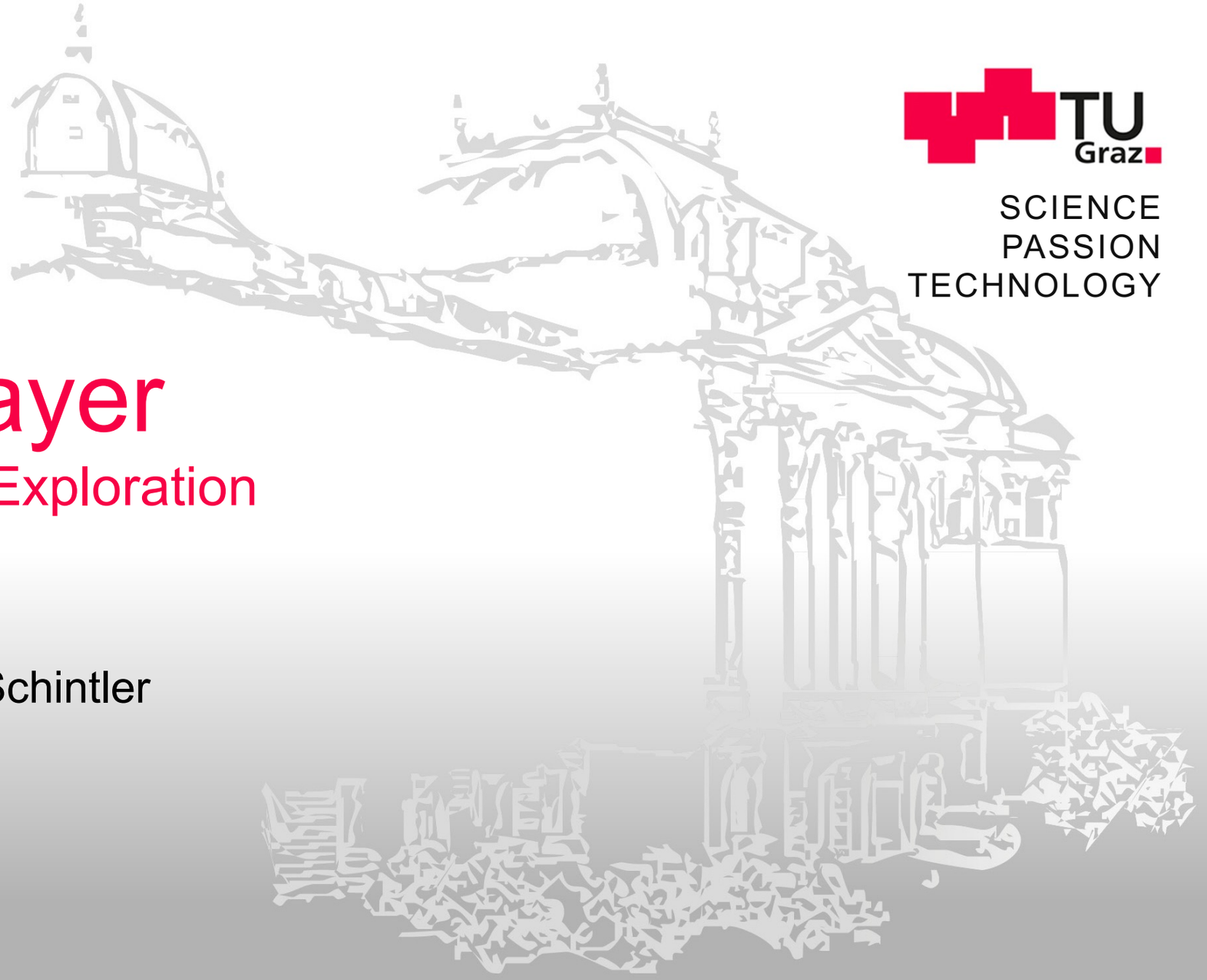


TU Graz

SCIENCE
PASSION
TECHNOLOGY

Gizual Data Layer

Enabling Browser-Based Exploration of Git Repositories

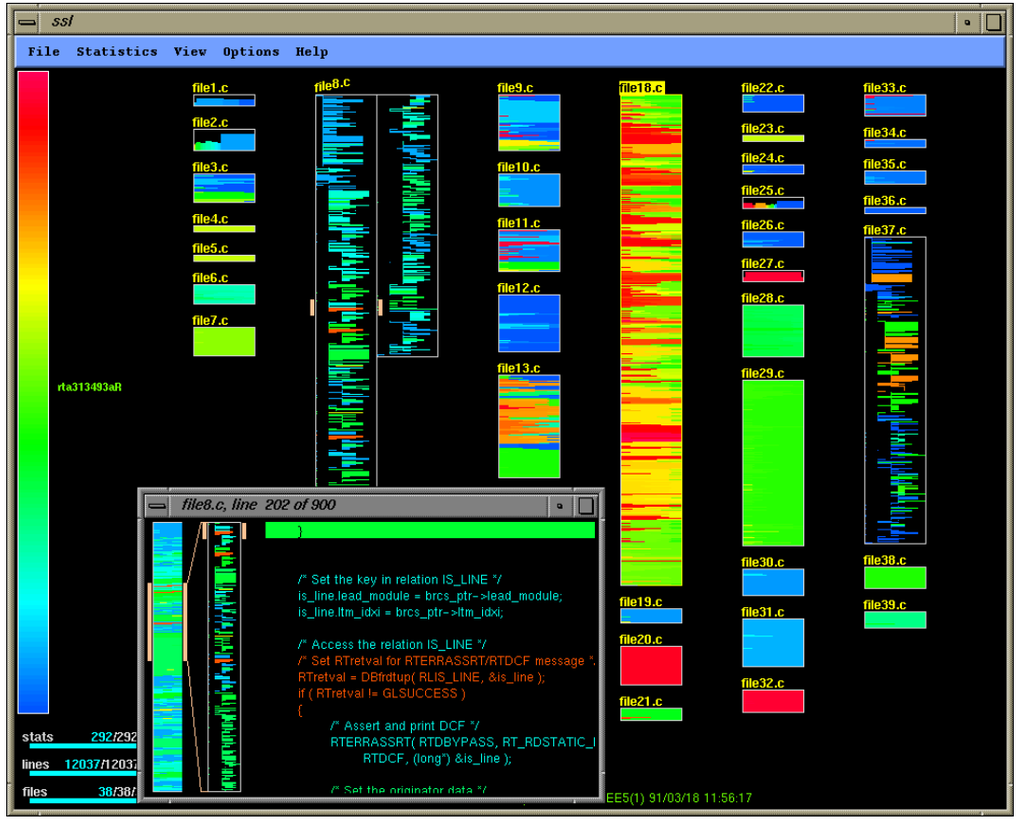
Master's Examination - Stefan Schintler

29 Jan 2025

gizual.com

Seesoft [1992]

- Stephen Eick et al, 1992[1].
- Visualisation of line-oriented software statistics
- Colour-coding each line of code:
  - Age (scale: blue = cold to red = hot).
  - Author (discrete palette).

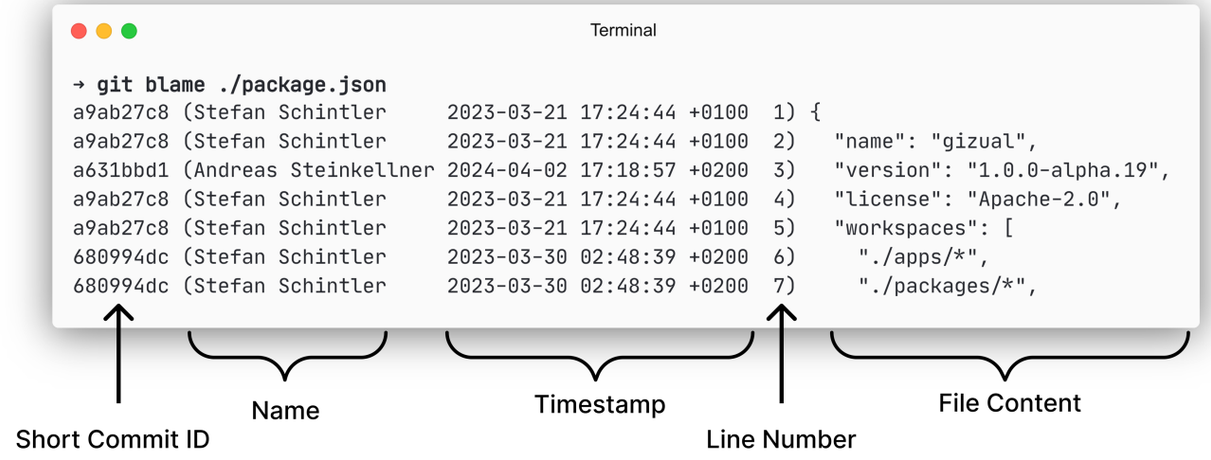


Seesoft. [Image extracted from Eick et al.[1] and used under §42f of Austrian copyright law.].

1. *Seesoft - A Tool for Visualizing Line Oriented Software Statistics*; Stephen Eick et al; IEEE Transactions on Software Engineering; No. 18, Vol. 11, Nov 1992. doi:10.1109/32.177365

Git Command Line Tool[1] [2006]

- Widespread adoption.
- Dependency on POSIX shell.
- Cross-platform availability. (Linux, macOS, Windows)
- Text-based Interface.

