SECURE EXTERNAL ACCESS TO ODOO

HAMK
HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

Bachelor’s thesis

Häme University of Applied Sciences,
Degree Programme in Business Information Technology

Spring 2020

Minna Katermaa

HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

TIIVISTELMÄ

Tietojenkäsittelyn koulutusohjelma
Hämeenlinnan korkeakoulukeskus

Tekijä	Minna Katermaa	Vuosi 2020
Otsikko	**Odoon ulkoinen käyttö turvallisesti**	
Työn ohjaaja	Lasse Seppänen	

TIIVISTELMÄ

Tämän kirjallisuustutkimuksena toteutettavan opinnäytetyön tavoitteena on tutkia kuinka Odoo-toiminnanohjausjärjestelmän (ERP) käyttöönotto toteutetaan turvallisesti ulkoisten toimijoiden kanssa. Lisäksi tavoitteena on löytää parhaita käytänteitä, miten ulkoiset käyttäjät liitetään ERP:iin vaarantamatta yrityksen ydintietoa. Työn toimeksiantaja on suomalainen teollisuusalan yritys, jolla on käytössä avoimen lähdekoodin ERP-järjestelmä Odoo. Yrityksessä on tulevaisuudessa vahva tarve laajentaa Odoon toiminnallisuuksia siten, että myös ulkoiset toimijat kuten partnerit, toimittajat sekä asiakkaat saadaan integroitua osaksi yrityksen Odoota.

Teoriaosuudessa perehdytään ERP-järjestelmien tarkoitukseen, etuihin ja haittapuoliin sekä eri toimitus- sekä hankintamalleihin. Lisäksi luodaan lyhyt katsaus tietoturvaan, keskittyen etenkin sovellusten ja tietokantojen tietoturvaan. Lisäksi tarkastellaan myös ERP-tietoturvamenetelmiä. ERP:n arkkitehtuurimalli yhdessä sovellus- ja tietokantaturvamekanismien kanssa nähdään ratkaisevaksi tietoturvahaasteisiin vastaamisessa. Kolmitasoinen arkkitehtuurimalli, jota myös Odoo tukee, nähdään turvallisempana ratkaisuna kuin yksi- tai kaksitasoiset mallit. Koska yritykset toimivat tänä päivänä verkostoissa, joissa liiketoimintapartnerit ovat osa ERP-järjestelmää, tarvitaan uudenlaisia tapoja mahdollistamaan pääsy ERP:iin mutta kuitenkin samalla suojaamaan yrityksen ydindataa. Perinteiset ERP-tietoturvamekanismit eivät välttämättä enää riitä. Tarvitaan uusia ratkaisuja, joilla ERP:n turvallinen ulkoinen käyttö voidaan mahdollistaa. Nämä parhaat käytänteet ovat vasta muovautumassa.

Avainsanat ERP, ERP-tietoturva, Odoo-ERP, ERP-arkkitehtuuri.

Sivut 50 sivua

HAMK
HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

ABSTRACT

Degree Programme in Business Information Technology
Hämeenlinna University Center

Author	Minna Katermaa	**Year** 2020
Subject	**Secure external access to Odoo**	
Supervisor	Lasse Seppänen	

ABSTRACT

The aim of this thesis is to investigate based on the literature review that how to implement Odoo enterprise resource planning system (ERP) securely with external connectivity. Additionally, the target was to form an understanding of the best practises available to create the external connections in ERP without risking the core data of the company. The commissioner of this thesis is a manufacturing company in Finland which is using open source ERP called Odoo. In the future there is a strong need to enable connectivity also with external partners meaning that several modules from the same ERP application need to be enabled for external usage.

The theory framework is introducing main purpose of ERP systems, its advantages, disadvantages, different delivery models and acquisition options. Information security on high level is introduced with focus on application, database and ERP specific security aspects. The architectural structure of ERPs, together with application and database security mechanisms, are seen crucial to respond to security challenges. The three tier architecture model, supported also by Odoo, is seen more secure than one or two tier models. As companies today are operating over the traditional company borders, secure business partner access to enterprise data is needed. Traditional security methods of ERP have to be re-considered to enable usage also with external connections to fulfil the security needs of companies. It seems that at the moment the best practise security mechanisms for web ERPs are not yet widely established.

Keywords ERP, ERP security, Odoo ERP, ERP architecture

Pages 50 pages

CONTENTS

1 INTRODUCTION

Enterprises are facing the need to expand business capabilities and provide real-time information access and richer user interactions. Companies are part of large global networks, breaking down the traditional enterprise boundaries. New business models in the global environment including the growing need for user interaction together with new internet capabilities require business to respond with latest technology applications and solutions. (Cisco, 2019)

An Enterprise Resource Planning (ERP) system is an integrated information system that brings together key departments of an organization and targets to automate business processes. It supports efficient data flows, real time data visibility and operations such as accounting, material resource planning, supply chain management, human resources, sales and marketing, customer relationship management etc.

ERPs host the core, integral data of a company and thus data security of an ERP solution is an important component of company's cybersecurity strategy. (Cisco, 2019) ERP systems are now facing digital transformation that will make them part of integrated and collaborative partner network with web and even wireless capabilities. This will raise new requirements towards ERP security because of the fact that ERPs carry data of high confidentiality but at the same time vulnerability of systems is increasing. Many ERP vendors have integrated their security solutions, but with new collaborative environments also new ERP security solutions are needed. (Panwar, Kumar & Pandey, 2016, 1)

Target of this thesis is to study and investigate how to implement securely an open source ERP system both in internal and external use in a company where several modules from the same ERP application need to be enabled for external connectivity and usage.

The research questions of this study are:

1) How to implement Odoo ERP securely and also provide external connectivity?

2) What are the best practises available to create the connections without risking the core data of the company?

2 ENTERPRISE RESOURCE PLANNING (ERP) SYSTEMS

Enterprise Resource Planning (ERP) systems are comprehensive, enterprise-wide modular software packages that seek to integrate people, business processes and functions holistically from company's information and IT architecture point of view (Lahti & Salminen, 2014, 40; Klaus et al., 2000, via Duan, Faker, Fesak & Stuart, 2012, 2.) ERP systems are typically combining several key organizational operations to functional modules like finance, marketing, human resources, operations, purchasing and logistics, which are using common database across them. Aggregation and integration of data in the ERP platform is enabling automated process links and data utilization across organization. (APICS, 2012; Berchet & Habchi, 2005) Integration of these functions increases automation and data correctness because data needs to be added only once to a common database instead of repeating the same data to several applications (Lahti & Salminen, 2014, 42). From functional perspective ERP systems needs to collect, process, present, analyse and store data (Pascu, 2013, p. 37).

According to Gartner (n.d.), benefits of an ERP applications are in the automation and supporting administrative and operational business processes across multiple industries. Even though ERP deployments can be complex and expensive investments which might cause doubts on ERP benefits in some companies, business benefits can be found from areas such as: ERP contributing to business innovation, more standardized and efficient business processes including reductions in IT expenses.

ERP implementation is often mentioned to be a remarkable investment. The IT cost saving aspect is coming from efficiency improvement and unified IT costs; ERP enables to centralize the cost base around one system only instead of supporting multiple systems that all needed dedicated staff, infrastructure, support teams and licenses. (O'Shaughnessy, n.d.)

The benefits of traditional ERP systems have been in maturity of the system functionality as well as ability to perform more customization and integration on them (Duan et. al., 2012, p. 1). Generally, the ERP systems in traditional mode are known for the modification needs to fulfil the business requirements. However, this is causing significant costs to companies when configuring the system for the implementation. Due to this customization, also ERP maintenance phase is causing high need or resources. Additional challenge that the customization might also cause is a vendor lock-in; changing the system supplier is not that easy anymore and could be even technically impossible. (Eskeli, Heinonen, Matinmikko, Parviainen & Pussinen, 2010) Focus on traditional ERP-systems was on strong optimization of data and processes but such factors as user-experience or user-engagement were not that much in the development focus (Aspirion, Schneider & Grimberg, 2018, 15-17).

ERP originally targeted to integrate internal business functions of a company. Due to the changes in business environment where business collaboration became a key success factor, companies needed to integrate more across its national borders and strategic partners in their network. This changed the ERP integration requirements to integrate and coordinate business processes with external stakeholders; customers, suppliers and other business partners. (Barki & Pinsonneault, 2002; Kelle & Akbulutb, 2005) Furthermore, requirements towards web interfaces with customers and mobile access to enterprise data has additional challenges to ERP environment (Vathanophas, 2007, p. 444). Nowadays IT-industry has faced a shift towards cloud computing. This has been the fastest growing segments of IT-industry during the past decade. Also, ERP systems are following this trend. (Duan et. al., 2012, p. 1-2)

In fact, Panwar et. al. (2016) discuss the relationship of e-commerce and the scope of the ERP being nowadays blurred. Traditionally EPR has been more concerned about organizations' internal functionalities whereas e-commerce is about business across companies. This blurs the company boundaries and thus discussion on the technical issues should be expanded from "internal" ERP context to cover the wider viewpoint.

2.1 Proprietary ERP vs Open Source ERP

One way to examine ERPs is to differentiate between proprietary and open source ERP systems. Proprietary ERP refers to systems where publisher is holding the intellectual property rights, source code and licencing rights. Companies can purchase the licence to access and use the software. Free/Open Source ERP - FOSERP can be community based or sponsored by some organization. Software can be loaded free of charge and there is full access to the source code which can be modified based on company's own need. Some smaller vendors utilize a free distribution system for their source code, relying on various business models for corporate success. Some ERP vendors use community developed components for various purposes to support their proprietorial software. (Olson, Johansson & Carvalho, 2018)

ERP systems became common for large organizations in the 1990s. Later on, in the 21st century, ERPs have been further expanded with functions such as supply chain management (SCM) and customer relationship management (CRM) as well as access through the Web. In the past ERP acquisition has required a heavy investment by enterprises because of high initial investment and resources required to deploy and maintain the system. For small businesses, investment often was too high, leaving them into a position where they could not afford the implementation costs. This created a niche market of ERP in the sector small and medium-sized enterprises (SME) to which vendors have developed ERPs with information technologies to lower the adoption costs and by developing more simple ERP versions. Moreover, new ways of delivering ERP via internet emerged creating opportunities also for Free/Open Source ERPs. Nowadays FOS-ERP has become a viable alternative especially for small and medium size enterprises. (Olson et. al., 2018)

2.2 **ERP delivery models**

ERP systems began as systems which only large manufacturing companies could afford to implement. Gradually ERPs have been gaining foothold also in other industry sectors e.g. retail, healthcare and government. Mainly this is because of the web ERP and web technology enablement with e-commerce and e-business capabilities. (Ip & Ng, 2013) Cloud technology has been seen enabler for SME enterprises to also adopt cloud-based ERPs. Reasons for choosing cloud ERP model are the speed to deliver and ease of use as it is freely accessible via internet. Cost reduction plays an important role and that is achieved e.g. via reduced demand on in-house IT-stuff, upgrades included in service, reduction on hardware costs. (Mattison & Raj, 2012, p. 4)

There are many different ERP delivery model options for a company to choose from (Figure 1). Aspects to consider are e.g. costs, workload, security, implementation efforts and scalability.

