

RDM Career Landscape & Jobs

Can one make a living doing RDM?



RDM Career Paths

Explore the people who make research data reusable, trustworthy and sustainable



You will investigate real Data-* (Data Steward, Data Engineer, Data Analyst, Data Scientist, Data Curator ...) roles through evidence, job ads and source texts.

Which Data-* role would you hire for which research-data problem?

You will investigate

Real job ads and role descriptions. You decide what each role is meant to do.

You will compare

Problems, verbs, outputs, stakeholders, evidence and neighbouring roles.

You will present

A compact role profile and a career-path insight for your group.

Part 1

A short history of Data-* roles



Why did Data-* roles emerge?

Data deluge

Research became increasingly digital, large-scale and collaborative. Datasets needed active management beyond publication.

Reproducibility pressure

Software, notebooks, data and documentation became part of the evidence behind research claims.

Open Science & FAIR

Funders, journals and infrastructures started asking for reusable data, persistent identifiers and FAIR practices.

Institutional services

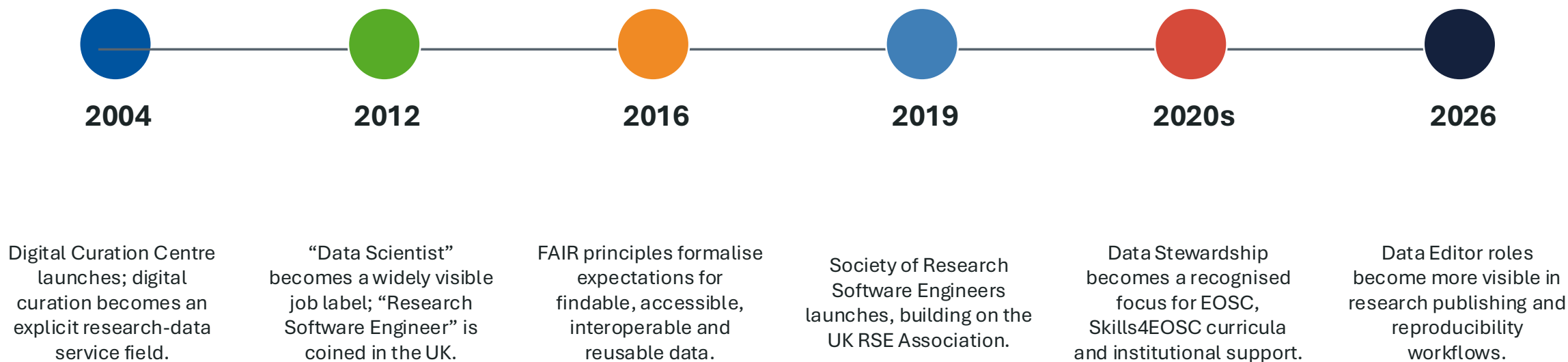
Libraries, IT centers and repositories developed new support roles.

Professionalisation

Communities now work on curricula, competencies, career tracks and recognition.

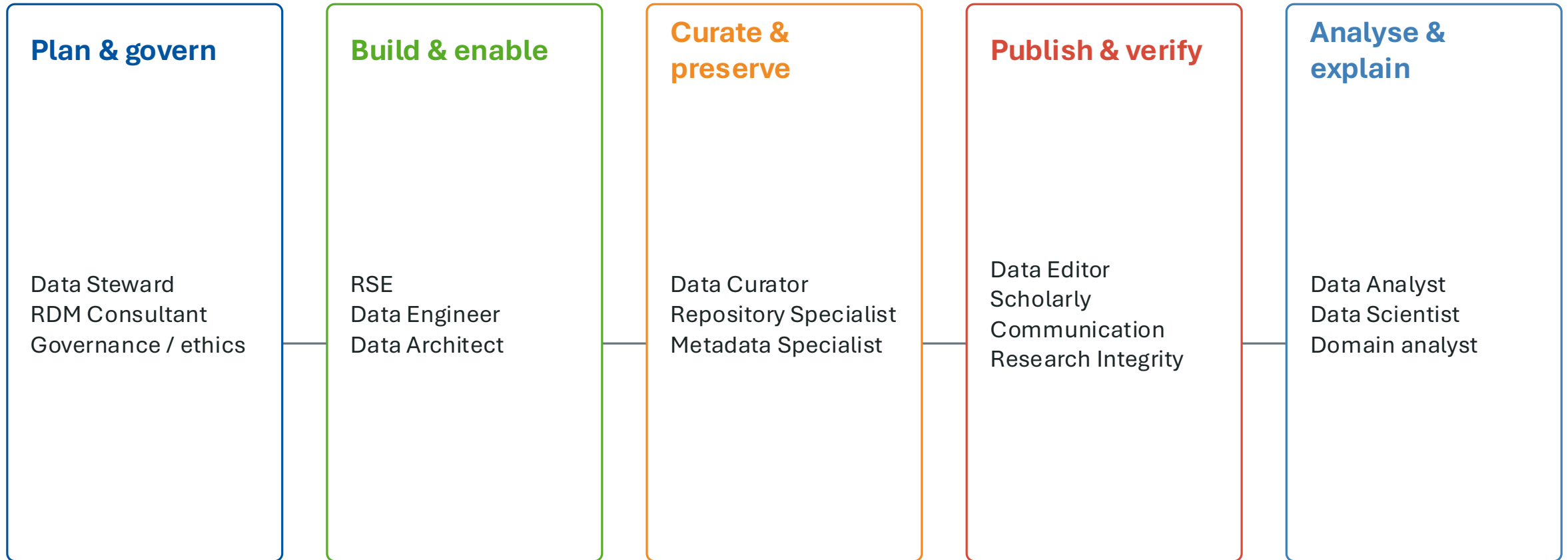
Milestones: from curation to stewardship

RDM Career Paths



Data-* roles today: one ecosystem, not one ladder

RDM Career Paths



Many job ads combine two or three of these functions.

Part 2

You investigate the roles yourself



Task 1 — Dataset emergency

A research group has collected two terabytes of sensor data. The project ends in six weeks.

File names are inconsistent.
The analysis code exists only on one laptop.
Nobody documented the software environment.
Some files contain sensitive location information.
No metadata standard has been selected.
The journal requires a data-availability statement.

Who should solve what?

Your output

Create a responsibility map with: problem · lead role · collaborating role · evidence for your choice.

Task 1 output — responsibility map

Problem	Lead role	Collaborating role	Reason / evidence
Code only on one laptop			
Sensitive location data			
Missing metadata standard			
Data availability statement			

Rule: one problem can require more than one expert. Mark that case clearly.

Task 1 **solution** — dataset emergency

Solution sheets

Problem	Likely lead role	Collaborating role	Reason / evidence
Inconsistent file names	Data Steward	Research team	Naming conventions are part of practical data-management guidance.
Code only on one laptop	Research Software Engineer	Researcher · IT	Reproducibility requires versioning, shared storage and runnable environments.
Missing software environment	Research Software Engineer	Data Curator	The analysis needs dependencies, versions and execution instructions.
Sensitive location data	Data protection / ethics specialist	PI · Data Steward	Access decisions require legal, ethical and risk assessment.
No metadata standard	Data Steward or Curator	Domain expert	Standards must fit both disciplinary meaning and reuse needs.
Data-availability statement	Researcher or Data Steward	Repository · Data Editor	The journal statement must match the actual deposit and access conditions.

Main insight: this is not one person's problem. It is a coordinated RDM workflow.

Task 2 — choose a role and become the expert

Step 1

Join a role group: Data Steward, Data Curator, Data Editor, RSE, Data Architect, Data Scientist/Analyst, RDM Librarian/Trainer.

Step 2

Read one source text and/or role page.
Highlight evidence, not slogans.

Step 3

Fill the overview table and prepare a 90-second role pitch.

**Your aim: explain what the role is meant to do —
without relying on the job title.**

Your investigation questions

1. What problem does the role solve?

2. Which five verbs describe the work?

3. What does the employee produce?

4. Who are the stakeholders?

5. Which role family fits best?

6. Which neighbouring role is most similar?

7. What evidence supports your decision?

Task 2 solution — role families at a glance

Solution sheets

Role family	Problem solved	Typical outputs	Closest neighbour
Data Steward	Researchers need practical, FAIR and responsible data practices.	DMP guidance · standards advice · workflow support · training	RDM consultant / Data curator
Data Curator	A dataset must be understandable, complete and reusable.	Curated package · metadata · quality checks · repository record	Repository specialist / Data steward
Data Editor	A publication needs data and code that support its claims.	Editorial assessment · data/code checks · reproducibility notes	Data curator / reviewer
RSE / Data Engineer	Research software and pipelines must be reliable and reusable.	Code · pipeline · package · reproducible environment	Data architect / Data scientist
RDM Librarian / Trainer	Researchers need guidance, skills and service orientation.	Consultation · course · checklist · service guide	Data steward / scholarly communication
Governance / Ethics	Data use must be lawful, responsible and risk-aware.	Access decision · agreement · risk assessment · policy	Data steward / legal office

These are role families. Real job ads often combine two or three of them.

Task 2 **solution** — five verbs that reveal the work

Solution sheets

Role	Five useful evidence verbs
Data Steward	advise · coordinate · translate · implement · train
Data Curator	check · enrich · document · preserve · publish
Data Editor	review · verify · assess · request · connect
RSE / Data Engineer	develop · automate · test · package · maintain
Data Architect	model · standardise · integrate · govern · assure
RDM Librarian / Trainer	teach · consult · guide · signpost · engage
Data Scientist / Analyst	analyse · model · visualise · interpret · report

If your evidence verbs differ, that is fine. They should still describe actual work in the source.

