AUTODISCOVERY. FANTASTIC BEASTS AND WHERE TO FIND THEM

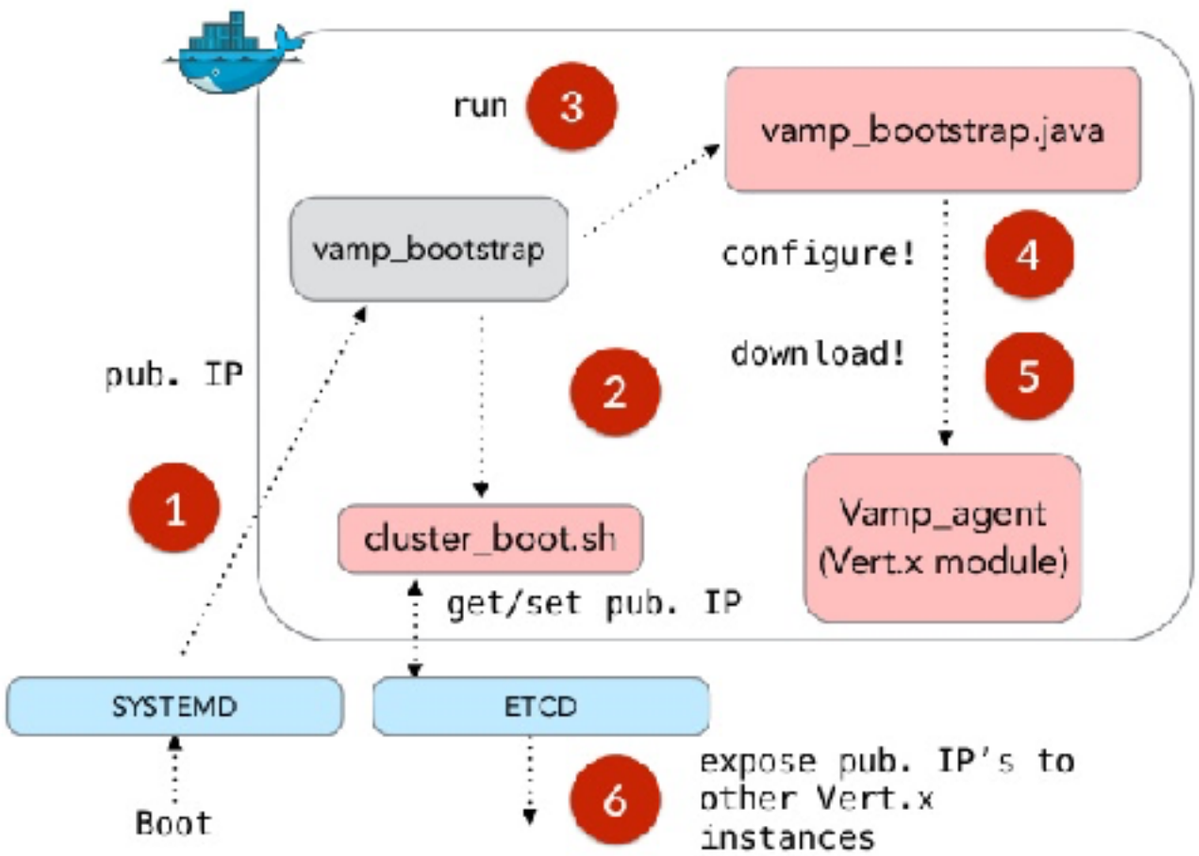
DNS Failure Is Not an Option.
Different approaches to glue together Golang applications in the Kubernetes, Docker and Westworld

SOLUTION LEADER

- ZX Spectrum Generation

- My first program in 11 years.
- Assembler, C/C++, Perl, Ruby and other crap
- Kubernetes, containers, etc.

- Mountain bike, snowboard, hiking



AGENDA

- Problem Statement
- Centralised Solutions
- Distributed Solutions
- Something more complicated
- DNS Failure Is Not an Option
- Surprise!

PROBLEM STATEMENT

Only the Dead Have Seen the End of War
- Plato

SIMPLE APPLICATIONS – PAYMENT GATEWAY

- Modules
  - Web frontend
  - iOS/Android app
  - RESTful backend
    - Processing module
    - Tax Calculator
- External RESTful API integration
- Local SQL database

