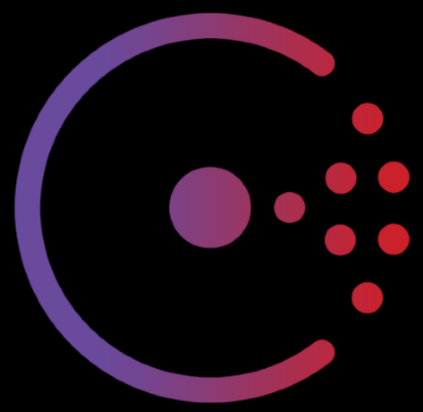
Introduction to Spring Cloud Consul



Presenter: David Lucas

L S E

Who am I ?

- Over 25 years in software industry
- Linux Enthusiast since 1994
- Working with Java since 1998
- Java & Spring Enthusiast since 2004
- Kotlin Enthusiast since 2016

David Lucas
Lucas Software Engineering, Inc.
www.lse.com
ddlucas@lse.com
@DavidDLucas

L S E

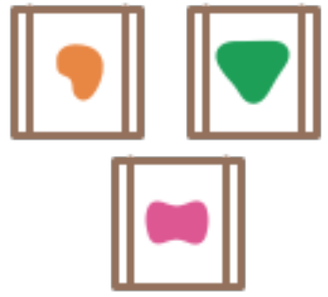
Spring Cloud Consul: Introduction

- Why are properties & discovery important ?
- What is Consul ?
- What is Spring Cloud Consul ?
- Summary

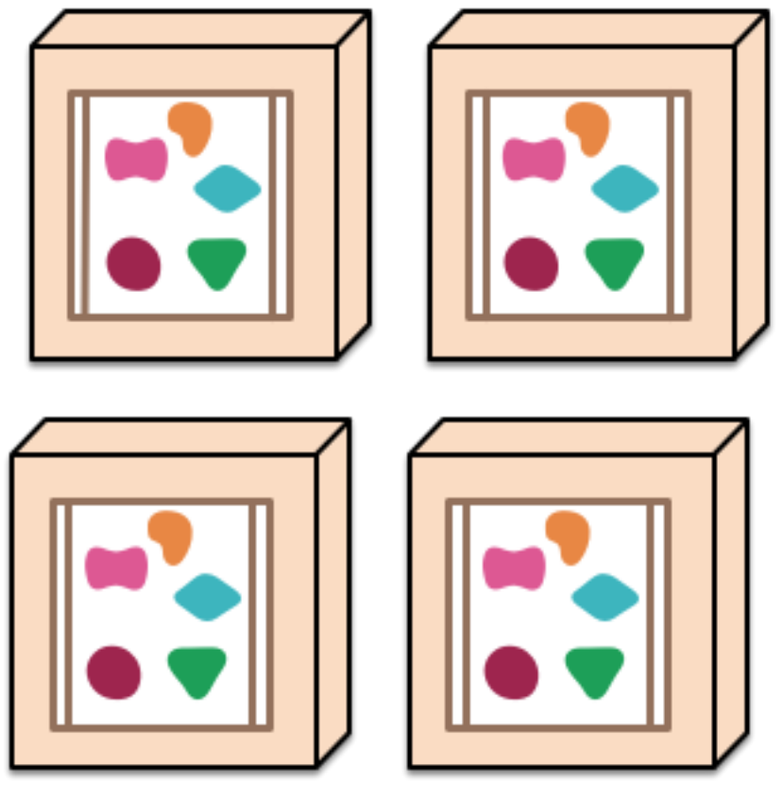
Microservices

A monolithic application puts all its functionality into a single process...

