Master of Science HES-SO in Engineering
Av. de Provence 6
CH-1007 Lausanne

Master of Science HES-SO in Engineering

Information and Communication Technologies (ICT)

GASP: Guide to Assess Security and Privacy levels on web services

Supported by the Department of Digital Technologies
at the University of Winchester, UK

Author
Loïc GUIBERT

Supervisors
Dr. Pascal BRUEGGER
HES-SO//Master,
HEIA-FR · iCoSys
and
Dr. Adriana WILDE
University of Winchester · WINTS,
University of Southampton · CHT

External expert
Dr. Robert VAN KOMMER
EPFL Innovation Park · Alliance

Winchester (UK), HES-SO//Master, 10th February 2023

Information about this Report

Contact information

Author:	Loïc Guibert
	MSE Student
	HES-SO//Master
	Switzerland
Emails:	*loic.guibert@master.hes-so.ch*
	loicguib@ik.me

Declaration of honour

I, undersigned Loïc Guibert, hereby declare that the work submitted is the result of a personal work. I certify that I have not resorted to plagiarism or other forms of fraud. All sources of information used, and the author quotes were clearly mentioned.

Place, date: Winchester (UK), 10th February 2023

Signature: 

Validation

Accepted by the HES-SO//Master (Switzerland, Lausanne) on a proposal from Loïc Guibert, author, and Dr. Pascal Bruegger, supervisor.

Place, date: Fribourg (CH), 3rd April 2022

Key people

Advisor:	Philippe Joye
Supervisors:	Dr. Pascal Bruegger
	Dr. Adriana Wilde
External Expert:	Dr. Robert Van Kommer
Internal Expert:	Dr. Jean Hennebert
Dean:	Dr. Bernard Masserey

Acknowledgments

This thesis would not have been possible without the support of many people. First, I am extremely grateful to my supervisor Dr. Pascal Bruegger, who always supported me both in my whole education curriculum and through this thesis.

I also want to thank my other supervisor Dr. Adriana Wilde, who accepted me as a visiting researcher and welcomed me at the University of Winchester. I am also very grateful for her supervision throughout my whole thesis despite her many obligations.

Thanks to the HES-SO//Master for allowing students to design and propose their own thesis subjects, and to ensure that the appropriate resources are provided.

And finally, I want to thank my parents, my sister, and my numerous friends who always supported me, not only during this thesis but also during my whole life.

Abstract

The vast majority of Internet users around the world accesses web services, for both personal and professional goals, often involving sensitive information. Therefore, web services must be secure to prevent attacks resulting from vulnerabilities being exploited. Against this background, recent advances in data science and Artificial Intelligence (AI) have led to an increased adoption of personalized, adaptive and customizable models in many areas, including in web services. However, these models need vast amounts of user data to achieve a satisfactory performance, raising important privacy concerns. Therefore, it is critical for organizations to ensure that the web services they provide are both developed securely and designed respectfully of the privacy of their end users.

This thesis addresses such need by providing an accessible, simple, and complete guide to evaluate web services for organizations. To meet this goal, we conducted a state-of-the-art review on multiple Information and Communications Technologies fields related to both the web environment and big data, focusing on security and privacy concerns. In addition, an evaluation of reputable existing guides (by NIST, NCSC, ENISA and CISA) was conducted to identify the gaps regarding web services, which informed the scope for the development of this new guide. Several guidelines were then developed to allow organizations to assess whether their web services are compliant with recommendations, mitigations or other useful information, incorporating the knowledge collected. Finally, an online web application has been developed specifically for a dynamic assessment of web services by organizations, against the entire knowledge collection we made in an easy, complete, efficient and accessible manner. The application, hosted at ohmygasp.com, allows organizations to receive heuristic scores on the security and privacy levels of their assessed web services, on multiple categories. Those scores and the requirement levels specified for the knowledge collection are provided in order to give the organization a prioritized list on the most sensitive concerns, in order to quickly activate levers to strengthen their web services. The accuracy of our proposal and guide has been reviewed by assessing an existing web service, which suggests that our approach is not only valid but also useful.

In its present form, the use of this guide already can benefit any organization developing web services. However, in addition, its generic design allows it to be extended for evaluations of systems from other contexts using customized guide contents, and therefore having the potential to become a centralized or decentralized hub that could store, compare and help design guide contents for various subjects, scopes or technologies. This growth would allow us to improve the security and privacy levels of many more systems.

Keywords: IT Security, User Privacy, Confidentiality, Web Services, Data Science

Contents

List of Figures

List of Tables

List of Listings

1 Introduction

This document is the report written during the master thesis that takes place at the end of the HES-SO//Master curriculum. This document starts with an introduction Chapter that will explain the context of this thesis.

First, some useful terms that will be used throughout this report will be defined. Then, the context in which our thesis lays will be exposed, followed by an explanation of the contribution that our thesis brings to the academical and industrial world. The objectives of the thesis, its research question its outcomes will also be exposed. Then, some similar works will be presented, followed by an explanation of the report structure. Finally, small words about the project management will conclude this Chapter.

Contents

1.1 Definitions

First and foremost, some terms used in this report must be defined so that they are understood by all readers. Indeed, these terms can be confusing.

- **Artificial Intelligence (AI) process**: any process that uses user data to train predictive or generative models, and/or to infer results using those models. Examples: big data, Machine Learning (ML), Deep Learning (DL) models.
- **Privacy**: a catch-all terms referring to user privacy and their data privacy at the same time. Used to express the right to respect these two concerns.
- **Topic**: a domain, concept, or technique coming from the Information and Communications Technologies (ICT) field. This is a generic term to group them altogether without differentiation. Examples: cloud computing, access control.
- **Web service**: a service exposed by a device which gives an answer depending on a challenge sent by another device. Both devices communicate using a connexion link through the Internet. No distinction is made on the used technologies, regardless of the misnomer word web. Can be shortened by service. Examples: *Google Search*, *GitLab*.

1.2 Context

Nowadays, web services are among the most used Internet resources around the world. Lots of applications communicate with their infrastructure to provide multiple features to their end users. Everyone uses a large variety of them throughout their daily life, both for personal than professional needs.

The latest impressive progress made in the AI field leaded a massive adoption of personalized, adaptive and customizable processes inside web services. Among that, the increasing capabilities of Neural Network (NN) models are among the strongest ones.

However, those models need tremendous amounts of user data in order to achieve satisfactory performances. This constraint raises crucial user privacy issues towards numerous service providers collecting as much data as they can with the goal of either improving their own services, or to resell them to other organizations. This approach is known as big data collection.

Globally speaking, every software project should be analysed, designed, implemented and tested appropriately to reach great security and privacy levels. But even with the strongest will to provide a robust and secure piece of software, this objective is hard to achieve. Indeed, lots of threats, issues, vulnerabilities and errors must be taken into account and mitigated, which make it difficult for developers to ensure a complete coverage of these concerns.

Developers also need to ensure that the software they develop respect the privacy of their end users. As for security concerns, the considerations to be taken into account are numerous and difficult to assess.

There is currently a lack of methods that allow developers to build software that is as secure and private as possible in a simple, complete, and accessible manner. This fact is true for both web services and the AI field.

1.3 Contribution

This thesis aims to fill the gap we found and explained in Section 1.2 (Context).

We wish to provide our contribution to the organizations and developers that develop web services which also include AI processes. Our contribution aims to offer them an accessible, simple, and complete guide to evaluate their services based on a knowledge collection made of various topics regarding known security and privacy issues. Those evaluations will allow developers to identify the weakest parts of their services: by doing so, some levers can be pointed out to improve the overall security and privacy levels.

1.3.1 Thesis Objectives

A specifications document has been made to describe the scope of this thesis: the objectives it defines are summarized below. For further details, the document is available at Appendix A.

In order to specify relevant objectives, three global questions have been asked based the scope of this thesis:

1. Which rules, best practices, technologies and aspects should be used in order to improve the security and privacy levels of web services?

2. Which rules, best practices, technologies and aspects should be used in order to improve the security and privacy levels of the AI field, particularly for ML and DL models?

3. How can we provide an understandable and complete method to present our findings?

Primary Objectives

To answer the above questions, three objectives have been defined. They must all be completed at the end of the timeframe for the thesis to be considered as completed.

1. Establish an up-to-date knowledge collection

2. Provide an understandable guide

3. Apply and test the guide on a web service

The specifications document originally defined one of the output as *Guide* or Framework. We simplified this statement by removing the framework part.

Secondary Objective

A secondary objective has been defined: *Publish the guide as an online resource*. It will only be completed if all primary objectives have been fulfilled and if there is still time left before the end of the thesis timeframe.

Constraint

If any data collected under real-life conditions is used, analysed or processed during this thesis, ethical considerations and obligations must be applied.

1.3.2 Research Question

A proper and answerable research question has been defined in order to efficiently lead the conduct of the thesis. To this end, we have chosen the *Patient-Intervention-Comparison-Outcome(s) (PICO) method*[1], which is mainly used in the medical research field but also adapted to broader scientific fields. This is the method the most adapted to our approach we found.

We have used the PICO method using the questions asked in Subsection 1.3.1 (Thesis Objectives). The two first questions are focused on the same goal, but on two different subjects: we can therefore combine them in only one research question, and integrate the third question as well into the method.

Each letter of the PICO method, which we will refer to as *item*, can have different meanings. As an example, the item *P* can either identify a population, a patients group or the problem itself. We have chosen the most appropriate meaning for each item based on our objectives and context, while also considering the specificity of the ICT field.

- **Population**: the aimed population is the organizations and developers that develop web services that include AI processes. Their end users are not concerned.
- **Intervention**: an accessible, simple and complete guide made of various topics used to evaluate the said services.
- **Control**: the guide is used to evaluate the security and privacy levels, which point out levers that can improve the said services and overcome their weaknesses.
- **Outcomes**: an improved awareness about both the security risks of the systems, and about end users' privacy protections.

By putting all the items together, we can define our final research question which is the following:

> How to help organizations to evaluate their web services that also use AI processes, by measuring the risk levels of both ICT security and user privacy concerns?

1.3.3 Outcomes

Based on the research question, this thesis will provide three outcomes.

The first outcome will be a knowledge collection built from all topics concerned by our scope: the security and privacy concerns of web services, including the topic of AI. A state-of-the-art review will be conducted to search all relevant data needed to explain the biggest risks related to each topic using adequate sources.

[1] *PICO method* source: `https://bit.ly/3y4N1E0` (accessed 30th September 2022)

The second outcome will be a new proposal that states guidelines to build a guide which allows evaluating web services based on specific knowledge. Once this proposal defined, it will be applied to the knowledge collection previously built to create our own guide content.

The third outcome will be an application that uses the guide content previously created to allow developers and decision makers to evaluate whether their web services are compliant with the various security and privacy issues.

1.4 Similar Projects

During our preliminary researches, we found a great amount of projects that allow to assess the security and privacy levels of ICT systems: yet, none of them have a complete approach. Indeed, those projects are specialized on particular topics, such as cloud computing security, password strength or vulnerabilities evaluation. Furthermore, none of those projects address the specific scope of web services.

Nevertheless, we found some projects that are close to our scope. These projects and their references could be useful when defining our own proposal. Here is a non-exhaustive list or the major projects we found:

- ***Open Web Application Security Project (OWASP) Projects***[2]: non-profit that publishes several projects to help developers to secure their software. Their projects are focused on vulnerabilities and risks.
- ***Privacy Guide***[3]: community-driven website, lists ethical and privacy-friendly applications. Limited to recommendations and best-practices for end users.
- ***Security-List***[4]: community-driven best practices regarding various security and privacy aspects of an information system. Oriented towards a usage of web services by end users.
- **International Organization for Standardization (ISO) 27000 Series**: multiple standards focusing on various fields, such as storage security or ICT disaster recovery programs. Yet, such standards are not accessible and can be difficult to assess.
- **General Data Protection Regulation (GDPR)**: regulation on data protection and privacy applied in the European Union. This set of laws does not offer any concrete applications or mitigations.
- ***National Institute of Standards and Technology (NIST) Cybersecurity framework***[5]: allows managing cybersecurity risks using standards, guidelines and best practices. Only focused on risk management in order to improve security and resilience levels.

[2] *OWASP Projects* source: `https://owasp.org` (accessed 4th October 2022)
[3] *Privacy Guide* source: `https://www.privacyguides.org` (accessed 4th October 2022)
[4] *Security-List* source: `https://security-list.js.org` (accessed 4th October 2022)
[5] *NIST Cybersecurity framework* source: `https://bit.ly/3Szsm3j` (accessed 4th October 2022)

- ***NIST Special Publication 1800***[6]: set of various specialized guides applicable to cybersecurity, using standards-based approaches and best practices. Many of these guides are not focused on web services.
- ***Cybersecurity & Infrastructure Security Agency (CISA) Cyber Essentials Toolkits***[7]: set of Portable Document Format (PDF) pages that lists some actions defined to help organizations to integrate cybersecurity processes. Does not integrate concerns on web services nor AI.

1.5 Report Structure

Our report is divided into several Chapters, each addressing specific steps of our thesis.

A first Chapter will cover our approach used to build our knowledge collection. A state-of-the-art review will be defined, and all the knowledge needed to answer our research question will be collected.

Then, a Chapter will analyse and compare several other guides that also aim to assess and evaluate software. Their characteristics will also be listed.

Afterwards, our proposal will be defined in details. This Chapter will explain the considerations toward this challenge and our vision to create an effective and complete guide that suits our scope. The process of creating the guide content by applying our proposal will also be explained.

Based on our proposal and on the guide content, an application will be analysed, designed, implemented and tested in order to simplify the usage of our proposal.

Once all the previous steps finished, the application will be tested on a web service in order to validate it on multiple points. This approach will also concern our proposal and our guide content as they will be integrated in the application.

Finally, a conclusion will be done on the entire thesis.

1.6 Project Management

The thesis will be managed using the Scrum method, with weekly meetings between the student and the supervisors to discuss and define the backlogs of tasks to be completed. Those backlogs will be executed each week.

All the resources used and produced within the scope of this thesis can be consulted on the School of Engineering and Architecture of Fribourg (HEIA-fr) software forge [1].

[6] *NIST Special Publication 1800* source: `https://bit.ly/3M6V35k` (accessed 4th October 2022)
[7] *CISA Cyber Essentials Toolkits* source: `http://bit.ly/3GOf3bY` (accessed 23rd November 2022)

2 Knowledge Collection

First and foremost, a complete knowledge collection must be built in order to obtain a complete and comprehensive state-of-the-art review on the topics that concern our thesis scope. This Chapter will respond to this need by exposing and explaining our approach in details. Our goal is to provide a complete and unbiased review of the rules, best practices, technologies and any aspects that can improve the security and privacy levels of web services.

We will start with an explanation of the methodology that will be used for the state-of-the-art review. This step includes the topic definition, the scope, the topics evaluations, and more. Then, various topics will be assessed using current, established and reviewed resources in order to define how they can contribute to the thesis goal. Finally, a summary will be made on our findings.

Contents

2.1 Methodology

A state-of-the-art review aims to establish the condition of a field in a given timeframe. To this end, all significant information related to the said field must be collected, a set of them must then be selected following defined rules, and those selected must finally be summarized.

2.1.1 Selection of the Approach

Such process can be conducted in several ways. To decide which one is the most adapted to our needs, the two major approaches used by the academic field will be explained, and a decision will be made based on their characteristics.

Systematic Review

A systematic review mainly consists of responding to a research question while minimizing the biases that can be found during the review, both in its conducting that in the used sources. Indeed, one of its objective is to reach the largest objectivity possible. The bias management consists of taking the said biases into account during the whole process and state them clearly in the results, for example by exposing the level of confidence of the findings. Those findings are finally exposed in a refined conclusion.

Those reviews can be either quantitative or qualitative. They can be used to produce various outputs, with the two major ones being meta-analyses and systemic-cartographies. Meta-analyses are the results of independent studies on a given problem within a timeframe that have been combined using a reproducible protocol and used as a statistical synthesis of the studies. Systemic-cartographies consist of explaining methods that highlight the distribution of knowledge according to explicit criteria.

Systematic reviews must include a research protocol composed by multiple steps: identify and define the research question, establish the topic inclusion and exclusion criteria, search for data sources, extract relevant data, assess data eligibility, analyse and combine the data, and then communicate the findings. Those steps are generic and can vary depending on the chosen approach.

Literature Review

A literature review aims to resume the current knowledge for a given field during a given situation, which can be a timeframe, a geographical zone, or both. Its goal is to answer to a research question or topic using an adapted methodology. By doing so and by collecting relevant resources, a proper context can be explained to the audience.

Such reviews are focused on a unique subject and are a form of meta-analyses that combine several primary sources, from which they identify similarities and differences and possibly analyse them to define the knowledge at a given time only. An assumption is made for the chosen sources: they are considered as unbiased because they have been validated by peers.

No official or widely-adopted methodology is defined for this kind of reviews.

Decision

Our research question and our project scope include topics from various backgrounds. We will mainly review academic papers which can include quantitative and qualitative data. Those papers can also be theoretical or technical. Furthermore, best practices will also be a part of our review. Our data sources are therefore quite diverse, which means that we will not always be able to compare them.

In addition, the time and resources at our disposal do not allow us to conduct a proper and complete systemic review: this is the reason why a literature review will be made. However, we will bring some aspects and processes from the systematic review to our methodology in order to bring more rigour, and to mitigate biases as much as possible.

2.1.2 Review Protocol

Our review will be conducted by applying a protocol that defines several steps completed sequentially. As explained in Subsubsection 2.1.1 (Decision), this protocol has been built using inspiration from both review approaches by selecting the steps that are useful in our research whilst being feasible in our timeframe.

1. **Define the research question**: how to answer and cover the thesis scope?
2. **Define the scope**: which timeframe will be covered? And in which location?
3. **Search for existing work**: have any similar projects already been published?
4. **Inclusion and exclusion criteria**: which terms have to be reviewed, and which ones are not included?
5. **List data sources**: how and where to get data?
6. **Extract relevant data**: how to identify data that can help to answer to the research question?
7. **Evaluate data quality and bias**: how to evaluate whether the data is objective and relevant for our knowledge collection?
8. **Assess Data**: how to select and/or combine the most adequate findings for each topic?
9. **Present and explain the findings**: how to explain the whole findings and how to combine them to create knowledge?
10. **Review update**: will the review be updated in the future?

Apart from the thesis supervisors, we do not need to identify any other actors: indeed, this thesis is independent of any funds or companies. The said supervisors are also completely independent.

Research Question

The research question has already been defined in Section 1.3 (Contribution). It guides the whole thesis scope and direction, including the review.

Research Scope

Regarding the security field, its selected timeframe will limit our review from what we consider a modern ICT context until now. We define modern ICT as the area of web services such as we know them: we fixed its arbitrary beginning around the 2000s. We will prioritize more recent data over its older equivalent. Please note that ground knowledges defined before this period but still relevant nowadays can be used.

The AI research field has been present for several decades, but its broad adoption by the industry and fully usable ML and DL models are more recent. The ICT privacy field is also quite young, because related problems have been especially raised lately.

There will be no coverage for immature technologies or topics such as quantum ICT. This kind of topics is not fully developed which means that it can not be correctly assessed and evaluated. However, advices and warnings can be made based on current knowledge. Please note that the ICT field is not officially considered as fully mature, but is considered sufficiently stable enough by its community.

No specific limitation on locations and languages will be applied, as no data will be collected based on populations and no cultural and/or ethnic aspects come into account.

Existing Work

In order to find existing work similar to our review, we searched for academical papers and articles using specialized search engines. To do so, we entered a mix of keywords related to this thesis such as ICT security, user privacy, confidentiality, web services, AI, ML and DL models, et cetera. While trying various combinations, we played with the search engine parameters by specifying mandatory terms, dates of publication, languages, and other filters.

Some academic papers already assessed our two generic subjects, being the ICT security and user confidentiality or privacy. However, none of them are oriented towards web services. In contrast, we found resources that directly concern web services, although specialized in a particular topic such as servers architecture. Some resources we found could however be useful for our review, our approach being to gather several topics altogether, not to establish a deep, specialized and exhaustive study of only one topic.

Some papers are similar to what we want to achieve with our review and represent similar approaches:

- **A framework for *Android* applications**: how to enhance user control on privacy when accessing to user resources by Ricardo Naisse et al. [2]. Paper scope limited to mobile platforms.
- **Preserving user privacy**: definition of a framework that abstracts the privacy and the utility requirements of smart meter data by Rajagopalan et al. [3], and an equivalent in the mobile healthcare and home-care systems fields by Kotz et al. [4].

Inclusion and Exclusion Criteria

To respond to the research question, criteria to classify topics must be defined before the review. Indeed, some topics must be included in the review, others must not. The two lists must be build by browsing all possible topics that compose the generic ICT field.

In order to be included in the review, each topic must concern the research question and be within the thesis scope. Included topic will be formally defined during the review by briefly describing them.

Topics that are not compliant with those two conditions will be placed in the exclusion list, being irrelevant to our thesis scope. The list is not exhaustive and could hold other topics not concerned by our scope according to the specificity of the topics research.

Data Sources

Data sources must meet four quality criteria to allow them to be used for our knowledge collection:

- **Official source**: must not relay information from an official source, except if new relevant material is added to the original data.
- **Trusted source**: must be reviewed by legitimate pairs.
- **Best practices**: must be both demonstrated and accepted by the community, with evidences.
- **Standardized source**: standards, regulations and laws must be defined by legitimate organizations and be adapted to the context.

There is no restriction on the format of the sources: sources can come from academic research through papers and articles, grey literature, websites, companies networks, et cetera.

Relevant Data

To be included to the review, the assessed data must respond to the question and included in the scope. If not, it is not relevant and must be discarded from the review.

Too specific or specialized data will not be considered. Indeed, our approach is to bring together the biggest risks regarding security and end users privacy. Therefore, we can not deeply explore each topic, but we can provide resources for developers and decision makers to go deeper.

Evaluate Data

Data quality must be validated by various evaluations:

- **Arbitrary decisions**: only include data because of its ability to answer the research question, not because it supports our opinion. The same goes for its ouster because it contrasts with our opinion.
- **Points of view**: if a topic has multiple perspectives, all of them must be considered.
- **References**: consider the qualification, amount, and relevance of the data references to avoid unsubstantiated selection.

If the data passes the three points of this evaluation, it can be included in the review.

Data Assessment

When multiple data have been found for the same topic, several approaches can be applied to determine the knowledge to be collected. Each unique situation will be assessed by the most adapted approach, which will be one of the following:

1. **Keep the most established finding**: if the data from different sources is evaluated as different, the most established source will be kept. This choice will be done following the evaluation process defined in Subsubsection 2.1.2 (Evaluate Data).

2. **Mix of findings**: if the data of different sources is evaluated as equivalent, both sources must be combined in a manner that preserve their different perspectives.

3. **One finding**: if only one source is found for a topic, this source will be used as it.

Present and Explain Findings

The final step of the review will consist of exposing the findings in an understandable manner to create the knowledge collection. To this end, each topic to be defined must have a dedicated space which explains the current state of the art, using findings previously found.

Additional sublevels of details and categories can be defined if they facilitate the interpretation of the findings.

Knowledge will be presented using plain text. It can be illustrated if needed. Collections of items that can be compared between each other will be formatted into tables which will expose their differences.

The result of this step will be the content of Subsection 2.2.2 (Review Results).

Updates

There is no plan to extend the state-of-the-art review in the immediate future, at least not during the thesis timeframe. However, the author might proceed to one or multiple updates in his personal time afterwards. No guarantee is given and updates can be realized in an informal manner.

2.2 Review

The following Subsections will expose the results of our review. To do so, we applied the protocol as explained in Subsection 2.1.2 (Review Protocol). We started with the topics inclusion and exclusion lists and continued with the knowledge collection on the included ones.

2.2.1 Topics Inclusion and Exclusion

The topics inclusion and exclusion lists have been defined based on the various criteria stated in the review protocol. We browsed the Internet to establish the most complete lists as possible.

Included Topics

The following list shows all the topics we have to assess.

1. **Access Control**: how to provide an adapted and enforced data and system access control?
2. **Anonymization**: when anonymization must be applied? How to ensure a proper anonymization process?
3. **Application Programming Interface**: how to ensure that the endpoints are secure?
4. **Artificial Intelligence**: how to preserve user data privacy while using ML and DL models? How to avoid attacks on those models?
5. **Authentication**: how to design a secure and complete authentication process?
6. **Authorization**: how to design a secure and complete authorization process?
7. **Best Practices**: are there generic advices to enforce security and privacy levels?
8. **Big Data Privacy**: how to handle large quantity of data while ensuring that user privacy is guaranteed?
9. **Cloud Hosting**: does its adoption bring additional security and privacy issues?
10. **Dark Patterns**: how to avoid end users manipulation?
11. **Data Management**: how to define data management policies that assure proper privacy and security levels?
12. **Dependencies**: are my dependencies secure?
13. **Distributed Computing**: how to securely and privately use several computers to collaborate on the same task?
14. **Encryption**: which methods are considered as secure? When does data need encryption?
15. **Hardware**: how to avoid data leaks and intrusions?
16. **Identification**: how to provide secure identification processes for users or external parties?
17. **Instant Messaging and Communication**: how to assess privacy and security issues?
18. **Intelligence**: how to stay up-to-date with the latest issues and risks?
19. **Legislation**: which laws the organizations must be compliant with?
20. **Mobile**: what are the security and privacy issues on the major platforms?
21. **Network**: how to limit security issues and intrusions on networks?
22. **Operating System**: what are the biggest security and privacy issues with the major Operating Systems (OSes)? How to mitigate them?
23. **Policies**: what types of policies the organizations must define? How to define them?

24. **Pseudonymization**: when pseudonymization must be applied? How to ensure a proper pseudonymization process?
25. **Sandboxing**: how to isolate processes to enforce security?
26. **Server Architecture**: how can the server architecture choice impact the security and privacy levels?
27. **Social Engineering**: how to define policies that mitigate the risks of social engineering attacks?
28. **Software**: how to limit security and privacy issues while developing software?
29. **Storage**: how to handle data security at rest?
30. **System Administration**: how to limit security issues on systems?
31. **Web Browsers**: do browsers have privacy and security issues? Are they all equal on those issues?

Excluded Topics

As for the inclusion list, we defined the topics which must not be considered. Indeed, it is unnecessary for us to go deeper than their security or privacy aspects, or they are unrelated to our project scope.

1. **Advanced Compiling**
2. **Advanced Concurrency**
3. **Advanced Cryptography**
4. **Advanced Mobile**
5. **Advanced Virtualization**
6. **Assembly**
7. **Blockchains**
8. **Efficiency and Optimization**
9. **Electronics**
10. **Embedded ICT**
11. **Graphical User Interface (GUI)**
12. **Human-computer interaction (HCI)**
13. **Internet of Things (IoT)**
14. **Quantum ICT**
15. **User eXperience (UX)**

2.2.2 Review Results

This Subsection shows the knowledge collection built during the review, as defined by the protocol explained in Subsection 2.1.2 (Review Protocol).

Access Control

Broadly speaking, access control is a set of policies that restrict access to virtual or physical resources. A proper access management must also support its implementation.

This topic is strongly related to user and administration roles, as to the notion of trust.

As explained by Schneider, 'Access control mechanisms are intended to protect programs and data from corruption, yet still allow sharing of these resources' [5].

A good practice is to apply the principle of least privilege: every entity of a system, which can be users, software modules or processes, must only have access to resources they explicitly need for their purpose. Any other resources access must be denied by default.

Blankstein and Freedman studied in 2014 how to correctly apply isolation and least privilege patterns [6]. Some principles must be followed during system design. First, the portions of the application must be split in isolated components with isolation boundaries. Then, the amount of privilege given to each component must be minimized. Finally, the required privileges of each component must be inferred using dynamic analysis, which is an automated version of the least privilege pattern.

Colombo and Ferrari reviewed in 2018 the particularities of access control for big data contexts [7]. The majority of platforms have basic access control mechanisms, which lead to multiple problems. Unconstrained access are given to high volume of data from multiple data sources, some sensitive and private data are illegitimately accessible, and advanced analysis and prediction capabilities are limited. Multiple requirements must be met for great access control: define fine-grained access control, allow context management, and guarantee the efficiency of access control without any compromises on the platform usability. Some issues are still open in the research field: how to unify the access control model and mechanism, how to provide policy analysis tools, to ensure GDPR fulfilment, or for federated environments, and how to define appropriate access control for streaming analytics, including adaptation for continuous flows.

A comparison of the Attribute-Based Access Control (ABAC) policy has been made in Table 2.1 (Attribute-based access control policies comparison). Some papers contradict others, some of them are complementary: we will combine them appropriately for our collection of knowledge.

Anonymization

Anonymization on data is a process that aims to remove all links related to an entity to its real identity, by modifying its content or structure. Such process is often applied on critical personal data collected from users of an information system.

Anonymization may be made mandatory in certain situations, such as in the medical field or when publishing data for an academic study. Various approaches can be used, but not all of them can be considered as secured.

This process must not be confused with pseudonymization, which is treated in Subsubsection 2.2.2 (Pseudonymization).

Liew et al. used in 1985 a probability distortion to protect the privacy of individuals [8]. The authors defined three steps: first, the underlying density function must be identified with its parameters on the original dataset. Then, series of data must be generated using the estimated function. Finally, the original data must be mapped and replaced with the generated ones. Both datasets have asymptotically the same statistical properties.

Domingo-Ferrer and Torra defined in 2002 an approach to handle sensitive information in tabular data [9]. Sensitivity rules are used to decide whether table cells are sensitive or not, which means that sensitive cells must not be published. Examples have shown that publishing non-sensitive cells may also disclose sensitive information. An a priori assessment on disclosure risks must be made using sensitivity rules, such as (n, k)-dominance, pq-rule or $p\%$-rule. This is explainable by the fact that disclosure risks of contributions increase as the percent within which they can be estimated by an intruder decreases. Two alternatives to sensitivity rules are proposed by the authors: entropy-based sensitivity rule, and complement the a priori risk assessment using a posteriori assessment.

Pfitzmann and Hansen defined in 2010 what are the different privacy-preserving techniques using data minimization [10]. Anonymity of a subject from an attacker's perspective means that the attacker cannot sufficiently identify the subject within a set of subjects, know as the anonymity set. Unlinkability of two or more items of interest from an attacker's perspective means that within the system, the attacker cannot sufficiently distinguish whether these items are related or not. Linkability is the negation of unlinkability. Undetectability of an item of interest from an attacker's perspective means that the attacker cannot sufficiently distinguish whether it exists or not. Unobservability of an item of interest means that the item is not detectable against all subjects not involved with it. A pseudonym is an identifier given to a subject that is different from of the subject's real names.

Kotschy studied in 2016 the requirements to be compliant with the GDPR [11]. If the data is anonymized, the GDPR is not applicable. However, there is still a risk of data being not fully anonymized, and no clear requirement is given in the regulation. If the data is pseudonymized, there is no precise legal consequences. Pseudonymization has no clear and immediate legal advantages.

Yüksel et al. reviewed in 2017 how anonymity is implemented in electronic health services [12]. Five approaches has been found by the authors. Data anonymity, which assures that no relationship can be made between users and their data. User anonymity, guarantees that messages do not give information about their users' identity. Communication anonymity, which hides the link between users and the system. This technique can use onion routing systems like Tor. Ensure unlinkability between users' exchanges. Usage of differential privacy by adding noise in the data.

Murthy et al. studied in 2019 some data anonymization techniques with their characteristics [13]. The generalization technique replaces data values with less specific ones, but keeps them semantically consistent. The suppression technique removes entire parts of data. The swapping technique randomly rearranges the variables. The masking technique changes the characters in attributes. The distortion technique changes the data itself, which a possibility of being reverted. Some techniques are more suitable for specific types of variables.

A comparison of the anonymization techniques has been made in Table 2.2 (Anonymization techniques comparison). Some papers contradict others, some of them are complementary: we will combine them appropriately for our collection of knowledge.

Application Programming Interface

An Application Programming Interface (API) is an interface used for communication between software. It usually exposes endpoints to realize actions in a given service, using the web. An API can be public or private, authenticated or anonymous.

The access of such interfaces can be a problem, both for end users than for the service provider. Sensitive information could be accessed, or unwanted actions could be made by malicious parties.

Several mechanisms can be implemented to restrict access and limit rights to the proper resources. Furthermore, a complete documentation should be designed before building an API.

Hussain et al. studied in 2020 the state of enterprises' API security and their GDPR compliance [14]. An API can be designed to perform actions or to provide access to objects. They can be one of three types: private with a closed access, for partners, designed with efficient access control and authorization mechanisms including rules and policies, or public, which bring potential security threats. An API can be implemented by following either REST or Simple Object Access Protocol (SOAP) approaches. SOAP is more adapted for sensitive data. ML security can fill various security gaps such as addressing new threats, identifying past attacks behaviour, or making predictions. However, it must be compliant with the GDPR, which restricts automated decision-making and profiling, causing an increase into AI-enabled services costs. This regulation requires explaining details of algorithmic decisions, ensuring right of data portability, ensuring the trade-off between algorithmic transparency and accuracy, and allowing users' right of data erasure. Automated decision-making is prohibited without human intervention with the need to be transparent to users. This issue brings new technical challenges, particularly on how to explain those black-boxes to users and on intellectual properties. The data can be localized anywhere, but the in-house approach is a more appropriate solution compared to public cloud. In general, data processing needs consent of data subjects.

Bonne et al. studied in 2017 how to increase transparency and legibility for users [15]. If data are shared with third parties, an API can be given to users for them to consult how their own data are being shared. User should be able to decide if a service is worth to be used by knowing how and what data is shared. They could accept or not such sharing thanks to an assessment of the value of their data, which is difficult to guess without actually knowing what is shared. Three limitations appear with such system: the sharing retroactivity must be handled, the users must use the system to have access to the API, and it does not show internal usage of data.

Ichario and Maarek studied in 2020 which terms of service and privacy policies should be defined for an API [16]. API providers must define the terms of service and privacy policies for developers that will use the said API. Developers can then assess the services compatibility and avoid breaches and threats of termination for non-compliance. The terms should include the guaranteed Service Level Agreement (SLA) level, the conditions to agree to before usage, the privacy policies, the indications on terms changes, the liability, or third parties usage conditions. The authors defined privacy policies as 'the channel through which internet services communicate to their users the data they collect

from them and what it is used for': users can either accept them, which means that they lose control of their data but obtain an access, or reject them, which guarantee them to keep control but without any granted access to the API. Privacy policies define provider's terms that API users must comply to. Figure 2.1 (Concept map of perceptions emerging from content analysis of discourse [16]) shows the biggest issues and their related mitigations.

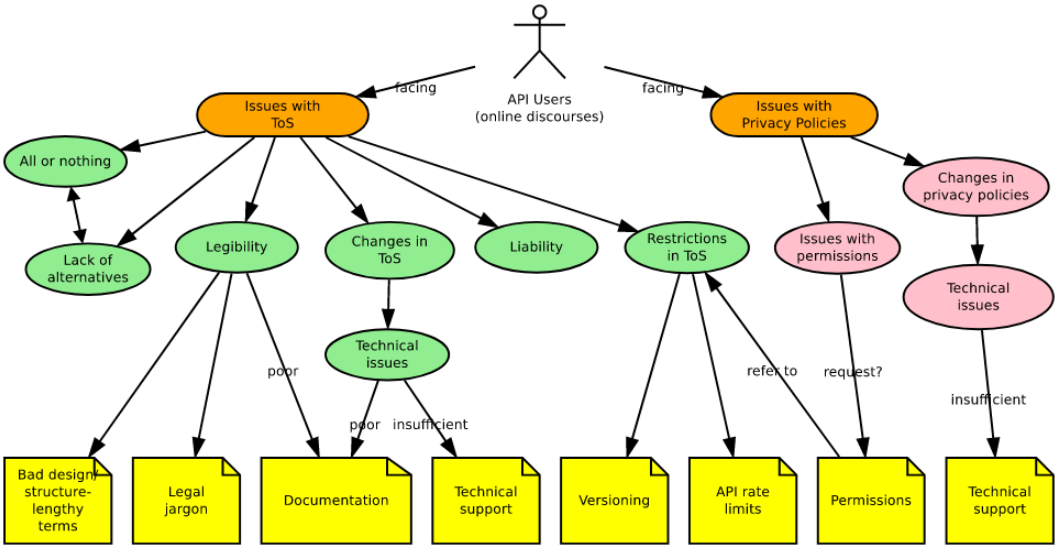


Figure 2.1 *Concept map of perceptions emerging from content analysis of discourse [16]*
Source: `http://bit.ly/3w1GkRO` (accessed 14th January 2023)

Sharieh and Ferworn studied in 2021 how to reinforce security using chaos engineering [17]. Security chaos engineering can be used to both expose vulnerabilities and enhance security. Multiple techniques exist to detect automated attacks, such as monitoring the traffic, apply a quota management, whitelisting [sic], or traffic throttling. HyperText Transfer Protocol (HTTP) header fields can be used to achieve code injection attacks. Chaos engineering is a method that simulates unpredictable failures to make systems more resilient. Distributed Denial of Service (DDoS) attacks are difficult to identify: each malicious client sends normal traffic volume, while adapting the said volume by detecting rate-limiting controls to avoid any detection. Bots can be detected by searching for patterns such as abnormal behaviour, persistent attempts, unusual error rates, suspicious client requests, or by using ML models. Those models need historical data and more research to achieve greater results. Chaos security applies empirical exploration to verify how a system behaves. It is implemented by building a hypothesis around steady-state behaviour, varying real-world events, running experiments in production, automating experiments to run continuously, and minimizing blast radius.

A comparison of the vulnerabilities, attacks and mitigations mentioned by the papers has been made in Table 2.3 (Application programming interface vulnerabilities, attacks and mitigations comparison). The sources being complementary, their combined knowledge will be used for our collection of knowledge. The `Summary` column has been removed for readability purposes.

Artificial Intelligence

AI brought broad and novel possibilities to a lot of different fields. It can enable new ways of exploiting data in order to gain knowledge on particular topics. The mastery of this technology has led to its adoption in many services today.

AI models require a big amount of data in order to be accurate in their classifications or predictions. This characteristic includes privacy risks for users, whose data is being collected from service providers in order to improve their models. New ways of protecting user data are being studied to improve the user privacy of services.

Shokri and Shmatikov studied in 2015 how to preserve privacy on DL models [18]. DL models require massive data collection, including sensitive user data which are kept indefinitely. A system can be designed to learn a model without sharing input datasets, using the characteristic of Stochastic Gradient Descent that can be parallelized and executed asynchronously. Only small subsets of key parameters are exchanged, whilst improving accuracy with external data without having access to them. Evaluations shown an accuracy close to centralized models, with negligible utility loss. The NN parameters leak risks is mitigated using differential privacy on their updates, thanks to the sparse vector technique.

Li studied in 2018 the AI-specific cybersecurity defences [19]. AI-powered systems can be attacked or deceived, resulting in incorrect classification or prediction results. Federated learning uses terminal devices that use their own data for the training phase. However, the user privacy must be assured and model manipulation and/or steal must be mitigated. There are various ways for building safe distributed models: One of them is to avoid gradients leakages using homomorphic encryption, based on the approach defined by Shokri and Shmatikov [18]. This method adds a large computational overhead. Otherwise, a federated learning environment can be built including an aggregation protocol which securely computes the sum of parameters computed by devices. Or, ML classifications can be done over encrypted data, but it lowers the model accuracy.

Phong et al. studied in 2018 how to preserve privacy using homomorphic encryption for DL models [20]. The authors' goal is to enable collaborative learning of a NN using local dataset of all participants, without actually sharing the data. All participant compute their local gradients, by training their local model, and then send a portion of their gradients to a central server. The latter use additively homomorphic encryption and asynchronous Stochastic Gradient Descent to compute a general model. However, a trade-off must be taken care of between accuracy and privacy, which consists of finding the correct amount of local gradients to share. Because a small faction of gradients can leak useful and therefore private information, homomorphic encryption is used to enable computation on the data without being able to know its value as a plaintext. This approach has three effects. On the security side, the central server can not leak any data. On the accuracy side, an identical accuracy is achieved compared to a corresponding model trained on a global dataset built from the joint local datasets. Finally, on the overheads side, an increase in communication is caused by the sharing of the gradients, and an increased computation time is required to achieve the same model accuracy.

Vellido studied in 2019 the societal issues brought by AI usages [21]. The GDPR mandates a 'right to explanation' made by 'automated or artificially intelligent algorithmic systems', which legally binds the data controller to provide explanation about AI tools to requesting citizens if their personal data are used. There is therefore a need for interpretable and explainable models in order to justify their decisions. DL models can be compared to opaque black boxes: such systems are not capable to self-explain their operating processes.

Xue et al. [22] listed in 2020 the threats and countermeasures on ML. Models security can be evaluated by testing their design, by using testing frameworks, by performing AI specialized penetration tests, and by choosing the most appropriate metrics to test models. Some of those metrics are the accuracy, confusion matrix, precision, recall, Receiver Operating Characteristic (ROC) curve, or the Area Under the ROC curve (AUC).

Zhang et al. reviewed in 2021 the ethical and privacy issues discussed in papers around AI [23]. The most concerned technologies that could cause ethical issues are ML, data analysis, robots, intelligent systems, and cloud technologies. The key concerns are to ensure fairness, discrimination barriers, data privacy, cybercrime mitigations, fraudulent behaviour detections, and machine ethics.

Liu et al. reviewed in 2021 how to design privacy-friendly ML models [24]. The authors found three private learning schemes: by applying homomorphic encryption on data or model, by applying obfuscation using differential privacy, which adds noise, and by using aggregation, which guarantees that parties keep their own dataset private whilst still being able to learn collaboratively.

Liu et al. reviewed in 2021 how to ensure privacy and security with DL models [25]. Future directions to improve DL models security and privacy has been found: lightweight privacy-preserving techniques, intellectual property protection of DL model, generic privacy-preserving techniques, and systematic evaluation of adversarial defences.

Zhu, et al. [26] studied in 2021 how to add differential privacy in AI processes. The calibrated randomization embedded in differential privacy brings benefits to some AI algorithms because of multiple properties: it preserves privacy, which is its original purpose, it improves stability, thanks to the unchanged models output probability if an individual record is changed, it brings better security by reducing the impact of malicious participants, it guarantees fairness by re-sampling the training data from the universe, and it enables composition, which means that any step that satisfies differential privacy principles can be integrated in the algorithm. All properties do not have the same effect on the different types of AI. For ML, it preserves privacy and improves both stability and fairness. In the other hand, an optimal trade-off between privacy and utility needs to be found and optimized. Furthermore, it is only suitable for loss functions that do not contain any regularization steps. Moreover, some situations do not have knowledge of the utility of each sample, which is used by the exponential mechanisms of the re-sampling step. Regarding DL models, which include both distributed DL and federated learning, differential privacy can be applied locally. A global implementation would not protect the system against an attacker pretending to be trustful. It can also be used to destroy redundancy in order to avoid model inversion attacks. More specifically for federated

learning, an aggregate of re-weighted loss functions can be used with clients having different weights to improve their learning accuracy, and then joint using differential privacy to make different model updates according to client's requirements.

A list and comparison of all attacks and mitigations that have been mentioned by the related papers has been made in Table 2.4 (Artificial intelligence attacks and mitigations comparison). The sources do not contradict themselves: we will combine their data for our collection of knowledge. The `Summary` column has been removed for readability purposes.

Authentication

Authentication is an action done by a system that verify whether the identity given by an entity is valid and trusted or not. Such evaluation must ensure that this identity can not be falsified.

This is mainly focused on security issues, the privacy being more related to user identification which must follow company policies.

Rabkin listed in 2008 the issues on security questions for lost passwords [27]. The author realized tests on personal banking websites using security questions as a lost password retrieval: many of them rely partially on security questions with serious usability and security weaknesses. The hardness of this method is weakened as personal information becomes ubiquitously available online. Two kinds of security questions have been found: sensitive questions, which are not necessary private, and personal questions, related to users' background or to their family. Allowing users to define their own questions is not very common. Alternatives exist, such as email-based resets, often considered as secured, use data already held by the organization, which imply that the level of security depends on nature of the source, or asking for a series of preferences judgements, technique not used in the industry. Automatic attacks must be blocked, by using for example Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHAs). The author found that personal questions are more secured than the sensitive ones because of questions being more varied and of public leaks of sensitive data. The biggest weaknesses in personal security question are that they are inapplicable, not memorable, ambiguous, guessable, attackable, and automatically attackable. Users treat memorability rather than security as the dominant factor in choosing security questions. Some well-known attacks are random guessing, automatically using online information, dedicated human attackers, and personal acquaintance. Some mitigations can be enforced, such as survey distribution of answers, users' education, usage of ephemeral answers, and ask users for durable and offline answers.

Schechter et al. reviewed in 2009 the security and reliability of authentication using secret questions [28]. The authors found that 17% of users' security answers can be found by their acquaintances. Users forget 20% of their own answers within six months, and 13% of answers could be guessed within five attempts by guessing the most popular answers of other participants. A single personal question is not sufficiently secure for authenticating users. User-written questions could be harder to attack, but only if they are sufficiently private and unpopular. The proportion of popular questions should be reduced.

Jain and Nandakumar studied in 2012 security and privacy concerns of biometric authentication [29]. Biometric systems recognize individuals based on their anatomical or behavioural traits. They are used to ensure that only legitimate or authorized users can get access to an entity. Their unique advantages are their deterrence against repudiation, and their multiple identity detection. Biometric systems rely on similarities between two biometric samples, not on a perfect match: challenges can lead to false non-matches or false matches. This approach leads to vulnerabilities such as denials of service, with legitimate users being not recognized, or intrusions, with impostors being incorrectly identified as legitimate. Multiple adversary attacks exists: coercing or colluding with insiders, exploiting insiders' negligence, manipulating the procedures of enrolment and exception processing, direct attacks on sensors, feature extractor, or matcher module. Those attacks can be carried out using trojan horses, man in the middle or replay attacks. They are also applicable to password-based authentication. The major vulnerabilities are spoof attacks on users' interfaces and template database leakages. A mitigation against spoofing is to detect liveness during the tests. Data leakages are sensitives because of biometric traits being irrevocable. A mitigation against template database leakages is to enforce template security by applying a trade-off between non-invertibility, discriminability and revocability. To this end, two generic approaches be be applied: biometric feature transformation and biometric cryptosystems. Generating a secure sketch of traits can be realized by using fuzzy commitment and fuzzy vault. However, biometric systems include some major issues that need to be answered: who own biometric data? Is this usage proportional to the need? What is the optimal trade-off between service security and user privacy?

Velásquez et al. listed in 2018 various authentication schemes and methods [30]. Multifactor authentication is a combination of different authentication factors. Choosing the adequate authentication schemes or methods depends on the contexts. The authentication factors come from knowledge, what users know, possession, what they physically own, or inherence, what users are. The combination of the knowledge and possession factors is very predominant in multifactor authentication methods. Three-factor authentication is well researched but less applied. For both methods, the combination of text passwords and smart cards is the most popular. The comparison and selection of schemes are made with usability, security and cost-related criteria. The authors gave advices on which frameworks can help in the decision of authentication schemes or methods, according to different contexts.

Ibrokhimov et al. studied in 2019 the details about multifactor authentication [31]. Digital multifactor authentication is one of the best method to implement a secure authentication, but can be frustrating for users. Some greatly used multifactor authentication methods are fingerprints and user-specific random projection, threshold cryptography (One Time Password (OTP) approach), multimodal biometrics, or cloud-based infrastructure. The latter can use third parties authentication. Different entities can be used for authentication, such as smart cards, OTP, cryptographic techniques, multi-modal biometric systems, or tokens.

Authorization

Authorization is a process that verify an entity access request in order to grant its access to resources, by following rules from the access control policy. The main challenge of authentication is to ensure that every access made to system resources must pass by its verification, without exception.

This process mainly concerns the security aspect of a system: authorizing a user must not be lightened in order to preserve their privacy. Indeed, collecting proof of access requests is a great method to fight an intrusion in a system. However, authorization details must not be accessible by a party without valid reasons.

Kagal et al. reviewed in 2004 the privacy and security concerns in semantic web services by defining semantically rich security and policy annotations for Web Ontology Language Semantic (OWL-S) service descriptions [32]. Policies should be part of the representation of (semantic) web services and respond to a bunch of questions, such as who can use a service under which conditions, how information should be provided to the service, and how provided information will be used later. Those policies should be of different kinds: privacy policies, that define under what conditions information can be exchanged and what are the legitimate uses of that information, and authorization policies. Single requests can have policies of their own. Ontologies and markup are some proposed approaches to capture security information of web service input and output parameters. The authors defined policies that are transformed into informal contracts represented in Rei (Resource Description Framework Schema (RDFS) based language) that also include a prioritization mechanism to resolve conflicts. Providers can be discovered and selected using the policies. A way of enforcing privacy and authentication is to use encryption standards for OWL-S communication independently of the transport protocol security.

Fett et al. studied in 2016 how to secure the OAuth framework [33]. OAuth allows users to grant websites access to their resources, which can be data or services, at other websites. This operation is called an authorization. Its central security properties are authorization, authentication, and session integrity. Four exploitable attacks have been found, but mitigations are given for new and existing deployments: multiple new RFC have been drafted from the respective working group, with guidelines given to secure OAuth implementations. A complete security model is given to enforce OAuth processes.

Best Practices

A best practice is a piece of advice given by a party, which has usually no official status in the concerned field but generally trusted because of its background or by past events. There is no obligation to apply such advices, but doing so is considered better than ignoring them.

They can concern both privacy advices than security ones. We did not find big amounts of generic best practices in the academic field.

Larsson and Sigholm studied in 2016 security issues of the web ecosystem [34]. The authors found three sources of problems that can affect the introduction of best practices. First, insecure configurations can remain widespread for over a decade. Secondly, introduction of best practices only affects moderately the decline of insecure configurations. However,

highly publicized security flaws have a significant impact. Thirdly, economic incentives for website owners to provide secure services are too weak. Other levers of influence as legislation or blocking noncompliant sites have a bigger impact.

Ma and Pearson [35] listed in 2005 the best practices ensuring information security, initially released by the ISO 17799 document. It answers questions such as which standards should an organization implement to achieve their information security objectives, or what management practices are perceived as critical by information technology professionals. ISO 17799 is widely accepted and recognized as best practices being applied by information security professionals. The authors found that most of the security dimensions and items covered under the ISO 17799 document are highly valid. This resource has nowadays been replaced by ISO 27002 with updated content.

Big Data Privacy

This topic is mainly focused on privacy, because security aspects can be compared to regular storage, processing or transport of data. Indeed, there is mainly changes on the amount of data attributes and on data quantity.

The security concerns of this topic is mainly brought by the distribution model: to process such amount of data, several computers must be used in parallel, which include this additional task to the whole data usage. This will be treated in the Subsubsection 2.2.2 (Distributed Computing)

Jain et al. studied in 2016 the major privacy and security concerns, with their requirements [36]. The data generation phase must restrict the access to data and allow data falsification. The data storage phase must perform attribute-based encryption, enforce homomorphic encryption, encrypt storage paths, use hybrid clouds, and allow data integrity checks. The data processing phase must be able to extract information without violating user privacy using de-identification. Various techniques exist to this end, such as K-anonymity, L-diversity, T-closeness, HybrEx model, privacy-preserving aggregation (homomorphism), differential privacy or identity-based anonymization. The data publishing phase must also include privacy-preserving techniques.

Abouelmehdi et al. studied in 2017 the health industry limitations on big data resources [37]. The authors found that the health industry is one of the most susceptible ones to publicly disclose data breaches. Possible mitigations are strong authentication, enabling encryption, data masking and strong access control. Some legislations regulate user privacy, but different countries have different policies and laws. Some privacy-preserving techniques can be used, such as de-identification (K-anonymity, L-diversity, T-closeness), the HybrEx model, and identity based anonymization.

Mehmood et al. made an overview of the big data privacy preservation mechanisms with their challenges [38]. Mitigations to avoid privacies breach have been defined by the authors. The data generation phase needs access restrictions and data falsifications. The data storage needs various schemes of encryption, a usage of hybrid clouds, and various schemas to ensure proper integrity verifications. The processing phase needs Platform Security Processor (PPDP) techniques, realize knowledge extraction using privacy preserving clustering or classification, and association rule mining techniques.

Soria-Comas and Domingo-Ferrer listed in 2015 the challenges raised by big data in privacy-preserving data management [39]. Some principles are requested in regulations that aim to protect personally identifiable information: lawfulness, consent, purpose limitation, necessity and data minimization, transparency and openness, individual rights, information security, accountability, and data protection by design and by default. Threats can appear if no anonymization is enforced, such as data breach, internal misuse by employees, unwanted secondary use, changes in company practices, or government access. Anonymization is a solution, but it must be effective. However, it can remove the purpose of big data analysis. Privacy models must comply with volume, variety and velocity, and satisfy the composability, computational cost, and linkability principles.

Cloud Hosting

Nowadays, a lot of services are hosted in the cloud. This paradigm brings new issues in terms of security, but more particularly for user and company privacies. Indeed, the data storage, transport and processing are made on someone else's computers. A new bond of trust must be established between service providers and cloud providers.

Organizations leasing shared resources from cloud providers become infrastructure tenants rather than owners.

Molnar and Schechter studied in 2010 the issues and mitigations of self-hosting environments versus cloud hosting [40]. Multiple threats appear when migrating from owning to housing:

- **Risks on the infrastructure assembly**: physical threats, which can be avoided by testing the components, using Trusted Platform Module (TPM), or making audits, or on software and human resources that fails to meet the promised standards or being compromised: can be mitigated by defining multiple admins, limiting admins' access, and carrying out background checks of employees.

- **Contractual threats**: cost-overrun attacks, can be avoided by setting quotas or ensuring that the provider absorbs bulks, deceptive billing, avoidable by enabling tenants to do their own infrastructure tests, or by reporting resource consumption, captivity, avoidable by ensuring providers homogeneity and by reviewing long-term contracts cost prediction, or bankruptcy, which need to assure that the rights to access infrastructure, and the funds to continue short operation are guaranteed.

- **Legal threats**: can create indirect legal coercion, secret search, or direct and indirect jurisdictional exposure. Can be avoided by enabling data location choice.

Some threats also appear when migrating from dedicated to shared infrastructure:

- **Threats from other tenants**: by direct breaches, can be mitigated by hypervisor and network isolations, by side channel attacks, avoidable using the same isolation techniques, or by denial of resources, resource thefts, and collateral damage to shared reputation. Those last threats can be mitigated by securing the mapping between communications and tenants.

- **Threats from legislation**: jurisdictional collateral damages

- **Threats on availability and costs of shared resources**, by under provisioning, avoidable with attestation-based audit mechanisms and spare capacity audits, or by collateral denial of shared resources, mitigated using resource quotas.
- **Threats caused by diminished audit, detection, or incident response capabilities**: can be caused by forensic restrictions, can be avoided by forcing providers to investigate breaches.

Some security benefits are brought by the economies of scale principle: providers can amortize fixed costs, which brings more specialization, adapted infrastructure, full service security, leverage data from multiple tenants, relationships through recurring interactions with regulators and law enforcement.

Zhou et al. investigated in 2021 what are providers concerns on security and privacy issues [41]. Providers have five goals to achieve adequate security: ensure availability, confidentiality, data integrity, control and audit. Some legal issues can be mitigated by creating additional roles from cloud infrastructures and by great handling of third parties. Some acts fail to protect user privacy from the government and third parties in a cloud environment. Multi location can bring issues in a legislation perspective.

Mathisen studied in 2011 the security-related challenges and solutions for cloud environments [42]. The author found various policies issues, such as inside threats, avoidable by creating adapted employees' governance, access control, can be mitigated by enabling additional authentication factors or by creating confidence between provider and tenant, system portability issues, avoidable by avoiding provider link-in or by using open standards. The software security issues are caused by virtualization technologies, which can be mitigated by applying updates and keeping tenants isolated, by host OS, avoidable by choosing a simple and with minimal services OS, by guest OS, whose issues can be mitigated by giving tenant responsibility and informing them about risks, or by weak data encryption. Some physical security-related issues can be caused by backups, that should be done by tenant directly and also by using offline storage, by the server location, avoidable by choosing adapted rooms, backup power and controlling entrances, or by firewalls, avoidable by activating a default deny mode, defining additional per-instance filters, and by enabling DDoS protections.

Sengupta et al. listed in 2011 the most important research directions in the cloud market [43] The most common concerns are related to cloud infrastructures, platforms and shared codes, data, accesses or compliances. To be mitigated, concerns related to those points should be addressed. Some more advanced issues are abstraction problems, lack of execution controls, third party control of data, and multi-party processing. A strong model to secure operations would be to enforce trust, to create context specific access model within data and to preserve privacy. The authors defined a framework that characterizes the security requirements of the application, then characterizes and reviews the cloud provider's security strengths and vulnerabilities, and finally maps the two previous steps to perform a fit analysis.

JPC Rodrigues studied in 2013 the electronic health records concerns when using the cloud [44]. The author defined suggestions to be considered before adopting the cloud: how are handled the data security, the regulatory compliances, the user authentication,

the data separation, and the legal issues. Providers' certifications must be reviewed: it could include *SAS70 Type II*, *PCI DSS Level 1*, *ISO 27001*, or *FISMA* certifications. The employee lifecycle policies must also be reviewed: how are defined the account provisioning, account review, access removal, and password policy. The business continuity management must also be known, such as the provider's availability, incident response, and company-wide executive review. Finally, the network security should be considered, with mitigations for DDoS, man in the middle, Internet Protocol (IP) spoofing, or port scanning attacks.

Xiao and Xiao reviewed in 2013 the threats related to the security and privacy in cloud [45]. The author found that the most representative security and privacy attributes are confidentiality, integrity, availability, accountability, and privacy preservability. The biggest challenges building secure and trustworthy cloud systems are outsourcing, multi-tenancy, massive data and intense computation issues. The security ecosystem should be modelled considering the three participants: the service user, the service instance, and the cloud provider. The authors found four kinds of vulnerabilities, with threats and mitigations: Virtual Machines (VMs) co-residence, loss of physical control, bandwidth under-provisionning, cloud pricing model. Figure 2.2 (A summary of research advances in cloud security and privacy [45]) shows threats and mitigations in a summarized overview.

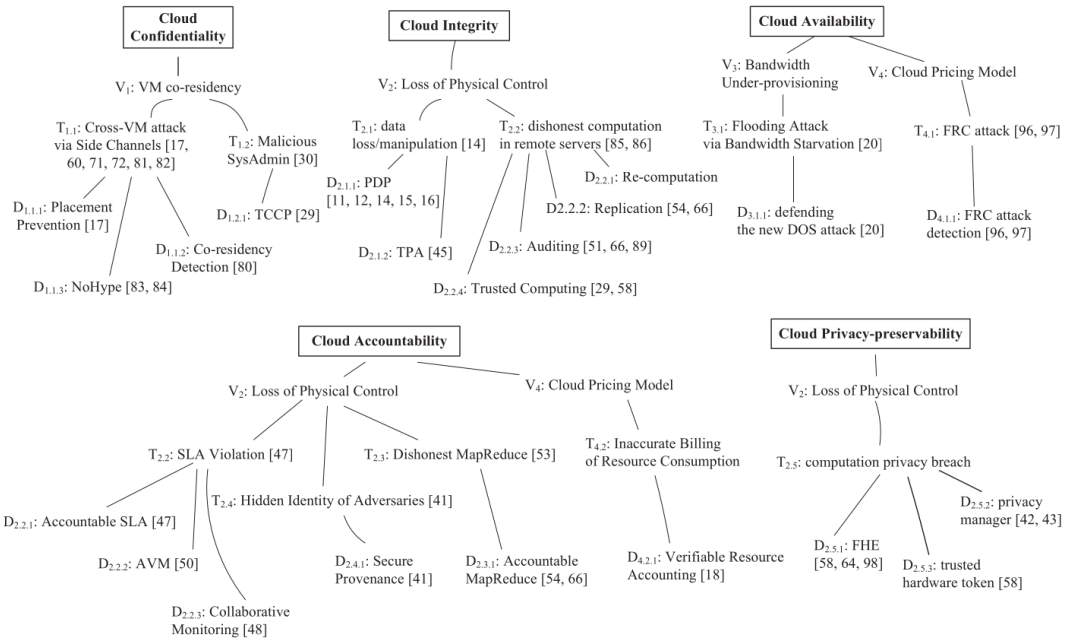


Figure 2.2 *A summary of research advances in cloud security and privacy [45]*
Source: https://bit.ly/3k15Zaq (accessed 10th January 2023)

Jathanna and Jagli studied in 2017 what are the threats in cloud computing [46]. The author found that the biggest threats are compromised credentials and broken authentication, data breaches, hacked interfaces and APIs, exploited system vulnerabilities, account hijacking, permanent data loss, inadequate diligence, cloud service abuses, and DDoS attacks. Designing a service model includes some security challenges such as malicious attacks, backup and storage issues, service hijacking, and VM hopping. The deployment model also integrates security challenges like Platform as a Service

(PaaS) security issues, third-parties relationship management, development life cycle issues, underlying infrastructure security, cloning and resource pooling, unencrypted data, authentication and identity management, network issues, eXtensible Markup Language (XML) signature element wrapping, browser security, flooding attacks, and Structured Query Language (SQL) injection attack.

