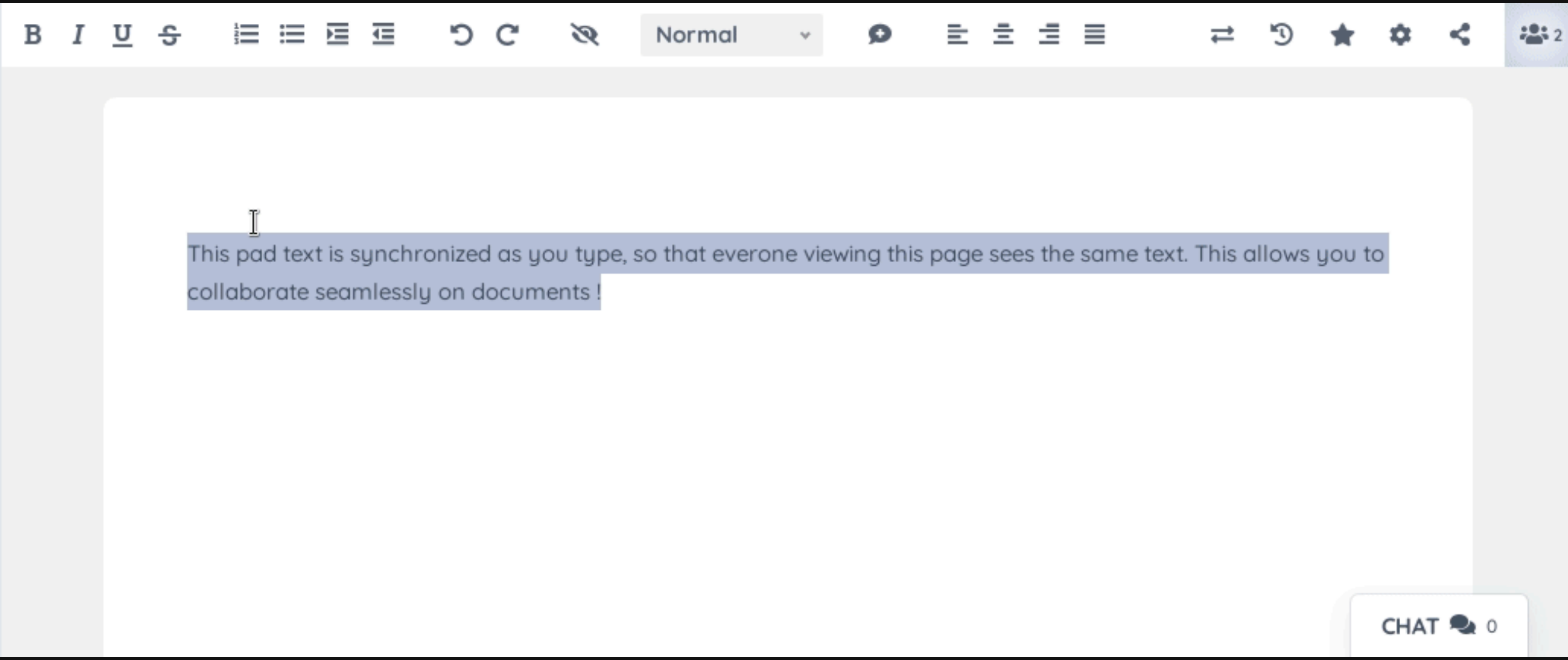
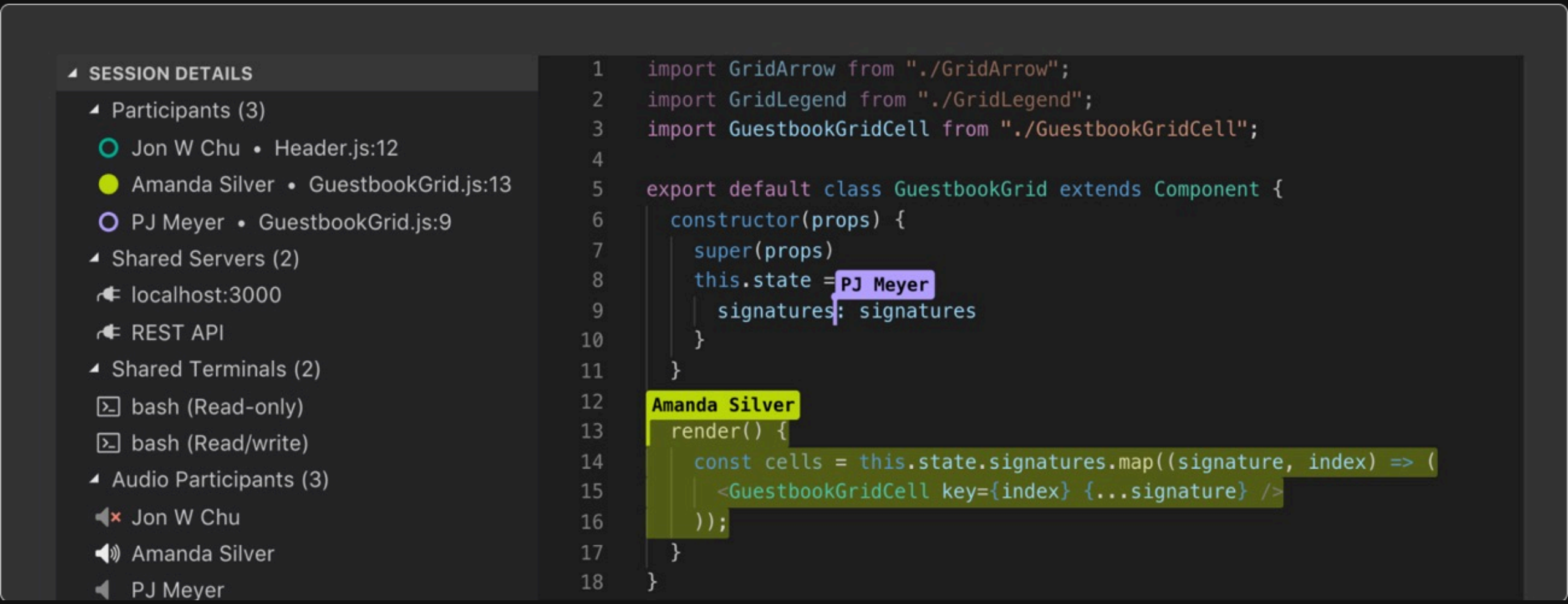
Ethersync

Real-time collabo|ration in your text editor!