Job-ad hunt — what to search for

Search titles

Data Steward · Research Data
Manager · Data Curator · Data Editor ·
Research Software Engineer

Search places

jobs.ac.uk · Euraxess · university job
portals · repository job pages ·
LinkedIn

Search evidence

Responsibilities · required skills ·
stakeholders · outputs · contract
context

Group workflow: 25 minutes

- 1 Read** Scan the source and job ad. Do not start with the title.
- 2 Mark** Highlight verbs, outputs, stakeholders, competencies and lifecycle stage.
- 3 Decide** Choose the best role family and the closest neighbouring role.
- 4 Fill** Complete the overview table with evidence.
- 5 Pitch** Prepare a 90-second explanation for the class.

Prepare your 90-second role pitch

Use this exact structure:

We investigated ...

This role solves the problem that ...

The five verbs that best describe the work are ...

The person produces ...

The key stakeholders are ...

We classify it as ...

The closest neighbouring role is ..., but the difference is ...

Our strongest evidence is ...

Task 3 **solution** — job-ad forensics checklist

Question	Strong answer should include
1. Problem solved	A need or pain point, not just a title: “researchers need reusable, compliant data.”
2. Five verbs	Action verbs from the source: e.g. curate, advise, automate, review, preserve.
3. Outputs	Concrete work products: DMP, metadata, software, repository record, policy, report.
4. Stakeholders	Who receives the service: researchers, PI, repository, journal, funder, IT, public.
5. Role family	The best fit from the evidence: steward, curator, RSE, editor, trainer, governance.
6. Neighbouring role	A plausible confusion case and the decisive difference.
7. Evidence	At least three quotes or paraphrased details from responsibilities/requirements.

Sample filled table — Data Steward

Question	Possible solution
Problem solved	Research teams need FAIR, documented, responsible and reusable data practices in daily work.
Five verbs	advise · coordinate · document · standardise · train
Employee produces	DMP guidance, metadata recommendations, repository decisions, workflows, training material.
Stakeholders	Researchers, PIs, repository staff, IT, ethics/legal services, funders.
Role family	Plan & govern / stewardship.
Neighbouring role	Data Curator — but the Steward supports processes across the lifecycle, not only dataset publication.
Evidence	Look for DMPs, FAIR, metadata, governance, training, project support and coordination.

Sample filled table — Data Curator

Question	Possible solution
Problem solved	A dataset must be checked, described, preserved and made reusable for other researchers.
Five verbs	curate · validate · enrich · publish · preserve
Employee produces	Curated dataset package, metadata record, quality notes, persistent identifier, repository entry.
Stakeholders	Authors, repository, future reusers, data stewards, publishers, funders.
Role family	Curate & preserve.
Neighbouring role	Data Steward — but the Curator works more directly on the dataset package and repository quality.
Evidence	Look for metadata, quality assurance, repository deposit, formats, preservation and reuse.

A curation role usually has stronger evidence around dataset quality and repository readiness.

Sample filled table — Research Software Engineer

Question	Possible solution
Problem solved	Research depends on software and pipelines that must be reliable, maintainable and reproducible.
Five verbs	develop · test · automate · package · maintain
Employee produces	Research software, analysis pipelines, documentation, tests, containers, reusable tools.
Stakeholders	Researchers, users, software team, infrastructure services, data analysts, funders.
Role family	Build & enable / research software.
Neighbouring role	Data Scientist — but the RSE’s primary output is robust software, not an analysis result.
Evidence	Look for software engineering, Git, testing, CI/CD, reproducibility, user collaboration and maintenance.

The decisive evidence is often the expected work product: software or pipeline.

Boundary debate — choose a side

Move to: AGREE · UNSURE · DISAGREE

1. A Data Scientist is automatically an RDM professional.
2. A Data Steward should personally solve all legal and technical problems.
3. Data curation starts when the publication is almost finished.
4. FAIR data must always be openly accessible.
5. Communication skills can be more important than programming skills in some RDM roles.

Exit ticket — where might you fit?

Complete these sentences:

The role I currently find most attractive is ...

The type of problem I would like to solve is ...

One relevant skill relevant for a Data-* -role I already have is ...

One skill I need to develop is ...

One concrete action I can take in the next four weeks is ...

Do not ask first: “What is the title?”

Ask first: “What research-data problem is this person hired to solve?”

That question is the key to understanding RDM career paths.

FAIR4Studium

Your takeaway for your daily (academic) life



Why you should care

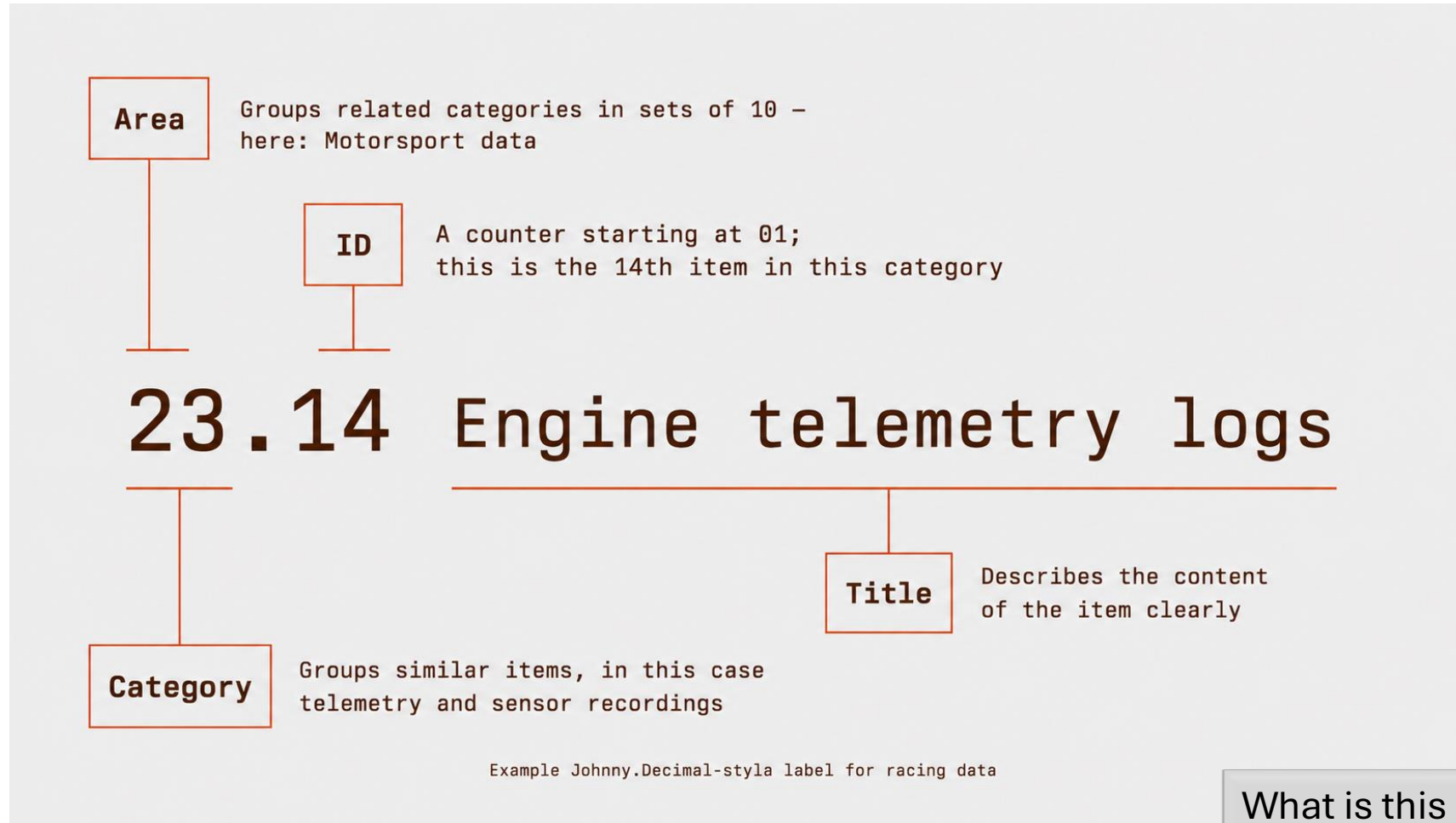
Research data management is not only for large funded projects.



Core message: data that is well organized is easier to trust.

Practice 1 — Make the project addressable

Johnny.Decimal gives every piece of information a stable address.



Source idea: Johnny.Decimal — a structured numbering system for giving information an address.

What is this „JohnnyDecimal“-System?
[Short Introduction to JD](#)



Hands-on 1 — Build a mini project map

Create for your experiment a structure according to JohnnyDecimal

INPUT: MESSY FOLDER

Survey_FINAL.xlsx

notes new.docx

data old.csv

graphs2.png

literature

thesis stuff



OUTPUT: ADDRESSABLE MAP

11 thesis-project

11.01 plan-and-question

11.02 literature

11.03 raw-data

11.04 analysis

11.05 outputs

Think about ...

What are your 3–5 project categories?
Where do raw data live?
What should never be edited directly?

Deliverable: JD-System

You can use the sandbox at
[JD Generator](#)



Practice 2 — Name files for sorting and searching

A good file name says what it is before you open it.

YYYY-MM-DD_topic_sample_stage__v###.ext

date

topic

sample

stage

version

Example

2026-07-26_survey-responses_raw__v01.csv

Good naming is metadata at file level.

