



Streaming functions



C.C. @ YoMo

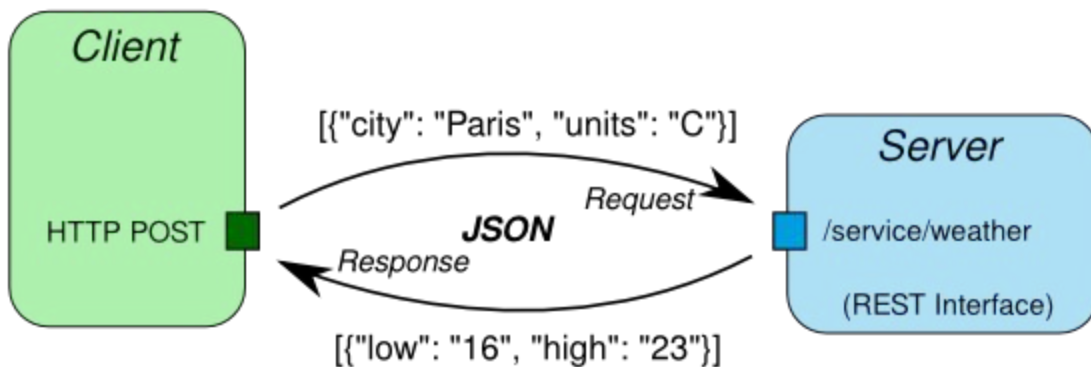
<https://github.com/yomorun/yomo>

2022 Sep

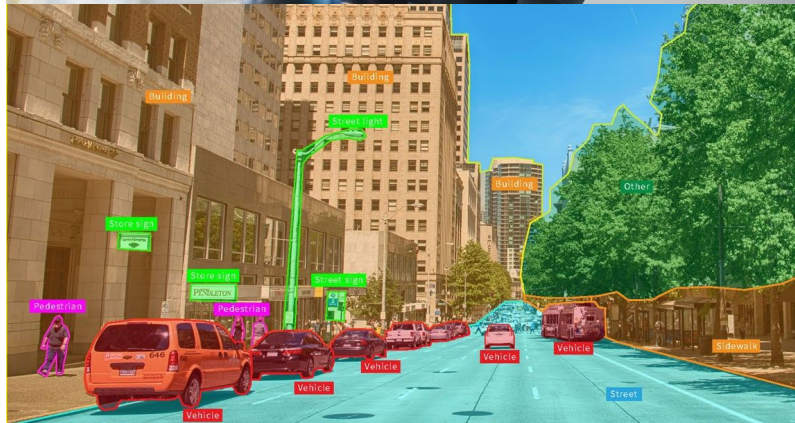
We familiar with RESTful



JSON / REST / HTTP



JSON -> Bytes stream & Requirements for low latency



Self driving cars



Facial Recognition

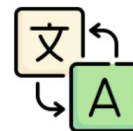


video stream

Artificial voices (Siri,
Google home)



Language Translation



audio stream

Detecting credit card Fraud

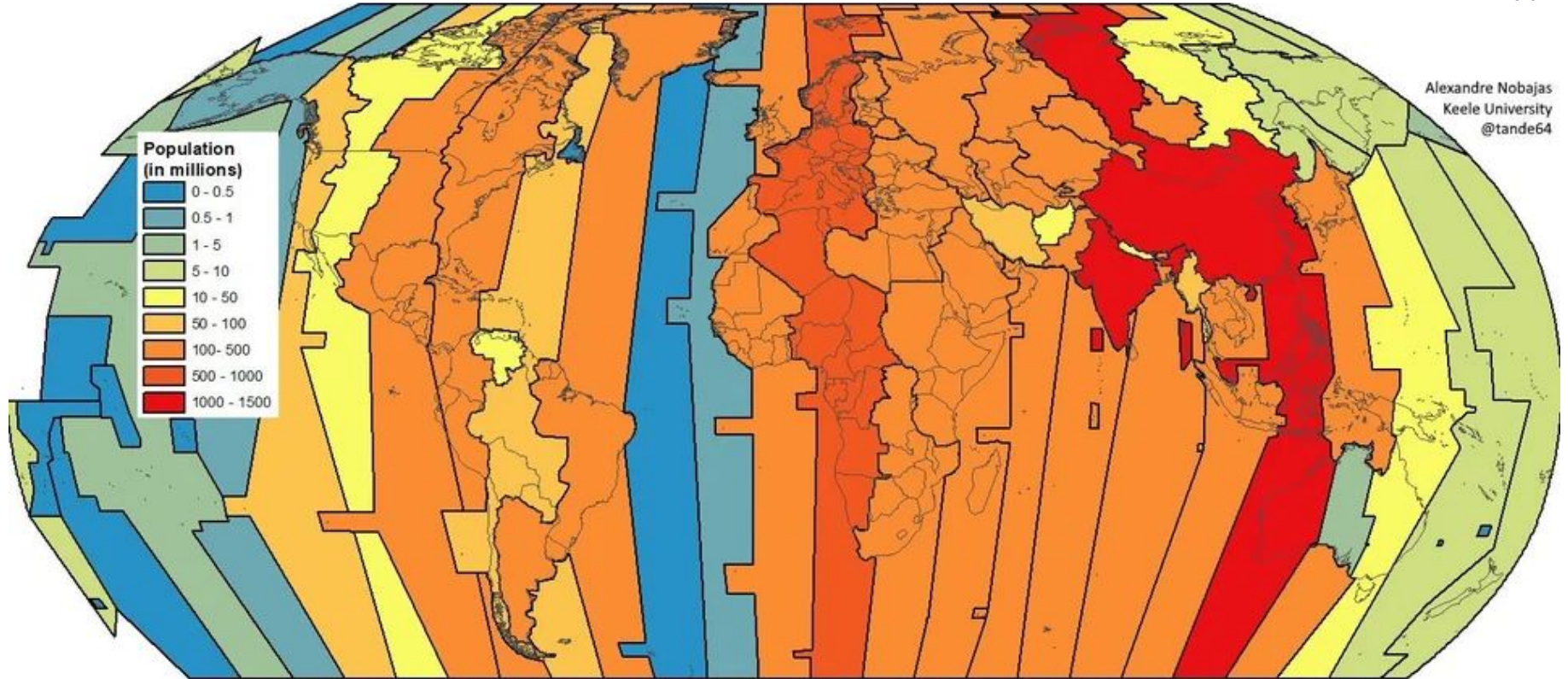


Recommendation
Engines



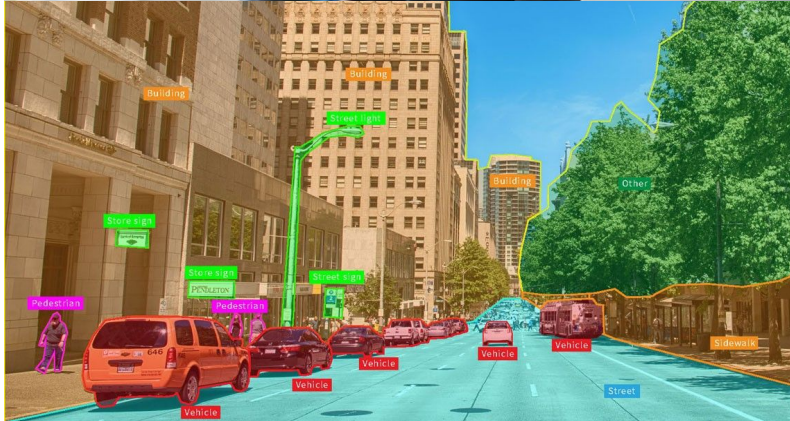
behavior sequences

Total population by time zone



Alexandre Nobajas
Keele University
@tande64

How to solve the problem



Self driving cars



Facial Recognition

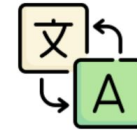


video stream

Artificial voices (Siri,
Google home)



