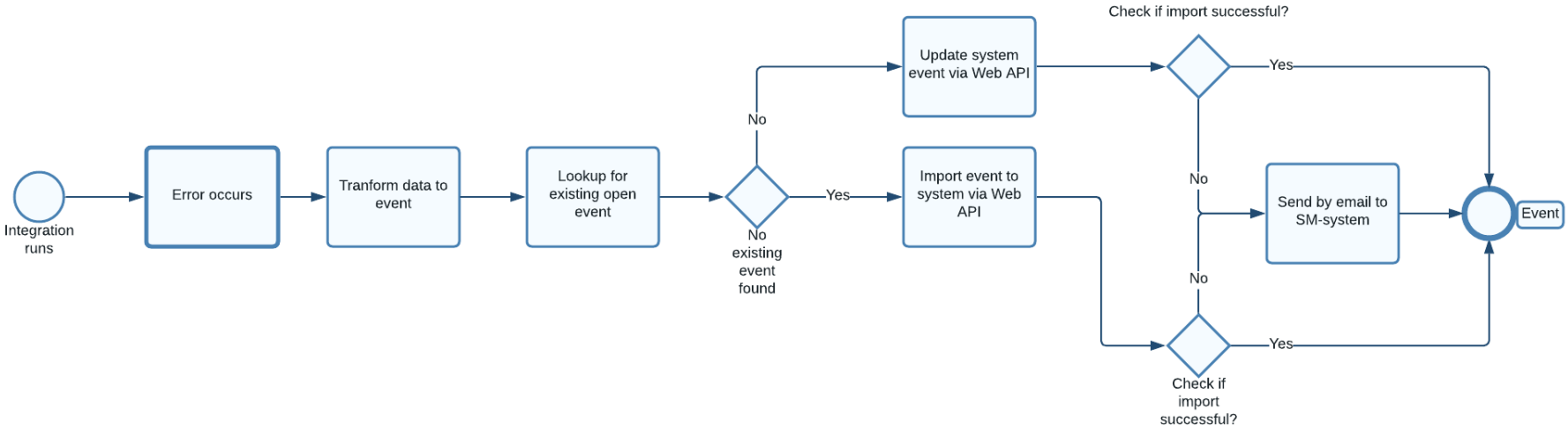

Figure 20: Developed integration process for EM.

If for some reason the import is not successful, the integration has a fail-safe, by sending an email with the data, which then will create an event. After the integration logs the error into an event to the SM system, a new workflow will begin for the handling of that event.

5.3 Proposal of Developed Event Management

This chapter will discuss the proposal of the developed EM process and give concrete steps how it should be implemented as well as including recommendations. As in the previous chapters, we took a look at how the proposal is built and the findings from data 2, this chapter will follow those guidelines and take the data into account when creating the proposal.

The first chapter will propose the event layer. The second chapter will discuss the activities related to the SD. After this, automation suggestions will be proposed, after which the EM process itself will be presented. Finally, all proposed aspects are summarized. Recommendations on actions to take to implement different steps of the proposed EM process will be also given after each proposal.

As seen from the CSA, the biggest issue that the case company encounters with the current EM process, are integration related events. This is derived from the nature of the integrations, as they create a lot of events. Taking a look back at figure 5 in the CSA, and from the interviews, we can see that amounts of integration events dominate all events created. This proposal will hence be focused on these integration events, however not forgetting other types of events. The process, templates and practices that will be proposed, are designed so that all events can be used since the solution should be uniform and holistic for all types of events, disregarding the origin of the event.

5.3.1 Event layer

As described in the CSA, the fundamental issues with the current EM solution, is that the data is unstructured and not enriched, there is no event correlation used, no integrations utilized, and most importantly, IM and EM share the same data template in the SM-system. This leads to that all created events are directly logged as incidents, even if they are only informative for example and should not be an incident. All created events from the monitoring system, should be first logged to an own Event-template, where they are stores and addressed. Currently, no such event template exists.

5.3.1.1 Templates

The first step, is to separate IM and EM into separate data-templates. This can be done by creating a new Event-template, which all events are logged into. This event template should be simple, quick to edit with only necessary metadata, that supports the Event handling process. The metadata is formed of attributes, or fields, that contains the data. These attributes should be carefully planned. The creation of a data template to accommodate attributes varies from system to system but is generally quite simple. A data-template will contain all attributes, automations, scripts, relations to other data-templates, and overall, everything related to the process it will support. Down below is an example of an event-template that could be implemented for the case company. The template includes classes, which divide the included attributes into logical sections. This template, is specifically designed for the case company, considering the current data model, templates, and other factors. However, it can be used as an example or as a baseline, depending on the system the EM is implemented to.

