



# POLITECNICO MILANO 1863

Master of Science in Computer Science and Engineering  
School of Industrial and Information Engineering

## **A new ERP software for a new type of organization**

Supervisor: Prof.ssa Chiara FRANCALANCI

Co-supervisor: Francesco FRUGIUELE

Master Thesis of:  
Davide MOREO  
Matr. 913410  
Alberto FIORUCCI  
Matr.919546

Academic Year 2019 - 2020



# Sommario

Lo scopo di questa tesi è quello di mostrare il lavoro svolto durante lo stage fatto in Kopernicana. Kopernicana è un'organizzazione di consulenza fondata lo scorso anno (2019) dal nostro co-supervisore Francesco Frugiuele e da altri due soci, con lo scopo di migliorare l'organizzazione delle aziende attraverso diversi servizi di consulenza e diffondere la nuova struttura organizzativa chiamata olocrazia (più conosciuta con il termine inglese *holacracy*) in Italia. Come sarà spiegato meglio nella tesi, l'olocrazia è una tecnologia sociale o un sistema organizzativo di governance nel quale l'autorità e le decisioni sono distribuite nell'ambito di una olocrazia di gruppi auto-organizzati, anziché fissati in una gerarchia di tipo manageriale [58]. Queste nuove aziende hanno una carenza nel sistema informativo, in particolare nell'ERP, in quanto quelli già esistenti sono progettati e implementati per le organizzazioni gerarchiche.

Nella tesi di laurea saranno descritti due lavori, il primo riguarda il progetto di un nuovo tipo di ERP destinato ad aziende olocratiche. L'innovazione che porta questa nuova piattaforma è quella di lasciare libero l'utente di utilizzare applicazioni web già esistenti e importare da esse solo i dati importanti per l'azienda che verranno poi integrati e visualizzati insieme nel nuovo ERP.

Il secondo lavoro che sarà presentato è l'implementazione di una sezione di questa nuova applicazione per Kopernicana. Creando così una prova valida del fatto che sia possibile utilizzare applicazioni differenti per svolgere diversi task di un lavoro per poi integrare i dati e creare un sistema unico e funzionale. In particolare è stato svolto un lavoro per la contabilità e la gestione dei progetti all'interno della azienda. Un nuovo sistema di importazione e scambio dati tra il CRM, il time tracking software e due Google Sheet è stato creato insieme a un nuovo workflow per i progetti e le offerte di Kopernicana.

**Italian keywords:** Kopernicana, nuovo ERP, organizzazioni gerarchiche, organizzazione olocratica, olocrazia, integrazione piattaforme.

# Abstract

The purpose of this thesis is to show the work done during the internship in Kopernicana. Kopernicana is a consulting organization founded last year (2019) by our co-supervisor Francesco Frugiuele and two other partners, intending to improve the organization of firms with several consulting services and to share the new organizational structure called holocracy in Italy. As will be better explained in the thesis, holocracy is a social technology or organizational system of governance in which authority and decisions are distributed within a holarchy of self-organized groups rather than fixed in a managerial hierarchy. These new companies have a weakness in the information system, in particular in the ERP, as the existing ones are designed and implemented for hierarchical organizations. In the thesis, two works will be described; the first is the design of a new type of ERP for companies that practicing holacracy. The innovation that brings this new platform is to leave the user free to use already existing web applications and import from them only the important data for the company that will then be integrated and displayed together in the new ERP. The second work that will be presented is the implementation of a section of this new application for Kopernicana. This will create a valid proof that it is possible to use different applications to perform different tasks of a job and then integrate the data and create a unique and functional system. In particular, work has been done for accounting and project management within the company. A new system of data import and exchange between CRM, time tracking software and two Google Sheets has been created with a new workflow for Kopernicana projects and deals.

**Keywords:** Kopernicana, new ERP, hierarchical organizations, holacratic organization, holacracy, platform integration.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>State Of Art</b>	<b>3</b>
2.1	Pyramidal Structure . . . . .	4
2.1.1	How does it work? . . . . .	5
2.1.2	What are the problems? . . . . .	6
2.2	How to improve these types of organizations? . . . . .	8
2.2.1	Examples of Companies that changed the type of organization . . . . .	8
2.2.2	Benefits . . . . .	11
2.3	Holacracy . . . . .	12
2.3.1	Introduction . . . . .	12
2.3.2	How does Holacracy work? . . . . .	13
2.3.3	How to apply Holacracy to an organisation . . . . .	18
2.4	Example of Holacray firm . . . . .	19
2.5	Problems of holacracy organisation . . . . .	21
2.5.1	Lack of software for these types of organisation . . . . .	21
<b>3</b>	<b>Methodology</b>	<b>23</b>
3.1	Kopernicana . . . . .	24
3.2	IT tool used in Kopernicana . . . . .	26
3.2.1	"Mandatory" tools . . . . .	26
3.2.2	"Free" tools . . . . .	26
3.2.3	Problems . . . . .	27
3.3	How can we help? . . . . .	29
3.4	The survey . . . . .	31
3.4.1	The interactive question . . . . .	31
3.4.2	Open ended questions . . . . .	32
3.4.3	Result . . . . .	32
3.4.4	The grid . . . . .	32
3.4.5	Open-ended questions . . . . .	34
3.5	The API provided by each tool . . . . .	37
3.6	The PONGO application . . . . .	38
3.6.1	The architecture . . . . .	38
3.6.2	The back-end . . . . .	39
3.6.3	The front-end . . . . .	40
3.7	The manifesto . . . . .	42

<b>4</b>	<b>Implementation</b>	<b>49</b>
4.1	Pipedrive and Clockify . . . . .	50
4.1.1	Clockify . . . . .	50
4.1.2	Pipedrive . . . . .	51
4.2	Operating System Kopernicano . . . . .	52
4.2.1	How does it work? . . . . .	53
4.2.2	Problems . . . . .	55
4.3	Tools used to improve the system . . . . .	57
4.4	Clockify and Operating System connection . . . . .	58
4.5	Pipedrive and Operating System connection . . . . .	60
4.6	Pipedrive and Clockify connection . . . . .	61
4.7	The new Operating System . . . . .	62
4.7.1	What is still missing? . . . . .	63
4.7.2	Project details . . . . .	63
4.8	The complete workflow . . . . .	67
<b>5</b>	<b>Conclusions and Future Works</b>	<b>69</b>
5.1	Conclusions . . . . .	69
5.2	Future Works . . . . .	70
	<b>Bibliography</b>	<b>71</b>

# List of Figures

2.1	Pyramidal organization . . . . .	4
2.2	Pyramidal structure . . . . .	5
2.3	How to improve these kinds of organizations . . . . .	9
2.4	Changes in Rendanheyi's company from 2005 to 2014 . . . . .	10
2.5	Holacracy structure . . . . .	13
3.1	The empty grid . . . . .	31
3.2	The structure of the application . . . . .	38
3.3	ER schema of the DB . . . . .	45
3.4	Flow of the application . . . . .	47
3.5	Complete flow . . . . .	47
3.6	Example of a project details page . . . . .	48
4.1	Clockify home page . . . . .	50
4.2	Pipedrive dashboard . . . . .	51
4.3	Project-budget into the Operating System Kopernicano . . . . .	53
4.4	Work days into the Operating System Kopernicano . . . . .	54
4.5	Non-billable projects . . . . .	55
4.6	Accounting system in Kopernicana before automations . . . . .	56
4.7	Accounting system 2.0 in Kopernicana . . . . .	57
4.8	Summary report of Clockify hours . . . . .	59
4.9	Pipedrive in the Operating System . . . . .	60
4.10	Accounting system 2.0 in Kopernicana . . . . .	62
4.11	Workflow of Project details Sheet . . . . .	65
4.12	Full schema of the implemented system . . . . .	66
4.13	Complete workflow . . . . .	68





# List of Tables

3.1	API list . . . . .	44
3.2	How we get the data . . . . .	46



# Chapter 1

## Introduction

This thesis has three different goals. The first one is to analyze the typical kind of organization (with a pyramidal structure) and compare it with the new one called *holacracy*. In order to do this, we focus our attention on the problems of the first one to describe how holacracy structure solves them and to explain the improvements that Holacracy gives to the companies that adopt it. The second scope is to design a new ERP for holacracy companies and in particular, for the organization we started working for, called Kopernicana. This new innovative ERP is also called 'pongo' ERP due to its adaptability and the possibility to aggregate various functionalities and tools in it. To design it and make it innovative and functional for these new companies, we had analyzed how Kopernicana adopts the Holacracy structure, the IT platforms that are used in the organization and then, a survey has been done to all the members. With all this information, finally, we were able to structure the 'pongo', in which will be possible to merge data and dashboards from several other platforms. This is made possible thanks to the use of API provided by the tools that will be integrated into the ERP. The last scope is to improve the accountability system of the company. It has been done implementing the Operating System Kopernicano, by automatically links the CRM and the time tracker application to the Google Sheet that is used to manage the system.

## Structure

In the state of Art, chapter 2, we presented the structure of the pyramidal organizations: how they are organized, how they work, and the problem that the various kind of pyramidal structures have. After that, we explain how it is possible to improve the efficiency of these structures by giving some examples of some companies that change the structure and then arrive to adopt the Holacracy structure. So we explain how Holacracy, an oligarchic structure, and how this new type of organization can afford the problems of the pyramidal one and how it can improve the efficiency by splitting the responsibilities between the employee and by destroying the pyramidal structure. We bring some examples of organizations that change from and hierarchical structure to an Holacracy based one.

In chapter 3, the Methodology one, we explain how Kopernicana, the Holacracy company for which we have worked, is structured and the tools that they use to do work and the problems that the employee has to use several different tools. After that, the survey that has been done to analyze the situation and determine what could be improved in the information system is shown with its response. From the survey, different helpful data comes out, and it was used to design a new type of ERP that will be presented in the last section of this chapter.

In chapter 4, the last one, we will show how Clockify and Pipdrive work, in specific how they are used in Kopernicana. Then will be described how these two tools have been connected to the already existing operating system in order to automate it and create a better workflow in the accountability system of Kopernicana.

# Chapter 2

## State Of Art

## 2.1 Pyramidal Structure

A pyramidal organization is the most standard type of structure that a company can use to start its life. These type of companies can be divided into three parts: the lead, the control and activity. Starting from the top of the pyramid, the lead part is the one that can determine almost all the decision-making strategies of the company. Then there is the control part of the company, where the people working there are responsible for what the activity sector must do. Lastly, but not for importance, we have the activity sector, that is the area with more workers than the preceding ones and its role is to produce what the company sells.



Figure 2.1: Pyramidal organization

In most cases, we can subdivide each part into more sectors to have more control. This type of organization can be divided not only vertically but also horizontally. Most companies have many different kinds of activities: the marketing, transport and production areas are just some examples, therefore it's possible to understand how to subdivide the activity sector and, consequently, the control area has to be subdivided in the same way. The only sector that is difficult to subdivide is the lead part, that has to control the general part of the company.

The structure that we have just illustrated is the common structure of a company that uses the pyramidal organization.

We can group the pyramidal organization into third sector, volunteer association, (including all the companies that are nonprofit and non-governmental), PMI (small to medium-sized companies) with less than 50 employees, PMI with more than 50 employees, large companies, a sector that includes: confederations, associations, incubators, networks, PA (public administration) and startups.

### 2.1.1 How does it work?

The pyramidal company has a particular structure; therefore, the work is quite complex. The lead sector takes care of each task. In general, as specified above, the control part is subdivided into more than one block, and each block manages a different kind of task. The lead must instruct each section and monitor how the control area is working and if necessary, correct it give or give instructions to better explain the task at hand. Next, the control area must understand what the lead has assigned to it and start organized the activity area. The control sector is responsible for the work of the activity area. In fact, when a worker in the activity area has a problem understating a specific task, they must ask the control area for advice because as far as the company concerned they haven't got the competences to be responsible for the project they are working on. If this happens, the control area analyzes the problem and has three options: firstly, it can solve the problem with its own competences, that is the control block in charge of the specific activity can directly resolve the issue of the worker; secondly, the problem can be solved only by a different block that is not the one that controls the activity area that has the problem. In this case, the manager of the control area must ask to a manager from another block or the activity area that he/she manages to resolve the problem. Once a solution has been identified, it must be referred to the initial control area that will provide the solution to the activity area with the problem. The final option is to refer the issue to the lead sector in turn, that will try to fix the problem and provide the solution to the manager.

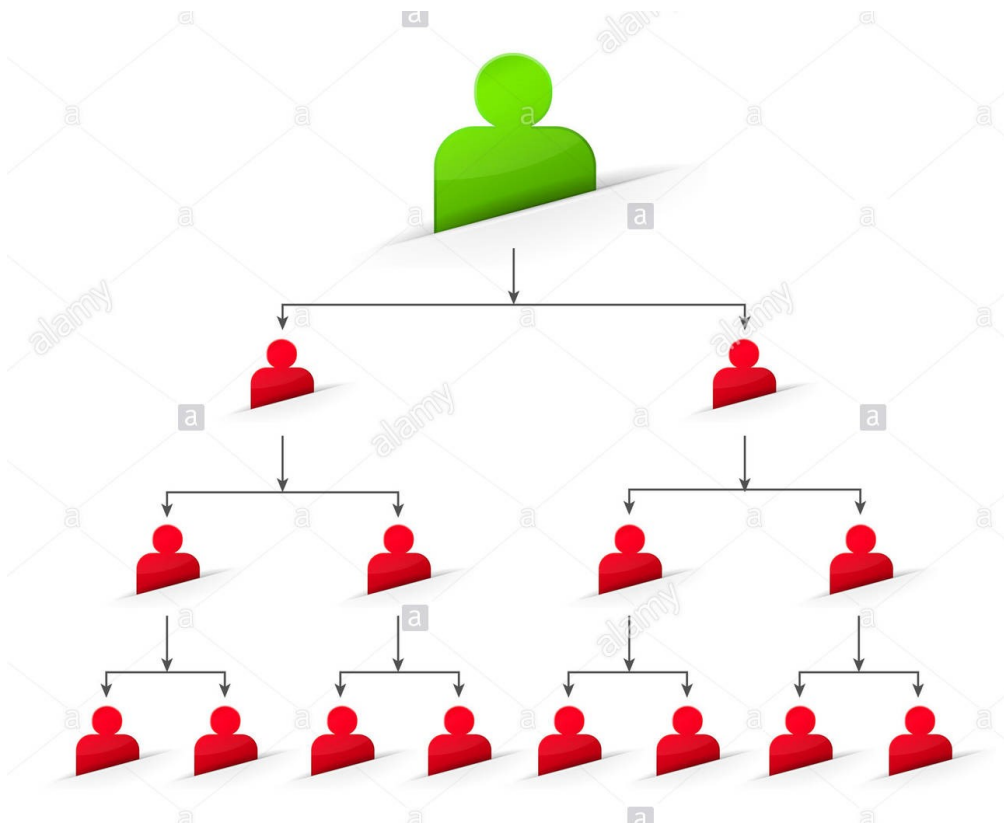


Figure 2.2: Pyramidal structure

### 2.1.2 What are the problems?

This type of pyramidal organization has a lot of organizational and coordination problems, based on the dimension of the company and the sector in which they work.

For each type of company, we can divide the problems into those expressed by the lead/control areas and those not expressed by the lead/control area. In the former, we have all the problems that the management class is aware of and that must be improved to make their company more comfortable, agile and productive; whereas, in the latter, we have all the problems that the management class is not aware of, or completely ignores.

The following are the different problems that each of the companies that have been listed above may face.

*-Volunteer Association:* The expressed problems include a minimum use of software as these associations usually use paper documentation and do not often use a database or a tool where they can group together all their activities. Another problem stems from this: those who must manage the workers remotely, do not have any kind of experience with the software. Another problem is the lack of communication and collaboration between the teams and often also inside each team. This is a problem with almost all the associations because with the pyramidal structure there are many issues that must be faced to ensure an efficient collaboration between different groups of workers that are involved in different types of production and that must collaborate. In fact, they must always contact their manager to operate with the other groups or to make a decision that can change the structure of the project they are working on. The final expressed problem is an inadequate and not satisfactory training of the management part. Regarding the non-expressed problems (the issues that the associations have but are not aware of): it is often not easy to align the strategies and the processes, make meetings and alignment tools more effective, focus the company's purpose (in a moment of change or crisis), refine awareness of how the organization works, making decision-making mechanisms faster and more effective, refine awareness of how the organization works.

*-PMI(<50 people):* three problems have been identified here, all of them expressed. They are difficulty in converting to the software mode as these enterprises use a database or software where they group all the activities but in general, the software or the ERP where the file is confusing, and there is not a real organization of the file. Another problem of this kind of firm, as in the voluntary associations, is little collaboration between the individual and the teams. This is a problem with almost all the companies because with the pyramidal structure there are many issues that must be faced to ensure an efficient collaboration between different groups of workers that are involved in different types of production and that must collaborate. In fact, they must always contact their manager to operate with the other groups or to make a decision that can change the structure of the project that they are working on. The last expressed problem is the formation of the management staff. *-PMI(>50 employees):* Considering that this type of organization has a considerable amount of workers, there are many problems to organize the people, such as: align strategy processes, management training, attract top talent, remotely manage distributed teams and monitor the teamwork. In general, this is also a software problem because



these companies can have a bad organization of their software, which makes difficult to plan the work. The final expressed issue in this type of company is to promote the empowerment and autonomy of the employees as working on an important project can cause them great stress and it's important that their happiness and wellbeing is taken care of. As far as the non-expressed problems are concerned, there are several problems that the managers do not understand: make meetings and alignment tools more effective, in fact in the PMI with more than 50 people, there are problems with the organization of documents and the kind of tools that each person uses. in order to improve the organization, the manager needs to focus on the company's purpose (in a moment of change or crisis), refine awareness of the functioning of the organization, make decision-making mechanisms faster and more effective.

*Big companies:* in this organization, we have many problems mainly because they must coordinate the work between many people, and the manager usually has too much responsibility and as a result to much stress. In this case, we have a lot of expressed problems: promoting the empowerment and autonomy of the employees, improve the onboarding of new workers in the organization, switch to software mode, align strategies and processes, monitor teamwork, simplify processes and procedures, simplify the relationship with users, improve collaboration and sharing between people and teams, management training, attract top talent. But there are other problems that are non-expressed by the managerial class because they believe that their methodology is consolidated and well-formed with respect to any other kind. For example, improve internal communication and the structure of meetings that generally follow an inadequate model, refine awareness of the functioning of the organization, make decision-making mechanisms faster and more effective.

*Confederations, associations, incubators, networks:* in this type of organization, there are not many problems. There are two expressed problems: switch to software mode and improve internal communication. These problems, whether expressed or not expressed, are present in all the other companies whose organization follows the pyramidal structure. Lastly, there are other two non-expressed problems: refine awareness of how the organization works and make decision-making mechanisms faster and more effective. These are considered not expressed because the managers do not think they have this problem, but like the other two are a common problem of all pyramidal structures.

*PA:* this organization is almost the same as the big companies that we have previously written about, so they have the same expressed and unexpressed problems.

*startup:* a startup has almost the same structure as the volunteer associations, so it has the same expressed and unexpressed problems.

From this list, we can observe that the problems that these companies have in common are to improve cooperation between the workers and to lighten the responsibility and the coordination between the groups of workers. Why are these problems present?

The pyramidal structure can't respond to the fast changes of the new market, in fact, all the decisions have to be approved by the manager. In this case, the new organizations need more flexibility to react to the fast changes of the market.

