Leveraging Spring Cloud Consul for Discovery

Presenter: David Lucas

LSE

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
- Do not work for HashiCorp

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

L S E

Discovering Springy Consoles

Hear of HashiCorp?

- **Consul** [key/values and discovery] (https://consul.io)
- Vagrant [developer setup] (https://www.vagrantup.com)
- Terraform [infrastructure as code] (https://terraform.io)
- Nomad [scheduling and orchestration] (https://nomadproject.io)
- Vault [certs and secret management] (https://vaultproject.io)

Spring Cloud Consul: Introduction

- What exactly is the problem?
- How does Consul help ?
- How does it integrate with Spring Cloud?
- Summary

Services

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

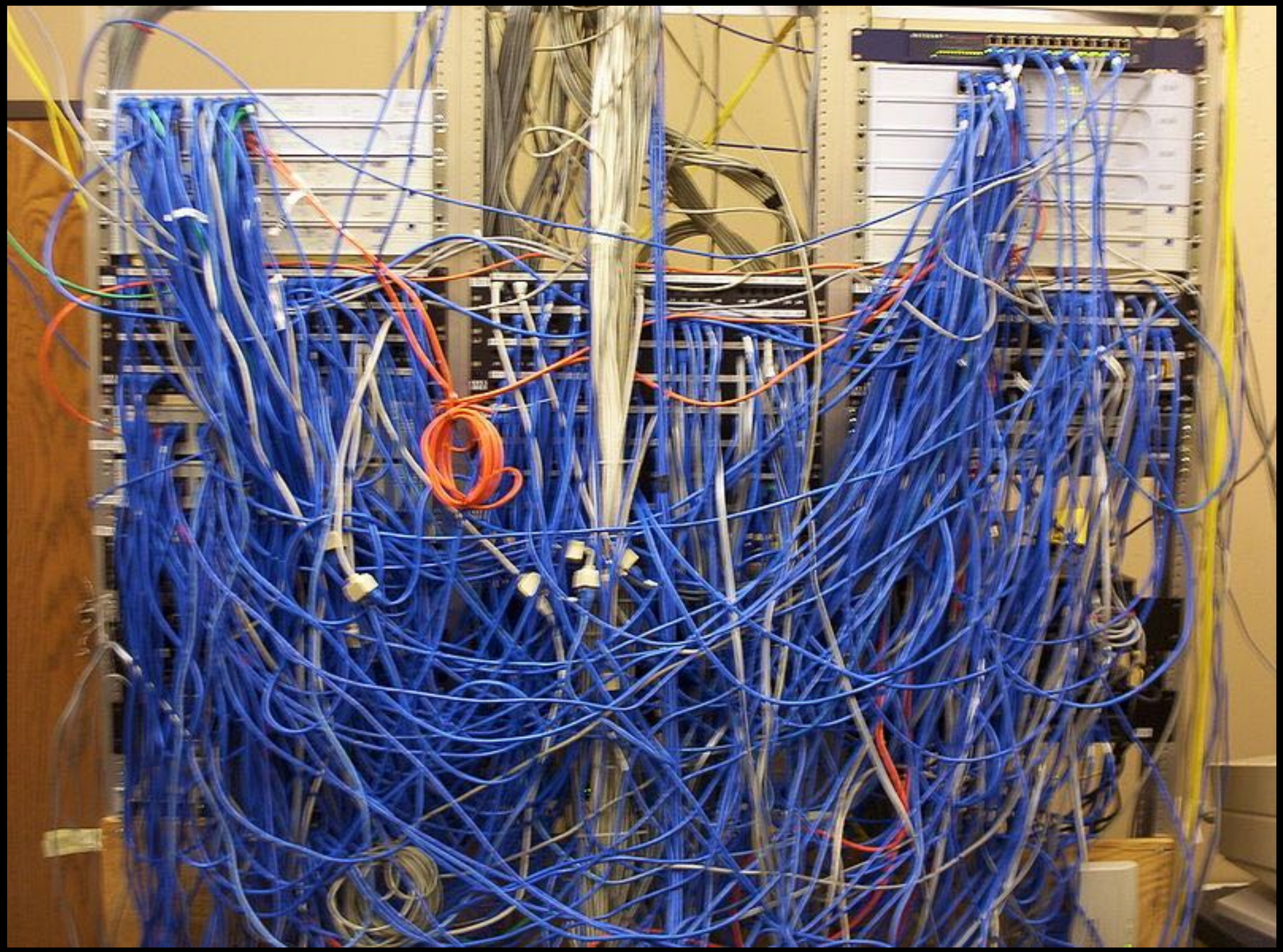
1. *One codebase tracked in revision control, many deploys*
2. *Explicitly declare and isolate dependencies*
3. *Store config in the environment*
4. *Treat backing services as attached resources*
5. *Strictly separate build and run stages*
6. *Execute the app as one or more stateless processes*
7. *Export 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*