Table 7: Proposed Event-template

Event Template			
Class:	**Attribute:**	**Datatype:**	**Source**:
Event information	Description	Text field	Monitoring system
	Error message	Text field	Monitoring system
	Integration id	String	Monitoring system
	Unique id	String	Monitoring system
	Event time	Datetime	Monitoring system
	Event status	Static	Integration process
Customer information	Company	Reference	Monitoring system
	Related integration	Reference	Integration template
	Integration owner	Reference	Integration template
Relations	Related events	Reference	Manual input
	Related incidents	Reference	Manual input
Priority & Classification	Impact	Static	Related integration error
	Urgency	Static	Related integration error
	Priority	Static	Impact & Urgency
	Event type	Static	Related integration
Handling and resolution	Error handling	Reference	Related integration
	Comment	Worklog	Manual / Integration
	Auto-resolve	Static	Related integration
	Resolution	Text	Related integration / Manual
General information	Data card id	String	System, Automatic
	Created	Datetime	System, Automatic
	Updated	Datetime	System, Automatic
	Creator	Reference	System, Automatic
	Latest update by	Reference	System, Automatic

The first class, Event information, contains specific information about the event. Attributes included in the class are description, error message, integration id, unique id, event time, event status. All of the attributes, will be directly populated by the new integration process, described in figure 20.

The second class, contains information about the customer, where attributes like company, related integration, and integration owner are displayed. Customer company is updated by the integration. The attributes ‘Related integration’ and ‘Integration owner’ will have logic applied to them, where they will be populated if a match is found on the Integration-Template, which will be presented shortly. This can be with a simple script, that checks the two templates when the event is created. If a match is found, the script links the integration data-card to the event template.

The third class in the template named ‘Relations’, include attributes Related events and Related incidents. This is where all the relations to other possible events and incidents are filled if needed and displayed. The related incident attribute should have an option to transform the event to an incident if needed, meaning that the event data-card will be converted to an incident data-card, by making mappings from the event template to the incident template. This transform option can be then triggered manually or by automation.

Next is the Priority & Classification class. This contains attributes Impact, Urgency, Priority, and Event type. All attributes are static by datatype, meaning they have values in the attribute metadata showing as a drop-down menu for example. Impact and urgency should be defined for the event error in question. Values for both can be e.g., Low, Medium, and High. These values are copied from the Error-Template automatically when the related integration is linked. Priority is calculated based on the impact and urgency and can be displayed in an impact urgency matrix as seen in table 8 below. The highest priority being 1 and the lowest 5. Lastly the Event type attribute is shown in this class, which indicates if the event is informational, a warning, or an exception, as described in section *4.1.3 Event management.* This is also copied from the Error-Template.

Table 8: Urgency and Impact Matrix

		Impact		
Urgency		High	Medium	Low
	High	1	2	3
	Medium	2	3	4
	Low	3	4	5

The fifth class in the Event-Template is Handling and Resolution. This class contains information about how to handle the event, any comments, if the event should be closed automatically directly, and the resolution. The auto-resolve attribute is also copied from the Error-template, where the error is defined as a non-action required error which can be automatically closed. If this is selected, the error closes automatically by updating the status to closed and setting a resolution to the resolution attribute. The comment attribute is used for when the integration checks for any relating events. The integration searches for matching events, by combining integration id + error message + unique id. If a match is found, the integration updates the event un the comment attribute.

The last class is general information about the data-card. This contains information like data-card id, when the event was created or updated, and which user created or updated the event.

In addition to this new Event-Template, two other templates would be created as well. These are done to support the event-template, by providing either information about the error or integration in question. These templates are also necessary for separating the errors from the integrations, meaning one integration can have multiple error. When both are their own templates, data cards can be conveniently linked to each other.

The first of the two is the Integration template. This template stores the information of the different integrations that are used by the case company's

customers environments. This template is divided into 3 different classes. The first contains information about the integration. The second class displays the relations of the integration data-card, and the last one shows general information. Some of the information on this template is matching the event template, which allows copying of values directly to an event data card. These data cards are filled in advance, so when an event is created to related integration, the event template can the copy directly values from this template.

Table 9: Proposal of Integration-template

Integration Template		
Class:	**Attribute:**	**Datatype:**
Integration information	Integration id	String
	Integration name	String
	Integration description	Text
	Environment type	Static
	Customer	Reference
	status	Static
Relations	Related errors	Multivalue reference
	Documentation link	Link
	Integration owner	Reference
General information	Data card id	String
	Created	Datetime
	Updated	Datetime
	Creator	String

The third template that is created is the Error-Template. This template is used to record different errors caused by the monitoring tool from different systems. Most of the errors are already known and can be filled in on this template as data cards in advance, but also new errors are generated with evolving products and configurations. If new errors are found, they can be easily created into this template. These errors can be directly linked to the Event-Templates data cards

or events, or then first linked to an integration data card on the Integration template.

The Error-Template is divided into 3 different classes: Error information, Error description and priority and classification. The Error information class contains information about the error itself, such as Integration id, error id and the option of Auto-resolve, which was described earlier. The Error description class describes the error and how to handle it. This will be copied to the event, if required. This error handling information can be used by the SD agent to resolve the event, or also transferred with the event to the incident if necessary. The last class contains information about priority and classification. These values will be copied to the event when correct integration or error is found as described earlier. Same urgency and impact matrix should be used for the impact and urgency in this class, as seen in table 8 above.

Table 10: Proposed Error-template

Error Template		
Class:	**Attribute:**	**Datatype:**
Error information	Integration id	Reference
	Error id	String
	Auto-resolve	Static
Error description	Description	Text
	Error handling	Text
Priority & Classification	Type	Static
	Impact	Static
	Urgency	Static
	Priority	Static

The architecture in the system between these three templates are displayed in figure 21 below. The templates are all linked together from different attributes. Between the Event-template and Integration-template, a one-to-many relationship is formed, meaning one Event card can have only one Integration card, and Integration cards can have many Event cards linked to it. Between the Integration-template and Error-template a many-to-many relation is formed. Between the Error-template and Event-template a one-to-many relation is applied. To clarify, these are not database relations, but rather Template relations in the system.

