Rise of the Machines

How we built and scaled the real Skynet, then open-sourced it so you can too.

Who are we? Who are you…

Peter DeMartini

Software Engineer @ Octoblu / Citrix

@PeterDeMartini

Aaron Herres

Software Engineer @ Octoblu / Citrix

@redaphid

Why you should trust us

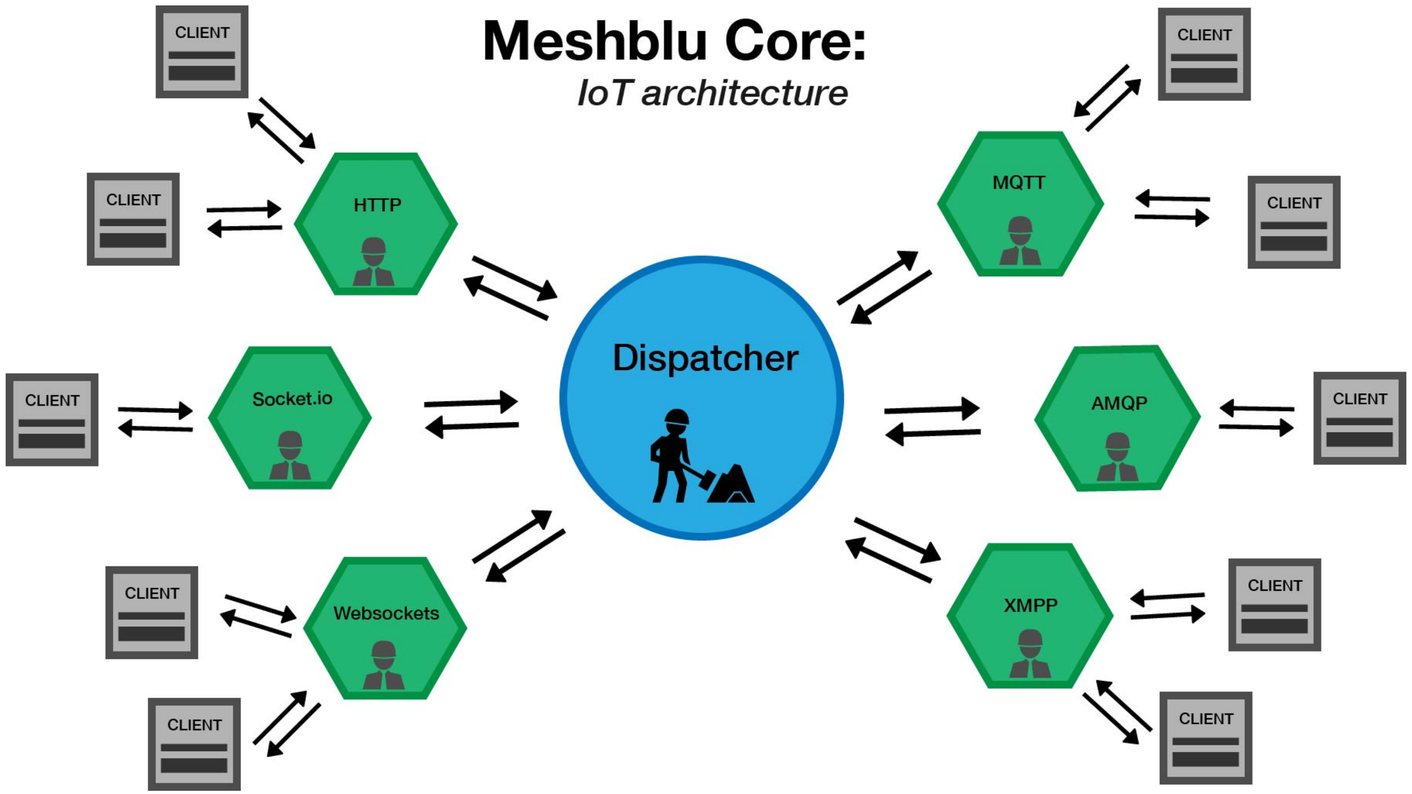
- We’ve built and deployed hundreds of microservices
- We’ve failed many times…and sometimes learned from our mistakes
- Experimentation is part of our development process

Who this is for

- Software Developers
- DevOps
- Members of the resistance

So, What's Skynet?

- It's Meshblu now. But it may still be evil.
- Open-Source, free to use
- Over 1000 MIT-licensed Github projects
- **5 billion** messages a month
- **1.5 million** connected devices
- Mostly Node & Docker



A Pretty Picture of Meshblu

Look, they're all microservices!

Microservices

- A philosophy, not an implementation
- They should be small, single-purpose, and dumb
- Docker works **extremely** well for this, and plays well with Node.
- They should be glued together with web standards (Meshblu helps with this!)
- This pushes the complexity to the interaction between microservices, which is “*easy*” to debug

Microservice Glue

- Microservices move the complexity to the interaction between the services, but how do you manage that complexity?
  - Keep them dumb
  - Make building / deploying micro services easy
  - Don’t build it if it already exists
- **Case in point**: Meshblu https://github.com/octoblu/meshblu

Managing Microservices

- Docker is fantastic for microservices, but not perfect for scaling
- **CI** (Continuous Integration)
- **CD** (Continuous Deployment)
- Auto-updating
- Self-healing
- Uptime monitoring

The Competition

- There’s a lot of solutions for managing clusters of docker instances, and most of them have problems
  - Kubernetes
  - CoreOS
  - Swarm
  - ECS / Google Containers / Docker Cloud

Micro…clusters?

- Microservices aren’t the only things that should be redundant
- Every solution we tried has some kind of trouble under serious load
- Redundant, small, focused clusters helped our uptime tremendously
- Scalable
  - Multi-node clusters, cross-cluster services
- Rebuild, Upgrade, and Experiment

P-Tier Architecture™

Managing Microclusters

- Declarative Infrastructure
  - Cluster Configuration `**cluster.json**`
  - Environment Files `**env.d/*.env**`
  - See https://github.com/peterdemartini/octoswarm-stacks
- Load Balancer configuration
- DNS configuration
- Uptime Alerts

Continuous Deployment

- Beekeeper
  - Track the latest version of a service
  - Gate CI tests and Docker builds
  - Auto-updating Docker Swarm

https:/github.com/octoblu/beekeeper-service

https:/github.com/octoblu/beekeeper-worker

https://github.com/octoblu/beekeeper-util

https://github.com/octoblu/beekeeper-updater-swarm

https://github.com/octoblu/beekeeper-updater-docker-compose

Demo Time

https://github.com/peterdemartini/octoswarm-stacks

“I'll be back.”

– The Terminator