Securing Large Language Model Software Supply Chains

Audris Mockus
The University of Tennessee, Knoxville, TN
Vilnius University, Vilnius
Meta, New York

Outline

- What are Software Supply Chains?
  - Dependencies, copying, knowledge transfer
  - Version history can help identify instances of copy more precisely
- How LLMs and SSCs are related?
  - LLMs copy/transform training data: another form of SSCs
- Does LLM output contain bugs?
  - LLMs suggest buggy code 2X more likely than fixed code (K. Jesse et all MSR'23)
- Does LLM training data contain vulnerabilities?
  - We found 250K vulnerable blobs in "the stack"
- What is World of Code and how it can help curate LLM training corpus?

SSC of the 1st kind

- Technical dependencies among projects with change effort as product flow
- Primary risks: unknown vulnerabilities, breaking changes, lack of maintenance, lack of popularity

Examples of SSC of the first kind

- Python: import re
- Java: import java.util.Collection;
- JavaScript: package.json

SSC of the 2nd kind

- Copying of the source code from project to project as product flow
- Primary risks: license compliance, unfixed vulnerabilities/bugs, missing updated functionality

Examples of SSC of the second kind

- Implementation of a complex algorithm
- Useful template
- Build configuration

SSC of the 3rd kind

- Knowledge (product) flow through code changes as developers learn from and impart their knowledge to the source code
- Primary risks: developers may leave, companies may discontinue support

Examples of SSC of the third kind

- Developers gaining skills with tools/packages/practices
- Developers spreading practices, e.g., testing frameworks

LLMs critically rely on curated data

- It is easy to forget that AI actually has no “Intelligence”
  - All intelligence comes from the training data
  - LLMs copy/translate/transform their training data
- What if there are problems with input data?
  - Use curated data
  - But LLM need lots of data
  - Auto-curate?
- All code LLM vendors face the same problem:
  - How to create a large corpus of good-quality training data?
- Source code version history can help auto-curate
  - Have not seen any vendor exploit that

LLMs are SSCs of type IV

- Source code to LLMs
  - LLMs are trained on extremely large curated corpus
  - LLMs are fine-tuned on developer actons (which suggestions are “accepted”)
  - Both sources carry problematic input
- LLMs to source code
  - LLMs generate code that ends up in public repositories
    - This code may then be depended upon (SSC I), copied (SSC II), learned from (SSC III), or fed back to LLM in the next training cycle.
  - LLMs generate suggestinons and explanations to developers
    - This may directly affect how they understand and write code even if the generated suggestions are not accepted
- LLMs make code copying harder to detect

SSCs Measurement: Key Needs

- Completeness: the entirety of OSS
- Autocuration: address data quality at scale
- Cross-referencing: to make analysis run in minutes not months

Why SSCs?

- OSS resulted in massive code reuse
  - everything is built on something
  - controlled by someone else
- Project-focused Software Engineering => ecosystem-aware SE
  - both upstream and downstream
- Project-focused measurement/practices no longer work
- Need for curated data resource
  - to measure across projects (e.g., what happens downstream)
  - to understand collaborations (e.g., what a developer does)
  - to study/do ecosystem aware SE
- Inspired by a company-wide measurement within AT&T/Lucent/Avaya [10,11]

What is World of Code (WoC)?

- Complete*
  - Captures data from all public git repos (approx 200 forges)
  - 20B blobs, 16B trees, 4B commits, 210M repos (130M de-forked), 76M author IDs (44M aliased)
- Current
  - As of June, 2023
- Curated
  - e.g., author aliasing, deforking, bot identification, core teams, communities
- Cross-referenced
  - first-class entities mapped to other first-class entities
  - authors, projects, commits, blobs, ctagged blobs, time, imports (17 languages)

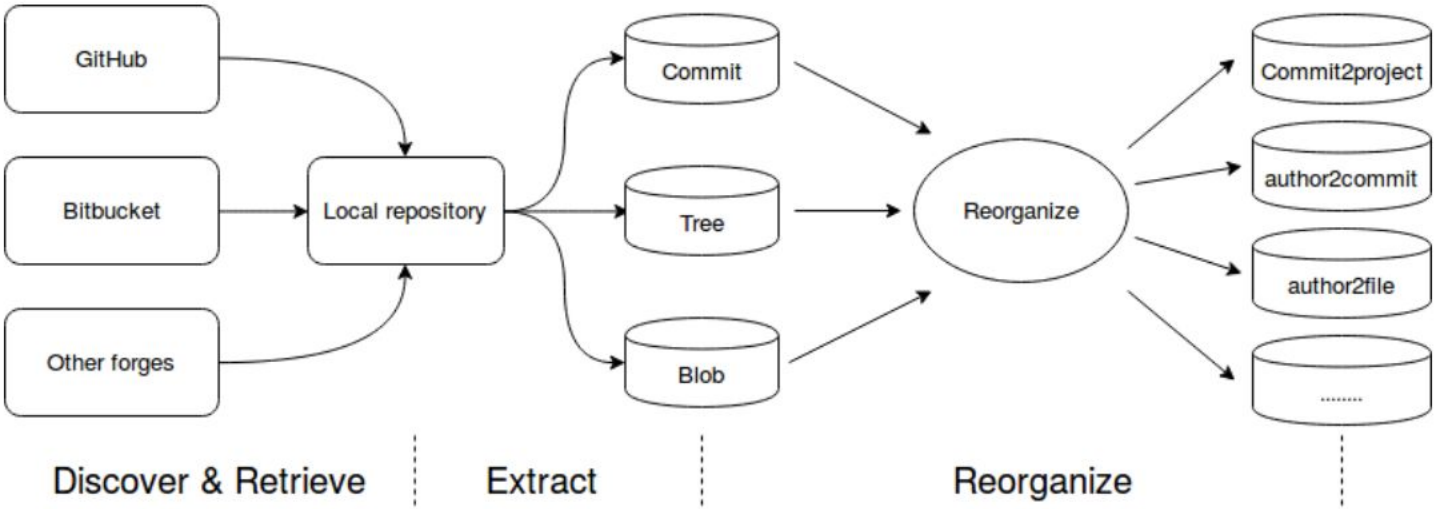
*WoC version V numbers are at https://bitbucket.org/swsc/overview/

