Go, Swarm and DevOps

vs

The Mighty Monolith

Igor Karpovich

Disclaimer

Not a true story at all
All persons and events are fictitious
No gophers were harmed

2013:

- Secure in-house infrastructure
- Cloud for customer facing products
- Monitoring by looking into server rooms
- F&G log analysis system (F for *find* and G for *grep*)
- Jenkins for tests, *manual* deployments
- 4 products

2018

- Multi cluster AWS environment
- Container orchestration for *all* products
- Logging, monitoring, tracing
- CI/CD for products and infrastructure
- 50+ products

It's been a long road...

- Culture
- Architecture
- Platform
- Legacy

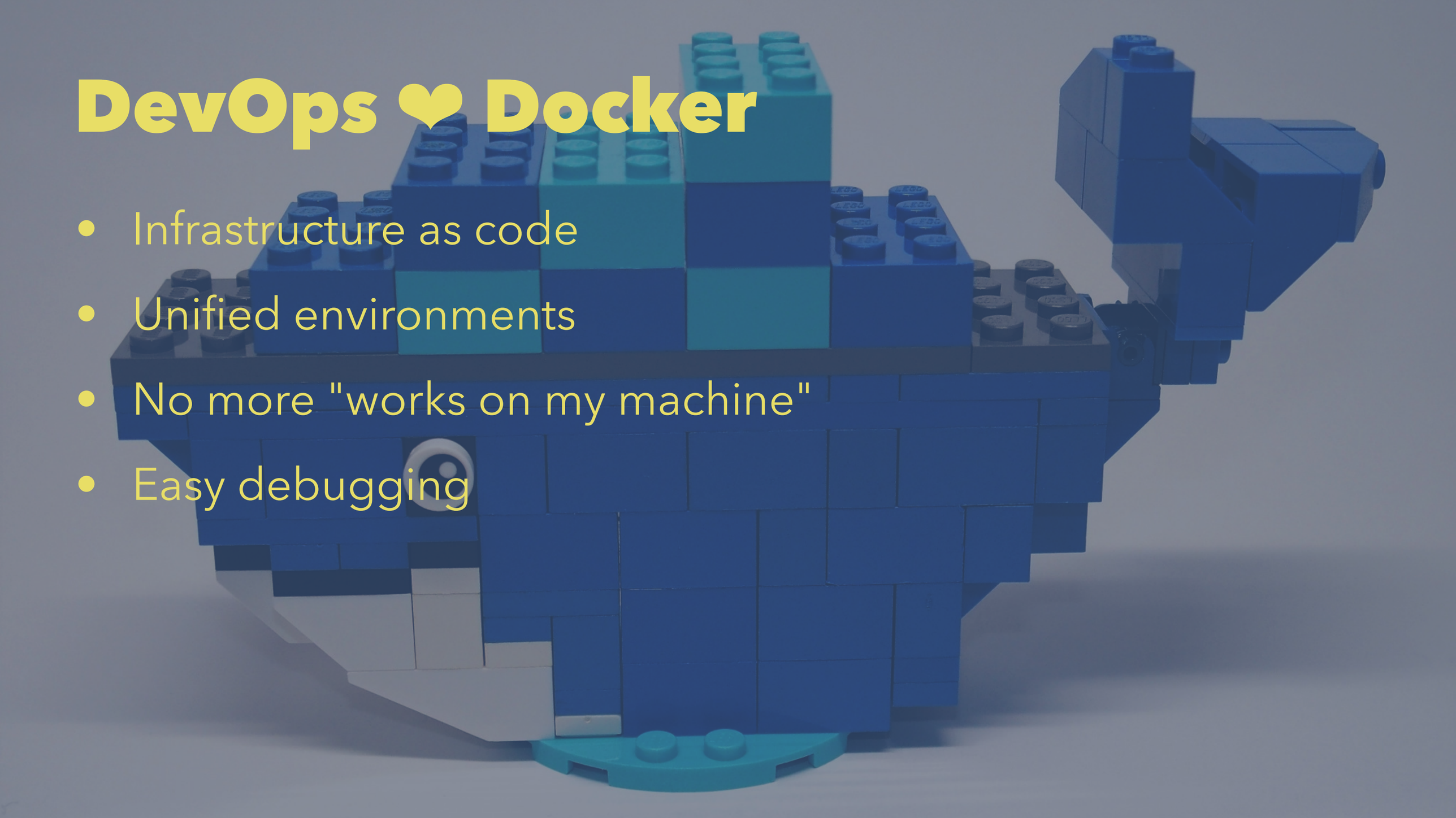
Moving away from CDD*

- No more favourite IT people to go to
- Ticketing system
- Change management and deployment awareness
- Feedback loop
- Incident management
- Iterative approach with steady value delivery