2.2.1 On-premise and hosted models

In the past ERP systems have been mainly implemented on premise. ERP delivered in an on-premise model requires a company to have their own physical servers in their premises to which software is installed to. Company also needs to acquire the licences to run the software and is responsible of maintaining the software. In case of new software versions company is required to purchase new licences for them. (Profiz Business Solutions Oyj, 2013)

In hosted ERP -model software is delivered via the internet, remotely by a provider, but company acquires and owns the licences. Hosting provider will place the software on his own secured data centre servers or external host servers. Server will be separate, dedicated server for the customer company. In this model vendor is responsible for server back-ups and maintenance. Software is accessed through virtual private network. This approach reduces the maintenance and hardware service costs. At the same time, it reduces the ability to control the internal data because provider is responsible of the data stability and data security. (Profiz Business Solutions Oyj, 2013; & Visma Software Oy, 2019)

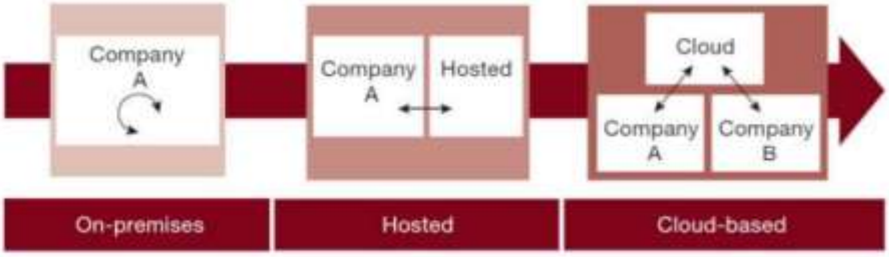

Figure 1. ERP delivery models (Rabaya & Graffi, 2019)

2.2.2 Cloud Computing and cloud ERP

Cloud ERP is defined by Salim, et. al. (2015, p. 220) as “commercial software packages that enable the integration of business processes and transaction oriented data throughout the organization using a model that enables ubiquitous, convenient, on-demand network access within minimal management effort or service provider reaction.”

Benefits of using the web services are in the easier integration and reduced costs. Web services provides clients and outsourcing vendors seamless access to the information and ERP applications without the need to install the ERP software. Partner and client communication happen via web services which can connect to actual “legacy” ERP software. (Panwar et. al., 2016)

2.2.3 Service delivery models in cloud computing

Cloud computing is a service delivery model based on internet where services, computing and storage is provided for users in all markets. There are different types of cloud services, and in order to understand the concept of cloud properly, it is important to understand also the different software delivery models (Figure 2). Typically cloud computing can be divided to three different categories in terms of service models: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). (Duan & al., 2012, p. 3; Sharma, Bansal & Sharma, 2012, p. 1)

Software as a Service (SaaS) is a capability to provide applications to the users via web browser or program interface i.e. applications that are running on cloud infrastructure and are accessible from various devices. Customer does now own the software licence but instead uses the service against subscription fee. Software is hosted by the software vendor, managed centrally and service is carefree for the end user for such responsibilities as managing or controlling the network, servers, operating systems and storage. Customer does not need to implement or deploy the software at the customer site. In addition, customers do not need handle software upgrades and patches. (NIST Special Publication 500-291 Version 2./2013; Sharma et al., 2012, p. 421; Profiz Business Solutions Oyj, 2013)

Platform as a Service (PaaS) provides the possibility to purchase server maintenance and data centres as a service. If the customer does not want to own and maintain servers and data centres by themselves, this can be purchased as a service. This means that hardware and software tools are provided to consumer onto the cloud infrastructure. Applications which are deployed to cloud can be consumer created or acquired. The provider is supporting and providing additional resources in terms of programming languages, libraries, services and tools. The consumer carries the control over the deployed applications and pays service fee for the application server maintenance, but provider is responsible of managing and controlling the cloud infrastructure for the network, servers, operating systems and storage. (NIST Special Publication 500-291 Version 2./2013; Visma Software Oy, 2019)

Infrastructure as a Service (IaaS) is the capability to provide the user with services covering processing, storage, networks, and other fundamental computing resources. I.e. service provider is hosting the infrastructure components like virtual data-centre and thus customer does not need to have servers, storage and networking hardware on-premise. However, customer has the control over the operating systems, storage, and deployed applications. (NIST Special Publication 500-291 Version 2./2013; Rouse, 2018)

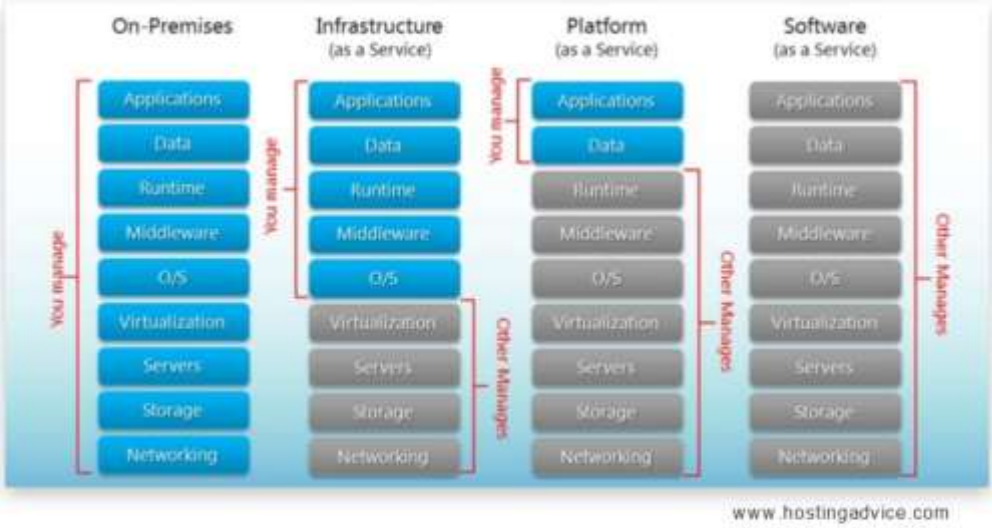

Figure 2. Breakdown of responsibilities in different delivery models (Bernheim, 2017)

2.2.4 Cloud deployment models

Cloud solutions can also be deployed with different methods. There are four different cloud deployment models existing, which are shortly explained below. (listing based mainly on NIST Special Publication 500-291 Version 2./2013)

In the private cloud model, the cloud infrastructure is used by a single tenant organization either in on or off-premise model. Responsibility split can be agreed with different options in private cloud model. Ownership, managing and operating the cloud can be either in the responsibility of the organization itself, a third party or combination of these. Community cloud is community specific e.g. it is for specific group of users with shared interests or concerns. Ownership lies in one or more of the community organizations or a third party (could be also combination of them).

Public cloud means that it is made available for the general public and is enabled from the premises of cloud provider. The cloud infrastructure can be owned, managed and operated by e.g. by a business or government organization. Hybrid cloud combines two or more cloud infrastructures (private, community or public) to unique entity. However, capability for data and application portability is needed to be enabled by the technology.

Benefits and advantages of cloud based services are seen in improved value creation via lower cost, faster time to market and new sources of value (Duan & al., 2012). However, concerns have been around the security aspects of cloud delivery model as companies need to place sensitive data outside the company firewall. From data security point of view traditional models with IT systems and security managed centrally has seen beneficial as data was protected behind the firewalls in a private network. Cloud computing is changing this, as data is stored in clouds, might be transported over open networks and can be managed in multiple cloud applications. Therefore, cloud computing also raises severe challenges especially regarding the security level required for the secure use of services provided by it. (Sharma et. al., 2012, p. 1)

2.3 **ERP architecture**

Architecture of the ERP system is a critical factor impacting on enterprise's successful ERP implementation. Currently there are four dominant ERP architectures for ERP systems (Figure 3):

1-Tier architecture means that a single-instance ERP system were used. This model was costly and too slow to change according to constant change of organizations and business models. 2-Tier architecture was introduced which was more cost efficient, had higher capability in processing and direct communication. This is the client-server architecture approach in which processing is divided between the presentation logic called Client and the Server which is the processing and storage logic layer. (Bahssas, AlBar & Hoque, 2015, p. 73-74)

3-Tier architecture is further separating the layers to three different tiers where the client is not communicating directly with the database but instead business logic tier is the interface in between. In the 3-tier architecture model the client tier is the presentation layer that is responsible for browsing the data and providing a user-friendly interface. In the business logic tier; application layer is responsible for the application logic and business rules. In application layer the data gets retrieved and transferred to the database servers which are in the 3rd tier called database layer. Benefits in 3-tier architecture are in better scalability, reliability, flexibility and easier implementation of reusable components. The drawbacks are in more complex and costly design. (Habadi, Samih, Almehdar & Aljedani, 2017, p.1)

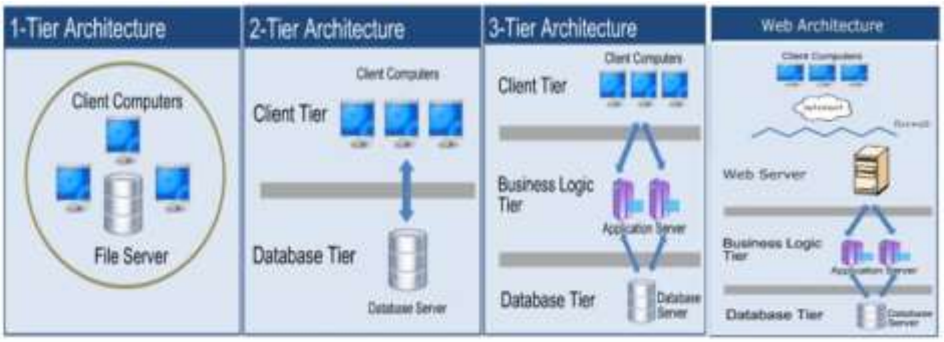

Figure 3. N-tier architecture models (modified). (Habadi et. al., 2017)

ERP systems require the database tier, application tier and user tier to have the system up and running and thus fulfilling the need of integrated system within the company. However, these models have been further enhanced to the increased need to integrate users via web services to ERP. This is called Web Architecture which is also enabling remote access to ERP system. (Habadi et. al., 2017, 2)

In web architecture model application layer as well as database layer follow the traditional 3-tier model, but the presentation layer is split to web services and web browser part. This has been quite a typical evolution path for enterprises because in the beginning it was sufficient to build architecture to support internal use only, later on use over the internet was needed and architecture enhanced to web architecture model. (Habadi et. al., 2017, 2) Web-based architecture utilizes technology (Web-based Object Oriented Model (WOOM)) that allows system-to-system integration. It also improves data integrity, enables easier modifications and has higher flexibility. Web service layer, which is integrating web browser with ERP applications and existing systems, is improving the performance. However, system and internet security are seen as drawbacks. (Bahssas et. al., 2015, p. 73-74) Also recent ERP cloud models drive organizations towards models where both the database as well as the application tier is migrated to cloud environment (Elragal, 2014, p. 246).

2.4 **Odoo ERP**

Odoo is an open-source software suite which is based on modular structure of different business application software that are customizable and fully integrated with each other. Odoo is an ERP system integrating business software such-as CRM, website/e-commerce, billing, accounting, manufacturing, warehouse, inventory and project management to mention a few Modularity enables companies to add needed modules gradually based on the need and grow the system step-by-step and Odoo is targeted to companies of all sizes and budgets. (Odoo a., n.d.)

There are three different Odoo editions available: Community, Enterprise, and Online. The Community edition is open source free-of-charge software which can be downloaded via the Odoo website. The Enterprise edition offers more additional proprietary features and services in on-premise model and that is with annual licence cost. The Online edition is the cloud hosted version of the Enterprise edition based on monthly prescribed fee. (Odoo a., n.d.)

The Odoo market offers numerous modules and apps that suitable for a variety of business needs (Figure 4). In addition, functionalities can be extended with 3rd party application modules. Amount of 3rd party applications is nearly 22 000 in the official Odoo apps web page. Some of these apps are officially validated by Odoo, whereas others developed by the community are dedicated to specific versions for specific needs. (Odoo a., n.d.)

