

TRACE: Transparency Certified

Lars Vilhuber et al.

2026-06-03



Follow along



[transparency-certified.github.io/trace-sivacor-presentation-](https://transparency-certified.github.io/trace-sivacor-presentation-2026/)

2026/



The Problem



Computational Reproducibility in the Social Sciences

- Research communities increasingly require sharing of **data, code, and methods**
- Trend is clear: **any data** that can reasonably be shared, and all code, **should be made available**
- Consumers combine data and code, **re-run it**, and validate the analysis



Computational Reproducibility in the Social Sciences

But what makes this hard in practice?

- Data withheld for **ethical**, legal, or contractual reasons



Computational Reproducibility in the Social Sciences

But what makes this hard in practice?

- Data withheld for ethical, legal, or contractual reasons
- Processing is time-consuming or requires **rare computing** resources



Computational Reproducibility in the Social Sciences

But what makes this hard in practice?

- Data withheld for ethical, legal, or contractual reasons
- Processing is time-consuming or requires rare computing resources
- Raw data may be transient, ephemeral, or **deleted**



Recent Evidence

A 2026 issue of *Nature* on social science reproducibility:

- Headline: **“Half of social-science studies fail replication test”**
 - But: only **30% of articles yielded data**

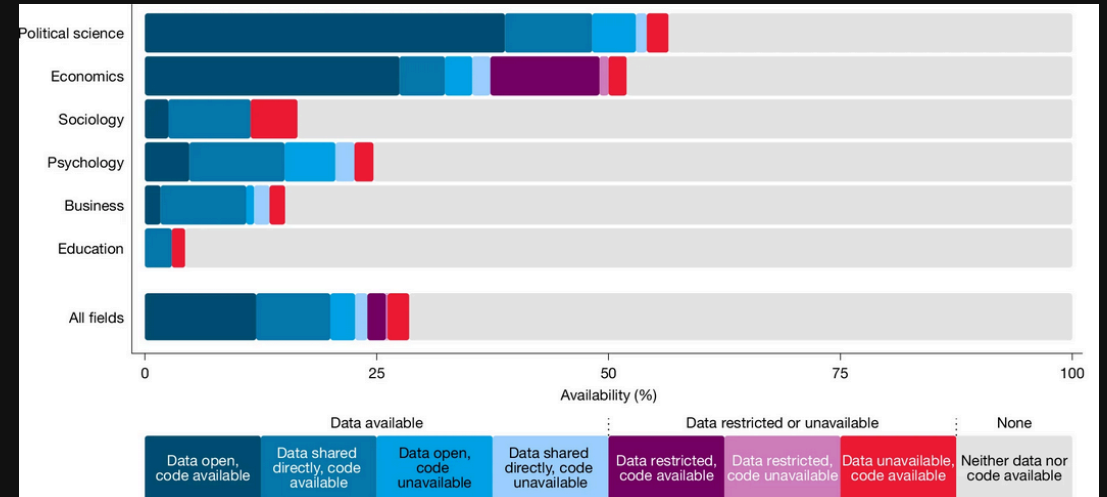


Fig 2, Tyne et al. (2026)

Recent Evidence

A 2026 issue of *Nature* on social science reproducibility:

- Headline: **“Half of social-science studies fail replication test”**
 - But: only 30% of articles yielded data; **only 24% were attempted**

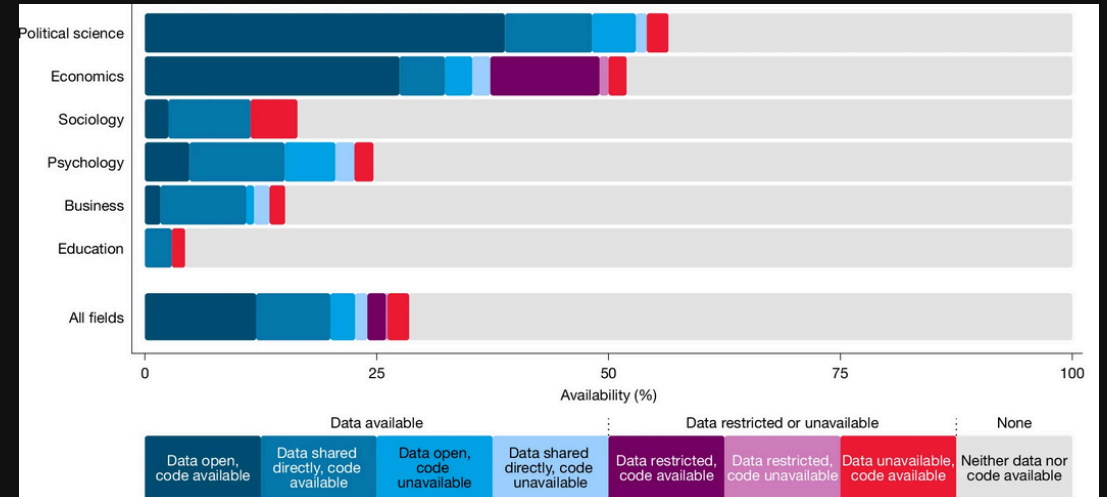


Fig 2, Tyne et al. (2026)

Recent Evidence

A 2026 issue of *Nature* on social science reproducibility:

- Headline: **“Half of social-science studies fail replication test”**
 - But: only 30% of articles yielded data; only 24% were attempted
 - **74% of those attempted** were exactly or approximately **reproducible**