PUBLIC SERVICE ANNOUNCEMENT:

OUR DIFFERENT WAYS OF WRITING DATES AS NUMBERS CAN LEAD TO ONLINE CONFUSION. THAT'S WHY IN 1988 ISO SET A GLOBAL STANDARD NUMERIC DATE FORMAT.

THIS IS ***THE*** CORRECT WAY TO WRITE NUMERIC DATES:

2013-02-27

THE FOLLOWING FORMATS ARE THEREFORE DISCOURAGED:

02/27/2013 02/27/13 27/02/2013 27/02/13
20130227 2013.02.27 27.02.13 27-02-13
27.2.13 2013.II.27. 27½-13 2013.158904109
MMXIII-II-XXVII MMXIII ^{LVII}/_{CCCLXV} 1330300800
((3+3)×(111+1)-1)×3/3-1/3³ 2013
10/11011/1101 02/27/20/13 0²1³2⁴3⁷8
5 67 8 

<https://xkcd.com/1179/>

Hands-on 2 — Rename the chaos

Doublecheck your files and translate weak file names into a convention.

Before

interview notes.docx

data_newer.csv

chart FINAL.png

transcript Anna neu.txt

summary_for_handover_LAST.docx



After

2026-07-26_interview-notes_p01_raw_v01.docx

2026-07-26_survey-responses_raw_v02.csv

2026-07-26_response-rate_figure_v01.png

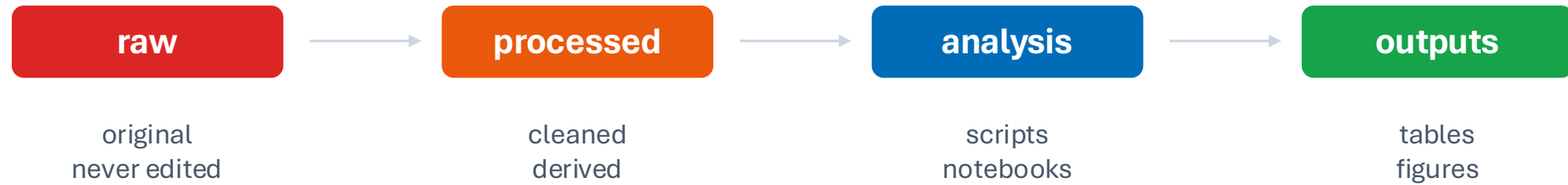
2026-07-26_interview-transcript_p03_raw_v01.txt

2026-07-26_project-handover_summary_v03.docx

Pair check: Can someone understand the file without opening it?

Practice 3 — Keep raw data raw

Make the difference between source material, processing, analysis, and outputs visible.



Golden rule If a result cannot be regenerated from raw data + documented steps, the project is fragile.

Practical habit: make raw folders read-only or add “DO_NOT_EDIT” to the folder note.

Practice 5 — Back up before you need it



Back Up Your Work

Protect your files with the 3-2-1 backup rule.



Prevent data loss

Accidents happen. Backups protect you.



Recover with confidence

Restore your work quickly and easily.



Stay secure

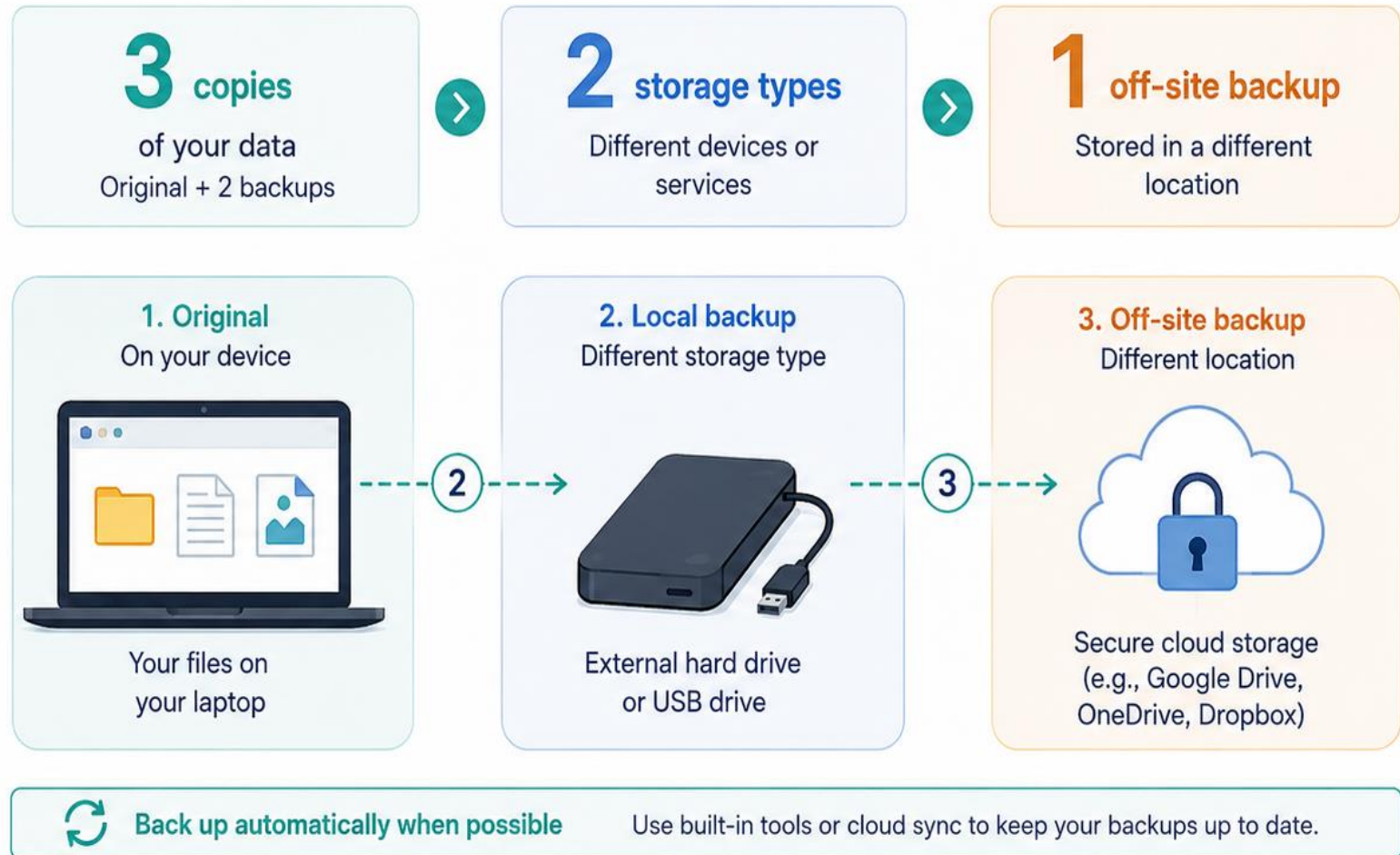
Keep a copy off-site for extra protection.



Back up automatically

Set it and forget it—let your tools do the work.

The 3-2-1 Backup Rule for Students



Practice 6 — Versioning without drama

Versioning means you can answer: what changed, when, and why?

Weak versioning

final.docx

final2.docx

really_final.docx

final_REALLY.docx



Better choices

Manual file versions

v01, v02 + change log

Git

code, scripts, notebooks, text

Repository / storage

larger data + released versions

Commit message formula

**verb + object +
reason**

clean survey data
add codebook
fix missing values

Do not use Git for huge raw binaries unless you know why.

Practice 7 — Share responsibly

Reusable does not always mean public.



Findable

Can others locate it?



Accessible

Can they access it under clear conditions?



Interoperable

Can tools and people combine it with other data?



Reusable

Is it described well enough to use again?

**Before sharing,
decide:**

license · sensitive data · consent · repository · citation · embargo/access restrictions

As open as possible, as closed as necessary.

Practice 8 — which file type to choose

ASCII

x-fmt/111
.txt
Editor
111

fmt/1149
.md
Markdown
111

x-fmt/195
.sgml
Docbook
111

fmt/471
.html
Chrome
111

fmt/103
.xhtml
110

fmt/119
.bmp
MS-Paint
001

x-fmt/276
.doc
MS-Word
001

fmt/412
.docx
MS-Word
101

fmt/291
.odt
LibreOffice
011

fmt/280
.tex
LaTeX
111

fmt/552
.indd
InDesign
001

fmt/1091
.sla
Scribus
110

fmt/160
.dvi
LaTeX
010

fmt/124
.eps
Ghostscript
111

fmt/476
.pdf
Acrobat
001

x-fmt/126
.ppt
PowerPoint
001

fmt/215
.pptx
PowerPoint
101

fmt/293
.odp
LibreOffice
011

fmt/119
.bmp
MS-Paint
001

fmt/730
.dng
001

fmt/13
.png
011

x-fmt/387
.tif
001

fmt/4
.gif
001

fmt/645
.jpg
001

x-fmt/392
.jp2
001

fmt/532
.dxf
Autocad
101

fmt/92
.svg
Inkscape
111

fmt/559
.ai
Illustrator
001

fmt/297
.odp
LibreOffice
011

fmt/427
.cdr
CorelDraw
000

fmt/1
.wav
001

fmt/134
.mp3
001

fmt/596
.m4a
001

fmt/203
.ogg
011

x-fmt/386
.mpg
001

fmt/640
.m2v
kaffeine
001

fmt/573
.webm
chrome
011

fmt/5
.avi
vlc
011

fmt/569
.mkv
vlc
011

fmt/199
.mp4
vlc
001

x-fmt/384
.mov
Quicktime
001

x-fmt/384
.mj2
001

fmt/506
.swf
Flashplayer
001

not listed
.pez
Prezi
001

x-fmt/18
.csv
111

x-fmt/57
.xls
MS-Excel
001

x-fmt/214
.xlsx
MS-Excel
101

fmt/295
.ods
LibreOffice
011

fmt/244
.kml
111

fmt/1134
.gpx
001

fmt/903
.blend
Blender
011

fmt/1210
.obj
101

fmt/1209
.dae
collada
111

fmt/582
.xthreed
110

fmt/831
.ply
001

fmt/410
.arc
WayBack
010

fmt/2099
.warc
WayBack
101

fmt/
.zip
001

fmt/
.tar
011

fmt/
.hdf5
C#
010

PRONOM-PUID: Click on it for more details

Mimetype-Suffix and Application Example: Click on it for more details

LTP-criteria	Format	License	Usage
Value	bin	text	prop open rare common
Code	0	1	0 1 0 1