*Terms of use: https://github.com/woc-hack/tutorial/blob/master/LICENSE (just over 200 international users)

github.com/woc-hack/tutorial, worldofcode.org

World of Code (WoC) Overview

- Data gathered and cleaned from multiple forges



World of Code (WoC) Overview

- Provides OSS-wide relationships

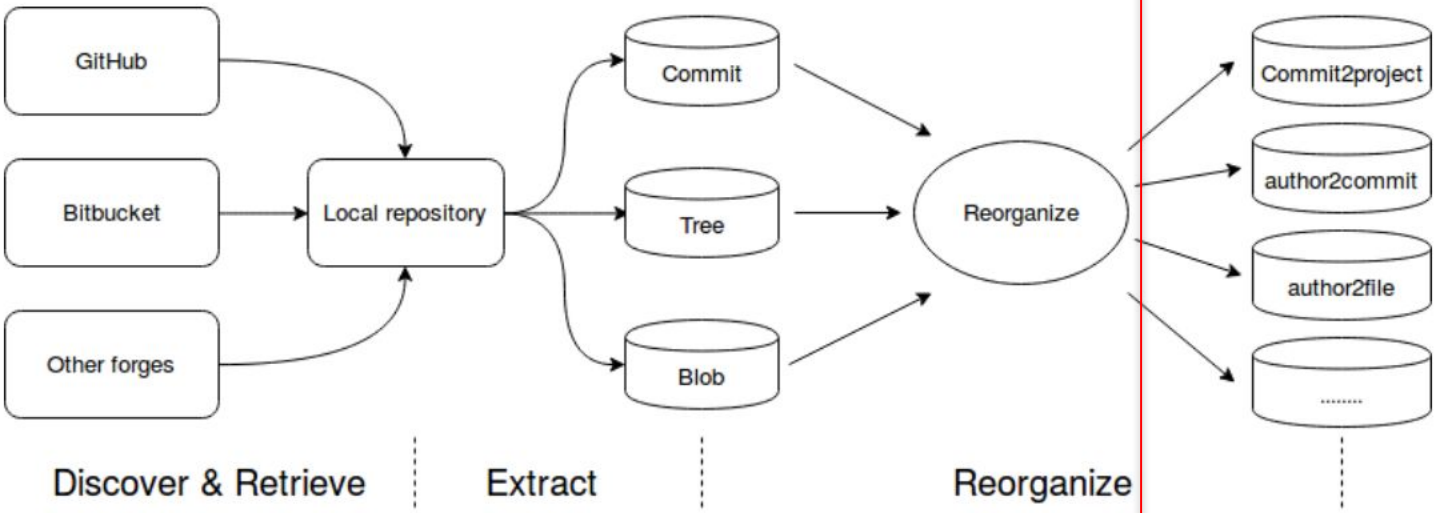


Fig. 1. Overarching data flow

Detect vulnerable code in LLM training data

- Open source “The Stack”
  - Raw collection (6TB) includes binaries, images, fork content
  - “Deduplicated” collection recommended for training LLMs
- Start with 3,615 CVEs from CVEFixes dataset:
  - The vulnerability is in one source code file
  - We know the vulnerable file and the fixing commit
  - Using the git history or b2ob/ob2b, we create 2 lists:
    - The vulnerable blob and all prior blobs (which are likely also vulnerable)
    - The fixed blob and all newer blobs (which are likely also fixed)
  - Compare above lists with bigcode-project.org’s the-stack and the-stack-dedup

WoC Provenance

Provenance
Code Copy

Provenance
Version History

WoC Provenance

Provenance
Code Copy

blob (git version)

contains: content of a file

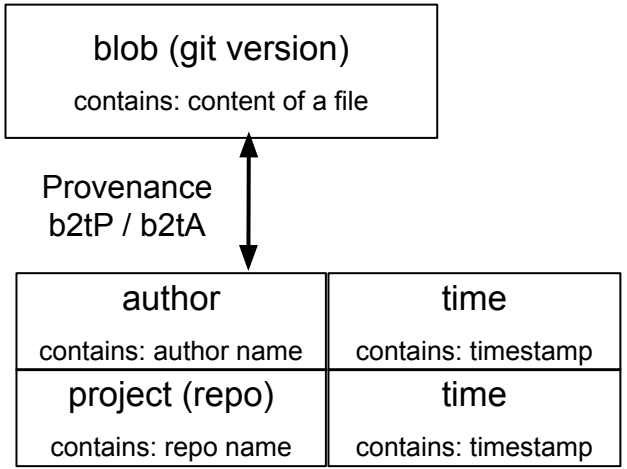
Provenance
Version History

blob (git version)