Because of the high complexity of showing information on cloud security issues, threats and mitigations, we did not use a table to compare them. A blend of those points will be made directly for our collection of knowledge.

Dark Patterns

Dark patterns are User Interfaces (UIs) that have been optimized to manipulate or mislead the users. This method aims to increase profits of the service provider, whether it is financial or in terms of time of use. Cognitive biases are often used to conceive such UIs.

Dark patterns are unethical, and can overtake on user privacy, mainly because of their optimizations to consume more time while using services, and therefore collecting more user data.

Two useful online sources on dark patterns: the *privacypatterns.eu*[8] and *privacypatterns.org*[9] websites.

Bösch et al. studied in 2016 what are the privacy dark strategies and patterns [47]. First, the privacy by design principle must be applied. In order to avoid to design interfaces that include dark patterns, some rules must be applied into the development process: proactive not reactive, privacy as the default setting, privacy embedded into design, ensure full functionality, enforce end-to-end security, assure visibility and transparency, and guarantee respect for user privacy. Privacy considerations must be included into the entire development process. Some strategies take advantage of the psychological constitution of human beings, which often cause users to not have the motivation or opportunity to resist them. Hoepman defined multiple privacy design strategies: minimize, hide, separate, aggregate, inform, control, enforce, demonstrate.

Mathur et al. reviewed in 2019 all the dark patterns used in eleven thousand shopping websites [48]. The authors' study is one of the firsts large-scale evidences for the most used dark patterns: 1818 instances of them have been found on 1254 websites, which represents 11.1% of the data set. The dark patterns most relevant characteristics are that they are asymmetric, covert, deceptive, hides information, and restrictive. The human biases that are used are anchoring effects, bandwagon effects, default effects, framing effects, scarcity biases, and sunk cost fallacies. Third-party entities can provide websites the ability to implement dark patterns.

Di Geronimo et al. studied in 2020 the most applied dark patterns into mobile applications [49]. The authors have tested 240 *Android* applications: 95% of them contain one or more dark patterns. Most of the time. users can not perceive the presence of malicious designs. This study included 584 respondents.

[8] *privacypatterns.eu* source: `https://privacypatterns.eu` (accessed 31st October 2022)
[9] *privacypatterns.org* source: `https://privacypatterns.org` (accessed 31st October 2022)

Luguri and Strahilevitz studied in 2021 what is the power of dark patterns [50]. The authors found that users exposed to mild dark patterns are more than twice as likely to sign up for a dubious service than a control group used in an experiment. Users in aggressive dark pattern conditions are almost four times as likely to subscribe. Aggressive dark patterns generate a powerful backlash, mild dark patterns do not. Less educated user are more susceptible to mild dark patterns than their well-educated ones. Some legal frameworks exist for addressing dark patterns, such as the Federal Trade Commission in the United States.

A comparison of the different dark patterns mentioned by the related papers has been made in Table 2.5 (Dark patterns categories comparison). All sources do not contradict themselves: we will therefore use their combined knowledge, including blending of similar data.

Data Management

Data management includes various aspects that handle data in a system, such as its storage, security, architecture or quality.

Our scope is restrained on security and privacy, but through the whole concerned aspects. Some of those aspects are review in topics of their own, because of data management being a broad concept.

A comprehensive guide to data management technologies has been published by Petković and Jonker [51] and is well recognized within practitioners.

Ashley et al. defined in 2002 how to manage collected personal data in a sensitive, trustworthy way [52]. The authors found that risks when personal information are not well handled, and can cause legislative penalties, brand and reputation erosions, or lawsuits. The Organisation for Economic Co-operation and Development (OECD) defined what are the privacy phases: notice, collection, cataloguing, control, release, recording, response. The authors created a framework with data management building blocks:

- Define an enterprise privacy policy
- Deploy a policy to the ICT systems
- Record consent of end users
- Enforce the privacy policy and create an audit trail of access to privacy-sensitive information
- Generate both enterprise wide and individualized reports showing accesses to privacy-sensitive information and their conformance to the governing privacy policy

Efraimidis et al. reviewed in 2009 how to integrate privacy in personal data management [53]. Data protection can be enforced by either the owner side or the provider side. Different schemes for representing personal data and policies exist, such as *P3P*, *CPExchange*, and *DISCREET*. Hierarchical categories have been defined to organize personal data, including some sub categories. The related policy components are principals (entities), data (every single item), purpose (entitles principals to retrieve data), and

usage restrictions (limit access rights). The policy includes the usage of licences which define the data involved, the valid purposes of data retrieval, and the rules to provide full or restricted access. Contracts are also included, which are arbitrary sets of licences. The paper exposes a system applying those concepts.

Squicciarini et al. studied in 2009 the problems of collaborative enforcement of privacy policies on shared data [54]. Two solution could be used. The first is to map the user collaborative policy specification to an auction based on the Clarke-Tax mechanism. This approach selects the privacy policy that maximizes the social utility using truthfulness among co-owners. The second solution is to apply data co-ownership. The potential owners of posted data can be identified using tagging features or files metadata. Some requirements must be met for valid collaborative privacy management: must ensure content integrity, is semi-automated, must be adaptive, and integrates group-preference. The private box implementation is proposed by the authors, which is a collaborative management of shared data based on pictures.

Mansour et al. created in 2016 a decentralized platform to share personal data [55]. User data is stored in web-accessible personal online datastores named pods. One or more pods can be used and easily switched across different providers. Applications can get access to the data using well-defined protocols, a decentralized authentication and access control mechanism to guarantee data privacy. This technology allows similar applications switching, applications on multiple platforms, and the advantages of decentralized architectures. The *Solid project*[10] implements this technology.

Dependencies

Software dependencies are a major component of almost every service, with features, capabilities or toolboxes that can directly used after their integration. A lot of time and energy are saved, but a new bond of trust must be established with such integrations. However, specialized libraries can bring more security to a module than trying to implement the feature ourselves.

There is also a domino effect: a dependency may include others. Such situations may lead to major security issues, or even *stability issues*[11].

The *OWASP organization*[12] has published two useful tools to mitigate dependencies risks: *Project Dependency Check*[13] and *Project Dependency Track*[14].

Zimmermann et al. studied in 2019 the security threats in the Node Package Manager (unofficial name) (NPM) ecosystem [56]. The openness of NPM has boosted its growth: more than 800,000 free and reusable packages available. This popularity brought security risks, as recent incidents of single packages have broken or attacked targets using software running on millions of computers. Individual packages can impact lots of projects, using maintainer accounts that can inject malicious code into the majority of all packages.

[10] *Solid project* source: https://solidproject.org (accessed 11th January 2023)
[11] *stability issues* source: https://xkcd.com/2347/ (accessed 17th October 2022)
[12] *OWASP organization* source: https://owasp.org (accessed 31st October 2022)
[13] *Project Dependency Check* source: https://owasp.org/www-project-dependency-check/ (accessed 31st October 2022)
[14] *Project Dependency Track* source: https://owasp.org/www-project-dependency-track/ (accessed 31st October 2022)

A lack of packages maintenance causes many packages to depend on vulnerable code. NPM suffers from single points of failure and unmaintained packages which threaten large code bases. One average package gives implicit trust on 79 third-party packages and 39 maintainers, which bring a large surface attack. Highly popular packages influence many other packages: often more than 100,000. Up to 40% of all packages depend on code with at least one publicly known vulnerability. The major security risks are locked dependencies, heavy reuse, micro-packages, no privilege separation (all packages have complete access to the application), no systematic vetting, and publishing model. The most known threat models are malicious packages, exploiting unmaintained legacy code, package takeover, account takeover, collusion attack. The authors defined some potential mitigations: raise developer awareness, warning about vulnerable packages, code vetting, training and vetting maintainers.

Wang et al. defined in 2022 a novel approach that integrates the interdependency among high-level security requirements [57]. Dependencies between security requirements may cause additional vulnerabilities. Vulnerabilities should be identified using static analysis, even if it raises high false positives and misses true vulnerabilities, and security testing, which is highly precise such as dynamic taint analysis and penetration testing. Precise tests should be launched when software is isolated, but security requirements may be violated on interactions. Up to 70% of total software errors are caused by interacting requirements. 20% of most dependent requirements are responsible for 75% of all dependencies. Another approach is to use automated requirements traceability based on information retrieval algorithms. The authors proposed a new approach to integrate horizontal and vertical traceability, using two levels of security requirements: a higher level for requirements specified in policies and regulations, a lower level for ones concretely implemented. A mixture of manual dependency is first done among higher level requirements, then automatically trace them with the lower level ones.

Distributed Computing

Distributed computing consists of sharing data and processes through multiple hosts using a network in order to complete a common task. Such methods must be handled by trusted hosts, even if they do not necessarily hold the whole data knowledge by themselves.

Privacy issues are caused by the trustworthiness of the participating nodes, which can access to complete or incomplete data that can concern sensitive information about the users. Security issues can emerge due to the data being shared across different nodes, that can have different policies or software stacks.

Georgiev and Georgiev studied in 2001 how to establish the trustworthiness and role of each component in distributed computing environments [58]. The authors found two categories of security threats: centralized systems threats, amplified by distribution, and distributed-specific threats, brought by distribution requirements such as scalability, interoperability, interconnection, untrusted nodes, different OS and applications suites, and multiple security policies. Security policies should be designed without regards to leaks and weaknesses of the nodes: they must be addressed independently. To this end, social and technical aspects must be considered. The authors stated a common security model optimized to provide interoperability, must establish the degree of trustworthiness of each component. It includes identification and authentication, access control, confidentiality,

non-repudiation, and availability. The issues that components face are untrusted partners (workstations or servers), untrusted communication media (physical links), untrusted intermediate systems (routers, gateways), untrusted clients (software), trusted user/client identity (unique identity), trusted server identity, trusted administration. Components can migrate between categories, for example with mobile roaming or changes on the network trust. The third parties' authentication services are one of the largest controversial and challenging issues. All distributed application servers and database servers should trust servers using two-way authentication, certificates, message addresses, or content certification. Partners should be trusted using levels of trust, with different evaluations of used software. All partners must be untrusted by default, except for security administrator and third-party authentication services.

Wang et al. analysed in 2004 how to handle security policies reconciliation [59]. A collaboration between two organizations includes that their policies must be resolved. Reconciliation algorithms find a policy that is consistent with all domain policies. If unsuccessful, requirements altering or abstinence can be applied. Policies provisioning includes complex dependencies which include decisions about some particular aspects of the policy that can affect subsequent options. Such processes are also subjects to preferential behaviours. Other reconciliation approaches exist, but are limited according to the authors. The authors' framework is defined using hierarchical graphs, including multiple partial orders resolution and preferences reconciliation. An implementation showed that inherent overhead is negligible in real-field applications.

Kher and Kim listed in 2005 the main challenges, techniques and systems regarding secured distributed storage [60]. Humongous quantities of generated data, which must be shared, replicated, and kept online for various performance, availability, and recovery requirements make systems more vulnerable to security breaches. Several security features should be considered:

- Authentication and authorization, through all data life cycle
- Availability, which includes backup and recovery
- Data confidentiality and integrity
- Key sharing and key management with an efficient and scalable management
- Auditing and intrusion detection
- Usability, manageability and performance

The authors defined three storage systems classification. The first is networked file systems: a server authenticates users and checks any access privileges. It assumes the file servers and the system administrators are trusted. It does not include end-to-end data security. The second is cryptographic file systems: they enable end-to-end security using cryptographic operations natively in the file system, on the client side in order to protect data from both the server and unauthorized users. The server is minimally trusted, and not included in the process. The third is storage-based intrusion detection systems: they monitor activities related to data and look for manifestation of an attack. Security issues are exposed into the paper for each category of storage. Categories can be compared

using the following criteria: used authentication for entities and messages, access control type, end-to-end data and metadata confidentiality support, end-to-end key management, revocation, non-repudiation, key storage, and long-term key management.

Tyagi studied in 2012 the problems of distributed function computation under privacy constraints [61]. A collective computation over correlated data must not reveal the value of a specified private function computed by each of the terminals. If so, such functions are therefore "securely computable". The paper gives necessary and sufficient conditions for secure computation of given functions. A class of functions are securely computable if and only if the conditional entropy of data given the value of private function is greater than the least rate of interactive communication required for an appropriately chosen multiterminal source coding task.

Encryption

Multiple algorithms can be used to encrypt data. The main issue is to ensure that the used algorithms, protocols and parameters are up-to-date with the current context.

Putting encryption is place do not resolve all privacy issues: metadata can still be useful to intruders or third parties to gain knowledge about users. For example, a service provider can easily understand why a female user called an abortion clinic number without being able to listen to the actual conversation, which can bring privacy disclosures.

El Makkaoui et al. studied in 2015 how to enable homomorphic encryption inside cloud environments [62]. Providers' ability to access sensitive user data is a major obstacle in the adoption of cloud services. Homomorphic encryption allows operations on encrypted data with the same results after treatment as with raw data. Several categories of encryption can be used, some of them have limited available operations or limited representation of data. The three main challenges of this technology is its efficiency with limited operations and performances, its robustness which is based on the size of the key, and its delay due by great encryption, decryption and processing times.

Yassein et al. reviewed in 2017 the major asymmetric and asymmetric key encryption algorithms [63]. Symmetric encryption uses one single secret key for encrypting and decrypting data between the sender and the receiver. Symmetric encryption uses public keys for encryption and different keys (secret) for decryption. Asymmetric encryption not very efficient for small devices due to more computations needed. Therefore, symmetric encryption algorithms are almost a thousand times faster than asymmetric algorithms, because of less processing power required. Some of the most used symmetric algorithms: Data Encryption Standard (DES), the first standard, 3DES which uses keys that are three time larger, Advanced Encryption Standard (AES), which is the DES replacement recommanded by NIST using different key lengths, Blowfish, that supports different key lengths, is licence free and the fastest of them. Some of the most used asymmetric algorithms: Rivest-Shamir-Adleman (RSA), supports variable length of key and block, Diffie-Hellmann, the first public key algorithm that exchanges keys under insecure channel, Digital Signature Algorithm (DSA), developed by NIST and for authentication and signature integrity verification, Elliptic Curve Cryptography (ECC), applies the elliptic curve theory that can be used to enhance other algorithms, designed to improve performances, power and battery consumption. The most useful attribute of compare them are: the block sizes, the larger block sizes for symmetric algorithms give faster

speed time, the key sizes, larger key sizes need more battery consumption and time processing, and the algorithm speed, which Blowfish often being the fastest depending on the used parameters.

Hardware

This topic is mainly focused on security, because privacy leaks occur through security issues in this particular context.

Web services are not really concerned by advanced hardware aspects: they often only use pre-designed servers and client devices without any particular needs, not like IoT or embedded projects.

Something to verify as a company is the trust placed into vendors and manufacturer, that they propose legit and audited products. Politics can also interfere in manufacturer processes to enable industry intelligence and surveillance, such as *China's infiltration into U.S. companies*[15].

Potlapally listed in 2011 the main challenges when one wants to correctly implement security in commercial hardware platforms [64]. Hardware security is getting increasingly more complex because of two trends. First, the skills and resources to counter well-funded criminals aiming for economic goals have been raising. Secondly, an increase of hardware-based attacks has been noticed: this kind of attacks leads to the most privileged entities, with lots of flexibility and power that can also escape OS detections.

In 2014, Rostami et al. reviewed threat models, metrics, and remedies on hardware attacks [65]. Algorithmically secure cryptographic processes rely on a hardware root of trust to deliver the expected protections when implemented in software. Critical control and communication functions assume that the hardware is resilient to attacks. Backends have been found in various systems, even military ones. Cost, power consumption, performance, and reliability are considered first while designing hardware, which leads to security issues being relocated as an afterthought. Several metrics can be used to evaluate them, some of them can be used for multiple threats. The location of the attackers can be anywhere, such as 3PIP vendors, System on a Chip (SoC) integrators, foundries, PCB assembly units, test facilities, end users, or the recycling/repackaging facilities.

Jin listed in 2015 the key concepts of hardware security [66]. One had the original assumption that the supply chain was well-protected, but it is actually spread around the globe and involves lots of third parties which make it difficult to fully verify and control processes. The research evolution is going towards trustworthy hardware development for the construction of the root of trust, security-enhanced hardware infrastructure for device protection, and various security-enhanced architectures under development. New protection schemes operating at the system-level such as *ARM TrustZone*, *Intel SGX*, *CHERI* or *LowRISK*, which bring new possibilities to secure processes.

[15] *China's infiltration into U.S. companies* source: https://bloom.bg/3FqhBwf (accessed 28th October 2022)

A comparison of hardware attacks and mitigations that have been mentioned by the related papers has been made in Table 2.6 (Hardware attacks and mitigations comparison). The two sources do not contradict themselves: we will therefore use their combined knowledge.

Identification

No relevant knowledge has been found for this topic. Indeed, if a service is privacy-friendly and does not generate profit by tracking its users, there is no legitimate need to identify them apart for their authentication.

Instant Messaging and Communication

Email and instant messaging applications have become a common way to communicate, for both individuals than companies employees. Those channels carry a broad variety of data, some of which that might be sensitive or confidential. Choosing an adapted service, provider and channel is therefore very important.

The National Cyber Security Centre (NCSC) from the United Kingdom government have published a guidance on this topic [67] for companies.

Solomon studied in 2007 the means of workplace issues [68]. Workplace issues such as disputes, harassment, employee performance, and others can be supported by e-messages. Organizations do not know which messages are of interest for this kind of problems until issues surface and restored messages are requested. This raises the question of how long backups must be kept. Backups can be of two types, either online or offline of the system. The biggest challenge is that expectations of privacy for company messages sent by employees vary between territories: the United States forces companies to store them, whilst the European Union (EU) states that messages are private, unless a disclosure is requested with appropriate reasons.

Ayodele and Adeegbe listed in 2013 what are the Cloud based emails boundaries and vulnerabilities [69]. The major issue is that users do not know where emails and sensitive data are stored. The data boundaries vary depending on laws, access privileges, data protection and privacy requirements. An interesting mitigation would be to use an intelligent cloud based machine encryption and decryption system.

Foster et al. studied in 2015 the native and embedded security features in the email technology and emails providers [70]. Emails have no Integrity, Authenticity, and Confidentiality (CIA) guarantees: users must use their own tools, such as Pretty Good Privacy (PGP), but few of them actually do. Transport-layer security mechanisms can protect users' privacy, but they are limited to transport. Sender protections also exist, such as DomainKeys Identified Mail (DKIM) and Sender Policy Framework (SPF). The authors made a survey on major providers with the following results: half of them supports Transport Layer Security (TLS) (increasing part), servers do not check certificates, the SPF enforcement is limited, few senders use DKIM, and even fewer reject invalid DKIM signatures. The global email system has some protection against passive eavesdropping, but has limited protection against unprivileged peer message forgery, and no protection at all against active network-based attacks. Proper enforcement is possible for the latter.

In 2015, Rana et al. reviewed the problems and issues of instant messaging in businesses [71]. The main problems are security-related risks, legal-related risks, information leakages, and productivity decreases. The authors have found essential features to be enforced by instant messaging applications: security, stability, efficiency, versatility (effective and rich set of features), compatibility, scalability, simplicity, affordability. Both the set of features and the architecture of the instant messaging applications must comply to the organization needs.

Chawathe defined in 2018 some fuzzy rules to classify malicious emails [72]: a semi-automated, rule-based system for detecting malicious email messages which aims to fill gaps left by other security mechanisms. This classifier is amenable to human understanding and modification. The author proved a competitive performance compared to other alternatives.

Englehardt et al. reviewed in 2018 the major privacy implications of email tracking [73]. The mere action of viewing emails contains privacy pitfalls for the unwary. Hundreds of third parties track email recipients via methods such as embedded pixels, with 30% of emails that also leak the recipient IP. Additional leaks occur if recipients click on links in emails. Some third parties can link email tracking to users' web cookies. 62% of senders intentionally leak email addresses to third parties. 85% of emails in the authors' corpus contain embedded third-party content, and 70% contain trackers with an average of 5.2 trackers by email and a median of 2 third parties per email. 900 third parties were contacted at least once during the study. Some defences can be implemented: content proxying, HyperText Markup Language (HTML) filtering, cookie blocking, referrer blocking, and request blocking. Reopening emails can also bring in new third parties. No email servers or clients offers complete protection. Emails provide much of same tracking opportunities as the web. Two improvements could be made: enable server-side email content filtering, and fill gaps in tracking-protection lists.

Huang et al. defined in 2018 a classification system to categorize emails into different security levels to avoid leaks [74]. The goal is to avoid that sensitive information from emails sent to external parties is exposed to the public or to competitive companies. The author proposed a tool which parses emails content and prevents sensitive information from leaking based on emails label. If classified security level does not reach the one of user email label, the message is not sent and is reported. Two issue have been found: no metadata can be used because of privacy policies, and the data can be imbalanced because of different email lengths. The authors tested their tool and obtained a high accuracy, which they claim proves that it can be used in the real-field.

Reisinger studied in 2022 the security and privacy risks in unified communication tool [75]. The author used two threat modelling methodologies: Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege (STRIDE) and Linkability, Identifiability, Non-repudiation, Detectability, Disclosure of information, Unawareness, Non-compliance (LINDDUN). The mitigation controls must be put in relation with the threats. Ten major platforms have been evaluated on their security and privacy features and most of them provide the obvious security features, but most do not provide privacy properties. The author defined guidelines to improve security and privacy: enforcing encryption by default and making sure it is end-to-end, locking and password-protecting meetings, holding unauthenticated users in a waiting room, monitoring the participant

list, acquiring consent from participants for meeting recordings, being aware that audio-only participants calling via a regular phone dial-in option or protocol gateways could disable end-to-end encryption, being aware that file and screen-sharing capabilities could accidentally disclose sensitive information or be used to spread malicious programs. End-to-end encryption and open-source architectures are two fundamental security and privacy mitigations. Finally, the author found that mitigations against privacy threats are far less available than the security ones.

Intelligence

Usually, companies use intelligence to monitor their competitors, in order to obtain or preserve technological or commercial knowledge. In our context, we talk about intelligence as the fact of staying informed about the latest news in the security field.

In sensitive environments, being quickly informed is critical. To this end, companies must find adapted ways to be alerted about issues, vulnerabilities or public attacks in an acceptable timeframe.

Here are some relevant sources for security issues:

- *Cybersecurity & Infrastructure Security Agency (CISA)*[16], from the United States government
- *NCSC*[17], from the United Kingdom government
- *Computer Emergency Response Team (CERT)*[18], from the European Union Commission
- *The CVE project*[19], from the *MITRE*[20] company

The *OpenCVE project*[21] can be used to optimize the security intelligence on the tools, software, or anything used in a company. This tool is open-source.

Hodgson et al. listed in 2008 some technology watch review, with their corresponding techniques [76]. The author found that the two major methods are bibliometric analyses and data mining. The relevant sources must be found both internally and externally to the organization. An evaluation of the company risks should be made, resulting with a ranked list.

Rovira analysed in 2008 how to realize technology watches and how to apply corresponding techniques [77]. A technology watch consists of obtaining technical information to make decisions in a company production department. It can also be applied to commercial decision-making processes. A strategic planning must be defined with the following steps:

[16] *Cybersecurity & Infrastructure Security Agency (CISA)* source: https://www.cisa.gov/uscert/ncas/alerts (accessed 26th October 2022)
[17] *NCSC* source: https://bit.ly/3DA2aAi (accessed 26th October 2022)
[18] *Computer Emergency Response Team (CERT)* source: https://bit.ly/3DzNlxH (accessed 26th October 2022)
[19] *The CVE project* source: https://cve.mitre.org/cve/ (accessed 26th October 2022)
[20] *MITRE* source: https://www.mitre.org (accessed 26th 2022)
[21] *OpenCVE project* source: https://github.com/opencve/opencve (accessed 6th November 2022)

1. Analyse the internal and external activities of a company
2. Perform a Strengths, Weaknesses, Opportunities and Threats (SWOT) analysis
3. Create a strategy plan (short and midterms)
4. Define the critical watch factors

Five watch phases are executed continuously and cyclically:

1. Identify and analyse the company information needs defining the critical watch factors
2. Search and obtain the necessary information to track the Critical Watch Factors (CWF)
3. Evaluate and analyse the information obtained
4. Internally disseminate the results
5. Use the information in the decision-making process

Some of the most used tools for sources: service alerts, webpage software monitoring, adding agents, search agents, search engines, Really Simple Syndication (RSS) feeds, data mining procedures, bibliographic databases, patent databases, distribution lists, and invisible web databases.

Legislation

Companies must comply to several laws and directives when they commercialize products or services. Regarding web services, personal data rules must be respected.

Good advice is to always analyse and verify the country and territories legislations that must be enforced, whether it is for the local state were the company is registered or for the foreign markets the company covers. The company must then comply to those legislations.

Different levels of regulations, directives or laws can concern a specific usage, such as local states, federated state or internation laws. Having an adapted service or contact of legal advisors or lawyers is a good practice.

Poullet questioned in 2009 what were the main privacy challenges raised by the at the time present and future information society [78]. The author found numerous important changes, such as Moore's law (growth of the computers capacity, increases analysis and the quantity of data), the Internet revolution (convergence of networks around a single platform), and the arrival of ambient intelligence (puts technology into the everyday life). The main tendencies are the privatization of the cyberspace, with private corporations being the major deciders, and increased service provider responsibility in everyday behaviour and interactions. Author gave three advices: first, we must look at the social impact and the transformation of human relations created by new developments, then keep in mind that although technology has risk, it can also offer solutions, and finally that developers must think about proportionality and transparency in their work.

In 2016, Albrecht studied the effects of the GDPR on personal data [79]. The GDPR regulates almost all the personal data questions directly: it only leaves exceptional and limited specification powers to EU Member States which then have to always justify any divergence from the aim of a fully harmonized legal frame. It brought two major changes: the major players of numerous markets have changed their strategies to become leaders on data protection friendly services, and the regulation added legal certainty and coherence.

Goddard studied in 2017 the impact of the GDPR [80]. The regulation covers all personal data, which encompasses data that can directly or indirectly identify an individual including identifiers. It concerns all EU residents regardless of the location of the data processing. It encourages both ethical approaches to data collection and public trust. This regulation brings multiple principles: fairness and lawfulness, purpose limitation, data minimization, accuracy, storage limitation, integrity and confidentiality. Its core attributes are data protection by design and by default, user consent must be freely given, any data processing is only fair if it is transparent, has a wide jurisdictional scope, and is user-centric. Two challenges remain for involved entities: there is a limited scope for countries to impose their own rules, and some uncertainties about legislative derogations still stand.

Jäntti made a study in 2020 about small and medium ICT companies on their actions and feelings on data privacy [81]. The author found that the top privacy-related challenges are the lack of commitment from the top management, weak management on stored personal data in the cloud, underestimations of the GDPR effects on the organizations, and a lack of GDPR understanding and bad interpretation of authoritative legal texts. The companies have prepared themselves for GDPR regulations by identifying data registers, outsourcing the maintenance of data registers, a better monitoring of applications, participating in GDPR training events, creating data balance sheets, reviewing contracts with suppliers, and analysing GDPR from the business perspective. The major challenges for ensuring data privacy are delivering enough data privacy related information and communicating with business users, having sufficient resources for GDPR preparation, and proceeding to the verification of many systems for GDPR compliance. Implementations of the GDPR have been made with companies' own resources: they relied on guidelines for national authorities, changed some traditional ways to work, and defined data privacy roadmaps.

Mobile

Mobile devices are now almost mandatory in our modern lifestyle. They bring us new opportunities and full access to the whole Internet, but also carry our personal and sensitive information. Moreover, the current context is oriented towards the collection of user data in order to improve applications or to resell personal data to data brokers.

Nowadays, we have two major providers of smartphone ecosystems, *Google* and *Apple*, that share *99%*[22] of the market.

Security is a major issue for mobile phones, because of their sensitive and personal data holding. A lot of attackers are specialized in this particular field. Furthermore, the preservation of users' privacy must be done on applications level as much as on the OS one.

[22] *99%* source: `https://bit.ly/3W0bHIf` (accessed 25th October 2022)

Garg and Baliyan made in 2021 a comparison of the security features of *Android* and *iOS* [82]. Generally speaking, *iOS* has stronger security mechanisms than *Android*. *Android* is open-source, use the Linux kernel (applies user restriction on system resources), and use the *Application Sandbox* mechanism which gives a unique user identifier for each application. *iOS* is more restrictive, is a closed system, implements device-level locking mechanisms, has a remote wipe feature, has a secure boot chain process, implements secure enclaves, a mandatory file encryption methodology, and is not easy to *jailbreak*. The permissions are more controllable on *Android*, but application origins are more controlled on the *iOS* store. *Android* lacks control over device manufacturers, whilst Apple controls both the hardware and the software of its products. Based on recent reports from Common Vulnerabilities and Exposures (CVE), *Android* has more vulnerabilities than *iOS*, and *Android* ones are more severe. *iOS* has more remote vulnerabilities, but are more complex to exploit. The increasing market shares and the open-source nature of *Android* influence its increasing vulnerabilities amount. Most used malware attacks: trojan, ransomware, backend, spyware, adware.

Mos and Chowdhury listed in 2020 how users can protect their *Android* phone [83]. The most common threats and weaknesses are premium call rates and Short Message Service (SMS), search engine optimization, botnets, and ransomware. There are two main types of countermeasures to improve safety. The first are static approaches which disassemble and analyse the source code, either with signature matches using a dictionary or with permission checks. The second one are dynamic approaches which examine the application behaviour during its execution. It uses anomaly detection, data and control flow monitoring, emulation techniques, permissions management, device locking (avoid device tampering), antiviruses installation, and verification that applications are only installed from trusted packages repositories.

Wijesekera et al. studied in 2017 how to align mobile privacy with user preferences [84]. The authors found that if users are asked to make privacy decisions too frequently or under circumstances that are seen as low-risk, they may become habituated to future, more serious, privacy decisions. But if they are asked to make too few privacy decisions, they may perceive that the system is acting against their wishes. There is permission types that are seen as more dangerous, which are the ones related to personal data. Others are seen as more regular ones.

Wu et al. studied the effects of the design on users' security perceptions [85]. Great interface usability and adapted design of notifications positively impact users' perceived application security. Furthermore, disruptive notifications irritate users and negatively influence those perceptions.

In 2012, Boyles et al. [86] realized a survey about smartphone usage. The authors have found that 57% of mobile users have uninstalled or decided not to install an application due to concerns about how their personal information is processed.

Poniszewska-Marańda et al. defined in 2021 a secure development model to overcome common mobile platforms threats [87]. The authors have listed security standards for each data cycle. In the data storage state, locally stored data must be limited, and alternatives for key stores must be used. For data access, developers' attention must be focused on features using geolocation, application-device identifiers, and user sessions.

During data transfer, adapted encryption must be enforced, digital signatures must be used, as well as security keys. Data transfer is the weakest link of the chain. A security framework is proposed in the paper.

An investigation of data shared by the mobile phones OSes with their developers has been made in 2021 by Leith [88]. To this end, the author used an iPhone and a *Google Pixel*, and they did not enable the optional services (mapping, cloud or photo applications), leaving the settings at their default values. The kind of shared data depends on the OS. Both systems transmit telemetry, even with opt-out configurations. *Google* collects around 20 times more mobile data than *Apple*. Both systems make devices periodically connect to their backend servers with an average of 4.5 minutes, even when the device is not used. Inserting a Subscriber Identity/Identification Module (SIM) card into the device generates connections that share the SIM details with *Apple*/*Google*. Browsing activities also generate multiple network connections to backend servers. Some pre-installed applications make network connections despite having never been opened or used. Two major concerns have been listed. First, device data can be linked to other data sources with other personal details, and potentially with other devices. Secondly, every connection with a backend server disclose the device IP address, which is a rough proxy that can be used for location. Two mitigations have been found: using an alternate OS for *Android* devices, and disable Internet access by default for all application plus manually disable problematic applications. Alternatives must then be installed via alternative stores, but they could therefore not use Google Play Services.

Network

Connecting services to a network allows remote access for a public or private usage. In all cases, it brings new threats in a system with the possibility for external parties to collect knowledge or to exploit vulnerabilities.

A network must be well configured to avoid those new threats. The difficulty is to restrict the possible actions or accesses as much as possible without disrupting or impacting the services.

Marin listed in 2005 the most important network security basic design and configuration issues [89]. The network traffic must be analysed, both on the flows and formats. Be aware that attackers can know the protocols intent and their rules to interpret the associated formats and flows. Network intrusions can be used for several goals, including to consume the resources uselessly, to interfere with the system or to gain knowledge. DDoS attacks intent to slow or to interrupt services. There is no single technique to detect network intrusion: signatures or anomaly detection are the most common.

Pawar and Anuradha listed the most known types of network attacks [90]. There is three type of them: active, passive and advanced. Active attacks are initiated by commands. They include spoofing (play on identity), routes modification, wormhole (tunnelling traffic), fabrication (false routing message), denial of services, sinkhole (prevent node to exchange information), and Sybil (insert multiple malicious nodes). Passive attacks do not require any action. They could be traffic analysis, eavesdropping (find credentials in communication), or monitoring access. Advanced attacks are more difficult to realize. Some of them are black hole (replace the best paths), rushing (make receiver busy), replay (repeat or delay data), Byzantine (disrupt or degrade routing), or location disclosure.

In 2016, Ghafir et al. listed the most appropriate approaches to securely monitor networks [91]. Three tool approaches can be used to find, report and resolve problems. The first is packet capture: it intercepts data packets that are crossing a node or moving over it. The second is Deep Packet Inspection (DPI): actions on packets are applied when they match specific data or code payloads. Finally, there is flow-based observation, that analyse packets in a specific transport connection or a media stream. The criteria for DPI tools are prototype support, developer friendliness, and extensibility. According to the authors using those criteria, the best DPI tool is *Bro*https://zeek.org/20230107.

Kavianpour and Anderson reviewed in 2017 what are the threats, vulnerabilities and mitigations of wireless networking [92]. Wireless communications imply additional threats: introduction of malicious activities, interception of data transmission, or passive eavesdrop. List of some attacks: malicious association (mock a legitimate access point), ad hoc networks attack (no central access point: access control issues), man in the middle, rogue access point (unsanctioned by administrator), lack of encryption. Some mitigations: chose good encryption parameters, educate users, limit access with explicit allowance, change factory router configuration, change default router identifier, disable broadcasting of identifiers, apply Mandatory Access Control (MAC) filtering, keep firmwares up to date.

Dimitrakos et al. explained in 2005 how to design great network access control policies [93]. The authors identified that the main problem with firewalls is the difficulty to configure them appropriately. To do so, administrators need a clear methodology and adapted supporting tools. They should use high level languages to specify a network security policy in order to avoid mistakes and to help further edits in the future. It is recommended to apply dual security policy, which specify both permission and prohibition rules. However, it requires rules ordering, which is difficult to assess. An alternative could be to apply closed access control policy, with permissions only. A complete concept is given in the paper.

Operating System

OSes are mandatory to handle all the operations in a machine. It brings lots of interfaces for the users, resources and hardware management, and runs programs. The current market includes several competitors: *Windows* and *macOS* are leading the personal computer market, Linux and *Windows* the server market, *Android* and *iOS* the mobile market.

OSes have different security mechanisms and implementations, but their general health and robustness are being periodically improved for several years.

On the privacy side, we can split them in two categories: the free systems (as in freedom) and the proprietary ones. The firsts are generally more privacy-friendly than the others. Moreover, such systems require user accounts to be either configured or to unlock access to all the features of the OS.

User privacy varies a lot among Linux distributions: some of them are focused on strong, complete privacy, some of them are oriented towards other goals.

Although *Android* is initially a free and open-source project, its most used and commercially distributed version include modifications and applications from *Google*, including a large part of tracking technologies.

Bassil made in 2012 a comparison of security features on the two most widespread and successful desktop and server OSes, (*Windows* and Linux) [94]. They both uniquely identify each entity. On access tokens, *Windows* stores restrictions where Linux uses Discretionary Access Control (DAC) and MAC. Furthermore, it does not store tokens type. Impersonation design is more secure in *Windows* than Linux. Regarding Access Control List (ACL), *Windows* uses privileges and restrictions, Linux uses MAC and DAC and do not handle logging. For privileges and user rights, *Windows* uses a separate process where Linux uses MAC and handles restrictions with separate daemon. They both have similar auditing and logging features. *Windows* implements a more secure but complicated authentication system than Linux. Linux has no native file system encryption. *Windows* has more security components within its kernel and is more complicated, where Linux use user-mode processes and is more efficient.

Adekotujo et al. made a comparative study of strengths and weaknesses of major OSes [95], back in 2020. Author found that more malware targets *Android* devices than *iOS* ones. *Windows* 10, Linux, UNIX and *macOS* are the most secured and reliable. *Windows* 10 and *macOS* have integrated firewalls. The following list summarizes facts about OSes:

- ***Windows***: has great support and compatibility with lots of functions, but costly, slow and exposed to viruses.
- **UNIX**: comes with a great user control, is very reliable, but needs expertize with a large learning curve.
- **Linux**: free, less vulnerable, great variety, but is complicated, has low applications compatibility and few vendors.
- ***macOS***: few viruses, high reliability, but is expensive, needs *Apple* computers and has a low application compatibility.
- ***Android***: open-source platform, with easy access to applications, continuous upgrades, adapted for programmers, but unstable, has lots of bugs in applications, has limited administrator access (rooting), and lots of applications need Internet access.
- ***iOS***: stable and safe, minimal viruses exposure thanks to a strong applications policy, but has a low operability because of mandatory *Apple* hardware which is also costly.

Yaswinski et al. listed in 2019 methods to secure a Linux system from internal and external threats [96]. First, administrators must apply security through repositories: they must avoid software from other sources than the repositories provided by the distribution. Then, usage of antivirus is recommended, such as *ClamAV*[23]. Precautions must be

[23] *ClamAV* source: `https://www.clamav.net` (accessed 24th October 2022)

taken if compatibility layers as used, such as *Wine*[24]. Administrators must always keep software up-to-date for security patches. They must also set up firewalls to avoid access gains. Different accounts with unique passwords must be provided for each person, including separate usages such as root access and regular users. Finally, adapted file access permissions must be enforced.

in 2007, Zhai and Li [97] studied some security mechanisms inside Linux. The biggest addition is Security-Enhanced Linux (SELinux), which has been developed to implement MAC policies. It supports multiple security models, is extensive but have low flexibility and difficult to manage.

Policies

Information security policies allow organizations to avoid data breaches, which are often caused by employees. Those are considered as the weakest point in organizations. Establishing and enforcing a complete and adapted policy is one of the most effective mitigations.

In general, the whole family of *ISO 27000 standards*[25] is the best yet one of the heaviest ways to comply to a great policy.

Knapp et al. [98] (2009) defined in 2009 an information security policy process model for professionals. Information security policies are the first step to protect organizations against attacks, and are used to implement effective deterrents for data CIA. A policy is a general rule implemented in an organization to limit the discretion of subordinates. The study showed ten internal and external influences, alongside of their relationships and influences. The model is repeatable.

Bulgurcu et al. studied in 2010 the role of employees on information security policies [99]. Author found that employees' compliance with policies is significantly influenced by attitude, normative beliefs, and self-efficacy to comply to them. Policies positively affects both attitude and outcome beliefs, and organizations security compliance increases if employees follow policies.

In 2015, Safa et al. defined Information security model to consider employees behaviour [100]. The authors found that users' poor information security behaviour is the main cause of security breaches. Such model leads to positive effects on information security awareness, information security organization policy, information security experience and involvement, attitude towards information security, subjective norms, threat appraisal, and information security self-efficacy.

Alotaibi et al. listed in 2016 the challenges for a successful implementation of information security policies [101]. Employees are seen as the biggest potential threats to organization cybersecurity: non-compliance with the policy is one of the main issue. Main challenges:

- **Security policy promotion**: dissemination, awareness raising, training, enforcement and monitoring

[24] *Wine* source: `https://www.winehq.org/` (accessed 7th January 2023)
[25] *ISO 27000 standards* source: `http://bit.ly/3fEdLW0` (accessed 6th November 2022)

- **Non-compliance with security policy**: malicious and negligent behaviour, unawareness
- **Security policy management and updating**: regular review and update, policy management, technology advances, designing good policy
- **Shadow security**: unclear security policies, unusable security mechanisms, high compliance costs

Two factors can influence the behaviour. It can be organizational with poor information quality, motivation, sanction, awareness and training, computer monitoring, or persuasion. It can also be human with personal traits that can impact the compliance such as perception, personality, technology democracy, cultural factors, gender, satisfaction, habits.

Soomro et al. reviewed in 2016 what are the roles and activities needed from management to handle information security [102]. The authors defined five aspects: information security and management, information security policy awareness and training, integration of technical and managerial activities for information security management, human aspects of information security management, information security as a business issue.

In 2020, Hina and Dominic analysed what makes sensitive infrastructure under risk [103], using a high education institutions context. The authors found several lacks in policy guidelines, in awareness of information security threats, and in irregular monitoring of misuse behaviour. Those items lead to threatening situations. A security framework to implement strategic security procedures for users should be defined to ensure compliance with security policies and protection of vital resources. An information security culture developed in organizations can reduce the risk of security breaches and potential incidents, given that compliance with rules and regulations becomes a habit.

Pseudonymization

Pseudonymization protects the privacy of users by paying on their identity in a way that their become unrecognizable even if the corresponding service has been compromised. All identification data is replaced by a specifier which can not be linked to the original user without the corresponding secret. Pseudonymization provides a form of traceable anonymity and requires legal, organizational or technical procedures [104]. In the scope of health data, data storage that guarantees privacy and strict control of access by the patient is a special and strong need. Pseudonymization is one way to fulfil this need.

Riegl et al. defined in 2007 a new architecture for the pseudonymization of medical data, using a two steps process [104]: service providers should identify all information uniquely associated with a certain person, and separate this information from the remaining data. Pseudonyms can be calculated by either encryption, using symmetric or asymmetric keys which enable reversal of the operation, or hashing, but it needs a list which is a weak point. Those approaches can be differentiated by how pseudonyms are created and shared, by used security techniques, or the owner of the secret. Author proposed an architecture with a central system which allows key recovering, with users in full control of their data.

Privacy preservation techniques in e-healthcare have been explored by Sahi et al. [105] in 2018. The authors used hybrid protocols to reach healthcare requirements. Access control and stored data security must be mixed with pseudonymization. Hash functions are limited for data sharing purposes. Usage of blank pseudo-identities is useful to avoid pseudonym correlation. Two-level pseudonymization should be designed to allow multiple pseudonyms for one identity, using an authority for correlation. Add control to patient: group identities for users to allow fine sharing with parties. Pseudo-identities which are derived from primary ones must be used independently. Data anonymization, de-identification and pseudonymization are necessary to share health data outside a patient's privacy and trust sphere.

In the same spirit, Fernández-Alemán et al. studied pseudonymization techniques used in health systems [106] in 2013. They stated that reversible pseudonym generation can be achieved using AES. Pseudonym alteration can be avoided by using integrity protection. Same pseudonym generation regardless of data source origin can be computed using a dual-pass pseudonymization scheme. Pseudonym trees can be used to differ identity sent to each provider.

In 2019, Ribeiro and Nakamura found a technique [107] to avoid problems in case of database leaks: store hashed pseudonyms, and use salts for password generation to avoid collisions.

Rai made an analysis of different pseudonymization techniques [108] in 2016. There are multiple ways of generating pseudonyms: they can be created remotely by a centralized third party or locally by the holder of identity. Some approaches: Peterson (keys stored in the database), pseudonymization of information in e-health (hull architecture), electronic health card (service-oriented architecture), Thielscher (identification data and anamnesis data stored in two different databases, using decentralized keys), Pommerening (two approaches, for one-time usage or re-linkable patients), Slamanig and Stingl (centralized database with smart cards for authentication).

Sandboxing

Sandboxing is addressed to malware threats by containing their malicious behaviour in controlled and isolated environments. All processes or instances contained in a sandbox can not communicate with other processes or instances.

Greamo and Ghosh studied in 2011 how to use sandboxing and virtualization to fight against malware in applications [109]. Multiple techniques exist to encapsulate processes: restrict account privileges, separate the file systems of applications, run applications in their own VM, separate untrusted code from the system. Several approaches exist with various level of protection: in-browser security, sandboxing (partial virtualization), full virtualization, secure virtualization. The latter is the most secured, it must have the following attributes: host and network isolation, real-time detection (previously unseen attacks), fast and complete recovery to a known clean state, forensic data collection on infection, hypervisor integrity checks.

Sandboxes are used in healthcare: Leckenby et al. have assessed their potential [110] in 2021. Sandbox is a safe space to test innovative products, services, business models and delivery mechanisms without immediately incurring all the normal regulatory consequences:

this approach is called test beds or testing labs. It confirms software compliance with existing regulation before implementing them in production. Real-world test beds are therefore controlled environment with real-world conditions. There are different categories of approaches: advisory, adaptive and anticipatory.

Damshenas et al. analysed in 2022 how to prevent malware propagation using virtualization [111]. To do so, VM environments must detect malware and prevent its propagation: it uses undocumented *VMWare* options to avoid its recognition, and alter the magic value of the *VMX* Central Processing Unit (CPU) flag, related to Intel virtualization (VTX) the communication channel.

In 2011, Delport et al. listed new techniques to isolate instances on a cloud environment [112]. It includes instance relocation, suspicious instance isolation without interruption, failover (backup the instance), address relocation for network traffic when problems occur, sandboxing (no interaction), let's hope for the best (terminate node, move to a controlled environment and make images).

Server Architecture

Nowadays, the two major and almost two only server architectures are the monolith and the microservice ones. The first architecture is the legacy approach, which consists of building all features into a single program. The second one is a more recent approach that have the following characteristics [113]:

- Isolation from other services, as well as from the execution environment based on a virtualized container.
- Autonomy: services can be deployed, destroyed, moved or duplicated independently.
- Open and standardized interface that describes all specific goals with effectiveness, efficiency and available communication methods.
- Fine-grained: each service should handle its own task.

Migrating monolithic architectures to the cloud is a complicated task, especially to gain access to the advantages of this kind of platform. Moreover, scaling monolithic applications is suboptimal because they include several services that might not need the same scaling factor. Microservices have therefore been adopted as the optimal and natural solution in their replacement.

The monolith architecture is the regular ground when talking about generic security or privacy features, which is why no specific paper have been found about this approach: the other topics already cover its scope.

Almeida et al. defined in 2017 which elements must be considered for the construction of solutions based on microservices [113]. Several and interconnected points of access blur the boundaries but do not obscure security vulnerabilities. Routine completions require services to communicate over network, which exposes more data and information (expands attack surfaces). Developer teams must define how services are interconnected and interacting to mitigate this concern. Network complexity makes it difficult to establish a complete view (debugging, monitoring, auditing and forensic analysis). Relationships

trust can use the OAuth standard between services. Service fragmentation brings a better availability (one failure do not impact the whole app) and better code portability (automation, independence, version management). Microservices are mainly deployed in clouds environments, which means additional security and privacy concerns. Some layers can be implemented to secure and privatize cloud models: physical and environmental security, cloud infrastructure security, network security, data and access control and privilege management.

A security framework for microservice [114] has been defined by Yarygina and Bagge in 2018. The main concern is to decompose microservices security into their components. Categories of security: hardware, virtualization, cloud, communication, service, and orchestration. The price of mitigations for each category is not the same. Particular perimeter of security: assume that other services may be compromised and hostile ("trust no one"). Particular security properties: do one thing and do it well, realize automated and immutable deployment, isolate through loose coupling, diversity through system heterogeneity (use N-version programming), fail fast (tolerate partial failures). Security practices: mutual authentication of services using mutual TLS (mTLS), principal propagation via security tokens, fine-grained authorization. Proposed security framework to establishing trust and securing microservice communication with mTLS, self-hosted Public Key Infrastructure (PKI) and security tokens.

Finally, a complete book of specialized knowledge for microservice-based application systems has been created by Chandramouli [115] in 2019. It has been commanded by the NIST institution and contains specialized security strategies.

Social Engineering

This kind of attacks exploits the human tendency to trust people, which is the weakest link in the whole security chain. It is realized by psychologically influence and manipulate key people to divulge confidential information or to break the security procedures, which threaten all systems and networks. Attacks can be easily automated, which enable large case scenarios. Social engineering is therefore a threat that can not be mitigated using technical ways.

In 2019, Salahdine and Kaabouch made a survey on social engineering attacks, classifications, detection strategies, and prevention procedures [116]. Social engineering phases: collect target information, develop relationship with it, exploit the information and attack, exit with no trace. Classification: human or computer based. Categories: social, technical or physical, and also directly or indirectly. Some prevention should be made in companies' risk management strategy, and also rise awareness within the employees. Defence approaches: encourage security education and training, increase social awareness, provide required detection tools, keep confidential information safe, report suspect activities, train new employees, advertise employees using sensitization and fraudulent emails.

Krombholz et al. made an overview of advanced social engineering attacks on the knowledge worker [117]. Author found multiple categories: social, technical, physical or reversed (sabotage, advertising and assisting). Most attacks often combine several or all categories. Computer-supported collaboration is a main entry point: office and external communication tools are increasingly used. Most used channels: e-mails, instant

messaging, telephone, social networks, cloud services, websites. Attacks' operators can be human or software. Attacks: online social networks (wealth of personal information), social phishing and context-aware spam, fake profiles, cloud services (shared resources), mobile applications (vulnerable applications).

Researches regarding mitigation of social engineering have been carried out by Chizari et al. [118] in 2015. The main goal is to gain victim's trust by various manners: reciprocation, commitment, social proof, friendliness, authority and scarcity. Attackers use public sources such as web search to perform profiling of the targets. A primary tactic used by attackers is impersonation.

A comparison of social engineering attacks mentioned by the related papers has been made in Table 2.7 (Social engineering attacks comparison). The two sources do not contradict themselves: we will therefore use their combined knowledge.

Software

To be functional, a piece of software go through multiple phases realized by multiple people. Considering security and privacy issues during its whole scope is therefore a mitigation to avoid late and expensive corrections.

Hochheiser defined in 2000 which strengths then-existing tools must clarify for their next generation [119]. The author found necessary principles for privacy protection systems (perhaps insufficient): simplicity, privacy by default, no penalities for privacy, users fully, accurately, and fairly informed, services built on trust must be accountable, and treat privacy as part of security.

Hadar et al. [120] made interviews of developers in 2018 to figure out how to apply better Privacy by Design (PbD). The PbD main challenge is to introduce privacy considerations into the technological design: translating the general abstract notion and the meaning of informational privacy into concrete guidelines. Interviews were made to understand what developers think of privacy. Developers hold a partial understanding of privacy, mostly limited to security concerns, prefer policy-based solutions to architectural solutions, are highly influenced by organizational privacy climate, are willing to trade off the level of privacy to achieve better usability. Fair Information Practice/Privacy Principles (FIPP) guidelines is a common ground to enforce privacy. Many developers do not have sufficient knowledge and understanding of privacy concepts, nor do they sufficiently know how to develop privacy preserving technologies. Other privacy challenges: testing, bug reporting, sharing information of defects.

McGraw studied in 2004 the software security domain [121]: he stated that systems must continue to function correctly under malicious attack. Developers must think about security early in the software life cycle, plus identify and understand common threats. All parties must be educated. Security should be explicit at the requirements level and must cover functional security plus emergent characteristics. Implementation flaws can be avoided using static analysis tools. Testing must be done with standard techniques, plus risk-based security testing. Use penetration testing and monitoring.

In the same spirit, Jones and Rastogi have studied [122] in 2004 how to integrate security into the Software Development Life Cycle (SDLC) process. Developers not having a security view from inception through deployment and beyond are identified as the root of most security and privacy breaches, despite efforts of their organizations. A solution is to fully integrate security in the SDLC process. Several points are listed and explained in the paper.

Finally, Potter and McGraw used and analysed [123] risk-based approaches to test systems in 2004. Author found that risk analyses help to identify potential security problems and their impact. Tasks should be defined to manage software security risks. The security testing approach must define who must do it, and how to think like attackers. Automating testing can be used for minimal human intervention and qualitative results. Furthermore, functional and nonfunctional testing should be made.

Storage

Stored data must have sufficient layers of protections to avoid any leaks or attacks. This topic also mentions cloud storage because it is used for a lot of services nowadays, but this approach can also be applied to personal servers. It only adds new security and privacy aspects, not removing ones.

Note that if anonymization or pseudonymization have been applied to the data, user privacy is improved as well for the storage perspective.

Issues and solutions regarding cloud data storage have been reviewed by Vurukonda and Rao in 2016 [124]: the main challenges are data breaches, data theft, and unavailability. Cloud providers have full of control over stored data, and bring virtualization and multi tenancy related security issues:

- **Storage issues**: data privacy, integrity, recoverability and vulnerability, improper media refinement, data backups.
- **Identity management and access control issues**: malicious insiders, outside intruders.
- **Contractual and Legal issues**: SLA and legal issues.

Some solutions found in literature are given for each issue.

Syed et al. showed multiple storage security concerns [125] back in 2020, with related implementations to prevent damages. First, the type of cloud must be chosen carefully. Threats: account control, malicious insiders, data control, management console security, multi-tenancy. Activity patterns and business reputation also need attention. Due to multiple implementations, a standardized security model is convoluted and inefficient. Risks: lack of control, shared servers, data leakage, API access and storage sinks. Some security practices: assessing cloud using frameworks, encryption, data classification, multifactor authentication, private encryption, in-transit encryption, ransomware protection.

In 2010, Hubis and Hibbart listed [126] standard for secured data storage. They described a method of encryption for data stored in sector-based devices, where the threat model includes possible access to stored data by the adversary. Specifies the encryption transformation, but not the encryption of data in transit.

Wang et al. made in 2010 recommendations on the usage of public auditability for cloud data storage security to check outsourced data integrity [127]. For third party auditing on cloud data storage, it must be done without demanding local copies, and must bring in no new vulnerabilities towards user data privacy. Utilization of homomorphic authenticator and random masking to guarantee can be good to guarantee that parties can not learn any knowledge about the data.

More specific technologies challenges must also be taken into account, which have been analysed by Thain et al. [128] in 2005: it includes distributed file systems, third party transfer, active storage, and group management. To be effective, all security mechanisms must be distributed with the nodes (decentralized). But specific problems appear: side effects on the file systems semantics and on active storage, securing third party transfers. Challenges appear with decentralized systems: unbounded set of users, multiple identities per user, new decision points, unexpected policy coupling. The main security mechanisms are great authentication and authorization. Recommendations: systems should store meaningful identities deep in the software stack, clients must be prepared for a wide array of failures, and users need tools for debugging security mechanisms.

System Administration

A system must support a web service, and must therefore be secured. Some aspects of this topic have already been processed in other topics. Furthermore, complete guides [129][130] can be useful for system administrators, with their security parts.

Yeh and Chang made a list of threats and countermeasures regarding information system security [131] back in 2007. They listed multiple categories of countermeasures: software, hardware, data, network, physical facilities and environment, personnel, regulation and legality. Each one has specific mitigations. A study has also been made on countermeasure adoption in companies, and several problems were found:

- Lack of relationship between the severity of the perceived threats and the scope of the countermeasures adopted.
- The protections of assets do not increase with greater managerial perceptions of threats severity.
- Countermeasures adoption is greatly influenced by the industry field of the company and organizational computerization level.

Web Browsers

Although not directly linked to an information system, a web browser is nowadays the entry point to a majority of web services. This kind of application is used daily, even hourly, by both end users and developers. Its security must be strong to avoid attacks while visiting websites, and the browsing habits must be protected to avoid user tracking: indeed, a lot of knowledge can be extracted from browsing history, downloads, frequency, et cetera.

For the developer side, critical information about the company must be protected. For the end users, their privacy must be respected to avoid tracking. For both of them, malicious attacks must be impossible to conduct.

An opened, exhaustive and community *comparison of modern browser*[26] can be a good indication about their current privacy and security levels.

Aggarwal et al. studied the security and privacy of private browsing in 2010 [132]: they noticed that this mode is used differently from their marketed message. This mode should not leave any traces on the user computer, and also complicate remote parties to identify users. However, the authors have noticed inconsistency between browsers and additional complexity are brought by additional plug-ins and extensions installed on browsers. Browsers fail to provide appropriate protections in various ways, which imply that corrections must be made by the browsers maintainers. Please note that this study was made a few years ago: we only kept the concerns, not the technical parts.

Snyder et al. found some risks [133] for users privacy and security with the new features that are implemented into browsers. This study made in 2017 shows that browsers compete on performance, security, and compatibility: this race imply that users and developers expect a large set of features into their browsers. However, each feature adds a benefit (capabilities) and a cost (security/privacy issues). The authors proved that removing over 60% of the standards had no noticeable effect on users' experience, with WebGL having the highest cost with the lowest benefit of all features. The authors concluded by saying that disabling all risky features is not necessary, but giving users the knowledge and possibility to chose is the best thing to do.

Englehardt and Narayanan measured in 2016 the tracking on the top one million websites [134]. Users are tracked by first parties (visited website) and third parties (hidden trackers) by being uniquely identified by a combination of tracking techniques. Third parties tracking is growing and diversifying in their techniques: *Google* can track users across nearly 80% of websites. Device fingerprinting can identify a user by its computer properties, without any trackers. Users can reduce their exposure with browsers' privacy features and extensions for regular tracking.

Virvilis et al. found limitations and related countermeasures to avoid rogue website using blocklists in 2015 [135]. Threats can come from both nefarious and benign websites which are compromised: yet, lots of web users do not know any security solution to mitigate those risks. A secured proxy that aggregate multiple blocklists could be used in order to block attacks on all devices.

Leith listed [136] in 2021 the privacy risks implied by data exchanges between a browser and its backend servers. Browsers operate locally and with their backend infrastructure, with discloses unique identifier of users. The browsers can be configured more privately: they are not privacy-friendly by default. All browsers do not track users using the same identifiers, persistency and use cases. Privacy risks can be reduced by configuring browsers properly, and by choosing the most adapted ones.

[26] *comparison of modern browser* source: `https://privacytests.org` (accessed 18th October 2022)

Bielova explained in 2017 [137] what users can do to protect themselves online. First, by tweaking the browser configuration and by explicitly blocking third-party cookies. Then, by installing specialized extensions [138], which can often being limited to rule sets. Finally, disabling JavaScript to avoid fingerprinting could be done, although being very heavy for the browsing experience.

