# PION

WEBRTC

# Contributors

18871002288
3Byuan
Aaron France
Adam Kiss
Aleksandr
Aleksandr Arofikin
Aleksandr Razumov
Alex Browne
Alex Harford
AlexWoo(武杰)
Alexey Kravtsov
Andrei Nistor
Andrew N. Shalaev
Anindya Chatterjee
Antoine Baché
Artur Shellunts
Assad Obaid
Ato Araki
Atsushi Watanabe
BN
Bao Nguyen
Ben Weitzman
Billy Lindeman
Bjørn Remseth
Bo Shi
Brendan Rius
Cameron Elliott
Carson Hoffman
Cecylia Bocovich
Cedric Fung
Cédric Verstraeten
Chad Retz
Chao Yuan
Chris Hiszpanski
Christian Muehlhaeuser
Christopher Fry
Clayton McCray
CloudWebRTC
Cory Schwartz
Daniele Sluijters
David Hamilton
David-dp-
Dean Eigenmann
Denis
Doug Cone
Egon Elbre
Emir Aganovic
EricSong
Gabor Pongracz
Gareth Hayes
Graham King
Greg Burd
Guilherme
Haiyang Wang
Hanjun Kim
Harrison
Hayden James
Hendrik Hofstadt
Henry
Herman Banken
Hongchao Ma
Hugo Arregui
Igor German
Ilya Mayorov
Ingmar Wittkau
Issac Trotts
Ivan Egorov
Ivelin Ivanov
JacobZwang
Jadon Bennett
Jake B
Jake Coffman
Jamie Good
Jannis Mattheis
Jason
Jason Brady
Jason Maldonis
Jeff Tchang
Jeffrey Barron
Jeffrey Stoke
Jerko Steiner
Jeroen de Bruijn
Jim Wert
Jin Gong
John Berthels
John Bradley
John R. Bradley
Jon Lundy
Jonathan Jackson
JooYoung
Jorropo
Josh Bleecher Snyder
Joshua Obasaju
Jozef Kralik
Julien Salleyron
Juliusz Chroboczek
Justin Okamoto
Kazuyuki Honda
Kevin Wang
Kgothatso Ngako
Konstantin Chugalinskiy
Konstantin Itskov
Kuzmin Vladimir
Lander Noterman
Leeward Bound
Levin Du
Lukas Herman

# WHAT IS WEBRTC?

1

# Protocol for Browsers

E2E Secure Connection between Peers

Multiple Audio/Video Tracks

Binary Data

- Can be lossy
- Can be unordered
- Multiple distinct DataChannels

2

# And More!

- aiortc (**Python**)
- GStreamer's webrtcbin **(C)**
- werift (**Typescript**)
- Pion (**Golang**)
- Shiguredo (**Erlang**)
- |pipe| (**Java**)
- rawrtc (**C++**)
- webrtc-rs (**Rust**)
- AWS WebRTC (**C/Embedded**)
- ?

