



Bachelor's Thesis

# Licensing Activation Solution

By

Dervis Avdic

Department of Electrical and Information Technology  
Faculty of Engineering, LTH, Lund University  
SE-221 00 Lund, Sweden

# **Abstract**

This thesis is developed with collaboration and on behalf of Tetra Pak AB. The assignment was to create an internal system to manage site information and a license server to validate licenses. These systems are uniquely tailored for Tetra Pak's requirements and the software they want to license. The internal system manages data that the sales division enters and creates text files from that site characteristic. It can also create keys based on hardware information collected from customer sites together with its associated site information. The license server is installed at customer site and handles license activation. It gathers information about hardware and activates the licenses based on the validation. Then the license server supervises the software if there is any deviation from the hardware information.

Keywords: Licenses, IT-security, usability, IT-systems and integration.

# Sammanfattning

Detta examensarbete är genomfört i samarbete med och på uppdrag av Tetra Pak AB. Syftet var att utveckla ett internt system för att hantera platsinformation och en licensserver för att validera licenser. Systemet är unikt skräddarsytt för Tetra Paks kravspecifikation och den mjukvara som ska licensieras. Det interna systemet hanterar data från försäljning och skapar textfiler baserat på informationen. Det kan även skapa nycklar baserat på hårdvaruinformation från kundplatser tillsammans med tillhörande platsinformation. Licensservern är installerad ute hos kunden och hanterar licensaktivering. Den ackumulerar hårdvaruinformation och aktiverar licensen beroende på valideringen. Licensservern övervakar därefter mjukvaran för att uppfatta avvikelser hos hårdvaruinformation.

Nyckelord: Licenser, IT-säkerhet, användarvänlighet, IT-system och integration.

# Acknowledgments

This bachelor's thesis would not exist without the support and guidance of;

Peter Bjernetun, Product Owner – Tetra Pak AB.

I want to thank you for letting me have this opportunity and its continued collaboration.

Peter has been my supervisor for this thesis and is also the functional product owner. He has helped me through the project, answering all my questions and has successfully managed to guide me and my work. He has also created a perception about licensing for me and for that I am very grateful.

Christoffer Jerrebo, IT-Consultant – Jayway AB.

Christoffer has been very helpful throughout the project and has helped me a lot in the begging and with certain difficulties with for example creating installations packages. To that I am very grateful.

I am also grateful for all the help I have gotten from the team working with the MES platform.

Kristina Åstrand, Manager Project Management – Tetra Pak AB.

I want to thank you for letting me have this opportunity and continued collaboration.

Christin Lindholm, Associate professor and Education Program Leader for the Bachelor program on Computer Science and Electrical Engineering with Automation – Lund University.

I want to thank you for your guidance throughout my thesis and report and answering all my question regarding my thesis.

Christian Nyberg, Associate Professor at the Department of Communication Systems – Lund University.

I want to thank you for answering my question regarding the thesis.

# Table of Contents

<b>Abstract</b> .....	<b>2</b>
<b>Sammanfattning</b> .....	<b>3</b>
<b>Acknowledgments</b> .....	<b>3</b>
<b>Table of Contents</b> .....	<b>6</b>
<b>1 Introduction</b> .....	<b>8</b>
<b>1.1 Background</b> .....	<b>8</b>
<b>1.2 Tetra Pak AB History</b> .....	<b>9</b>
<b>1.3 Purpose and Goal</b> .....	<b>10</b>
1.3.1 Purpose.....	10
1.3.2 Goals.....	11
<b>1.4 Problem definition</b> .....	<b>12</b>
<b>1.5 Limitations</b> .....	<b>12</b>
<b>2 Technical Background</b> .....	<b>16</b>
<b>2.1 Software licensing</b> .....	<b>16</b>
2.1.1 Hardware-locked licensing.....	17
<b>2.2 Encryption</b> .....	<b>18</b>
2.2.1 Symmetric encryption.....	18
2.2.2 Asymmetric encryption.....	19
2.2.3 Digital certificates.....	20
<b>2.3 Licensing</b> .....	<b>21</b>
2.3.1 License server.....	21
<b>2.4 ASP.NET MVC</b> .....	<b>23</b>
<b>3 Methodology &amp; Analysis</b> .....	<b>26</b>
<b>3.1 Working methods</b> .....	<b>26</b>
3.1.1 Agile & Scrum.....	28
<b>3.2 Problem solution</b> .....	<b>28</b>
3.2.1 Problem solution – Back-Office.....	29
3.2.2 Problem solution – License server.....	31
<b>3.3 Source criticism</b> .....	<b>34</b>
<b>4 Results</b> .....	<b>38</b>
<b>4.1 Work result</b> .....	<b>38</b>
4.1.1 Back-Office Part 1 – Register site.....	41
4.1.2 Back-Office Part 2 – View sites.....	41
4.1.3 Back-Office Part 3 – Details/Export.....	42
4.1.4 License Server Part 1 – Upload.....	44

4.1.5	License Server Part 2 – Export .....	45
4.1.6	Back-Office Part 4 – Upload SITEKEY .....	46
4.1.7	Back-Office Part 5 – Export LIC .....	47
4.1.8	License Server Part 3 – Activate .....	48
4.1.9	License Component.....	48
<b>5</b>	<b>Conclusions .....</b>	<b>50</b>
5.1	Result of problems .....	50
5.2	Conclusion.....	52
5.3	Product utility.....	53
<b>6</b>	<b>Future Work.....</b>	<b>56</b>
6.1	Future development.....	56
<b>7</b>	<b>References .....</b>	<b>58</b>
7.1	Picture References.....	60
7.2	Appendix References .....	61
	<b>List of Acronyms.....</b>	<b>64</b>
<b>1.</b>	<b>Workflow .....</b>	<b>66</b>
1.	Workflow-picture of a new installation .....	66
2.	Workflow-picture of extension to existing installation .....	67
3.	Workflow-picture of replacing existing installation .....	67
<b>2</b>	<b>Working methods .....</b>	<b>68</b>

# CHAPTER 1

## 1 Introduction

---

*This chapter gives a brief introduction to the thesis, its purpose and problem definition. It also presents relevant background information about software licensing and why it's commonly used. Finally, limitations of the thesis work and what could be implemented in future versions are discussed. I would also recommend taking a look at the workflow of this licensing solution for better background understanding, see Appendix 1 (picture 1, 2 and 3).*

### 1.1 Background

This thesis is developed with collaboration and on behalf of Tetra Pak AB. Tetra Pak doesn't have licensing of its software today. This is partly because Tetra Pak trusts their customers and that they won't use the software in any way other than what is stated in the contract for usage of the software.

Licensing Activation Solution is a licensing solution that has been developed as the assignment of this thesis. The licensing solution consists of one back-office system, one license server

and one license component. The back-office system will be able to create license keys based on site characteristics and hardware information. The license server will control license activation and will communicate with the license component. The license component will be implemented into different software's designed by Tetra Pak. One example is the new MES Platform which will be licensed by this solution. In addition it shall it be possible to keep track of where the licensed software is installed and for example be able to view licenses used in different countries for different customers.

Licensing solutions are already implemented in other companies where their purpose is to create a surveying control. Therefore the unique ability Licensing Activation Solution possessing is its way of sending and receiving files and keys. The system is also implemented with a unique feature designed to generate site requests based on the customer sites characteristic. Naturally the unique algorithm for generating the product keys is a step that distinguishes this system from other systems that generate keys.

## **1.2 Tetra Pak AB History**

Tetra Pak AB was founded in 1951 in Lund, Sweden by Ruben Rausing. Along with it, the first packaging system for dairy products was designed and manufactured by Tetra Pak on the 18 of May that same year. In November 1952 the year after the first product was created, a one deciliter cream-package was introduced (Tetra Pak, 2016).

In 1956 the company moved to their own factory at Råbyholm in Lund. Tetra Pak then began to create some unique packaging products for milk and other dairy products like Tetra Classic, Tetra Brik and Tetra Prisma. This got attention from outside of Sweden and this was the first time the company reached off to the world. Råbyholm is still one of their quarters today and was their headquarters until late 20th century before they moved to Switzerland (Tetra Pak, 2016).

Tetra Pak is still today the world leader in food processing and packaging solutions for food. The company has more than 23,000 employees in over 85 different countries and it's increasing for every year. When the company was founded by Ruben Rausing in 1951 they specialized in creating and designing packaging systems. This is still a part of their income source. Today they focus in creating complete solutions for processing, packaging and distributing food products such as dairy products, juices, ice cream and cheese. (Tetra Pak, 2016)

## **1.3 Purpose and Goal**

### **1.3.1 Purpose**

The purpose of this thesis is to collect information and manage the data for extracting files and keys. Then it should be possible to activate the license at the license server. The license server then controls the activation and monitors deviations of the hardware at the sites of Tetra Pak's customers. This validation is based on the communication from the license component installed in the software. The expected results of this product are

based on three parts, as mentioned earlier in 1.1 Background. The back-office system that is established at Tetra Pak site, where there is the ability to enter site characteristics, such as functionality, equipment and connectivity. This system ultimately generates a valid license after the file has been sent to the license server, where it gathered information about the machine that is going to have the software installed. The license server is the second part of the Licensing Activation Solution that is going to be developed. It gathers information about machine hardware and is the final step for comparing the license keys. The third and final step of the product is a license component that is going to be installed at all the software that is going to take advantage of the licensing solution. This communicates with the license server and sends information about the machine hardware.