Paper	Summary	Useful Data
Li et al. [139]	Design of sole-based trust management languages	Such languages are useful for ABAC in decentralized collaborative systems: access control based on identity can be ineffective if entities do not know each other. ABAC systems have multiple capabilities. They can handle decentralized attributes, using entity asserts that another entity has a certain attribute. They can give delegations of attribute authority, which allow to trust another entities judgements. ABAC systems can control the inference of attributes and attributes fields. Finally, they handle attributes-based delegation of attributes authority, which gives them the ability of delegating to strangers whose trustworthiness is determined based on their own certified attributes.
Yuan and Tong [140]	Study of the ABAC model for web services	This model is based on subjects, objects, environments and attributes. ABAC is both mandatory and discretionary, and it can not predict how data must be shared in Service Oriented Architectures (SOA) environments: it is ad-hoc and dynamic in nature. Web services have rich semantics, which means that simple, static, and coarse-grained access control models should be avoided. Two access control models exist. The first one is DAC, which can restrict access to objects based on the identity and need-to-know of entities. The permissions can be passed from a subject to other entities. The second one is MAC, which can restrict access to objects following fixed security attributes given to users and objects. The controls are system-enforced, and it can not be modified. Both models can be used in conjunction. Three models are based on those two models. The Identity Based Access Control (IBAC) model uses permissions linked to identities. The Role Based Access Control (RBAC) model uses permissions linked to business functions or roles, including levels of indirection. The Lattice Based Access Control (LBAC) model solves the MAC problem of non-modification by using an ordered set of security labels combined with a set of categories. However, it has a lack of flexibility and scalability. Two main aspects are defined within ABAC: the policy model, which defines policies, and the architecture model, which applies the policies. The ABAC model defines permissions on any security relevant characteristics (attributes), includes both IBAC and RBAC functionalities and is more flexible with the attribute approach. Compared to the other models, ABAC is intuitive, more flexible and powerful, the security management can be distributed, and it uses a divide and conquer approach.
Hu et al. [141]	The ABAC model explained	The authors stated that 'ABAC is a logical access control model that controls access to objects by evaluating rules against the attributes of entities (subject and object), operations, and the environment relevant to a request'. It allows a high amount of inputs in the evaluation process, which brings an almost infinite amount of possible combinations. The relationships are not modified if updates must be done on access decisions, only the attributes are altered. The NIST has published the Special Publication (SP) 800-162 to help companies to understand and implement the ABAC model. However, it can be complex to apply in large organizations.

Paper	Summary	Useful Data
Servos and Osborn [142]	The limitations in the ABAC model	No standardization of ABAC has been published, but an acceptation of high level descriptions (NIST SP 800-162) has been accepted into the community. Some problems are caused by its infancy. No references are made to foundational models. The capability of emulating ABAC models has only been demonstrated informally in research context. The support of hierarchy is lacking, which is emulated by either using complex data types in attributes or by unmaintainable complex policies. A solution would be to use attribute user groups. Compliance is complicated to prove during audits. Would be simpler with hybrid models. The separation of duties is still unclear in research. The delegation feature is limited, must be done in the implementation. The attribute storage and sharing make it hard to evaluate trustworthiness of attributes and their compatibility when multiple attribute sources exist. It would require a commonly accepted namespace or ontology. The scalability must still be proven. The administration and user comprehension must be understood. Formal security analyses can be difficult to realize, some tools are compatible, but none are specialized for the ABAC model.

Table 2.1 Attribute-based access control policies comparison

Paper	Summary	Useful Data
Machanavajjhala et al. [143]	Privacy problems of k-anonymization	The values of sensitive attributes can be recovered if they have little diversity. Privacy can not be guaranteed against attackers who have background knowledge. The main mitigation is to use an extension of k-anonymization named l-diversity which adds diversity in data groups attributes.
Li et al. [144]	How to use slicing to preserve privacy	The k-anonymity technique looses considerable amount of information, especially for high-dimensional data. Bucketization does not prevent membership disclosures and breaks attribute correlation between sensitive attributes and quasi-identifiers. The slicing technique partitions data both horizontally by grouping tuples into buckets and then randomly permuting them, and vertically by grouping attributes into columns based on correlations. Slicing has a better data preservation utility compared to generalization, can be used for membership disclosure protection, can handle high-dimensional data, and can respect the l-diversity requirements.
Kumar et al. [145]	Comparison of multiple privacy preservation techniques	An additional approach to k-anonymization and l-diversity has been found by the authors, which is the t-closeness approach. It expends the l-diversity by reducing the granularity of data representations. t-closeness can use various techniques: generalization, multi set-based generalization, one-attribute-per-column slicing, slicing, or slicing with suppression. Those techniques give different results depending on the considered parameters, with variations on revealed correlation on quantity, the information loss, the data type, the level of privacy preservation, or membership disclosures.

Table 2.2 Anonymization techniques comparison

Paper	Useful Data
Hussain et al. [14]	The major vulnerabilities of APIs are script insertions, SQL injections, bound of buffer overflows, DDoSes, login attacks, and application or data attacks. Some security models can help to mitigate those vulnerabilities, such as authentication, throttling, communication security, or anomaly detection. The access control management can be enforced following the OAuth or OpenID standards. Communication security can be enforced using HTTPS for JSON transfers for the REST approach, or by using web services security and XML built-in security for the SOAP approach. Client throttling can be implemented in order to avoid attacks. The gateways security can be enforced by performing message analysis, by granting access tokens and authorization parameters, by acting like a traffic police, and by only authorizing legitimate users. A major limitation has been found by the authors on some APIs: a traditional approach is to limit access to the API instead of mitigating the attacks. An improvement into general security would be to integrate AI into API security. ML security consists of identifying malicious intents in data transactions. The models must learn patterns of normal behaviours for each context.
Diaz-Rojas et al. [146]	The authors found 68 security threats. The most common are eavesdropping, leakage of sensitive information, code injection, denial of service attack, man in the middle attacks, API hijacking, replay attack, brute forcing credentials, and broken authentication. 66 design advices have been found to harden security based on those threats, which are mainly focused on the network channels. The major mitigation techniques are to ensure separation of entities, strong authentication, strong authorization, strong encryption, strong access control, access revocation, validation of messages, enforce logging, input validation, input sanitization, set up rate limits, set up redirections, appropriate testing, realize design reviews, ensure high availability, great role engineering, regulate the traffic, enable load balancing, set up service degradation, and ensure proper monitoring. Multiple patterns can help to design secure APIs, such as the principle of least privilege, parameter forest, one factor security, two factor security, three factor security, client-server basic security, using API gateway, defence in depth, default denial, command pattern, and data minimization. Multiple methods can be used during implementation: the most used and appropriate are token-based authentication, digital signing, RBAC, ABAC, token-based authorization, and multifactor authentication. Threat modelling can be done using various schemes, such as STRIDE, DREAD, OSSTMM, sequence diagram, use case, user story, the NIST guide to cybersecurity, or OWASP testing guide.
Sharieh and Ferworn [17]	The authors determined a list of the most critical vulnerabilities for APIs: broken authentication, sensitive data exposure, using vulnerable components, improper use of CORS, and DDoS. Three categories of attacks have been found by the authors: post-login attacks, that aim for data and the application, pre-login attacks, which use authentication services, credential stuffing, fuzzing, or stolen credentials, and fundamental API security attacks, using resources such as access control, tokens, authorization, authentication, rate limiting, client throttling, quotas, network privacy, and TLS configuration issues.

Table 2.3 Application programming interface vulnerabilities, attacks and mitigations comparison

Paper	Useful Data
Li [19]	An adversarial attack can be done by injecting poisoned data in order to manipulate data distribution, which can lead to incorrect classifications or predictions. Among others, three defence methods can be applied. First, the training process and input data can be modified by continuously adding new adversarial samples, which requires a lot of data and could deceive network. Random rescaling on inputs can be introduced, or foveation mechanism can be used. Secondly, the network can be modified in several ways, such as by applying input gradient regularization, by using nonlinear activation functions, or by using dense associative memory models. Thirdly, an additional network which is separately trained can be used.
Xue et al. [22]	The authors found multiple vulnerabilities, such as outsourced training procedures, usage of pre-trained models that include intellectual properties, or unvalidated data sources coming from third parties. One example based on those vulnerabilities are adversarial attacks that use incompletion in training data, or that use overfitting and influence mechanisms to recover the sensitive data used for training. Some major security threats have been found by the authors. Data poisoning can lead to mislead predictions. backends implemented into training data can lead to misclassifications for specific trigger conditions. Adversarial attacks can be realized, either in an error-generic way which make models go wrong, or by an error-specific way that makes misclassifications based of adversarial examples. Model extraction attacks can be done in order to steal the model by observing the output labels and confidence levels with respect to used inputs. A recovery of sensitive training data can be realized using membership inference to determine if a sample is used in training phase, or by inversion attacks that infer information on the training data. Some defences exist against poisoning attacks and backend attacks, such as data sanitization and anomaly detectors. One possible defence against adversarial examples attacks is model outputs smoothing, which reduces the model output sensitivity regarding its input. Multiple defences can be enforced against sensitive information leakage: distributed learning frameworks, traditional cryptographic primitives-based approaches such as differential privacy or homomorphic encryption, and trusted platform-based approaches. Actual defence implementations depend on the type of models and the approaches.
Hu et al. [147]	Different categories of threats exist during the data collection phase. It could be software-based, with data biases, fake data, data breaches, or it could be hardware-based using sensor spoofing. The data pre-processing phase is mainly concerned by scaling attack with images. Some mitigation include data randomization, quality monitoring, or image reconstruction. The training phase has two major threats: poisonous data injection combined with availability attacks, which deteriorate the general performances of the model, and integrity attacks, which only deteriorate specific inputs. Some mitigations exist, such as data sanitization, robustness training, or certified defences. Regarding the inference phase, the biggest threat are evasion attacks, that degrade or interfere the predictive performances using adversarial attacks that alter the input without changing the targeted model. Some mitigations can be used, such as distillation, detectors, network validation, adversarial training, data randomization, or input reconstruction. Finally, the integration phase includes threats on the confidentiality of the model or on the data, vulnerabilities brought by the code, AI biases, and generic ICT threats.

Paper	Useful Data
Liu et al. [24]	The most known model attacks are model extraction, feature estimation, membership inference, and model memorization. The major privacy attacks are (re)identification, inference, which allows to illegitimately gain knowledge, and linkage, which gathers information by correlating data sources. Some privacy protection schemes exist, such as obfuscation, anonymization, reducing information sharing, cryptography, privacy risk assessment and prediction, personal privacy management assistant, and private data release, which consists of publishing data with guaranteed privacy.
Liu et al. [25]	The most used privacy attacks are model extraction, which duplicates the model parameters or hyperparameters, and model inversion, which infers sensitive information by utilizing available information. Some well-known security threats are adversarial attacks, which are invisible perturbations that mislead predictions, and poisoning attacks which brings training data pollution crafted by adversaries that misclassify malicious samples or activities. Several privacy-preserving techniques exist, such as differential privacy, homomorphic encryption, secure Multi-Party Computation (MPC), or Trusted Execution Environment (TEE) usage. Those techniques can bring one or multiple drawbacks, such as a significant increase of the computational overhead, or they can require customizing specific incompatible models. The authors found no universal approach to ensure privacy and/or security. Multiple defences have been found for adversarial attacks: apply input pre-processing, which reduces the influence of immunity, enable malware detection, which introduces regulations, adversarial training, feature denoising, models robustness improvement, or models modification and retraining, or improve the model robustness by detecting attacks using stateful detection, image transformation detection, or adaptive denoising detection. Again, no universal defence method has been found by the authors. Two defences for poisoning attacks has also been found, such as outlier detection mechanism, which removes outliers outside the applicable set, and improving the NN robustness.

Table 2.4 Artificial intelligence attacks and mitigations comparison

Paper	Summary	Useful Data
Bösch et al. [47]	Study of the privacy dark strategies and patterns	Some of the most used dark patterns are privacy zuckering, bad defaults, forced registration, hidden legalese stipulations, immortal accounts, address book leeching, and user profiles shadowing.
Gray et al. [148]	Ethical concerns in UX dark patterns	The authors found multiple general types of dark patterns: baits and switch, disguised ad, forced continuity, friend spam, hidden costs, misdirection, price comparison, prevention, privacy zuckering, roach motel, sneak into basket, and trick questions. The primary dark patterns, which are strategic motivators for designers, are nagging, obstruction, sneaking, interface interference, and forced action. Dark patterns are not always intentional.
Mathur et al. [48]	Review of dark patterns used in eleven thousand shopping websites	The authors have found seven categories for fifteen types of dark patterns: sneaking (sneak into basket, hidden costs, hidden subscription), urgency (countdown timer, limited-time message), misdirection (confirm shaming, visual interference, trick questions, pressured selling), social proof (activity message, testimonials), scarcity (low-stock message, high-demand message), obstruction (difficulties to cancel actions), and forced action (forced enrolment).
Luguri and Strahilevitz [50]	The power of dark patterns	The authors defined dark patterns taxonomies: nagging, social proof (activity messages, testimonials), obstruction (roach model, price comparison prevention, intermediate currency, immortal accounts), sneaking (sneak into basket, hidden costs, hidden subscription/forced continuity, baits and switch), interface interference (hidden information/aesthetic manipulation, preselection, toying with emotion, false hierarchy/pressured selling, trick question, disguised ad, confirm shaming, cuteness), forced action (friend spam/social pyramid/address book leeching, privacy zuckering, gamification, forced registration), scarcity (low stock message, high demand message), urgency (countdown timer, limited time message).

Table 2.5 Dark patterns categories comparison

Paper	Summary	Useful Data
Potlapally [64]	Main challenges to correctly implement security in commercial hardware platforms	The most known attacks types are active adversarial manipulation of control signals, exploit security gaps in interactions of multiple platform features, insecure platform initialization by boot-up firmware, ability of untrusted or lesser privileged entities to maliciously influence operation. A mitigation would be to apply secured SDLC.
Jin [66]	Key concepts of hardware security	The most common threats are hardware trojan, intellectual property piracy and integrated circuit overbuilding, reverse engineering, side-channel analysis, and counterfeiting. The most useful countermeasures are design obfuscation, intellectual property watermarking, intellectual property fingerprinting, integrated circuit metering, split manufacturing, integrated circuit camouflaging, integrated circuit information leakage reduction, key-based authentication, noise injection, secure-scan, physical non-clonable function or unique ID(s), and ageing sensors.

Table 2.6 Hardware attacks and mitigations comparison

Paper	Summary	Useful Data
Salahdine and Kaabouch [116]	Survey made on parts of social engineering	How to detect attacks: verify call sources, assign PINs to help desk callers, set up honeypot for spam, verify the emails sources, use anti-phishing tools, use ML algorithms, make employees aware of their environment, destroy discarded documents, limit personal computers access and USB ports, monitor the network, use *SERA*[27], apply allowlists and blocklists, identify vulnerable users. Other mitigations: report all the attacks (stops the spread), spread awareness about the psychological triggers, apply human techniques (auditing and policy, plus education, training, and awareness), apply technology techniques (biometrics, sensors, artificial intelligence, and social honeypot), have a ransomware policy (preparation, detection, containment, eradication, recovery).
Chizari et al. [118] (2015)	Researches regarding mitigation of social engineering.	Some human based detections and mitigations: education, training and awareness approach, policy and auditing approach. Human judgement is subjective, and such approaches suffer from lacks of details, leads to technology based techniques: biometrics, AI, social honeypots, sensors. Issues with technical approaches: adding cost and complexity to the system, increase attack surface, find large and up-to-date datasets.

Table 2.7 Social engineering attacks comparison

[27] *SERA* source: https://bit.ly/3s8egdH (accessed 20th October 2022)

2.3 Summary

In this Chapter, we explained all the tasks we conducted in order to complete our knowledge collection by doing a state-of-the-art review.

The methodology we used to build the whole collection of knowledge was adapted. We managed to do more than a literature review by adding additional steps ensuring a better quality and bias management, while managing to do it in our limited timeframe.

The protocol we defined could be used for similar project in the ICT field. We designed it in the most generic way possible to make it possible.

Although being convinced by our methodology, we are fully aware of the limitations of our knowledge collection. Indeed, each topic we assessed could be explored more deeply, other sources could have been included, and specific issues can have been unprocessed. However, this approach is adapted to our thesis scope, and those compromises were necessary given our limited timeframe. Nevertheless, we are satisfied with our results. In addition, the knowledge collection can be expended in the future following the same methodology.

3 Comparison of Other Guides

Once the collection of knowledge done, our next step is to compare other guides that allow assessments on ICT systems. This step will allow us to explore multiple approaches with their own characteristics and usages, which will bring useful information for our proposal that will be defined later in Chapter 4 (Guide Proposal).

We will start by defining the methodology that will be applied in order to find, analyse and compare the various guides. Then, we will apply this methodology which will lead to a comparison of those methods based on their analysis. Finally, a summary will be made on the whole Chapter content.

Contents

3.1 Methodology

A methodology has been defined to help us analyse and compare the most appropriate guides close to our scope. We designed it in a simple way while ensuring that an accurate and coherent comparison can be carried out.

3.1.1 Selection

We need to select guides that are well recognized and relevant in the security and privacy fields. They should ideally be specialized for web services and should also integrate AI-related concerns. We found citations of such guides during our knowledge collection made in Subsection 2.2.2 (Review Results).

Their format is not fixed; it could be a checklist, an evaluation, a framework, an online tool, et cetera. It could be issued by any organization, but its expertise must be established.

We will select the most adapted guides in our opinion, based on facts, legitimacy and general recognition.

3.1.2 Characteristics

Some characteristics will be extracted from the selected guides in order to expose and compare their approaches, features and objectives. Those characteristics can be exclusive to a guide or common for some of them.

The definition of the characteristics have been defined by taking our thesis scope and objectives into account.

- **Accessible**: is the guide easy and light to use?
- **AI-oriented**: does the guide include specific issues coming from the AI field? If so, are those issues sufficient?
- **Format**: how is the guide used? For example, by going through checklists or sets of directives.
- **Scoring**: does the guide provide a final score to represent the security and/or privacy levels?
- **Volume**: is the guide workload heavy for a small or medium-sized organization?
- **Web-orientated**: does the guide include specific issues coming from the web environment? If so, are those issues sufficient?

If some useful and accurate characteristics are found during the analyses, we will include them in the comparison process as well.

3.1.3 Assessment

The assessment of the guides will consist of listing them and comparing their characteristics. The goal of this comparison is to get an understandable and clear overview of the guides we found. Their analysis will allow us to present a factual, unbiased and complete comparison based on their said characteristics. This comparison will be useful to define our own defined guide.

3.2 Guides Analysis

The following Subsections contain an analysis of their respective guide.

3.2.1 NIST Cybersecurity Framework

The *NIST Cybersecurity Framework*[28] has been developed with the objective of 'Helping organizations to better understand and improve their management of cybersecurity risk'. Its format is a PDF file which can be found online on the NIST official website.

This framework is organized into five categories: *identify*, *protect*, *detect*, *respond*, and *recover*. Those categories themselves have additional categories, subcategories and references to go into specifics. It includes standards, guidelines, and practices that are useful to guide cybersecurity activities. It is focused on multiple areas which are the ICT field, industrial control systems, cyberphysical systems and the IoT topic. This tool can be used in various ways, but is oriented towards risk management practices analysis.

This tool integrates the possibility to create *profiles*, which are roadmaps that include some requirements for various specific sector goals. It enables organizations to describe their current and desired targeted states, which can also be shared across entities.

There is no evaluation or scoring capabilities with this tool, its users can only consult the descriptions of items and use them as guidance to assess their systems.

Alongside of the framework itself, NIST has published a quick start guide to help organizations to use their tool by explaining the different categories with additional information. They also provide an online learning platform to answer some frequently asked questions and to show various examples.

This framework does not guarantee a complete and exhaustive guidance, but it does try to be as complete as possible regarding cybersecurity management in the previously mentioned areas. No particular assessment process is given to use this tool, which can be confusing for its users.

Its usage is quite complex, and it has a lot of content. A lack of technical advices and implementation details can be noticed: it is mostly designed for managers than for developers. However, the framework content is quite expressive and is well referenced.

No item is specifically oriented towards AI processes nor web services. Furthermore, this tool is specialized into cybersecurity and does not include dedicated concerns about privacy.

3.2.2 NCSC Cyber Assessment Framework

The *NCSC Cyber Assessment Framework (CAF)*[29] tool aims to guide organizations to assess which entity is responsible in terms of security risks. It defines four objectives: *managing security risks*, *protecting against cyberattacks*, *detecting cybersecurity events*, and *minimizing the impact of cybersecurity incidents*. Each objective includes principles of specific security aspects which list and explain various rules to be respected.

[28] *NIST Cybersecurity Framework* source: `https://www.nist.gov/cyberframework` (accessed 18th November 2022)

[29] *NCSC CAF* source: `http://bit.ly/3AxqEs7` (accessed 18th November 2022)

This guide can be accessed online by browsing the principles, which are also categorized and indicated by a letter depending on the objective they cover. Each rule that a principle includes is defined with a title and its explanation. No scored evaluation can be conducted, but the rules are described in a way that allows to verify the assessed system compliance with them.

No specificities are given for AI processes concerns, neither for user privacy ones. Furthermore, this tool is not oriented towards web services. However, each principle can be overviewed in tables that efficiently summarize how to be compliant to each given rule.

A notation based on the compliance level for each rule is given, which can be either achieved, not achieved, and sometimes partially achieved. Levels are determined by the compliance observed within the organization by evaluating the description of each level contained in rules.

Additional online resources are given to explain and help on how to use the guide. They are available on the same website as the tool. A complete guidance can also be consulted.

The CAF is somewhat heavy to approximate because of its verbosity. Moreover, the rules descriptions can be complex to assess, with a lot of details. However, we noticed that this tool is quite complete in its coverage. It also provides additional and multiple resources for each principle, such as other relevant guides.

3.2.3 *ENISA* Resources

ENISA[30] is the European Union agency for cybersecurity whose activity is to spread awareness about cybersecurity across Europe. They do not provide any complete tool for ICT systems security and privacy assessment in their entirety, but they do publish multiple projects for organizations on specific scopes. We selected two of them that best suit our scope: The *Risk Level Tool*[31] and the *SecureSME project*[32]. The latter includes various guides that aim to raise the cybersecurity levels in organizations.

The *Risk Level Tool* evaluates risk levels of a personal data processing operation by doing a security risk assessment. The evaluation is composed of five steps: *definition and context*, *impact evaluation*, *threat analysis*, *risk evaluation* and *security measures*. It is doable online and provides a list of security measures with their related risk level based on the form previously filled. Measures are given for each category of security issues listed in tables.

This tool is highly specialized on privacy concerns and has been designed for the organizations decision makers, without providing any technical details. It is quite complete and therefore long to conduct, however its online availability and simple navigation makes the tool accessible given the complex field of risk management.

Once filled, the result can then be shared, printed or saved. No overall score of the risk level is given.

[30] *ENISA* source: https://enisa.europa.eu (accessed 18th November 2022)
[31] *Risk Level Tool* source: http://bit.ly/3Xqkkwi (accessed 19th November 2022)
[32] *SecureSME project* source: http://bit.ly/3EApptm (accessed 19th November 2022)

The *SecureSME project* includes several specialized guides. They are all different, but they all share the common goal to help organizations to secure their ICT systems. They are focused on three categories, which are employees' protection, process enhancement and reinforcements of technical measures. The guides can be written in English, French or German, because each one of them is produced by local European agencies.

A summarized version of the guides named *Cyber Tips* can be found online: it lists questions for each category that must be answered to reach higher security levels. This resource is useful to get an overview of the *ENISA* guides in a lighter way. However, no guidance or details are given to go further.

The *Cybersecurity guide for SMEs - 12 steps to securing your business* leaflet provides twelve high level steps to help organizations to secure their systems. It is based on the *Cybersecurity for SMEs - Challenges and Recommendations* report, which contains more details and specifics. Each step has a single or multiple advices that can be applied in order to enhance security, with a small description of them.

The *Cybersecurity Guide for SME* is an equivalent of the leaflet above: it also lists various categories that includes multiple items to be compliant with to raise security levels. Each item are explained by small descriptions.

Other resources exist for particular topics such as COVID-19 concerns or supply chain attacks risks: yet, none of them are useful for a generic guidance. Overall, no resource is web-oriented or includes AI processes. Apart for the risk assessment, no evaluation can be done on systems.

The resources we analysed are quite accessible and easy to understand, but they are shallow and have a lack of specifics. They are useful for simple guidances, but the *ENISA* project is harder to browse and to understand as a whole than the other guides that have a unique method and guide.

3.2.4 CISA Cyber Essentials Toolkits

The *CISA Cyber Essentials Toolkits*[33] is a collection of actions designed to help organizations to bring cybersecurity in their processes. Each toolkit is specialized in a specific area of organizations: *leader*, *users*, *systems*, *surroundings*, *data* and *crisis response*.

The toolkits are presented on two PDF pages using a user-friendly and well-presented design. A generic task is defined to improve security levels, with a linked set of actions that must either be applied by leaders, ICT staff or service providers. The actions contain a brief description and give additional resources to take action.

One of the toolkits gives an action that directly concerns data privacy issues, but nothing generic or centred on user privacy is given in the other ones. Moreover, neither the web area nor AI processes are mentioned or taken into consideration.

This tool is not intended to be used as a guide, just as an entry point to provide further resources that can be used to improve cybersecurity levels by adequate actions.

[33] *CISA Cyber Essentials Toolkits* source: `http://bit.ly/3GOf3bY` (accessed 23rd November 2022)

Few rules are given: the toolkits have been designed to be short and concise. However, they are easy to use and to understand thanks to their brevity.

Some webinars on this tool are available on *Youtube*, which can be useful when actions need to be taken in an organization. A small overview of each area of specialization including some advices is also provided on the toolkits website.

3.3 Guides Comparison

As explained in Section 3.1 (Methodology), a comparison between the resources has to be realized. The final results are shown in Table 3.1 (Comparison of the selected guides based on their characteristics). Each guide name has been shortened to the title of their respective organization for readability purpose.

Characteristic	NIST	NCSC	*ENISA*	CISA
Accessible	✓	∼	✗	✓
AI	✗	✗	✗	✗
Format	list of items	tables of rules	multiple	set of actions
Privacy	✗	✗	∼	✗
Scoring	✗	compliance levels	risk matrix	✗
Volume	high	high	medium	low
Web	✗	✗	✗	✗

✓: fills the criteria; ✗: does not fill the criteria; ∼: almost fills the criteria.

Table 3.1 Comparison of the selected guides based on their characteristics

We can see that for all the guides we analysed, none of them include specific content focused on AI processes. Furthermore, technologies used by web services are not represented, despite their wide adoption.

The format of the guides varies a lot between them. Their volume also changes according to their diverse workload. The *ENISA* resources and CISA Toolkits are the lightest guides, with the latter being the most accessible one.

The resources have a lack of scoring capacities, which would be useful for developers and decision makers whom wish to get an overall score of their security and privacy levels. Furthermore, getting scores would allow to prioritize the most sensitives parts of an evaluated service.

Finally, the guides we found are focused on security issues only. We noticed some actions oriented towards data privacy in one of the *ENISA* resources, but nothing has been specified in terms of user privacy. Hence, we added the *privacy* characteristic into the comparison table. This lack could be explained by the fact that privacy in ICT systems is considered as a separate concern from security. But in our opinion, those two concerns are closely related, and additional serious issues can appear from weak privacy levels.

3.4 Summary

This Chapter has presented an analysis and a comparison of multiple guides close to our thesis scope. The organizations that created them represent the countries that have the stronger impact and capacities in the security domain and cybersecurity surveillance: the United States of America, the United Kingdom, and the European Union.

We discovered that the guides have various characteristics, but none of them have integrated specific risks and issues from the privacy field, the AI processes field or the web environments in their content. This observation is concerning because of the recognition of each of the organizations that created those guides: indeed, a lot of ICT decision makers and developers follow their recommendation, which could lead to a lack of security and privacy levels for those specific three fields. There is therefore a lack of consideration towards lots of major risks, which we can contribute to fill with a new proposal.

The said new proposal should bring additional considerations to the previously identified lacks, alongside a coverage on the more classic ICT security issues. Moreover, its format, volume and accessibility should be optimized and designed to simplify its adoption and usage. By doing so, this proposal would meet all the characteristic we defined, which means that it would bring a strong added value for developers and ICT decision makers.

4 Guide Proposal

Once the collection of knowledge and the comparison of other guides done, we have to propose a new way of defining a guide. This Chapter will expose our considerations toward this challenge and how we defined an effective, accessible and complete guide. We will start with the global considerations of the guide and finish with its precise parts.

First, an explanation of the capabilities that the guide must have will be made. Then, the medium that will distribute the guide will be discussed and chosen, followed by the definition of its structure. Once those parts done, the guidelines on how to create the guide content will be defined. Afterwards, the ability to get a score through evaluations made by the guide will be discussed and then defined. The creation of the guide content by following our proposal will then be explained. We will conclude by a comparison between our proposal and the other guides already compared between each other in Chapter 3 (Comparison of Other Guides).

The other guides will play a role in the definition of our proposal: we analysed various ways of building, exposing and formatting them, and we noticed some of their useful approaches. We will use this knowledge in order to refine, in our opinion, the best guide definition that serves our goal.

Contents

4.1 Guide Capabilities

First and foremost, we have to define what the guide must be able to do for its users. We will refer to them as assessors to avoid any confusion with the end users of their web services. Furthermore, we must define who will be using it, as well as the final output of the guide. Those definitions are essential for an adequate, adapted and well-defined work process.

The capabilities of our guide must meet the global objectives of this thesis, described in Subsection 1.3.1 (Thesis Objectives). In order to define those capabilities appropriately and to be sure that our approach is complete, we defined a set of question that must be answered to cover all the context.

1. **Why** should the guide be used?
2. **What** must the guide be able to **do**?
3. **How** to use the guide?
4. **Who** are the guide targeted users?
5. **When** should the guide be used?
6. **What** are the **results** of the guide usage?

After reflexion, here are the answers for those prior questions:

1. To **improve** the security and privacy levels of a web service, including considerations on the AI topic.
2. An easy, quick and accessible **assessment** on security and privacy risks.
3. By evaluating whether the service is **compliant** with a set of directives designed to mitigate security and privacy risks.
4. It is addressed to the **developers and/or decision makers** of the service. The latter must have an adapted technical understanding in order to use the guide.
5. Ideally during the **(Secured) SDLC** process done for the service development. Its usage would also be possible on existing services, but it would be less convenient because of the costly and painful changes that come when modifying already built pieces of software.
6. A **scored evaluation** of the service security and privacy levels, which allows to get an accessible overview of its compliance with the set of directives designed to mitigate risks.

The following Sections will describe more specific aspects of the guide based on the answers of the questions.

4.2 Medium Choice

The medium of the guide is determining for its ease of use and accessibility. Indeed, potential assessors must not be frightened to understand how to use the guide, nor be afraid of its workload.

In our case, we wish to make our guide as available as possible for assessors. To this end, we found two mediums that are the most appropriate choices given their universality, ease of use, and availability: PDF files and web applications. The PDF files must be published in order to access them and websites are often used for this purpose. Those websites often include additional information about the PDF files themselves.

The medium choice is not decisive for the next steps of our proposal definition. Indeed, the content of the guide is the same regardless of the medium choice, but its layout and presentation will be optimized for the medium we choose.

The two mediums have different advantages, which is why we defined the characteristics that are the most relevant for our needs. Those mediums will then be evaluated based on their compliance with those characteristics to make an appropriate choice.

- **Accessibility**: do assessors need particular knowledge about how to use the medium?
- **Availability**: is the medium available under all circumstances? Takes into account the Internet connection.
- **Extendability**: is the medium easily upgradable with new items or resources?
- **Interactivity**: can assessors interact with the medium?
- **Interoperability**: can assessors access and use the medium on any platform and/or client?
- **Sustainability**: is the medium maintenance easy and light?
- **Technicity**: does the medium require a technical infrastructure?

Table 4.1 (Advantages of the considered mediums for the guide) shows an evaluation of each medium on the characteristics we defined. Because of their complementarity, we would like to publish both mediums. Indeed, the interactivity that web applications provide is a strong advantage for a guide that includes an evaluation process. This point is ever more relevant considering the fact that we need to design a light guide for the assessors. Furthermore, new resources or updates can be distributed more conveniently. However, a PDF file can be accessed using any device, does not require any Internet connection if already downloaded, and is less sensitive to technological evolutions. Its main limitation for our use case is its ability of integrating scoring capabilities that must be compatible on all major PDF readers.

Due to the thesis timeframe, we have chosen to restrict ourselves to the web application medium only. The creation of a PDF file could be conducted outside the thesis scope as further addition.

Characteristic	PDF file	Web app.
Accessibility	equivalent	
Availability	++	+
Extendability	+	++
Interactivity	limited	broad
Interoperability	+	++
Sustainability	lighter	heavier
Technicity	lighter	heavier

++: totally fills the criteria; +: somehow fills the criteria.

Table 4.1 Advantages of the considered mediums for the guide

With this medium choice, new considerations must be taken into account to ensure adapted access to the guide for the majority of assessors:

- **Browsable**: the navigation and UX must be as simple as possible.
- **Compatible**: the web application must be compatible with all the major modern browsers.
- **Offline usage**: the web application must provide an offline usage.
- **Usable**: the web application must be designed as simple as possible.

Other mediums have been considered but have been evaluated as unsuitable for various reasons:

- **Mobile application**: the added benefits of this medium comparing to a web application are not adapted to our needs. Furthermore, doing an evaluation on a mobile platform is less convenient because of the screen size. Finally, the fact that mobile applications must be installed first can be an adoption barrier.
- **Desktop application**: same reasons as for the mobile application medium plus the fact that desktop applications can be difficult to be installed in professional environments. Furthermore, the maintenance of such applications is heavier.
- **Paper-based**: difficult to distribute. Moreover, a PDF file can be easily printed.
- **Single or multiple video clips**: not suited for an evaluation process that requires non-predictable break times while assessing the guide content. The production of video clips is also too heavy for this project. However, the guidance provided by video clips is very high, which is great to lead a such processes.

4.3 Structure Guidelines

In order to build a simple yet complete guide, we need to define a structure that allows guidance in an understandable manner. To this end, we defined several structural parts to classify related content in a hierarchical manner: content nature, categories, and objectives.

Each structural part has been defined in order to be both compatible and distinguishable from the others. The main goal of this approach is to efficiently guide the assessors in the process without creating any complexity. The visual features defined for each structural part have been defined in order to limit the readability overhead.

Figure 4.1 (Flowchart of the guide structure) shows a summary of each structural part defined in this Section in the form of a flowchart.

4.3.1 Content Nature

The guide content must be classified based on its relative importance: indeed, some elements are more useful and essential than others. In order to keep the guide as light as possible in its original state while allowing it to be able to go into specifics, we defined a binary classification on the content nature.

A **primary content** has been defined as any item that is considered as mandatory to understand the context of an assessment. Following this statement, a **secondary content** has been defined as any related item to a primary content that is not as important for the assessment understanding. A secondary content must be directly link to only one primary content.

Due to the subjectivity of this classification, we defined a set of rules to determine a content nature:

1. **Assessable**: a primary content must enable an assessment.
2. **Brevity**: a primary content must be brief.
3. **Specificity**: a primary content must assess a single item.

The chosen guide medium being a web application, the secondary content must be displayed or hidden using non-modal specific containers relatively to their primary content through buttons, information icons or other interactive elements. Navigating to separated pages can also be considered, provided that the navigation is reversible.

If a PDF file is developed, the main part of the file must contain the primary content, with links to the appendices part that contains all the secondary content.

This classification allows the items placed in the primary content to be evaluated sequentially without interruption, while having links to further explanations and/or resources represented by the secondary content when necessary.

4.3.2 Categorization

The ICT field is very broad and contains a lot of different areas of specialization. Even considering only web services, many areas are necessary to build such systems. The Section 2.2 (Review) shows this diversity.

Exposing a bunch of uncategorized list of items to be assessed would include a massive and shapeless workload. In order to lighten this assessment, we defined a categorization on the guide content.

No research or common ground on ICT categorization has been found. Therefore, we defined our own approach.

The categorization must be done on two layers, with global categories acting as the less specific layer, and their related subcategories that are more specific. All the items to be assessed must be assigned to a subcategory.

We limited the amount of layers to two for simplicity and understanding purposes. Adding another layer of subcategories would bring a lot of complexity: being more specific would be a greater pain in the guide usage than adding it to get a more precise categorization. Furthermore, there are no needs to go deeper.

The categorization must be defined in a way that optimizes the understanding of the assessors. Following this principle, we defined a set of rules to be respected while defining categories:

- **Complete**: all items must be able to fit in a category and in one of its related subcategory.
- **Exclusive**: an item must not be duplicated or included in two subcategories.
- **Familiar**: the categories and subcategories must be similar to other ones used in the ICT field, such as articles, websites or job descriptions.
- **Granular**: each category must be able to be separated into several more specific subcategories where items are assigned to.
- **Numbered**: the amount of categories and subcategories must be limited in order to avoid drowning assessors in too many topics.
- **Specialized**: the categories and subcategories must have enough inner specificities so that they can be distinguished from each others.

We did not split the items depending on their topic, security and privacy, for reasons explained later in Subsection 4.4.1 (Item Topics).

4.3.3 Objectives

The last structural part of our proposal are the objectives. Displaying sets of items to be assessed directly into the subcategories was the simplest solution in terms of structure, but we noticed that the assessors might fell overwhelmed if the sets of items are too

long. Furthermore, the evaluation process would give assessors no reason to comply with these sets of items if no objective is given. For those two reasons, we decided to include this last piece of structure into our guide proposal.

Objectives are ideal for providing an understandable list of the considerations that must be met to reach appropriate security and privacy levels for each part of an evaluated web service. They can even be useful when taken alone without their related items as an overview to get a summary of what should be enforced for each (sub)category.

Objectives must be consistent with each others in their formulation. They must form basic sentences with a subject, a verb and a complement, finished by a final point. The sentences must qualify the goal of the objectives in a clear and comprehensive way. The verbs must be in the simple present passive tense because the sentences subject are not the ones that realizes the actions. The length of the sentences is not bounded, but the amount of words should be as short as possible to optimize the reading process. Furthermore, objectives must always encapsulate the set of items that contribute to their fulfilment. An example is given below:

> The distributed systems are compliant with the defined security protocols.

Each objective must be consistent and must be achievable. To this end, they must be defined by applying the Specific, Measurable, Achievable, Relevant, Time-bound (SMART) method. This method has been chosen because of its broad adoption both globally and in the ICT field. Here is an explanation of this method:

- **Specific**, must be well-defined, understandable and focused on a single goal.
- **Measurable**, must include metrics to indicate its fulfilment. The metric of the objectives is the compliance of the evaluated web service with the objects contained by the objective.
- **Achievable**, must (eventually) be possible to reach.
- **Relevant**, must be oriented toward the project scope. All the objectives must aim to improve the security and/or privacy levels.
- **Time-bounded**, must include a deadline for its fulfilment. All objectives must be met by the evaluated web service when using the guide.

The definition of objectives must be realized alongside the definition of items that is explained in Subsection 4.4.3 (Items Formulation). The related items must be grouped together based on their common ground and appropriate objectives must then be defined to express the goal that each set of item is aiming to reach.

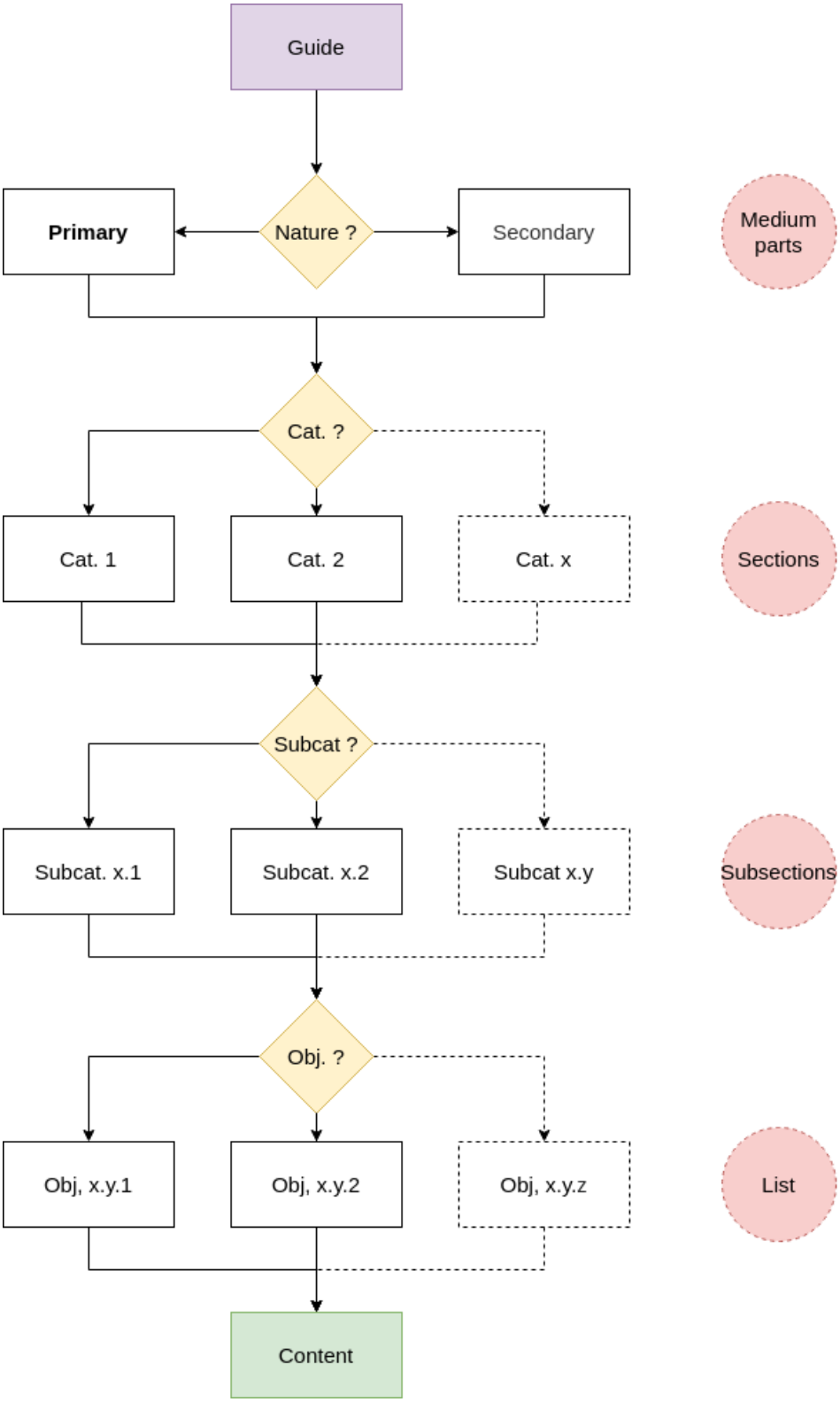


Figure 4.1 Flowchart of the guide structure

4.4 Content Guidelines

Once the guide structure specified, the process to create its content will be defined. The guide content refers to the sets of items that must be evaluated by the assessors during the guide usage.

Each part of the content must respect the same principles as described in Section 4.3 (Structure Guidelines): each element of the guide must be both compatible and distinguishable from the others, for both structural and content elements.

Figure 4.3 (Flowchart of the guide content) shows a summary of each content part defined in this Section in the form of a flowchart.

4.4.1 Item Topics

Our thesis scope includes two different fields: security and privacy. This combination can make it difficult for assessors to understand on which field an item is focused. To avoid such issues, we defined a labelling system for the items: by displaying a visual feature, assessors have an indication of what field the items are about.

We have called this labelling *topics* because we found this word as the most appropriate to differentiate the two fields. We did not choose the word *fields* in order to avoid any confusion with the guide categorization. The topics that qualify the guide items have therefore not the same meaning that the definition we made in Section 1.1 (Definitions), which refers to the various ICT fields.

We considered multiple visual features to indicate the topics: images, colours, icons, symbol, text labels, or an additional categorization on top of the one defined in Subsection 4.3.2 (Categorization). This choice must be done by considering our needs: an easy recognition, a low visual load, and a simple feature. Once those needs considered, we decided to use icons that must be added alongside the corresponding items, composed of the following characteristics:

- A unique **shape** that is not used by other elements of the guide.
- A unique yet lightly loaded **colour** that is not used by other elements of the guide, applied as a background colour.
- The **first letter** of the topic contained into the shape, using a contrasting colour from the background colour.

Figure 4.2 (An example of topic icons) shows an example of two icons that could be used in the guide medium.

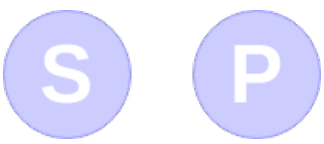


Figure 4.2 An example of topic icons

In order to have a consistent policy, an item that concerns both topics must have both icons assigned. Indeed, removing the icons for such items would break the consistency inside the sets of item, and a third icon would add more complexity for the recognition of topics. Displaying both icons is the simplest way to show this case, despite an extra visual addition.

If our proposal is used to evaluate other fields that the security and privacy levels, the topics can be changed for more appropriate terms.

4.4.2 Requirement Levels

While building our knowledge collection, we have noticed that non-compliance with the rules, technologies, or principles that aim to improve the security or privacy levels of systems leads to a variety of risks with varying degrees of severity. This variety causes a need to give different priorities to objects according to the risk levels related to them. To this end, different requirement levels for items have been defined.

Method

Multiple ways of expressing requirement levels have been explored by Javed et al. [149], and the Must-have, Should-have, Could-have, Won't-have (MoSCoW) method has the best advantages according to our needs. This method is widely used, is easily understandable, and has been peer-reviewed, although it has not been evaluated as the best prioritization assessment. Indeed, no inner priority is given to items in the same priority levels, and the *Won't have* priority can lead to confusion regarding its scope. However, those two limitations do not impact our approach because we do not have any needs on inner prioritization of items for simplicity purposes, and we made an assumption that every guide items must be covered at the evaluation time, which rules out the utility of the *Won't have* level.

We created a hybrid method using almost all the MoSCoW priority levels for expressing the requirement levels, and a risk matrix to classify the guide items in the said levels. Explanations and descriptions of the requirement levels are the following:

- **Must Have**: the evaluated web service must be compliant with the corresponding items. Mandatory, because a non-compliance includes severe risks.
- **Should Have**: the evaluated web service should be compliant with the corresponding items. Compliance is not mandatory because of the lower severity of risks, but would greatly improve the security and/or privacy levels.
- **Could Have**: same condition that the *Should Have* level, but a non-compliance brings lower risks.

Two levels can be used for the not mandatory items in order to prioritize the efforts to be done by assessors to improve their web service overall security and privacy levels. Non-compliant items that include higher risks should be processed before the lower risky ones. However, the evaluated web service should eventually be compliant with all the guide items to be considered as totally secure and private according to our knowledge collection.

Visual Feature

The requirement levels must be differentiated by **colours**. We thought of similar visual features as the ones explained in Subsection 4.4.1 (Item Topics) and found out that adding colours to items would be the lighter and most adapted way of expressing a requirement level. This choice has mainly been motivated by fact that we have three levels of risks, which can be likened to the information that is communicated by the status of traffic lights in almost all regions of the world. The more risky situations are shown by a red colour, and the less risky ones by a green colour. Following this encoding method, the colours that must be used are the following:

- **Red**, for the *Must Have* level.
- **Yellow**, for the *Should Have* level.
- **Green**, for the *Could Have* level.

Risk Matrix

In order to classify the items into their corresponding requirement level, we defined and used a risk matrix. By assessing each item with their probability to occur and their severity if they happen, we are able to classify them in the most appropriate requirement level with regard to their characteristics, being probability and severity.

Multiple organizations have defined their own risk matrix with their own sensitivity or bearings, but the core of this tool is always the same. Different levels of probabilities and severities are defined to express the risk of each item. Those levels define their strength and are linked to a coefficient according to their said strength. Then, each intersection between the two characteristics give a risk score based on the multiplication of the two coefficients. Finally, intervals of score, or bearings, are given to classify items into a risk level. In our case, those risk levels are requirement levels.

Our risk matrix is shown in Table 4.2 (Our defined risk matrix). It contains five levels of probability and sensitivity grades in order to ensure a certain granularity on the assessment of items. Linear coefficients have been defined in order to have a consistent and distributed grading. Furthermore, we respected the Likert scale principle in order to guarantee equal distances and same amount of levels for each side of the characteristics.

Our bearings have been defined with intervals in a way to be consistent with our definition of risk levels. Indeed, we gave the biggest weight to *Must Have* items, a similar yet lighter weight to *Should Have* items, and the lightest weight for the *Could Have* items. This way, we are able to indicate assessors what are the most risky situations. The following is the definition of our bearings:

- **Could Have**: $[0, 5)$
- **Should Have**: $[5, 15)$
- **Must Have**: $[15, 25]$

			Severity				
			Insignificant	Minor	Moderate	Major	Catastrophic
			1	2	3	4	5
Probability	Very unlikely	1	1	2	3	4	5
	Unlikely	2	2	4	6	8	10
	Possible	3	3	6	9	12	15
	Likely	4	4	8	12	16	20
	Very likely	5	5	10	15	20	25

Table 4.2 Our defined risk matrix

However, we are aware that risk matrices are not optimal. Anthony has found four limitations on them [150]: they provide poor resolutions, have a low resistance to errors, they carry a suboptimal allocation of resource, and can be ambiguous because of subjective interpretations. However, this method has been broadly used in various different domains despite its limitations. Furthermore, it still offers a simple, easy and accessible way of evaluating risks: those characteristics allow risk matrices to provide a great quality-time compromise which is optimal for our thesis scope, especially regarding our project timeframe.

The risk evaluation can be easily changed for another approach or method. The guide structure and content have been defined in a generic way that allows such modifications.

4.4.3 Items Formulation

Finally, each item must be formulated appropriately to provide a good understanding to assessors. This formulation must help them to determine whether their web service is compliant with all the guide items. To this end, we made a list of the most appropriate formulation possibilities based on that we saw in other guides and on our knowledge:

- **Actions**: sentences composed around verbs that imply to do something.
- **Rules**: injunctions given to parties to comply to one or multiple acts.
- **Questions**: sentences that request information.
- **Statements**: sentences without any specific feature that describe a state.

Each formulation comes with advantages and disadvantages, as shown in Table 4.3 (Advantages and disadvantages of the potential items formulations). The most appropriate formulation in our opinion is to express items as rules, because of the precision that rules provide. Their disadvantages are the less disruptive ones according to our scope: being too guided and be somewhat harsh can help to lead the assessors using the role given by being the author of the guide. Our second formulation choice was the statements, but the lack of consistency would have harmed too much the consistency of the guide. The disadvantages of the other formulations are too large.

Formulation	Advantage(s)	Disadvantage(s)
Actions	Easily applicable	Not in the same timeframe Difficult to assess
Rules	Precise Exhaustive	Feeling of guidance and harsh
Questions	Help to project oneself	Longer sentences Additional complexity
Statements	Free form Permissive	Lack of consistency

Table 4.3 Advantages and disadvantages of the potential items formulations

The rules must be formulated as imperative sentences with a final point. The sentences must start with their verb in the imperative tense, and this verb must give the nature of the compliance. The verb must then be followed by the subject that is targeted by the rule. Then, some context must be brought in order to give information on how to assess whether the system is compliant or not with the item. An example is given below.

Include the dependencies in the testing process.

4.4.4 Items Descriptions

Items can have none, one or multiple descriptions to provide additional information about them. Those descriptions are classified as secondary content, as described in Subsection 4.3.1 (Content Nature). A description can contain anything: paragraphs, external links, external resources and so on. Basic English rules must however be respected.

4.4.5 Items Evaluation

Every item must give to the assessors the possibility to state an evaluation about the web service compliance with the rule it represents. A compliance is a binary classification, which bring two evaluation values to be provided.

However, some use cases can lead the evaluated web services to not be concerned by items. For example, an item can express a risk on cloud hosting, but the web service can be hosted by the developers' organization itself. For those scenarios, a third evaluation value should allow to remove the subject from the evaluation process to avoid any penalizations of the web service without choosing the compliant value.

The evaluation of items can have three values:

- **Compliant**: the web service is compliant with the item.
- **Non-compliant**: the web service is not compliant with the item.
- **Not concerned**: the web service is not concerned by the item.

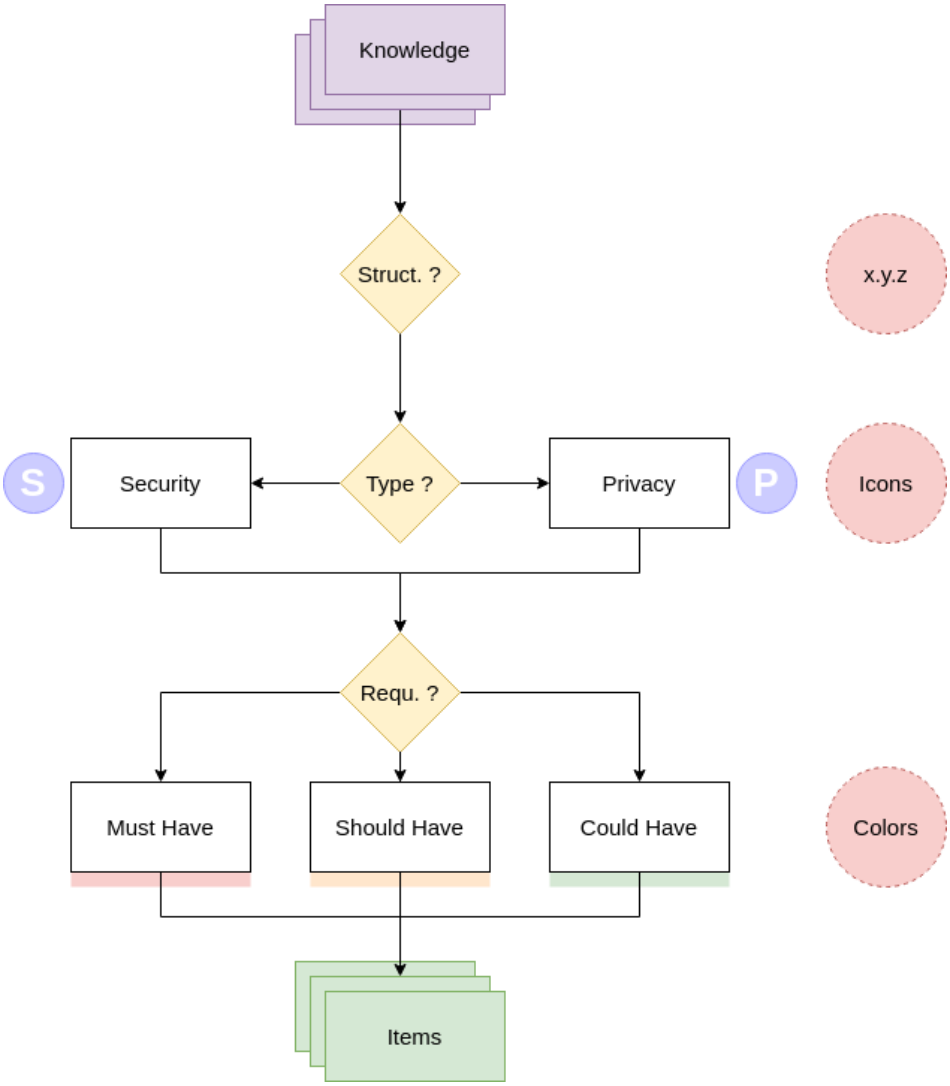


Figure 4.3 Flowchart of the guide content

4.5 Scoring Capacity

One goal of our proposal is to provide a score of the results following evaluations made on web services. This method enable assessors to get a comprehensive, summarized and simple indication of their service compliance with the items risks. To this end, an adequate way of expressing those results is needed.

The score must be able to inform the assessors on the items and categories where their web services are the most exposed to risks. To this end, two types of scores must be given to them:

- **An overall score**, which indicates the level of compliance with the whole guide. Only one value is given.
- **Multiple category scores**, which indicate the level of compliance with each defined category of the guide. Multiple values are given.

The overall score provides an overview and can be used as a basis for further comparisons. The category scores provide useful information about which area of the system should be prioritized for a quick and efficient improvement of the security and privacy levels.

The overall score must be independent of the category scores to ensure a proper evaluation of all items with respect to their own weighting without consideration made on the categorization.

4.5.1 Scoring Methods

Multiple methods have be considered to express score values. We found three different major ways of doing so:

- **Percentage progress**, which indicates proportions of compliance and non-compliance on a scale going from 0 to 100.
- **Gradings**, which give grades on a pre-defined scale to indicate a level of compliance. Could be expressed by numbers, letters, symbols, or other symbols.
- **Pass or fail notations**, which are binary values indicating a compliance status without intermediate states.

In order to decide on the most appropriate method for our needs, we defined the most relevant characteristics with our scope:

- **Granularity**: can the method express various levels of compliance in its results?
- **Simplicity**: is the method easy to understand?
- **Universality**: is the method recognizable and understandable regardless of cultural and/or educational contexts?

Table 4.4 (Characteristics of considered scoring methods for the guide) shows each scoring method compliance with the characteristics we defined. Because of multiple local ways of expressing a grade in schools and universities, the gradings method may not be understood by everyone. The pass or fail notations method would be an optimal solution because of its simplicity and universality, but no granularity can be expressed. Yet, granularity is needed in our guide to express the level of compliance of web services. The percentage progress is nevertheless a great choice appropriate with our needs. We could argue that a percentage progress is a kind of gradings method, but encoded using a universal method.

Characteristic	% progress	Gradings	Pass or fail
Granularity	++	+	None
Simplicity	+	Not guaranteed	++
Universality	+	Not guaranteed	++

++: totally fills the criteria; +: somehow fills the criteria.

Table 4.4 Characteristics of considered scoring methods for the guide

4.5.2 Calculations

Appropriate calculations must be computed in order to get an output (score) from a set of inputs (items evaluations). As explained in Subsection 4.4.5 (Items Evaluation), each item must either be classified as compliant, non-compliant or not concerned by the assessors during evaluation. A not concerned value is treated as a compliant one in the calculation, because the items that do not apply to the web services context must not degrade the scores. The scoring method must therefore show the web service level of compliance given the set of binary inputs.

As explained in Subsection 4.4.2 (Requirement Levels), each item must receive a level of requirement based on the risks that its non-compliance brings to the evaluated web services. In order to build a coherent scoring system, the scores should take into the risks for each item: to this end, the items risks value item must be used as weighting factors for the calculations.

We also considered taking into account all the items without any weighting factor for simplification purposes, but the simplicity that its would bring is not significant and is not very useful. Indeed, we do not have any particular computational limitations to realize the calculations. Another approach would have been to use weighting factors based on the level of requirement with a fixed scale, for example 1 for *Could Have*, 2 for *Should Have* and 3 for *Must Have*. This would lead to a loss of precision in the score, and we would still have the same calculation complexity as the risk-based method, which explains why it has not been chosen.

We defined the formula that must be applied on the items evaluation values. The formula scope changes with respect to whether the overall score or a category score is computed by changing the set of inputs. Furthermore, each category score is limited to the set of items contained in its corresponding category.

Equation 4.1 shows how the scores are computed, with N being the set of inputs, n_{risk} the risks value of each item and $n_{compliance}$ a binary indicator on whether each item is evaluated as compliant/not concerned (1) or non-compliant (0). We have chosen a linear function because the weighting factor given by the risks values already degrade the non-compliant items. However, a logarithmic scale or a normalization of the risks values may be necessary to avoid too large weighting and outliers in the evaluation results. This aspect will be tested in Chapter 6 (Web Service Evaluation).

$$Score[\%] = \frac{\sum_{n=1}^{N} n_{risk} \times n_{compliance}}{\sum_{n=1}^{N} n_{risk}} \tag{4.1}$$

This scoring method is our heuristic based on our considerations, knowledge and aimed objectives. Other methods could also be adapted to provide a scoring capacity, which our proposal could easily adopt.

4.6 Definition of the Guide Content

Now that our proposal has been formally defined, and its relative guidelines explained, the actual guide content will be created by applying our said proposal. We will use a platform-agnostic method in order to allow our guide content to be compatible with any other mediums that the one we have chosen at Section 4.2 (Medium Choice).

4.6.1 Support

We decided to use a spreadsheet to create our guide content. This structured support is the best compromise for a generic way of storing data: it is available on all major platforms and devices, is not linked to a proprietary technology, does not need any particular software apart from the spreadsheet program, and is broadly adopted across the world.