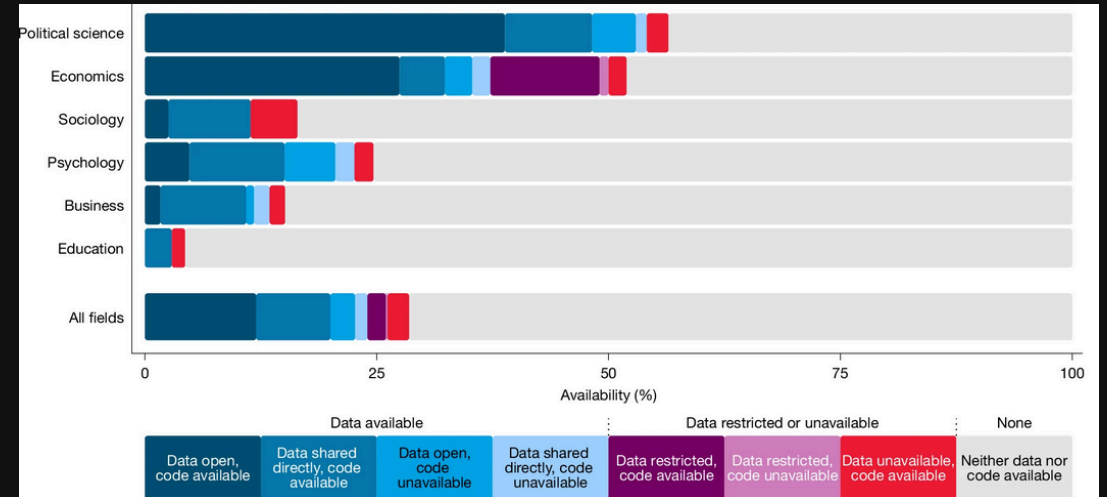


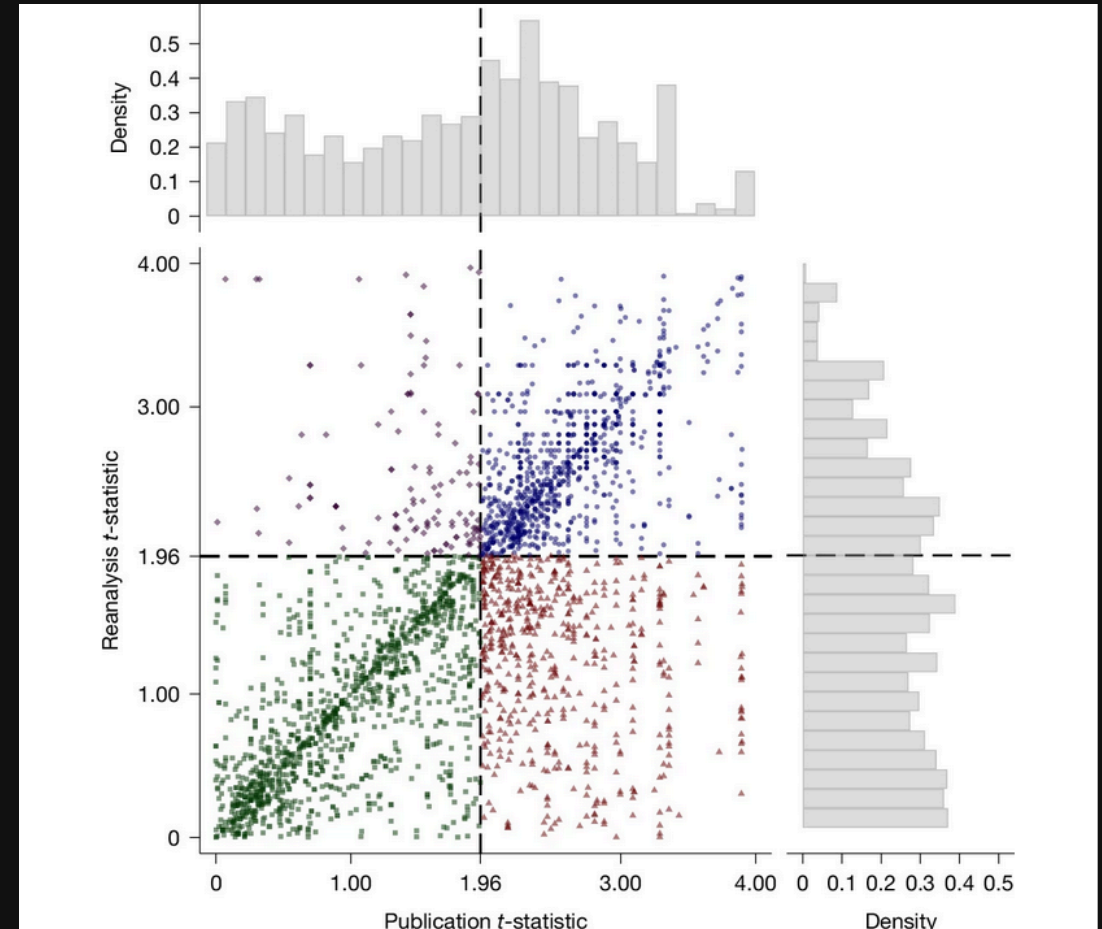
Fig 2, Tyne et al. (2026)



Enormous effort

A 2026 issue of *Nature* on social science reproducibility:

- Brodeur et al. (2026), crowdsourced with Sloan funding:
 - **85% of 110 articles** (2022–2023) were reproducible
 - Required **80 replication games** and **3,500+ researchers**