## 2.2 How to improve these types of organizations?

### 2.2.1 Examples of Companies that changed the type of organization

A lot of organizations are becoming aware that they have to react more efficiently and faster to the requests of the market. To do this, they have to improve their structure; the problem is: how? The common answer is to delegate the responsibility of the manager to the other workers. In this way, the employees can directly face the issue, without losing time to ask their superiors. A first example is the ‘Morning start Company’ a company that produces tomatoes and all other products that derive from them. As described in the paper ‘The Morning Star Company:Self-Management at Work’ [32], "this company is a self-managed structure. The only person that has a title is Chris Rufer, only because the organization needs a president for legal purposes. The other people that work for the company, have the same responsibility and duty. With this technique, all the workers don’t have the concept of the superior and don’t have to think to do the job only to satisfy what the superior. Each colleague can do which job he prefers and with the skills, he has to have the skills to perform the best of what he can do. With time the ‘colleague’ role gets his personal colleague principals, that are:

- An organization is an idea, a concept. People, as individuals, are the ultimate reality and the only operative element.
- Profitability and growth are the natural outgrowths of excellent execution and innovation, developing insights into market desires and producing goods and services which meet customer desires.
- People think, innovate, communicate, and execute best when they feel good about themselves. Individuals feel good about themselves when they are in control of their lives—when they have achieved freedom of choice over what is important to them.
- The principles of self-management and total responsibility, which underlie family and community relationships, are also the most efficient and effective principles for organizing commercial relationships.
- Competition is to be embraced as a significant source of personal and process improvement."

All Morning Star employees had to write a personal mission statement, outlining how they would individually contribute to the company’s goal of “producing tomato products and services which consistently achieve the quality and service expectations of our customers.” Green explained, “an individual who was responsible for sorting tomatoes on the line would not define his/her mission as, ‘my job is to remove bad tomatoes’, but instead might say, ‘My mission is to ensure that our customers receive pristine tomato products free of foreign material.’” Rufer added, “[At Morning Star] you are responsible for the accomplishment of your mission and for acquiring the training, resources, and cooperation that you need to fulfill your

mission.” How is it possible to see from this brief extract from ‘The Morning Star Company: Self-Management at Work’ [32] that all the structure is built on the workers, to make them feel satisfied and accomplished. Another example is Haier’s Rendanheyi, a company in China, and how it also evolved its structure with a high level of VUCA ( volatility, uncertainty, complexity and ambiguity ). From the introduction of Rendanheyi in 2005, Haier was transformed from a conventional hierarchical appliance manufacturer into a highly responsive, diversified online-based entrepreneurial conglomerate organized as an entirely autonomous micro-enterprises. This company started to use a new method to improve its productivity, in fact, as is written in the paper Management innovation made in China: Haier’s Rendanheyi [37]: “the Birkinshaw et al. process model proposes that there are four distinct phases in management innovating. The first of these comes about when change agents are motivated to consider creating a management innovation because of some dissatisfaction with the status quo. The second stage, invention, occurs when a thought or real experiment is started to create a new hypothetical management practice. In the third stage, implementation, this hypothetical practice is put into practice to establish its value. Finally, in the fourth stage, theorization and labelling, there is retrospective sense-making whereby innovation can receive legitimacy.”

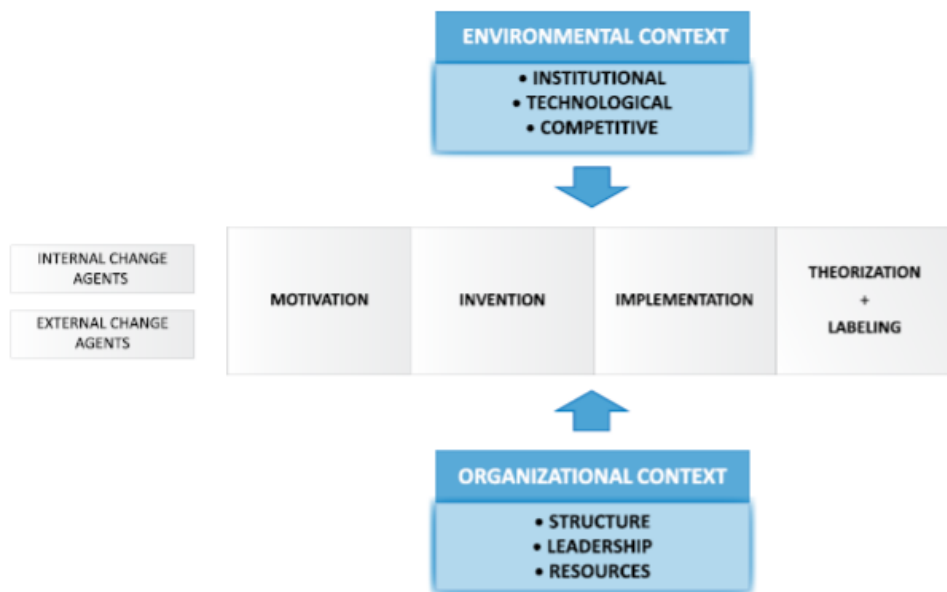


Figure 2.3: How to improve these kinds of organizations

When the Haier company applied this technique, it started to expand its business in a really short time, becoming one of the top three white goods sellers in the world. In 2005, it decided to try to become the number one in the world, but the problem was *how to do that??*

The answer was simple: re-organize the hierarchy structure that the company used and in 2014, the organization was restructured. In this table we can see what the changes were from 2005 to 2014: From the table taken from the paper ‘Management innovation made in China: Haier’s Rendanheyi’ [37], we can see that the most important change is passing from a hierarchy organization to an

	<b>Rendanheyi (2005-2014)</b>	<b>Rendanheyi 2.0 (from 2014)</b>
Organizational structure	Organizational hierarchy with increasing autonomy for corporate units—moving from strategic business units to micro-divisions called ZZJYT	Collection of independent micro-enterprises called xiaowei
Ownership	100% Haier Group-owned	Variable ownership of micro-enterprises with stakes from external partners, initially majority Haier Group-owned but increasingly diluting ownership
Corporate unit performance evaluation	Baseline targets for sales and profitability set by headquarters for SBUs/ZZJYTs and unit leaders	Performance measured against a mix of financial KPIs and user value added
Individual performance remuneration	Basic salary plus individual performance-related bonuses	Remuneration based entirely on micro-enterprise performance

Note: ZZJYT = *zi zhu jing ying ti*, translated as “independent operating unit; SBU = strategic business unit; KPI = key performance indicator.

Figure 2.4: Changes in Rendanheyi’s company from 2005 to 2014

organization subdivided into groups and each group has its own responsibility and does not have to respond to any other structure or person. With this method, the happiness and the satisfaction of the workers improved.

### 2.2.2 Benefits

As we have seen in the previous chapters, each time that a new company gets larger, it also needs to transform the structure of the organization into a new one, which must respond faster and more efficiently to the changes of the market. How does this happen in general? First of all, the company needs to have a leader with an open mind because he must recognize that the hierarchy is not the most efficient way to structure the organization. Subsequently, the organization begins to change its structure and the workers became more independent from the CEO until the position of the CEO is completely abolished. At this point, the work projects are assigned to individual groups that are independent of each other. This is more efficient with respect to the pyramidal structure because the workers can choose in which sector of the company they can work and establish what kind of work they can do and when they can do it. The employee is more motivated to work and can react more efficiently and faster than with the other structure because they are freer to choose how to react to an emergency or a change in the market. This obviously implicates more responsibility on the workers but it is in fact a contraindication. However, with respect to the possibility of growth with this restructuring, it is not considered a risk.

## 2.3 Holacracy

### 2.3.1 Introduction

As a city grows in size, its efficiency goes up about 15 percent, but when a hierarchy company grows quickly, what happens? Its productivity decreases, based on how many people are working in that company. The more people the manager has to coordinate, the more time he/she needs to do his/her work and the more the company must pay the manager to do things that could be delegated to the workers. For this reason, adopting Holacracy is a very good choice for a company that is growing. Why is this so? Holacracy demolishes the pyramidal structure. In fact, each person has a role that has its own domain where decisions can be made that allow the worker to operate freely within their domain. Each person can have one or more roles and when there is the necessity to assign a role to a person, the decision to accept that specific role is left to the person concerned. With this simple procedure that is not the only one that Holacracy implements, the worker is freer and can decide how, when and what kind of work he/she has to do, making them more comfortable and with a quicker reaction to the changes of the market. In this way, when the company grows and hires more employees, it can increase its efficiency. Obviously, this method is not so simple to apply, especially when a company has to pass from a hierarchy structure to a Holacracy structure. In general, in a pyramidal structure, the standard worker must only work and not have to make any kind of decision, whereas with this new type of organization all the employees have the same responsibility.

As we can see from the image 2.5, Holacracy has a circle organization where it is easier to find which role and subsequently which person does a certain job, and if an employee needs specific information, for a project, they will know immediately who has that information. In a pyramidal structure, it is rare that an employee will consult the organization chart to find out something that they need to know. In this case, they would ask their superior and continue going up the ladder until they arrive at the manager who knows where to find the problem. This method takes much time to resolve the problem of just one employee, so if this method is applied to all the company problems, we would have a significant loss of time. For this reason, the Holacracy structure is very efficient. Another advantage of Holacracy is that the structure of the organization is not static but dynamic, because it is based on what we will call tension from now on, for example, when an employee sees that his/her role creates too much stress on the job, they can report the issue at a specific meeting ( that we will see in the next section) and try to resolve the situation by creating a new role. Therefore, the structure of the Holacracy organization is in constant evolution, making the job of the workers easier and less stressful.

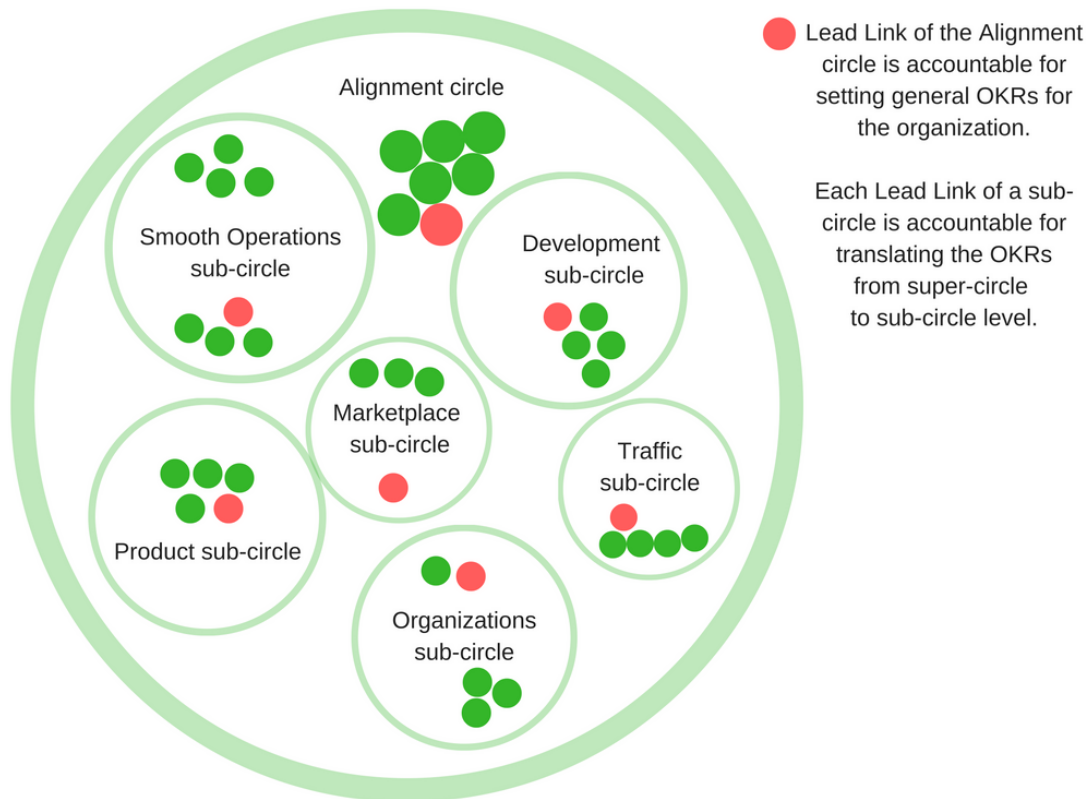


Figure 2.5: Holacracy structure

### 2.3.2 How does Holacracy work?

At this point, let's discuss Holacracy in more detail. To really start making our company a holacracy based structure, we must define the roles that a company needs to be governed. In holacracy, there are various types of meetings, so let's start from the most important to build the organization chart and the meeting where we will begin to implement Holacracy: the governance meeting. First of all, two important roles must be defined in each meeting: the first one is the facilitator that has the purpose of managing circle governance and operational practices aligned with the constitution and is accountable for facilitating the circle's constitutional-required meetings and auditing the meetings and records of Sub-Circles as needed, and initiating the restorative process defined in the constitution once a Process Breakdown has been discovered. The second one is the secretary that has the role of steward and stabilizes the circle's formal records and record keeping process. His/her domains are to constitutionally-required all records of the Circle, with the accountabilities of:

- Scheduling the Circle's required meetings, and notifying all Core Circle Members of scheduled times and locations
  - Capturing the outputs of the Circle's required meetings, and maintaining a compiled view of the Circle's current Governance, checklist items, and metrics
  - Interpreting Governance and the Constitution upon request
- After the assign of these two role, the governance meeting can start. The purposes of the governance

meeting are:

- Creating, amending, or removing roles within the circle
- Creating, amending, or removing policies governing the circle's domain
- Electing circle members to fill elected roles (facilitator, secretary, and rep link)
- Creating, amending, or dissolving sub- circles

With practice, circle members will learn what kinds of issues are best addressed through governance changes. If a key task has been consistently falling through the cracks, and it becomes clear that it needs to be added to the accountabilities of a particular role, that's a likely target for a governance meeting. If the relationship between the two roles is unclear, creating tension and communication problems, that can be clarified in a governance meeting. If the individual filling a role wants the authority to make certain decisions, or place limits on others' authority, that, too, can be brought to the table. Governance meetings are not the place to be dealing with the marketing strategy, the product offerings for next year, or any other execution-oriented decisions facing the team. Those are operational questions and should be addressed day-to-day outside of meetings, or, sometimes, in a tactical meeting. I'll explain that process later. After understand what can be done in a governance meeting, we can start to explain the various phase of the meeting. The first phase is the check-in: this phase is really important because all the participants have to outside their filling one by one, in a circle round, to let know to the other people if they are stressed if they have some personal issue, if they are a little distract to other question or simply to let know their emotion to the other members. This operation is really important, in fact, based on the feeling of each person, the other can adapt their language or can determine why a person is more depressed or happy in a determinate period. This is a method to lets the meeting less formal and more comfortable for all the participants. The second phase is the administrative concerns: this is a little space where the facilitator let know to the other people the information about the meeting, like the duration, if someone has to left first the meeting or information like these. While the facilitator is listing these questions, the important point is that he is the only person allowed to talk and there is no space, for any other member, to makes a discussion. The third phase is the building agenda one, where the members of the meeting list all the tensions about the role that they have, again in this phase there is no discussion. The fourth phase is the integrative decision Making, this phase is divided into six more phases that have to be done for each tension that was listed in the previous phase. This phase is:

-*Present Proposal*: only the speaker proposes unless help is requested. The proposer has space to describe a tension and state a proposal to resolve it, with no discussion. The proposer can optionally request discussion just to help craft a proposal, but not to build consensus or integrate concerns.

-*Clarifying Questions*: in this phase anyone can ask a clarifying question to seek information or understanding. The proposer can respond or say "not specified". No reactions or dialogue allowed.

-*Reaction Round*: Each person is given space to react to the proposal as they see fit; reactions must be made first or third person comments. No discussion or responses.

-*Amend and Clarify*: the proposer can optionally clarify the intent of the proposal further or amend the proposal based on the reactions, or just move on. No discussion allowed.



*-Objection Round:* the facilitator asks: “Do you see any reasons why adopting this proposal would cause harm or move us backwards?” (an “objection”). Objections are stated, tested, backward without discussion; the proposal is adopted if none surface.

*-Integration:* Focus on each objection, one at a time. The goal is to craft an amended proposal that would not cause the objection, but that would still address the proposer’s tension. Once all are integrated, go back to the Objection Round with the new proposal.

The last phase is the closing round, where once the agenda is complete or the meeting is nearing its scheduled end, the Facilitator gives each person space to share a closing reflection about the meeting. With this, we have done our organization structure, and with this method, we can update the organization of the company where we want. Now we have the operations part where we find how to manage all the other think that is not part of the internal structure of the company and the strategy of the company. So we have to distinguish between project and next-action: a project is any desired outcome that requires more than one action step, the next action is the next physical, visible activity that needs to be engaged in order to move the current reality towards completion. Now the person has a role and based on the role, whom that person accepts, he has assigned some responsibilities on the operation part that are:

*Sensing and processing tensions around the role’s purpose and accountabilities, through the various channels available*

*Processing accountabilities:* regularly identifying specific next- actions you could take and defining projects you could work toward to fulfil the role accountabilities

*Processing projects:* regularly identifying next actions that would move each of the role’s projects forward

*Tracking projects and next actions:* capturing all of the role’s projects and next-actions in a database or tangible form accessible to others, outside your own mind

*Directing attention and resources:* consciously and continually choosing the next-action or other activity that it makes the most sense to direct your attention and resources toward, all things considered, and then taking that action.

This is the basic responsibilities that members have, but they have a specific duty based on witch circle member they belong. This duty is: transparency about project and workflow, processing the thing that they can do when a member of the circle ask to him to do and the prioritization of the time paying attention to the other resource. The last step is to make the tactical meeting, a periodically meeting, that is needed to share what the circle have done till now and where the person can express his tension to the company and discuss them. The tactical meeting has seven steps that we will describe in the next lines:

*Check-in Round* Goal: Notice what’s got your attention, call it out, let it go. Sacred space: no cross-talk. Get present, here and now; grounds the meeting.

*Checklist Review* Goal: Bring transparency to recurring actions Facilitator reads checklist of recurring actions by role; participants respond “check” or “no check” to each for the preceding period(e.g., the prior week.)

*Metrics Review* Goal: Build a picture of current reality. Each role assigned a metric report on it briefly, highlighting the latest data.

*Progress Updates* Goal: Report updates to key projects of the circle. The facilitator

reads each project on the circle's project board and asks: "Any updates?" The project's owner either responds "no updates" or shares what has changed since the last meeting. Questions allowed, but no discussion.

*Agenda Building* Goal: build an agenda with placeholder headlines. Build an agenda of

tensions to process; one or two words per item, no discussion.

*Triage Issues* Goal: Get through all agenda items in the allotted time. To resolve each agenda item:

- The facilitator asks: "What do you need?"
- Agenda item owner engages others as needed.
- Capture any next- actions or projects requested and accepted.
- The facilitator asks: "Did you get what you need?"

*Closing Round* Goal: Harvest learning from the meeting. Each person can share a closing reflection about the meeting; no discussion.

With these seven steps, the company can complete a tactical meeting that updates the operational situation of the company for all the employees and resolve all the tensions that afflict the members of the company. Obviously, they will be all resolve if and only if there is the time to respond to all tensions. If the time is not sufficient, the tensions remained will be dropped and it will be solved separately in another tactical meeting or if possible with a meeting with specific people who are involved in that tension. The last step to complete the dynamic company applying Holacracy is the strategic meeting, that is the meeting where the members of the company decide what kind of strategy the organization have to adopt. In general, is very difficult to predict an entire strategy for the live life of the company, so in general every six months the lead link can ask for a strategically meeting with the duration of four/five hours to redesign the strategy of his circle. Now, like for the previous meeting, we list the phase that this kind of meeting needs to have, to be more efficient possible:

Check-in Round (the same of the other two kinds of the meeting)

Orientation: review purpose, domain, and accountabilities of the circle, and any strategies of super- circle

Retrospective:

- Each participant silently reflects and captures notable facts, data, events, and history.
- Participant post their notes on the wall, and collectively organized/group related ones.
- Participants describe/clarify keynotes and share reflections; facilitator makes a list of key tensions raised while sharing

Strategy Generation:

- Capture and post ideas individually: what should we emphasize, given these tensions?
- Discuss collectively: what strategy or strategies should guide us, looking forward?
- Lead link proposes one or more strategies; these are processed via Integrative Decision- Making

Unpack the Strategy

- Each participant individually reflects and captures projects and next- actions to take in his or her role.