A spreadsheet can also be easily adapted to fit into a database: its tabs can be seen as tables, its columns as attributes, and rows as records. Furthermore, relations between records can be defined as well.

We chose to work with the *ODS* file format which is a free and open file format used by *LibreOffice*[34] and other free and open office suites. This choice allows us to work efficiently on the file while using full capacities of spreadsheet programs. If needed, *ODS* files can easily be exported as *CSV* files, which are comma-separated file formats optimized for exchanging data between applications.

We also considered using the XML file format, but we wanted to make our data file accessible and clear for potential readers without having to use a specific tool. The JSON structure has also been considered, but this format is not well suited for structured data: it is therefore not used for the guide content creation part.

4.6.2 Spreadsheet File Structure

The spreadsheet file structure must be designed in a way that allows us to define our content while applying our proposal. Furthermore, we must not be limited in the content we want to create.

As shown in Figure 4.4 (File structure), we used the system of tabs to define each part of the guide structure and content. Each part have attributes that are needed for their definition, as shown in Table 4.5 (Attributes of the file). The attributes types have been designed in a way to avoid replication and to allow relationships between each others. If any, attributes constraints and formats are also given into the Table.

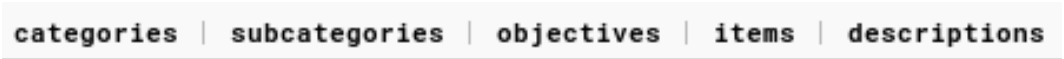


Figure 4.4 *File structure*

[34] *LibreOffice* source: https://www.libreoffice.org (accessed 5th December 2022)

Tab	Attribute	Type	Description
cat.	id	integer	Identifier, must be positive and continuous
	name	string	Displayed category title
subcategories	category	integer	Reference to the category identifier
	id	integer	Identifier, must be positive and continuous
	name	string	Displayed subcategory title
	description	string	Displayed subcategory description
	PK	string	Unique reference calculated from subcategory id and category id
objectives	subcategory	string	Reference to the subcategory PK
	id	integer	Identifier, must be positive and continuous
	name	string	Displayed objective title
	PK	string	Unique reference calculated from objective id and subcategory PK
items	objective	string	Reference to the objective PK
	id	integer	Identifier, must be positive and continuous
	name	string	Displayed item rule
	topic	string	Displayed item topic (S, P or SP)
	probability	integer	Evaluation based on our risk matrix, must be between 1 and 5
	severity	integer	Evaluation based on our risk matrix, must be between 1 and 5
	risk	integer	Computed based on probability and risk, must be between 1 and 25
	requirement	string	Computed based on risk (M, S or C)
	PK	string	Unique reference calculated from item id and objective PK
	remarks	string	Facultative field, remarks about the risk assessment
descriptions	item	string	Reference to the item PK
	id	integer	Identifier, must be positive and continuous
	name	string	Displayed description title
	value	string	Open text
	link	string	Facultative field, link for an external resource
	alt	string	Facultative field, text to be shown for the link
	PK	string	Unique reference calculated from the description id and item PK

Table 4.5 Attributes of the file

4.6.3 Our approach

The last step to complete our proposal is to actually create the guide content. We went through multiple steps to translate our knowledge collection into the spreadsheet file structure, which will be explained in this Subsection. Our whole work has been conducted by applying and respecting the guidelines defined by our proposal.

First, we defined a set of categories and subcategories once the knowledge collection done, based on the various topics we saw. Then, we went through each topic one by one, and read their corresponding source one by one. We selected the most appropriate subcategory for each source, and created a new one if none of them was appropriate. Then, we assessed whether the source content can be fitted in the scope of an existing objective if we have had chosen a subcategory. If not, we defined a new objective.

The source content was then translated to fit our proposal guidelines. If the topic subject has already been processed by an item with a similar approach, we added the source content as a description. Otherwise, a new item was defined, with a separation on the source content nature between the defined item and its related description. The risk assessment was then realized once the item created.

Once our whole knowledge collection processed, we reviewed the formulations of the content we defined to ensure that our proposal guidelines were respected. Then, we reviewed the first risk assessment we made when we defined the items and added the reasons for the values we gave to the probability and severity fields in the `remarks` attribute. This last step was not done while defining the content in the first place in order to force us to think twice to produce a more accurate assessment.

Then, we reviewed the categorization. The categories we initially defined have not been changed, but some subcategories and objectives have been moved to other parents or merged if they concerned a similar subject. Some items have also been moved to a different objective if a more appropriate one has been found.

Finally, we checked the spelling and grammar of the text and corrected our mistakes.

We have chosen to parse the spreadsheet several times to ensure that we would read each part of it multiple times to find all our errors and mistakes. Although being time-consuming, this approach allowed us to produce a high quality guide content.

Regarding the spreadsheet, we created a formula to automatically compute the requirement levels based on the value of the items risks. In the same spirit, the `PK` attributes are automatically calculated based on the parent `PK` and the object `id`.

The content can not be shown in this report for visibilities reasons. However, the spreadsheet is available at Appendix B. It can also be consulted on the thesis repository [1].

4.7 Comparison with the Other Guides

Now that a guide content has been created using our proposal, we are able to compare it to the other guides we analysed in Chapter 3 (Comparison of Other Guides). Table 4.6 (Comparison of the selected guides and our proposal) shows the same table as Table 3.1 (Comparison of the selected guides based on their characteristics) with an added grey column that shows our guide characteristics.

We decided to name our proposal Guide to Assess Security and Privacy (GASP), based on our scope and research question.

Charact.	NIST	NCSC	*ENISA*	CISA	**GASP**
Accessible	✓	∼	✗	✓	✓
AI	✗	✗	✗	✗	✓
Format	list of items	tables of rules	multiple	set of actions	**checklist**
Privacy	✗	✗	∼	✗	✓
Scoring	✗	compliance levels	risk matrix	✗	✓
Volume	high	high	medium	low	**medium**
Web	✗	✗	✗	✗	✓

✓: fills the criteria; ✗: does not fill the criteria; ∼: almost fills the criteria.

Table 4.6 Comparison of the selected guides and our proposal

We can see that our guide checks all the characteristics we wanted our proposal to meet. Its allows to evaluate all the topics we wanted, using an easy format. Its accessibility will be guaranteed by developing a specific application. This step will be explained in Chapter 5 (Web Application). The volume, although not being low, is at a totally appropriate level given the complexity and the amount of content to assess. Furthermore, our guide is the only one that has a scoring capacity based on its evaluations.

We believe that our guide offers a strong added value compared to the others. We managed to design it in the simplest way possible given the complexity of the ICT security and privacy topics. However, our complete guide accessibility, ease of use and utility will be reviewed in Chapter 6 (Web Service Evaluation) using the web application.

4.8 Summary

We are very satisfied with the quality of our proposal. All the parts we defined have been challenged, well thought, and chosen appropriately with both our thesis needs and the assessors' needs. We truly believe that a guide built using such structure and content guidelines help to efficiently evaluate a system, regardless of which ICT field it comes from.

We are aware of the limitations of our risk assessment. Indeed, risk matrices are not originally designed for this kind of threats, as explained in Table 4.2 (Our defined risk matrix). However, one of the main goals of our proposal is to give priorities on which

items bring more risks in case of non-compliance: being precise would be better, but it does not significantly impact the quality of our proposal. We are also aware that using colours might not be optimal for colour-blind people. Replacing this information with another symbol could be a useful improvement.

The guide content can be improved by adding a reference to the source that has been used to create each item and description. All the content defined in the spreadsheet file comes from our previously done knowledge collection, but adding this information could be useful for the assessors. However, this information is not lost: each source has related comments in the report source files that list which items and/or descriptions have been created using it.

As explained in Section 4.2 (Medium Choice), a PDF file could be created as an additional medium to the web application. This would allow more use cases for the assessors.

5 Web Application

The term *web application* will be simplified as *application* for readability purposes.

Our proposal has been defined, and a guide content has been created using our said proposal. However, assessors still do not have an easy and accessible way of evaluating their web services if they only use the spreadsheet file we built. To this end, an application will be designed and implemented, as decided in Section 4.2 (Medium Choice).

We will start by explaining how to adapt the guide content to a readable format for the application. Then, we will analyse the needs, technologies and other important aspects before developing the application. Then, its design will be defined, followed by its implementation. Finally, technical tests will be made on all the software we developed.

Contents

5.1 Analyse

This Section will explain what our application must be able to do and which technologies will be chosen to implement it.

5.1.1 Needs and Capabilities

Our needs are the same as for the proposal: our guide must be accessible and easy to use. Furthermore, the application availability must cover most of the use cases: usable with or without an Internet connection on all the most used platforms included in our scope.

We included the possibility for assessors to save and restore their results or their evaluation progress: by doing so, we enable them to keep their evaluation data, to perform their evaluations in instalments, or to share their results to other parties without storing any data on the application side. The latter is important to enable full confidentiality on the evaluation process, and to avoid any storage costs for the hosting side.

We made the choice not to use any backend. Indeed, the maintenance and workload of a two tiers architecture are too heavy. Moreover, there is no need to provide any centralized infrastructure: indeed, our data (the guide content) is static, without any state to keep on the backend side, and does not include any data process steps.

The following list defines the capabilities of our application:

- **Compatibility**: the application must be usable on all the major desktop browsers.
- **Expandability**: the application must be able to receive updates.
- **Offline usage**: if previously downloaded, the application must be usable without any Internet connection.
- **Online availability**: the application must be reachable using the web.
- **Standalone**: the application must work without any backend.
- **Stateful capacity**: the application must be able to restore results of evaluations if desired by assessors.

Mobile platforms are not considered, as explained in Section 4.2 (Medium Choice). However, we can integrate their support through browsers if this additional part is doable in minimal time and that no major changes must be done.

5.1.2 Technologies

Multiple technologies have the ability to support the capabilities we defined. Nowadays, we have web frameworks that encapsulate lots of libraries, helpers and components that help developers to build complete and robust applications. Starting from scratch is a viable option for large, innovative and specific contexts carried out by entire teams. However, this is not necessary for more basic needs: we will therefore use one of them.

There is three major web frameworks: *Angular*[35], *React*[36] and *Vue*[37]. All three of them are comparable in terms of performances and capabilities. Wohlgethan made an analysis and comparison [151] of those three frameworks in 2018 and stated that all of them are quite similar with a few disparities. Daityari made a similar analysis [152] in a more recent context and came to a similar conclusion.

Based on our past projects and experiences, we decided to choose *Vue*: it is the most light, simple and flexible solution, although its poor code allowance. However, this aspect can be countered by well-designed components.

5.2 Design

Before implementing the application, design steps must be conducted to ensure an appropriate and well-thought implementation. To this end, we will define our application the use cases, its data management, architecture, and mock-ups of its UIs.

5.2.1 Use Cases

Figure 5.1 (The application use cases) shows the interactions between the application and its actors. This diagram explains what the application must be able to do without any implementation specifics. Relationships are also represented. As designed, the application is standalone, without any external dependencies or connections to backend servers. Furthermore, it will be used by only one type of actors, the assessors that evaluate their web services.

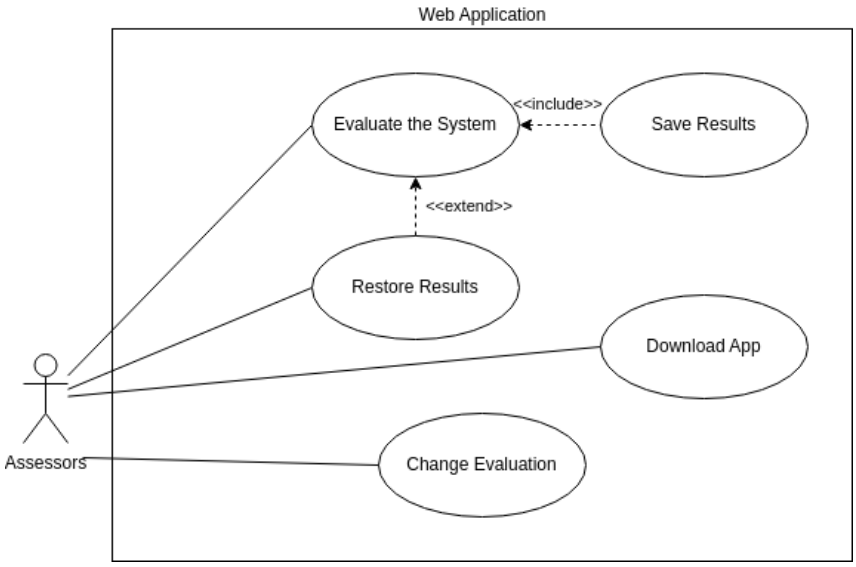


Figure 5.1 The application use cases

5.2.2 Data Management

Apart from the guide content, no additional data needs to be stored on the application side. To represent the guide content, the application will use its same structure using the TypeScript language by implementing classes, interfaces and enumerations.

[35] *Angular* source: `https://angular.io` (accessed 12th December 2022)
[36] *React* source: `https://reactjs.org` (accessed 12th December 2022)
[37] *Vue* source: `https://vuejs.org` (accessed 12th December 2022)

In its `3.x` versions, *Vue* natively supports the *Pinia*[38] module which provides application-wide storing capabilities. One or multiple data stores that contains one or multiple states, which can be seen as data, are shared across the whole application in a simple and reactive way.

Vue components, which are reusable and independent pieces composing the application, can have access to those stores by using getters and setters implemented by the developers, that directly communicate with the states values. The store implementation details will be explained through Section 5.3 (Implementation).

5.2.3 Architecture

The majority of *Vue* applications have the same, original structure as shown in Table 5.1 (Structure of a *Vue* application). By following our needs, we will use this same structure and develop our application parts into its corresponding folders.

Name	Type	Description
assets	Folder	Non-technical files needed by the application
components	Folder	Defines the independents pieces of the UI
router	Folder	Used by the *Vue Router*[39] module to declare the navigation routes through the application pages
stores	Folder	Used by the *Pinia* module to manage the application state
tests	Folder	Used to declare the unit tests
views	Folder	Used to build the displayed pages using components
App.vue	File	The entry point for the *Vue* application
main.ts	File	The entry point for the *JavaScript* environment

Table 5.1 Structure of a Vue application

Furthermore, additional information about the structure of the application can be seen on Figure 5.4 (Architecture of the application).

5.2.4 User Interfaces

Based on the use cases, we have defined the UIs that will compose the application. Their most significant mock-ups can be seen on Figure 5.2 (Mock-up of the home page) and Figure 5.3 (Mock-up of the evaluation page).

Mock-ups are defined in a way that does not consider any graphical design: they are used to represent the overall layout only. The graphical design part will be defined and done at the implementation time. Indeed, we decided that a thorough study of the UIs is not necessary because our thesis scope is not focused this concern. Furthermore, our knowledge built around past similar projects allows us to design appropriate layouts

[38] *Pinia* source: `https://pinia.vuejs.org` (accessed 12th December 2022)
[39] *Vue Router* source: `https://router.vuejs.org` (accessed 12th December 2022)

easily. Regarding the UX, we will rely on something very simple yet accessible. However, those two aspects could be the subjects of studies for a separate project once this thesis completed.

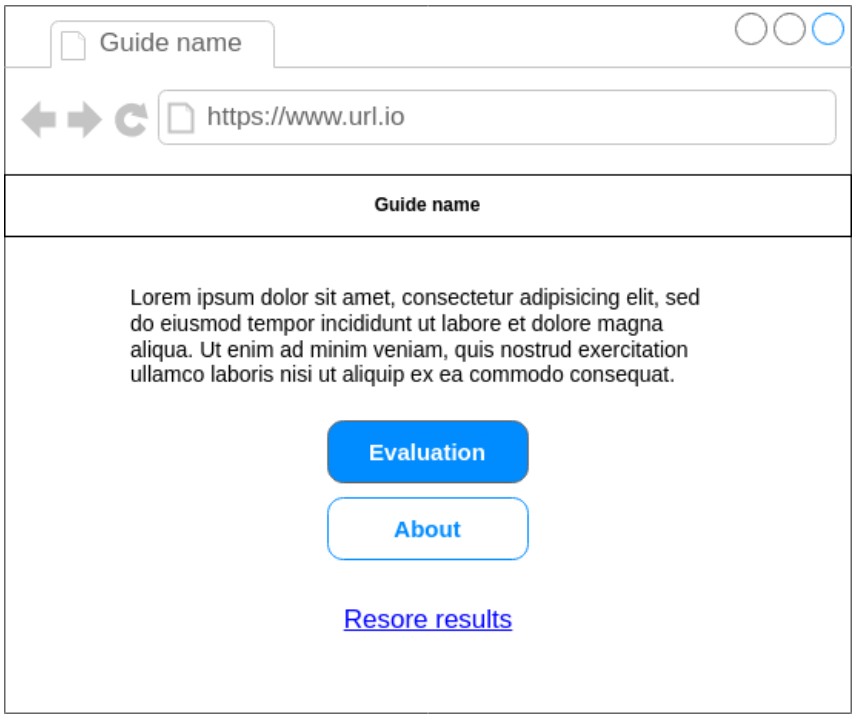


Figure 5.2 Mock-up of the home page

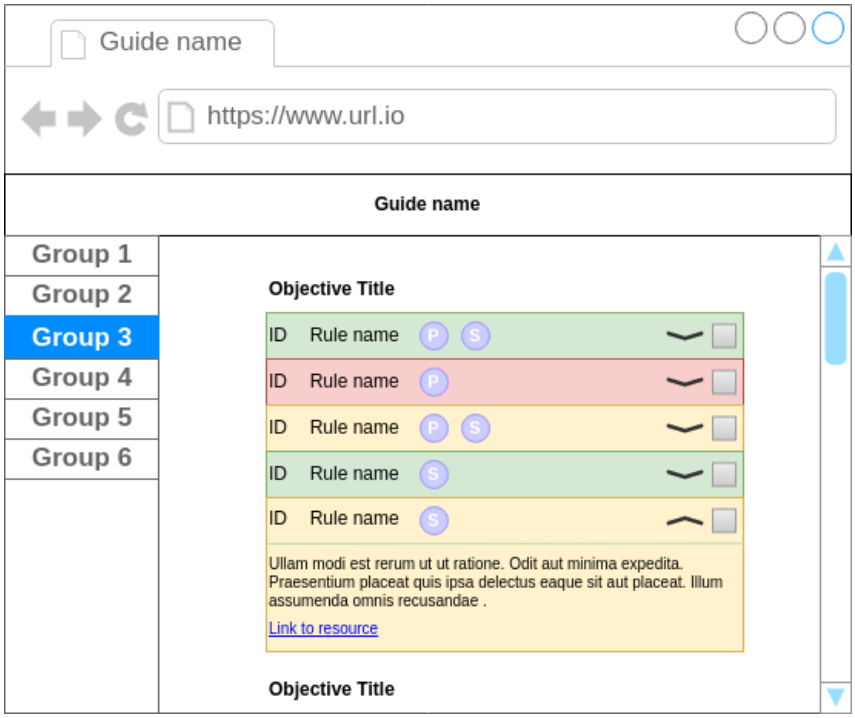


Figure 5.3 Mock-up of the evaluation page

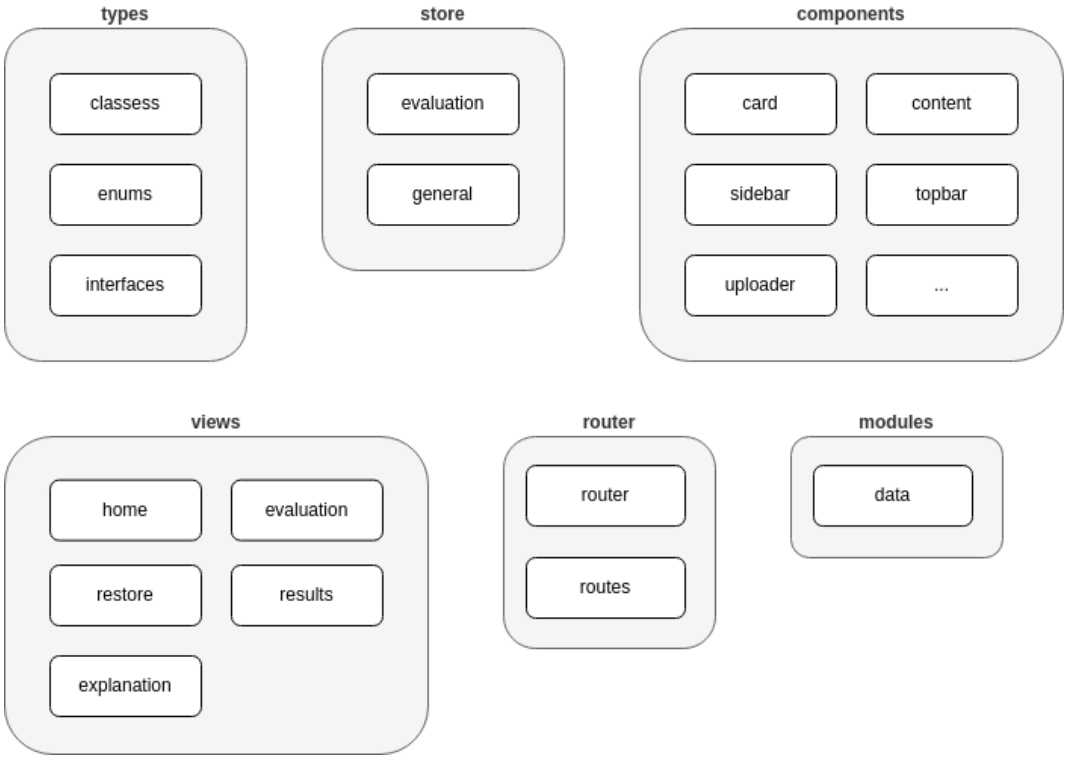


Figure 5.4 Architecture of the application

5.3 Implementation

During the whole implementation, we followed the *Vue official documentation*[40] in order to create an application that respects as much the recommendations from the development team as possible.

Each Subsection will describe the most important, interesting and relevant parts of the application implementation process. The full code is browsable and available on the project repository [1].

5.3.1 Data Conversion and Validation

As explained in Section 4.6 (Definition of the Guide Content), we decided to define our guide content in a spreadsheet file: this support can be easily read by assessors, but is less adapted for systems. However, spreadsheet files can use the *CSV* format which is a great support to transfer data between two different and independent systems. In the other hand, we wanted to design and implement data validation and integrity checks on the guide content to avoid invalid values in its definition. If those checks are performed, we can take advantage of the parsing steps through the spreadsheet to export the data in another format more suitable for applications.

One of the most used data format in the web environment is the JSON format: it has been initially designed for *JavaScript* applications and is based on a key-value pattern.

[40] *Vue official documentation* source: `https://vuejs.org/guide` (accessed 9th December 2022)

We created a *Python* script for this validation and conversion step, which is one of the most convenient and compatible scripting methods for all the major environments and OSes.

Script Environment

First, we used a library named *read-ods-with-odfpy*[41] to read the data from the *ODS* file, with built-in functions to parse its data. The library can be installed using the following command:

```
git clone https://github.com/marcoconti83/read-ods-with-odfpy ODSReader
```

Pipenv[42] is used as the package manager for our script. It allows developers to create and manage *Python* virtual environments, and to add or remove packages from a *Pipfile*. This *Pipfile* lists all the needed packages and the appropriate *Python* version in order to create a virtual environment as defined by the developers. Using this file, a new virtual environment can be initialized by running this following command:

```
pipenv install
```

Once the environment created, it must be activated in the current terminal using the corresponding `shell` command. The script can be then launched inside the virtual environment by running the `convert.py` file.

```
pipenv shell
```

```
python ./convert.py -i [input file] -o [output file]
```

The version `3.11` of *Python* has been used, which is the latest stable release at the implementation time.

Script Features

Here is a list of what the script must be able to do:

- **Data conversion**: transfer data from raw data to a valid JSON file
- **Data verification**: verify whether each value is valid based on its type
- **Error management**: handle the errors appropriately, and also show the error source
- **Argument management**: support command arguments to specify the input and output files

Script Development

The script is designed in two files: one for the classes (`classes.py`) and one for the main script (`convert.py`). The first is imported into the second in order to lighten the scripting part by declaring classes in another file.

[41] *read-ods-with-odfpy* source: https://bit.ly/3C835qu (accessed 29th December 2022)
[42] *Pipenv* source: https://bit.ly/3PYG7YC (accessed 29th December 2022)

An extract of the `classes.py` file is shown in Listing 5.1 (Extract of the data converter classes). It contains two *Python* `Exceptions` that we implemented, which allows us to manage errors with more granularity. One of those is shown from lines 1 to 2. The class file also defines three `Enumerations` that allows us to identify attributes and cell types more easily. One of those `Enumerations` can be seen from lines 5 to 10.

```
1  class ArgumentException(Exception):
2                  Exception.args
3
4
5  class SHEETS(Enum):
6          categories = "categories"
7          subcategories = "subcategories"
8          objectives = "objectives"
9          items = "items"
10         descriptions = "descriptions"
```

Listing 5.1 Extract of the data converter classes

The script is launched using an entry point, as shown in Listing 5.2 (The script entry point). The arguments are given to the `main` function using the `sys.argv` object, where they are then processed.

```
301 # Run the main function of the script, with the arguments
302 if __name__ == '__main__':
303     sys.exit(main(sys.argv))
```

Listing 5.2 The script entry point

The main function of the script is then called. All operations are wrapped into `try` and `except` blocks to allow us to catch the errors during the execution. Each type of error is processed adequately at the end of the main scripting part, as shown in Listing 5.3 (The data converter error processing). Additional information is displayed in the terminal according to the nature of the errors.

```
283 # Error management, with additional print of each Exception type
284 except ArgumentException as msg:
285     print("An error occurred when reading the arguments.")
286     print(msg)
287     print(arg_help)
288 except OSError as msg:
289     print("Problem while reading the ods file, or creating the json file, or while
    ↪  creating the JSON file.")
290     print(msg)
291 except getopt.GetoptError:
292     print("One specified argument is not valid. Please check the correct format below.")
293     print(arg_help)
294 except Exception as msg:
295     print("An unknown problem occurred.")
296     print(msg)
297 finally:
298     quit()
```

Listing 5.3 The data converter error processing

The first process of the script is to parse and verify the given arguments, as shown in Listing 5.4 (The data converter argument management). We have referred to the *OpenSourceOptions*[43] website. A help message is defined if needed, as well as a verification step to check whether each argument has been given. The `-input` or `-i` argument gives the *ODS* file to be read, and the `-output` or `-o` argument gives the name of the JSON file to be generated.

```
241 # Argument management
242 # {source link}
243 arg_input = ""
244 arg_output = ""
245 arg_help = "{0} -i <input> -o <output>".format(argv[0])
246
247 opts, args = getopt.getopt(argv[1:], "hi:u:o:", ["help", "input=", "output="])
248
249 for opt, arg in opts:
250     if opt in ("-h", "--help"):
251         # prints the help message and quit
252         print(arg_help)
253         sys.exit(2)
254     elif opt in ("-i", "--input"):
255         arg_input = arg
256     elif opt in ("-o", "--output"):
257         arg_output = arg
258
259 # making sure that each argument is not empty
260 if not arg_input:
261     raise ArgumentException("The input argument is empty.")
262 if not arg_output:
263     raise ArgumentException("The output argument is empty.")
```

Listing 5.4 The data converter argument management

Then, some operations are done to prepare the data conversion and verification: each spreadsheet tab content will be stored into its corresponding key in a *Python* `dictionary`.

Each tab passes through a parsing of its values, as shown in Listing 5.5 (The data converter part which reads the spreadsheet tabs). The attributes can be defined by reading the first row of the tab, as shown on line 208. Then, a dictionary is created, which will store each record found. From line 216 to 219, situations where the last attribute value is empty are handled. Afterwards, we parse through all the records and all their attributes to read their values, which are sent to the `integrity` function.

The `integrity` function tests whether the value it receives is valid given the tab and the attribute it comes from. If the value passes the tests, it is either returned as it or returned converted to another type if needed. If the value does not pass a single test, an `Exception` is raised and a verbose error message is shown in the terminal.

Finally, a hash is computed on the whole guide content using the `SHA512` hash function, and is then stored into the `dictionary` into the `hash` key, alongside the guide content. This `dictionary` is exported in the JSON format at the end of the script.

[43] *OpenSourceOptions* source: https://bit.ly/3WRYmBC (accessed 2nd January 2023)

```
202 def readSheet(ods, activeSheet, output):
203     sheetName = activeSheet.value
204     # Select the sheet
205     arrays = ods.getSheet(sheetName)
206     print("Getting the " + sheetName + " sheet...")
207     # Extracting attributes from data (first column)
208     attributes = arrays[0]
209     # Creating variable to receive data by their index and attributes
210     dictionary = [dict() for i in range(len(arrays) - 1)]
211     # Iterate on each record
212     for i, row in enumerate(arrays[1:]):
213         # Iterate on each attribute
214         for attribute in range(len(attributes)):
215             # Avoid error if last column of row is empty
216             if attribute < len(row):
217                 cell = row[attribute]
218             else:
219                 cell = ""
220 
221             try:
222                 # Run the integrity test
223                 verified = integrity(activeSheet, attributes[attribute], cell)
224                 if verified:
225                     # store value if inside the guide content scope
226                     dictionary[i][attributes[attribute]] = verified
227             except CellException as msg:
228                 print("Error in sheet " + sheetName + " for attribute " +
                    ↪  attributes[attribute] + ", value '" + str(cell) + "' is invalid! (line
                    ↪  " + str(i + 2) + ")")
229                 print(msg)
230                 quit()
231 
232     # Save results
233     output[sheetName] = dictionary
234     print(sheetName + " extracted.")
235     print("------------------------------")
```

Listing 5.5 The data converter part which reads the spreadsheet tabs

5.3.2 Application Basics

Some setups and patterns have been defined when the application has been initialized. This Subsection describes those parts only, and the more specific processes will be presented by features in further Subsections.

Project Creation

We used the official Command Line Interface (CLI) command of Vue to create the basic environment. This creation method is based on the *Vite*[44] frontend tooling. *Vite* allows to improve the development process by optimizing various parts of the *JavaScript* ecosystem. The official installer asks whether it should install other components (modules) to support specific development functionalities. We installed the following components:

- **TypeScript support**: to include typing support and new capabilities on top of *JavaScript*
- ***JSX***: to enable structure component rendering

[44] *Vite* source: https://vitejs.dev (accessed 24th January 2023)

- ***Vue Router***: to enable a single-page application structure with built-in navigation
- ***Pinia***: to manage application-wide data stores
- ***Vitest***: to perform unit tests
- ***ESLint***[45]: to detect code errors by static code analysis
- ***Prettier***[46]: a code formatter

The *Cypress*[47] component could have been used as well. It is an end-to-end testing solution for web applications. We did not install it because of the sufficient coverage provided by *Vitest* for testing capacities.

As shown in Table 5.1 (Structure of a *Vue* application), our application is contained in various folders. We added two new folders named `modules` and `types`. The first one is used for our customs modules. We only created one module, `dataHandler`, which uses the global store to manage the guide content. The second is used to define the TypeScript classes, enumerations and interfaces. Elements from both folders will be described in further Subsections.

Data Storage

The data storage is implemented using the *Pinia* module, which enables application-wide data stores definition and usage. It is defined into the `store` folder.

The module must be registered in the main definition of the *Vue* application made in the `main.ts` file. Afterwards, one or multiple stores can be defined and can be made available for all the application components.

Usually, one store is defined for each data origin: remote data can be handled in one store, application state in another, user data in another, et cetera. Our case includes only one origin, which is the one related to the guide. However, another store will be defined to provide data regarding the general state of the application.

Splitting data into multiple stores does not change the general behaviour or performances of the application, but it brings more clarity in the code, especially for large applications.

A store consists of three components:

- **The state**, structure that contains the store data using key-values pairs.
- **The getters**, set of functions that use the state to provide information and values about it.
- **The actions**, set of functions that alter the state with new data, bring changes on it, or remove either some parts or all of it.

[45] *ESLint* source: `https://eslint.org` (accessed 8th February 2023)
[46] *Prettier* source: `https://prettier.io` (accessed 8th February 2023)
[47] *Cypress* source: `https://www.cypress.io` (accessed 8th February 2023)

Only using getters and actions from *Pinia* allows the developers to implement safe, reactive and non-blocking operations on data. Those functions can then be used in *Vue* components, and even in regular TypeScript modules.

Examples given for the application data state come from the store that contains the guide content. The other store managing the global application state follows the same structure and implementation details.

Listing 5.6 (State definition) shows what the state contains. Each attribute can be seen as a standalone typed data, that is then used by getters and actions.

```
20 state: () => ({
21         hash: "" as String,
22
23         guide: new Map as Map<number, Category>,
24
25         subcategoriesCounter: new Map as Map<string, NumberedCounter>,
26
27         statistics: undefined as ObjectsCounter | undefined
28 }),
```

Listing 5.6 State definition

Getters and actions must be defined as functions in their own attribute, that is then passed to the *Pinia* store controller that creates the store itself. An example of two getters is given in Listing 5.7 (An example of two store getters).

```
30 getters: {
31         getStatus: (state) => {
32                 return state.guide.size !== 0
33         },
34
35         getAllCategories: (state): Category[] => {
36                 let categories = [] as Category[]
37                 state.guide.forEach((category) => {
38                         categories.push(category)
39                 })
40
41                 return (categories)
42         },
```

Listing 5.7 An example of two store getters

Modules

JavaScript or TypeScript modules are useful to create new features or capabilities to a piece of software that are independent of the application components, or used by several of them. They are defined into the `modules` folder of the application, and are then imported by the components that need to use them.

Our use case only needs one module, which is the data management part. Indeed, the *Pinia* store should only define getters and setters that directly manipulate the data it manages. Any additional steps that have no direct impact on the state should be defined in the other parts of the applications.

This module has three capabilities, each implemented as functions. They are then exported to make them callable by other components, except for a local one that is used several times. Those capacities are:

- **initializeData**: handles the guide content verification before importing it into the store.
- **restoreResultsFromFile**: reads a JSON file given by assessors to restore their results or their evaluation progress.
- **isIdValid**: local function used by the `initializeData` function to ensure that given identifiers are valid.

The `initializeData` function can be used in two different ways: if no argument is given, the local guide content contained in a JSON file is used. Otherwise, custom guide content given in a string are used. This allows assessors to use their own guide contents directly into the application.

Pages

A set of pages has been defined based on our features and needs. They all follow the same layout thanks to the components we defined, except for the `Evaluation` one that includes a `Sidebar` to enable subcategories navigation. Table 5.2 (Application pages) shows the complete list of them.

Name	Layout	Usage
HomeView	One page	Displays links to either start an evaluation or to restore results and progress from a file. It also contains further information about the project, allows to download the report and explain how to install the application.
ExplanationView	One page	Displays some information about how the guide works before accessing to it. Also allows assessors to change the guide content using a JSON file.
EvaluationView	`Sidebar`	Displays the objectives, items and descriptions of the subcategory selected by the assessors.
ResultsView	One page	Displays the global score and the category ones. It also allows to download the results and to browse all the non-compliant items.
RestoreView	One page	Allows assessors to upload a JSON file in order to restore their results or their progress.

Table 5.2 Application pages

Each page, also called view, is a collection of components. The pages themselves are components. They are all defined into the `views` folder of the application.

Navigation

The application navigation is managed by the *Router* module. It allows defining paths between multiple routes that are displayed into a single basic page, just by changing its content. The navigation is defined into the `router` folder of the application.

A `Router` instance must be created using a dedicated function, and its routes are then specified. A route must lead to a component, define a path, and can also have other specificities. Listing 5.8 (Extract of our application Router) shows an extract of some different routes we defined:

- Lines 4 to 8: a basic route
- Lines 9 to 13: a redirection to another route
- Lines 14 to 18: a route with a dynamic argument named `subcategory`
- Lines 19 to 28: a route with an access control, defined by the `beforeEnter` guard

```
1  const router = createRouter({
2          history: createWebHistory(),
3          routes: [
4                  {
5                          path: "/restore/",
6                          name: "Restore",
7                          component: RestoreView,
8                  },
9                  {
10                         path: "/evaluation/",
11                         name: "Evaluation Landpage",
12                         redirect: { path: '/evaluation/1.1' }
13                 },
14                 {
15                         path: "/evaluation/:subcategory",
16                         name: "Evaluation",
17                         component: EvaluationView,
18                 },
19                 {
20                         path: "/results",
21                         name: "Results",
22                         component: ResultsView,
23                         beforeEnter: (_to, _from, next) => {
24                                 const evaluationStore = useEvaluationStore();
25                                 if (!evaluationStore.isEvaluationComplete) return
   ↪  next('/evaluation');
26                                 else next()
27                         }
28                 },
29         ],
30 });
```

Listing 5.8 Extract of our application Router

Types

One major addition from TypeScript to the *JavaScript* language is the usage of types. It allows developers to work with typed data and attributes to allow more robust code and an improved error and exception management. We defined them into the `types` folder of the application.

We defined three sorts of types, depending on our needs:

- **Enumerations**, for basic enumerations to bring a better code readability
- **Interfaces**, to specify which attributes must be included in *JavaScript* objects
- **Classes**, to create instances of a class by using the object-oriented pattern

Documentation

The code documentation has been made with the *JSDoc*[48] markup language, which is the most used of its type for *JavaScript* and TypeScript source files. It allows a standardized and heterogeneous documentation through the application. through comments.

All comments are not shown on some Listings for for readability reasons.

5.3.3 Store Guide Data

As explained before in Subsection 5.3.2 (Application Basics), a *Pinia* store has been defined using TypeScript types and our `dataHandler` module. Their interactions and processes will be detailed here.

Types

Two different types have been used for this specific step: interfaces when data is loaded from a JSON file, and classes for their stored version. We made this choice because stored data must have a hierarchical structure that should allow us to store objects in their corresponding parents, and we need to instantiate this structure. This hierarchic approach allows us to optimize the amount of operations and the access time when retrieving stored objects, and to build a coherent structure.

Listing 5.9 (An example of a TypeScript interface) and Listing 5.10 (An example of a TypeScript class) show an example with the `Category` objects. First, the interface defines which attributes must be given by the JSON file, which acts as an integrity check on the attributes type and existence then importing the file. Then, the class implements the interface which means that all attributes must be existing as well, and a constructor instantiates an empty `Map` object for the object related children. The `Maps` use each child `id` as key and is corresponding class as a value. The same approach is used for the categories stored in the store, as shown in Listing 5.6 (State definition) on line 23.

```
1  interface CategoryData {
2          id: number;
3          name: string;
4  }
5
6  export type { CategoryData };
```

Listing 5.9 An example of a TypeScript interface

[48] *JSDoc* source: https://jsdoc.app (accessed 2nd January 2023)

```
1  import type { CategoryData } from "../interfaces/CategoryData";
2  import type { Subcategory } from "./SubcategoryObject";
3
4  class Category implements CategoryData {
5          id: number;
6          name: string;
7          subcategories: Map <number, Subcategory>
8
9          constructor(categoryData: CategoryData) {
10                 this.id = categoryData.id;
11                 this.name = categoryData.name;
12                 this.subcategories = new Map<number, Subcategory>();
13         }
14 }
15
16 export { Category };
```

Listing 5.10 An example of a TypeScript class

Data Verification

This part is done by the `dataHandler` module: it starts by importing a `string` representing a JSON file of the guide content into the corresponding TypeScript interfaces. This first step acts like a verification on each attribute, whether the value is null or not and if the type is correct.

Afterwards, we browse the guide content as shown in Listing 5.11 (Two iterations made by the `dataHandler` module when loading a guide content). We verify that at least one category is defined, and then we iterate on each category, as shown on line 56.

Then, each `id` is verified, and we use a custom integrity structure named `verification` to check the unicity of the object. If another object with the same `id` has already been defined, we cancel the process. Otherwise, we instantiate the integrity structure with another one in a recursive approach, for the further integrity check on the object children. This integrity process is shown on lines 61 to 67 of Listing 5.11 (Two iterations made by the `dataHandler` module when loading a guide content).

If the two tests have been passed, we search for all the children of the object within the JSON `string`, as shown on line 73. We then check whether at least one child exists: if so, the same process is started again with the related child or children.

Every child is then stored into an instance of the `Category` class, which is then sent to the store when all children have been browsed and defined. The counter is also incremented to keep a track of the amount of objects. An example of this process is shown from line 69 to 71 of Listing 5.11 (Two iterations made by the `dataHandler` module when loading a guide content), and another one is shown on Listing 5.12 (An item being saved into a `Category` instance).

If every object has been successfully verified, the guide content is sent to the store, as well as its hash.

```
53  try {
54          if (!categoriesData.length) throw ("No categories are defined.")
55
56          categoriesData.forEach((category) => {
57
58                  if (isIdValid(category.id, categoriesData.length)) {
59                          throw ("Category's ID " + category.id.toString() + " is not
    ↪  valid.")
60                  }
61                  if (verification[category.id]) {
62                          throw ("Description " + category.id + " is not unique.")
63                  }
64                  verification[category.id] = {
65                          "hasItem": true,
66                          "children": [] as Integrity[]
67                  }
68
69                  let newCategory = new Category(category)
70
71                  objectsCounter.categories++
72
73                  let relatedSubcategories = subcategoriesData.filter(record =>
    ↪  record.category === category.id) as SubcategoryData[]
74                  if (!relatedSubcategories.length) throw ("No subcategory is linked to
    ↪  category " + category.id + ".")
75
76                  relatedSubcategories.forEach((subcategory)  => {
```

Listing 5.11 Two iterations made by the `dataHandler` module when loading a guide content

```
1  newCategory.subcategories.get(subcategory.id)?.objectives
2  .get(objective.id)?.items.set(item.id, new Item(item))
```

Listing 5.12 An item being saved into a `Category` instance

5.3.4 Evaluation Progress

The assessors progress in the evaluation process must always be known: it allows us them to know their current progress, but also to avoid any access to the `ResultView` page when the evaluation is not complete. To do so, several getters have been defined into the store: Listing 5.13 (The evaluation progress computed by the store) shows them.

The line 25 of Listing 5.6 (State definition) shows a TypeScript `Map` that acts like a counter for the evaluation progress. The keys of the `Map` refer to *ids* of each subcategory. The `NumberedCounter` class is used for this need. It has two attributes: `count` is used for the number of item being evaluated, and `amount` for the total amount of items into a subcategory. This class can be seen on Listing 5.14 (The `NumberedCounter` class).

When the guide content is imported, all items are browsed depending on their subcategory and their total amount within each subcategory is calculated using the `NumberedCounter` class. This process can be seen on Listing 5.15 (Definition of the counter for each subcategory), were an instance of `NumberedCounter` is created if an item refers to a subcategory for the first time.

```
129 isCategoryComplete: (state) => (categoryPK: number): boolean  => {
130
131         let subcategories = [...state.subcategoriesCounter].filter( ([key] ) => {
132                 let [catId] = new Indexes(key).getSubcategoryIndexes()
133                 return categoryPK == catId
134         })
135
136         // deal with empty categories (should not happen)
137         if (!subcategories.length) return true
138
139         if (subcategories[0]) {
140                 let overallCount: number = 0;
141                 let overallAmout: number = 0;
142                 subcategories.forEach(map => {
143                         overallCount += map[1].count
144                         overallAmout += map[1].amount
145                 });
146                 return overallAmout === overallCount
147         }
148
149         return false
150 },
151 isSubcategoryComplete: (state) => (subcategoryPK: string): boolean => {
152         let subcategoriesStatus = state.subcategoriesCounter.get(subcategoryPK)
153         if (subcategoriesStatus)
154                 return subcategoriesStatus.count === subcategoriesStatus.amount
155         else return false
156 },
157 isEvaluationComplete(state): boolean {
158         let overallCount: number = 0;
159         let overallAmout: number = 0;
160         state.subcategoriesCounter.forEach(map => {
161                 overallCount += map.count
162                 overallAmout += map.amount
163         });
164         return overallAmout === overallCount
165 },
```

Listing 5.13 The evaluation progress computed by the store

Afterwards, the corresponding `NumberedCounter` is increased every time that an item is evaluated by the assessors. This is handled by the `setCheckbox` action of the store, visible on Listing 5.16 (The store action called when an item is evaluated). The `Item` class has a method named `changeEvaluation` that changes the value of the item evaluation, based on the corresponding `Enumeration`.

Lines 231 to 238 of Listing 5.16 (The store action called when an item is evaluated) show how the tracking of the subcategory count is realized: if the item was previously not evaluated, which can be determined whether its old value was `undefined`, we retrieve the parent subcategory of the item and increase the appropriate `NumberedCounter`. Since an item evaluation value can only be `undefined` when unchecked, there is no need to change the counter for other evaluation values.

The instances of `NumberedCounter` can be used to compute whether subcategories, categories or the whole evaluation are completed. Listing 5.13 (The evaluation progress computed by the store) shows the corresponding store getters.

```
class NumberedCounter {
        amount: number;
        count: number;

        constructor(countOne: boolean) {
                if (countOne) this.amount = 1;
                else this.amount = 0
                this.count = 0;
        }

        increaseCount(): void {
                if (this.count < this.amount)
                        this.count = this.count + 1;
        }
        increaseAmount(): void {
                this.amount = this.amount+1;
        }

        setAmount(newAmount: number): void {
                this.amount = newAmount;
        }

        getAmount(): number {
                return this.amount;
        }
        getCount(): number {
                return this.count;
        }
}

export { NumberedCounter };
```

Listing 5.14 The `NumberedCounter` class

```
populateCategories(data: Category) {
        this.guide.set(data.id, data)

        // Init item evaluation index
        this.guide.get(data.id)?.subcategories
        .forEach((subcategory) => {
                subcategory.objectives.forEach((objective) => {
                        objective.items.forEach( () => {
                                let oldValue =
↪  this.subcategoriesCounter.get(subcategory.PK)
                                if (oldValue) {
                                        oldValue.increaseAmount();
                                }
                                else
                                        this.subcategoriesCounter.set(subcategory.PK, new
↪  NumberedCounter(true))
                                        });
                                })
        })
        return true
},
```

Listing 5.15 Definition of the counter for each subcategory

A subcategory progress status can be easily computed: we only need to retrieve the corresponding `NumberedCounter` instance and test whether its two attributes have the same value.

```
221 setCheckbox(
222 newStatus: Evaluation,
223 oldStatus: (Evaluation | undefined),
224 itemPK: string) {
225         let [catId, subId, objId, itemId] = new Indexes(itemPK).getItemIndexes()
226
227         this.guide.get(catId)?.subcategories
228         .get(subId)?.objectives
229         .get(objId)?.items
230         .get(itemId)?.changeEvaluation(newStatus)
231         if (!oldStatus) {
232                 let [catId, subId] = new Indexes(itemPK).getItemIndexes()
233                 let subcategory = catId.toString() + "." + subId.toString();
234                 let oldValue = this.subcategoriesCounter.get(subcategory)
235                 if (oldValue) {
236                 oldValue.increaseCount()
237                 }
238         }
239 },
```

Listing 5.16 The store action called when an item is evaluated

A category progress status could have use the subcategory getter and iterate on it, but a limitation on *Pinia* avoids us to do so. Instead, we parse all the `NumberedCounter` and select the ones that concern our category. Then, we compare the two attributes on all the `NumberedCounter` instances as for the subcategory verification.

The entire evaluation progress status is computed using the same approach as for the categories one, but using all the `NumberedCounter` instances.

The three store functions are watched during the whole evaluation process by multiple components, and are updated every time that an item is being evaluated for the first time only. For this reason, we implemented the `NumberedCounter Map` instead of assessing all items from the entire guide content to optimize the application workload.

5.3.5 Compute and Show Results

Two getters have been developed to retrieve the scores, as shown in Listing 5.17 (The store getters to compute scores). They behave almost in the same way: they both select the set of items that must be used to compute a score, which could be all of them from the guide or the ones coming from a specific category. Every related item is then stored in a variable, which is sent to the `computeScore` intern function shown in Listing 5.18 (The score computation) that computes the score. The approach was chosen to avoid code repetition between the two getters. The details of the score computation is explained in Subsection 4.5.2 (Calculations).

For the numerator part, each item the evaluated web service is not compliant with will have a value of zero, and the others items will give a value equals to their risks level. This is done using a `map-reduce` approach.

Almost the same operation is done for the denominator, which needs all the items risk levels. Then, we use the `Percentage` class that we created to express the score.

```
167 getCategoryScore: (state) => (categoryId: number): Percentage => {
168
169         let items = [] as Item[]
170         state.guide.get(categoryId)?.subcategories
171         .forEach((subcategory) => {
172                 subcategory.objectives?.forEach((objective) => {
173                         objective.items?.forEach( (item) => {
174                                 items.push(item)
175                         })
176                 })
177         })
178
179         return computeScore(items)
180 },
181
182 getOverallScore: (state): Percentage => {
183         let items = [] as Item[]
184         state.guide.forEach ( (category) => {
185                 category.subcategories.forEach((subcategory) => {
186                         subcategory.objectives?.forEach((objective) => {
187                                 objective.items?.forEach((item) => {
188                                         items.push(item)
189                                 })
190                         })
191                 })
192         })
193         return computeScore(items)
194 },
```

Listing 5.17 The store getters to compute scores

```
243 function computeScore(items: Item[]): Percentage {
244         let numerator = items.map((item) => {
245                 switch (item.evaluation) {
246                 case Evaluation.checked:
247                 case Evaluation.unrelated:
248                         return item.risk
249                 default:
250                         return 0
251                 }
252         })
253         .reduce((a, b) => a + b, 0);
254
255         let denominator = items.map((item) => {
256                 return item.risk
257         })
258         .reduce((a, b) => a + b, 0);
259
260         return new Percentage(numerator, denominator)
261 }
```

Listing 5.18 The score computation

5.3.6 Save and Restore Results

The results of a completed evaluation can be saved on the `ResultsView` page. To do so, a download button can be clicked, and a JSON file is downloaded using the function shown on Listing 5.19 (Save the results).

This function retrieves all the guide items of the store and only keep their `PK` and `evaluation` attributes in order to avoid large files exports. This step can be seen from lines 2 to 4, with the data being saved into an array using the `StoredItems` interface. Then, we build an object using the `StoredResults` interface that will store both the items and the guide hash. Finally, the data are transformed into a `string` value following the JSON format and made available to download.

The two interfaces we used have been defined in order to standardize the save and restore process: the payloads we create and read at those two steps can then be verified when exported and imported.

```
1  download() {
2          let itemData = this.getAllItems.map((item) => {
3                  return (({ PK, evaluation }) => ({ PK, evaluation }))(item);
4          }) as StoredItems[]
5
6          let storedItems = {
7                  "checkHash": this.getHash,
8                  "items": itemData
9          } as StoredResults
10
11         const jsonData = encodeURIComponent(JSON.stringify(storedItems))
12         this.myUrl = `data:text/plain;charset=utf-8,${jsonData}`
13         this.myFilename = 'exportData.json'
14 },
```

Listing 5.19 Save the results

The restoration process of the results is handled by the `dataHandler` module, using the `restoreResultsFromFile` function. This function is shown on Listing 5.20 (The `restoreResultsFromFile` function of `dataHandler`).

The first thing the function does is to verify that the store holds a valid guide content. If so, the results are parsed and stored into a `StoredResult` interface. The structure is afterwards verified to ensure that the JSON structure is valid. Then, the hash of the guide content used to generate the results is compared to the one that is stored into the application: if they are the same, the results have been obtained using the same guide content that the one that is currently stored, and the results are restored. To do so, we used the `setCheckbox` store function shown in Listing 5.16 (The store action called when an item is evaluated) to restore the evaluation values of each item.

5.3.7 Save and Restore Progresses

The application has the ability to save and to restore the progress of an evaluation. The well-design structure of the application and the generic functions we created to handle the data allow this feature to use the same logic as for the feature to save and restore results, described in Subsection 5.3.6 (Save and Restore Results).

A button is displayed in the `Sidebar` on the `EvaluationView` page, which allows assessors to download a JSON file to save their progress.

```
188 export function restoreResultsFromFile(oldResults: string) {
189
190         const evaluationStore = useEvaluationStore();
191
192         return new Promise((resolve, reject) => {
193                 if (!evaluationStore.getStatus) return reject("The stored evaluation items
    ↪  are invalid.")
194
195                 const oldResultsJson = JSON.parse(oldResults) as StoredResults
196                 if (!oldResultsJson)
197                         return reject("Your file is not a JSON file")
198                 if (!oldResultsJson.checkHash || !oldResultsJson.items ||
    ↪  !oldResultsJson.items[0].PK)
199                         return reject("Inadapted JSON structure")
200
201                 let storedHashes = evaluationStore.getHash
202                 if (storedHashes !== oldResultsJson.checkHash)
203                         reject("Verification of hashes failed. Your results have been
    ↪  generated with another guide content.")
204
205                 try {
206                         oldResultsJson.items.forEach((newItem) => {
207                                 if (newItem.evaluation !== undefined)
    ↪  evaluationStore.setCheckbox(newItem.evaluation, undefined, newItem.PK)
208                         })
209                 } catch (error) {
210                         console.log(error)
211                         reject(error)
212                 }
213
214                 resolve(true)
215         })
216 }
```

Listing 5.20 The `restoreResultsFromFile` function of `dataHandler`

5.3.8 Use Store Data

The store getters are used by components to display data in their content. To this end, a component must import them from the store before using them. An example of this method will be presented using the `EvaluationView` component.

First, the store must be imported by the component and its getters made available, as shown on Listing 5.21 (The `setup` part of the `EvaluationView` component). This part is done in the `setup` section of components, which is where links to resources, methods, data or other *JavaScript* processes are declared. Line 57 shows the reference to the store, line 59 shows the getters imports from the store and lines 61 to 65 show how to make them available for the whole component.

Listing 5.22 (Extract of the `template` part of the `EvaluationView` component) shows how to use the data imported from the store using the getters. Data retrieved by getters must be placed between double curly brackets to show its computed value in the component content. The `v-for` attribute on line 11 shows how to generate components based on a list of objects. In this example, we use the `Objective` component that requests two parameters, the objective name and its PK, to build the view.

```
55  setup() {
56
57          const evaluationStore = useEvaluationStore();
58
59          const { getObjectivesFromSubcategory, getStatus, getSubcategory } =
    ↪  storeToRefs(evaluationStore)
60
61          return {
62                  getObjectivesFromSubcategory,
63                  getStatus,
64                  getSubcategory
65          };
66
67  },
```

Listing 5.21 *The `setup` part of the `EvaluationView` component*

```
5   <Content>
6           <h1> {{ active }} - {{ getSubcategory(active)?.name }}</h1>
7
8           <p> {{ getSubcategory(active)?.description }}</p>
9
10          <Objective
11          v-for="objective in getObjectivesFromSubcategory($route.params.subcategory as
    ↪  string)"
12          :objectiveName="objective.name"
13          :objectivePK="objective.PK"/>
14
15  </Content>
```

Listing 5.22 *Extract of the `template` part of the `EvaluationView` component*

5.3.9 Progressive Web Application

The application has been made compatible with the *Progressive Web Application (PWA) standard*[49], which enables web applications to be installed into browsers. It allows website users to open those applications natively on their device using their browser. If the device has no access to the Internet, those applications can still be accessed.

An application compatible with the PWA standard must include various criteria to be fulfilled by an application in order to enable it:

- Have a valid **Web App Manifest**
- Serve the application using the **HTTPS** protocol
- **Redirect** the HTTP traffic to HTTPS
- Have a **robots.txt** file for search engines crawling
- Have **adapted images** for favicons
- Declare mandatory link to resources in the webpages **head** sections

[49] *PWA standard* source: `http://bit.ly/3DQruBU` (accessed 8th February 2023)

We made our application compatible and PWA-ready by using the *PWA Vite Plugin*[50] module. We followed its documentation and applied the mandatory configurations in order to be compliant to the standard.

The application can be installed on all *Chrome*-based browsers, on *Firefox* Mobile and *Safari*. Unfortunately, *Firefox* desktop does not support this standard, except by using the unofficial *Progressive Web Apps for Firefox*[51] add-on.

Other methods could have been used to make our application available natively, such as using *Electron*[52]. Give the limited usage of our tool in native mode, which will only be useful for offline evaluations, we selected the easiest and fastest approach.

5.3.10 Data Restoration Improvement

While working on the application, we realized that our guide could become unusable in a specific use case: if an assessor saves their guide results or progress before an update of the guide content and then tries to restore their data afterwards, the application would detect that their data have been generated using an old version of the guide and would block the restoration process. This behaviour is doing exactly what we defined to avoid misleading evaluations, but then the assessor will not have the possibility to load their data.

In order to prevent this issue, we modified some parts or our application.

First, we modified the `StoredResults` interface, which is used as a support to store the data to be included in the downloaded JSON file, to integrate the guide content as well. This guide content also uses an interface, named `GuideData`, that represents the JSON object that is generated by our *Python* script and imported into our application at its startup.

As shown in Listing 5.23 (New store getter to get the guide content), the guide content is generated using a store getter named *getWholeGuide* that converts the data from the store under its TypeScript *Map* form into arrays for each guide content object. The hash is also stored in the `GuideData` interface and the latter is returned to the caller.

Then, we modified the *restoreResultsFromFile* function from the *dataHandler* module. If the stored hash is different from the one sent by the assessor, the legacy guide content is sent to the *initializeData* function of the same module in order to change the guide content stored in the application. If this change occurs, the function fulfilment will return a boolean *false* value to specify this change to the caller. Based on that, the *Vue* component that calls the *restoreResultsFromFile* function displays a warning to the assessors to explain the situation, as shown in Figure 5.5 (A message specifying that the restoration process uses an old version of the guide content)

In addition, the components that used the *restoreResultsFromFile* function have also been modified in order to use the correct *GuideData* interface. We also made sure that the generated JSON files are compliant with this same interface.

[50] *PWA Vite Plugin* source: `https://bit.ly/3WQAnTK` (accessed 4th January 2023)
[51] *Progressive Web Apps for Firefox* source: `https://bit.ly/3i88fwf` (accessed 4th January 2023)
[52] *Electron* source: `https://www.electronjs.org` (accessed 4th January 2023)

```
getWholeGuide: (state): GuideData => {
        let categories = [] as CategoryData[]
        let subcategories = [] as SubcategoryData[]
        let objectives = [] as ObjectiveData[]
        let items = [] as ItemData[]
        let descriptions = [] as DescriptionData[]
        state.guide.forEach((category) => {

                categories.push(category)
                category.subcategories.forEach( (subcategory) => {

                        subcategories.push(subcategory)
                        subcategory.objectives.forEach((objective) => {

                                objectives.push(objective)
                                objective.items.forEach((item) => {

                                        items.push(item)
                                        item.descriptions.forEach((description) => {

                                                descriptions.push(description)
                                        })
                                })
                        })
                })
        })

        return {
                categories: categories as CategoryData[],
                subcategories: subcategories as SubcategoryData[],
                objectives: objectives as ObjectiveData[],
                items: items as ItemData[],
                descriptions: descriptions as DescriptionData[],
                hash: state.hash as string
        } as GuideData
},
```

Listing 5.23 *New store getter to get the guide content*

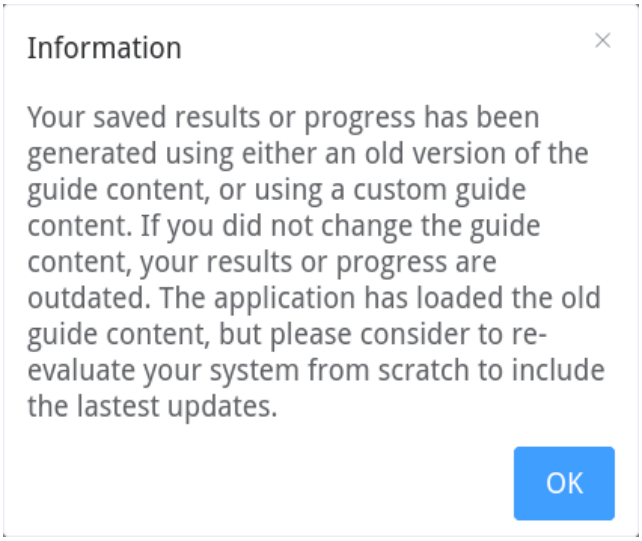


Figure 5.5 *A message specifying that the restoration process uses an old version of the guide content*

5.3.11 Miscellaneous

Some aspects were not described, such as the visual design part. This part has been done by first defining Cascading Style Sheets (CSS) rules to obtain appropriate UIs, and then by tweaking them to reach the most satisfying results as possible. We tried a lot of different styles, and we kept the ones we liked the most while respecting the most-known UI and UX principles. We used the *Element Plus*[53] visual framework which provides pre-built and well-designed components. Some of them have been used for our designs.