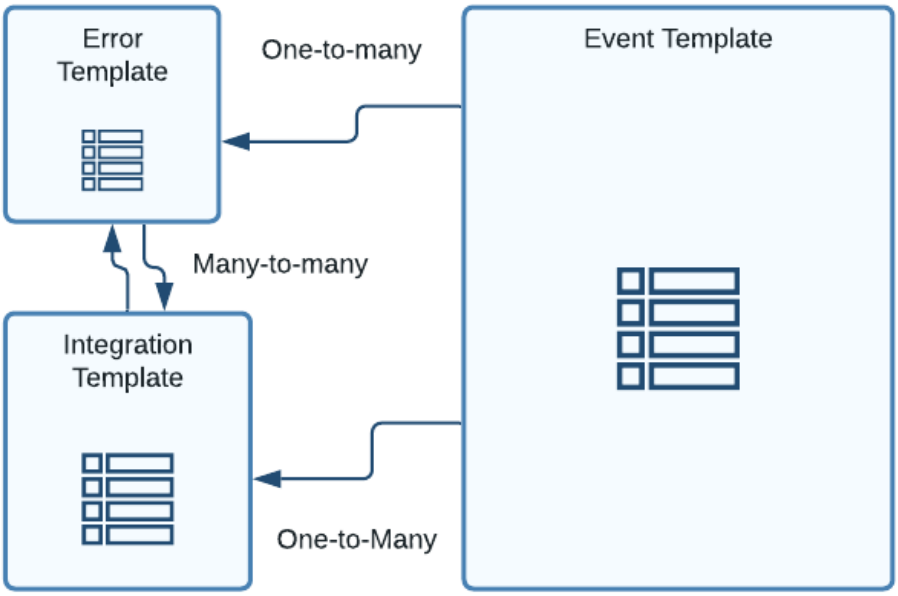

Figure 21: Architecture of Event-, Integration-, and Error-templates.

5.3.1.2 Data

The data is structured and enriched along the newly developed integration process. In this process, relevant data is collected and logged to the event and transferred to the event through Web API.

This way the collected data can be defined from the monitoring tool and more importantly the specified that can be transferred to the event generated in the

Event-template. This allows for uniform data for the event, giving good reporting capabilities, which the system already supports.

5.3.1.3 Event correlation

Event correlation will be utilized in the developed EM process. The correlator will be implemented into the integration platform and connected to the SM-system to get the needed knowledge, as described in chapter *4.4 Event correlation.* The correlation will bundle the events, before they are created as event data cards in the Event-template. The development of event correlation is not discussed in this study since the development of such a system is highly complex and is not in the scope of this study. However, it's recommended to use some kind of Event correlation in the EM process and technical solution, to filter out the main event if masses of events are generated. A third-party service or solution for event correlation can be an option, or it can be developed by the case company itself.

5.3.2 Service Desk activities

The CSA described the activities that the SD agents does currently with event related incidents. To recap, the SD agents need to categorize, classify, link, and correct the metadata of the generated event as an incident. This is all done manually and is time consuming. In the new developed EM process, the SD agents' activities change. Since the data is enriched and structured into the same format the metadata does not need to be corrected. Also, the classification of the event will be automatically fetched from the related Integration and Error templates. In cases of a new error, which have not been logged into the system before, the SD agent needs to create the error, and check for handling instructions for that specific error. This method improves the automatic linking continually and makes the SD agents work easier by time. To summarize, the SD agents need to quickly review the event, and decide if it needs to be created to an incident or escalated. The transformation is done simple, by pressing one button, the event is then converted to an incident.

5.3.3 Developed EM process

All of the chapters so far lead to the developed EM process, as shown in the chapter *5.1 Overview of proposal building* in figure 19. The developed EM process can be built up capitalizing the areas described in the chapters *5.2 Findings of Data* 2 and *5.3 Proposal of Developed Event Management.* These areas include requirements from the case company, the integration process, the event layer including templates, data, event correlation, and finally SD activities. By utilizing the theory and best practices found in the Literature review and understanding the current solution and process from the CSA, a comprehensive understanding is formed to build the EM process, which will be displayed and described in this chapter.

The newly developed EM process, which can be seen in figure 22 below, is divided into 4 areas. These areas represent the different systems or where the process operates, which are the customer environment, monitoring system, integration layer and the SM-system. The main difference, compared to the EM process, which was described in the CSA, in figure 3, is that a new integration layer is brought in, as well as the process in the SM-system. This new integration layer, which will be located in the case company's integration platform, allows the usage of EAI and Event Correlation, in conjunction with the EM process.

Figure 22 below is marked with numbers from 1-13, which represents the different stages of the process. These stages will be described in detail below:

1. The first stage of the EM process is the Server, which includes applications that customers use. This is called the customer environment. The customer server can include multiple applications that are linked to each other, or completely separate.

