



THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
EUROPE

Containerized Full-Stack Rust for IoT

Claus Matzinger (@claus__m)
<https://blog.x5ff.xyz>

October 24, 2017



Hi

I am Claus

Claus?

Rustacean for 1+ years

Rust Meetup Munich

From an Open Source startup (IoT)

Now Microsoft (Azure)

Must be
this tall to
write multi-
threaded
code.





Rust Is Great

Safe & Fast

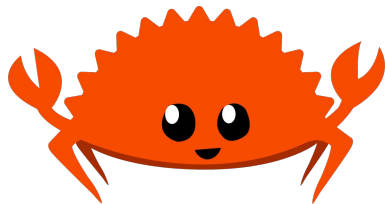
Efficient (no GC)

Great memory management

Fearless concurrency ;)

About IoT

And DevOps, Containers, Rust, and the World



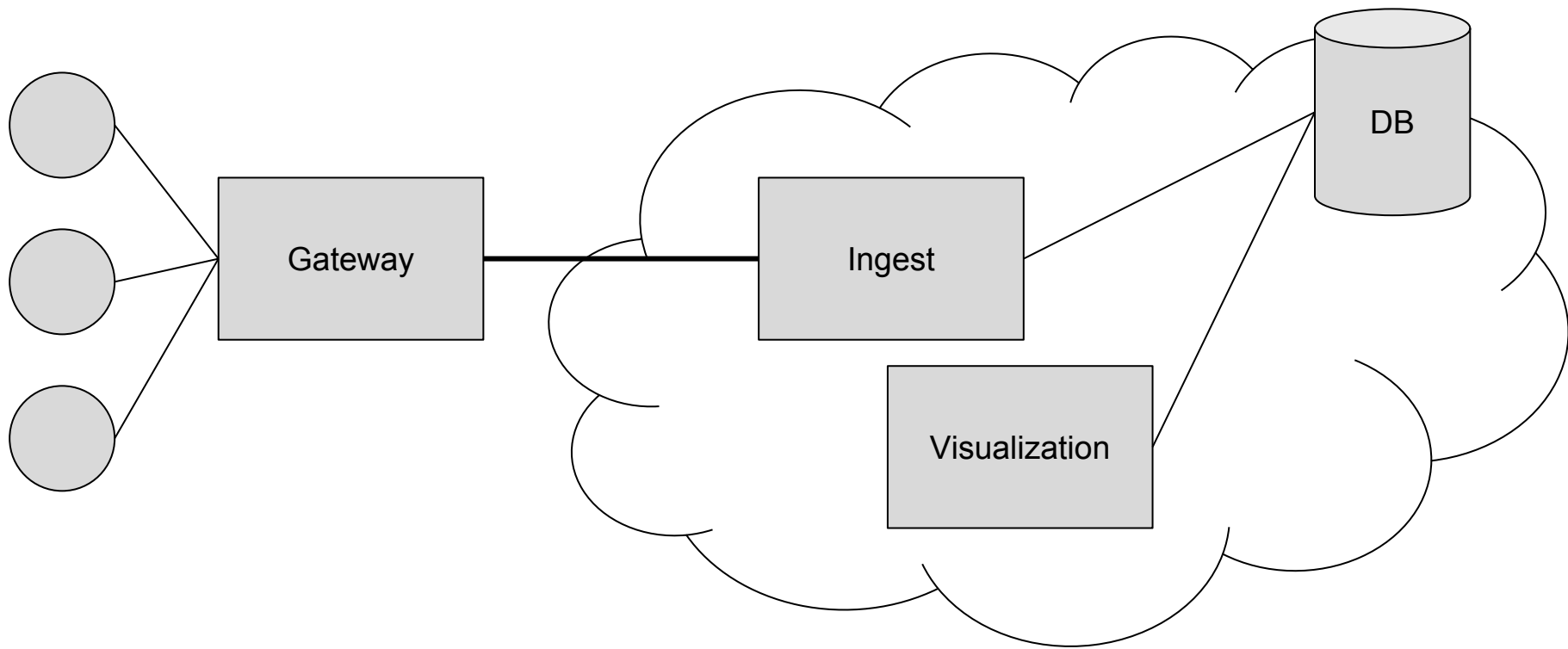
IoT Tech Is Solved

Off-the-shelf solutions available
Integration is key!

Choices

Lots of them!

- AWS IoT
- MS IoT Hub/Edge
 - Watson IoT



Roll Your Own

Using Open Source technology

- CrateDB/Druid/Cockroach/Elastic
- Grafana/OpenMCT/Superset
- Mosquitto/RabbitMQ/Kafka

How and Why

Should Rust be a part?

A full-page background image of a sunset over the ocean. The sun is a bright yellow-orange orb on the right side, partially obscured by clouds. Its light reflects on the water, creating a shimmering path. The sky is a mix of orange, yellow, and blue, with scattered clouds. The water in the foreground is dark with gentle ripples.