Services

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*

Configuration



Configuration

application.properties:
service.foo.url=http://foo-node-1:1234
service.foo.username=service-id-01
service.foo.password=password

Configuration

Configuration

application.properties:
service.foo.url=http://localhost:8080

config:/application:
service.foo.url=http://foo-node-1:1234
service.foo.username=service-id-01
service.foo.password=password

Discovery

Discovery

Discovery

Discovery

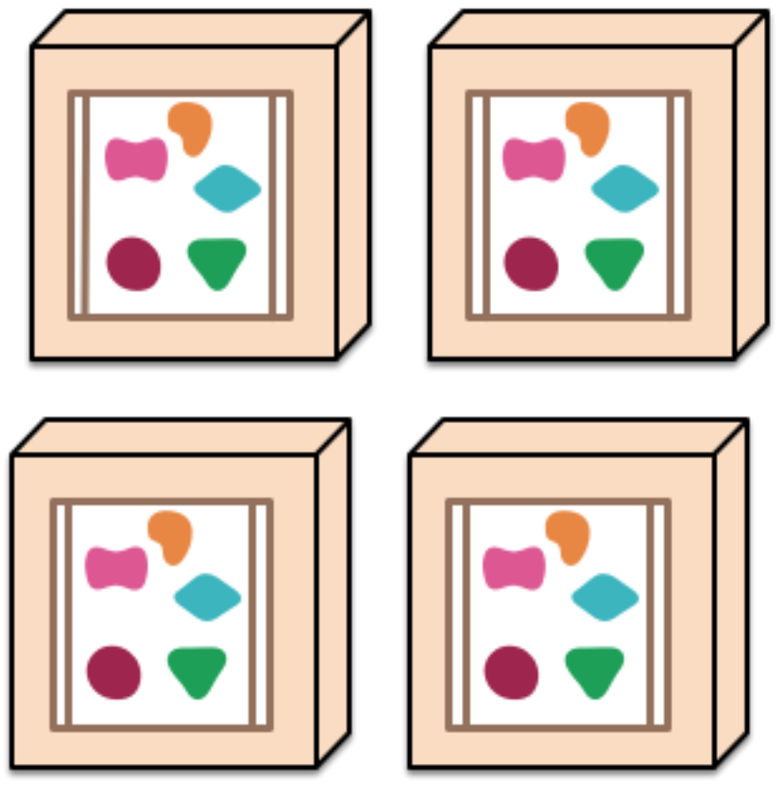
application.properties:
service.foo.url=http://localhost:8080

config:/application:
service.foo.url=http://foo-service
service.foo.username=service-id-01
service.foo.password=password

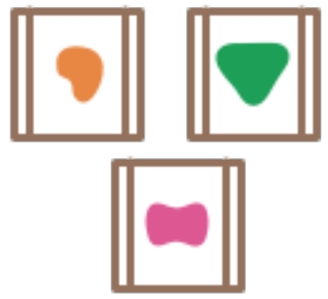
Services

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

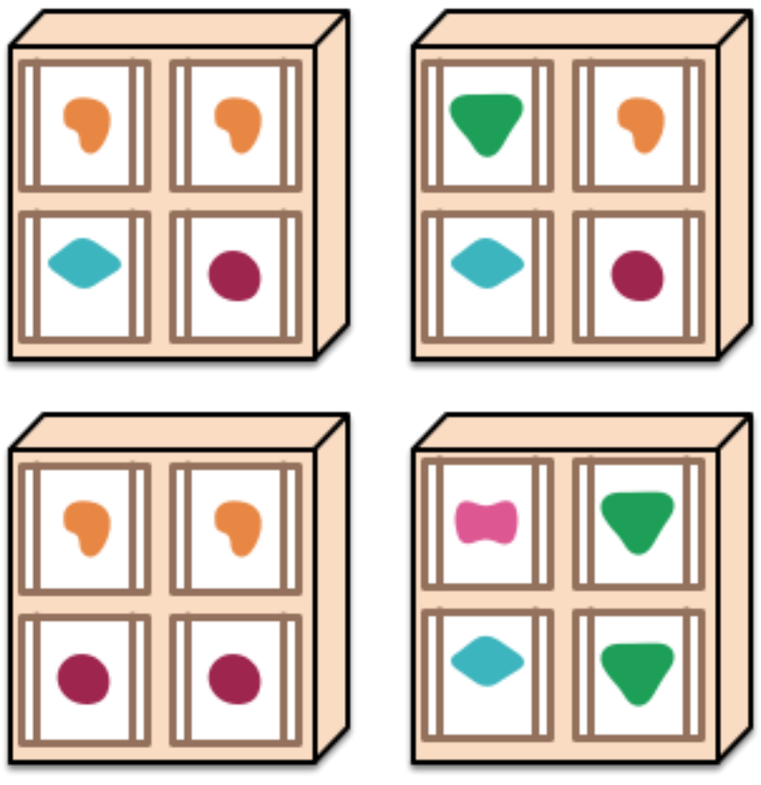
... and scales by replicating the monolith on multiple servers