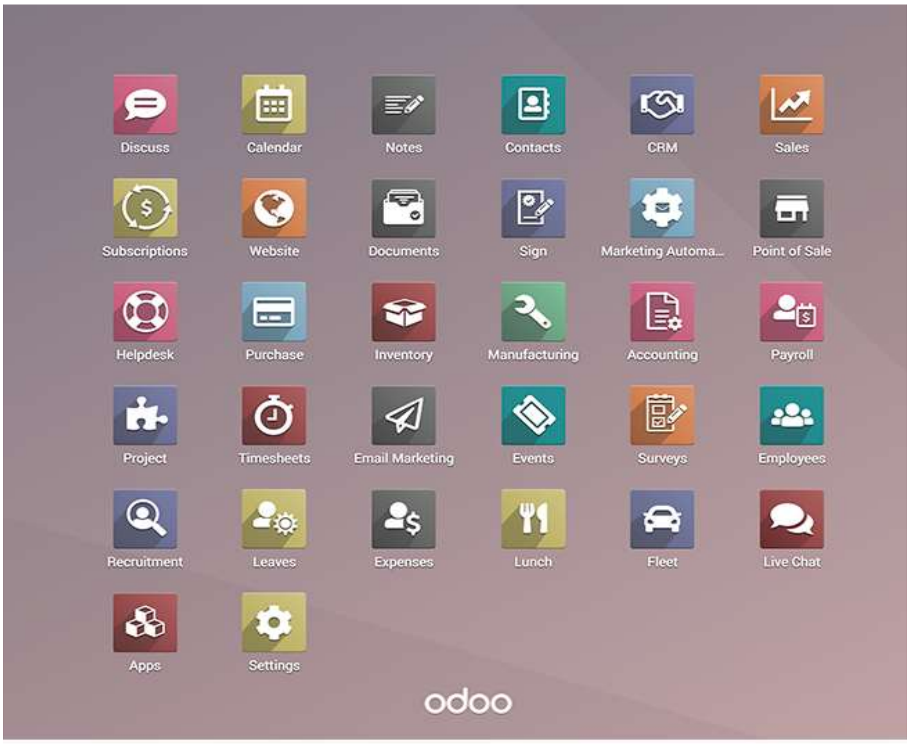

Figure 4.	Example of Odoo-modules (Odoo a., n.d.).

2.4.1 Odoo architecture

Architectural structure of Odoo is based on three main elements: data tier, logic tier and presentation tier which are also presented in Figure 5. (Reis, 2018.)

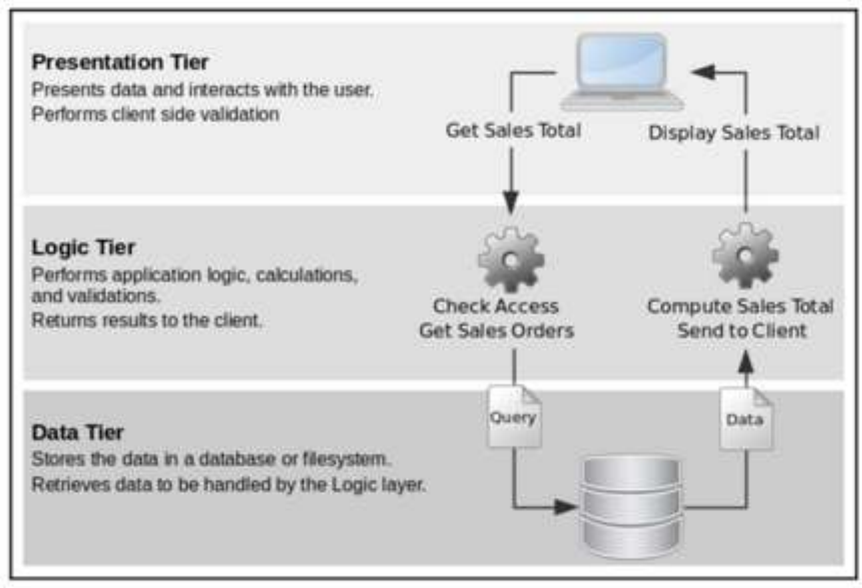
Presentation Tier
Presents data and interacts with the user.
Performs client side validation
Get Sales Total
Display Sales Total
Logic Tier
Performs application logic, calculations, and validations.
Returns results to the client.
Check Access
Get Sales Orders
Compute Sales Total
Send to Client
Data Tier
Stores the data in a database or filesystem.
Retrieves data to be handled by the Logic layer.
Query
Data

Figure 5. The Odoo architecture (Reis, 2018).

Lowest level is the Data Tier which is the data storage layer containing all databased for data storing. Data tier is using PostgreSQL server, which is the only supported database server. Odoo is a multi-tenant system which means that a single Odoo system may run and serve a number of database instances. Logic Tier is the layer interacting with data layer. (Reis, 2018)

Logic tier is the application server containing the enterprise logic and processing layer. This logic tier is on Odoo application server which should be the only access point to lowest level data tier. This is to ensure security access control and data consistency. Odoo server contains the object-relational mapping (ORM) engine which is used for interfacing the data layer. ORM also provides the API (application programming interface) for interaction with additional add-on modules. (Reis, 2018)

Presentation tier is the web client providing the user interface and enables the connections to the system; e.g. containing user login session, navigation menus, forms etc. (Reis, 2018.)

2.4.2 Odoo security

From Data security point of view Odoo provides two main data-driven mechanisms manage or restrict access to data. This is based on the usage on user groups. User is assigned to specific groups and the security mechanisms are applied on the user group level. (Odoo a., n.d.)

Odoo software security is based on continuous observation and contribution of Odoo community users worldwide. As Odoo is an open source software, the code base is accessible by the whole community that is reporting the bugs and also act as source of feedback on Odoo security issues. Odoo developers are encouraged to continuously audit the code and report security issues found. (Odoo a., n.d.)

Odoo design considers the most common security vulnerabilities preventing SQL injections, XSS attacks and preventing RPC access to private methods. Additionally, there are regular audits and penetration tests by independent companies hired by Odoo customers and prospects. In case faults reported, corrective measures are taken by the Odoo Security Team. (Odoo a., n.d.)

Odoo’s security aspects have been much discussed, partly due to its open source nature. Additional security risk might be seen in the 3rd party add-on modules as those are developed by the open source community and there are thus no certification proc*e*dure*s* to validate their security issues. This may result in a compromise of and creating risk to all connected systems. There are also arguments that due to the nature of open source these applications could be more secure than proprietary ones. (Odoo b., n.d.)

One of the listed security advantages is that open source community provides a large group for finding and fixing the problems. Another advantage is the fast fixing of the found problems. With licenced applications there is a typically waiting time for vendors to respond. The code problems can be raised to open source community to investigate and fix it fast. (Korolov, 2018)

Sometimes cloud based ERPs are considered being even more secure than traditional on-premise installations due to cloud providers having more extensive security measures and resources available than single organisations. Especially private clouds which are not shared with other tenants are considered to be safer than public clouds. (Panorama Consulting Group, 2019; Odoo b., n.d.)

2.4.3 PostgreSQL Database server

As mentioned above Odoo is using PostgreSQL server, which is the only supported database server. Database security is outlined in chapter 3.2. Here also PostgreSQL security aspects are studied to see what options that is providing.

PostgreSQL database is a general purpose, object-relational database server that is with free open source user licence. It underlines extensibility, creativity and compatibility. There is also a dedicated and active open source community developing new solutions and contributing to best practices and new feature requests. It is DBMS which runs on most operating systems such as Windows, MAC and Linux. Its users can be from different sectors such as government agencies, public and private sectors. (Juba, Vannahme, & Volkov, 2015, p. 31-36)

PostgreSQL system architecture is based on a client/server model. A PostgreSQL session consists of the below mentioned co-operative processes: Server process which is about management of database files; client applications are connecting to the database which server process is accepting. It is also performing database actions on behalf of the clients. Server program (postgres) which is about frontend applications (e.g. web server, graphical application etc.) that connect to database operations. (Juba, Vannahme, & Volkov, 2015, p. 31-36)

From security point of view when concentrating on limiting access to the database and to the data there are general recommendations existing. Database security should be integrated with enterprise level authentication and authorization models. Preventing unauthorized access by front-end application should not be solely relied on but instead it is important to have consistency and not to look the accesses on silos. (EnterpriseDB, 2016)

Users should have minimum access granted, not such access that is not essential for work performance. Only administrators should have access to configuration files and to log files. Users must have their own login. Superuser roles should be disabled to have host system login accesses. Additionally, the recommendations to keep the database patched, regular back-ups and tested recovery plan are also suggested. (EnterpriseDB, 2016)

3 INFORMATION SECURITY

Information security defined by NIST (National Institute of Standards and Technology) is about “The protection of information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction in order to provide confidentiality, integrity, and availability.”(NIST SP 800-12 Rev. 1. /2017, p. 2)

There are many security areas that need to be considered to ensure comprehensive assurance of company assets. These processes and technologies are physical security, data security, systems and network security, business communications security, wireless security, web and application security, security policies and procedures and security employee training and awareness. (Vacca, 2013)

Underlying basic concepts that form the information security are the concepts of confidentiality, integrity and availability a.k.a. CIA-triad (Figure 6.) CIA-triad model guides the way how to think about and discuss security concepts related to security on data. (Andress, 2014, p. 6-7) Confidentiality of data means that data can accessed only by those users who are authorized to use it. Data integrity refers to modification or destruction of data by those who are entitled to do that. It concerns data in storage, during processing, and while in transit. Data availability ensures that data and information systems can be used only by the entitled users. (Traficom, 2019; NIST SP 800-12 Rev. 1. /2017)

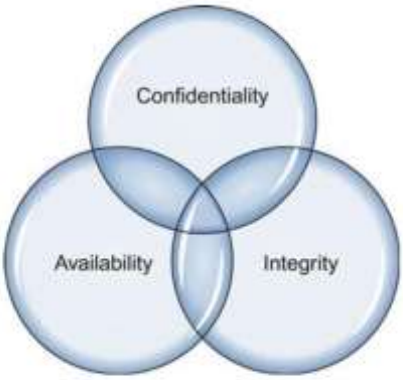

Figure 6. CIA-triad. (Andress, 2014, 6-7)

Later, this model has been enhanced with additional critical characteristics as C.I.A. triad model has been seen non adequate to respond to the constantly changing environment of computer science. Additional critical characteristics are Accuracy, Authenticity, Utility and Possession. (Whitman & Mattord, 2009, p. 8-13)

Information accuracy is error free information; it has the value that is expected, it has not been intentionally or unintentionally modified. Authenticity of information refers to information being original and genuine, not a reproduction or fabrication. Authenticity is when information is in the same state when it was created, placed, stored or transferred. Utility is about information serving a purpose. It should have value from some purpose e.g. it needs to be useful. Possession of information is control or ownership of some item or object. (Whitman & Mattord, 2009, p. 8-13)

3.1 Application security

According to WMware “Application security describes security measures at the application level that aim to prevent data or code within the app from being stolen or hijacked (VMware, n.d.)” It contains not only the security considerations during application development and design phase, but also considers systems and approaches after deployment in order to protect applications against different kind of threats. (VMware, n.d.)

Importance of application security is very high because applications often are available over various networks or connected to the cloud which is increasing the vulnerabilities. Security needs to be also ensured within applications not just at network level. Application security consists of authentication, authorization, application security testing, encryption, and logging. (VMware, n.d.)

Authentication defined that only authorized users can have access and authentication procedures ensure that a user is who they claim to be. This can be accomplished user specific usernames and passwords. In case of Multi-factor authentication then requires more than one form of authentication is required (could be e.g. something you have (a mobile device) or a thumb print or facial recognition). After authentication users need authorized access to use the system. This means system validation of user’s permission to access the application. (VMware, n.d.)

Encryption of the data protects sensitive data from being seen or even used by cyber attackers. In case of a security breach in an application, logging helps to report with timestamps who got access to the data and how. Application security testing is a continuous process to test and ensure that all of these security controls work properly. (VMware, n.d.)