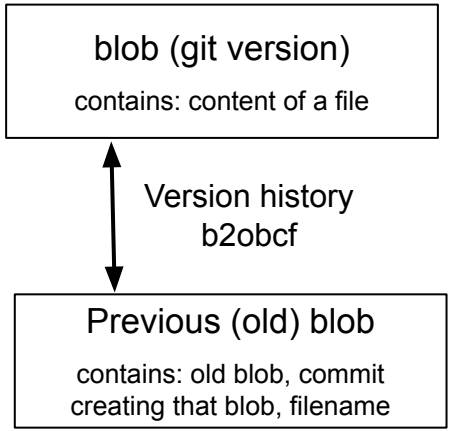
contains: content of a file

WoC Provenance

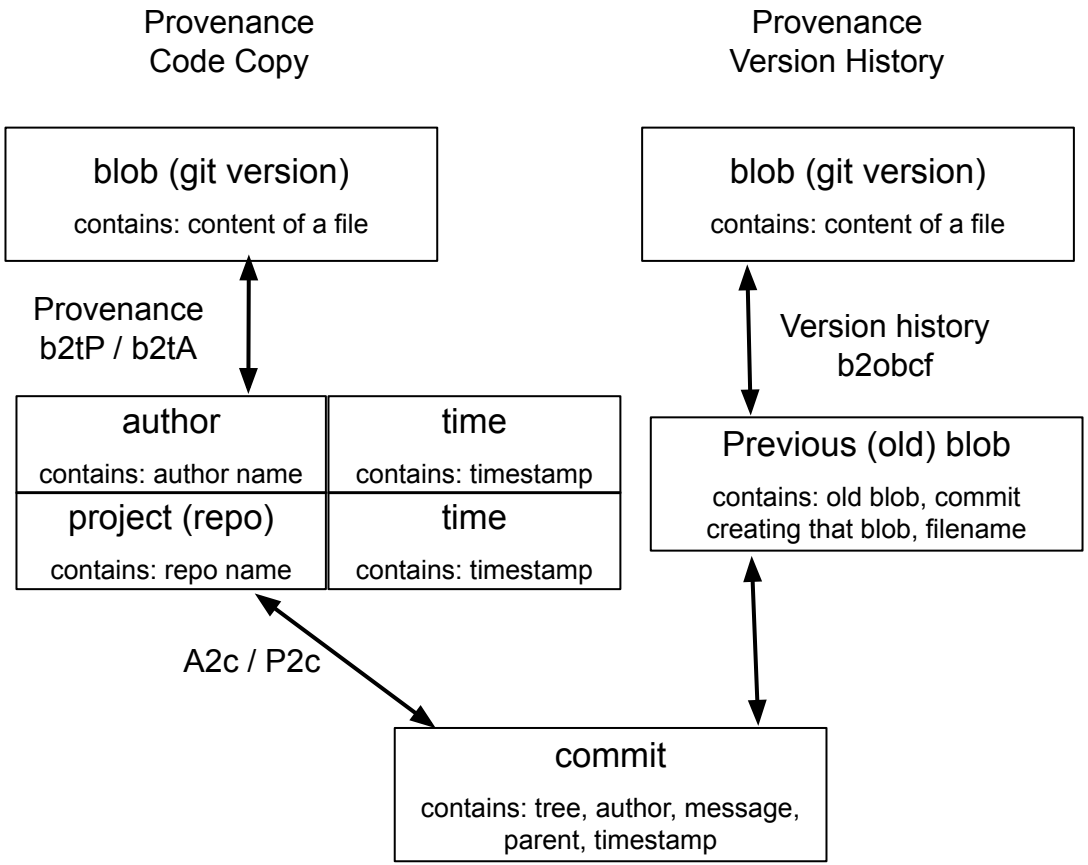
Provenance
Code Copy



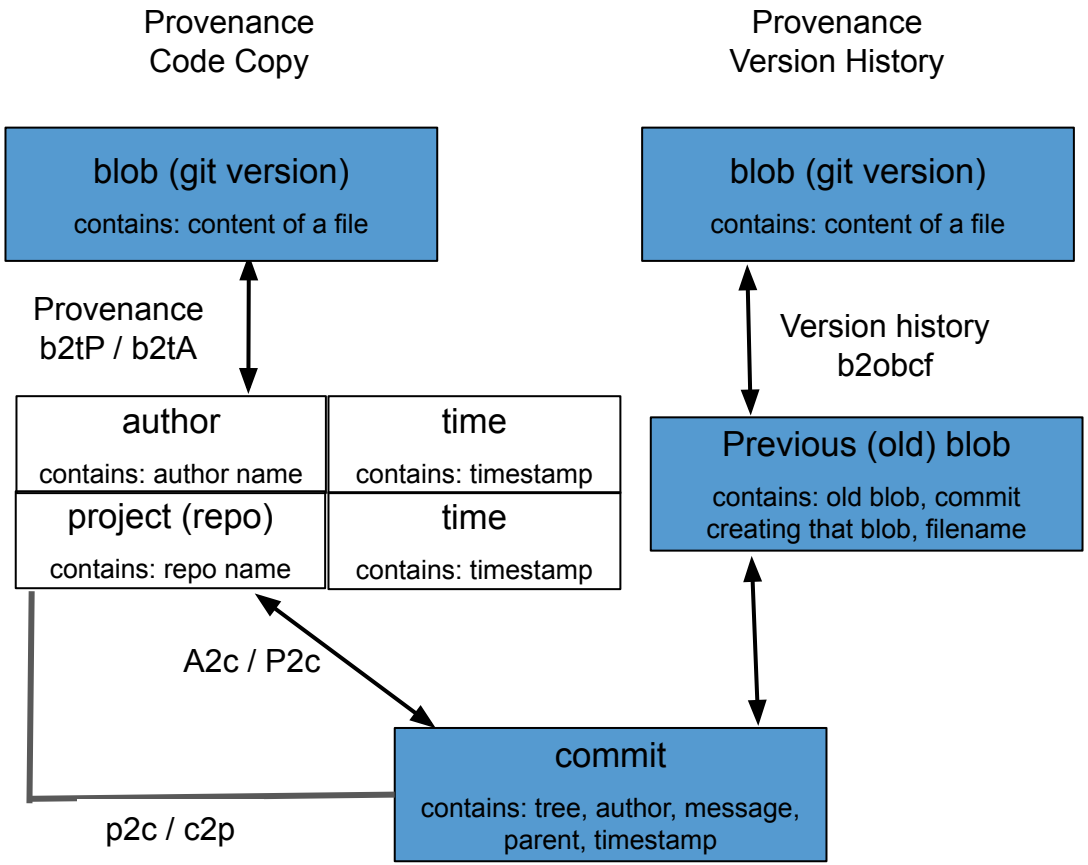
Provenance
Version History



WoC Provenance



WoC Provenance

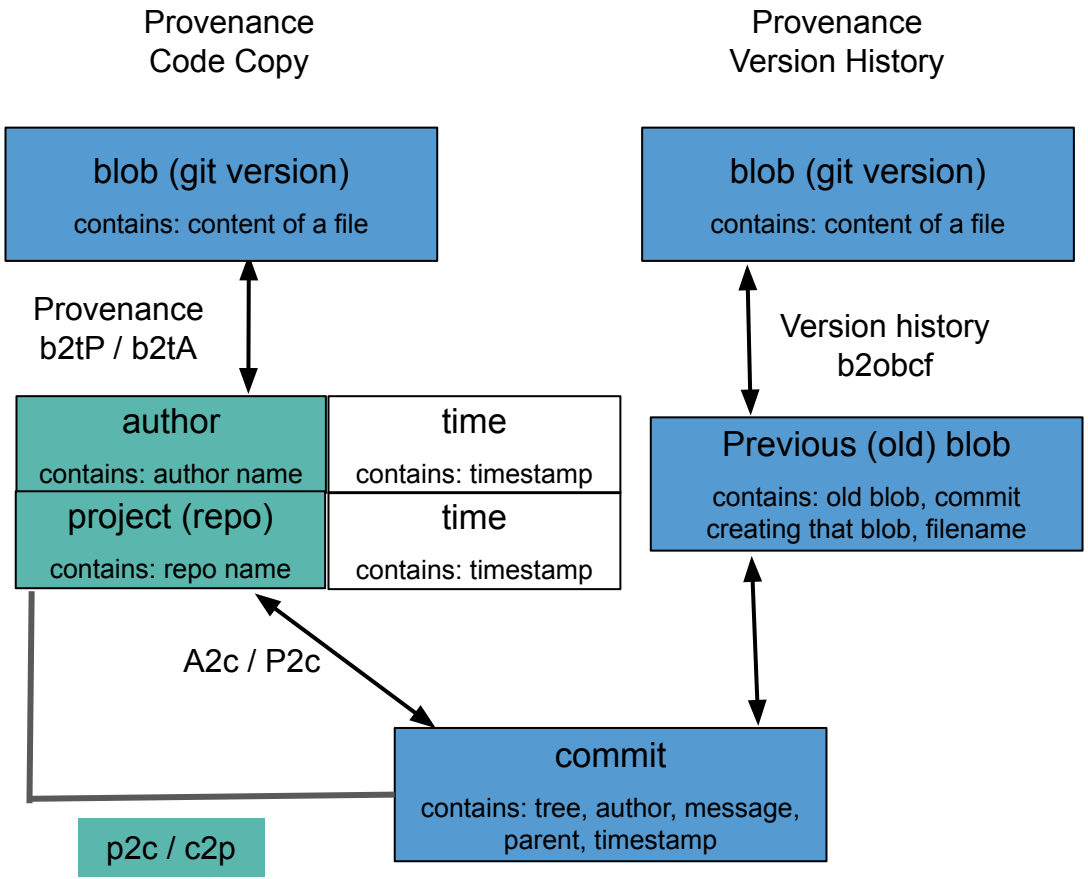


WoC layers

Layer 1: raw content of commits, trees or blobs

(access trough showCnt[ID])