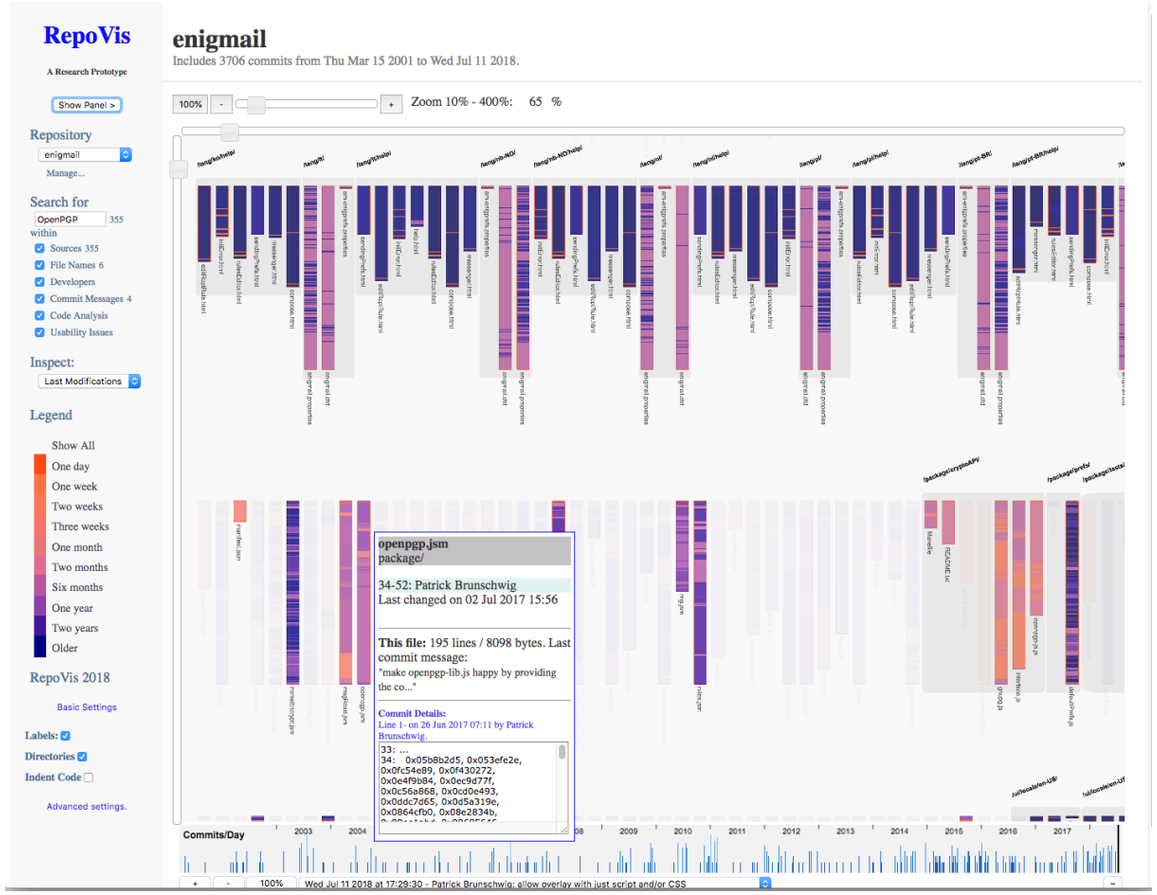


Output of Git CLI blame command.
[Image created by the author of this presentation]

1. Git Command Line; https://git-scm.com/

RepoVis [2018]

- Johannes Feiner and Keith Andrews[1]
- Line-oriented visualisation.
- Age and Author colour-coding.
- Backend: Rack, Ruby Sinatra, CouchDB, libgit2.
- Frontend: JavaScript, PixiJS, WebGL.



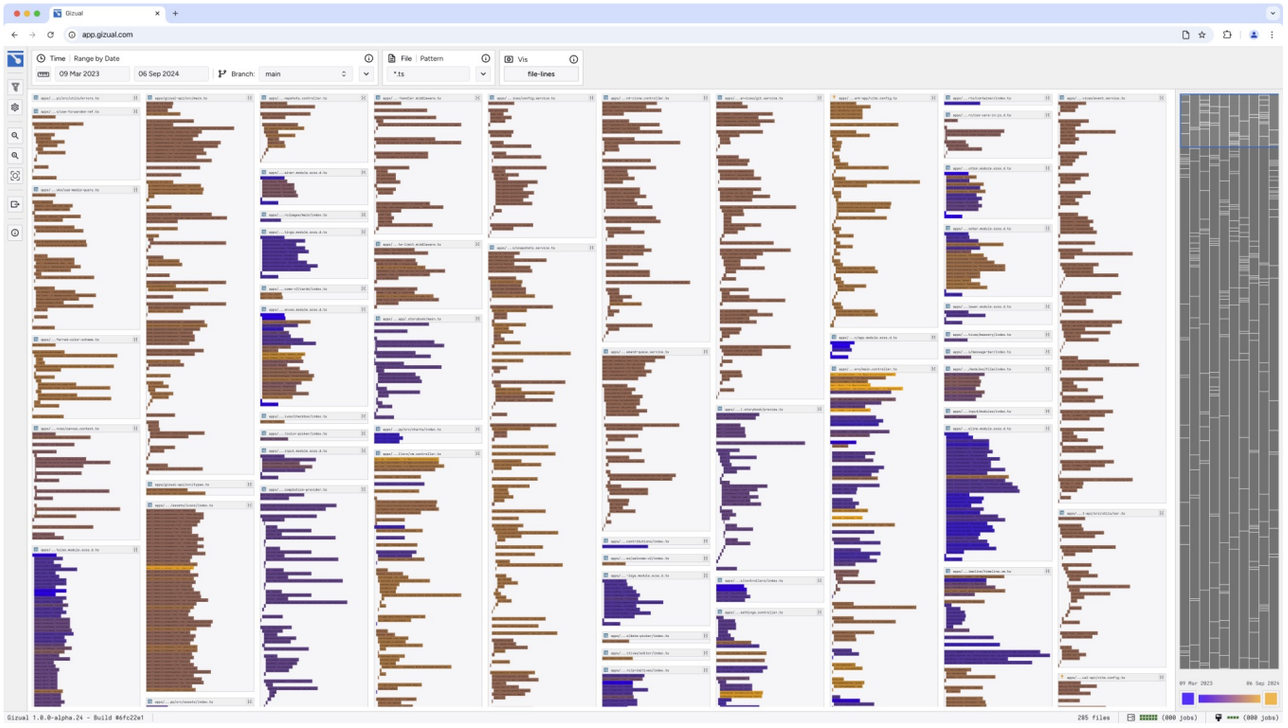
RepoVis; 2018 [Image extracted from Feiner and Andrews[1] and used under §42f of Austrian copyright law.].

1. *RepoVis: Visual Overviews and Full-Text Search in Software Repositories;* Johannes Feiner and Keith Andrews; Proc. 6th IEEE Working Conference on Software Visualization (VISSOFT 2018); Madrid, Spain, 24 Sep 2018. doi:10.1109/VISSOFT.2018.00009

Gizual [2022 →]

gizual.com

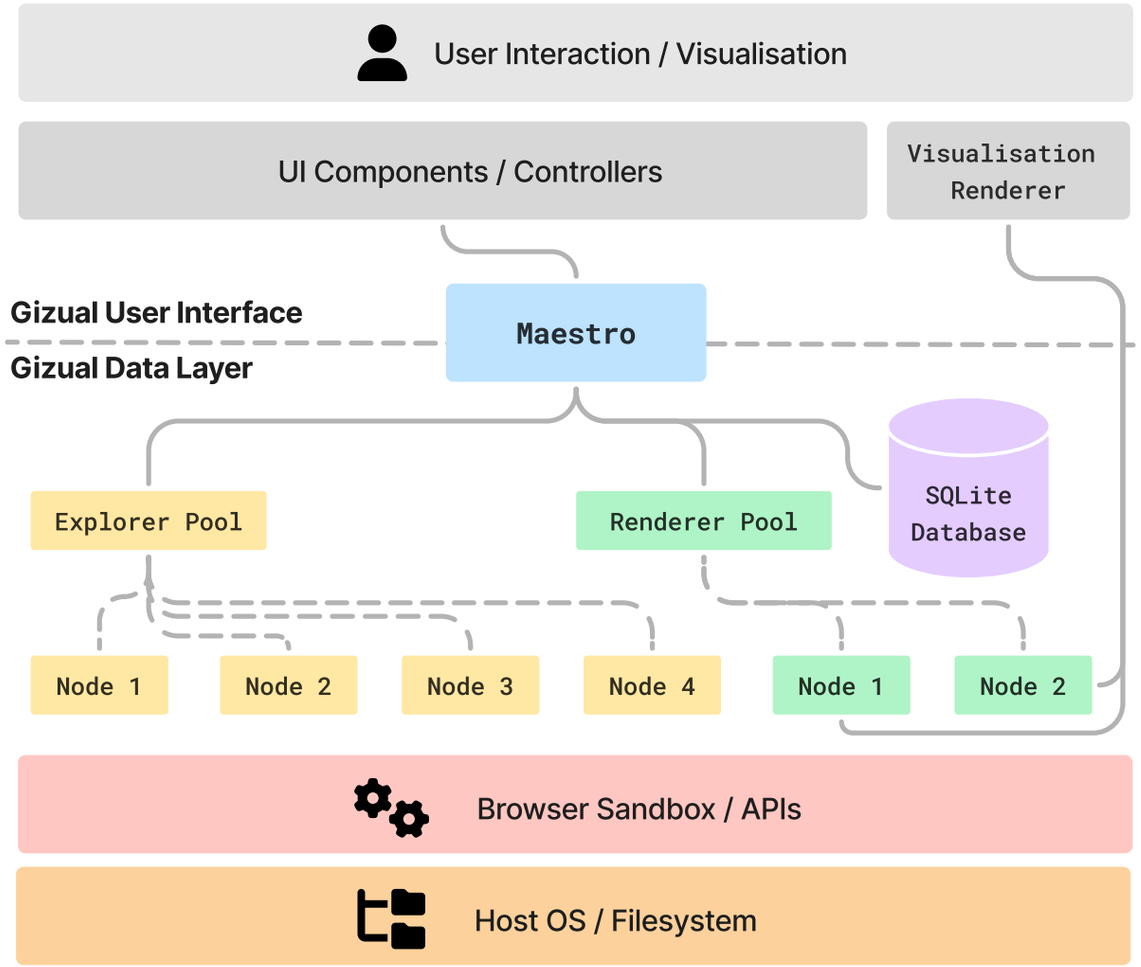
- **Intuitive Insights:**
  Line-oriented visualisation with Age and Author colour-coding.
- **High Performance:**
  Fast Visualisations on-demand.
- **Privacy First:**
  Local data never uploaded.
- **Stack:**
  TypeScript, React, Mantine, Rust, WebAssembly (libgit2), Web Workers.



Gizual, 2024.

Gizual‘s Architecture

- Gizual User Interface[1]
  by Andreas Steinkellner
  - Visualisation.
  - User Interaction.
- Gizual Data Layer
  - In-browser data processing (WebAssembly).
  - Broad browser support.
  - Performance (pools of WebWorkers).