We also defined the `Indexes` class which allows us to retrieve indexes of parents given an object `PK`. We defined five methods, one for each object type, in order to retrieve the correct amount of indexes. Those are useful to navigate through the `Maps` that store categories and their children. One example of its usage can be seen at line 232 on Listing 5.16 (The store action called when an item is evaluated).

Some other types have also been defined to use the standardized way of TypeScript to handle data.

5.4 Final Results

Some screenshots of the application will be shown, related to its multiple parts we explained.

Figure 5.6 (The evaluation `Sidebar`) shows how the `Sidebar` is displayed during the evaluation process. When a category or a subcategory has been completely evaluated, its corresponding red dot is removed. Two buttons allow respectively to consult the evaluation results, and to save the evaluation progress. We also added the possibility to toggle the `Sidebar` visibility for mobile users.

Figure 5.7 (An example of an objective) shows how an objective is displayed on the screen. The colours given to items are defined appropriately to their requirement level. Evaluating items can be done by clicking on the checkboxes, which change their state at every click. If a description is available for an item, a collapsing icon is displayed.

Figure 5.8 (An example of the scores) shows how the scores are shown to the assessors. The colours are based on the percentage of the score: if a score is below fifty percent, the progress bar is red. If it is below eighty-five percent, it will be orange. If it is below a hundred percent, it will be displayed on light green, and a full one hundred percent shows the progress bar is displayed on darker green. The thresholds have been chosen subjectively, they can be changed in the future.

Finally, Figure 5.9 (An example of an upload card) shows an example of a drag and drop space to upload files. Two features need this component: when the assessors want to restore their results, and when they want to upload a custom version of the guide content.

[53] *Element Plus* source: `https://element-plus.org` (accessed 3rd January 2023)

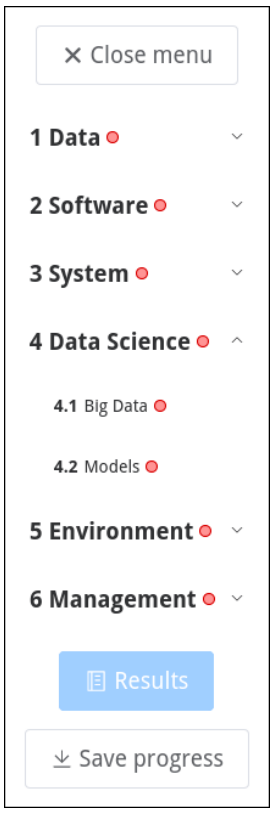


Figure 5.6 The evaluation `Sidebar`

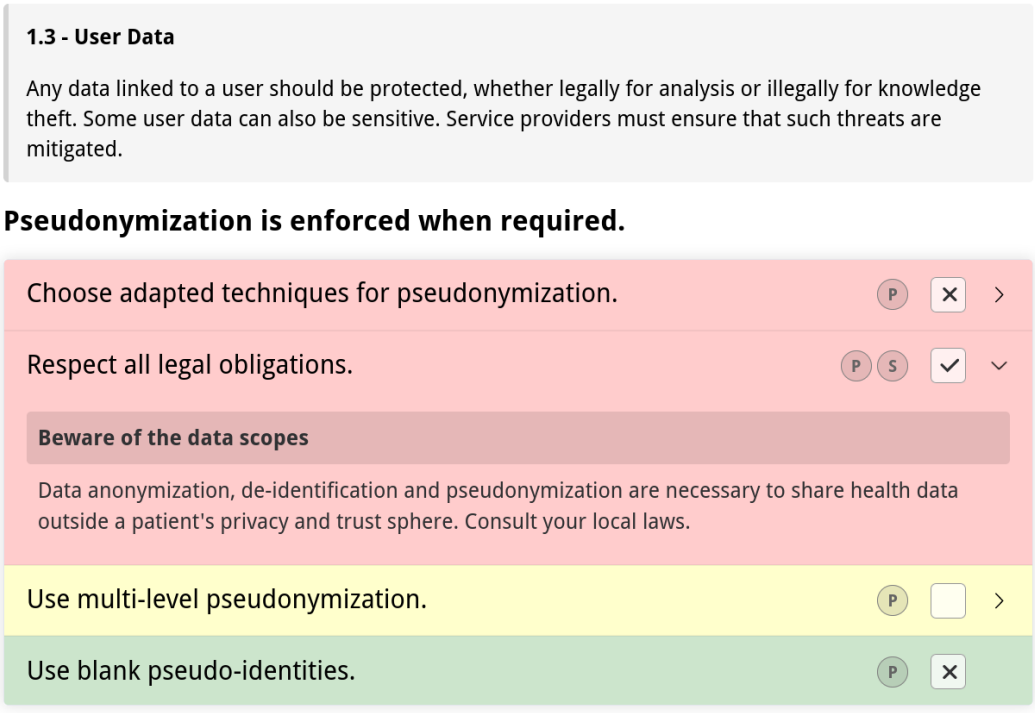


Figure 5.7 An example of an objective

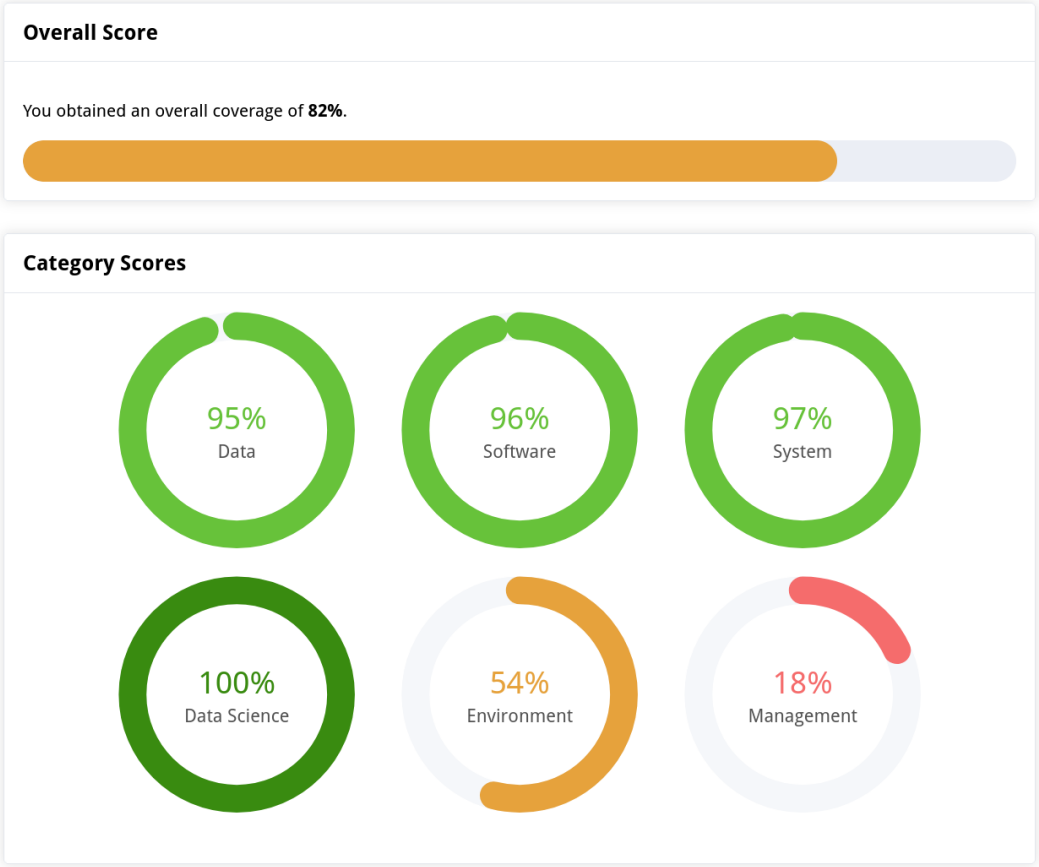


Figure 5.8 An example of the scores

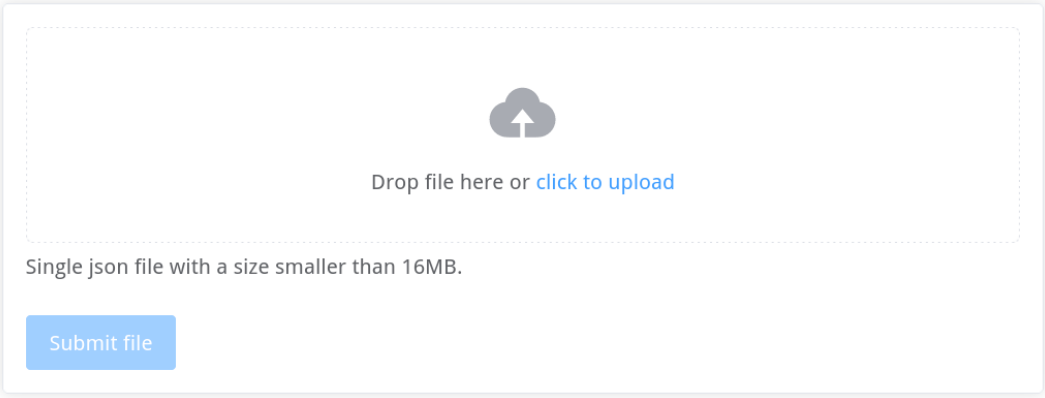


Figure 5.9 An example of an upload card

5.5 Software Tests

Software needs to be tested in order to prove that their processes behave accordingly to their goals and design. Furthermore, we need to think about all the scenario that can happen when using our software and show that they can handle problems without bugs or failures.

Two devices have been used for the tests. As a personal computer, we used a laptop running *Fedora Workstation 37*. As a smartphone, we used an *Android Samsung Galaxy S10 Plus*. Two browsers have been used on both devices, *Firefox* and *Chrome*. This selection allows us to cover most of the web browsing configurations.

The tests are separated based on the software piece they verify. Their details, expected results and obtained results are available at Appendix C.

5.5.1 Data Conversion and Validation Tests

The tests done on the data converter script are based on what has been explained in Subsection 5.3.1 (Data Conversion and Validation).

All the tests have been passed. Invalid values to determine whether an input is valid or not are based on the content in Table 4.5 (Attributes of the file).

5.5.2 Application Tests

Tests of several natures have been conducted on the application. We tested its UIs, its behaviour, and its data store.

Some tests shown in this Subsection have also been implemented as unit tests that can be triggered either manually or during deployment. They are explained in Subsection 5.5.3 (Unit Tests).

All the tests have been passed.

User Interfaces

The UI tests are based on actions done by the assessors while using the application. Each action must give a result that is visible on the interface. We realized tests on each specific page, and on the overall application regarding its global actions.

In addition to the two devices used for the tests, we used the *Firefox* and *Chrome* built-in responsive design modes to simulate various displayed screen size.

All the tests have been passed.

Application Behaviour

The tests on the application behaviour are based on how the application reacts to specific states. Automatic rendering, values or computations should be carried out without any actions from the assessors.

All the tests have been passed.

Application Stores

The tests on the stores are based on the values and computations that are carried on the state. The state can be altered by the assessors, processed automatically or reacting to changes.

All the tests have been passed.

5.5.3 Unit Tests

Due to time constraints, we have selected the tests that are the most crucial to assess whether the application works. We decided to test whether the guide content store are valid, and to simulate the basic path and actions that assessors will do while evaluating their web service.

Each unit test is based on its corresponding tests defined before in this Section. The states, actions, and expected results are the same.

Environment

As explained in Subsubsection 5.3.2 (Project Creation), we used the *Vitest* unit test framework. This choice has been mainly motivated by the fact that *Vitest* is recommanded and supported by the *Vue* development team.

A few additional NPM modules had to be installed in order to complete our testing environment. They are only needed in development mode, not for production mode.

- ***pinia/testing***[54]: a tool that creates *Pinia* instances specifically designed for unit testing
- ***jest***[55]: another testing framework whose features can be used into *Vitest*
- ***happy-dom***[56]: a web browser simulator without any embedded GUI
- ***vue/test-utils***[57]: a testing utility that brings features to specifically test *Vue* components
- ***unplugin-auto-import***[58]: a utility used to automatically import APIs into *Vite* projects
- ***unplugin-vue-components***[59]: a utility used to automatically import *Vue* components into *Vue* applications

Usage of *unplugin-auto-import* and *unplugin-vue-components* has been motivated by our usage of the *Element Plus* visual framework, which needs all its components to be registered into the *Vue* application. Otherwise, *Vitest* is not able to reach their references.

Once added to the *Vue* project, the *unplugin-auto-import* and *unplugin-vue-components* modules have to be added to the *Vite* configuration, and register the *Element Plus* components, as shown in the Listing 5.24 (The modules added to the *Vite* configuration).

[54] ***pinia/testing*** source: `http://bit.ly/3Xh9ibq` (accessed 8th February 2023)
[55] ***jest*** source: `https://jestjs.io/` (accessed 8th February 2023)
[56] ***happy-dom*** source: `http://bit.ly/3YilNVJ` (accessed 8th February 2023)
[57] ***vue/test-utils*** source: `https://test-utils.vuejs.org` (accessed 8th February 2023)
[58] ***unplugin-auto-import*** source: `http://bit.ly/3HOLRR8` (accessed 8th February 2023)
[59] ***unplugin-vue-components*** source: `http://bit.ly/3HLgZRJ` (accessed 8th February 2023)

```
1  export default defineConfig({
2          [...]
3          plugins: [
4                  AutoImport({
5                          resolvers: [ElementPlusResolver({ ssr: true })],
6                  }),
7                  Components({
8                          resolvers: [ElementPlusResolver({ ssr: true })],
9                  }),
10         ]
11         [...]
12 });
```

Listing 5.24 The modules added to the Vite configuration

Unit Tests Implementation

The unit tests are defined in the `tests` folder, as explained in Table 5.1 (Structure of a *Vue* application). If some unit tests had been specific to a component or a module, they could have been defined in other folders. We made that choice because of our tests being focused on the whole application.

We created two testing files based on the application part they target. This approach is recommanded to guarantee a clear separation of duties and a better code understanding.

Vitest provides multiple functions to build tests. The `describe` function allows to define a set of tests. It allows organizing tests that target the same context in order to bring clarity. The `test` function allows to define a set of expectations that are related between each others. Some pre-attached functions can be defined in `describe` blocks in order to realize actions during the life cycle of the tests: we used the `beforeEach` function in order to keep the same *Pinia* store instance through all the tests. The test expectations are defined using the `expect` function that specifies, with the help of other functions assessing values, the result that must be observed after execution.

Vitest includes other useful functions, but we did not use them for our tests. It can also benchmark the application on its performances.

The first of our testing files is focused on the application store, as shown in Listing 5.25 (Unit tests on the application store). The first line encapsulates the two tests we defined for this application part. Before running each test, *Vitest* will create and pass a *Pinia* instance to them. Then, the lines 6 to 11 show the test on the initial state of the guide content store which must be empty by default. Then, the second test defined from line 13 to 19 applies the same approach but initializes the data loading process into the store before testing its values. In this configuration, the guide content data must have verified by the verification checks done by the *initializeData* function, as described in Subsubsection 5.3.3 (Data Verification), and then the store must have been populated using this data.

Our second testing file is focused on simulating the whole path of an assessor that evaluates its web service through the application. To this end, the different links that allows to navigate through the pages are searched for, and a click is emulated on them. This approach allows us to also test the whole interface of our application, as if an

```
describe('Evaluation Store', () => {
        beforeEach(() => {
                setActivePinia(createPinia())
        })

        it('Default state', () => {

                const store = useEvaluationStore()
                expect(store.getStatus).toBe(false)

        })

        it('Loaded state', () => {

                initializeData()
                const store = useEvaluationStore()
                expect(store.getStatus).toBe(true)

        })
})
```

Listing 5.25 Unit tests on the application store

assessor would have use their browser. Once the pages loaded, we make sure that the correct content is displayed, and we define the appropriate expectations. All the references to application content are made using dedicated `data` HTML attributes, with a value describing their role.

Here are the different tests we defined, all part of the same *Vitest* `description`:

1. Going from *HomeView* to *ResultsView* without the evaluation being completed
2. Going From *HomeView* to *ExplanationView*
3. Going From *ExplanationView* to *ResultsView*
4. Click on the result button in *EvaluationView* without the evaluation being completed
5. Evaluate all items in *EvaluationView* as compliant
6. Test if the overall score is of one hundred percent
7. Test if the overall score is lower than one hundred percent if only one item is evaluated as non-compliant and the others compliant
8. Test if the overall score is of one hundred percent if only item is evaluated as not concerned and the others compliant

Only the most interesting parts of the tests will be explained.

Simulating the whole application needs to mount and use a *Vue* instance as browsers do. This step is shown in Listing 5.26 (Creation of an instance of the *Vue* application). The instance declaration is done at the root of the `describe` block to allow it to be accessible to all tests.

We used the `mount` function brought by the *vue/test-utils* module, which allows us to use the *Vue Router* and *Pinia* stores we previously defined during implementation time to build the *Vue* instance used by *Vitest*. A specific constructor brought by the *pinia/testing* module allows *Pinia* stores to be tested inside the *Vitest* environment. Its `stubActions` argument states whether the actions should be disclosed to the state or not.

As shown from lines 12 to 13 on Listing 5.26 (Creation of an instance of the *Vue* application), every router route update must be followed by assessing whether the view is ready before proceeding to the rest of the test.

```
11  // Defining basic App entry.
12  router.push("/")
13  await router.isReady();
14
15  // Mount the app as it would be in a browser, with the router and the store
16  const appWrapper = mount(App, {
17          global: {
18                  plugins: [router, createTestingPinia({ stubActions: false })]
19          }
20  });
```

Listing 5.26 Creation of an instance of the Vue application

Listing 5.27 (Unit test on the `ResultsView` access without a completed evaluation) shows the test that assess whether the `ResultsView` page is not accessible if the evaluation is not complete. Line 25 shows the route being changed, then the next line waits for the navigation to be ready. Line 27 shows the `flushPromises` function from the *vue/test-utils* module that must be used when any update is made on a *Vue Router* instance in testing mode to avoid errors. Then, line 30 shows an expectation that assess whether the active route is the one linked to the `EvaluationView` page, because of the redirection made by a navigation guard we defined for this use case. Finally, we navigate to the root page of the application before the following test being run.

```
22  test("From HomeView to ResultsView without complete evaluation", async () => {
23
24          // Finding Router button
25          router.push("/results")
26          await router.isReady();
27          await flushPromises()
28
29          // Should be on the Evaluation page
30          expect(appWrapper.vm.$route.name).toBe("Evaluation");
31
32          // Going back to the home page
33          router.push("/")
34          await router.isReady();
35          await flushPromises()
36
37  });
```

Listing 5.27 Unit test on the `ResultsView` access without a completed evaluation

An example of a navigation test is shown in Listing 5.28 (Unit test on the navigation from the `HomeView` page to the `ExplanationView` one). This kind of tests is similar to the one done in Listing 5.27 (Unit test on the `ResultsView` access without a completed evaluation), with a different approach. Lines 45 to 49 show how the navigation buttons are found and clicked on using functions from the *vue/test-utils* module.

```
39  test("From HomeView to ExplanationView", async () => {
40
41          // Expecting to start on the Home page
42          expect(appWrapper.vm.$route.name).toBe("Home");
43
44          // Finding Router button
45          const navigationButton = appWrapper.find("[data-test=navToExplanation]");
46
47          // Trigger the button to navigate on the other page
48          await navigationButton.trigger('click')
49          await flushPromises()
50
51          // Should be on the Explanation page
52          expect(appWrapper.vm.$route.name).toBe("Explanation");
53
54  });
```

Listing 5.28 Unit test on the navigation from the `HomeView` page to the `ExplanationView` one

Listing 5.29 (Unit test extract that evaluates all items as compliant) shows an extract of the tests done on the `EvaluationView` page that evaluate all the items of the guide content as compliant. Line 93 allows us to find and store all the subcategories listed in the `Sidebar`. Then, an iteration is made on all the subcategories in order to click on them to display their corresponding objectives and items, as shown on line 97. Once each subcategory clicked on, the same approach is applied to find each of its items on the page to evaluate them as compliant by a click on their checkbox.

```
93  const subMenus = appWrapper.findAll("[data-test=subcategories]");
94  for (let i = 0; i < subMenus.length; i++) {
95
96          // Displaying all the related items
97          subMenus[i].trigger('click')
98          await flushPromises()
99
100         const items = appWrapper.findAll("[data-test=checkboxes]");
101
102         // Getting all the related items to click on them
103         for (let i = 0; i < items.length; i++) {
104                 items[i].trigger('click')
105         }
106 }
```

Listing 5.29 Unit test extract that evaluates all items as compliant

Listing 5.30 (Unit test on the value of the overall score) shows an extract of the unit tests that assess the overall score. Once the `ResultView` page accessed, the value of the overall score is obtained using the `find` method, and we test whether its value is equal to 100% by using the `toBe` assert function brought by *Vitest*.

```
130 const overallScore = appWrapper.find("[data-test=overallScore]");
131 expect(overallScore.text()).toBe("100%");
132
```

Listing 5.30 Unit test on the value of the overall score

5.5.4 Test Summary

All the tests we defined have been successfully validated. We did not find any problem during their validation, both for the data conversion and validation step than for the application.

The implemented unit tests are also validated. They are launched at every code update on the repository [1] before the application is deployed. The failure of any unit test causes the deployment to be stopped to avoid any problems in production.

5.6 Further Application Improvements

Our application is complete, works perfectly, and covers all the features and capabilities we defined. However, some improvements could still be done on its usage or its source code.

First, the UI and UX can be improved by following recognized guidelines or design languages. The *Material Design*[60] design language developed by *Google* could be an interesting lead.

The application can be adapted to follow internationalization and localization standards, although translating the guide content would be a heavy and risky task.

The CSS rules can be optimized, and the Syntactically Awesome Style Sheets (SASS) preprocessor can be used to both simplify and generalize the style rules throughout the components.

New *Vitest* unit tests can be created to cover all the tests defined in Subsection 5.5.2 (Application Tests) in order to make sure that all the features of the application and the actions done by assessors are valid.

A new page can be created to show all the categorized objectives, with or without their corresponding items, in a print-friendly layout. This would allow to provide a great summary of the guide content.

Finally, some *Vue* components can be improved on their logic part, or divided into multiple more specialized ones.

[60] *Material Design* source: https://material.io (accessed 31st January 2023)

5.7 Summary

We successfully developed an application that covers the thesis scope and its goals, while implementing the capabilities and features it had to provide. The web application medium was not explicitly planned in the thesis specifications, but we managed to develop it appropriately are we are satisfied with the results we obtained.

The analysis and the design of the application helped us during its implementation phase, but parts have been adapted once tested. For example, some parts of the UI have been changed to optimize the usability, and the progress saving and restoration feature has been added afterwards when we noticed that its lack could lead to discomfort while using the application.

The *vue* framework was a perfect fit: no feature or behaviour has been limited because of this technology choice. Furthermore, our prior knowledge helped us to quickly build a stable and robust software.

At some point, we totally reimplemented the *Pinia* store that manages the guide content in order to change the architecture of the state by storing objects in the structure of their corresponding parent. Initially, each object was stored in their own state, and heavy sorting operations had to be done to retrieve an object when referenced using their parent PK. By doing so, the store performance has been improved by reducing the amount of operations during data retrieval.

We encountered some problems during the implementation phase: most of them were minor, but we did lose some time on two major issues. First, the support of the PWA standard took us some time: the *Firefox* mobile bowsers were able to install the application, but not any *Chrome*-based browser. The source of the problem was an invalid link to the web manifest resource put in the HTML file used as a template to mount the *Vue* instance. Secondly, the *PWA Vite Plugin* module was misconfigured and blocked all navigation to internal links that was not defined within *Vue Router*. As a result, the report PDF could not be accessed. The reason was that the said module had a fallback configuration that redirects such routes to itself. Adding an appropriate rule in the TypeScript environment configuration fixed this issue.

As explained in Section 5.6 (Further Application Improvements), some further work could be done in order to improve our application.

6 Web Service Evaluation

Now that the application we developed allows to use our guide by loading our guide content, it will be applied on conditions as close as the real field as possible. This step will allow us to determine whether our proposal can actually be used.

We will start by explaining the methodology which will be applied during the guide utilization. Then, the results be obtains following the utilization will be detailed in order to bring more information on the conditions of the usage. Based on those results, we will discuss what could be done for further improvements.

Contents

6.1 Methodology

Using our proposal on an actual web service serves multiple objectives. To ensure that all of them will be covered, we defined a methodology that will be applied during the application usage.

The *development team* term refers to all the parties involved with the development process of the web service that has been used in this Chapter. The *developer* term will also be used to refer to people in the development team.

6.1.1 Objectives Definition

As stated, the guide usage step includes multiple objectives to be meet. Those objectives are the following:

- Find undetected **mistakes** in the guide content
- Find undetected **errors** in the application usage
- Assess the **accuracy** of the evaluation
- Discuss the whole process with the web service **development team**

Those purposes allow us to cover the three outcomes of our thesis that had been defined in Subsection 1.3.3 (Outcomes).

6.1.2 Metrics Definition

Some metrics must be defined and then measured in order to provide a useful and impartial evaluation of our complete guide. Those have been defined accordingly to the objectives that have been defined in Subsection 6.1.1 (Objectives Definition).

- **Duration**: how much time has been necessary to evaluate the web service?
- **Overall score**: with how many items is the evaluated web service compliant?
- **Category scores**: what are the highest-risk categories?
- **Feedback**: does the web service development team agree with the different score?

6.1.3 Web Service Choice

The web service we chose is the *Hestia*[61] project administered by Dr. Pascal Bruegger who is also one of the supervisors of this thesis. *Hestia* aims to connect medical patients with their doctors through a data exchange platform in a secure and private way.

The service is composed by one central server and by multiple client applications. Those applications are developed for desktop, mobile and web platforms. The whole project is still under development and is still in a prototyping phase. However, the central server implementation has been completed before being evaluated using our guide.

[61] *Hestia* source: https://icosys.ch/hestia (accessed 1st February 2023)

This choice has been motivated by the fact that this web service includes strong security and privacy concerns due to its sensitive use case. In addition, it also includes AI-related concerns, especially on the big data topic. Moreover, the fact that the project is still in prototype form will enable useful levers to improve its security and privacy levels to the development team for the future implementation steps.

6.1.4 Evaluation Process

We planned a meeting with the *Hestia* development team without them knowing any specifics on the thesis. This approach aims to assess whether newcomers are able to understand the explanations of the guide made on the `ExplanationView` page.

The whole evaluation process has been made with our monitor being shared to the development team in order to display the same view of our GASP application to everyone.

The web service has been evaluated by formally expressing out loud the objectives and their related items. Then, discussions between the developers occurred to decide on the web compliance with each item. If additional information on any item was requested by a developer, the corresponding item descriptions were expended.

6.1.5 Confidentiality

No particular ethical consideration or obligation have been applied during the evaluation process. The developers agreed their feedbacks to be used whilst staying anonymous. Furthermore, no sensitive or personal data were collected during this phase.

6.2 Obtained Results

The metrics we defined in Subsection 6.1.2 (Metrics Definition) will be measured and explained.

6.2.1 Duration

The entire evaluation process of the web service lasted for two hours and ten minutes, including the guide explanation page consultation. Some items took more time to be evaluated than others, often because of their higher complexity or larger scope.

We noticed that every evaluation done through our guide can have different durations: indeed, multiple factors influence the evaluation pace:

- **Assessors' confidence**: some people are more certain of their choices and capacities than others.
- **Assessors' investment**: all organizations have not the same resources, willingness, and motivation to perform such evaluations.
- **Topic knowledge**: some topics are more difficult to understand and/or more complex than others.
- **Web service complexity**: some web services and smaller and less complex than others, which has an impact on the amount of work.
- **Web service knowledge**: the assessor can be the one that developed the whole evaluated web service or one out of many in an entire organization.

6.2.2 Scores

Once the web service evaluation completed, we obtained the overall and the category scores. Those scores are displayed on Figure 6.1 (The overall score from the *Hestia* evaluation) and on Figure 6.2 (The category scores from the *Hestia* evaluation).

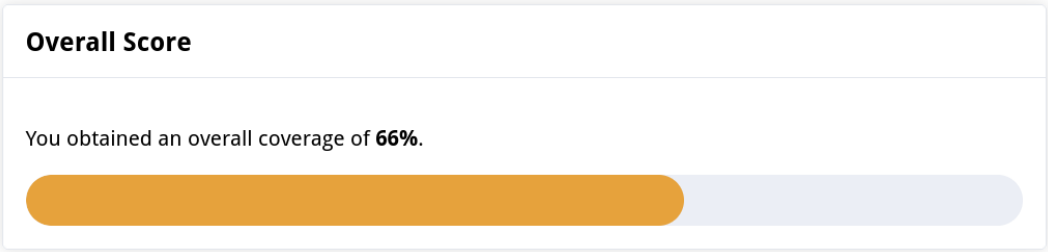


Figure 6.1 *The overall score from the Hestia evaluation*

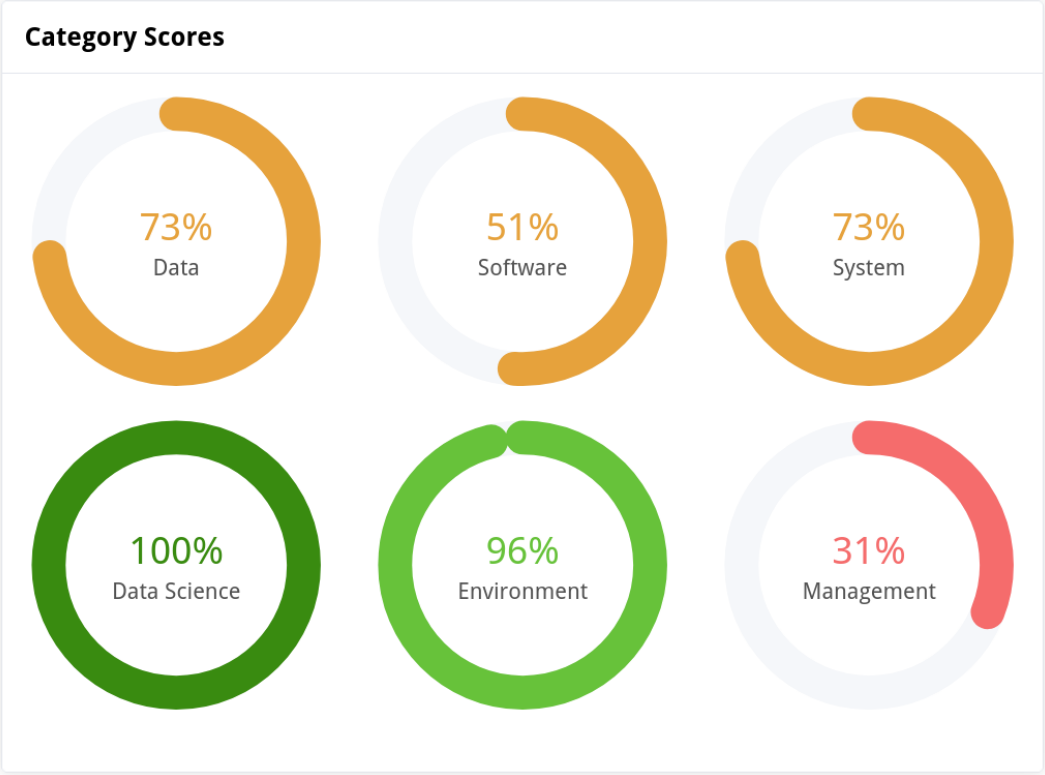


Figure 6.2 *The category scores from the Hestia evaluation*

A discussion about the scores values has been initiated with the developers to attest their veracity. The developers found the results consistent with the idea they have on their web service.

The lack of compliance with the management can be explained by the fact that policies or risk management are concerns that will be handled in future stages of the project development. Furthermore, the organization the development team is attached to is not the one that will eventually handle the production stage of the project in its final shape.

The data and the system categories have a consistent score, with a majority of items already being handled or integrated in the current state of the project. Some parts of those two categories will be processed in future stages of the project, which explains the remaining gap to get complete score values.

The software category shows a few lacks in the development process. Some items could have been integrated into the current development process, which helped to identify this weakness. However, the testing part of the project is planned at the end of its timeframe.

The great scores of the environment and data science categories are mainly explained by the fact that some techniques or approaches related to those categories are not yet integrated. However, the ones that are already implemented attest that the web service is compliant with the categories remaining items.

From our perspective, we share the opinion of the development team on the accuracy of the scores. The insights given by our guide give a comprehensible yet complete status of the current state of the evaluated web service. Some scores can be explained by the fact that the evaluated web service is still at a prototype phase, which caused the lacks shown by the evaluation process. The results seem to us to represent correctly the security and privacy levels of the *Hestia* project.

6.2.3 Feedback

Everyone in the development team found the guide accessibility, format, and volume as appropriate to its purpose, even if the amount of items is significant. However, the developers agreed that this workload is appropriate given the complexity of the security and privacy concerns.

The application itself has been evaluated as optimized and well-designed, and also offers a great usability. However, some improvements have been proposed by the developers:

- Add some navigation buttons to go to the next and previous subcategories.
- Add the possibility to ignore a whole category, subcategory or objective.
- Add a feedback form to ask for assessors' advices or new content.
- Add a guard to avoid any data loss when refreshing the application.

The listed improvements are noteworthy. Because of the limited time at the end of the thesis project, these remarks will be addressed on our free time.

6.3 Summary

We are satisfied with the feedbacks and the remarks we obtained during this evaluation process. The accuracy of our guide have been reviewed and tested by external parties which shows that our proposal is valid, taking into account the sample size.

The methodology to test the usage of our proposal has been designed according to the scope of our thesis. However, our proposal could also be tested by organizing a qualitative and/or quantitative studies, using a standardized approach and an appropriate protocol. Furthermore, the UI and UX could be included in the process with additional user testing. Those steps could be part of a future work on the same subject.

7 Conclusion

Now that we have completed our thesis, we will summarize the work we realized during it and discuss its major aspects. The content of the specifications document, which is available at Appendix A, will be used through this Chapter.

We will start by a definition of the final state of this thesis, which will be compared with the thesis objectives and with our research question. We will also discuss the improvements that can be done. Then, the choices made during the thesis will be discussed and assessed on their correctness. Then, the problems we met will be explained, and the limitations of our work will be listed. The future work we foresee for our thesis will then be identified, followed by an explanation on our planning differences. We will conclude by a personal feedback.

Contents

7.1 Thesis State

All the activities and tasks we initially planned were completed during the completion of our thesis. Some small discrepancies occurred, but the global journey stayed the same. The knowledge collection, the guide proposal, the guide content creation, the application development and the evaluation of the guide parts have all been realized as defined in the specifications document. The publishing part has also been carried out.

7.1.1 Comparison with the Objectives

All the thesis objectives were achieved on time. We will describe each of them in more detail to explain how they were achieved.

Establish an Up-To-Date Knowledge Collection

We did make a knowledge collection composed by rules, best practices, technologies and aspects that contribute to enforce the security and privacy levels of web services. Although being non-exhaustive because of the vastness of these two areas, our collection successfully allows to have a global and broad view of the topics we treated.

This objective is fulfilled. However, further knowledge additions would be appreciated.

Provide an Understandable Guide

We stated that either a guide of a framework would be defined during our thesis. We did propose a new way of evaluating systems with guidelines, methods, and precise rules to be respected. This outcome as been referred as a guide through our thesis, but it can also be considered as a framework. Our proposal allows evaluating web services, but is generic and is also appropriate for evaluating other types of systems if desired.

On top of the guide, an application has been developed to make its usage simpler, more accessible, and more efficient to use. The result we obtained totally meets the objective.

This objective is fulfilled.

Apply and Test the Guide on an Online Service

We did perform an assessment of our proposal by evaluating a web service using our guide. Meaningful metrics have been defined and measured, and we manage to determine that the approaches of both our guide and application were valid. The application itself has been evaluated has optimized and well-designed, and also allows a simple usage.

This objective is fulfilled.

Secondary Objective: Publish the Guide as an Online Resource

This secondary objective has been achieved: we did publish our work on a publicly accessible server. The application we developed is hosted on a public address:

ohmygasp.com

This objective is fulfilled.

Constraint

A constraint had been defined to address a situation where data would have been collected under real-life conditions. We did not conduct any data collection task during our thesis that had to respect this constraint.

7.1.2 Research Question

As stated in Subsection 1.3.2 (Research Question), a research question had been posed in order to obtain an answer based on the work done during this thesis:

> How to help organizations to evaluate their web services that also use AI processes, by measuring the risk levels of both ICT security and user privacy concerns?

The best way we found to help organizations on this point is to allow them to evaluate whether their web services are compliant with the major risks identified by researchers, using an accessible and simple guide to evaluate the said compliance. Different scores and requirement levels are provided by the guide in order to give them a prioritization on the most sensitive concerns, in order to quickly activate levers to strengthen their web services.

7.1.3 Further Improvements

Our guide content can be further improved by adding the items and descriptions source. Indeed, we kept a trace of their origin while we built the knowledge collection, but we should have added this data into the spreadsheet. By doing so, the sources could then be consulted on the application by assessors.

The knowledge collection can be completed with other relevant sources such as other papers, grey literature resources, or the content provided by other guides we found during the analysis made in Section 3.2 (Guides Analysis).

If an update is made on the guide by adding new content, any saved progress or results made before the update would be outdated. In this case, assessors can not take advantage of the new added content without redoing the entire evaluation process. In order to allow them to effortlessly access to new additions, a tool could be developed to convert results or progress generated using any old guide version to new ones adapted to the updated content.

As explained in Section 5.6 (Further Application Improvements), the application can be improved on multiple points. This consists mainly of minor improvements on its UI and on its navigation, with some optimizations in the source code. Those changes would not greatly impact the UX, our application being already well-designed.

Those different improvements will certainly be implemented in our free time, out of the thesis scope.

7.2 Choices Made

We made several choices during this thesis. Those choices will be reviewed in order to establish whether the decisions we made were appropriate.

The topics we selected to build our knowledge collection were totally appropriate. We managed to include a great amount of sources which constituted a comprehensive vision of the security and privacy concerns. Moreover, we managed to identify complementary yet diverse sources for each of the topics. This allowed us to guarantee a great coverage of our knowledge collection.

Regarding our proposal, the guidelines of its structure and content have allowed us to provide a simple yet complete and effective way of evaluating web services. The combination of hierarchy, categorization, and information about each item ended up being a perfect compromise between simplicity and completeness.

The guide content being stored into a spreadsheet file was a good choice. It allowed us to easily add new content, to define cell references and to sort the content, which saved us some time while building it. Moreover, this support is human-readable without using specialized tools.

The technologies chosen to develop the application were totally appropriate to our needs. *Vue* and the other modules allowed us to implement each capability we wanted, and our experience with this framework helped us to build a stable and robust application.

The design we chose for the application UIs is entirely adapted to our proposal characteristics, and allows a clear understanding by the assessors.

Once our thesis completed, we were very happy with the definition and the approach of the risk assessment. This choice allowed us to reach of objectives at time considering the great amount of knowledges we had to collect. In addition, the risks prioritization is consistent and satisfactory, with the most risky items being ranked as the highest priority and the less risky items as low priority. The majority of the items, having limited risks, have been classified as medium priority. This distribution is a result that fully meets our objectives and the scope of our guide.

No other major choices had to be done through this thesis.

We are very satisfied with the results of our choices and their positive impact on the quality of our work. Those choices were questioned and validated by our supervisors.

7.3 Encountered Problems

We did not encounter a great amount of problems. Furthermore, those problems did not cause us much harm. However, we did meet some issues during the application implementation, as explained in Section 5.7 (Summary).

7.4 Limitations

We are aware that our approach has some limitations which we will address.

First, the protocol we defined in Subsection 2.1.2 (Review Protocol) has some limits. Although being adapted to the scope of our thesis and allowed us to reach our objectives, it did not allow us to build a knowledge collection as complete as one that would have been done using a systematic review. In addition, consultation with specialists in each topic would be appreciated. However, further additions are still possible in order to develop and improve our guide.

Secondly, the risk assessment as defined in Subsection 4.4.2 (Requirement Levels) is not optimal. This choice was however explained and justified, and is still the best solution we found according to our thesis scope. In case of a new approach being chosen, this method can be easily changed for another one, with adapted intervals to classify the requirements levels.

7.5 Future Work

The generic approach of our proposal allows us to imagine the development of a centralized or decentralized hub that would enable to store, compare and design different guide contents for different subjects, scopes or technologies. The application we developed could include this hub as a repository and let assessors chose the most appropriate guide for their needs.

Our proposal and guide can be assessed by qualitative and/or quantitative studies to evaluate their real and measured capacity of help organization to improve their security and privacy levels. This could be the subject of further research.

Our guide could be offered as a PDF file alongside the web application medium. As the guide content is defined in a generic way, the generation of this new medium would be achievable without any modification on it, while allowing new use cases for assessors. Alternatively, the application could implement a functionality to generate a print-friendly web page using its data coming from the guide content.

7.6 Planning Differences

When planning the thesis, we did not know what format our guide would use: therefore, we defined our planning in a way to be suitable for any choice. This approach has been successful given the slight changes that has resulted.

One of the two major planning variations was caused by the fact that we worked on both the guide content definition and on the application development on the same time. This allowed us to take a step back from those two parts of our work and to make changes or improvements when we returned to those tasks.

The second major variation was the time that the *Test and Evaluation* activity actually took us to conduct. We planned it across too many days: those were taken for the guide content definition instead.

An updated version of the planning, whose original version is available at Appendix A, can be consulted at Appendix D. Please note that the goal of the updated planning is to give an overview of the activities and tasks we did, and does not reflect all we work we did.

7.7 Personal Feedback

We greatly appreciated working on this thesis. Its subject, scope and outcomes are totally suited to our centres of interest, are meaningful, and can be useful to anyone interested.

All the activities done during this thesis were appropriate. The whole work process went well, and we were not stuck or bothered with major problems. We are happy with our project management as well.

We are very proud and satisfied with the final state of our thesis. Our knowledge collection, proposal and application have a high level of quality, and we are convinced that we fully reached our goal. Furthermore, we were able to answer to our research question and achieved all the objectives we defined.

In the future, we will certainly continue to contribute to this project, whose software repository will be made available to the public. The different improvements that could be done will be implemented, and we will also think about how to develop the future work we discussed.

Finally, this thesis allowed us to learn numerous new knowledges about security and privacy. We discovered new attacks, weaknesses, technologies, and other meaningful aspects on those two subjects. We greatly appreciated this new experience, and we truly believe that using our guide would bring great benefits to any organization developing web services.

References

[1] *Loïc Guibert / Master Thesis · GitLab*. Loïc GUIBERT, Sept. 2022. URL: https://gitlab.forge.hefr.ch/loic.guibert/mt.

[2] Ricardo Neisse et al. "A privacy enforcing framework for Android applications". In: *Computers & Security* 62 (2016), pp. 257–277. ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2016.07.005. URL: https://www.sciencedirect.com/science/article/pii/S0167404816300840.

[3] S. Raj Rajagopalan et al. "Smart meter privacy: A utility-privacy framework". In: *2011 IEEE International Conference on Smart Grid Communications (SmartGridComm)*. Oct. 2011, pp. 190–195. DOI: 10.1109/SmartGridComm.2011.6102315.

[4] David Kotz, Sasikanth Avancha, and Amit Baxi. "A Privacy Framework for Mobile Health and Home-Care Systems". In: *Proceedings of the First ACM Workshop on Security and Privacy in Medical and Home-Care Systems*. SPIMACS '09. Chicago, Illinois, USA: Association for Computing Machinery, 2009, pp. 1–12. ISBN: 9781605587905. DOI: 10.1145/1655084.1655086. URL: https://doi.org/10.1145/1655084.1655086.

[5] F.B. Schneider. "Least privilege and more [computer security]". In: *IEEE Security & Privacy* 1.5 (2003), pp. 55–59. DOI: 10.1109/MSECP.2003.1236236.

[6] Aaron Blankstein and Michael J. Freedman. "Automating Isolation and Least Privilege in Web Services". In: *2014 IEEE Symposium on Security and Privacy*. 2014, pp. 133–148. DOI: 10.1109/SP.2014.16.

[7] Pietro Colombo and Elena Ferrari. "Access Control in the Era of Big Data: State of the Art and Research Directions". In: June 2018, pp. 185–192. DOI: 10.1145/3205977.3205998.

[8] Chong K. Liew, Uinam J. Choi, and Chung J. Liew. "A Data Distortion by Probability Distribution". In: *ACM Trans. Database Syst.* 10.3 (Sept. 1985). Place: New York, NY, USA Publisher: Association for Computing Machinery, pp. 395–411. ISSN: 0362-5915. DOI: 10.1145/3979.4017. URL: https://doi.org/10.1145/3979.4017.

[9] Josep Domingo-Ferrer and Vicenç Torra. "A critique of the sensitivity rules usually employed for statistical table protection". In: *Internat. J. Uncertain. Fuzziness Knowledge-Based Systems* 10.05 (Oct. 2002). Publisher: World Scientific Pub Co Pte Lt, pp. 545–556.

[10] Andreas Pfitzmann and Marit Hansen. *A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management*. 2010.

[11] Waltraut Kotschy. *The new General Data Protection Regulation-Is there sufficient pay-off for taking the trouble to anonymize or pseudonymize data*. 2016.

[12] Buket Yüksel, Alptekin Küpçü, and Öznur Özkasap. "Research issues for privacy and security of electronic health services". In: *Future Generation Computer Systems* 68 (Mar. 2017), pp. 1–13. ISSN: 0167-739X. DOI: 10.1016/j.future.2016.08.011. URL: https://www.sciencedirect.com/science/article/pii/S0167739X16302667.

[13] Suntherasvaran Murthy et al. "A Comparative Study of Data Anonymization Techniques". In: *2019 IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*. 2019, pp. 306–309. DOI: 10.1109/BigDataSecurity-HPSC-IDS.2019.00063.

[14] Fatima Hussain et al. "Enterprise API Security and GDPR Compliance: Design and Implementation Perspective". In: *IT Professional* 22.5 (2020), pp. 81–89. DOI: 10.1109/MITP.2020.2973852.

[15] Bram Bonne, Peter Quax, and Wim Lamotte. "The Privacy API: Facilitating Insights in How One's Own User Data is Shared". In: *2017 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. Paris: IEEE, Apr. 2017, pp. 72–75. ISBN: 978-1-5386-2244-5. DOI: 10.1109/EuroSPW.2017.54. URL: http://ieeexplore.ieee.org/document/7966974/ (visited on 11/09/2022).

[16] Aidah Ichario and Manuel Maarek. "Vision: Investigating Web API Developer Experience in Relation to Terms of Service and Privacy Policies". In: *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. Genoa, Italy: IEEE, Sept. 2020, pp. 166–171. ISBN: 978-1-72818-597-2. DOI: 10.1109/EuroSPW51379.2020.00030. URL: https://ieeexplore.ieee.org/document/9229791/ (visited on 11/09/2022).

[17] Salah Sharieh and Alexander Ferworn. "Securing APIs and Chaos Engineering". In: *2021 IEEE Conference on Communications and Network Security (CNS)*. Tempe, AZ, USA: IEEE, Oct. 2021, pp. 290–294. ISBN: 978-1-66544-496-5. DOI: 10.1109/CNS53000.2021.9705049. URL: https://ieeexplore.ieee.org/document/9705049/ (visited on 11/09/2022).

[18] Reza Shokri and Vitaly Shmatikov. "Privacy-Preserving Deep Learning". In: *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. CCS '15. event-place: Denver, Colorado, USA. New York, NY, USA: Association for Computing Machinery, 2015, pp. 1310–1321. ISBN: 978-1-4503-3832-5. DOI: 10.1145/2810103.2813687. URL: https://doi.org/10.1145/2810103.2813687.

[19] Jian-hua Li. "Cyber security meets artificial intelligence: a survey". en. In: *Frontiers of Information Technology & Electronic Engineering* 19.12 (Dec. 2018), pp. 1462–1474. ISSN: 2095-9184, 2095-9230. DOI: 10.1631/FITEE.1800573. URL: http://link.springer.com/10.1631/FITEE.1800573 (visited on 11/10/2022).

[20] Le Trieu Phong et al. "Privacy-Preserving Deep Learning via Additively Homomorphic Encryption". In: *IEEE Transactions on Information Forensics and Security* 13.5 (2018), pp. 1333–1345. DOI: 10.1109/TIFS.2017.2787987.

[21] Alfredo Vellido. "Societal Issues Concerning the Application of Artificial Intelligence in Medicine". en. In: *Kidney Diseases* 5.1 (2019), pp. 11–17. ISSN: 2296-9381, 2296-9357. DOI: 10.1159/000492428. URL: https://www.karger.com/Article/FullText/492428 (visited on 11/10/2022).

[22] Mingfu Xue et al. "Machine Learning Security: Threats, Countermeasures, and Evaluations". In: *IEEE Access* 8 (2020), pp. 74720–74742. DOI: 10.1109/ACCESS.2020.2987435.

[23] Yi Zhang et al. "Ethics and privacy of artificial intelligence: Understandings from bibliometrics". en. In: *Knowledge-Based Systems* 222 (June 2021), p. 106994. ISSN: 09507051. DOI: 10.1016/j.knosys.2021.106994. URL: https://linkinghub.elsevier.com/retrieve/pii/S0950705121002574 (visited on 11/10/2022).

[24] Bo Liu et al. "When Machine Learning Meets Privacy: A Survey and Outlook". In: *ACM Comput. Surv.* 54.2 (Mar. 2021). Place: New York, NY, USA Publisher: Association for Computing Machinery. ISSN: 0360-0300. DOI: 10.1145/3436755. URL: https://doi.org/10.1145/3436755.

[25] Ximeng Liu et al. "Privacy and Security Issues in Deep Learning: A Survey". In: *IEEE Access* 9 (2021), pp. 4566–4593. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.3045078. URL: https://ieeexplore.ieee.org/document/9294026/ (visited on 11/10/2022).

[26] Tianqing Zhu et al. "More Than Privacy: Applying Differential Privacy in Key Areas of Artificial Intelligence". In: *IEEE Transactions on Knowledge and Data Engineering* (2021), pp. 1–1. ISSN: 1041-4347, 1558-2191, 2326-3865. DOI: 10.1109/TKDE.2020.3014246. URL: https://ieeexplore.ieee.org/document/9158374/ (visited on 11/10/2022).

[27] Ariel Rabkin. "Personal Knowledge Questions for Fallback Authentication: Security Questions in the Era of Facebook". In: *Proceedings of the 4th Symposium on Usable Privacy and Security*. SOUPS '08. event-place: Pittsburgh, Pennsylvania, USA. New York, NY, USA: Association for Computing Machinery, 2008, pp. 13–23. ISBN: 978-1-60558-276-4. DOI: 10.1145/1408664.1408667. URL: https://doi.org/10.1145/1408664.1408667.

[28] Stuart Schechter, A.J. Bernheim Brush, and Serge Egelman. "It's No Secret. Measuring the Security and Reliability of Authentication via "Secret" Questions". In: *2009 30th IEEE Symposium on Security and Privacy*. 2009, pp. 375–390. DOI: 10.1109/SP.2009.11.

[29] Anil K Jain and Karthik Nandakumar. "Biometric authentication: System security and user privacy." In: *Computer* 45.11 (2012), pp. 87–92.

[30] Ignacio Velásquez, Angélica Caro, and Alfonso Rodríguez. "Authentication schemes and methods: A systematic literature review". en. In: *Information and Software Technology* 94 (Feb. 2018), pp. 30–37. ISSN: 09505849. DOI: 10.1016/j.infsof.2017.09.012. URL: https://linkinghub.elsevier.com/retrieve/pii/S0950584916301501 (visited on 11/04/2022).

[31] Sanjar Ibrokhimov et al. "Multi-Factor Authentication in Cyber Physical System: A State of Art Survey". In: *2019 21st International Conference on Advanced Communication Technology (ICACT)*. PyeongChang Kwangwoon_Do, Korea (South): IEEE, Feb. 2019, pp. 279–284. ISBN: 979-11-88428-02-1. DOI: 10.23919/ICACT.2019.8701960. URL: https://ieeexplore.ieee.org/document/8701960/ (visited on 11/04/2022).

[32] L. Kagal et al. "Authorization and privacy for semantic Web services". In: *IEEE Intelligent Systems* 19.4 (2004), pp. 50–56. DOI: 10.1109/MIS.2004.23.

[33] Daniel Fett, Ralf Kuesters, and Guido Schmitz. *A Comprehensive Formal Security Analysis of OAuth 2.0*. 2016. DOI: 10.48550/ARXIV.1601.01229. URL: https://arxiv.org/abs/1601.01229.

[34] Emil Larsson and Johan Sigholm. "Papering over the cracks: The effects of introducing best practices on the web security ecosystem". In: *2016 International Conference on Information Networking (ICOIN)*. 2016, pp. 1–6. DOI: 10.1109/ICOIN.2016.7427064.

[35] Qingxiong Ma and J. Michael Pearson. "ISO 17799: "Best Practices" in Information Security Management?" en. In: *Communications of the Association for Information Systems* 15 (2005). ISSN: 15293181. DOI: 10.17705/1CAIS.01532. URL: https://aisel.aisnet.org/cais/vol15/iss1/32 (visited on 10/17/2022).

[36] Priyank Jain, Manasi Gyanchandani, and Nilay Khare. "Big data privacy: a technological perspective and review". In: *Journal of Big Data* 3.1 (Nov. 2016), p. 25. ISSN: 2196-1115. DOI: 10.1186/s40537-016-0059-y. URL: https://doi.org/10.1186/s40537-016-0059-y.

[37] Karim Abouelmehdi et al. "Big data security and privacy in healthcare: A Review". In: *Procedia Computer Science* 113 (2017). The 8th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2017) / The 7th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH-2017) / Affiliated Workshops, pp. 73–80. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2017.08.292. URL: https://www.sciencedirect.com/science/article/pii/S1877050917317015.

[38] Abid Mehmood et al. "Protection of Big Data Privacy". In: *IEEE Access* 4 (2016), pp. 1821–1834. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2016.2558446.

[39] Jordi Soria-Comas and Josep Domingo-Ferrer. "Big Data Privacy: Challenges to Privacy Principles and Models". In: *Data Science and Engineering* 1.1 (Mar. 2016), pp. 21–28. ISSN: 2364-1541. DOI: 10.1007/s41019-015-0001-x. URL: https://doi.org/10.1007/s41019-015-0001-x.

[40] David Molnar and Stuart E Schechter. "Self Hosting vs. Cloud Hosting: Accounting for the Security Impact of Hosting in the Cloud." In: *WEIS*. Vol. 2010. 2010, pp. 1–18.

[41] Minqi Zhou et al. "Security and Privacy in Cloud Computing: A Survey". In: *2010 Sixth International Conference on Semantics, Knowledge and Grids*. 2010, pp. 105–112. DOI: 10.1109/SKG.2010.19.

[42] Eystein Mathisen. "Security challenges and solutions in cloud computing". In: *5th IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2011)*. 2011, pp. 208–212. DOI: 10.1109/DEST.2011.5936627.

[43] Shubhashis Sengupta, Vikrant Kaulgud, and Vibhu Saujanya Sharma. "Cloud Computing Security–Trends and Research Directions". In: *2011 IEEE World Congress on Services*. 2011, pp. 524–531. DOI: 10.1109/SERVICES.2011.20.

[44] Joel JPC Rodrigues et al. "Analysis of the Security and Privacy Requirements of Cloud-Based Electronic Health Records Systems". In: *J Med Internet Res* 15.8 (Aug. 2013), e186. ISSN: 14388871. DOI: 10.2196/jmir.2494. URL: http://www.ncbi.nlm.nih.gov/pubmed/23965254.

[45] Zhifeng Xiao and Yang Xiao. "Security and Privacy in Cloud Computing". In: *IEEE Communications Surveys & Tutorials* 15.2 (2013), pp. 843–859. DOI: 10.1109/SURV.2012.060912.00182.

[46] Rohan Jathanna and Dhanamma Jagli. "Cloud Computing and Security Issues". In: *International Journal of Engineering Research and Applications* 07 (June 2017), pp. 31–38. DOI: 10.9790/9622-0706053138.

[47] Christoph Bösch et al. "Tales from the Dark Side: Privacy Dark Strategies and Privacy Dark Patterns." In: *Proc. Priv. Enhancing Technol.* 2016.4 (2016), pp. 237–254.

[48] Arunesh Mathur et al. "Dark Patterns at Scale: Findings from a Crawl of 11K Shopping Websites". In: *Proc. ACM Hum.-Comput. Interact.* 3.CSCW (Nov. 2019). Place: New York, NY, USA Publisher: Association for Computing Machinery. DOI: 10.1145/3359183. URL: https://doi.org/10.1145/3359183.

[49] Linda Di Geronimo et al. "UI Dark Patterns and Where to Find Them: A Study on Mobile Applications and User Perception". In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. event-place: Honolulu, HI, USA. New York, NY, USA: Association for Computing Machinery, 2020, pp. 1–14. ISBN: 978-1-4503-6708-0. DOI: 10.1145/3313831.3376600. URL: https://doi.org/10.1145/3313831.3376600.

[50] Jamie Luguri and Lior Jacob Strahilevitz. "Shining a Light on Dark Patterns". In: *Journal of Legal Analysis* 13.1 (Mar. 2021). _eprint: https://academic.oup.com/jla/article-pdf/13/1/43/36669915/laaa006.pdf, pp. 43–109. ISSN: 2161-7201. DOI: 10.1093/jla/laaa006. URL: https://doi.org/10.1093/jla/laaa006.

[51] Milan Petković and Willem Jonker, eds. *Security, Privacy, and Trust in Modern Data Management*. en. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. ISBN: 978-3-540-69860-9 978-3-540-69861-6. DOI: 10.1007/978-3-540-69861-6. URL: http://link.springer.com/10.1007/978-3-540-69861-6 (visited on 10/18/2022).

[52] Paul Ashley, Calvin Powers, and Matthias Schunter. "From Privacy Promises to Privacy Management: A New Approach for Enforcing Privacy throughout an Enterprise". In: *Proceedings of the 2002 Workshop on New Security Paradigms*. NSPW '02. event-place: Virginia Beach, Virginia. New York, NY, USA: Association for Computing Machinery, 2002, pp. 43–50. ISBN: 1-58113-598-X. DOI: 10.1145/844102.844110. URL: https://doi.org/10.1145/844102.844110.

[53] Pavlos S. Efraimidis et al. "Towards privacy in personal data management". In: *Information Management & Computer Security* 17.4 (Jan. 2009). Publisher: Emerald Group Publishing Limited, pp. 311–329. ISSN: 0968-5227. DOI: 10.1108/09685220910993971. URL: https://doi.org/10.1108/09685220910993971 (visited on 10/18/2022).

[54] Anna Cinzia Squicciarini, Mohamed Shehab, and Federica Paci. "Collective Privacy Management in Social Networks". In: *Proceedings of the 18th International Conference on World Wide Web*. WWW '09. event-place: Madrid, Spain. New York, NY, USA: Association for Computing Machinery, 2009, pp. 521–530. ISBN: 978-1-60558-487-4. DOI: 10.1145/1526709.1526780. URL: https://doi.org/10.1145/1526709.1526780.

[55] Essam Mansour et al. "A Demonstration of the Solid Platform for Social Web Applications". en. In: *Proceedings of the 25th International Conference Companion on World Wide Web - WWW '16 Companion*. Montréal, Québec, Canada: ACM Press, 2016, pp. 223–226. ISBN: 978-1-4503-4144-8. DOI: 10.1145/2872518.2890529. URL: http://dl.acm.org/citation.cfm?doid=2872518.2890529 (visited on 11/02/2022).