WoC Provenance



WoC layers

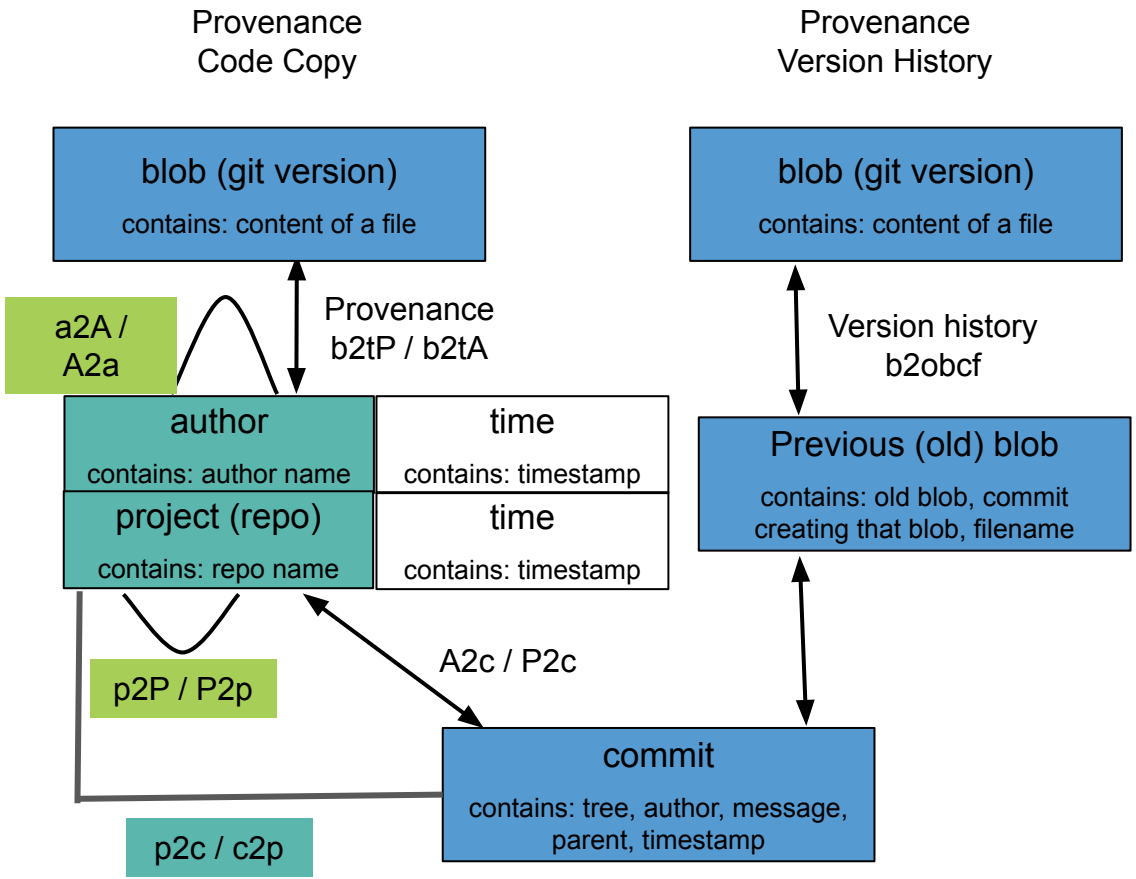
Layer 1: raw content of commits, trees or blobs

(access trough showCnt[ID])

Layer 2: repo names, author IDs, file names

(access through maps such as a2c, c2b, etc.)

WoC Provenance



WoC layers

Layer 1: raw content of commits, trees or blobs

(access trough showCnt[ID])

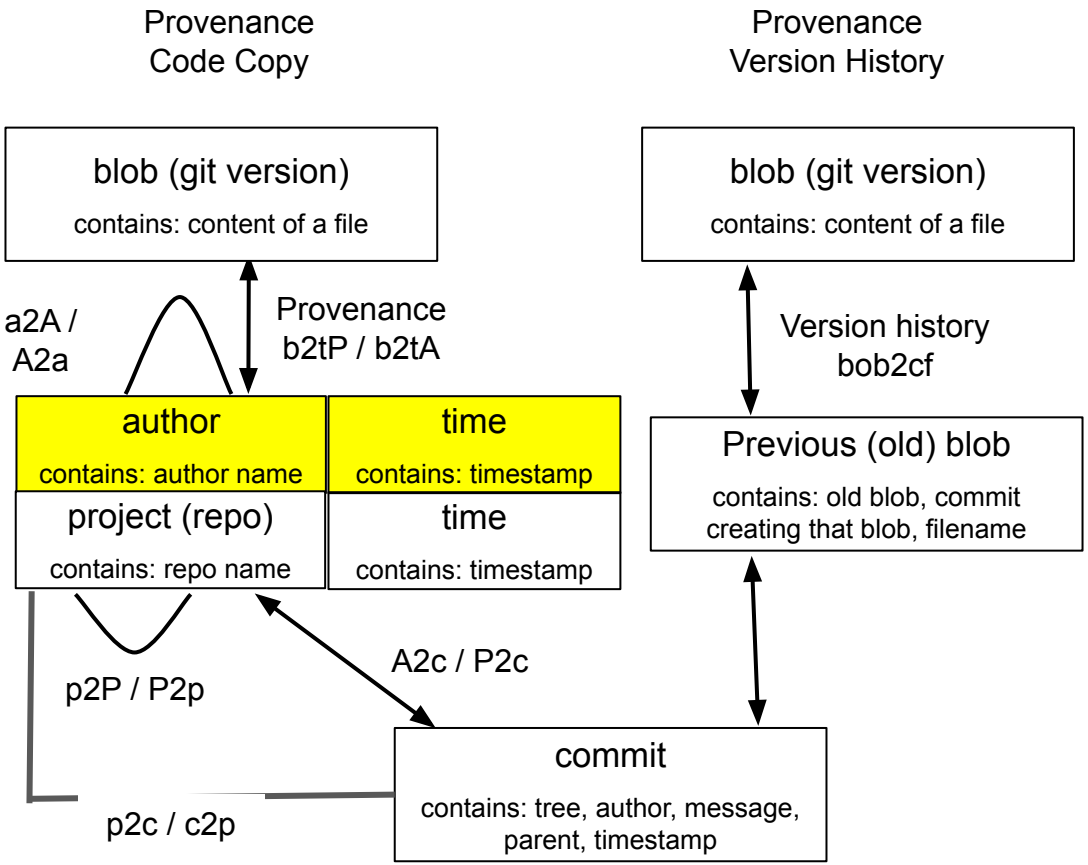
Layer 2: repo names, author IDs, file names

(access through maps such as a2c, c2b, etc.)