FOSDEM 2025 — blinry


Normal
2
This pad text is synchronized as you type, so that everone viewing this page sees the same text. This allows you to collaborate seamlessly on documents !
CHAT 0


SESSION DETAILS
Participants (3)
Jon W Chu • Header.js:12
Amanda Silver • GuestbookGrid.js:13
PJ Meyer • GuestbookGrid.js:9
Shared Servers (2)
localhost:3000
REST API
Shared Terminals (2)
bash (Read-only)
bash (Read/write)
Audio Participants (3)
Jon W Chu
Amanda Silver
PJ Meyer
import GridArrow from "./GridArrow";
import GridLegend from "./GridLegend";
import GuestbookGridCell from "./GuestbookGridCell";
export default class GuestbookGrid extends Component {
constructor(props) {
super(props)
this.state =
PJ Meyer
signatures: signatures
}
}
Amanda Silver
render() {
const cells = this.state.signatures.map((signature, index) => (
<GuestbookGridCell key={index} {...signature} />
));
}
}

Ethersync

An editor-agnostic, open-source tool + protocol for editing local files together

blinry

zormit

Making a connection

Person 1:

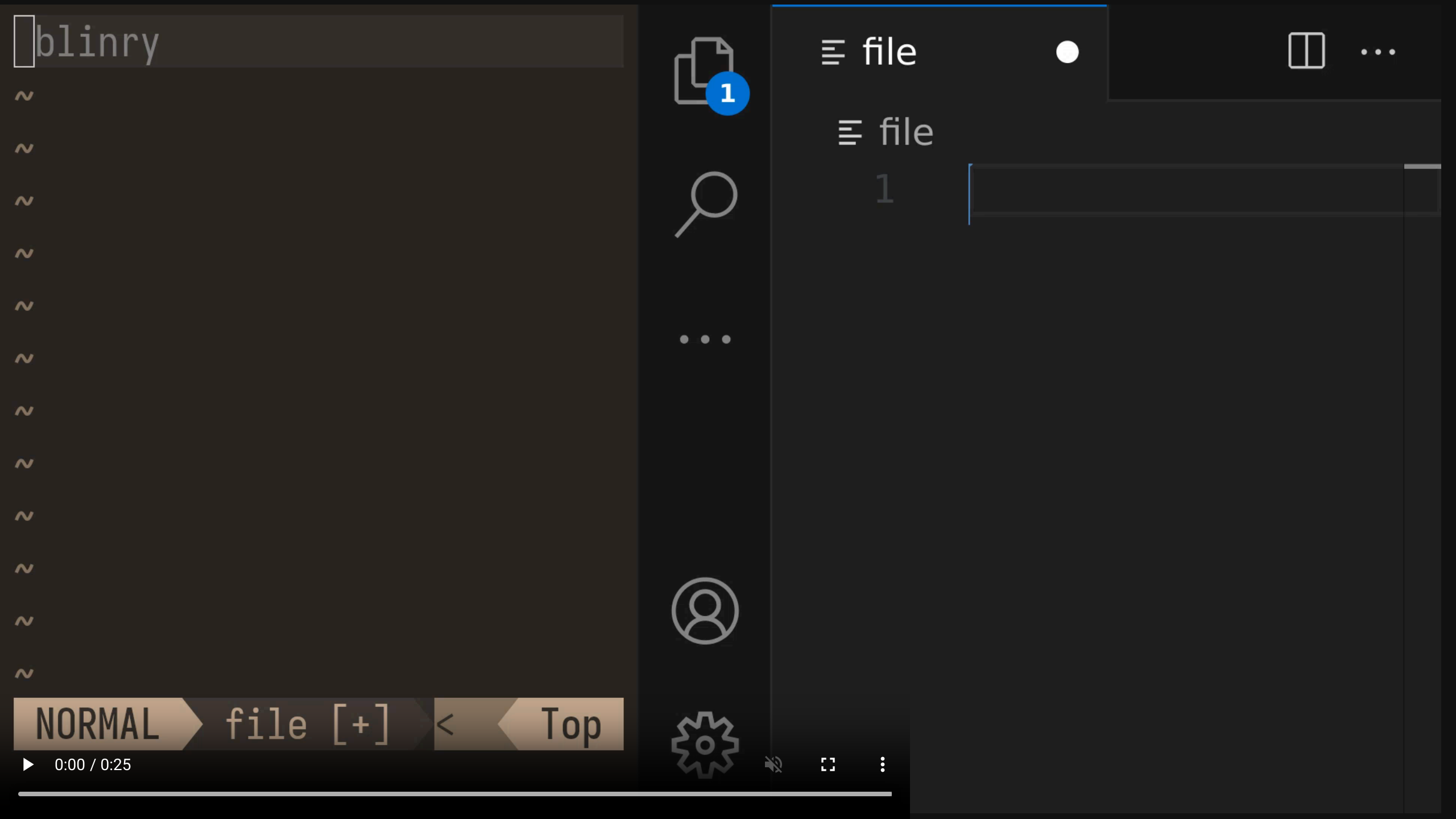
```
$ ethersync daemon
    Others can connect with:
    ethersync daemon --peer /ip4/192.168.178.36/tcp/4242
```

Other people:

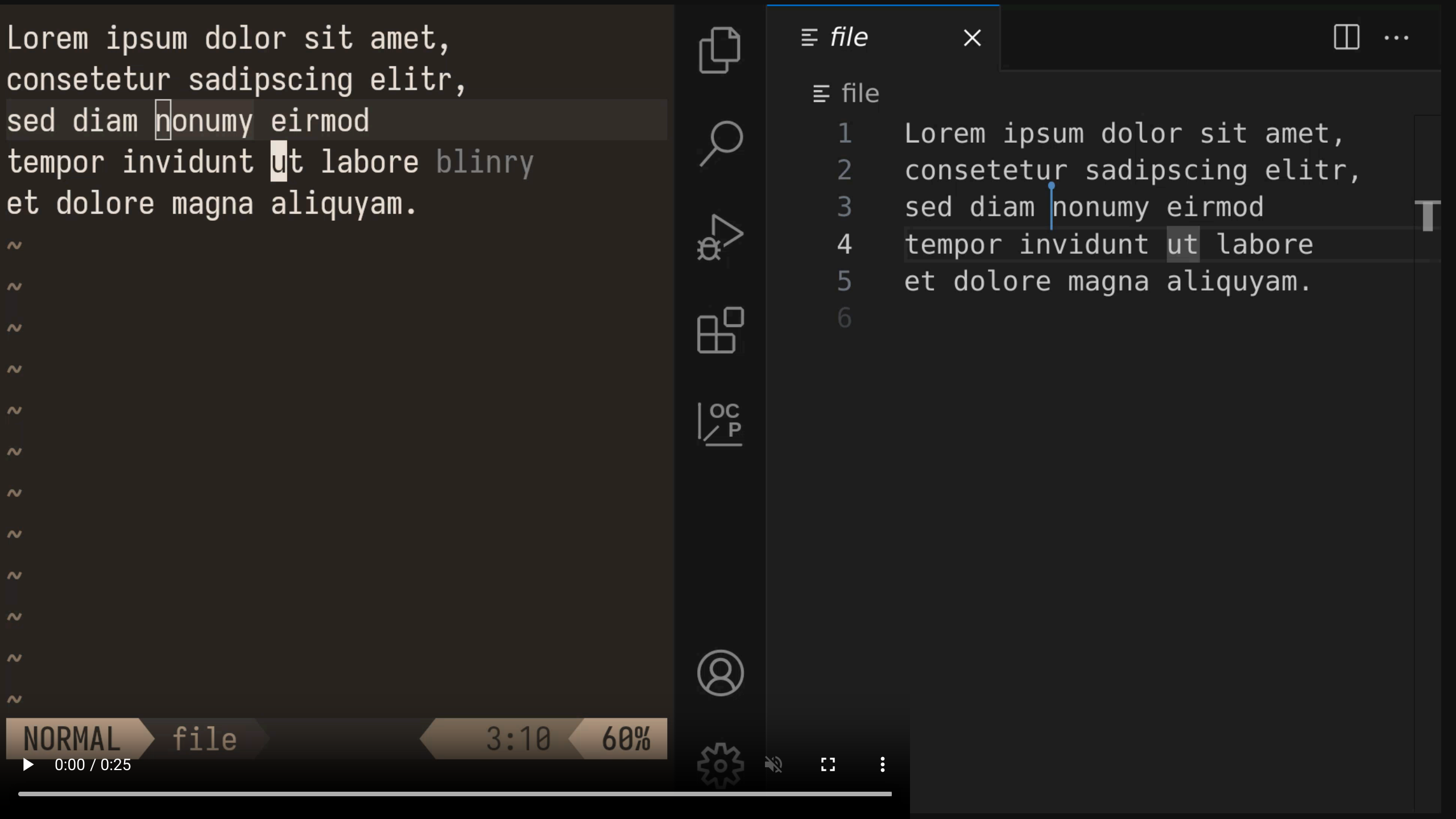
```
$ ethersync daemon --peer /ip4/192.168.178.36/tcp/4242
```