- One at a time, each participant shares what she has captured and solicits/receives input and ideas

Closing Round (the same of the other two kinds of meeting) Finally, we have introduced all the meeting and all we need to implement an Holacracy based Company.

### 2.3.3 How to apply Holacracy to an organisation

The first thing to understand is that a company cannot use only a part of the Holacracy method, for example the meeting to really improve the efficiency of the Company, but all the parts that are present in the Holacracy structure must be involved. This is because if the company maintains the pyramidal structure and tries to adopt the meeting of Holacracy, it would be a very serious problem. The Holacracy structure views all the people on the same level, the only thing that changes is the role, and there is no role that is superior to another and in case of a discussion, it would be between two roles and not two people. In the pyramidal structure, some people are superior and have more importance with respect to another, so the dialogue is not on the same level. When partially applying the Holacracy method, a manager can discuss with a normal employee and this could be very dangerous because it could anger the manager to have to discuss with a person that is inferior with respect to him/herself, or likewise, the employee may feel resentment towards his/her superior. This will happen because the discussion is between two people and not between two roles. For this reason, we cannot partially apply Holacracy or the risk would be less efficiency in the firm. So there are two different methods to apply the Holacracy paradigm to a Company: the first one is to gradually start applying the Holacracy rules, like when a child starts to learn a sport. The instructor or the parent teaches the child the rules of the sport that he wants to learn. In regards to Holacracy, there is a constitution that the employees will refer to so they know what to do, but when does a child really understand and get fully involved in the sport? Obviously when he/she starts to practice! The same goes for a company, Holacracy needs practice. With this method, the employees need time to get used to it, but at a certain point, the Holacracy constitution will be automatically applied. The second option to consider is that while you can't really practice Holacracy by adopting only part of the rules, you can apply all of the rules to only a part of the company, as when just one manager decides to adopt Holacracy without enlisting the broader organization. In that case, the manager adopts the Holacracy constitution and cedes power to it, but the "organization" that the constitution refers to is just his/her team or department. An adoption as such in just part of a company can be useful even if the entire organization eventually intends to follow suit. Especially in larger companies, it's not uncommon to see one department or division pilot Holacracy before the organization goes all in. These two methods are the only two that can be applied because all the other methods are not sufficiently efficient, and the risk, as discussed in the previous paragraph, is too high to not be considered.

## 2.4 Example of Holacray firm

A Company that has gone far by applying Holacracy is Zappos. Zappos is an online clothing company, founded in 1999 by Nick Swinmurn, in 2009 became one of the sub-company of amazon. From the 2014 Zappos go under the eye of the journalist because it was trying to implement Holacracy structure and eighteen percent of its employees chose to leave the company. “Tony Hsieh, Zappos’ CEO, offered a generous severance package to these employees who were uncertain about the reliability, adaptability, accountability, and efficiency of the firm’s operations.” Is write in Kumar S, V.a-Holacracy-the-future-of-organizing [57] paper. Zappos, with his 1500 workforce was known for his self-management vision of the company before to try to apply holacracy. But there were a lot of employees who were septic respect the holacracy structure because other companies were trying to apply it, not completely, and they fail. The employees who remained with Zappos envisioned the initial years of holacracy implementation as a transitional stage which would lead to learning and innovation. So the employee can customize his career and salary based on the skills and role they acquire. So more skills and the role the employee learn more will be the remuneration is. This amplified the opportunity of the employee to find new functions, earn more money and go deeper with their passion. But how this system managed? In Kumar S, V.a-Holacracy–the-future-of-organizing [57] paper is reported that: “Self Management enables heterogeneity in expertise in its demographic of circles. These circles are further divided into sub-circles managed by lead links, with employees freely moving across circles based on their new interests and desire to learn new functions. Zappos is enhancing a collaborative mindset and making the system more appealing by introducing a coworker reward policy. This enables employees to spread satisfaction by rewarding coworkers for their vital contribution and diligent adherence to cultural values. This is facilitated by a stringent recruitment process which continues to focus on cultural fit. Therefore, employees are motivated to work as mini-entrepreneurs, running their respective circles as mini-enterprises.” From the experience with holacracy the Zappos tries to implement four principles, how is write in Kumar S,V.a-Holacracy–the-future-of-organizing paper [57] :“ First, the CEO realized that current thinking was insufficient to break the mould. Self Management structure in a growing organization was implemented to improve productivity, despite an increase in organization size. Although it was in its prime, Zappos did not shy away from the new generation’s concept of holacracy. Second, new ideas and collective individualism have together enabled Zappos to sustain their position. This was also evident in its focus on symbiotic growth through the coworker rewards systems and its shared-leadership structure. Third, Zappos adopted holacracy as its collaborative organizational structure to foster an understanding of the changing role of self and others in a team. Finally, the synergy between ideas, individuals, and the structure was integrated by organizational culture.” The case of Zappos and its journey toward an organization that is ahead of its time has three noteworthy lessons: 1. The conflict between old and new generations is inevitable when implementing new ideas. However, a total focus on the new generation is insufficient. The best of both worlds should be embraced. 2. Do not wait for critical events to force change. Initiate the change process when the firm has enough buffers to absorb the shocks.

3. Companies should also take care of the “too-much-of-good-thing” effect (Pierce and Aguinis, 2013) to reduce chaos and to avoid detrimental effects in the long run. Structure alone is insufficient for producing desired change.

## 2.5 Problems of holacracy organisation

### 2.5.1 Lack of software for these types of organisation

For this type of Company, there is a greater use of tools and software with respect to other structures and the reason is that tools or software optimized for this type of organization do not exist. So there is a big problem to structure a workflow that is collectable to a standard ERP. For example, the financial part is a big problem to standardize because the company's employees do not have a fixed salary but their wages are based on how many hours they work, their role in the project, the project budget and how they work. To take care of this payment several tools are needed, one to manage the CRM, one to manage the hours that the employee spends on that project, a document to manage how much the worker is paid by the hour, and a program (excel file) where all this data must merge to see how much the employee has earned. If all this is needed just for the financial part, in order to implement a Holacracy organization, we need many tools and software to manage all the different parts, with the risk of over-stressing the employees. What is the solution to merge all these tools? A new kind of ERP that we will explain in the next section.





# Chapter 3

## Methodology

The main objective of this project is to fill in the IT gaps that holacracy organizations, as explained in the last section of chapter 2, still have. The standard ERP software are structured following the hierarchical organizations of firms and not designed for the holacracy ones. This chapter aims at showing how we arrive to design a new kind of ERP designed for holacracy organizations. In section 3.1 what is Kopernicana and the feelings we had during the first month working with the member about this new kind of organisation will be described. Next, in section 3.2, some of the IT tools used by the organization's members are presented with all the problems and difficulties they have to work with it. In section 3.5 the APIs that make possible the implementation of this innovative project are described. Later, in section 3.4, will be shown a survey that was done to better understand how Kopernicana works and the necessities of the members to improve their working life. Next, the design of the new ERP is explained in a detailed way, in particular, in section 3.6 all the technical details are discussed. At the end of the chapter, in section 3.7, all the features that make the application innovative are shown and illustrated.

## 3.1 Kopernicana

Kopernicana is a consulting Holacracy firm born last year (2019) to help companies improve their organization. Kopernicana is composed of eight members. Each of them comes from different work fields, with several knowledge and skills; in the team, entrepreneurs, designers, researchers and innovators are present. All of them have been working for years into different hierarchical organizations with different roles, but after discovering holacracy, they understood that it could be a great reality in the working world of the future and decide to merge their capabilities together to spread more and more this new type of organization.

The main goal of Kopernicana is to try to share holacracy into companies, but also helps firms to improve their efficiency in different ways with various services: *organization cartography*, *enterprise design* and *clarity and OKRs* are some examples. It would be very interesting to explain in a more detailed way how they work with companies and describe their main services, but it would take too much time and space in the thesis; in order to have more information about the company check their website<sup>1</sup>.

What is more interesting for us and more important now for our work is to show how a company that practicing holacracy, like Kopernicana, works from an IT standpoint and how its employee, or better members, behave and interact with each other and with IT tools, this to understand the problems present in this type of organizations and try to solve them.

To have a complete overview of the work, how were the firsts contacts with Holacracy, how slowly we got into the dynamics of Kopernicana and how we got in touch with the members' workday will be explained now. The internship started at the end of February (2020) after a meeting with the co-supervisor, where he easily convinced us to try this experience in this new type of company. The idea of work into a new kind of organization, future-oriented and without hierarchy was really exciting. In the beginning, we didn't have a specific job to work on, we were free to help Kopernicana the way we thought the best. The freedom of each member is an important feature of holacracy organization; no one will tell you what you have to do or not to do, everyone is free to bring new ideas in order to help the organization. As has been written in the last section of chapter 2, there is a lack of software for Holacracy companies and this is the only thing our co-supervisor told us before starting the internship. This is really interesting because we have the freedom to work on what we prefer but at the same time a little bit tricky because we had to understand really well all the necessities of the company to come out with a satisfactory idea. Our first step was to enter in the life of Kopernicana to better understand the internal dynamics, the problems and the strength of this organization.

Kopernicana doesn't have a physical place where members work together (this also before Covid-19), they use to meet only (virtually) once a week on Zoom to do Tactical and Governance meetings, for the rest of the time they work from where they prefer. Hence the only contact we had with all the members of the teams together was the weekly meeting on Zoom. Due to the big news of the Holacracy

---

<sup>1</sup>[www.kopernicana.com](http://www.kopernicana.com)

world and the little amount of time spent with Kopernicana members, we needed at least a month to figure out some easy dynamics of the company and to adapt ourselves into the organization.

This is normal since some technical details of this organization still have to be well understood by the members to achieve a perfect use of the holacracy structure, for example, some meetings dynamics are not clear to everyone and sometimes there are difficulties to follow the strict rules of the tactical and governance meetings. Holacracy is completely a new working world and the differences with the common companies are enormous. What came out from these meetings is that holacracy can be really positive for a firm. The members of Kopernicana are all on the same level, all the responsibilities and roles are clear and the workflow of the meetings gives the possibility to everyone to interact and to expose his ideas without losing time, hence the efficiency is highest. For sure, it is not an easy transformation and change for a company, but the power of holacracy is evident and could be a great improvement in the future of the work.

## 3.2 IT tool used in Kopernicana

During this month of adoption, in addition to attending all Kopernicana's meetings, studying the structure and the principal features of Holacracy, we also read a very interesting book [50], written by Robertson (the creator of this new type of organization) and we had some meetings only with our co-supervisor where he explained us some details about the way of working, he showed us some applications used by all Kopernicana's members and some that he only use to do his jobs in the company.

Since the way of work and which tools are used in Kopernicana are two fundamental aspects of the work done in the internship, here will be presented the main tools used in the company.

### 3.2.1 "Mandatory" tools

The co-supervisor firstly showed us Clockify <sup>2</sup> a time tracking software used by all the members to keep their amount of time spent working for Kopernicana updated. In this web application is possible to insert the hours spent working, selecting the project and the task; it is used by the organization not to control how much each member works but because, how we will explain later, the members of Kopernicana aren't normal employee, and their salaries depend on several factors included the amount of time and what project they worked on.

Then he explained to us how Kopernicana uses Holaspirit<sup>3</sup>: a platform for self-managed organization where it is possible to insert all the information about members, roles, circle, internal projects and all the others information of an Holacracy company. It is also used in the meetings to save everything that happened during them. An application like this must be used by all the holacracy organizations as the base of the structure, as we already described in detail in chapter 2 roles, members, meeting and activities are the keys of these companies and without Holaspirit (or another similar application) would be impossible to have a place to save and modify all these information easily.

These two software, in addition to Zoom, used for the meetings, are the only ones that all the members of Kopernicana are forced to use. This is really important for our project because the freedom that every member of an Holacracy organization has to choose which application use to work is a key point for our study. Now we will show several tools chosen by some of the members to achieve their goals.

### 3.2.2 "Free" tools

Our tutor and other dealers of Kopernicana, which manage the commercial part of the company, also work with Pipedrive<sup>4</sup>, a CRM used to keep clients and projects up to date but also where all the contracts of the organization's members are saved. To pay his members, as we said before, Kopernicana uses a particular method based on the amount of time spent working on each project of the organization and with

---

<sup>2</sup><https://clockify.me/>

<sup>3</sup>[www.holaspirit.com](http://www.holaspirit.com)

<sup>4</sup>[www.pipedrive.com](http://www.pipedrive.com)

a different hourly rate for each member, until now everything is calculated with the “Operating System Kopernicano”: a Google Sheet created by our tutor, it also extract some important data about the finance of the organization, but it has some gaps, the most critical one is that is not automatic, and all the data have to be inserted manually by our tutor, they don’t arrive directly from the other tool like Clockify or Pipedrive. The improvement of this module of the company will be our implementation part of the project. It is described in Chapter 4

During the tactical and governance meetings, we also saw several applications used by all the other members. Some members use Miro<sup>5</sup> to create interactive boards where they insert notes, ideas and use it also for some projects with clients. with these aims also Notion<sup>6</sup> is used in Kopernicana. It has been discovered at the beginning of our internship by a member and it has been shown during a meeting as an application that could improve the work-life of Kopernicana. This is a normal workflow in the company: if a member discover a software that could be interesting for the organization, he will show it to the others and then, after testing it for some time, the organization decides if use it or not. As cloud platform for the storage of the documents, both Google Drive, OneDrive and Dropbox are used by the organization’s members. To create contracts and offers Pandadoc<sup>7</sup> is used. Then, to communicate, as we said, Zoom is used for all the meetings, not only the internal ones but often also to do meetings with clients. At the beginning of our internship, only Whatsapp was used as a messaging tool but then, (in the same way explained for Notion) Slack has been tested by the members and after passed the test phase, it is daily used in the organization.

### 3.2.3 Problems

It’s clear that really a lot of applications are used together and of course, it could be a problem not only for the inconvenience to work with so many different tools but, above all, because the interactions between all these software are minimal or even zero. Minimal interaction means that there is no automatic data transfer between applications. For example, both Clockify and Pipedrive manage projects, but if a member creates a new project on Pipedrive it doesn’t automatically appear also in Clockify, it has to be manually added in it and in all the others applications that manage the object project (Operating System Kopernicano for example). This is a big problem because when an object (projects, offers, deals, contracts or documents) is added in two or more different platforms is easy to do error, insert wrong information or duplicate them.

In the beginning, in a small company composed by eight people, the use of all these different software, sometimes also with the same functionalities (OneDrive, Google Drive and iCloud for example), may seem wired and is normal to think “why don’t they agree on what to use?” It was also our first thought, but the answer is easy: this is Holacracy, everyone is free, to achieve the best result possible, to follow the way that seems better for him or her and of course also the software that he or she prefers. This has been already explained but is really important to

---

<sup>5</sup>[www.miro.com](http://www.miro.com)

<sup>6</sup>[www.notion.so](http://www.notion.so)

<sup>7</sup>[www.pandadoc.com](http://www.pandadoc.com)

highlight in order to better understand how and why the new application, shown later in section 3.6, is designed. After we got in touch with the use of the members under an IT point of view, we started to think about what we could do in order to improve the organization and facilitate the work to the members.

### 3.3 How can we help?

From the first month in Kopernicana, the IT problems that have been found and that came up during the meetings are mostly related to the difficulty to work with such a high number of different applications, more than one member complained about continuing switch application depending on the task he has to do, they said that often they have to work with more application in the same time and for sure this is not comfortable at all. Others lament about the difficulties in finding the information they need. If someone needs a specific document or some info about a project or a client its could be in more than one application and inside an application could be in a different location; remember everything is not easy. At the beginning to solve these problems, create a new application with the most important features of each application that now are used was the idea, this to put all together the functionalities, give the possibility to the members to work with only one software and to merge all the data in only one database. This could be really interesting since the application will be created on the ideas and specific of Kopernicana, but, it has no sense for other reasons:

- *How many applications are used now by the members?* We will see later with the survey in section 3.4 the exact number, until now we can say that for sure are many, at least ten.
- *Is it useful to reprogram features that already exist in other applications?* Maybe, but it needs really a lot of work.
- *Are we able to integrate all the features of these applications into only one?* Definitely not! It's impossible to create a platform with all the different features.

The answer to the last question is a key of our project, from it we understood that the biggest problem to program a completely new application is that we will never have the same power and functionalities of the platform that already exists since a lot of years, that are used by millions of user around the world and that are created by big IT companies. For example, how can we program an application with the same functionalities and power of Google Drive and Zoom? This is definitely out of our hands.

These are technical problems, but let's also think about the following ones, in an Holacracy organization everyone is completely free to work using what is better for him/her, and is also free to try and test new software (as it happens for Slack and Notion during our stage), and what can be done if after we finish to insert in our application the functionalities of the new tested tool, a member starting to use a new one with new features? We would have worked for nothing. Another big problem: we want help Kopernicana, but we also would like to create something that could be sell in the future to other Holacracy organizations or maybe to small startup with few employees; again, which software is used by other organization? Which features do they want to put in the application that should be implemented? Impossible to know for all the organizations.

Let's review the situation, we would like to create a tool that merges the data coming from several applications, but these applications are not fixed since they

come from the needs of the users; hence something that is flexible and adaptable to more applications is needed. Is there any way to do this without replicate all the applications into only one?

Using API is possible to retrieve data from an application and some time is even possible to embed some functionalities of the app in another one. This is what we were looking for! We did some research and we found out that most of the tools we were studying provide APIs, so probably there was the possibility to continue following this idea. If we exploit the power of API, we could integrate really a lot of different applications extracting what we would like to use of each application and merge the data with the others one. In this way, the user is free to choose which application use to work and we don't have to reprogram anything that already exists and is already well colluded.

Now the questions are: How should the applications interact between them? Again, Which applications? What are the specific problems Kopernicana's members have and we have to solve? What API provides every app? How can we use the data of every API? is possible to embed some functionalities?

To answer the first three questions probably, we needed to spend some more time in the company to understand deeper these problems, while for the lasts we have to do some detailed researches and studies for each application. Unfortunately, due to the Covid-19 situation and the way of work of the team (not all in the same office), we didn't have the possibility to stay closer to members during their workdays and get more input from them. Therefore, to better analyze the situation, the problems, the difficulty of everyday work life, a survey to all the Kopernicana's members is the best choice possible. The last two questions regarding API are studied in section 3.5



## 3.4 The survey

A survey using Miro has been created, it was split in two separate part, the first one is interactive, each member had to fill up a grid using virtual post-it, to tell us which tools he uses and which are the most important for him, while the second one is composed by three open-ended questions about problems and necessities of members.

### 3.4.1 The interactive question

In figure 3.1 the empty grid on the left with the “post-it” on the right.