Gizual‘s Architecture [Image created by the author of this presentation]

1. Gizual User Interface: Browser-Based Visualisation for Git Repositories; Andreas Steinkellner; Master’s Thesis; Graz University of Technology, Austria; 09 Dec 2024 https://ftp.isds.tugraz.at/pub/theses/asteinkellner-2024-msc.pdf

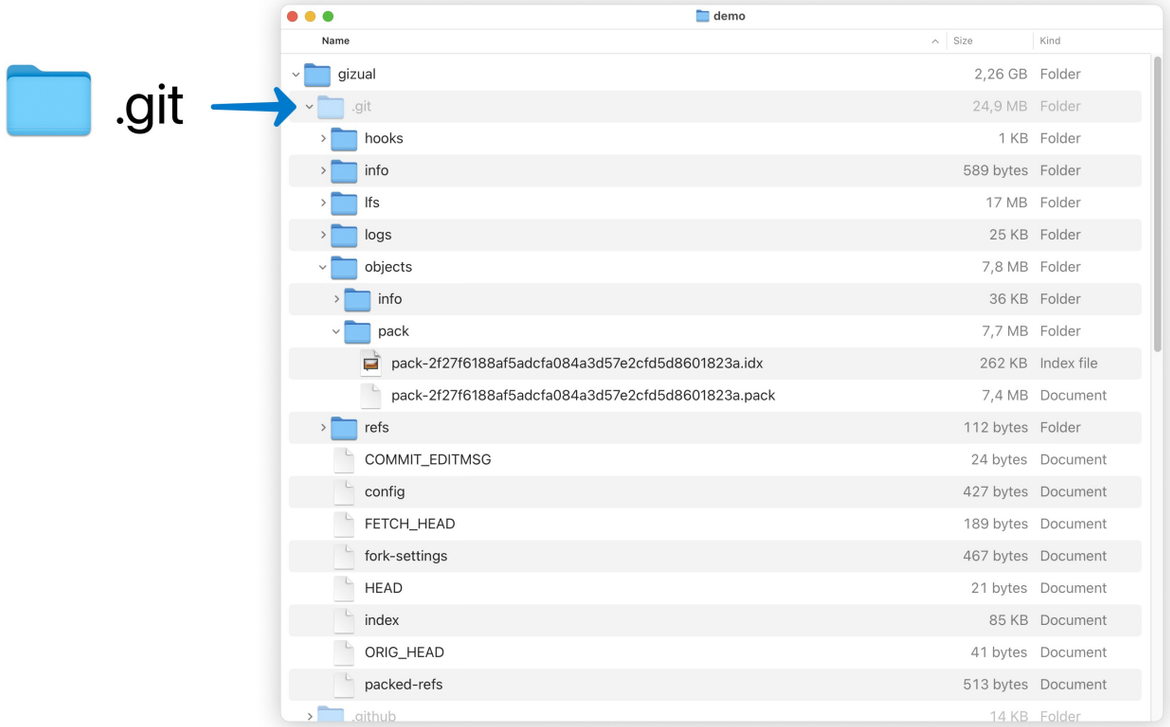
Goals of Gizual‘s Data Layer

- **Local:** Avoid uploading to a server.
- **Fast:** Provide results quickly and / or incrementally.
- **Scalable:** Support large repositories:

Repository	GitHub Path	Size	.git/ Size	First Commit	# Commits
Vue 2	*vuejs/vue*	small	35 MB	10 Apr 2016	6,694
React	*facebook/react*	medium	657 MB	29 May 2013	25,873
VS Code	*microsoft/vscode*	large	1,025 MB	13 Nov 2015	134,419
Linux Kernel	*torvalds/linux*	very large	5,836 MB	16 Apr 2005	1,310,317

The four GitHub repositories used to test Gizual. Statistics collected on 29 Oct 2024

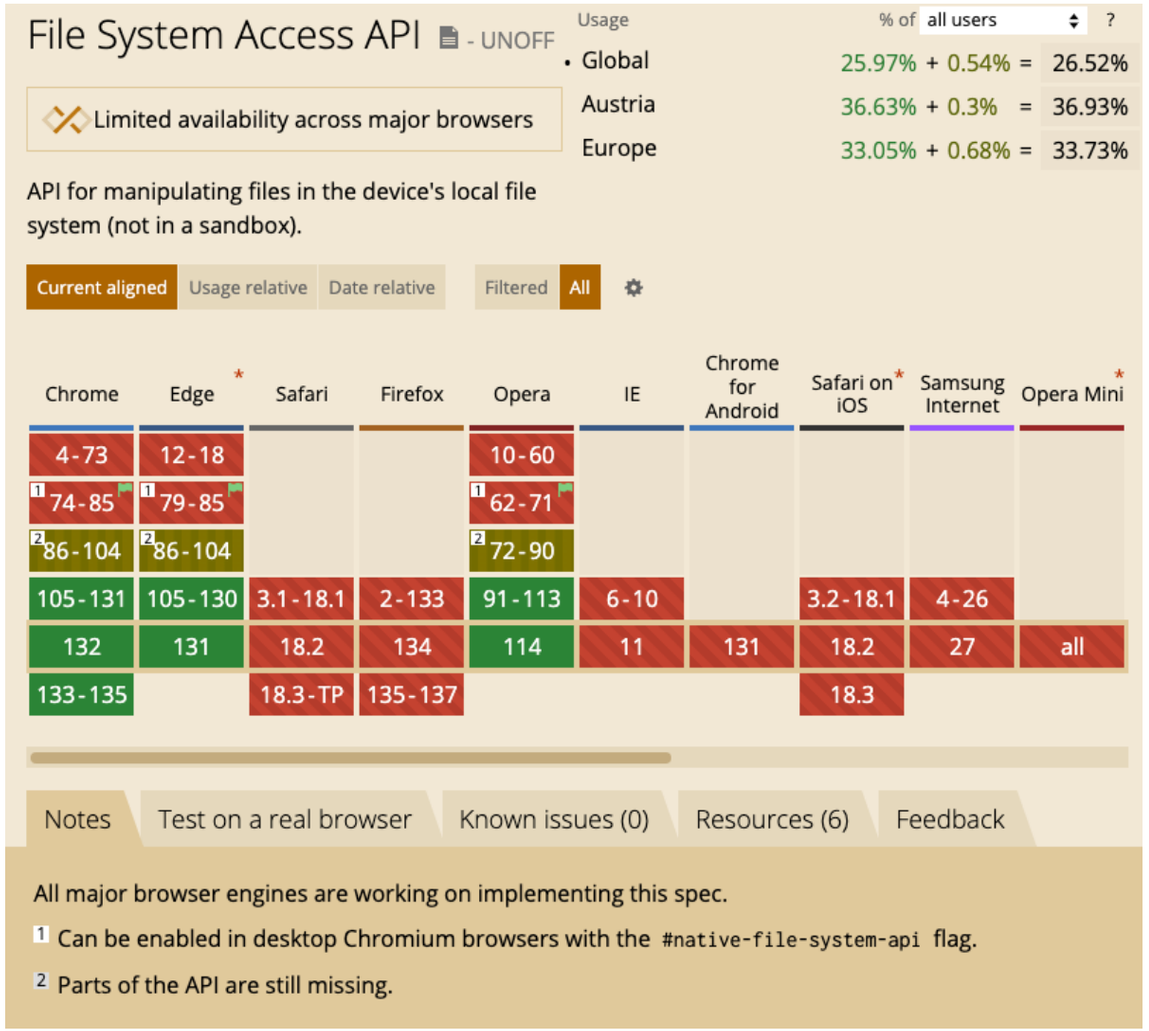
Loading Data into the Browser



[Image created by the author of this presentation]

File System Access API[1]

- Abstraction of file system.
- Mostly async API.
- Access local files directly.
- Supported by Chromium.



Browser support for the File System Access API
[Screenshot taken by the author of this presentation from caniuse.com[2]]

1. File System Access API; https://wicg.github.io/file-system-access/
2. File System Access API Support; caniuse.com; 21 Jan 2025; https://caniuse.com/native-filesystem-api

Alternatives for Non-Chromium Browsers

- Drag and Drop API[1]
- Origin Private File System[2]
  - Similarities to File System Access API.
  - Broad Browser Support.
  - Not stored in memory.

	Input	Storage
Chromium Chrome, Edge, Opera	File System Access API	
Firefox Waterfox, SeaMonkey	Drag & Drop API	Origin Private File System
Safari	Drag & Drop API	WIP[3] 🚧
Safari on iOS	only Remote-Clone	Origin Private File System
Chrome on Android	only Remote-Clone	Origin Private File System

1. Drag and Drop API; https://developer.mozilla.org/en-US/docs/Web/API/HTML_Drag_and_Drop_API
2. Origin Private File System; https://developer.mozilla.org/en-US/docs/Web/API/File_System_API/Origin_private_file_system
3. Safari – Incomplete OPFS Implementation; https://bugs.webkit.org/show_bug.cgi?id=254726

Parsing Git Data

- Git Command Line[1]
  - Not portable (POSIX Shell etc.).
- Isomorphic-git Library[2] (JS)
  - No blame support.
- Gitoxide Library[3] (Rust)
  - No blame support (yet).
- Libgit2 Library[4] (C)
  - Mediocre blame performance[5].

1. Git Command Line; https://git-scm.com/
2. Isomorphic-git Library; Hilton, William; https://isomorphic-git.org/
3. Gitoxide Library; Thiel, Sebastian; https://github.com/Byron/gitoxide
4. Libgit2 Library; https://libgit2.org/
5. Libgit2 Performance Issue; https://github.com/libgit2/libgit2/issues/3027

WebAssembly[1]

- Modern binary instruction format.
- Widespread support, even within the browser sandbox.
- C / C++ / Rust / Go
- No Kernel ABI / Syscalls available.

