

Connecting everything, everywhere, all at once with libp2p

[@2color](#) / Daniel
[@TheDiscordian](#)
[@jochasinga](#) / Pan
[@achingbrain](#) / Alex
[@maschad](#) / Chad
[@p-shahi](#) / Prithvi
[@marcopolo](#) / Marco
[@thomaseizinger](#) / Thomas
[@mxinden](#) / Max



libp2p

- Peer-to-peer networking library
- One specification, many implementations (Go, JS, Rust, Nim, C++, Java, ...)
- Low level features like encryption, authentication and hole punching
- High level features like DHT or Gossiping
- All you need to build peer-to-peer applications



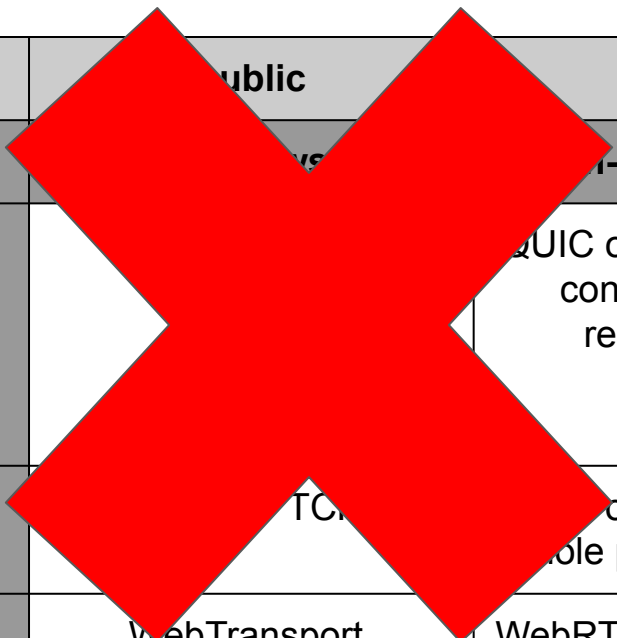
LIBP2P

Connectivity Options

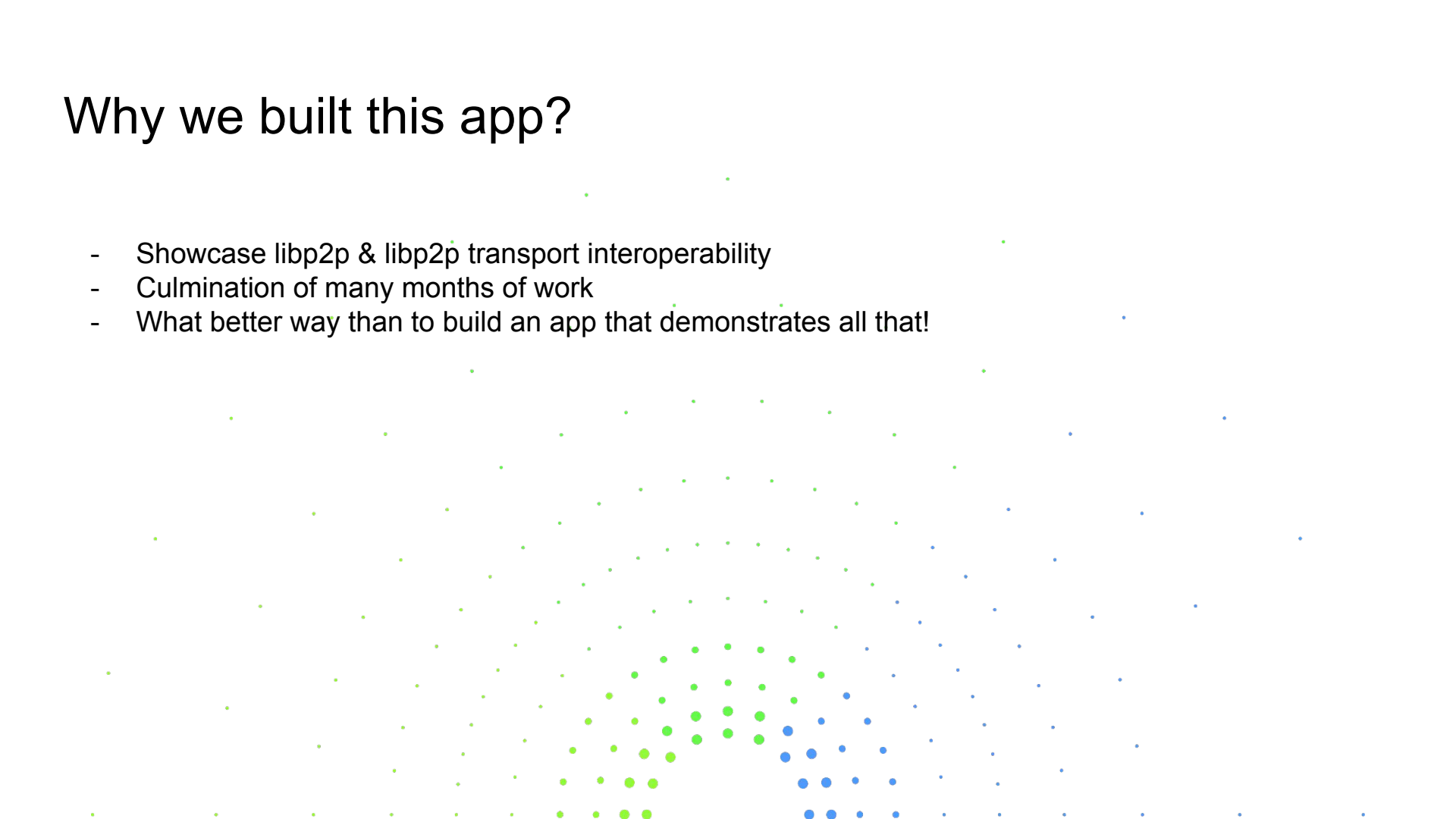
reachability	public		private	
	platform	non-browser	non-browser	browser
public	non-browser	QUIC or TCP	QUIC or TCP with connection reversal	WebTransport, WebSocket, WebRTC with connection reversal
private	non-browser	QUIC or TCP	QUIC or TCP with hole punching	WebRTC with hole punching
	browser	WebTransport, WebSocket or WebRTC	WebRTC with hole punching	WebRTC with hole punching

Connectivity Options

reachability	public		private	
	platform	non-browser	non-browser	browser
public	non-browser	QUIC or TCP with connection reversal	QUIC or TCP with connection reversal	WebTransport, WebSocket, WebRTC with connection reversal
private	non-browser	QUIC or TCP with hole punching	QUIC or TCP with hole punching	WebRTC with hole punching
	browser	WebTransport, WebSocket or WebRTC	WebRTC with hole punching	WebRTC with hole punching



Why we built this app?

- Showcase libp2p & libp2p transport interoperability
 - Culmination of many months of work
 - What better way than to build an app that demonstrates all that!
- 
- A decorative background consisting of numerous small dots in shades of green and blue, scattered across the lower half of the slide. The dots are more densely packed in the bottom center and become sparser towards the edges.