Directories are now synchronized! ✨

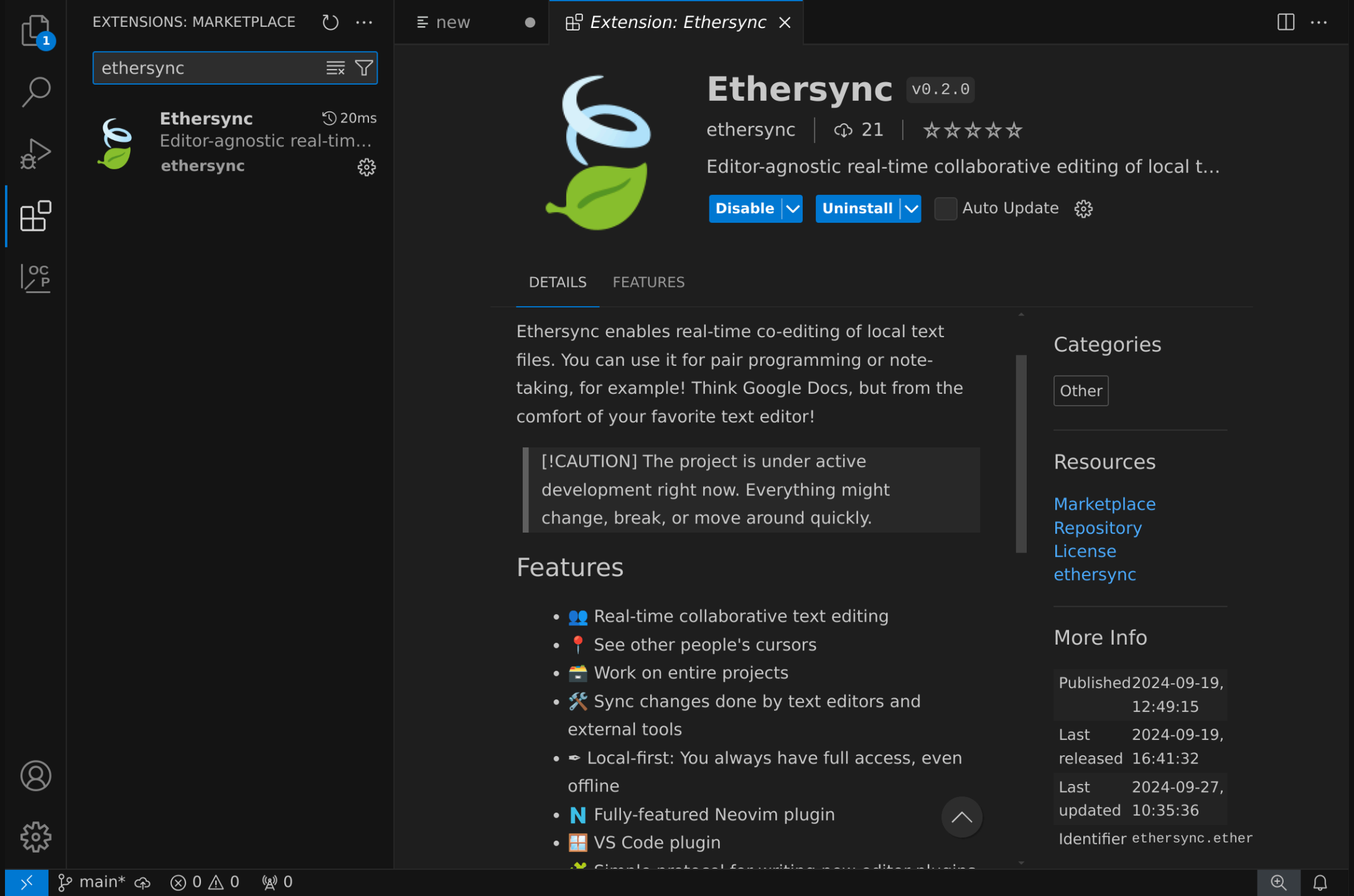
Real-time collaboration


blinry
NORMAL
file [+]
<
Top
0:00 / 0:25
file
file
1

📍 See other people's cursors

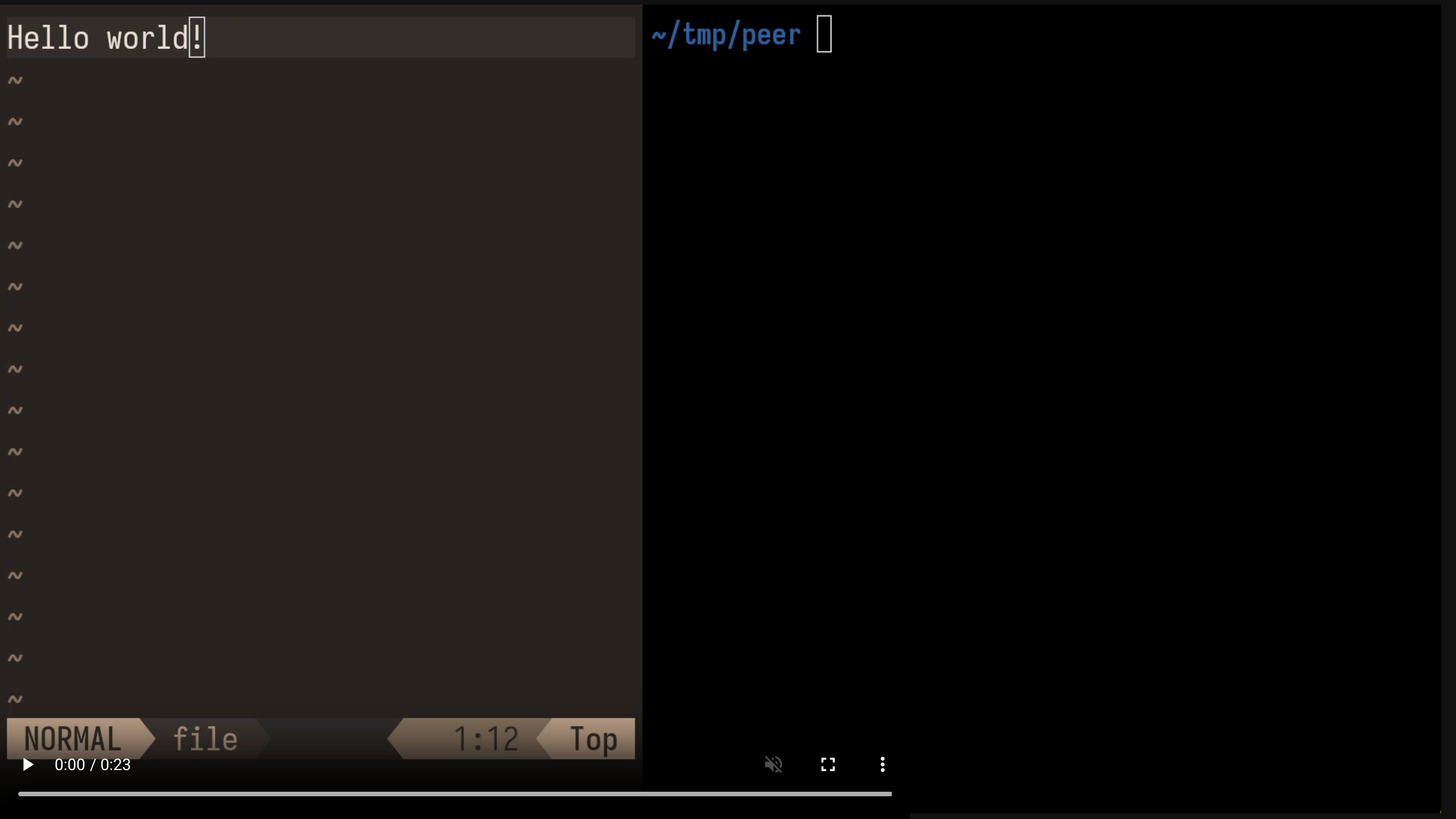

Lorem ipsum dolor sit amet,
consetetur sadipscing elitr,
sed diam nonumy eirmod
tempor invidunt ut labore blinry
et dolore magna aliquyam.
NORMAL file 3:10 60%
file
file
1 Lorem ipsum dolor sit amet,
2 consetetur sadipscing elitr,
3 sed diam nonumy eirmod
4 tempor invidunt ut labore
5 et dolore magna aliquyam.
6
0:00 / 0:25

🧩 Plugins for Neovim and VS Code


EXTENSIONS: MARKETPLACE
ethersync
Ethersync
20ms
Editor-agnostic real-tim...
ethersync
new
Extension: Ethersync
Ethersync
v0.2.0
ethersync
21
Editor-agnostic real-time collaborative editing of local t...
Disable
Uninstall
Auto Update
DETAILS
FEATURES
Ethersync enables real-time co-editing of local text files. You can use it for pair programming or note-taking, for example! Think Google Docs, but from the comfort of your favorite text editor!
[!CAUTION] The project is under active development right now. Everything might change, break, or move around quickly.
Features
👥 Real-time collaborative text editing
📍 See other people's cursors
🗃️ Work on entire projects
🛠️ Sync changes done by text editors and external tools
✒ Local-first: You always have full access, even offline
Fully-featured Neovim plugin
VS Code plugin
Categories
Other
Resources
Marketplace
Repository
License
ethersync
More Info
Published 2024-09-19, 12:49:15
Last released 2024-09-19, 16:41:32
Last updated 2024-09-27, 10:35:36
Identifier ethersync.ether
main*
0 0
0

Synchronize entire projects

🛠️ Use external tools


Hello world!
~/tmp/peer
NORMAL file 1:12 Top
0:00 / 0:23

✒️ Local-first: You always have full access, even offline

Build local-first software

Automerge is a library of data structures for building collaborative applications.

Get started

Automatic merging

Automerge is a Conflict-Free Replicated Data Type (CRDT), which allows concurrent changes on different devices to be merged automatically without requiring any central server.

Network-agnostic

Use any connection-oriented network protocol: client-server, peer-to-peer, or local. Or use unidirectional messaging: send an Automerge file as an email attachment or store it on a file server.