Apply Filters

Filter Formats recommended for Long Term Availability

Filter Document Formats

Filter Text Formats

Filter Multimedia Formats

Filter Image Formats

Filter Data and 3D Formats

Reset Filters

<https://www.q-terra.de/lzv/>

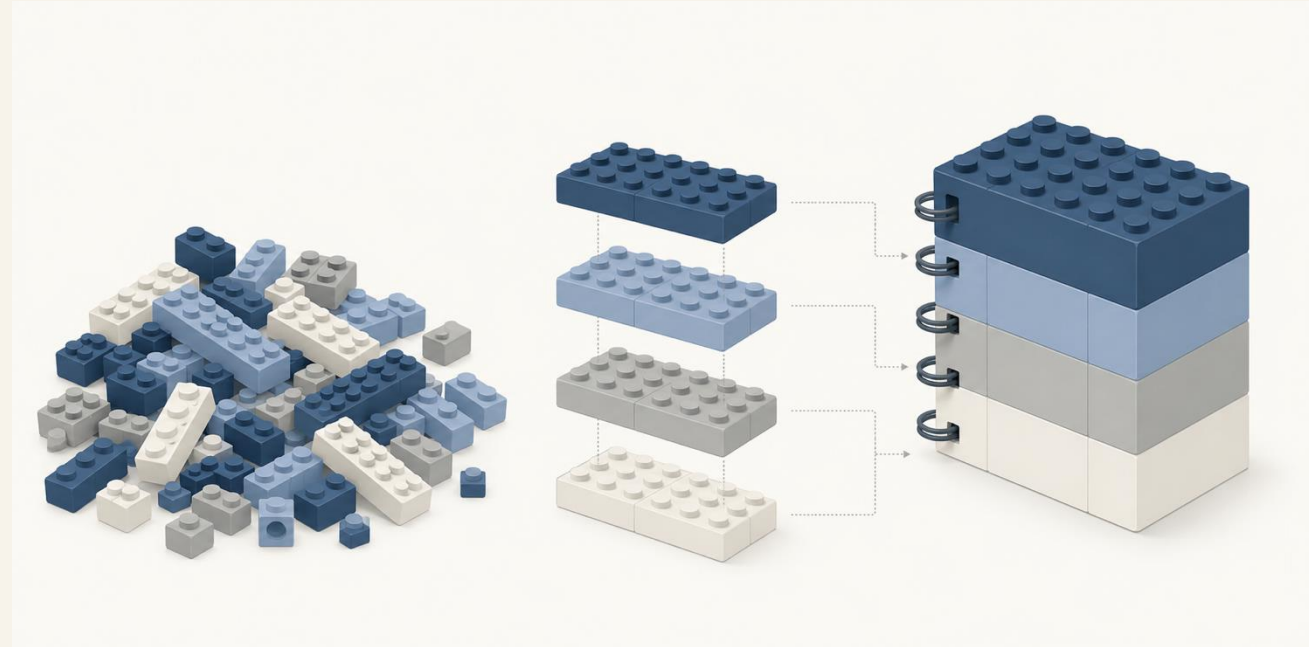
13

Practice 9

Literate Programming

Build research code like a rebuildable instruction model.

**Not a pile of code bricks.
A build logic someone can follow.**



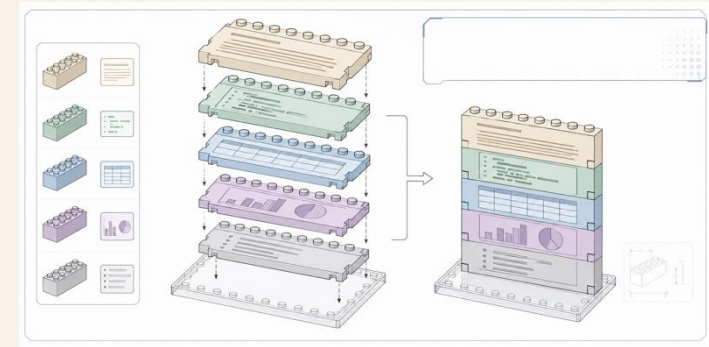
Hoare's principle: documentation belongs inside the build

“Documentation must be regarded as an integral part of the process of design and coding.”

“A good programming language will encourage and assist the programmer to write clear, self-documenting code, and even perhaps to develop and display a pleasant style of writing.”

— C. A. R. Hoare, Hints on programming language design, in *Computer Systems Reliability*, 1973, p. 195

LEGO analogy: the instruction manual is not extra packaging — it is part of what makes the model rebuildable.



Where literate programming shows up

Examples of tools that combine explanation, code, data, and output in one rebuildable document.

TeX / WEB family

WEB · CWEB · noweb

classic literate programming: weave documentation, tangle executable code

LaTeX ecosystem

Sweave · knitr · R Markdown · Quarto

analysis reports where text, code, tables, and figures live together

Emacs Org mode

org-babel

plain-text documents with executable source blocks and export targets

Notebook systems

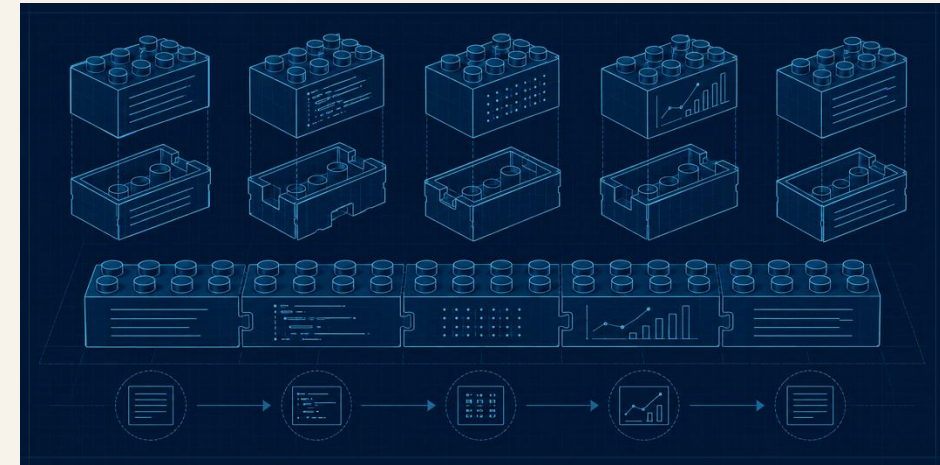
Jupyter · Wolfram · Pluto.jl

cells combine narrative, code, output, and interpretation

Web/data notebooks

Observable · marimo

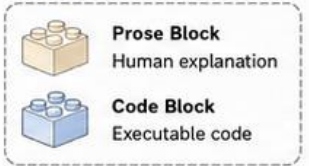
reactive or shareable documents for data stories and exploration



LEGO analogy: different brick systems, same principle — the reader can see how the model was assembled and rebuild it.

Not every notebook is automatically literate. The method depends on readable explanation, visible assumptions, and reproducible execution.

A single document interleaves explanation (prose) and code.



LITERATE PROGRAMMING SOURCE

Single Source Document

1. Introduction
This document explains the analysis pipeline and its components.

1.1 Load libraries
`import pandas as pd
import numpy as np
import matplotlib.pyplot as plt`

1.2 Overview of the data
We use the dataset described below. Key variables are summarized.

2. Data processing
`df = pd.read_csv("data.csv")
df = df.dropna()
summary = df.describe()`

2.1 Results
The results show a strong relationship between X and Y.

2.2 Visualization
`plt.figure()
plt.plot(df['X'], df['Y'])
plt.xlabel('X')
plt.ylabel('Y')
plt.savefig('plot.png')`

WEAVE
Render prose + code + results into human-readable documentation.

HTML / PDF REPORT (For Humans)

1. Introduction
This document explains the analysis pipeline and its components.

1.1 Load libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

1.2 Overview of the data
We use the dataset described below. Key variables are summarized.

2. Data processing

```
df = pd.read_csv("data.csv")
df = df.dropna()
summary = df.describe()
```

2.1 Results
The results show a strong relationship between X and Y.

Metric	Value
Correlation	0.82
RMSE	1.24
R ²	0.67

2.2 Visualization

```
plt.figure()
plt.plot(df['X'], df['Y'])
plt.xlabel('X')
plt.ylabel('Y')
plt.savefig('plot.png')
```

1.1 Load libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

2. Data processing

```
df = pd.read_csv("data.csv")
df = df.dropna()
summary = df.describe()
```

2.2 Visualization

```
plt.figure()
plt.plot(df['X'], df['Y'])
plt.xlabel('X')
plt.ylabel('Y')
plt.savefig('plot.png')
```

Extract code blocks into executable source files.