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



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



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

Services

- Traditional Application Service Lookup
  - JNDI within cluster
  - HTTP Server plugged into to cluster
  - HTTP URL DNS / VIP mapped to nodes

- Microservices Lookup
  - Hard Coded host : port ???
  - Discovery ???
  - HTTP URL DNS / VIP mapped to nodes

Consul: Introduction

- HashiCorp's Swiss Army Knife
- Introduced 0.10 in 2014
- Light Weight Go Binary (client/server)
- Key Value Pairs
- Registry, Discovery and Health Check
- DNS
- Event Bus

Consul Introduction

- Configuration
  - Key / Value (properties)
  - Hierarchy of key / values
  - Watch (change notification)
- Discovery
  - service registers location
  - client finds healthy service
- DNS
- Event Bus

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
  @LoadBalanced

Consul: Introduction

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

- Leverage 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)

- aedile (useful for import / export)

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: Introduction

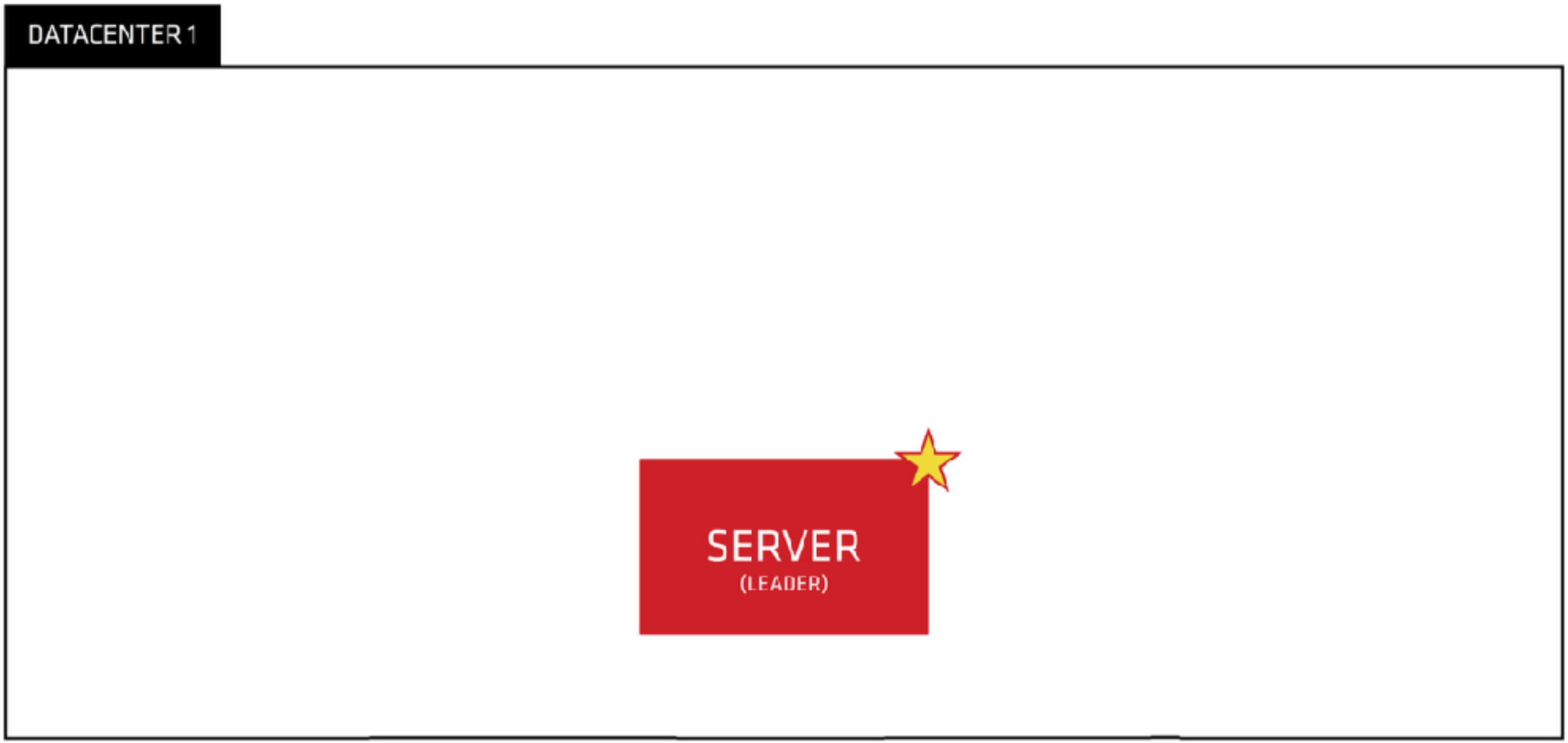
- Highly Available Clustered Information
- 3-5 (odd count) consul agent server (can run just one in development mode)
- consul agent client cached proxy
- communicates over Serf protocol
- Gossip over Serf, synchronizes data