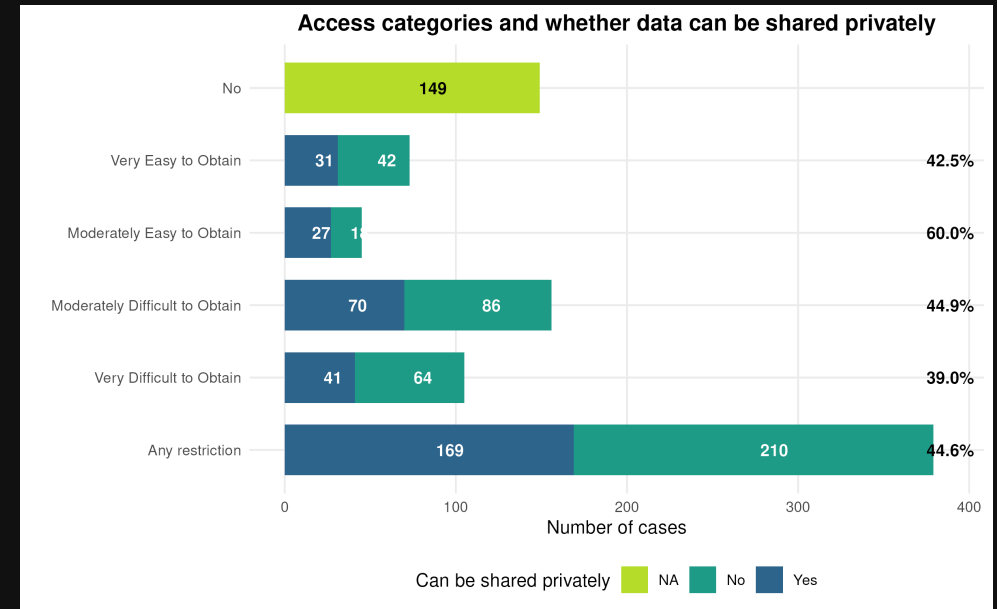
But: Restricted Data Remains Unassessed

None of these studies could access restricted data — entire swaths of social science literature remain outside scope.

The Scale of the Challenge

From the AEA Data Editor's experience (2025, 384 papers assessed):

- **38%** used data with no access restrictions — *in scope for replication studies*
- **62%** used data subject to access restrictions



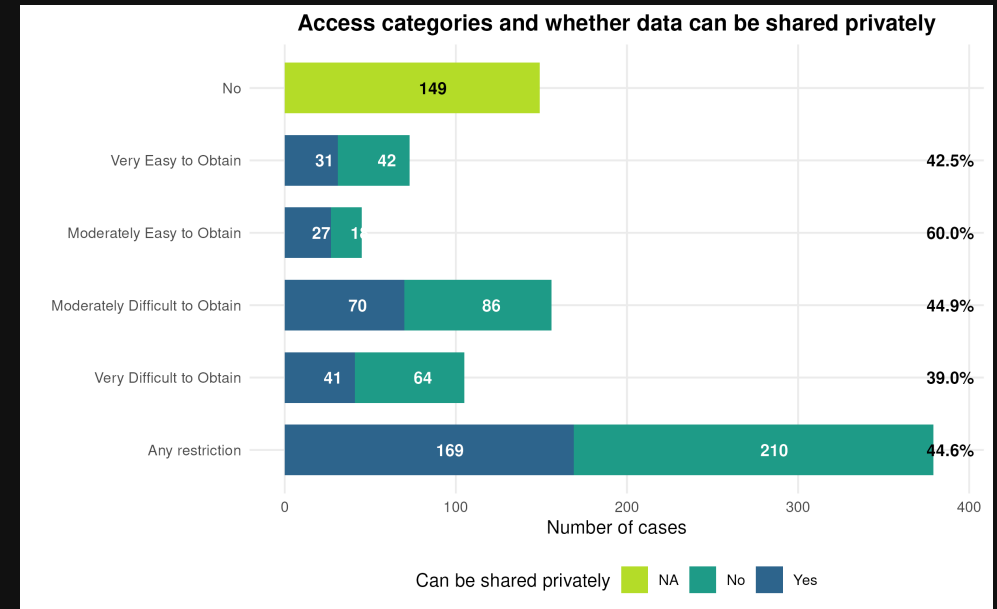
Access categories and data sharing



The Scale of the Challenge

Of those restricted papers:

- The AEA team obtained private access to **45%** of the 62%
- Conducted reproducibility checks despite data not being in public packages



Access categories and data sharing



An information and trust problem



An information and trust problem

- Verification at large cannot ascertain whether replication packages are reproducible.
- Yet many of those same packages are reproducible.
- That fact might rely on trust in data editors, or others.

The Key Question

What if it were possible to credibly demonstrate that the original execution of the computational artifacts occurred in a transparent fashion, even when data cannot be published, and is consistent with the deposited computational artifacts (code) and outputs (figures and tables)?

The Key Question

What if it were not necessary to re-run the code?



Certification

If reproducibility can be **certified at the source**, then:

- Readers can trust results and focus on **robustness**, not reproduction
- Researchers can convey **credibility** even when data cannot be shared
- The **scale** of verification effort **shrinks** dramatically



Certification is not new



Services now provide active verification:

| Service | Approach |
|----------------------|--|
| <i>cascad</i> | Certification service, access to confidential data |
| World Bank | Internal service, access to confidential data |
| Codeocean | Containerized capsules with manual compliance checking |



The Transparency Gap

When a reproducibility service has re-run code and issued a certificate — **what actually happened?**



The Transparency Gap

Questions that remain unanswered:

- Was work done with input/influence from authors, or in isolation?
- Was internet access available during the run?
- What state was a database in when first queried vs. when the service ran it?
- Was code modified (inadvertently or intentionally) by verifiers?