3

# WebRTC for the Curious

**Book on how WebRTC really works**

- Not just about the public APIs!
- Deep dive on protocols

**History of WebRTC**

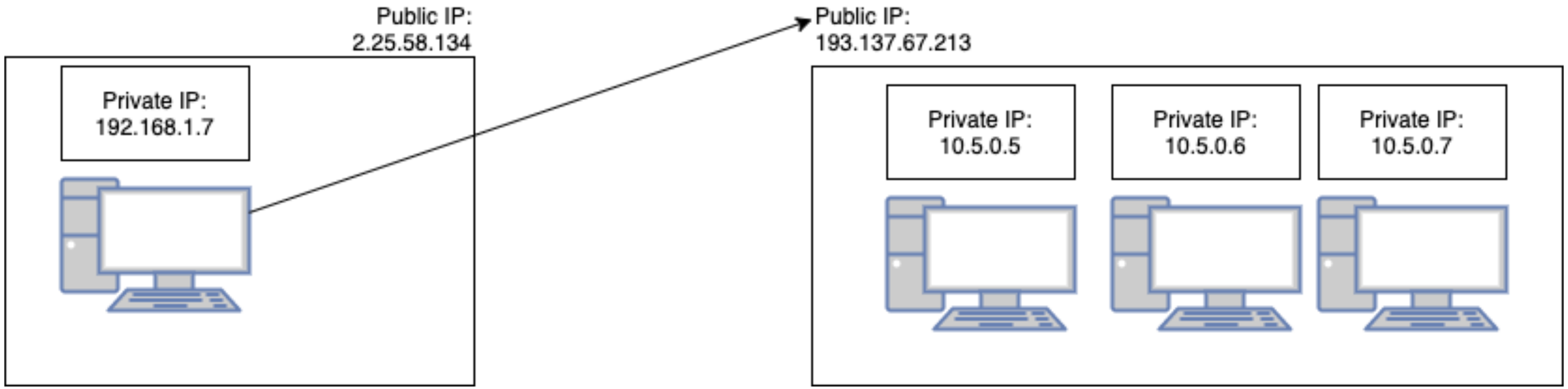
- Interviews with RFC authors

**WebRTC in practice**

- Debugging every piece
- Teach the sharp edges

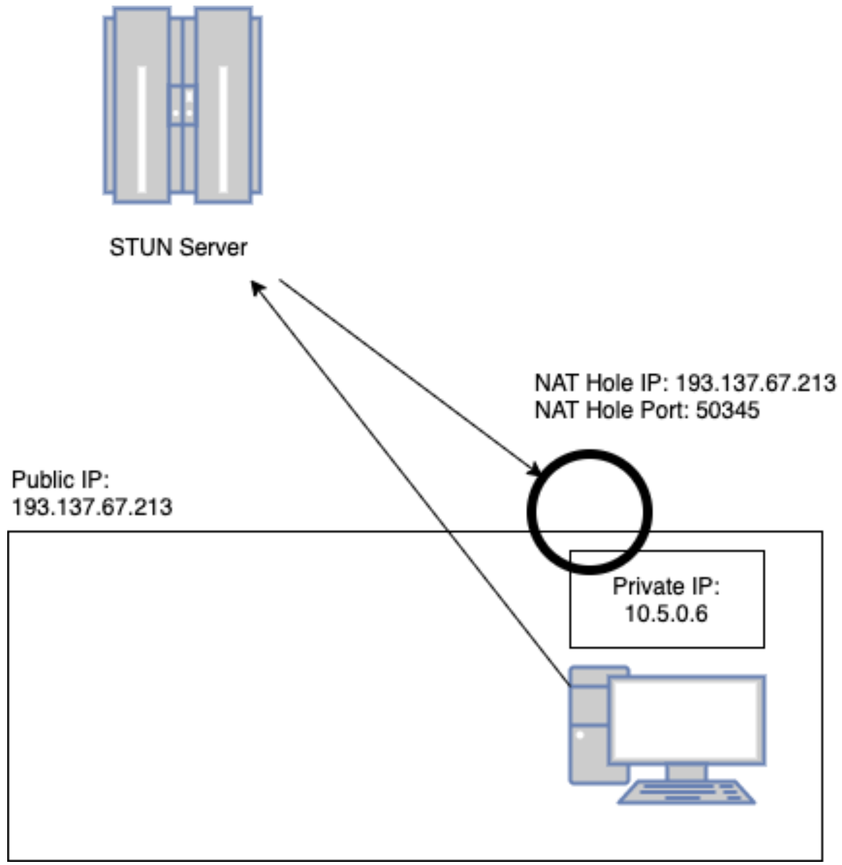
# WHAT DOES IT SOLVE

# Problem: Connect two users with no Public IP

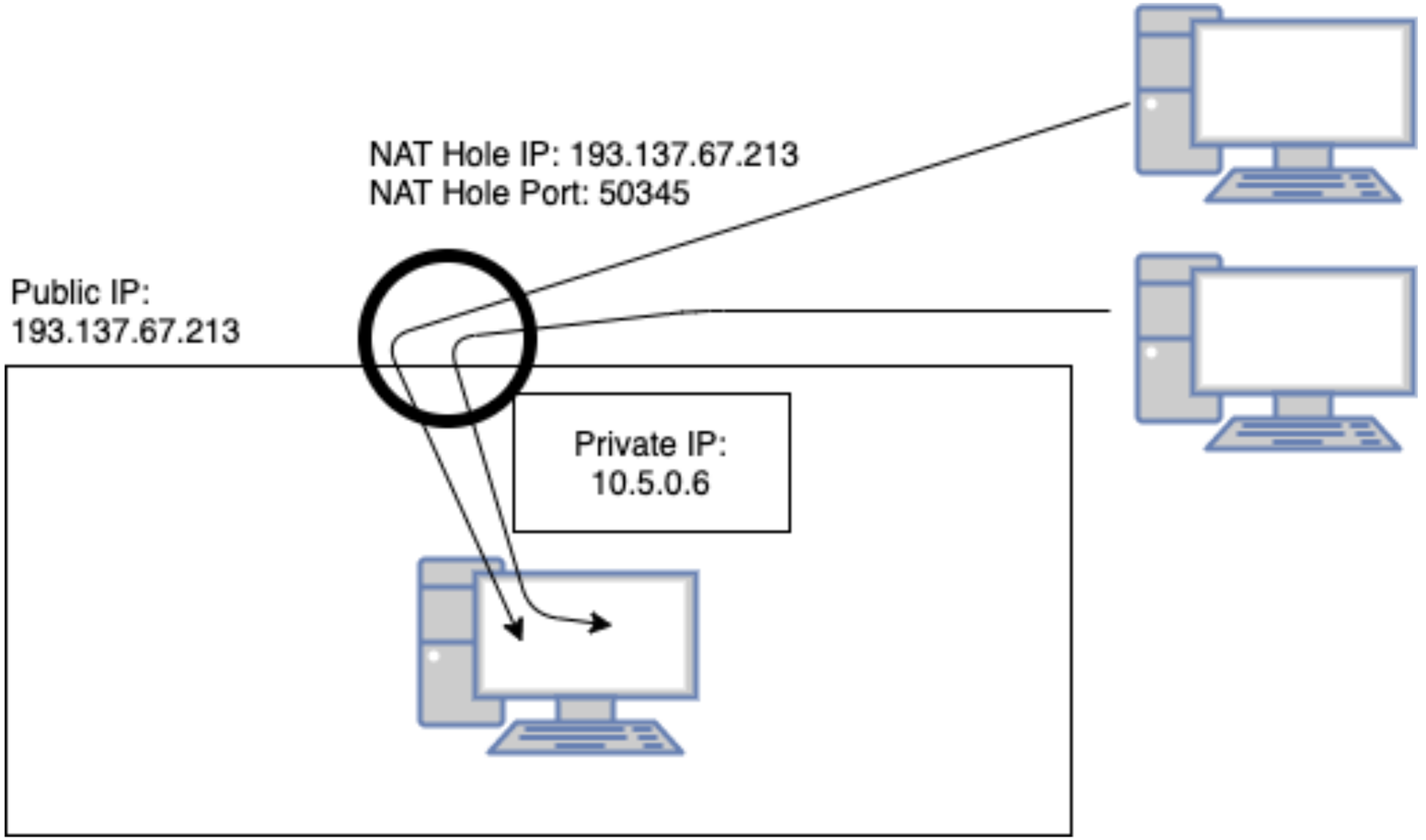


5

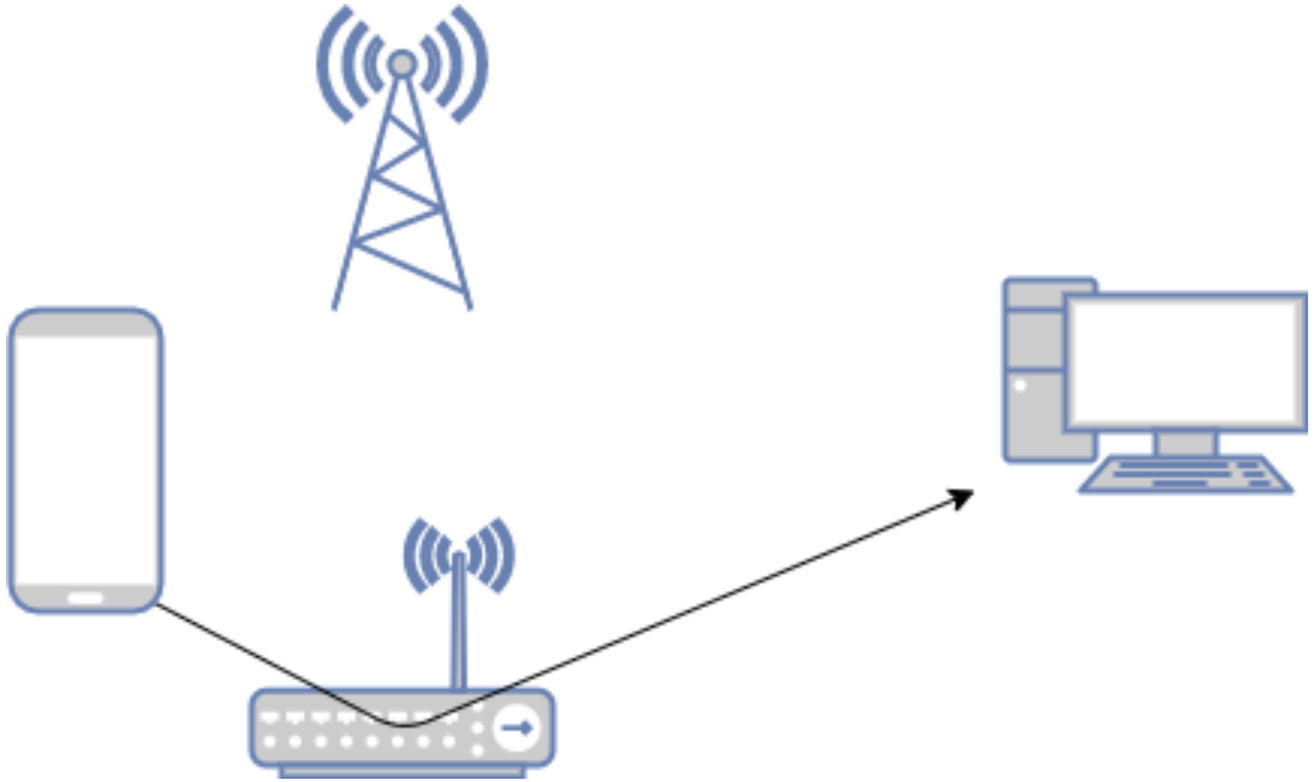
# Solution: NAT Traversal



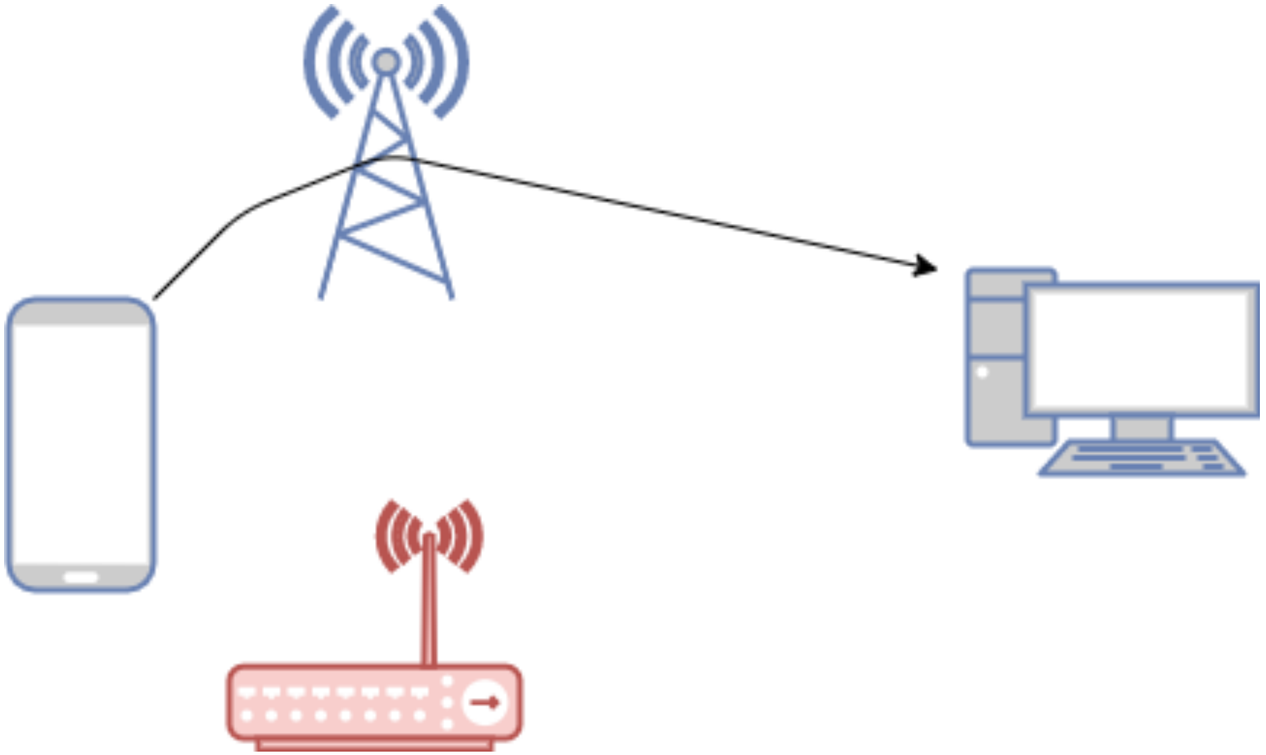