Web application deployment generally is built with 3-tier logic, see Figure 7, (web, application and database). Sometimes web and application tier could be combined to a single tier which already deviates from the classical security architecture practise. This is due to the confusion on where this hybrid web and application instance should be positioned in the DMZ. (Woody, 2013)

When considering full-featured web application it might require access to enterprise core data which is stored in internal databases and that would require access permissions for the web users. Typically, this is implemented with approach where presentation and logic tiers are within DMZ infrastructure, but backend data is located in the internal network. This database relationship makes web applications target of the exploitation. (Woody, 2013)

Alternatively, it could be possible to implement the hybrid solution to web tier and create connections directly to database tier in the core internal network. However, this is not preferred alternative from security point of view. Also, hybrid instance could be placed in the application tier, but also in this approach the security is compromised as business partners and internet access is directly to application tier. It is with firewalls and segmented tiers that needed security layers and measures can be ensured. (Woody, 2013)

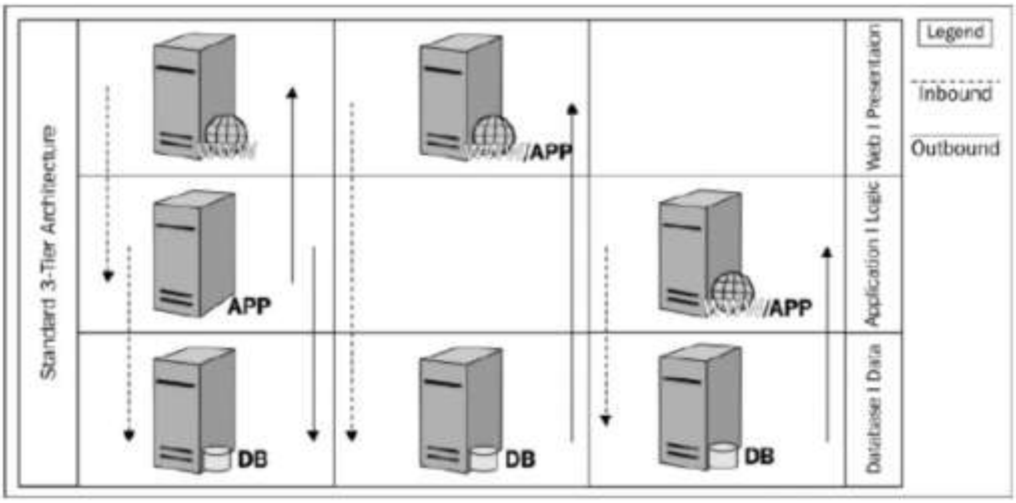

Figure 7. Architecture considerations with web applications from security point of view (Woody, 2013 p. 46.)

Woody (2013) also notes the dilemma of granting the access for business partners to the enterprise data. He says that many times it is not possible to deny the access to enterprise data by externals as it may be needed for business-critical functions or to provide a service enabling function with the business partners accessibility. The problem is that companies might be lacking the methods for data feeds to multiple destinations and thus data accessibility is being served to business partners to the most critical systems and network segments.

3.2 **Database security**

Database can be defined as: “a collection of data, arranged in some meaningful way” (Gollman, 2011, 156). Database management system (DBMS) enables users to retrieve the data organized by the DBMS system. Because database is not only about storing the data but also it is providing information to its users, it is required to have mechanisms to the controlled way or retrieving information in addition to protect the data itself. (Gollman, 2011, 155 - 156)

Databases hold information e.g. about individuals with employee records with bank details and personal information or customer data with payment information. Sensitive data may be protected by law with privacy legislation and therefore database security has an important role in within security considerations. (Gollman, 2011, 156 - 157)

3.2.1 Database threats and security

Database threat can be any situation or event that can cause damage, either intentional or incidental. This damage can cause and adverse effect on the database structure. Possible threats to database security can be e.g. theft and fraudulent, loss of confidentiality or secrecy, loss of data privacy, integrity and availability caused by excessive or unauthorized privileges, operating system vulnerabilities, SQL injection, malware, inappropriate passwords or incorrect implementation of the database (w3schools, n.d.; Ali & Afzal, 2017)

Database security is about protecting and securing the database against intentional or accidental threats with different techniques. Database management system has database specific security requirements. These requirements are different from security requirements on operating system level. DBMS security together with operating system security mechanisms can enforce the security. In case operating system does not have adequate controls in place or it is not feasible for other reasons to have operating system involved, then DBMS can be the tool for defining the security controls in the application layer. On operating system level data is manged without considering the content of it. This data management is about users with defined access controls being able to perform activities such as creation, deletion, reading or writing, depending on permissions granted for them. On database level the content of data, information starts to play a role. Sensitive information might need to be restricted and requires rules who can access and what kind of information. (Gollman, 2011, 155 - 158)

3.2.2 Database security best practises

Next also database security best practises are shortly visited and described to get an understanding how the database itself can be better safeguarded against attackers and threats.

Database should be in a separate database server secured behind the firewall. It should not be in the DMZ zone with the webserver because this is making it easier for attackers to get access to database if they manage to get the administrator accesses to web server. Using web application firewalls improves also database security. Application firewall protects not only against cross-site scripting vulnerabilities and web site vandalism but also against SQL injections thus helping to safeguard the sensitive data in database. (Applicure Technologies, n.d)

It is extremely important to keep the latest patches updated. Outdated versions can be targeted to exploit attempts. Keeping patches up-to-date plays significant role especially e.g. web sites that contain third-party applications, widgets, components and various other plug-ins and add-ons, as all of them should be kept updated. (Applicure Technologies, n.d)

Encryption of the stored files protects the sensitive data because the stored files of web applications can contain database information to which the software needs to connect to. Encryption thus protects the data from attackers. Also, encryption of backup data needs to be ensured to protect data also from internal threats. (Applicure Technologies, n.d)

It is also advised not to use shared web server if database is holding sensitive information. If, however, hosting provider services with shared server are used, it is advised to review their security policies and agree on roles and responsibilities to secure the sensitive data. Usage of 3rd party applications should be also carefully considered because those might create additional risk to database as well. Threats coming with 3rd party apps are in the possible design faults as the quality of these applications might vary a lot, thus these could have design faults and discontinued support with them in the long run. (Applicure Technologies, n.d)

In database access control maintenance default user password must be changed upon database installation, unused user accounts should be locked or removed. Also, all public accounts should be used as well as public access from all accounts. Stronger password policies should be used. Proper authentication method needs to be chosen, either domain authentication or database authentication. User permissions, roles and user groups need to be carefully defined and regularly reviewed to ensure that only the needed authorization is provided. Database administrative functions must be kept with administrators only, not to be delegated to users. (Lane, 2009)

Database configuration should be assessed to determine security and operational integrity. Database configuration needs to be known, understood and approved configuration baseline should be also documented. Unnecessary modules and services should be removed. (Lane, 2009)

Database-platform interaction assessment is important to identify potential security gaps in this area. All databases have functions to directly call operating system commands for administrative tasks which is offering a bidirectional portal to the database. Therefore, it is recommended e.g. to make sure that domain administrators are not also database administrators and externally stored procedures are disabled. (Lane, 2009)

For secure communications encryption needs to be used for communication between applications and database, especially with web application connections. Default database port numbers must be changed to non-default. Ad-hoc connections should be blocked by implementing login triggers, database firewalls and access control systems. (Lane, 2009)

There should be a segmentation of users when viewing application usage of the database. Common users should be separated from application administration accounts. Application processing should be divided in different groups and have different database user accounts assigned to them. Application-to-database connection should allow only such queries which has associated end user. This helps the audit analysis and policy enforcement. (Lane, 2009)

Because protection of sensitive data is typically also a governmental obligation, the most common prescribed regulatory requirements to be established are auditing and encryption. Auditing database is a mechanism to capture a record of database transactions. This can be used to detect suspect activity. Logging and event review help to identify system probing. Reports together with log files also help to demonstrate the compliance in practice. (Lane, 2009)

Finally, despite of all the security measures in place there is still the risk to fail. Companies should have an up-to-date inventory of the databases, sensitive data identified and catalogued and to have disaster recovery plan in place.

3.3 ERP security

In the previous chapter application and database security have been introduced on general level. In this chapter security is viewed especially from ERP point of view and what aspects are raised about that in the literature.

ERP systems host data that is the core, business critical data of a company. These systems also act as corporate data hubs having connections with other systems, customers, suppliers and mobile workers etc. This is also a reason why ERP systems can be tempting target to cyberattacks. (Kowalke, 2017; Rindasu, 2018).

As an example of one ERP application Pascu (2013, p. 37) is listing the most often encountered risks for Oracle E-Business Suite implementations (in the order of importance): Default database passwords, default applications passwords, external application access, database direct access, poor application security design, incomplete patching and update procedures, no encryption on sensitive data, no change management procedures, no database or applications audit, poor password control.

Traditionally ERP has been secured with access control and logging methods (She & Thuraisingham, 2007, p. 152.) Jhawar, Nirwal & Shivhare (2013) explain the concept of the Role-Based Access Control (RBAC) approach. This is a widely used concept in ERPs with some vendor specific variations. An RBAC model consists of different components described in Table 1.

Table 1. RBAC-model components (Jhawar et. al., 2013)

RBAC-model components	
Components:	Description:
Roles	A role is a defined job function within an organization, possibly including hierarchical set-up.
Permissions	Permission is the granted access to subject one or more objects in the ERP system.
Users	A user is a person who is assigned one or more roles.
Constraints	Ability of the senior administrator to restrict the junior's right to grant or deny the permissions.

Also, Panwar et. al. (2016) list typical ERP security aspects which are summarized in the Table 2. These security aspects should be controlled with specific controls per area. Here are some examples of the controls. In the ERP systems one of the biggest vulnerabilities is excessive access rights granted for users. Controls need to be in place to follow if users are able to perform activities to which they should not have rights. Also, actions done to database or its objects need to be analysed and controlled in such way that there is continuous monitoring and detecting illegal activities. (Panwar et. al. (2016)

Table 2. ERP Security aspects (Panwar et. al, 2016.)

ERP SECURITY ASPECTS	
1. Security policy and administrator	Policies maintained and defining the rules for the access of subject to object including constraints on administrators granting the access rights.
2. User authentication	Confirmation of a user's authenticity
3. Separation of duties	Tasks classified and assigned only to certain users or roles
4. Authorization	Verification of the user's access to relevant resources. Granting access based on authorization rules.
5. Time restriction	Access is valid only for certain time period
6. Log and trace	Prevention of log files from breach.
7. Database security	

System logging analysis controls provide data of both successful and unsuccessful login activities. This control helps to prevent a brute-force attack but can also reveal e.g. former employee's login attempts to the system in case user profile is not yet disabled in the system. (Rindasu, 2018)

Qin & Wei (2013, p. 617) consider that the risk in the three-tier model from security point of view is the openness of business logic of the system to the client. Bahssas et. al. (2015, p. 76) mention that the introduction of web-ERP; mobile and cloud has impacted to a complex configuration of security issues. ERP is mainly offering security inside the company. ERP architecture contains different vulnerability areas from network level to application level. Additional security measures are needed to reach compatible security. Maheshwari & Sharma (2014, p. 41) are concerned about increased data security risk that have dramatically increased but at the same this business practises to protect the data has not developed with same speed.

In ERPs, the application layer security needs large efforts in order to effectively secure the business processes and data. Also, the decisions on database security need to made if to activate or deactivate the security functions provided by the database vendor. (Jhawar et. al., 2013)

Database is one critical factor in the ERP system vulnerability protection. If database security measures are not well implemented that is causing most of the issues. Inconsistencies at the database level could be endangering the fundamentals of data security: privacy, integrity and availability. On the application logic layer security deployment also needs to be in place to prevent to data security issues occurring. These can occur in the form of brute force or SQL injection attacks. Database security measures also impact to this level because if database security not well established, attacks will get through. (Rindasu, 2018)

Common security principles for ERP databases are restriction of users not to have direct access to ERP database. Database access must be built via ERP program components (application services, client applications) using secured database connections. Administrator access needs to be limited and controlled; for database maintenance and administrative tasks only limited highly privileged accounts activated. Usage of them needs to be controlled and audited based on group policies. Technical capabilities and sufficient resources need to be ensured onto dedicated database instances hosting the databases (e.g. enough storage space, performance capacity, RAM volume etc.) Creation of specific service accounts with additional security settings is required to run database services. (Antonova & Georgiev, 2019)

For the database security in the ERPs a practical example is e.g. if access passwords are stored without encryption in the ERP application database it might enable attackers to obtain users credentials. If SQL Server databases is used, then it is possible to store encrypted user authentication data in another database, hided even from the database administrator. One of the most significant database weaknesses is the SQL injection which is used for data stealing. The importance of the human factor is being one of most important components of data security. Employees need to be trained and instructed about the impact of security incidents. (Rindasu, 2018)

Additional challenge comes with the ERP customizations. The critical nature of the ERP systems with valuable and confidential data combined with the customized solutions and laborious implementations is causing challenges to IT administrator in terms of security. Even though ERP systems are developed with best practice targets sometimes the business specific requirements are so unique that ERP is implemented with customized and alternative solutions. (Pascu, 2013, p. 36-37)

Rindasu (2018) highlights that because of the complex nature of ERP systems, there is no common security framework applicable for all companies using ERP. Instead, to protect sensitive data in ERP applications is to identify risks by using best practices correctly. She also points out that in order to analyse the vulnerabilities of ERP application, the architectural set-up and functionalities of ERP applications must be understood.