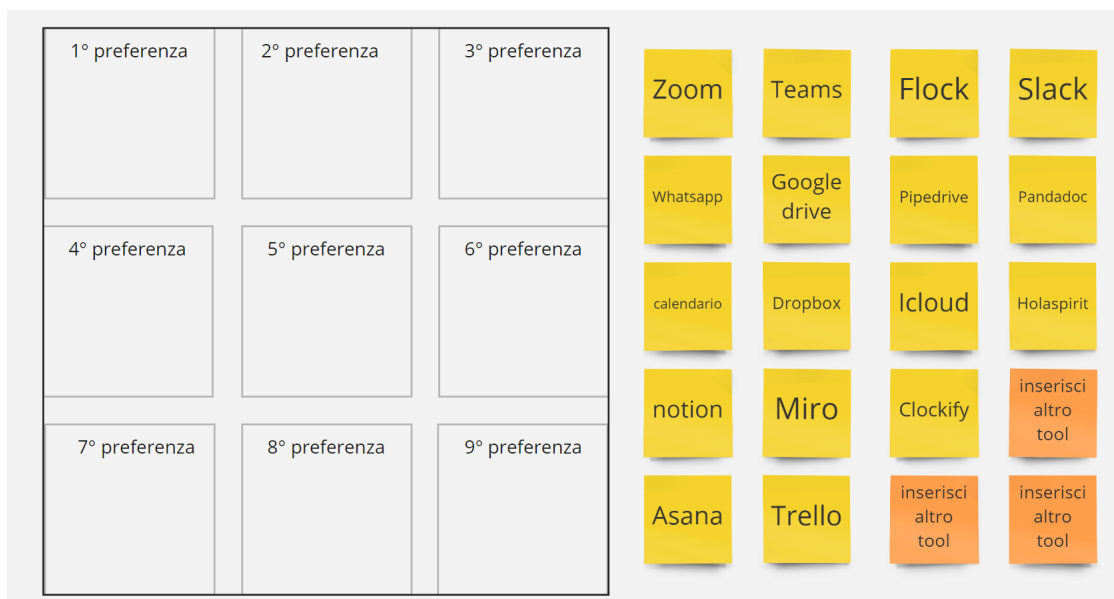


Figure 3.1: The empty grid

The members had to move each “post-it” (if they use that tool) on the position they want according to the importance they give to each tool. For example, if for me Slack is the more important tool (in terms both of the time of use and task are done with it) to do my jobs, I will put slack in the first square of the grid and so on. Some post-it has been created with the tools we know they use, while others are left “free” to give the possibility to add more tools. The order of the tools used in the organization is important since from it it’s possible to extract some interesting information on which tool is more relevant for the members or if there are some specific functionalities that are used more than others, this to understand if it is necessary to focus on some features or not. At the same time, a complete list with all the platforms used is important to start to study the API of these applications in order to create the understand what is possible to do for each software and then starting to do a prototype of the application. The aim of asking to insert all the tools used is also to try to split them into different groups according to their functionalities.

### 3.4.2 Open ended questions

The second part of the survey consisted of three open-ended questions:

1. *What are the problems or difficulties you have working with such a large number of different platforms?*
2. *Which of these platforms would you like to merge so that data is shared between them?*
3. *Are there any platforms that you currently consider indispensable to carry out your daily work? Why?*

Obviously, this part with the open-ended questions is important for us to have more specific information than what we understood in the first month about problems and needs of the members, this to decide how to implement the application, which data merge together and at the end to create a structure that can really help the organization. The survey has been done only to all the Kopernicana's members. Since the experience of the members in this kind of organization is really a lot and their answers came from several years spent studying the holacracy world, this survey can be used as a reliable baseline to start to design the application.

### 3.4.3 Result

The survey has been submitted to all eight members of Kopernicana; six of them compiled the first part, four of these six also answered the open-ended questions. Six out eight is enough to extract trustworthy data from the grids, while for the second part, even if maybe four out of eight is not the best result possible, the answers of these four members are very detailed and we found really a lot of interesting information in it.

Let's start to analyze the first section of the survey.

### 3.4.4 The grid

The maximum number of applications utilized by one member is 20 (really a lot!!), the minimum is seven and the average is 13.3. This list is the result of the first part. We gave a value for each position and we sum the points of each tool from all the grids (one for each participant).

1-Google Drive(64)	9-PipeDrive(37)	17-OneDrive(11)
2-Zoom(62)	10-DropBox(36)	18-Lark,whereby(10)
3-Notion(61)	11-Slack(34)	19-Asana(9)
4-Miro(54)	12-PandaDoc(30)	20-iCloud(7)
5-Calendar(51)	13-Teams(23)	21-Adobe CC(6)
6-Holaspirt(43)	14-Gmail(15)	22-WeTransfer(5)
7-Clockify(42)	15-Meet(15)	
8-whatsapp(41)	16-Google Sheet(13)	

First of all, we didn't expect such high number of applications for user and we thought that they put in the list some applications they rarely use, hence we check every grid and what we saw is that these (in the following list) are the last tools insert in the grid by each member:

-Holaspirt    -Holaspirt  
 -PandaDoc   -Slack  
 -Holaspirt   -iCloud

Other than iCloud, which could be used less, maybe only to save some documents or as a second cloud platform, the other apps are used for sure more than once a week. Holaspirt is used at least every Thursday for the meeting by all the members, but probably more to check project, activities and info about the organization, Slack should always be online to be updated and also PandaDoc, which is used to create contract and documents for clients, if it is used by a user, it is used more than once a week; hence all of these are used several times for a week. This is really important, and it means that we can't focus on the top of the list, but we need to consider all the applications that have been inserted in the grid by the members because all of them are often used. Then, we tried to split the applications into different groups, one for each kind of application. These are the groups we identified:

#### **Document Management**

-Google Drive  
 -Dropbox  
 -OneDrive  
 -iCloud

#### **Holacracy tool**

-Holaspirt

#### **Calendar**

-Google Calendar

#### **CRM**

-Pipedrive  
 -Pandadoc

#### **All in one workspace**

-Notion  
 -Miro

#### **Project management**

-Asana  
 -Trello  
 -Monday

#### **Social collaboration**

-Slack  
 -Flock  
 -Whatsapp  
 -Lark

#### **Video communication**

-Zoom  
 -Teams

#### **Time tracking**

-Clockify

#### **Mail**

-Gmail  
 -Outlook

We have divided 22 apps into 10 different groups and if we came back to the list and then underline the first app that appear for each group in the list

<b>1-Google Drive</b>	<b>9-PipeDrive</b>	17-OneDrive
<b>2-Zoom</b>	10-DropBox	18-Lark,whereby
<b>3-Notion</b>	11Slack	19-Asana
4-Miro	<b>12-PandaDoc</b>	20-iCloud
<b>5-Calendar</b>	13-Teams	21-Adobe CC
<b>6-Holaspirit</b>	<b>14-Gmail</b>	22-WeTransfer
<b>7-Clockify</b>	15-Meet	
<b>8-hatsapp</b>	16-Google Sheet	

Is possible to notice that except from Miro with Notion, Dropbox with Google Drive, Slack with Whatsapp and Teams with Zoom, even if all of these couple have some specific different functionalities (maybe we should also split this group but we don't want to create too many), until the fourteenth position all the applications are in different groups. What does it mean? that all the groups have the same importance and that the media number of applications used by each members (do you remember? 13.3) has sense, in average each member use at least one tool for each group.

Since the beginning we didn't want to give priority to some applications but now is clear how all of them should be included in our new software.

### 3.4.5 Open-ended questions

Now that we have a clear idea about the applications used by the members, let's study the open-ended questions.

- *What are the problems or difficulties you have working with such a large number of different platforms?*

1. "Duplication of files and documents."
2. "Not fully integrated platforms."
3. "Limits in sharing materials and files (especially notions)"
4. "Lack of a repository of all materials available to all."
5. "No summary representation of what's going on in the organization."
6. "Too much data/replicated information."
7. "Few connections between the various systems."
8. "Too many accounts, passwords."
9. "Too much scattered material."
10. "Slowing down the machine."

11. "Need to work with at least two screens."
12. "You have to remember whom you're working with and on what platform."

There are twelve answers from the survey, an average of three for each member. First of all, the problems have been split into two different groups: the problems that are possible to solve with the new ERP (from 1 to 9) and those who can't be solved with the application (10-11-12) because are hardware problems or not related to the use of different applications.

As expected, some problems are found by more members and can be split into three big clusters. Numbers 2-7-8 are all related to the few connections of the tools, numbers 1-3-4-6-9 are related to bad management of the documents, there is not a single repository and is difficult to know where a specific file is located, while number 5 is a problem about visualization of important info about the organization (the financial part especially). These problems have been already founded these during meetings but reading them also in the answer that member gave to the survey confirm that the feeling we had about the difficulties and problems of the organization were true. All these problems will be solved with the new application that will be designed, and the last one will also solve practically since the implementation of the accountability of Kopernicana has been executed during the internship.

- *Which of these platforms would you like to merge so that data is shared between them?*

1. "Notion + Slack + DropBox + Gmail + Mirò + Google Drive"
2. "**Clockify + Holaspirit + Google Drive**"
3. "Notion + **Google Drive + Holaspirit**"
4. **PandaDoc + Pipedrive + Clockify**"
5. "**Google Drive** con i "gestionali" come **Pipedrive, PandaDoc e Clockify**"

As is easy to see, most of the answers included Google Drive, the first necessity in Kopernicana, as it has been founded out also in the first question, is to improve the document management workflow. We also understand that will be important to merge the financial part; in fact, Clockify (where they keep updating the working hours), Pipedrive (CRM) and Holaspirit, often appear together in the answers. We are happy about these because it is a need that our co-supervisor has and told us since the beginning of our experience, we didn't know that is also a request of other members, this is important for us because it means that we will solve (always with the automation of the "operating system Kopernicano") a problem of more than one member.

- *Are there any platforms that you currently consider indispensable to carry out your daily work? Why?*
1. “Notion, Slack, Miro, Zoom, Google Drive.”
  2. “Zoom, Pipedrive, PandaDoc, Google Drive.”
  3. “Google Drive, Whatsapp.”
  4. “Gmail, Google Drive, Slack, Notion, Calendar, Zoom, Miro, Dropbox, iCloud, Clockify, Whatsapp.”

Going back to the group list and underling the platform of the answers:

**Document Management**

-*Google Drive*  
 -Dropbox  
 -OneDrive  
 -*iCloud*

**Calendar**

-*Google Calendar*

**Holacracy tool**

-Holaspirt

**CRM**

-*Pipedrive*  
 -*Pandadoc*

**All in one workspace**

-*Notion*  
 -Miro

**Project management**

-Asana  
 -Trello  
 -Monday

**Social collaboration**

-*Slack*  
 -Flock  
 -*Whatsapp*  
 -Lark

**Video communication**

-*Zoom*  
 -Teams

**Time tracking**

-*Clockify*

**Mail**

-*Gmail*  
 -Outlook

Two really interesting points came out from the table: Holaspirt that, as we already said, is an indispensable platform for an Holacracy organization as Kopernicana, has not been mentioned in this answer. Probably the members are not completely satisfied by this tool; this should keep in mind and could be an interesting implementation in the future. The second is that except for “project management” all the other groups contain at least one application indispensable for some members, hence we have to add all of these different kinds of applications in our software.

## 3.5 The API provided by each tool

As regards the API, different platforms have been studied, not only the ones came out from the survey, but also some others, to understand if there are some similarities in the API provided by applications of the same type.

In table 3.1, the tools are ordered by groups, firstly the video conference tools, then the calendars, and so on. The most important aspect is that except for Notion (that will public its API in the future <sup>8</sup>) all the applications that have been checked provide API. Hence all of them could be included in the new innovative ERP. There are two different kinds of applications, the ones who managed personal or detailed information as the CRM (work with clients and deals data) or Holaspirit (used to store information about the organizations and members) and it's more important the information itself than how it is shown and others applications that works with dashboards or specific functionalities (for example a calendar or a Miro board or a zoom video-conference) where it's really important how this information is shown. From the table 3.1 is possible to see that the first type of applications (CRM, Holacracy tools and time tracking software) provides only API to get and post data, while the others, that belong to the second group (video conference tools, team collaboration, project management tools and so on) also provides API to embed some specific functionalities or boards. This is really important since it gives the possibility to extract data that we need from the first group of applications and embed the functionalities of the second group of applications where the features are more important than the data. Another important characteristic that came out from the table is that all the application that belongs to the same group provide the same type of API and use the same type of data. For example, all the CRM APIs return a JSON file, the same happens in the time tracking software and for all the other groups. Also, to embed, except for the video conference tools or messaging platforms that need some specific HTML function (check the Zoom example <sup>9</sup> only a script of HTML code is needed for all the others. In this way, as it's described in the next section 3.6, to design the application, it is possible to focus on the type of platform that has to be integrated and not on the specific one. It means that is not important the specific application but the specific functionality of it, hence it will be possible to create a defined structure following the functionalities and the data and not the applications. This is what we were looking for, an application that can integrate several, not fixed applications but with a defined structure. Also, some other tools have been quickly checked and hence not inserted in the paper; anyway, almost all of them have the same characteristic of the ones in the table, so it is realistic to think that this new software can integrate many more applications than the ones in the table.

---

<sup>8</sup><https://www.notion.so/Does-Notion-have-an-API-I-can-use-4541b07a5caa46dba0026624646118c0>

<sup>9</sup><https://github.com/zoom/sample-app-web>

## 3.6 The PONGO application

Merging what we understood from the experience in Kopernicana, from the survey and the possibilities offered by APIs the idea of our application was beginning to take shape. This is a nice pun because the shape is something that the application should not have; our application should take the shape of the users' work-life; it should be like a pongo. What does textpongo mean? Pongo is a modeling plastic that is used by kids to create whatever they want; the main characteristic is that is possible to change the shape and add or remove some material at any time. This is what we would like to do also with our applications. The users should be able to modify the structure as freely as possible and add or remove an integration with a platform at any time without having problems. This is possible thanks to the APIS explained before, now the architecture of the new ERP will be shown..

### 3.6.1 The architecture

The new ERP will be a web application The application architecture is a three-tier one composed of a presentation tier, an application server and a database. An API management system will use API to extract the data we need from the applications and fill the database. This architecture has been chosen due to its independence between the three modules. In this way, they can be updated or modified without problems. Let's see how the presentation layer has been designed.

To show the idea, we think it is easier to start with the structure of the application.

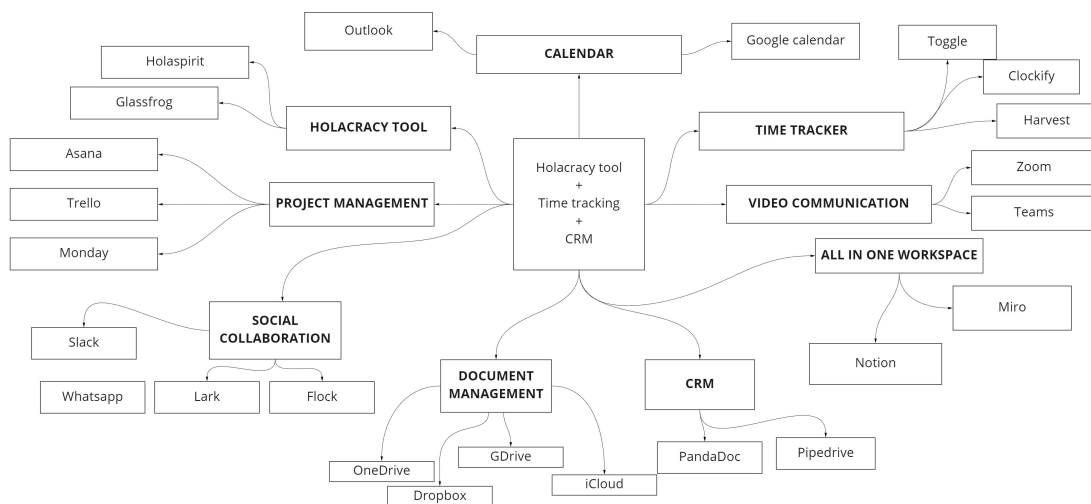


Figure 3.2: The structure of the application

In Figure 3.2, there is a baseline box composed of an Holacracy platform, a time tracking and a CRM. These three groups of applications have been chosen as the base of the structure because:

- An Holacracy tool is mandatory to manage an Holacracy organization
- A CRM nowadays is necessary for every organization
- If an organization (as Kopernicana does) pays its member according to the numbers



of hours worked for each projects, a time tracking software is a must-have for it. We put them in a unique box because, as it turned out from the survey (opened question 2), these applications should communicate and merge data between them. These kinds of applications manage data that are really important for the organization and should not be errors or duplicates in it. For example, it's really important that the same project has the same name and details both on the CRM and on the time tracker, we don't want two different names for the same project or deals in two different tools, if we add a new project during a meeting managed on Holaspirt we would like to open it also in the CRM and add even in the time tracker, so that is possible to create a working time on that specific project. Around this box, all the other groups of applications could be inserted in the ERP at the user's convenience. Not all the tools and the application groups that we included in the schema must be in the app; we create the structure to insert all the different group of applications that have been studied in this project, then each organization that will use this software will be free to choose what to integrate or not. This is the first advantage of this application. Another note that has to be made is that we know there are more applications on the market for each group that can be used by the users; as mentioned before, here only the tools that are studied in detail have been inserted. Anyway, we know that also the other tool with the same functionalities provides similar APIs and can be easily inserted in the application in the future. To structure the application, we started from a baseline box in which we put an holacracy tool, a time tracker and a CRM.

### 3.6.2 The back-end

The data will be extracted from the apps by APIs and insert them into a database where is easier to have a general view of all the data of all the apps and from which is easier to create connection and links between the applications. In figure 3.3, it is possible to see the database that has been designed to host all the data of an holacracy organization coming from the three different applications of the box.

Is important to have a database because, as possible to see from the ER schema in the figure, all these entities have a lot of connection between them, without a database that merge the data, coming from a different platform, would be impossible to create these connections and use them in the application. This will give to possibility to the user to continue to create and modify data in the applications that they already used before, where the user interface is designed for that specific operation, while the ERP will merge them and show to the user within all the others data.

Hence, the database layer has been presented, now the application layer that made possible the interaction between the external applications and the database will be shown.

In table 3.2 is possible to see for each entity of the database, all the tools that could be used to manage the data by the organization and the relative API that should be used by the application layer to extract the data to fill our database.

Let's see an example to better explain the table and how the data are extracted from the applications and inserted in the database. Take the entity member; in

an Holacracy organization all the information about a member are saved into an Holacracy tool, that can be Glassfrog or Holaspirit, which one is used by an organization is not important for the application, since with APIs it's easily possible to extract what is needed from both the tools. From an holacracy tool is possible to extract data about the organization, members, role or action. For all of this objects both the applications provide API to extract the data that will be used by the application layer to fill the database. Here is the pongo, any organization is free to choose the best tool to work with, depending on what they think is better, and we extract the data and merge them together. The user only have to connect our application with his account of the tools he uses and then the API management system, that knowing for each tools which API use, will manage the data.

### 3.6.3 The front-end

After explaining how the database layer and the application one are structured, let's analyze the presentation layer. As has been explained in section 3.5, there are two different kinds of applications that the application works with, the ones that provide only API to extract data and the ones that provide an HTML script to embed some functionalities. With the first group, we fill the database, while for the second one, we only need to insert the script where the user wants. To better understand the flow, we start to analyze how we use the data from the database, and then we add all the other groups subsequently. In figure3.4 the flow of the application. In yellow the pages, while in green the entities of the database used to dynamically fill them. For example, with the entity member, we will create the members' page and then clicking on one of the members, the details page will be shown. All the data will be retrieved with a query in our database In this way, we managed the CRM, the holacracy applications and the time trackers; how do we interact with the others?

Continuing with the pongo idea, so give the freedom to the user to model to one's liking, the applications can be embedded in several different pages.