Language Translation



audio stream

Detecting credit card Fraud



Recommendation
Engines



behavior sequences

Key 1: latency matters



Excellent

> -50 dBm



Good

-50 to -60 dBm



Fair

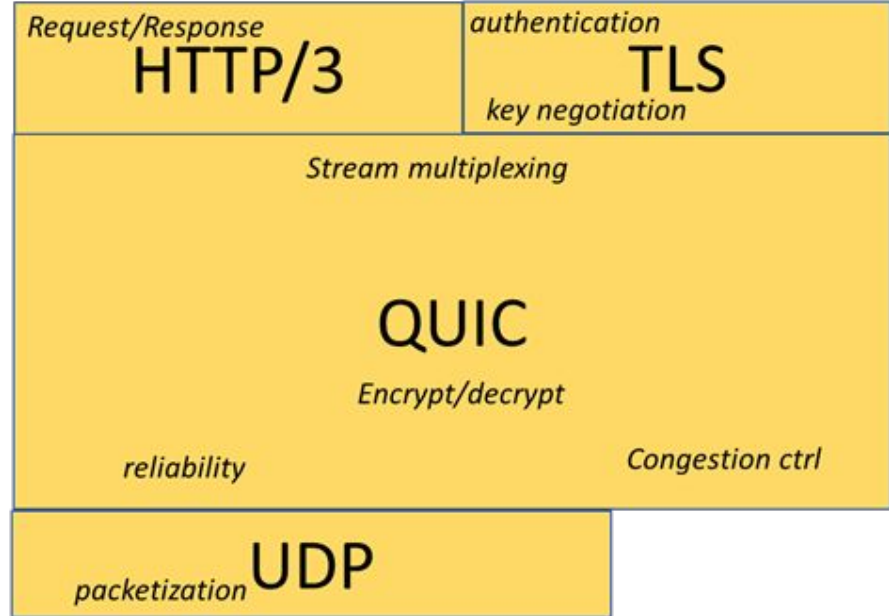
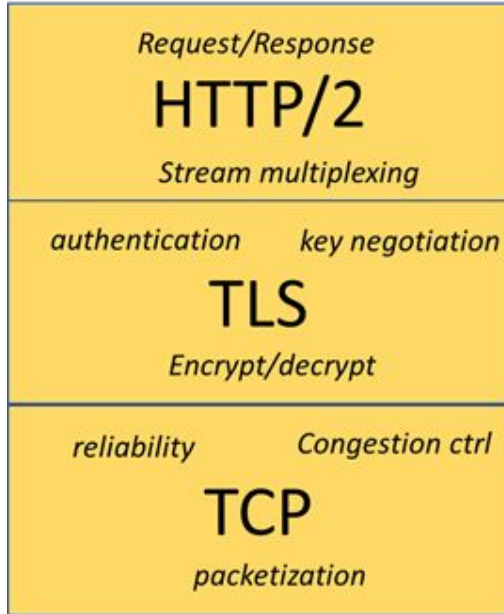
-60 to -70 dBm



