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

objectiv	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-lex	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://solidpr	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	