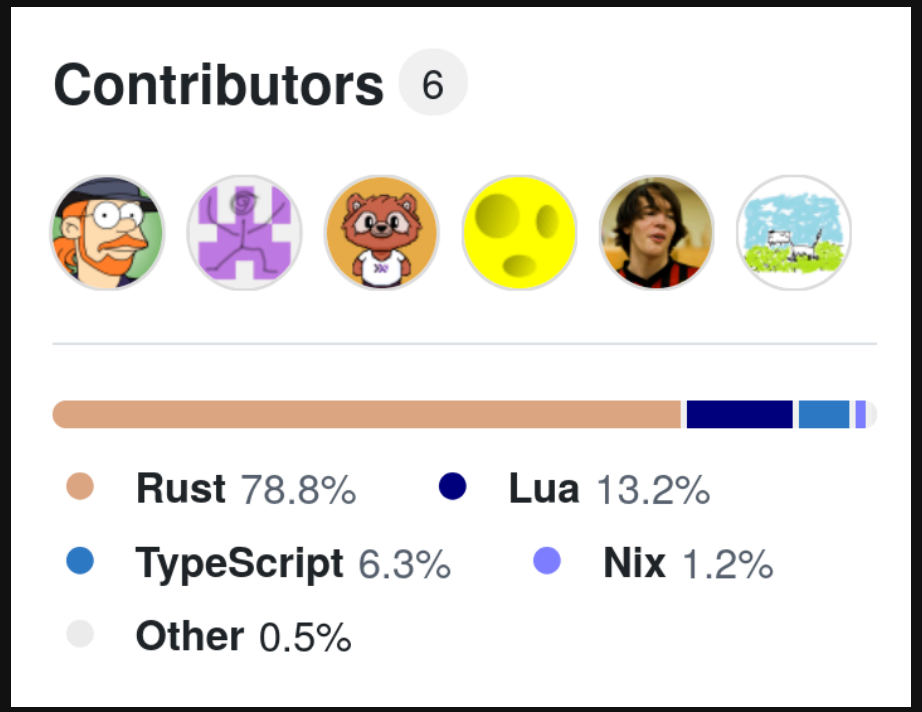
Portable

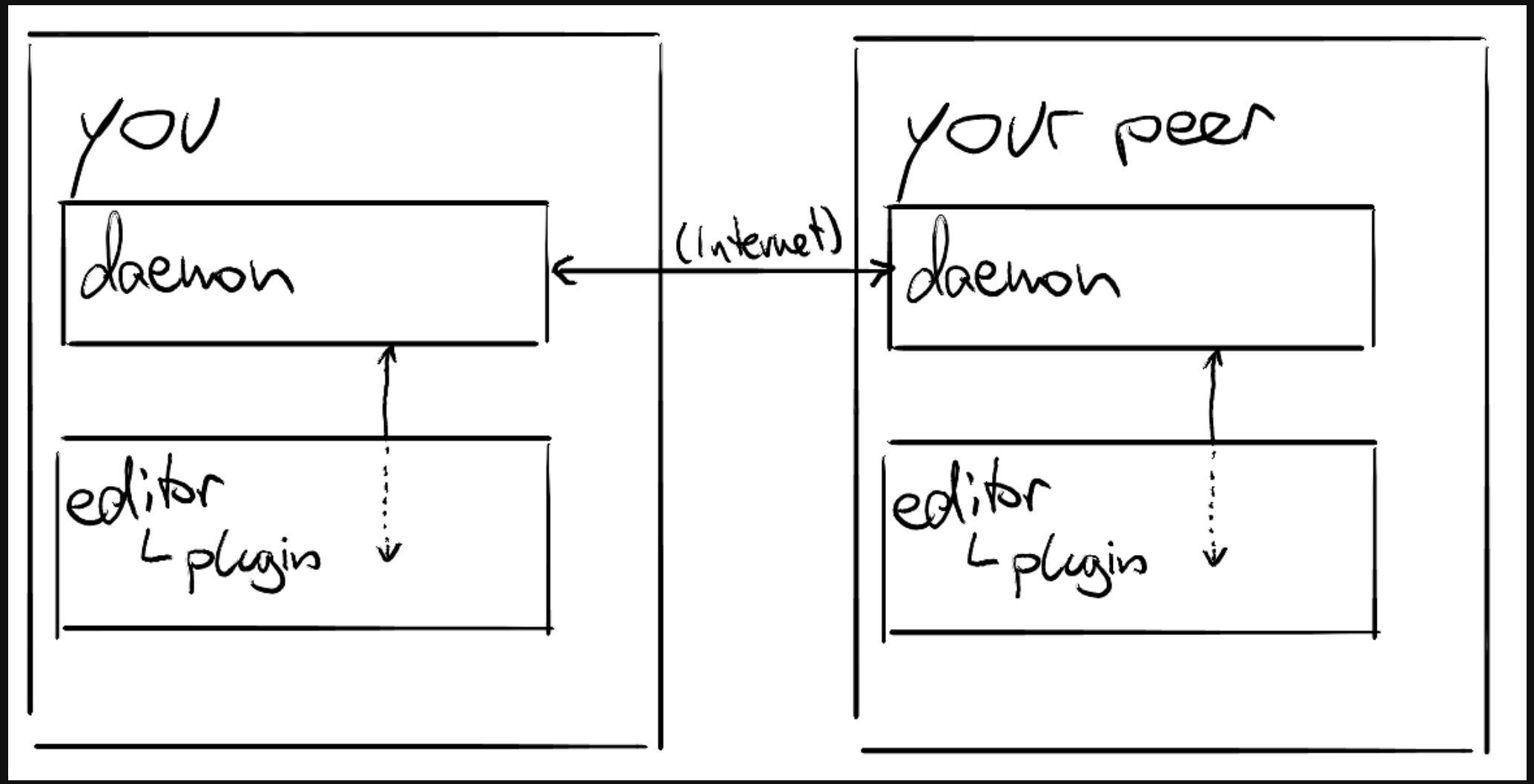
Implemented in JavaScript and Rust, with FFI bindings across platforms including iOS, Electron, Chrome, Safari, Edge, Firefox, and more.

Peer-to-peer connections

How does Ethersync work?

https://github.com/ethersync/ethersync




you
daemon
editor
plugin
(internet)
your peer
daemon
editor
plugin

Daemon ↔️ Daemon

- libp2p connection
  - Using their "private network" feature
- Inside: raw Automerge sync protocol

Daemon ↔️ Editor

- Our goal: make it as easy as possible to write new editor plugins
- Editors open a JSON-RPC connection to `ethersync client`
- Defined a small syncing prototol
  - A bit like LSP
  - "One-sided Operational Transform" (editor can be lazy)

Ethersync
@ethersync@chaos.social

Follow

Speak JSON-RPC with `ethersync client`.

When editor opens/closes a file, send "open"/"close" {"uri": "file:///..."}.

For opened files, set editor + daemon revision to 0.

On user edits, send

```
"edit" {
 "uri": "file:///...",
 "revision": daemon_revision,
 "delta": [{
  "range": {"start": p1, "end": p2},
  "replacement": text
 }]
}
```

with positions as {"line": l, "character": c} & increment editor revision.

Apply incoming edits if they match your editor revision, then increment daemon revision.

September 6, 2024 at 7:50 PM · Edited September 26, 2024 at 6:35 PM · 3 · 3 · 3 · Open in web

What Ethersync is not

- A replacement for Git
- A system for rich-text editing
- Production-ready software

Open question: How to make connections across local networks?