Absent **standardized protocols or vocabularies**, services remain opaque.

Readers, journals, and researchers cannot compare:

- Codeocean vs. World Bank
- cascad vs. AEA



The TRACE Framework



What is TRACE?

TRACE = Transparency Certified

A framework that allows inquiry into the reproducibility workflow **at any stage** — without requiring re-running the code.



Key insight

Document the **process**, not just the **outputs**

- **File arrangements** (manifests with checksums)
- **Processing steps** (software, timing, method, isolation)
- **Cryptographic signatures** by certifying organizations

Result

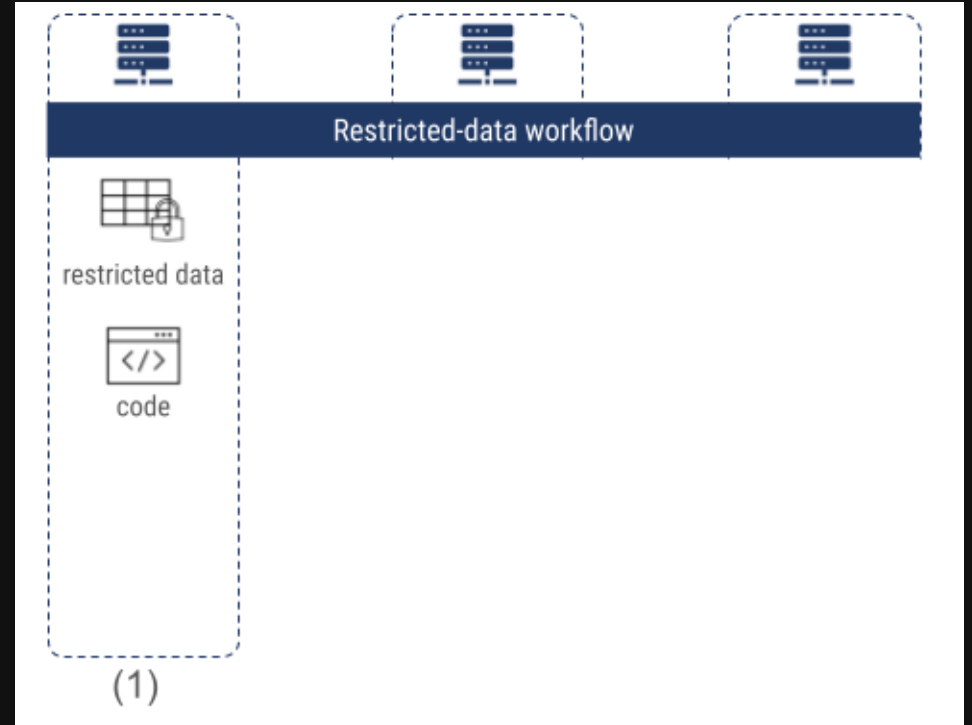
TRACE-compliant packages can be:

- Compared across services (Codeocean vs. World Bank)
- Inspected both by humans and automated scripts
- Trusted via organizational credibility chains

Generic Workflow (Before TRACE)

Consider a researcher using confidential data in a Restricted Access Data Center (RADC):

1. Researcher gets environment with confidential data, writes code

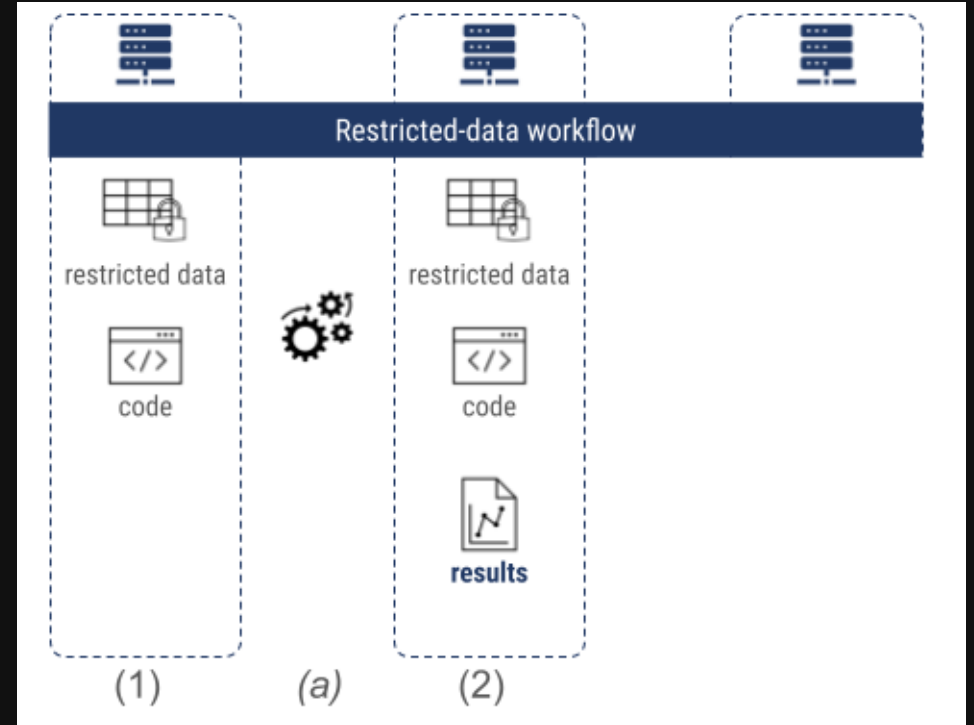


Generic workflow with confidential data

Generic Workflow (Before TRACE)

Consider a researcher using confidential data in a Restricted Access Data Center (RADC):