Consul: Agents

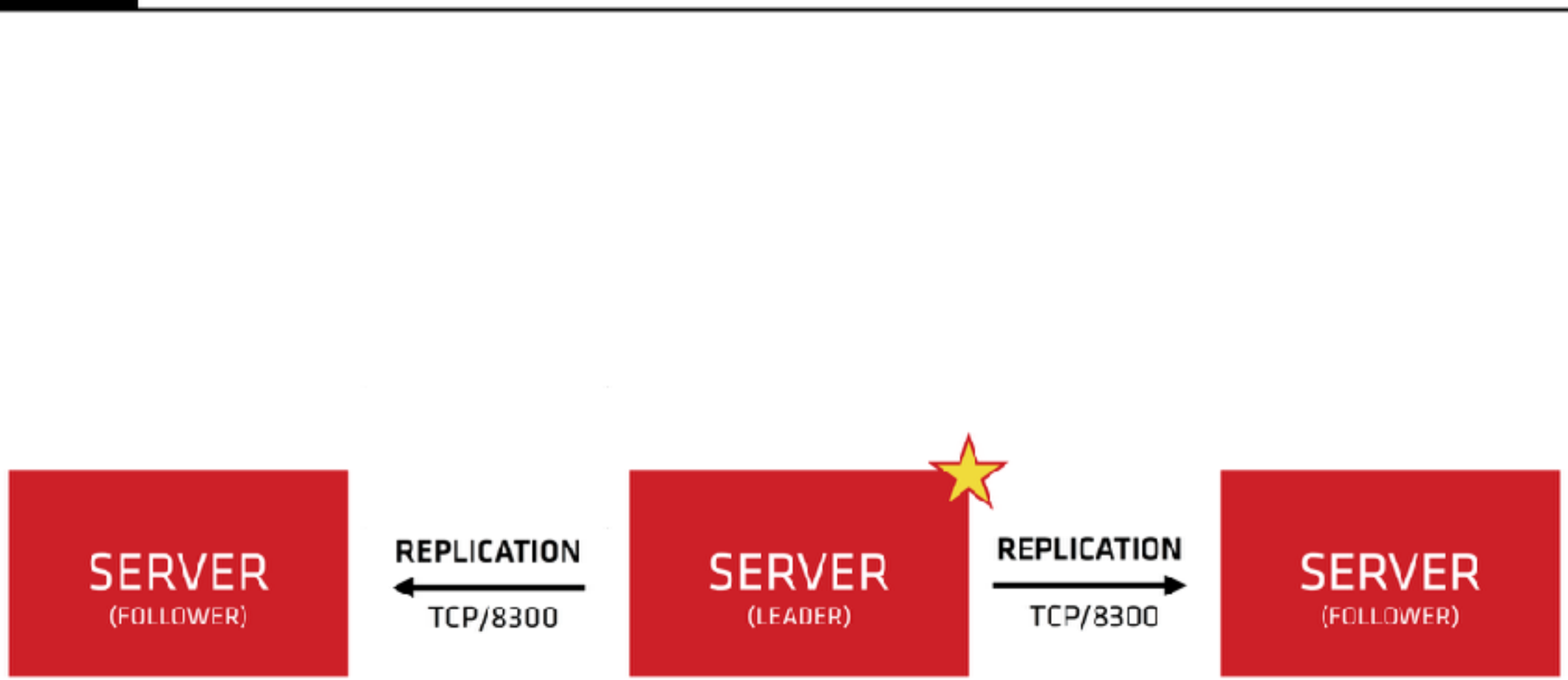
Consul: Agents