In figure 3.5 is possible to see the different possibilities. Of course, functionalities like Calendar or Mail are fixed in their pages, while tools like a social collaboration or document management could be embedded in different pages. Let's take a document or a folder as an example. A document could be an offer for a client, a project idea, a contract of a member; therefore in all these pages (client, project and member) should be given to the user the possibility to embed a specific object, hence there will be a button to add documents. An example (Figure 3.6) of a project details page has been created using Adobe XD<sup>10</sup>, in that page a Google Drive folder a Trello notice board and a Miro board have been embedded.

This is the HTML code that has to be inserted in the code of the page to show the Google folder.

```
<iframe src="https://drive.google.com/embeddedfolderview?
id=1GEaUJn9ZjRj7AV00WZW2bWeU_76czWLh"
style="width:600px; height:600px; border:0;"></iframe>
```

<sup>10</sup>[www.adobe.com/it/products/xd.html](http://www.adobe.com/it/products/xd.html)

These embedded objects are also link to the pages, so clicking on them a new page in the browser with the mother application will be automatically opened. In this way, all the members of an organization can easily find all the object related to this project, and who has the rights can modify them in the application.

### 3.7 The manifesto

This new application that has been designed will help all the holacracy organizations to have data of several applications used, merged and shown all in one workspace. This will be a big improvement in the way of work and in the economy of the company. Of course, another important feature of this new ERP is that it could be use not only by the holacracy firms, but also by every company that wants to have more freedom on the tools its members can use to work. The aspects that make our application innovative and different from any other ERP already existing on the market are summarized here in three big points:

***Free choice:*** *"individual at the centre" not "processes at the centre".*

What does it mean? The application gives priority to the members (the individuals) of an holacracy organization and not to its processes. The members are completely free to work on the tools that they think are better, use all the functionalities of it, and then the data will be integrated into the app. Usually, the opposite happens: the processes have the priority, hence the members are forced to use a specific tool that do some specific functionalities because the data are saved on it and can't be extracted. That is the first and more important news of our application that gives the possibility to holacracy companies, or better, to all the company that would like to use a different platform to merge data and information that, without this new software, would be spread.

This is a great innovation for a lot of companies. At the moment, where the working world constantly changes and the remote working is more and more used, having the possibility to see data coming from applications that are used by other members all in one web application is something that will save a lot of time and makes the members of the teams of firms closer between them even if they work from different places. For example, a member is working on a new idea using a Miro board; if he would like to share it in a normal way, it has to send the sharing code using a messaging tool or an email to all the other members, which have to enter in Miro (hoping to have the right permission) and check it. While using the new ERP, the member has only to connect the board with the application; then it is visible to everyone that is already connected to that application without do nothing more.

The second features that makes our application innovative is:

***Intelligent integration:*** *"integrate for tension (when needed)" not "integrate everything (always)".*

"Tension" is a key world for holacracy; during a tactical meeting, when a member opens a tension means that he has a problem, news or everything that he wants to discuss with the team. "Tension" is also called necessity and this is how we integrate the applications and the data: around the necessities of the members. As explained before, the user is free to insert an object of the other application where he wants. Also, all the connection of the data coming from the API are designed based on the necessities expressed from the members in the survey (in particular in the second open-ended question).

This is also really important; sometimes there isn't the necessity to integrate everything, or better, a lot of things are integrated in a bad way and all together. In this new application is integrated and merged together only what the user really want to integrate and share with the team, in addition to this, the object that has to be merged can be connected in several different sections of the application. The last change that this ERP brings to the market of the work applications concern the payment of the tools, that often is a major impact on the expenses of a company:

***Payment 2.0:** "pay per use" not "pay per workstation".*

Usually, when an application is used by a company, it is paid according to the number of users that are registered. It doesn't matter if the application will be really used or not. With this new application, each member is free to choose which application use and will be registered only to the ones that really need. Hence the company will pay only the applications that are really used. This is a great innovation for companies that too often sign deals with application providers and pay the fees for all the members even if the application is used only by few employees of the organization. In this way, no more money will be wasted for unused applications.

<b>Tool</b>	<b>Has API?</b>	<b>Is possible to embed?</b>	<b>Type of data</b>
<b>Zoom</b>	YES	YES	Web SDK to embed the meeting board
<b>Google meets</b>	YES	YES	HTML code to embed the video in your application
<b>Google calendar</b>	YES	YES	HTML code to embed the calendar in your application
<b>Outlook calendar</b>	YES	YES	HTML code to embed the calendar in your application
<b>Notion</b> Team collaboration	NO	NO	
<b>Miro</b> Team collaboration	YES	YES	HTML code to embed a board in your application
<b>Asana</b> Project management	YES	NO	JSON
<b>Trello</b> Project management	YES	NO	HTML code to embed a board in your application
<b>Monday</b> Project management	YES	YES	HTML code to embed a board in your application
<b>Flock</b> Messenger	YES	NO	JSON
<b>Slack</b> Messenger	YES	YES	HTML code to display a slack channel into any webpage
<b>Hubspot</b> CRM	YES	NO	JSON
<b>Pipedrive</b> CRM	YES	NO	JSON
<b>Pandadoc</b> Document creator	YES	YES	WITH APIs is possible to create documents from templates
<b>Google drive</b>	YES	YES	HTML code to show a folder or a document
<b>Dropbox</b>	YES	YES	HTML code to show a folder or a document
<b>Onedrive</b>	YES	YES	HTML code to show a folder or a document
<b>iCloud</b>	YES	YES	HTML code to show a folder or a document
<b>Holaspirit</b> Holacracy tool	YES	NO	JSON
<b>Glassfrog</b> Holacracy tool	YES	NO	JSON
<b>Clockify</b> Time tracking	YES	NO	JSON
<b>Harvets</b> Time tracking	YES	NO	JSON
<b>Toogle</b> Time tracking	YES	NO	JSON

Table 3.1: API list

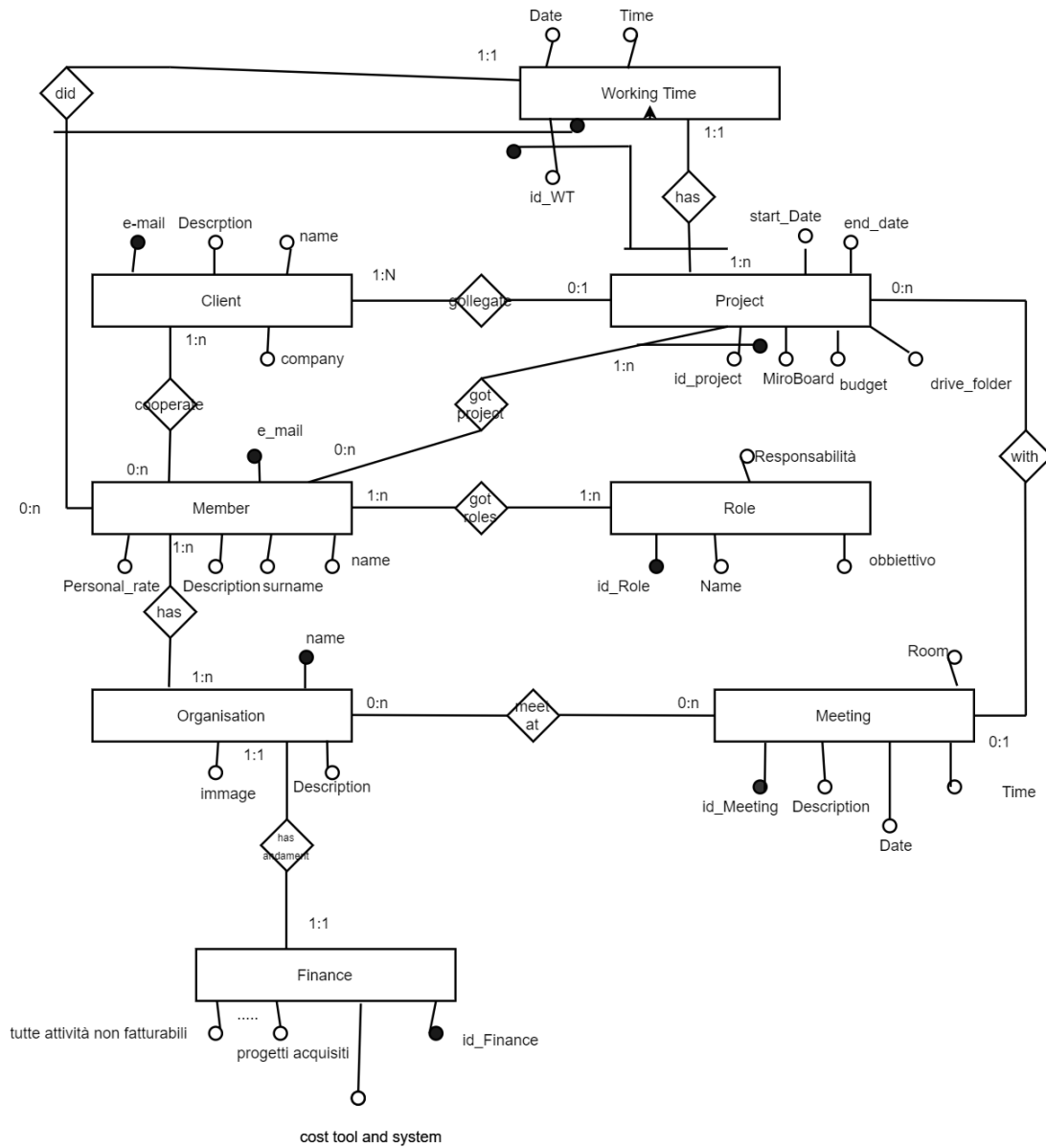


Figure 3.3: ER schema of the DB

Entity	Application	Type of data	API
Member	Holaspirt	JSON	-Display organization members
	Glassfrog	JSON	-List everyone in an organization
Client	Pipedrive	JSON	-Get all deals -Get details of deal
	HubSpot	JSON	-Get all deals -Get a deal
	Insightly	JSON	-Gets a list of leads -Gets a lead
organization	Holaspirt	JSON	-Displays an organization -Displays checklists
	Glassfrog	JSON	-Displays an organization -Displays checklists
Role	Holaspirt	JSON	-Displays roles
	Glassfrog	JSON	-Displays roles
Action	Holaspirt	JSON	-Displays projects -Display actions
	Glassfrog	JSON	-Displays projects -Display assignments
Project	Pipedrive	JSON	-Get all deals
	HubSpot	JSON	-Get all deals
	Insightly	JSON	-Gets a list of leads
OKR	Holaspirt	JSON	-Get the okr
Meeting	Holaspirt	JSON	-Display tactical meetings -Displays meeting's tension
	Glassfrog	JSON	-Displays governance meetings -Displays meeting's tension
Working time	Clockify	JSON	-Get time entry on workspace
	Toggle	JSON	-Get time entry details -report API for complete report
	Harvest	JSON	-Time entries. -Report API

Table 3.2: How we get the data



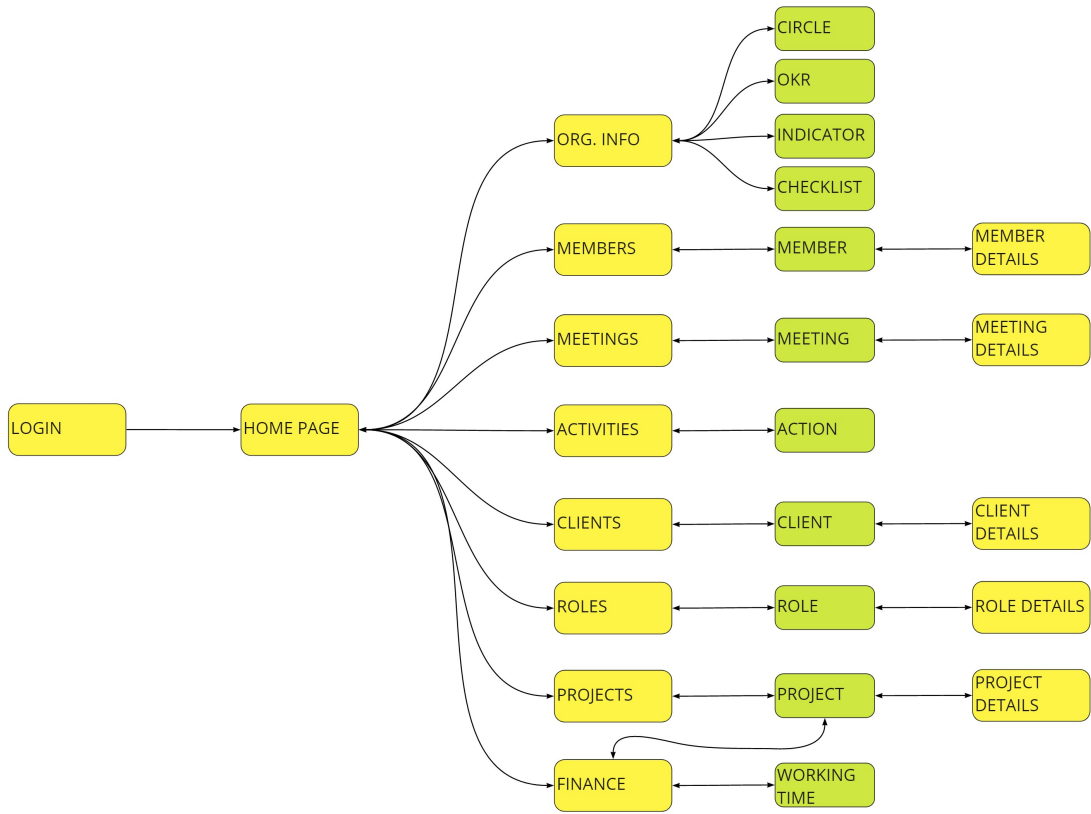


Figure 3.4: Flow of the application

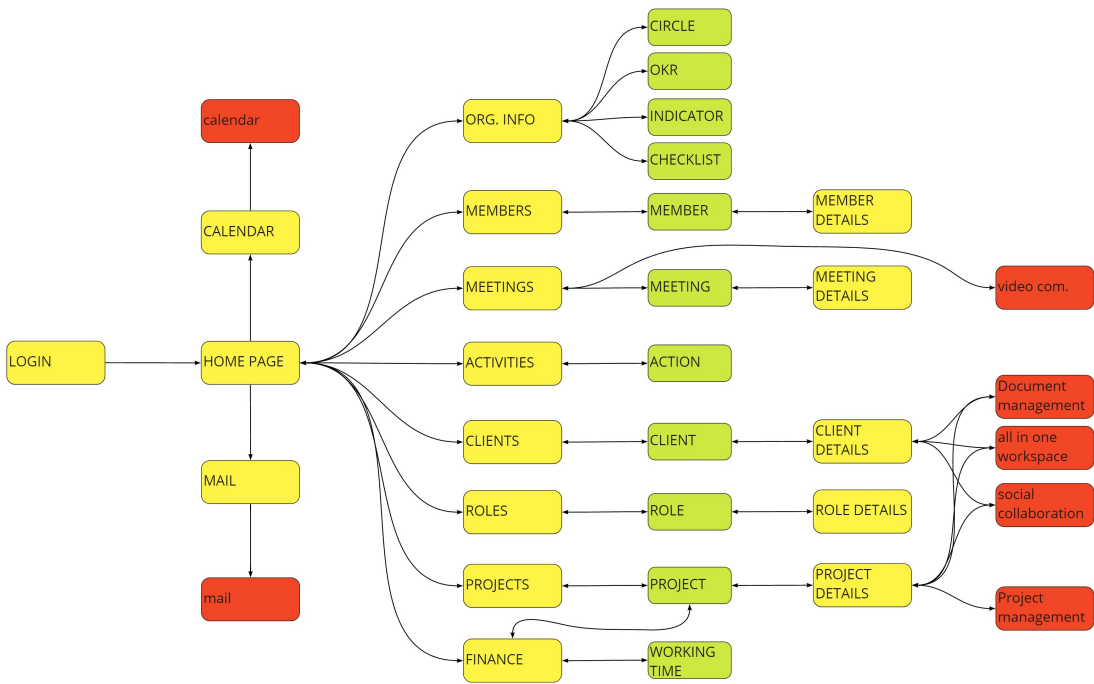


Figure 3.5: Complete flow

**XOPERNICANA**

**Musement - OKR** Owner: Francesco Frugluele [View for yourself](#)

Organizzazione di due incontri con dipendenti di Musement sul tema OKR

**Miro board**

**Google Drive folder**

TITOLO	ULTIMA MODIFICA
Musement-OKR	21 apr daivide moreno
Musement-Presentation	21 apr daivide moreno

**Trello notice board**

Card title	Is completed	Info
Meeting 15/3	Creazione presentazione dipendenti	Done meeting 15/3
Presentazione per dirigenti		Invia commenti

[Add to page at home](#) [Details](#) [Modify](#)

Figure 3.6: Example of a project details page

# Chapter 4

## Implementation

This chapter aims to explain the practical work done in the internship for Kopernicana, in addition to the one already explained in chapter 3, where the design work of the new application has been shown. This work follows the design of the application and regards the automation of the accountability system of the organization. The connection between this system and two web applications (a CRM and a time tracking software) have been created to merge the data coming from them. The system that has been improved is the Operating system Kopernicano. As explained at the beginning of the thesis, the operating System Kopernicano is a google sheet where all the data coming from the CRM (Pipedrive) and the time tracking app (Clockify) are merged to extract the values of the salaries of the members and some interesting data for the company. Consider developing and implementing all the application designed in the methodology chapter was impossible, for lack of time but also for lack of people. It would take a lot of working hours, and it maybe will be ready in one year or more. This work could be done by the next internship in Kopernicana or by some professional developers if the organization would like to invest money in this new type of ERP.

Hence, why the Operating System has been chosen to be improved and not another part of the information system of the company? First of all, it was the first information component that the tutor showed to us, so our attention since the beginning was on it. Also, there was a specific request by the tutor to make some improvement in it and from the survey 3.4, done to the members, turned out the necessities to automatized and merge different applications as Clockify and Pipedrive to facilitate the work in Kopernicana. All this suggested focusing our attention on the Operating System Kopernicano.

In the first section, 4.1, the main components of the system will be shown to explain how they work and what they are used for in Kopernicana. Later, in section 4.2, the old system will be discussed to extract the weaknesses and strengths. Next in section 4.3, two new tools used to automate the new system will be presented. In sections 4.4, 4.5, 4.6 and in 4.7 the work done will be described in detail. In the last section 4.8, to conclude the thesis, a complete overview of the system, with the new workflow, will be done. To clarify a bit, some words will be used in the chapter, the *system* is the complete part that will be implemented, composed by the *external application* (Clockify and Pipedrive) and the *Operating System Kopernicano* (OSK) that is the group of Google Sheets used.

## 4.1 Pipedrive and Clockify

This section is not to make a tutorial of the external applications used by Kopernicana; it would not be interesting and would be useless for the work. Here only some features of these external applications will be shown due to their importance in the work of the organization and to better understand how the system works.

### 4.1.1 Clockify

Clockify is the time tracking software of Kopernicana. It means that it is an application used to keeping track of the worked hours of all the members of the organization. Each member has to add all his working hours into the application to keep update the total amount of worked hours and at the end of the month, be paid with the correct salary.

Two details of this way of work have to be understood before proceeding with the description of the work done. First of all, is not a method used by all Holacracy company of the world, Kopernicana decided to use this method since it gives to the members even more freedom, but also the "normal" method of payment with a standard contract is allowed in an Holacracy organization. The second error that a lot of people could do reading this section is to think that asking to the members to insert each hour of work with all the details of what they did (this part will be explained better later) is a way to control the member; is not like that, but the opposite, each member has not a working time to follow, each member is free to work where and when he wants, so the freedom is maximum, of course, if someone noticed that a members insert a lot of hours and his work doesn't match what's being inserted it's possible to ask for clarification; this could be done by everyone to everyone, hence if I'm the last arrived I can easily ask the founder of the organization clarification about his working hours. The holacracy idea of a non-hierarchical organization is followed in every aspect of Kopernicana's life.