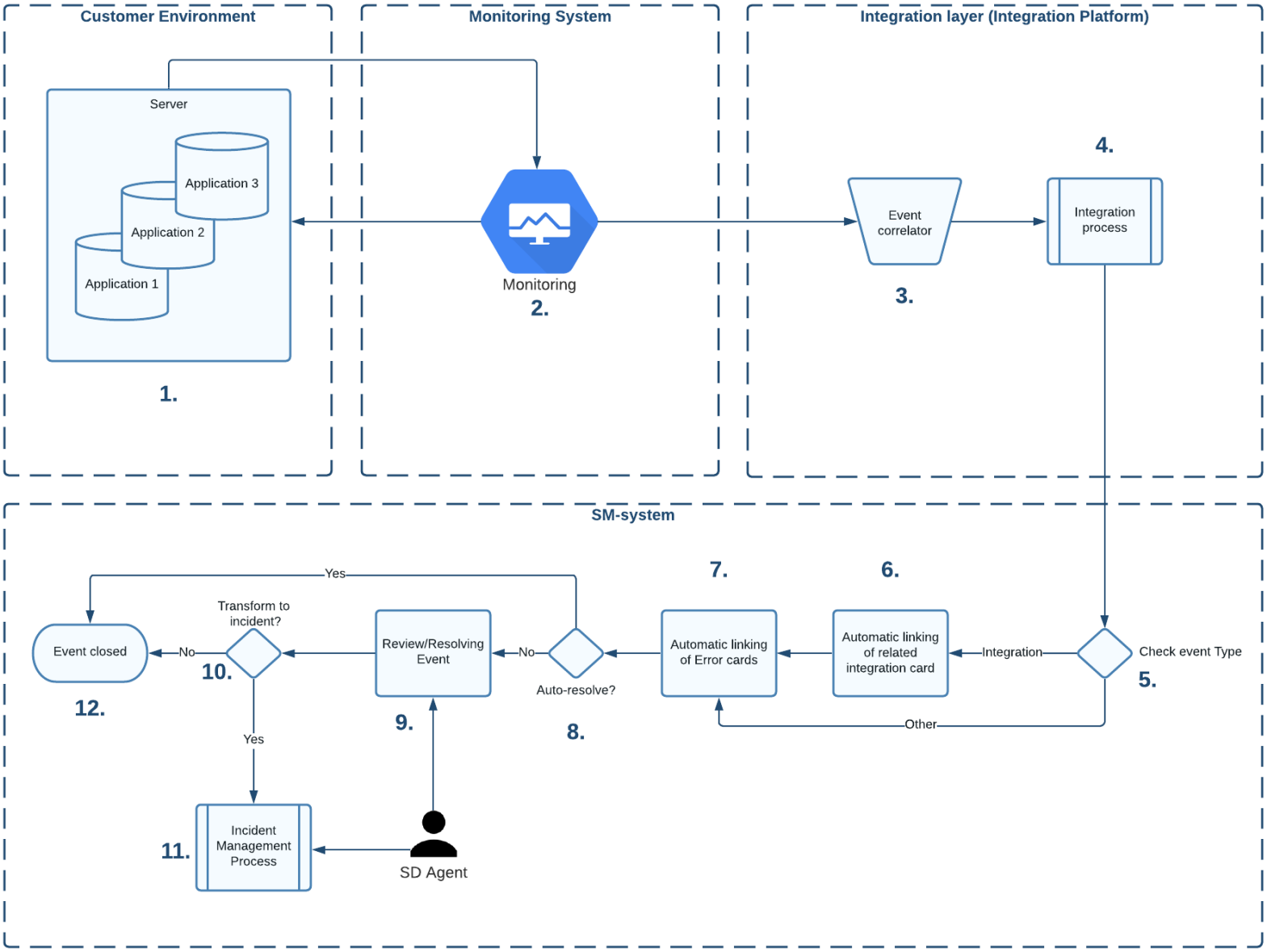

Figure 22: Developed EM process.

2. In the second area, the monitoring system monitors the customer environment, including both the server, and any applications that are running on it. The monitoring runs all the time and detects any changes or errors from the environment. Thresholds and parameters are set to the monitoring system based on the case company's needs. When the monitoring system detects an anomality, it creates an event of it and feeds the information forward.

3. When one or multiple events are created in the monitoring system, they are sent to the Event Correlation system in the Integration Layer. The event correlation system then finds the specific event that caused multiple events to be generated and sends it forward. If only one event is created, the correlator passes it onwards.

4. The Event with data is passed through to the new integration process, described in *5.2.2 Integration process.* The integration process is constantly running, catching any events that are coming from the correlator. The event then proceeds to go through the defined process, which utilizes Web API to transfer the event to the SM-system

5. Next, all of the following stages of the process happens in the SM-system. Firstly, the event is checked if it is an Integration related event from the customer environment, or any other type of an event. If the event is related to an integration error detected by the monitoring system, it continues to stage 6. If it's any other type of an event, it goes directly to stage 7.

6. When the incoming event is detected to be an integration related event, automatic linking of related integration data card from the Integration-template, that is described in *5.3.1 Event layer.* Automatic linking is done by a script, by comparing the data from the event to the integration data cards, which are prefilled. If no match is found, no linkage is generated. Relevant information is copied to the event from the integration data card, which helps the SD agent to process the event.

