# Security, Compliance and Reproducibility

*Data protection, security, reproducible practice, and the capstone project.*

This guide accompanies Module 6, the final module, which covers the safeguards that make automation responsible and the reproducible practices that make it lasting, before introducing the capstone project. It addresses data protection under the General Data Protection Regulation, practical security and safe data handling, and reproducibility and documentation, drawing on legal-scholarly analysis, privacy-engineering research, and the reproducibility literature.

A caveat applies throughout: this material provides the working understanding an engineer needs, not legal advice. Health data is a special category under EU law precisely because its misuse can seriously harm the people it describes, and analyses of the GDPR show that it functions as both an enabler and a barrier to the secondary use of health data, protecting individuals while imposing real obligations on those who process their data [1]. Knowing the limits of one's own expertise, and when to consult a specialist, is part of doing this work well.

### Learning outcomes

After working through this guide you will be able to:

- Explain why health data is specially protected and how core data-protection principles shape design.
- Apply practical security measures and automate routine security checks.
- Use pseudonymisation and de-identification appropriately and understand their limits.
- Make automated work reproducible and document it for others.
- Scope and complete the capstone project to a professional standard.
- Relate the module's practices to the published reproducibility and privacy literature.

## 1. Data protection and the GDPR

### 1.1 Why health data is special

Under the GDPR, health data is a special category of personal data given heightened protection, because its misuse can cause lasting harm. Unlike a password, which can be reset, a person's medical history cannot be un-disclosed, and this permanence justifies a higher standard of care at every step.

### 1.2 Principles that shape design

Several principles shape engineering decisions directly: processing must have a lawful purpose; data collection and retention should be minimised to what is necessary; data must be secured against loss and unauthorised access; and controllers must be accountable, able to demonstrate how data was handled. Legal analysis highlights that the regulation's principle of purpose limitation constrains further processing beyond the purpose for which data were collected, and that the boundary between permissible secondary

use and new processing is often ambiguous, exposing researchers to compliance risk and making specialist advice necessary [2].

### 1.3 Minimisation in practice

Data minimisation is the principle an engineer controls most directly, and it is doubly beneficial: reading and keeping only what a task needs reduces both the risk if data is exposed and the burden of protecting it. In practice it means reading only the fields a task requires, not copying data speculatively, preferring identifiers to names, and deleting working copies when done. Empirical study of the regulation finds it acts as an enabler for user rights, transparency, and the sharing of anonymised statistics, but as a barrier when identifiable, individual-level data are involved, owing to the time, workload, and stricter interpretations required [1].

### 1.4 Pseudonymisation

Pseudonymisation replaces identifying details with a code, so that data cannot be tied to a person without separate, protected information; for example, results are stored against a patient code while the link between code and name is held separately and access-controlled. It lets you work with realistic data while greatly reducing the harm if the data is exposed, though it is not a complete shield, since re-identification can sometimes be possible.

### 1.5 Designing for individuals' rights

Data protection gives people rights over their own data: to know what is held, to have errors corrected, and, in certain cases, to have data deleted. Systems should be designed so these requests can be honoured. A good test is that if your automation cannot find or remove one person's data when asked, it is not ready for real personal data; building for this from the start is far easier than retrofitting it.

> **Key idea.** The safest data is the data you never collected. Minimise by default, secure what you keep, design for individuals' rights, and seek specialist advice where the lawful basis for processing is unclear.

## 2. Security and safe data handling

### 2.1 The foundations

Practical security rests on a small number of foundations: encrypting data at rest and in transit, controlling access so that only those who need data can reach it, and logging who accessed what and when. Most real-world exposures stem from missing basics rather than sophisticated attacks, so getting these right prevents the majority of incidents.

### 2.2 Least privilege

The principle of least privilege, granting each person or program only the access it needs, limits the damage when something goes wrong. A reporting script that only reads data should never have permission to delete it, because that permission adds risk with no benefit. Applying least privilege to people and programs alike is one of the highest-value habits in security.