Reordered & Assembled

1.1_load_libraries.py

2_data_processing.py

2.2_visualization.py

main.py

```
from libs import *

def main():
    process()
    visualize()

if __name__ == "__main__":
    main()
```

analysis.py

```
import pandas as pd
import numpy as np

def process(path):
    df = pd.read_csv(path)
    df = df.dropna()
    return df.describe()
```

plotting.py

```
import matplotlib.pyplot as plt

def make_plot(df):
    plt.figure()
    plt.plot(df['X'], df['Y'])
    plt.xlabel('X')
    plt.ylabel('Y')
    plt.savefig('plot.png')
```

EXPLANATION FIRST.
Code is modular and embedded within the narrative.

EXECUTABLE SOURCE FILES (For the Machine)

1 One source of truth
All explanation and code live in one version-controlled document.

2 Executable code
Tangle produces clean, ordered source files that run.

3 Readable documentation
Weave generates polished reports that explain, visualize, and reproduce results.

KEY
→ Tangle (to machine)
→ Weave (to humans)
- - - Extracted code blocks

1 R — ROBUST



- local failure stays local
- modular repair
- recoverable structure

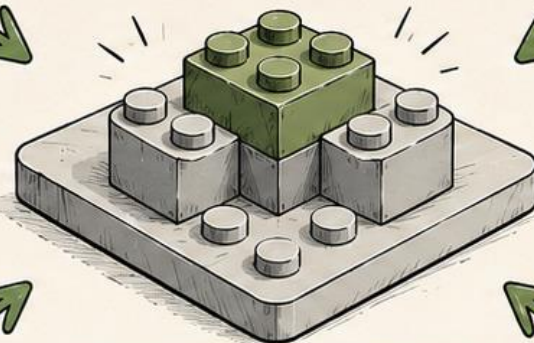
ROOT

A framework for resilient technology choices

2 O — OPEN



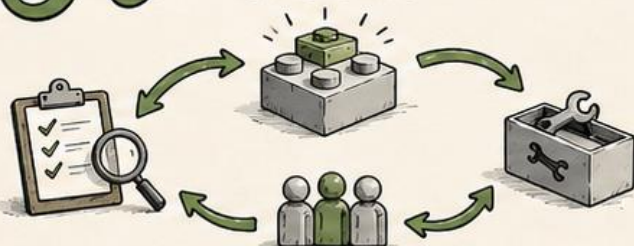
- inspectable
- interoperable
- shared standards



SHARED FOUNDATION

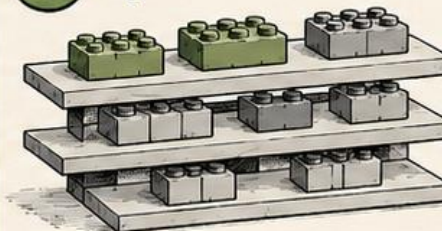
A common reference keeps compatibility and clarity strong.

3 O — ONGOING



- active maintenance
- continuous improvement
- community stewardship

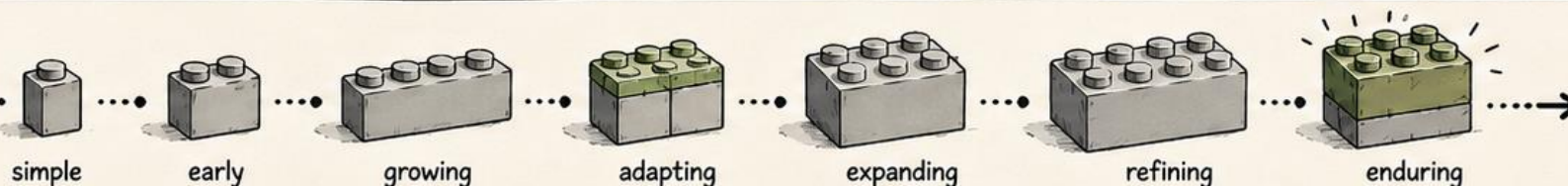
4 T — TIME-TESTED



- proven in practice
- mature
- trusted over time

- TODAY
- YEARS
- DECADES
- FOUNDATION

EVOLUTION OVER TIME.
SHARED INTERFACE,
MANY IMPLEMENTATIONS.



WHY ROOT?



CLEAR VALUES
Know what matters.



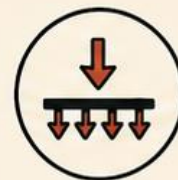
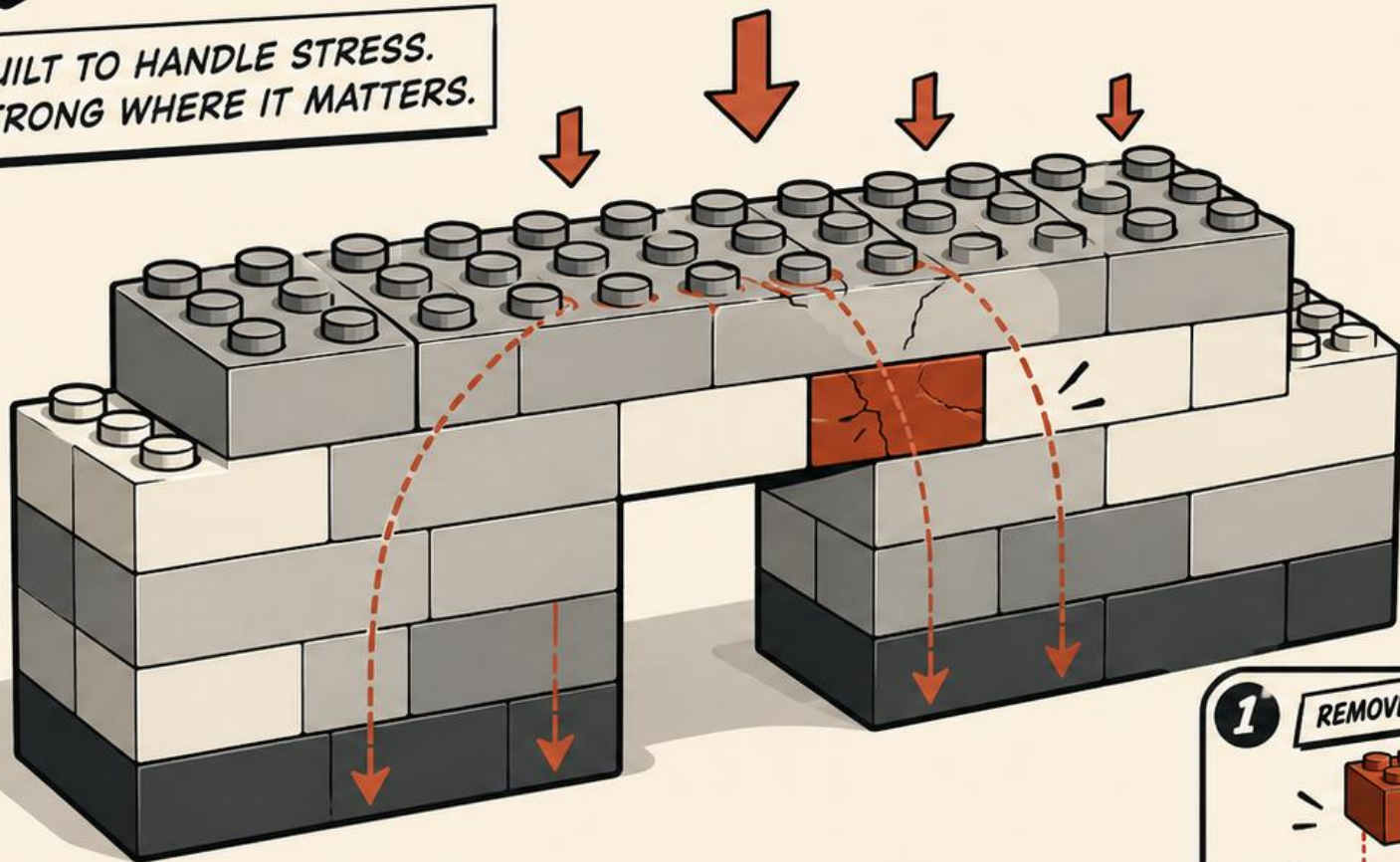
BALANCED TRADE-OFFS
Practical, not perfect.



LONG-TERM THINKING
Build for what lasts.

R — ROBUST

BUILT TO HANDLE STRESS.
STRONG WHERE IT MATTERS.



