3 Design

The checklist is dynamic, not exhaustive, and will be updated regularly. If you have any suggestions or comments, we would like to hear from you.

Data oriented design requirements

Minimise and limit. The amount of personal data collected and processed should be limited to what is lawful and what is strictly necessary. The data shall be deleted when storage is no longer required for the purposes. Follow the principle "*Select before you collect*".

Examples:

- Review the Data Protection Impact Assessment (DPIA).
- Be sure that the need for personal data matches the amount and scope of data collected. Do not collect more information than necessary. Limit the amount of information processed on units or in areas with lower trust and security assurance.
- Avoid, limit, or minimise the need to collect and process sensitive personal data.
- Limit and minimise the exposure of unnecessary functionality and personal data in the user interface. Consider, for example, whether it is necessary to store directly identifying information in the software itself, or if pseudonymised information is sufficient.

Hide and protect. Personal and their interrelationships, should not be communicated, processed, or stored in plain view. By hiding directly identifiable personal data from plain view, the risk of abuse and the scope of potential incidents is significantly reduced. Examples include pseudonymisation, encryption, and aggregation of personal data.

Examples:

- Conduct *threat modelling* and "*attack surface analysis*" when designing software.
- Use encryption mechanisms for secure transfer, communication, processing and storage. This is particularly important if there is a need to transfer personal data over uncontrolled areas and networks.
- Anonymise or pseudonymise personal data wherever possible.
- Avoid unnecessary exposure to communication patterns and connections (such as APIs, feeds, gateways, login interfaces, etc.).

Separate. Personal data should be separated from other data when possible. Data should be stored in separate databases, entities, and areas for each purpose and process. By separating the processing or storage of several sources of personal data that belong to the same person, the possibility of creating complete profiles of one person is reduced. Separation is also an effective means to achieve purpose limitation and to avoid linkability between different data sets. Tables containing personal data should have shorter storage times and a deadline for automatic deletion, while tables without personal data can be stored for longer. Measures to achieve this can include access control to tables, splitting database tables, distinguishing between components, units and areas with a high level of trust and those with a lower level of trust, and dividing access to specific areas according to necessity.

Examples:

- Separate sensitive personal data from less sensitive personal data (in the database, access to sites, for clients and units, etc).
- Separate tables in the database and associated access rights to tables and areas, according to job necessity (the principle of least privilege).
- Keep less highly trusted units and areas separate from more highly trusted units.
- Rows in tables should be hard to link to one another.

Aggregate. Personal data should be gathered and processed in as aggregated manner as possible to ensure the enforceability of the data subject's rights, without prejudice to the business value and purpose of the collection and use.

Examples:

- Reduce the use of detailed and sensitive personal data.
- Remove unnecessary and excessive information. For example,
    - "Raise" the measures of time by using weeks instead of days or hours
    - use counties or regions, rather than street addresses, when allocating people or units
    - use groupings rather than individuals
- Use anonymising techniques whenever possible.

Data protection by default. All settings should, by default, be configured in the most privacy-friendly setting. The user should have to make a conscious choice to change any settings that would result in a less privacy-friendly configuration. For example, it should be the user's choice to share more data with others.

Examples:

- All privacy-friendly configurations need to be on by default.
- Device tracking should be disabled by default.
- Bluetooth should be disabled by default.
- Tracking from one website to another should be disabled by default.
- By default there should be no reuse of information about which websites the data subject has visited.

Process oriented design requirements

Inform. The software should be designed and configured so that the data subject is sufficiently informed, on how the software works and how personal data is processed. When profiling or automated processing of personal data are being conducted the data subject should be informed on how it is being done. It is important to remember that special requirements apply if the software is aimed at children.

Examples:
A website containing information about the software should be established and made available to the data subjects before they begin using the software:

- Provide a point of contact, contact form, or similar, that the data subject can use to request information on
    - the purpose and need for data collection

    - the security of the software (how the data is protected)
    - use of subcontractors, and possible sharing of data with subcontractors
- Use multiple approaches and channels to ensure that you reach all users and user groups.
- Use simple, comprehensible language.
- Use multiple languages if necessary.
- Enrich and vary with photographs, icons, audio, video, etc to target users.

Control. The data subject has the right to control their own personal data. This includes requesting access to view, update, and/or delete their own data. Where automated processing is taking place, or decisions are being made without human intervention, the data subject can demand manual processing. The software should be designed so that the data subject can exercise these rights as easily as possible.