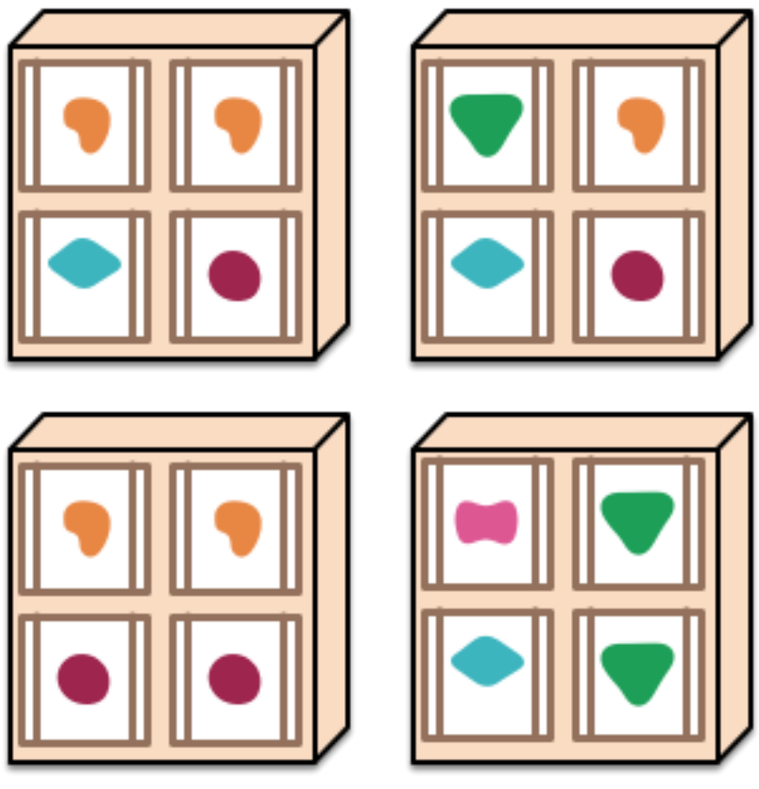
A microservices architecture puts each element of functionality into a separate service...



... and scales by replicating the monolith on multiple servers



... and scales by distributing these services across servers, replicating as needed.



https://martinfowler.com/articles/microservices.html

BTW: Unix got it right !

cat data.csv | **grep** “LOGIN“ | **tr** -d’,’ -f1-3 > login.csv

<stream> | <process> | <sink>

- simple
- each module does one thing well
- leverages other capabilities

Microservices

12 Factors (https://12factor.net)

1. *One codebase tracked in revision control, many deploys*
2. *Explicitly declare and isolate dependencies*
3. *Store config in the environment*
4. *Treat backing services as attached resources*
5. *Strictly separate build and run stages*
6. *Execute the app as one or more stateless processes*
7. *Export services via port binding*
8. *Scale out via the process model*
9. *Maximize robustness with fast startup and graceful shutdown*
10. *Keep development, staging, and production as similar as possible*
11. *Treat logs as event streams*
12. *Run admin/management tasks as one-off processes*

Microservices

12 Factors (https://12factor.net)

1. *One codebase tracked in revision control, many deploys*
2. *Explicitly declare and isolate dependencies*
3. **Store config in the environment**
4. Treat backing services as attached resources
5. Strictly separate build and run stages
6. **Execute the app as one or more stateless processes**
7. **Export services via port binding**
8. *Scale out via the process model*
9. *Maximize robustness with fast startup and graceful shutdown*
10. **Keep development, staging, and production as similar as possible**
11. *Treat logs as event streams*
12. *Run admin/management tasks as one-off processes*

Consul: Introduction

- HashiCorp's Swiss Army Knife
- Light Weight Go Binary (client/server)
- Key Value Pairs
- Registry
- DNS

Consul: Introduction

- Event Bus
- Semaphore
- Health Checks
- Data Center capable

Consul: Introduction

- Highly Available Clustered Information
- 3-5 (odd) consul agent -server
  (can run just one in development mode)
- consul agent cached proxy for servers
- communicates over Serf protocol
- Gossip over Serf, synchronizes data

Consul: Introduction

- Leader Forwarding to migrate changes to primary server
- LAN and WAN ports for local and remote datacenter conversations
- TLS encryption support
- Security ACL token support
- Eventually Consistent Model

Consul: Introduction

- Top Level Service Discovery

  /v1/agent/service/register

  - API Discovery (port 8500)
  - DNS Discovery (port 8600)

    dig @localhost -p 8600 dev-consul-01.consul

- Spring Consul Discovery
  @EnableDiscoveryClient
  @DiscoveryClient

Consul: Introduction

- Health Checks
  /v1/health/checks/<service>

- Spring Actuator
  /health (define health indicator)
  /refresh (post to refresh scope)

Consul: Introduction

- Key Value Store (time to live)
  /v1/kv/<key>

- Spring Consul Configuration
  Key hierarchy to PropertySource Context
  Key can point to value or properties

- git2consul (useful to migrate)