# Solution: NAT Traversal



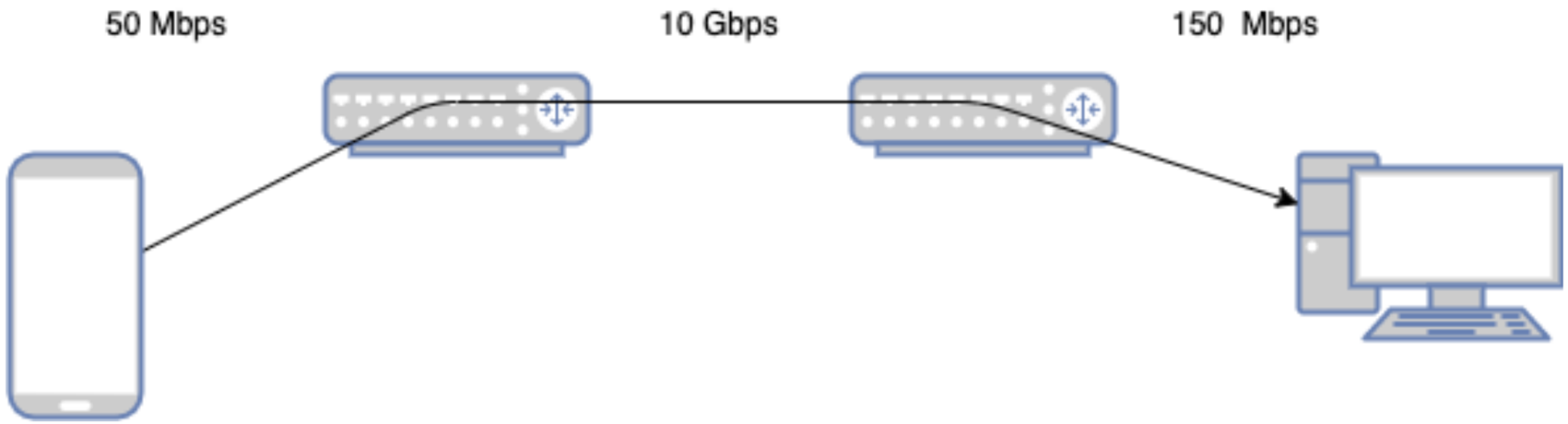
# Problem: Stay connected on the move



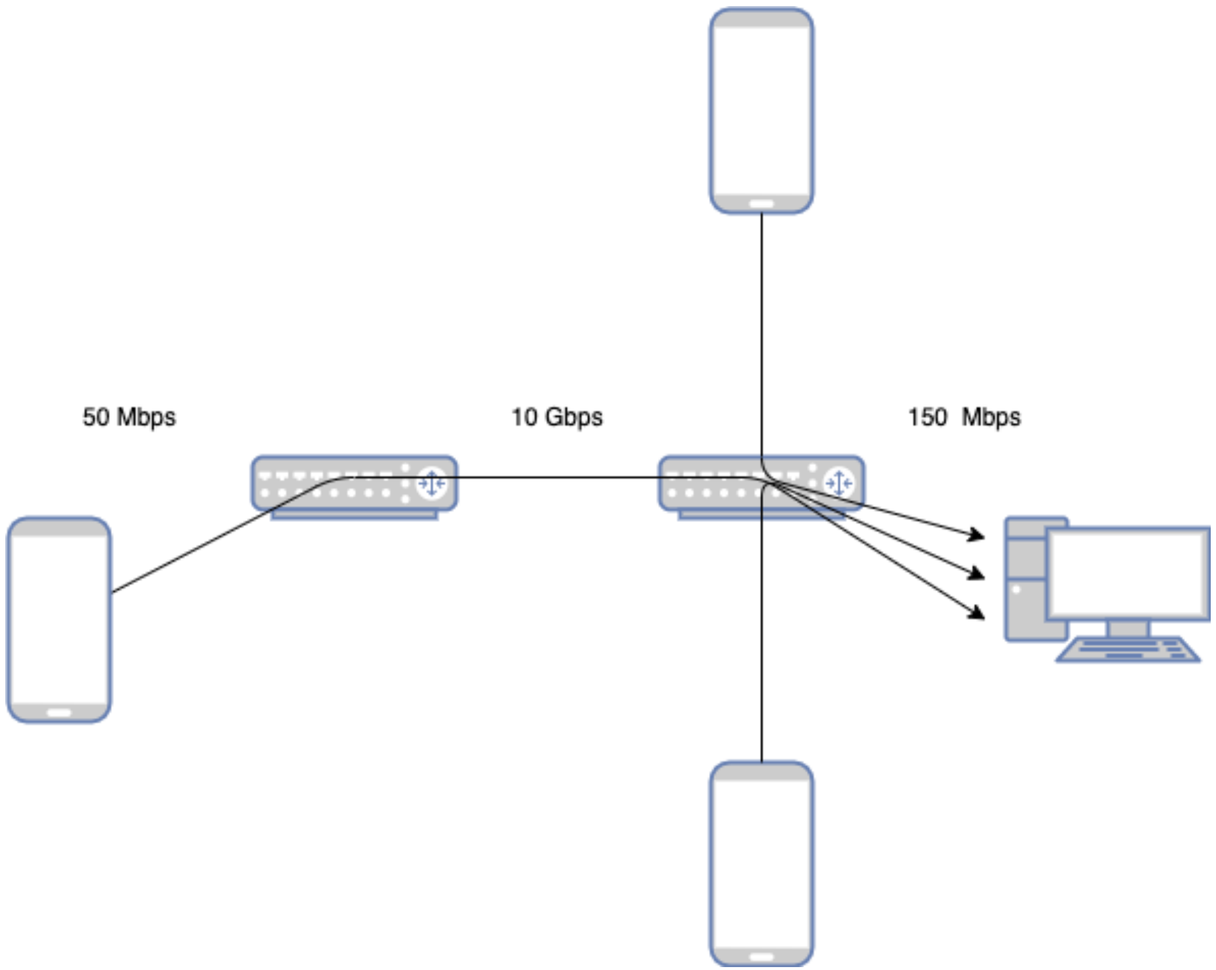
# Solution: ICE Restart



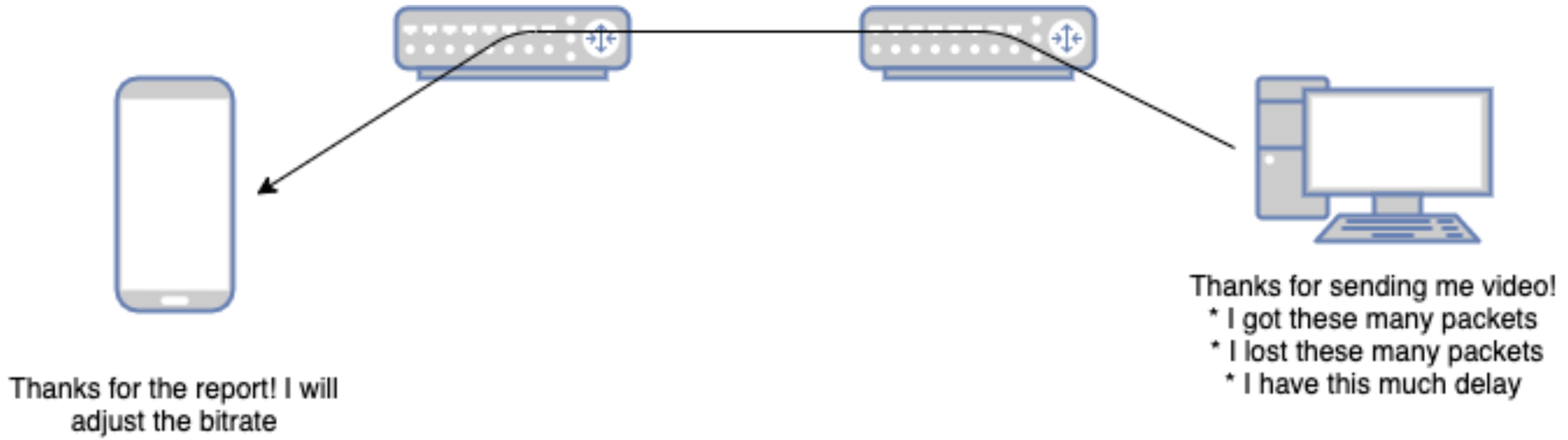
# Problem: What bitrate do I upload?



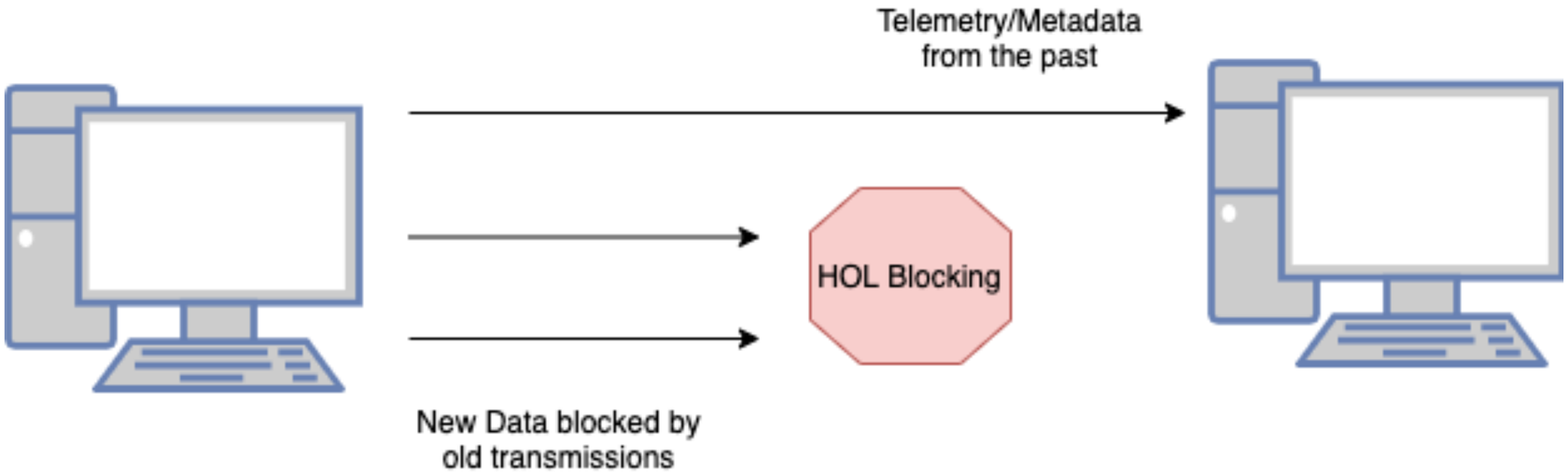
# Problem: What bitrate do I upload?



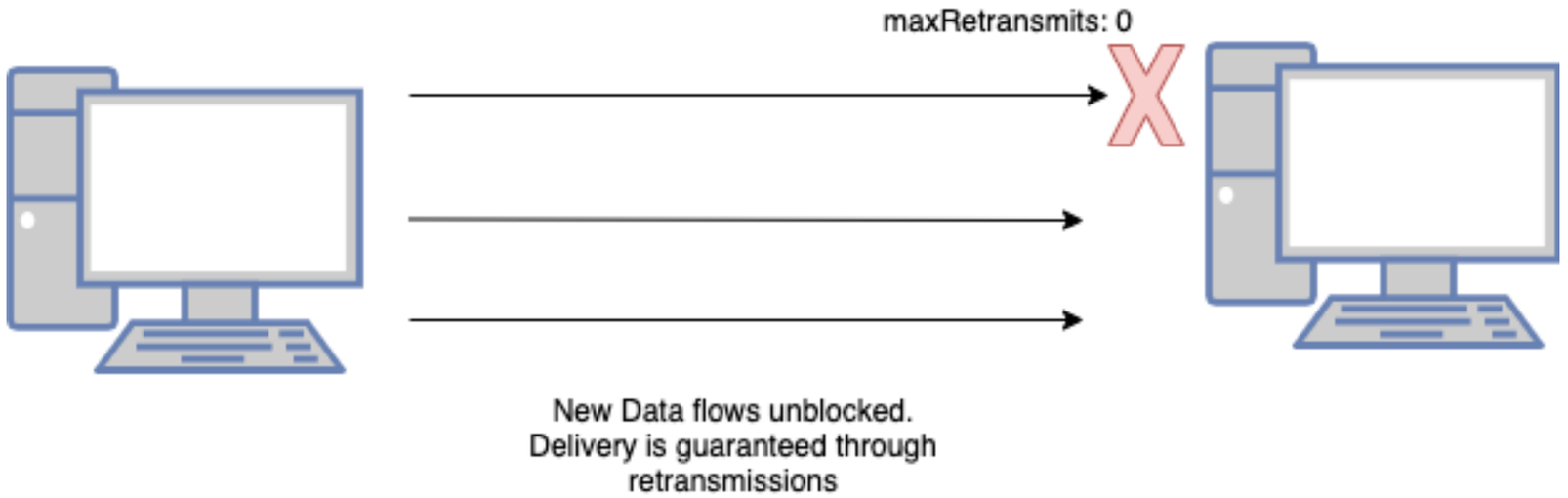
# Solution: Congestion Control



# Problem: Don't block on the past



# Solution: SCTP



TIME TO BUILD

14

# Connecting (Signaling Offer/Answer)

```go
package main

import (
    "github.com/pion/webrtc/v2"
)

func main() {
    peerConnection, err := webrtc.NewPeerConnection(webrtc.Configuration{})
    if err != nil {
        panic(err)
    }

    offer, err := peerConnection.CreateOffer(nil)
    if err != nil {
        panic(err)
    }

    err = peerConnection.SetLocalDescription(offer)
    if err != nil {
        panic(err)
    }

    // send Offer to remote PeerConnection via any protocol
    // receive Answer from remote PeerConnectian
    answer := webrtc.SessionDescription{}
    err = peerConnection.SetRemoteDescription(answer)
    if err != nil {
        panic(err)
    }

    // You are connected
}
```