7. After the integration data card is linked, automatic linking of Error data cards is applied, this works similarly as the linking of the integration data cards – by comparing the event data to the Error card data from the event template. Also, if the event type is other than integration in step 5, the linking is done here to those events as well.

8. When the possible integration- and error data cards are linked, the value for auto-resolving the event is checked from the error data card. If this value is set to “Yes” in the error data card, and a linkage is created, the system resolves the event automatically and updates the status and resolution fields. If no auto-resolve selection is found, the event is displayed to the service desk.

9. After the auto-resolution check, the event is brought to the SD agents, in a view. These views can be customized and should facilitate the SD agents work. The first view, where the newly generated event is shown, could show all events with status 01. New, for example. This is when the event is shown for the first time to the SD agent. The agent’s responsibility is to quickly review the new event. If the integration- or error data cards have been linked, actions to resolve the event are displayed on the event data card. The service desk Agent then either resolves the event to their best ability or escalates the event.

10. If the event needs escalation, it can be wither passed to a second level SD agent, a specialist, or consultant that can handle the event, or it can be transformed to an incident. Automatic transformation to an incident could also be implemented based on the priority of the event.

11. If the event is transformed to an incident, the event will be handled according to the case company’s incident management process.

12. If the event is not transformed to an incident the event is set as closed and will be archived.

5.4 Summary of proposal

The current systems and processes that the case company is using, will have adequate capabilities to support this new EM process and technical solution that is proposed in this section. The process is a developed version of the EM process displayed in the CSA, utilizing new methods and technologies and subprocesses, to achieve a more comprehensive solution that solves the issues with the current solution.

Considering the requirements from the case company, designing the new integration process, designing, and fitting in the new templates and automations to the SM-system by following the workflow of the developed EM process, the initial version of a developed EM process is complete.

6 Validation of the proposal

This section discusses the validation of the proposal presented in the last section as well as displaying the benefits of the proposal by inspecting the weaknesses from Data 1. Data 3 is briefly discussed to present the results of the workshop held for validating the proposal. Lastly next steps are given on how to move forward with the implementation of the proposed EM process. Since the tight schedule of this study, implementing and testing the proposal was not possible in the timeframe. However, the implementation will be carried out in the following months when resourcing is planned for the development work by the case company.

6.1 Pain points, proposal, and benefits

To validate the proposal presented in the last section, the weaknesses and requirements found in the CSA and Data 2 respectively, are recapped and analysed by taking a look on how those issues were addressed in the proposal. Referring to table 4 from section *3.3 Strengths and weaknesses of current EM proposal,* following weaknesses were found:

- Too many unnecessary incidents are created
- Lack of data for event related incidents
- Automatic linking missing
- Biased reporting capabilities
- Lack of efficiency
- Lack of forecasting possibilities
- Too much manual work

These weaknesses can be displayed in pain point, proposal, and benefits matrix seen in figure 23 below.