- Current approach: libp2p
- Problem: The "private network" feature doesn't seem to play nice with relays/hole punching
- Dream: `magic-wormhole`-style connection making

How to get involved

1 Write an editor plugin!

1️⃣ Write an editor plugin!

Editor	Status	Who is working on it?
Neovim	✅	Official plugin
VS Code	✅	Official plugin

1️⃣ Write an editor plugin!

Editor	Status	Who is working on it?
Neovim	✅	Official plugin
VS Code	✅	Official plugin
IntelliJ	🛠️	@schrieveslaach (#257)
Emacs	🛠️	@cosmicexplorer / @_sohalt (#77)

1️⃣ Write an editor plugin!

Editor	Status	Who is working on it?
Neovim	✅	Official plugin
VS Code	✅	Official plugin
IntelliJ	🛠️	@schrieveslaach (#257)
Emacs	🛠️	@cosmicexplorer / @_sohalt (#77)
Eclipse	❌	You?
Web	❌	You?

1 Write an editor plugin!

Editor	Status	Who is working on it?
Neovim	✅	Official plugin
VS Code	✅	Official plugin
IntelliJ	🛠️	@schrieveslaach (#257)
Emacs	🛠️	@cosmicexplorer / @_sohalt (#77)
Eclipse	❌	You?
Web	❌	You?
Nano	❌	…You?

2️⃣ Help improve the editor sync protocol!

- Anyone with LSP experience?
- Naming suggestions?
- Dream: Form a small group of people to refine & standardize it!
- Goal: Make editor plugins usable with other collaborative applications

3️⃣ Try Ethersync and give us feedback!

- What's unexpected/annoying/broken?
- What are you missing?
- Interest in regular online user meetups?

What to take home: Desire!