### 2.3 Handling secrets

Secrets, meaning passwords, keys, and tokens, require particular care: they must never be written into code or committed to version control, since a secret placed in a repository's history is difficult to remove and should be treated as compromised. Keep secrets in a dedicated, protected place, and replace any secret that is exposed.

### 2.4 De-identification and its limits

A recurring hazard is the use of real patient data in less-guarded settings such as testing, which is why this course uses synthetic data throughout. Where realistic data is genuinely required, de-identification and pseudonymisation reduce harm by removing or replacing identifiers. A systematic review of de-identification and anonymisation strategies documents heuristic, lexical, pattern-based, and statistical-learning approaches, and cautions that true anonymisation is challenging and that residual identifiers often remain [3]. Modern automated tools combine deep learning with rule-based methods to detect and replace personal information at scale with high recall and precision [4], and standards-based approaches now apply configurable de-identification directly to FHIR resources for the secondary use of healthcare data [5].

### 2.5 Automating security

Much of routine security can itself be automated, scanning code for secrets before it is shared, flagging dependencies with known vulnerabilities, and verifying that access rules stay as intended, all as part of a continuous-integration pipeline. Security you must remember is security you will eventually forget; automating the routine checks so they run every time turns good intentions into reliable protection.

> **Practical rule.** Encrypt at rest and in transit; apply least privilege; keep secrets out of code and version control; use only synthetic or de-identified data in testing; and automate security checks so they run every time.

## 3. Reproducibility and documentation

### 3.1 What reproducibility means

Reproducibility is the property that another person can run your work and obtain the same result, using the same data, code, and instructions. The phrase another person is important: reproducibility is not merely whether the work runs on your own machine, but whether it runs on someone else's, without you present.

### 3.2 The ingredients

The reproducibility literature converges on a small set of ingredients: literate programming, version control and code sharing, control of the compute environment, persistent data sharing, and documentation, described as the five pillars of reproducible computational research [6]. The widely cited ten simple rules for reproducible computational research stress recording how every result was produced, archiving exact versions of programs and data, and noting the provenance of results, and observe that good reproducibility habits ultimately save the researcher time [7]. In practice, four things must travel together: the code under version control, the data or precise instructions to obtain it, the compute environment with exact library versions, and the steps to run it.

### 3.3 The environment trap

The single most common reproducibility failure is the phrase it works on my machine. It arises because your machine has libraries and versions that others lack, and a different version can change behaviour or break the code. The fix is to record the exact libraries and versions your code needs in a file that ships with the code, so that any machine can recreate a matching environment. Best-practice recommendations for computational science make the same case for sharing code and data and controlling the software environment as the basis for reproducible and extensible research [8].

### 3.4 Documentation people use

Documentation completes the package. It should start with how to run the project in a few clear steps, say what it does and what it produces, note the inputs it needs, and stay short and current. The operational test is whether a new colleague could run your project from the README alone, without asking you; if not, the documentation needs more.

> **Key idea.** Package four things together, code, data, environment, and steps, and pin exact versions. The test of documentation is whether a newcomer can run your project from the README alone.

## 4. The capstone project

### 4.1 The task

The capstone asks you to design and build a small automation for a healthcare or research task of your choice, including a data step, at least one test, and a note on data protection, and to accompany it with a short report and a recorded presentation. It is an assembly of skills already developed across the course rather than anything new.

### 4.2 What good looks like

The marks of a strong submission are modest but demanding: it works end to end, it is tested, it is responsible in its use of synthetic data and consideration of data protection, and it is reproducible from your documentation. None of these asks for scale or cleverness; a working, tested, responsible, reproducible project demonstrates the whole discipline of the course.

### 4.3 Scope

Scope is where such projects succeed or fail. A finished, small project is worth far more than an ambitious unfinished one, so choose a task you could complete in a short time and then refine it. The commonest capstone mistake is scoping too big, so err toward small.

### 4.4 A path to done

A dependable order of work guarantees you always have something to submit: choose the task and synthetic data; build the simplest version that runs end to end; harden it with a test, validation, and a log; and finally document it. After the second step you always have a complete, if simple, project, and every later step only improves it.

