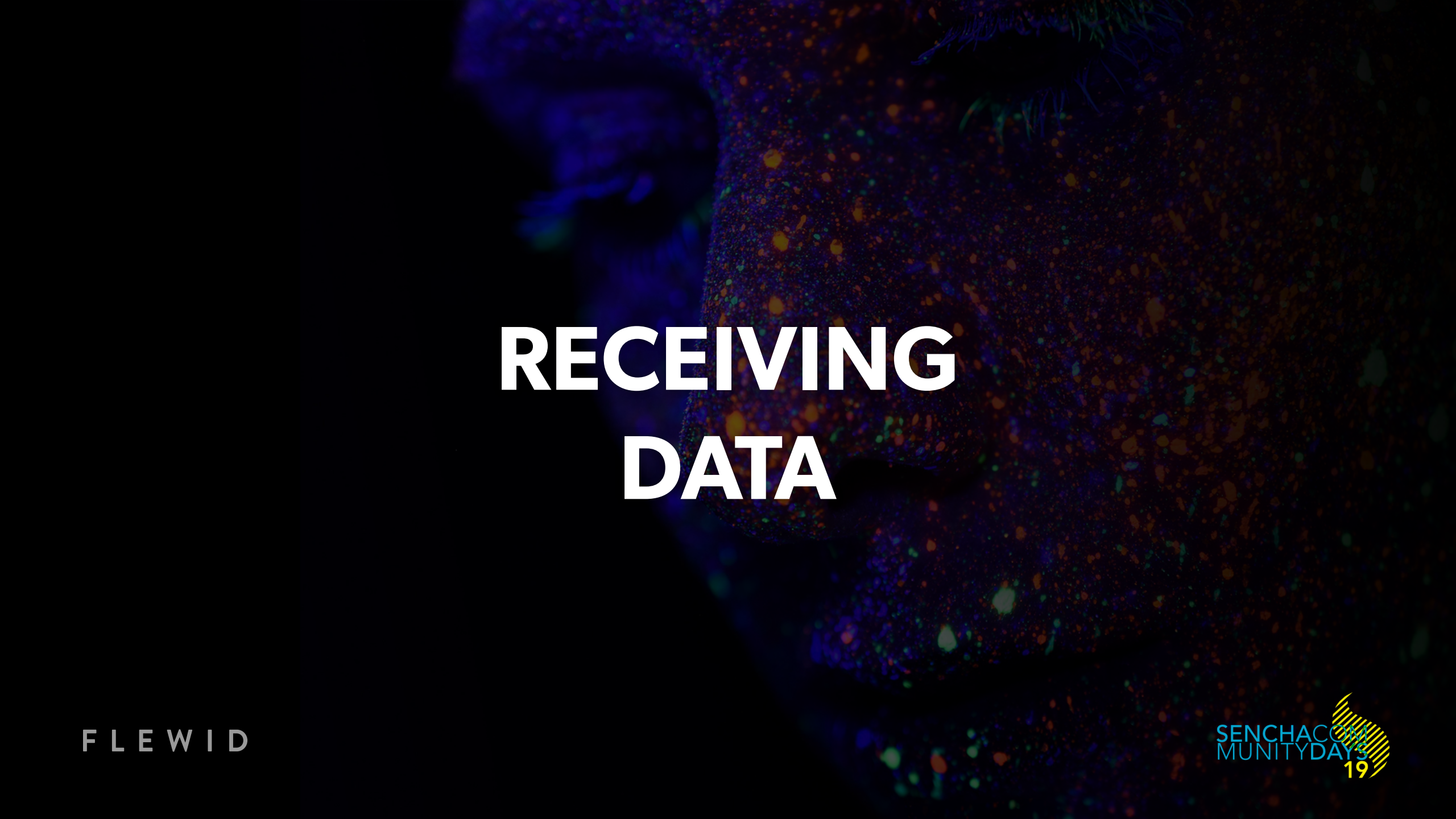