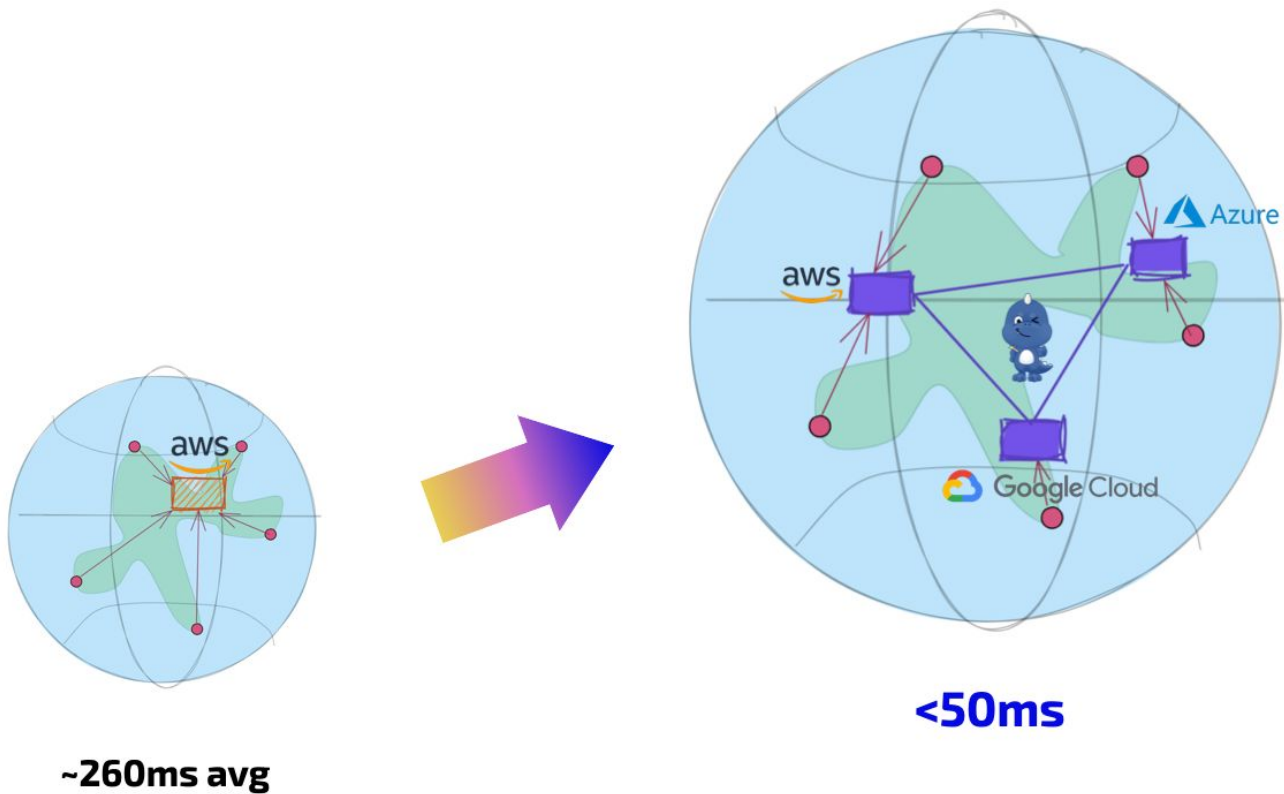
Weak

< -70 dBm

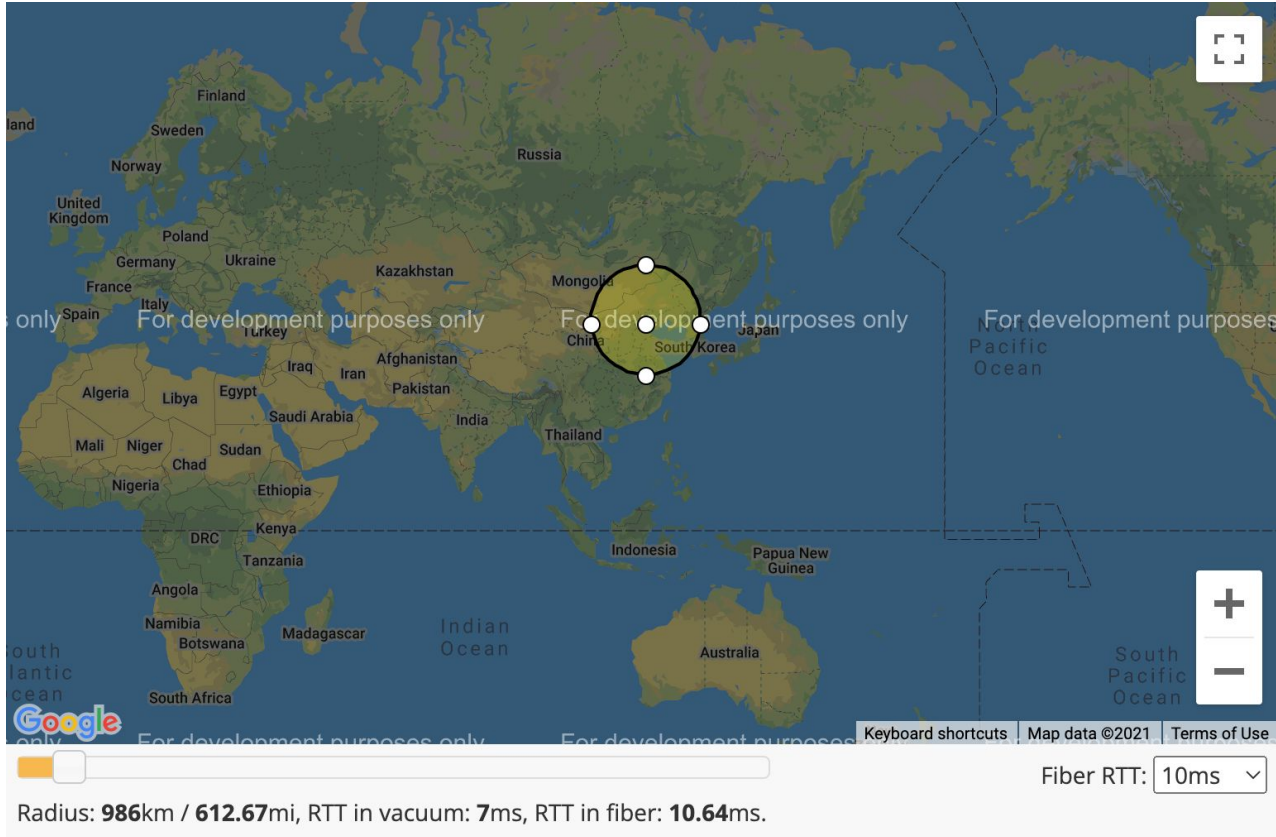
RFC 9000 - QUIC protocol



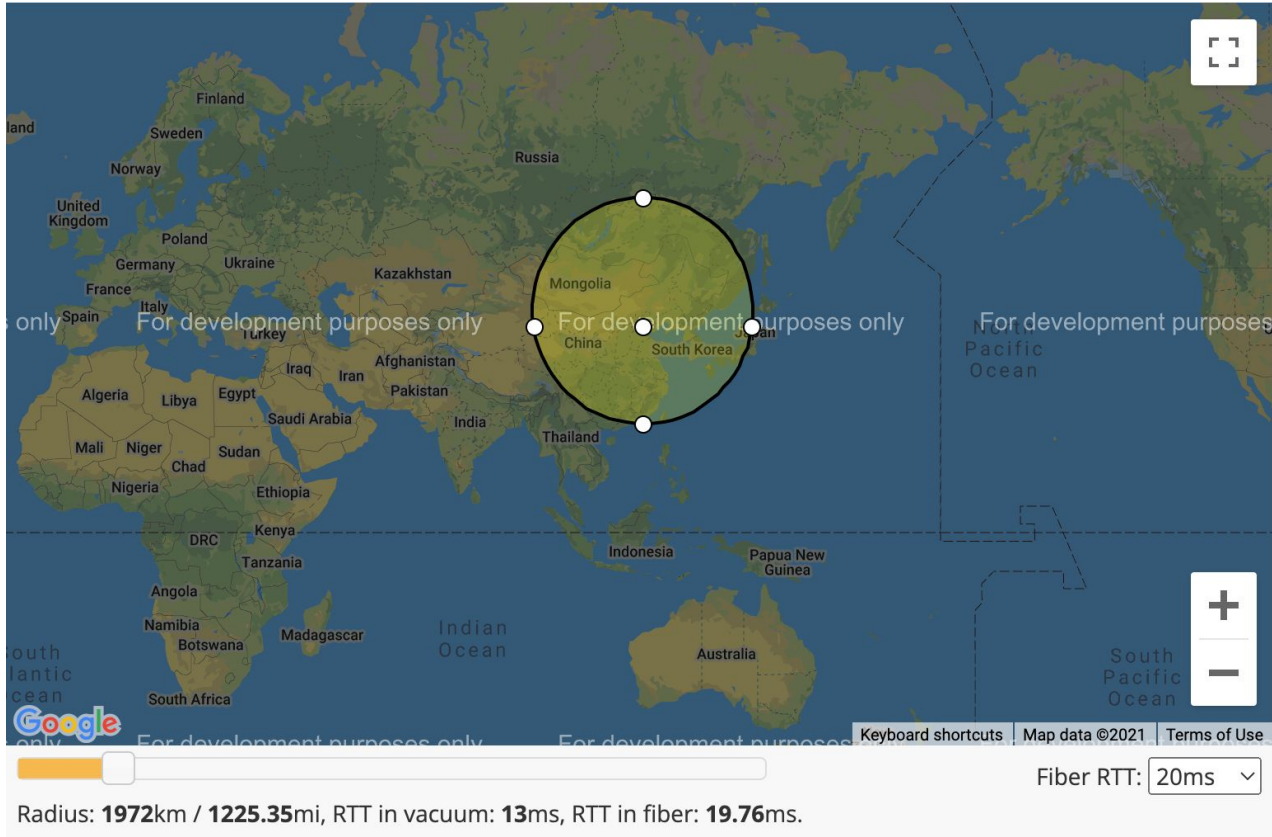
Key 2: Geo-distributed & Global Workloads