Panwar et. al (2016) note that most of the ERP systems are considered as closed environments. However, current trends already change ERPs to be open systems and thus the security solutions also need to be updated to respond to current trends. They also say that security mechanisms of ERP systems are not yet openly discussed even though research work is ongoing on this area.

4 CASE STUDY –ARHCITECTURAL OPTIONS FOR A SECURE ODOO IMPLEMENTATION

The actual study part concentrates on outlining different possible options for the case company to consider for the Odoo ERP implementation in the future. There are already some preliminary ideas considered in the case company. These ideas as still in drafting phase and consider option to distribute Odoo to two separate instances for internal and external usage to ensure needed security level.

The focus was to find potential options for the future Odoo set-up when considering the integration of external entities from security point of view. The focus in this study is to get more information on the Odoo deployment options and knowledge to support the final decision making.

4.1 Research method and framework

The focus of the study is on open ERP environment from a small and medium business (SMB) size company point of view. Type of this thesis is literature research. Literature review scope is to introduce basic concepts of Enterprise Resource Systems (ERP); purpose of ERP systems, main benefits and disadvantages of them. Also, different delivery models of ERP systems as well as architecture set-up are viewed. Introduction to the Odoo ERP, system that case company was using, is shortly introduced.

Basics of information security are also briefly visited. As information security is a vast area, focus is on application security and database security because those are in key role when discussing about ERP related security aspects.

After establishing the theory groundwork, the actual case study is presented. First research method and framework are explained, then the commissioner (case company) of the study is introduced. Additionally, this includes explanation how Odoo is currently set-up in the case company.

Next the study focuses that what is the current approach and solution available from Odoo to enable external usage of ERP securely. Also, findings from literature are collected to explain the best practises found. Open source community forums are also used as information source. Finally, example solution as reference model is briefly viewed.

Target is to find additional information to support the decision making in the company for the option proposal to implement the ERP system used into two ERP instances. This research will contribute to add new information aspects to the knowledgebase of open source ERP implementation from security point of view.

4.2 Case Company introduction

The commissioner of this thesis is a company operating in manufacturing industry in Finland in small to medium enterprise category. Its' headquarters is based in Finland and one subsidiary in US. During the past ten years the Company has had a strong focus on R&D operations but recently however, it is now expecting growth by expanding its operations not only in domestic markets but also internationally. Company is currently expanding its footprint globally through distributors and partners.

The commissioner of this thesis has already implemented an open source ERP system called Odoo. Currently ERP is used for company internal purposes only but in the future target is to enable modules and functions which will integrate company's customers, re-sellers and suppliers to the company's ERP.

With Odoo the company operates purchases, manufacturing, inventory management, sales and invoicing functions. Recently also financial accounting module has been implemented to Odoo environment. In the next phase company is planning to enable external usage for CRM and customer portal functions. This would enable real-time communication with customers e.g. customer is able to check on order and invoice statuses, enabling customer invoice approvals etc. Seamless connectivity of re-sellers is required to enable CRM usage to offer them e.g. product related information. Implementation of web-shop module is also on the future roadmap.

4.3 Case company Odoo set-up

ERP has been implemented in an on-premise hosting model where Company has its own virtualized servers hosted by 3rd party service provider. On-premise model was considered more secure option in order not to risk and expose company's sensitive data outside company's firewall. However, in the future there is a need to get external connectivity established so that customers and re-sellers, later on also suppliers will get connected fast and seamlessly to data relevant for them. Utilization of these Odoo functions would require more open access by external stakeholders.

Odoo is implemented in the case Company with one server set-up and Odoo licences are installed to one server only, see Figure 8. One server approach was selected as that was seen fully sufficient from the capacity point of view. Simple set-up is easier to manage and this it is cost efficient. Simple set-up reducing complexity is seen beneficial due to a simplified IT architecture and management effort of only one database.

Currently Odoo is used only internally, by the personnel of the company, behind firewall. Odoo usage also via VPN is possible. Access controls on user groups are applied based on Odoo best practises.

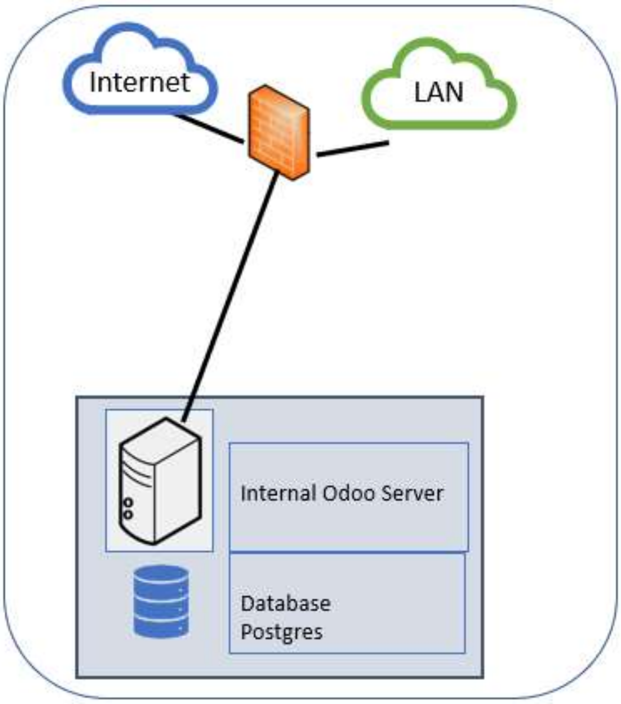

Figure 8. Current Odoo set-up in the case company

Generally, the security level of Odoo is considered to be high enough in many occasions, even with the one server set-up with enablement of VPN usage and a controllable access from the outside. These are the findings from Odoo forum help discussion threads as well as from Odoo partner side. Eventually, it is the company's security policy that defines if the Odoo basic set-up is considered safe enough. Furthermore, if considering web shop implementation for the future, then VPN and controllable access is not a possible control point for external users because web shop is open to all internet users.

When considering this approach with future plan, then especially from security perspective this set-up is not considered feasible anymore. Quite recently the accounting module was also implemented to Odoo which in turn increased the amount of the sensitive data in the system. Adding to that functionalities like CRM, web-shop or other modules requiring external connectivity, risks to external vulnerabilities is increasing as all data utilises the same database server.

4.3.1 Odoo two-tier set-up

The basic principle of ERP is the modular structure which integrates the different applications used in different organizational units to a one integrated platform of software applications. This modular integrated system utilizes central common database, which enables the streamlined processes as the same data can be used by all units in real-time. This serves the purpose well when ERP is used only internally. When new additional functionalities or features are needed usually Odoo is extended internally via modules, see Figure 9. This is very efficient as the ERP modularity can be utilized to its full extent and real benefits of an ERP will realize. At the same time this is the dilemma that how to ensure sensitive data protection because all functionalities are in one and the same ERP system, but also external entities need to be integrated to be part of the same system.

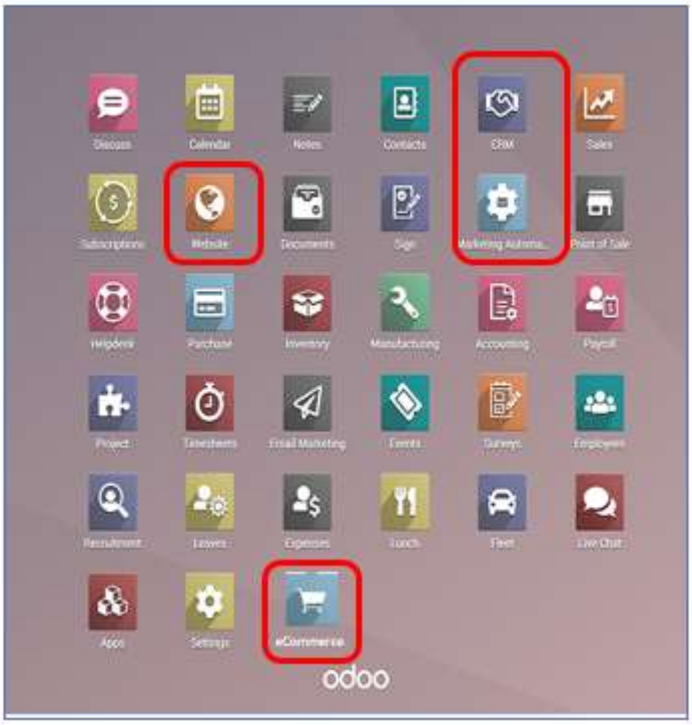

Figure 9. Odoo standard approach to add functionalities by expanding usage to new modules

In Odoo community support it was explained that technically it is possible to separate Odoo to frontend and backend. This means that Odoo codebase/frontend is installed on one server and the backend (PostgreSQL server and database) on another server, Figure 10. It was noted, that this two-tier set-up will bring additional security, but it was mentioned that this would also increase the complexity due to increased tiers. It was mentioned that focus should be put on securing the server and not anymore hide it behind DMZ because current ERPs need to be online and collaborative by nature. From the case company point of view this does not still fully answer the original key question because this approach does not support the required security level to protect the core data of the company.

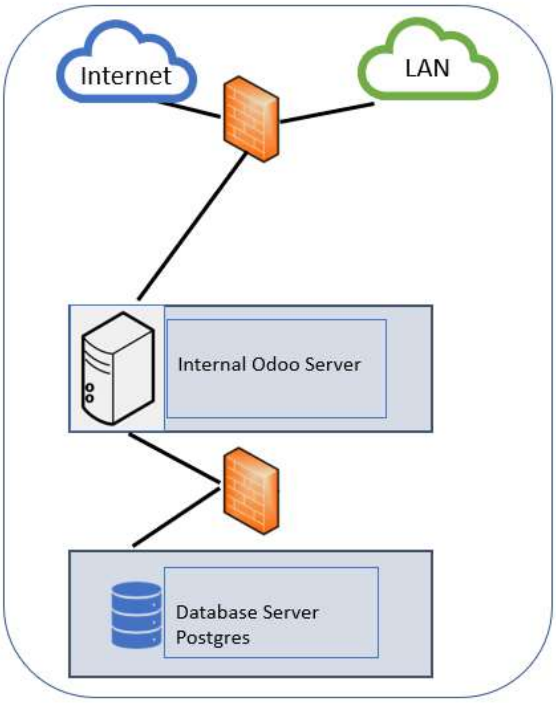

Figure 10. Option to implement 2-tier model in Odoo

If thinking, that 2-tier structure is used, meaning that frontend and backend are separated, but in addition, there would be own instances for internal and external usage (Figure 11.). In this approach application layer would be duplicated to two separate instances but having still the common central database serving both instances Database is placed to a separate server. This would still be 2-tier approach. Here the improved security mechanism comes from separating the internal and external usage to separate application servers, adding the mechanisms of both the application layer and standard database security mechanisms explained in Chapter 3.