1. Researcher gets environment with confidential data, writes code
2. Code is executed (a) → output produced

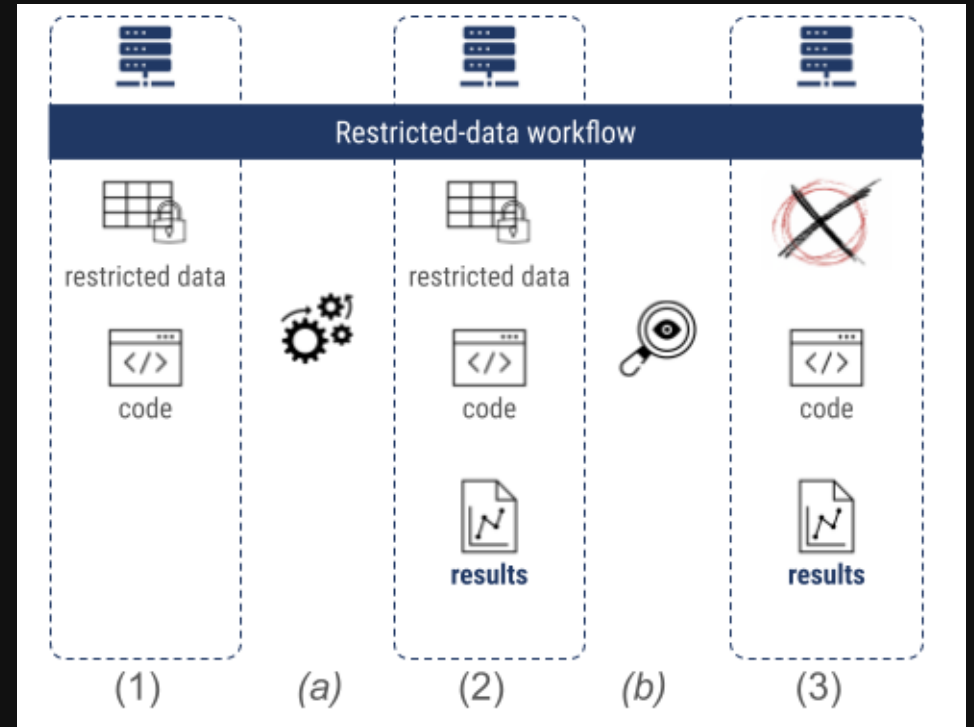


Generic workflow with confidential data

Generic Workflow (Before TRACE)

Consider a researcher using confidential data in a Restricted Access Data Center (RADC):

1. Researcher gets environment with confidential data, writes code
2. Code is executed (a) → output produced
3. Data custodian inspects, removes confidential data (b)
4. Researcher receives code + output only (arrangement 3)



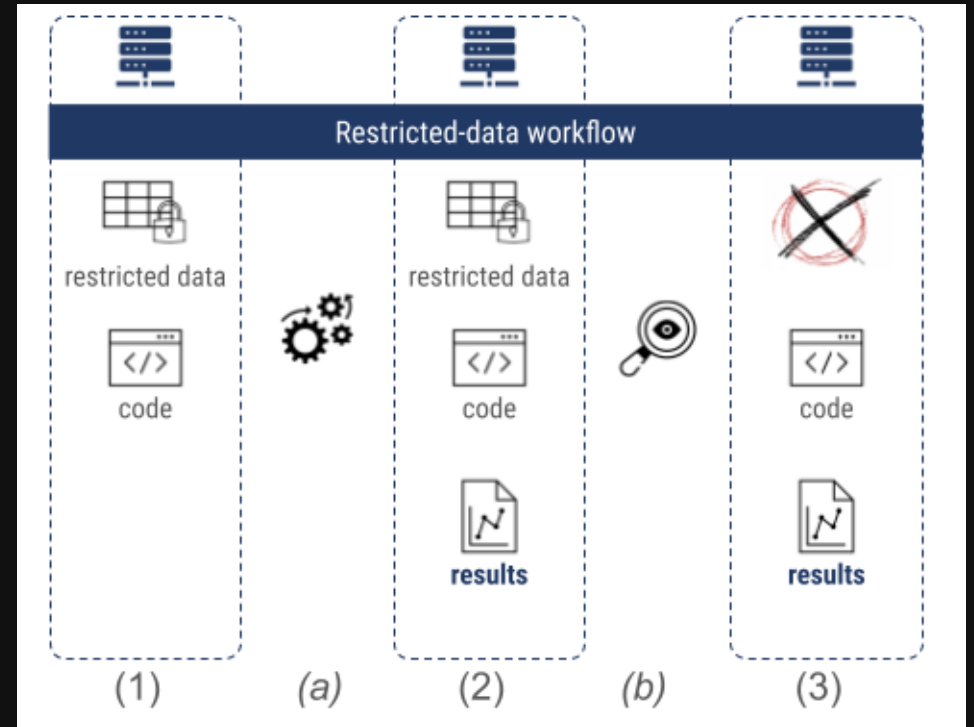
Generic workflow with confidential data

Generic Workflow (Before TRACE)

Problem:

- The World Bank certificate states in English that this process was followed.
- Codeocean points to an FAQ.

Neither is verifiable or machine-readable.