LOAD DISTRIBUTED

Across many connections



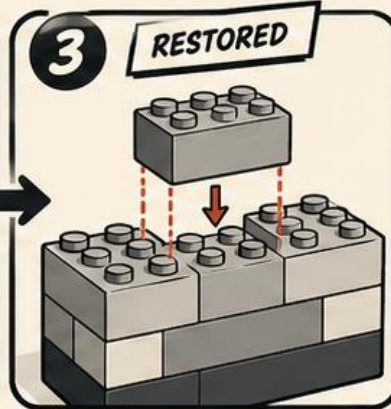
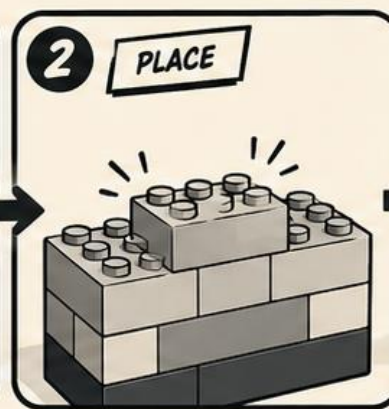
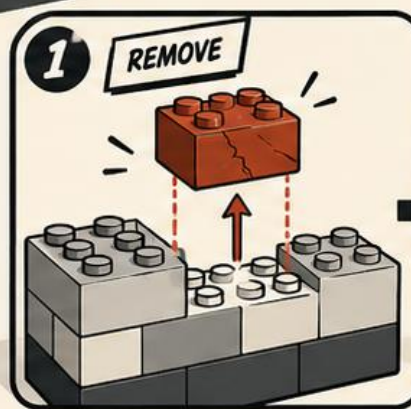
DAMAGE IS LOCAL

The rest stays strong



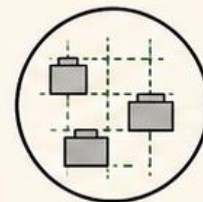
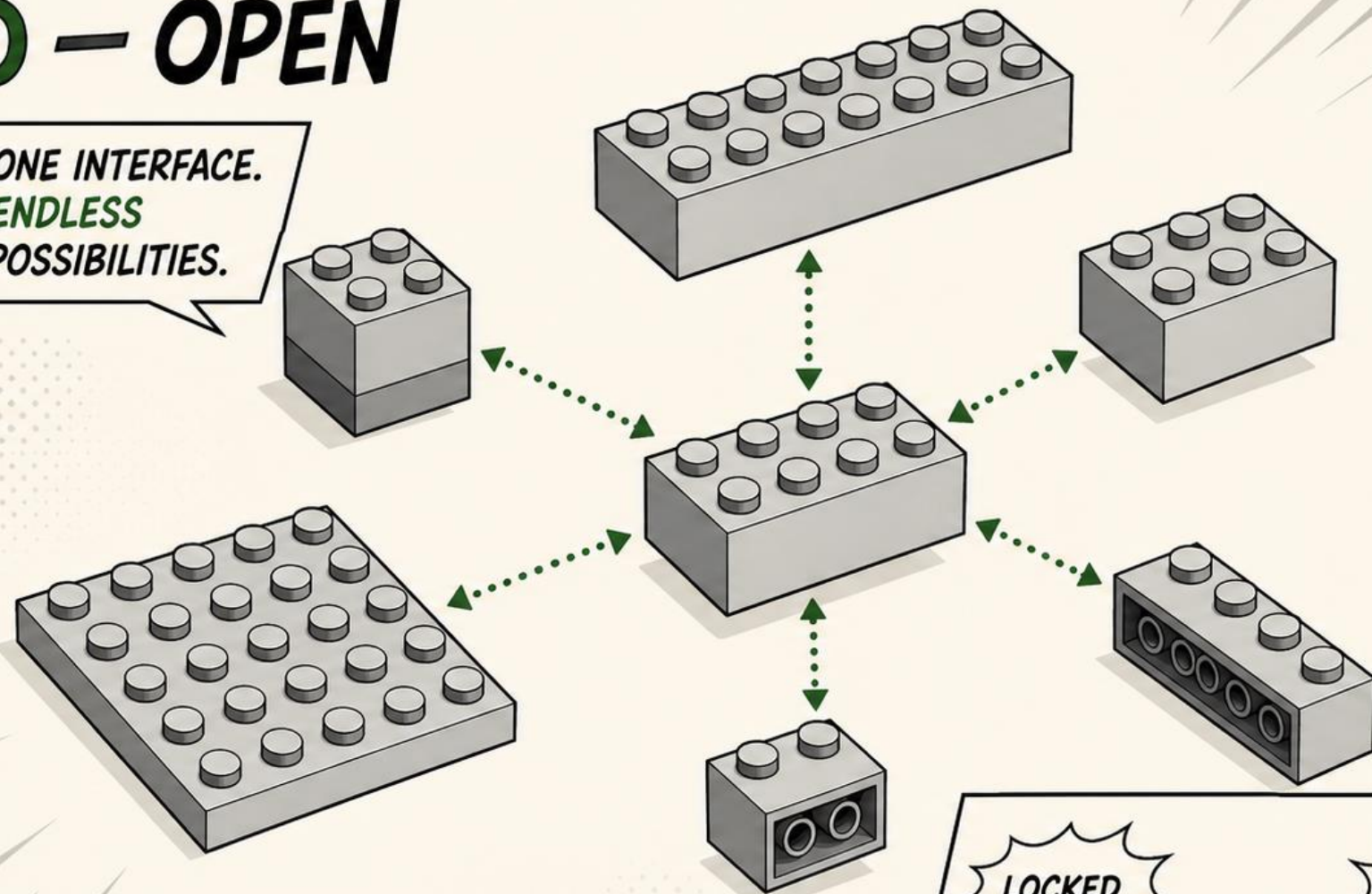
EASY TO REPLACE

Swap the part, not
the whole



O — OPEN

ONE INTERFACE.
ENDLESS
POSSIBILITIES.



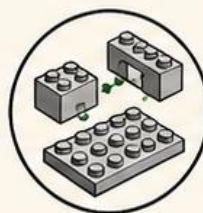
SHARED GEOMETRY

All parts align
on one grid.



VISIBLE RULES

Studs on **top**,
tubes on the **bottom**.



INTEROPERABLE

Different parts,
one system.

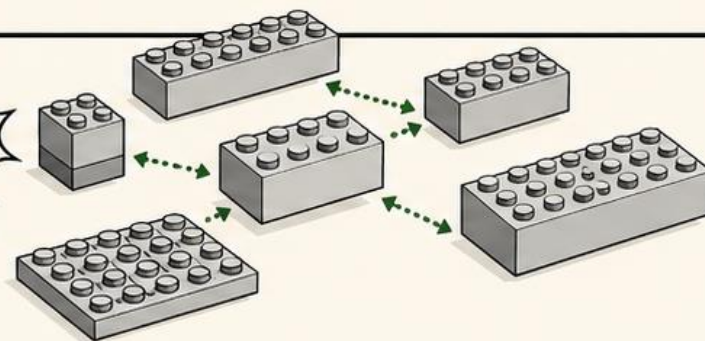
LOCKED

Solid, limited.

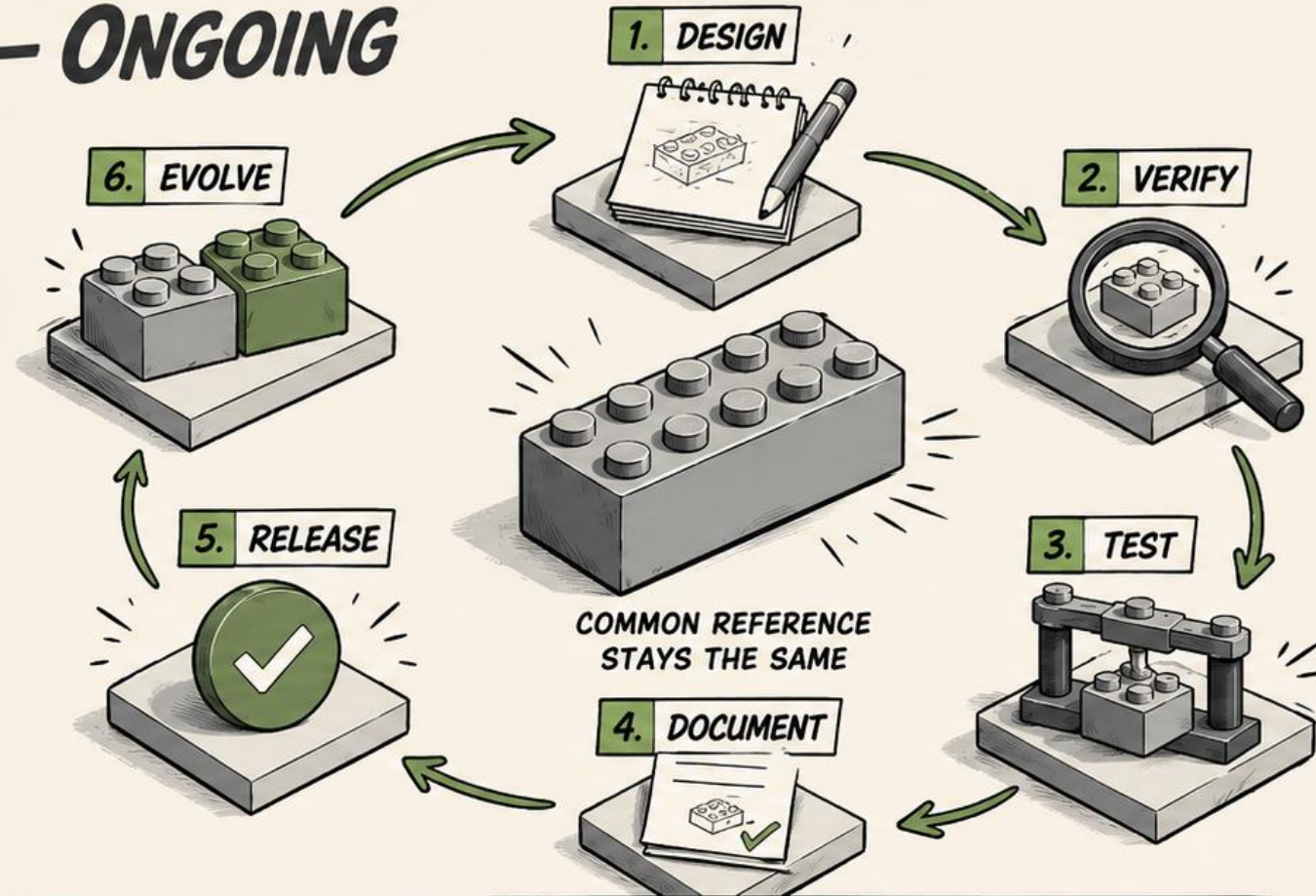


OPEN

Expandable,
compatible.