The license server receives the information and compares with the activated license.

See figure 1.

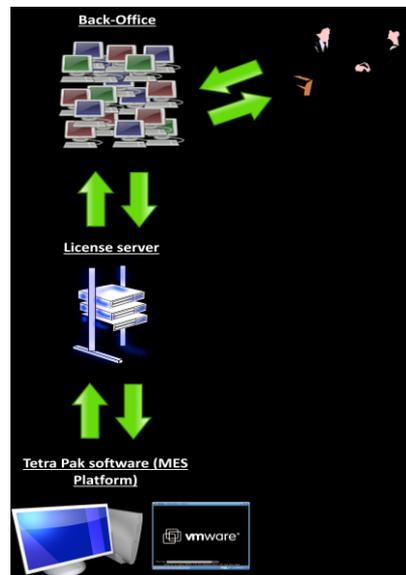


Figure 1 – Picture of the licensing solution

### 1.3.2 Goals

The goal of this thesis is therefore to develop all three components mentioned in the expectant result of the purpose above. The goal is also to fully make a functional licensing product (Licensing Activation Solution) in version 1.0. Along

with the development a database is designed and implemented. Here the goal is to make the design very flexible. This enables the database to add new data, and change present data without endangering the functionality. The goal is furthermore to create an installation package for the back-office system and the license server. This is so that the back-office system can be installed at different locations on Tetra Pak sites and the license server at customer sites.

## **1.4 Problem definition**

The problems/challenges of this thesis are the following questions below. Those are the core problems of this thesis.

- How to control which software the customer has installed?
- What to do to avoid that the software can be duplicated on a machine?
- How to develop an algorithm to identify a secure way for license key generation?
- How to implement flexibility to the system and the database for easily adding new application and functionalities?

## **1.5 Limitations**

A limitation of the Licensing Activation Solution system is the log-in system to the back-office system. A log-in system has been implemented that Microsoft provided in the start template, but nothing further. This is because of the time pressure and the priority of the thesis. There is going to be an implementation of a

new log-in system independent of the Microsoft template further on.

Working with the log-in system also requires work with Tetra Pak's Active Directory (AD), which is very time consuming and out of the time frame for this thesis. The Active Directory requires skills and experience because you are handling sensitive information about the employees. The part about the license cost in relative to the site characteristics isn't in the thesis area either. There is another division working with that, but our systems must be fully functionally before they can be fused. Therefore there is a certain importance of making the form user-friendly and easy to understand for users. This is because there is an economical aspect that is included for making price calculations on a site.

There are some limitations with the database design, because the Licensing Activation Solution is going to license Tetra Pak's new software, the MES Platform. The design must therefore be adapted to the new platform and the requirements that are included. The database design is for that reason created very flexible for adding data and fully functional for the MES Platform and other system that Tetra Pak may want to sublicense.

Licensing Activation Solution is developed on a web-platform which means that it will work on any operating system such as Windows, OS X and Linux. This is along with browsers Internet Explorer 7, 8, 9 and 10, Chrome, Firefox, Safari and Opera.

Today the servers are physical and there is no certain request for having the servers cloud based. That is of course a goal in the future for Tetra Pak, but for this thesis, there is no further involvement.



# CHAPTER 2

## 2 Technical Background

---

*This chapter gives an all through introduction to the technical background of the project. It gives a description about which technique there is present today in license servers and license keys.*

### 2.1 Software licensing

Software licensing is an agreement between the developer of the software and the end-user. The agreement specifies which terms of use the end-user are committed to. The agreement also specifies that all software must be legally licensed before it may be installed. The confirmation of product that is bought must be maintained by the individual client or corporation that using the software (University of North Carolina, 2016).

Software licensing is developed for economical viability. It's a system of controlling whom and how users are using the software. There is therefore a variety of different licensing

solutions that can be implemented on dissimilar systems. The most advantageous way of determine which licensing solution suits your system best, is by distinguish how valuable the product is. The Licensing Activation Solution system is based on hardware identifiers where the workstation license (licensed software) is limited to a single physical computer (University of North Carolina, 2016).

### **2.1.1 Hardware-locked licensing**

Hardware-locked licensing with online activation is the licensing mechanism this system is built and structured on. The main advantage of hardware-locked licensing is that the software developer (company) has full control over their product and how many times it is installed. It's very common method and most people that have absolute no technical skills have encountered whit it when installing for example Microsoft Office (wDay, 2016).

A hardware-locked licensing system works like this; The customers enter a serial number that they have received when purchasing a copy of the software. The serial number looks very similar to an example like this, *ABCD-EFGH-IJKL-MNOP*. The system can both create serial numbers and identify already created serial numbers. The generated serial number from the back-office system is based on the physical hardware information from the computer that the server is installed on at the customer site.. Then these two serial numbers are compared. If the serial numbers are equal the software will be activated and

the full functionality will be unlocked for this user (wDay, 2016).

## 2.2 Encryption

The Licensing Activation Solution system is utilizing encryption when sending the files between the back-office system and the license server for security causes. This system is based on an asymmetric encryption, which includes a key-pair. For better background understanding symmetric encryption is explained in (Microsoft, 2007).

### 2.2.1 Symmetric encryption

Symmetric encryption is one the oldest and most well-known encryption techniques. The symmetric encryption is based on a secret key, which can be almost anything, from single characters, to a digit or even a sentence. This secret key is imported when there is a message or a file to encrypt. Likewise if that identical message or file is going to be decrypted it has to be decrypted by the same secret key. Encryption can therefore have various methods of encrypting messages and files. Here is one example

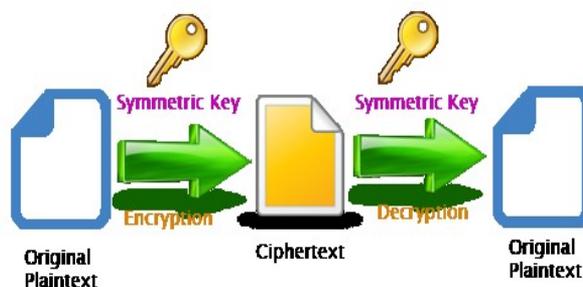


Figure 2 - Flowchart of symmetric encryption

demonstrated where the secret key is number 2. The secret key then shifts each letter two steps to the right (Microsoft, 2007):

You have a message – “Hello”.

Secret key – “2”.

Message after encryption – “Jgnnq”.

This is the very basic concept of encryption, see figure 2 for a visual overview. More advanced versions of these concepts can be implemented in systems with high security requirements (Microsoft, 2007).

## 2.2.2 Asymmetric encryption

Asymmetric encryption is based on a key-pair, containing a public key and a private key. The public key is open to anyone who might want to send a message. The purpose of the public key is therefore to encrypt the message that you want to send. The private key is kept secret. Only the person that wants to decrypt the message can have access to the private key. You can for that reason decrypt any message that has been encrypted with the public key in the same key-pair and encryption-algorithm. See figure 3 for a more visual overview (Microsoft, 2007).

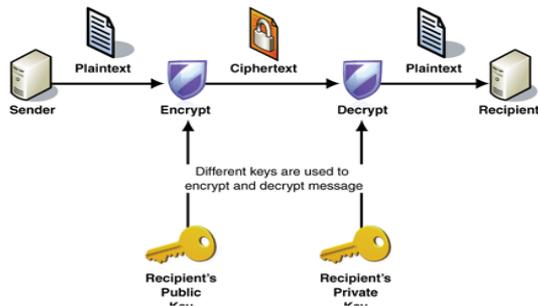


Figure 3 - Flowchart of asymmetric encryption

This means that you don't have to worry about passing the public key to the wrong end-user or server. Only possessing the public key can encrypt a message which won't do any harm to the system. This is even though it is equally important as the private key, because otherwise the encryption technique isn't going to proceed (Microsoft, 2007).

### **2.2.3 Digital certificates**

For the use of asymmetric encryption in larger systems that are more complex digital certificates are often mentioned. It's a method for other people to exchange keys. The most known technique is to use digital certificates, also known as just certificates. A certificate is package containing information that identifies a user or a server. It also contains information about the user. If it's an employment it keeps track of organization name, organizations that issued the certificate, the user's e-mail address, country and of course the users public key. This is because when you validate the certificate you must be sure that it comes from a reliable source (Microsoft, 2007).

When a communication is established between a server and a user they want a secure encrypted communication. The server or the user therefore sends a query over the network, which then sends back a copy of the certificate. From this certificate you can for example extract a public key. This is how you can exchange keys in a secure way if you want both end-users to have the keys for encryption and decryption (Microsoft, 2007).

## **2.3 Licensing**

Licensing is a wide term and it's used in various types of situations. Not only in software licensing, where a computer is connected to a server. Licensing is a practice where you legally lease a name, logo, design, characters or a combination of them. That is often called a product (system). Licensing is therefore a tool that's widely used by most of the business corporations for control and management of their software. If the business corporation has a very valuable product (software) there are very well-designed licensing programs. For example the owner can control the product and how it's displayed to the end-users by contractual agreements (Licensing EXPO, 2016).