Browser support for WebAssembly
[Screenshot taken by the author of this presentation from caniuse.com[2]]

1. WebAssembly; https://developer.mozilla.org/en-US/docs/WebAssembly
2. WebAssembly Support; caniuse.com; 21 Jan 2025; https://caniuse.com/wasm

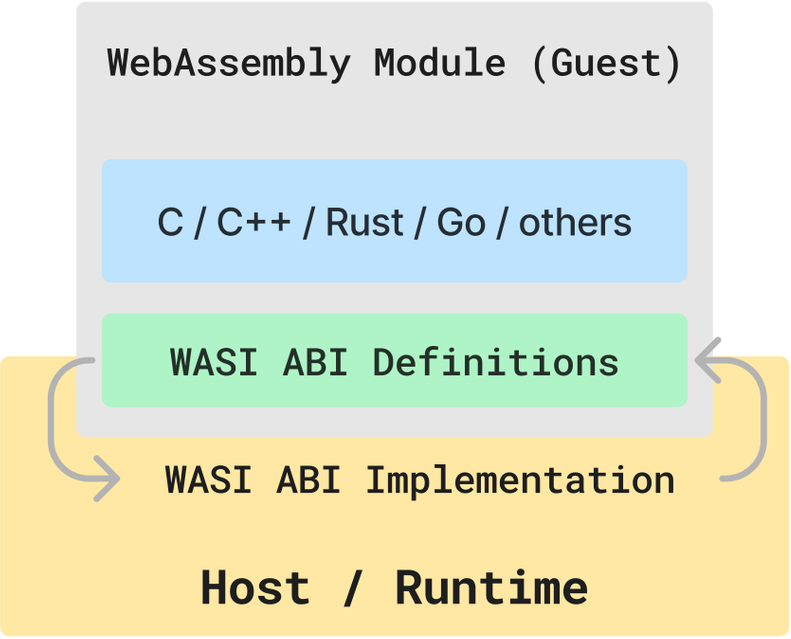
WebAssembly System Interface (WASI)[1]

- Used: preview1 (Version 0.1)
- Set of common API specifications.
- Command-Line Interface (CLI)-like behavior.

```typescript
interface WasiFunctions {
  args_sizes_get(argc: number, argv_buf_size: number): number;
  args_get(argv: number, argv_buf: number): number;

  fd_close(fd: number): number;
  fd_read(fd: number, iovs_ptr: number, iovs_len: number, nread_ptr: number): number;
  fd_write(fd: number, iovs_ptr: number, iovs_len: number, nwritten_ptr: number): number;
  fd_filestat_get(fd: number, filestat_ptr: number): number;
  // ... and many more
}
```

TypeScript Interface showcasing API of WASI

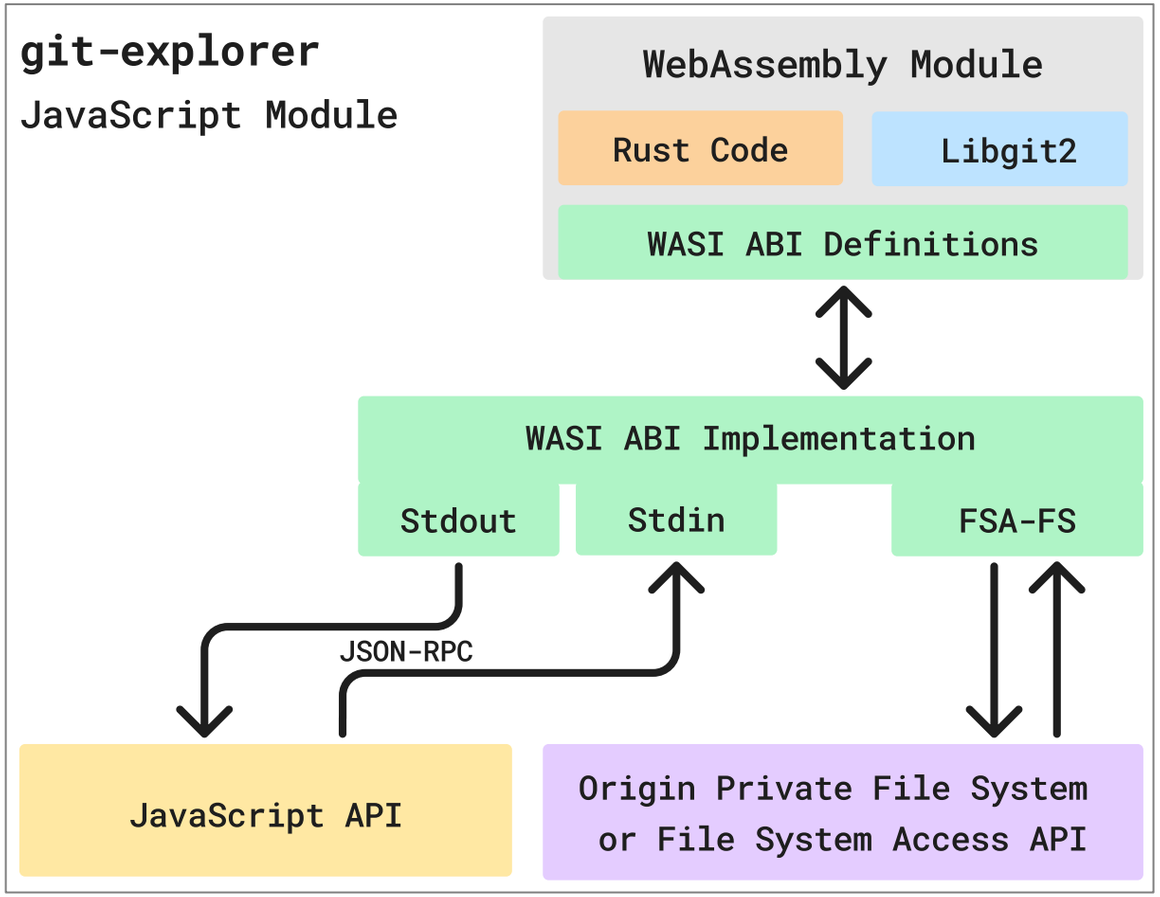


WASI Visualisation
[Image created by the author of this presentation]

1. WebAssembly System Interface (WASI); https://wasi.dev/

Challenges of Native Code in the Browser

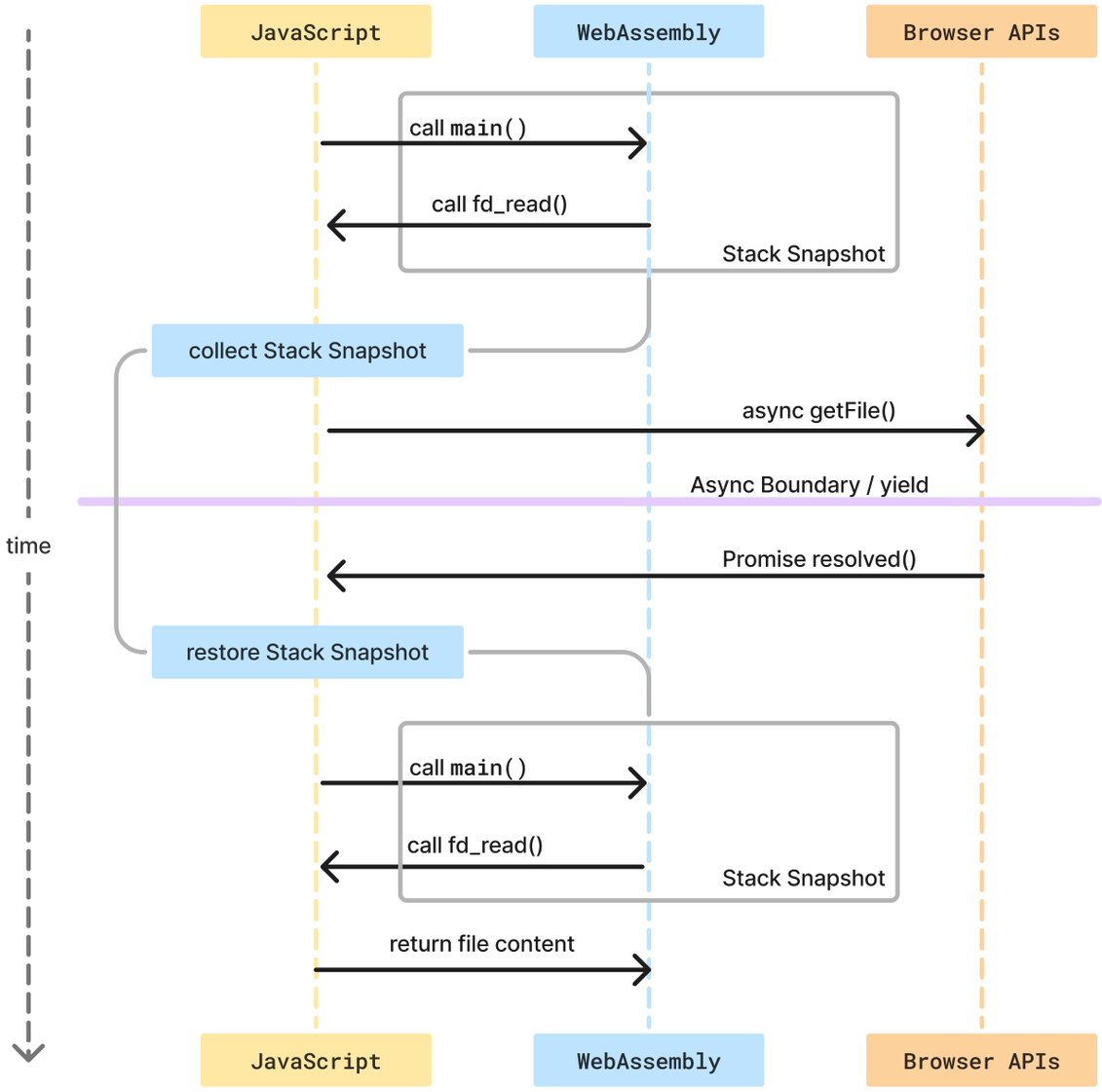
- File APIs
  - Native Code: Sync
  - Browser: Async
- JavaScript ↔ Rust Communication



Overview of git-explorer module
[Image created by the author of this presentation]

Async File I/O Interface

- Based on approach by A. Zakai[1].
- Using Asyncify[2] .
- Drawback:
  Significant increase of module size.



