Annex 4 – IT Data Protection Checklist

Checklist to be integrated into the specifications of any IT development

Privacy Design Strategies	Elements to be implemented for any IT development, in particular in the event of resorting to a third party
Data-oriented strategies	
Minimise	▪ **Identify the data strictly necessary for the purpose of the processing operation(s) relating to the solution developed and in particular:** □ Collection of the minimum required for the processing of sensitive data □ Minimisation regarding the collection of logging data □ Non-storage of sensitive or critical logging data ▪ **Association of a retention period to all processed data** ▪ **Implementation of a system for archiving, purging or anonymising data at the end of retention periods** ▪ **Implementation of an automatic deletion procedure**
Abstract	▪ **Processing and storage of data so as to reduce their identifying nature by mechanisms of pseudonymisation, generalisation, addition of noise, etc.**
Separate	▪ **Separation of environments for development and testing, integration and production** □ Restricting access to the production environment for development teams □ Use of fictitious or anonymised data in development and testing environments □ When using production data in the test environment, provide the same level of protection as for the production environment
Hide	▪ **Management of developer and end-user authorizations prior to development** □ Access authenticated with a unique and nominative ID prior to any access to personal data □ Implementation of a mechanism for managing user authorizations via roles (Role based access) □ Implementation of a specific policy for administrator passwords □ Privilege accounts cannot be used for day-to-day operations □ Tracing activities via a logging system □ Documentation and automation, if possible, of the management of employee movements (departure, end of contract, unsubscription to the solution, change of position, etc.) that can be based on an LDAP □ Use of a password management tool during the project □ Prohibition of generic accounts shared between several people ▪ **Securing communications:** □ Implementation of the TLS protocol for websites and mobile applications □ Limitation of communication ports to what is strictly necessary for the proper functioning of the installed applications □ Passwords with a sufficient level of complexity and always stored as hash (bcrypt) □ Encouraging strong authentication ▪ **Implementation of encrypted and regularly verified backups** ▪ **Database management:**

	□ Use of nominative accounts for access to the databases and creation of specific accounts for each application □ Implementation of measures against attacks by injection of SQL code, scripts... □ Enabling encryption of data on storage disks and in databases
Process oriented strategies	
Inform	▪ **Identification of data collection points in case of collection of data directly with the data subject** □ Inclusion of an information notice containing at least the contact details of the data controller, the purposes of the processing and the description of the rights and how to exercise them, as well as a link to more complete information for each collection point. □ Implementation of a means of direct transmission of information in case of direct contact with data subjects ▪ **Identification of categories of data collected indirectly** □ Identification of the means of transmission of information to data subjects ▪ **Establishment or updating of complete information relating to the service developed** □ Easy to access, clear and intelligible □ Distinct from other provisions not specific to data protection (Terms and conditions, contractual clauses...) ▪ **Provide the means of informing individuals in the event of data breaches entailing a high risk for the rights and freedoms of data subjects**
Control	▪ **Implementation of a default settings strategy taking into account the principles of data protection regarding IT developments for third parties (Privacy by Default)** □ Password complexity □ Collection of end-user consent for the processing of optional data (*e.g. geolocation*) and prohibition of further use of such data for other purposes ▪ **Implementation of mechanisms allowing for the exercise of the rights of data subjects (of access, to object, to rectification, to erasure, to data portability,) in particular with regard to the data archived in the solution developed** □ Establishment of a mechanism for the formal identification of the person wishing to exercise his or her rights □ Use of a secure channel for the communication of the data or of a secure space for making it available to the data subject wishing to exercise his or her right of access or to data portability □ Logging of the operations regarding the processing of the request ▪ **Collecting consent and implementing withdrawal of consent as simply as it was given** □ Regarding the processing that is the object of the IT development when it is based on the data's subject's consent □ Regarding the deposit of cookies and other tracers, the deposit of which on users' terminals requires the collection of their consent
Enforce	▪ **Inclusion of security and data protection clauses in the contract with the processor in charge of development** □ Identification of the geographical location of the servers hosting the data ▪ **Compliance of the proposed architecture with internal security policies** ▪ **Training of the teams in charge of development in the good practices regarding secure development** □ Provision of guidelines for good coding practices, development procedures adapted to new threats and vulnerabilities □ Demonstration by the processor that these measures are implemented as part of its development process ▪ **List of all the assets involved in the development of the solution and documentation of the operations related to the developed solution**

	◻ Assessment of the risks to which each tool is exposed ◻ Securing servers and workstations in a homogeneous and reproducible way (Ansible, Puppet or Chef tools can be used) ▪ **Follow-up of a secure development methodology in accordance with the state of the art and known practices in the field** ◻ Avoiding the top 10 vulnerabilities identified by the OWASP ▪ **Guarantees of integrity and source code protection during build and run phases** ▪ **Contractual guarantees that the source code will be maintained and updated throughout the life of the project** ◻ Backup and archiving of validated versions ◻ User role-based access to the Source Code Manager ◻ Implementation of metrics tools to check the quality of the code ◻ Library Access Logging ◻ Source code purge ▪ **Minimisation of the attack surface when using libraries or development kits** ◻ Evaluation of the value of each dependency ◻ Choice of cryptographic libraries or open source software maintained, recognized, easy to use and benefiting from regular updates ◻ Use of dependency management systems (yum, apt, mavent, pip...) in order to maintain an up-to-date list of dependencies ▪ **Regular patching of solution components so that known vulnerabilities cannot be exploited by malicious people** ▪ **Use of vulnerability detection tools to identify possible security breaches** ◻ Test to be carried out regularly and before any production of a new software version
Demonstrate	◻ Contractual definition of additional technical and organisational measures when the processor does not present sufficient guarantees in terms of security or data protection ◻ Implementation of recurring audits according to the criticality of the solution, including on-site audits when the solution is developed by third parties. Formal acceptance, implementation, and efficiency of safety controls prior to the production phase ◻ Implementation of KPIs to identify the degree of compliance of the developed solution and the correct implementation of the processes. ◻ Provision of deliverables and internal procedures regarding security and data protection to demonstrate the processes implementation