### 4.5 The recorded presentation

The recorded presentation should state the problem and why it matters, show the working automation, explain how it was tested and how data was protected, and reflect honestly on one limitation and a next step, the same intellectual honesty that the evaluation and reproducibility literature of this course commends. Aim for five to seven minutes, and show the working automation rather than only slides.

> **Guidance.** Aim for small and complete over large and unfinished. Build end to end first, then harden and document. Quality, not scale, is what the capstone rewards.

## 5. Case studies and worked discussion

The safeguards of this module are grounded in legal analysis and privacy-engineering research, and several sources reward closer attention because they show both the constraints and the practical techniques for working within them.

### 5.1 The GDPR as enabler and barrier

Vuković and colleagues studied the secondary use of health data across Europe and found that the GDPR functions in two directions at once: it acts as an enabler when it comes to individuals' rights, transparency, and the sharing of anonymised statistics, but as a barrier when identifiable, individual-level data are involved, owing to the time, workload, and stricter interpretations required [1]. Becker and colleagues sharpen one source of this difficulty, analysing when reusing data counts as further processing under the regulation and showing that the boundary is often ambiguous, which exposes researchers to compliance risk and makes specialist advice necessary [2]. For an engineer, the practical takeaway is to design for anonymised or minimised data wherever possible, because that is where the regulation is most enabling, and to seek advice early when identifiable data are unavoidable.

### 5.2 De-identification techniques and their limits

Kushida and colleagues review strategies for de-identifying and anonymising electronic health record data for multicentre research, documenting heuristic, lexical, pattern-based, and statistical approaches, and cautioning that true anonymisation is difficult and that residual identifiers often remain [3]. Murugadoss and colleagues demonstrate how far automation has come, building a de-identification tool that combines deep learning with rule-based methods to detect and replace personal information at scale with high recall and precision [4]. Raso and colleagues bring the techniques into the FHIR era, applying configurable anonymisation and pseudonymisation directly to FHIR resources for the secondary use of healthcare data [5]. The combined lesson is that de-identification is powerful and increasingly automated, but never a guarantee, so it should be layered with minimisation, access control, and the use of synthetic data for testing.

### 5.3 Reproducibility as a shared standard

The reproducibility literature offers concrete, widely adopted checklists. Ziemann and colleagues distil five pillars, literate programming, version control, environment control, data sharing, and documentation [6]; Sandve and colleagues offer ten simple rules that emphasise recording how every result was produced and archiving exact versions [7]; and Stodden and colleagues set out best practices for the software infrastructure and environments that make research reproducible and extensible [8]. These sources agree

that reproducibility is less about any single technology than about disciplined habits, and that those habits ultimately save time as well as protecting integrity.

> **Key idea.** Design for minimised or anonymised data where the regulation is most enabling; treat de-identification as a strong but imperfect safeguard; and adopt the reproducibility community's checklists as shared, disciplined habits.

## 6. Further reading and practice

This final module ties the whole course together, and its practices are best consolidated by preparing the capstone itself. The exercises below are designed to take an afternoon and to move your project from an idea to a working plan. Throughout, use only synthetic or openly published data, and treat data protection as a design consideration from the outset rather than an afterthought.

For deeper reading, the GDPR analyses [1], [2] explain how the regulation both enables and constrains the use of health data, the de-identification references [3], [4], [5] survey the techniques and their limits, and the reproducibility literature [6], [7], [8] provides concrete, widely adopted checklists for making computational work repeatable. The regulation itself [9] is the authoritative source, though most engineers will meet it through institutional guidance rather than the primary text.

- Choose a capstone task and identify what personal or sensitive data it would touch, then redesign it to use synthetic data only.
- List the fields your task truly needs and remove any others, applying data minimisation.
- Pseudonymise a sample dataset by replacing identifiers with codes held separately.
- Write a short requirements file pinning the exact libraries and versions your project needs.
- Draft a README that a colleague could follow to run your project without asking you.

Completing these exercises will leave you with a scoped, privacy-conscious, reproducible starting point for the capstone, which is exactly what the final assignment builds upon.

