Measuring and Curating Code LLM Supply Chains

Audris Mockus
The University of Tennessee, Knoxville, TN

AIWare Leadership Bootcamp, Toronto, Nov 5

Outline

- What are Software Supply Chains (SSC)?
  - I- Dependencies, II - Copying, III - Knowledge transfer
  - WoC: Measuring SSCs
- LLMs - as type IV SSC
  - LLMs copy/transform training data
- Does LLM output contain bugs?
  - LLMs suggest buggy code 2X more likely than fixed code
- Does LLM training data contain bugs/vulnerabilities?
  - We found 250K vulnerable blobs in “the stack”
  - How to fix these buggy training data?
- Recognizing generated code
- Approaches to reduce data leakage
- Supply chain of LLMs: model reuse, data reuse, parameter reuse, software reuse

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

What is World of Code (WoC)?

- Complete
  - Captures data from all public git repos (approx 200 forges)
  - 26B blobs, 20B trees, 5B commits, 210M repos (130M de-forked), 76M author IDs (44M aliased)
- Current: As of June, 2024; next update Nov, 2024
- Curated
  - e.g., author aliasing, repo deforking, bot identification, core teams, communities
- Cross-referenced: full SSC
  - first-class entities mapped to other first-class entities
  - APIs (17 languages), authors, projects, commits, blobs, time

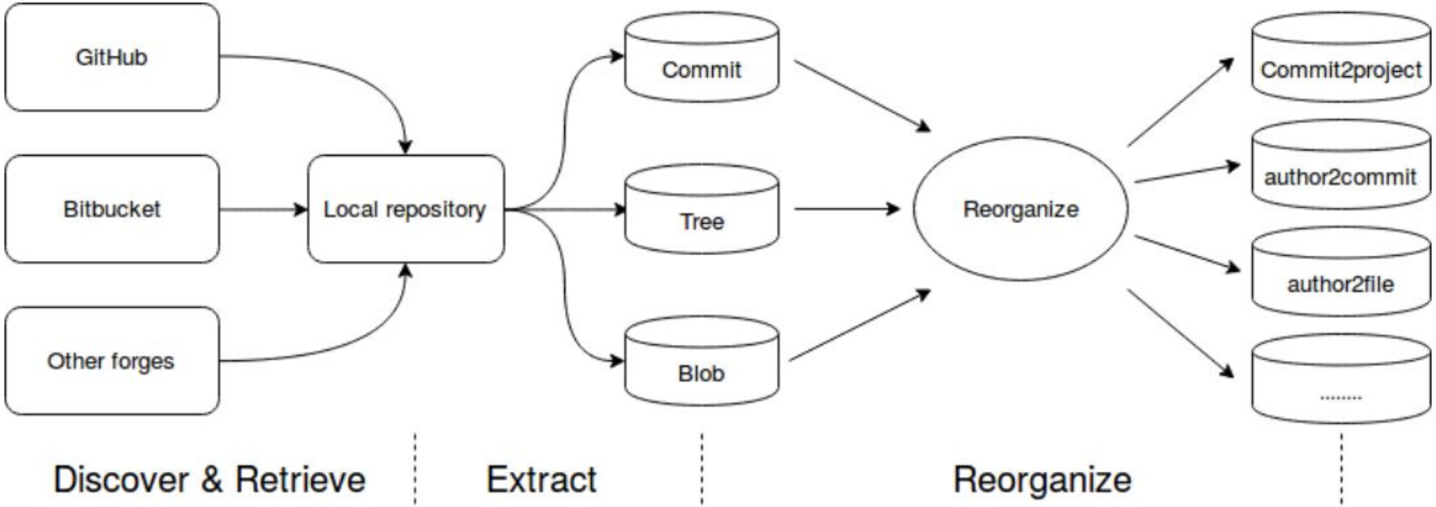
*WoC version V3 numbers are at https://bitbucket.org/swsc/overview/

*Terms of use: https://github.com/woc-hack/tutorial/blob/master/LICENSE (just over 300 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

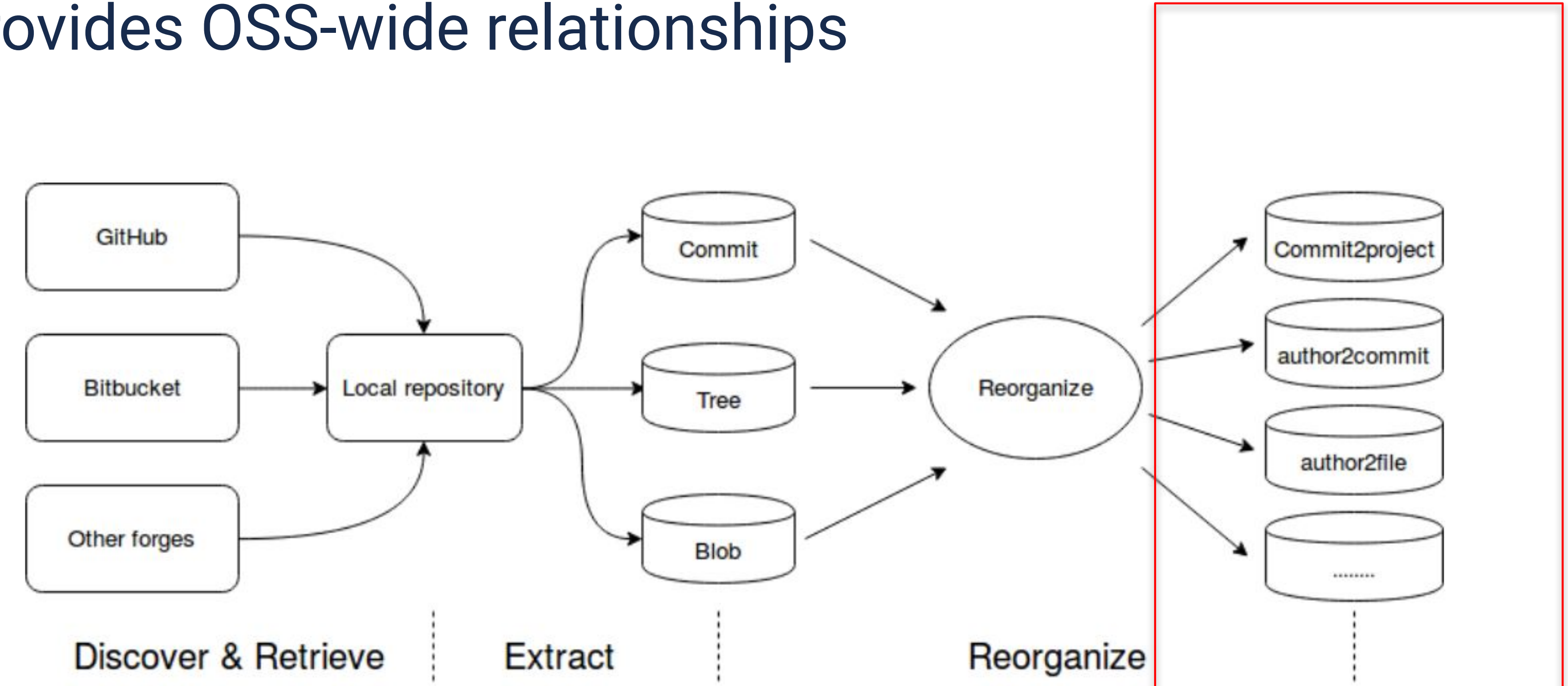


Fig. 1. Overarching data flow

WoC: SSC of the 1st kind

W<🌍/>C

- Technical dependencies among projects with change effort as product flow
- WoC indexes
  - Import statement to blob, commit, project Author
  - Export indicators to blob, commit, project Author
  - No need for package managers

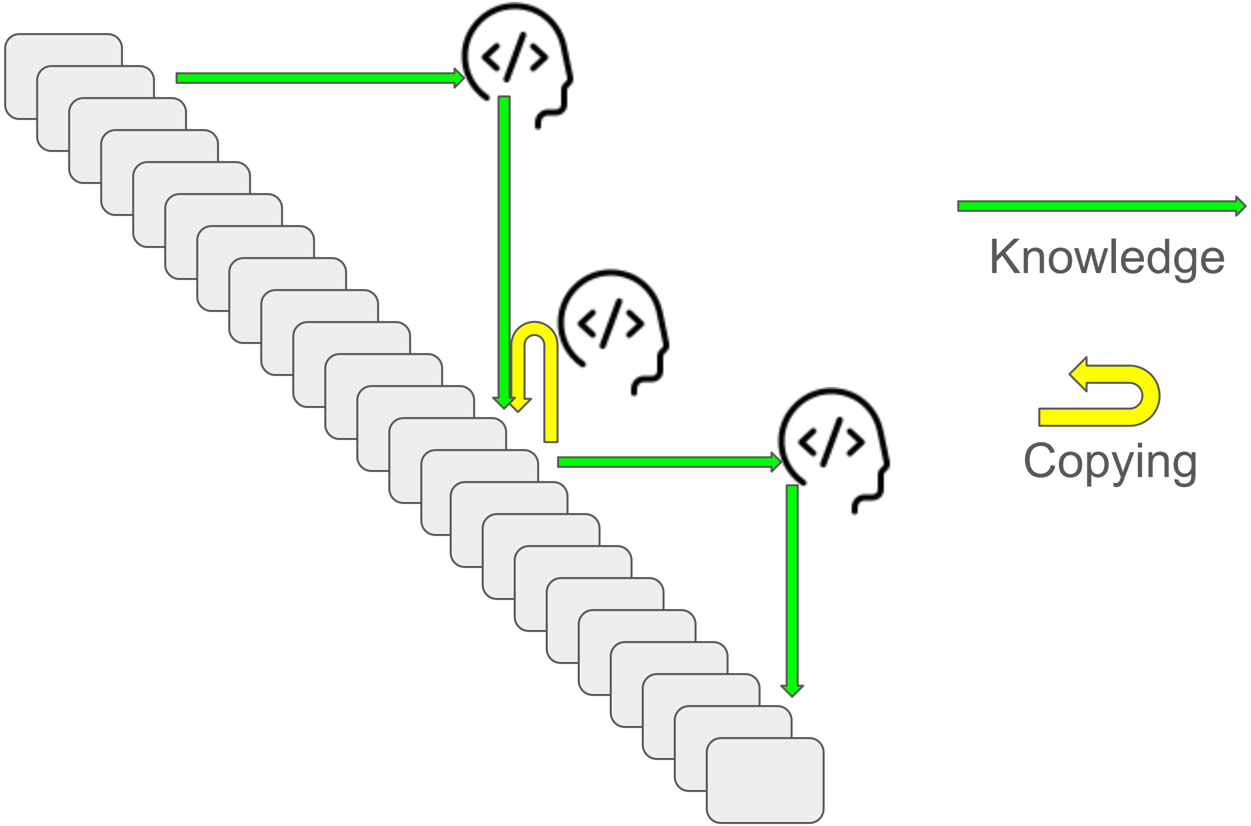
SSC of the 2nd kind

- Copying of the source code from project to project as product flow
- WoC indexes
  - Originating project to
    - Target project
    - Time of first copy
- Prevalence (arXiv:2409.04830)
  - 23B instance of reuse (version U had 16B)
  - 31M Originating projects
  - 86M Destination projects