Generic workflow with confidential data

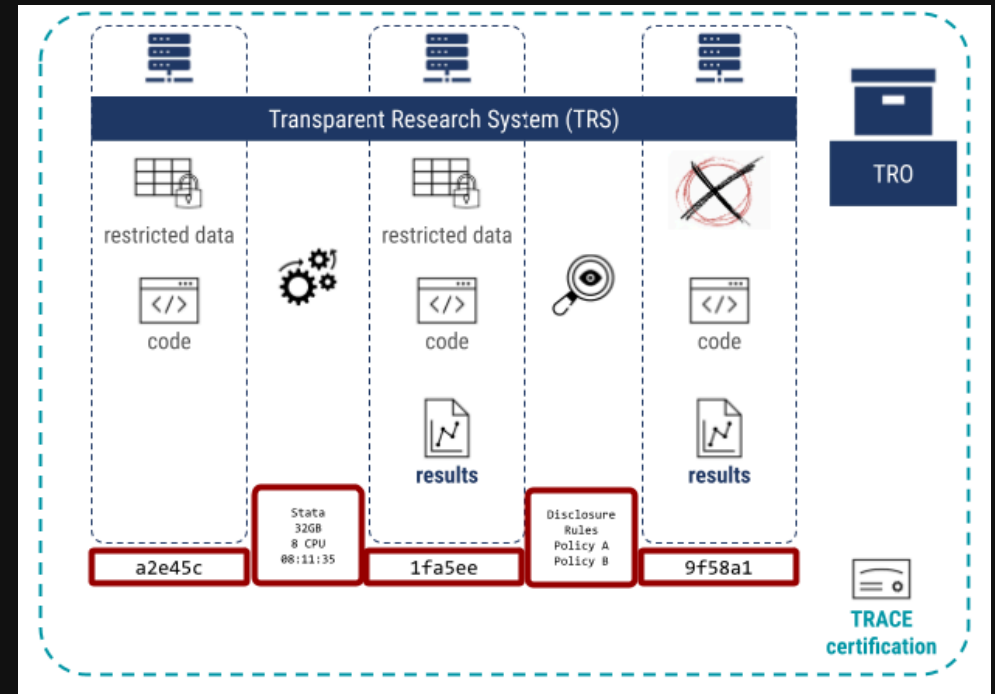
Making a Workflow TRACE-Compliant



Making a Workflow TRACE-Compliant

Add a few computationally easy steps:

1. Document each **file arrangement** (1, 2, 3) with manifests + checksums
2. Describe each **processing step** (software-driven or manual) with salient info

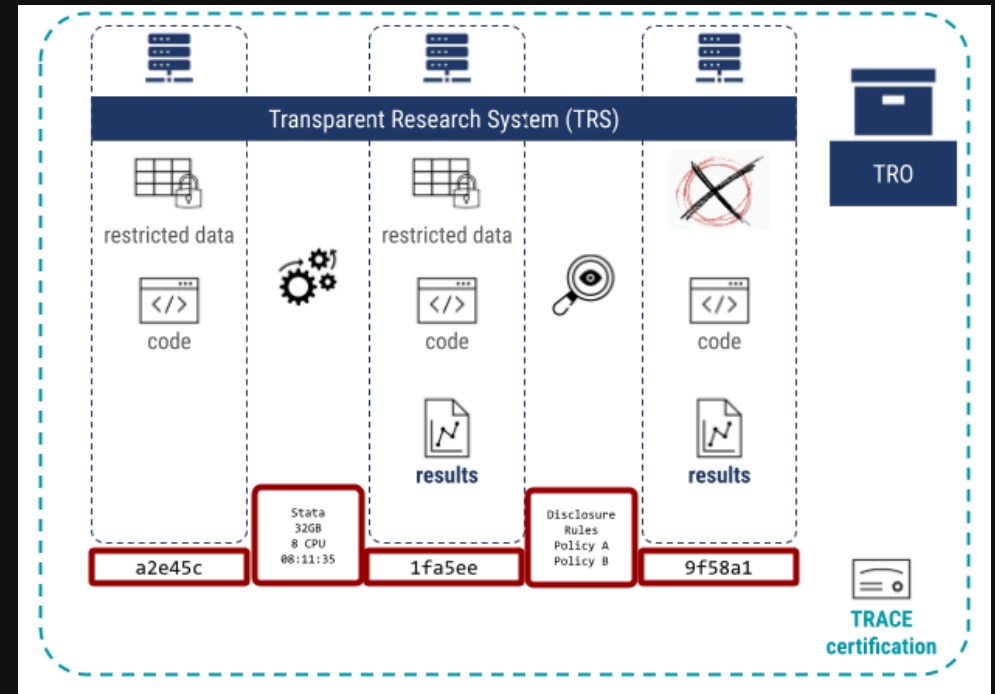


Example Trusted Research System

Making a Workflow TRACE-Compliant

Add a few computationally easy steps:

- Express in a **controlled vocabulary** (TROV)
- Wrap everything into a package with a **cryptographic signature** → creates a **TRO**



Example Trusted Research System

Scenarios TRACE Addresses



SCN1 — Journal reproducibility checks

Authors use SIVACOR to demonstrate push-button reproducibility before submission. Journals with and without data editors can rely on the TRACE-compliant package.

Estimated **20-40%** of AEA submissions are amenable to this.



SCN2 — Certification at the source

Long-running jobs on university clusters via SLURM can be configured with TRACE-compliant queues.

Universities become producers of TROs, providing enhanced credibility to affiliated researchers.

SCN3 — Restricted access data environments

RADCs have no vested interest in any particular paper — they satisfy the arms-length requirement.

Partners at **central banks** (and World Bank!) are already implementing TRACE-compliant capabilities.

SCN4 — Comparing reproducibility services

TRACE provides a systematic, standardized vocabulary — machine- and human-readable — to compare Codeocean, cascada, World Bank, and others.

SCN5 — Transient resources (5/5)

When data or services used in computation may no longer be available, TRACE provides a record of what existed at the time of execution.