## Key terms

**Special category data —** Under the GDPR, sensitive personal data, including health data, given heightened protection.

**Purpose limitation —** The principle that personal data should be used only for the purposes for which it was collected.

**Data minimisation —** Collecting and keeping only the personal data genuinely needed for a purpose.

**Least privilege —** Granting each person or program only the access it needs, and nothing more.

**Pseudonymisation —** Replacing identifying details with a code so data cannot be tied to a person without separate, protected information.

**De-identification —** Removing or obscuring identifiers so that data can no longer be readily attributed to an individual.

**Reproducibility —** The property that another person can run your work and obtain the same result from the same data, code, and steps.

**Compute environment —** The specific libraries and versions a program needs to run as intended, recorded so others can recreate it.

## Self-check questions

1. Explain why health data is a 'special category' and how the minimisation principle translates into concrete design choices.
2. What does purpose limitation constrain, and why can it create compliance risk for secondary use of data?
3. State the principle of least privilege and give an example of applying it to a program rather than a person.
4. Why must secrets never be committed to version control, and what should you do if one is exposed?
5. Describe two approaches to de-identification and explain why true anonymisation is difficult.
6. List the four ingredients that must travel together for work to be reproducible, and explain the role of pinning library versions.
7. What is the operational test of good documentation, and why does it matter?
8. Outline a plan for your capstone that guarantees you always have a complete project to submit, even if time runs short.

# References

*Peer-reviewed sources located via the Consensus academic search tool; standards cited from their issuing bodies. Formatted in IEEE style.*


**[1]** J. Vuković et al., “Enablers and barriers to the secondary use of health data in Europe: general data protection regulation perspective,” Archives of Public Health, vol. 80, 2022. [Online]. Available: https://consensus.app/papers/details/2e8c2446a9e85d66a244b3ca615c2bb6/

**[2]** R. Becker et al., “Secondary use of personal health data: when is it ‘further processing’ under the GDPR, and what are the implications for data controllers?,” SSRN Electronic Journal, 2022. [Online]. Available: https://consensus.app/papers/details/1bcd945ca29e58759bf67bc9ec6a1439/

**[3]** C. A. Kushida et al., “Strategies for de-identification and anonymization of electronic health record data for use in multicenter research studies,” Medical Care, vol. 50, 2012. [Online]. Available: https://consensus.app/papers/details/2e2177a546df5a7794e02cee788b7352/

**[4]** K. Murugadoss et al., “Building a best-in-class automated de-identification tool for electronic health records through ensemble learning,” Patterns, vol. 2, 2021. [Online]. Available: https://consensus.app/papers/details/c61078f983075ab988c5a5b5c908121e/

**[5]** E. Raso et al., “Anonymization and pseudonymization of FHIR resources for secondary use of healthcare data,” IEEE Access, 2024. [Online]. Available: https://consensus.app/papers/details/cf418172135357ceb25e237607b8c39b/

**[6]** M. Ziemann, P. Poulain, and A. Bora, “The five pillars of computational reproducibility: bioinformatics and beyond,” Briefings in Bioinformatics, 2023. [Online]. Available: https://consensus.app/papers/details/09aac82bfb2550a7a72f0fd387a4fb91/

**[7]** G. K. Sandve, A. Nekrutenko, J. Taylor, and E. Hovig, “Ten simple rules for reproducible computational research,” PLoS Computational Biology, vol. 9, no. 10, 2013. [Online]. Available: https://consensus.app/papers/details/844402aaa5105a5280cdc99952699243/

**[8]** V. Stodden and S. Miguez, “Best practices for computational science: software infrastructure and environments for reproducible and extensible research,” Journal of Open Research Software, vol. 2, 2014. [Online]. Available: https://consensus.app/papers/details/39572ffa44795b7ebddce589c7b62305/

**[9]** European Parliament and Council, “Regulation (EU) 2016/679 (General Data Protection Regulation),” Official Journal of the European Union, 2016. [Online]. Available: https://eur-lex.europa.eu/eli/reg/2016/679/oj