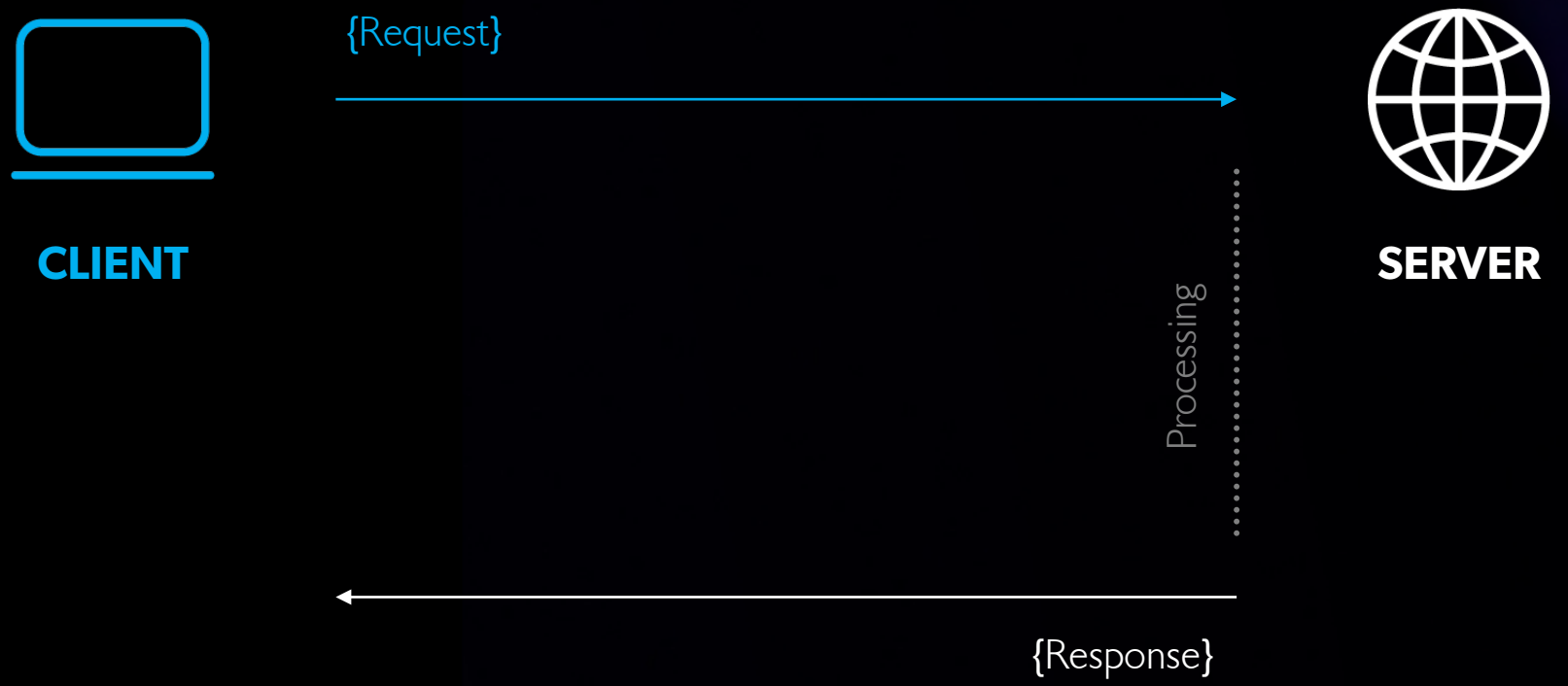