# Sending Data (DataChannels)

```
    datachannel, err := peerConnection.CreateDataChannel("my-fun-channel", nil)
    if err != nil {
        panic(err)
    }

    datachannel.OnOpen(func() {
        err = datachannel.SendText("Hello World!")
        if err != nil {
            panic(err)
        }
    })
}
```

# Receiving Data (DataChannels)

```
peerConnection.OnDataChannel(func(datachannel *webrtc.DataChannel) {
    datachannel.OnOpen(func() {
        fmt.Printf("New stream %s \n", datachannel.Label())
    })

    datachannel.OnMessage(func(msg webrtc.DataChannelMessage) {
        fmt.Printf("%s \n", msg.Data)
    })
})
```

17

# Deploy to the browser!

```
seaduboi@38f9d359441f:~/go/src/github.com/Sean-Der/kranky$ GOOS=js GOARCH=wasm go build -o main.wasm
seaduboi@38f9d359441f:~/go/src/github.com/Sean-Der/kranky$ cat index.html
<html>
        <head>
                <meta charset="utf-8"/>
                <script src="wasm_exec.js"></script>
                <script>
                        const go = new Go();
                        WebAssembly.instantiateStreaming(fetch("main.wasm"), go.importObject).then((result) => {
                                go.run(result.instance);
                        });
                </script>
        </head>
        <body></body>
</html>
```

18

# Send Video

```
videoTrack, err := peerConnection.NewTrack(webrtc.DefaultPayloadTypeVP8, 50000, "video", "pion")
if err != nil {
    panic(err)
}

_, err = peerConnection.AddTrack(videoTrack)
if err != nil {
    panic(err)
}

for {
    frame, _, err := ivf.ParseNextFrame()
    if err != nil {
        panic(err)
    }

    err = videoTrack.WriteSample(media.Sample{Data: frame, Samples: 90000})
    if err != nil {
        panic(err)
    }
}
```

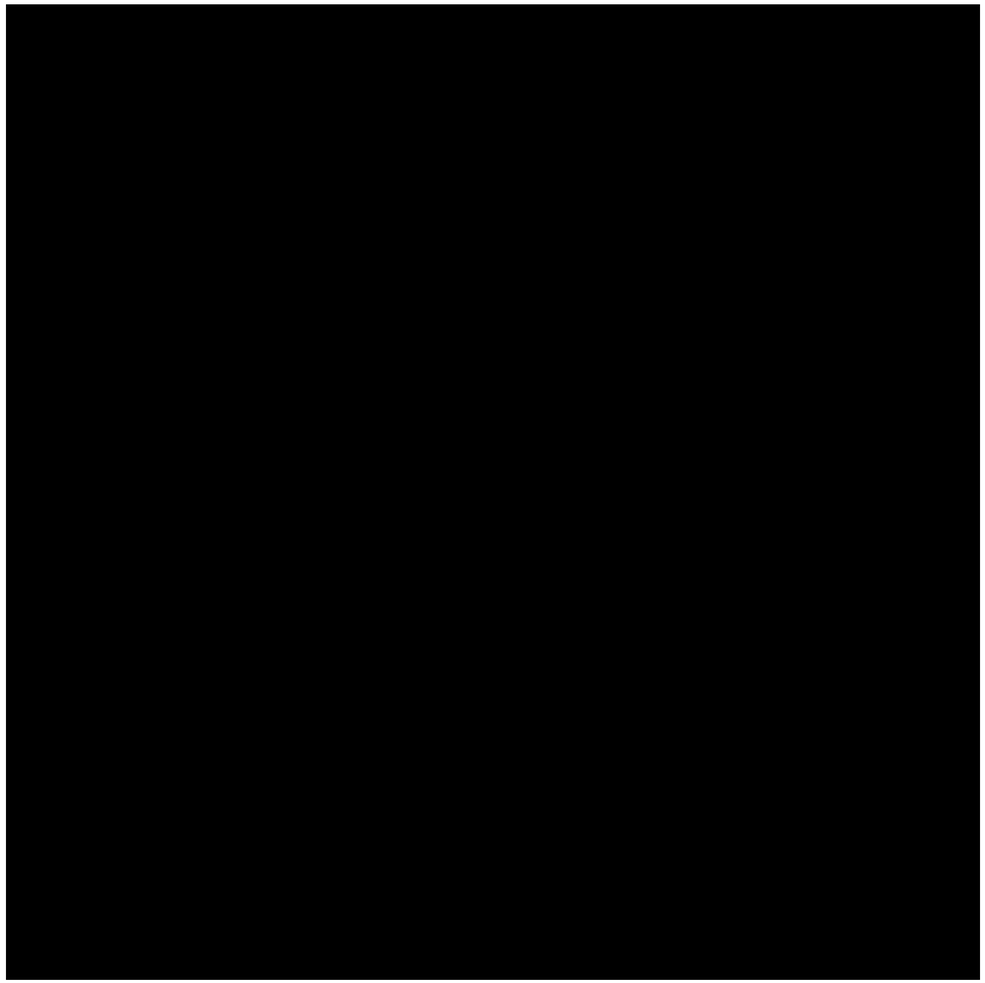
19

# Receive Video

```
peerConnection.OnTrack(func(track *webrtc.Track, receiver *webrtc.RTPReceiver) {
    if track.Codec().Name == webrtc.Opus {
        for {
            packet, err = track.ReadRTP()
            if err != nil {
                panic(err)
            }
            // Playback using Media library of your choice
        }
    }
})
```