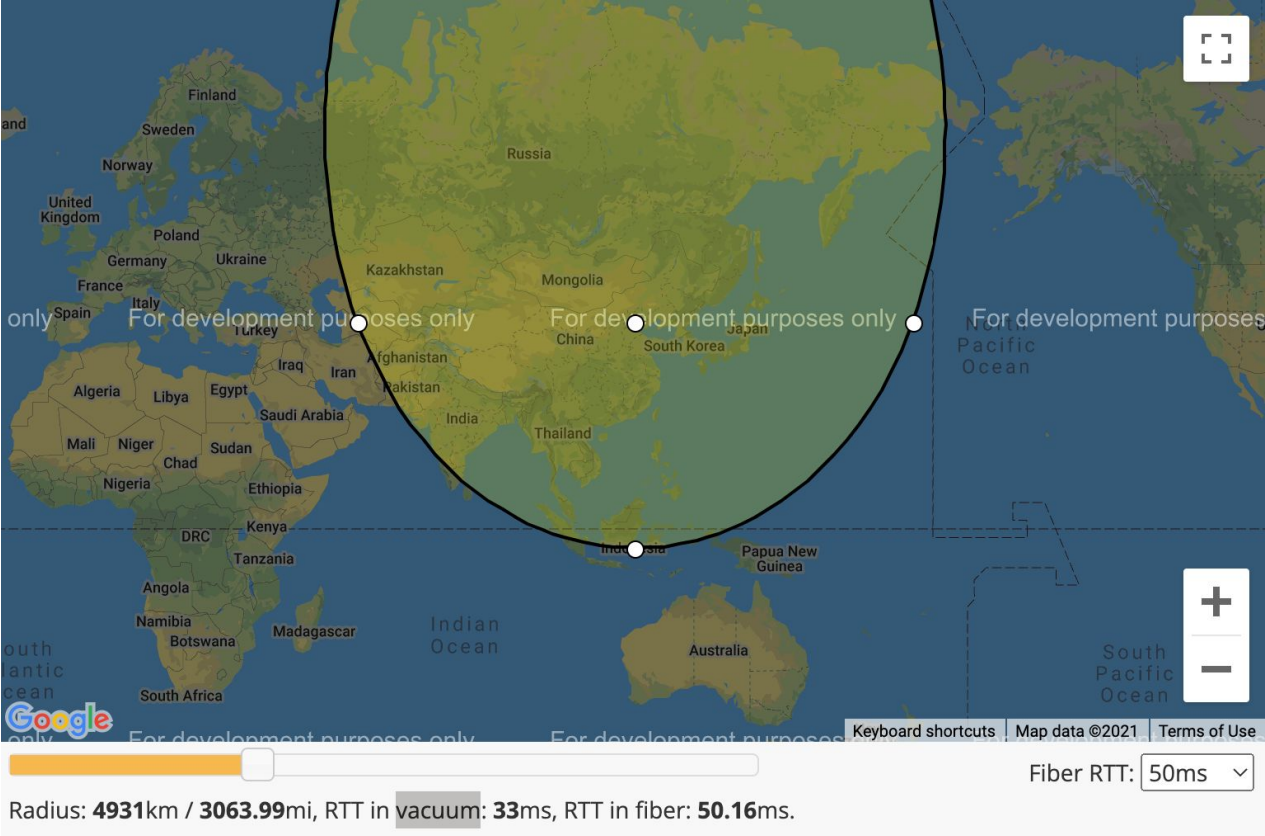
10ms



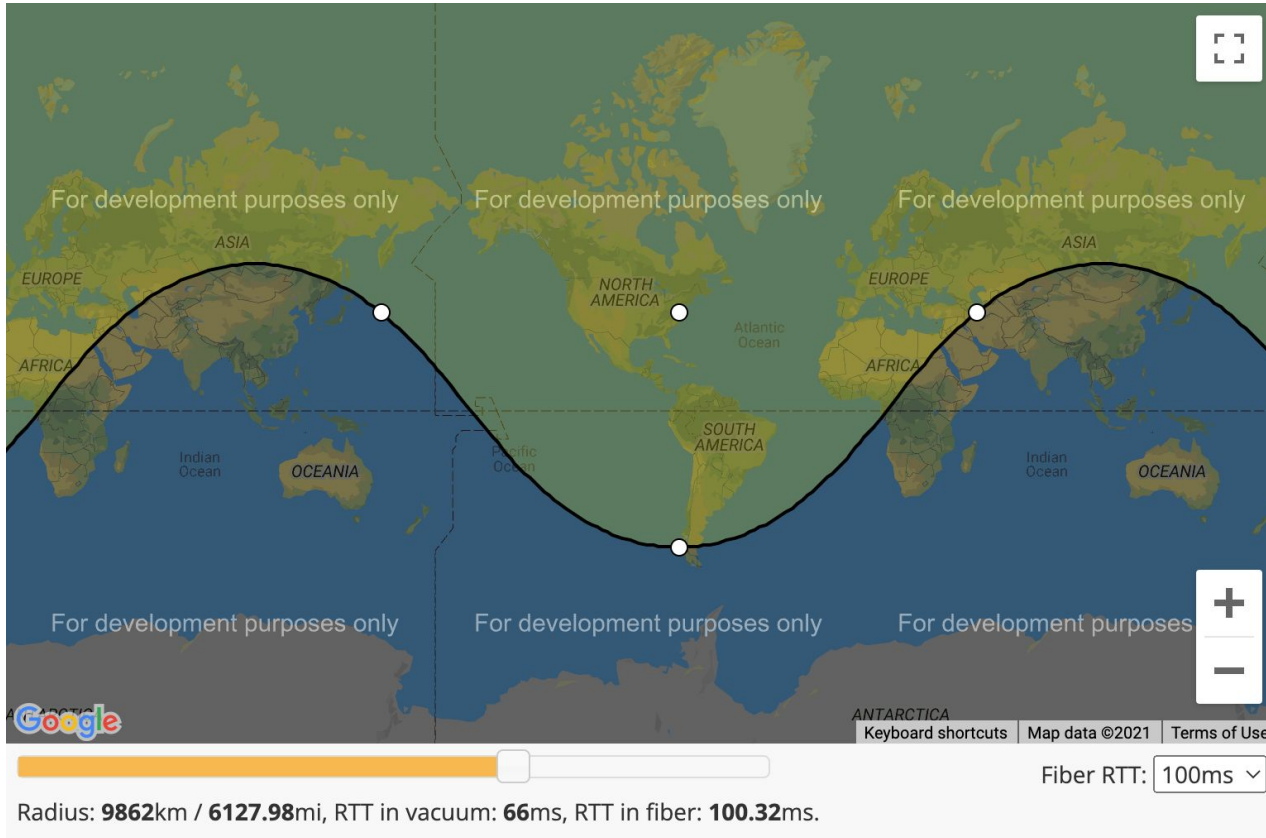
20ms



50ms



100ms



Key 3: Developer friendly



Unix Pipeline

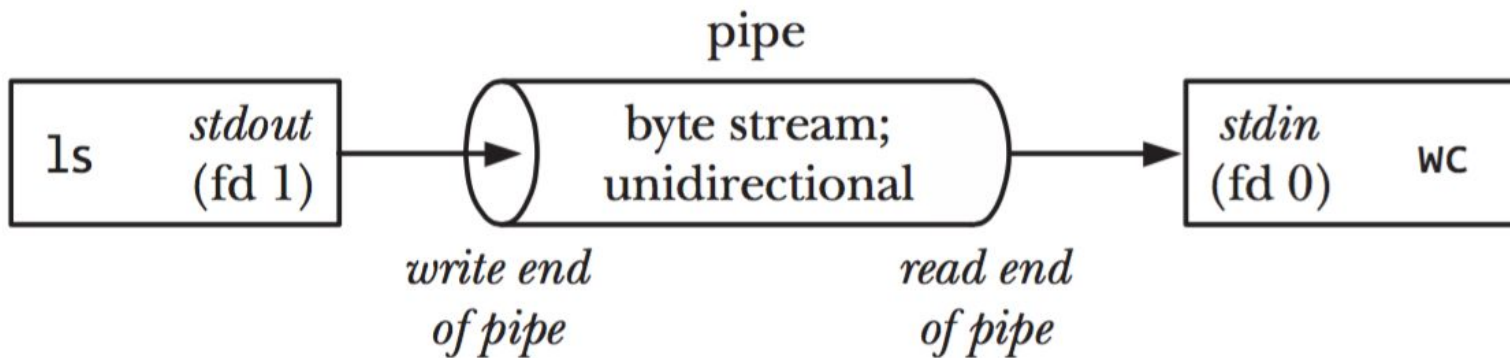
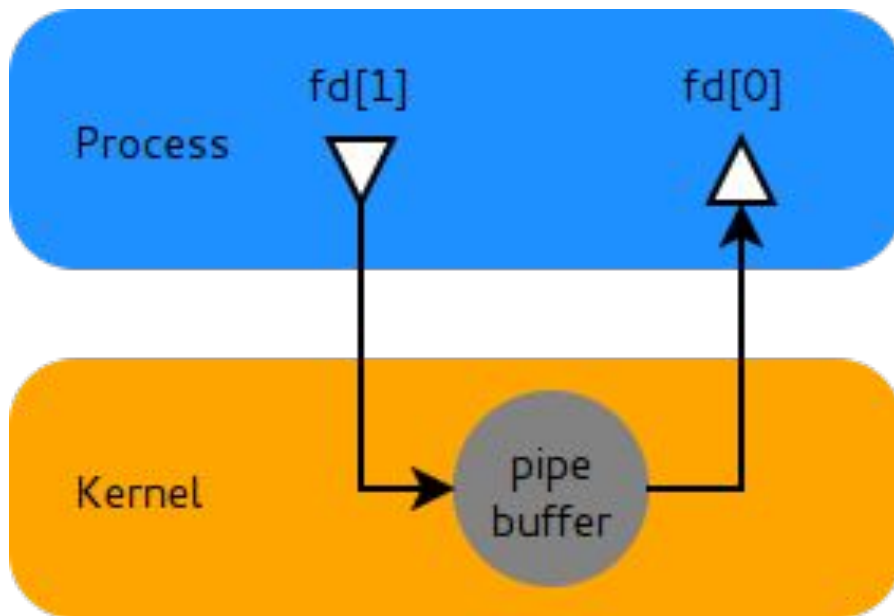


Figure 44-1: Using a pipe to connect two processes

Unix Pipeline

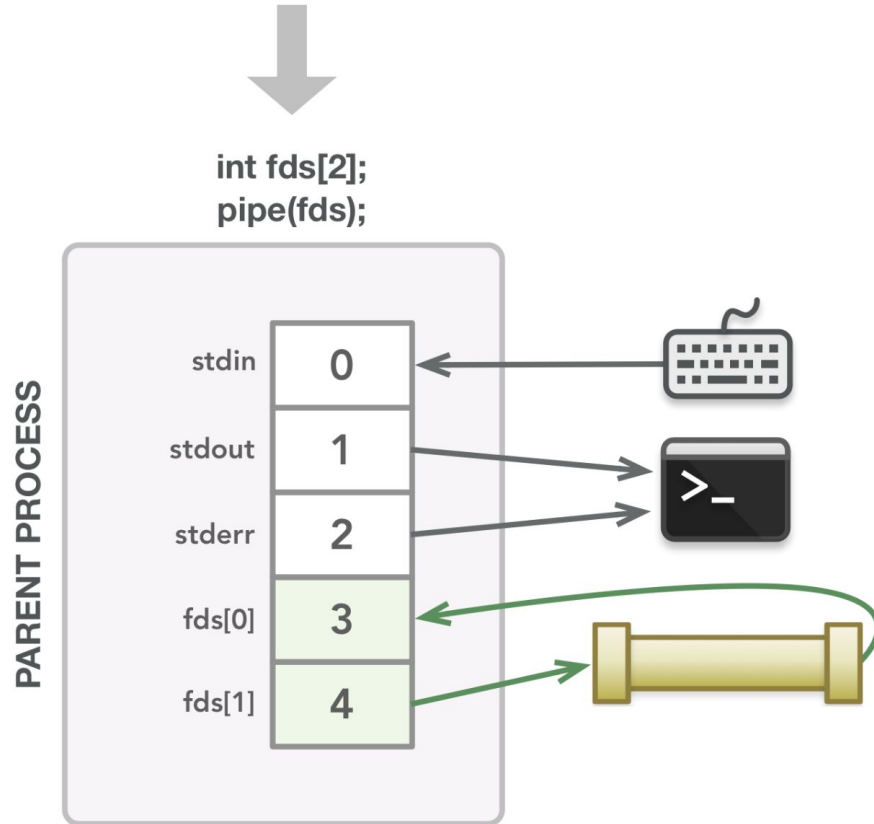


Unix Pipeline

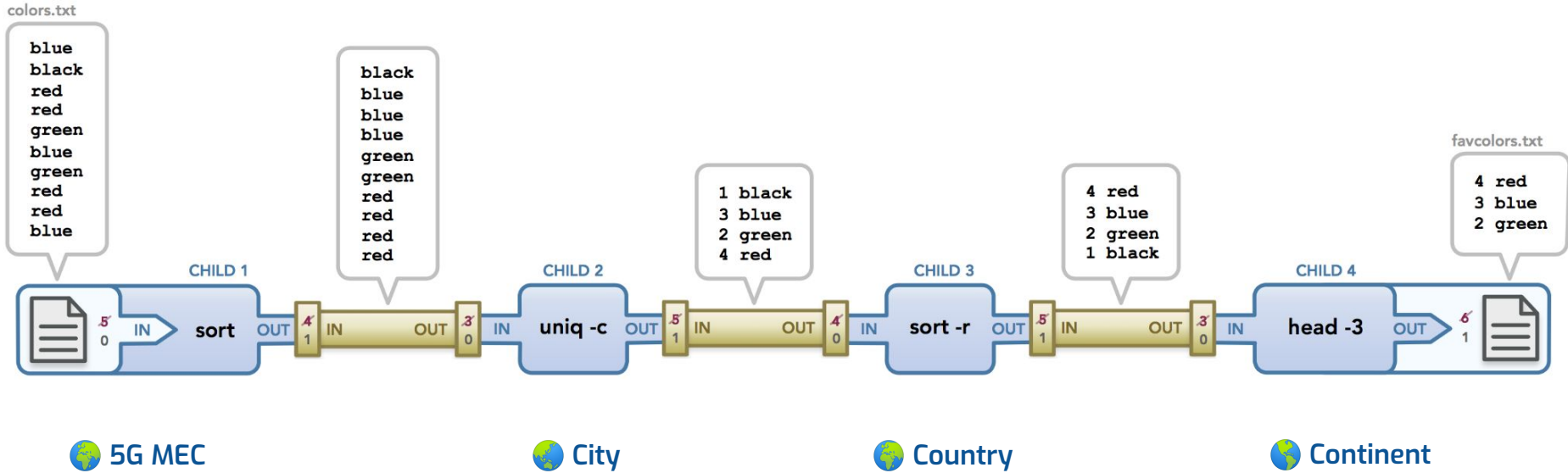


```
fanweixiao@CCs-RMBP2019:~  
~ → cat ~/_wrk/yomo/zipper.go | sort | uniq | grep func  
go func(dsZipper *zipper) {  
func (z *zipper) AddDownstreamZipper(downstream Zipper) error {  
func (z *zipper) Addr() string {  
func (z *zipper) Close() error {  
func (z *zipper) ConfigMesh(url string) error {  
func (z *zipper) ConfigWorkflow(conf string) error {  
func (z *zipper) ListenAndServe() error {  
func (z *zipper) RemoveDownstreamZipper(downstream Zipper) error {  
func (z *zipper) Stats() int {  
func (z *zipper) configWorkflow(config *config.WorkflowConfig) error {  
func NewDownstreamZipper(name string, opts ...Option) Zipper {  
func NewZipper(conf string) (Zipper, error) {  
func NewZipperWithOptions(name string, opts ...Option) Zipper {  
func createZipperServer(name string, addr string) *zipper {  
~ → █
```