Consul: Introduction

- Event Bus (Surf P2P Event Library)
  /v1/event/fire/<event-name>

- Spring Cloud Bus

Shared communications between services
Specific Server changes
Not for business application use

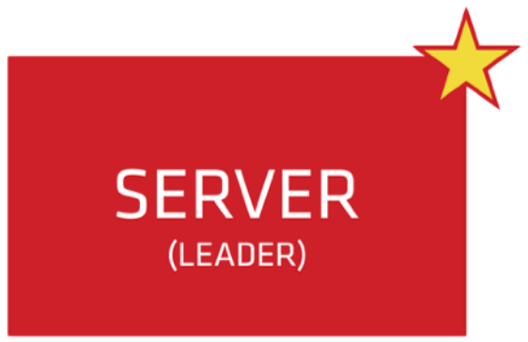
Consul: Introduction

- Security ACL
  /v1/acl/create

- Allows hiding of keys by token access
- Each service could have its own api token

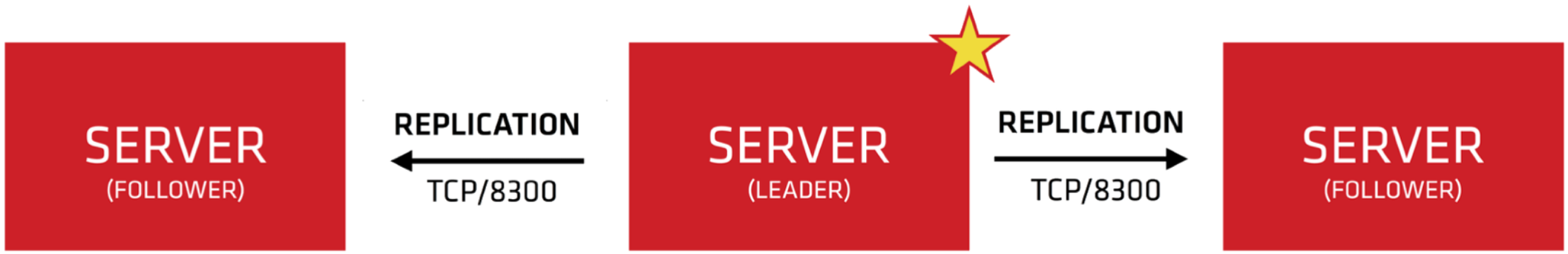
Consul: Agents

DATACENTER 1

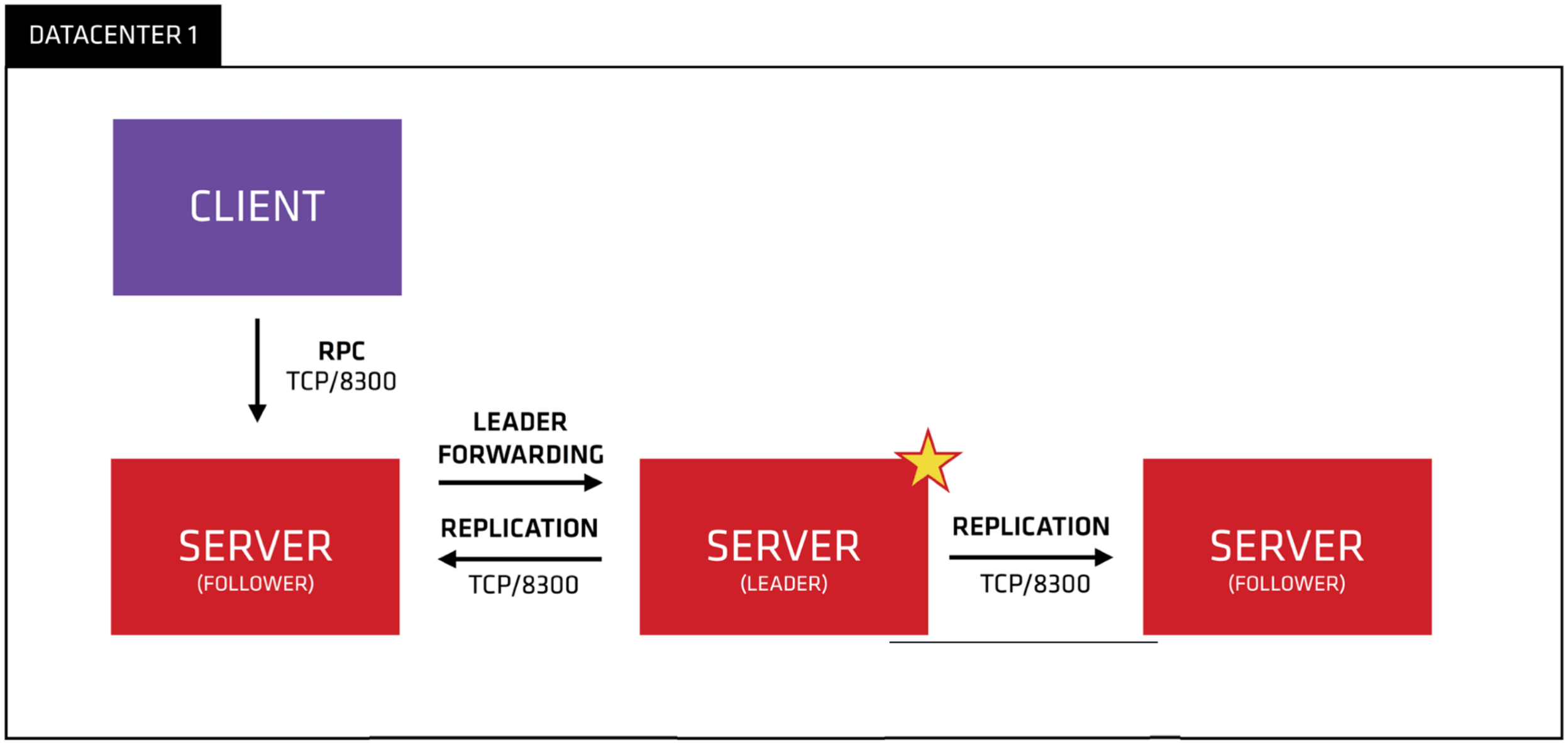


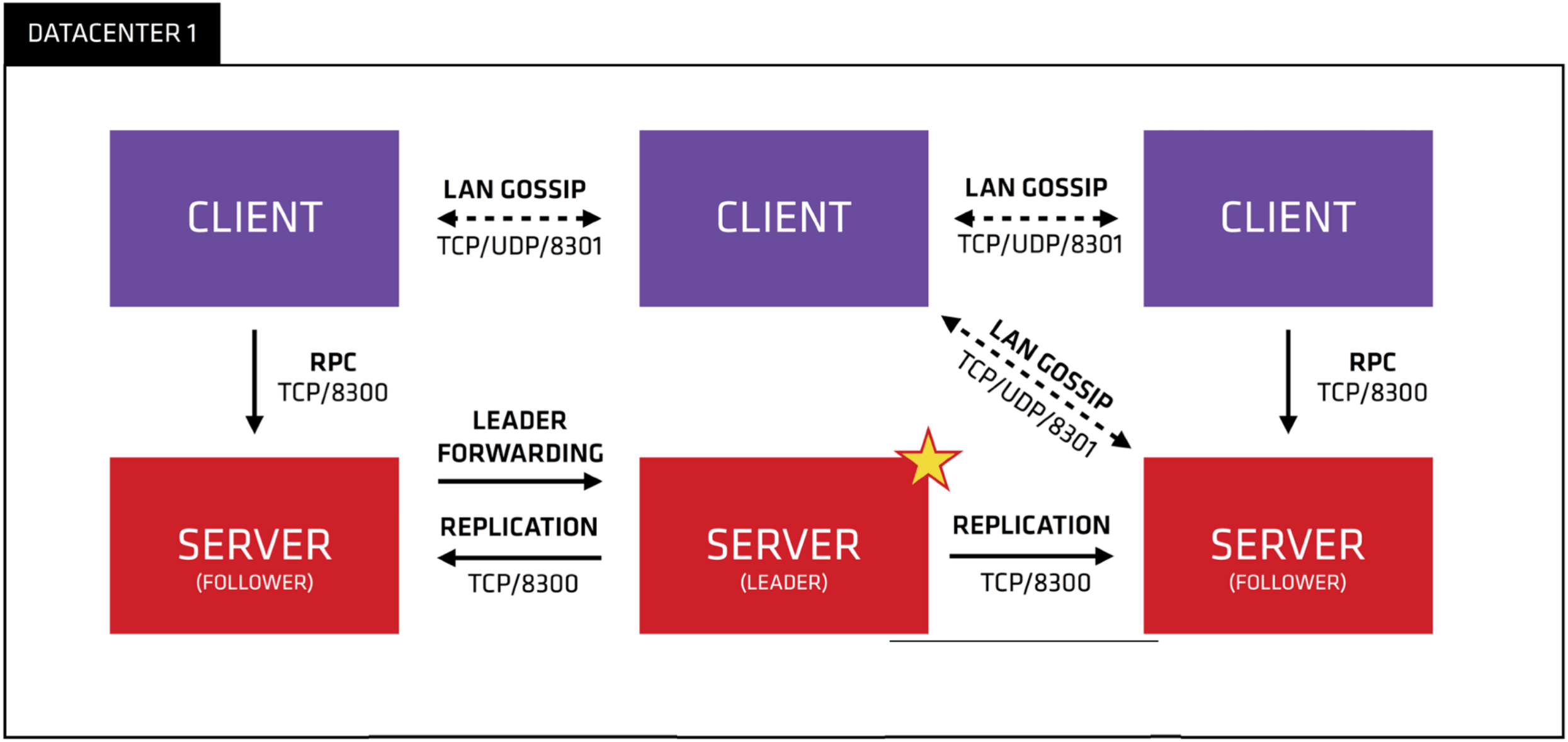
HashiCorp

DATACENTER 1