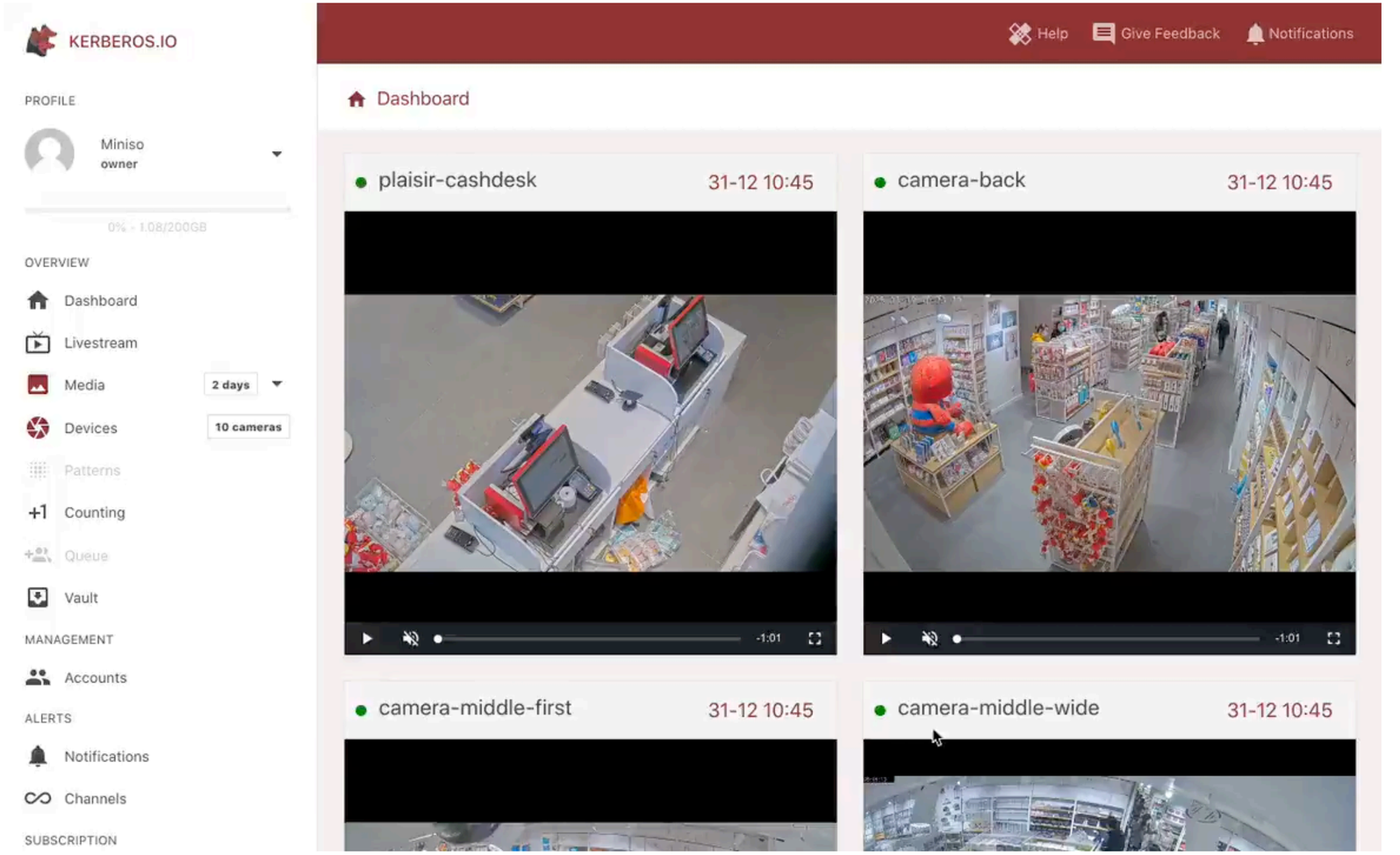
# PION IN ACTION

# ns-remote



21

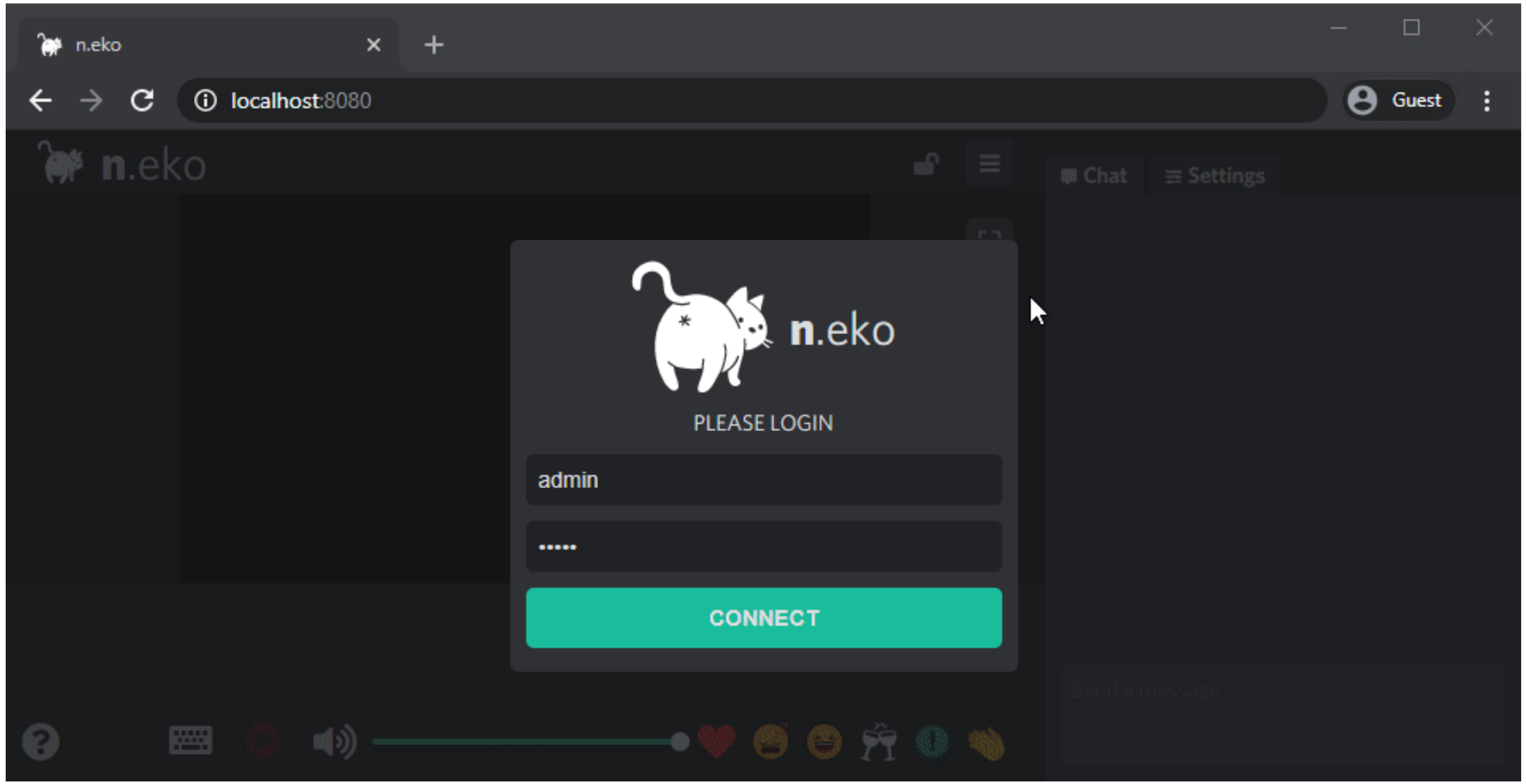
# kerberos.io



# CloudRetro.io

# Neko

## github.com/nurdism/neko



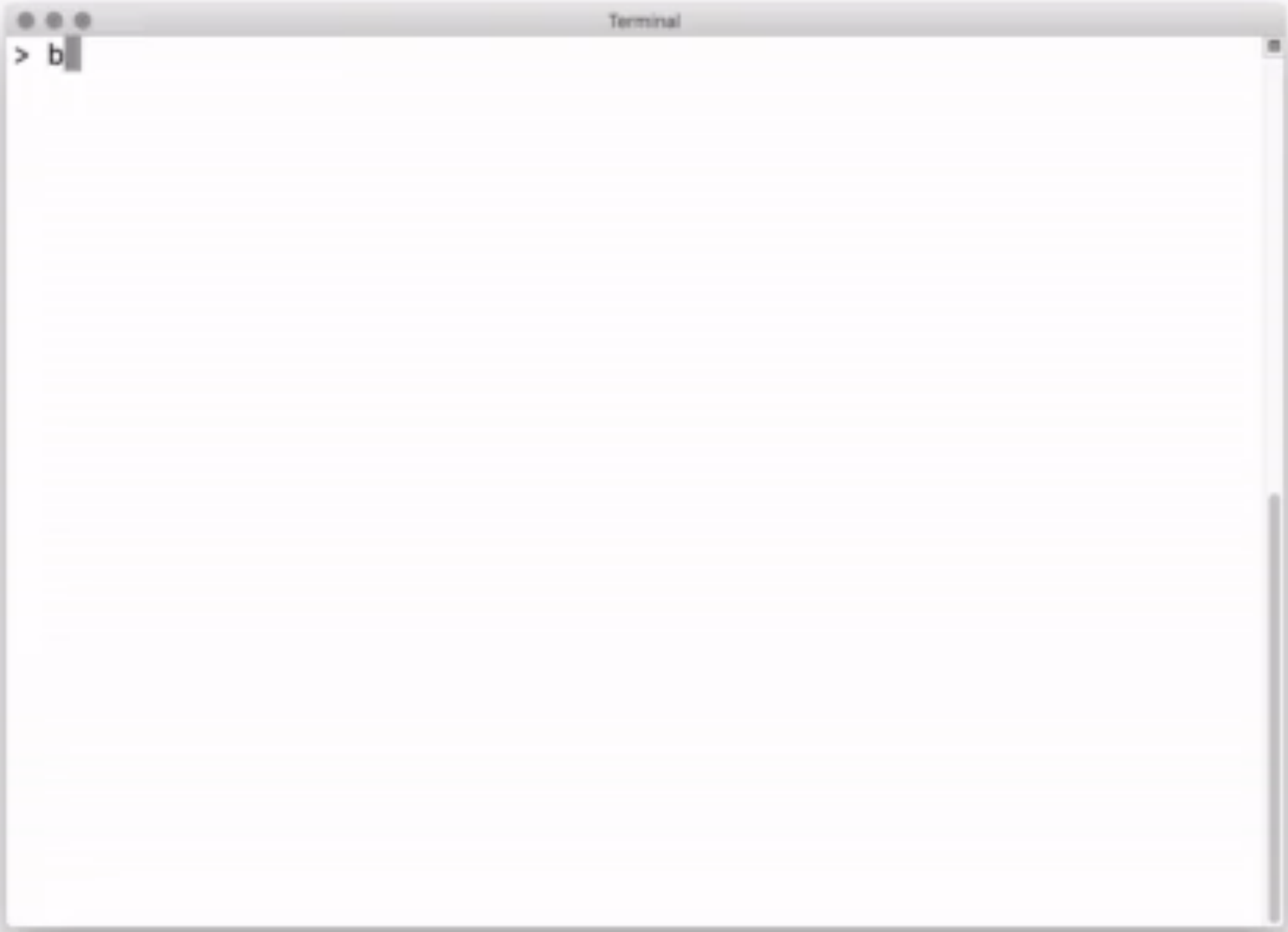
24

# TelloGo


localhost:3000
Distance
Speed
Distance
Speed
Distance
Speed
Distance
Speed
Distance
Speed
Distance
Speed
Distance