Examples:
Establish and enable the ability to:

- maintain an overview of which elements of data processing are necessary to fulfil the contract, and which are subject to voluntary consent
- allow the data subject to provide consent on an information page with a checkbox before using the software
- withdraw consent via a menu within the software. Keep in mind that collection of personal data must cease if consent is withdrawn
- give access for the rectification, blocking, or deletion of personal data, e.g. by allowing collected data to be viewed directly by the data subject within the software
- ensure the permanent deletion of personal data in the database and wherever else it is stored (e.g., backup). The information can also be exported to a file or paper version for manual review with a corresponding procedure allowing the data subject to correct, block, or delete data (within 30 days)
- terminate a contract/agreement, install, uninstall, enable and disable an application, service, technical component, or system, by using functionality within a menu, on a dedicated page, or manually using a form
- submit questions or complaints relating to data protection and security. Alternatively, the information can be made available on a website with contact information connected to a staffed communications channel for handling enquiries (in this case, a documented procedure must be established)
- object to profiling by enabling users to make a conscious choice to reject profiling and the redistribution of personal data. This can, for example, be done via a menu with a checkbox and a flag to be stored in the database, or carried out manually via a staffed communications channel
- define requirements for openness and information on how automated decisions are made, as well as the ability for data subjects to demand manual processing. This is described in the system documentation, and should be made available on an information page within the software.

Enforce. The software should be designed so that it can document that how it ensures the enforceability of the data subject's rights. The documentation should cover accountability and how the data protection regulation is enforced. It should be available for audits and

inspections of the processing. This also includes artificial intelligence, profiling, and automated processing.

Examples:

- The software must apply the highest privacy settings by default:
    - Settings should be presented in a menu where the data subject must make a conscious choice to actively "change" to less privacy-friendly settings.
    - If the software at a later date has to be changed to a less privacy-friendly setting, the data subject must be clearly informed of the change, and may consent to it by clicking a checkbox after notification of the change has been received/read.
- The software must meet the dataportability requirement:
    - There must be a way for the data subject to request the disclosure of their own personal data, or request that the data be transferred to another service provider, in a standardised and reusable format.
    - This right applies when processing is based on consent or a contract.
    - The data subject may request that personal data be exported and delivered in a secure manner (manually by traditional post, or another secure delivery method) to new service providers.
- Consent must require the active participation of users:
    - Consent, or change of consent, is signalled by filling out a text box or clicking a checkbox that marks a flag in the database.
    - Software aimed at young people and minors must contain features requiring consent from the parent or guardian before access is granted. This feature must ensure or confirm that they are authorised to grant this consent. Alternatively, procedures can be established to request manually documented consent from the parent or guardian.
- The data subject should be given an overview of what access (including access and changes by the data subject) has taken place, when, by whom, and what types of consent has been given. All access should be recorded in logs that can be made available to the data subject.

Demonstrate. The controller must be able to document compliance with the data protection regulation and security of processing. The software must be designed and developed so that the controller can document and demonstrate how the requirements of the data protection regulation have been implemented. Examples include documentation demonstrating that the software has been developed using a methodology that ensures data protection by design and information security (SSDLC - Secure Software Development Life Cycle), reports from security audits, vulnerability scanning, security tests such as penetration testing, and reports on practicing data breach management.

Analyse and reduce the attack surface of the software being developed

- Analyse the attack surface of the pre-designed software to reduce opportunities to exploit weaknesses and vulnerabilities in the software.
- Review designs and analyse where it is possible to receive input data, get output and where data is transported and stored.

- Check if the same type of information is being collected in multiple places (duplicate functionality), and assess whether functionality can be simplified.
- Reduce the likelihood of errors by simplifying the software and eliminating unnecessary functionality.
- Reuse the assessments of security risks and data protection impacts that were completed during the requirements phase.
- Implement vulnerability-reducing measures to achieve acceptable tolerance levels for data protection and security, if the analysis reveals that the existing levels are not satisfactory.
- Reuse the tolerance level developed during the requirements phase.
- Be sure to document the analysis and the reduction of the attack surface.

Threat modelling

- Analyse components, access points, data flow and processing of data in the software.
- Ensure that those involved in the development team analyse how the software could be misused in different scenarios.
- Review each scenario to see how the design can be improved to avoid any threats identified. This can be done by implementing vulnerability-reducing measures, resulting in stronger and more robust software.
- Perform a risk assessment of any vulnerabilities that remain and which must be mitigated using other measures. Ensure that these vulnerabilities are included in a risk log.

Examples of tools

- Design principles for good security, including:
  - The principle of least privilege
  - Defence in depth
  - Fail Securely
- Threat Modelling
- Attack-Surface-Analysis
- Use Case and Misuse-Case Modelling
- Attack-Trees
- DREAD/STRIDE
- The Data Protection Authority's guidelines on anonymisation and anonymisation techniques (https://www.datatilsynet.no/aktuelt/2015/Hvordan-anonymisere-personopplysninger-Ny-veileder/)

Why are design requirements necessary?

- Personal data must be processed lawfully and the design of the software must reflect this, cf. Article 6 of the General Data Protection Regulation. It is therefore important to be aware of the type of personal data being processed, the purpose and terms of the processing, and the compilation of personal data.
- The rights of the data subject must be reflected in the design, cf. the General Data Protection Regulation articles 12-23.