For instance, **deprecated commercial LLM APIS!**

TROV, TRO, and TRS



The Three Core Concepts

| Acronym | Name | Description |
|-------------|-------------------------------|--|
| TROV | TRO Vocabulary | Controlled vocabulary for describing computational processes |
| TRO | Trusted Research Object | A replication package described using TROV |
| TRS | Trusted Research System | The system or process used to create a TRO |



What is a TRO?

A **TRO** is a replication package that contains:

- All **code** used in the analysis
- All **outputs** produced
- A **system description** (core features of the environment)
- Information about **how files changed** across arrangements
- **Organizational signatures** — cryptographically signed affirmations



What is a TRO?

Important limitation

The organization only asserts that *its process was followed* — it does **not** assert scientific correctness or endorse findings.

The Trust Chain

TROs must be issued by **organizations** (not individuals):

1. A journal accepts its first TRO → verifies the producing organization's process meets its standards
2. Journal publishes the TRO → adds another layer of credibility
3. Other journals and researchers can rely on that credibility **without further verification**

The Trust Chain

This mirrors:

- **PGP** “circle of trust”
- **Web PKI** — chain of certificate authorities down to root CA

Institutions emitting TROs have strong incentives to **maintain trust** — their credibility depends on it.



Example

Arrangements

Arrangements define how artifacts, typically files, are organized before and after each TRP. Artifacts are defined by their location and a checksum of their contents. Artifacts may be local or remote, defined by an URI. They may be included or excluded from the associated archive.

Step: Before executing workflow

| Artifact | SHA-256 | Status |
|-------------------|----------------------------------|--------|
| fred_apikey.txt | 067430f7c54898a6e8f9d094d53d3... | |
| requirements.txt | 79056c6a6c4bf5c13958768148cbc... | |
| run.sh | c38107430682ca6466f61421f40aa... | |
| results/sp500.png | 30a7f0712996ede7e0a3b7da59473... | |
| scripts/fred.py | 25f9160f2bd6fb2049775e75f710d... | |

Step: After executing workflow

| Artifact | SHA-256 | Status |
|-------------------|----------------------------------|-----------|
| data/sp500.csv | 6e69b0bb10277374181f0db85645b... | Created |
| data/sp500.json | 773bfb5e017777aac769cedaca... | Created |
| fred_apikey.txt | 067430f7c54898a6e8f9d094d53d3... | Unchanged |
| requirements.txt | 79056c6a6c4bf5c13958768148cbc... | Unchanged |
| results/sp500.png | 30a7f0712996ede7e0a3b7da59473... | Unchanged |
| run.sh | c38107430682ca6466f61421f40aa... | Unchanged |
| scripts/fred.py | 25f9160f2bd6fb2049775e75f710d... | Unchanged |



Arrangements

Arrangements define how artifacts, typically files, are organized before and after each TRP. Artifacts are defined by their location and a checksum of their contents. Artifacts may be local or remote, defined by an URI. They may be included or excluded from the associated archive.

Step: Before executing workflow

| Artifact | SHA-256 | Status |
|------------------|----------------------------------|--------|
| fred_apikey.txt | 067430f7c54898a6e8f9d094d53d3... | |
| requirements.txt | 79056c6a6c4bf5c13958768148cbc... | |
| run.sh | c38107430682ca6466f61421f40aa... | |
| scripts/fred.py | 25f9160f2bd6fb2049775e75f710d... | |

Step: After executing workflow

| Artifact | SHA-256 | Status |
|-------------------|----------------------------------|-----------|
| data/sp500.csv | 6e69b0bb10277374181f0db85645b... | Created |
| data/sp500.json | 773bfb5e017777aac769cedaca... | Created |
| fred_apikey.txt | 067430f7c54898a6e8f9d094d53d3... | Unchanged |
| requirements.txt | 79056c6a6c4bf5c13958768148cbc... | Unchanged |
| results/sp500.png | 30a7f0712996ede7e0a3b7da59473... | Created |
| run.sh | c38107430682ca6466f61421f40aa... | Unchanged |
| scripts/fred.py | 25f9160f2bd6fb2049775e75f710d... | Unchanged |



What TROs Do and Don't Do

TROs can:

- **Confirm** that artifacts not present initially were created by the provided code
- **Enable** principled compliance checking for data/code availability policies
- **Allow** human and automated comparison across services
- **Provide** file checksums that detect unauthorized modification

TROs cannot:

- **Confirm** completeness of the replication package
- **Confirm** correctness of the analysis
- **Replace** journal inspection for policy compliance
- **Guarantee** scientific validity