1. Zakai, Alon; https://kripken.github.io/blog/wasm/2019/07/16/asyncify.html
2. Binaryen Asyncify; https://github.com/WebAssembly/binaryen/blob/main/src/passes/Asyncify.cpp

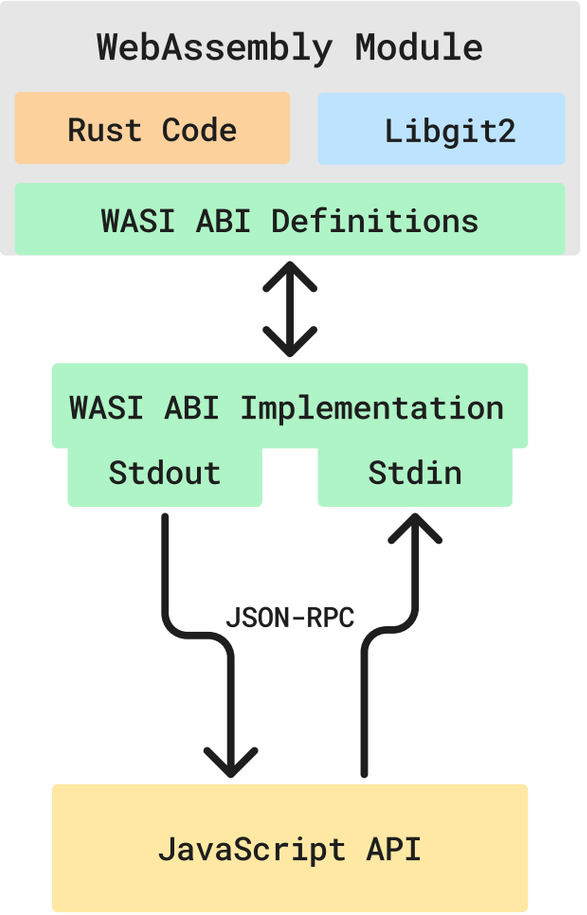
Simplified diagram showcasing Asyncify
[Image created by the author of this presentation]

JavaScript to Rust Command Interface

- JSON-RPC
- Use of FD stdin / stdout.
- Async wait for next job.

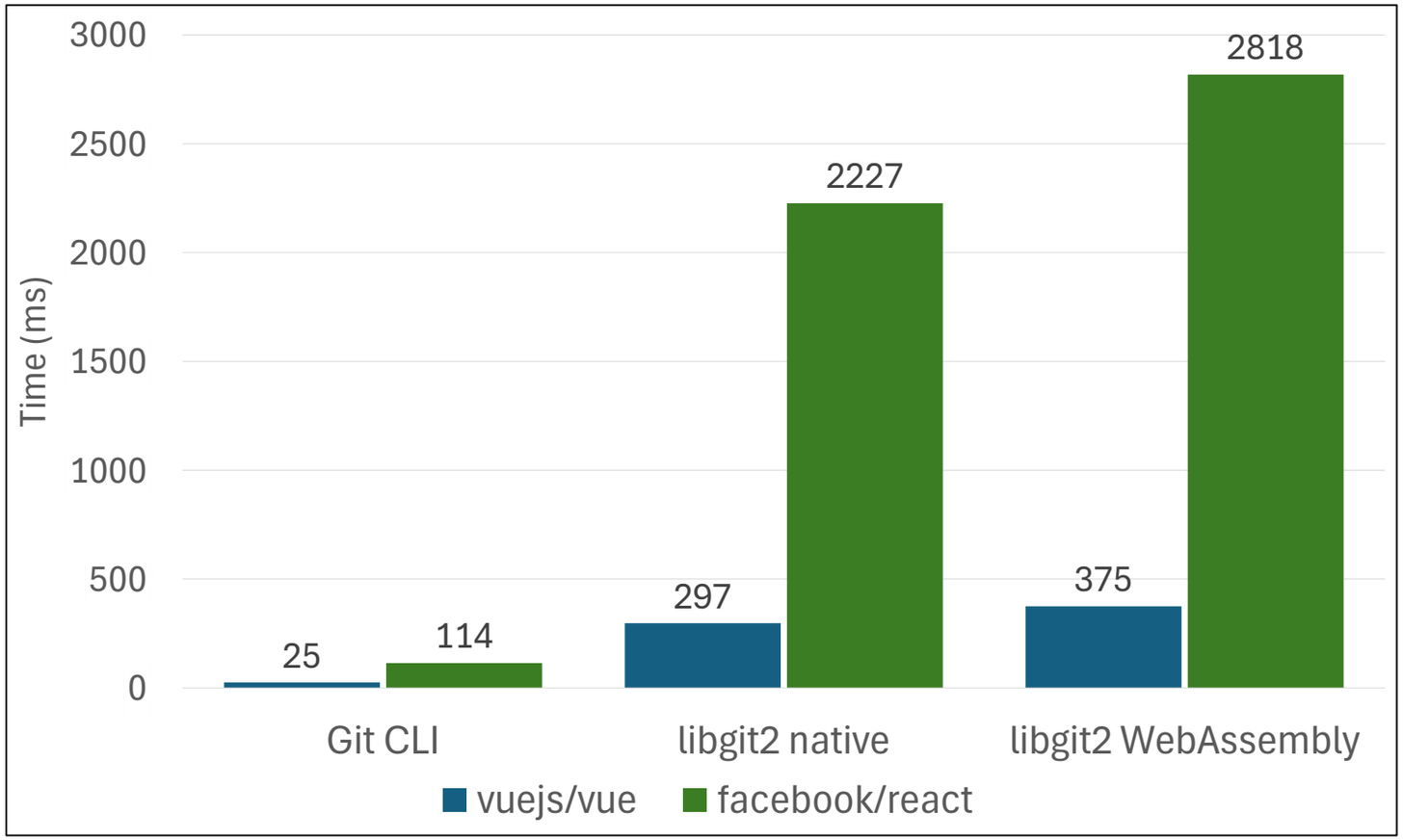
```
{
   jsonrpc: "2.0",
   id: 5,
   priority: 1,
   method: "get_blame",
   params: {
      path: "./package.json",
      rev: "v2",
      sinceRev: "v1",
   },
};
```

JSON-RPC Request used within Gizual.



Overview of JSON-RPC Usage
[Image created by the author of this presentation]

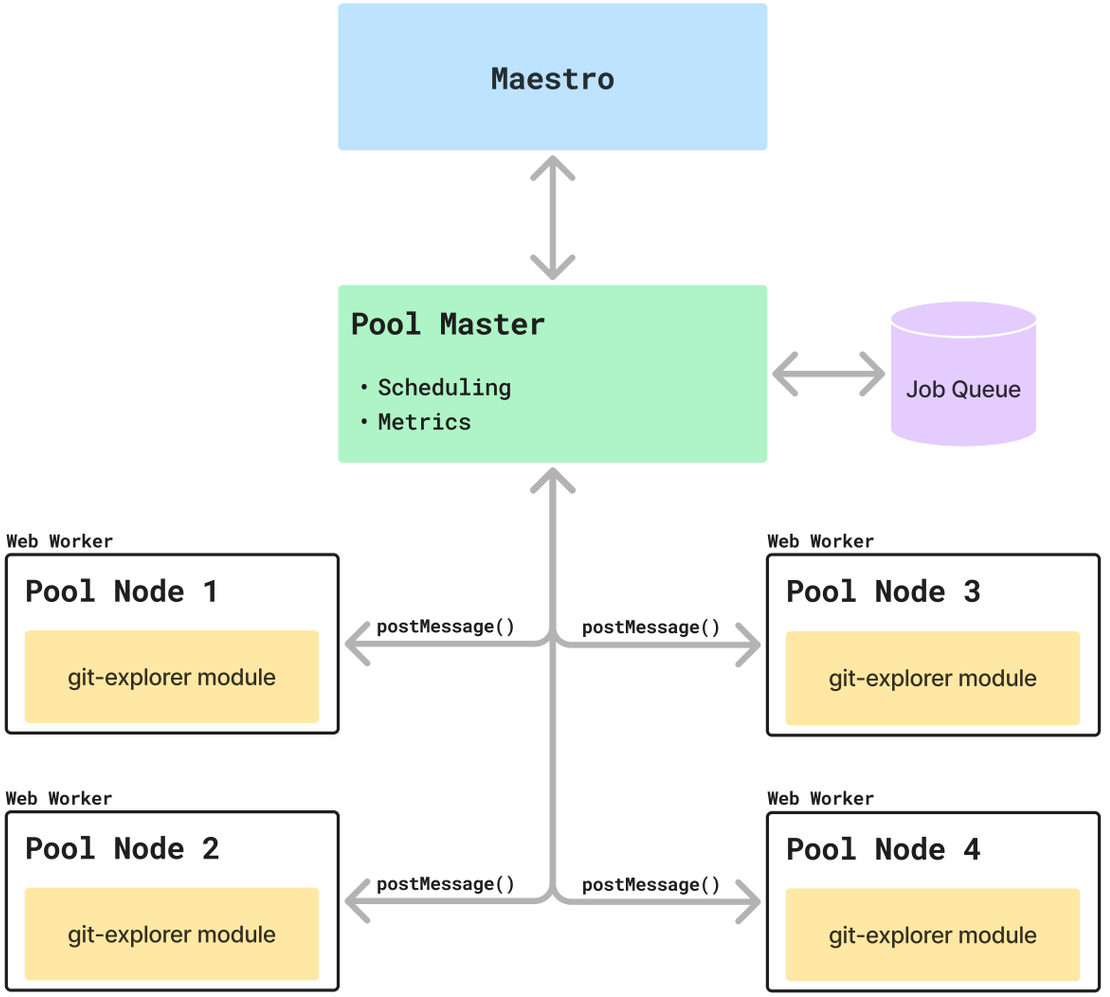
Performance



Comparison of blame performance; Blame was calculated for the single file package.json in both repositories on MacBook Pro 2023, M3 Max, 48 GB RAM. [Chart created by the author of this presentation]

Multi-Threading

- Pools of Web Workers for:
  - Rendering.
  - Data Processing.
- Prioritised Job Queue.
- Interval-based scheduling.



Pools of Web Workers
[Image created by the author of this presentation]

Showcase / Demo

Limitations

- Safari desktop support[1]
- More efficient memory management.
- Support even very large repositories:

Repository	Size	.git/ Size	First Commit	# Commits	🖥️ Desktop	📱 Mobile
Vue 2	small	35 MB	10 Apr 2016	6,694	✅	✅
React	medium	657 MB	29 May 2013	25,873	✅	🚧
VS Code	large	1,025 MB	13 Nov 2015	134,419	✅	🚧
Linux Kernel	very large	5,836 MB	16 Apr 2005	1,310,317	❓	❓

1. Safari – Incomplete OPFS Implementation; https://bugs.webkit.org/show_bug.cgi?id=254726

Thank You!