Layer 3: author aliasing, project deforking

(access through capitalized maps: A for authors (a2A, A2a), P for projects (p2P, P2p))

WoC Provenance



Example

Q1: Who/What project created this code?

Shell code:

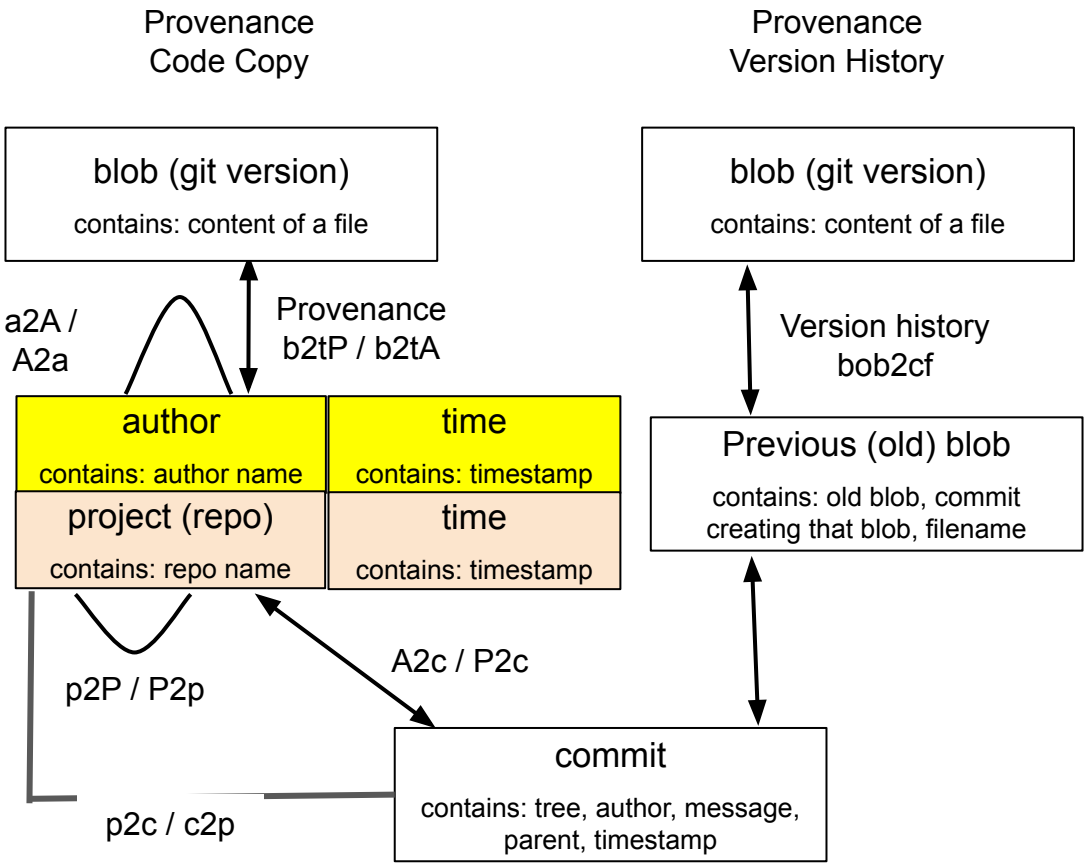
```
echo 99600f  | getValues b2tAc
echo 99600f  | getValues b2tP
```

Result:

- List of blob;time;Repository/Author;commit:
  - 99600f;1620159367;Sanjoy Das <sanjoy@debian>;6ec5f4a03e1181fbfcfdffa10a82cd52d9724ae9
  - 99600f;1620159367;tensorflow_tensorflow

…

WoC Provenance



Example

Q1: Who/What project created this code?

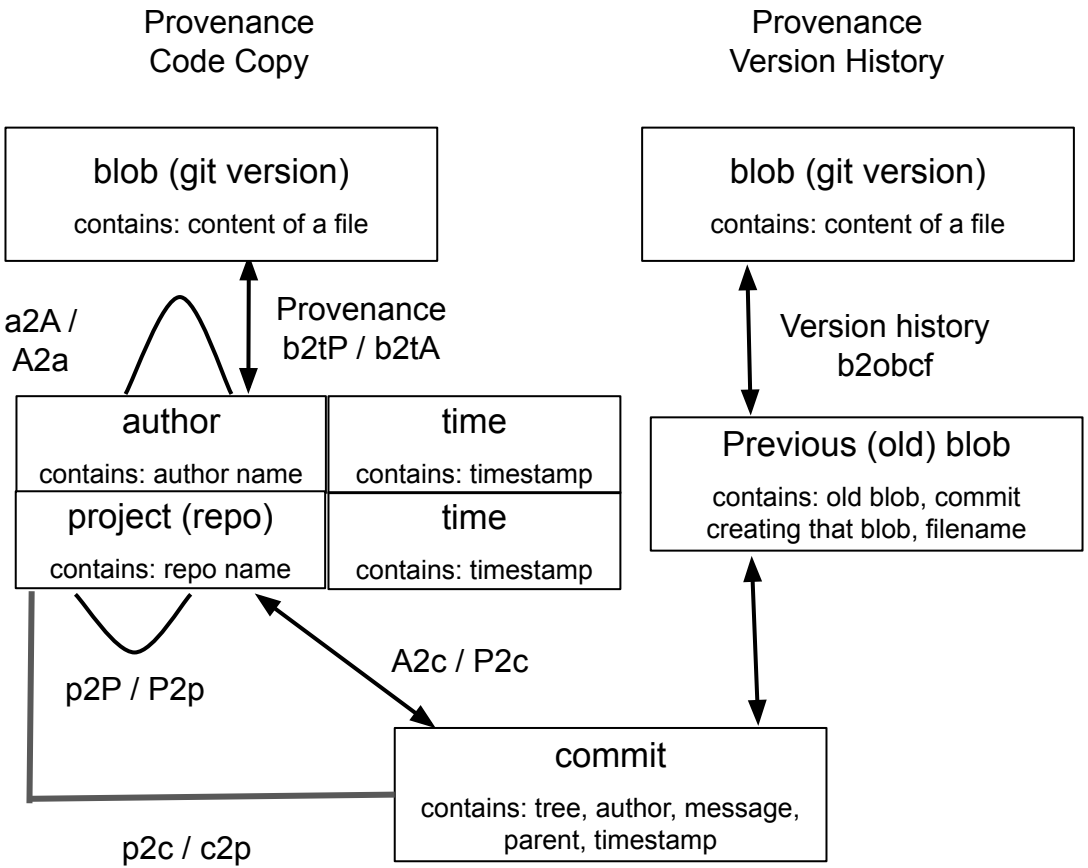
Shell code:

```
echo 99600f  | getValues b2tAc
echo 99600f  | getValues b2tP
```

Result:

- List of blob;time;Repository/Author;commit:
  - 99600f;1620159367;Sanjoy Das <sanjoy@debian>;6ec5f4a03e1181fbfcfdffa10a82cd52d9724ae9
  - 99600f;1620159367;tensorflow_tensorflow

WoC Provenance



Example

Q2: What commit(s) created this code?
Q3: What are previous version of this code?
Q4: What are next version of this code?