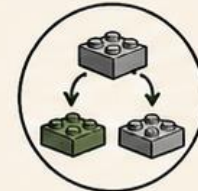
0 — ONGOING



ACTIVE MAINTENANCE
Continuous cycle of care keeps compatibility strong.

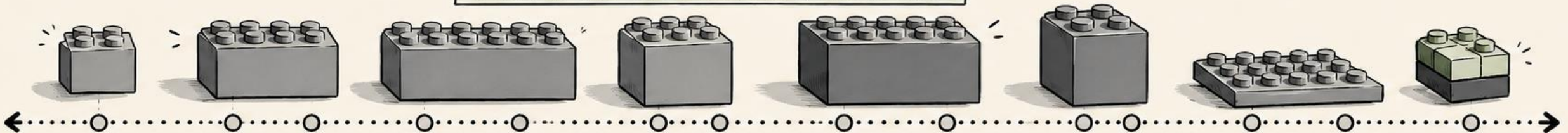


REFERENCE CHECKS
Regular verification ensures everything still fits.



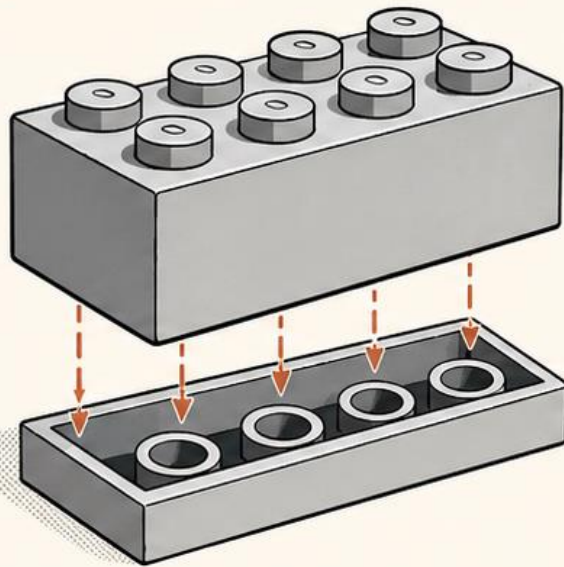
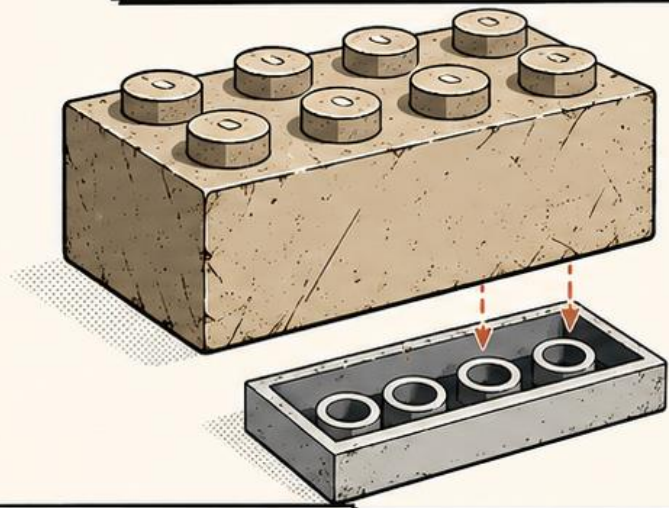
EVOLVE WITHOUT RESET
New bricks, same connection rules. Compatibility preserved.

EVOLUTION OVER TIME. ONE INTERFACE, MANY FORMATS.



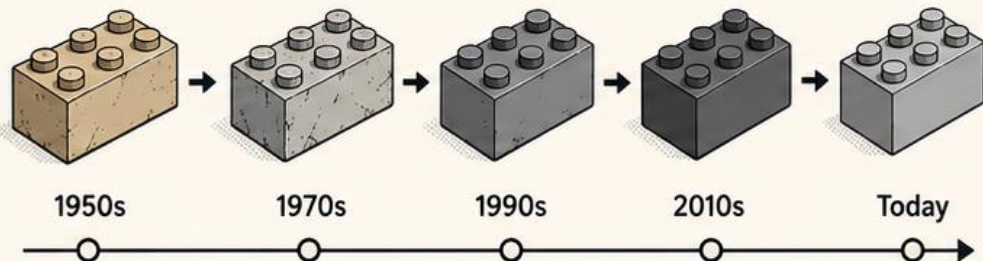
T — TIME-TESTED

BUILT TO LAST. DESIGNED TO CONNECT.

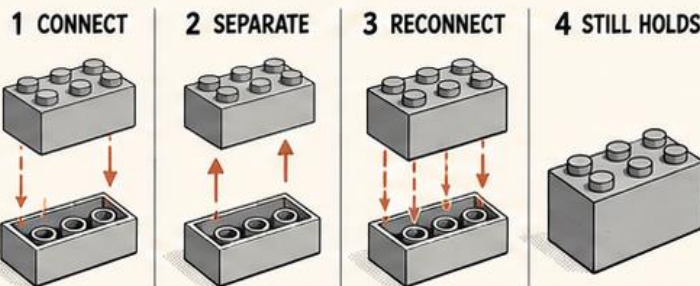


**SAME GEOMETRY.
ALWAYS
COMPATIBLE.**

BUILT ACROSS GENERATIONS



REPEATED CONNECTION CYCLES



TESTED OVER AND OVER. STILL HOLDS STRONG.



DECADES OF USE

Proven across generations and billions of builds.



REPEATED CONNECTION CYCLES

Designed to connect, disconnect, and connect again.



STILL USEFUL TODAY

Old and new bricks work together in everything you build.

ROOT and Resilient Digital Technologies

Why interlocking bricks are a model for long-lived digital infrastructure

PATENTED INTERFACE — STUD-AND-TUBE CONNECTION

FIG. 1 PERSPECTIVE VIEW

- Patent No. 3,005,282
- Stud-and-tube principle
- Publicly documented geometry

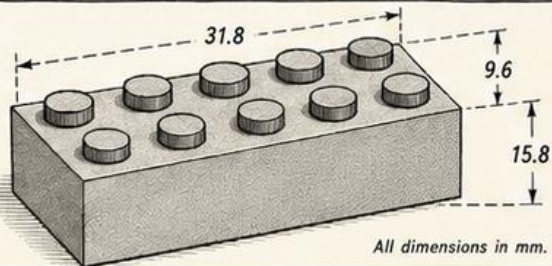
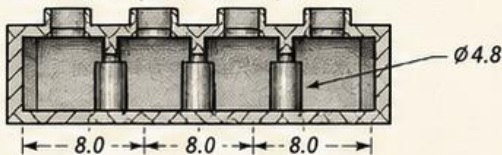
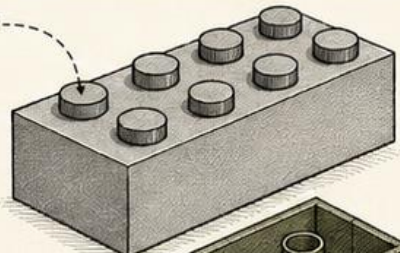


FIG. 2 CROSS-SECTION (LONGITUDINAL)



ONE SHARED CONNECTION PRINCIPLE

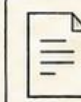
- 8 studs



- 3 support tubes

Stud-and-tube geometry creates a strong, simple, and repeatable connection.

RESILIENT DIGITAL TECHNOLOGY ECOSYSTEM



OPEN
FILE FORMATS
(plain text)



APIS
(well-defined
interfaces)



MODULAR
TOOLS
(plug, replace,
reuse)



DOCUMENTATION
(clear, accessible,
searchable)



VERSIONING
(track changes,
ensure stability)

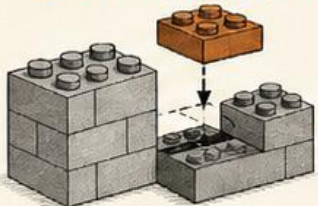


COMMUNITY
MAINTENANCE
(shared stewardship,
continuous care)

LONG-TERM REUSE & VALUE
(interoperable, adaptable, future-proof)

R — ROBUST

Local failure, global continuity



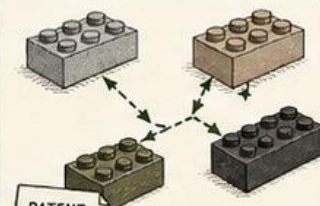
- modular repair
- replaceable parts
- fault tolerance



Replace one part.
Keep the system running.

O — OPEN

Shared rules, interoperable parts



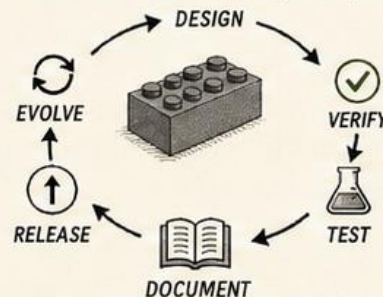
- open standards
- visible interfaces
- no permanent gatekeeper



Publicly documented interfaces
allow compatibility.

