**Networking with "RUST"**

Why Rust ?

**Low footprint**

Take control over resource usage to keep memory and CPU footprint to a minimum. Get help from the compiler to make sure you've got it right. And do this with an ecosystem that is productive and pleasant to use.

**Secure and reliable**

Rust's powerful type checker prevents whole classes of bungd. Make sure you know exactly when and where state is shared and mutated. Get help catching points of filure - before deployment.

**Concurrent at scale**

Use any mixture of concurrency approaches that works for you. Rust will make sure you don't accidentally share state between threads or tasks. It empowers you to suqeeze every last bit of scalling, fearlessly.