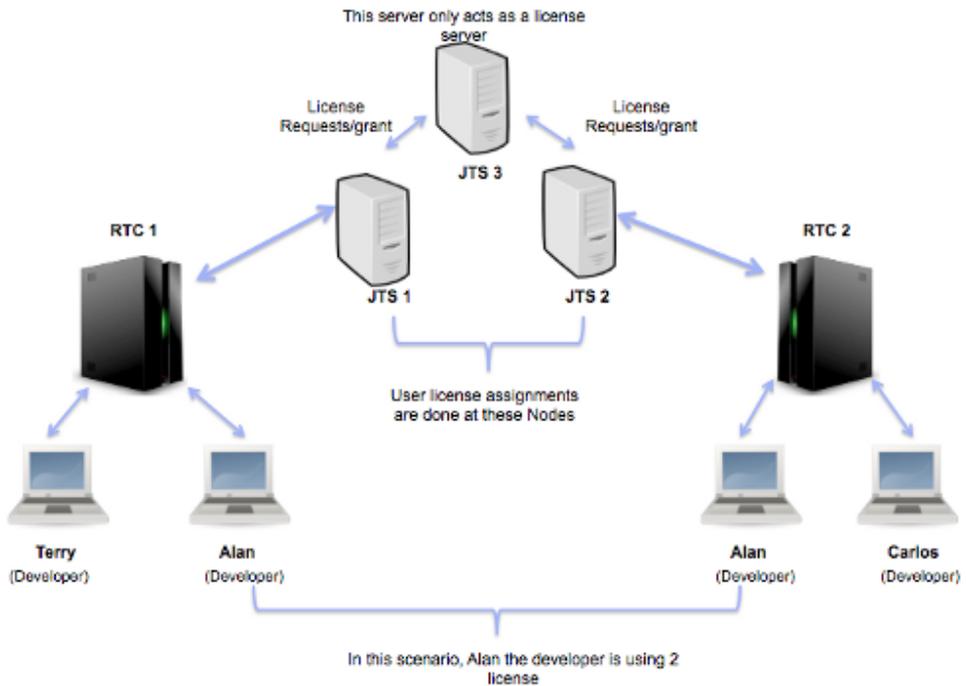
### **2.3.1 License server**

Software license server is computer software which provides access tokens and license keys to client computers. This is made in order to enable licensed software to run on them. Client computers send back requests to the license server and querying if their software is enabled. They then receive a response based if they are enabled or not (Sassafras, 2016).

In the beginning of 1989 a company named Sassafras Software Inc developed their first license management tool called KeyServer. This came to be one of the first license servers ever to be built. Since then other developing companies has begun to develop license servers (Microsoft, 2016).

The license server's assignment is to maintain control of how many copies of certain software is permitted by the license

agreement. This is done by installing a license server at the customer site. Every computer in the network, virtual or physical then communicates with the server for response in continuing use of software. See figure 4 below for a visual overview (Sassafras, 2016).



**Figure 4 - Picture of how license server works in an organization**

There are a lot of different licensing solution softwares available on the market today. Most of the software withholds a solution that can be implemented on your current system. Something that is very common among industry software is uniquely designed third party developed software. That is for example software that matches companies demands and functionality more accurately. This is often because of high security risk that comes along

when buying licensing software from an untrustworthy source (Sassafras, 2016).

## 2.4 ASP.NET MVC

Model View Controller, MVC is an architectural pattern that divides an application in three main components, model, view and controller. ASP.NET is a highly testable presentation framework that is integrated with existing ASP.NET features. The features are for example, default pages and log-in systems. The MVC model is the standard framework for many developments, but

some more traditional and outdated systems prefer the traditional ASP.NET. This

depends of course on how complex your

application is and the strategy for further development.

Preferably the MVC pattern is used in more multifaceted and larger system development like for example in industrial programming. This is because the separation using an advanced 3-layer system Model, View and Controller. This creates a separation where all the business logic is independent. See figure 5 for more visual overview and understanding the layers connection (Microsoft, 2016).

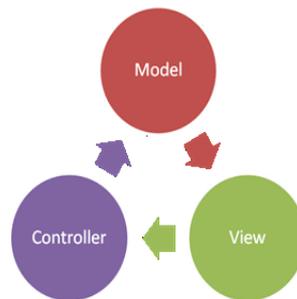


Figure 5 - Picture of the MVC pattern

In a MVC application the model objects are the parts that represent the logic. A very common area for models in

ASP.NET is that they work with the database. They tend to saving themselves as a model state, called model-view. In the Licensing Activation System, the models work with site characteristics. This means that sometimes a database object has to be updated if a change occurs. Then you retrieve information from a model, update it and put the model object back into the database server. Sometimes you have two models for the same database object (table). One is for the front-end (user view) and one for the back-end (database view). The front-end, where the user inputs information doesn't need to handle identification numbers (ID's). This is because that is not important for a user. Therefore there is a distinction between what the user visually sees and what is being stored in the database (Microsoft, 2016).

The view is the part that comes second in Model View Controller. It is the components that display the applications interface (UI). In the most common development the view is created from the model data. The view can for example have have pages with forms where the user inputs data. The view then renders all the buttons and the text-layers for the page and form (Microsoft, 2016).

The last component is the controller. It handles all the user interaction and works with storing and receiving data from the model. It also works with a view that renders all the information in a UI. The controller handles all the code that actually performs something to the system. The view just renders the information, so that the user can visually see it (Microsoft, 2016).

Overall the Model View Controller pattern facilitates the development and creation of an application. This is done by separating the structure into three components, input logic, business logic and UI logic with a connection between them. These three separations help a project manage complexity when building an application. This is because the focus is narrowed to one aspect of the implementation at a time, which could otherwise be very time consuming. For that reason this concept is used when developing and implementing Licensing Activation Solution. There are a lot of parts that need to be implemented and that could easily lead to confusion or to forget something in the code.

# CHAPTER 3

## 3 Methodology & Analysis

---

*This chapter is about the methodology used during the thesis work. It is also about which problems/challenges this thesis has encountered and how they have been solved.*

### 3.1 Working methods

The working methods in this thesis had to be very flexible. This is because of the amount of research that had to be performed and evaluated. The evaluations course of action was in the form of testing the different code examples. This led to more and less successful attempts, but eventually came to a reliable solution. All the testing and evaluation had to be completed before implementing it with the system. The lack of knowledge in the MVC pattern also led to a resolution where both learning the MVC pattern, reading about licensing and products keys were the best and only options for the time frame in the beginning.

The most challenging part in the beginning was to master the ASP.NET MVC pattern. This created a structure partly built on Scrum where the workflow had to be flexible. More information about Scrum is located in the next section, 3.1.1 Agile & Scrum. Briefly explained it's an iterative and incremental agile software development framework for product development.

There were certain limitations that this structure didn't include and that was that the method didn't have a backlog. A backlog with assignments prioritized relative to time and importance. Except for the backlog the structure followed the concept of that you could start programming from day one. If you then had issues there is the possibility to go back and make changes. The backlog was of course created in another way than what is explained here. See the summary of the structure for this thesis in appendix 2 page 64. The example is from the development of week 16. There every week was divided with assignments. In that case every week had one assignment inside the first system, back-office. This working method is partly built on the appendix 1, where one part is one assignment. Then one assignment from the back-office system can be divided into other smaller assignments.

The priority was divided into different steps. Step number 1 was the first part that was implemented and complete and then came the next step, number 2. This method is also using colors for a better visual picture. The orange color represents that the assignment has been initiated. The green means completed, but need adjustment and finally the red means that is hasn't been initiated.

### **3.1.1 Agile & Scrum**

Scrum is a branch of agile development. The agile software development is a set of principles for software development. The requirements and solutions evolve through collaboration between self-organizing cross-functional teams (CPRIME, 2016).

Scrum is a framework for agile development and the most common among product development projects. Scrum is mostly used in large product development divisions where they use iterative and incremental practices. This is to reach the goal as efficient and quickly as possible. The method has proven to increase productivity significantly compared to other product development techniques. The huge benefit with Scrum is the base concept of Agile. The rapid adjustments you can make on requirements to give a high priority to what is needed to be completed first (CPRIME, 2016).

### **3.2 Problem solution**

Problem solution has been an extensive part of this thesis. The first problem began when the structures were built in ASP.NETs MVC pattern. This was accomplished with guidance and help from the world wide web and tutorials.

The Licensing Activation Solution could be developed in numerous different ways. Tetra Pak is building their products in ASP.NETs MVC pattern and that was the most obvious choice of programming structure. There was a possibility to build the system in a less learning manner. However they would

eventually have to rebuild the whole system when it came to further development.

Detailed information about the back-office system and the license server is located in section 1.1 Background.

The problems that occurred in the first system, Back-Office:

- Finding the most secure and non breakable algorithm for generating the products keys based on the information from the site characteristics and the hardware information.
- Make the database design flexible for adding new functionality and data.

The problems that occurred in the second system, the license server:

- Encrypt the hardware information from license server that could enable a security risk.
- Retrieve hardware information from physical computers that has the license server installed.
- If the license server is installed on a virtual machine, the challenge is to retrieve the hardware information from the physical host machine.

### **3.2.1 Problem solution – Back-Office**

In the back-office system there a secure hash-algorithm had to be implemented. The algorithm basically retrieves the hardware information and hashes it. A hash-function is to map a key to a random number, hence that the data is distributed. See figure 6 for a visual overview. For this thesis a tailor-made hash function

was developed. It uses the hex symbols 1-9 and A-F. The hash-functions course of action is to obtain the hardware information, character by character and replacing each character with a number from 1-9 or a letter from

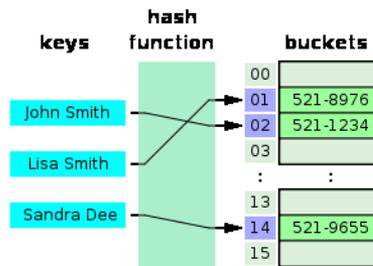


Figure 6 - Picture of hash-function

A-F. This will finally create a product key like for example *ABCD-EFGH-IJKL-MNOP-1234*. The algorithm has not been tested more detailed for security, but it has been evaluated and researched by system architect and was approved.