Fail fast - fail often

rustc

Safety & Efficiency

Great for smaller devices

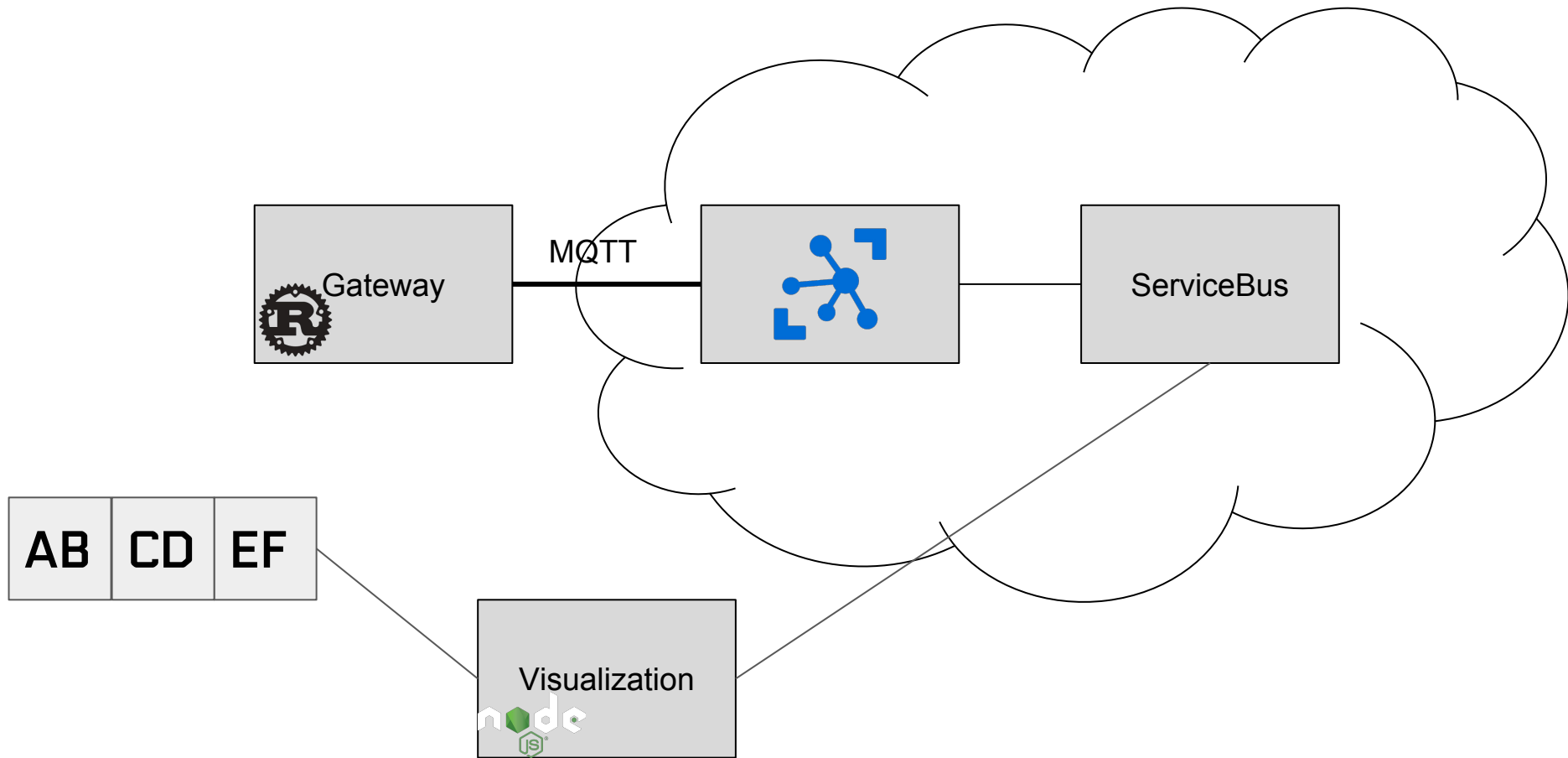
Performance of a compiled language

Highly productive

The compiler knows

A black and white photograph of a scuba diver underwater. The diver is on the left, facing right, with a large rock formation on the right. The text "Dive In" is overlaid in the center. The background is dark with many small fish swimming around.

Dive In



Step 1

Rust $\leq$ MQTT $\Rightarrow$ IoT Hub

easy.

Orchestrate

- Kubernetes
- Docker Swarm
- (Service Fabric)
 - ~~Manual~~

Containerize

- Single platform
- Lots of tools available

Step 2

[Rust] $\Leftarrow$ MQTT $\Rightarrow$ IoT Hub

Issues

Dynamic linking
Dependencies (libopenssl >:()

Fixing it...

```
docker run -it my/base:image bash
```

Streamline dependencies (Cargo.lock)

Cross Containerization

- ARM binary
- Alpine Linux, musl libc?
 - Docker tags
- Device or VM required

Step 3

{ Rust } $\leq$ MQTT $\Rightarrow$ IoT Hub

Cross Compilation

1. Don't set up a cross toolchain
2. Check dependencies for compatibility
3. Use `gh:japaric/cross` or `gh:japaric/rust-cross`

Or a docker container:

`dockerhub:dlecan/rust-crosscompiler-arm/`

Cross Containerization

- Build the binary
- Get a base image for target arch
 - Build container:
ARM v6 vs v7, qemu?

Compile Times

Great for 1-10 coffees

Step 4

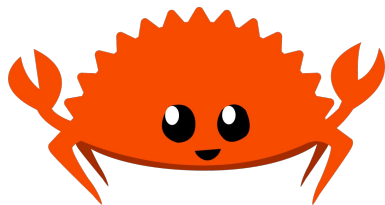
{ Rust } $\leq$ MQTT $\Rightarrow$ IoT Hub

A CI Pipeline

- Multi-architecture
 - Publish binaries
- Test container startup ;)

Travis

<https://github.com/japaric/rust-cross/blob/master/.travis.yml>



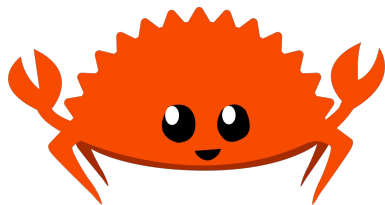
Tips

Use more Rust :)

Stick with Rust stable

Use pure Rust libs

Specify arch-specific dependencies



More Tips

Check & share community efforts

Check & fix dependencies if easy

Check out Rust meetups

...



Thanks!

<https://github.com/celaus/>

<https://blog.x5ff.xyz>

@claus__m