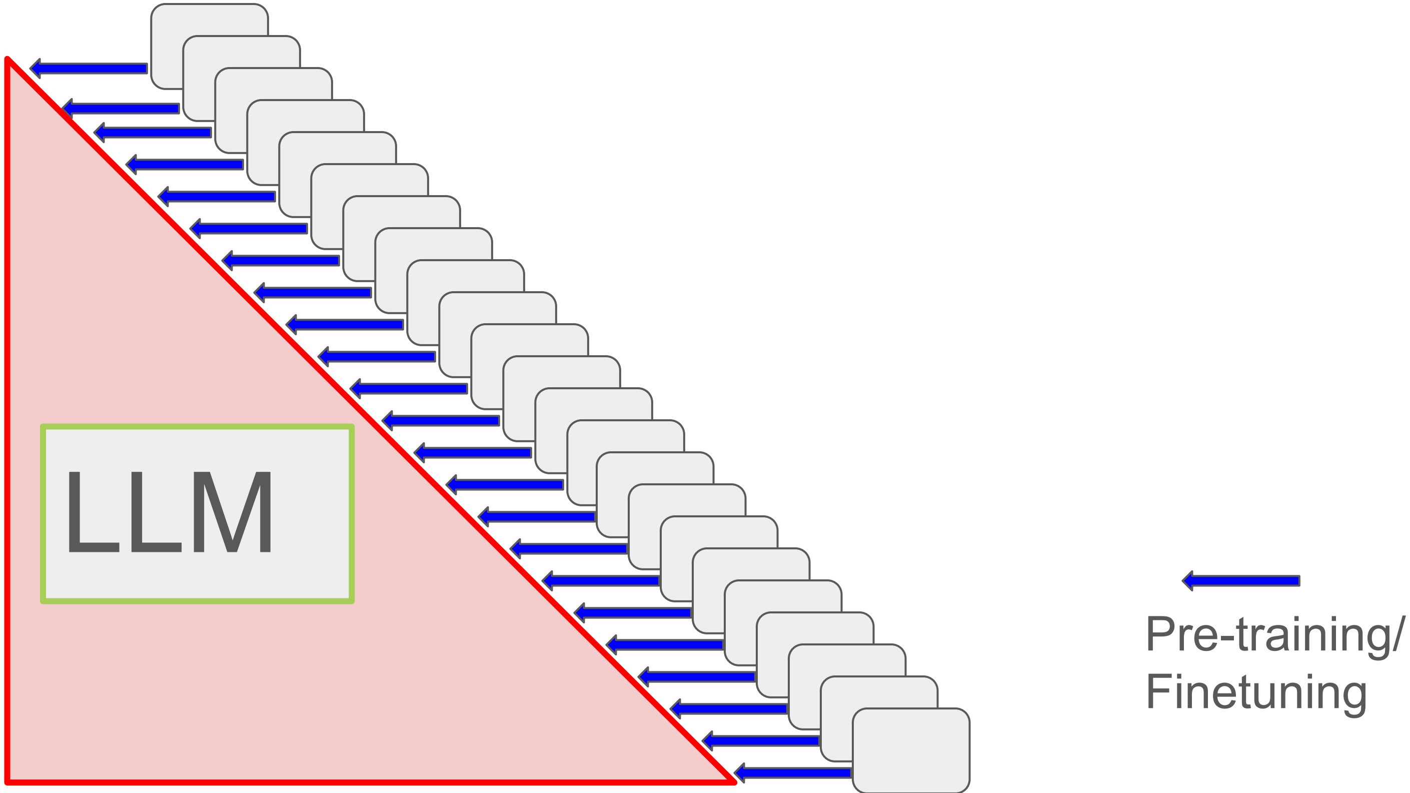
SSC of the 3rd kind

- Knowledge (product) flow through code changes as developers learn from and impart their knowledge to the source code
- WoC index
  - Commit authors (aliased) to
    - APIs
    - Commits