Ethersync

🌐 ethersync.github.io

🐘 @ethersync@fosstodon.org

 github.com/ethersync/ethersync

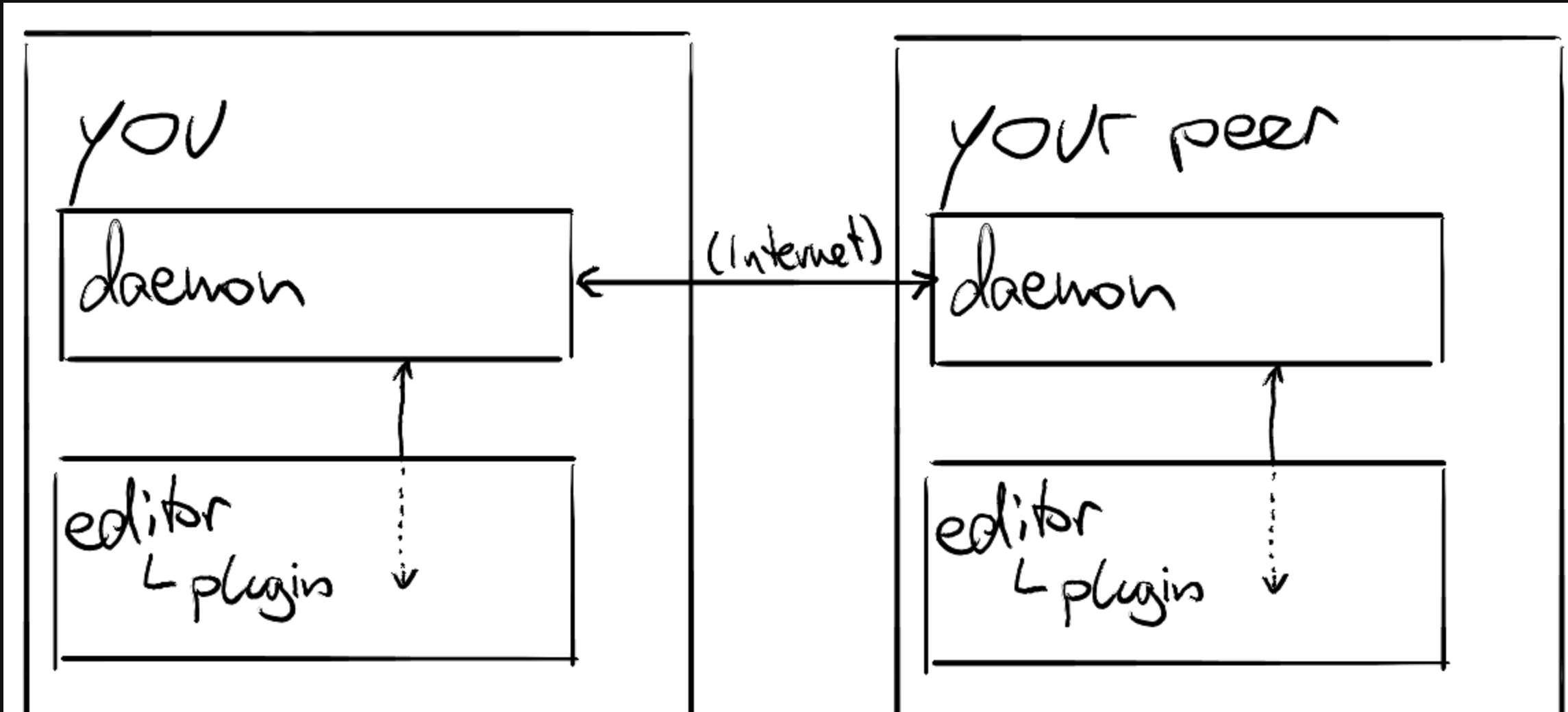
blinry

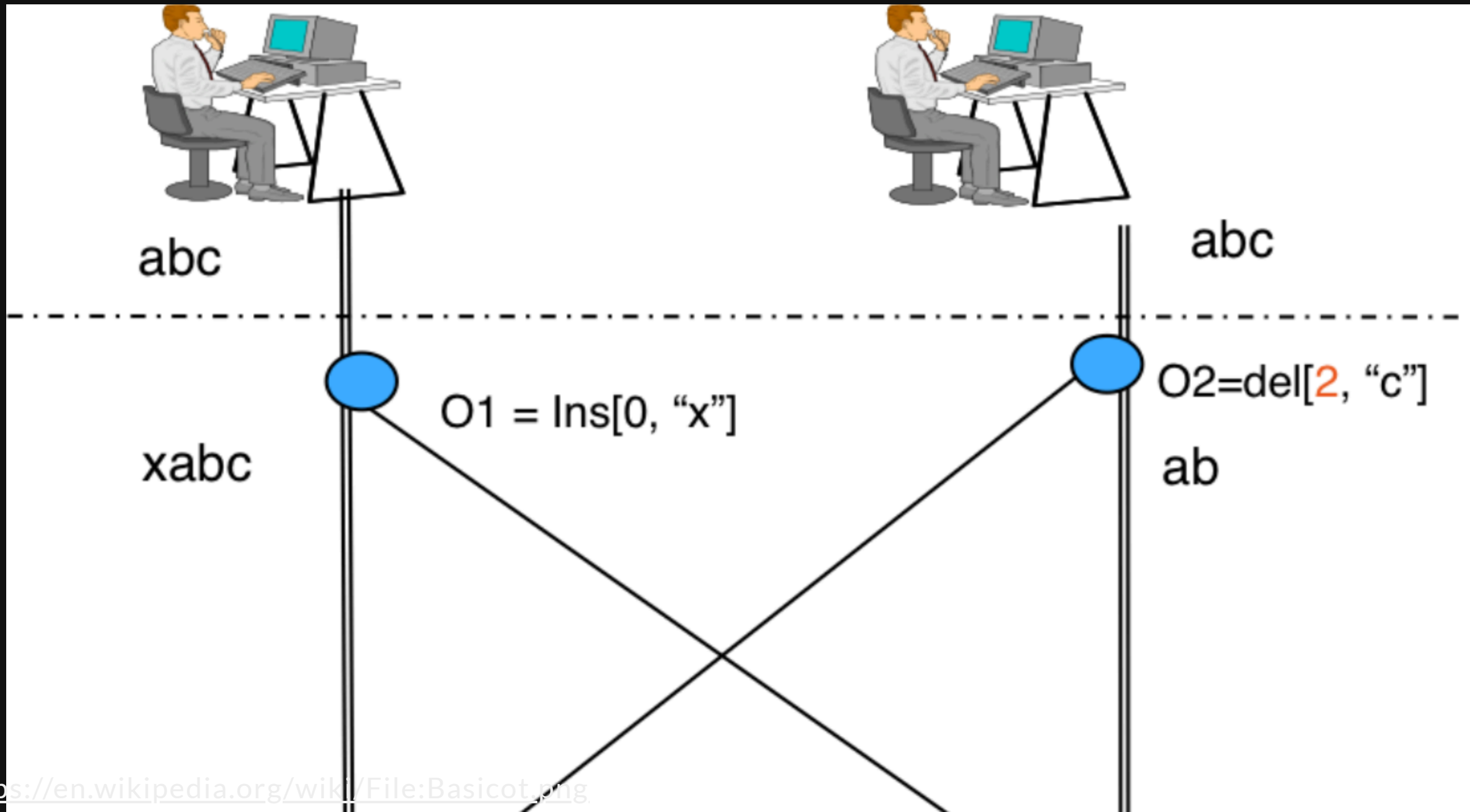
🌐 blinry.org

🐘 @blinry@chaos.social

Bonus slides

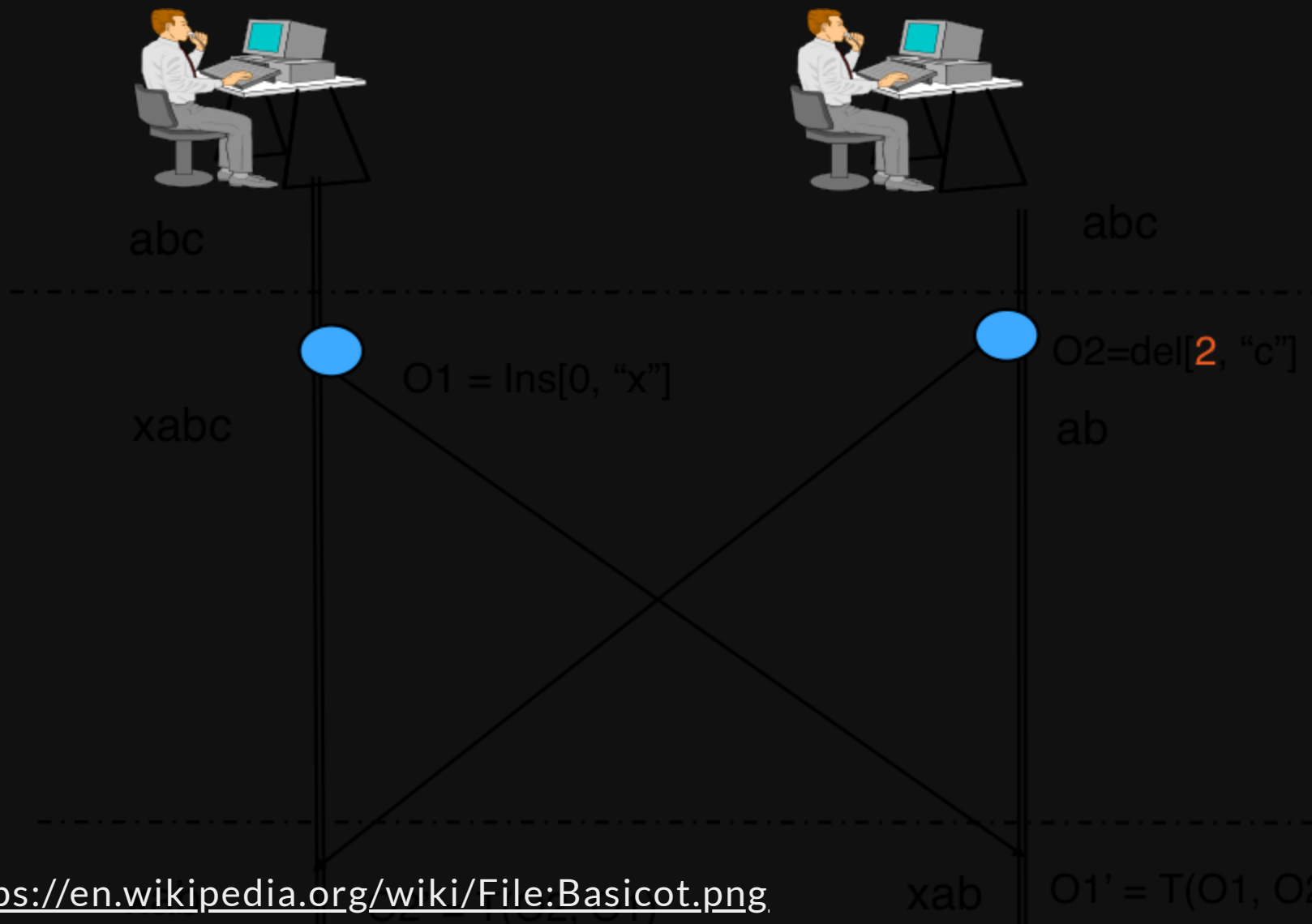
Sending edits between processes





https://en.wikipedia.org/wiki/File:Basicot.png

Solution: Operational Transformation



https://en.wikipedia.org/wiki/File:Basicot.png

Speak JSON-RPC with `ethersync client`

Request:

```
{"jsonrpc": "2.0", "id": 1, "method": "open", "params": {"uri": "file:///path/to/file"}}
```

Response:

```
{"jsonrpc": "2.0", "id": 1, "result": "success"}
```

LSP

3.17 (Current)

3.18 (Upcoming)

3.16 (Previous)

LSIF

0.6.0 (Current)

0.5.0 (Previous)

Base Protocol

0.9 (Upcoming)

Language Server Protocol Specification - 3.17

This document describes the 3.17.x version of the language server protocol. An implementation for node of the 3.17.x version of the protocol can be found here.