ENHANCE YOUR JAVASCRIPT APPLICATIONS WITH
WEBSOCKETS
SENCHACONMUNITYDAYS19


RECEIVING
DATA

HTTP



AJAX


{Request}
{Response}
{Request}
{Response}
{Request}
{Response}
CLIENT
SERVER


GETTING
UPDATES FROM
THE SERVER
FLEWID
SENCHACON
MUNITYDAYS
19

POLLING

CLIENT

Anything new?

Nope

Anything new?

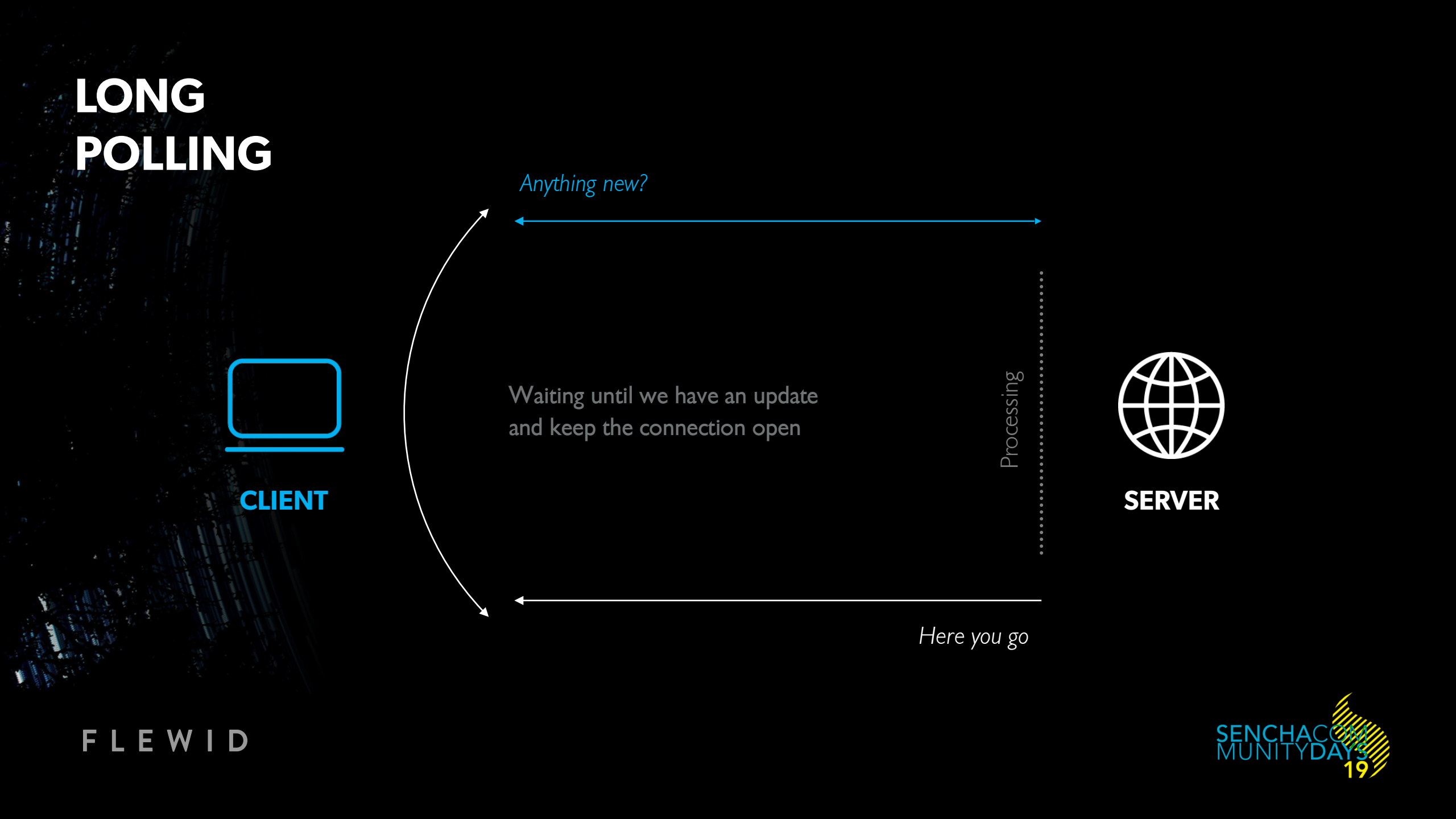
Nope

Anything new?

Ah, yeah, I have a new post, here you go!