Copy/Knowledge SSC



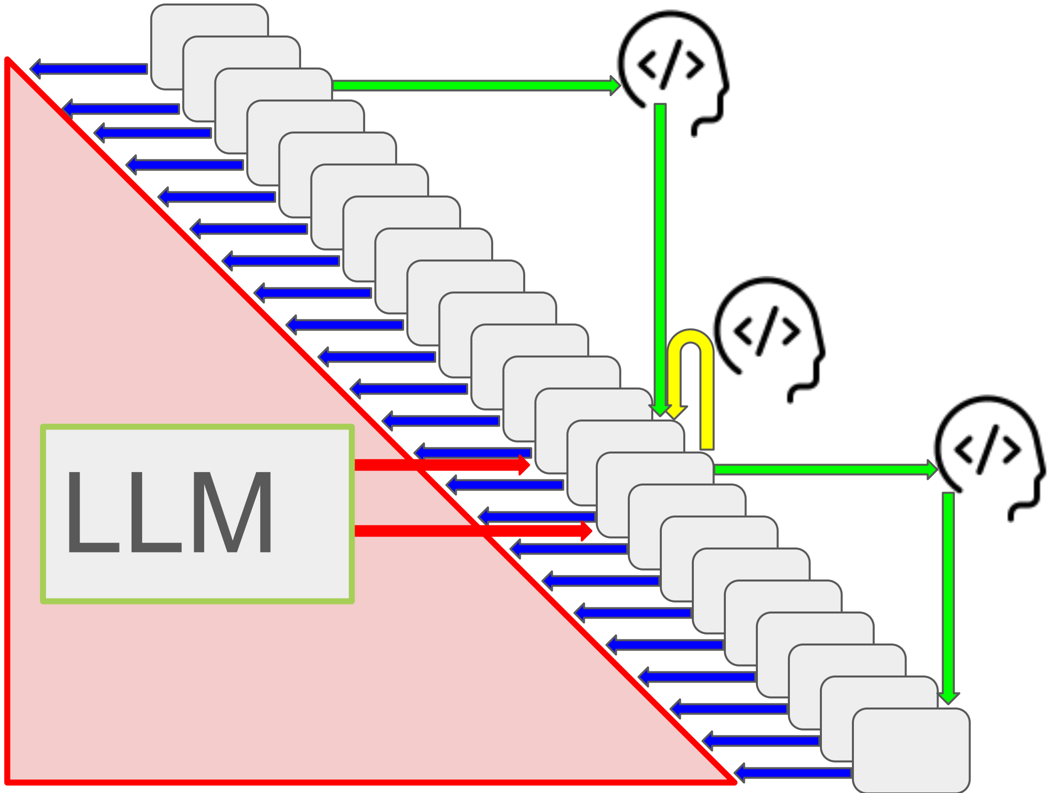
LLMs: pre-trained on *ALL* code



LLM

Pre-training/
Finetuning

LLMs: Copy/Knowledge SSC on Steroids



→ Knowledge

↩ Copying

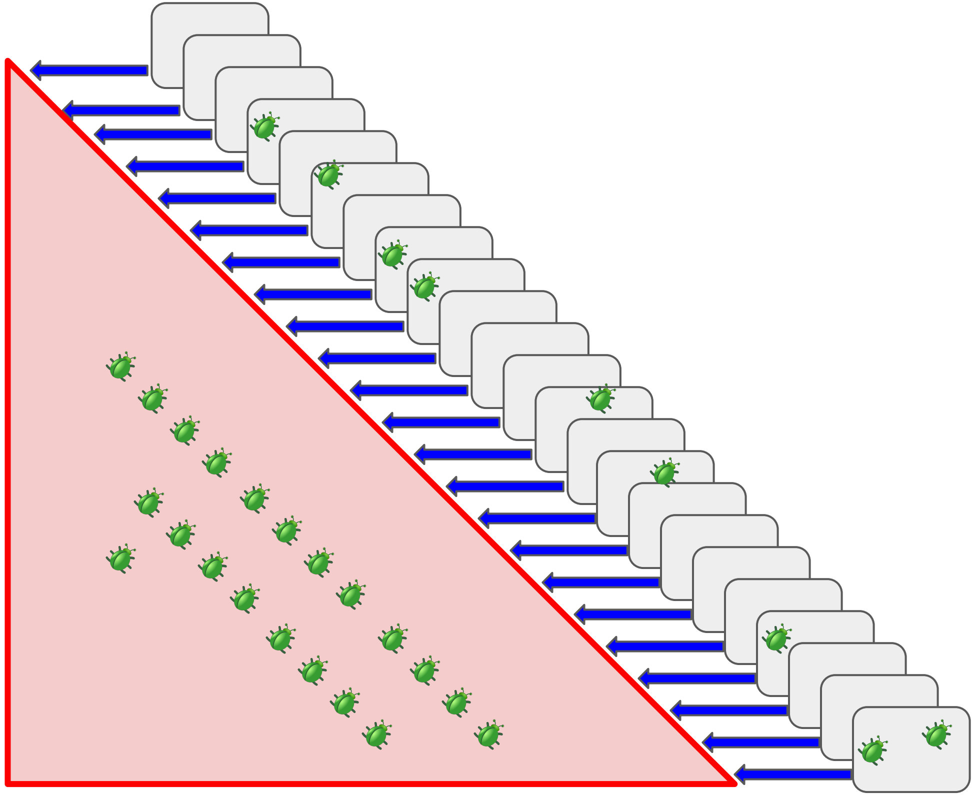
→ Autocompletion
LLM Suggestion

← Pre-training/
Finetuning

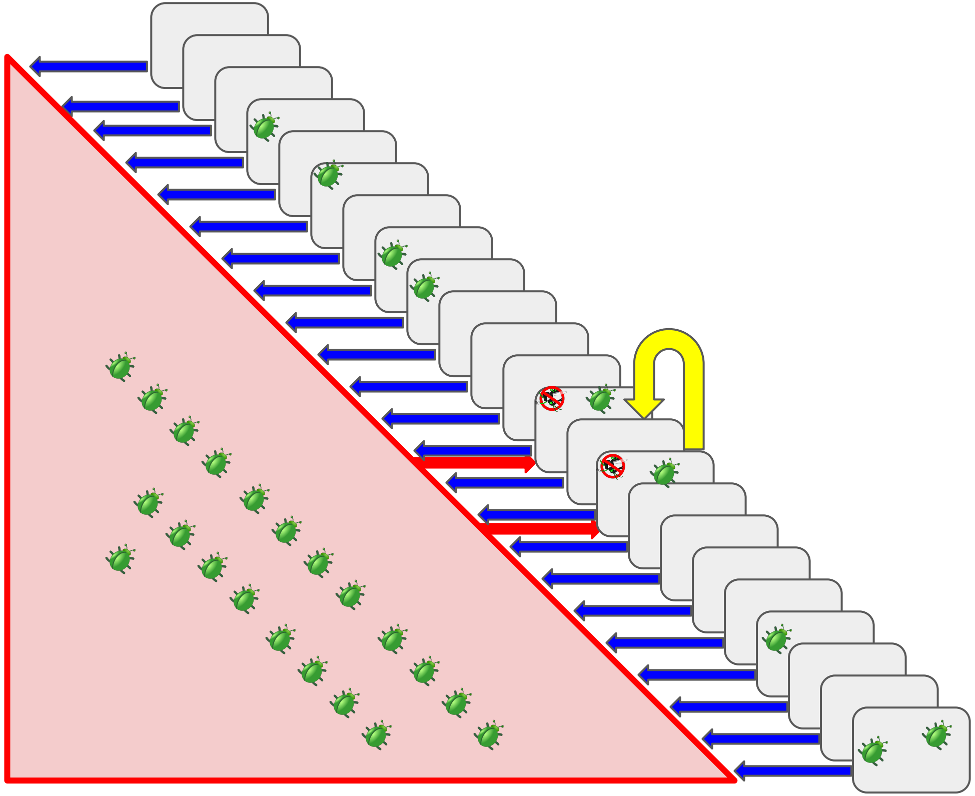
LLMs: SSCs of type IV

- Source code to LLMs
  - LLMs are trained on extremely large curated corpus
  - LLMs may be guided by developer actons (which suggestions are “accepted”)
  - Both input sources are highly problematic
- 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 (SSC IV) .
  - LLMs generate suggestions 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

Is LLM input data of high quality?



Do LLMs introduce bugs?



LLMs do introduce bugs

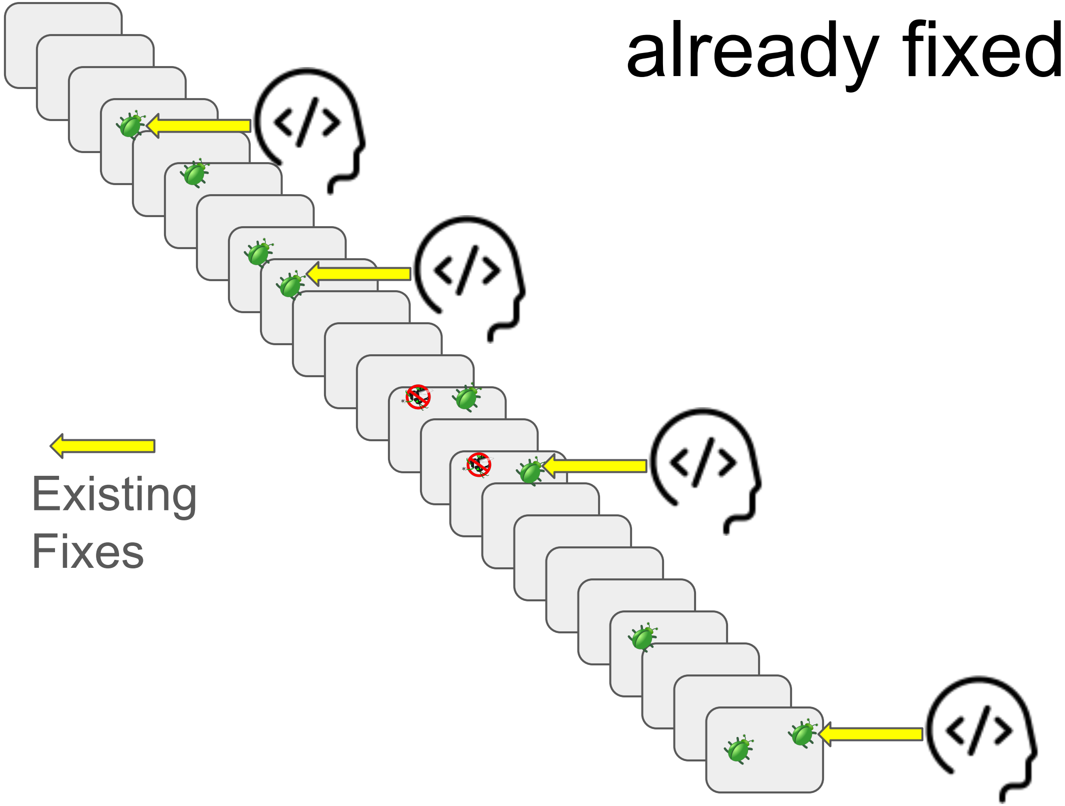
- “Disappointed with … Copilot because they’re not getting the **accuracy** and the response that they want,... customers are left cleaning up the mess…” Benioff
- Use of GitHub Copilot introduced **41% more bugs** (Upleavel study of 800 developers)
- nearly **40%** of the suggestions by GitHub's Copilot code-generation tool are **erroneous** from a security point of view “An Empirical Cybersecurity Evaluation of GitHub Copilot's Code Contributions."
- ..ChatGPT could diagnose coding errors: wrong 52% of the time .. human developers preferred the ChatGPT answers more than 35% of the time, even though 77% of those preferred answers were in fact wrong.

LLMs: Copy/Knowledge SSC on Steroids

- It is easy to forget that AI has no “Intelligence”
  - **All intelligence comes from the training data**
  - **LLMs copy/translate/transform their training data**
- How to avoid problems with input data?
  - Use only **high-quality curated data**
  - But **LLM need lots of data**
  - **Auto-curate?**
- Option A: use LLM to fix pre-training data --- no good
- Option B: use source code version history to apply fixes done by developers

How to clean massive training data?

Many bugs have been already fixed by develop



Existing Fixes

How to find these fixes?

- Need the entirety of OSS: use WoC
- Fix VCS data quality at scale: use WoC
- Make analysis run in minutes not months: use WoC

Open source LLM training data

- Open source
  - The Stack: 6TB of permissively-licensed source code files covering 358 languages

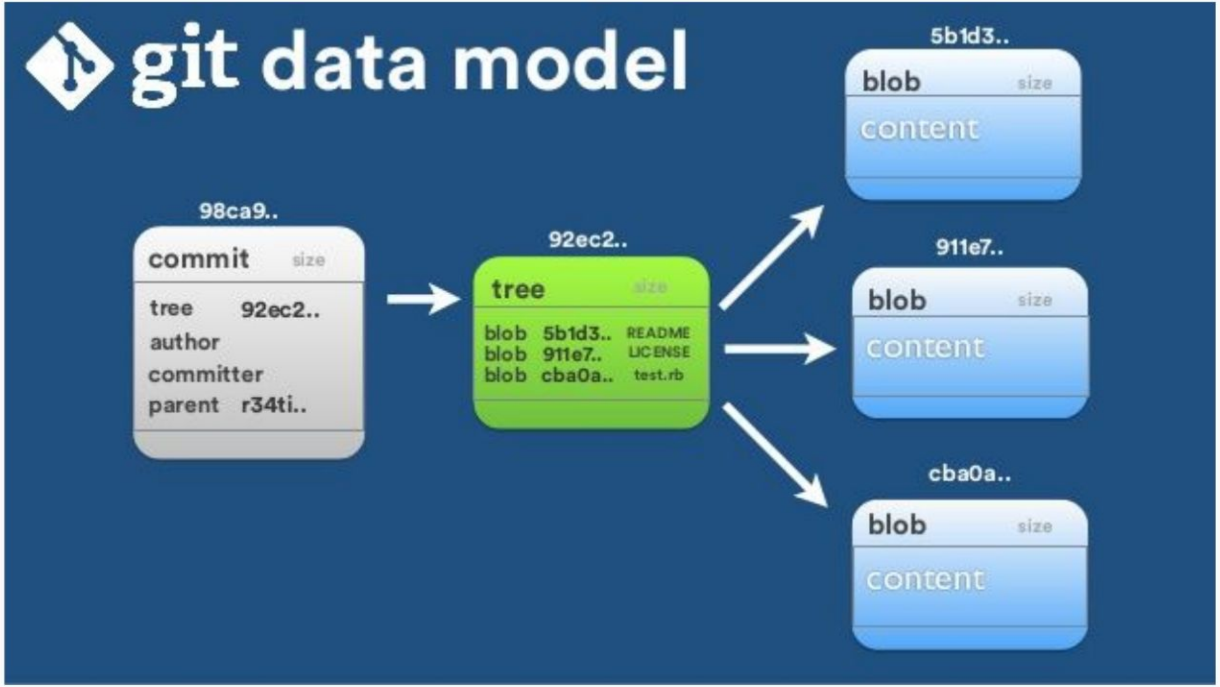
https://huggingface.co/datasets/bigcode/the-stack

  - The Stack V2: 67.5TB, 600 languages

https://huggingface.co/datasets/bigcode/the-stack-v2

  - Includes binaries, images, forked content
  - “Deduplicated” collection 32TB for V2) recommended for training LLMs

Git Model


git data model
98ca9..
commit size
tree 92ec2..
author
committer
parent r34ti..
92ec2..
tree size
blob 5b1d3.. README
blob 911e7.. LICENSE
blob cba0a.. test.rb
5b1d3..
blob size
content
911e7..
blob size
content
cba0a..
blob size
content

How many blobs in Stack V2 have fixes?