The product key that the system created is added to another key which summarizes the site characteristics for one site. It shall not be hashed because that information should be public for everyone. For the perspective of design is good practice to find a key solution for the site key. Therefore that key is developed on a unique algorithm that is under construction and not ready yet.

An additional problem or challenge for the Back-office system was to make the database design very flexible. This is for adding new functionality and data. The solution for this was to create look-up tables for the database. This is basically that you have all the raw data in one table and the user input data in one table like showed in the figure 7 below.

In the example of figure 6, here the look-up tables enable the possibility to create flexibility for adding new data. This is done inside the tables of figure 6 called “Table 1” and “Table 2”. Here it is allowed to add new data without disrupting the design and structure in the table “Products”. This example states what the problem or challenge were in the database design in this thesis. The license solution is created for the new software, MES. If it’s necessary to add new data functionality to the database it is possible with this database structure. The advantage of this database structure is that it’s implementable for any software.

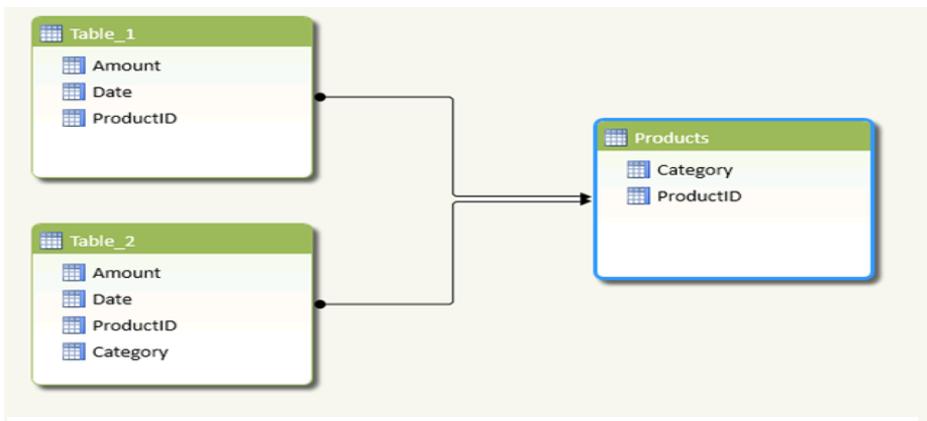


Figure 7 – Look-up table connected to two tables

### 3.2.2 Problem solution – License server

Encryption is a very important part of this project because it manages a great amount of sensitive information. The problem with symmetric encryption is exchanging the secret key over the internet or a large network. The communication could be intercepted by an attacker and the whole system would be a security risk. This is because the secret key is used for both encryption and decryption. The best solution for this thesis is

asymmetric encryption where there are two related keys, a key pair.

The thesis initialized with symmetric encryption in the system with the benefit of just handling one secret key. This is less difficult to implement, but the problem took place when transporting the file to another system. The file is logically going to be decrypted and the file size has to be the same as when encrypted in ASP.NET. If the file sizes aren't equal it will not decrypt the file because of security risks. The method for extracting the key from the file didn't proceed like the encryption-method allowed. This meant that the key couldn't be placed in the file. It would therefore in the end be forced to asymmetrically encrypt the file with associated key nevertheless. Therefore Licensing Activation Solution ended up with the asymmetric encryption where it had a public key. This key was made without restraint available to any user who might want to send a message and encrypt it. Along with the public key came a second private key that is kept secret where only the developer knows about its generating algorithm. One of the disadvantages with asymmetric encryption is that the encryption technique is slower than the symmetric technique. This can occur if the files are large, because of the processing power that is supplying the encryption and decryption.

The asymmetric encryption technique is slower because of the CPU speed. It has approximately a speed of 2000 megabyte per second for one core, on a standard good computer. The decryption of a 1024-bit message, e.g. cryptogram program can run a maximum of 4000 bytes/second, thus a throughput of 0.4

megabyte/second. That result in a speed which is 5000 times slower included the very colossal amount of processing power that is supplying this. Therefore it's a fact that the symmetric cryptography is faster and less power supplying than the asymmetric encryption. This is explained in World Scientific and Engineering by Shahzadi Farah, Younas Javed, Azra Shamim and Tabassam Nawaz, 2016 and then demonstrated in another example (Cryptography, Stack exchange, 2016).

The retrieval of hardware information became a key issue in the license solution for both the back-office system and the license server. This is because the information becomes a unique fingerprint for a physical computer. To retrieve the hardware information from a physical computer was less difficult. The challenge where located at retrieving physical hardware information from a host on a virtual machine. The problem with this is that the developers that create a virtual machine want to totally isolate the virtual machine entirely from retrieving any physical host machine data.

No useful information could be retrieved from the physical host computer through the virtual machine. Therefore it was necessary to find what physical host information that wasn't hidden by the virtual machine. The solution was as a result based on creating a program that retrieves all information about hardware from a virtual and physical computer. Then this information was compared and summarized to which information were equal. The information that wasn't virtualized was therefore the key information that linked a remote machine to its physical host

machine. This information is located in the CPU and it's not brought up publicly for Tetra Pak's security.

Although some information from the CPU was equal it wasn't enough for making a computer unique. The MAC-address for either the physical computer or a virtual computer was then extracted for higher identification of uniqueness. The reason for combining the MAC address with CPU information is; when the license server is installed on a virtual machine, the user could easily move the virtual machine to another computer. The license solution won't work because the hardware information is comparing the virtual machines information. Now you can have virtual machine on a physical host where the MAC-address is the one thing that identifies the license server, because the hardware information is the same. If the virtual machine moves to another physical computer in an attempt to break the licensing solution the CPU information will not be equal and this attempt would fail.

### **3.3 Source criticism**

The source criticism is very important in this thesis. For the history facts about Tetra Pak I have used Tetra Pak homepage when collecting information. This is very safe and reliable information. When writing about software licensing, hardware-locked licensing and licensing in general there is a certain impartial meaning of this. This is because one part is written by a company, one part by a university and one by a blog. Here the questions that should be produced, that's also written in Source Criticism on internet by Kristina Alexanderson is. Who is behind

the source? Is it a company? Is it an organization? Is it a private person? Is it someone you rely on? After these questions it is very important to ask why the source is created. Is it to inform? Is it to convince me into something? Is it to sell me something? The company produces information that wants me to buy their product. They're telling how complex and secure it is. I am just collecting information about their product and using it to my advantage in my thesis, so I am not affected. The university writes about software licensing and their purpose is to teach. The blogs primary purpose is to attract readers, but the underlying purpose can be to sell me something, to convince me into something or to inform me of something. The purpose for this thesis is to collect information about licensing and therefore no purchase will be complete. The will maybe convince their readers about their product, but for the readers who just collect information it doesn't matter.

Microsoft is a gigantic company and they have a wide library with information about IT and security. Their purpose is of course like all companies to sell their product and services. When for example writing about the Model View Controller pattern in ASP.NET. The text that Microsoft has written, points out how great and useful the tool is, from their perspective. In this case the MVC pattern is the best tool, because it's most compatible for this thesis. Along with the information Microsoft provided there is also information from Windows development center. Their purpose is basically the same as Microsoft, but the difference is that is depending on another source, Microsoft. When writing about symmetric and asymmetric encryption Licensing Activation Solution takes advantage of their structure

in cryptography security, because of their reputation and their experience in IT. This also applies when writing about license servers. The difference here is that the reader is restricted to only collect information and can't do anything concrete. This information is to inform and present fact to the reader (Kristina Alexanderson, 2016).

The source Sassafras software is a company, which answers the question. What is behind the source? This source reveals information about which license server was the first on the market and they claim that it was theirs. This may be true, but there is also a second necessary question to ask. Why is the source created? Is this to just inform about the history or to sell their license server? They claim that they were first on the market and that they know the most about license servers. This may not be true. The readers who collects information is very dependent on that the information is reliable, but in this case it's very complicated and blurry who created the first license server. For this thesis the option was very clear and that was to presume that they were the first because of the founding age (Kristina Alexanderson, 2016).

In this report there is a section about product development, Agile & Scrum. Here the selected source is CPRIME. It is a company that provides a service to improve structure in product development. They explain what scrum and agile is. The purpose of the page creation is though unclear. The most liable reason is to provide information and inform. Their goal could also be to slowly affect the reader and convince the reader to purchase their service. It could also be to try out their service because of their

quality and reviews. Here the source was chosen because they give realistic information about agile product development (Kristina Alexanderson, 2016).

There are two sources that explain the performance of encryption. One source is a study written by students for different universities and one is from a known developer site where everybody could state the fact and comment. Two sources were chosen because the example in the webpage, stack exchange may not be accurate. Therefore the reader of this report can ensure its calculation and information by reading the report study “*An experimental study on Performance Evaluation of Asymmetric Encryption Algorithms*”. The purpose of the sources is to inform the reader and make more logic into asymmetric performance in relative to symmetric performance (Kristina Alexanderson, 2016).

The last source is about source criticism on internet and the whole section 3.3 Source Criticism is based on the question on page 11 in the PDF document.