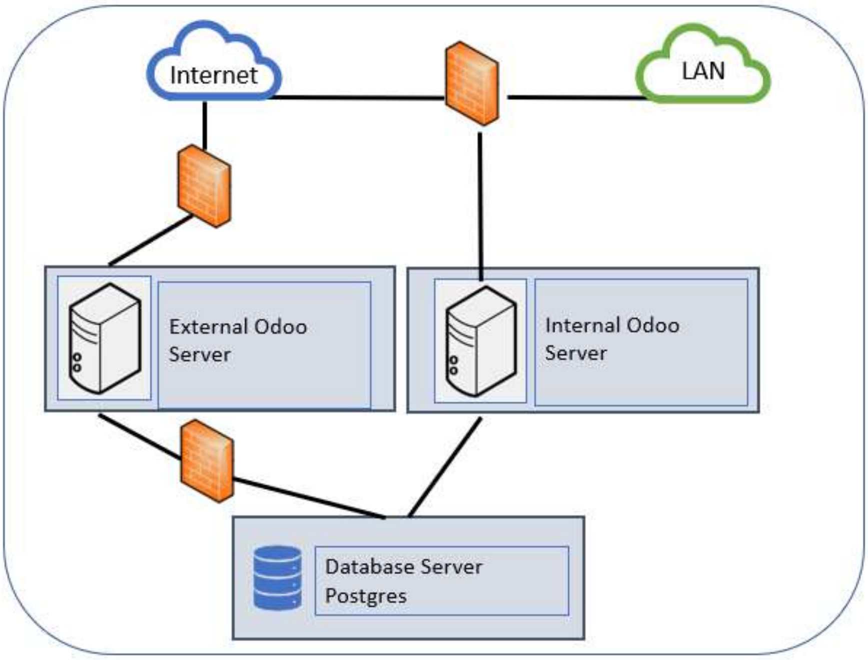

Figure 11. Odoo instance duplication for internal and external usage

This approach could be usable in such cases where the number of modules in external Odoo side is very low. In case more modules would be activated then also the number of attack vectors would increase. Here also the server costs and integration costs would increase due to duplication of Odoo application servers.

In this two tier model the web and application tiers are combined to a single tier. This creates the challenge on where to place this hybrid instance. If placed to web tier, then it is having direct database access to internal network, which is not recommended. Or, if the hybrid instance is placed to second tier, then partner access would be directly to application tier which is also not recommended. Based on the statements from literature review this approach could make the web applications target to exploitation due to the database relationship, which is of interest for attackers. Furthermore, security is compromised because of the internet and business partner access directly to application tier. Duplicating the frontend servers would increase the server costs as well and so this approach would be more expensive and still unsecure.

4.3.2 Odoo Three-tier set-up

Odoo architecture is supporting the three-tier model which is separating the presentation layer, application layer and database layer to different tiers. Based on literature review 3-Tier architecture was seen more secure than client-database architecture model because in the 3-Tier model the client is not communicating directly with the database. There is the business logic tier in between. As per the database security recommendation, database should be in a separate database server secured behind the firewall. Therefore, this 3-Tier architecture model described in Figure 12, could improve the application security level in the case company too. At the same time three-tier set-up will increase server costs and create more complex environment to maintain and manage.

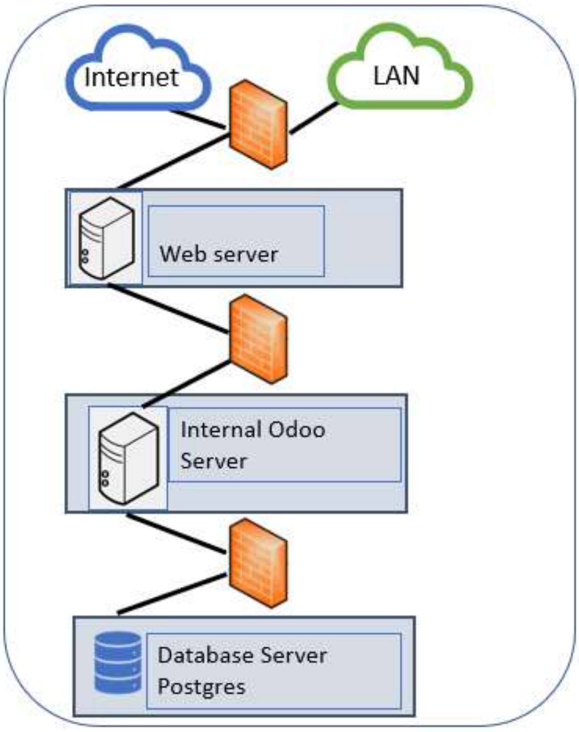

Figure 12. Three-tier implementation approach

In the application security considerations, it was mentioned that it is the database relationship that makes web applications target of the exploitation creating risk to database security. In the literature review, it was also seen as a challenge that how to manage the external users; because business partners would need to have business partner accesses in case companies do not have any other methods to data feeds with external users. Again, with web-shop implementation any users could access the web-shop via internet. From security point of view, this solution is still based on usage of common shared database. This means that then access for business critical functions might not be avoided and risk is still not mitigated even with this solution.

4.3.3 Planned set-up with distributed instances

Because the requirement from the case company is to eventually find a feasible option to separate the internal and external usage of with more powerful methods than defining access rights and user groups. Company has considered an option to implement two Odoo instances for internal and external purposes (Figure 13).

Odoo integration with various software is supported. From Odoo capability points of view, at present Odoo does not provide own Odoo-Odoo connector. Odoo does provide 3rd party connector framework which enables to develop bi-directional connector between Odoo and any other software or service. Odoo connector is an add-on functionality which provides extensibility with additional modules for new features or customizations. Connector can be used for e.g. Odoo E-bay connector, Odoo Amazon connector, Odoo Wordpress connector, Odoo SugarCRM connector. Connectors are also available for Odoo-Odoo connections but provided by 3rd party providers only. (Odoo a., n.d.)

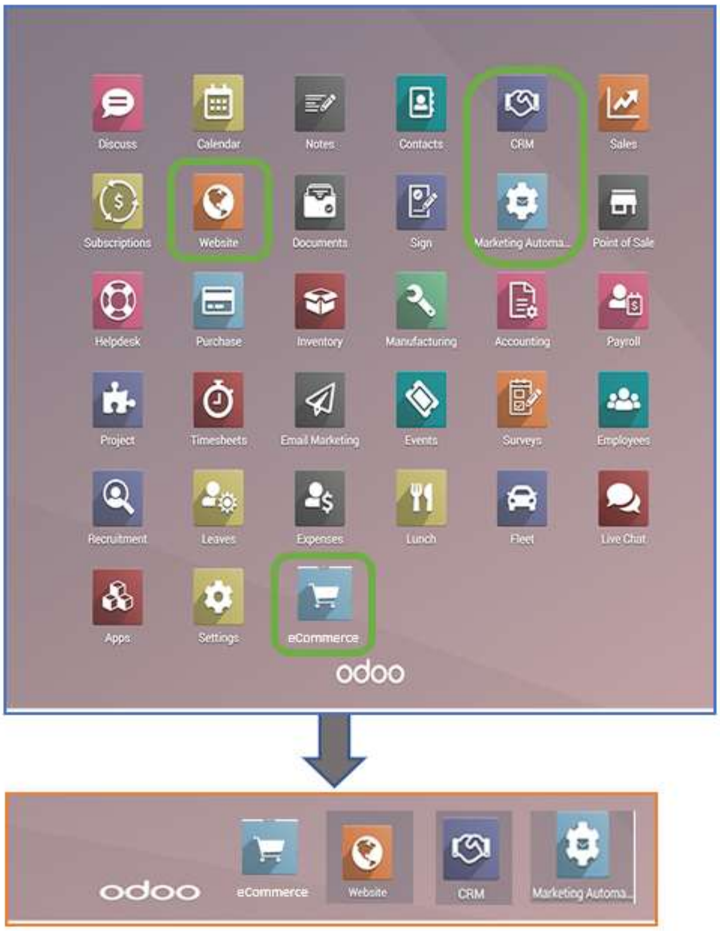

Figure 13. Target stage alternative is to separate Odoo for two separate instances for internal and external use.

Research on Odoo Forum/help community pages indicates that also other companies are facing similar challenges and considerations for the security aspects in Odoo deployment for external usage. Based on the learnings from several discussion threads it seems that Odoo is not designed to run on two different instances, one facing the public, and the other one for internal use only, behind additional security layers. It was stated that the big advantages of integration do outweigh the potential security risks, when Odoo being deployed properly. Solution proposal for users with doubts on Odoo existing approach, was to utilize Odoo Connector framework instead and have Odoo connected to other available online shops. WooCommerce etc. (Odoo b., n.d.; Quora, 2020)

Figure 15 describes the future planned architecture set-up with Odoo for the case company. In this plan Odoo instances would be deployed to separate servers; Odoo server for the public Odoo usage and server for the internal Odoo usage, both instances having own databases. Internal Odoo would be the master instance and external Odoo would be the slave instance. In between the ERP instances there is integration platform which acts as a message bus between the applications taking care of the communication and information exchanges between ERP instances. At first communication would be established only one-way from master to slave, later on communication is planned to be enhanced to bi-directional.

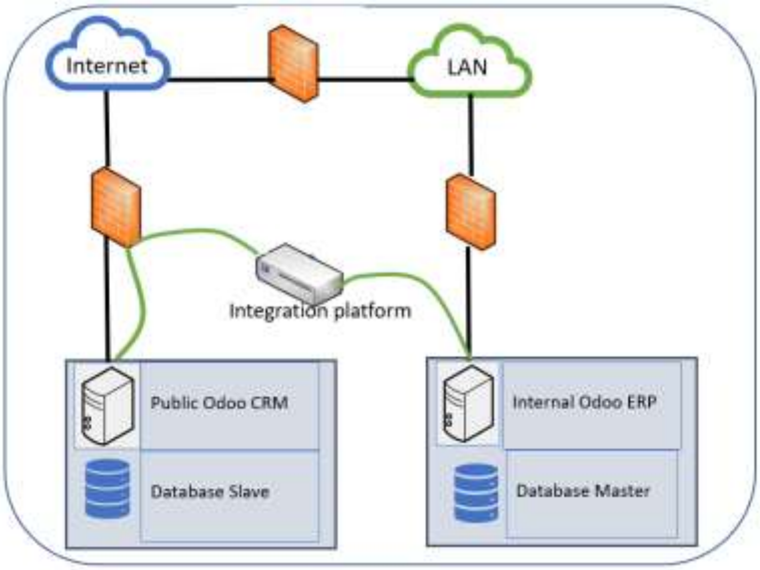

Figure 14. Planned Odoo architecture for high security level (Modified from original source: Contrasec Oy)

This planned approach would enable to separate the internal and external usage thus creating additional needed security level for the company. However, based on these findings there is not a supported capability existing from Odoo to this kind of integration set-up. This will require additional effort from the case company to figure out the architectural solution for it. Respectively data integration between the systems will create more complicated architecture and additional costs due to Odoo instance duplication and building the communication logic in between.

The scope of this study is to consider the solution with two distributed instances. For the future, architectural solution could be scaled up even with three instances, Figure 15. This model would be the enhanced model for the two-instance model with very high security level solution.

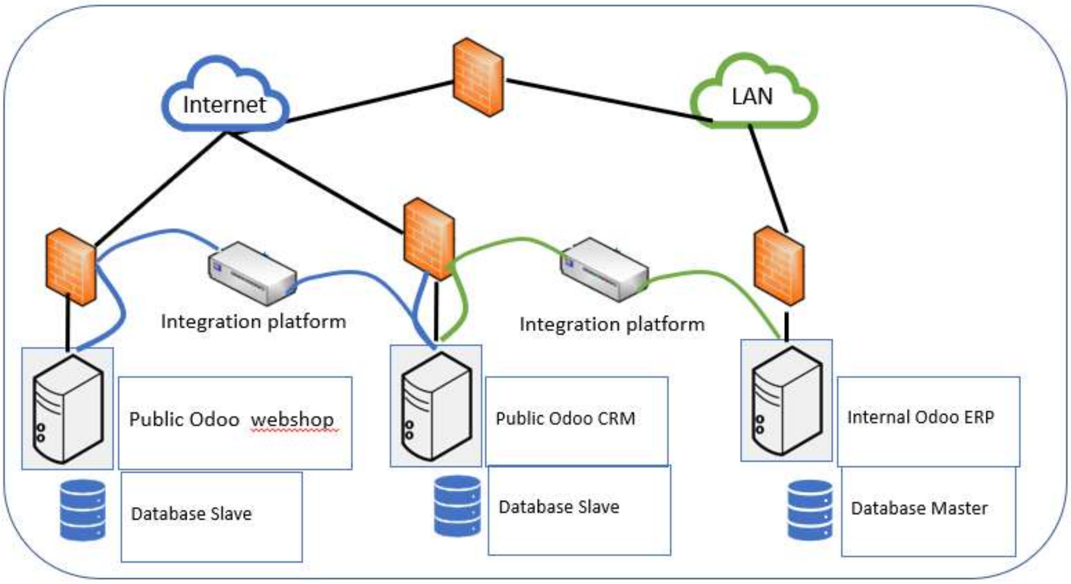

Figure 15. Planned Odoo architecture for very high security level (Modified from original source: Contrasec Oy)

4.4 Reference solution – Aras Innovator

The planned option of the case Company for Odoo instance duplication would help the company in being control what data is being shared from database layer. This approach is not at present is not such that Odoo would have supporting capabilities in place. Possible reference implementations were asked from one of the Odoo partners. Based on the feedback, few distributed Odoo solutions have been built by some companies but those solutions are not shared in public.