- Start with Stack V2
- Has 582M blobs << 26B blobs in WoC
- 74% are for files that have never been edited
    - Join with bb2cf and obb2cf
- 14% are for files that have been ever edited and have no new version
- 18% for files with future versions (70% of all edited files)
- 20% of the commits for the edited files are fixes: (correct|fix|bug|problem)

- In summary **140M blobs have known fixes**

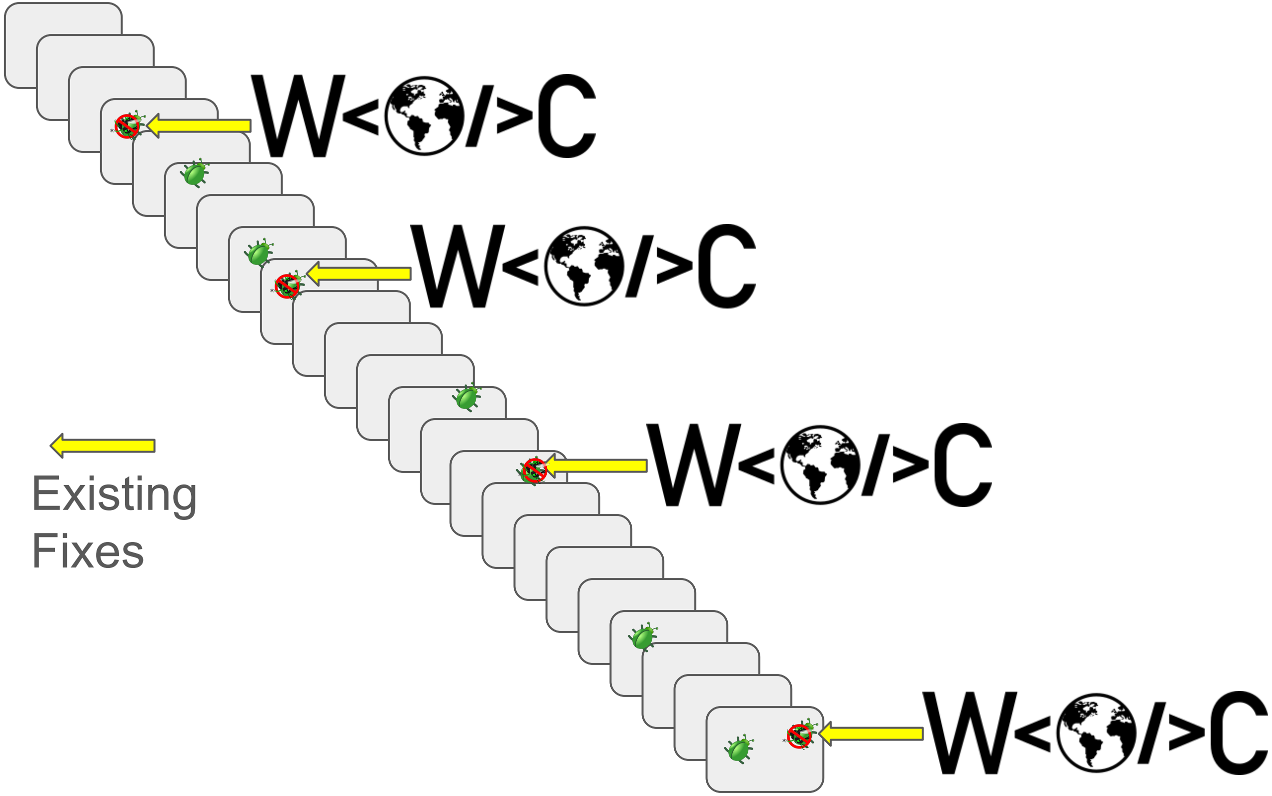
How many blobs have known (orphan) vulnerabilities in the Stack?

- 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

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

Now we can auto-fix bugs in the training data: n



Existing Fixes

Open Challenges in Securing LLM SSCs

+ **Identify/fix problems in the training data**

- **Identify data leakage - version control to the rescue**
- **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

Data leakage

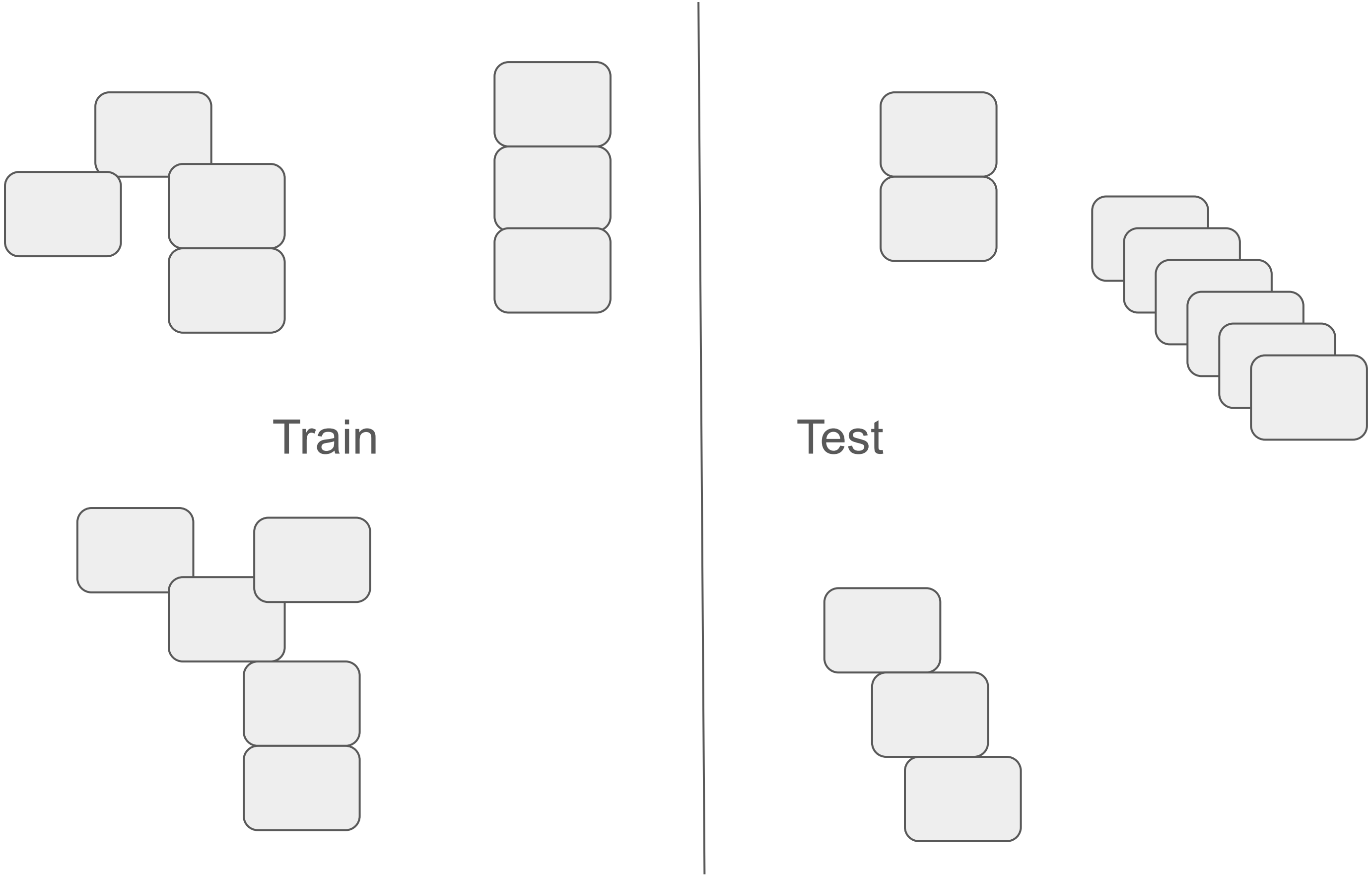
+ **Versions of a file (parent-child blob) are similar**
+ **Find cliques in parent-child blob graph**
+ **Ensure each clique is in test or train, not in both**

- **Challenges**
    - **Blobs with empty/trivial/template content: exclude from graph**
    - **Use of models that do not reveal pretraining data: infer that**

Largest cliques

15,158,550	C282
58,114	C13212
54,279	C46069
33,757	C135746

Blob parent-child relationship --- Cliques



Identifying generated code

- Probabilistic, e.g, perplexity is lower for generated text
- Language patterns and style [29, 25]
- Factual errors [93]
- Deep learning [70]
- Repeated higher-order n-grams [27]
- Basic machine learning models [0, 13]

References

[0] Idialu, O. J., Mathews, N. S., Maipradit, R., Atlee, J. M., & Nagappan, M. (2024, April). Whodunit: Classifying Code as Human Authored or GPT-4 generated-A case study on CodeChef problems. In *Proceedings of the 21st International Conference on Mining Software Repositories* (pp. 394-406).

[13] Heather Desaire, Aleesa E Chua, Madeline Isom, Romana Jarosova, and David Hua. Distinguishing academic science writing from humans or chatgpt with over 99% accuracy using off-the-shelf machine learning tools. Cell Reports Physical Science, 2023.

[25] Leon Fröhling and Arkaitz Zubiaga. Feature-based detection of automated language models: tackling gpt-2, gpt-3 and grover. PeerJ Computer Science, 7:e443, 2021.

[29] Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. arXiv preprint arXiv:2301.07597, 2023

[27] Matthias Gallé, Jos Rozen, Germán Kruszewski, and Hady Elsahar. Unsupervised and distributional detection of machine-generated text. arXiv preprint arXiv:2111.02878, 2021.

[70] Juan Rodriguez, Todd Hay, David Gros, Zain Shamsi, and Ravi Srinivasan. Cross-domain detection of gpt-2-generated technical text. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1213–1233, 2022.

[93] Wanjun Zhong, Duyu Tang, Zenan Xu, Ruize Wang, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. Neural deepfake detection with factual structure of text. arXiv preprint arXiv:2010.07475, 2020.