# CHAPTER 4

## 4 Results

---

*This chapter presents the results and what I have achieved with this thesis based on the challenges and the requirements. I would also recommend taking a look at the workflow of this licensing solution for better background understanding, see Appendix A1 (1, 2 and 3).*

### 4.1 Work result

The result of this thesis includes three different components. The back-office system, a license server and a test component that communicates with the license server to validate that the license key is valid. The successful and implemented parts are the back-office system and the license server. The license component isn't finished and therefore isn't explained in the results. Although there is an almost completed plan for how to proceed with its implementation. There is more information about this in section 4.1.9 License Component and in 6 Future Work.

See figure 8 for a visual overview. If a sales person working for Tetra Pak manages to sell a product, they input all the information about a site. The information is for example site characteristics, features to include and market company information. After that a form is being created and saved in the database. Later on the form is evaluated by a back-office user (Tetra Pak employee) and approved. Then a LICBASE-file is generated that include all the information about that site and sent to the customer site where the license server is installed. Here an engineer working on the customer site imports the file to the license server and exports another file called the SITEKEY. This file includes information about the site and the hardware information about the host machine. The file is then sent back to the back-office where it's uploaded to the system. After completing that step the engineer can successfully create a LIC-file containing a serial key generated based on the hardware information and the site information. The key for the site information isn't completed yet, but in the later result there is a prototype created for this. Currently the key is only hashed and based on the hardware information.

When the user successfully creates the key, the user can enter the key in the license server installed at customer site. If the hardware information present at the license server is equal to the key it's a success and the software is ready to use. For security reasons there is also a license component that must validate the licenses. Therefore the license component in the software that's licensed has to communicate with the license server all the time.



## Licensing Workflow – New installation

S = Sales user  
 BO = Back-Office user  
 E = Engineer user

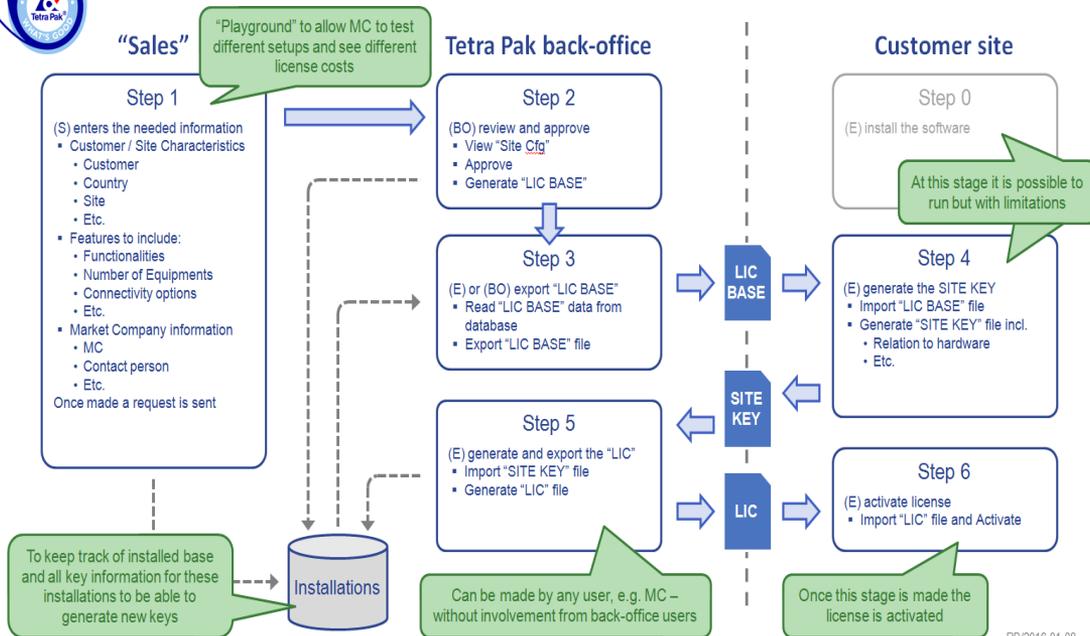
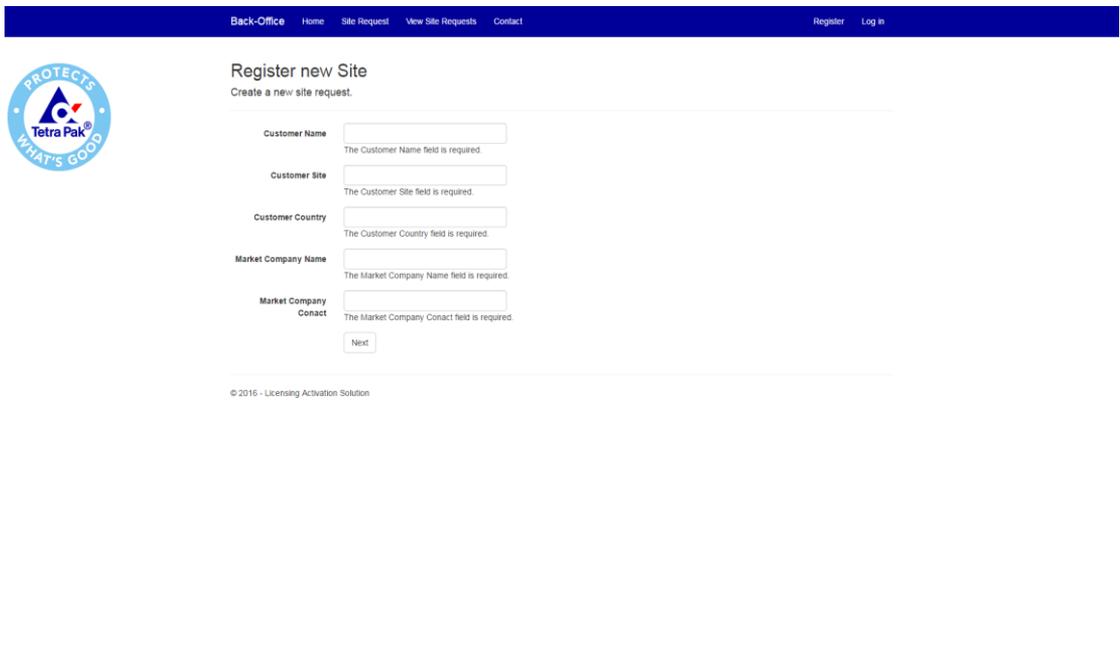


Figure 8 - Licensing Activation Solution

## 4.1.1 Back-Office Part 1 – Register site

In the back-office system you can register a new site with all its characteristics. In this case the characteristics are for example customer name, customer city, customer site, market company name, market company contact. Functionalities, equipment and connectivity are displayed on different form pages where you press next when you filled out the page. There are totally 6 different pages. See figure 9 for the first page.



The screenshot shows a web interface for registering a new site. At the top, there is a dark blue navigation bar with links: Back-Office, Home, Site Request, View Site Requests, Contact, Register, and Log in. On the left side, there is a circular logo for Tetra Pak with the text 'PROTECTS WHAT'S GOOD'. The main content area is titled 'Register new Site' and contains the instruction 'Create a new site request.' Below this, there are five input fields, each with a 'Next' button and a validation message: 'The Customer Name field is required.', 'The Customer Site field is required.', 'The Customer Country field is required.', 'The Market Company Name field is required.', and 'The Market Company Contact field is required.' At the bottom left, there is a copyright notice: '© 2016 - Licensing Activation Solution'.

Figure 9 - Back-Office Part 1 - Registration of new site

## 4.1.2 Back-Office Part 2 – View sites

In figure 9 you will see a page called “View Site Request” where all the sites that you request are stored. I haven’t reached as far as the goal was because of the time left and that was to

implement a search function for the whole site part. Also a modification button, so that the sales person or the back-office user can manage changes on a site. Here a search field should be implemented and no sites should be visible until you search for them. There is also a difference in user authority and only an engineer or a back-office user can approve a site request. If site request is approved you can see that the “Approved” label changes to yes. If you then press Details at the right of every site you come to the details of the site.

**View Requests**

Title:

Details

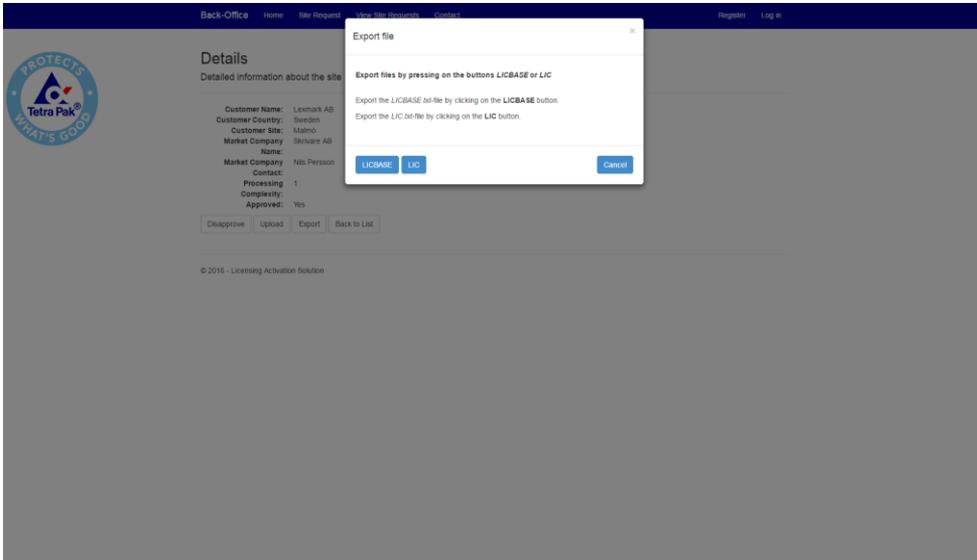
Customer Name	Customer Country	Customer Site	Approved	Market Company Name
Tetra Pak	Sweden	Bryggaregatan	Yes	
IKEA AB	Sweden	Älmhult	Yes	
ICA AB	Sweden	Helsingborg	Yes	
Svedbank AB	Sweden	Malmö	Yes	Klarna AB
ICA AB	Sweden	Perstorp	Yes	Frukt AB
Telenor AB	Sweden	Stockholm	Yes	Telia AB
Arla AB	Sweden	Ähus	Yes	Tetra Pak AB
Lexmark AB	Sweden	Malmö	Yes	Skivare AB
Pelles AB	Sweden	Malmö	No	Maskin AB
jk	jk	jk	Yes	jk
jk	jk	jk	No	jk
9	8	89	No	89