[56] Markus Zimmermann et al. "Small World with High Risks: A Study of Security Threats in the npm Ecosystem". In: *28th USENIX Security Symposium (USENIX Security 19)*. Santa Clara, CA: USENIX Association, Aug. 2019, pp. 995–1010. ISBN: 978-1-939133-06-9. URL: https://www.usenix.org/conference/usenixsecurity19/presentation/zimmerman.

[57] Wentao Wang et al. "Detecting Software Security Vulnerabilities Via Requirements Dependency Analysis". In: *IEEE Transactions on Software Engineering* 48.5 (2022), pp. 1665–1675. DOI: 10.1109/TSE.2020.3030745.

[58] Iliya Georgiev and Ivo Georgiev. "A security model for distributed computing". In: *The Journal of Computing in Small Colleges* 17 (Nov. 2001).

[59] Hao Wang et al. "Security policy reconciliation in distributed computing environments". In: *Proceedings. Fifth IEEE International Workshop on Policies for Distributed Systems and Networks, 2004. POLICY 2004.* 2004, pp. 137–146. DOI: 10.1109/POLICY.2004.1309160.

[60] Vishal Kher and Yongdae Kim. "Securing Distributed Storage: Challenges, Techniques, and Systems". In: *Proceedings of the 2005 ACM Workshop on Storage Security and Survivability*. StorageSS '05. event-place: Fairfax, VA, USA. New York, NY, USA: Association for Computing Machinery, 2005, pp. 9–25. ISBN: 1-59593-233-X. DOI: 10.1145/1103780.1103783. URL: https://doi.org/10.1145/1103780.1103783.

[61] Himanshu Tyagi. "Distributed computing with privacy". In: *2012 IEEE International Symposium on Information Theory Proceedings*. 2012, pp. 1157–1161. DOI: 10.1109/ISIT.2012.6283035.

[62] Khalid El Makkaoui, Abdellah Ezzati, and Abderrahim Beni Hssane. "Challenges of using homomorphic encryption to secure cloud computing". In: *2015 International Conference on Cloud Technologies and Applications (CloudTech)*. 2015, pp. 1–7. DOI: 10.1109/CloudTech.2015.7337011.

[63] Muneer Bani Yassein et al. "Comprehensive study of symmetric key and asymmetric key encryption algorithms". In: *2017 International Conference on Engineering and Technology (ICET)*. 2017, pp. 1–7. DOI: 10.1109/ICEngTechnol.2017.8308215.

[64] Nachiketh Potlapally. "Hardware security in practice: Challenges and opportunities". In: *2011 IEEE International Symposium on Hardware-Oriented Security and Trust*. 2011, pp. 93–98. DOI: 10.1109/HST.2011.5955003.

[65] Masoud Rostami, Farinaz Koushanfar, and Ramesh Karri. "A Primer on Hardware Security: Models, Methods, and Metrics". In: *Proceedings of the IEEE* 102.8 (2014), pp. 1283–1295. DOI: 10.1109/JPROC.2014.2335155.

[66] Yier Jin. "Introduction to Hardware Security". In: *Electronics* 4.4 (2015), pp. 763–784. ISSN: 2079-9292. DOI: 10.3390/electronics4040763. URL: https://www.mdpi.com/2079-9292/4/4/763.

[67] *Video conferencing services: Security guidance for organisations*. Apr. 2020. URL: https://www.ncsc.gov.uk/guidance/video-conferencing-services-security-guidance-organisations.

[68] Dianne Solomon. "Balancing Privacy and Risk in the E-Messaging World". In: *IEEE Security & Privacy* 5.5 (2007), pp. 72–75. DOI: 10.1109/MSP.2007.105.

[69] Taiwo Ayodele and Dennis Adeegbe. "Cloud based emails boundaries and vulnerabilities". In: *2013 Science and Information Conference*. 2013, pp. 912–914.

[70] Ian D. Foster et al. "Security by Any Other Name: On the Effectiveness of Provider Based Email Security". In: *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. CCS '15. event-place: Denver, Colorado, USA. New York, NY, USA: Association for Computing Machinery, 2015, pp. 450–464. ISBN: 978-1-4503-3832-5. DOI: 10.1145/2810103.2813607. URL: https://doi.org/10.1145/2810103.2813607.

[71] Muhammad Ehsan Rana, Gong Wei, and Peter Hoornaert. "An Enterprise Instant Messaging (EIM) solution to cater issues associated with instant messaging (IM) in business". In: *2015 IEEE Student Conference on Research and Development (SCOReD)*. 2015, pp. 187–192. DOI: 10.1109/SCORED.2015.7449321.

[72] Sudarshan Chawathe. "Improving Email Security with Fuzzy Rules". In: *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. 2018, pp. 1864–1869. DOI: 10.1109/TrustCom/BigDataSE.2018.00282.

[73] Steven Englehardt, Jeffrey Han, and Arvind Narayanan. "I never signed up for this! Privacy implications of email tracking". In: *Proceedings on Privacy Enhancing Technologies* 2018 (Jan. 2018). DOI: 10.1515/popets-2018-0006.

[74] Jen-Wei Huang, Chia-Wen Chiang, and Jia-Wei Chang. "Email security level classification of imbalanced data using artificial neural network: The real case in a world-leading enterprise". In: *Engineering Applications of Artificial Intelligence* 75 (2018), pp. 11–21. ISSN: 0952-1976. DOI: https://doi.org/10.1016/j.engappai.2018.07.010. URL: https://www.sciencedirect.com/science/article/pii/S0952197618301544.

[75] Thomas Reisinger, Isabel Wagner, and Eerke Albert Boiten. "Security and Privacy in Unified Communication". In: *ACM Comput. Surv.* 55.3 (Feb. 2022). Place: New York, NY, USA Publisher: Association for Computing Machinery. ISSN: 0360-0300. DOI: 10.1145/3498335. URL: https://doi.org/10.1145/3498335.

[76] Allan Hodgson, Husam Arman, and Nabil Gindy. "An intelligent technology watch function for the high technology enterprise". In: *International Journal of Industrial and Systems Engineering* 3 (Jan. 2008). DOI: 10.1504/IJISE.2008.015913.

[77] Cristòfol Rovira. "Technology watch and competitive intelligence for SEM-SEO". In: (2008). Publisher: Hipertext. net.

[78] Yves Poullet. "Data protection legislation: What is at stake for our society and democracy?" In: *Computer Law & Security Review* 25.3 (Jan. 2009), pp. 211–226. ISSN: 0267-3649. DOI: 10.1016/j.clsr.2009.03.008. URL: https://www.sciencedirect.com/science/article/pii/S0267364909000612.

[79] Jan Philipp Albrecht. "How the GDPR Will Change the World Forword". eng. In: *European Data Protection Law Review (EDPL)* 2.3 (2016), pp. 287–289. URL: https://heinonline.org/HOL/P?h=hein.journals/edpl2&i=314.

[80] Michelle Goddard. "The EU General Data Protection Regulation (GDPR): European Regulation that has a Global Impact". en. In: *International Journal of Market Research* 59.6 (Nov. 2017), pp. 703–705. ISSN: 1470-7853, 2515-2173. DOI: 10.2501/IJMR-2017-050. URL: http://journals.sagepub.com/doi/10.2501/IJMR-2017-050 (visited on 10/18/2022).

[81] Marko Jäntti. "Studying Data Privacy Management in Small and Medium-Sized IT Companies". In: *2020 14th International Conference on Innovations in Information Technology (IIT)*. 2020, pp. 57–62. DOI: 10.1109/IIT50501.2020.9299050.

[82] Shivi Garg and Niyati Baliyan. "Comparative analysis of Android and iOS from security viewpoint". In: *Computer Science Review* 40 (May 2021), p. 100372. ISSN: 1574-0137. DOI: 10.1016/j.cosrev.2021.100372. URL: https://www.sciencedirect.com/science/article/pii/S1574013721000125.

[83] Aaron Mos and Md Minhaz Chowdhury. "Mobile Security: A Look into Android". In: *2020 IEEE International Conference on Electro Information Technology (EIT)*. 2020, pp. 638–642. DOI: 10.1109/EIT48999.2020.9208339.

[84] Primal Wijesekera et al. "The Feasibility of Dynamically Granted Permissions: Aligning Mobile Privacy with User Preferences". In: *2017 IEEE Symposium on Security and Privacy (SP)*. 2017, pp. 1077–1093. DOI: 10.1109/SP.2017.51.

[85] Dezhi Wu et al. "Effects of the design of mobile security notifications and mobile app usability on users' security perceptions and continued use intention". In: *Information & Management* 57.5 (July 2020), p. 103235. ISSN: 0378-7206. DOI: 10.1016/j.im.2019.103235. URL: https://www.sciencedirect.com/science/article/pii/S0378720617301313.

[86] Jan Lauren Boyles, Aaron Smith, and Mary Madden. "Privacy and data management on mobile devices". In: *Pew Internet & American Life Project* 4 (2012), pp. 1–19.

[87] Aneta Poniszewska-Marańda, Łukasz Chomatek, and Joanna Ochelska-Mierzejewska. "Secure Development Strategy Model Framework for Security of Mobile Applications". In: *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. 2021, pp. 1294–1299. DOI: 10.1109/TrustCom53373.2021.00178.

[88] Douglas J. Leith. "Mobile Handset Privacy: Measuring the Data iOS and Android Send to Apple and Google". en. In: *Security and Privacy in Communication Networks*. Ed. by Joaquin Garcia-Alfaro et al. Vol. 399. Series Title: Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Cham: Springer International Publishing, 2021, pp. 231–251. ISBN: 978-3-030-90021-2 978-3-030-90022-9. DOI: 10.1007/978-3-030-90022-9_12. URL: https://link.springer.com/10.1007/978-3-030-90022-9_12 (visited on 10/25/2022).

[89] G.A. Marin. "Network security basics". In: *IEEE Security & Privacy* 3.6 (2005), pp. 68–72. DOI: 10.1109/MSP.2005.153.

[90] Mohan V. Pawar and J. Anuradha. "Network Security and Types of Attacks in Network". In: *International Conference on Computer, Communication and Convergence (ICCC 2015)* 48 (Jan. 2015), pp. 503–506. ISSN: 1877-0509. DOI: 10.1016/j.procs.2015.04.126. URL: https://www.sciencedirect.com/science/article/pii/S1877050915006353.

[91] Ibrahim Ghafir et al. "A Survey on Network Security Monitoring Systems". In: *2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*. 2016, pp. 77–82. DOI: 10.1109/W-FiCloud.2016.30.

[92] Alireza Kavianpour and Michael C. Anderson. "An Overview of Wireless Network Security". In: *2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud)*. 2017, pp. 306–309. DOI: 10.1109/CSCloud.2017.45.

[93] F. Cuppens et al. "A Formal Approach to Specify and Deploy a Network Security Policy". en. In: *Formal Aspects in Security and Trust*. Ed. by Theo Dimitrakos and Fabio Martinelli. Vol. 173. Series Title: IFIP International Federation for Information Processing. New York: Springer-Verlag, 2005, pp. 203–218. ISBN: 978-0-387-24050-3. DOI: 10.1007/0-387-24098-5_15. URL: http://link.springer.com/10.1007/0-387-24098-5_15 (visited on 10/18/2022).

[94] Youssef Bassil. *Windows And Linux Operating Systems From A Security Perspective*. 2012. DOI: 10.48550/ARXIV.1204.0197. URL: https://arxiv.org/abs/1204.0197.

[95] Akinlolu Adekotujo et al. "A Comparative Study of Operating Systems: Case of Windows, UNIX, Linux, Mac, Android and iOS". In: *International Journal of Computer Applications* 176 (July 2020), pp. 16–23. DOI: 10.5120/ijca2020920494.

[96] Matthew R. Yaswinski, Md Minhaz Chowdhury, and Mike Jochen. "Linux Security: A Survey". In: *2019 IEEE International Conference on Electro Information Technology (EIT)*. 2019, pp. 357–362. DOI: 10.1109/EIT.2019.8834112.

[97] Gaoshou Zhai and Yaodong Li. "Analysis and Study of Security Mechanisms inside Linux Kernel". In: *2008 International Conference on Security Technology*. 2008, pp. 58–61. DOI: 10.1109/SecTech.2008.17.

[98] Kenneth J. Knapp et al. "Information security policy: An organizational-level process model". In: *Computers & Security* 28.7 (Oct. 2009), pp. 493–508. ISSN: 0167-4048. DOI: 10.1016/j.cose.2009.07.001. URL: https://www.sciencedirect.com/science/article/pii/S0167404809000765.

[99] Bulgurcu, Cavusoglu, and Benbasat. "Information Security Policy Compliance: An Empirical Study of Rationality-Based Beliefs and Information Security Awareness". In: *MIS Quarterly* 34.3 (2010), p. 523. ISSN: 02767783. DOI: 10.2307/25750690. URL: https://www.jstor.org/stable/10.2307/25750690 (visited on 10/18/2022).

[100] Nader Sohrabi Safa et al. "Information security conscious care behaviour formation in organizations". In: *Computers & Security* 53 (Sept. 2015), pp. 65–78. ISSN: 0167-4048. DOI: 10.1016/j.cose.2015.05.012. URL: https://www.sciencedirect.com/science/article/pii/S0167404815000863.

[101] Mutlaq Alotaibi, Steven Furnell, and Nathan Clarke. "Information security policies: A review of challenges and influencing factors". In: *2016 11th International Conference for Internet Technology and Secured Transactions (ICITST)*. 2016, pp. 352–358. DOI: 10.1109/ICITST.2016.7856729.

[102] Zahoor Ahmed Soomro, Mahmood Hussain Shah, and Javed Ahmed. "Information security management needs more holistic approach: A literature review". In: *International Journal of Information Management* 36.2 (Apr. 2016), pp. 215–225. ISSN: 0268-4012. DOI: 10.1016/j.ijinfomgt.2015.11.009. URL: https://www.sciencedirect.com/science/article/pii/S0268401215001103.

[103] Sadaf Hina and P. Dhanapal Durai Dominic. "Information security policies' compliance: a perspective for higher education institutions". en. In: *Journal of Computer Information Systems* 60.3 (May 2020), pp. 201–211. ISSN: 0887-4417, 2380-2057. DOI: 10.1080/08874417.2018.1432996. URL: https://www.tandfonline.com/doi/full/10.1080/08874417.2018.1432996 (visited on 10/18/2022).

[104] Bernhard Riedl et al. "A secure architecture for the pseudonymization of medical data". In: *The Second International Conference on Availability, Reliability and Security (ARES'07)*. 2007, pp. 318–324. DOI: 10.1109/ARES.2007.22.

[105] Muneeb Ahmed Sahi et al. "Privacy Preservation in e-Healthcare Environments: State of the Art and Future Directions". In: *IEEE Access* 6 (2018), pp. 464–478. DOI: 10.1109/ACCESS.2017.2767561.

[106] José Luis Fernández-Alemán et al. "Security and privacy in electronic health records: A systematic literature review". In: *Journal of Biomedical Informatics* 46.3 (June 2013), pp. 541–562. ISSN: 1532-0464. DOI: 10.1016/j.jbi.2012.12.003. URL: https://www.sciencedirect.com/science/article/pii/S1532046412001864.

[107] Sérgio Luís Ribeiro and Emilio Tissato Nakamura. "Privacy Protection with Pseudonymization and Anonymization In a Health IoT System: Results from OCARIoT". In: *2019 IEEE 19th International Conference on Bioinformatics and Bioengineering (BIBE)*. 2019, pp. 904–908. DOI: 10.1109/BIBE.2019.00169.

[108] Bipin Kumar Rai. "Pseudonymization Techniques for Providing Privacy and Security in EHR". In: *International Journal of Emerging Trends & Technology in Computer Science* 5 (Aug. 2016).

[109] Chris Greamo and Anup Ghosh. "Sandboxing and Virtualization: Modern Tools for Combating Malware". In: *IEEE Security & Privacy* 9.2 (2011), pp. 79–82. DOI: 10.1109/MSP.2011.36.

[110] Emily Leckenby et al. "The Sandbox Approach and its Potential for Use in Health Technology Assessment: A Literature Review". en. In: *Applied Health Economics and Health Policy* 19.6 (Nov. 2021), pp. 857–869. ISSN: 1175-5652, 1179-1896. DOI: 10.1007/s40258-021-00665-1. URL: https://link.springer.com/10.1007/s40258-021-00665-1 (visited on 10/18/2022).

[111] Mohsen Damshenas, Ali Dehghantanha, and Ramlan Mahmoud. "A survey on malware propagation, analysis, and detection". English. In: *International Journal of Cyber-Security and Digital Forensics* 2.4 (Oct. 2013). 10, pp. 10+. ISSN: 23050012. URL: https://link.gale.com/apps/doc/A359172420/AONE?u=ucwinch&sid=googleScholar&xid=0b805b60 (visited on 10/18/2022).

[112] Waldo Delport, Michael Kohn, and Martin Olivier. *Isolating a cloud instance for a digital forensic investigation.* Jan. 2011.

[113] Washington Henrique Carvalho Almeida et al. "Survey on microservice architecture-security, privacy and standardization on cloud computing environment". In: *ICSEA 2017* (2017), p. 210.

[114] Tetiana Yarygina and Anya Helene Bagge. "Overcoming Security Challenges in Microservice Architectures". In: *2018 IEEE Symposium on Service-Oriented System Engineering (SOSE)*. 2018, pp. 11–20. DOI: 10.1109/SOSE.2018.00011.

[115] Ramaswamy Chandramouli. *Security strategies for microservices-based application systems*. Tech. rep. NIST SP 800-204. Gaithersburg, MD: National Institute of Standards and Technology, Aug. 2019, NIST SP 800–204. DOI: 10.6028/NIST.SP.800-204. URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-204.pdf (visited on 10/27/2022).

[116] Fatima Salahdine and Naima Kaabouch. "Social Engineering Attacks: A Survey". In: *Future Internet* 11.4 (2019). ISSN: 1999-5903. DOI: 10.3390/fi11040089. URL: https://www.mdpi.com/1999-5903/11/4/89.

[117] Katharina Krombholz et al. "Advanced social engineering attacks". In: *Special Issue on Security of Information and Networks* 22 (June 2015), pp. 113–122. ISSN: 2214-2126. DOI: 10.1016/j.jisa.2014.09.005. URL: https://www.sciencedirect.com/science/article/pii/S2214212614001343.

[118] Hassan Chizari et al. "Social Engineering Attack Mitigation". In: *International Journal of Mathematics and Computational Science* 1 (Jan. 2015), pp. 188–198.

[119] Harry Hochheiser. "Principles for privacy protection software". In: *Proceedings of the tenth conference on Computers, freedom and privacy: challenging the assumptions*. 2000, pp. 69–72.

[120] Irit Hadar et al. "Privacy by designers: software developers' privacy mindset". en. In: *Empirical Software Engineering* 23.1 (Feb. 2018), pp. 259–289. ISSN: 1382-3256, 1573-7616. DOI: 10.1007/s10664-017-9517-1. URL: http://link.springer.com/10.1007/s10664-017-9517-1 (visited on 10/18/2022).

[121] G. McGraw. "Software security". In: *IEEE Security & Privacy* 2.2 (2004), pp. 80–83. DOI: 10.1109/MSECP.2004.1281254.

[122] Russell L. Jones and Abhinav Rastogi. "Secure Coding: Building Security into the Software Development Life Cycle". en. In: *Information Systems Security* 13.5 (Nov. 2004), pp. 29–39. ISSN: 1065-898X, 1934-869X. DOI: 10.1201/1086/44797.13.5.20041101/84907.5. URL: http://www.tandfonline.com/doi/full/10.1201/1086/44797.13.5.20041101/84907.5 (visited on 10/20/2022).

[123] B. Potter and G. McGraw. "Software security testing". In: *IEEE Security & Privacy* 2.5 (2004), pp. 81–85. DOI: 10.1109/MSP.2004.84.

[124] Naresh vurukonda and B. Thirumala Rao. "A Study on Data Storage Security Issues in Cloud Computing". In: *2nd International Conference on Intelligent Computing, Communication & Convergence, ICCC 2016, 24-25 January 2016, Bhubaneswar, Odisha, India* 92 (Jan. 2016), pp. 128–135. ISSN: 1877-0509. DOI: 10.1016/j.procs.2016.07.335. URL: https://www.sciencedirect.com/science/article/pii/S1877050916315812.

[125] Aamir Syed, Keerthana Purushotham, and Ganeshayya Shidaganti. "Cloud Storage Security Risks, Practices and Measures: A Review". In: *2020 IEEE International Conference for Innovation in Technology (INOCON)*. 2020, pp. 1–4. DOI: 10.1109/INOCON50539.2020.9298281.

[126] Walt Hubis and Eric Hibbard. "IEEE Standard for Cryptographic Protection of Data on Block-Oriented Storage Devices". In: *IEEE Std* (), pp. 1619–2018.

[127] Cong Wang et al. "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing". In: *2010 Proceedings IEEE INFOCOM*. 2010, pp. 1–9. DOI: 10.1109/INFCOM.2010.5462173.

[128] D. Thain et al. "The consequences of decentralized security in a cooperative storage system". In: *Third IEEE International Security in Storage Workshop (SISW'05)*. 2005, 12 pp.–82. DOI: 10.1109/SISW.2005.11.

[129] AEleen Frisch. *Essential system administration*. 3rd ed. Beijing ; Sebastopol, CA: O'Reilly, 2002. ISBN: 978-0-596-00343-2.

[130] Gregory B. White, Eric A. Fisch, and Udo W. Pooch. *Computer System and Network Security*. en. Ed. by Gregory B. White, Eric A. Fisch, and Udo W. Pooch. 1st ed. CRC Press, Dec. 2017. ISBN: 978-1-315-14006-3. DOI: 10.1201/9781315140063. URL: https://www.taylorfrancis.com/books/9781351458726 (visited on 10/18/2022).

[131] Quey-Jen Yeh and Arthur Jung-Ting Chang. "Threats and countermeasures for information system security: A cross-industry study". In: *Information & Management* 44.5 (July 2007), pp. 480–491. ISSN: 0378-7206. DOI: 10.1016/j.im.2007.05.003. URL: https://www.sciencedirect.com/science/article/pii/S0378720607000523.

[132] Gaurav Aggarwal et al. "An analysis of private browsing modes in modern browsers". In: *19th USENIX Security Symposium (USENIX Security 10)*. 2010.

[133] Peter Snyder, Cynthia Taylor, and Chris Kanich. "Most Websites Don't Need to Vibrate: A Cost-Benefit Approach to Improving Browser Security". In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. CCS '17. event-place: Dallas, Texas, USA. New York, NY, USA: Association for Computing Machinery, 2017, pp. 179–194. ISBN: 978-1-4503-4946-8. DOI: 10.1145/3133956.3133966. URL: https://doi.org/10.1145/3133956.3133966.

[134] Steven Englehardt and Arvind Narayanan. "Online Tracking: A 1-Million-Site Measurement and Analysis". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. CCS '16. event-place: Vienna, Austria. New York, NY, USA: Association for Computing Machinery, 2016, pp. 1388–1401. ISBN: 978-1-4503-4139-4. DOI: 10.1145/2976749.2978313. URL: https://doi.org/10.1145/2976749.2978313.

[135] Nikos Virvilis et al. "Security Busters: Web browser security vs. rogue sites". In: *Computers & Security* 52 (July 2015), pp. 90–105. ISSN: 0167-4048. DOI: 10.1016/j.cose.2015.04.009. URL: https://www.sciencedirect.com/science/article/pii/S0167404815000590.

[136] Douglas J. Leith. "Web Browser Privacy: What Do Browsers Say When They Phone Home?" In: *IEEE Access* 9 (2021), pp. 41615–41627. DOI: 10.1109/ACCESS.2021.3065243.

[137] Nataliia Bielova. "Web Tracking Technologies and Protection Mechanisms". In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. CCS '17. event-place: Dallas, Texas, USA. New York, NY, USA: Association for Computing Machinery, 2017, pp. 2607–2609. ISBN: 978-1-4503-4946-8. DOI: 10.1145/3133956.3136067. URL: https://doi.org/10.1145/3133956.3136067.

[138] Georg Merzdovnik et al. "Block Me If You Can: A Large-Scale Study of Tracker-Blocking Tools". In: *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*. 2017, pp. 319–333. DOI: 10.1109/EuroSP.2017.26.

[139] Ninghui Li, J.C. Mitchell, and W.H. Winsborough. "Design of a role-based trust-management framework". In: *Proceedings 2002 IEEE Symposium on Security and Privacy*. 2002, pp. 114–130. DOI: 10.1109/SECPRI.2002.1004366.

[140] E. Yuan and J. Tong. "Attributed based access control (ABAC) for Web services". In: *IEEE International Conference on Web Services (ICWS'05)*. 2005, p. 569. DOI: 10.1109/ICWS.2005.25.

[141] Vincent C. Hu et al. "Attribute-Based Access Control". In: *Computer* 48.2 (2015), pp. 85–88. DOI: 10.1109/MC.2015.33.

[142] Daniel Servos and Sylvia Osborn. "Current Research and Open Problems in Attribute-Based Access Control". In: *ACM Computing Surveys* 49 (Jan. 2017). DOI: 10.1145/3007204.

[143] A. Machanavajjhala et al. "L-diversity: privacy beyond k-anonymity". In: *22nd International Conference on Data Engineering (ICDE'06)*. 2006, pp. 24–24. DOI: 10.1109/ICDE.2006.1.

[144] Tiancheng Li et al. "Slicing: A New Approach for Privacy Preserving Data Publishing". In: *IEEE Transactions on Knowledge and Data Engineering* 24.3 (2012), pp. 561–574. DOI: 10.1109/TKDE.2010.236.

[145] Atul Kumar, Manasi Gyanchandani, and Priyank Jain. "A comparative review of privacy preservation techniques in data publishing". In: *2018 2nd International Conference on Inventive Systems and Control (ICISC)*. 2018, pp. 1027–1032. DOI: 10.1109/ICISC.2018.8398958.

[146] Josue Alejandro Diaz-Rojas et al. "Web API Security Vulnerabilities and Mitigation Mechanisms: A Systematic Mapping Study". In: *2021 9th International Conference in Software Engineering Research and Innovation (CONISOFT)*. San Diego, CA, USA: IEEE, Oct. 2021, pp. 207–218. ISBN: 978-1-66544-361-6. DOI: 10.1109/CONISOFT52520.2021.00036. URL: https://ieeexplore.ieee.org/document/9653437/ (visited on 11/09/2022).

[147] Yupeng Hu et al. "Artificial Intelligence Security: Threats and Countermeasures". In: *ACM Comput. Surv.* 55.1 (Nov. 2021). Place: New York, NY, USA Publisher: Association for Computing Machinery. ISSN: 0360-0300. DOI: 10.1145/3487890. URL: https://doi.org/10.1145/3487890.

[148] Colin M. Gray et al. "The Dark (Patterns) Side of UX Design". In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. event-place: Montreal QC, Canada. New York, NY, USA: Association for Computing Machinery, 2018, pp. 1–14. ISBN: 978-1-4503-5620-6. DOI: 10.1145/3173574.3174108. URL: https://doi.org/10.1145/3173574.3174108.

[149] Department of Software Engineering University of Science and Technology Bannu, 28100 Pakistan et al. "Comparison of Requirement Prioritization Techniques to Find Best Prioritization Technique". In: *International Journal of Modern Education and Computer Science* 7.11 (Nov. 2015), pp. 53–59. ISSN: 20750161, 2075017X. DOI: 10.5815/ijmecs.2015.11.06. URL: http://www.mecs-press.org/ijmecs/ijmecs-v7-n11/v7n11-6.html (visited on 12/09/2022).

[150] Louis Anthony TonyCox. "What's Wrong with Risk Matrices?: **What's Wrong with Risk Matrices?**" en. In: *Risk Analysis* 28.2 (Apr. 2008), pp. 497–512. ISSN: 02724332. DOI: 10.1111/j.1539-6924.2008.01030.x. URL: https://onlinelibrary.wiley.com/doi/10.1111/j.1539-6924.2008.01030.x (visited on 12/08/2022).

[151] Eric Wohlgethan. *Supporting Web Development Decisions by Comparing Three Major JavaScript Frameworks: Angular, React and Vue.js*. 2018. URL: https://reposit.haw-hamburg.de/handle/20.500.12738/8417.

[152] Shaumik Daityari. *Angular vs React vs Vue: Which Framework to Choose in 2022*. Aug. 2022. URL: https://www.codeinwp.com/blog/angular-vs-vue-vs-react/.

Glossary

ABAC Attribute-Based Access Control. 15, 54, 55, 57

ACL Access Control List. 43

AES Advanced Encryption Standard. 33, 46

AI Artificial Intelligence. 2–4, 6, 10, 13, 17, 19, 20, 57, 58, 62, 66–71, 74, 92, 135, 141

API Application Programming Interface. 13, 17, 18, 27, 51, 57, 125

AUC Area Under the ROC curve. 20

backend A functionality unknown to the service provider, the developers and the legitimate user which gives secret, and otfen complete, access to a piece of software. 34, 40, 41, 53, 58, 96, 97

big data Refers to information resources whose characteristics in terms of volume, velocity and variety require the use of adapted technologies. They generally exceed the capacities of a single machine and often require parallel processing. 2, 15, 135

blockchain A dynamic list of data grouped in blocks linked by cryptography mechanisms. Each block brings additional data to the chain and validates the integrity of the previous bloc. 14

CAF Cyber Assessment Framework. 67, 68

CAPTCHA Completely Automated Public Turing test to tell Computers and Humans Apart. 21

CIA Integrity, Authenticity, and Confidentiality. 35, 44

CISA Cybersecurity & Infrastructure Security Agency. 6, 69, 70, 92

CLI Command Line Interface. 105

cloud An architectural model where computing resources are made available to the user without direct active management. Usually describes data centres that are accessible to many users over the Internet. 2, 5, 17, 20, 25–28, 33, 35, 39, 41, 47–51, 85

CORS Cross Origin Resource Sharing. 57

CPU Central Processing Unit. 47

CSS Cascading Style Sheets. 121, 130

CVE Common Vulnerabilities and Exposures. 40

CWF Critical Watch Factors. 38

DAC Discretionary Access Control. 43, 54

DDoS Distributed Denial of Service. 18, 26–28, 41, 57

DES Data Encryption Standard. 33

differential privacy A protection technique applied on results of queries made to a database. The goal is to minimize the risks of identifying the entities the database contains. A secondary goal is to also maximising the relevance of the results. 16, 19–21, 24, 58, 59

DKIM DomainKeys Identified Mail. 35, 36

DL Deep Learning. 2, 3, 10, 13, 19–21

DPI Deep Packet Inspection. 42

DREAD Damage, Reproducibility, Exploitability, Affected users, Discoverability. 57

DSA Digital Signature Algorithm. 34

ECC Elliptic Curve Cryptography. 34

EU European Union. 35, 39

federated learning A distributed and collaborative ML method were the training phase of the model is done on the device of different users of an application. The results are then shared to each users to benefit from all users' knowledge without knowing their data. 19, 21

FIPP Fair Information Practice/Privacy Principles. 50

framework In the software industry, a framework is a set of software components used to bring new possibilities for the development of an IT tool, especially by providing elements related to its infrastructure. Outside of the ICT field, it describes a set of rules, practices or must-haves in order to achieve a goal. 3, 5, 10, 20, 22, 23, 26, 29, 32, 41, 48, 51, 58, 66, 67, 96, 97, 121, 125, 131, 140, 142

frontend The part of a software program that includes its graphical presentation layer, which is presented to users. 105

GASP Guide to Assess Security and Privacy. 92, 135, 235

GDPR General Data Protection Regulation. 5, 15–17, 20, 39

gradient The direction of the greatest change of a scalar function. Useful to train AI models with their activation functions. 19, 20, 58

GUI Graphical User Interface. 14, 125

HCI Human-computer interaction. 14

HEIA-fr School of Engineering and Architecture of Fribourg. 6

homomorphic encryption An encryption made on data that protect them during its whole lifetime: at rest, in transit and while being processed. Use homomorphism operations. 19, 20, 33, 59

honeypot A computer defense method that draws attackers to spcific resources to identify and eventually neutralize them. These devices are specifically designed to fool attackers. 62

HTML HyperText Markup Language. 36, 127, 131

HTTP HyperText Transfer Protocol. 18, 118

HTTPS HTTP Secure. 57, 118

hyperparameters In the AI field, refer to the configuration that is not directly linked with the model but still used during the process. They are not inferred from data. 59

IBAC Identity Based Access Control. 54

ICT Information and Communications Technologies. vii, 2, 4, 5, 10, 11, 14, 29, 39, 58, 63, 65, 67–71, 78, 79, 81, 92, 141

IoT Internet of Things. 14, 34, 67

IP Internet Protocol. 27, 36, 41

ISO International Organization for Standardization. 5, 24, 44

JSON JavaScript Object Notation: a file format widely used for sending information between different information systems. Its structure is very simple and is easily readable by humans. 57, 89, 101–104, 107–110, 116, 117, 119, 232–236

LBAC Lattice Based Access Control. 54

Likert scale A type of scales that define an equally-distributed set of responses to a question in terms of distances. It includes a neutral answer and the same amount of answers for both directions of opinions other that neutral. 83

LINDDUN Linkability, Identifiability, Non-repudiation, Detectability, Disclosure of information, Unawareness, Non-compliance. 36

Linux Also known as GNU is Not UNIX (GNU)/Linux. A family of open source UNIX-like OSes based on the Linux kernel. Numerous Linux distributions have integrated this Linux in their structure and can be used for a wide variety of tasks. 40, 42–44

loss function A function used as a criterion for determining the best solution to an optimisation problem by associates a value with an instance of an optimisation problem. The optimisation problem must minimise or maximise this function to its optimum using algorithms. 20, 21

MAC Mandatory Access Control. 42–44, 54

metadata Any information related to other data, apart from the content of the data itself. Example, the time and location of a photograph.. 30, 33, 36

microservice An architectural style which structures applications as a set of services. A microservice must follow specific rules in order to be efficient as conceptualized, and to be considered as one. 47, 48

ML Machine Learning. 2, 3, 10, 13, 17–20, 57, 62

MoSCoW Must-have, Should-have, Could-have, Won't-have. 82

MPC Multi-Party Computation. 59

mTLS mutual TLS. 48

NCSC National Cyber Security Centre. 35, 37, 67, 70, 92

NIST National Institute of Standards and Technology. 5, 6, 33, 34, 48, 54, 55, 57, 67, 70, 92

NN Neural Network. 2, 19, 59

NPM Node Package Manager (unofficial name). 31, 125

OAuth An authorization protocol used to authorize access to resources without providing credentials. Its 2.0 version is the current industry-standard. 23, 48, 57

OECD Organisation for Economic Co-operation and Development. 29

OpenID Open standard used for decentralized authentication for websites, using third-party identity providers. 57

open-source The fact to freely share the source code used to build a program to the public, generally associated with a licence. It is also a tech and social movement. 37, 40, 43

OS Operating System. 13, 26, 32, 34, 40–43, 101

OSSTMM Open Source Security Testing Methodology Manual. 57

OTP One Time Password. 22

OWASP Open Web Application Security Project. 5, 30, 57

OWL Web Ontology Language. 23

OWL-S Web Ontology Language Semantic. 23

PaaS Platform as a Service. 28

parameters In the AI field, refers to the characteristics of the training data that will be learned during the learning phase. For DL, they include the model weight and bias. 19, 59

PbD Privacy by Design. 49

PDF Portable Document Format. 6, 67, 69, 75–77, 93, 131, 143, 232

PGP Pretty Good Privacy. 35

PICO Patient-Intervention-Comparison-Outcome(s). 4

PKI Public Key Infrastructure. 48

plaintext A term used to describe any data that is not encrypted and readable by an information system or a human. 19

PPDP Platform Security Processor. 24

proxy A program that acts as an intermediary to access another network by placing itself between two hosts to facilitate or monitor their exchanges. 53

PWA Progressive Web Application. 118, 119, 131, 232, 235

RBAC Role Based Access Control. 54, 57

RDFS Resource Description Framework Schema. 23

regularization A process that adds information to a problem if it is badly posed or to avoid an overfitting model.. 20, 58

REST A set of guidelines for the architecture of an API, to ensure interoperability between information systems on the Internet. 17, 57

RFC Request for Comments. 23

ROC Receiver Operating Characteristic. 20

RSA Rivest-Shamir-Adleman. 34

RSS Really Simple Syndication. 38

SASS Syntactically Awesome Style Sheets. 130

Scrum Project management approach that aims to divide the work into sprints, with tasks defined by the team. 6

SDLC Software Development Life Cycle. 50, 61, 74

SELinux Security-Enhanced Linux. 44

SIM Subscriber Identity/Identification Module. 41

SLA Service Level Agreement. 18, 50

SMART Specific, Measurable, Achievable, Relevant, Time-bound. 79

SMS Short Message Service. 40

SOA Service Oriented Architectures. 54

SOAP Simple Object Access Protocol. 17, 57

SoC System on a Chip. 34

SP Special Publication. 54, 55

SPF Sender Policy Framework. 35

SQL Structured Query Language. 28, 57

Stochastic Gradient Descent An iterative method for optimizing an objective function with suitable smoothness properties. It can be interpreted as an approximation of the general gradient descent optimization. 19

STRIDE Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege. 36, 57

SWOT Strengths, Weaknesses, Opportunities and Threats. 38

TEE Trusted Execution Environment. 59

TLS Transport Layer Security. 35, 48, 57

token A token is a string of characters issued to users by an entity, allowing it to identify the authors of queries within their information system. In general, a token is unique and can be revoked. It is a kind of label, rarely in human-readable form. 22, 57

toolkit Refers to an utility program, several software capabilites or a complete integrated set of software utilities which can be used to develop and maintain applications. 6, 69, 70

Tor A global, decentralised, overlay computer network. Composed of servers (called network nodes) which are publicly listed. 16

TPM Trusted Platform Module. 25

TypeScript A superset put on top of the JavaScript language that allow static types, among other functionalities. 97, 105, 106, 108–111, 119, 121, 131, 237

UI User Interface. 28, 97, 98, 121, 124, 130, 131, 138, 141, 142

UNIX A family of multi-tasking, multi-user OSes derived from the original Unix created by *AT&T* in the 1970s. It is based on an interpreter or supervisor named the shell and small utilities, each performing a specific action. 43

UX User eXperience. 14, 60, 76, 98, 121, 130, 138, 141

VM Virtual Machine. 27, 28, 47

VTX Intel virtualization. 47

WebGL A JavaScript API used by various platforms but mainly in browsers. It allows to create two or three dimensional graphics. 52

XML eXtensible Markup Language. 28, 57, 89

Appendices

This Chapter provides all the appendixes we mentioned during our thesis. Their original files can be accessed on the thesis repository [1] as well.

Appendix A: Specifications Document

The next twenty-nine pages present the specifications document that has been defined for this thesis.

MASTER OF SCIENCE IN ENGINEERING

Hes·so
Haute Ecole Spécialisée de Suisse occidentale
Fachhochschule Westschweiz
University of Applied Sciences and Arts Western Switzerland

Master of Science HES-SO in Engineering
Av. de Provence 6
CH-1007 Lausanne

Master of Science HES-SO in Engineering

Orientation: Information and Communication Technologies (ICT)

Provide Secured Environment for AI Projects
Specification document

Author
Loïc GUIBERT

Under the direction of
Dr. Pascal BRUEGGER
HES-SO//Master, iCoSys
and
Dr. Adriana WILDE
University of Winchester, WINTS

Winchester, HES-SO//Master, 29th September 2022

Contents

List of Figures

1 Introduction

This specifications document describes all the elements necessary to understand the characteristics of the Master Thesis that takes place in the *HES-SO//Master* curriculum.

A Master thesis takes place at the end of all the Master lectures during a whole semester. An amount of nine hundred hours must be dedicated to this end and at least one weekly meeting must be organized through the whole project duration.

This thesis will be realized in collaboration between the *HES-SO//Master* and the *University of Winchester*. The student will realize his thesis abroad, at the city of Winchester.

In this document will be explained the context, objectives, activities and planning of the thesis.

All the documents produced during this project will be stored on the School of Engineering and Architecture of Fribourg (HEIA-fr) software forge[1].

2 Context

This chapter will describe the context around the subject of the thesis. We will explain the actual state that contains a problem to be solved, and the author's contribution that will contribute to fix the previously described problem. We will also introduce the different actors that will intervene during the thesis.

2.1 Actual State

In a world where Informational Technologies (IT) security is increasingly valuable and necessary, the need for new ways to secure and trust information systems is growing. This need is particularly expressed in the Artificial Intelligence (AI) field, where big amounts of data are periodically collected in order to improve services performances or to monetize them. Furthermore, such data is often personal and highly related to their user, which raise ethical and privacy-related questions.

Nowadays, end users of public or private online services are more and more aware of personal data related risks and a change in consumption patterns is being noticed. Therefore, new approaches for the whole IT field must be developed in order to provide secured and privacy-first online services that can nevertheless enable a personalized experience.

2.2 Contribution

By enabling secured and privacy-oriented personalized experience on online services, companies would be able to provide ethical, modern and respectful offers to their customers. All stakeholders would benefit from such implementations, as long as the performance, time or processing capabilities do not restraint them in their activities.

This thesis aims to provide which technologies, best practices or safeguards can be integrated to information systems in order to ensure secured environments to the end users, particularly regarding AI projects. These components must then be implemented in a functional information system, which includes AI processes, while providing a conclusion on the changes of such integration compared to the initial information system.

We want to offer a suitable solution to those wishing to increase the security and confidentiality of their online services. A focus will also be made on the AI field.

2.3 Actors

The thesis will be conducted by Loïc Guibert.

There are two advisors for this thesis: Dr. Pascal Bruegger and Dr. Adriana Wilde.

The thesis subject has been proposed by Loïc Guibert as a personal project and is related to the field of research of the two advisors. A secured online service is being developed by Dr. Pascal Bruegger and includes aspects similar to this thesis subject.

An expert will be assigned to the thesis in order to evaluate it when completed and returned. This assignment will be made later during the semester.

3 Objectives

Following the enumeration made in 2.2 (Contribution), questions need to be asked in order to complete this project.

- Which rules, best practices, technologies and aspects should be used in order to improve the security and confidentiality of online services?
- Which specific rules, best practices, technologies and aspects should be used in order to improve the security and confidentiality of the AI field, particularly for Machine Learning (ML) and Deep Learning (DL) models?
- How can we provide an understandable and complete model of our findings?

The objectives below intent to provide answers to the questions asked.

3.1 Primary Objectives

All those objectives must be fulfilled in order to successfully complete the thesis.

3.1.1 Establish an Up-To-Date Knowledge Collection

We need to collect an up-to-date and complete knowledge of the rules, best practices, technologies and aspects that contribute to enforce the security and privacy of online services. To this end, a proper literature review must be conducted. Its results will then be used as a foundation for the next phase of the thesis.

The content related to this objective must be included into the thesis report. It will list, explain and analyse the concepts found during the collection.

This objective must be completed before the end of the thesis, which is the 10th February 2023.

3.1.2 Provide an Understandable Guide or Framework

Using the previously collected knowledge about security and privacy, a guide or framework must be built. It must provide an understandable and appliable methodology in order to evaluate online services.

The content related to this objective must be included into the thesis report. It will list, explain and analyse the methodology used to build the guide or framework and its content. The result of this objective will be an independent document, which must be attached to the thesis report as an appendix.

This objective must be completed before the end of the thesis, which is the 10th February 2023.

3.1.3 Apply and Test the Guide or Framework on an Online Service

Proper metrics must be used in order to conduct a proper evaluation of the guide or framework. Once defined and explained, an evaluation of an online service including one or more AI processes will be made, and the results will then be analysed and discussed.

The online service should be accessible and open, which means that we should have access to the source code. To this end, we could use the *Hestia*[1] ecosystem, which is an ongoing project leaded by Dr. Pascal Bruegger.

The content related to this objective must be included into the thesis report. It will explain and analyse the results obtained during the evaluation.

This objective must be completed before the end of the thesis, which is the 10th February 2023.

3.2 Secondary Objectives

The secondary objective will only be completed if all primary objectives have been fulfilled and if there is still time left before the end of the project's timeframe.

3.2.1 Submit the Guide as an Online Resource

Because of the nature of this thesis being an academical work and the ethics of such approaches, we believe that publishing the guide or framework on the Internet would be useful to the public. It should therefore be published on some online platforms, such as a custom website, a forum or something equivalent.

The content related to this objective must be included into the thesis report. It will explain and analyse the approaches made in order to publish the guide or framework online.

If this objective is considered, it must be completed before the end of the thesis, which is the 10th February 2023.

3.3 Constraints

Apart from the ones described in the respective objectives, an additional constraint must be respected regarding the whole thesis scope.

If any data collected under real-life conditions is used, analysed or processed during this thesis, ethical considerations and obligations must be applied.

[1] *Hestia* source: `https://bit.ly/3BzeDSN` (accessed 22nd September 2022)

4 Activities

The previously defined objectives imply several work activities in order to fulfil them. These activities are separated into more specific tasks. Please note that those tasks will not necessarily be done in a chronological order.

4.1 Specifications

This activity defines the scope and objectives of the thesis.

1. Define the project specification
2. Define the project planning

4.2 State of the Art

This activity covers the 3.1.1 (Establish an Up-To-Date Knowledge Collection) objective.

1. Gather scientific articles and papers
2. Gather online resources
3. Gather standards
4. Gather technologies
5. Assess, analyse and explain the relevant resources

4.3 Methodology

This activity partially covers the 3.1.2 (Provide an Understandable Guide or Framework) objective.

1. Define and explain the guide or framework chosen format
2. Gather resources for the chosen format
3. Prepare the document

4.4 Build the Guide or Framework

This activity partially covers the 3.1.2 (Provide an Understandable Guide or Framework) objective.

1. Define the categories
2. Define the items to evaluate
3. Define the evaluation process

4.5 Test and Evaluation

This activity covers the 3.1.3 (Apply and Test the Guide or Framework on an Online Service) objective.

1. Define the proper metrics to evaluate the guide of framework
2. Get access to an online service
3. Apply the methodology previously defined
4. Evaluate and explain the results

4.6 Publish the Guide or Framework

This activity covers the 3.2.1 (Submit the Guide as an Online Resource) secondary objective.

1. Assess the suitable online platforms
2. Publish the document on selected platform(s).

4.7 Documentation

The thesis report will be realized throughout the duration of the whole project, but this activity will be conducted at its end in order to change last details in said report. A poster must also be produced.

1. Produce the thesis poster
2. Finalize the thesis report

5 Planning

Around nine hundred hours of work are required for this thesis: this amount depends on the number of indicated credits.

The public holidays are taken into account: one week of break has been planned during winter holidays. Weekends are not included into the planning but can be used to catch up possible delays.

We will work with the Scrum method, which is an iterative agile method. The duration of the sprints is of one week excepted for exceptions due to special occasions (sprints number 1, 13, 14 and 18), with a backlog definition made at each weekly meeting. The tasks defined in 4 (Activities) can be directly included into the backlog or can be divided into smaller tasks.

The major observed differences will be notified into meeting minutes.

Figure 5.1 (Gantt planning of the thesis) shows a graphical view of the corresponding planning.

5.1 First Sprint (1st)

Timeline:

- Start - 19.09.2022
- End - 28.09.2022

Activity to complete:

- 4.1 (Specifications)

Deliverable:

- Specifications document

5.2 Second Sprint (2nd)

Timeline:

- Start - 29.09.2022
- End - 05.10.2022

Activity to complete:

- 4.2 (State of the Art)

Deliverable: none

5.3 Third Sprint (3rd)

Timeline:

- Start - 06.10.2022
- End - 12.10.2022

Activity to complete:

- 4.2 (State of the Art)

Deliverable: none

5.4 Fourth Sprint (4th)

Timeline:

- Start - 13.10.2022
- End - 19.10.2022

Activity to complete:

- 4.2 (State of the Art)

Deliverable: none

5.5 Fifth Sprint (5th)

Timeline:

- Start - 20.10.2022
- End - 26.10.2022

Activity to complete:

- 4.2 (State of the Art)

Deliverable: none

5.6 Sixth Sprint (6th)

Timeline:

- Start - 27.10.2022
- End - 02.11.2022

Activity to complete:

- 4.2 (State of the Art)

Deliverable: none

5.7 Seventh Sprint (7th)

Timeline:

- Start - 03.11.2022
- End - 09.11.2022

Activity to complete:

- 4.2 (State of the Art)

Deliverables:

- An up-to-date state of the art
- Report content

5.8 Eighth Sprint (8th)

Timeline:

- Start - 10.11.2022
- End - 16.11.2022

Activity to complete:

- 4.3 (Methodology)

Deliverable: none

5.9 Ninth Sprint (9th)

Timeline:

- Start - 17.11.2022
- End - 23.11.2022

Activity to complete:

- 4.3 (Methodology)

Deliverables:

- Defined methodology for the guide or framework
- Report content

5.10 Tenth Sprint (10th)

Timeline:

- Start - 24.11.2022
- End - 30.11.2022

Activity to complete:

- 4.4 (Build the Guide or Framework)

Deliverable: none

5.11 Eleventh Sprint (11th)

Timeline:

- Start - 01.12.2022
- End - 07.12.2022

Activity to complete:

- 4.4 (Build the Guide or Framework)

Deliverable: none

5.12 Twelfth Sprint (12th)

Timeline:

- Start - 08.12.2022
- End - 14.12.2022

Activity to complete:

- 4.4 (Build the Guide or Framework)

Deliverable: none

5.13 Thirteenth Sprint (13th)

Timeline:

- Start - 15.12.2022
- End - 23.12.2022

Activity to complete:

- 4.4 (Build the Guide or Framework)

Deliverable: none

Winter vacations

5.14 Fourteenth Sprint (14th)

Timeline:

- Start - 02.01.2023
- End - 11.01.2023

Activities to complete:

- 4.4 (Build the Guide or Framework)
- 4.5 (Test and Evaluation)

Deliverables:

- Guide or framework
- Report content

5.15 Fifteenth Sprint (15th)

Timeline:

- Start - 12.01.2023
- End - 18.01.2023

Activity to complete:

- 4.5 (Test and Evaluation)

Deliverable: none

5.16 Sixteenth Sprint (16th)

Timeline:

- Start - 19.01.2023
- End - 25.01.2023

Activity to complete:

- 4.5 (Test and Evaluation)

Deliverable: none

5.17 Seventeenth Sprint (17th)

Timeline:

- Start - 26.01.2023
- End - 01.02.2023

Activity to complete:

- 4.5 (Test and Evaluation)

Deliverable: none

5.18 Eighteenth Sprint (18th)

Timeline:

- Start - 02.02.2023
- End - 08.02.2023

Activity to complete:

- 4.5 (Test and Evaluation)

Deliverables:

- Evaluation of an online service
- Report content

5.19 Nineteenth Sprint (19th)

Timeline:

- Start - 09.02.2023
- End - 10.02.2023

Activity to complete:

- 4.7 (Documentation)

Deliverables:

- Master poster
- Completed master thesis

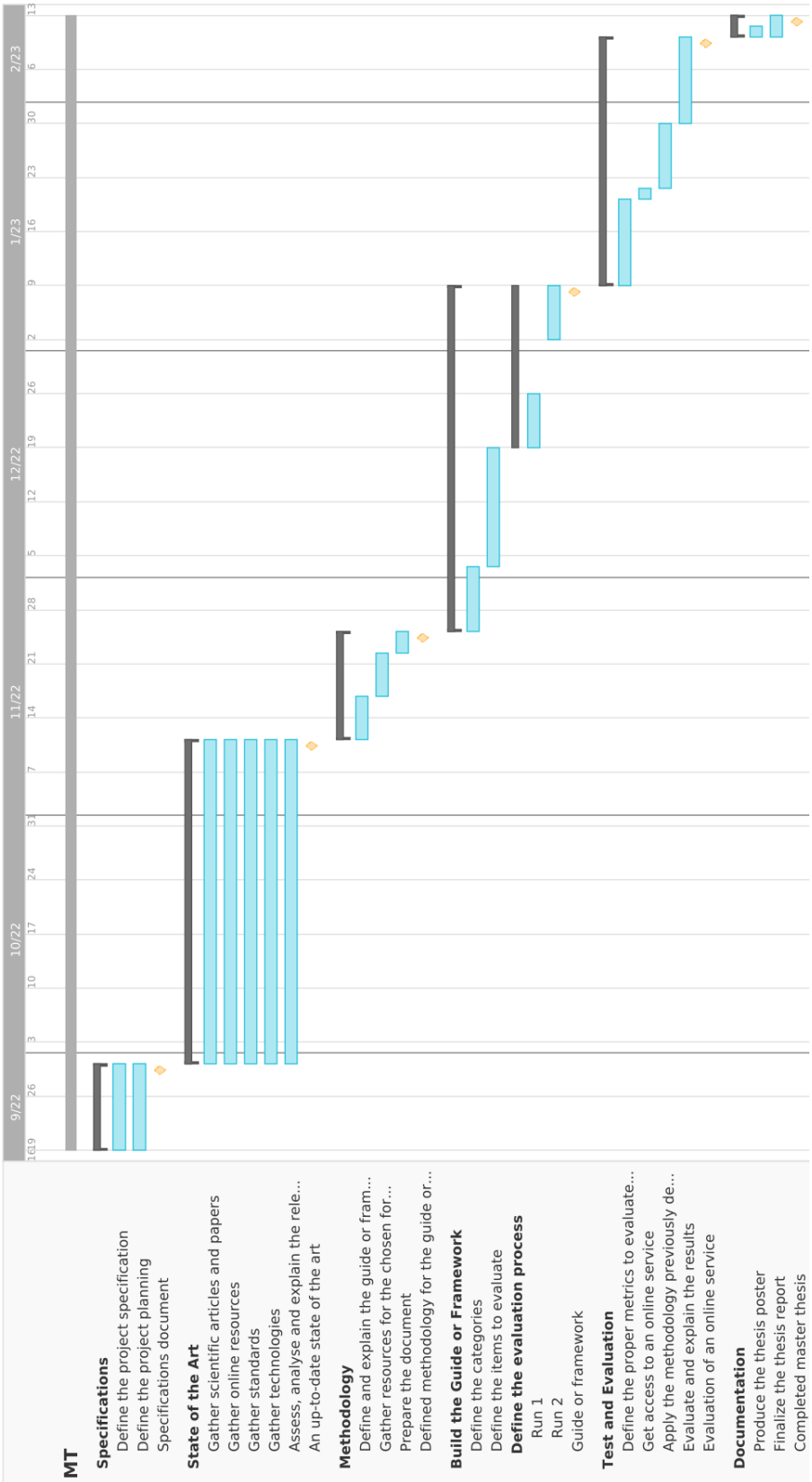


Figure 5.1 Gantt planning of the thesis

A | Appendices

A.1 Project Proposal

The next two pages are this thesis proposal form, submitted to the *University of Winchester*.

UNIVERSITY OF WINCHESTER

BS7205 – MSc Project

Project Proposal Form

Student Name: Loïc Guibert

Student Route:

Project Title: Provide Secured Environments for Artificial Intelligence Projects

Context:

In a world where Informational Technologies (*IT*) security is more and more proven and necessary, the need for new ways to secure and trust information systems is growing. This need is particularly expressed in the Artificial Intelligence (*AI*) field, where big amounts of data are periodically collected in order to improve services performances. Furthermore, such data is often personal and highly related to their user, which raise ethical and privacy-related questions.

Nowadays, end users of public or private online services are more and more aware of the personal data related risks and a change in consumption patterns is being noticed. New approaches must be developed in order to provide secured and privacy-first online services that can nevertheless enable a personalized experience.

Contribution:

By enabling secured and privacy-oriented personalized experience on online services, companies would be able to provide ethical, modern and respectful offers to their customers. All stakeholders would benefit from such an implementation, as long as the performance, time or processing capabilities do not restraint them in their activities.

This thesis aims to provide which technologies, best practices and safeguards can be integrated to information systems in order to ensure secured environments to the end users, particularly regarding *AI* projects. These components must then be implemented in a functional information system which includes *AI* processes, while providing a conclusion on the changes of such integration compared to the initial information system.

Literature Review:

Several emerging technologies that could fulfil the purpose of this thesis were found during preliminary research. Indeed, the academic world has new tools and increasingly permissive computing power, which opens up new possibilities.
Fully Homomorphic Encryption (*FHE*) shemes, that allows systems to process encrypted data without knowing its content, has make significant progress in terms of security, speed, and simplicity [Acar et al., 2017, 10.1145/3214303]. In terms of trustworthiness, a proposal of a new protocol, based on *HTTPS*, named HTTPA [King and Wang, 2021] aims to ensure to users of an online service that their related processes are executed in a trustable and attestable environment. This technology is not yet standardized nor reviewed. Regarding decentralized approaches, Federated Learning techniques could enable users to stay in control of their data by bringing the Machine Learning processes in their devices, which avoid data sharing to centralized systems. In latter researches, several models have been tested in various situations, where some aspects brought by this technique must be handled such as the heterogeneity of the data and the parties [Li et al., 2021, 10.1109/TKDE.2021.3124599]

Research Methodology and Research Design:

Seven steps have been defined in order to reach this thesis objectives. Each of them focuses on a particular subject and will be documented. The first steps will supply an academical view, and the last ones will use this knowledge to suggest a usable proposal.

- Provide an up-to-date literature review
- Analyse the current state-of-the-art
- List and compare related technologies, papers and resources

- Select the most appropriate items
- Build a guide or framework to explain how to evaluate an online service
- Test and validate the guide or framework

Change notification report

The following table is for you to report any changes to your project. After this proposal has been agreed, you should only change the table below – the content above should remain the same. For all changes, you must consult your academic supervisor

Date of Change	Brief Description of Change

Project Proposal Sign Off

I confirm:

a) I have discussed my proposed project with my allocated supervisor
b) I will not collect any data until my ethics application has received a favourable opinion from the appropriate university representatives
c) I will not collect any data without prior agreement from my academic supervisor
d) I will inform my academic supervisor of any changes to this proposal
e) This project is appropriate for my registered programme route

Student signature	
Print name	
Date	

I confirm:

a) This project is academically appropriate for a level 7 qualification
b) This project is appropriate for the route the student is registered to complete

Supervisor signature	
Print name	
Date	

References

[1] *Loïc Guibert / Master Thesis · GitLab*. Loïc Guibert, Sept. 2022. url: https://gitlab.forge.hefr.ch/loic.guibert/mt.