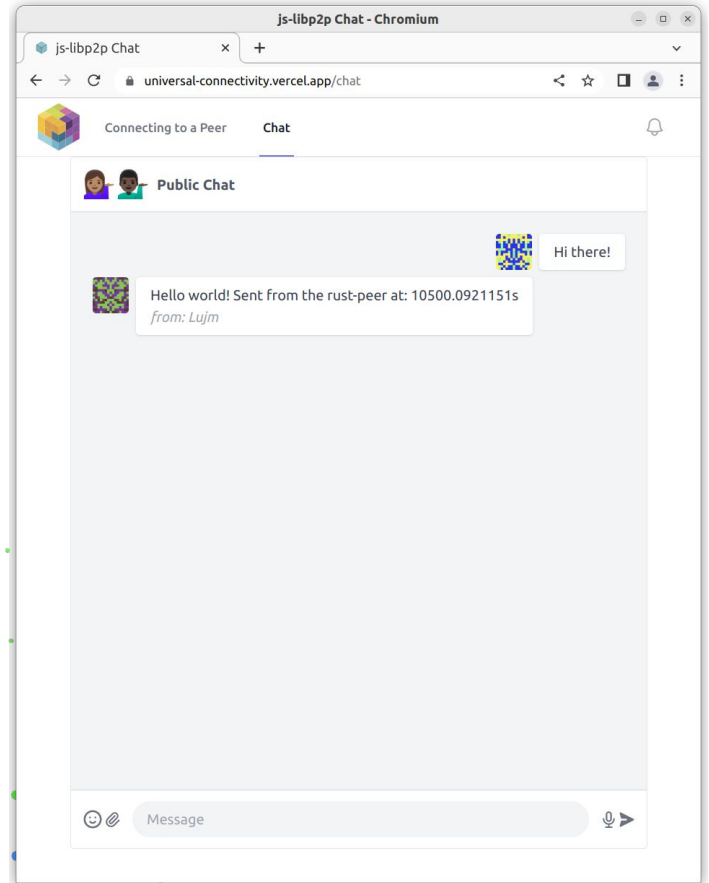
Everything Everywhere All at Once

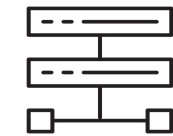


Look, this is your universe,

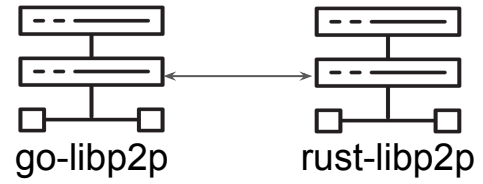
libp2p Chat

- Simple peer-to-peer chat application
- Demonstrate how easy it is to connect
 - *everything* (browsers & non-browsers),
 - *everywhere* (public or private)
 - *all at once* (using libp2p)!



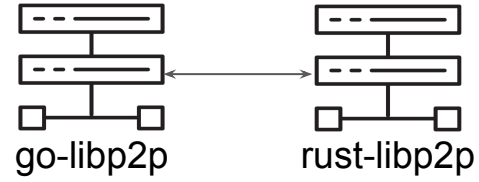


rust-libp2p





go-libp2p



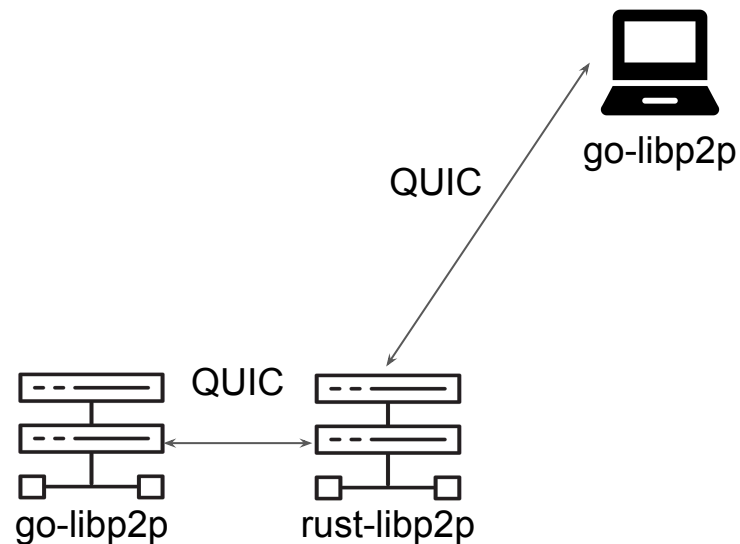
QUIC in libp2p

Lifecycle Stage	Maturity	Status	Latest Revision
3A	Recommendation	Active	r1, 2022-12-30