© 2016 - Licensing Activation Solution

Figure 10 – Back-Office Part 2 – View registered site requests

### 4.1.3 Back-Office Part 3 – Details/Export

In the details of the site there are a few options to choose from. See figure 11. Here is partly the information about the site displayed and should be all the information such as equipment, functionality and connectivity. In the previous figure 10, you could only see the most necessary information about the site. In



**Figure 11 - Back-Office Part 3 - Details of site, Export Pop-up**

details you should be able to view it all. Here you can approve and disapprove a site. Depending on if the site is approved you can also download a LICBASE-file which is a text file containing information about the site, such as company name and site functionality. Also in this case you shouldn't be able to see the export button for LIC-file and Upload-button because everything is done by steps. These steps follow a principle that you can't accomplish something if the previous steps aren't finished.

There should also be displayed a status field to the right of the details information. This is done to display for the user where in the process they are right now. If we now follow the workflow the user approves a site and can later on generate a LICBASE-file that can be downloaded.

## 4.1.4 License Server Part 1 – Upload

The next step is the license server where you upload the LICBASE-file to the license server. When this is completed successfully we export the SITEKEY.enc file. The SITEKEY.enc file is a combination with information from LICBASE.txt-file and the hardware information on the current computer. If the license server is installed on a virtual machine the system retrieves host information from the guest remote machine. See figure 12.

There are pages with help text that guides the user through the system if the user uploads or exports a file. When you've successfully uploaded the LICBASE.txt file the system tells you that the SITEKEY file is ready for download.

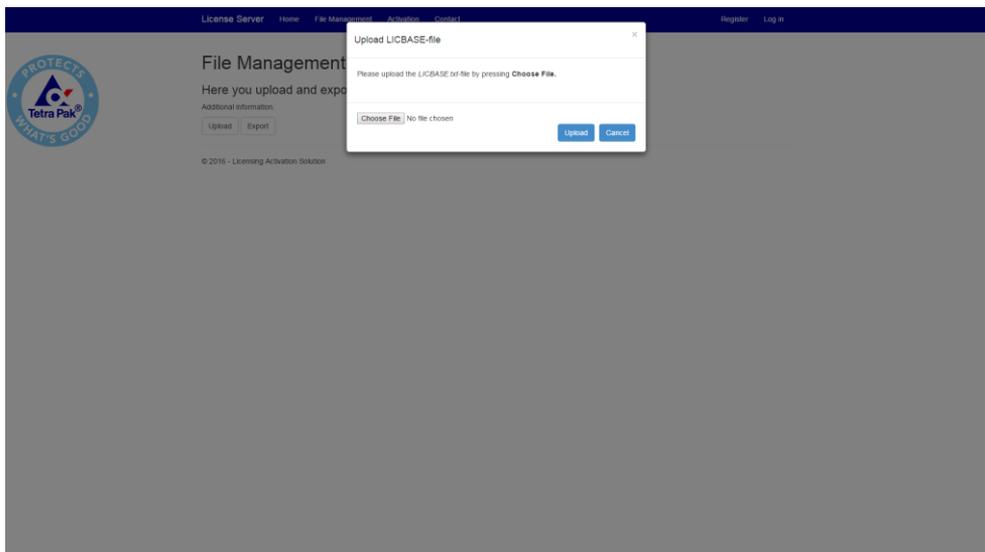


Figure 12 - License Server Part 1 - Upload LICBASE

## 4.1.5 License Server Part 2 – Export

In part two of the license server the system encourages the user to download the SITEKEY.enc file if the upload of the LICBASE.txt file was successful. Here you go to the other button on the page “File Management” and press Export. You will then see the option to export the SITEKEY.enc-file by pressing SITEKEY, see more in figure 13. Here you choose where to put your download again.

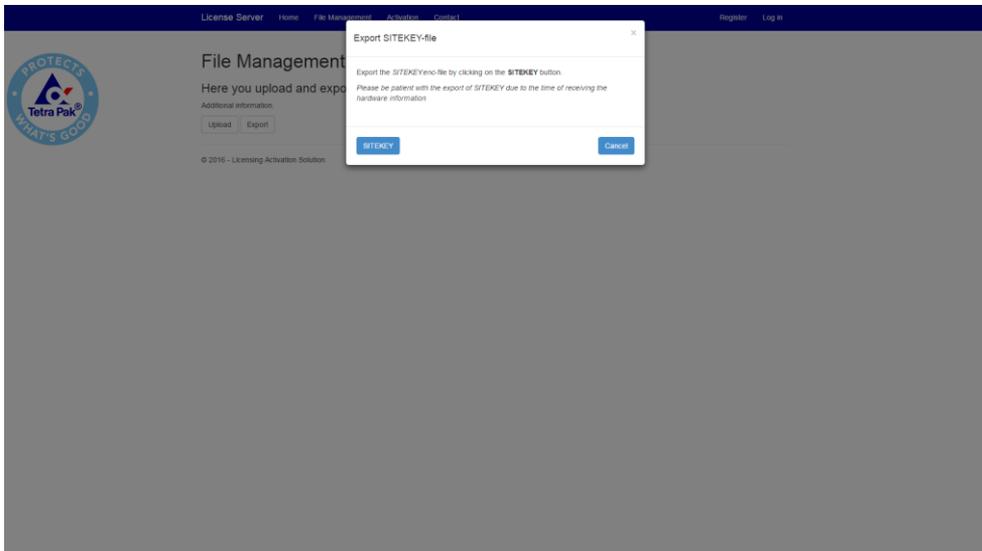


Figure 13 - License Server Part 2 - Export SITEKEY

## 4.1.6 Back-Office Part 4 – Upload SITEKEY

In part four of the Back-Office system the SITEKEY.enc is uploaded to the system, by pressing “Upload” on the Details of the right site as shown in the figure 14 below. If you upload the wrong SITEKEY to the wrong site you will get an error that tells you to find the right site. This is required by the engineers and back-office users because the system can’t tell which site the SITEKEY includes. The encryption won’t allow it, so the engineer at customer site must communicate with the person uploading the SITEKEY-file. If the right SITEKEY-file is successfully uploaded to the system, the system tells the user that the LIC-file is ready to export.

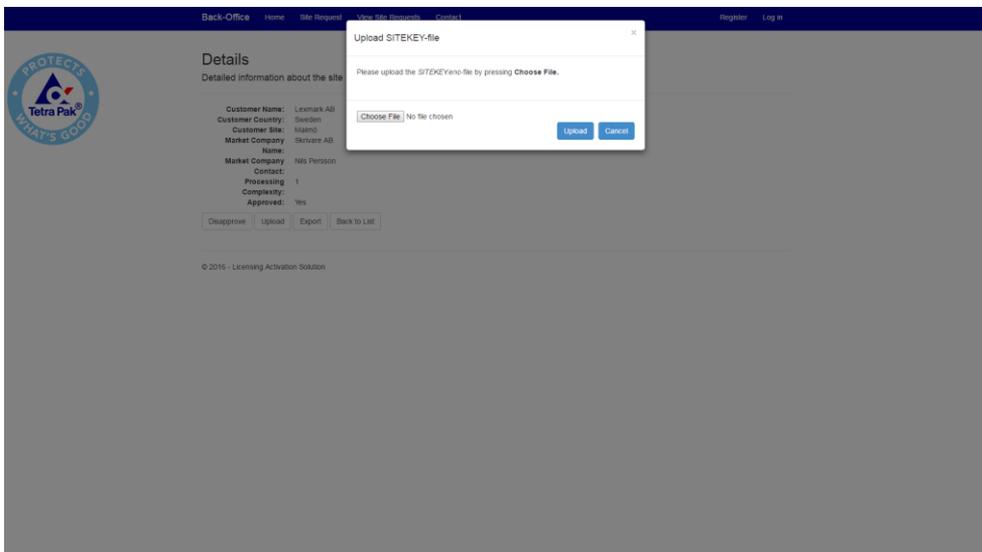


Figure 14 - Back-Office Part 4 – Upload SITEKEY

## 4.1.7 Back-Office Part 5 – Export LIC

In the last step of the Back-Office system the LIC-file is exported, see figure 15. The LIC file contains a product key combined with a site key. The product key is the hardware information in hash form retrieved from the SITEKEY, e.g. ABCD-EFGH-IJKL-1234. The site key is generated based on all the information about a site. 107-A10-987-B09-121 for example, which represent 107 tanks at the customer sites factory.