Unix Pipeline



YoMo - Unix pipeline at internet scale



Core Concepts of YoMo



Source

source

Connect data to YoMo



Zipper

zipper

Routing data to process unit



Stream
Functions

sfn

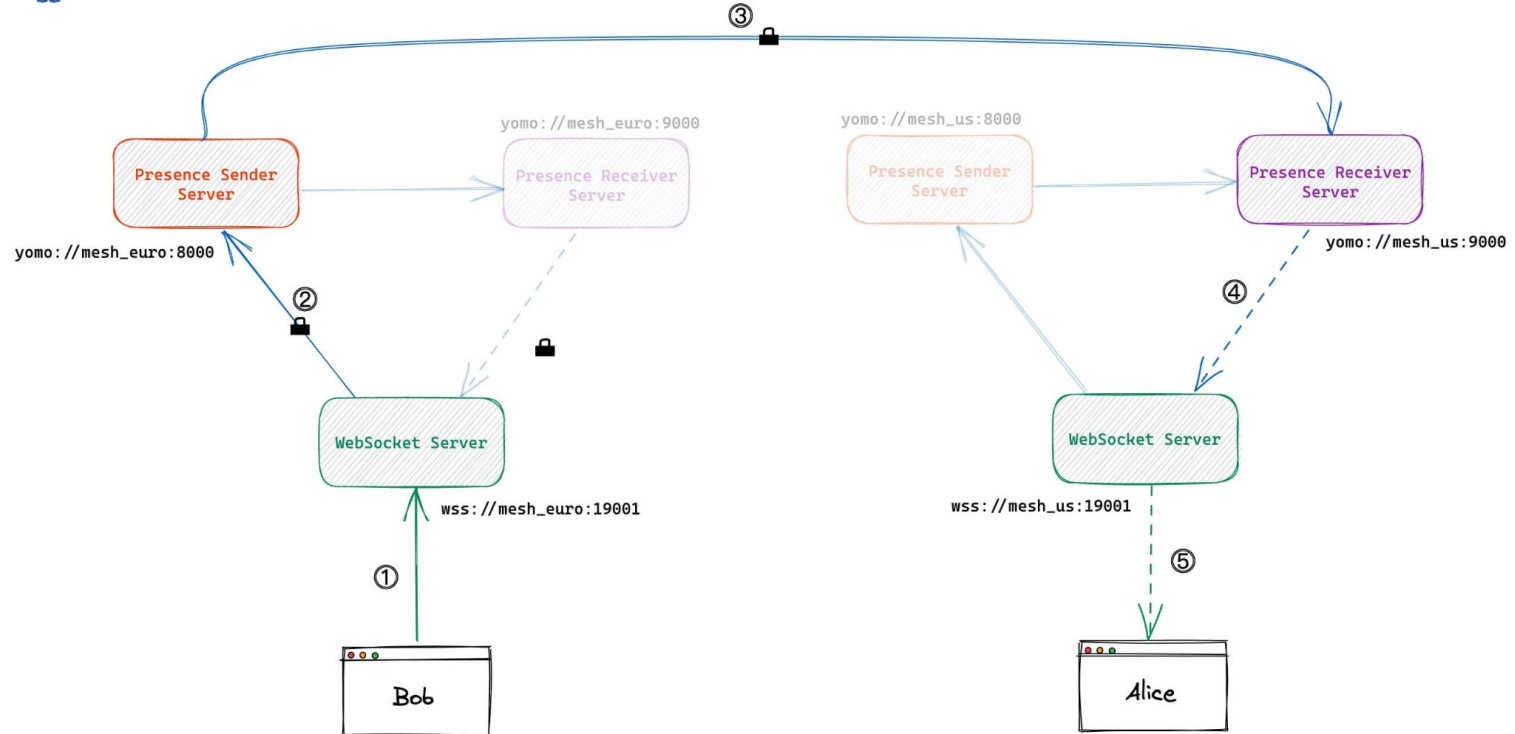
Processing data



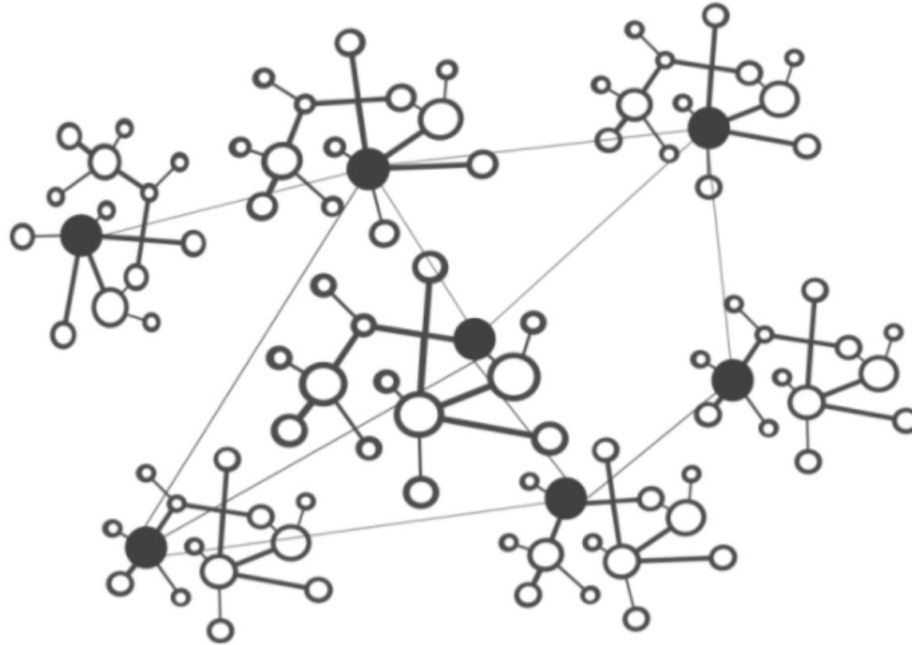
Cross region state-syncing



2. When Bob and Alice are in different mesh regions

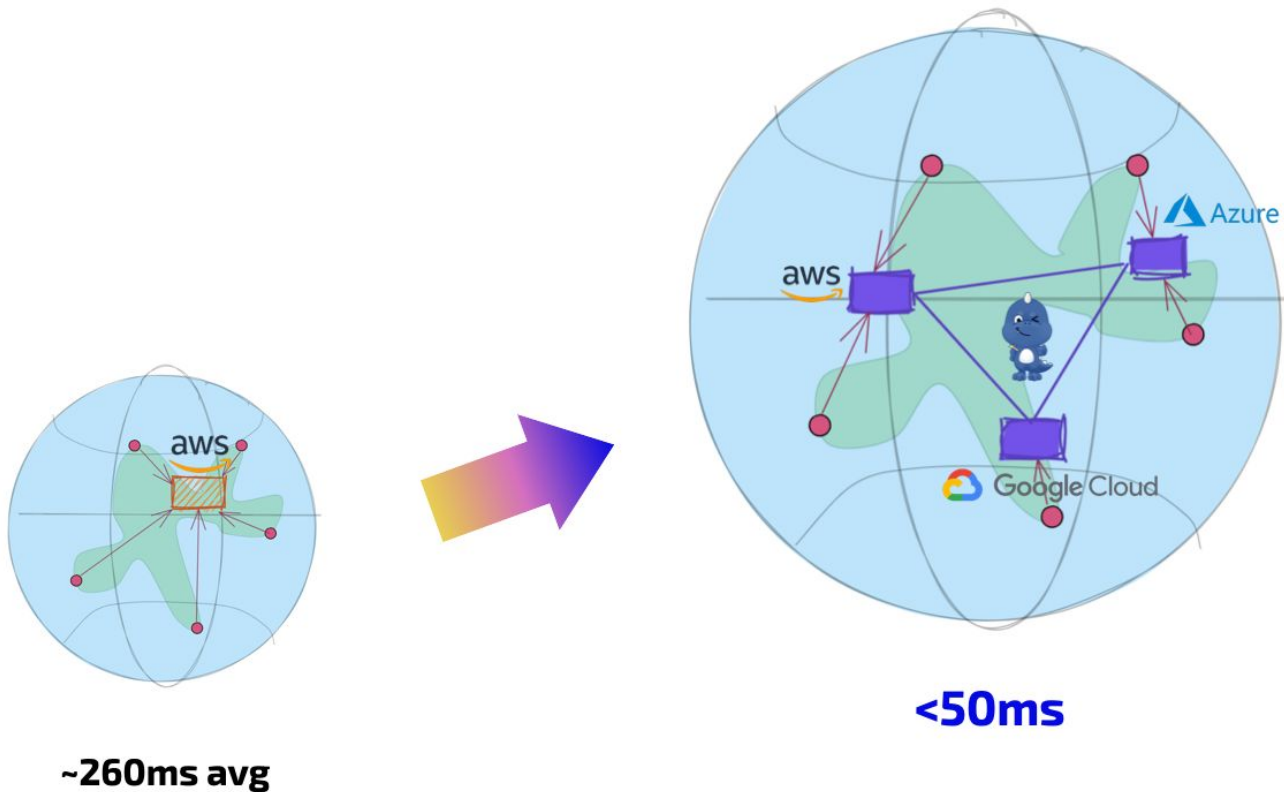


Self-governance



The Internet - A Network of Networks

Self-governance, provisioning close to users



△ **Introduce WebAssembly to YoMo Stream Function**

Implement stream function by WebAssembly



```
..[fanweixiao@CCs-MacBook-Pro] - [~/_wrk/webtransport-wasm-serverless-demo] - [Thu Sep 22, 01:14]
..[$] <(git)-[main]-> exa -Tl
drwxr-xr-x  - fanweixiao 22 Sep 01:14 .
.rw-r--r-- 2.4k fanweixiao 22 Sep 01:07 Cargo.lock
.rw-r--r-- 284 fanweixiao 22 Sep 01:05 Cargo.toml
.rw-r--r-- 1.1k fanweixiao 22 Sep 01:05 LICENSE
.rw-r--r-- 352 fanweixiao 22 Sep 01:05 README.md
drwxr-xr-x  - fanweixiao 22 Sep 01:05 src
.rw-r--r-- 2.0k fanweixiao 22 Sep 01:05 lib.rs
```



```

22  #[no_mangle]
23  pub extern "C" fn yomo_handler(input_length: usize) {
24      println!("yomo wasm sfm received {} bytes", input_length);
25
26      // load input data
27      let mut input = Vec::with_capacity(input_length);
28      unsafe {
29          yomo_load_input(input.as_mut_ptr());
30          input.set_len(input_length);
31      }
32
33      // parse Vec<u8> to String
34      let str = str::from_utf8(&input).unwrap();
35      println!("> received: {}", str);
36
37      // Step 1: Deserilize input data as struct `Message`
38      // deserilize String to struct `Message`
39      let mut parse: Message = serde_json::from_slice(&input).unwrap();
40      // base64 decode the `data` field
41      let data = parse.data;
42      let res = decode(&data).unwrap();
43      let ddd = str::from_utf8(&res).unwrap();
44      println!("> data: {:?}", ddd);
45
46      // Step 2: Process the input data, we just uppercase all inputs here
47      let response = ddd.to_ascii_uppercase();
48      // then base64 encode the response
49      parse.data = encode(response.to_string());
50
51      // Step 3: Serilize the response as Vec<u8>