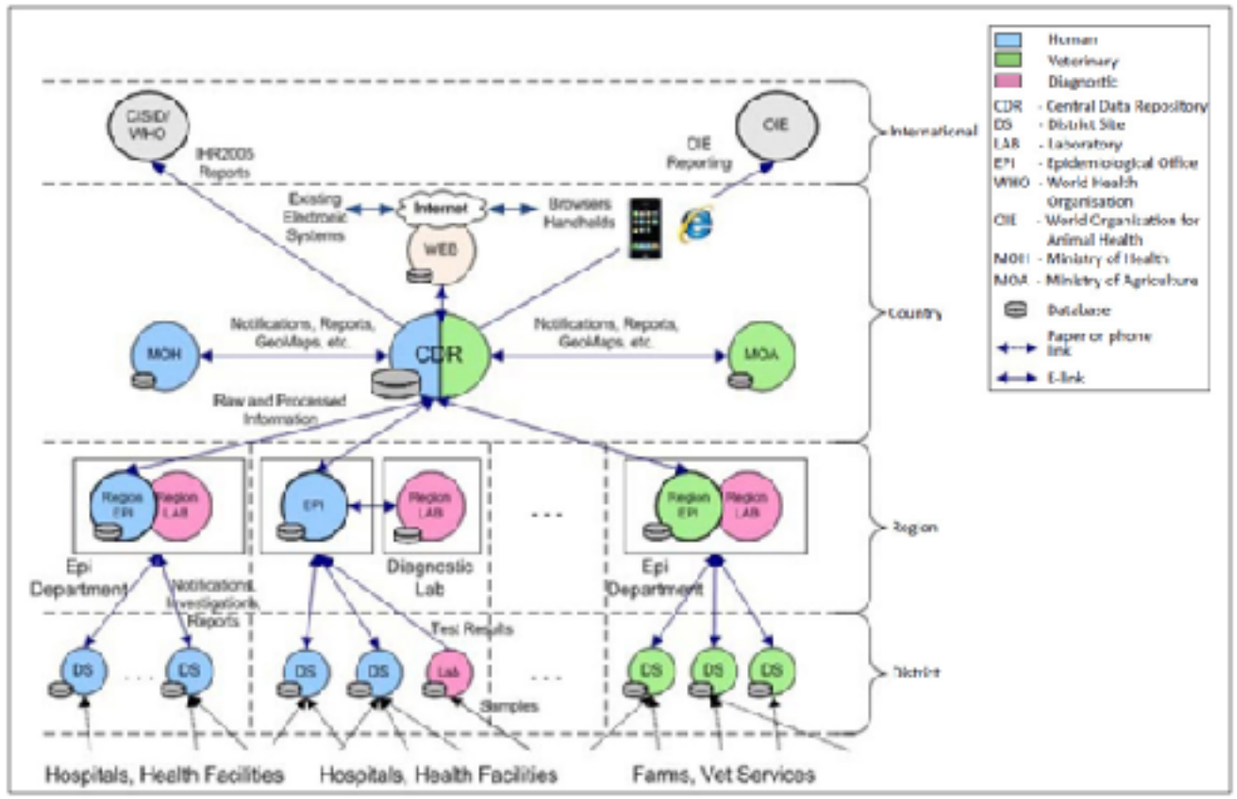


FIGURE 1: Typical Electronic Integrated Disease Surveillance System architecture.

CENTRALISED SOLUTIONS

*This is it? One f***ing platoon?*
- John McClane

FRAK THE ANSIBLE

- DNS with bunch of the Load Balancer
- CMDB (Configuration Management Database) + some scripts, playbooks, etc.
- AWS LB + AWS Route 53
- etc.

PROS AND CONS

1. **PROS**
    1. Simple enough for understanding for mediocre developers
    2. Usually ease to implement
    3. Minimum changes on Applications level
2. **CONS**
    1. One point of failure
    2. Green/Blue update approach is not easy to implement
    3. One woodpecker can destroy whole civilization

SOLUTION EXAMPLES

1. Configuration Management and CMDB
    1. OpsCode Chef
    2. Puppet
    3. Ansible
    4. Salt
    5. CFEngine
    6. Something custom

DISTRIBUTED SOLUTIONS

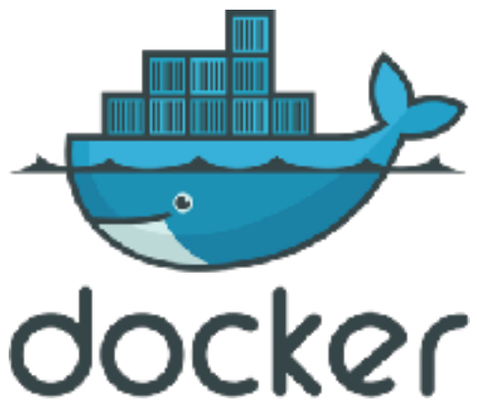
They're 6-year-old kids. How much trouble can they be?
- Detective John Kimble