SERVER

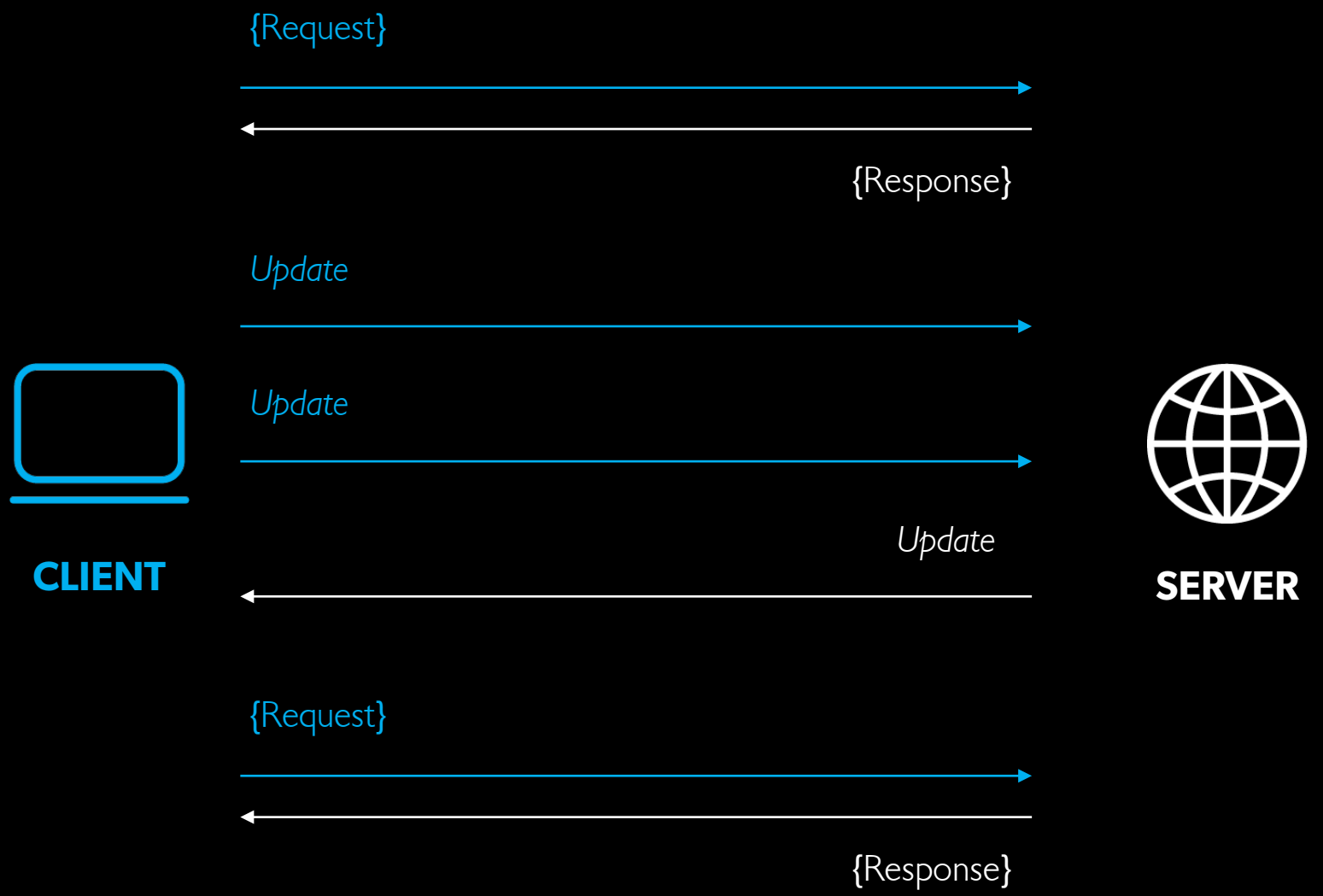
LONG POLLING



SERVER SIDE EVENTS

BIDIRECTIONAL COMMUNICATION

WEBSOCKETS


{Request}
{Response}
Update
Update
Update
CLIENT
SERVER
{Request}
{Response}

WEBSOCKETS

- Full-duplex bi-directional communication between client and server
- Only one HTTP request, then the connection is upgraded into a TCP connection
- Headers are only sent once (see above)
- Capable of sending binary data
- Standardised (uses protocol handshakes)