HashiCorp


DATACENTER 1
CLIENT
RPC
TCP/8300
LEADER
FORWARDING
REPLICATION
TCP/8300
SERVER
(FOLLOWER)
SERVER
(LEADER)
REPLICATION
TCP/8300
SERVER
(FOLLOWER)


DATACENTER 1
CLIENT
LAN GOSSIP
TCP/UDP/8301
CLIENT
LAN GOSSIP
TCP/UDP/8301
CLIENT
LAN GOSSIP
TCP/UDP/8301
RPC
TCP/8300
RPC
TCP/8300
LEADER
FORWARDING
REPLICATION
TCP/8300
REPLICATION
TCP/8300
SERVER
(FOLLOWER)
SERVER
(LEADER)
SERVER
(FOLLOWER)

DATACENTER 1

CLIENT

LAN GOSSIP
TCP/UDP/8301

CLIENT

LAN GOSSIP
TCP/UDP/8301

CLIENT

RPC
TCP/8300

LAN GOSSIP
TCP/UDP/8301

RPC
TCP/8300

LEADER
FORWARDING

REPLICATION
TCP/8300

SERVER
(FOLLOWER)

SERVER
(LEADER)

REPLICATION
TCP/8300

SERVER
(FOLLOWER)

WAN GOSSIP
TCP/UDP/8302

TCP/8300

INTERNET

WAN GOSSIP
TCP/UDP/8302

REMOTE DC FORWARDING
TCP/8300

DATACENTER 2

LEADER
FORWARDING

REPLICATION
TCP/8300

SERVER
(FOLLOWER)

SERVER
(LEADER)

REPLICATION
TCP/8300

SERVER
(FOLLOWER)

Consul

DEMO Setup

Consul

- Setup Consul

  consul agent -server -bootstrap-expected=1 -bind 127.0.0.1 -data-dir ./data -ui

- Key Values

  consul kv put config/new-entry/foo bar
  consul kv get config/new-entry/foo
  bar