Glossary

AI Artificial Intelligence. 3, 5, 6

DL Deep Learning. 5

framework A framework is a set of software components used to bring new possibilities for the development of an IT tool, especially by providing elements related to its infrastructure. Outside of the IT field, it describes a set of rules, practices or must-haves in order to achieve a goal. 5–8, 12, 14

HEIA-fr School of Engineering and Architecture of Fribourg. 1

IT Informational Technologies. 3

ML Machine Learning. 5

Scrum Project management approach that aims to divide the work into sprints, with tasks defined by the team. 9

Appendix B: Guide Content

The next fifty pages present the spreadsheet file acting as the dataset that has been defined during this thesis.

categories

id	name
1	Data
2	Software
3	System
4	Data Science
5	Environment
6	Management

subcategories

category	id	name	description	PK
1	1	Basic Security	Some mechanisms should be applied by default to protect data from unwanted, unauthorized, or more generally unanticipated actions and access.	1.1
1	2	Authentication	Authentication is an action done by a system that verifies whether the identity given by an entity is valid and trusted or not. Such evaluation must ensure that this identity can not be falsified.	1.2
1	3	User Data	Any data linked to a user should be protected, whether legally for analysis or illegally for knowledge theft. Some user data can also be sensitive. Service providers must ensure that such threats are mitigated.	1.3
2	1	Development	Security and privacy problems are common during the process of analysing, designing, developing, and testing software. Threats can occur from mistakes, external sources, organizational issues, and more. Safeguards must ensure that this process avoid common mistakes.	2.1
2	2	Environments	Software runs in environments which ensure that specific tasks can be performed. These environments include a collection of libraries, utilities, or programs that can lead to security issues.	2.2
3	1	Distributed Systems	Distributed computing consists of sharing data and processes through multiple hosts using a network in order to complete a common task. Such methods must be handled by trusted hosts, even if they do not necessarily hold the whole data knowledge by themselves.	3.1
3	2	Cloud Hosting	A numerous amount of services is hosted in the cloud. This paradigm brings new issues in terms of security, but more particularly for user and company privacies. Indeed, the data storage, transport and processing are made on someone else's computers. A new bond of trust must be established between service providers and cloud providers.	3.2
3	3	Durability	Durability refers to the CIA Triad, an acronym for Confidentiality, Integrity, and Availability. Those three principles must be respected in order to provide users a durable and resilient service.	3.3
3	4	Architecture	The two major and almost two only server architectures are the monolith and the microservice ones. The first architecture is seen as the legacy approach, which consists of building all features into a single program. The second one is a more recent approach that follows specific characteristics. A microservice architecture brings new security issues to be mitigated.	3.4
3	5	Operating Systems	Operating systems are mandatory to handle all the operations in a machine. It brings lots of interfaces for the users, resources and hardware management, and runs programs. The current market includes several competitors: Windows and macOS are leading the personal computer market, Linux and Windows the server market, Android and iOS the mobile market.	3.5
3	6	Network	Connecting services to a network allows remote access for a public or private usage. In all cases, it brings new threats in a system with the possibility for external parties to collect knowledge or to exploit vulnerabilities. Networks must be well configured to avoid those new threats. The difficulty is to restrict the possible actions or accesses as much as possible without disrupting or impacting the services.	3.6

3	7	Hardware	Web services are not really concerned by advanced hardware aspects: they often only use pre-designed servers and client devices without any particular needs, not like Internet of Objects or embedded projects. As a company, the trust placed into vendors and manufacturer must be verified, in order to ensure that they propose legit and audited products. Politics can also interfere in manufacturer processes to enable industry intelligence and surveillance.	3.7
4	1	Big Data	The security concerns of this topic are mainly brought by the distribution model: to process such amount of data, several computers must be used in parallel, which includes this additional task to the whole data usage. By sharing data between computers, users' privacy can also be a concern.	4.1
5	1	User Tracking	User tracking enables organizations to track an Internet user by various means. One of the most used method is by following user habits through browsers. Tracking can be used for marketing, statistical, or commercial purposes.	5.1
5	2	External Tools	Every organization uses various external tools to provide and conceive their services. They can be directly included into processes, used for administrative tasks, or used through the development phases. Those tools bring new threats for both security and privacy levels into an organization.	5.2
6	1	Risk Management	An organization that knows its risks can enforce mitigations in order to strengthen their operations and resilience. Risk management involves several steps related to the activities of an organization. First, risks must be identified, assessed and prioritised. Then, these risks must be dealt methodically in order to know and control both the likelihood and the impact of their related events. Risks can be of different natures and	6.1
6	2	Audits	Audits aim to verify organizations activities to ensure their compliance to known requirements. An audit can be conducted on the entire organization or be specific to a single function, process, or production step. Audits can be of different natures, can be done for different reasons and can expect different goals.	6.2
6	3	Policies	Policies allow organizations to define sets of guidelines that must be respected by every person. More specifically, information security policies help to avoid data breaches, which are often caused by human mistakes. People are considered as the weakest point in organizations. Establishing and enforcing a complete and adapted policy is one of the most effective mitigations to breaches.	6.3
6	4	Best Practices	A best practice is a piece of advice given by a party, which has usually no official status in the concerned field but generally trusted because of its background or by past events. There is no obligation to apply such advices, but doing so is considered better than ignoring them.	6.4
1	4	Authorization	Authorization is a process that verifies an entity access request in order to grant its access to resources, by following rules from the access control policy. The main challenge of authentication is to ensure that every access made to system resources must pass by its verification, without exception.	1.4

subcategories

1	5	Access Control	Broadly speaking, access control is a set of policies that restrict access to virtual or physical resources. A proper access management must also support its implementation. Access control is strongly related to user and administration roles, as to the notion of trust.	1.5
2	3	APIs	An API is an interface used for communication between software. It usually exposes endpoints to realize actions in a given service, using the web. An API can be public or private, authenticated or anonymous. The access of such interfaces can be a problem, both for end users than for the service provider. Sensitive information could be accessed, or unwanted actions could be made by malicious parties.	2.3
4	2	Models	Artificial intelligence models require a big amount of data in order to be accurate in their classifications or predictions. This characteristic includes privacy risks for users, whose data is being collected from service providers in order to improve their models. Various threats exist on their security side, whit attacks that aim to gain knowledge on their behaviour.	4.2

subcateg	id	name	PK
1.1	1	Adapted data protections are planned and enforced.	1.1.1
1.1	2	Data are encrypted appropriately.	1.1.2
1.2	1	The organization accounts accesses are strongly secure.	1.2.1
1.3	1	Pseudonymization is enforced when required.	1.3.1
1.3	2	The service privacy is aligned with user preferences.	1.3.2
1.3	3	Local laws, regulations and obligations are respected.	1.3.3
2.1	1	Systems include an appropriate debugging and logging capability.	2.1.1
2.1	2	Software is delivered with an appropriate testing phase.	2.1.2
2.1	3	Privacy and security concerns are integrated into the development.	2.1.3
2.1	4	The mobile platforms specific threats are mitigated during the development.	2.1.4
2.2	1	Applications and processes are sandboxed.	2.2.1
3.1	1	The distributed systems are compliant with the defined security protocols.	3.1.1
3.2	1	The cloud provider choice is adapted to the use case.	3.2.1
3.2	2	The cloud assets are managed appropriately.	3.2.2
3.1	3	Isolation techniques are applied on the instances.	3.1.3
3.3	1	A back up process is designed and tested.	3.3.1
3.3	2	The service availability is guaranteed.	3.3.2
3.4	1	Security issues specific to the microservice architecture are mitigated.	3.4.1
3.5	1	The operating systems of servers are adapted to the use case.	3.5.1
3.6	1	Network specific security issues are mitigated.	3.6.1
3.6	2	The network is monitored appropriately.	3.6.2
3.7	1	The chosen hardware is secure and trustworthy.	3.7.1
3.7	2	The security mechanisms embedded in hardware pieces are activated and configured.	3.7.2
4.1	1	Big data specific issues are mitigated using appropriate techniques.	4.1.1
5.1	1	The browsers are configured to avoid security and privacy threats.	5.1.1
5.1	2	The user tracking is configured ethically and legitimately.	5.1.2
5.1	3	The mobile providers' user tracking is understood and limited.	5.1.3
5.2	1	Email and instant messaging technologies are set up appropriately.	5.2.1
6.1	1	Reach an adapted level of countermeasures based on the threat levels.	6.1.1
6.1	2	The organization management controls the security and privacy levels.	6.1.2
6.1	3	A technology watch is conducted within the organization.	6.1.3
6.2	1	Auditors have no access to assessed data.	6.2.1

6.3	1	A set of policies is defined within the organization.	6.3.1
6.3	2	A policy to mitigate social engineering attacks is defined.	6.3.2
6.3	3	A policy on workplace management is defined.	6.3.3
6.3	4	The security policies reconciliation is handled when collaborating with another organization.	6.3.4
3.1	2	The storage of distributed systems is defined in a secure way.	3.1.2
2.1	5	The issues brought by software dependencies are understood and mitigated.	2.1.5
3.4	2	A decentralized architecture has been considered.	3.4.2
1.3	4	The personal user data is managed appropriately.	1.3.4
2.1	6	No dark pattern is used on the client applications.	2.1.6
3.2	3	The choice of using the cloud has been well thought.	3.2.3
6.4	1	A set of best practices have been defined within the organization.	6.4.1
1.4	1	Privacy and security concerns are considered when implementing authorization processes.	1.4.1
1.2	2	The user accounts recuperation process is secure.	1.2.2
1.2	3	Concerns brought by biometric systems are mitigated.	1.2.3
1.2	4	Concerns brought by multi-factor authentication are mitigated.	1.2.4
1.5	1	The data access policies are adapted to the needs.	1.5.1
4.1	2	The data access policies take into account the big data particularities.	4.1.2
1.3	5	The data anonymization techniques are applied appropriately.	1.3.5
2.3	1	The API policies are defined appropriately.	2.3.1
2.3	3	Users have the ability to know which of their data are shared and with whom.	2.3.3
2.3	2	The APIs security concerns are handled appropriately.	2.3.2
4.2	1	The artificial intelligence processes respect the regulations.	4.2.1
4.2	2	The users’ privacy is respected.	4.2.2
4.2	3	The machine learning and deep learning models security issues are mitigated.	4.2.3

objecti	id	name	topi	proba	severit	risk	requi	PK	remarks
1.1.1	1	Encrypt data at rest.	SP	4	5	20	m	1.1.1.1	Very recommended, not this hard to apply. Secure data in breaches.
1.1.1	2	Encrypt data in transit.	SP	5	5	25	m	1.1.1.2	Mandatory, impossible not to implement it.
1.1.2	1	Encrypt data using strong and adapted encryption.	SP	4	5	20	m	1.1.2.1	Adapted encryption parameters and ciphers must be used.
1.2.1	1	Enable two-factors authentication on third parties or providers accounts.	S	4	3	12	s	1.2.1.1	Can help to restrict malicious accesses by adding an extra step.
1.3.1	1	Choose adapted techniques for pseudonymization.	P	3	5	15	m	1.3.1.1	Some pseudonymization are done without testing the results. Access to de-pseudonymized data can harm the users.
1.3.1	2	Respect all legal obligations.	SP	3	5	15	m	1.3.1.2	Most known laws are considered, but some aspects / laws / limitations can be missed. Lawsuits have a strong severity.
1.3.1	3	Use multi-level pseudonymization.	P	2	3	6	s	1.3.1.3	Great way of mitigating external third parties surveillance. Limited severity if other protections exist.
1.3.1	4	Use blank pseudo-identities.	P	2	2	4	c	1.3.1.4	To add nice into the data, increase the processing time for intruders.
1.3.2	1	Ask for user permission if access to personal data is needed.	P	3	5	15	m	1.3.2.1	Permissions on specific aspects can be forgotten, and lawsuits have a strong severity on the organization image.
1.3.3	1	Verify and comply with all mandatory laws, regulations and obligations that concern your activities.	SP	5	5	25	m	1.3.3.1	Mandatory step.
1.3.3	2	Handle the biggest challenges on data privacy compliance.	P	4	5	20	m	1.3.3.2	Such challenges are known, so limited probability. But missing one can be very harmful.
2.1.1	1	Collect all the debugging and logging data from software using a centralized tool.	SP	3	4	12	s	2.1.1.1	Big amount of data, can not do that without a tool. Can sometime help to identify problems before they hit.
2.1.1	2	Analyse the errors and irregularities using a centralized tool.	SP	3	4	12	s	2.1.1.2	Big amount of data, can not do that without a tool. Can sometime help to identify problems before they hit.
2.1.1	3	Design applications to improve users' security perceptions.	S	2	2	4	c	2.1.1.3	Make them feel secure.
2.1.2	1	Define a policy for software testing.	S	2	4	8	s	2.1.2.1	To strengthen the development process and avoid later problems.
2.1.2	2	Ensure that all parties follow the software testing policy.	S	2	3	6	s	2.1.2.2	
2.1.2	3	Report bugs, problems and issues to the appropriate parties.	SP	3	3	9	s	2.1.2.3	Allow tracking of problems and make sure they are resolved.
2.1.2	4	Detect flaws by using static analysis tools.	SP	3	3	9	s	2.1.2.4	Complete the developers' tests.
2.1.2	5	Realize penetration testing periodically.	SP	3	4	12	s	2.1.2.5	Before having intruders breaching in, impossible to predict whether they would have a small or big access to the system.
2.1.2	6	Realize functional and non-functional testing periodically.	SP	4	5	20	m	2.1.2.6	Essential to find problems.
2.1.2	7	Realize automated testing periodically.	SP	4	4	16	m	2.1.2.7	Ensure compliance of software before releasing it.
2.1.2	8	Test mobile applications with appropriate techniques.	SP	3	3	9	s	2.1.2.8	Some specificities, not that critical.
2.1.3	1	Apply the FIPP principle.	P	4	4	16	m	2.1.3.1	Simple and quite complete asset, to avoid any major privacy issues in the future.
2.1.3	2	Apply the "privacy by default" principle.	P	4	4	16	m	2.1.3.2	Recognised principle
2.1.3	3	Allow users to make privacy-friendly choices without any penalties.	P	3	4	12	s	2.1.3.3	Can have serious consequences if an application limits users.
2.1.3	4	Integrate security concerns into the entire software life cycle.	S	5	5	25	m	2.1.3.4	To build better code quality and less changes afterwards, proofed as essential to avoid vulnerabilities in the development.
2.1.4	1	Apply appropriate protections for each data cycle.	S	2	3	6	s	2.1.4.1	Mobile devices bring new threats, but they are known and not that serious.
2.2.1	1	Apply the highest level of virtualization possible.	S	3	5	15	m	2.2.1.1	If a component is infected, it should not propagate to others. Low level.
2.2.1	2	Add additional virtualization techniques to processes.	S	3	4	12	s	2.2.1.2	If a component is infected, it should not propagate to others. High level.
3.1.1	1	Distribute the security mechanisms with the nodes.	S	4	4	16	m	3.1.1.1	Some security mechanisms are deployed system-wide but forgotten for the components: intrusion propagation risks.
3.2.1	1	Choose the type of cloud that is compliant with your needs.	P	2	2	4	c	3.2.1.1	Conditions, services and policies can vary a lot. Security should be OK.
3.2.1	2	Check whether your cloud provider should be compliant with standards, certifications or other labels.	SP	2	4	8	s	3.2.1.2	Mandatory for some markets, services, needs, etc. Are normally known.

3.2.1	3	Verify the reputation of your provider and whether it is audited by trusted and recognized sources.	SP	2	4	8	s	3.2.1.3	Is normally handled appropriately by companies because of its importance.
3.2.2	1	Define a policy on users accesses and identities.	SP	3	4	12	s	3.2.2.1	Limit access to data for people that must not access to it, easy to have leaks because of changes.
3.2.2	2	Manage the assets using categories, inventories and assessments.	SP	3	2	6	s	3.2.2.2	The assets can be concerning in terms of privacy for misplaced sensitive data, or a security concern for accesses. Moderate severity, and possible. Managing them to avoid mistakes on intern audits.
3.2.2	3	Encrypt the assets sent to a cloud platform when possible.	S	3	3	9	s	3.2.2.3	The someone else's computer. We never really know who has access to assets, can mitigate the risks.
3.2.2	4	Use homomorphic encryption whenever possible.	P	2	4	8	s	3.2.2.4	Big big improvement, especially for privacy. But not suitable for every situation
3.1.3	1	Isolate suspicious instances without interruption.	S	2	4	8	s	3.1.3.1	Suspicions are not rare, and they can have big impacts if malicious (more rare).
3.1.3	2	Define a failover policy.	S	4	2	8	s	3.1.3.2	Severity limited to availability, can sometimes lead to data losses.
3.1.3	3	Enable address relocations.	S	3	2	6	s	3.1.3.3	Severity limited to availability.
3.1.3	4	Define a "let's hope for the best" policy.	S	2	3	6	s	3.1.3.4	Severity limited to availability, analyse for proofs.
3.3.1	1	Schedule periodic backups that include all data to be saved.	S	3	5	15	m	3.3.1.1	Backups are sometimes not periodic, or too sparse in time. Major severity if not valid.
3.3.1	2	Test the backups periodically.	S	3	5	15	m	3.3.1.2	Lots of backups are not checked and some of them are non valid. Severe if it happens.
3.3.1	3	Test the backups resilience.	S	3	5	15	m	3.3.1.3	Lots of backups are not checked and some of them are non valid. Severe if it happens.
3.3.2	1	Enable the service to restore itself alone in case of a failure.	S	2	4	8	s	3.3.2.1	Generalized failure are unlikely, but the impact would be important.
3.3.2	2	Protect the service against denial of service attacks.	S	3	3	9	s	3.3.2.2	Those attacks became quite common, moderate impact (mostly availability).
3.4.1	1	Ensure that all security mechanisms are implemented locally in each service.	S	4	4	16	m	3.4.1.1	Quite likely that some security mechanisms are not local to services, can then expose it and becoming a major problem.
3.4.1	2	Apply the "trust no one" principle between services.	S	3	3	9	s	3.4.1.2	Principle either applied or not. Concerning but not major.
3.4.1	3	Enforce relationships between services using mutual and fine-grained authorization.	S	3	2	6	s	3.4.1.3	Help the trust no one implementation.
3.4.1	4	Ensure and verify trust between each service.	S	3	4	12	s	3.4.1.4	Help the trust no one by applying rules.
3.5.1	1	Enable security features of the used operating systems.	S	2	4	8	s	3.5.1.1	Depending on the system, they must be configured. They are embedded because of the severity of the threats they address.
3.5.1	2	Take into account the strengths and weaknesses of each operating system while making a choice.	SP	2	3	6	s	3.5.1.2	Should normally be done every times. Has a controlled impact if not adapted.
3.5.1	3	Understand and configure the operating system security features.	S	4	4	16	m	3.5.1.3	Can be very specific, which can imply that some of them are not configured. And they can go into deep details.
3.5.1	4	Choose a Linux distribution adapted to the needs.	S	2	2	4	c	3.5.1.4	Only minor changes.
3.5.1	5	Understand and configure the operating system privacy features.	P	2	2	4	c	3.5.1.5	Changes vary depending on the operating system.
3.6.1	1	Mitigate the most common network attacks.	S	4	5	20	m	3.6.1.1	Very often used attacks, can be very severe.
3.6.1	2	Mitigate the threats specific to wireless connectivity.	S	3	5	15	m	3.6.1.2	Common threats are often mitigated, but if not, can cause critical outrages (network= backbone of organizations).
3.6.1	3	Design and apply adapted network access control policies.	S	3	5	15	m	3.6.1.3	Often well design, but errors can lead to critical outrages (network= backbone of organizations)..
3.6.2	1	Monitor the whole network.	S	3	4	12	s	3.6.2.1	Sometimes implemented, sometimes not. Can help to detect problems and intrusions before their impact.
3.7.1	1	Verify that countermeasures are enforced on pieces of hardware.	S	3	5	15	m	3.7.1.1	If hardware pieces are infected, access is given to the lowest layer of systems: critic severity.
3.7.1	2	Review and validate the supply chain parties.	S	2	5	10	s	3.7.1.2	They should be valid, web environments are not as specific as embedded systems. Again, lowest layer of systems, critic severity.
3.7.2	1	Configure and enable the system-level protection schemes.	S	3	5	15	m	3.7.2.1	If hardware pieces are infected, access is given to the lowest layer of systems: critic severity.

4.1.1	1	Apply anonymization techniques to data at the collection phase.	P	3	3	9	s	4.1.1.1	Depending on the context and the type of service, not always done. Can lead to legal problems.
5.1.1	1	Educate organization parties on the limitations of private browsing mode to protect privacy.	P	2	1	2	c	5.1.1.1	Limited impact on privacy, none on security.
5.1.1	2	Verify the privacy policies, security and source of the browser extensions.	SP	2	3	6	s	5.1.1.2	If an extension is compromised, it could lead to larger security and/or privacy breaches. Unlikely.
5.1.1	3	Disable all features that are not needed.	SP	2	2	4	c	5.1.1.3	Could help if a major breach is discovered. Plus, WebGL can disclose private information.
5.1.1	4	Choose browsers accordingly to the providers' tracking policy.	P	3	1	3	c	5.1.1.4	Users not always aware of that. Very unlikely to have an impact.
5.1.1	5	Disable browser third-party cookies support.	P	3	1	3	c	5.1.1.5	Not necessary and not vital, but simple to configure.
5.1.1	6	Improve browsing privacy by installing verified and adapted browser extensions.	P	3	1	3	c	5.1.1.6	Not necessary and not vital, but simple to configure.
5.1.2	1	Collect user data only if they gave explicit consent.	P	3	5	15	m	5.1.2.1	Legal obligation. Can lead to severe problems. Often properly handled.
5.1.2	2	Collect user data only for legitimate interests.	P	4	5	20	m	5.1.2.2	Legal obligation. Can lead to severe problems. Less often properly handled.
5.1.2	3	Limit user tracking in sent emails to the minimum.	P	2	3	6	s	5.1.2.3	Unlikely to be filled because of third party tools often used. Moderate impact, but third party included.
5.1.3	1	Limit the mobile providers' user tracking within developed applications.	P	3	2	6	s	5.1.3.1	Collections by default. Minor impact limited to mobile providers.
5.2.1	1	Apply email security features in the organization.	S	3	4	12	s	5.2.1.1	Communication breaches can disclose sensitive information. Some security features are often missing, low adoption of PGP.
5.2.1	2	Mitigate the threats specific to instant messaging applications.	SP	2	4	8	s	5.2.1.2	Communication breaches can disclose sensitive information. Unlikely, but can have major impacts depending on the problem (integrations in systems).
5.2.1	3	Avoid malicious emails using appropriate tools.	S	2	3	6	s	5.2.1.3	Well known threat now, but can still cause damages.
5.2.1	4	Avoid sensitive information leaks using appropriate tools.	P	1	5	5	s	5.2.1.4	Rare, but can lead to critical leaks for the organization.
6.1.1	1	Assess risks using a standardized and recognised method.	S	4	4	16	m	6.1.1.1	Mainly for security risks. Likely that the assessment is not complete, and they can cause serious harm (unknown).
6.1.1	2	Ensure that countermeasures of identified threats are tested and validated.	S	4	4	16	m	6.1.1.2	Coverage not always complete, can cause serious damages.
6.1.2	1	Define and verify the organization policies.	SP	4	4	16	m	6.1.2.1	Incomplete coverage can enable threats with large severity.
6.1.3	1	Define a continuous and cyclic technology watch.	SP	4	3	12	s	6.1.3.1	Not often done in the privacy/security part, mainly focused on competition. Can avoid recent threats.
6.2.1	1	Apply homomorphic encryption to prevent auditors from accessing sensitive data plaintexts.	P	4	2	8	s	6.2.1.1	Extra protection to avoid unlikely problems, unlikely that it is implemented.
6.3.1	1	Define and apply an information security policy.	S	5	5	25	m	6.3.1.1	Mandatory for everyone. Important on all levels.
6.3.1	2	Ensure parties involvement and compliance to policies.	S	4	5	20	m	6.3.1.2	Important to be compliant with them, likely that some parties are not aware of all the policies.
6.3.2	1	Provide prevention to parties.	SP	3	4	12	s	6.3.2.1	Often provided once or twice, but not consistent though time. Can lead to severe breaches.
6.3.2	2	Provide training to parties.	SP	4	4	16	m	6.3.2.2	Theory differs from practice, and can be evaluated. Can lead to severe breaches.
6.3.2	3	Enable human detection of attacks.	SP	3	4	12	s	6.3.2.3	People often notice attacks, but this task is not often systematic and framed by the organization. Can avoid serious damages.
6.3.2	4	Enable technical detection of attacks.	SP	4	4	16	m	6.3.2.4	They exist but their coverage is not always complete. Can avoid serious damages.
6.3.2	5	Mitigate the most common attacks.	SP	3	5	15	m	6.3.2.5	Supposed to be mitigated, but has to be verified. Common attacks, so well-designed with very severe damages.
6.3.3	1	Assure that necessary backups of the internal and external communications are private and secure.	P	3	3	9	s	6.3.3.1	Often done in general backups or by the provider, but not clearly defined in the policies. Can avoid later problems (legal obligations).

6.3.4	1	Apply policies reconciliation if collaborating with other organizations.	P	3	4	12	s	6.3.4.1	Problems can cause serious damages, such things are discussed but not always systematically framed.
3.1.1	2	Compute distributed function by respecting privacy constraints.	P	3	3	9	s	3.1.1.2	Limited impact with third parties, not that hard to fulfil constraints
3.1.1	3	Establish the trustworthiness and role of each component.	SP	3	3	9	s	3.1.1.3	Avoid data breaches if another node is malicious, but moderate probability and severity.
3.1.2	1	Secure the distributed storage using appropriate techniques.	SP	3	5	15	m	3.1.2.1	Any error can cause massive damages, those errors can come from a lot of vulnerabilities. Not easy to assess all of them.
2.1.5	1	Review the software dependencies for security issues.	S	3	5	15	m	2.1.5.1	Small breaches can lead to complete and critic access to a whole software. Massive vulnerabilities are rare, but not uncommon.
2.1.2	9	Include the dependencies in the testing process.	S	3	3	9	s	2.1.2.9	Often included, but by default and not specifically and systematically. Can detect problems, not as severe as big vulnerabilities.
3.4.2	1	Use a decentralized architecture.	P	1	4	4	c	3.4.2.1	Very uncommon, but help the privacy a lot. Is restrictive for some processes.
1.3.4	1	Define and apply a privacy policy.	P	5	5	25	m	1.3.4.1	Mandatory and central.
6.3.4	2	Handle collaborative enforcement of privacy policies on shared data.	P	3	5	15	m	6.3.4.2	Impact can be enormous on the other parties side, possible to get problems through them.
1.3.4	2	Integrate privacy concerns into the personal data management.	P	3	3	9	s	1.3.4.2	Should be integrated, but not often fully complete. Limited impact, major issues often taken into account.
2.1.6	1	Avoid dark patterns in the development process.	P	4	3	12	s	2.1.6.1	Moderate impact on privacy, with data that can be shared with parties by default for example. But they are very common and sometimes unintentional.
3.2.3	1	Mitigate the threats specific to migrations done from self hosting to cloud hosting.	SP	3	4	12	s	3.2.3.1	Problems severity should be contained, cloud providers are normally serious. But external threats but be handled and are not rare. If leaks, a large part of the organization data can be disclosed. Same with unauthorized access.
3.2.3	2	Mitigate the threats specific to migrations done from dedicated infrastructure to shared infrastructure.	S	3	4	12	s	3.2.3.2	Focused on security, should not happen (but if it does, major impact). A few mitigations should be enforced, they are not always applied.
3.2.1	4	Verify whether cloud providers assure sufficient security levels.	S	2	5	10	s	3.2.1.4	Can have catastrophic legal and public image consequences. Some specific data / markets need appropriate protections, often already known.
4.1.1	2	Use appropriate protections on the data generation phase.	P	4	4	16	m	4.1.1.2	Lowest control on the data, can involve a large variety of devices (high variability). Leaks and intrusion can occur and are difficult to mitigate.
4.1.1	3	Use appropriate protections on the data storage phase.	P	2	4	8	s	4.1.1.3	The storage can be outsourced, and involve third parties. Normally, data are encrypted.
4.1.1	4	Use appropriate protections on the data processing phase.	P	3	3	9	s	4.1.1.4	Mostly done internally, lower risks to have problems.
4.1.1	5	Use appropriate protections on the data publishing phase.	P	2	5	10	s	4.1.1.5	Legal obligations are often covered. But errors can lead to massive outrages.
4.1.1	6	Apply legislative obligations.	P	2	5	10	s	4.1.1.6	Normally handled by default when processing big data. Consequences can be massive (public image, lawsuits).
6.4.1	1	Mitigate the threats specific to the web.	S	3	4	12	s	6.4.1.1	Threats are normally known, large life span. But can be major if exploited.
6.4.1	2	Adopt all the recognized best practices.	SP	3	3	9	s	6.4.1.2	Limited probability and severity if exploited.
1.4.1	1	Design the authorization policies appropriately.	S	2	4	8	s	1.4.1.1	Should normally be implemented, but some doors can be missed. And can lead to intrusions into the system, with serious threats.
1.4.1	2	Configure the OAuth 2.0 framework appropriately.	S	2	4	8	s	1.4.1.2	Easy to apply, modern configurations should already integrate them. But Can lead to intrusions with serious threats.
1.2.2	1	Implement the security guidelines of the “lost password” feature.	S	3	3	9	s	1.2.2.1	Often follow guidelines, but can also follow basic rules and be improved. Can lead to leaks, but for limited subset of users.
1.2.3	1	Enforce the security of biometric systems.	S	3	3	9	s	1.2.3.1	Often follow guidelines, but can also follow basic rules and be improved. Can lead to intrusions (depends on the system), limited because should not be used alone.
1.2.3	2	Mitigate the threats specific to biometric systems.	S	3	3	9	s	1.2.3.2	Often follow guidelines, but can also follow basic rules and be improved. Can lead to intrusions (depends on the system), limited because should not be used alone.
1.2.4	1	Choose multi-factor schemes adapted to the needs.	S	2	2	4	c	1.2.4.1	Should be chosen according to needs. Limited impact if not adapted.

1.5.1	1	Apply the isolation and least privilege patterns on applications components.	S	3	4	12	s	1.5.1.1	Patterns already known so should already applied, but not always complete. Great impact if a breach occurs.
4.1.2	1	Integrate big data specific requirements into the data access policies.	P	3	3	9	s	4.1.2.1	Often implemented, but can be improved because of the large entry doors. Can lead to leaks.
1.5.1	2	Choose the access control method appropriately.	S	3	4	12	s	1.5.1.2	The method itself don't mitigate the threats, but it changes the easiness of the rules. Can lead to strong errors if misconfigured.
1.3.5	1	Handle sensitive information in tabular data appropriately.	P	2	4	8	s	1.3.5.1	Quite niche, but can lead to serious damage if not filled (public image, regulations).
1.3.5	2	Select the appropriate privacy-preserving techniques.	P	3	5	15	m	1.3.5.2	Usage of pre-built techniques can lead to inappropriate choices, that can have serious damages due to the sensitive nature of anonymized data.
1.3.3	3	Ensure compliance for anonymization of data.	P	3	5	15	m	1.3.3.3	Should be OK, but some data can be forgotten. Sensitive data leaks lead to lawsuits .
1.3.5	3	Select the appropriate anonymization techniques.	P	3	4	12	s	1.3.5.3	Errors lead to major problems, should be mitigated in the compliance.
2.3.1	1	Define the terms of service and privacy policies appropriately.	P	3	4	12	s	2.3.1.1	Templates are often used for those two things, which can cause errors for specificities. Some serious legal problems can occur if not appropriate.
2.3.3	1	Provide an API for users to explore which of their data are shared.	P	2	2	4	c	2.3.3.1	Would not be used by all users, and would give information that they should already know by reading the terms and the policy.
2.3.1	2	Ensure that the machine learning-enforced APIs are compliant with the corresponding regulations.	P	3	4	12	s	2.3.1.2	Supposed to be checked, but has to be verified. Can cause severe legal damages.
2.3.2	1	Choose the APIs type accordingly to the needs.	S	2	3	6	s	2.3.2.1	Choice pretty staightforward, and consequences not that big if other security mechanisms are implemented.
2.3.2	3	Test the APIs using the chaos engineering method.	S	4	3	12	s	2.3.2.3	Likely to not have this testing method not implemented. If not there, some risks can be missed.
2.3.2	2	Mitigate the threats specific to APIs.	S	4	4	16	m	2.3.2.2	Common threats are normally known, large life span. But can be major if exploited, and can be automated.
4.2.1	1	Ensure that the artificial intelligence processes are compliant with the corresponding regulations.	SP	3	5	15	m	4.2.1.1	Templates are often used to ensure compliance, which can cause errors for specificities. Some serious legal problems can occur if not appropriate.
4.2.2	1	Avoid privacy leakages.	P	3	4	12	s	4.2.2.1	Can disclose sensitive information about some users' context and data. Not straightforward to apply.
4.2.2	2	Use homomorphic encryption.	P	2	3	6	s	4.2.2.2	Difficult to apply and restrictive as well. But allows to avoid major privacy concerns
4.2.2	3	Use differential privacy techniques.	P	2	3	6	s	4.2.2.3	Difficult to apply and restrictive as well. But allows to avoid major privacy concerns
4.2.3	1	Mitigate the security threats.	S	3	5	15	m	4.2.3.1	Can allow to learn specific and central knowledge about the model's behaviour, which can lead to very serious damages depending on the target. But require adapted access to model.
4.2.3	2	Mitigate the privacy threats.	P	3	5	15	m	4.2.3.2	Can disclose sensitive information about users' context and data. Mitigations can impact performances. Take some time.
4.2.3	3	Mitigate the model life cycle threats.	SP	3	5	15	m	4.2.3.3	Well known, but numerous things to mitigate. Can strongly impact the model behaviour, which can lead to very serious damages depending on the target.
4.2.3	4	Mitigate the model datasets threats.	SP	3	5	15	m	4.2.3.4	Hard to detect. Can strongly impact the model behaviour, which can lead to very serious damages depending on the target.
1.2.4	2	Mitigate multi-factor specific risks.	S	3	2	6	s	1.2.4.2	Should be known and mitigated, but specificities can avoid full mitigation. Limited impact.
1.2.1	2	Back up the restoration keys.	S	2	3	6	s	1.2.1.2	In an manner that avoid any unwanted access to them.

descriptions

item	id	name	value	link	alt	PK	Comments
1.1.1.1	1	Stored data can be vulnerable	It avoids problems if untrusted pairs have access to the data. Be aware that encrypting data is not sufficient: a policy must be defined in order to ensure that data is protected using appropriate and strong protocols.			1.1.1.1.1	
1.1.1.2	1	Every data leaving a system or device must be encrypted	Encrypting the data is not sufficient: a policy must be defined in order to ensure that data is protected using appropriate and strong protocols.			1.1.1.2.1	
1.1.2.1	1	Symmetric versus asymmetric encryption	Symmetric encryption uses one single secret key for encrypting and decrypting data between the sender and the receiver. Symmetric encryption uses public keys for encryption and different keys (secret) for decryption. Asymmetric encryption is not very efficient for small devices due to additional computations needed. Therefore, symmetric encryption algorithms are almost a thousand times faster than asymmetric algorithms, because of less processing power being required.			1.1.2.1.1	
1.1.2.1	5	Use adapted ciphers and parameters	Every encryption mechanism can have different configuration. Be aware that your configuration is adapted to the latest security needs, and that the cipher suites are strong enough.			1.1.2.1.5	
1.3.1.1	1	Who should generate them?	Pseudonyms can be created remotely by a centralized third party, or locally by the holder of identity. The latter is the most private solution, but the process must be trusted.			1.3.1.1.1	
1.3.1.1	2	With which features?	Pseudonymization techniques can provide different features: operation reversal, key recovering, sharing capabilities. Hash functions are limited for data sharing purposes.			1.3.1.1.2	
1.3.1.1	3	List of some techniques	Pseudonyms can be calculated by either encryption, using symmetric or asymmetric keys which enable reversal of the operation, or hashing, but it needs a list which is a weak point. Some approaches: Peterson (keys stored in the database), pseudonymization of information in e-health (hull architecture), electronic health card (service-oriented architecture), Thielscher (identification data and anamnesis data stored in two different databases, using decentralized keys), Pommerening (two approaches, for one-time usage or re-linkable patients), Slamanig and Stingl (centralized database with smart cards for authentication).			1.3.1.1.3	

1.3.1.2	1	Beware of the data scopes	Data anonymization, de-identification and pseudonymization are necessary to share health data outside a patient's privacy and trust sphere. Consult your local laws.			1.3.1.2.1	
1.3.1.3	1	Give control to users	Users should be able to group their identities to allow fine sharing with other parties.			1.3.1.3.1	
1.3.1.3	2	How to handle multi generation?	Same pseudonym generation regardless of data source origin can be computed using a dual-pass pseudonymization scheme. Pseudonym trees can be used to differ the identity sent to each provider.			1.3.1.3.2	
1.3.2.1	1	Give the habit to users to decide for their privacy settings	Users are asked to make privacy decisions too frequently or under circumstances that are seen as low-risk may become habituated to future, more serious, privacy decisions. But if they are asked to make too few privacy decisions, they may perceive that the system is acting against their wishes. There is permission types that are seen as more dangerous, which are the ones related to personal data. Others are seen as more regular ones.	https://ieeexp	Link to the study	1.3.2.1.1	
1.3.3.1	1	One of the biggest regulation	The GDPR regulation covers all personal data, which encompasses data that can directly or indirectly identify an individual, including identifiers. It concerns all EU residents regardless of the location of the data processing. It encourages both ethical approaches to data collection and public trust.	https://gdprhu	Help to be compliant with the GDPR	1.3.3.1.1	
1.3.3.1	2	GDPR principles	Fairness and lawfulness, purpose limitation, data minimization, accuracy, storage limitation, integrity and confidentiality.	https://eur-le	Legal text	1.3.3.1.2	
1.3.3.2	1	Example for the GDP	The top privacy-related challenges are the top management's lack of commitment, weak management on stored personal data in the cloud, underestimations of the GDPR effects on the organizations, a lack of GDPR understanding and bad interpretation of authoritative legal texts.			1.3.3.2.1	
1.3.3.2	2	Major challenges for ensuring data privacy	Delivering enough data privacy related information and communicating with business users, having sufficient resources for GDPR preparation, and proceeding to the verification of many systems for their GDPR compliance.			1.3.3.2.2	
1.3.3.2	3	Some leads to prepare compliance	The companies have prepared themselves for GDPR regulations by identifying data registers, outsourcing the maintenance of data registers, a better monitoring of applications, participating in GDPR training events, creating data balance sheets, reviewing contracts with suppliers, and analysing GDPR from the business perspective.			1.3.3.2.3	

1.3.3.2	4	Note about the GDPR focus	A focus has been made on the GDPR because of its broad application and the importance of the European market. However, the challenges and issues presented in this item are still valid for other regulations.			1.3.3.2.4	
2.1.1.3	1	Can be enforced by simple things	Great interface usability and adapted design of notifications positively impact users' perceived application security. Furthermore, disruptive notifications irritate users and negatively influence those perceptions.			2.1.1.3.1	
2.1.1.3	2	Major concern for users	57% of mobile users have uninstalled or decided not to install an application due to concerns about how their personal information is processed.	https://www.p	Link to the study	2.1.1.3.2	
2.1.2.2	1	Have a designated responsible	The policy must also include which person must be contacted when an issue is found.			2.1.2.2.1	
2.1.2.3	1	Use a tool	Some tools can be used to track and resolve issues in organizations.			2.1.2.3.1	
2.1.2.8	1	Use static approaches	Static approaches disassemble and analyse the source code, either with signature matches using a dictionary or with permission checks.			2.1.2.8.1	
2.1.2.8	2	Use dynamic approaches	Dynamic approaches examine the application behaviour during its execution. It uses anomaly detection, data and control flow monitoring, emulation techniques, permissions management, device locking (avoid device tampering), anti viruses installation, and verification that applications are only installed from trusted packages repositories.			2.1.2.8.2	
2.1.3.1	1	What is FIPP?	Key principles useful to integrate in every software developments when information about people are processed.	https://www.f	Further explanation by the FPC agency	2.1.3.1.1	
2.1.3.2	1	What is this principle?	Broadly speaking, a piece of software should not go beyond its purpose. The EU data protector defined a guidance on how to guarantee this principle during software developments.	https://edps.e	More information by the EU data protector	2.1.3.2.1	
2.1.3.4	1	Apply Secured SDLC	This approach allows to ensure that every step of a software development includes useful security mechanisms, concerns and designs.	https://www.c	Secured SDLC seen by the CISA institute	2.1.3.4.1	

2.1.4.1	1	A secure development model	Security standards have been defined for each data cycle. In the data storage state, locally stored data must be limited, and alternatives for key stores must be used. For data access, developers' attention must be focused on features using geolocation, application-device identifiers, and user sessions. During data transfer, adapted encryption must be enforced, digital signatures must be used, as well as security keys. Data transfer is the weakest link of the chain.			2.1.4.1.1	
2.2.1.1	1	Multiple levels of virtualization possible	Going from the lowest level of protection to the most advanced: in-browser security, sandboxing (partial virtualization), full virtualization, secure virtualization. Secure virtualization must have the following attributes: host and network isolation, real-time detection (previously unseen attacks), fast and complete recovery to a known clean state, forensic data collection on infection, hypervisor integrity checks.			2.2.1.1.1	
2.2.1.2	1	Multiple techniques of virtualization possible	Some examples to encapsulate processes: restrict account privileges, separate the file systems of applications, separate untrusted code from the system.			2.2.1.2.1	
3.1.1.2	1	How to evaluate it?	A collective computation over correlated data must not reveal the value of a specified private function computed by each of the terminals. If so, such functions are therefore "securely computable". A class of functions is securely computable if and only if the conditional entropy of data given the value of private function is greater than the least rate of interactive communication required for an appropriately chosen multi-terminal source coding task.			3.1.1.2.1	
3.1.1.3	1	Two kinds of threats	Centralized systems threats, amplified by distribution, and distributed-specific threats, brought by distribution requirements such as scalability, interoperability, interconnection, untrusted nodes, different operating systems and applications suites, and multiple security policies.			3.1.1.3.1	
3.1.1.3	2	How to design security policies?	Security policies should be designed without regards to leaks and weaknesses of the nodes: they must be addressed independently. To this end, social and technical aspects must be considered. A common security model should be optimized to provide interoperability, while establishing the degree of trustworthiness of each component.	https://www.r	An example of security policies	3.1.1.3.2	

3.1.1.3	3	What should security policies include?	It includes identification and authentication, access control, confidentiality, non-repudiation, and availability. The issues that components face are untrusted partners (workstations or servers), untrusted communication media (physical links), untrusted intermediate systems (routers, gateways), untrusted clients (software), trusted user/client identity (unique identity), trusted server identity, trusted administration.			3.1.1.3.3	
3.1.1.3	4	Other useful facts	Components can migrate between categories, for example while mobile roaming or during changes on the network trust. The third party authentication services are one of the largest controversial and challenging issues. All distributed application servers and database servers should trust servers using two-way authentication, certificates, message addresses, or content certification. Partners should be trusted using levels of trust, with different evaluations of used software. All partners must be untrusted by default, except for security administrator and third-party authentication services.			3.1.1.3.4	
3.1.2.1	1	Why is it important?	Humongous quantities of generated data, which must be shared, replicated, kept online for performance reasons, always available, and recovery requirements make systems more vulnerable to security breaches.			3.1.2.1.1	
3.1.2.1	2	Security concepts to be implemented	1) Authentication and authorization, through all data life cycle. 2) Availability, which includes backup and recovery. 3) Data confidentiality and integrity. 4) Key sharing and key management with an efficient and scalable management. 5) Auditing and intrusion detection. 6) Usability, manageability and performance.			3.1.2.1.2	
3.1.2.1	3	Three storage systems classification	The first is networked file systems: a server authenticates users and checks any access privileges. It assumes that the file servers and the system administrators are trusted. It does not include end-to-end data security. The second is cryptographic file systems: they enable end-to-end security using cryptographic operations natively in the file system. Cryptographic operations are done on the client side in order to protect data from both the server and unauthorized users. The server is minimally trusted, and not included in the process. The third is storage-based intrusion detection systems: they monitor activities related to data and look for manifestations of attacks.			3.1.2.1.3	

3.1.2.1	4	Storage systems comparison	Categories can be compared using the following criteria: used authentication by entities and messages, access control type, end-to-end data and metadata confidentiality support, end-to-end key management, revocation, non-repudiation, key storage, and long-term key management.			3.1.2.1.4	
3.2.1.1	1	Multiple choices of Cloud types	A cloud platform can be shaped in various ways. Its type can either be private, public or hybrid. Cloud platform can provide one or multiple service models, such as PaaS, IaaS, SaaS or other more specific ones. This choice must be done according to your needs, constraints and obligations.			3.2.1.1.1	
3.2.1.2	1	Beware of your business case, and its related data	Some type of sensitive data, such as medical reports, might need specific compliances. It also concerns the cloud provider. A hybrid cloud solution might be adapted.			3.2.1.2.1	
3.2.2.2	1	Tagging or labelling can help	To help you in this task, cloud providers consoles have built-in features to tag or label assets. This is very helpful for managing your data.			3.2.2.2.1	
3.2.2.3	1	An additional layer of protection	The major issue is that users do not know where sensitive data is stored. The data boundaries vary depending on laws, access privileges, data protection or privacy requirements. An interesting mitigation would be to use an intelligent cloud-based machine encryption and decryption system.			3.2.2.3.1	
3.2.2.4	1	Strong but restrictive technology	Providers' ability to access sensitive user data is a major obstacle in the adoption of cloud services. Homomorphic encryption allows operations on encrypted data with the same results after treatment as with plain data. Several categories of encryption can be used, some of them have limited available operations or limited representation of data. The three main challenges of this technology is its efficiency with limited operations and performances, its robustness which is based on the size of the key, and its delay due by great encryption, decryption and processing times.			3.2.2.4.1	
3.3.1.3	1	The 3-2-1 backup rule	Keep 3 copies of any important file (1 primary, 2 backups). Keep the files on 2 different media types, for protection against different types of hazards. Store 1 copy outside of your facility.	https://www.c	Backup options by the US-CERT	3.3.1.3.1	
3.4.1.2	1	Security within perimeter	Assume that other services may be compromised and hostile.			3.4.1.2.1	

3.4.1.3	1	Various technologies can be useful	One of the most used technology for this need is the OAuth standard. Others can be used as well.			3.4.1.3.1	
3.4.1.4	1	Some technologies	Some leads: securing communication with MTLS, self-hosted PKI and security tokens.			3.4.1.4.1	
3.5.1.1	1	Major security features of the two major operating systems for servers	Windows and Linux both uniquely identify each entity. On the access tokens, Windows stores restrictions where Linux uses DAC and MAC. Furthermore, it does not store the token types. Impersonation design is more secure in Windows than in Linux. Regarding ACL, Windows uses privileges and restrictions, Linux uses MAC and DAC and does not handle logging. For privileges and user rights, Windows uses a separate process where Linux uses MAC and handles restrictions with a separate daemon. They both have similar auditing and logging features. Windows implements a more secure but more complicated authentication system than Linux. Linux has no native file system encryption. Windows has more security components within its kernel and is more complicated, where Linux uses user-mode processes and is more efficient.			3.5.1.1.1	
3.5.1.2	1	Characteristics of the most used operating systems	Windows has a great support and compatibility with lots of functions, but is costly, slow and exposed to viruses. UNIX comes with a great user control and a high reliability, but it needs expertise with a large learning curve. Linux is free, less vulnerable, has a great variety, but is complicated, has a low application compatibility and too few vendors. MacOS is exposed to few viruses, has a high reliability, but is expensive, is only compatible with Apple computers and has a low application compatibility.			3.5.1.2.1	
3.5.1.3	1	Some leads for all systems	Administrators must apply security through repositories: they must avoid software from other sources than the repositories provided by the distribution or vendors. Then, using an anti virus is recommended. Precautions must be taken if compatibility layers are used, such as Wine. Administrators must always keep software up-to-date with security patches. They must also set up firewalls to avoid access gains. Different accounts with unique passwords must be provided for each person, including separate usages such as root access and regular users. Finally, adapted file access permissions must be enforced.			3.5.1.3.1	

3.5.1.3	2	Some leads for Linux	One of the biggest security feature is SELinux, which has been developed to implement MAC policies. It supports multiple security models, is extensive but have low flexibility and difficult to manage.			3.5.1.3.2	
3.5.1.4	1	Various goals	User privacy varies a lot among Linux distributions: some of them are focused on strong, complete privacy, some of them are oriented towards other goals.			3.5.1.4.1	
3.6.1.1	1	Main issues	The network traffic must be analysed, both on the flows and formats. Be aware that attackers can know the protocols intents and their rules to interpret the associated formats and flows. Network intrusions can be used for several goals, including to consume the resources uselessly, to interfere with the system or to gain knowledge. The DDoS attacks intent is to slow or to interrupt services. There is no single technique to detect network intrusion: signatures or anomaly detections are the most common.			3.6.1.1.1	
3.6.1.1	2	Active attacks	Active attacks are initiated by commands. They include spoofing (play on identity), routes modification, wormhole (tunnelling traffic), fabrication (false routing message), denial of services, sinkhole (prevent node to exchange information), and Sybil (insert multiple malicious nodes).			3.6.1.1.2	
3.6.1.1	3	Passive attacks	Passive attacks do not require any action by attackers. They could be traffic analysis, eavesdropping (find credentials in communication), or monitoring access.			3.6.1.1.3	
3.6.1.1	4	Advanced attacks	Advanced attacks are more difficult to realize. Some of them are black hole (replace the best paths), rushing (make receiver busy), replay (repeat or delay data), Byzantine (disrupt or degrade routing), or location disclosure.			3.6.1.1.4	
3.6.1.2	1	Wireless connectivity brings additional threats	Introduction of malicious activities, interception of data transmission, or passive eavesdrop.			3.6.1.2.1	
3.6.1.2	2	Some well-known attacks	Malicious association (mock a legitimate access point), ad hoc network attack (no central access point: access control issues), man in the middle, rogue access point (unsanctioned by administrators), lack of encryption.			3.6.1.2.2	
3.6.1.2	3	Some mitigations	Only chose adapted and strong encryption parameters, educate the users, limit the network access with explicit allowance, change factory router configuration, change default router identifier, disable broadcasting of identifiers, apply MAC filtering, and keep firmwares up to date.			3.6.1.2.3	

3.6.1.3	1	Backbone of the network	Firewalls configurations have been identified as the main problem of weak network security levels. To configure them appropriately, administrators need a clear methodology and adapted supporting tools. They should use high level languages to specify a network security policy in order to avoid mistakes and to help further edits in the future. It is recommended to apply a dual security policy which specifies both permission and prohibition rules. However, it requires rules ordering, which is difficult to assess. An alternative could be to apply closed access control policy, with permissions only.			3.6.1.3.1	
3.6.2.1	1	Three approaches	The first approach is packet capture: it intercepts data packets that are crossing a node or moving over it. The second is DPI: actions on packets are applied when they match specific data or code payloads. Finally, there is flow-based observations that analyse packets in a specific transport connection or a media stream.			3.6.2.1.1	
3.6.2.1	2	DPI tool selection	The criteria to choose between DPI tools are to check their prototype support, developer friendliness, and extensibility.			3.6.2.1.2	
3.7.1.1	1	Raising complexity of hardware security	Hardware security is getting increasingly more complex because of two trends. First, the skills and resources to counter well-funded criminals aiming for economic goals have been raising. Secondly, an increase of hardware-based attacks has been noticed: this kind of attacks leads to the most privileged entities, which brings lots of flexibility and power with the ability to escape operating systems detections.			3.7.1.1.1	
3.7.1.1	2	Some background context	Algorithmically secure cryptographic processes rely on a hardware root of trust to deliver the expected protections when implemented in software. Critical control and communication functions assume that the hardware is resilient to attacks. Backdoors have been found in various systems, even in the military field. Cost, power consumption, performance, and reliability are considered first while designing hardware, which causes security issues to be considered as an afterthought. The location of the attackers can be anywhere, such as 3PIP vendors, SoC integrators, foundries, PCB assembly units, test facilities, end users, or the recycling/repackaging facilities.			3.7.1.1.2	

3.7.1.1	3	Most known attacks	The most known attacks types are active adversarial manipulation of control signals, exploit security gaps in the interactions of multiple platform features, insecure platform initialization by boot-up firmware, ability of untrusted or lesser privileged entities to maliciously influence operations, hardware trojan, intellectual property piracy, integrated circuit overbuilding, reverse engineering, side-channel analysis, and counterfeiting.			3.7.1.1.3	
3.7.1.1	4	The most useful mitigations	Some mitigations would be to apply secured Software Development Life Cycle, design obfuscation, intellectual property watermarking, intellectual property fingerprinting, integrated circuit metering, split manufacturing, integrated circuit camouflaging, integrated circuit information leakage reduction, key-based authentication, noise injection, secure-scan, physical non-clonable function/unique ID(s), and ageing sensors.			3.7.1.1.4	
3.7.1.2	1	A complex issue	The supply chain is often considered as well-protected, but it is actually spread around the globe and involves lots of third parties which makes it difficult to fully verify and control processes.			3.7.1.2.1	
3.7.2.1	1	A few examples	Depending on the hardware manufacturer, ARM TrustZone, Intel SGX, CHERI or LowRISK bring the possibility to secure running processes.			3.7.2.1.1	
5.1.1.1	1	Private mode is limited	Private mode can prevent many tracking techniques, but it has a lot of limitations. Furthermore, it does not make users anonymous.	https://www.r	More information from Mozilla	5.1.1.1.1	
5.1.1.3	1	What is an adapted configuration?	That can vary depending on the usage and the browser.	https://securi	List of actions	5.1.1.3.1	
5.1.1.4	1	Consult trusted online resources	Lots of public and community-driven projects have been created to evaluate browsers on their privacy features, such as the privacytests project.	https://privac	The privacytests project	5.1.1.4.1	
5.1.1.5	1	A simple action	This kind of cookies is generally used only for tracking purposes. The majority of websites can still work without allowing their usage.	https://suppo	More information from Mozilla	5.1.1.5.1	
5.1.2.3	1	Emails are massively tracked	Hundreds of third parties track email recipients via methods such as embedded pixels, with 30% of emails that also leak the recipient IP. Additional leaks occur if recipients click on links in emails. Some third parties can link email tracking to users' web cookies.	https://doi.org	Link to the study	5.1.2.3.1	

5.1.3.1	1	Big amounts of data are shared by mobile phones	The kind of shared data depends on the operating system. Android and iOS systems transmit telemetry data even with opt-out configurations. Google collects around 20 times more mobile data than Apple. Both systems make the devices periodically connect to their backend servers with an average of 4.5 minutes, even when the device is not used. Inserting a SIM card into the device generates connections that share the SIM details with Apple/Google. Browsing activities also generate multiple network connections to backend servers. Some pre-installed applications make network connections despite having never been opened or used.	https://www.s	Link to the study	5.1.3.1.1	
5.2.1.1	1	Emails have security issues	Emails have no out of the box CIA guarantees: users must use their own tools such as PGP, but few of them actually do. Transport-layer security mechanisms can protect users' privacy, but they are limited to this layer. Sender-side protections also exist, such as DKIM and SPF.			5.2.1.1.1	
5.2.1.1	2	Some security advices	Assure TLS support, make servers checking the certificates, ensure a proper SPF enforcement, use DKIM for in the sender side, and reject invalid DKIM signatures.			5.2.1.1.2	
5.2.1.2	1	Why could they cause harm to businesses?	The main problems that instant messaging bring are security-related risks, legal-related risks, information leakages, and productivity decreases.			5.2.1.2.1	
5.2.1.2	2	Assess their security levels	Messaging application have different security levels depending on their stability, efficiency, versatility (effective and rich set of features), compatibility, scalability, simplicity, and affordability. Both their set of features and their architecture must comply with the organization needs.			5.2.1.2.2	
5.2.1.2	3	Regarding unified communication tools	Some guidelines exist to improve their security and privacy levels: enforce encryption by default and make sure it is end-to-end, lock and password-protect meetings, hold unauthenticated users in a waiting room, monitor the participant list, acquire consent from participants for meeting recordings, be aware that audio-only participants calling via a regular phone dial-in option or protocol gateways could disable the end-to-end encryption protection, be aware that file and screen-sharing capabilities could accidentally disclose sensitive information or be used to spread malicious programs. End-to-end encryption and open source architectures are two fundamental security and privacy mitigations against unified communication threats.			5.2.1.2.3	

5.2.1.3	1	An example	Fuzzy rules could be used to classify malicious emails, such as a semi-automated rule-based system that aims to fill the gaps left by other security mechanisms.			5.2.1.3.1	
5.2.1.4	1	Why is it important?	Confidential or sensitive information could be shared by employees, either intentionally or unintentionally. Such emails should be detected to avoid further issues.			5.2.1.4.1	
5.2.1.4	2	Example of a mitigation	Install a tool which parses emails content and prevents sensitive information from leaking based on emails label. If the classified security level does not reach the one of the user's email label, the message is not sent and is reported.			5.2.1.4.2	
6.1.1.1	1	Evaluating risks is difficult	Evaluating risks is difficult: assessors must follow a complete, unbiased and tested method or framework to do so. Multiple solutions might be adapted to each situation: rigorous research should be done before choosing one.			6.1.1.1.1	
6.1.1.2	1	Defining plans is great, testing them is better	Evaluating risks is difficult: assessors must follow a complete, unbiased and tested method or framework to do so. Multiple solutions might be adapted to each situation: rigorous research should be done before choosing one.			6.1.1.2.1	
6.1.3.1	1	What is a technology watch?	A technology watch consists of obtaining technical information to make decisions in a company production department. It can also be applied to commercial decision-making processes. The relevant sources must be found both internally and externally to the organization.			6.1.3.1.1	
6.1.3.1	2	Define a strategic planning	1) Analyse the internal and external activities of a company. 2) Perform a SWOT analysis. 3) Create a strategy plan, both for short and midterms. 4) Define the critical watch factors.			6.1.3.1.2	
6.1.3.1	3	The continuous and cyclic watch phases	1) Identify and analyse the company information needs by defining the critical watch factors. 2) Search and obtain the necessary information to track the critical watch factors. 3) Evaluate and analyse the obtained information. 4) Internally disseminate the results. 5) Use the information in the decision-making processes.			6.1.3.1.3	
6.1.3.1	4	Some useful tools	Service alerts, webpage software monitoring, adding agents, search agents, search engines, RSS feeds, data mining procedures, bibliographic databases, patent databases, distribution lists, and invisible web databases.			6.1.3.1.4	
6.1.3.1	5	The technology watch outputs	A technology watch can help an organization to define an evaluation of their risks, resulting with a ranked list.			6.1.3.1.5	

6.3.1.1	1	Why is it important?	Information security policies are the first step to protect organizations against attacks, and are used to implement effective enforcements towards CIA (Confidentiality, Integrity and Availability). A policy is a general rule implemented in an organization to limit the discretion of subordinates.			6.3.1.1.1	
6.3.1.1	2	Main challenges	The biggest challenges are grouped in four categories. 1) Security policy promotion: challenge on its dissemination, on how to raise its awareness, on the training, on the enforcement, and on its monitoring. 2) Non-compliance with security policy: challenges from malicious behaviour, from negligent behaviour, and from unawareness. 3) Security policy management and updating: challenges on its regular review and update, on policy management, on technology advances, and to design a good policy. 4) Shadow security: challenges on unclear security policies, on unusable security mechanisms, and on high compliance costs.			6.3.1.1.2	
6.3.1.1	3	Roles and activities needed from management	The management must be involve into five policy aspects of an organization: on the information security and management definition, on the information security policy awareness and corresponding training, on the integration of technical and managerial activities in information security management, on the human aspects of information security management, and on the information security as a business issue.			6.3.1.1.3	
6.3.1.1	4	Reduce risks in infrastructures	Policies can have lacks in their policy guidelines, in the awareness of information security threats, and in irregular monitoring of misuse behaviour. Those lacks can lead to threatening situations. A security framework to implement strategic security procedures for users should be defined to ensure both compliance with security policies and protection of vital resources. An information security culture developed in organizations can reduce the risk of security breaches and potential incidents, given that compliance with rules and regulations becomes a habit.			6.3.1.1.4	
6.3.1.2	1	The biggest impact on security compliance	Parties' compliance with policies is significantly influenced by their attitude, normative beliefs, and self-efficacy to comply with them. Policies positively affect both attitude and outcome beliefs, and an organization security compliance increases if all parties follow the policies.			6.3.1.2.1	