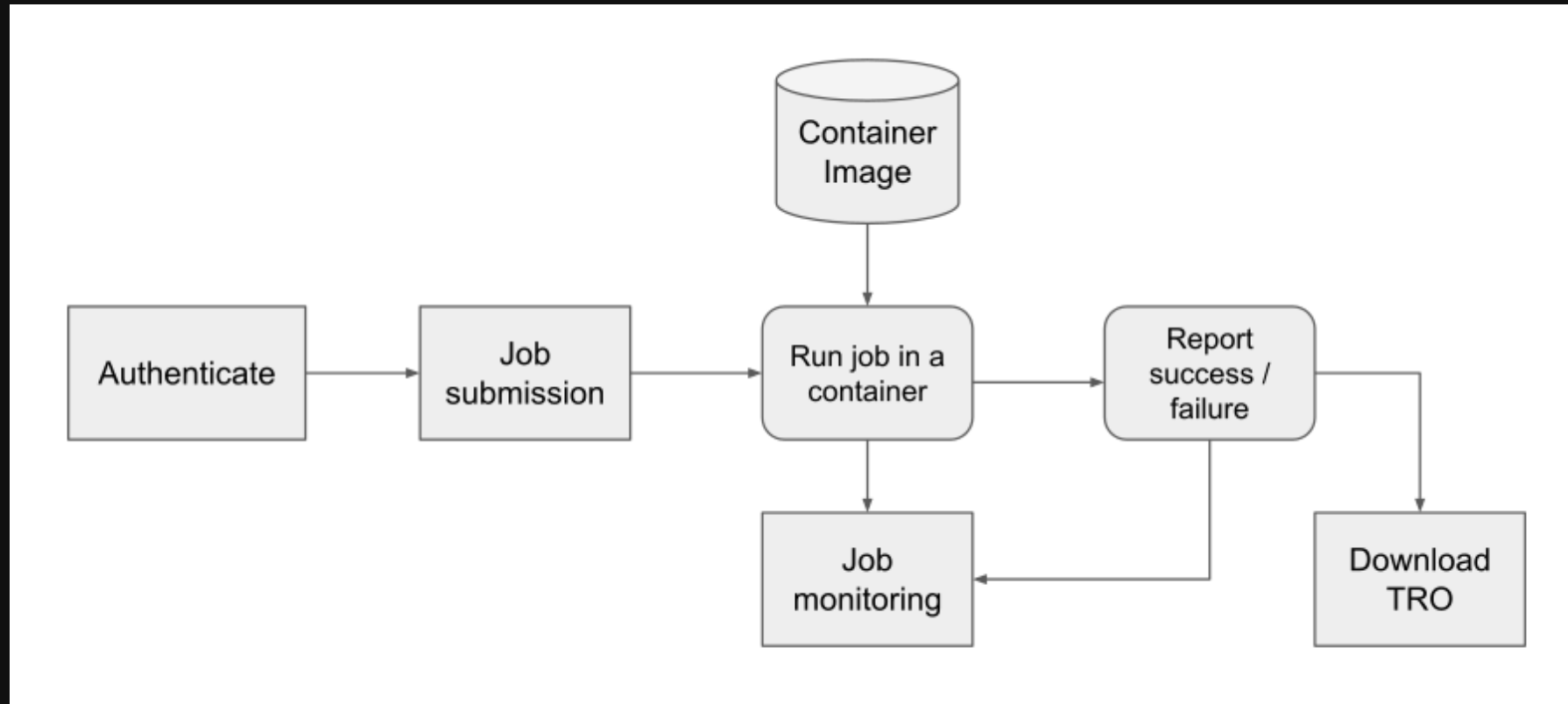


SIVACOR



The Reference Implementation

SIVACOR is the reference implementation of a TRACE-compliant Trusted Research System (TRS).





Scalable Infrastructure for Validation of Computational Social Science Research



Sign In

Choose your authentication provider to continue



Globus



ORCID





SIVACOR

Scalable Infrastructure for Validation of Computational Social Science Research

 Hello, Lars Vilhuber

 LOGOUT



SIVACOR Submission

Track and manage your last job



New Submission

Upload a single file (tarball or zipball), choose a Docker image and tag, then specify your main execution file to run your job.



File Upload

Select a file to upload for processing



Choose a file


or drag it here Maximum file size: 500MB







SIVACOR Submission

Track and manage your last job

Submission
sizzling-sensor SUCCESSJob ID
6925d19d164758ad437dbd27 **Job Completed Successfully!**
Results are ready for download and review[+ RUN NEW JOB](#) Job Logs

```
Preparing SIVACOR submission
New submission: 'sizzling-sensor' created.
Creating workspace from source folder.
Running TRO utilities in the workspace. (add_arrangement)
Executing workflow on workspace.
Running TRO utilities in the workspace. (add_arrangement)
Running TRO utilities in the workspace. (add_performance)
Running TRO utilities in the workspace. (sign)
Uploading executed replication package to Girder.
Submission job finalized successfully.
```

 Downloadable Files TRS Signature[DOWNLOAD](#) TRO Declaration[DOWNLOAD](#) Trusted
Timestamp[DOWNLOAD](#) Run output log[DOWNLOAD](#) Run error log[DOWNLOAD](#) Replicated
Package[DOWNLOAD](#)

Who runs SIVACOR?

A SIVACOR instance is run by a **trusted organization**

- journal
- university
- research institution hosting other researchers

What SIVACOR Provides

SIVACOR allows authors to:

1. Demonstrate **push-button reproducibility** in a transparent environment
2. Fix minor bugs **before submission** — without journal resources
3. Receive a **TRACE-compliant TRO** with organizational signature



What SIVACOR Provides

For journals:

- Packages arrive **already verified**
- TRACE metadata is machine-readable and comparable
- Works for journals **with and without** dedicated data editors



The Bigger Picture

*Universities, research data centers, and compute facilities become **producers of credibility** — not just reproducibility services.*

Development for TRACE / SIVACOR



TRACE

Implementations bring up issues, which need review and implementation.

| We need you!



Current known issues

These are in scope for inclusion in the next release of TRACE:






- **Explicit** rather than implicit disclaimers (issue 37 🕒)
- TRO software: what created the TRO? (issue 11 ✅, included in 0.1)
- Describe the **software environment** used to create an arrangement (issue 10 🕒)

SIVACOR

Allows to demonstrate features that are relevant for TRACE in an open testable environment



Recent additions

- Automatic file-level redaction (removing files) (frontend #27 , backend #81 )
- Multi-software workflow support (backend #14 )
- Environment variables (“secrets”) (frontend #29 , backend #82 )