Supporting Long-term Reproducible Software Execution

How can reproducibility be facilitated?

Reproducible execution: Executed by others
Using independent artifact ➔ same conclusion

Replicable execution: Executed by others
Using original artifact ➔ same result

Repeatable execution: Executed by author
Using own artifact ➔ same result

Where is the error?
In the original artifact? In one of its dependencies? In the reproduction?

Transparency:
Track, preserve, and inspect all software

Source code required to find and correct errors.
Making the software useful again.

All dependencies and respective source code accessible

What if the results are not reproduced?

Occam: repeatable, replicable, and reproducible execution

Software can be automatically built and run using the instructions on the metadata.

Manifests contain a digest of the software (main artifact and its dependencies).

Automatically generated!

Replicable Execution Environments are created on-demand (e.g., on Docker).

Results are packed with metadata that track their lineage and provenance.

Enforced by the system

Artifacts contain metadata with dependencies and build/run instructions

Packaged Artifact

Dependencies and artifacts must be encapsulated and preserved in Occam.

Repositories

Jupyter
Python
G++
GLIBC

的一切包等同 故事在 Occam!