- Connecting (public) non-browser to (public) non-browser

Support in go-libp2p since forever

Support in rust-libp2p since Q4 2022



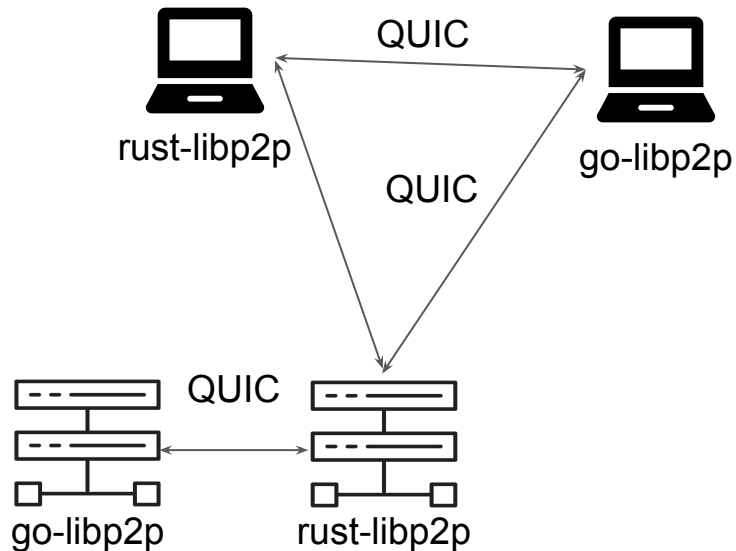
QUIC in libp2p

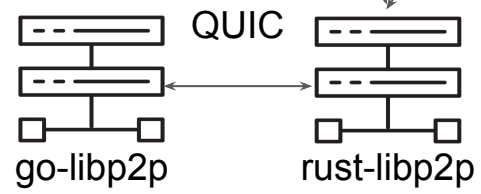
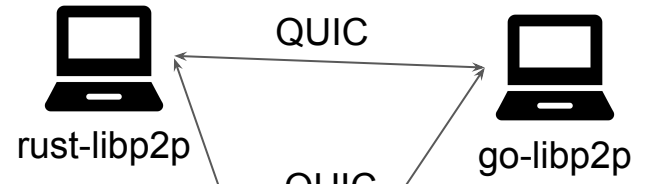
Lifecycle Stage	Maturity	Status	Latest Revision
3A	Recommendation	Active	r1, 2022-12-30

- Connecting (public) non-browser to (public) non-browser
- Allows hole punching between two non-browsers