- Refresh

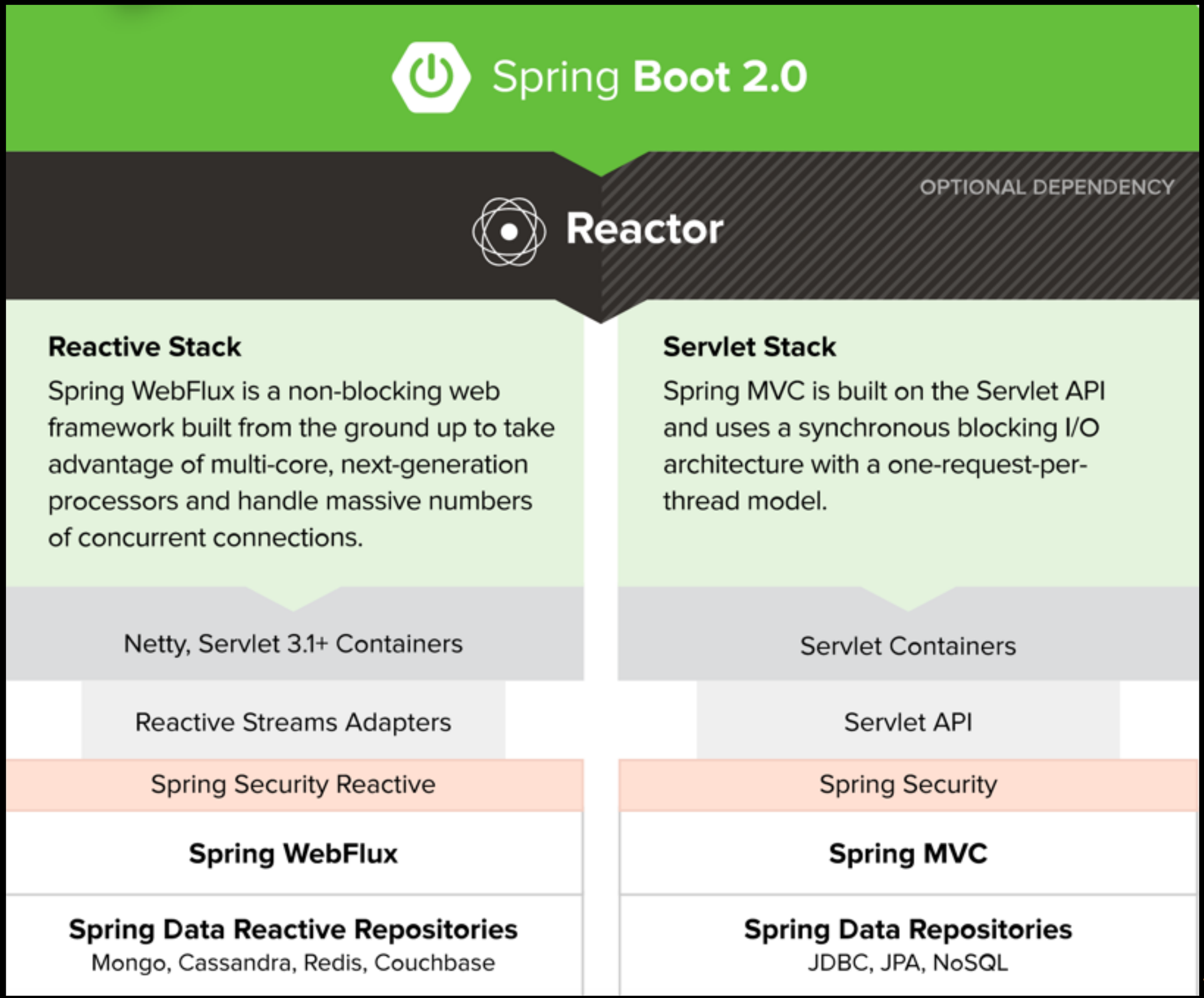
  curl -s -X POST http://localhost:8080/application/refresh -o-