PROS AND CONS

1. **PROS**
    1. Distributed by design
    2. Simple enough to implement and support
    3. Minimum changes on Applications level
2. **CONS**
    1. Sometimes one point of failure. SkyDNS for example
    2. Some technology is not ready for PROD
    3. Sometimes dependencies on the containers solutions

SOLUTION EXAMPLES

- **Distributed CMDB**
  - HashiCorp Consul
  - CoreOS etcd with OR without SkyDNS
- **Schedulers**
  - Google Kubernetes
  - Docker Swarm
  - Apache Mesos



SOMETHING MORE COMPLICATED

Just give me fraked CPU!
- unknown customer

BUSINESS REQUIREMENTS

- Per customer transaction processes isolation
- Easy to scale. From 100k to 3m transactions per minute
- Ability to shutdown whole system OR one customer workflow in 5 sec. In case of security breach
- 34 geo-distributed untrusted DC (Data Centers)

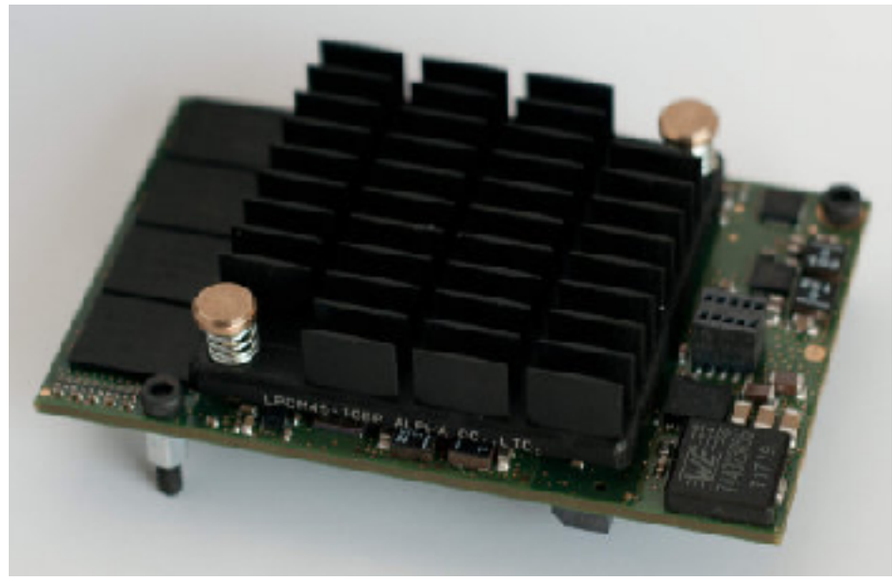
POSSIBLE SOLUTIONS: VM'S VS CONTAINERS

- **VM's**
    - Good enough isolation
    - Very expensive
    - Over 60sec to spinoff new VM

- **Containers**
    - No PCI complain isolation
    - Easy to manage and create
    - Only 2 sec to spinoff new Container

PLATFORM

- **Customised ODROID C2 ARMv8**
  - 4 Core CPU
  - 2Gb RAM



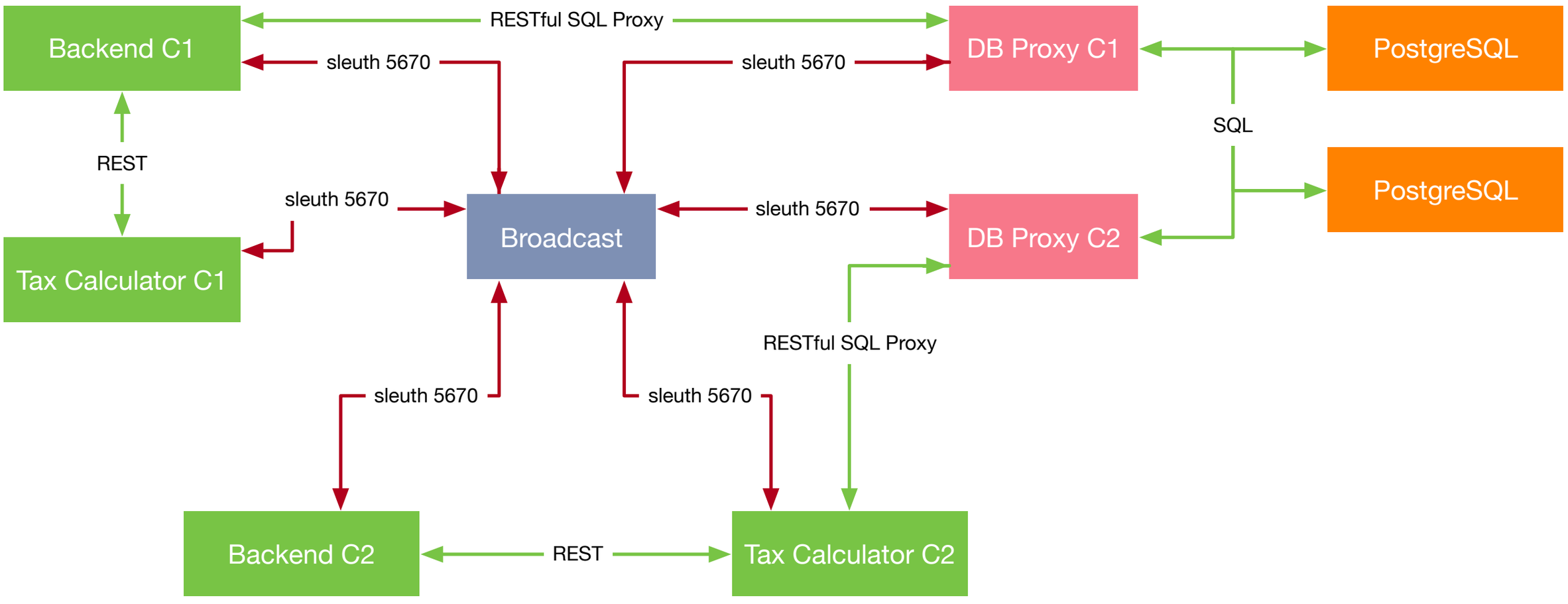
- **18** SBC (Simple Board Computer) per one rack unit
- **684** physical server per rack cabinet