LLM supply chain

- LLMs are not just data

  Parameters

  Adaptors

  Fine-tunes

  Merges

- Documentation/Publications

  Structured README on Huggingface

  Version control of parameters/models

  Links to source code (relatively few)

- Execution environment
  - https://huggingface.co/icputrd

Summary

- LLM output is only as good as LLM training input
  - May get worse over time via user feedback, and reuse of public generated code
  - Data curation, including curriculum, etc is the only solution likely to work
  - Model/data now use version control, relationships to training data in README
- Curation leveraging SSC 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

Hiring: Foundational AI: EECS UTK

https://www.eecs.utk.edu/resources/employment-opportunities/tenure-track-positions-foundational-ai/

WoC Provenance

Provenance
Code Copy

Provenance
Version History

WoC Provenance

Provenance
Code Copy

blob (git version)

contains: content of a file

Provenance
Version History

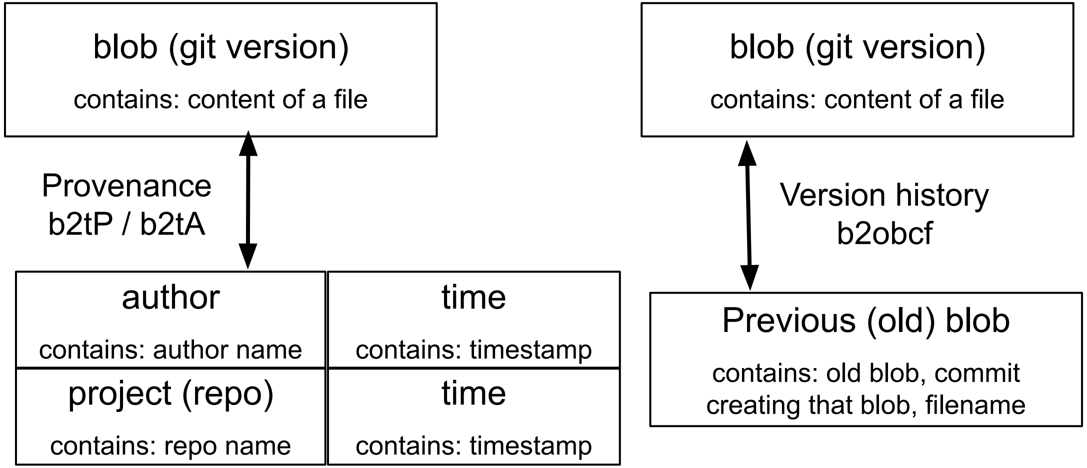
blob (git version)

contains: content of a file

WoC Provenance

Provenance
Code Copy

Provenance
Version History



blob (git version)

contains: content of a file

Provenance
b2tP / b2tA

author

contains: author name

time

contains: timestamp

project (repo)

contains: repo name

time

contains: timestamp

blob (git version)

contains: content of a file

Version history
b2obcf

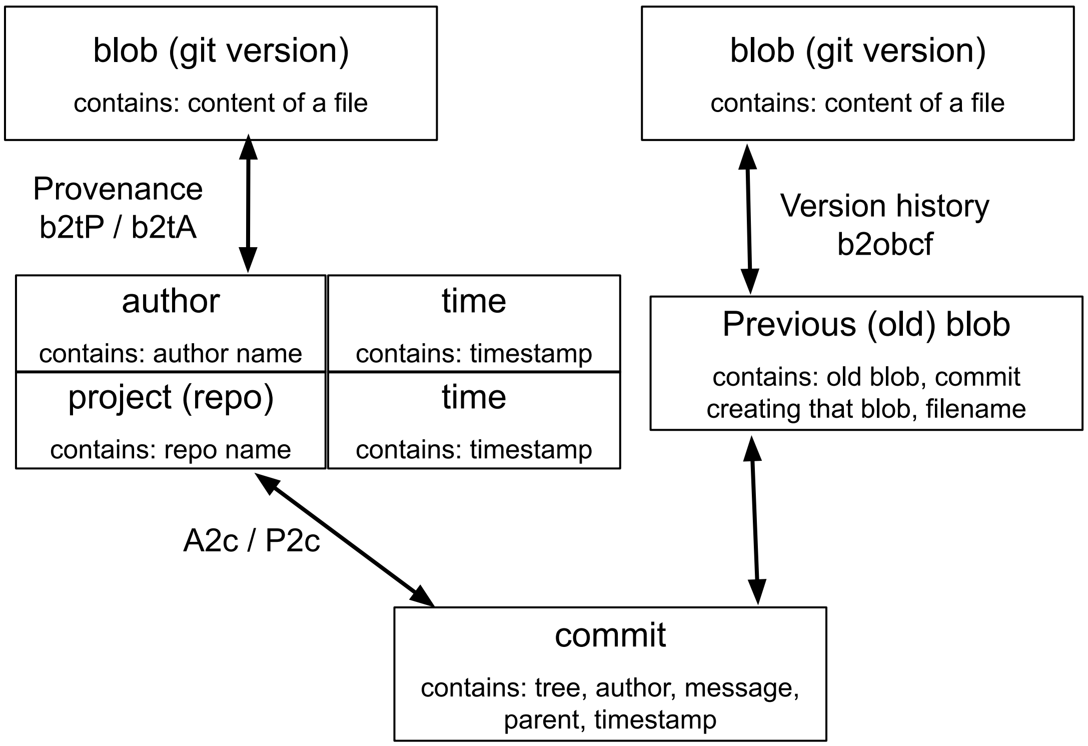
Previous (old) blob

contains: old blob, commit creating that blob, filename

WoC Provenance

Provenance
Code Copy

Provenance
Version History



blob (git version)

contains: content of a file

Provenance
b2tP / b2tA

author

contains: author name

time

contains: timestamp

project (repo)

contains: repo name

time

contains: timestamp

A2c / P2c

blob (git version)

contains: content of a file

Version history
b2obcf

Previous (old) blob

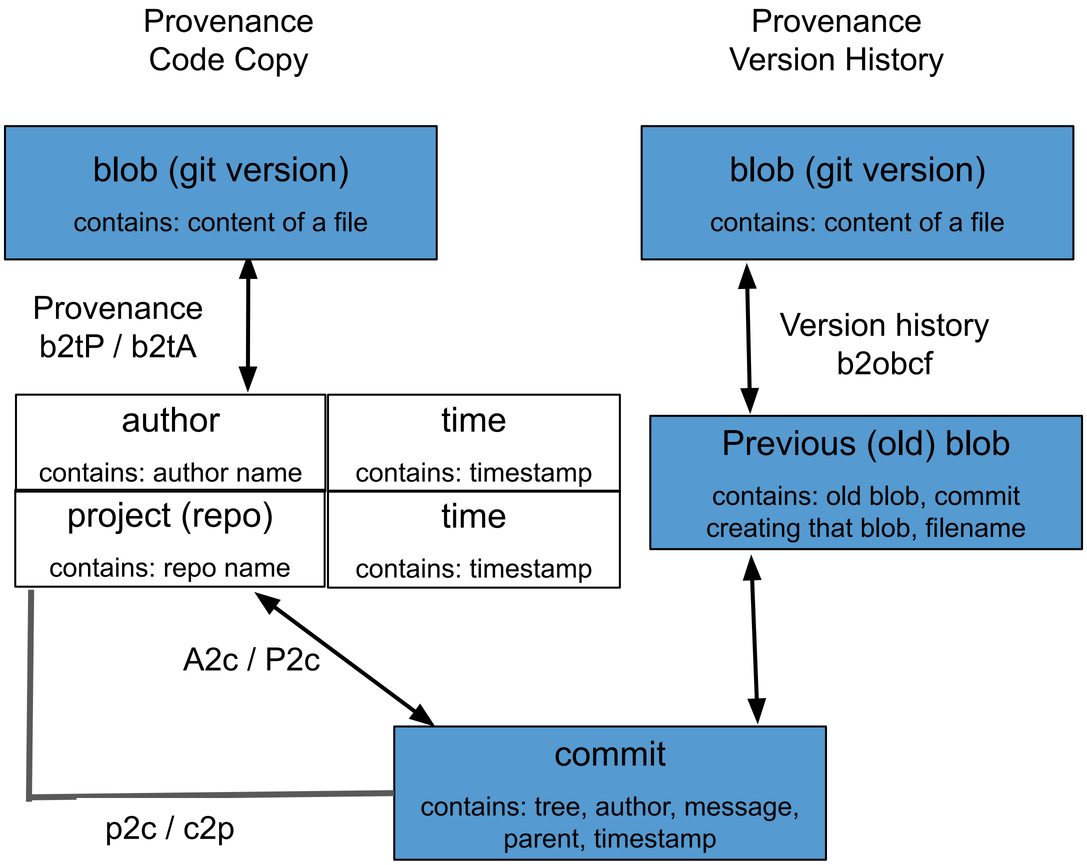
contains: old blob, commit creating that blob, filename

commit

contains: tree, author, message, parent, timestamp

WoC Provenance

WoC layers



Provenance
Code Copy

blob (git version)

contains: content of a file

Provenance
b2tP / b2tA

author

contains: author name

time

contains: timestamp

project (repo)

contains: repo name

time

contains: timestamp

A2c / P2c

p2c / c2p

commit

contains: tree, author, message, parent, timestamp

Provenance
Version History

blob (git version)

contains: content of a file

Version history
b2obcf

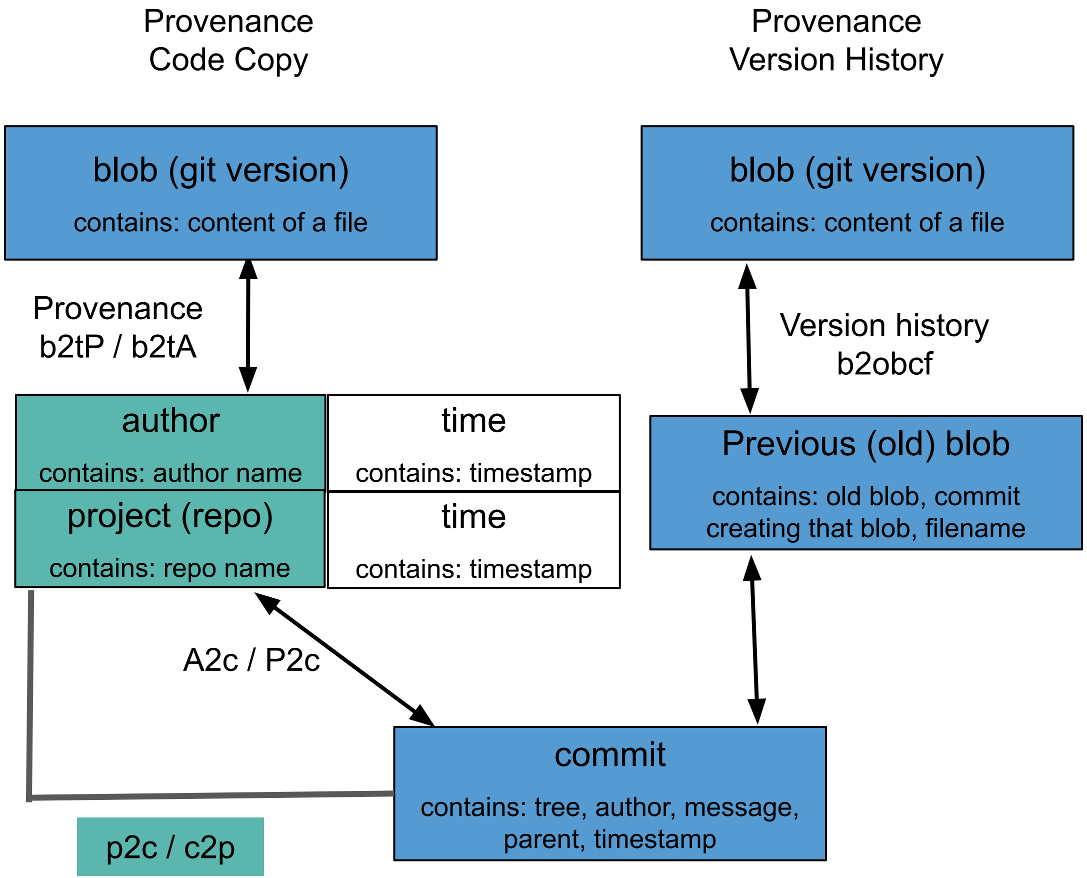
Previous (old) blob

contains: old blob, commit creating that blob, filename

Layer 1: raw content of commits, trees or blobs

(access trough showCnt[ID])