- Registration / DNS

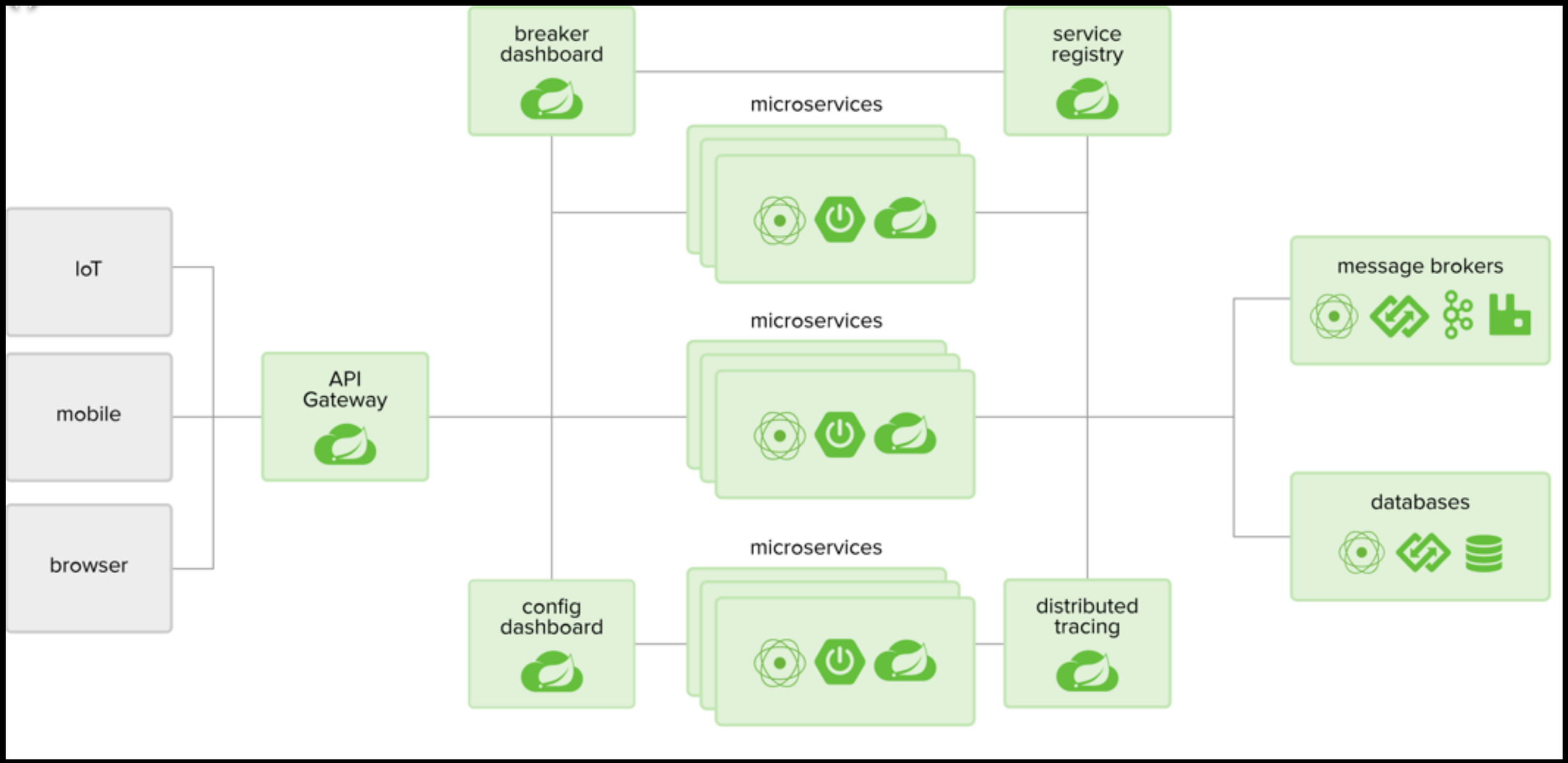
  dig $HOST.node.consul @localhost -p 8600
  dig $SERVICE.service.consul @localhost -p 8600

Spring Platform

Spring Platform



Spring Platform



Spring Cloud Consul

- Create Simple Service
- Demonstrate Properties via Consul
- Discover Service

Spring InitializR

Secure https://start.spring.io

Apps

SPRING INITIALIZR bootstrap your application now

Generate a Maven Project with Java and Spring Boot 1.5.8

Project Metadata

Artifact coordinates

Group

com.example

Artifact

demo

Dependencies

Add Spring Boot Starters and dependencies to your application

Search for dependencies

Web, Security, JPA, Actuator, Devtools...

Selected Dependencies

Web × Actuator × Consul Configuration × Consul Discovery ×

Generate Project ⌘ + ⏎

Don't know what to look for? Want more options? Switch to the full version.

LSE

Spring Cloud Consul

DEMO

Spring Cloud Consul: Summary

- Easy to setup and run on Spring Services
- Can manage properties centrally
- Can register and discover services
  Spring / JVM and DNS based
- And much, much more…

Spring Cloud Consul: Resources

- Spring Cloud Consul
  https://cloud.spring.io/spring-cloud-consul
- HashiCorp Consul
  https://consul.io
- Consul with .NET services
  https://www.dotnetcatch.com/2016/12/30/intro-to-distributed-config-with-consul-on-asp-net-core/
- Spring Actuator
  http://www.baeldung.com/spring-boot-actuators
- Spring Zuul Gateway (proxy)
  http://www.baeldung.com/spring-rest-with-zuul-proxy

Questions ?

- Slides: https://github.com/lseinc/intro-consul.git

Thanks Everyone !!!
Including TEKsystems and Manifest Solutions

David Lucas
Lucas Software Engineering, Inc.
www.lse.com
ddlucas@lse.com
@DavidDLucas

L S E

LSE