6.3.1.2	2	Security breaches origin	Users' poor information security behaviour is the main cause of security breaches. Such model leads to positive effects on information security awareness, information security organization policy, information security experience and involvement, attitude towards information security, subjective norms, threat appraisal, and information security self-efficacy.			6.3.1.2.2	
6.3.2.1	1	How to provide prevention?	Prevention should be included in every organizations risk management strategy, and must raise awareness within the parties. Some approaches: encourage security education and training, increase social awareness, keep confidential information safe, report suspect activities, and train new employees. Physical intrusions must not be forgotten.			6.3.2.1.1	
6.3.2.1	2	Which channel are used by attackers?	Most used channels: emails, instant messaging applications, phone calls or messages, social networks, cloud services, and websites.			6.3.2.1.2	
6.3.2.1	3	What are the most common attacks?	By using online social networks which are wealthy of personal information, by doing social phishing and context-aware spam, by using fake online profiles, by passing through cloud services using shared resources, or through mobile application vulnerabilities.			6.3.2.1.3	
6.3.2.2	1	How to train people?	A few leads to train parties: advertise them using sensitization and fraudulent emails, provide them the required detection tools and explain them, or include social engineering scenario in penetration tests.			6.3.2.2.1	
6.3.2.3	1	Part of the prevention	Based on what is explained in the prevention phase, ensure that parties have all the needed tools and processes to identify and report problems. Physical intrusions must not be forgotten.			6.3.2.3.1	
6.3.2.3	2	Most used techniques	Attackers try to gain victims' trust: some of their most used techniques are to play on reciprocation, on commitment, on social proofs, on friendliness, on authority, and on scarcity.			6.3.2.3.2	
6.3.2.3	3	A few advices	Verify the call sources, verify the emails sources, identify the most vulnerable users, and report all the attacks.			6.3.2.3.3	
6.3.2.4	1	How to detect problems?	Various technical detection exist: honeypots, anti-phishing tools, machine learning algorithms, or network monitoring.			6.3.2.4.1	
6.3.2.4	2	New issues to be mitigated	Bringing additional layers of technology creates new issues: it adds costs and complexity to the system, it increases the attack surface, and a need to find large and up-to-date datasets appears.			6.3.2.4.2	
6.3.2.5	1	Techniques-based mitigations	Some well known mitigations: limit the access to personal computers and to their USB ports, apply allowlists and blocklists, and use biometrics verification steps.			6.3.2.5.1	

6.3.2.5	2	Human-based mitigations	Some well known mitigations: destroy discarded documents, assign PINs to help desk callers, and define a ransomware policy.			6.3.2.5.2	
6.3.3.1	1	For what needs?	Workplace issues such as disputes, harassments, employee performances, and others can be disclosed by e-messages. Organizations do not know which messages are of interest for this kind of problems until issues surface and related messages are requested and restored. This raises the question of how long backups must be kept. Backups can be of two types, either online or offline of the main system.			6.3.3.1.1	
6.3.3.1	2	Beware of the different territory regulations	The biggest challenge is that expectations of privacy for company messages sent by employees vary between territories: the United States forces companies to store them, whilst the European Union states that messages are private unless a disclosure is requested with appropriate and legitimate reasons.			6.3.3.1.2	
6.3.4.1	1	Use reconciliation algorithms	Reconciliation algorithms help to define a policy that is consistent with all domain policies. If unsuccessful, requirements altering or their abstinence can be applied. Policies provisioning includes complex dependencies which include decisions about some particular aspects of the policy that can affect subsequent options. Such processes are also subject to preferential behaviours. Other reconciliation approaches exist, but are limited.			6.3.4.1.1	
2.1.5.1	1	A single package can have a big impact	Using packages from repositories bring security risks, as recent incidents shown that single packages have broken or attacked targets using software running on millions of computers. Individual packages can impact lots of projects, using for example maintainer accounts that can inject malicious code into them. A lack of packages maintenance causes many packages to depend on vulnerable code.			2.1.5.1.1	
2.1.5.1	2	An example with NPM	NPM suffers from single points of failure and unmaintained packages which threaten large code bases. One average package gives implicit trust on 79 third-party packages and 39 maintainers, which brings a large surface attack. Highly popular packages influence many other packages: often more than 100,000. Up to 40% of all packages depend on code with at least one publicly known vulnerability.	https://www.u	Link to the study	2.1.5.1.2	
2.1.5.1	3	Major security risks	The major security risks are locked dependencies, heavy reuse, micro-packages, no privilege separation (all packages have complete access to the application), no systematic vetting, and vulnerable publishing model.			2.1.5.1.3	

2.1.5.1	4	Most known threat models	The most known threat models are malicious packages, exploiting unmaintained legacy code, package takeover, account takeover, and collusion attack.			2.1.5.1.4	
2.1.5.1	5	Some potential mitigations	Raise developer awareness, give warning about vulnerable packages, do code vetting, provide training and vet maintainers.			2.1.5.1.5	
2.1.2.9	1	Include them into security requirements	Dependencies between security requirements may cause additional vulnerabilities. Those vulnerabilities should be identified using static analyses, even if they raise high false positives and miss true vulnerabilities, and security tests, which is highly precise. Security tests can be as dynamic taint analysis or penetration testing. Precise tests should be launched when software is isolated, but security requirements may be violated on interactions. Up to 70% of total software errors are caused by interacting requirements. 20% of most dependent requirements are responsible for 75% of all dependencies. Another approach is to use automated requirements traceability based on information retrieval algorithms.	https://doi.org	Link to the study	2.1.2.9.1	
3.4.2.1	1	Allow users to be in control of their data	The goal is to enable users to be in full control of their data. The Solid project implements this approach: user data is stored in web-accessible personal online datastores named pods. One or more pods can be used and easily switched across different providers. Applications can get access to the data using well-defined protocols, a decentralized authentication and access control mechanism to guarantee data privacy. This technology allows similar applications switching, applications on multiple platforms, and the advantages of decentralized architectures.	https://solidp	Solid project	3.4.2.1.1	
1.3.4.1	1	Non-compliance includes high risks	Some risks when personal information is not well handled can cause legislative penalties, brand and reputation erosions, or even lawsuits.			1.3.4.1.1	
1.3.4.1	2	Be aware of the privacy phases	The OECD defined what are the privacy phases: notice, collection, cataloguing, control, release, recording, response.	https://www.c	Privacy as seen by the OECD	1.3.4.1.2	
1.3.4.1	3	Data management building blocks	1) Deploy a policy to the ICT systems. 2) Record the consent of end users. 3) Enforce the privacy policy and create an audit trail of access to privacy-sensitive information. 4) Generate both enterprise wide and individualized reports showing accesses to privacy-sensitive information and their conformance to the governing privacy policy.			1.3.4.1.3	

6.3.4.2	1	How to do it?	Two solutions can be used. The first is to map the user collaborative policy specification to an auction based on the Clarke-Tax mechanism. This approach selects the privacy policy that maximizes the social utility using truthfulness among co-owners. The second solution is to apply data co-ownership. The potential owners of posted data can be identified using tagging features or files metadata.			6.3.4.2.1	
6.3.4.2	2	Requirements	Some requirements must be met to enable valid collaborative privacy management: must ensure content integrity, must be semi-automated, must be adaptive, and must integrate group-preference.			6.3.4.2.2	
1.3.4.2	1	How to integrate them?	Data protection can be enforced by either the data owner side or the provider side. Different schemes for representing personal data and policies exist, such as P3P, CPExchange, and DISCREET.			1.3.4.2.1	
1.3.4.2	2	An example of implementation	Hierarchical categories can be defined to organize personal data, including some sub categories. The related policy components are principals (entities), data (every single item), purpose (entitles principals to retrieve data), and usage restrictions (limit access rights). The policy includes the usage of licences that define the data involved, the valid purposes of data retrieval, and the rules to provide full or restricted access. Contracts that hold arbitrary sets of licences are also defined.	https://doi.org	Link to the implementation	1.3.4.2.2	
2.1.6.1	1	Similar to the “privacy by design” principle	In order to avoid to design interfaces that include dark patterns, some rules must be applied in the development process: proactive privacy instead of reactive, privacy as the default setting, privacy embedded in design, ensure full functionality, enforce end-to-end security, assure visibility and transparency, and guarantee respect for user privacy. Privacy considerations must be included in the entire development process. Some strategies take advantage of the psychological constitution of human beings, which often cause users to not have the motivation or opportunity to resist them. Dark patterns are not always intentional.			2.1.6.1.1	
2.1.6.1	2	Understand their characteristics	The most relevant characteristics of dark patterns are that they are asymmetric, covert, deceptive, hides information, and restrictive.			2.1.6.1.2	
2.1.6.1	5	They use human biases	The human biases that are used are anchoring effects, bandwagon effects, default effects, framing effects, scarcity biases, and sunk cost fallacies.			2.1.6.1.5	
2.1.6.1	6	Be aware of third parties	Third-party entities can cause implementation of dark patterns, by integrating their in software through libraries or external resources.			2.1.6.1.6	

2.1.6.1	7	Also in mobile applications	Based on a study, 95% of mobile applications contain one or more dark patterns. Most of the time, users can not perceive the presence of malicious designs.	https://dl.acm	Link to the study	2.1.6.1.7	
2.1.6.1	8	The power of dark patterns	Users exposed to mild dark patterns are more than twice as likely to sign up for a dubious service than others. Users in aggressive dark pattern conditions are almost four times as likely to subscribe. Aggressive dark patterns generate a powerful backlash, mild dark patterns do not. Less educated users are more susceptible to mild dark patterns than their well-educated ones. Some legal frameworks exist for addressing dark patterns, such as one provided by the Federal Trade Commission in the United States.	https://www.f	Link to the FTC Report	2.1.6.1.8	
2.1.6.1	3	The dark patterns categories	Nagging, social proof (activity messages, testimonials), obstruction (roach model, price comparison prevention, intermediate currency, immortal accounts, difficulties to cancel actions), sneaking (sneak into basket, hidden costs, hidden subscription/forced continuity, baits and switch, bad defaults), interface interference (hidden information/aesthetic manipulation, pre-selection, toying with emotion, false hierarchy/pressured selling, trick question, disguised ad, confirm shaming, cuteness, hidden legalese stipulations, user profiles shadowing), forced action (friend spam/social pyramid/address book leeching, privacy zuckering, gamification, forced registration or enrolment), scarcity (low stock message, high demand message), urgency (countdown timer, limited time message), misdirection (confirm shaming, visual interference, trick questions, pressured selling).	https://privac	Used patterns from privacypatterns Europe	2.1.6.1.3	
2.1.6.1	4	The privacy design categories	Minimize, hide, separate, aggregate, inform, control, enforce, demonstrate.	https://privac	Used patterns from the privacypatterns organization	2.1.6.1.4	
3.2.3.1	1	Risks on the infrastructure assembly	The physical threats can be avoided by testing the components, by using TPM, or by making audits. The risks brought by software and human resources that fail to meet the promised standards or compromised can be mitigated using various techniques: define multiple admins, limit administrators' access, and carry out background checks of employees.			3.2.3.1.1	

3.2.3.1	2	Some contractual threats	Cost-overrun attacks, can be avoided by setting quotas or ensuring that the provider absorbs bulks, deceptive billing, avoidable by enabling tenants to do their own infrastructure tests or by reporting resource consumption, captivity, avoidable by ensuring providers homogeneity and by reviewing long-term contracts cost prediction, or bankruptcy, users must be assures that their would still have rights to access the infrastructure and that minimal funds are guaranteed to continue short operations.			3.2.3.1.2	
3.2.3.1	3	Legal Threats	Can create indirect legal coercion, secret search, or direct and indirect jurisdictional exposure. Can be avoided by enabling data location choice.			3.2.3.1.3	
3.2.3.2	1	Threats from other tenants	Threats can be brought by direct breaches, mitigated by hypervisor and network isolations, by side channel attacks, mitigated using the same isolation techniques, or by denial of resources, resource thefts, and collateral damage to shared reputation. Those last threats can be mitigated by securing the mapping between communications and tenants.			3.2.3.2.1	
3.2.3.2	2	Threats from legislation	Caused by various jurisdictional collateral damages.			3.2.3.2.2	
3.2.3.2	3	Threats on availability and costs of shared resources	Caused by under provisioning, avoidable with attestation-based audit mechanisms and spare capacity audits, or by collateral denial of shared resources, avoidable using resource quotas.			3.2.3.2.3	
3.2.3.2	4	Threats caused by diminished audit, detection, or incident response capabilities	Can be caused by forensic restrictions, can be avoided by forcing providers to investigate breaches.			3.2.3.2.4	
3.2.1.4	1	How to assess the security levels?	Providers have five goals to achieve an adequate security: ensure availability, confidentiality, data integrity, control and audit. Some legal issues can be mitigated by creating additional roles from cloud infrastructures and by great handling of third parties. Some acts fail to protect user privacy from the government and third parties in a cloud environment. Multi location can bring issues in a legislative perspective.			3.2.1.4.1	

3.2.1.4	2	Some major security challenges and their mitigations	Inside threats, avoidable by creating adapted employees' governance, access control issues, can be mitigated by enabling additional authentication factors or by creating confidence between provider and tenant, and system portability issues, avoidable by avoiding provider link-in or by using open standards. The software security issues are caused by virtualization technologies, which can be mitigated by applying updates and keeping tenants isolated. On the host OS side, it can be avoided by choosing a simple and minimalistic OS. On the guest OS side, issues can be mitigated by giving tenant responsibility and informing them about risks and weak data encryption.			3.2.1.4.2	
3.2.1.2	2	Multiple aspects to review	How are handled the data security, the regulatory compliances, the user authentication, the data separation, and the legal issues. Providers' certifications must be reviewed: it could include SAS70 Type II, PCI DSS Level 1, ISO 27001, or FISMA certifications.			3.2.1.2.2	
3.2.1.2	3	Other concerns	The employee life cycle policies must also be reviewed: how are defined the account provisions, account reviews, access removals, and password policies. The business continuity management must also be known, such as the provider's availability, incident response, and company-wide executive review. Finally, the network security should be considered, with mitigations enforced for DDoS, man in the middle, IP spoofing, or port scanning attacks.			3.2.1.2.3	
3.2.1.4	4	Designing an appropriate service model	It should handle security challenges such as malicious attacks, backup and storage issues, service hijacking, and VM hopping.			3.2.1.4.4	
3.2.1.4	5	Designing an appropriate deployment model	It should handle security challenges like PaaS security issues, third-party relationships management, development life cycle issues, underlying infrastructure security, cloning and resource pooling, unencrypted data issues, authentication and identity management, network issues, XML signature element wrapping, browser security, flooding attacks, and SQL injection attacks.			3.2.1.4.5	
4.1.1.1	1	Multiple techniques	Various techniques exist to this end, such as K-anonymity, L-diversity, T-closeness, HybrEx model, privacy-preserving aggregation (homomorphism), differential privacy or identity-based anonymization.			4.1.1.1.1	
4.1.1.2	1	How to implement them?	The data generation phase must restrict the access to data and allow data falsification.			4.1.1.2.1	

4.1.1.3	1	How to implement them?	The data storage phase must perform attribute-based encryption, enforce homomorphic encryption, encrypt storage paths, use hybrid clouds, and allow data integrity checks.			4.1.1.3.1	
4.1.1.4	1	How to implement them?	The data processing phase must be able to extract information without violating user privacy using de-identification, PPDP techniques, privacy preserving clustering or classification, and association rule mining techniques.			4.1.1.4.1	
4.1.1.5	1	How to implement them?	Apply anonymization techniques: K-anonymity, L-diversity, T-closeness, HybrEx model, privacy-preserving aggregation (homomorphism), differential privacy or identity-based anonymization.			4.1.1.5.1	
4.1.1.6	1	The most known legal principles	Some legislations regulate user privacy, but each countries have different policies and laws. Some principles are requested in regulations to protect any personally identifiable information: lawfulness, consent, purpose limitation, necessity and data minimization, transparency and openness, individual rights, information security, accountability, and data protection by design and by default.			4.1.1.6.1	
6.4.1.1	1	Three sources of problems	First, insecure configurations into web services can remain widespread for over a decade. Secondly, introduction of best practices only affects moderately the decline of insecure configurations. However, publicizing highly security flaws have a significant impact on awareness. Thirdly, economic incentives on website owners to provide secure services are too weak. Other levers of influence as legislation or blocking non-compliant sites have a bigger impact.			6.4.1.1.1	
6.4.1.2	1	Adopt best practices to ensure information security	The ISO 17799 document answers questions such as what standards should an organization implement to achieve their information security objectives, or what management practices are perceived as critical by information technology professionals. It is widely accepted and recognized as best practices being applied by information security professionals. Most of the security dimensions and items covered under this document are highly valid. This resource has nowadays been replaced by the ISO 27002 document with updated content.	https://www.i	Link to the ISO 27002 document	6.4.1.2.1	

1.4.1.1	1	Questions to answer	Policies should be part of the representation of (semantic) web services and respond to a bunch of questions, such as who can use a service under which conditions, how information should be provided to the service, and how provided information will be used later. Those policies should be of different kinds: privacy policies, that define under what conditions information can be exchanged and what are the legitimate uses of that information, and authorization policies. Single requests can have policies of their own.			1.4.1.1.1	
1.4.1.1	2	A possible approach	Ontologies and markup are some proposed approaches to capture security information of web service input and output parameters. Policies can be transformed into informal contracts that also include a prioritization mechanism to resolve conflicts. Providers can be discovered and selected using the policies. A way of enforcing privacy and authentication is to use encryption standards for communication independently of the transport protocol security.			1.4.1.1.2	
1.4.1.2	1	Check the OAuth implementations	OAuth allows users to grant access to their resources, which can be data or services, at other websites. This operation is called an authorization. Its central security properties are authorization, authentication, and session integrity. Four exploitable attacks have been found, but mitigations are given for new and existing deployments: multiple new RFCs have been drafted from the respective working group, with guidelines to secure OAuth implementations. A complete security model is given to enforce OAuth processes.	https://arxiv.c	Link to the security model	1.4.1.2.1	
1.2.2.1	1	Could be an entry door	Tests were realized on personal banking websites using security questions as a lost password retrieval process: many process rely partially on security questions with serious usability and security weaknesses. The hardness of this method is weakened as personal information becomes ubiquitously available online. 17% of users' security answers can be found by their acquaintances. Users forget 20% of their own answers within six months, and 13% of answers can be guessed within five attempts by guessing the most popular answers of other participants. A single personal question is not sufficiently secure for authenticating users. User-written questions could be harder to attack, but only if they are sufficiently private and unpopular. The proportion of popular questions should be reduced.	https://ieeexp	Link to the study	1.2.2.1.1	

1.2.2.1	2	Two kinds of security questions	Two kinds of security questions exist: sensitive questions, which are not necessarily private, and personal questions, related to users' background or to their family. Allowing users to define their own questions is not very common. Alternatives exist such as email-based resets which are often considered as secure, use data already held by the organization which implies that the level of security depends on the nature of the source, or asking for a series of preference judgements, a technique not very used in the industry. Personal questions are more secure than the sensitive ones because of questions being more varied and because public leaks of sensitive data are less irrelevant for the questions asked.			1.2.2.1.2	
1.2.2.1	3	Some attacks	Automated attacks must be blocked, by using for example CAPTCHAs. Some well used attacks: random guessing, automatically using online information, dedicated human attackers, and personal acquaintance.			1.2.2.1.3	
1.2.2.1	4	Some weaknesses	The biggest weaknesses in personal security questions are that they are inapplicable, not memorable, ambiguous, guessable, attackable, and automatically attackable. Furthermore, users treat memorability rather than security as the dominant factor in choosing security questions.			1.2.2.1.4	
1.2.2.1	5	Some mitigations	Some mitigations can be enforced, such as survey distribution of answers, users' education, usage of ephemeral answers, and ask users for durable and offline answers.			1.2.2.1.5	
1.2.3.1	1	What are they?	Biometric systems recognize individuals based on their anatomical or behavioural traits. They are used to ensure that only legitimate or authorized users can get access to an entity. Their unique advantages are their deterrence against repudiation, and their multiple identity detection. Biometric systems rely on similarities between two biometric samples, not on their perfect match: challenges can lead to false non-matches or false matches.			1.2.3.1.1	

1.2.3.1	2	Some attacks	Biometric systems match approach leads to vulnerabilities such as denials of service, with legitimate users being not recognized, or intrusions, with impostors being incorrectly identified as legitimate. Multiple adversary attacks exist: coercing or colluding with insiders, exploiting insiders' negligence, manipulating the procedures of enrolment and exception processing, direct attacks on sensors, feature extractor, or matcher module. Those attacks can be carried out using trojan horses, man in the middle attacks or replay attacks. They are also applicable to password-based authentication. The major vulnerabilities are spoof attacks on user interfaces and template database leakages.			1.2.3.1.2	
1.2.3.2	1	Must be answered	Biometric systems include some major issues that need to be answered: who own biometric data? Is this usage proportional to the need? What is the optimal trade-off between service security and user privacy?			1.2.3.2.1	
1.2.4.1	2	Multi-factor authentication is a combination of different authentication factors	Choosing the adequate authentication schemes or methods depends on the contexts. The authentication factors come from knowledge, what users know, from possession, what they physically own, or from inherence, what users are. The combination of the knowledge and possession factors is very predominant in multi-factor authentication methods. Three-factor authentication is well researched but less applied. For both methods, the combination of text passwords and smart cards is the most popular.			1.2.4.1.2	
1.2.4.1	3	Compare and select schemes	The comparison and selection of schemes are made with usability, security and cost-related criteria. Some frameworks can help in the decision of authentication schemes or methods, according to different contexts.	https://www.r	Link to one of the frameworks	1.2.4.1.3	
1.2.4.1	1	Some details about multi-factor authentication	Digital multi-factor authentication is one of the best methods to implement a secure authentication, but it can be frustrating for users. Some greatly used multi-factor authentication methods are fingerprints and user-specific random projection, threshold cryptography (OTP approach), multi-modal biometrics, or cloud-based infrastructure. The latter can use a third party's authentication. Different entities can be used for authentication, such as smart cards, OTPs, cryptographic techniques, multi-modal biometric systems, or tokens.			1.2.4.1.1	

1.5.1.1	1	Some principles must be followed during system design	First, the portions of the application must be split into isolated components with isolation boundaries. Then, the amount of privilege given to each component must be minimized. Finally, each component required privileges must be inferred using dynamic analysis, which is an automated version of the least privilege pattern.			1.5.1.1.1	
4.1.2.1	1	How to adapt them?	The majority of platforms have basic access control mechanisms, which leads to multiple problems. Unconstrained access is given to high volumes of data from multiple data sources, some sensitive and private data is illegitimately accessible, and advanced analysis and prediction capabilities are limited. Multiple requirements must be met for better access control: define fine-grained access control, allow context management, and guarantee the efficiency of access control without any compromises on the platform usability.			4.1.2.1.1	
4.1.2.1	2	Some issues still need to be resolved	Some issues are still open in the research field: how to unify the access control models and mechanisms, how to provide policy analysis tools, how to ensure GDPR fulfilment, how to comply with federated environments, and how to define appropriate access control for streaming analytic, including adaptation for continuous flows.			4.1.2.1.2	
1.5.1.2	1	ABAC is seen as the most appropriate for web services	ABAC is a logical access control model that controls access to objects by evaluating rules against the attributes of entities (subject and object), operations, and the environment relevant to a request. It allows a high amount of inputs in the evaluation process, which brings an almost infinite amount of possible combinations. The relationships are not modified if updates must be done on access decisions, only the attributes are altered. The NIST has published the SP 800-162 to help companies to understand and implement the ABAC model. However, it can be complex to apply in large organizations.	https://www.r	NIST SP 800-162	1.5.1.2.1	

1.5.1.2	2	More details on ABAC	ABAC is both mandatory and discretionary, and it can not predict how data must be shared in SOA environments: it is ad hoc and dynamic in nature. Web services have rich semantics, which means that simple, static, and coarse-grained access control models should be avoided. Two access control models exist. The first one is DAC, which can restrict access to objects based on the identity and need-to-know of entities. The permissions can be passed from a subject to other entities. The second one is MAC, which can restrict access to objects following fixed security attributes given to users and objects. The controls are system-enforced, and it can not be modified. Both models can be used in conjunction.			1.5.1.2.2	
1.5.1.2	3	Other models	The IBAC model uses permissions linked to identities. The RBAC model uses permissions linked to business functions or roles, including levels of indirection. The LBAC model solves the MAC problem of non-modification by using an ordered set of security labels combined with a set of categories. However, it has a lack of flexibility and scalability. Two main aspects are defined within ABAC: the policy model, which defines policies, and the architecture model, which applies the policies. The ABAC model defines permissions on any security relevant characteristics (attributes), includes both IBAC and RBAC functionalities and is more flexible with the attribute approach. Compared to the other models, ABAC is intuitive, more flexible and powerful, the security management can be distributed, and it uses a divide and conquer approach.			1.5.1.2.3	
1.5.1.2	4	ABAC is useful in decentralized collaborative systems	Access control based on identity can be ineffective if entities do not know each other. ABAC systems have multiple capabilities. They can handle decentralized attributes, using entity asserts that another entity has a certain attribute. They can give delegations on attribute authority, which allow to trust another entities judgements. ABAC systems can control the inference of attributes and attributes fields. Finally, they handle attributes-based delegation of attributes authority, which gives them the ability of delegating to strangers whose trustworthiness is determined based on their own certified attributes.			1.5.1.2.4	

1.5.1.2	5	ABAC limitations	No standardization of ABAC has been published, but an acceptation of high level descriptions (NIST SP 800-162) has been accepted into the community. Some problems are caused by its infancy. No references are made to foundational models. The capability of emulating ABAC models has only been demonstrated informally in research context. The support of hierarchy is lacking, which is emulated by either using complex data types in attributes or by unmaintainable complex policies. A solution would be to use attribute user groups. Compliance is complicated to prove during audits: it would be simpler with hybrid models. The separation of duties is still unclear in research. The delegation feature is limited, must be done in the implementation. The attribute storage and sharing make it hard to evaluate trustworthiness of attributes and their compatibility when multiple attribute sources exist. It would require a commonly accepted namespace or ontology. Its scalability must still be proven. The administration and user comprehension must be understood. Formal security analyses can be difficult to realize: some tools are compatible, but none is specialized for the ABAC model.			1.5.1.2.5	
1.3.5.1	1	Sensitivity rules can be used	Sensitivity rules are used to decide whether table cells are sensitive or not, which means that sensitive cells must not be published. Examples have shown that publishing non-sensitive cells may also disclose sensitive information. An a priori assessment on disclosure risks must be made using sensitivity rules, such as (n, k)-dominance, pq-rule or p%-rule. This is explainable by the fact that disclosure risks of contributions increase as the percent within which they can be estimated by an intruder decreases.			1.3.5.1.1	
1.3.5.1	2	Alternatives	Two alternatives to sensitivity rules are entropy-based sensitivity rule, and complement the a priori risk assessment using a posteriori assessment.			1.3.5.1.2	

1.3.5.2	1	Different terms and definitions	Anonymity of a subject from an attacker's perspective means that the attacker can not sufficiently identify the subject within a set of subjects, know as the anonymity set. Unlinkability of two or more items of interest from an attacker's perspective means that within the system, the attacker cannot sufficiently distinguish whether these items are related or not. Linkability is the negation of unlinkability. Undetectability of an item of interest from an attacker's perspective means that the attacker cannot sufficiently distinguish whether it exists or not. Unobservability of an item of interest means that the item is not detectable against all subjects not involved with it. A pseudonym is an identifier given to a subject that is different from of the subject's real names.			1.3.5.2.1	
1.3.3.3	1	An example with GDPR	If the data is anonymized, the GDPR is not applicable. However, there is still a risk of data being not fully anonymized, and no clear requirement is given in the regulation. If the data is pseudonymized, there is no precise legal consequences. Pseudonymization has no clear and immediate legal advantages.			1.3.3.3.1	
1.3.5.2	2	Some approaches	Five approaches have been found for anonymization into electronic health services. Data anonymity, which assures that no relationship can be made between users and their data. User anonymity, guarantees that messages do not give information about their users' identity. Communication anonymity, which hides the link between users and the system. This technique can use onion routing systems like Tor. Ensure unlinkability between users' exchanges. Usage of differential privacy by adding noise in the data.			1.3.5.2.2	
1.3.5.2	3	Additional techniques	The generalization technique replaces data values with less specific ones, but keeps them semantically consistent. The suppression technique removes entire parts of data. The swapping technique randomly rearranges the variables. The masking technique changes the characters in attributes. The distortion technique changes the data itself, with a possibility of being reverted. Some techniques are more suitable for specific types of variables.			1.3.5.2.3	

1.3.5.3	1	Multiple techniques	The values of sensitive attributes can be recovered if they have little diversity. Privacy can not be guaranteed against attackers who have some background knowledge. The main mitigation is to use an extension of k-anonymization named l-diversity which adds diversity in data groups attributes. The t-closeness approach is an additional approach to k-anonymization and l-diversity. It expands the l-diversity by reducing the granularity of data representations.			1.3.5.3.1	
1.3.5.3	3	Slicing to preserve privacy	The k-anonymity technique looses considerable amount of information, especially for high-dimensional data. Bucketization does not prevent membership disclosures and breaks attribute correlation between sensitive attributes and quasi-identifiers. The slicing technique partitions data both horizontally by grouping tuples into buckets and then randomly permuting them, and vertically by grouping attributes into columns based on correlations. Slicing has a better data preservation utility compared to generalization, can be used for membership disclosure protection, can handle high-dimensional data, and can respect l-diversity requirements.			1.3.5.3.3	
2.3.1.1	1	Why is it important?	API providers must define the terms of service and privacy policies for developers that will use the said API. Developers can then assess services compatibility, avoid breaches and mitigate the threats of termination for non-compliance.			2.3.1.1.1	
2.3.1.1	2	What are the terms?	The terms should at least include the guaranteed SLA level, the conditions to agree to before usage, the privacy policies, the indications on terms changes, the liability, and third parties usage conditions.			2.3.1.1.2	
2.3.1.1	3	What are the privacy policies?	Privacy policies are defined as the channel through which internet services communicate to their users the data they collect from them and what it is used for. Users can either accept them, which means that they lose control of their data but obtain an access, or reject them, which guarantee them to keep control but without any granted access to the API. Privacy policies define provider's terms that API users must comply to.			2.3.1.1.3	
2.3.1.1	4	What are the common issues with terms?	All or nothing, lack of alternatives, legibility, changes in terms, technical issues, liability, and restrictions in terms.			2.3.1.1.4	
2.3.1.1	5	What are the common issues with privacy policies?	Issues with permissions, changes in policy, and technical issues.			2.3.1.1.5	

2.3.3.1	1	Increase transparency and legibility for users	If data is shared with third parties, an API can be given to users for them to consult how their own data is being shared. Users should be able to decide if a service is worth to be used by knowing how and what data is shared. They could accept or not such sharing thanks to an assessment of the value of their data, which is difficult to guess without knowing what is shared. Three limitations appear with such system: the sharing retro-activity must be handled, the users must use the system to have access to the API, and it does not show internal usage of data.			2.3.3.1.1	
2.3.2.1	1	Different scopes	An API can be one of three types: private with a closed access, for partners, designed with efficient access control and authorization mechanisms including rules and policies, or public, which brings potential security threats.			2.3.2.1.1	
2.3.2.1	2	Different approaches	Usually, an API is implemented by following either REST or SOAP approaches. SOAP is more adapted for sensitive data.			2.3.2.1.2	
2.3.1.2	1	Why implementing machine learning security?	Machine learning security can fill various security gaps such as addressing new threats, identifying past attacks behaviour, or making predictions. However, it must be compliant with regulations, which restricts automated decision-making and profiling. It also causes an increase the costs.			2.3.1.2.1	
2.3.1.2	2	An example with the GDPR	This regulation requires explaining details of algorithmic decisions, ensuring right of data portability, ensuring the trade-off between algorithmic transparency and accuracy, and allowing users' right of data erasure. Automated decision-making is prohibited without human intervention with the need to be transparent to users. This issue brings new technical challenges, particularly on how to explain those black-boxes to users and on intellectual properties. The data can be localized anywhere, but the in-house approach is a more appropriate solution compared to public clouds. In general, data processing needs consent of data subjects.			2.3.1.2.2	
2.3.2.3	1	What is it?	Security chaos engineering can be used to both expose vulnerabilities and enhance security. Multiple techniques exist to detect automated attacks, such as monitoring the traffic, applying a quota management, applying allowlisting, or implementing traffic throttling. HTTP header fields can be used to achieve code injection attacks. Chaos engineering is a method that simulates unpredictable failures to make systems more resilient.			2.3.2.3.1	

2.3.2.3	2	How can it help?	DDoS attacks are difficult to identify: each malicious client sends normal traffic volume, while adapting the said volume by detecting rate-limiting controls to avoid any detection. Bots can be detected by searching for patterns such as abnormal behaviour, persistent attempts, unusual error rates, suspicious client requests, or by using machine learning models. Those models need historical data and more research to achieve greater results.			2.3.2.3.2	
2.3.2.3	3	How is it implemented?	Chaos security applies empirical exploration to verify how a system behaves. It is implemented by building a hypothesis around steady-state behaviour, varying real-world events, running experiments in production, automating experiments to run continuously, and minimizing blast radius.			2.3.2.3.3	
2.3.2.2	1	Major vulnerabilities	APIs major vulnerabilities are script insertions, SQL injections, bound of buffer overflows, DDoS attacks, login attacks, application or data attacks, eavesdropping, leakages of sensitive information, code injections, man in the middle attacks, API hijacking, replay attacks, brute forcing credentials, broken authentications, usage of vulnerable components, and improper usages of CORS.			2.3.2.2.1	
2.3.2.2	2	Some mitigations	Some security models can help to mitigate those vulnerabilities, such as authentication, throttling, communication security, or anomaly detection. The access control management can be enforced following the OAuth or OpenID standards. Communication security can be enforced using HTTPS for JSON transfers for the REST approach, or by using web services security and XML built-in security for the SOAP approach. Client throttling can be implemented in order to avoid attacks. The gateways security can be enforced by performing message analysis, by granting access tokens and authorization parameters, by acting like a traffic police, and by only authorizing legitimate users. A common mistake is to limit access to the API instead of mitigating the attacks.			2.3.2.2.2	

2.3.2.2	3	Design advices to harden security	Those advices are mainly focused on the network channels. The major mitigation techniques are to ensure separation of entities, strong authentication, strong authorization, strong encryption, strong access control, apply access revocation, validate the messages, enforce logging, enforce input validation, enforce input sanitization, set up rate limits, set up redirections, do appropriate testing, realize design reviews, ensure high availability, define great role engineering, regulate the traffic, enable load balancing, set up service degradation, and ensure proper monitoring.			2.3.2.2.3	
2.3.2.2	4	Implementation advices to harden security	Multiple patterns can help such as the principle of least privilege, parameter forest, one factor security, two factor security, three factor security, client-server basic security, using an API gateway, defence in depth, default denial, command pattern, and data minimization. Multiple methods can be used during implementation: the most used and appropriate are token-based authentication, digital signing, RBAC, ABAC, token-based authorization, and multi-factor authentication.			2.3.2.2.4	
2.3.2.2	5	Use thread modelling	Threat modelling can be done using various schemes, such as STRIDE, DREAD, OSSTMM, sequence diagram, use case, user story, NIST guide to cybersecurity, or OWASP testing guide.			2.3.2.2.5	
2.3.2.2	6	Three categories of attacks	Post-login attacks, that aim for data and the application, pre-login attacks, which use authentication services, credential stuffing, fuzzing, or stolen credentials, and fundamental API security attacks, using resources such as access control, tokens, authorization, authentication, rate limiting, client throttling, quotas, network privacy, or TLS configuration issues.			2.3.2.2.6	
4.2.1.1	1	The issue of explaining models	The GDPR mandates a right to explanation on decisions made by automated or artificially intelligent algorithmic systems, which legally binds the data controller to provide explanations about artificial intelligence tools to requesting citizens if their personal data is used. There is therefore a need for interpretable and explainable models in order to justify their decisions. Deep learning models can be compared to opaque black boxes: such systems are not capable to self-explain their operating processes.			4.2.1.1.1	

4.2.2.1	2	Multiple techniques for federated learning	The federated learning model is composed of terminal devices that use their own data for their training phase. However, the user privacy must be assured when the results are shared between terminals, and model manipulation and/or stealing must be prevented. There are various ways of building safe distributed models: one of them is to avoid gradient leakages using homomorphic encryption. This method adds a large computational overhead. Otherwise, a federated learning environment can be built using an aggregation protocol which securely computes the sum of parameters computed by devices. Otherwise, machine learning classifications can be done over encrypted data, but it lowers the models accuracy.			4.2.2.1.2	
4.2.2.2	1	How is it used?	The goal is to enable collaborative learning on a neural network using the local dataset of all participants, without actually sharing the data. All participants compute their local gradients by training their local model, and then send a portion of their gradients to a central server. The latter use, for example, additively homomorphic encryption and asynchronous SGD to compute a general model which is then shared. However, a trade-off must be taken care of between accuracy and privacy, which consists of finding the correct amount of local gradients to share. Because a small faction of gradients can leak useful and therefore private information, homomorphic encryption is used to enable computation on the data without being able to know its value as a plaintext. This approach has three effects. On the security side, the central server can not leak any data. On the accuracy side, an identical accuracy is achieved compared to a corresponding model trained on a global dataset built from the joint local datasets. Finally, on the overheads side, an increase in communication is caused by the sharing of the gradients, and a greater computation time is required to achieve the same model accuracy.			4.2.2.2.1	
4.2.2.1	1	Some privacy-preserving models	Three major private learning schemes exist: apply homomorphic encryption on data or model, apply obfuscation using differential privacy, which adds noise, and apply aggregation, which guarantees that parties keep their own dataset private whilst still being able to learn collaboratively. Secure MPC or TEEs can also help. Those techniques can bring one or multiple drawbacks, such as a significant increase of the computational overhead, or they can require customizing specific incompatible models.			4.2.2.1.1	

4.2.2.3	1	Five benefits	The calibrated randomization embedded in differential privacy brings benefits to some AI algorithms because of multiple properties: it preserves privacy, which is its original purpose, it improves stability, thanks to the unchanged model output probability if an individual record is changed, it brings better security by reducing the impact of malicious participants, it guarantees fairness by re-sampling the training data from the universe, and it enables composition, which means that any step that satisfies differential privacy principles can be integrated in the algorithm. All properties do not have the same effect on the different types of artificial intelligence.			4.2.2.3.1	
4.2.2.3	2	For machine learning	For machine learning, differential privacy preserves privacy and improves both stability and fairness. In the other hand, an optimal trade-off between privacy and utility needs to be found and optimized. Furthermore, it is only suitable for loss functions that do not contain any regularization steps. Moreover, some situations do not have sufficient knowledge of the utility of each sample, which is needed by the exponential mechanisms of the re-sampling step.			4.2.2.3.2	
4.2.2.3	3	For deep learning	Regarding deep learning models, which include both distributed deep learning and the federated learning model, differential privacy can be applied locally. A global implementation would not protect the system against an attacker pretending to be trustful. It can also be used to destroy redundancy in order to avoid model inversion attacks. More specifically for federated learning, an aggregate of re-weighted loss functions can be used with clients having different weights to improve their learning accuracy by joining their knowledge using differential privacy to make different model updates according to client's requirements.			4.2.2.3.3	
4.2.2.3	4	Deep learning problems	Deep learning models require massive data collection, including sensitive user data which is kept indefinitely. A system can be designed to learn a model without sharing input datasets, using a characteristic from SGD that allows it to be parallelized and executed asynchronously. Only small subsets of key parameters are exchanged, whilst improving accuracy with external data without having access to them. Evaluations have shown an accuracy close to centralized models, with a negligible utility loss. The neural network parameters leak risks can be mitigated using differential privacy on their updates, thanks to the sparse vector technique.			4.2.2.3.4	

descriptions

4.2.3.2	1	Most used privacy attacks	The most used privacy attacks are model extraction, which duplicates the model parameters or hyperparameters, model inversion, which infers sensitive information by utilizing available information, (re)identification, inference, which allows to illegitimately gain knowledge, and linkage, which gathers information by correlating data sources.			4.2.3.2.1	
4.2.3.2	2	Some mitigations to those attacks	Some privacy protection schemes exist, such as obfuscation, anonymization, reducing information sharing, cryptography, privacy risk assessment and prediction, personal privacy management assistant, and private data release, which consists of publishing data with a guaranteed privacy.			4.2.3.2.2	
4.2.3.1	1	Most known security threats	Some well-known security threats are brought by adversarial attacks, which are invisible perturbations that mislead predictions, and the ones brought by poisoning attacks, which add training data pollution crafted by adversaries that misclassifies malicious samples or activities.			4.2.3.1.1	
4.2.3.1	2	Most used security attacks	The most known model attacks are model extraction, feature estimation, membership inference, and model memorization.			4.2.3.1.2	
4.2.3.1	3	Some mitigations for security attacks	Multiple defences have been found for adversarial attacks: apply input pre-processing, which reduces the influence of immunity, enable malware detection, which introduces regulations, adversarial training, feature denoising, models robustness improvement, or models modification and retraining, or improve the model robustness by detecting attacks using stateful detection, image transformation detection, or adaptive denoising detection. Two defences for poisoning attacks has also been found, such as outlier detection mechanism, which removes outliers outside the applicable set, and improving the neural networks robustness.			4.2.3.1.3	

4.2.3.3	1	The five phases of the life cycle	Different categories of threats exist during the data collection phase. It could be software-based, with data biases, fake data, data breaches, or it could be hardware-based using sensor spoofing. The data pre-processing phase is mainly concerned by scaling attack with images. Some mitigations include data randomization, quality monitoring, or image reconstruction. The training phase has two major threats: poisonous data injection combined with availability attacks, which deteriorate the general performances of the model, and integrity attacks, which only deteriorate specific inputs. Some mitigations exist, such as data sanitization, robustness training, or certified defences. Regarding the inference phase, the biggest threat is evasion attacks, that degrade or interfere the predictive performances using adversarial attacks that alter the input without changing the targeted model. Some mitigations can be used, such as distillation, detectors, network validation, adversarial training, data randomization, or input reconstruction. Finally, the integration phase includes threats to the confidentiality of the model or on the data, vulnerabilities brought by the code, artificial intelligence biases, and generic ICT threats.			4.2.3.3.1	
4.2.3.4	1	The biggest impact on the process	The authors found multiple vulnerabilities, such as outsourced training procedures, usage of pre-trained models that include intellectual properties, or unvalidated data sources coming from third parties. One example based on those vulnerabilities is an adversarial attack that uses incomplete training data, or that use overfitting and influence mechanisms to recover the sensitive data used for training. Some major security threats have been found by the authors. Data poisoning can lead to mislead predictions. Backdoors implemented in training data can lead to misclassifications for specific trigger conditions. Adversarial attacks can be realized, either in an error-generic way which make models go wrong, or by an error-specific way that makes misclassifications based on adversarial examples. Model extraction attacks can be done in order to steal the model by observing the output labels and confidence levels with respect to used inputs. A recovery of sensitive training data can be realized using membership inference to determine if a sample is used in the training phase, or by inversion attacks that infer information on the training data. Some defences exist against poisoning attacks and backdoor attacks, such as data sanitization and anomaly detectors.			4.2.3.4.1	

4.2.3.4	2	How to mitigate adversarial example attacks	Some defences exist against adversarial examples attacks. Model outputs smoothing, which reduces the model output sensitivity regarding its input. The training process and input data can be modified by continuously adding new adversarial samples, which requires a lot of data and could deceive network. Random rescaling on inputs can be introduced, or foveation mechanism can be used. The network can be modified in several ways, such as by applying input gradient regularization, by using non-linear activation functions, or by using dense associative memory models. An additional network which is separately trained can be used.			4.2.3.4.2	
4.2.3.4	3	Mitigate sensitive information leakage	Multiple defences can be enforced against sensitive information leakage: distributed learning frameworks, traditional cryptographic primitives-based approaches such as differential privacy or homomorphic encryption, and trusted platform-based approaches. Actual defence implementations depend on the type of models and the approaches.			4.2.3.4.3	
3.2.2.1	1	Keep track of the changes	The accesses and identifies given to users can change through time. It is important to keep an inventory of the current data, but also to have a policy to handle creations, edits and removal of accesses and identities.			3.2.2.1.1	
1.1.2.1	2	Symmetric algorithms	Some of the most used symmetric algorithms: DES, the first standard, 3DES, which uses keys that are three times larger, AES, which is the DES replacement recommended by NIST and supporting different key lengths, Blowfish, that supports different key lengths, does not have any licence and is the fastest of them.			1.1.2.1.2	
1.1.2.1	3	Asymmetric algorithms	Some of the most used asymmetric algorithms: RSA, supports variable lengths of key and block, Diffie-Hellmann, the first public key algorithm that exchanges keys under insecure channels, DSA, developed by NIST and for authentication and signature integrity verification, ECC, applies the elliptic curve theory that can be used to enhance other algorithms, designed to improve performances, power and battery consumption.			1.1.2.1.3	
1.1.2.1	4	How to compare them?	The most useful attribute of compare them are the block sizes (larger block sizes for symmetric algorithms give faster computation times), the key sizes (larger key sizes need more battery consumption and require more processing), and the algorithm speed (with Blowfish often being the fastest, depending on the used parameters).			1.1.2.1.4	

1.2.3.1	3	Some mitigations	A mitigation against spoofing is to detect liveness during the tests. Data leakages are sensitive because of biometric traits being irrevocable. A mitigation against template database leakages is to enforce template security by applying a trade-off between non-invertibility, discriminability and revocability. To this end, two generic approaches are applied: biometric feature transformation and biometric cryptosystems. Generating a secure sketch of traits can be realized by using fuzzy commitment and fuzzy vault.			1.2.3.1.3	
1.3.5.3	2	t-closeness variations	t-closeness can use various techniques: generalization, multi set-based generalization, one-attribute-per-column slicing, slicing, or slicing with suppression. Those techniques give different results depending on the considered parameters, with variations on revealed correlation on quantity, the information loss, the data type, the level of privacy preservation, or membership disclosures.			1.3.5.3.2	
3.2.1.4	3	Some physical issues	Some physical security-related issues can be caused by backups, that should be done directly by tenants and also by using offline storage, by the server location, avoidable by choosing adapted rooms, backup power and controlling entrances, or by firewalls, avoidable by activating a default deny mode, defining additional per-instance filters, and by enabling DDoS protections.			3.2.1.4.3	
4.1.1.6	2	Some related threats	Threats can appear if no anonymization is enforced, such as data breaches, internal misuses by employees, unwanted secondary uses, changes in company practices, or government accesses. Anonymization is a solution, but it must be effective. However, it can remove the purpose of big data analysis.			4.1.1.6.2	
4.1.1.6	3	Define adapted privacy models	Privacy models must comply with volume, variety and velocity, and satisfy the composability, computational cost, and linkability principles.			4.1.1.6.3	
5.1.3.1	2	Concerning threats	Two major concerns have been listed. First, device data can be linked to other data sources with other personal details, and potentially with other devices. Secondly, every connection with a backend server discloses the device IP address, which can be used as a proxy for location tracking.			5.1.3.1.2	
5.1.3.1	3	How to reduce the quantity of shared data	Three mitigations have been found: use an alternate OS for Android devices, disable Internet access by default for all applications, and manually disable problematic applications. Alternatives must then be installed via alternative stores, but they can then not have any access to the Google Play Services.			5.1.3.1.3	

1.1.1.1	2	Regardless of the medium	Data is stored on numerous devices such as servers, personal devices, or external storage supports. They should be protected in any context.			1.1.1.1.2	
2.1.3.3	1	What is a penalty?	A penalty is trade-off that can discourage users from making choices that protect their privacy. Such penalties can concern the service performances, utility, or usability.			2.1.3.3.1	
5.1.1.1	2	Private browsing limited	Organization parties should be aware of the limited impact of the private browsing mode when handling processes, data or tasks of the organization.			5.1.1.1.2	

Appendix C: Software Tests

The results of the conducted software tests are summarized in this appendix.

ID	Action	Expected result	Result
1.1	Run script with valid arguments	The arguments are passed to the script and used by it	OK
1.2	Run script with invalid arguments	The script exits after showing the arguments	OK
1.3	Run script without arguments	The script exits after showing the arguments	OK
1.4	Run script with the `help` argument	The script exits after showing the arguments	OK
1.5	Run script with as invalid *ODS* file	The script exits after showing the reason	OK
1.6	The *ODS* file has an invalid attribute	The attribute is not processed	OK
1.7	The *ODS* file has an invalid value	The script shows the error	OK
1.8	The hash is calculated based on the guide content	The hash is valid	OK

Data conversion tests

ID	Action	Expected result	Result
2.1	Click on the evaluation button.	The `ExplanationView` page opens.	OK
2.2	Click on the restoration button.	The `ResorationView` page opens.	OK
2.3	Click on the report button.	The report PDF file opens in another tab.	OK
2.4	Click on the PWA browser titles.	Additional installation information is shown.	OK
2.5	Click on the PWA links.	Links are opened in another browser tab.	OK

Application tests on the `HomeView` user interface

ID	Action	Expected result	Result
2.6	Click on the upload form.	A system modal window opens to select a JSON file.	OK
2.7	Click on the upload submit button.	The integrity of the JSON file is verified, the progress or results are saved into the store and the assessor is redirected to either the `EvaluationView` page or the `ResultsView` page.	OK

Application tests on the `RestoreView` user interface

ID	Action	Expected result	Result
2.8	Click on the text to change guide content.	Additional information and an upload form are shown.	OK
2.9	Click on the upload form.	A system modal window opens to select a JSON file.	OK
2.10	Click on the upload submit button.	The integrity of the JSON file is verified, the guide content is saved into the store, and the assessor is redirected to the `EvaluationView` page.	OK
2.11	Click on an item of the guide example.	If at least one description exists for the item, the item expends to give additional information, and collapses back on second click.	OK
2.12	Click on an item checkbox of the guide example.	The store state is not updated.	OK
2.13	Click on the evaluation button.	The `EvaluationView` page opens.	OK

Application tests on the `ExplanationView` user interface

ID	Action	Expected result	Result
2.14	Click on a subcategory name in the `Sidebar`.	Objectives, items and descriptions of the subcategory are displayed.	OK
2.15	Click on the close option in the expended `Sidebar`.	The `Sidebar` is hidden.	OK
2.16	Click on the expend icon to collapse the `Sidebar`.	The `Sidebar` is expended.	OK
2.17	Click on a collapsed category in the `Sidebar`.	The group containing children of the category is expended.	OK
2.18	Click on an expended category in the `Sidebar`.	The group containing children of the category is collapsed.	OK
2.19	Click on the results button in the `Sidebar`.	If the evaluation is complete, the `ResultsView` page opens.	OK
2.20	Click on the saving button in the `Sidebar`.	A JSON file is downloaded containing the evaluation progress and the guide content.	OK
2.21	Click on an item.	If at least one description exists for the item, the item expends to give additional information, and collapses back on second click.	OK
2.22	Click on an item checkbox.	The store state is updated, and the checkbox is updated with the corresponding state value.	OK
2.23	Click on an item checkbox.	The evaluation values change accordingly (unchecked to compliant, compliant to non-compliant, non-compliant to not concerned, not concerned to compliant).	OK
2.24	Click on a description link.	Links are opened in another browser tab.	OK
2.25	The last item of a subcategory is evaluated.	The corresponding subcategory is shown as completed in the `Sidebar`.	OK
2.26	The last item of a category is evaluated.	The corresponding category is shown as completed in the `Sidebar`.	OK
2.27	The last item of the guide is evaluated.	The result button is activated.	OK

Application tests on the `EvaluationView` user interface

ID	Action	Expected result	Result
2.28	If displayed, click on the non-compliant button.	The page scrolls to the non-compliant items.	OK
2.29	Click on the download button.	A JSON file is downloaded containing the evaluation progress and the guide content.	OK
2.30	Click on the evaluation button.	The `EvaluationView` page opens.	OK

Application tests on the `ResultsView` user interface

ID	Action	Expected result	Result
2.31	Click on the GASP logo.	The `HomeView` page opens.	OK
2.32	The assessor changes the browser size.	The user interface is responsive.	OK
2.33	The assessor installs the PWA.	The PWA is installed.	OK

Application tests on the general user interface

ID	State	Expected result	Result
2.34	The application is accessed.	The store is initialized and filled with data.	OK
2.35	The application is accessed.	The values of the items are undefined.	OK
2.36	A JSON file is processed.	The data are verified on its integrity.	OK
2.37	A JSON file contains an error.	A comprehensive and detailed error dialogue is shown.	OK
2.38	A restoration is done for a custom or an older version of the guide content.	The custom or older version of the guide content is loaded into the store and an information message is shown.	OK
2.39	The `EvaluationView` is opened.	A default subcategory is displayed.	OK
2.40	The `EvaluationView` is opened.	The `Sidebar` is displayed.	OK
2.41	The evaluation is not complete.	The `ResultsView` is not accessible and a redirection to `EvaluationView` is made.	OK
2.42	The results are shown.	The colour of the progress change based on their value (less than 50% in red, less than 85% in orange, less than 100% in green, 100% in dark green).	OK
2.43	The overall score is not of 100%.	The non-compliant items are shown.	OK
2.44	A mobile phone is used.	The user interface is responsive.	OK

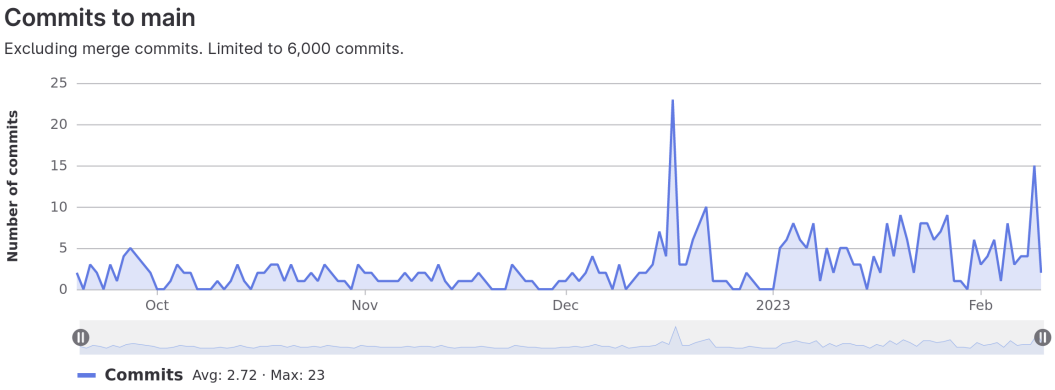
Application tests on its behaviour

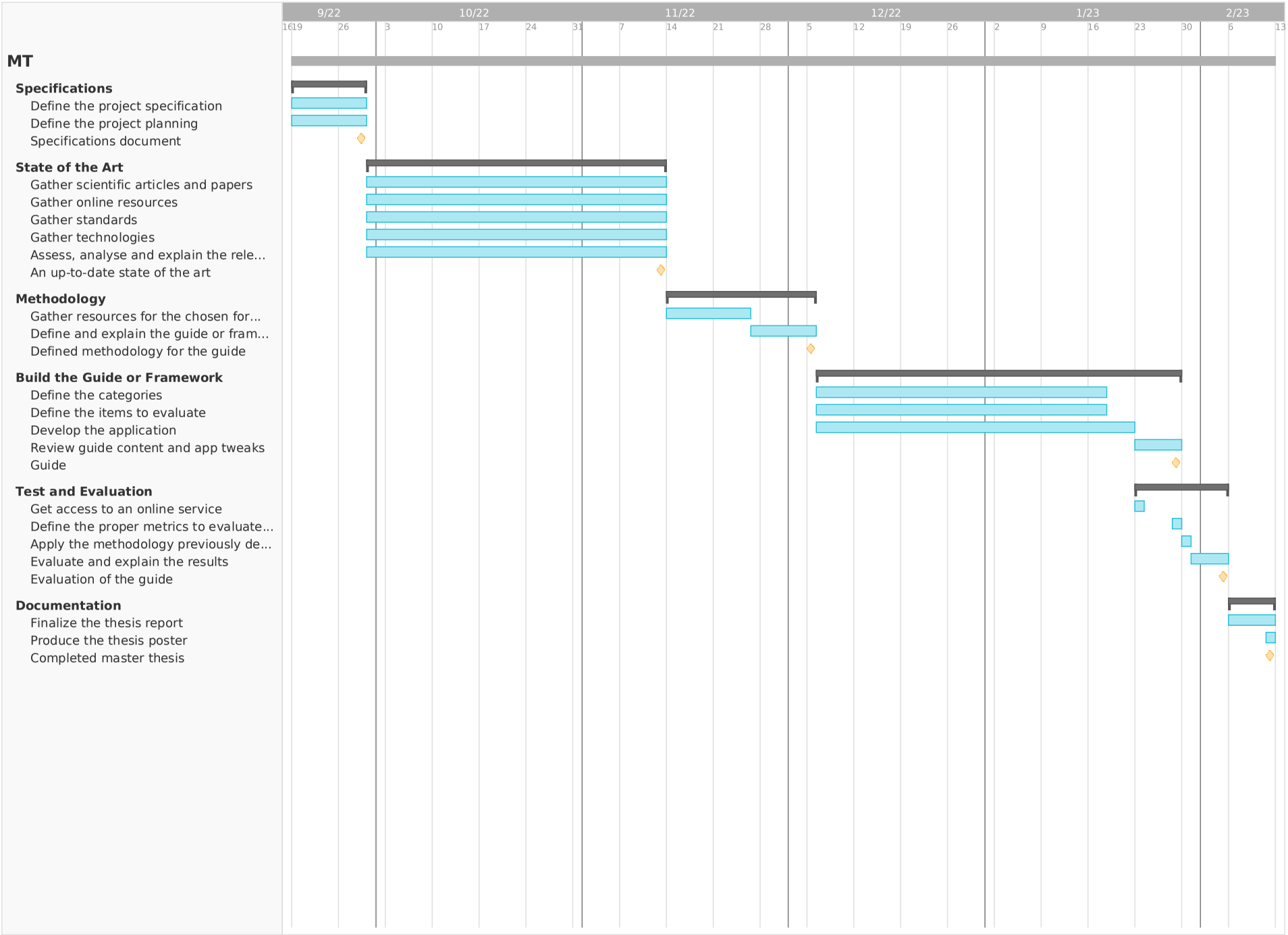
ID	Object	Expected result	Result
2.45	Guide content	Integrity checks are realized before storing the guide content.	OK
2.46	Evaluation status	When complete, the store changes the status of the evaluation.	OK
2.47	Category status	When complete, the store changes the status of the corresponding category.	OK
2.48	Subcategory status	When complete, the store changes the status of the corresponding subcategory.	OK
2.49	Score computation	The score computation is done accordingly to the defined formula from the proposal.	OK
2.50	Object getters	The store getters can return lists of all objects or of filtered objects giving unique identifiers.	OK
2.51	Object types and classes.	The store getters and its related data are using the appropriate TypeScript types and classes.	OK

Application tests on the store

Appendix D: Project Management

- The next image shows some statistics of the activities done on the thesis repository [1].
- The next page is the Gantt planning representing our thesis progress. Its originally planned version can be consulted at Appendix A.
- The minutes of the meetings done during the thesis can be consulted on the thesis repository [1].





Appendix E: Software and Tools

This appendix does not list the software dependencies or tools if they have already been presented before in this report.

Office Tools

All the software and tools used for the administrative or documentation tasks are listed here.

LaTeX Suite - pdfTeX, used to generate the documents for this project.

- Version 3.141592653-2.6-1.40.22 (TeX Live 2021)
- Licence pdfTeX copyright, Lesser GNU General Public

Zotero, used to manage the dissertation references.

- Version 6.0.13
- Licence AGPL-3.0

Draw.io, used to draw the different graphs.

- Version 19.0.0
- Licence APACHE Licence, version 2.0

Microsoft Teams, used to conduct organizational and follow-up sessions.

- Version 1.5.00.10453 (64-bit) (64 bits)
- Licence Proprietary

Development Tools

All the software and tools used for the development tasks are listed here.

LibreWolf, used for research, implementation and testing.

- Version from 104.0.2-1
- Licence MPL-2.0 (d), GNU General Public Licence (GPL) et GNU Lesser General Public Licence (LGPL)

ungoogled-chromium, used for research, implementation and testing.

- Version 105.0.5195.125
- Licence BSD-3-Clause

GitLab, used to manage the versioning of project resources.

- Version GitLab Enterprise Edition 15.3.2-ee
- Licence MIT Licence

VSCodium, used to implement *JavaScript* software and to redact the LaTeXdocuments.

- Version 1.71.2
- Licence MIT Licence

PyCharm Professional, used to implement the *Python* script.

- Version 2022.2.4
- Licence Copyright © 2010-2022 JetBrains s.r.o.

LibreOffice Calc, used to create the spreadsheet file.

- Version 7.4.3.2
- Licence MPL-2.0