---

* Chaos Driven Development

DevOps ❤ Docker



- Infrastructure as code
- Unified environments
- No more "works on my machine"
- Easy debugging

DevOps is *not*:

- A job title
- Ops telling Dev to DIY
- Giving out unlimited access to production

Version Zero
First microservice

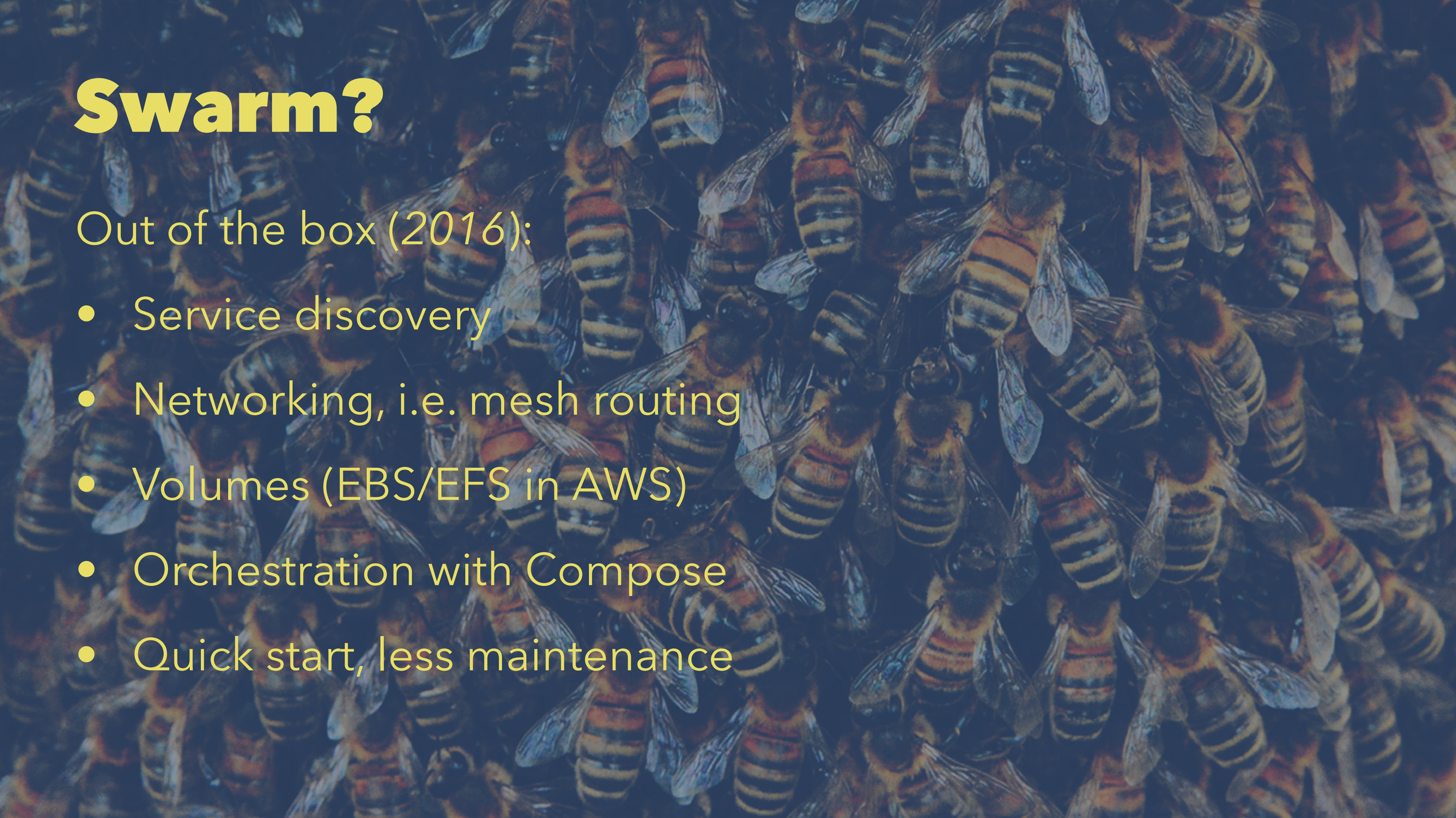
- REST API only
- Monitoring coverage
- Plugged into core app
- Traditional deployment
- Written in PHP