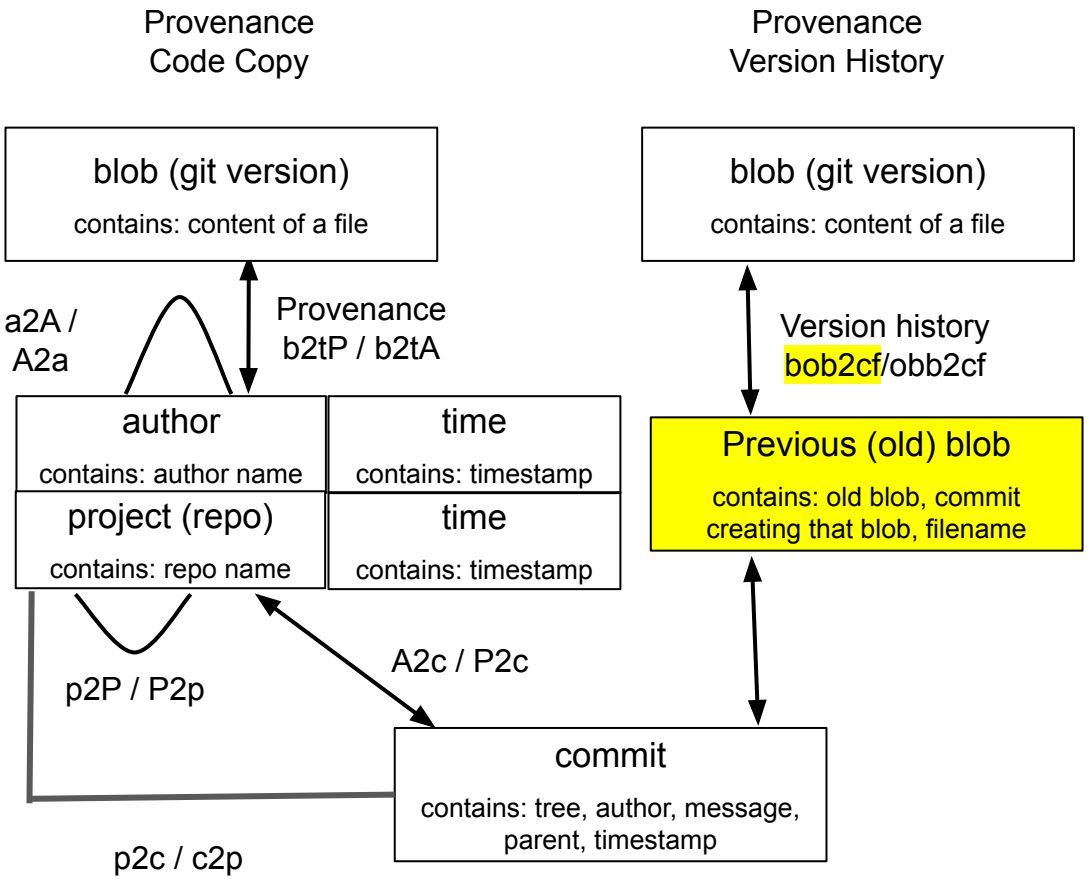
Shell code:

```
echo 99600f  | getValues bb2cf
echo 99600f  | getValues obb2cf
```

Result:

- List of blob;old blob;commit;filename:
99600f;de97e63a49cd2982e6e0f391146be4c35ae726b1;6ec5f4a03e1181fbfcfdffa10a82cd52d9724ae9;tensorflow/core/kernels/sparse_fill_empty_rows_op.cc
- List of old blob;new blob;commit;filename:
99600f;c1ed592b965e247d6105e416c0e74d888d2993f8;faa76f39014ed3b5e2c158593b1335522e573c7f;tensorflow/core/kernels/sparse_fill_empty_rows_op.cc

WoC Provenance



Example

Q2: What commit(s) created this code?
Q3: What are previous version of this code?
Q4: What are next version of this code?

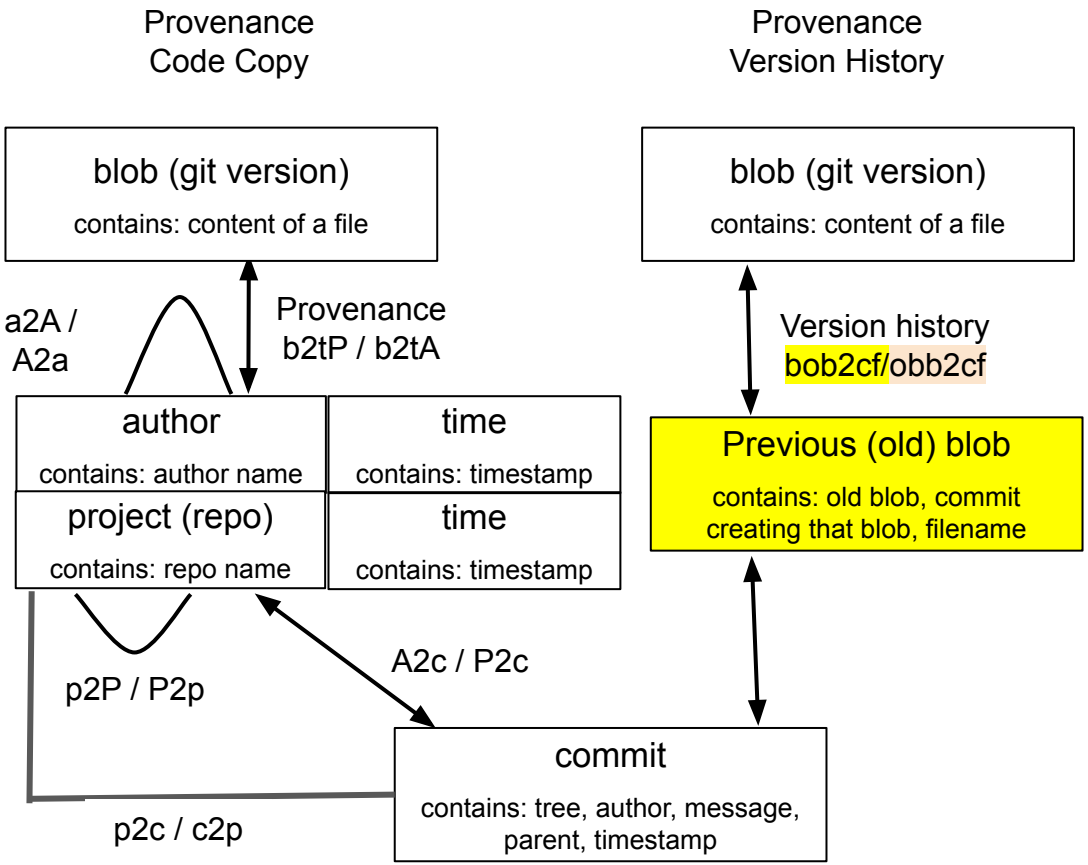
Shell code:

```
echo 99600f  | getValues bb2cf
echo 99600f  | getValues obb2cf
```

Result:

- List of blob;old blob;commit;filename:
99600f;de97e63a49cd2982e6e0f391146be4c35ae726b1;6ec5f4a03e1181fbfcfdffa10a82cd52d9724ae9;tensorflow/core/kernels/sparse_fill_empty_rows_op.cc
- List of old blob;new blob;commit;filename:
99600f;c1ed592b965e247d6105e416c0e74d888d2993f8;faa76f39014ed3b5e2c158593b1335522e573c7f;tensorflow/core/kernels/sparse_fill_empty_rows_op.cc

WoC Provenance



Example

Q2: What commit(s) created this code?
Q3: What are previous version of this code?
Q4: What are next version of this code?