Support in go-libp2p since forever

Support in rust-libp2p since Q4 2022





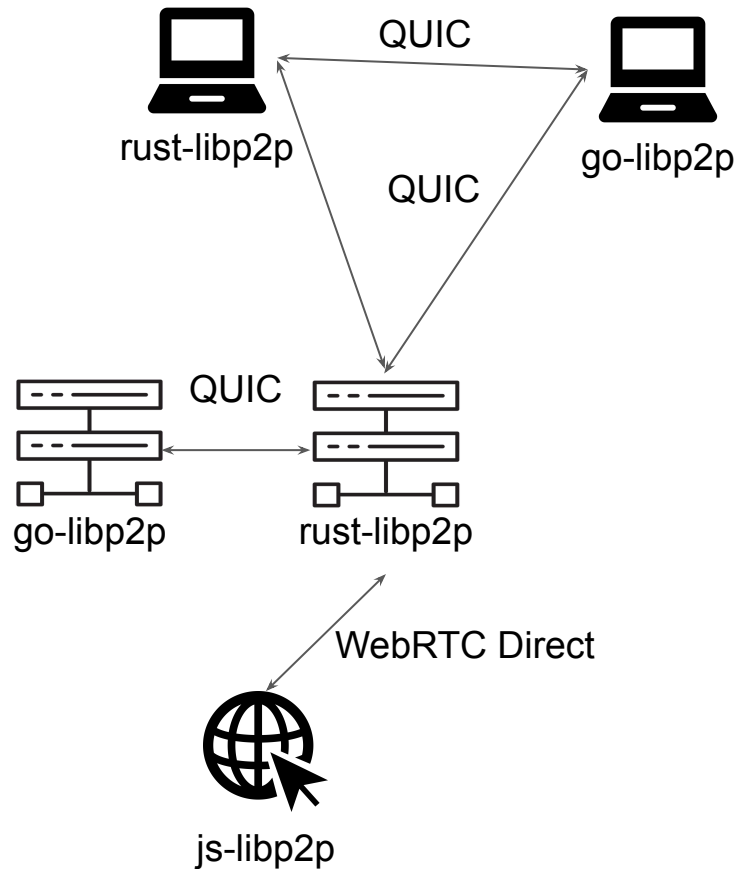
WebRTC Direct

Lifecycle Stage	Maturity	Status	Latest Revision
2A	Candidate Recommendation	Active	r1, 2023-04-12

- Connecting browser to public non-browsers
- Self-signed certificates

Implemented in js-libp2p link Q4 2022

Implemented in rust-libp2p link Q4 2022



libp2p WebTransport

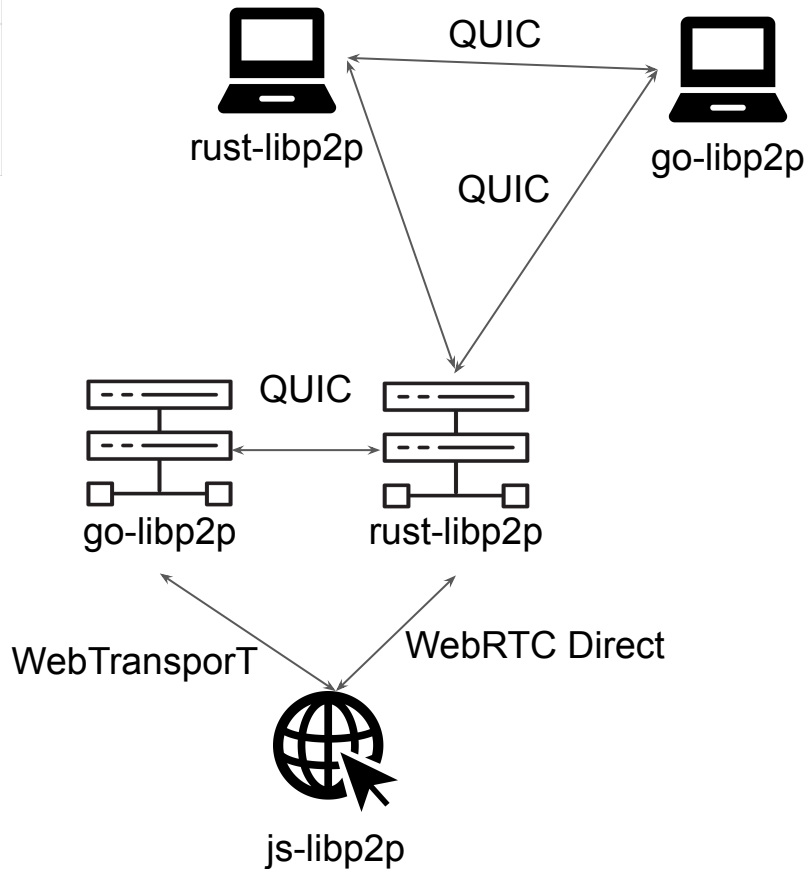
Lifecycle Stage	Maturity	Status	Latest Revision
2A	Candidate Recommendation	Active	r0, 2022-10-12