On the way to the bright future

- Maintaining stability
- Security by design
- Increasing velocity
- Supporting innovation

Honouring legacy

- Minimising incidents
- Shutdown planning for software and infrastructure
- Replacing monolith pieces with microservices calls



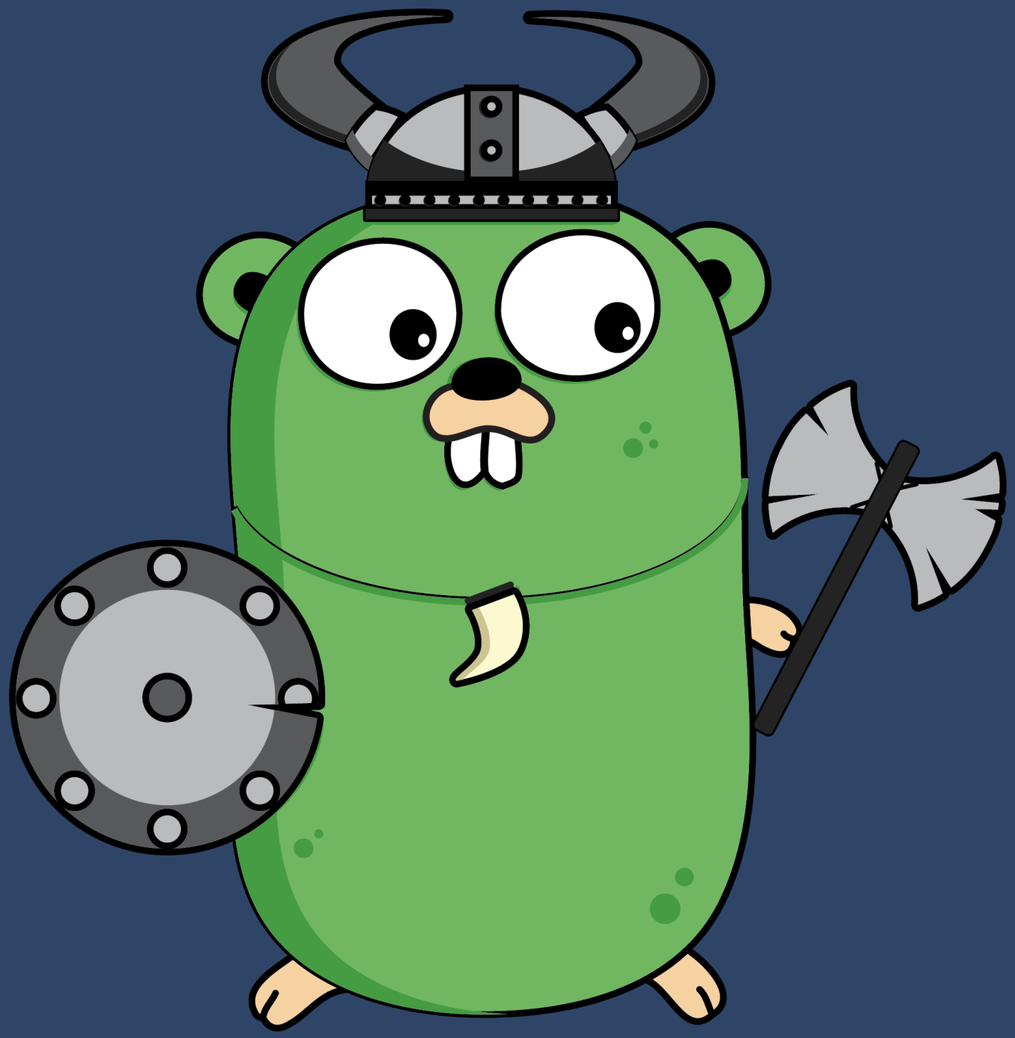
Swarm?