Performance

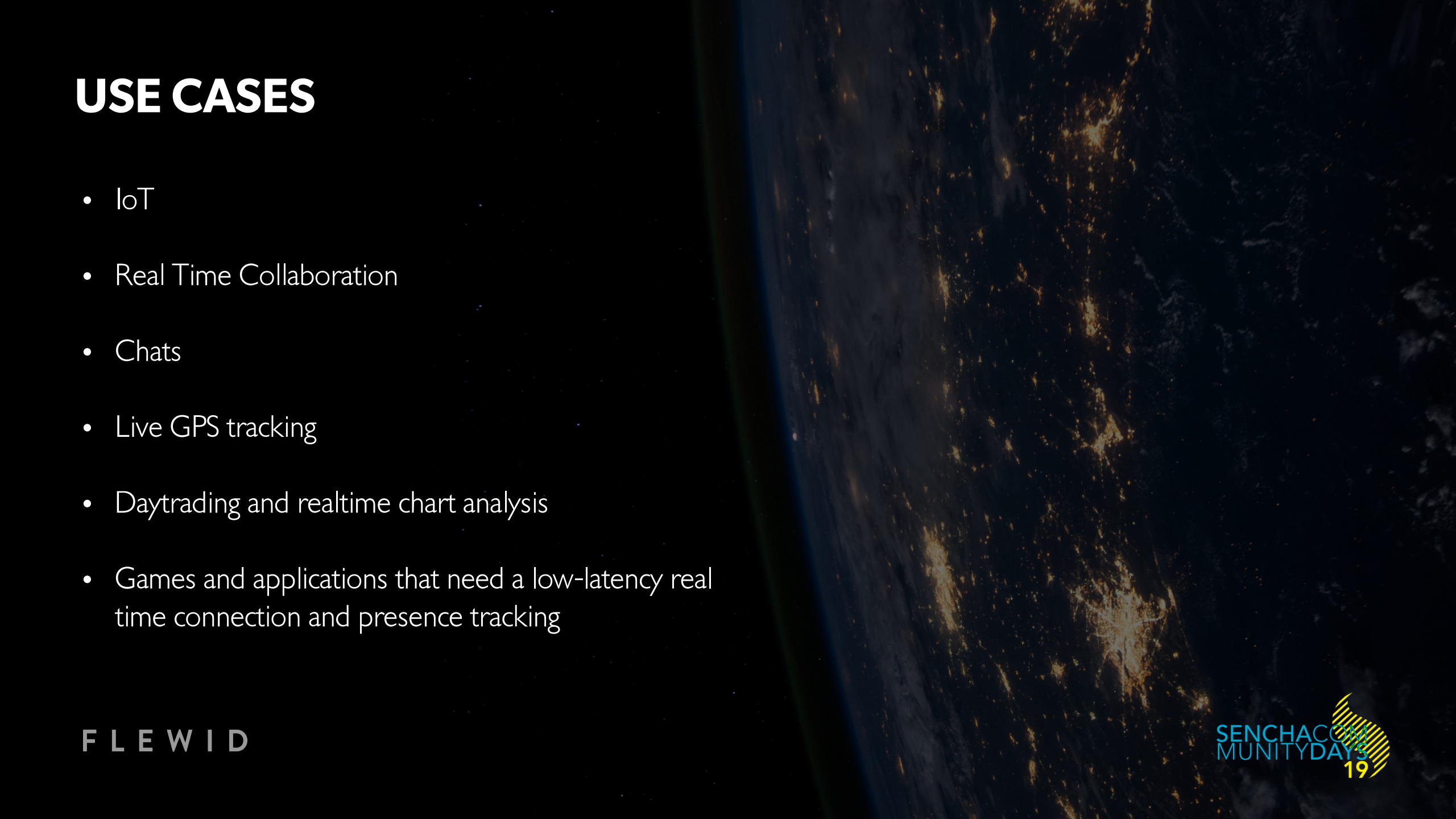
- Multiplexing: use multiple „channels“ through the same websocket connection
- No capping of simultaneous requests
- Avoid the additional payload of HTTP headers
- TCP ensures that all messages are sent in order

Securing your socket

- Use Secure connections! (wss://)
- Authentication can be on a connection level (connect requires JWT)
- Authorization can apply for certain channels (join requires certain role/capability)
- Additional authorization / authentication can occur on an event level (user needs role/capability)
- Presences can be restricted (user with same ID can not be present more than x times
- Most secure way is to handle authentication on the server and only accept connections from signed in users

USE CASES

- IoT
- Real Time Collaboration
- Chats
- Live GPS tracking
- Daytrading and realtime chart analysis
- Games and applications that need a low-latency real time connection and presence tracking



FLEWID

SENCHACOMMUNITYDAYS 19

Wanna try?

https://sockets.juliawill.com

SERVER SIDE – THE GOOD

- Depending on the hardware, websockets can manage up to 1M concurrent connections
- Low overhead
- Good support - a lot of frameworks have builtin support. For pretty much all others libraries exist