52      let result = serde_json::to_string(&parse).unwrap();
53      println!("< response: {}", result);
54      let output = result.as_bytes();
55
56      // dump output data and set tag to 0x31 to response to https://webtransport.day/serve
57      rless
58      unsafe {
59          yomo_dump_output(0x31, output.as_ptr(), output.len());
60      }
61  }

```

cargo build --target wasm32-wasi



```
..[fanweixiao@CCs-MacBook-Pro] - [~/_wrk/webtransport-wasm-serverless-demo] - [Thu Sep 22, 01:06]
..[$] <( (git)-[main]-)> cargo build --release --target wasm32-wasi

  Updating crates.io index
  Downloaded syn v1.0.100
  Downloaded 1 crate (236.7 KB) in 1.76s
  Compiling proc-macro2 v1.0.43
  Compiling unicode-ident v1.0.4
  Compiling quote v1.0.21
  Compiling syn v1.0.100
  Compiling serde v1.0.144
  Compiling serde_json v1.0.85
  Compiling serde_derive v1.0.144
  Compiling ryu v1.0.11
  Compiling itoa v1.0.3
  Compiling base64 v0.13.0
  Compiling sfn v0.1.0 (/Users/fanweixiao/_wrk/webtransport-wasm-serverless-demo)
  Finished release [optimized] target(s) in 31.34s
```

yomo run -r wasmedge /path/to/sfn.wasm



```
..[fanweixiao@CCs-MacBook-Pro] - [~/ wrk/webtransport-wasm-serverless-demo] - [Thu Sep 22, 01:07]
..[$] <( (git)-[main]-)> yomo run -r wasmedge target/wasm32-wasi/release/sfn.wasm
i YoMo Stream Function file: target/wasm32-wasi/release/sfn.wasm
⌚ Create YoMo Stream Function instance...
i Starting YoMo Stream Function instance with executable file: target/wasm32-wasi/release/sfn.wasm
m. Zipper: [fra.algr.yomo.dev:9001 sgp.algr.yomo.dev:9001 usa.algr.yomo.dev:9001].
i YoMo Stream Function is running...
2022-09-22 01:08:59.132 [core:client] use credential: [app-key-secret]
2022-09-22 01:09:01.195 [core:client] ♥ [cctest][ne6X_WQEovetZjW3U893S]([:]:54609) is connected t
o YoMo-Zipper fra.algr.yomo.dev:9001
2022-09-22 01:09:01.196 [core:client] use credential: [app-key-secret]
2022-09-22 01:09:02.852 [core:client] ♥ [cctest][2KoBfYLg8fFN8\AIIVCLhZ]([:]:59821) is connected t
o YoMo-Zipper sgp.algr.yomo.dev:9001
2022-09-22 01:09:02.852 [core:client] use credential: [app-key-secret]
2022-09-22 01:09:03.394 [core:client] ♥ [cctest][MLu6TUB_5b_svMqI_Kc2Z]([:]:52076) is connected t
o YoMo-Zipper usa.algr.yomo.dev:9001
```