Together the key will look for example like ABCD-EFGH-IJKL-1234-107-A10-987-B09-121 where the first part will be taken and compared to the hardware information at customer site. The second part will develop into an explore field where you can see all the information about the site on the customer site.

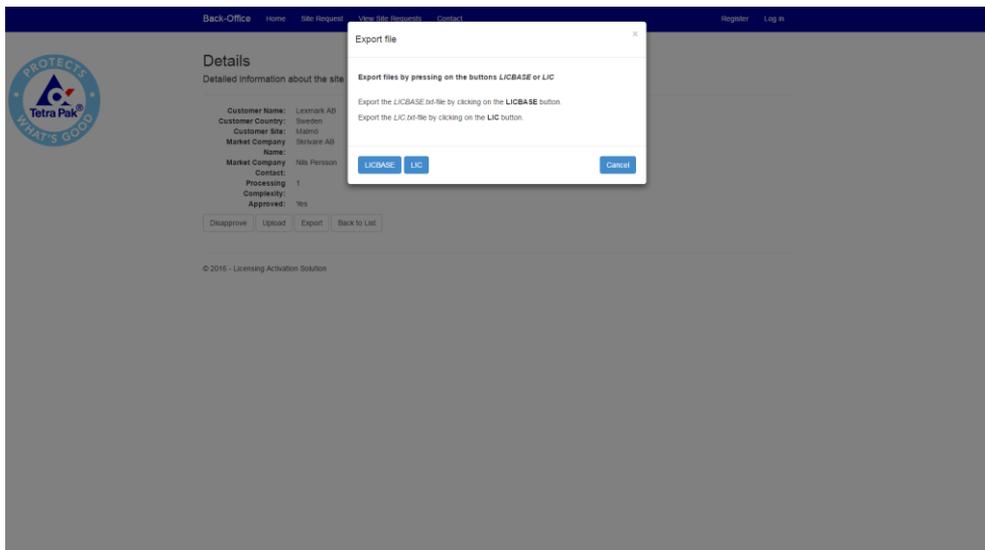


Figure 15 - Back-Office Part 5 - Export LIC

### **4.1.8 License Server Part 3 – Activate**

This is the overall last step for the Licensing Activation Solution, where you activate your key. In the field the user enters the key and presses activate. If the hardware information is equal on this machine as it was when you first exported the SITEKEY, it will grant access to the software. Later on a file is saved in the system by the license server. This is done for the license component where it asks the license server if the license is granted. The license server will check in the file and answer thereafter.

### **4.1.9 License Component**

The license component will be a plug-in to test the license server's communication. When the license key is approved in the License Server Part 3 – Activation, the license server saves a file with that activation code. Later on when you send a request from software that is licensed the hardware information will be compared. After that it will check if the license server still has grant access. If it has it will send a message back to the software telling that you can use the software with 100% functionality.

If the license server tells the software that the hardware doesn't match, which could happen if the machine that licenses the server is installed on is a virtual machine. If the machine has been moved in an attempt to use the software in a way that isn't allowed the system will identify that break by hardware information and send out a popup message to the software. This popup will tell that the software's functionality isn't allowed and

would then try to close that section in the program that the license component is programmed to control.

The component as mentioned earlier isn't completed, but will be in the future because it's Licensing Activation Solutions last step. The design is completed and will soon be implemented and tested.

# CHAPTER 5

## 5 Conclusions

---

*This chapter contains the conclusions of the result and the problems that the thesis was challenging, but also how this product is going to fulfil a purpose at Tetra Pak.*

### 5.1 Result of problems

This section is about answering the problem definition that you can find in 1.3 Problem definitions. Here the questions are presented and answered how and why the solution was produced.

- **How to control which software the customer has installed?**

A control is established through the software by license keys and will save all the information in the database such as who the customer is and the customers country and site. Everything is connected to a site identification number and later on there will be pages where you can list all the

site requests that the user/users create. More information on this can be found in the sections 4.1 Work Result and 4.1.2 Back-Office Part 2 – View Sites.

- **What do you do to avoid that the software can be duplicated on a machine?**

In the license agreement it is stated that a machine license can be installed on a physical and virtual machine. The user can create infinite numbers of virtual machine to run the software. You can activate a license, run software and then try to run the software on another computer. The license is connected to the hardware information and if you move, the license server will detect that and stop the functionality on the software. In the section 3.2 Problem Solution – License Server there is more detailed information about the virtual hardware information.

- **How to develop an algorithm to identify a secure way for license key generation?**

The algorithm is based on hex code and hash codes. It takes the hardware information and first hashes it by Microsoft's built in hash-tool. Then turns the symbols into hex characters, either a number from 1-9 or a letter from A-F. This information is located in 4.1.7 Back-Office Part 5 – Export LIC and in 3.2.1 Problem Solution – Back-Office.

- **How to implement flexibility to the system and the database for easily adding new application and functionalities?**

The database is designed and structured in a way of look-up tables, making the database flexible for adding new application or raw data. This information is located 3.2 Problem Solution.

## **5.2 Conclusion**

The conclusion of this thesis is a system called Licensing Activation Solution that has been designed and developed. The purpose is to serve and license Tetra Pak's future product called the new MES Platform (present name) that is under development. A person that is handling sales can now enter the information about a site and later on a back-office user or Tetra Pak engineer can approve the site and export the site characteristics to a LICBASE-file. The file is sent to the customer site where the license server is installed.

While at the license server the LICBASE-file is being uploaded to the system and a SITEKEY-file is exported containing information of LICBASE and the hardware information from a physical computer or a physical host through a virtual machine. When the file later is transported back to the back-office system and uploaded the LIC-file is being created that is similar to a product key. This product key is being entered into the license server at customer site and if comparison with current computer at customer site is equal the license key is activated.

When the Licensing Activation Solution system is complete you can sell a machine license that's connected to one license and you can create an infinite number of virtual machines on that license. There is also the possibility to purchase a site license that covers all the physical computers on the site. Also here you can then create infinite numbers of virtual machines on the site license. The price of a license isn't an assignment in this thesis. The system had to be developed to adapt the sale division though and for that making the form very user friendly.

Validation of the licenses is controlled by a license server that is installed on the customer site. The license server has a communication after LIC activation, with the license component. The license component is installed on the new MES Platform.

Licensing Activation Solution is at version 1.0 and is going to be maintained for upgrades and bugs by the author of this thesis. This is during a couple of more months on behalf of Tetra Pak and by the technical and function product owners and their developers. Therefore it is very important to make the complex implementations accurate from the beginning, such as encryption, retrieval of hardware information and the generating algorithms for keys.

## **5.3 Product utility**

This product will be used and commissioned by Tetra Pak together with their new product MES in August 2016. They are

then able to control in which extent the product will be distributed and sold. This is of course part for an economical aspect, but also for setting up statics for how the product is handled and received by the market. The product almost fulfils the requirements of the thesis.



# CHAPTER 6

## 6 Future Work

---

*This last chapter of the thesis will be about future work in Tetra Pak with the Licensing Solution product and what I am going further develop.*

### 6.1 Future development

Licensing Activation Solution is almost completed. The system works, but the license component that should be installed into the licensed software isn't complete. Therefore the author of this thesis is going to continue working with the product over the summer 2016 until it's completed on behalf of Tetra Pak. The parts that should be improved are the form, to make it more user friendly which means to simplify the form. Also to send out more help information that includes errors if mistakes are made. A new key file that can be exported is going to be implemented. It is called LIC Limited and it will expire after a chosen number of days. A complex search implementation is going to be designed and developed for further user mobility and flexibility.

Later on a log-in system is going to be set with an AD against Tetra Pak users.

The system will also need an upgrade on the database design because of the options in the form. This is because a sales person that doesn't understand the terms needs to have guidance and different options. There is also the option of changing a site request.

There is a vision of creating multiple site requests for comparison in the price calculation, but that under development by Process Management division at Tetra Pak.

# 7           References

Tetra Pak AB (). *Tetra Pak History*. Available at:  
<http://www.tetrapak.com/se/about/history> [2016-04-20]

Tetra Pak AB (). *Short description of Tetra Pak*. Available at:  
<http://www.tetrapak.com/se/about/tetra-pak-in-brief> [2016-04-20]

The University of North Carolina (). *Software Licensing*. Available at: <https://its.uncg.edu/Software/Licensing/> (Accessed: 2016-04-06)

wDay (). *What is hardware-locked licensing?* Available at:  
<http://wyday.com/lime/m/features/why/> (Accessed: 2016-04-25)

Microsoft (Last Review: 10/26/2007 18:38:36 - Revision: 1.3). *Description of symmetric and asymmetric encryption*. Available at: <https://support.microsoft.com/en-us/kb/246071> (Accessed: 2016-04-25)

Licensing EXPO (). *What is licensing?* Available at:  
<http://www.licensingexpo.com/licensing-expo/education/what-licensing> (Accessed: 2016-04-27)

Microsoft (). *Planning the License Server*. Available at:  
<https://technet.microsoft.com/en-us/library/cc738769%28v=ws.10%29.aspx?f=255&MSPPError=-2147217396> (Accessed: 2016-04-27)

Sassafras Software (). *KeyServer*. Available at:  
<http://www.sassafras.com/> (Accessed: 2016-04-27)

O'REILLY Windows development center (). *What is ASP.NET?*  
Available at:  
<http://archive.oreilly.com/pub/a/dotnet/2005/09/19/what-is-asp-net.html> (Accessed: 2016-04-27)