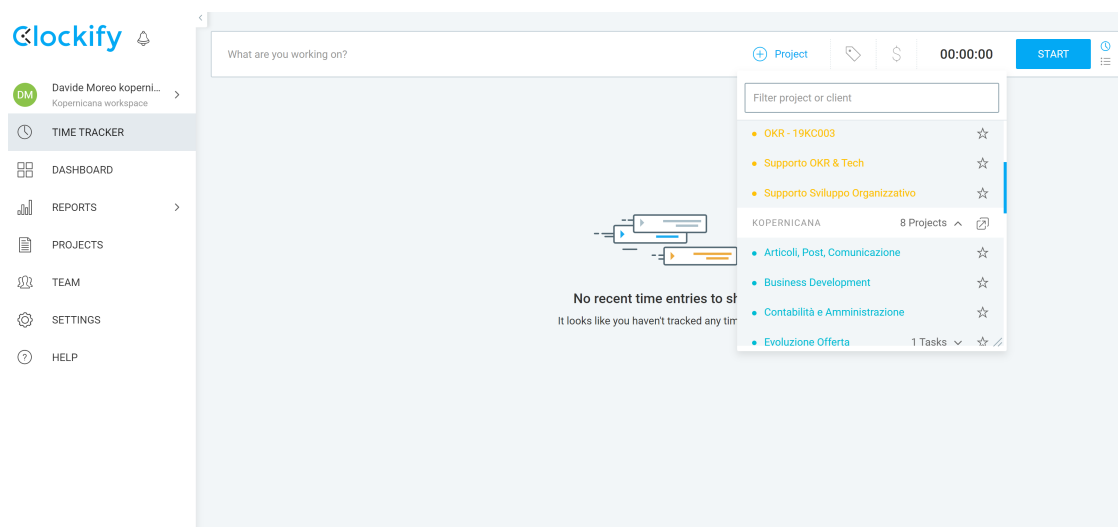


Figure 4.1: Clockify home page

In figure 4.1, the home page of Clockify is shown, it's possible to see the user-friendly interface and how the members insert their hours. Each member has to specify, from left to right of the bar at the top of the page, what he worked on, the project and the total amount of the time spend on that work. The project can be chosen from a list of projects inserted previously in the application. Until we started working on the system, these projects (that are the projects Kopernicana is working on at the moment) are manually inserted in the application by a designed member, of course, this is a problem that has to be solved. The manual insertion makes error very likely and also replication and misalignment with other applications that manage projects. Another important feature of this application that is used by Kopernicana is the report one; the report gives the possibility to the user to extract the total amount of work of all members filtered in a different way (on members, on projects, for the specific project and so on). This extracted file, which could be a PDF, CSV, or an excel one, will be manually imported into the Google Sheet (Operating System Kopernicano) to do all the calculations needed. At the moment of our arrival in Kopernicana, the excel file was extracted from clockify to be imported in the OSK.

### 4.1.2 Pipedrive

As already said several times, Pipedrive is the CRM of Kopernicana and as CRM is used to keep up to date deals and clients with all their detailed information. When we started working for Kopernicana, this application wasn't included in the system and all the useful data had to be automatically merged with the other applications (Clockify and Google Sheet). In the picture 4.2 above, it's possible

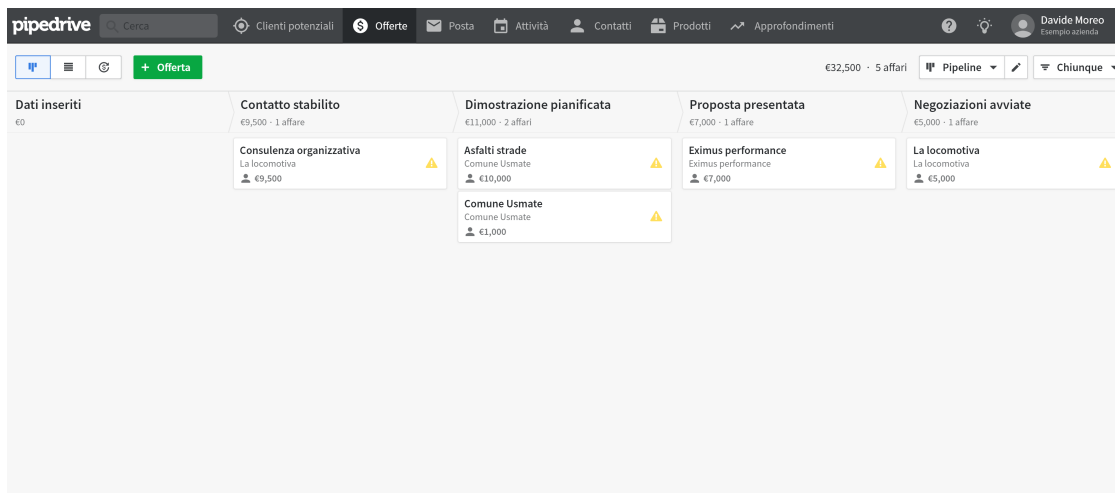


Figure 4.2: Pipedrive dashboard

to see the dashboard of the web application. It's important because it gives the possibility to see how the deals/projects are subdivided into the organization. As soon as the first contact with the customer takes place, the details of the customer and of the deal are inserted in the app, and a new deal appears in the first stage on the left called *contatto stabilito*, then as the negotiations near to conclusion, the

deal is moved in the others stages. The stages are important for our work because they will be used as triggers for some operations later. After all the negotiations, the deal could be won (if the contract will be signed) or lost (if the contract won't be signed) and it disappears from the dashboard, but it remains in the application and will still be visible to the users from others pages of the application. Pipedrive, as it has already said, was not included in the system before our work, it was only used to keep updated projects and clients, but everything was extracted and then inserted manually in the system. Since this application is the first place where all the deals or projects are written by Kopernicana it should be also the application that is in charge to send and update all the others applications that manage projects. Hence it will be really important to include Pipedrive in the system and create an automatic workflow for the deals. In the next section will be shown how this feature has been implemented in the system.

These are the only two web applications used by Kopernicana to manage the projects and the working time, that merged together, create the accountability part of the organization. The data coming from Clockify and Pipedrive are inserted, as we said, in the Operating System Kopernicano, a Google Sheet created by our tutor.

## 4.2 Operating System Kopernicano

The Operating System Kopernicano, as is understandable from the name, is the station that makes the calculus of the accounting of the organization possible. It is a Google Worksheet composed of four different spreadsheets:

- Actual*: a dashboard with all the parameters coming from the other spreadsheets, like the total amount of the projects, the total costs of billable projects, and so on. Here are also inserted manually the cost of all the platform used in the organization.
- Detailed report*: is a spreadsheet created importing a CSV file extracted from Clockify. As the name says, the CSV file contains a detailed report of all the worked hours inserted in the application by each member. To do this operation, the user in charge has to open the Clockify web application, extract the CSV file, save it in a folder of his computer and then import it in the spreadsheet.
- Summary of billable and non-billable projects*: is a spreadsheet with the three tables that will be shown later (4.3, 4.4, 4.10) and the cost for each specific projects. From here comes the data inserted in the actual spreadsheet. The name of the projects, the cost for each project, the budget hours of each member and all the others data except the total worked hours are manually inserted in the spreadsheet, while the total worked hours of each member are inserted using a SUMIFS() function over the *detailed report* spreadsheet.
- Personal fee*: here are inserted the personal fee of each member and is used to calculate the salary for the billable projects.

### 4.2.1 How does it work?

To understand how it works, explain how the salary is calculated for each of Kopernicana's members is necessary. Everything rotates around the idea of the project. A *project* is every work done with customers (Kopernicana is a consulting company, so each consultation is a project) but also all the internal activities of the organization. For example, the work done to find a new customer is a project too, called *prevendita*, all the activities that are done to improve what the organization "sell" are called *evoluzione offerta* or the meeting did in the organization are included in the internal project called *governance e gestione*. The projects done with the customers are marked as billable activities because the organization makes money with it, while the internal ones are marked as non-billable activities since the organization doesn't make money with them. Then, every member has its own daily fee, accorded with the founders when the contract with Kopernicana was signed.

The founders of the company, at the moment, decide not to have percentage gains on turnover, so every income is split among the members. When a member works for a billable project, it will be paid according to the daily fee of his contract. While for the non-billable projects, the daily fee is calculated summing the gain coming from the non-billable projects and split it by the total non-billable day worked by all the members. Here a brief description of how the salary is calculated has been done to better understand the following part of the thesis. If it is still not clear reading the following section, the idea behind the salary of the Kopernicana's members will be understood.

After this introduction in the Kopernicana accountability is easier to understand how the salaries are calculated and how each table is filled in the worksheet.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Project ID	Nome prog	Total	Owner	Fee sales	FF	AG	RL	JV	ED	GL	IM	MF	Total days	Members total cost	
2	1	Project 1	€ 40.000	Owner A	€ 4.000	15	10				13	3		41	€ 20.050	
3	2	Project 2	€ 15.000	Owner B	€ 1.500			7	1				4	12	€ 5.050	
4	3	Project 3	€ 25.000	Owner C	€ 2.500	10	5	6		10		8		39	€ 16.800	
5	4	Project 4	€ 20.000	Owner A	€ 2.000	7			2		11	6	7	33	€ 13.900	
6	5	Project 5	€ 13.000	Owner B	€ 1.300		2		6			8	3	19	€ 6.800	
7																
8																
9																
10																
11																
12																
13																
14		Total	€ 113.000		€ 11.300	32	17	13	9	10	24	25	14	144	€ 62.600	

Figure 4.3: Project-budget into the Operating System Kopernicano

The first two tables refers to the external project, so those that are billable since done with clients. Starting from the Project-budget table, in picture 4.3 it's possible to see the filled table. Before signing a contract with a client for a project, an established member is in charge of defining a hypothetical schedule of all the amount of hours that every member should work for that specific project. This to have an idea of the total hours that Kopernicana will work for that client and create the offer and also give an idea of the amount of time needed for the project. This amount is inserted in the table where it is possible to notice that for each project (here the real name of the projects are modified to respect the privacy of

Kopernicana) is signed, for each member (From FF to MF), a number of days. What does it mean? It means that the organization thinks that to conclude the project, it needs that amount of workday for that member. For example, for project number 1, Kopernicana supposes that FF will work 15 days, RL 10, GL 13, while IM only three days. The column "member total count" multiply each day for the day fee of each member (coming from another sheet). So, in the end, from the total income, removing the total member cost and the external cost (the fee sales and other costs that could be the trip for the customers, room rented for some meeting and so on) it possible to estimate the gain over each project. As already said, this is only hypothetical numbers of days that each member will work, used to estimate the final income. This table is manually filled by a member. The real amount of days worked for each project, coming from the CSV file imported in the worksheet, are inserted in the following table using an easy SUMIFS().

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Project ID	Nome prog	Totale	Owner	Fee sales	FF	AG	RL	JV	ED	GL	IM	MF	Tot worked days	Budget-days	percentage of completion	
2	1	Project 1	€ 40.000	Owner A	€ 4.000	10	7	0	0	0	12	3	0	32	41	78,05%	
3	2	Project 2	€ 15.000	Owner B	€ 1.500	0	0	3	0	0	0	0	2	5	12	41,67%	
4	3	Project 3	€ 25.000	Owner C	€ 2.500	10	5	6	0	12	0	7	0	40	39	102,56%	
5	4	Project 4	€ 20.000	Owner A	€ 2.000	1	0	0	1	0	3	2	0	7	33	21,21%	
6	5	Project 5	€ 13.000	Owner B	€ 1.300	0	0	0	1	0	0	0	0	1	19	5,26%	
7																	
8																	
9																	
10																	
11																	
12																	
13																	
14		totale	€ 113.000		€ 11.300	21	12	9	2	12	15	12	2	85	144	58,03%	

Figure 4.4: Work days into the Operating System Kopernicano

This table, which is updated less than once a week, when the new CSV file is imported in the spreadsheet, shows the progress of each project with the worked hours of each member. There are some Projects that are already ended (project 3) or close to the end (project 1), and others that have only just begun. All the projects of the year, also the ended one, will be kept in the table because they enter in the year’s accountability. From the table is possible to see that the hypothesized amount of hours is often really close to the real one (the only ended project has only one day of difference). These worked hours will be paid following the daily fee agreed during the recruitment, hence For project 3, FF will receive an amount equal to his daily fee multiply by 10.

This is how the hours for billable projects are paid. Since until now, the founders of Kopenernicana have decided not to share between them some percentage of the income, all the gain coming from the billable projects is used to pay the non-billable ones. Of course from the gain has to be removed the cost of the members for the day working on the project, the cost of each project (can be some trip for the members or for the client), the cost of all the applications used in Kopernicana and other internal costs of the organization. After removing the cost, the amount is split for the total non-billable days of the organization, and then each member will be paid on his number of worked days.

On the left of figure 4.10 the name of the non-billable projects of Kopernicana; these projects don’t change over time but remain equal, of course, is possible to modify or add a new one for some new job in Kopernicana, but usually, they don’t change. For the rest, the table is really similar to figure 4.3: for every member, the



	A	B	C	D	E	F	G	H	I	J	K	L
1				FF	AG	RL	JV	ED	GL	IM		Total
2	Business Development			5,0			4,0	2,0	5,0		6,0	22,0
3	Governance e Gestione			3,0	8,0		2,0	5,0			3,0	21,0
4	Contabilità e Amministrazione			7,0								7,0
5	Articoli, Post, Comunicazione			4,0	4,0				6,0		7,0	21,0
6	Prevendita			7,0	6,0	5,0				5,0		23
7	Evoluzione Offerta			2,0	4,0	8,0	3,0				4,0	21,0
8												
9	<b>Total amount</b>		GG NON FATTURABILI	<b>28,0</b>	<b>22,0</b>	<b>19,0</b>	<b>10,0</b>	<b>11,0</b>	<b>5,0</b>	<b>20,0</b>		<b>115,0</b>
10	€ 61.180,00											
11										Total per day		€ 532,00

Figure 4.5: Non-billable projects

total of days worked for each non-billable project. Under the grey line, the results. The total amount is the amount to be split, and the total per day is the amount divided for the total non-billable days; it means that at the moment, the gain for each member for each day worked on a non-billable project is 532 €.

### 4.2.2 Problems

Let's see what the problems are, or better where is possible to do some improvements to this system. First of all, it is important to say that the system works well and the final output (all the total amount and data) is correct, anyway, some problems related to the lack of automation are present and these create a big waste of time in the organisation. As we already said in the past section, all the data are manually inserted or are imported from other application, but always without automation; for example the user in charge to update the data in the worksheet, has to export the CSV file from Clockify and then import it every time (usually once a week). There are three improvements that could be done and that could help Kopernicana to improve its accountability system and save a lot of time:

- Automatic importing of the Clockify reports into the system.
- Create a connection with Pipedrive to automate the creation of a new Project in both the OSK and in Clockify.
- Improve the dashboard of the system, including more organization data and create some graphs or tables to show them in a better way.

The first two points will be studied in a more detailed way and developed in the system, while the third one is left to a possible future work.

Actually, as explained before, the Clockify reports that are used to calculate the amount of all the hours worked for each project are imported manually into the system after exporting the CSV report file from the web application. This doesn't require a lot of time but someone has to do it at least once a week. Since the total worked hour is also used to calculate the total amount of daily salary for the non-billable project could be nice to have the update more than once a week, in order to have the correct amount every day and every time a member enter in the system to check that data. It's important to think that the Operating System Kopernicano is open to all the members so everyone can check the statistics whenever they want; for this reason, a daily update would be really interesting. In addition to the time spent to import the CSV file the user in charge of the

system every time has to check that everything works well because from Clockify it's possible to export different type of report, grouped and subgrouped by different subjects (user, projects are task some example) and if the import file has a different structure, the system doesn't work. This couldn't seem a big problem but if repeated for two or three times a week, in the end, the time spend doing this operation is a lot and the probability of making some errors grows considerably.

The second possible improvement that has been thought for the system is related to the connection with Pipedrive. Pipedrive is the first information system where all the information about deals and clients are inserted. At the moment, the connection between Pipedrive and the other two systems used for the accounting (the OSK and the time tracking software) doesn't exist at all, all the data born in Pipedrive are inserted manually in the other two systems. The data interesting for us in Pipedrive are the new projects that start, and hence marked as won in the CRM, these projects have to be added in the list of projects of Clockify (to give the possibility to the members to create working hours with that project) and also two new rows (one for the budget and one for the worked hours table) have to be created in the Operating System Kopernicano. Here, the work that has to be done by the members in charge of doing these operations is a lot and the probability of making some errors even bigger. Since the deals and projects can be inserted in the CRM by different members in different moments, it's easy to understand the difficulties of these operations. At the moment this operation could be done manually since the numbers of projects in Kopernicana are still not so many, but in the future (already during the problem of Covid-19, where the company has had a great growth due to its knowledge and preparation in the remote and smart working field) where the numbers of projects will be bigger, this implementation has to be done for sure.

To summary the situation a bit, a graph of the complete system without implementation is shown in the following figure.

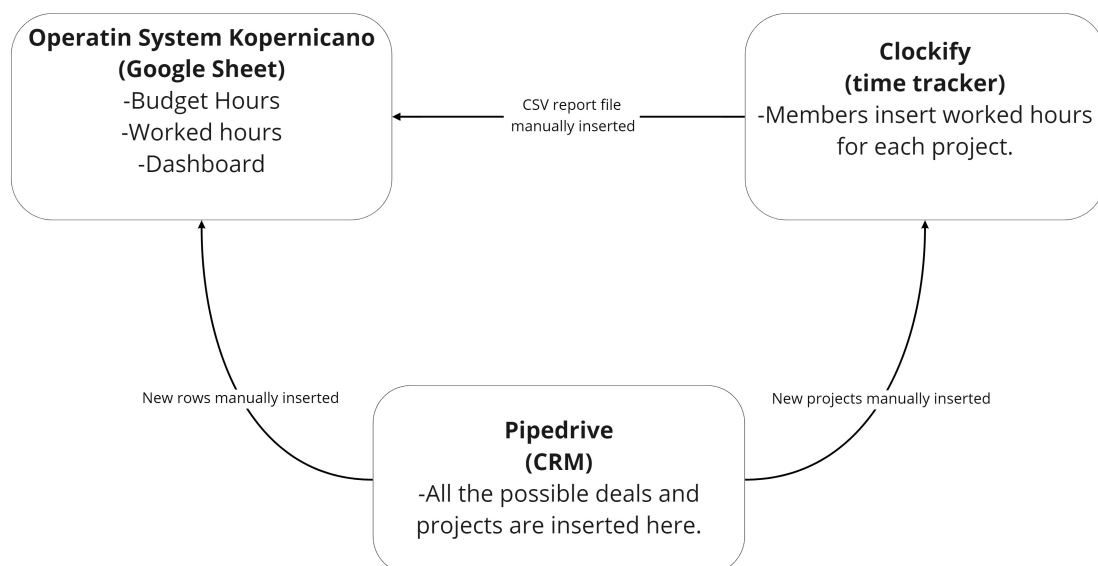


Figure 4.6: Accounting system in Kopernicana before automations

## 4.3 Tools used to improve the system

To create these connections that have been explained before, two tools will be used, Zapier <sup>1</sup> and Coupler.io <sup>2</sup>.