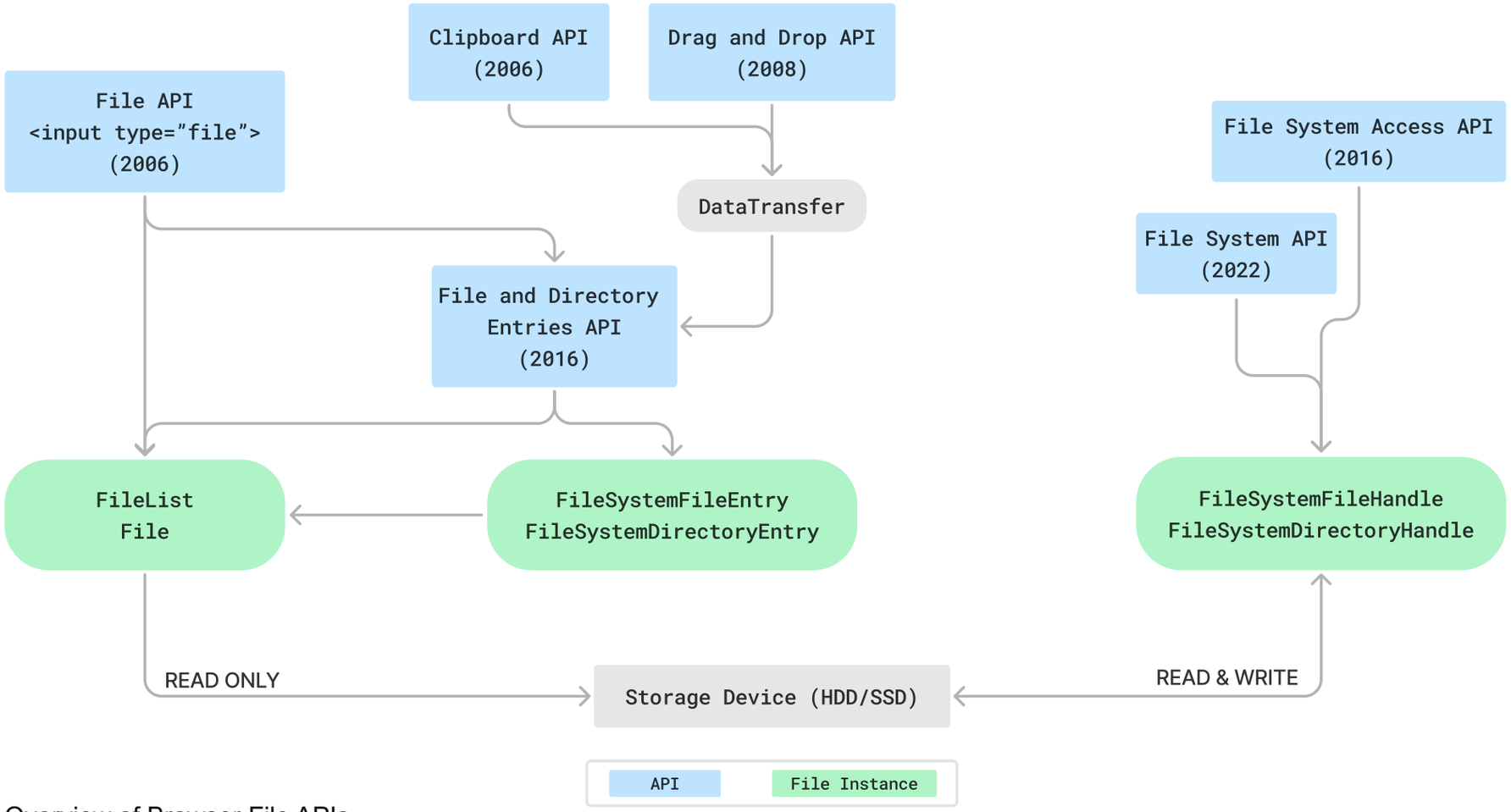
gizual.com

Future Work

- Use WASI version 0.2 (preview2).
- Use JS promise integration in WebAssembly[1] (when available).
- Enhance performance of git blame (e.g. gitoxide).
- Improve file I/O caching / use SharedArrayBuffer.
- Optimise size of Web Worker Pools.
- Harvest more data than just blame information.

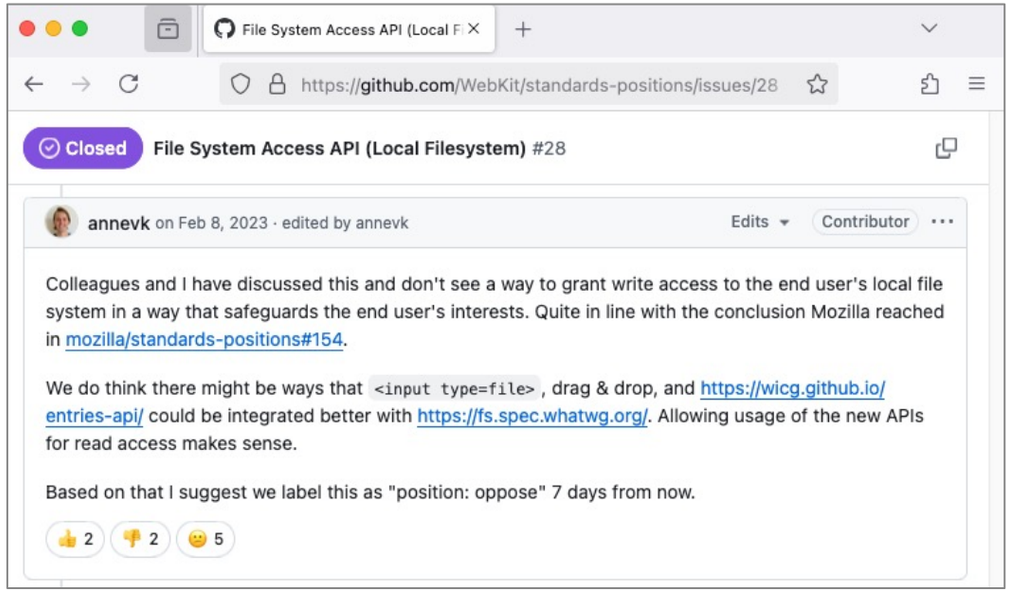
1. JavaScript-Promise Integration Proposal; https://github.com/WebAssembly/js-promise-integration/blob/main/proposals/js-promise-integration/Overview.md

File I/O within the Browser

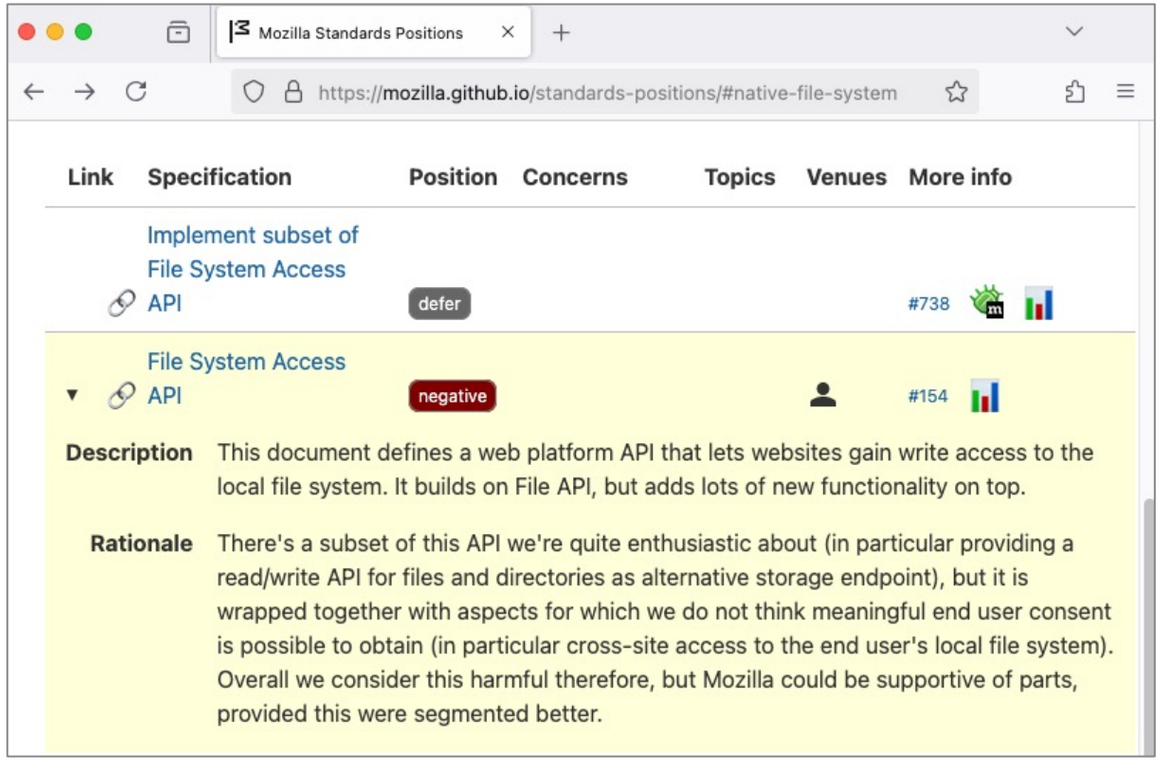


Overview of Browser File APIs
[Image created by the author of this presentation]

File System Access API – State in other Browsers



Position of WebKit on File System Access API[1]
[Screenshot created by the author of this presentation]



Position of Firefox on File System Access API[2]
[Screenshot created by the author of this presentation]