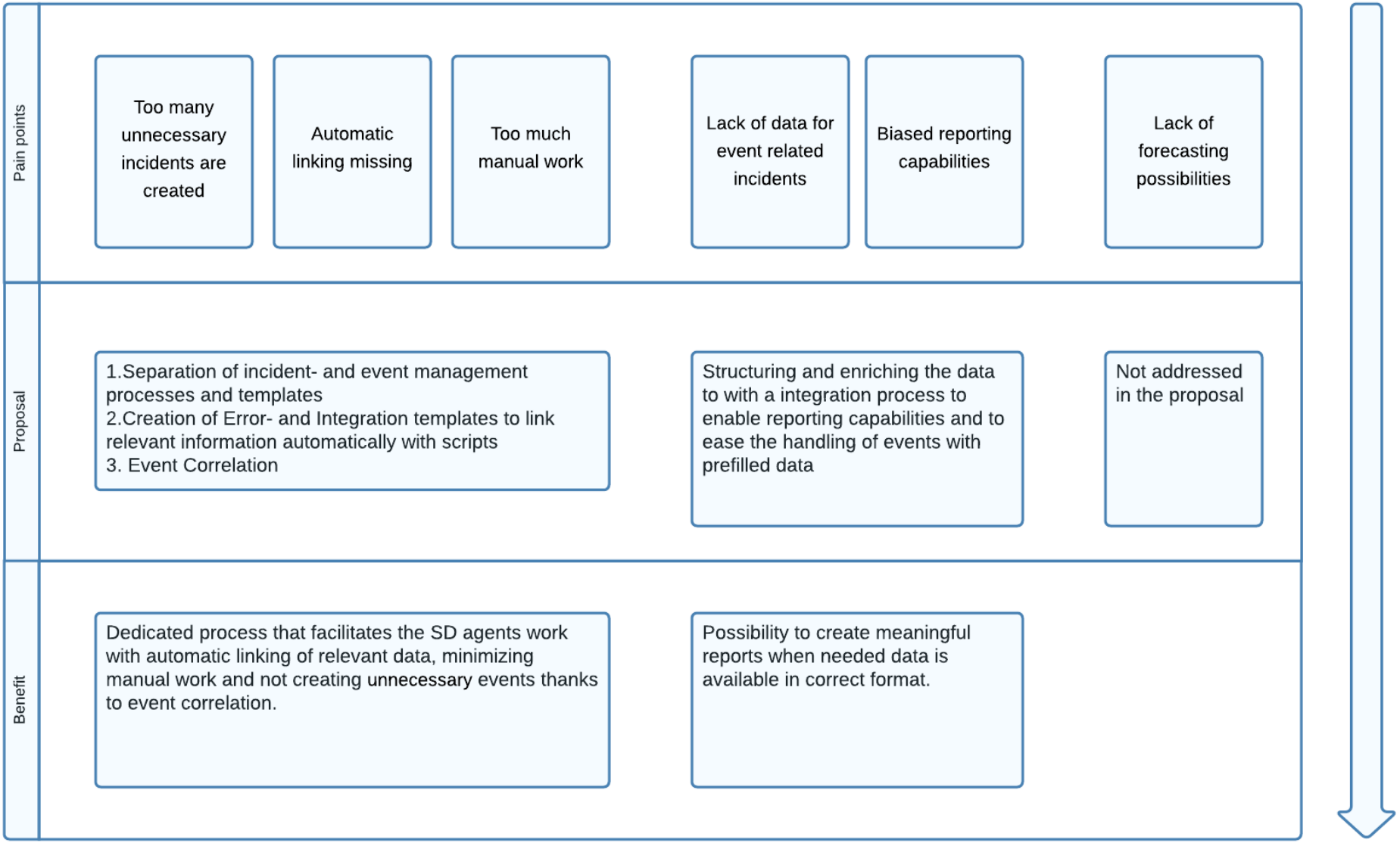

Figure 23: Pain points, proposal, and benefits

All the weaknesses except lack of forecasting possibilities were considered in the proposal. This is due to forecasting capabilities should be created in a monitoring tool, which is out of scope of this study. However, this is an excellent filed to study separately in the future.

Requirements for the developed EM process from the Data 2 are displayed in table 11 below:

Table 11: Requirements and how they are addressed in proposal.

#	Requirement	How requirement was addressed in proposal
1	The EM process should be separated from the current IM process.	By creating separate process and templates.
2	The solution should not be as demanding and should facilitate the SD agents work.	Automating steps in the process, such as linking and priority information.
3	The developed EM process should enable accurate reporting capabilities.	By enriching the data and improving the dataflow from the monitoring system to the SM-system.
4	An event correlation logic should be considered in the process.	Designing the process to utilize correlation. No technical solution for correlation was addressed.

As seen in table 11 above, all four main requirements were addressed in the proposal. The technical solution for Event correlation was not addressed, since it is out of scope for this study, but it was considered in the process.

6.2 Inspection of Data 3

A workshop was held with a Senior Architect and PM, one of the key personnel in the case company related to this study. The workshop discussed the proposal overall, how it should be implemented and if the proposal is valid for implementation to the test environment next.

Proposal was considered valid in the workshop, with a few notes. The event correlation should be developed by the case company itself and implemented in the integration platform. The second observation was that the proposal should be implemented and tested but should be working in conjunction with the old model, but only as long as the proposed EM process is set up and running for all customers. There should not be any gaps in EM any time. This has to be considered in the implementation of the proposed EM process.

More thorough validation of test implementation of the developed EM process should and will be done in the next stage after this study, which is implementation.

6.3 Next steps

The proposed EM solution should be carefully implemented first to the case company’s own test-environment, where it is thoroughly tested with different scenarios and situations. Any needed changes should be implemented that are brought up from the testing, and further developed in an iterative process. When testing is done, all changes should be moved to the production environment, where everything should also be tested and validated. Any changes to the daily workflows of personnel should be considered, trained, and informed for a smooth go-live.

7 Thesis overview and conclusions

This last section of this study summarizes and discusses the conclusions. First, an executive summary is presented, giving an overview of the study in general, and the proposal. After the executive summary, evaluation of the study is discussed, including topics such as relevance, validity and reliability, and lastly final words are presented.

7.1 Executive summary

This study focused on building an Event Management model that can be implemented easily, considering the requirements for the proposal. The objective of this study was to develop the current event management process, to match industry standards, best practices, and internal as well as external requirements.

The study was conducted in seven stages, starting with an introduction followed by discussing the methods and materials used in this study. A research design was presented, displaying the structure of the research, what data is used, and what is the outcome. Next a current state analysis was conducted, to gain an understanding of the current methods and processes. Here, interviews with key personnel were conducted, and technical solutions inspected. Also, discussions with employees were held to gain an insight of the different aspects of event management. Challenges were found in the CSA, including missing reporting capabilities, lack of efficiency in handling event related incidents due to manual work and unstructured data. To create a well-considered proposal, theory about relevant areas was studied, such as ITIL, Event and Incident Management Best Practices, and Enterprise Application Integration. By having a clear understanding of the relevant theory, a proposal of a Developed EM was built, following a validation of the proposal.

The proposal is focused on to ease the workload of SD agents, dealing with masses of events created from various environments. The proposed EM process, utilizes automations, integrations, data enriching and structuring, event

correlation, to create a seamless workflow for handling with events. The main findings were to separate the incident and event management processes and to utilize integrations to enhance data flow from monitored environments, to the service desk. Creating automatic linking of relevant information for the events, manual work of the SD agents minimizes, and reporting capabilities are made better with more accurate information.