O — ONGOING

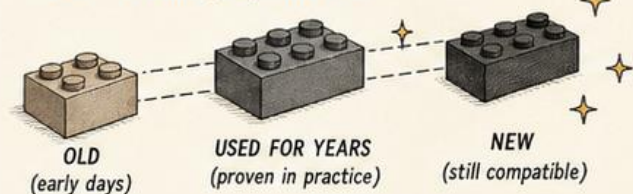
Maintenance keeps compatibility alive



- stewardship
- backward compatibility
- continuous improvement

T — TIME-TESTED

Proven by repeated use



- long-term usefulness
- stable conventions
- trusted through decades

Resilient digital technologies are like good bricks:
understandable, compatible, repairable, and still useful years later.

RESILIENT TECHNOLOGIES

WHY DECADES-OLD TOOLS DEFINE THE ROOT OF MODERN RESEARCH DATA MANAGEMENT

LUKAS C. BOSSERT
bossert@ic.rwth-aachen.de 0000-0003-3070-3968

Research data management (RDM) today relies on a growing landscape of platforms and specialized tools. While these innovations advance research, they also bring risks: short life cycles, proprietary dependencies, and limited sustainability. In contrast, long-established tools have proven to be remarkably resilient – distinguished by openness, interoperability and active communities. This poster demonstrates how such **resilient technologies** – despite their age – remain central to sustainable, transparent, and reproducible research data management today.

emacs (1976) / org-mode (2003)

Emacs is not just an editor but a fully extensible research and productivity environment. It can be adapted to cover the entire workflow—from notes and coding to analysis and publishing—without switching applications. The real strength unfolds with **org-mode**: plain text enriched with semantic markup and lightweight metadata. An Org file can act simultaneously as notebook, task manager, analysis script, and manuscript – a robust foundation for **reproducible research**. Org-mode supports **structured metadata** at the top of every file, also for export/provenance tracking. You can use the metadata for querying files.

Org-mode has **semantic markup**, similar but more extensible than markdown. It even supports native citing and referencing literature and publications.

Personal knowledge management and linking by creating a network of thoughts, snippets and information with **org-roam**. This is the backbone of your research project and can and will be extended continuously.

With **org-babel** you have embedded, executable code blocks for transparent analyses, best way to do **iterate programming**. This is like a Jupyter Notebook but more flexible in regard of programming languages to be used.

org-mode provides a lightweight macro system that lets you define reusable text snippets with parameters. Tables are plain-text, too, however you have full control over formulas and calculations.

curl (1997)

curl is one of the most universal and resilient tools for data transfer. Unlike graphical or browser-based downloads, **curl** can be scripted and parameterized for reproducibility, transparency, and automation—core principles of **FAIR** and **Good Scientific Practice**. The minimal usage pattern:

```
snippet (bash)
curl -DPTIME2 [URL]
```

sed (1974)

sed is a stream editor for transforming text—fast, scriptable, and reliable even on very large files. It excels at data cleaning tasks: normalizing metadata, harmonizing formats, and fixing systematic errors.

```
snippet (bash)
sed -i PLACE 's/pattern/replacement/g' [FILE]
```

The leading **s** stands for substitute. It tells **sed** to search for the given pattern and replace it with the specified replacement. The trailing **g** applies the substitution globally across the line, so every occurrence of the pattern is replaced, not just the first.

grep (1973)

When it comes to **searching and validating data**, **grep** is one of the oldest and most resilient tools. It is extremely fast, works on arbitrarily large files, and supports regular expressions for flexible pattern matching. The basic syntax is:

```
snippet (bash)
grep [FLAGE] PATTERN [FILE ...]
```

When it comes to verifying reproducibility and provenance, **diff** is one of the simplest yet most powerful verification tools. It compares files line by line and reports any deviations. This allows researchers to document data cleaning steps, valuable outputs, or monitor version transformations. After applying transformations (e.g., normalization with **sed**), **diff** can confirm the expected changes and provide a readable audit trail.

The basic syntax looks like:

```
snippet (bash)
diff [FLAGE] [FILE1] [FILE2]
```

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

#FDMRWTH

EMACS OKZ 2R

Version: 1.1.0
LuarBITEX, Version 1.24.0 (TeX Live 2024)
GNU Emacs 30.2 (org-mode: 9.8)
DOI: 10.5811/meda.17157398

RESILIENT TECHNOLOGIES

WHY DECADES-OLD TOOLS DEFINE THE ROOT OF MODERN RESEARCH DATA MANAGEMENT

LUKAS C. BOSSERT

bossert@itc.rwth-aachen.de

0000-0003-3976-3968

Research data management (RDM) today relies on a growing landscape of platforms and specialized tools. While these innovations advance research, they also bring risks: short life cycles, proprietary dependencies, and limited sustainability. In contrast, long-established tools have proven to be remarkably resilient—distinguished by openness, interoperability, and active communities.

This poster demonstrates how such **resilient technologies**—despite their age—remain central to sustainable, transparent, and reproducible research data management today.

emacs (1976) / org-mode (2003)

Emacs is not just an editor but a fully extensible research and productivity environment. It can be adapted to cover the entire workflow—from notes and coding to analysis and publishing—without switching applications. The real strength unfolds with **org-mode**: plain text enriched with semantic markup and lightweight metadata. An Org file can act simultaneously as notebook, task manager, analysis script, and manuscript—a robust foundation for **reproducible research**. Org-mode supports **structured metadata** at the top of every file, also for export/provenance tracking. You can use the metadata for querying files.

Org-mode has **semantic markup**, similar but more extensible than markdown. It even supports native citing and referencing literature and publications.

Personal knowledge management and linking by creating a network of thoughts, snippets and information with **org-roam**. This is the backbone of your research project and can and will be extended continuously.

With **org-babel** you have embedded, executable code blocks for transparent analyses, best way to do **literate programming**. This is like a Jupyter notebook but more flexible in regard of programming languages to be used.

org-mode provides a lightweight macro system that lets you define reusable text snippets with parameters. Tables are plain-text, too, however you have full control over formulas and calculations.

curl (1997)

curl is one of the most universal and resilient tools for data transfer. Unlike graphical or browser-based downloads, **curl** can be scripted and parameterized for reproducibility, transparency, and automation—core principles of **FAIR** and **Good Scientific Practice**. The minimal usage pattern:

```
snippet (bash)
curl [OPTIONS] [URLs]
```

sed (1974)

sed is a stream editor for transforming text—fast, scriptable, and reliable even on very large files. It excels at data cleaning tasks: normalizing metadata, harmonizing formats, and fixing systematic errors.

```
snippet (bash)
sed [FLAGS] 's/pattern/replacement/g' [FILE]
```

The leading **s** stands for substitute. It tells **sed** to search for the given pattern and replace it with the specified replacement. The trailing **g** flag applies the substitution globally across the line, so every occurrence of the pattern is replaced, not just the first.

grep (1972)

When it comes to **searching and validating data**, **grep** is one of the oldest and most resilient tools. It is extremely fast, works on arbitrarily large files, and supports **regular expressions** for flexible pattern matching.

The basic syntax is:

```
snippet (bash)
grep [FLAGS] PATTERN [FILE ...]
```

diff (1974)

When it comes to verifying reproducibility and provenance, **diff** is one of the simplest yet most powerful verification tools. It compares files line by line and reports any deviations. This allows researchers to document data cleaning steps, validate outputs, or monitor versioned transformations. After applying transformations (e.g. normalization with **sed**), **diff** can confirm the expected changes and provide a readable audit trail.

The basic syntax looks like:

```
snippet (bash)
diff [FLAGS] [FILE1] [FILE2]
```

make (1976)

A **Makefile** (without suffix) is an integral part of every project. It serves as a **Standard Operating Procedure (SOP)** that defines how individual tasks depend on each other and how they are executed.

In the context of research data management, **make** functions as a lightweight yet powerful workflow engine. Each rule represents not just a command, but a clearly defined dependency chain that transparently records how results are generated from inputs. This approach ensures traceability, reusability, and automation—core principles of **Good Scientific Practice** and **reproducible research**. Here is the basic design of a **makefile**:

```
snippet (make)
VAL := [7-1-1] := VALUE
.PHONY: TARGET TARGET TARGET
all: TARGET
TARGET: PREREQUISITE(S)
COMMAND
```

Version 1.1.0
Lualatex, Version 1.24.0 (TeX Live 2020)
GNU Emacs 30.2 | org-mode: 9.8
DOI: 10.5281/zenodo.17157588



#FDMRWTH



IT Center @ RWTH Aachen University
Department: Research Process & Data Management
Group: Technical Training & Consulting

Not all tools of lasting relevance can be introduced in depth here. The following list presents additional resilient technologies that remain fundamental to reproducible and sustainable research workflows.

find (1974) BSD/GP: Recursively searches complex directory structures to locate files, match patterns, or filter metadata. Essential for navigating large research repositories with nested folders or evolving data versions.

LaTeX (1984) L^AT_EX: The enduring standard for scientific writing and documentation—used for DMPs, reports, posters, and publications. Combines typographic precision with reproducibility and platform independence.

perl (1987) Artistic/GP: A versatile scripting language for text manipulation, pattern recognition, and metadata transformation. Ideal for parsing logs, transforming XML/JSON, or validating research data.

rsync (1996) GP: Synchronizes and mirrors directories across systems efficiently. Commonly used for backups, replication, and maintaining mirrored repositories. Transfers only differences, saving time and bandwidth.

SQLite (2000) Public Domain: A serverless SQL database stored in a single file—lightweight yet powerful. Excellent for metadata catalogs, portable archives, and small-scale analytical datasets.

Approved
ROOT TECHNOLOGY
ROBUST · OPEN
ONGOING · TIME-TESTED