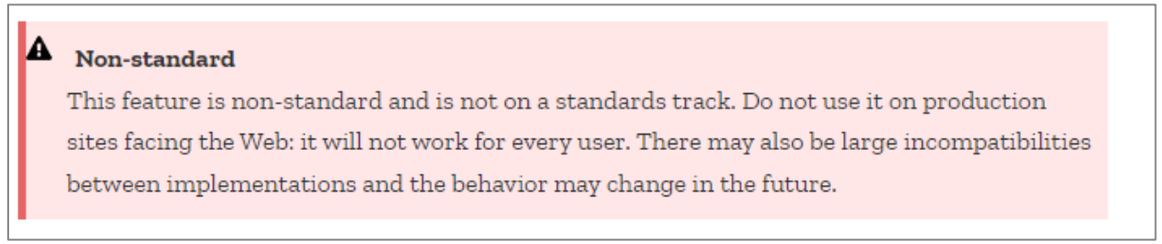
1. Github discussion about File System Access API in WebKit; https://github.com/WebKit/standards-positions/issues/28
2. Mozilla Standards Positions – File System Access API; https://mozilla.github.io/standards-positions/#native-file-system

File API / File Input Element

- Designed for single files.
- Lags if many (1000+) files are selected.

```
5   <input type="file" webkitdirectory />
6   <script>
7     const picker = document.querySelector('input');
8     picker.addEventListener('change', (e) => {
9       for (let file of picker.files) {
10        console.log(file.webkitRelativePath);
11        // 'repo/README.md'
12        // 'repo/.git/config'
13        // 'repo/.git/HEAD'
14        // 'repo/.git/objects/22/ee7b7ca19c25..'
15        //  and many more ...
16      }
17    });
18  </script>
```

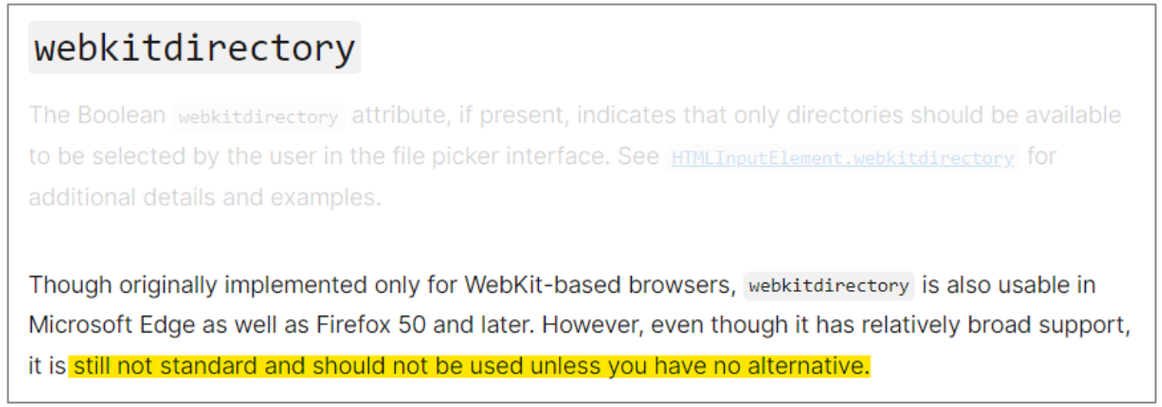
Simple Code Example of File Input Element



Non-standard

This feature is non-standard and is not on a standards track. Do not use it on production sites facing the Web: it will not work for every user. There may also be large incompatibilities between implementations and the behavior may change in the future.

Warning about *webkitdirectory* attribute
[Screenshot created by the author of this presentation from UDN web docs[1]]



`webkitdirectory`

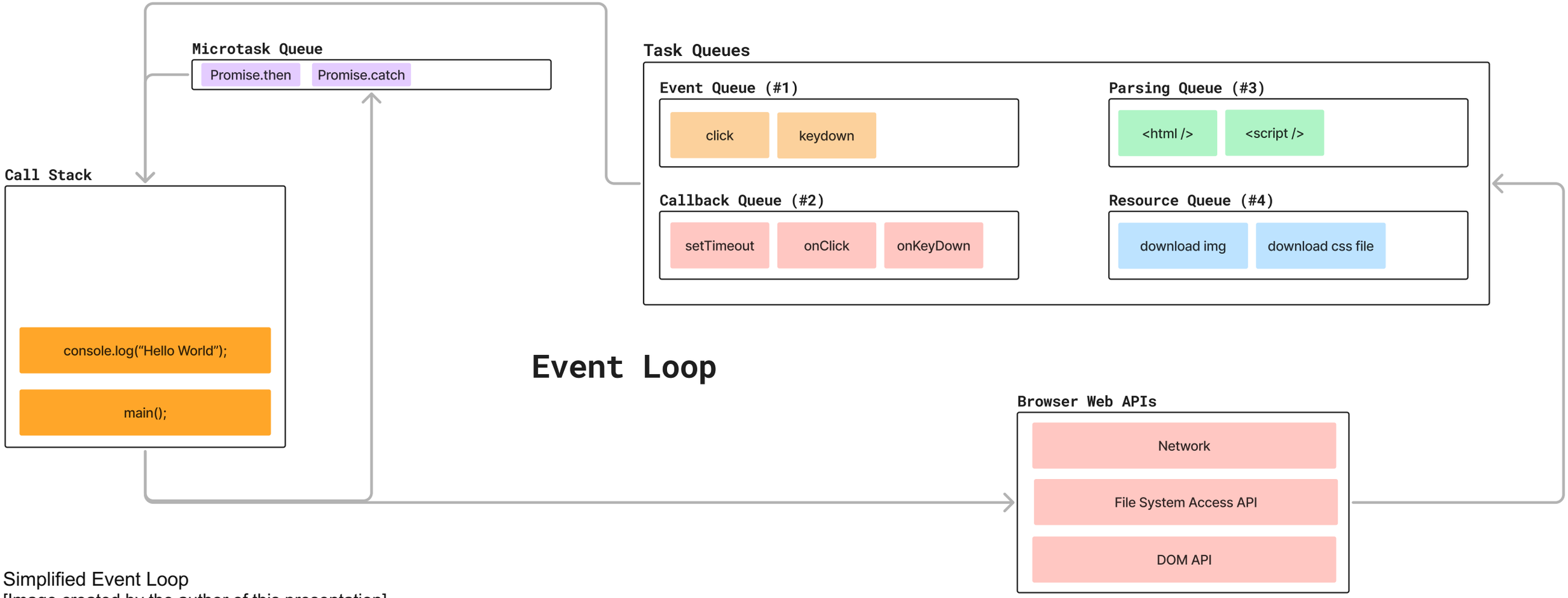
The Boolean `webkitdirectory` attribute, if present, indicates that only directories should be available to be selected by the user in the file picker interface. See `HTMLInputElement.webkitdirectory` for additional details and examples.

Though originally implemented only for WebKit-based browsers, `webkitdirectory` is also usable in Microsoft Edge as well as Firefox 50 and later. However, even though it has relatively broad support, it is still not standard and should not be used unless you have no alternative.

Documentation of *webkitdirectory* attribute
[Screenshot created by the author of this presentation from developer.mozilla.org[2]]

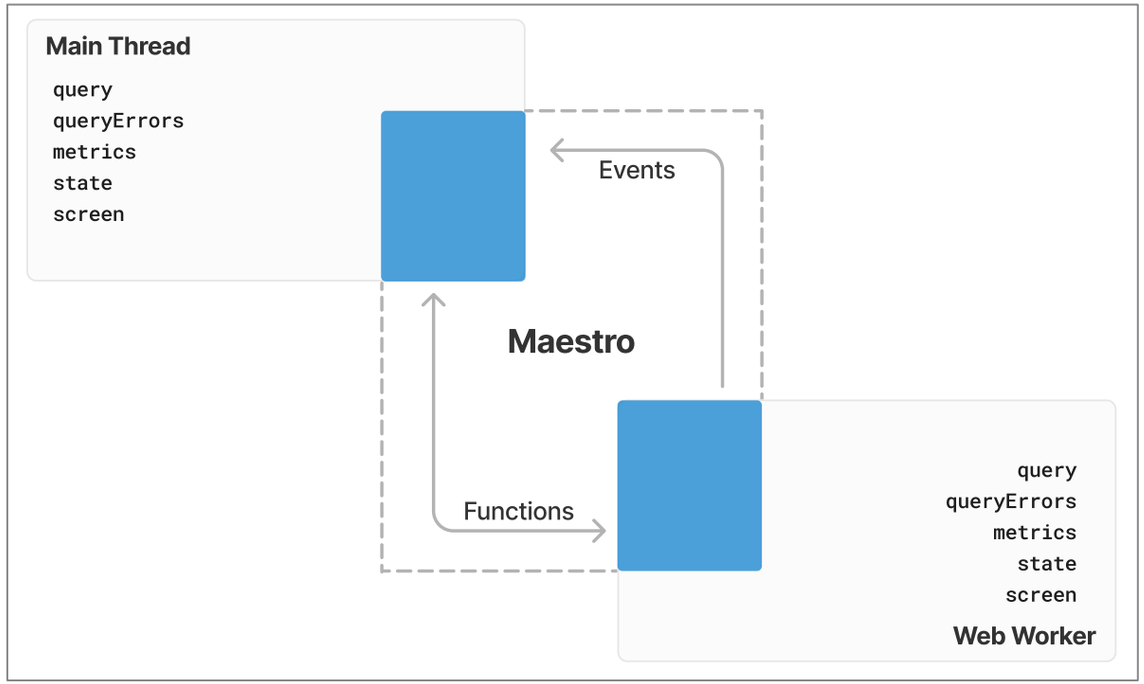
1. UDN web docs; HTMLInputElement.webkitdirectory; https://udn.realityripple.com/docs/Web/API/HTMLInputElement/webkitdirectory
2. Mozilla Documentation – File Input Element; https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/file