The next step is to implement the proposed EM process, following this study. The first step is to create new templates. After this the new integration process and event correlation should be built to support the new templates and process. Lastly, the new process with new workflows and automations should be implemented.

7.2 Evaluation of the thesis

In this section, the thesis is self-evaluated, considering aspects such as what could have been done better, is the proposal relevant, how valid it is and is there any shortcomings.

7.2.1 Relevance

The results presented based on the proposal, compared to the weaknesses and requirements from the case company, seem beneficial and promising. However, the proposal has not been piloted or tested yet in any environment, so it is difficult to tell, how much more efficient the developed process and workflows are for the case company.

The developed process and solution are based on the original solution, even though heavily developed. This means that the processes, use-cases, needs, and systems can be very different in other companies, which can make this exact solution hard to copy exactly. However, the process and methods are based on best practices, and the proposal works as a good example on how an event management solution should look like anyways.

7.2.2 Validity and Reliability

The proposal was considered from multiple aspects and angles, studying relevant best practices and technologies to create a reliable and valid proposal. By interviewing key employees of the case company, relevant information and wanted features and requirements were brought up to be considered in the proposal.

Since the rather short time span of the project, the whole study was conducted in under seven weeks, meaning not everything could be considered in this study. This can also mean that something is missing, or some proposed solutions do not work as intended. However, these things will be noticed and acted on in the implementation and testing stage, after this study is completed.

The proposal was focused mostly on integration related events, since seen from the CSA, these were the biggest pain point, creating loads of manual work for the SD agents. However, all other types of events should also be considered in the proposal. The proposal was designed to support all types of events, but other cloud-related events is something that might need some further development and planning, to make forecasting example possible, which was found as a weakness, but not addressed in the proposal.

7.2.3 Final Words

Looking back at the thesis, I am overall happy of the end results, and exited to implement this proposed EM process for the case company when the time comes. The proposal brings a lot of benefits to the SD agents and their daily work, making their life a little bit easier. The proposal also benefits the company in whole, even though EM is a small part of the whole operations, it is important to have it properly implemented to work as efficiently as possible in all aspects of the business.

Personally, I have learned a lot from this thesis. Areas like integrations, were mostly completely new for me and this gives me new competences to advance in my career in the IT industry. I enjoy always learning new skills and practices that can help make any business more efficient and more pleasant to work in.

References

2011. *ITIL® Service Operation*. Norwich: TSO, the Stationery Office.

2017. *The Open Group IT4IT™ Reference Architecture, Version 2.1*. The Open Group.

2020. *ITIL®4 EVERYTHING YOU HAVE TO LEARN*. 1st ed. SERVIEW GmbH.

Aashik, A., 2020. *Web APIs*. [online] Medium. Available at: <https://aashikahamed.medium.com/web-apis-216b3a5f36a5> [Accessed 19 November 2021].

Axelos.com. 2021. [online] Available at: <https://www.axelos.com/certifications/itil-service-management/what-is-itil> [Accessed 19 November 2021].

Bayes, S., 2021. *A View from the Frontline 2019*. [online] Servicedeskinstitute.com. Available at: <https://www.servicedeskinstitute.com/wp-content/uploads/2019/08/A-View-From-The-Frontline-2019.pdf> [Accessed 19 November 2021].

Brajesh, D., 2017. *API Management*. Bangalore, Karnataka: Apress.

Dictionary.cambridge.org. 2021. *best practice*. [online] Available at: <https://dictionary.cambridge.org/dictionary/english/best-practice> [Accessed 19 November 2021].

Gruschke, B., 1998. *INTEGRATED EVENT MANAGEMENT: EVENT CORRELATION USING DEPENDENCY GRAPHS*. Munich: Department of Computer Science, University of Munich.

Hohpe, G. and Woolf, B., 2020. *Home - Enterprise Integration Patterns*. [online] Enterpriseintegrationpatterns.com. Available at: <https://www.enterpriseintegrationpatterns.com/> [Accessed 19 November 2021].

LIMITED, A., 2019. *ITIL® Foundation, edition ITIL 4*. London: The Stationery Office Ltd.

Linthicum, D., 2020. *Enterprise Application Integration*. [online] Metcat-Finna. Available at: <https://metropolia.finna.fi/Record/3amk.16613> [Accessed 19 November 2021].

Watts, S., 2017. *4 P's of ITIL Service Strategy*. [online] BMC Blogs. Available at: <https://www.bmc.com/blogs/friday-focus-on-itil-part-four-4-ps-of-strategy> [Accessed 19 November 2021].

Interview with Employee of case company, Service Desk Team Lead

Interview with Employee of case company, Cloud Ops Team Lead & Product manager

Interview with Service Desk Team Lead for CSA

What is your role in the case company?

Running and managing service desk team, day to day operations.

How is your role related to Event Management?

My team handles all incoming events, either by handling the tickets or escalating them for further investigation. For example, handling integration errors as incident tickets that are generated from customer environments.

What is Service Desks relation to Event Management?

Analyzing and handling event related tickets. Escalating events that require more investigation to consultants/other specialists. Handling all event related incidents and customer communication.

Could you describe the current Event Management process?

Customer environments are monitored with case company's used monitoring tools which detects different events in the monitored systems, which are then sent by the monitoring system via email to the case company's own ticketing tool, without categorization, customer information and instructions of how to handle (for example KB article). Every event generates a new ticket which are directed into first level queue, with all other incidents and tickets.