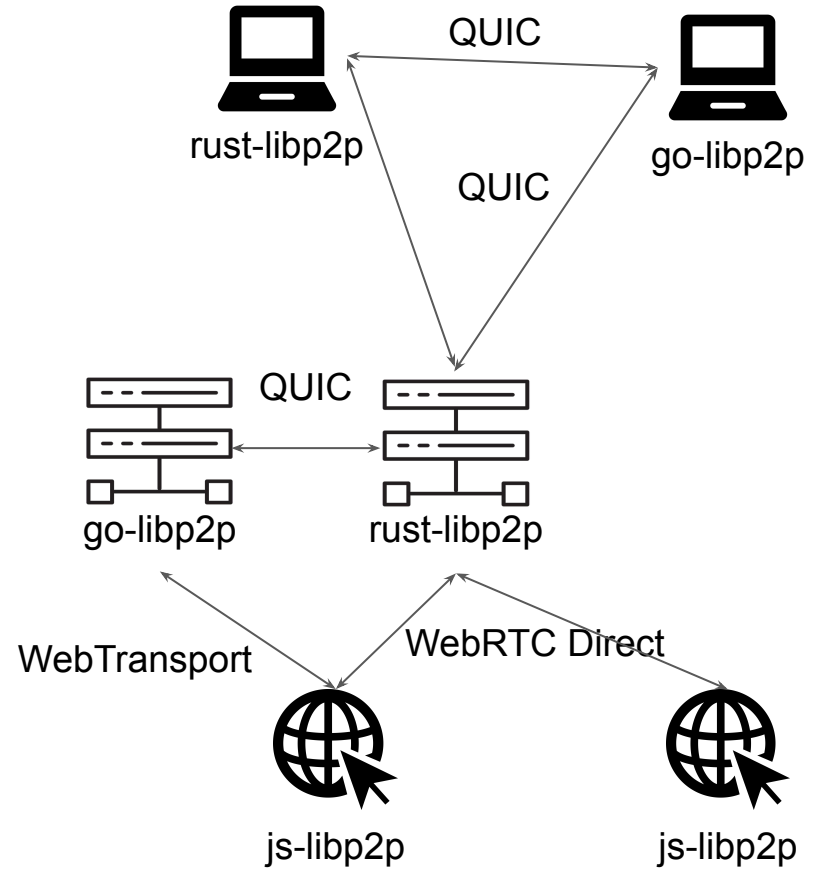
- Connecting browser to public non-browsers
- Self-signed certificates

Implemented in js-libp2p Q4 2022

Implemented in go-libp2p Q4 2022

Ships in go-libp2p and [on by default in Kubo](#)



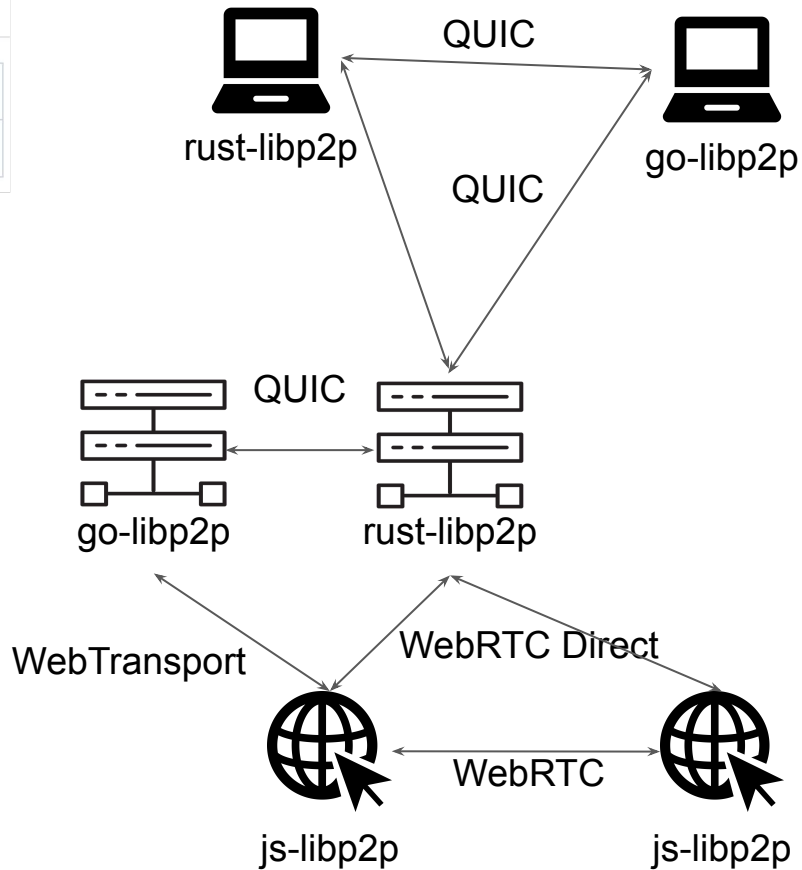


WebRTC

Lifecycle Stage	Maturity	Status	Latest Revision
2A	Candidate Recommendation	Active	r0, 2023-04-12