- Boot from iPXE
- 45drives storage solutions

DNS FAILURE IS NOT AN OPTION

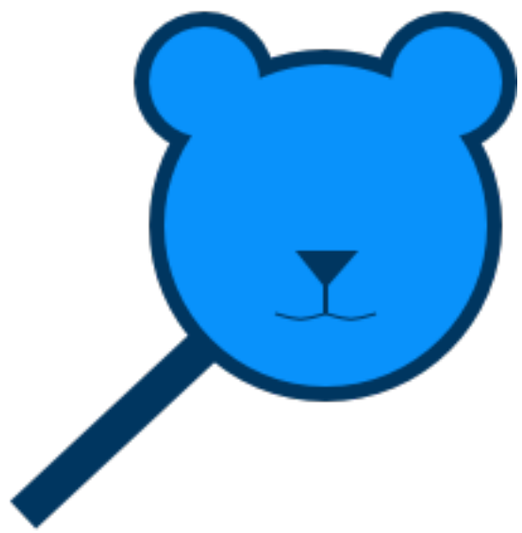
There no a "sherif" in Westworld
- MR

SCHEMA


Backend C1
RESTful SQL Proxy
DB Proxy C1
PostgreSQL
sleuth 5670
sleuth 5670
SQL
REST
sleuth 5670
Broadcast
sleuth 5670
DB Proxy C2
PostgreSQL
Tax Calculator C1
RESTful SQL Proxy
sleuth 5670
sleuth 5670
Backend C2
REST
Tax Calculator C2

SLEUTH – HOW THIS WORKING

- Nodes in the network discover each other using a UDP beacon on port **5670**
- Sleuth client provides ability to **wait** for required services before sending any requests
- Sleuth automatically **round-robin** the requests each client makes to all services that share the same name
- Sleuth using simple **REST API**
- **ØMQ** (ZeroMQ) library



FACTS

- BETA stage
- 4 USA customers
- Highly customised ODROID C2
- Already over 1200 physical nodes
- More than 30 different applications
- Master-slayer PostgreSQL behind RESTful API proxy

USEFUL INFORMATION

- Sleuth lib URL http://ursiform.github.io/sleuth/
- Implementation example http://darian.af/post/master-less-peer-to-peer-micro-service-autodiscovery-in-golang-with-sleuth/
- Raft: The Understandable Distributed Consensus Protocol https://speakerdeck.com/benbjohnson/raft-the-understandable-distributed-consensus-protocol/
- The Raft Consensus Algorithm https://raft.github.io
- Serve a RESTful API from any PostgreSQL database https://github.com/nuveo/prest

QUESTIONS?


esc
tab
caps lock
shift
ctrl
alt
F1
F2
F3
F4
F5
F6
F7
F8
Q
W
E
R
T
Y
U
I
O
P
A
S
D
F
G
H
J
K
L
Z
X
C
V
B
N
M

CONTEST

ENLI

small and promising startup company

WE ARE HIRING

HTTP://ENLI.IO

- Projects
  - VoIP
  - Distributed systems and mesh networks. ARM platform
  - IoT
  - Kubernetes
- **Engineers**
  - Go Developers
  - DevOps Engineers
  - iOS/macOS Developers

INTERESTING?

myroslav@enli.io