SERVER SIDE
– THE BAD & THE UGLY

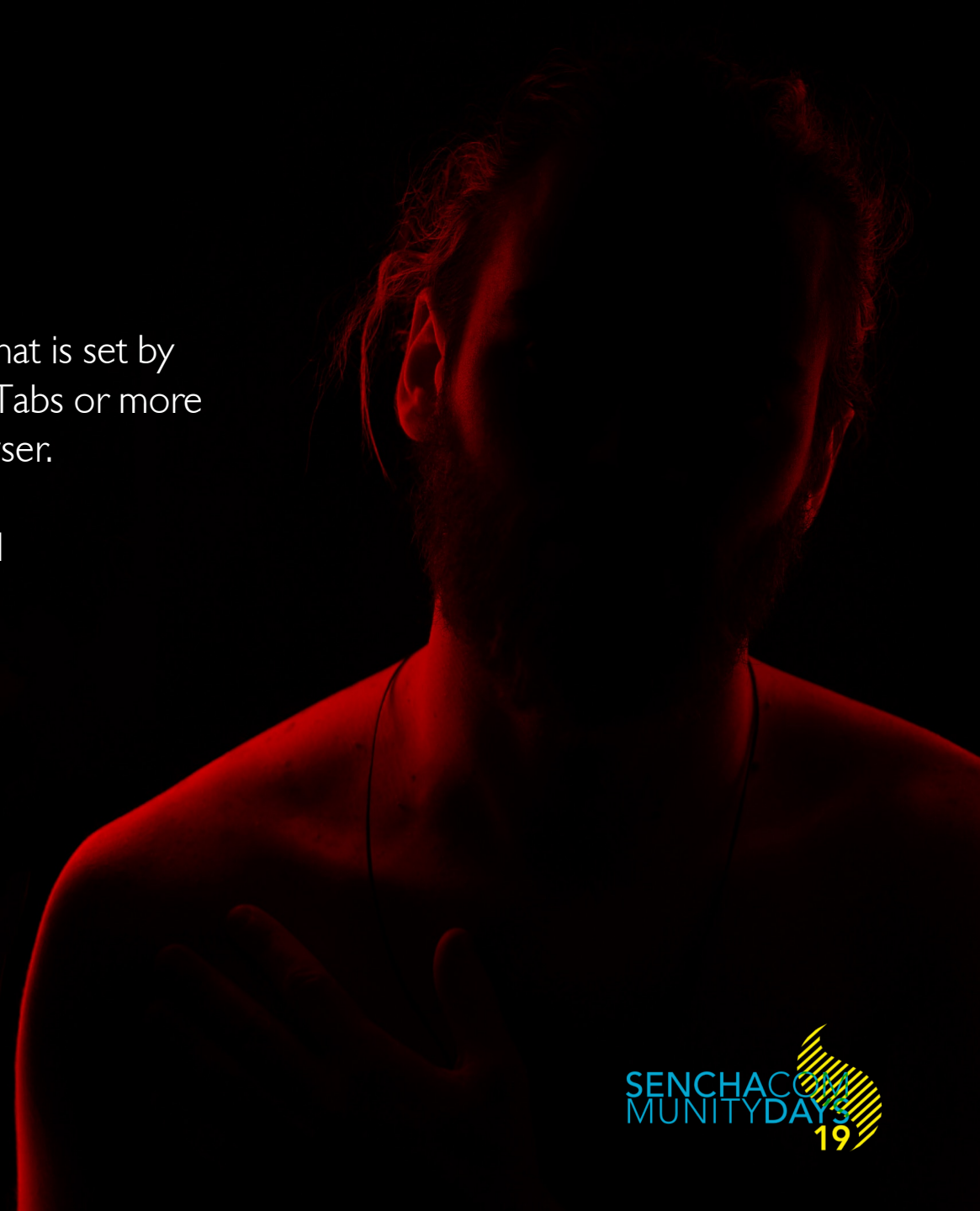
- Websocket connections are not HTTP connections, so the cap that is set by the HTTP-Server will not apply. Thus, it is possible to open 100 Tabs or more and establish the same amount of connections - in just one browser.