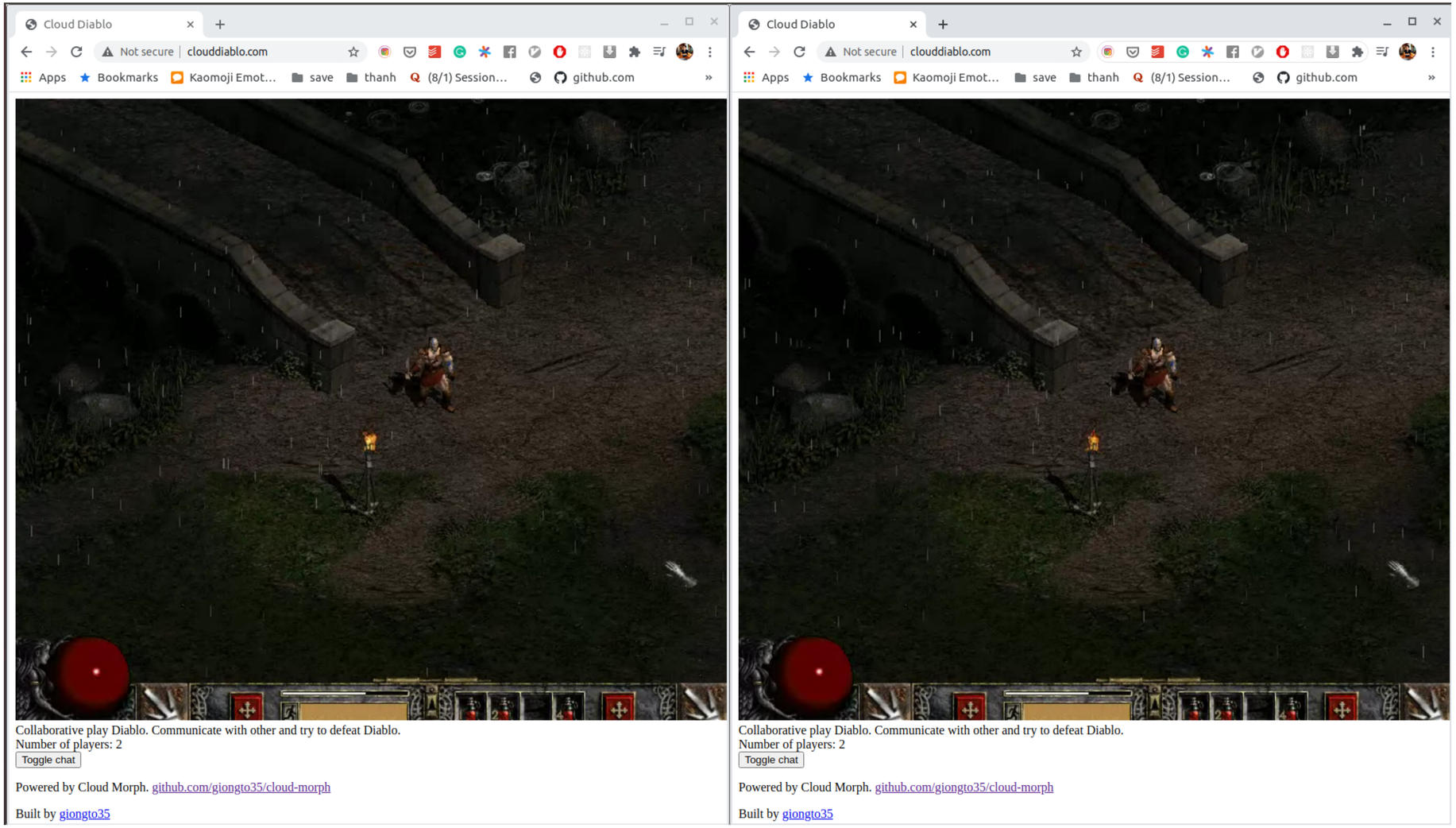
25

# ascii

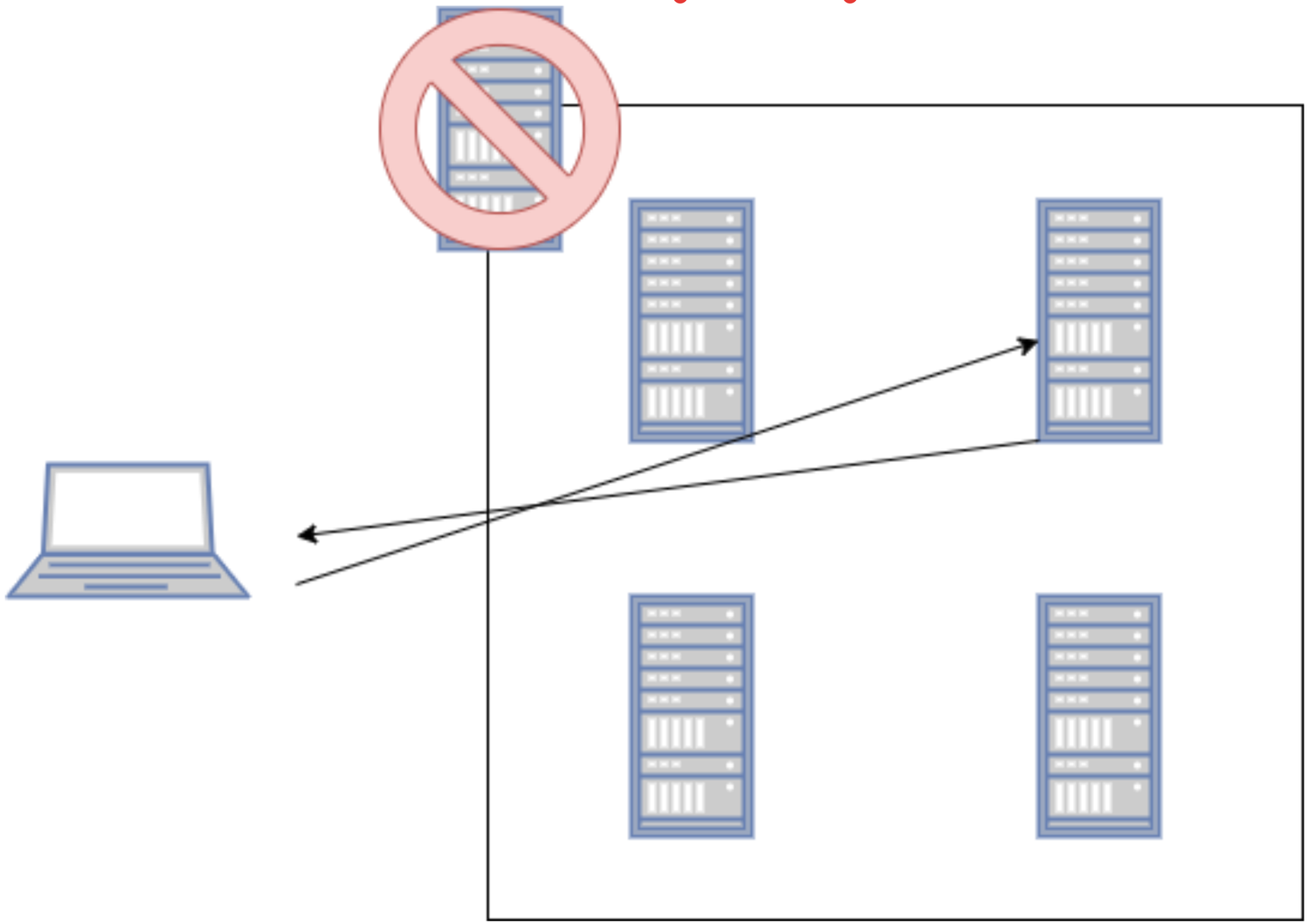


26

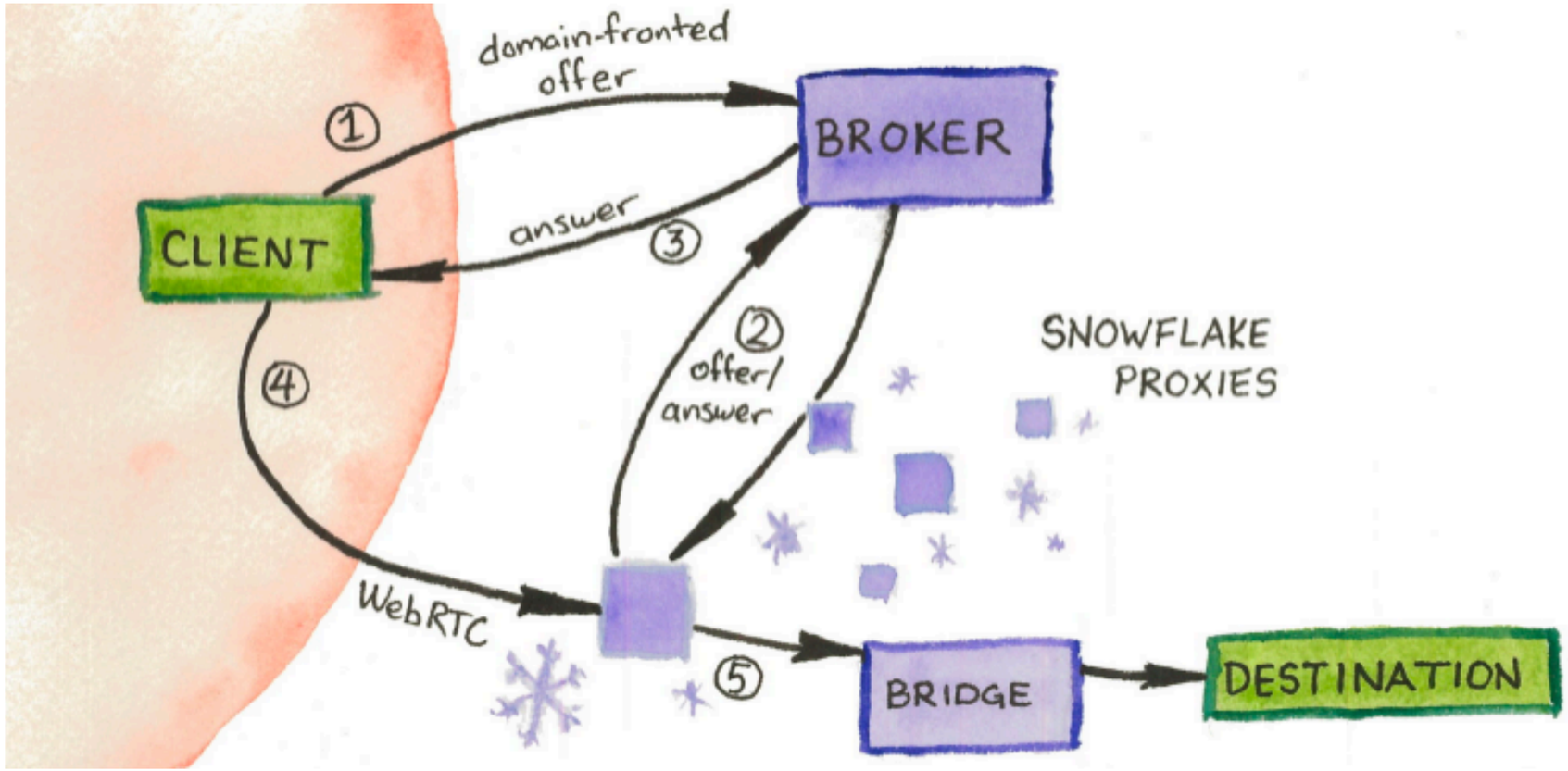
# cloud-morph



# ssh-p2p



# Snowflake

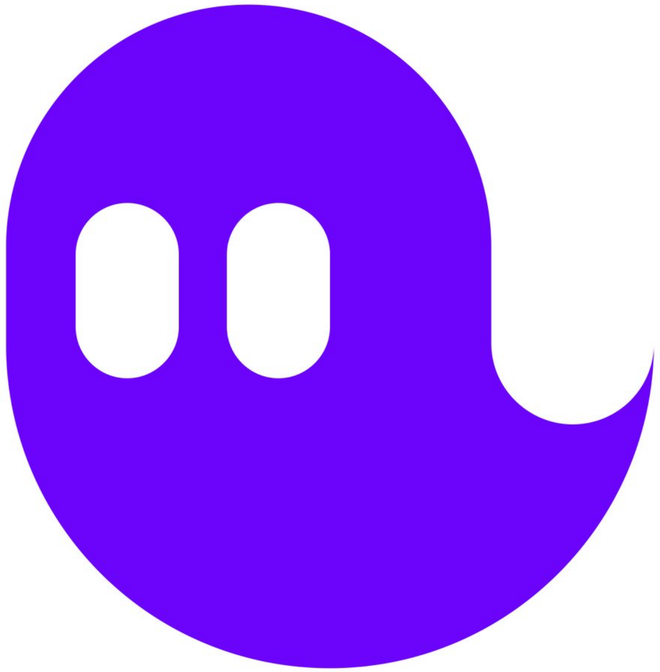

domain-fronted
offer
1
BROKER
CLIENT
answer
3
2
offer/
answer
SNOWFLAKE
PROXIES
4
WebRTC
5
BRIDGE
DESTINATION

# webwormhole.io

# s4y/space

31

# Project Lightspeed



# Even more projects

GitHub.com/pion/awesome-pion

We want to share your projects!

# Pion needs you

Empower those helping the internet!

Gain deep WebRTC knowledge

A fun challenge where you pick the goals

github.com/pion

pion.ly/slack

twitter.com/_pion