WoC Provenance



Provenance
Code Copy

blob (git version)

contains: content of a file

Provenance
b2tP / b2tA

author

contains: author name

time

contains: timestamp

project (repo)

contains: repo name

time

contains: timestamp

A2c / P2c

p2c / c2p

commit

contains: tree, author, message, parent, timestamp

Provenance
Version History

blob (git version)

contains: content of a file

Version history
b2obcf

Previous (old) blob

contains: old blob, commit creating that blob, filename

WoC layers

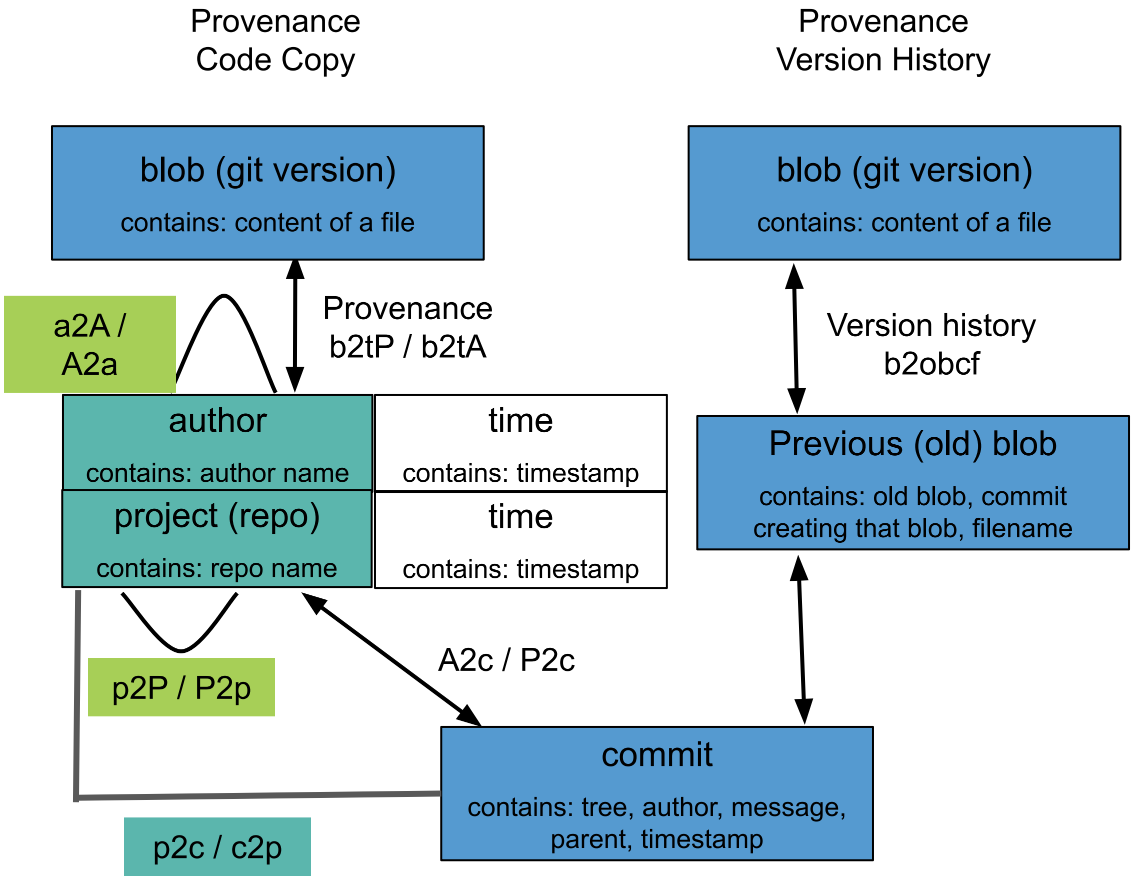
Layer 1: raw content of commits, trees or blobs

(access trough showCnt[ID])

Layer 2: repo names, author IDs, file names

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

WoC Provenance



Provenance
Code Copy

blob (git version)

contains: content of a file

a2A /
A2a

Provenance
b2tP / b2tA

author

contains: author name

time

contains: timestamp

project (repo)

contains: repo name

time

contains: timestamp

p2P / P2p

A2c / P2c

p2c / c2p

commit

contains: tree, author, message, parent, timestamp

Provenance
Version History

blob (git version)

contains: content of a file

Version history
b2obcf

Previous (old) blob

contains: old blob, commit creating that blob, filename

WoC layers

Layer 1: raw content of commits, trees or blobs

(access trough showCnt[ID])

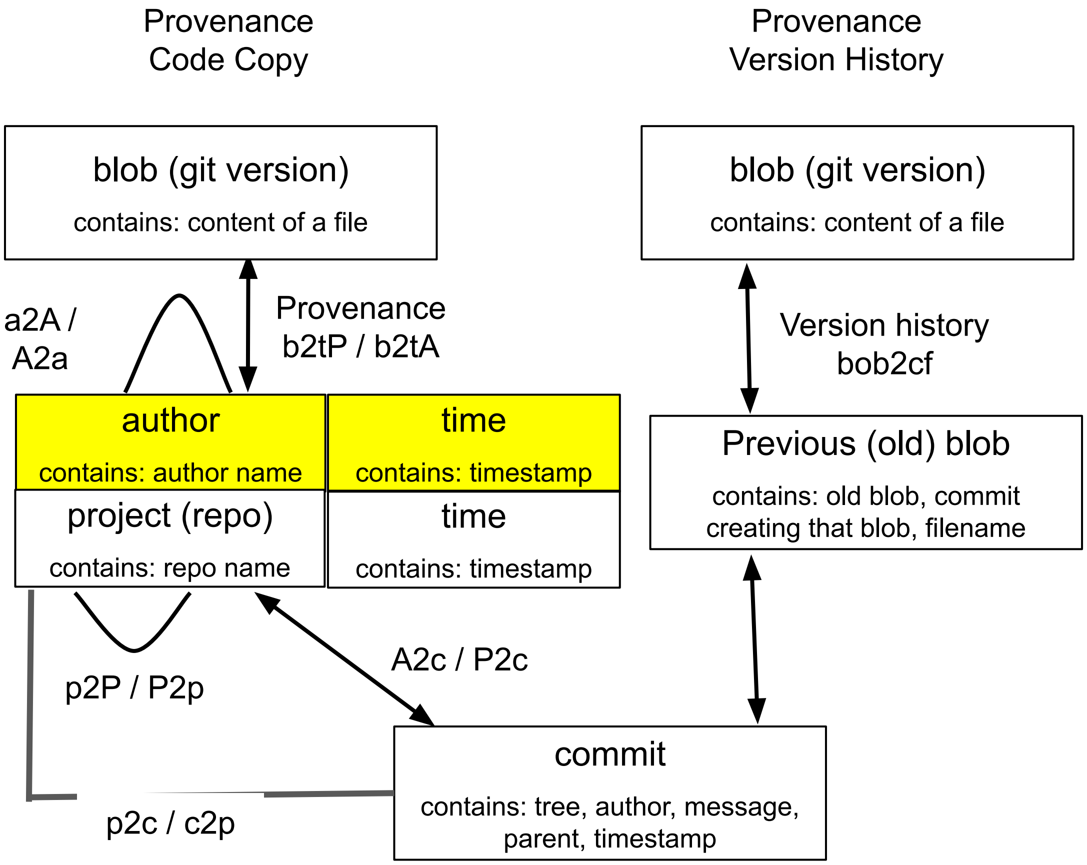
Layer 2: repo names, author IDs, file names

(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:

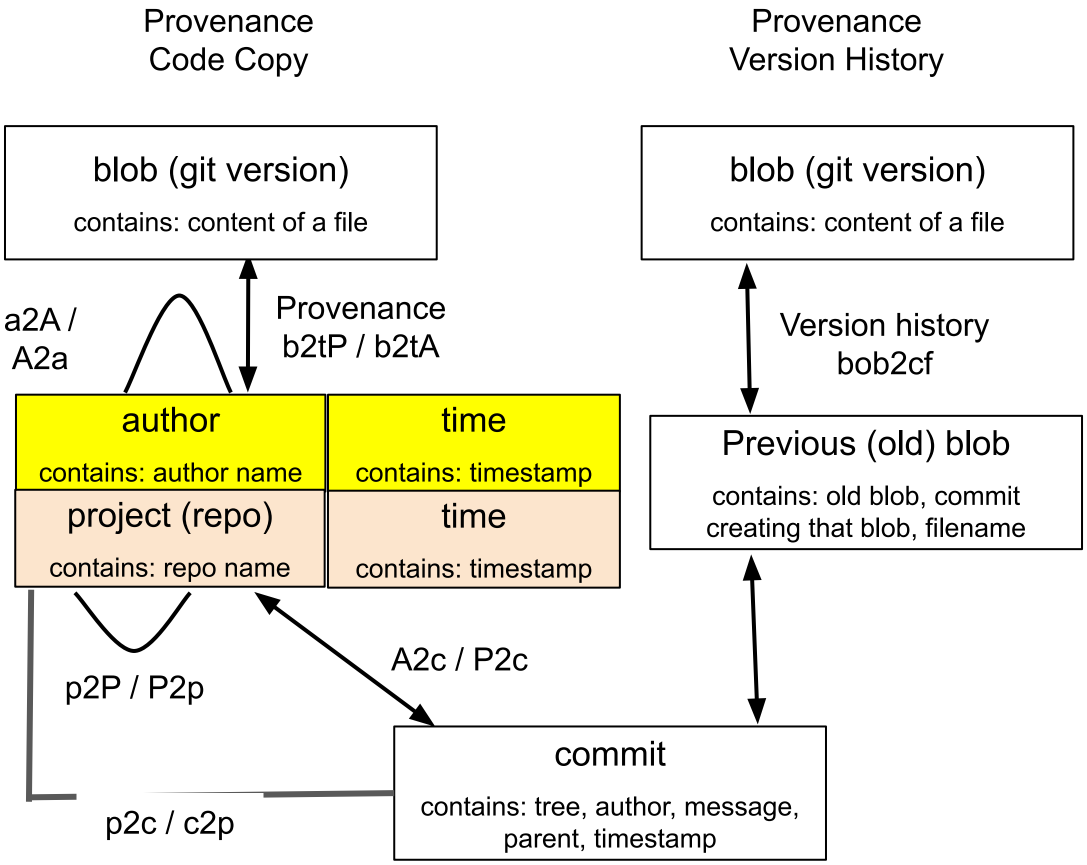
```
echo 99600f  | getValues b2tAc
echo 99600f  | getValues b2tP
```

Result:

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

…

WoC Provenance



Provenance
Code Copy

Provenance
Version History

blob (git version)
contains: content of a file

a2A /
A2a

Provenance
b2tP / b2tA

author
contains: author name

time
contains: timestamp

project (repo)
contains: repo name

time
contains: timestamp

p2P / P2p

A2c / P2c

p2c / c2p

commit
contains: tree, author, message, parent, timestamp

blob (git version)
contains: content of a file

Version history
bob2cf

Previous (old) blob
contains: old blob, commit creating that blob, filename

Example

Q1: Who/What project created this code?