- Connecting browser to browsers
- Hole punching

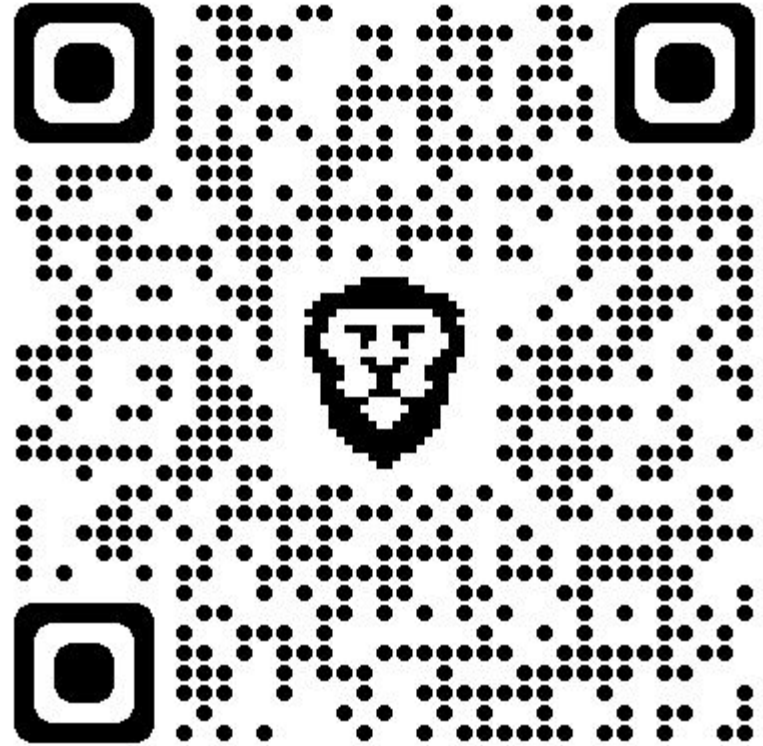
Implemented in js-libp2p Q1 2023



Join the chat!

- Scan the code to join the chat
- or
- <https://universal-connectivity.vercel.app/>

([Small bug fix](#) for Firefox not yet merged.)



What is next?

- [Bug fix for libp2p WebRTC in Firefox](#)
 - [WebRTC Direct \(browser-to-server\) in go-libp2p](#)
 - [Firefox support for WebTransport](#)
 - Eventually
 - WebTransport support in rust-libp2p
 - WebRTC (browser-to-private) in go and rust libp2p
 - Emoji support
 - Feature to travel universes
 - Zig libp2p
- 
- A decorative background consisting of numerous small dots in shades of green and blue, scattered across the lower half of the slide.

Pull request 46

- Community is already contributing

Construct addresses using MultiAddr, defaulting to IPv6 #46



DougAnderson4... wants to merge 5 commits into `Libp2p:main` from `DougAnderson444:address-IPv6`

Conversation 5

Commits 5

Checks 1

Files changed 5



DougAnderson444 commented 2 days ago

First-time contributor Tip ...

Rust-Peer (server) improvements

I was working in my own repo, but I figure I would start using this repo instead of building in a parallel silo. Here are some changes I would propose for this repo.

I know this Rust server is meant for the cloud, but seeing as I am a big fan of "running a peer at home" this PR started out as changing the default IP to v6 so home servers are easier to connect, and then I added a few more cleanup items along the way. EC2 should run on IPv6 addresses just fine too, even though it has a public IP.