But when cross-checking other possible solutions, there is a solution presented by a company Aras, which is aiming to solve the similar problem what the case company is trying to solve in ERP environment. This is presented as a reference case below.

When investigating the solution options provided by Odoo for the external connectivity, such a solution was not found which would support distribution of instances. However, there is an interesting option provided by Aras in the area of product life-cycle management. They are presenting a concept to improve the challenge to secure external connectivity.

In this chapter the reference literature is based on Aras Whitepaper: Making the Connection : The How-To's of Connecting Suppliers, Partners, and Manufacturers. Aras Corporation (Aras) is providing solutions for the engineering, manufacturing and maintenance of complex products. Aras customers are e.g. Airbus, GE, GM Hitachi. Aras platform and product lifecycle applications target to connect the users across the extended enterprise to critical product data. There is a need to exchange data with partners and suppliers on controlled way ensuring that only valid data is being transferred. Process needs to collaborative so that partners and vendors are included in the change process. For this purposes Aras has Aras Innovator platform which is an open-source software used for product lifecycle management (PLM).

Manufacturing industries, e.g. automotive, industrial and electronics, have faced challenges in establishing efficient and secure capabilities in data sharing with their collaboration partners. Manufacturing industry is also on the path of digital transformation and at the same time product complexity has grown massively. Production has been outsourced and there is growing need to improve design collaboration with outsourced manufacturing partners. Current legacy tools and processes have been serving simpler and more mechanically focused era. However, modern product development with even product design being outsourced is setting a new level to requirements to efficient, seamless and digital collaboration processes.

Aras mentions that legacy systems are often unsecure and do not support efficient workflows which has driven companies to develop workaround solutions and home grown tools to run the processes. Still today different manual methods (e-mail, FTP, file sharing services) are being used to share design data, drawings, CAD files. etc.

Aras has presented their approach to secure external access by presenting three dimensional data control layers for partner access (Table 3). Dimensions are data access, data location and client access.

Table 3. Three-dimensional data control layers for partner access (Aras, 2018.)

Dimension	Lighter control	Stronger control
Data access	Standard permissions	Advanced access control
Data location	Single database	Multi-database with synchronization
Client access	Standard web client	Web services portal client

Every company security policy defines the required security levels and thus that guides the decisions to data location. Data Location dimension describes if the data to be accessed by external users is located either inside or outside your company's firewall.

In the option of having only single database inside the company firewall means that external users can directly access the Aras PLM database (Figure16.). This model is very efficient but to ensure needed security protections also the other two dimensions need to be established.

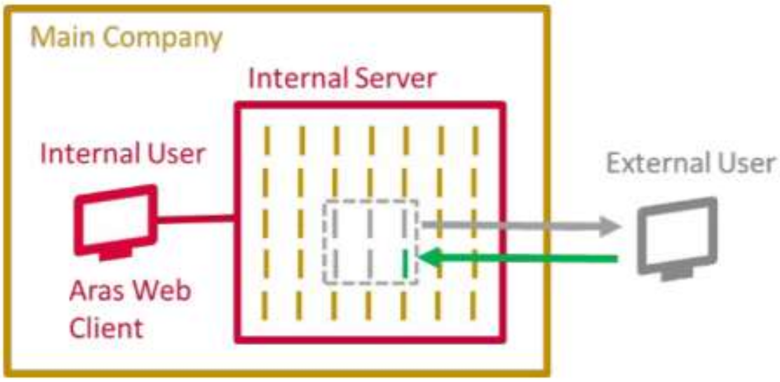

Figure 16. Data location – inside the firewall with single database model (Aras, 2018)

In the multi-database model, a separate Aras server and database instance are placed outside the firewall, either in the DMZ or in the cloud (Figure 17). This model utilizes Aras Data Synchronization Service to copy (selected data synch based on identification of the necessary data) the needed data for external users to external server. Administrator functions are used to identify, submit, and monitor the data being synchronized. Also, many different architecture options are possible if e.g. there is need to separate partners from each other. For example, different combinations having one Aras server and one database or one server and multiple databases (one for each partner), or multiple servers each with multiple databases.

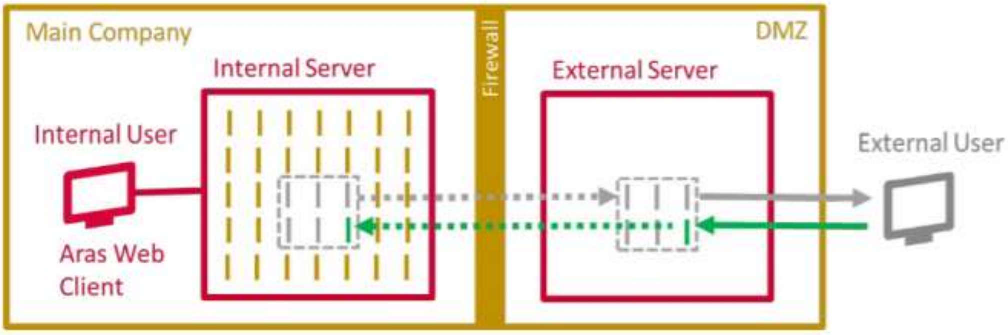

Figure 17. Multi-database model (Aras, 2018)

In the solution approach defined by Aras the security with external connectivity is solved with support for having separate servers and building a synchronization flow in between. This would help in cases where sensitive data is not needed in the external server side and that could reside only in the internal server. However, in case also the external service needs the sensitive data, then this solution does not help either.

5 FINDINGS

ERP systems are facing the challenge of transformation. New business models, real-time information access as well as richer user experiences are capabilities required also from ERP systems. ERP systems are changing in fast pace towards web-based and cloud-based systems. This change is also bringing new players to the market and different kind of ERPs from proprietary to open source ERP systems are competing especially on the SME market.

Globalization, internet and company's networking capabilities require new ERP capabilities. At the same time security aspects of the ERP systems need to be considered. Security in a company needs to be based on security policy which need to be in place, trained to everyone and respected. Auditing, logging and tracing are mechanisms to follow the requirements defined in security policy. ERP system are secured both with application and database security mechanisms such as user authentication, authorization, user groups, and encryption. Generally, the architectural structure of ERPs is seen crucial to respond to security challenges and to prevent vulnerabilities. Also respecting the three tier architecture model is seen more secure than one or two tier approaches. This is also the Odoo supported model.

Odoo has several security mechanisms in place. From Data security point of view Odoo provides data-driven mechanisms manage or restrict access to data based on usage of user groups to which the security mechanisms are applied on. Additionally, Odoo design considers the most common security vulnerabilities preventing SQL injections, XSS attacks and preventing RPC access to private methods. Regular audits and penetration tests are performed by independent companies hired by Odoo customers and prospects. In case faults reported, corrective measures are taken by the Odoo Security Team. Also, Odoo software security is based on continuous observation and contribution of Odoo community users worldwide. Odoo architecture is supporting the three-tier model which is separating the presentation layer, application layer and database layer to different tiers. Three-Tier architecture is more secure than client-database architecture model because in the 3-Tier model the client is not communicating directly with the database.

However, there are also considerations that it is still the database relationship that makes web applications target of the exploitation and thus risking also the database security and company core data. Odoo approach is to rely on its architectural set-up and other traditional security mechanisms. If this is still seen risky in the companies using Odoo, other options need to be considered to integrate external users to be part of ERP.

Odoo integration with various software is supported. But from Odoo capability points of view, at present Odoo does not provide own Odoo-Odoo connector. Odoo does provide 3rd party connector framework which enables to develop bi-directional connector between Odoo and any other 3rd party software or service.

It might not be enough for companies to rely on ERP architectural layer security mechanisms only. New solution might be needed to fulfil the security needs of the companies and provide solutions that support ERP usage also with external connection. For the case company the recommendation is in minimum to consider changing the Odoo ERP architecture to three-tier model and gaining more security from that set-up. This is needed because the amount of sensitive data in ERP has increased by the implementation of the accounting module and because of the intended external usage of the system. This three-tier model is the current, generally recommended practise available. But, the case company has already alternative high security architecture plan under consideration. To ensure better security level, it is worthwhile to continue developing that plan still further. Literature review did not indicate directly similar considerations for ERP security solutions. But on the other hand, ERP security mechanisms are still based on the best practises built for internal ERPs. Changing ERPs to more open systems indicate that new solutions are needed. This is the reason why three-tier architecture might not be enough for the case company either. Reference solution by Aras Innovator is already a step towards this direction. Even though this is not in ERP environment, it is indicating, that in some areas new approaches are developed to enable secure external access to company data. Therefore, with the high security requirements that the case company has, they should continue to evaluate their internal plan in order to secure the core data.

The conclusion is that ERP environment is in turbulent change and based on the literature review, security mechanisms need to be adjusted to respond to new requirements. Especially opening the ERP to external entities seem to be still an open question to be answered comprehensively. Research has been started but best practises for the future are not yet widely established.

6 CONCLUSIONS

This study aimed at investigating how to implement Odoo ERP securely with external connectivity and what are the best practises available to create the external connections in ERP without risking the core data of the company. Research questions were answered based on the literature review findings and literature review revealed that ERP security mechanisms are serving the purpose quite well when ERPs were used only internally. Generally, managing the external users was seen as a challenge in ERP systems. Findings were the same also when investigating the solution in Odoo ERP.

When new additional functionalities or features are needed usually Odoo is extended internally via modules. This is very efficient as the ERP modularity can be utilized to its full extent and real benefits of an ERP will realize. At the same time this is the dilemma that how to ensure sensitive data protection because all functionalities are in one and the same ERP system, but also external entities need to be integrated to be part of the same system.

The three tier architecture model, supported also by Odoo, is seen as more secure than one or two tier models. As companies today are operating over the traditional company borders, secure business partner access to enterprise data is needed. It is not always enough for companies to rely on ERP architectural layer security mechanisms only. New solutions are needed enabling secure ERP usage also with external connections to fulfil the security needs of companies. It seems that at the moment best practise security mechanisms for web ERPs are not yet widely established.

This study was agreed to be focused on literature review. In order to improve the quality of the findings, scope of work for the future research could be improved e.g. with comparison of other ERP vendors security solutions for secure external access and by interviewing ERP providers on planned future capabilities to solve this kind of problems.

I have previous work history with ERP system development but no experience on open source ERP systems or more in-depth security related topics within ERP environment. During my studies in Linux and open source systems I course studied Odoo ERP as one of my course assignment. This assignment evoked my interest to learn more. I managed to get in contact with a company using Odoo software and which was interested in cooperation around this subject. Because I was doing the study project alongside with my main job the timeline was agreed to be very flexible.

Even though I have long experience in ERP development I have previously not concentrated on specifically to ERP security or architecture related topics. I had no previous knowledge on these subject except the basic courses on databases and database design. I had to start from very basics to get understanding on architecture, especially ERP architecture and its impact on security related mechanisms. Also, information security as a subject is such a vast area that defining the scope relevant to my research was also a bit cumbersome. Writing this thesis has required me to step out of my own comfort zone and it proved to be a really good learning experience.

