



A note from the student behind this lab

This is an **independent, free, learning-only student project**. It respects every company and vendor, and every trademark belongs to its owner. It is **not affiliated with or endorsed by** any of them.

If anyone has a concern, the content will be removed or corrected promptly: please write via <https://cyphermonkeylabs.com/#contact>. Nothing here is sold, and I am happy to volunteer for any organization.

WebRTC and SIP over WebSocket

WebRTC names a browser protocol suite. JSEP leaves the signaling protocol to the application. SIP over WebSocket is one such protocol.

WebRTC is a suite, not a SIP method

RFC 8825 is an applicability statement for real-time communication in browsers. It does not itself specify a protocol. It points at the other specifications an implementation follows to be compliant with WebRTC.

RFC 8829, the JavaScript Session Establishment Protocol, says the browser API should control the signaling state while the JavaScript chooses the signaling protocol. The document names the existing SIP call signaling protocol as one option, and a custom protocol as another. The object that has to move is the session description.

SIP over WebSocket is still SIP

RFC 7118 specifies a WebSocket subprotocol so SIP requests and responses can cross a WebSocket connection. It is a transport document. It does not replace INVITE, REGISTER, or the status codes.

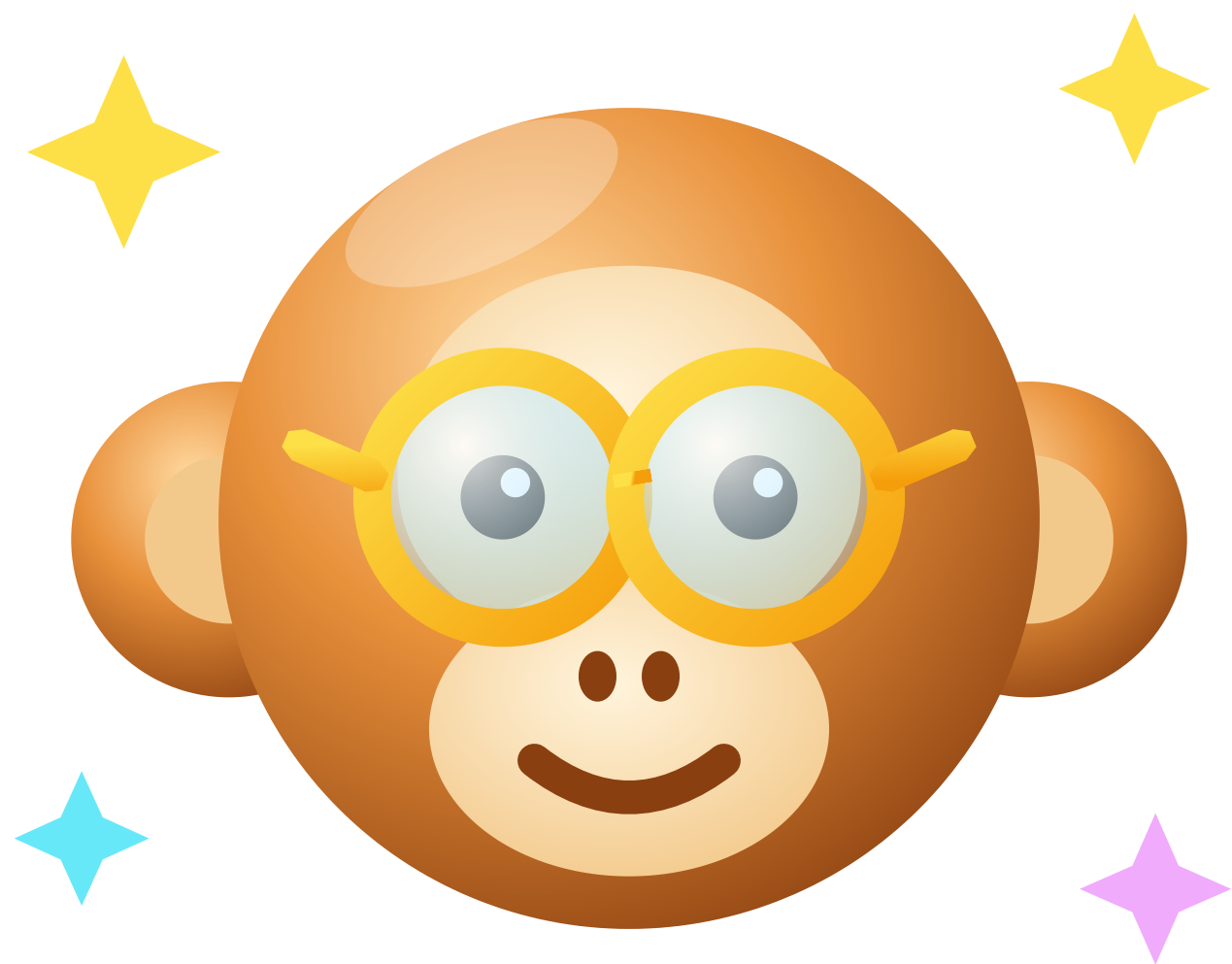
RFC 6455 is the WebSocket protocol under that subprotocol: an opening handshake, then message framing over TCP. A browser softphone that uses SIP is combining that transport with ordinary SIP methods, then a media stack that is specified elsewhere.