WasmEdgeRuntime

Wasmtime

https://github.com/fanweixiao/webtransport... Public WebTransport E...
AllegroPulse Orbit GDrive Cloudcraft DeepL
Get your free WebTransport Dev Server on [AllegroPulse](https://webtransport.day/serverless)

Establish WebTransport connection

zipper:

app key:

app secret:

url:

Send data over WebTransport

Send a datagram
 Open a unidirectional stream
 Open a bidirectional stream

Event log

- Initiating connection...
- Connection ready.
- Datagram writer ready.
- Datagram reader ready.
- Sent datagram: hello akraino
- Datagram received: HELLO AKRAINO
- Datagram received: HELLO AKRAINO
- Opened bidirectional stream #1 with data: hi there, nextarch
- Received data on stream #1: HI THERE, NEXTARCH
- Received data on stream #1: HI THERE, NEXTARCH

```
yomo run -r wasmedge target/wasm32-wasi/release/sfn.wasm
Compiling itoa v1.0.3
Compiling base64 v0.13.0
Compiling sfn v0.1.0 (/Users/fanweixiao/_wrk/webtransport-wasm-serverless-demo)
Finished release [optimized] target(s) in 31.34s
..[fanweixiao@CCs-MacBook-Pro] - [~/_wrk/webtransport-wasm-serverless-demo] - [Thu Sep 22, 01:07]
..[$] <( git)-[main]-> yomo run -r wasmedge target/wasm32-wasi/release/sfn.wasm
i YoMo Stream Function file: target/wasm32-wasi/release/sfn.wasm
h Create YoMo Stream Function instance...
i Starting YoMo Stream Function instance with executable file: target/wasm32-wasi/release/sfn.wasm.
m. Zipper: [fra.algr.yomo.dev:9001 sgp.algr.yomo.dev:9001 usa.algr.yomo.dev:9001].
i YoMo Stream Function is running...
2022-09-22 01:08:59.132 [core:client] use credential: [app-key-secret]
2022-09-22 01:09:01.195 [core:client] ♥ [cctest] [ne6X_WQEovetZjw3U893S] (:::54609) is connected
o YoMo-Zipper fra.algr.yomo.dev:9001
2022-09-22 01:09:01.196 [core:client] use credential: [app-key-secret]
2022-09-22 01:09:02.852 [core:client] ♥ [cctest] [2K0BfYLg8fFN8lAIVCLhZ] (:::59821) is connected
o YoMo-Zipper sgp.algr.yomo.dev:9001
2022-09-22 01:09:02.852 [core:client] use credential: [app-key-secret]
2022-09-22 01:09:03.394 [core:client] ♥ [cctest] [MLu6TUB_5b_svMqI_Kc2Z] (:::52076) is connected
o YoMo-Zipper usa.algr.yomo.dev:9001

yomo wasm sfn received 72 bytes
> received: {"meta":{"mode":"datagram","stream_id":0},"data":"aGVsbG8gYWtyYWlubw==" }
> data: "hello akraino"
< response: {"meta":{"mode":"datagram","stream_id":0},"data":"SEVMTE8gQUtSQL0TW==" }

yomo wasm sfn received 74 bytes
> received: {"meta":{"mode":"stream","stream_id":4},"data":"aGkgdGhlcmsIG5leHRhcmNo" }
> data: "hi there, nextarch"
< response: {"meta":{"mode":"stream","stream_id":4},"data":"SEkgVEhFUKUsIE5FWFRBUKNI" }
```

Have a try: <https://webtransport.day/serverless>

<https://yomo.run>


```
..[fanweixiao@CCs-MacBook-Pro] - [~/_wrk/yomo/example/1-pipeline] -  
[Thu Sep 22, 01:25]  
..[$] <(git)-[master]-> cat /dev/urandom | go run source/pipe.go
```

```
2022-09-22 01:25:41.003 [core:client] use credential: [none]  
2022-09-22 01:25:41.006 [core:client] ♥ [source-pipe][tCTYCw0mDEIb  
cr4dMosY6]([:]:58961) is connected to YoMo-Zipper localhost:9000  
2022-09-22 01:25:41.006 Read: 0x17 0x61 0x2c 0x38, written: 4  
2022-09-22 01:25:41.507 Read: 0x7b 0x54 0xa6 0x54, written: 4  
2022-09-22 01:25:42.007 Read: 0xe0 0x15 0x67 0x3b, written: 4  
2022-09-22 01:25:42.508 Read: 0xe0 0x7a 0x94 0x53, written: 4  
2022-09-22 01:25:43.009 Read: 0x7d 0x14 0x6f 0x91, written: 4  
2022-09-22 01:25:43.510 Read: 0x5b 0x4a 0xd8 0xe9, written: 4  
2022-09-22 01:25:44.011 Read: 0xc8 0xae 0x84 0xe5, written: 4  
2022-09-22 01:25:44.512 Read: 0x3e 0x45 0x7e 0xc6, written: 4
```

```
..[fanweixiao@CCs-MacBook-Pro] - [~/_wrk/yomo/example/1-pipeline] -  
[Thu Sep 22, 01:24]  
..[$] <(git)-[master]-> yomo run -n rand serverless/rand.go
```

```
YoMo Stream Function file: serverless/rand.go  
Create YoMo Stream Function instance...  
Starting YoMo Stream Function instance with Name: rand. Zipper:  
[localhost:9000].  
YoMo Stream Function building...  
Success! YoMo Stream Function build.  
YoMo Stream Function is running...  
Run: /Users/fanweixiao/_wrk/yomo/example/1-pipeline/serverless/  
sl.yomo  
2022-09-22 01:24:46.004 [core:client] use credential: [none]  
2022-09-22 01:24:46.009 [core:client] ♥ [rand][m01jCCKpPV1NM8A9Ira  
WC]([:]:49639) is connected to YoMo-Zipper localhost:9000  
2022-09-22 01:25:41.007 Generate random uint32: 942432535 (0x17 0x6  
1 0x2c 0x38)  
2022-09-22 01:25:41.508 Generate random uint32: 1420186747 (0x7b 0x  
54 0xa6 0x54)  
2022-09-22 01:25:42.008 Generate random uint32: 996611552 (0xe0 0x1  
5 0x67 0x3b)  
2022-09-22 01:25:42.509 Generate random uint32: 1402239712 (0xe0 0x  
7a 0x94 0x53)  
2022-09-22 01:25:43.010 Generate random uint32: 2439976061 (0x7d 0x  
14 0x6f 0x91)  
2022-09-22 01:25:43.511 Generate random uint32: 3923266139 (0x5b 0x  
4a 0xd8 0xe9)  
2022-09-22 01:25:44.012 Generate random uint32: 3850677960 (0xc8 0x  
ae 0x84 0xe5)  
2022-09-22 01:25:44.513 Generate random uint32: 3330164030 (0x3e 0x  
45 0x7e 0xc6)
```





 **Thanks**



<https://github.com/yomorun/yomo>

2022 Sep