Out of the box (*2016*):

- Service discovery
- Networking, i.e. mesh routing
- Volumes (EBS/EFS in AWS)
- Orchestration with Compose
- Quick start, less maintenance

Goblin

Microservice framework

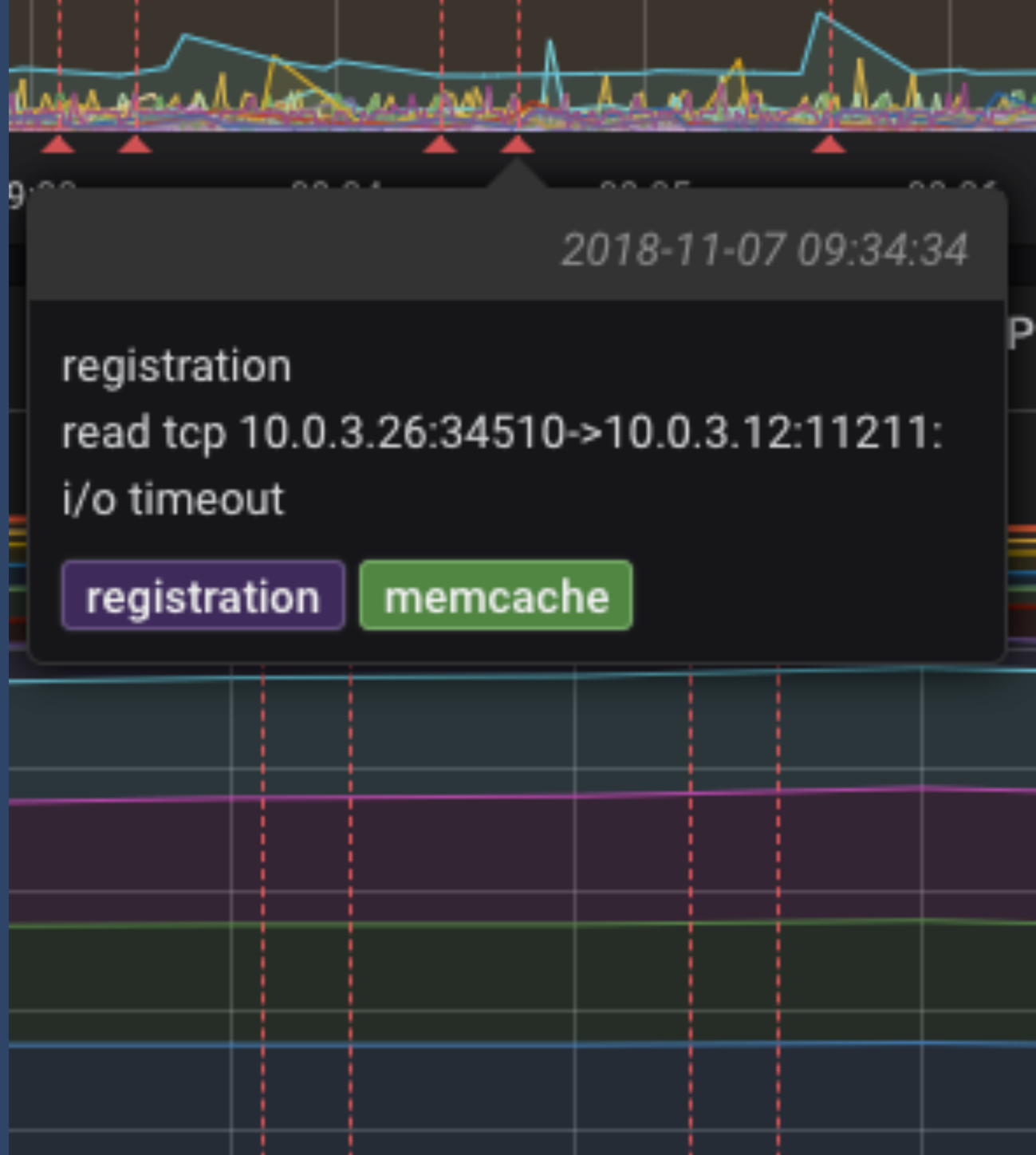


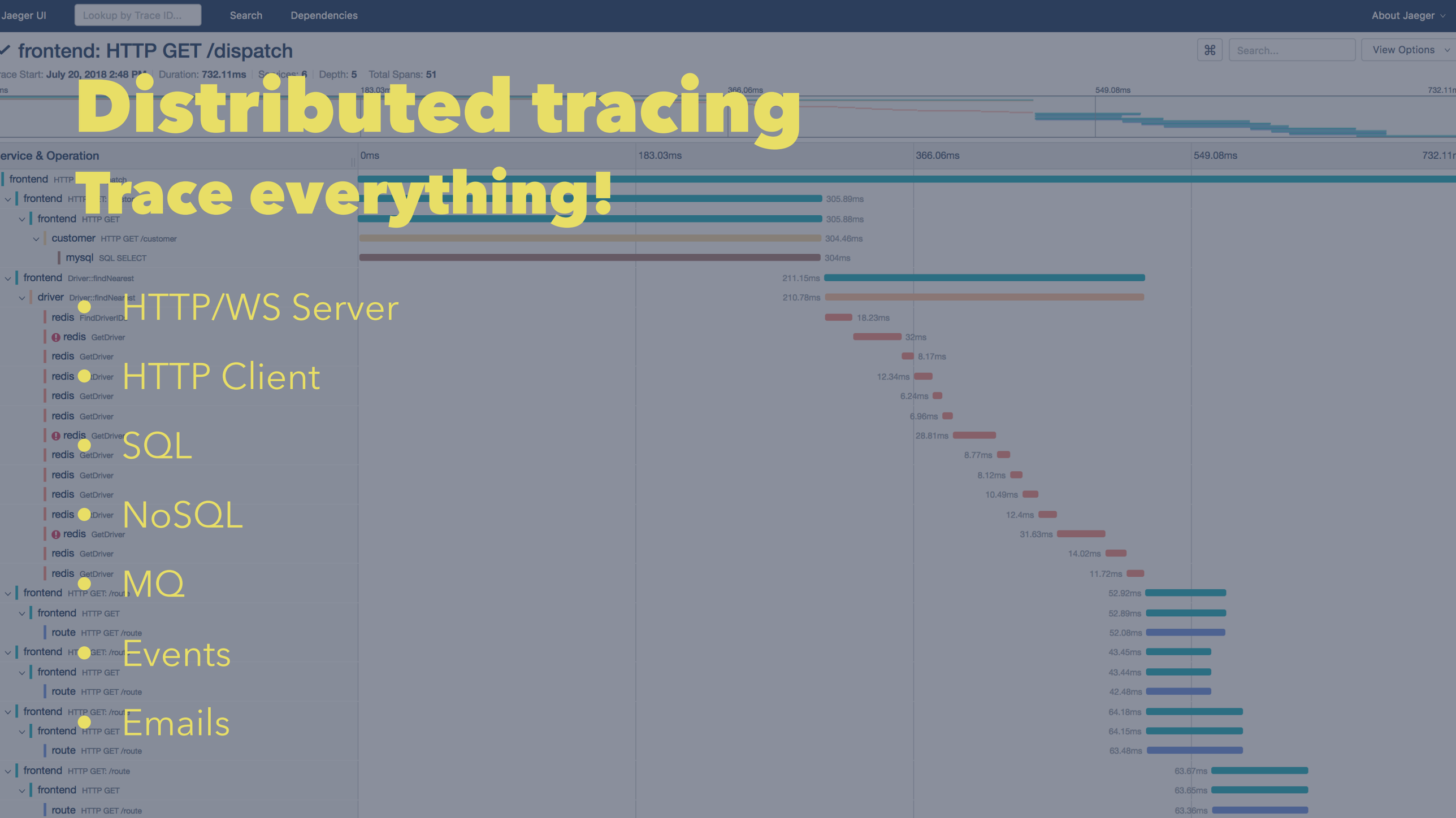
Some of Goblin's features

- Runtime (module) management
- Config management
- Storage & Cache
- Transport (HTTP Server/Client, WS router)
- Monitoring client and metrics (Gin, WS router, go runtime)
- Logging (context data injection, sentry integration)
- Healthchecks
- Tracing

Healthchecks as metrics

- Docker native support
- Expose metrics
- Push to InfluxDB
- Visualisation




Distributed tracing
Trace everything!
HTTP/WS Server
HTTP Client
SQL
NoSQL
MQ
Events
Emails

Platform toolkit

- Clusters self-deployment (traefik, CI, monitoring, logging, tracing)
- GitOps
- Secret management
- Service wide event broadcast
- SSO, identity management (ACL, JWT, SAML)
- Scheduler
- Job runner
- Change Audit
- Multi-route email delivery
- Monitoring (InfluxDB, Grafana, Slack, TV dashboards)

Palantir

Monitoring backend and TV Dashboard featuring visual Swarm

- Pulls metrics from InfluxDB
- Metric format is Telegraf compatible
- YAML definition for Docker Stacks
- Metric templates with YAML anchors

Questions!

Twitter: @ikarpovich