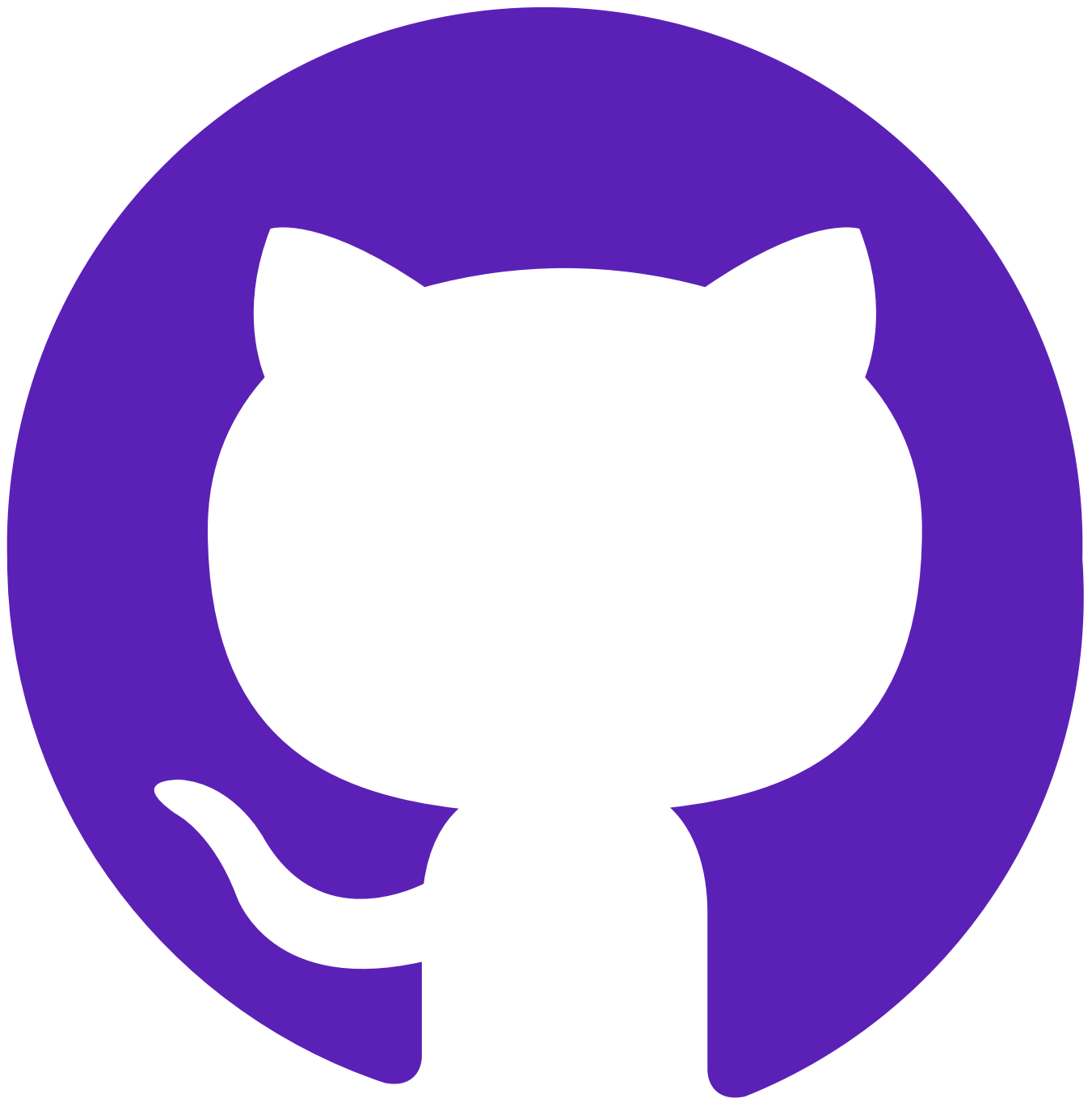
Room: [Signaling, JSEP, and the media plane](#)

Sources and go deeper

- [RFC 8825](#) — last checked 2026-10
- [RFC 8829](#) — last checked 2026-10
- [RFC 7118](#) — last checked 2026-10
- [RFC 6455](#) — last checked 2026-10

CypherMonkey Labs







© 2026 CypherMonkeyLabs 🐒

[Open app](#) · [View repo](#)