Summary of changes

- use IPv6 by default, as they offer globally available IP addresses (IPv4 is so 1998)
- note that `ipv6.is_global()` is **only available on nightly**, I hope that's not a problem? I added a toolchain config for that. If not we can code around it with logic I suppose...
- use address variables (`address_webrtc` & `address_quic`)
- only add globally available external addresses to the `swarm`
- construct address by building with `MultiAddr.with(...)`
- move constants to beginning of `main.rs`
- match on `opt_listen_address` (`String`) to ensure it's a valid IP address (`IpAddr`)
- cleanup/remove unused deps, & ping

Remaining work

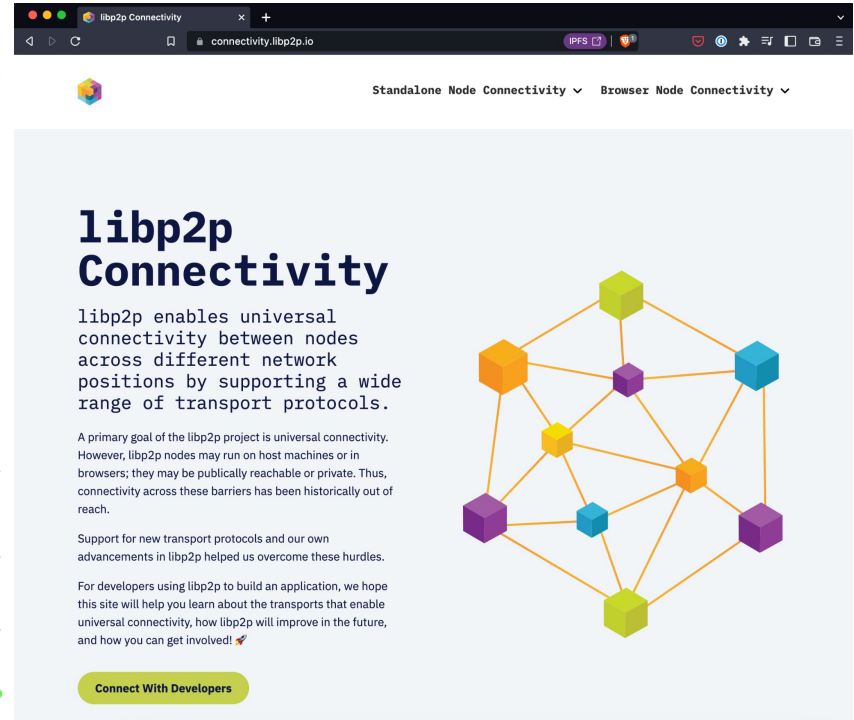
- I left the dependencies as-is for now, whenever I moved off the branches things broke, and I wanted to get these changes in incrementally. It's late at night now, so perhaps we can migrate/patch the deps in another PR?



2

Where to learn more

- github.com/libp2p/universal-connectivity/
- connectivity.libp2p.io
- github.com/libp2p/specs

A screenshot of a web browser displaying the libp2p Connectivity website. The browser's address bar shows the URL 'connectivity.libp2p.io'. The website has a dark header with the libp2p logo and navigation links for 'Standalone Node Connectivity' and 'Browser Node Connectivity'. The main content area features the title 'libp2p Connectivity' in a large, bold font. Below the title is a paragraph explaining that libp2p enables universal connectivity between nodes across different network positions by supporting a wide range of transport protocols. A second paragraph states that a primary goal of the libp2p project is universal connectivity, but nodes may be on host machines or in browsers, making connectivity across these barriers historically difficult. A third paragraph mentions that support for new transport protocols and their own advancements in libp2p helped overcome these hurdles. A final paragraph expresses hope that the site will help developers learn about transports that enable universal connectivity, how libp2p will improve in the future, and how they can get involved. A green button labeled 'Connect With Developers' is positioned at the bottom of the text. To the right of the text is a network diagram consisting of several interconnected nodes represented by colored cubes (orange, purple, blue, green) connected by orange lines.