  To avoid clogging the server, restrictions should be implemented

- In load balanced environments, it might be problematic to
- keep sockets in sync

- No Logging by default

- Always use a secure connection (wss)



SERVICES – YOU DON'T HAVE TO CODE YOUR OWN

KAAZING

PUSHER

PubNub

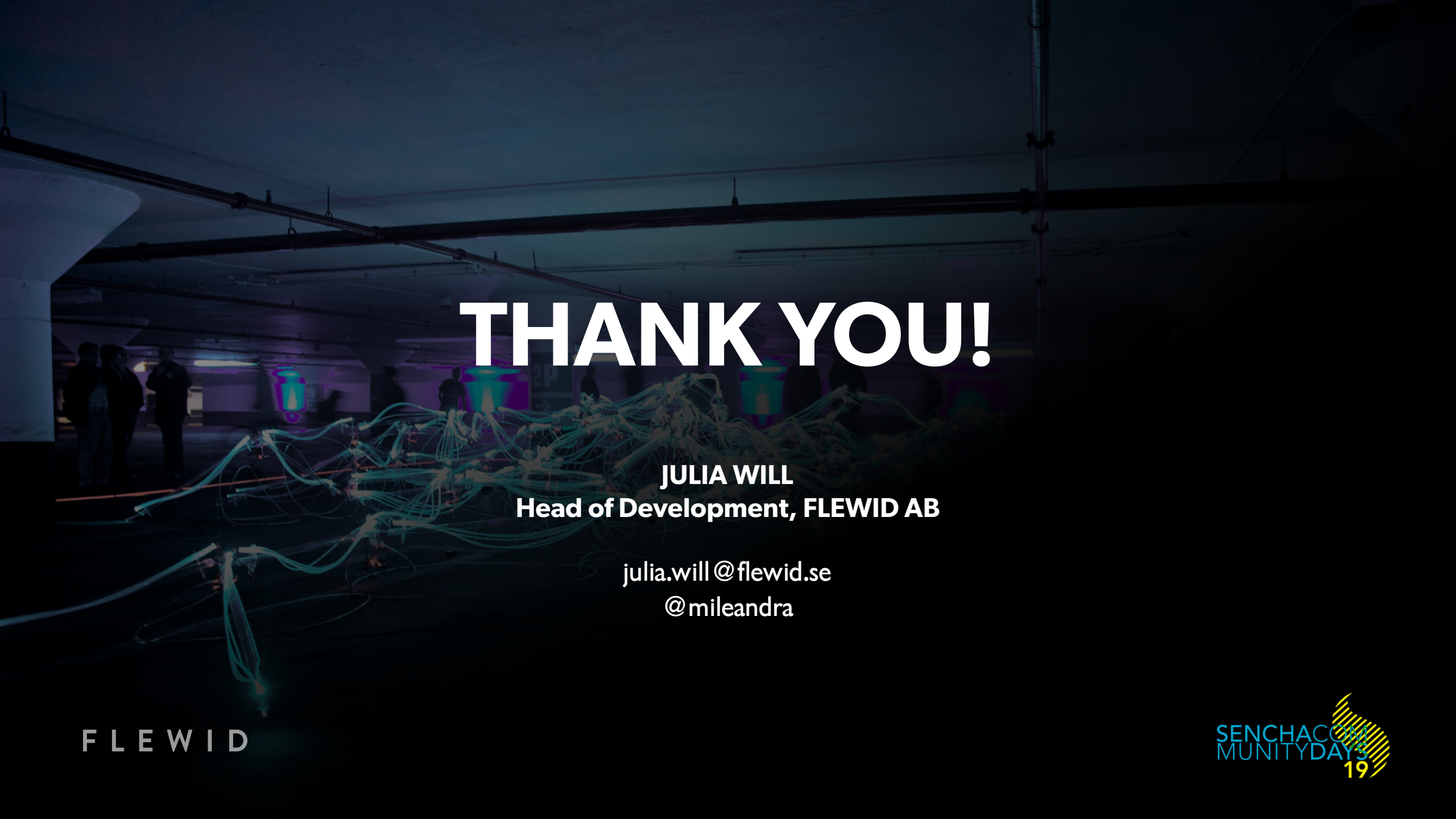
Firebase

SENCHACONMUNITYDAYS 19

CLIENT

- Native browser support in HTML 5
- Well maintained client libraries exist (socket.io)
- Easy and standardized implementation


QUESTIONS?



THANK YOU!

JULIA WILL
Head of Development, FLEWID AB

julia.will@flewid.se
@mileandra

FLEWID

SENCHACOMMUNITYDAYS 19

References

- Demo Application Repo: https://github.com/mileandra/sockets_app

- Libraries And Services:
  - Socket IO https://socket.io/
  - Pubnub https://www.pubnub.com/
  - Pusher https://pusher.com/
  - Kaazing https://kaazing.com/
  - Firebase https://firebase.google.com/

- Native Websocket API https://webplatform.github.io/docs/apis/websocket/