Browser Event Loop

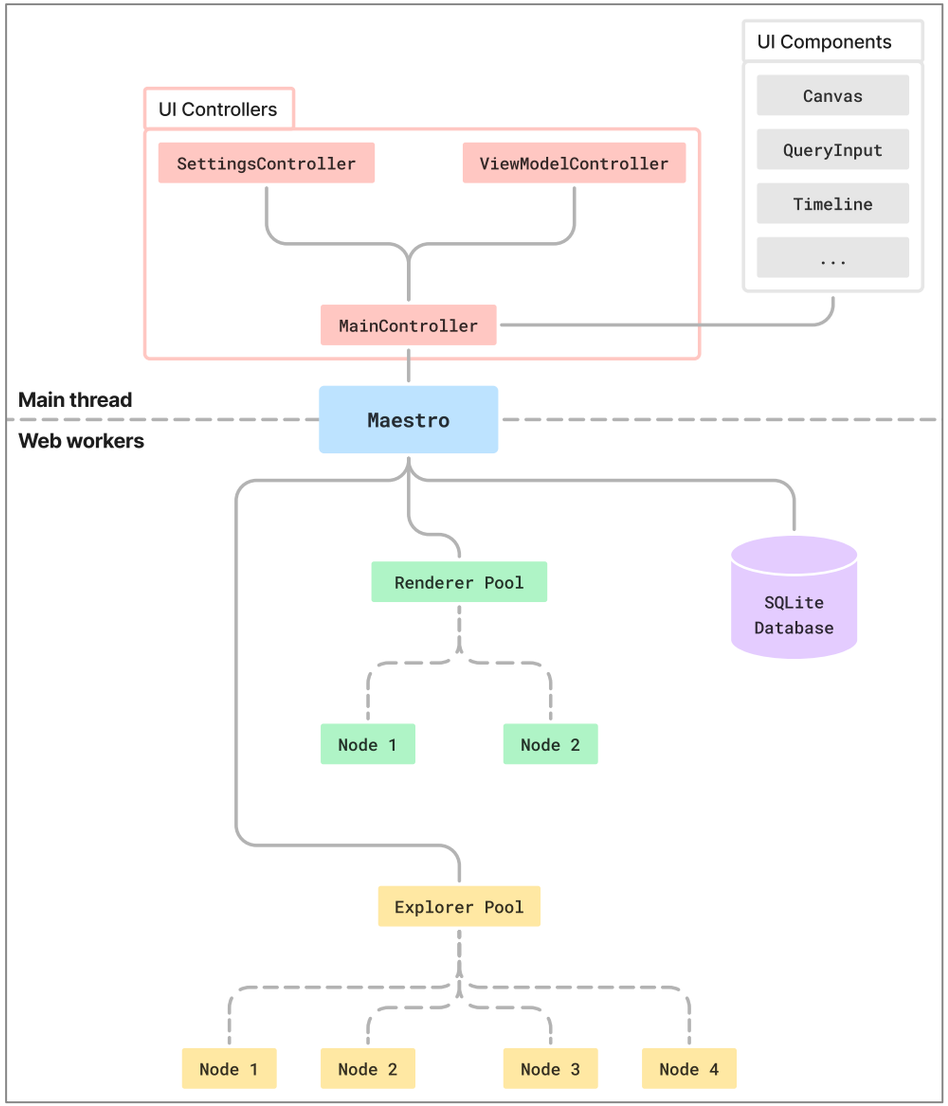


Simplified Event Loop
[Image created by the author of this presentation]

Architecture – Maestro



Maestro Controller
[Image jointly created by the author of this presentation and Andreas Steinkellner[1]]

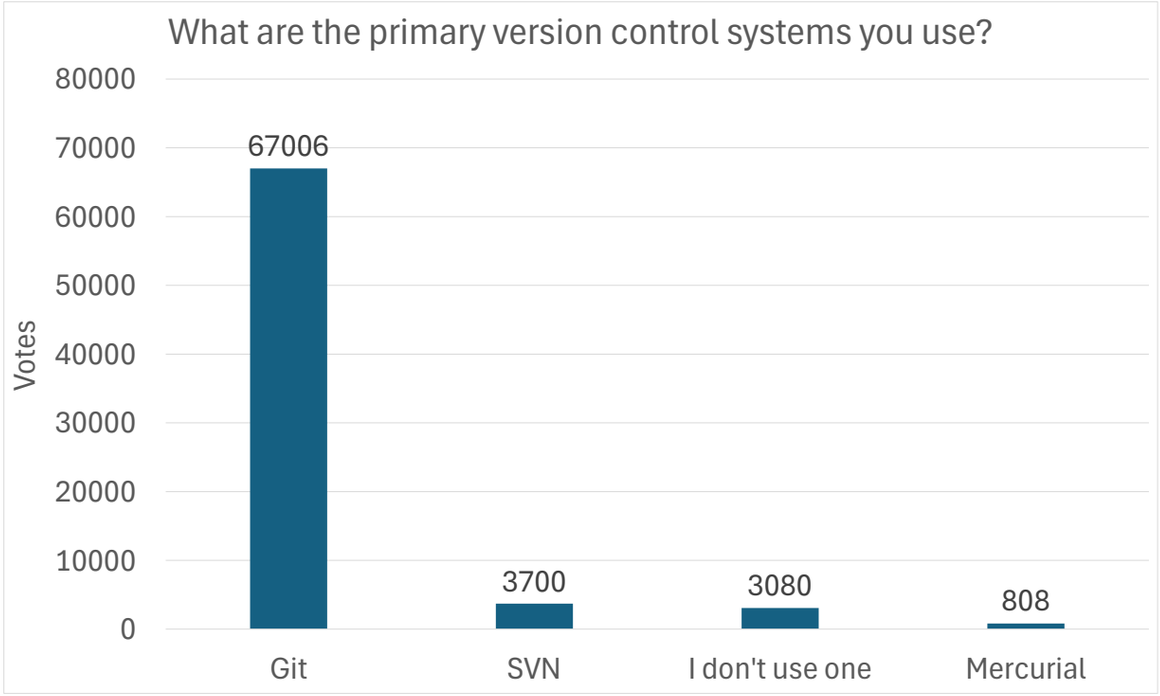


Gizual Architecture
[Image jointly created by the author of this presentation and Andreas Steinkellner[1]]

1. Gizual User Interface: Browser-Based Visualisation for Git Repositories; Andreas Steinkellner; Master's Thesis; Graz University of Technology, Austria; 09 Dec 2024 https://ftp.isds.tugraz.at/pub/theses/asteinkellner-2024-msc.pdf

Source-Code Version Control Systems

- Team collaboration.
- History of changes.
- Branching and merging for parallel development.
- Traceability of the author.

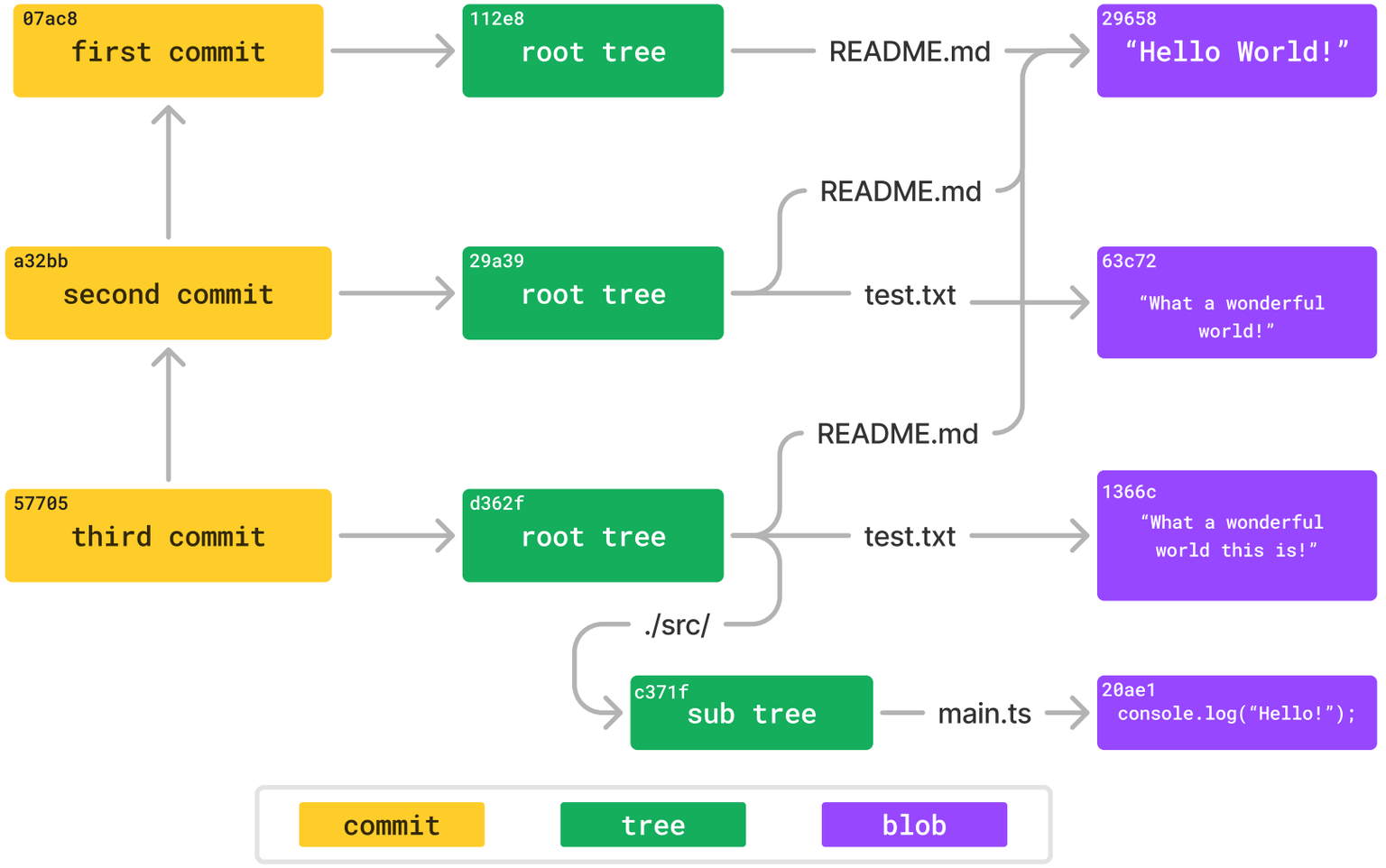


Survey results for the question “What are the primary version control systems you use?” from the Stack Overflow Developer Survey 2022. [based on Data of the Public 2022 Stack Overflow Developer Survey Results, licensed under the Open Database License (ODbL)][1]

1. Stack Overflow Developer Survey 2022
https://survey.stackoverflow.co/2022/

Git Internals

- Directed Acyclic Graph
- Node: Commits
- Edges: Ref to parent(s)



Git Internals: Overview of Structure
[Image created by the author of this presentation]