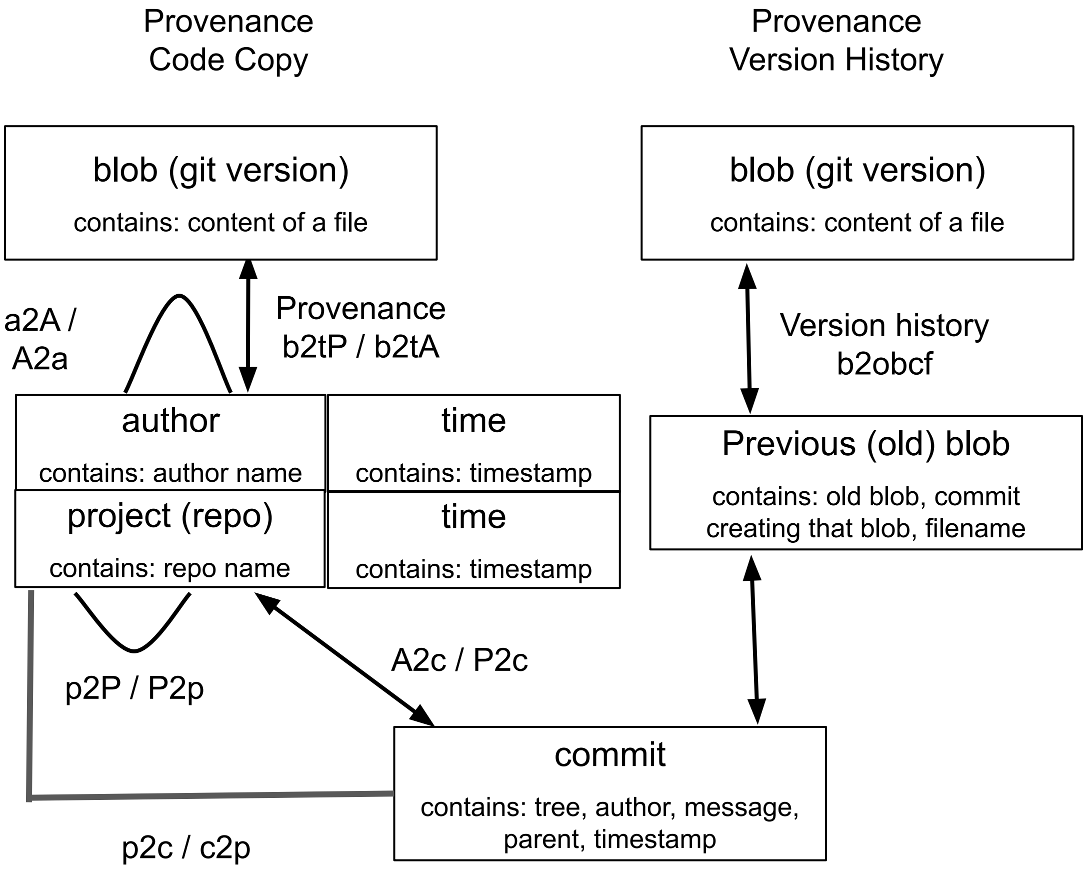
Shell code:

```
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



Provenance
Code Copy

Provenance
Version History

blob (git version)
contains: content of a file

a2A / A2a

Provenance
b2tP / b2tA

author
contains: author name

time
contains: timestamp

project (repo)
contains: repo name

time
contains: timestamp

p2P / P2p

A2c / P2c

p2c / c2p

commit
contains: tree, author, message, parent, timestamp

blob (git version)
contains: content of a file

Version history
b2obcf

Previous (old) blob
contains: old blob, commit creating that blob, filename

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:

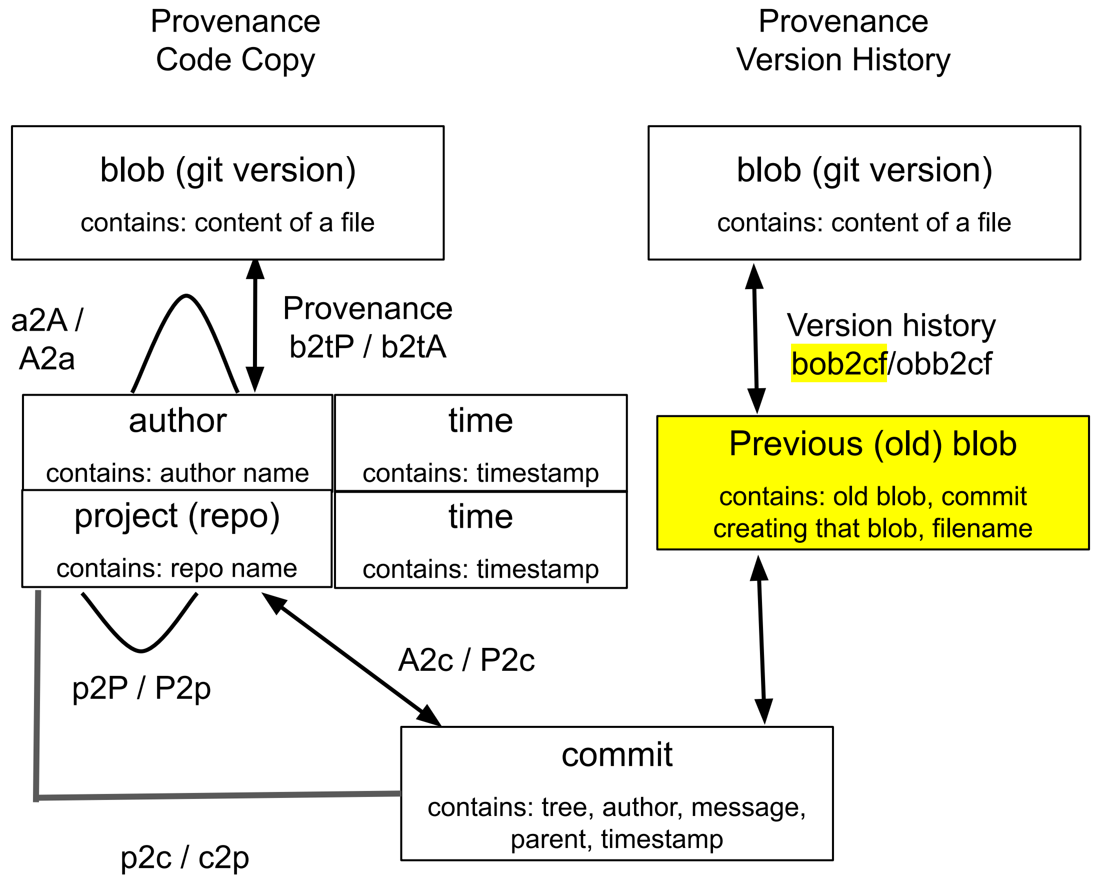
- List of blob;old blob;commit;filename:

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?

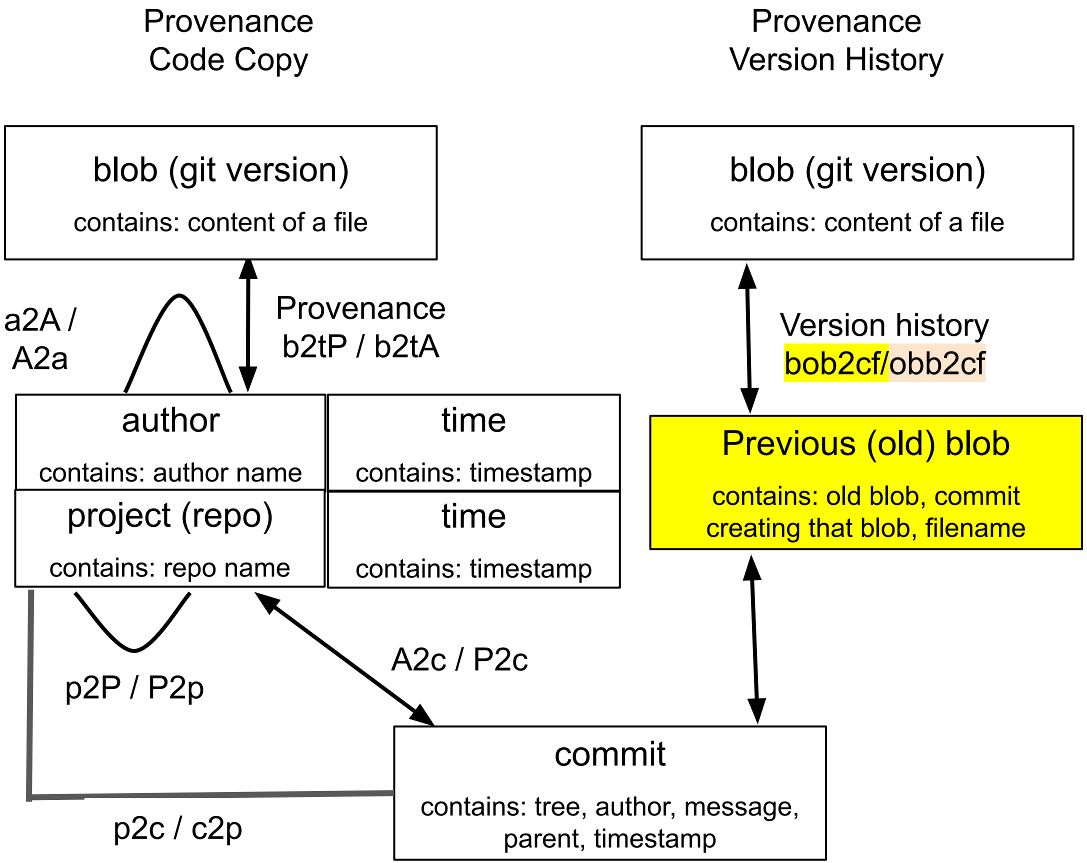
Shell code:

```
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



Provenance
Code Copy

Provenance
Version History

blob (git version)
contains: content of a file

a2A /
A2a

Provenance
b2tP / b2tA

author
contains: author name

time
contains: timestamp

project (repo)
contains: repo name

time
contains: timestamp

p2P / P2p

A2c / P2c

p2c / c2p

commit
contains: tree, author, message, parent, timestamp

blob (git version)
contains: content of a file

Version history
bob2cf/obb2cf

Previous (old) blob
contains: old blob, commit creating that blob, filename

Example

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

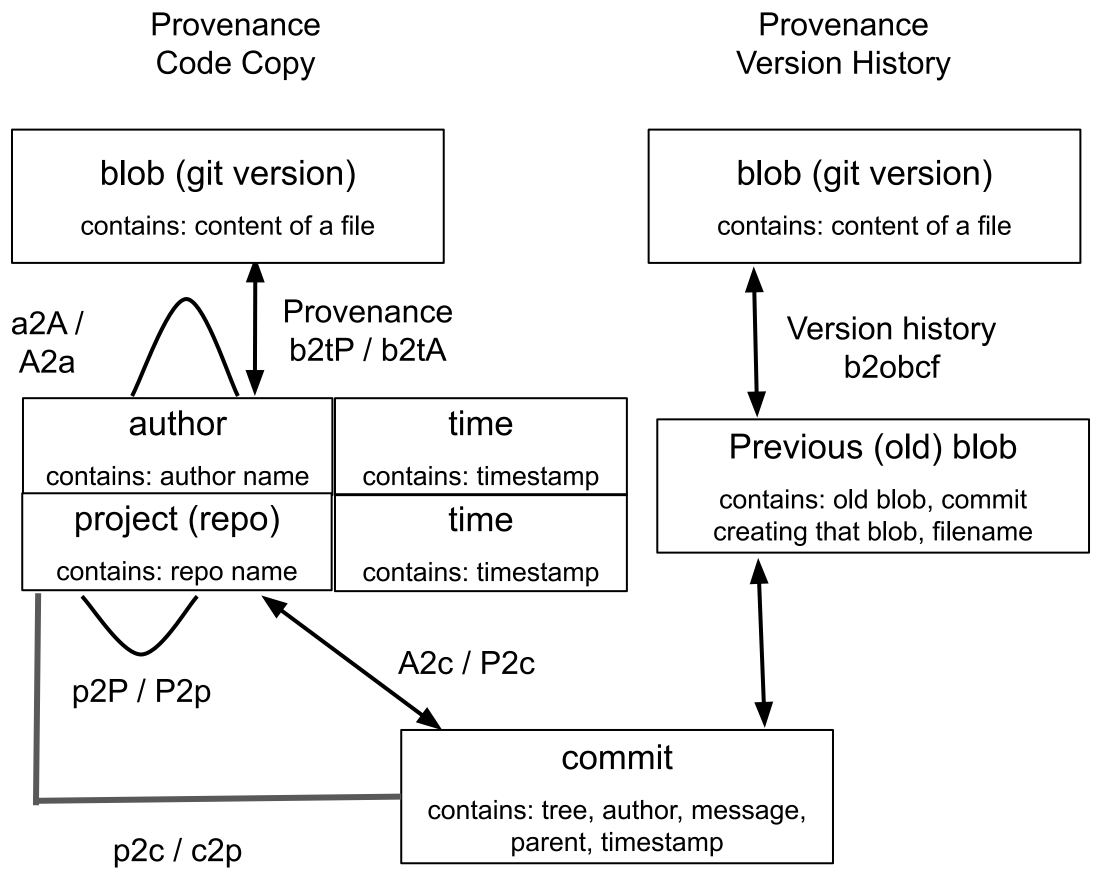
Shell code:

```
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

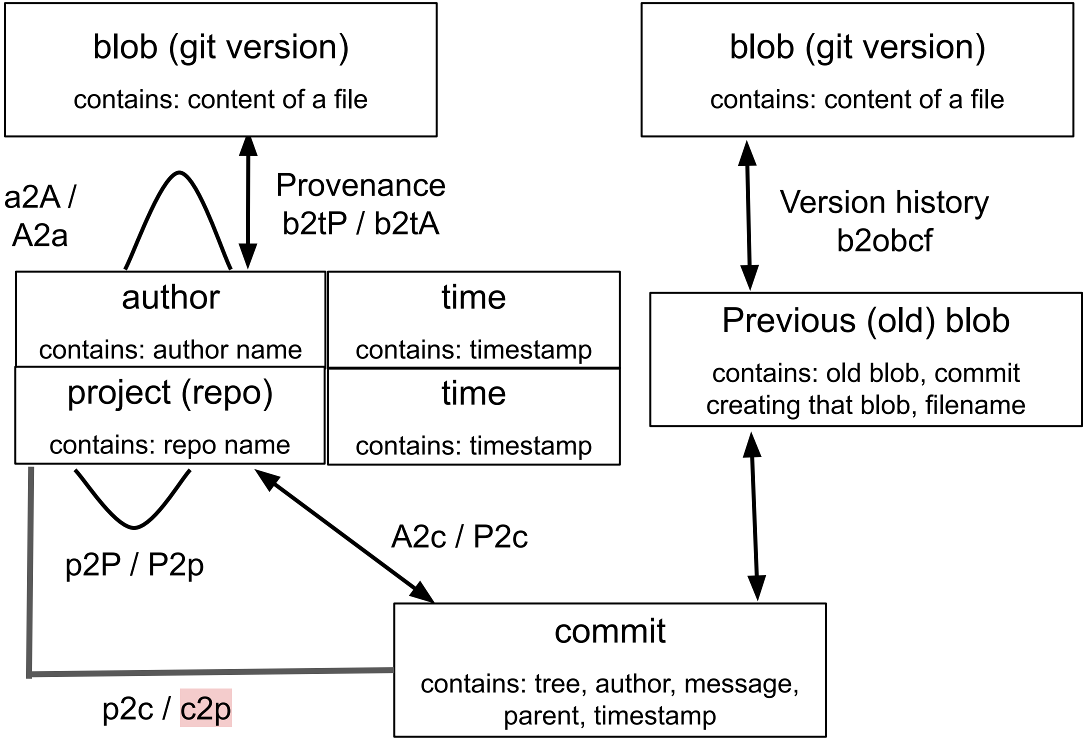
```
724ae9;47-studio-org_tensorflow
724ae9;8bitmp3_tensorflow
724ae9;Apidwalin_tensorflow-master
724ae9;OlexanderMiroshnychenko_tf_patch_test
724ae9;aakash30jan_tensorflow
724ae9;aaudiber_tensorflow
724ae9;abhilash1910_tensorflow
724ae9;adaalarm_tensorflow
724ae9;adamhillier_tensorflow
724ae9;adhadse_tensorflow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

WoC Provenance

Example

Provenance
Code Copy

Provenance
Version History



Q5: Which project(s) contain this commit?

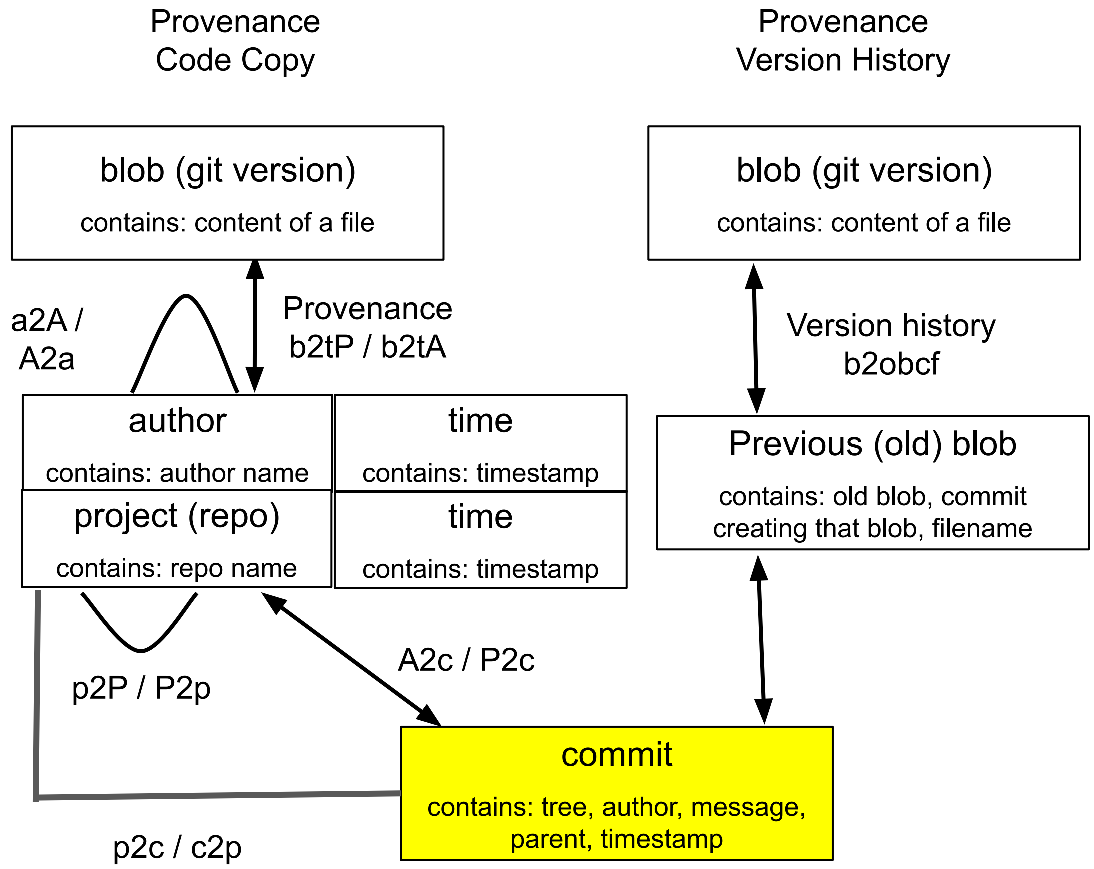
Shell code:

```
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;
55338bb43c76edad2557be1cd62dc315c410631
4;
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-Rev
Id: 371980850__NEWLINE__Change-Id:
```