REFERENCES

Ali, A. & Afzal, M. M., (2017). Database Security: Threats and Solutions, pp. 25-27). *International Journal of Engineering Inventions.* (6)2. Cited 6th of March 2020. http://www.ijeijournal.com/papers/Vol.6-Iss.2/D06022527.pdf

Andress, J., (2014). *The basics of information security: Understanding the fundamentals of InfoSec in theory and practice, 2nd edition*. Beaverton: Ringgold Inc. Cited 3rd of April 2020. Retrieved from https://search-proquest-com.ezproxy.hamk.fi/docview/1651928488?accountid=27301

Antonova R., Georgiev G. (2019) ERP Security, Audit and Process Improvement. In: A., Al-Masri, K. Curran, (eds) *Smart Technologies and Innovation for a Sustainable Future. Advances in Science, Technology & Innovation* (IEREK Interdisciplinary Series for Sustainable Development). Springer, Cham. Cited 4th of April 2020. DOIhttps://doi.org/10.1007/978-3-030-01659-3_14

APICS Operations Management Body of Knowledge Framework, Third Edition (2011). Cited 9th of Dec 2019. https://www.apics.org/apics-for-individuals/apics-magazine-home/resources/ombok/apics-ombok-framework-table-of-contents/apics-ombok-framework-8.1

Applicure Technologies. *Database Security Best Practices*. Cited 3rd of April 2020. http://www.applicure.com/blog/database-security-best-practice.

Aras Corporation (2018). Whitepaper: *Making the Connection: The How-To's of Connecting Suppliers, Partners, and Manufacturers*. Increasing Product Complexity Tests the Limits of Legacy Systems. Cited and downloaded 11th of Feb. 2020. https://www.aras.com/en/resources/all/sea-making-the-connection

Aspirion, P. M., Schneider, B. & Grimberg, F. (2018). ERP systems towards digital transformation. In Dornberger R. (ed.) *Business Information Systems and Technology 4.0 : New Trends in the Age of Digital Change.* Cham, Switzerland, Springer, 15-29. Cited 9th of Dec 2019. DOI 10.1007/978-3-319-74322-6.

Bernheim, L. (2017). *IaaS vs. PaaS vs. SaaS Cloud Models (Differences & Examples).* Cited 15th of February 2020. https://www.hostingadvice.com/how-to/iaas-vs-paas-vs-saas/

Bahssas, D. , Albar, A. & Hoque, M. (2015). Enterprise Resource Planning (ERP) Systems: Design, Trends and Deployment. *International Technology Management Review 5(2), 72-81*. ISSN: 2213-7149, Atlantis Press.

Barki, H. & Pinsonneault, A. (2002). *Explaining ERP Implementation Effort and Benefits with Organizational Integration*. Cited 15th of Jan 2020. https://www.researchgate.net/.

Berchet, C. and Habchi, G., (2005). The Implementation of deployment of an ERP system; An industrial case study. Computers in Industry, 56 (6), p (588-605).

Cisco (2009). *Blueprint for Collaborative Application Architecture.* Whitepaper. Cited 6th of March 2020.https://www.cisco.com/c/dam/en/us/solutions/collateral/enterprise/benefit-unified-communications/C11-503429-00-CollaArchit.pdf .

Duan, J., Faker, P., Fesak, A. & Stuart, T. (2012). *Benefits and drawbacks of could-based versus traditional ERP systems.* Proceedings of the 2012-13 course on Advanced Resource Planning. W.J.H. van Groenendaal (ed.)

Elragal, A. (2014). *ERP and Big Data: The Inept Couple*. Cited 21st Feb 2020. DOI: 10.1016/j.protcy.2014.10.089. Published by Elsevier Ltd.

EnterpriseDB. (2016). *Security Best Practices for Postgres*. Cited 3rd of April 2020. https://info.enterprisedb.com/rs/069-ALB-339/images/security-best-practices-for-postgres.pdf .

Eskeli, J., Heinonen, S., Matinmikko, T., Parviainen, P. & Pussinen, P. (2010). *Challenges and Alternative solutions for ERP's*. VTT Technical Research Center of Finland. Research report.

Gartner. Information Technology, Gartner Glossary.
Enterprise Resource Planning (ERP). Cited 28th of March 2020. https://www.gartner.com/en/information-technology/glossary/enterprise-resource-planning-erp

Habadi, A., Samih, Y., Almehdar, K. & Aljedani, E. (2017). An Introduction to ERP Systems: Architecture, Implementation and Impacts. *International Journal of Computer Applications,* 167(9).

Häkkinen, T. (2020). Planned Odoo architecture for very high security level. Contrasec Oy.

Jhawar, M., Nirwal, D. & Shivhare, S. (2013). Modelling Security Concerns in Web Based ERP. *International Journal of Innovative Research and Development*, 2(12).

Juba, S., Vannahme, A., & Volkov, A. (2015). *Learning PostgreSQL*. Packt Publishing. Cited 26th of Apr 20 EBSCOhost Ebook Academic Collection - World Wide.

Kelle, P. & Akbulutb, A. (2005). The role of ERP tools in supply chain information sharing, cooperation, and cost optimization. *International Journal of Production Economics,* 93–94(8), p. 41-52. Cited 15th of Jan 2020. https://doi.org/10.1016/j.ijpe.2004.06.004..

Korolov, M. (2018). *Open source software security challenges persist*. Cited 4th of March 2020. www.csoonline.com/article/3157377/open-source-software-security-challenges-persist.html.

Kowalke, P. (2017) *Six ERP Security Risks to Watch*. Cited 2nd of Feb. 2020. https://it.toolbox.com/blogs/erpdesk/six-erp-security-risks-to-watch-040317

Kraft, P. & Thome, R., (2013) Semantically integrated business applications for enterprise resource planning systems. In: Kurosu, M. (Ed) *Human-Computer Interaction. Users and Contexts of Use*. HCI 2013. Lecture Notes in Computer Science, vol 8006. Springer, Berlin, Heidelberg. Cited 20th of Jan. 2020. DOI https://doi.org/10.1007/978-3-642-39265-8_46.

Lahti, S. & Salminen, T. (2014). *Digitaalinen taloushallinto*. Helsinki: Sanoma-Pro.

Lane, A. (2009). *Basic Database Security: Step by Step*. 10th of Dec 2009. Cited 3rd of April 2020 https://searchsecurity.techtarget.com/magazineContent/Basic-Database-Security-Step-by-Step.

Mattison, J. B. & Raj, S. (2012). Key questions every IT and business executive should ask about cloud computing and ERP. Accenture. Cited: 11 Dec 2019. https://www.itselector.nl/wp-content/uploads/2013/10/Accenture-Key-Questions-Executive-Ask-About-Cloud-Computing-ERP.pdf.

Ng, J. K., & Ip, W. H. (2003). Web-ERP: the new generation of enterprise resources planning. *Journal of Materials Processing Technology*, 138(1), 590-593. Cited 18th of Feb 2020. DOI: 10.1016/S0924-0136(03)00153-5.

NIST SP 800-12 Rev. 1 (2017). *An Introduction to Information Security*. Special Publication 800-12 Revision 1. Cited 23 February 2020. https://csrc.nist.gov/publications/detail/sp/800-12/rev-1/final.

NIST Special Publication 500-291, Version 2. (2013). *NIST Cloud Computing Standards Roadmap*. Cited 12 February 2020. https://www.nist.gov/system/files/documents/itl/cloud/NIST_SP-500-291_Version-2_2013_June18_FINAL.pdf

Odoo a.(2019). https://www.odoo.com/. Cited 9th of December 2019.

Odoo b. (2019). Odoo Community Forum help. https://www.odoo.com/forum. Cited 10th April 2020.

Olson, D. L., Johansson, B. & De Carvalho, R. A. (2018) Open source ERP business model framework. Robotics and Computer-Integrated Manufacturing 50, April 2018, 30-36. Cited 6th of March 2020.

O'Shaughnessy, K. (n.d.). *Benefits of ERP: Advantages and Disadvantages of an Enterprise Resource Planning System*. Cited 1st of March 2020. https://www.selecthub.com/enterprise-resource-planning/erp-advantages-and-disadvantages/

Panorama Consulting Group (2019). *Cloud vs. On-premise ERP Security: The Advantages and Disadvantages.* https://www.panorama-consulting.com/cloud-vs-on-premise-erp-security-the-advantages-and-disadvantages/ Cited 24th of April 2020.

Panwar, R., Kumar, A. & Pandey, N. (2016). Enterprise Resource Planning Systems: Security. *International Journal of Innovative Computer Science & Engineering (IJICSE)*. Bhagwant University, Ajmer (National Conference on recent trends in CSE. 1(1) 2016.

Profiz Business Solutions Oyj, (2013). *ERP toiminnanohjausjärjestelmän ostajan opas PK-yrityksille*. Modified 31st of May 2017, Cited 10th of Feb 2020. https://www.profiz.com/profiz/wp-content/uploads/2017/05/ERP-Ostajan-opas.pdf.

Qin, Y & Wei, J. (2013). *The Solution of Enterprise ERP Based on Six-tier Architecture.* 2013 International Conference on Computational and Information Sciences. IEEE. Cited 22 Feb 2020. DOI 10.1109/ICCIS.2013.169.

Quora. How secure are online ERP systems like Odoo?

A security audit of Odoo - The fastest growing OSS ERP - To raise the standard of business software. Cited 2nd of Feb 2020. https://www.quora.com/How-secure-are-online-ERP-systems-like-Odoo

Reis, D. (2018). *Odoo 12 Development Essentials*. 4th Ed. Packt Publishing. E-book central. Cited 25th of Jan 2020.

Rouse, M. (2018*). Infrastructure as a Service (IaaS).* Cited 26th of February 2020. https://searchcloudcomputing.techtarget.com/definition/Infrastructure-as-a-Service-IaaS

Salim, S. A., Sedera, D., Sawang, S., Alarifi, A.H.E. & Atapattu, M. (2015). Moving from Evaluation to Trial: How do SMEs Start Adopting Cloud ERP? *Australasian Journal of Information Systems*, 19, p 219-254.

Sharma, M., Bansal, H. & Sharma, K. A. (2012). Cloud Computing: Different Approach & Security Challenge. *International Journal of Soft Computing and Engineering*. 2(1), p 421-424).

She, W. & Thuraisingham, B. (2007). Security for Enterprise Resource Planning Systems. *Information Systems Security*, 16, 152–163. Cited 15th of Feb 2020. DOI: 10.1080/10658980701401959.

Traficom, (2019). *Information security*. Finnish Transport and Communication Agency, National Cyber Security Center. Cited 16 Feb 2020.https://www.kyberturvallisuuskeskus.fi/fi/toimintamme/saantely-ja-valvonta/tietoturva Updated 11th of Dec 2019.

Vathanophas, V. (2007) Business process approach towards an inter-organizational enterprise system. *Business Process Management Journal* Vol. 13(3), pp. 433-450. Emerald Group Publishing Limited 1463-7154. Cited 21st of Feb. 2020. DOI 10.1108/14637150710752335.

Visma Software Oy (2019). Pilvi vai on-premise? ERP opas oikean ratkaisun löytämiseen. Cited 1st of Feb 2020. http://images.efficiency.visma.com/Web/Visma/%7Bef588d73-59ae-43ec-b1a1-9089d97e187a%7D_FI_SW_Pilvi_vai_On-Premise_ERP.pdf?utm_source=Eloqua&utm_medium=email&utm_content=FI_SW_Suunta%20-%20Tiedoston%20lataus%20%28V5%29&utm_campaign=&optin=1 Modified 11th of Feb 2019.

VMware Inc (n.d.). *Application Security*. Cited 3rd of April 2020 www.vmware.com/topics/glossary/content/application-security

Whitman, M. E., & Mattord, H. J. (2009). *Principles of Information Security,* 3rd Ed. Boston: Thomson Course Technology.

Woody, A. (2013). *Enterprise Security : A Data-Centric Approach to Securing the Enterprise*. ProQuest E-book Central. Created from Hamk-ebooks on 21st of Feb 2020. http://ebookcentral.proquest.com/lib/hamk-ebooks/detail.action?docID=1103988

W3schools (n.d.). Database security. Cited 28th of March 2020. https://www.w3schools.in/dbms/database-security/