These tools have been chosen after long studies of all the market option for tools that automate the workflow of companies. Zapier is an online automation tool that connects apps. You can connect two or more apps to automate repetitive tasks without coding or relying on developers to build the integration. For example, maybe you get a lot of email attachments in your Gmail account and you want to save them to Dropbox. Every time you get an attachment, you could open up the email, click the attachment, and then save it to Dropbox. Or you can have Zapier automate this for you, saving you time and effort. It's easy enough that anyone can build their own app workflows with just a few clicks. This is really important because it gives the possibility to Kopernicana in the future, also without developers in the team to change or modify the settings of the workflow. Hence this solution to connect apps has been chosen instead of the creation of an application that would be more difficult for Kopernicana to modify in the future. Zapier will be used to create the connection between Pipedrive and Clockify.

The second tool used is Coupler.io; it is a Google Sheets' add on that automates integration with different data sources as Pipedrive or HubSpot, it is also possible to use to create a personal import from a CSV URL (using API) of others application. Hence, Coupler.io will be used to create the connection from Pipedrive (using the Pipedrive connection offered the tools) and Clockify (using the API) to the Operating System Kopernicano.

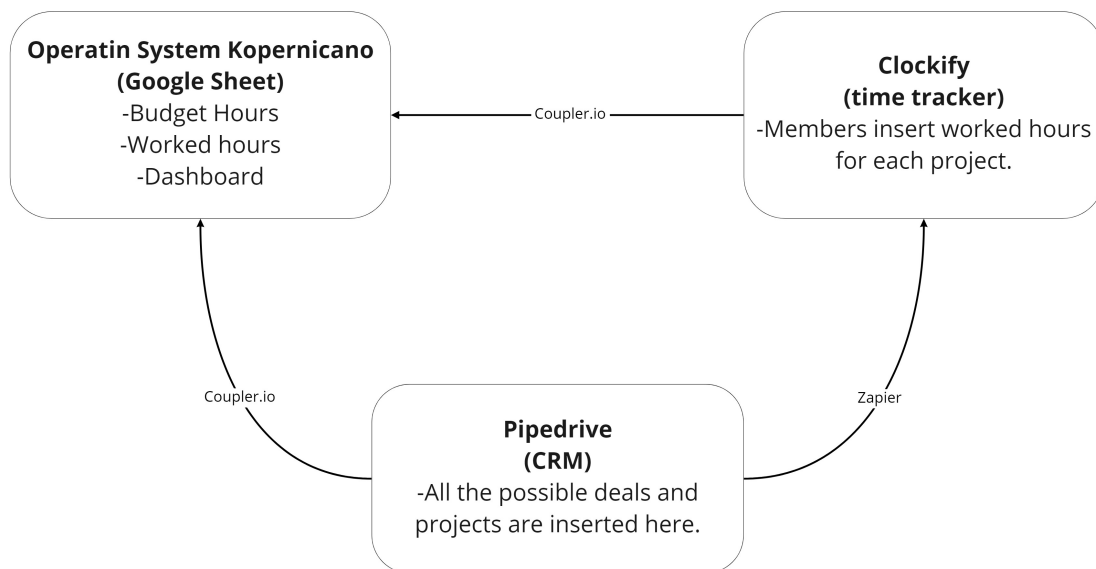


Figure 4.7: Accounting system 2.0 in Kopernicana

In figure 4.7 the schema with the tolls instead of the manually operation is shown. Now these connections will be explained in a detail way.

<sup>1</sup><https://zapier.com/how-it-works>

<sup>2</sup><https://www.coupler.io/>

## 4.4 Clockify and Operating System connection

As explained before the connection between these two systems has been implemented using Coupler.io. The integration is done with a CSV importer, exploiting the Clockify's API that provides the possibility to automatically extract the reports from the time tracker application.

```
https://global.api.clockify.me/workspaces/5e675e64418bd329e0cf4f61/reports/-
summary?export=csv
```

This is the URL of the API, in the endpoint are inserted the location of the resources to be accessed. In particular this API extract the CSV file of the Summary reports of the workspaces with that specific ID. In the request headers is specified the API key of the Clockify account that is necessary to have the permissions.

Here the request body:

```
startDate: '2020-01-01T00:00:00.000Z'
endDate: '2020-12-31T23:59:59.999Z'
isDetailed: false
billable: BOTH
description: ""
sortDetailedBy: timeAsc
firstTime: true
archived: Active
me: TEAM
includeTimeEntries: true
name: ""
groupingOn: true
groupedByDate: false
page: 0
count: 500
roundingOn: false
groupBy: PROJECT
subgroupBy: USER
weeklyGroupBy: PROJECT
weeklySubgroupBy: TIME
userGroupIds: []
userIds: []
projectIds: []
clientIds: []
taskIds: []
tagIds: []
```

Let's check the more interesting lines: From the first one, it's possible to choose the date, since the report of all the years is needed, first of January is the *startDate* and 31<sup>st</sup> of December at midnight is the *endDate*. The parameter *isDetailed* is really important. setting it to false means that the report will be a summary reports, hence not all the inserted worked hours will be shown in the report but only a sum for each user for each project (how the report is group by is decided

in the following parameters). Next, *billable* equal to "both", means that in the report will be present both the hours worked for a billable project and the ones for non-billable projects. Billable and non-billable is a parameter of a project that has to be inserted in Clockify when the project is created. Next, in the parameter of the request body, the other important lines are the *groupBy* and *subgroupBy* that are the parameters used to decide how the report is grouped. Selecting Project and User as parameters means that in the report will be shown for every project the total amount of hours of every member worked for that project.

	A	B	C	D	E	F
1	Project	Client	User	Hourly rate (EUR)	Time (h)	Time (decimal)
2	Evoluzione Offerta	Kopernicana	Roberto	0	96:26:09	96,44
3	Evoluzione Offerta	Kopernicana	Albertog	0	34:00:00	34
4	Evoluzione Offerta	Kopernicana	Joshua	0	08:10:00	8,17
5	Evoluzione Offerta	Kopernicana	Ilaria	0	21:45:00	21,75
6	Evoluzione Offerta	Kopernicana	FrancescoF	0	19:30:00	19,5
7	MMN Day - 20KC003	MMN	FrancescoF	150	08:00:00	8
8	Holacracy 2020 - 20KC001	ScuolaTao	FrancescoF	150	75:51:23	75,86
9	Governance e gestione	Kopernicana	Albertog	0	50:59:57	51
10	Governance e gestione	Kopernicana	Roberto	0	28:45:00	28,75
11	Governance e gestione	Kopernicana	Ilaria	0	41:30:00	41,5
12	Governance e gestione	Kopernicana	Joshua	0	29:05:58	29,1
13	Governance e gestione	Kopernicana	FrancescoF	0	87:00:00	87
14	Purpose & Strategy - 20KC006	Domicare	Albertog	0	26:30:00	26,5
15	Purpose & Strategy - 20KC006	Domicare	Ilaria	0	22:30:00	22,5
16	Attività per Talent Garden - 20KC012	TAG	FrancescoF	150	20:00:00	20
17	Consulenza Direzionale - 20KC005	Regola	Albertog	0	24:03:00	24,05
18	Consulenza Direzionale - 20KC005	Regola	Ilaria	0	19:00:00	19
19	Consulenza Direzionale - 20KC005	Regola	FrancescoF	0	17:30:00	17,5

Figure 4.8: Summary report of Clockify hours

The figure above presents the data coming from the API. As explained before, the first column is for the project, then the client, the user and the total time for that project. In this way, automatically, every day this spreadsheet will be updated and no more work by the user is needed. With the function SUMIFS(), comparing the Project and the user it's really easy to extract the total hours worked for each project and insert them in the main table (figure 4.4).

## 4.5 Pipedrive and Operating System connection

Always with Coupler.io, the connection between Pipedrive and the Operating System Kopernicano has been created. Create this connection is easier since Coupler.io provides a connection between Google sheet and Pipedrive. It's only necessary to give the permission of the Pipedrive account and select the details of which objects have to be imported in the google sheet. It's possible to choose between deals, organization or person, of course, what it's needed for the system are the deals. With this connection created using the add on, the page with all the deals and its data will be updated every day.

fx						
	A	AD	AE	AF	AG	AN
1	id	title	stage_id	stage.name	value	status
2	1	La locomotiva	5	Negoziazioni avviate	5000	open
3	2	Eximus performance	4	Proposta presentata	7000	won
4	3	Comune Usmate	3	Dimostrazione pianificata	1000	open
5	4	Asfalti strade	3	Dimostrazione pianificata	10000	open
6	5	Consulenza organizzativa	5	Negoziazioni avviate	9500	open
7	6	Consulenza OKR	2	Contatto stabilito	15000	open

Figure 4.9: Pipedrive in the Operating System

In the figure above, only the interesting columns for the system are present; several other columns with useless information about deals have been hidden. The details important for the system now are, of course, the name of the deal, the value, the status and also the stage where the deal is at the moment. The stage will be important for future improvement, hence it has been shown here. Since also this table is updated every day, as soon as a deal change status from open to won, it will be added in the tables with the projects (figure 4.3 and 4.4).

## 4.6 Pipedrive and Clockify connection

This connection has been implemented using the potential of Zapier. Zapier gives the possibility to automate workflows; after an event (called trigger) happens in an application "A", it's possible to decide to do an action on another application (or on the same, of course) "B".

Zapier exploits the API of each connected web application. This means that the possible triggers and actions are the ones provided by the API. For example, if the API of a specific web application provides the possibility to extract a new object inserted on it, *new object created* could be a trigger in Zapier; if using API of this application is possible to insert another object, *create object* could be an action to do on this web application.

The second step after checking that both Pipedrive and Clockify have integration with Zapier is to decide the trigger and the action of this zap (this is the name of an operation made by Zapier).

The aim of this connection is to automatically create the new project into the time tracker application in order to start to insert worked hours on it. Hence, since creating a new project is a possible action to do over Clockify the action chose is *create project* in Clockify. The choice of the trigger is a little bit more difficult. When a deal present in Pipedrive becomes a project for Kopernicana and has to be inserted in Clockify? The possible triggers offered by Zapier are an update over the stage of a deal or the change of status always of the deals, from open to won.

Since in Kopernicana all the activities done before a deal is signed, are inserted in the internal project called *prevendita*, it is necessary to have the projects in Clockify only after the signature of it, so after the deals in Pipedrive will be marked as won. Hence the trigger *deal won* has been chosen. As parameters of the zap, the name of the project in Clockify, obviously, will be the name of the deals in Pipedrive. The project will be billable as it is a project with clients and not an internal one and of course, it will also be public that means that will be possible to add on it working hours by all the members of Kopernicana.

## 4.7 The new Operating System

After these three implementations, the operating system for the accountability of Kopernicana works better and doesn't need a member that creates connections between the system manually every day.

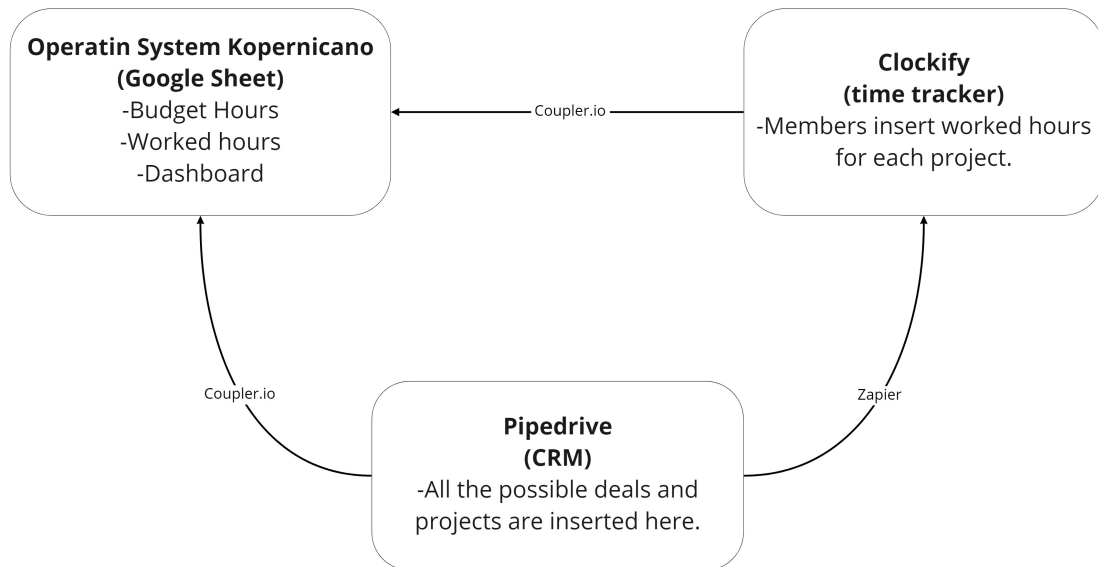


Figure 4.10: Accounting system 2.0 in Kopernicana

The figure above is the one that has already been inserted at the beginning of this chapter to show which are the tools that could be used to create these three connections, but nothing was said about the details of these connections. The three implementations have been separately explained in the sections above. Here the aim is to give an overview of the improvement and summary the work done.

- From Clockify to the Google sheet: the report is imported and the worked hours of the Operating System filled from it.
- From Pipedrive to Clockify: when a deal is marked as won in the CRM a new project is inserted as billable in the time tracker app using a zap.
- From Pipedrive to Google sheet: A page with all the details of all the deals in Pipedrive is updated once a day using Coupler.io. All the won deals are inserted in the tables *budget hours* and *worked hours*.

These may seem little improvement but summing all together and calculating the time saved to repeat all the operations that should be done more than once a week is a great job. In addition to save time, creating an automatic workflow will eliminate the human error that could be done, like a different name of projects in two platforms or a repetition of the same project with two names or the possibility to forgive to insert something and so on. When a system is automated, it is impossible to have all these errors.



### 4.7.1 What is still missing?

Two are the more important tables of the Operating System Kopernicano, the one with the budget hours and the one with the worked hours, that is split in billable and non-billable projects. The second table was already studied and is automatically filled by the System from the report imported from Clockify, while nothing was saying about the first one. To remember a bit, the budget hours are useful to do a forecast of the hours that will be spent by each member over each project. For example, when a deal is on the way to be signed by the client it is important to have a forecast of how many members will work on that project and for each one how many hours will work on it. This to have a concrete idea about the value of the project and the total time needed to complete it. Having this forecast is also important to understand if the organization has time to work on more projects or it is full and new projects can't be scheduled. At the moment this operation is done by a member of the organization, that for each project create a forecast of each task that has to be done, and then knowing the capabilities of each member, associates each task to one or more member. After this work is completed, another member (the one in charge of the Operating System Kopernicano) will fill the budget hours table. As is possible to imagine, this is a complex operation either because doing a precise forecast of the hours is really difficult either because every time the members start with a white page to be filled with tasks and hours, hence a lot of time is needed to complete the operation. In addition to these difficulties, of course, all the problems already explained, that could occur when something is done by human and has to be repeated several times are present in this operation.

*Is it possible to create some automatic workflow in order facilitate the operation?*

Obviously it's not possible to automate the creation of the document because the task and the number of hours are specific for each project, consequently is also not possible to automatically fill the tables of the budget-hours. Therefore the idea could be to create a template for the budget-hours document to give the possibility to the member in charge to forecast the hours not to start every time from a white document but from a pre-compiled one. If the template is a Google spreadsheet, is also possible to extract from it all the data necessary to fill the budget hours table in the Operating System Kopernicano.

### 4.7.2 Project details

Project details are the name of the new Google worksheet created to reach the goals that have been explained above. The workflow of this worksheet is the following.

1. create a list with only the won deals of Pipedrive
2. for each won deal create a page starting from a template
3. extract the data from the template and use them to fill the budget hours table in the Operating System Kopernicano.

First of all, as for the Operating System Kopernicano, Coupler.io has been used to import in the worksheet all the information about the deals from Pipedrive, that are inserted in the spreadsheet called *Pipedrive deals*. In order to achieve these goals, the standard functions of Google Sheet are not powerful enough, so it was necessary to create some script exploiting Google app script. Then the main sheet (Called *Home*, has been inserted to host all the won projects and the summary of the budget-hours. It has a column for the name of the projects and one column for each member that will host the budget amount of hour for each project. The column project will be filled using the following function:

```
function projects() {
  var ss = SpreadsheetApp.getActiveSpreadsheet();
  var sheet1 = ss.getSheetByName("Pipedrive deals");
  var y = 2;
  for (i = 1; i < 20; i++) {
    var project = sheet1.getRange(i, 31).getValue();
    var status = sheet1.getRange(i, 39).getValue();
    if (status == "won") {
      var sheet = ss.getSheetByName("Home");
      var cell = sheet.getRange(y, 2);
      cell.setValue(project);
      y = y + 1;
    }
  }
}
```

From Pipedrive deals, all the name of the projects (column 31 of the sheet) that have the status (column 39) equal to won will be orderly inserted in the Home page.

After this operation is completed, a page for each project present in the list has to be created, starting from a template. The template has been created and is a page in the worksheet. Using the following function all the pages are created.

```
function templatePages() {
  var ss = SpreadsheetApp.getActiveSpreadsheet();
  var sheet = ss.getSheetByName("Home");
  for (i = 20; i > 1; i--) {
    var title = sheet.getRange(i, 2).getValue();
    if (title != 0) {
      var template = ss.getSheetByName("Template");
      var itt = ss.getSheetByName(title);
      if (!itt) {
        template.copyTo(ss).setName(title);
      }
    }
  }
}
```

The function above checks all the projects in the Home page and for each one, if the page doesn't exist yet, create a new page called as the name of the project

and that is equal to the template one.

Now the last operation that has to be done is to extract the data from each page and fill the columns of the members in the home page. Unfortunately is not possible to use the standard formulas to copy the value of some cells because the name of the pages are not known, they are dynamically created with the name of the projects, hence the following function has been used to do this task.

```
function fillData () {
  var ss = SpreadsheetApp.getActiveSpreadsheet ();
  var sheet1= ss.getSheetByName("Home");
  for (i=2;i<6;i++){
    var value =sheet1.getRange(i, 2).getValue ();
    var sheet = ss.getSheetByName(value);
    var FF = sheet.getRange("B2").getValue ();
    sheet1.getRange(i,4 ).setValue(FF);
    var AG = sheet.getRange("C2").getValue ();
    sheet1.getRange(i,5).setValue(AG);
    var RL = sheet.getRange("D2").getValue ();
    sheet1.getRange(i,6).setValue(RL);
  }
}
```

For each project on the home page, the function extracts the value of the budget hours of each member (that are in knowing cells) and copy it in the exact column of the Home page. After all the functions run the Home page will be like a dashboard with all the data needed to fill the budget hours of the Operating System Kopernicano.

In the following figure, the schema of all that has been explained before.