Note: edits to this specification can be made via a pull request against this markdown document.

What's new in 3.17

All new 3.17 features are tagged with a corresponding since version 3.17 text or in JSDoc using `@since 3.17.0` annotation. Major new feature are: type hierarchy, inline values, inlay hints, notebook document support and a meta model that describes the 3.17 LSP version.

A detailed list of the changes can be found in the change log

The version of the specification is used to group features into a new specification release and to refer to their first appearance. Features in the spec are kept compatible using so called capability flags which are exchanged between the client and the server during initialization.

Base Protocol

Language Server Protocol

Basic JSON Structures

Lifecycle Messages

Document Synchronization

Language Features

Workspace Features

Window Features

Miscellaneous

Change Log

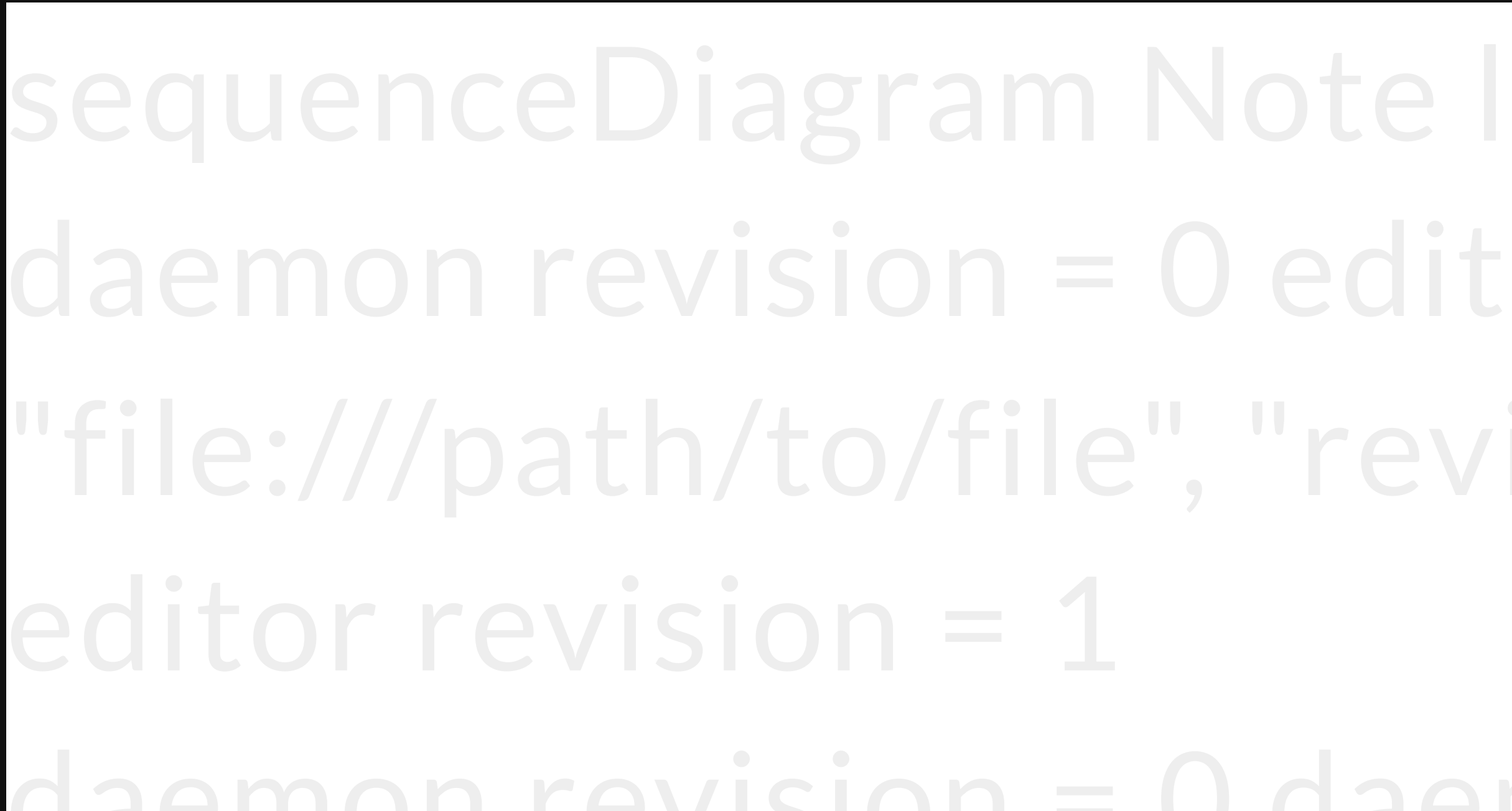
"open" and "close" messages

<script type="module"> import mermaid from 'https://cdn.jsdelivr.net/npm/mermaid@11/dist/mermaid.esm.min.mjs'; mermaid.initialize({ startOnLoad: true }); </script>

sequenceDiagram editor->>da {"uri":"file:///path/to/file"} dae >>daemon: ... editor->>daemo daemon->>editor: "ok"

"edit" messages

```
"edit" {
    "uri": "file:///path/to/file",
    "revision": 0,
    "delta": [
        {
            range": {
                "start": {"line": 0, "character": 0},
                "end": {"line": 0, "character": 0}
            },
            "replacement": "boop"
        }
    ]
}
```


sequenceDiagram Note l
daemon revision = 0 edit
"file:///path/to/file", "revi
editor revision = 1
daemon revision = 0 daem

"cursor" messages from editor

```
"cursor" {
    "uri": "file:///path/to/file",
    "ranges": [{
        "start": {"line": 0, "character": 5},
        "end": {"line": 0, "character": 5}
    }]
}
```

"cursor" messages from daemon

```
"cursor" {
    "uri": "file:///path/to/file",
    "ranges": [{
        "start": {"line": 0, "character": 5},
        "end": {"line": 0, "character": 5}
    }],
    "userid": "123",
    "name": "sam"
}
```