Microsoft (). *ASP.NET MVC Overview*. Available at:  
[https://msdn.microsoft.com/en-us/library/dd381412\(v=vs.108\).aspx](https://msdn.microsoft.com/en-us/library/dd381412(v=vs.108).aspx) (Accessed: 2016-04-27)

CPRIME (). *What is agile? What is Scrum?* Available at:  
<https://www.cprime.com/resources/what-is-agile-what-is-scrum/>  
(Accessed: 2016-04-28)

World Scientific and Engineering (). *An experimental study on Performance Evaluation of Asymmetric Encryption Algorithms* by Shahzadi Farah, Younas Javed, Azra Shamim and Tabassam Nawaz. Available at: <http://www.wseas.us/e-library/conferences/2012/Paris/ECCS/ECCS-19.pdf> (Accessed: 2016-05-17)

Cryptography, Stack exchange (). *Why is asymmetric cryptography bad for huge data?* Available at:  
<http://crypto.stackexchange.com/questions/5782/why-is-asymmetric-cryptography-bad-for-huge-data> (Accessed: 2016-05-17)

IIS (). *Källkritik på Internet* by Kristina Alexanderson. Available at: <https://www.iis.se/lar-dig-mer/guider/kallkritik-pa-internet/> (Accessed 2016-05-18)

## 7.1 Picture References

[Figure 1] Picture of licensing solution – Creator: Dervis Avdic (Accessed: 2016-06-01)

[Figure 2] <https://www.cybrary.it/0p3n/symmetric-encryption/> (Accessed: 2016-05-16)

[Figure 3] <http://codingatschool.weebly.com/asymmetric-keys-and-encryption-methods.html> (Accessed: 2016-05-16)

[Figure 4] <https://jazz.net/library/article/548> (Accessed: 2015-05-16)

[Figure 5] <http://www.codeproject.com/Tips/669195/MVC-Introduction> (Accessed: 2015-05-16)

[Figure 6] [https://en.wikipedia.org/wiki/Hash\\_function#/media/File:Hash\\_table\\_4\\_1\\_1\\_0\\_0\\_1\\_0\\_LL.svg](https://en.wikipedia.org/wiki/Hash_function#/media/File:Hash_table_4_1_1_0_0_1_0_LL.svg) (Accessed: 2016-05-16)

[Figure 7] <https://javierguillen.files.wordpress.com/2012/07/image26.png> (Accessed: 2016-05-16)

[Figure 8] Picture of project – Creator: Peter Bjernetun  
(Accessed: 2016-04-16)

[Figure 9] Picture of project – Creator: Dervis Avdic (Accessed:  
2016-04-16)

[Figure 10] Picture of project – Creator: Dervis Avdic  
(Accessed: 2016-04-16)

[Figure 11] Picture of project – Creator: Dervis Avdic  
(Accessed: 2016-04-16)

[Figure 12] Picture of project – Creator: Dervis Avdic  
(Accessed: 2016-04-16)

[Figure 13] Picture of project – Creator: Dervis Avdic  
(Accessed: 2016-04-16)

[Figure 14] Picture of project – Creator: Dervis Avdic  
(Accessed: 2016-04-16)

[Figure 15] Picture of project – Creator: Dervis Avdic  
(Accessed: 2016-04-16)

## **7.2 Appendix References**

[1 – 1] Appendix of project – Creator: Peter Bjernetun  
(Accessed: 2016-04-16)

[1 – 2] Appendix of project – Creator: Peter Bjernetun  
(Accessed: 2016-04-16)

[1 – 3] Appendix of project – Creator: Peter Bjernetun  
(Accessed: 2016-04-16)

[2 – 1] Appendix of project – Creator: Dervis Avdic (Accessed:  
2016-04-16)



# List of Acronyms

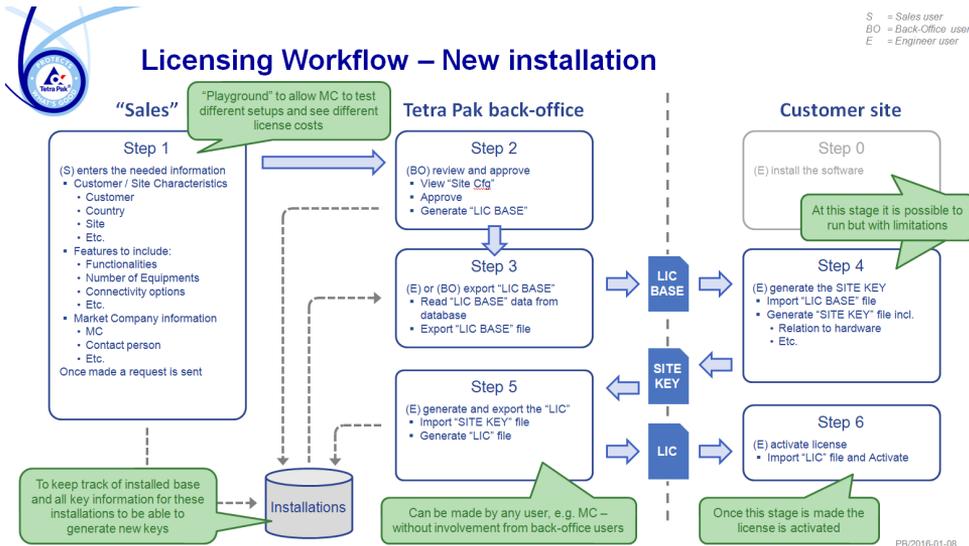
SQL	Structured Query Language
AD	Active Directory
UI	User Interface
IT	Information Technology
ASP	Active Server Pages
NET	NET Framework for ASP
MVC	Model View Controller
MES NG	Manufacturing Execution System New Gen.



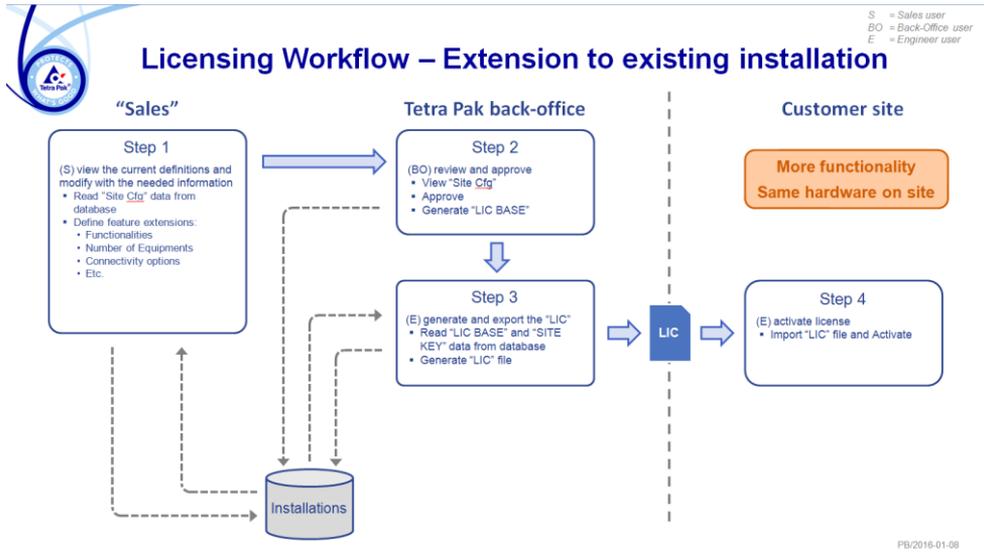
# Appendix 1

## 1. Workflow

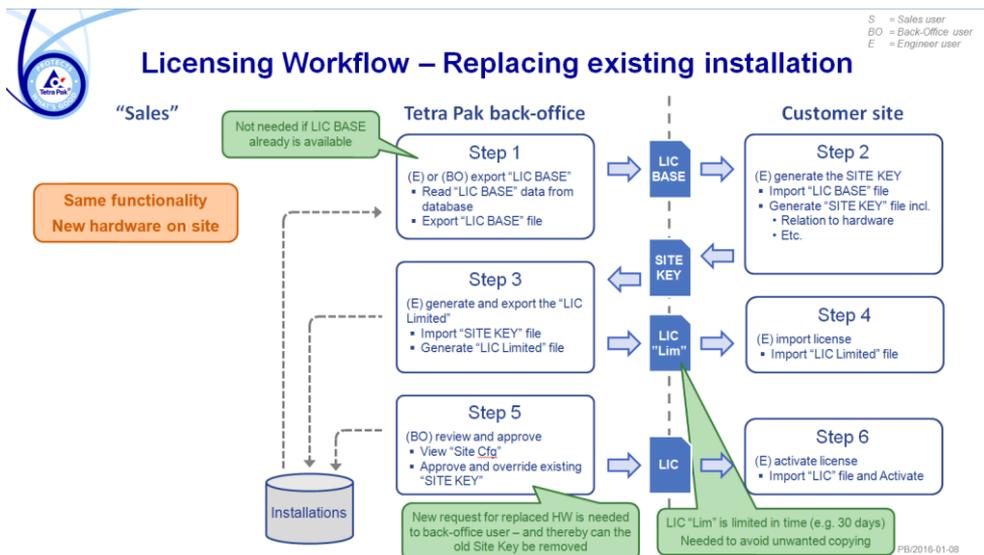
### 1. Workflow-picture of a new installation



## 2. Workflow-picture of extension to existing installation



## 3. Workflow-picture of replacing existing installation



# 2

# Working methods