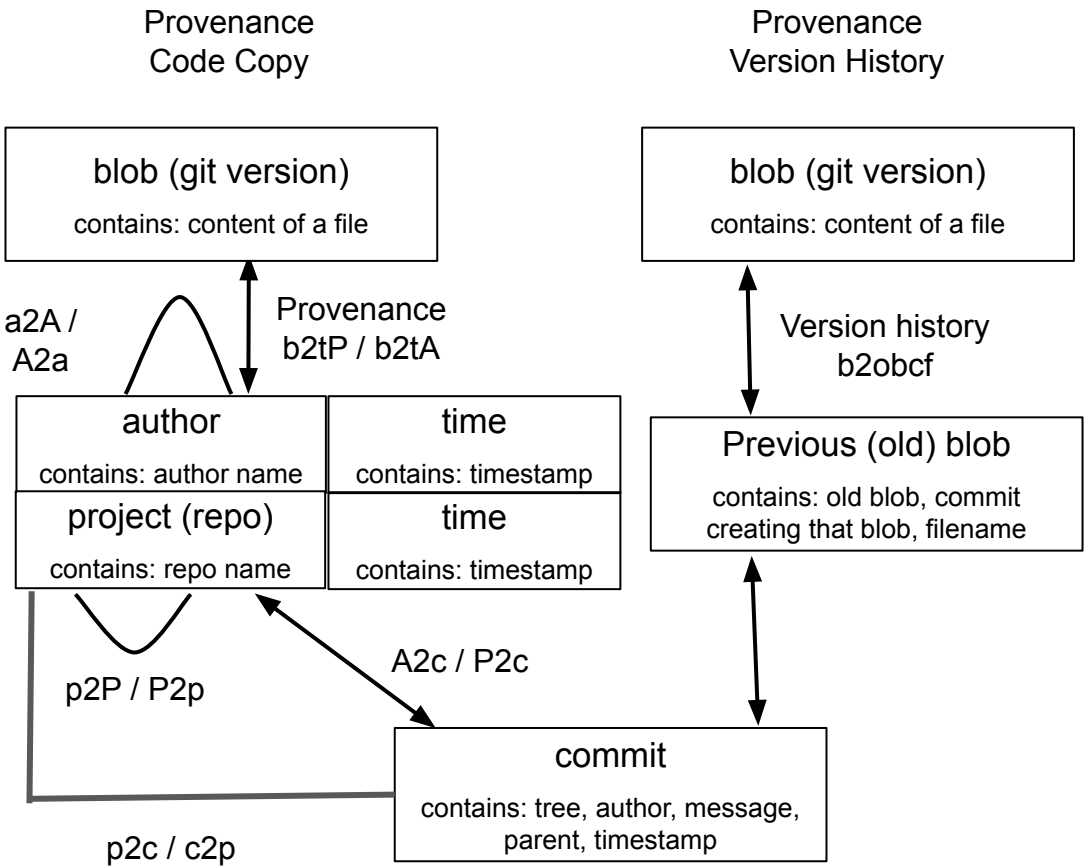
Shell code:

```
echo 99600f  | getValues bb2cf
echo 99600f  | getValues obb2cf
```

Result:

- List of blob;old blob;commit;filename:
99600f;de97e63a49cd2982e6e0f391146be4c35ae726b1;6ec5f4a03e1181fbfcfdffa10a82cd52d9724ae9;tensorflow/core/kernels/sparse_fill_empty_rows_op.cc
- List of old blob;new blob;commit;filename:
99600f;c1ed592b965e247d6105e416c0e74d888d2993f8;faa76f39014ed3b5e2c158593b1335522e573c7f;tensorflow/core/kernels/sparse_fill_empty_rows_op.cc

WoC Provenance



Example

Q5: Which project(s) contain this commit?

Shell code:

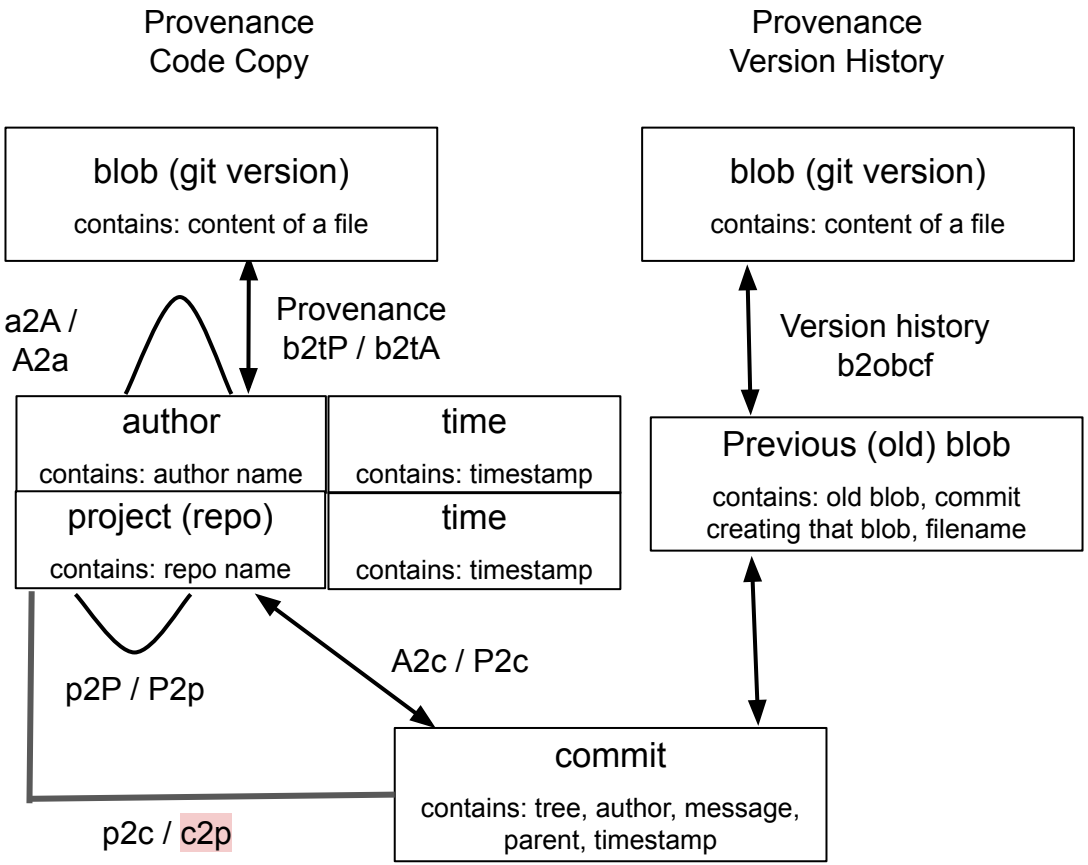
echo 724ae9 | getValues c2p

Result:

- List of commit;project

724ae9;47-studio-org_tensorflow
724ae9;8bitmp3_tensorflow
724ae9;Apidwalin_tensorflow-master
724ae9;OlexanderMiroshnychenko_tf_patch_test
724ae9;aakash30jan_tensorflow
724ae9;aaudiber_tensorflow
724ae9;abhilash1910_tensorflow
724ae9;adaalarm_tensorflow
724ae9;adamhillier_tensorflow
724ae9;adhadse_tensorflow
….

WoC Provenance



Example

Q5: Which project(s) contain this commit?

Shell code:

```
echo 724ae9  | getValues c2p
```

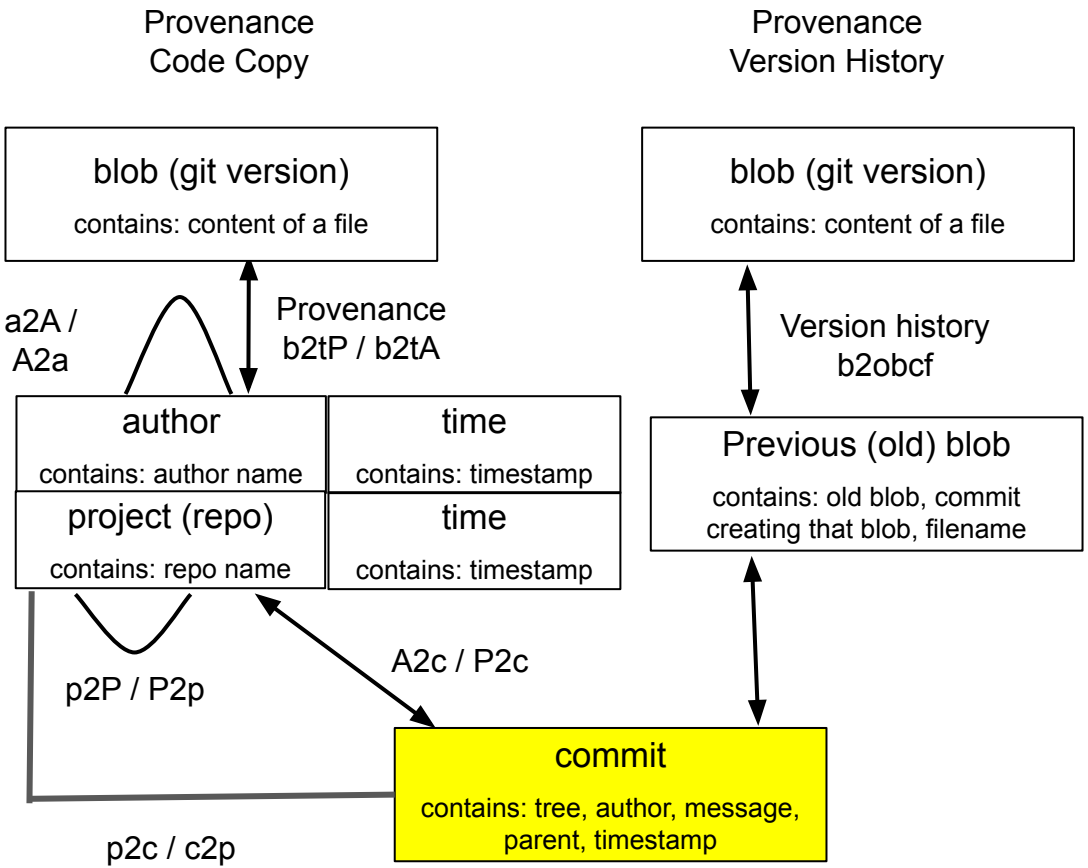
Result:

- List of commit;project

724ae9;47-studio-org_tensorflow
724ae9;8bitmp3_tensorflow
724ae9;Apidwalin_tensorflow-master
724ae9;OlexanderMiroshnychenko_tf_patch_test
724ae9;aakash30jan_tensorflow
724ae9;aaudiber_tensorflow
724ae9;abhilash1910_tensorflow
724ae9;adaalarm_tensorflow
724ae9;adamhillier_tensorflow
724ae9;adhadse_tensorflow

….

WoC Provenance



Example

Q5: What is the content of this commit?

Shell code:

echo 724ae9 | showCnt commit

Result:

- List of commit;parent;tree;author;commiter;time;comment

724ae9;
9454fd13698405f86d7aa84485a99ad3e988e5fe;
55338bb43c76edad2557be1cd62dc315c4106314;
Sanjoy Das <sanjoy@google.com>;
Tenso rFlower Gardener <gardener@tensorflow.org>;
1620159367;
Disable SparseFillEmptyRows[Grad] on GPU\nIt breaks an internal workload with the error message "segment ids are not increasing", which probably means that the output indices are not sorted in some cases.\nPiperOrigin-RevId: 371980850__NEWLINE__Change-Id:

Results (Security Issues in “The Stack”)

- Out of 267,798 potentially vulnerable blobs (blobs prior to the fixing commit).
  - 11,266 are in bigcode/the-stack dataset.
  - 1,482 are in bigcode/the-stack-dedup
- Out of 3,293 known vulnerable blobs (looking just at the one blob before the fixing commit):
  - 119 are in thebigcode/the-stack-dedup dataset

Open Challenges in Securing LLM SSCs

Identify/fix problems in the training data

Identify/fix problems at the time of generation via fine-tuning

Identify LLM output in public repositories

- **Statistical and LLM-based approaches**
  - **Generated files**
  - **Code completions**

Trace LLM output to most likely LLM inputs

- **General AI traceability**
- **Use WoC + LLM training/fine-tuning to capture these relationships**

Summary

- LLM output is only as good as LLM training input
  - May get worse over time via user feedback
- Curation is one way to address the problem
  - WoC may be a good resource to
    - Identify problems in training data
    - Identify problems in generated data
    - Recognize the extent of generated code

Tutorial: github.com/woc-hack/tutorial

Hybrid hackathon Nov 17-19

https://github.com/woc-hack/2023Hackathon