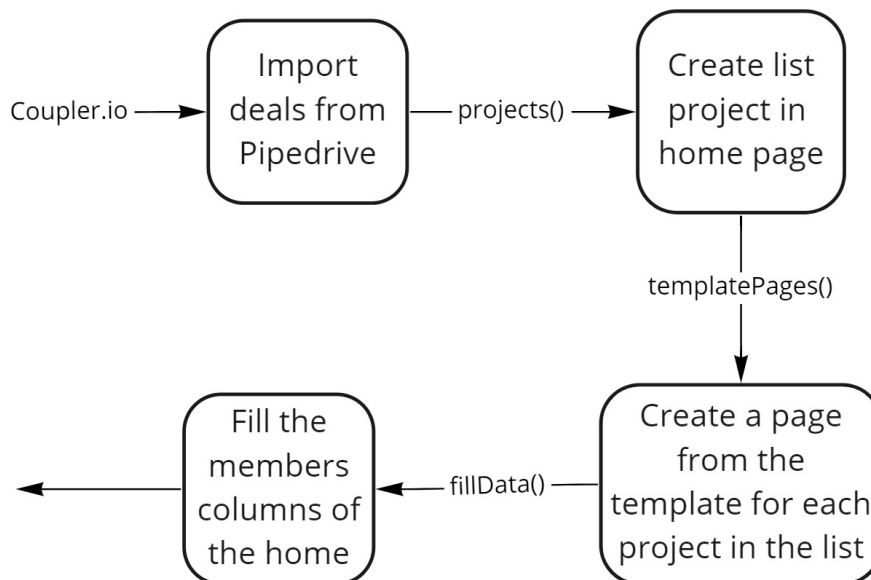


Figure 4.11: Workflow of Project details Sheet

After add this implementation in the work the initial schema is modified as follow.

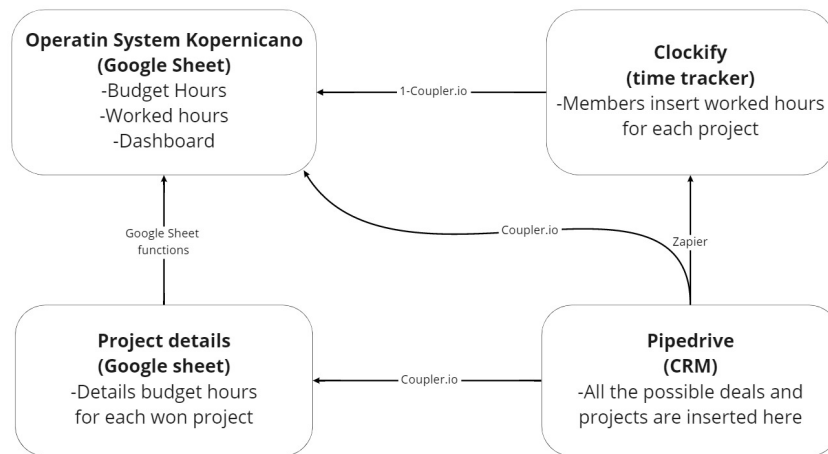


Figure 4.12: Full schema of the implemented system

The project details of course is the box for the schema 4.11 shown before.

## 4.8 The complete workflow

It seems that everything works well, and there is no need for further improvement. For the technical part, it's true, what was scheduled to be automated is automatic now, but there are some external aspects of the system that have to be analyzed. The following are the problems that have to be solved before saying that the work is completed.

- Who is in charge of compile the budget hours in the *project details* sheet?
- When this sheet should be created and filled?
- Who is in charge of control all the tools used for the system?
- At the moment doesn't exist a complete workflow that follows every stage of the deals in Kopernicana.

Let's start from the first and then we will conclude with the last problem that is a complete overview of the work.

At the moment in Kopernicana there is a member in charge of fill in the documents of the budget hours. The changes in this part of the system have also been done with him and following his advice, a final template will be done and insert in the system. Probably the same member will continue to manage this part also when the internship is finished.

The second problem is more complex, a paper from the template has to be created for each project, but it has to be already filled when the contract with the client will be signed; hence this operation has to be done before signing the contract with the client, and so before be certain to win the negotiation. This is tricky; if the new page is created too late in the negotiation, there is no time to fill it; however, if it is created too early, there is much more probability that the deals won't be signed and useless work would have been done. To solve this problem, the different possible stages of a deal in Pipedrive come to us. As already explained, Pipedrive manages the deals using stages, based on the moment of the negotiation. There are five different stages, the first one for the first contact between the company and the client, and the last one when almost all the details of the deals are accorded. Following the reasoning done before, the perfect stage to create the new page is the fourth one, that is not too far from the signed but at the same time, there is time to study the project and fill the project details page.

The third question is more an holacracy problem. As said more times every job in Kopernicana is done by a member that has a specific role, for example, there is a role called *website* manager and the member that has this role, obviously is in charge to manage the website. For the new tools used to automate the system (Coupler.io, Zapier and the new Google Worksheet) doesn't exist a role yet. So it is important that a new role will be created during a governance meeting and that a member takes control of it. The role name will be *tools manager* and the responsibilities of the role are:

-Host the Coupler.io adds on on his Google Sheet. Coupler.io is a client-side service so when a Coupler.io add on are inserted in a Google Sheet page is not visible to all the user with the permission to see the page but only to the user that has created

it.

-Host the zap to create the project in Clockify on his Zapier account.

-Control that everything works well. Some updates of the tools could be done and maybe some modifications have to be done on the settings of the tools.

For the last point (that is a complete overview of the system) a schema has been done to makes everything clear.

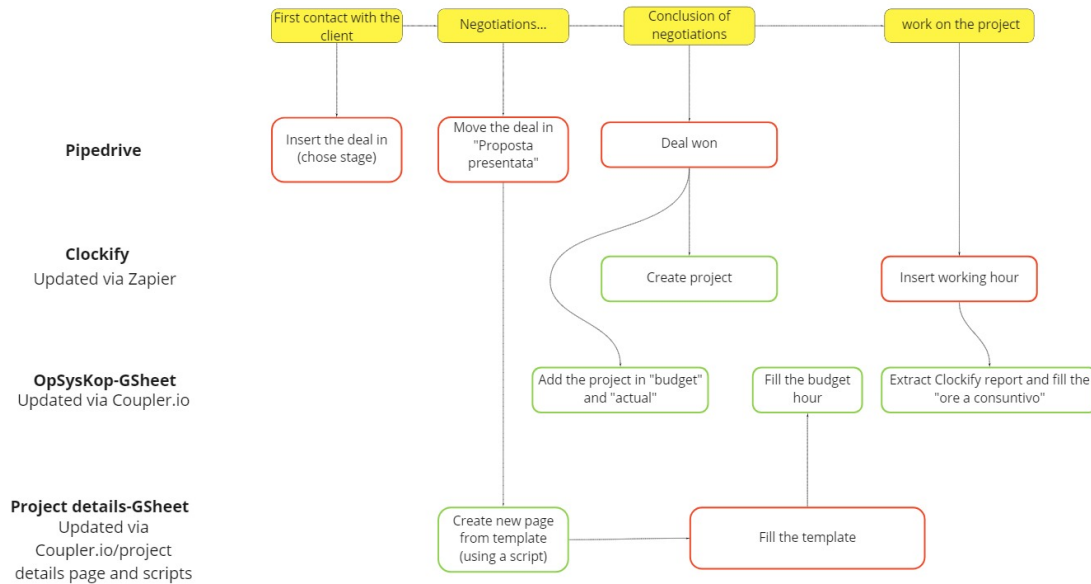


Figure 4.13: Complete workflow

On the top (in yellow), the different steps of a project in Kopernicana, from the first contact with the client to the work by the members on the project. On the left, there are four different platforms used to manage the accountability of the organization. In red, the operations that have to be done manually by the member, the ones of Pipedrive by the tradesmen of Kopernicana, the one of the project details Google Sheet by a declared member, while inserting the working hours in Clockify is done by all the members. In green, the operations that have been automated by the implemented system.

# Chapter 5

## Conclusions and Future Works

This chapter illustrates the conclusive considerations on five months internship done in Kopernicana and forward-looking perspectives of the work described so far.

### 5.1 Conclusions

The work presented in this document consists of the design of a new web application and the implementation of a semi-automated system for the project management and the accountability in Kopernicana. We enter in Kopernicana five months ago with the goals to do experience in the world of work, after the university studies and to discover the idea of holacracy. In these five months, Covid-19 create problems in all the world and also our experience has been affected. Luckily for us, this situation creates few problems in our work, since Kopernicana was already a reality where all the members work from home and from different places, of course, we didn't have any possibilities to meet the members except using some video conference platforms, hence our integration was really slow and sometimes difficult also due to the big news of Holacracy. In the moment that we are writing the conclusion of the thesis the situation of Covid-19 in Italy is much better than three or four months ago. We can say that the Kopernicana gives us the possibility to see how the world of work could be in the future, after this incredible pandemic, from a privileged point of view since the aims of Kopernicana is also helps companies to learn how to practicing remote working and they were already prepared to this kind of situations. This, in addition of being in contact with the Kopernicana team, composed by people coming from several work sectors bringing into the organisation the knowledge and experience of their previous works give us the possibility to learn a lot, on a didactic point of view with the work and studies are done but also from a personal point of view thanks to all the members. As regards the works were done during these months, the first one, that is the design of a new type of ERP is an ambitious project but we think is really innovative and we had put the bases to something that in the future could be really developed and then used not only by Kopernicana but also by other holacracy or not firms. The second work done, that is the implementation of a part of the designed application, in addition to be a valid proof of the power of the new ERP that has been designed, is an important improvement in the already existing information system of the organisation and

it will save a lot of time to the members. This system shows that is possible to work with different applications that provides different functionalities, but merge together all their data in order to analyze them and extract important information.

## 5.2 Future Works

Future developments could focus on both the works done. A considerable amount of work need to be performed to implement a complete prototype of the holocratic ERP system that we have designed. But, since the holacracy world is getting much wider, implement this new ERP not only with new trainees of Kopernicana, but also with the help of some expert developers could be an important investment for the organisation. In addition to that ERP could be interesting the implementation of a new application for holacracy compaanies, since the ones that already exist (Holaspirt and Glassfrog) aren't perfect and this is highlighted by the result of the survey. The accountability system is in a good state of automation but probably in the future, when the company will grow, more integration will be necessary to update the system, for example with the bank application to extract the invoices. If the complete ERP will be created also the accountability system will be included in it and no more updates will be necessary.



# Bibliography

- [1] Carlo Altomonte and Armando Rungi. “Business Groups as Hierarchies of Firms Determinants of Vertical Integration and Performance”. In: *European Central Bank* (June 2013).
- [2] Debora Ancona, Henrik Bresman, and Katrin Kaeufer. *The Comparative Advantage of X-Teams*. URL: [https://www.researchgate.net/publication/39322924\\_The\\_Comparative\\_Advantage\\_of\\_X-Teams](https://www.researchgate.net/publication/39322924_The_Comparative_Advantage_of_X-Teams).
- [3] *API Asana, how to integrate Asana*. URL: <https://developers.asana.com/docs/#object-hierarchy>.
- [4] *API Clockify, how to integrate Clockify*. URL: <https://clockify.me/developers-api>.
- [5] *API Drive, how to integrate Drive*. URL: <https://developers.google.com/drive/api/v3/about-sdk>.
- [6] *API dropbox, how to integrate dropbox*. URL: <https://www.dropbox.com/developers/documentation/http/documentation>.
- [7] *API Flock, how to integrate Flock*. URL: <https://docs.flock.com/display/flockos/Methods>.
- [8] *API Glasfrog, how to integrate Glasfrog*. URL: <https://documenter.getpostman.com/view/1014385/2SJViY?version=latest>.
- [9] *API holaspirit, how to integrate holaspirit*. URL: <https://app.holaspirit.com/api/doc/#patch--api-me>.
- [10] *API Hourstack, how to integrate Hourstack*. URL: <https://hourstack.com/integrations>.
- [11] *API HubSpot, how to integrate HubSpot*. URL: <https://developers.hubspot.com/docs/overview>.
- [12] *API icloud, how to integrate icloud*. URL: <https://reincubate.com/it/support/ricloud/icloud-api-v1/#overview>.
- [13] *API Miro, how to integrate Miro*. URL: <https://miro.com/api/>.
- [14] *API Monday, how to integrate Monday*. URL: <https://monday.com/developers/v2>.
- [15] *API notion, how to integrate notion*. URL: <https://www.notion.so/Does-Notion-have-an-API-I-can-use-4541b07a5caa46dba0026624646118c0>.
- [16] *API oneDrive, how to integrate oneDrive*. URL: <https://docs.microsoft.com/en-us/onedrive/developer/rest-api/?view=odsp-graph-online>.

- [17] *API PandaDoc, how to integrate PandaDoc*. URL: <https://developers.pandadoc.com/reference#about>.
- [18] *API PipeDrive, how to integrate PipeDrive*. URL: <https://developers.pipedrive.com/docs/api/v1/#/>.
- [19] *API Slack, how to integrate Slack*. URL: <https://api.slack.com/rtm>.
- [20] *API Trello, how to integrate Trello*. URL: <https://developers.trello.com/reference/>.
- [21] Richard J. Zeckhauser Apostolos Filippas John J. Horton. “Owning, Using and Renting: Some Simple Economics of the “Sharing Economy””. In: (May 2019).
- [22] *Author’s personal copy Leadership is like fine wine*. URL: <http://www.sciencedirect.com>.
- [23] Gisela Bäcklander. “Doing complexity leadership theory: How agile coaches at Spotify practise enabling leadership”. In: *DOI* (2018).
- [24] Liliana David Bernhard Bockelbrink James Priest. “Sociocracy 3.0 - A Practical Guide”. English. In: (June 2019).
- [25] NEIL HUNT CARLOS A. GOMEZ-URIBE. “The Netflix Recommender System: Algorithms, Business Value, and Innovation”. In: *ACM Transactions on Management Information Systems*, 6 (Dec. 2015), p. 19.
- [26] Frank Shipper Greg L. Stewart Charles C. Manz. “Every one a TeamLeader Shared Influence at WLGore associated”. In: *Science Direct* (2018).
- [27] George Serafeim Claudine Gartenberg Andrea Prat. “Corporate Purpose and Financial Performance”. In: (June 2016).
- [28] Valve Corporation. “Valve Corporation”. In: (Mar. 2012).
- [29] *Data to Knowledge to Results- Building an Analytic Cpability*. URL: [www.accenture.com](http://www.accenture.com).
- [30] Igor Denisov Emil Velinov Vasko Vassilev. “Twitter earthquake detection: Earthquake monitoring in a social world”. In: *Problems and Perspectives in Management* (Mar. 2018). ISSN: 1727-7051.
- [31] Teppo Felin and Thomas C. Powell. *Designing Organizations for Dynamic Capabilities*. URL: [https://www.researchgate.net/publication/282704874\\_Designing\\_Organizations\\_for\\_Dynamic\\_Capabilities](https://www.researchgate.net/publication/282704874_Designing_Organizations_for_Dynamic_Capabilities).
- [32] Bradley R. Staats Francesca Gino. “The Morning Star Company: Self-Management at Work”. In: (Jan. 2014).
- [33] Dianne J Hall and Jiahe Song. *The Conceptualization of Data-driven Decision Making Capability*. URL: [https://www.researchgate.net/publication/280051851\\_The\\_Conceptualization\\_of\\_Data-driven\\_Decision\\_Making\\_Capability\\_The\\_Conceptualization\\_of\\_Data-driven\\_Decision\\_Making\\_Capability](https://www.researchgate.net/publication/280051851_The_Conceptualization_of_Data-driven_Decision_Making_Capability_The_Conceptualization_of_Data-driven_Decision_Making_Capability).
- [34] Anders Ivarsson Henrik Kniberg. “Scaling Agile at Spotify”. In: *Annals of geophysics = Annali di geofisica* (Oct. 2012).

- [35] Francis Heylighen and Clement Vidal. *Getting thing done behind the science*. URL: [https://www.researchgate.net/publication/222552899\\_Getting\\_Things\\_Done\\_The\\_Science\\_Behind\\_Stress-Free\\_Productivity](https://www.researchgate.net/publication/222552899_Getting_Things_Done_The_Science_Behind_Stress-Free_Productivity).
- [36] Nancy M.P. Bocken Ilka Weissbrod. “Developing sustainable business experimentation capability”. In: *Cleaner Production* (2017).
- [37] Kamel Mellahi Jędrzej George Frynas Michael J. Mol. “Management Innovation Made in China : Haier’s Rendanheyi”. In: *California Management Review* 61.1 (Sept. 2014), pp. 71–93.
- [38] Thomas Diefenbach John Sillince. *Formal and Informal Hierarchy in Different Types of Organization*. URL: <https://journals.sagepub.com/doi/full/10.1177/0170840611421254>.
- [39] Brendan Julian, James Noble, and Craig Anslow. *Agile Practices in Practice: Towards a Theory of Agile Adoption and Process Evolution*. URL: [https://link.springer.com/chapter/10.1007%2F978-3-030-19034-7\\_1](https://link.springer.com/chapter/10.1007%2F978-3-030-19034-7_1).
- [40] Pepijn van de Kamp. “Holacracy – A Radical Approach to Organizational Design”. In: *University of Amsterdam* (Apr. 2014).
- [41] Christian Hitz Karlheinz Schwer. “DESIGNING ORGANIZATIONAL STRUCTURE IN THE AGE OF DIGITIZATION”. In: *EASTERN EUROPEAN AND CENTRAL ASIAN RESEARCH* 5.1 (2018).
- [42] Jhon Kay. “Obliquity”. In: *Financial time* (Jan. 2014).
- [43] Peter E Rossi Emily Oehlsen M Keith Chen Judith A Chevalier. “NBER WORKING PAPER SERIES THE VALUE OF FLEXIBLE WORK: EVIDENCE FROM UBER DRIVERS”. In: *national bureau of economics research* (June 2017).
- [44] Juan Nicolas Montoya Monsalve Mercedes Rubio Andres Santiago Gutierrez Broncano. “Could innovative teams provide the necessary flexibility to compete in the current context?” In: 15.1 (2015), pp. 145–164. ISSN: 1131-6837.
- [45] Amy C. Edmondson Michael Y. Lee. “Self-managing organizations: Exploring the limits of less-hierarchical organizing”. In: *Frontiers in Earth Science* (Jan. 2017).
- [46] Christophe Midler. *Projection of the Firm : The Renault Case*. URL: <https://hal.archives-ouvertes.fr/hal-00262524>.
- [47] Eric Too Patrick Weaver. “The management of project management: A conceptual framework for project governance”. In: *International Journal of Project Management* 32.8 (Nov. 2014), pp. 1382–1394.
- [48] Marielba Zacarias Paula Ventura Martins. “An Agile Business Process Improvement Methodology”. In: *Science Direct* (2017), pp. 129–136.
- [49] Dr. Rishipa. “Analytical Comparison of Flat and Vertical Organizational Structures”. In: *European Journal of Business and Management* 6.36 (2014). ISSN: 2222-1905.
- [50] Brian J. Robertson. In: *Holacracy, Come superare la gerarchia*. 2015.

- [51] Bryan J. Robertson. “Organization at the Leading Edge: Introducing Holacracy”. In: (2007), p. 33.
- [52] Georges Romme. “Management as a science-based profession: a grand societal challenge”. In: 40 (Oct. 2016), pp. 5–9.
- [53] Philip Russom. *Big Data Analytics*. URL: <https://tdwi.org/research/2011/09/best-practices-report-q4-big-data-analytics.aspx>.
- [54] Sukhbir Sandhu Samudi Perera Claudine Soosay. “Investigating the Strategies for Supply Chain Agility and Competitiveness”. In: *Asian Journal of Business and Accounting* 12 12 (2019).
- [55] Ziyuan Wang Shimei Jiang Yimei Hu. “Core Firm Based View on the Mechanism of Constructing an Enterprise Innovation Ecosystem: A Case Study of Haier Group”. In: (June 2019).
- [56] Kate Starbird and Jeannie Stamberger. “Tweak the Tweet: Leveraging Microblogging Proliferation with a Prescriptive Syntax to Support Citizen Reporting”. In: (Jan. 2010).
- [57] Subhasree Mukherjee Vijay Kumar S. “Holacracy—the future of organizing? The case of Zappos”. In: 26.7 (2018), pp. 12–15.
- [58] Wikipedia. *Olocrazia*. URL: <https://it.wikipedia.org/wiki/Olocrazia>.