Service desk agent takes the generated event ticket into handling, adds customer information, checks, and links possible duplicates, checks if there is guidance (e.g., Integration error cards, which contains error handling instructions for integration related errors). If instructions found, ticket will be handled accordingly. If instructions are not found and the agent does not know how to handle, the agent escalates to specialist/consultant for further investigation.

If new solution for handling event is found, agents update the integration error data card to contain instructions.

If needed, the agent contacts the customer for more information and updates the ticket.

When the ticket is escalated to a specialist/consultant, ticket ownership is still with SD, and agents closes the ticket when resolved. If there is a change request, the SD agent will direct it to the case company's consultant, whom then has the ownership of the ticket.

no feedback from events is generated (only for incident which need customer contact from SD specifically.)

What's the shortcomings of the current EM process?

A lot of tickets are generated, since all monitored events are generated as incidents, which causes a lot of manual work which is time consuming.

Generated events do not include customer information, which is an additional step for the SD agent to add. Agent needs to find the related customer from ticket subject/description.

Automatic linking missing to relevant KB articles or integration error cards.

Also, a lack of reporting is relevant since all tickets does not contain needed data and are possibly closed by mass editing.

What's the strengths of the current EM process?

Possibility to react to events proactively (before customer notices themselves) that are generated to incidents.

How much workload does event management in general generate for SD / for one SD Agent? (By estimate)

0,5-1h per SD Agent per day. 5 SD agents.

How many tickets are generated from the monitoring for EM?

Service desk handles weekly around 50-100 event related incidents.

By estimate 10-20 events that needs consultant to check why event error was generated.

What does the current event data look like?

Integration events:

- What integration / process is in question

- Link to integration platform
- Generated error message

How do you see that the EM process should be?

- Only one ticket should be created per event, no duplicates of one event should be created as tickets.
- Customer information should be prefilled to the ticket
- The generated incident should have categorization prefilled, so the SD agent can see does the event need any actions or is it just information
- Linkage to documentation should be created
- Ideally, incidents would only be generated if actions are needed from agent / customer communication is required

Interview with Cloud Ops Team Lead & Product Manager for CSA

What is your role in the case company?

I am responsible for all Cloud-related operations, platform & capacity management, new customer cloud onboarding, quarterly released deployment and second level support for cloud related incidents.

How is your role related to Event Management?

To manage daily operations that are related to cloud operation related events.

What is Cloud ops relation to Event Management?

Cloud operations generates, utilizes, and manages events.

Could you describe the current Event Management process? (In terms of Cloud Ops)

Monitoring data is collected and utilized to applied extent for event management at this moment. The utilization is not on the wanted level to unlock full potential of proper event management.

There are many different processes and operations that uses event management, but they are on different maturity levels.

Cloud server storage monitoring and events:

- When a specific threshold is crossed, in this case GB, the monitoring tool that monitors cloud environments creates an event, which then sends an email to case company's service desk, which creates an incident. If the event is critical, the monitoring software sends an alert to a third party automated service, which will call trough the cloud ops team in defined escalation policy to the on-call personnel.

Cloud environment certificates

- When a specific threshold is crossed, in this case a date, the monitoring tool that monitors cloud environments creates an event, which then sends an email to case company's service desk, which creates an incident. Also, a notification is generated of the event in a communication tool that the case company uses in a specific channel for Cloud ops.

All environments update related events

- A lot of events are generated when environment updates are performed, but these are not utilized.
- We use 2 different monitoring platforms; one monitors for events and the second visualizes logs.
- Events are generated from servers, server platforms and applications. Also, from logs.

Capacity management related events

- For capacity management related operations, a lot of events are also generated but not utilized.
- These are for example: Mass migrations in blade servers
- These are not utilized almost at all, sometimes updates are visualized, which can be used to create hotfixes.
- By estimate, 90% of the potential is not utilized.

What's the shortcomings of the current EM process?

Does not allow us to operate as efficiently as we could since the lack of a standardized and centralized EM process. We could increase the performance of our operations, for example, by using events to forecast possible issues when upgrading customer test environments. These issues could be reported to our QA team, which then could fix the issues before production upgrades.

What's the strengths of the current EM process?

It helps a lot for example in certificate expiring, to get the information beforehand, and to renew them so customers are not affected.

Being able to proactively handle different events before they are generated into issues.

How much workload does event management in general generate for Cloud ops?

The events that are escalated from Service Desk to Cloud Ops generate by estimate from the whole team 2-4 hours weekly.

How many tickets are generated from the monitoring for EM?

Certificate related tickets: by estimate 0-10 weekly.
Capacity management related tickets: by estimate 0-5 weekly.

What does the current event data look like?

Certificate:

- Certificate name
- Status information (e.g., Warning)
- Date and time

Capacity management:

- Customer information
- Threshold information (e.g., 10%)

How do you see that the EM process should be?

That the most important events would be utilized collectively. Following trends and doing conclusions to better and make different activities and operations more efficient.

Most important:

- Continuous monitoring
- Update related monitoring
- Capacity management related monitoring

Further ideas to EM?

To use artificial intelligence to find and analyze patterns and to find anomality's in events, what would then be brought up.