Consul: Agents

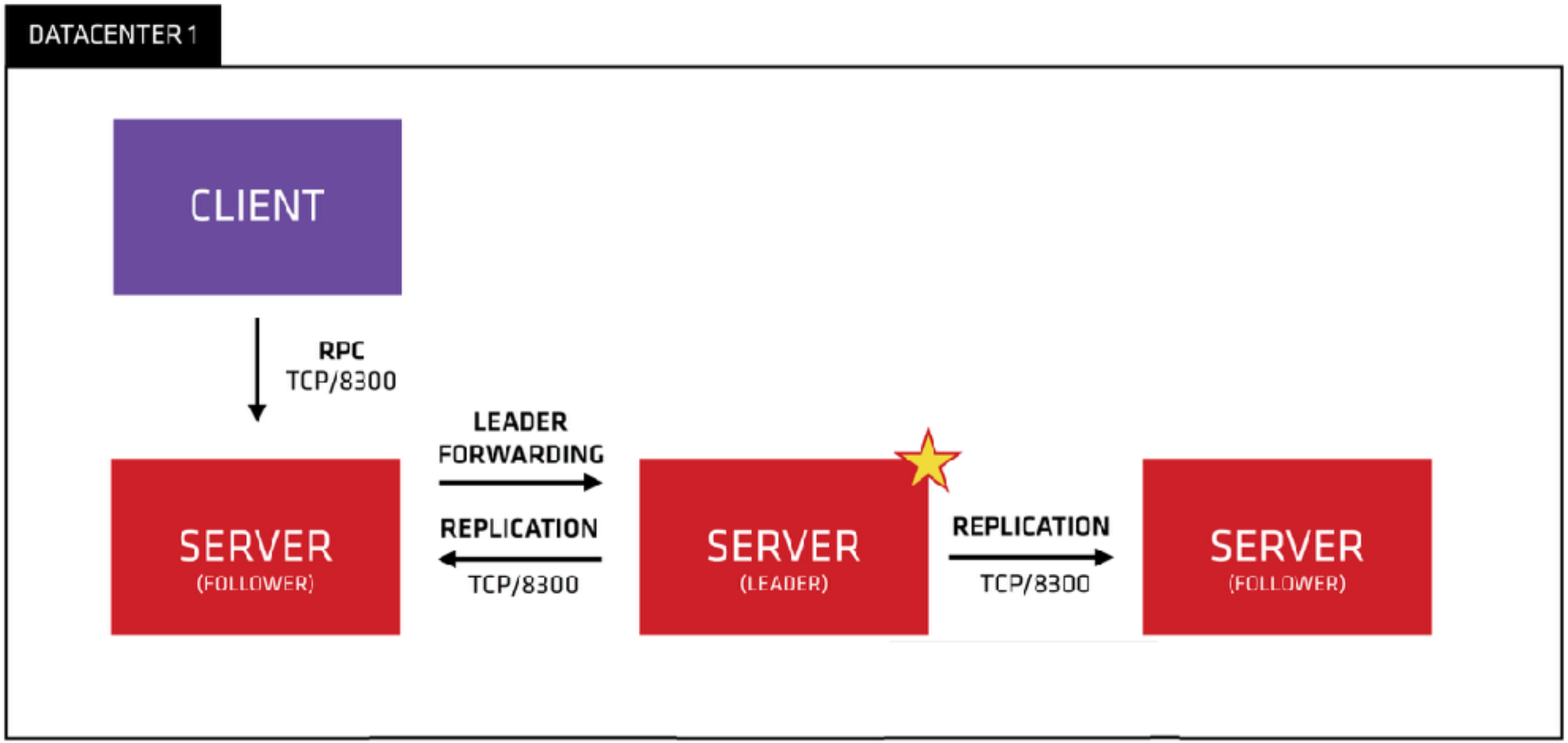
Consul: Agents

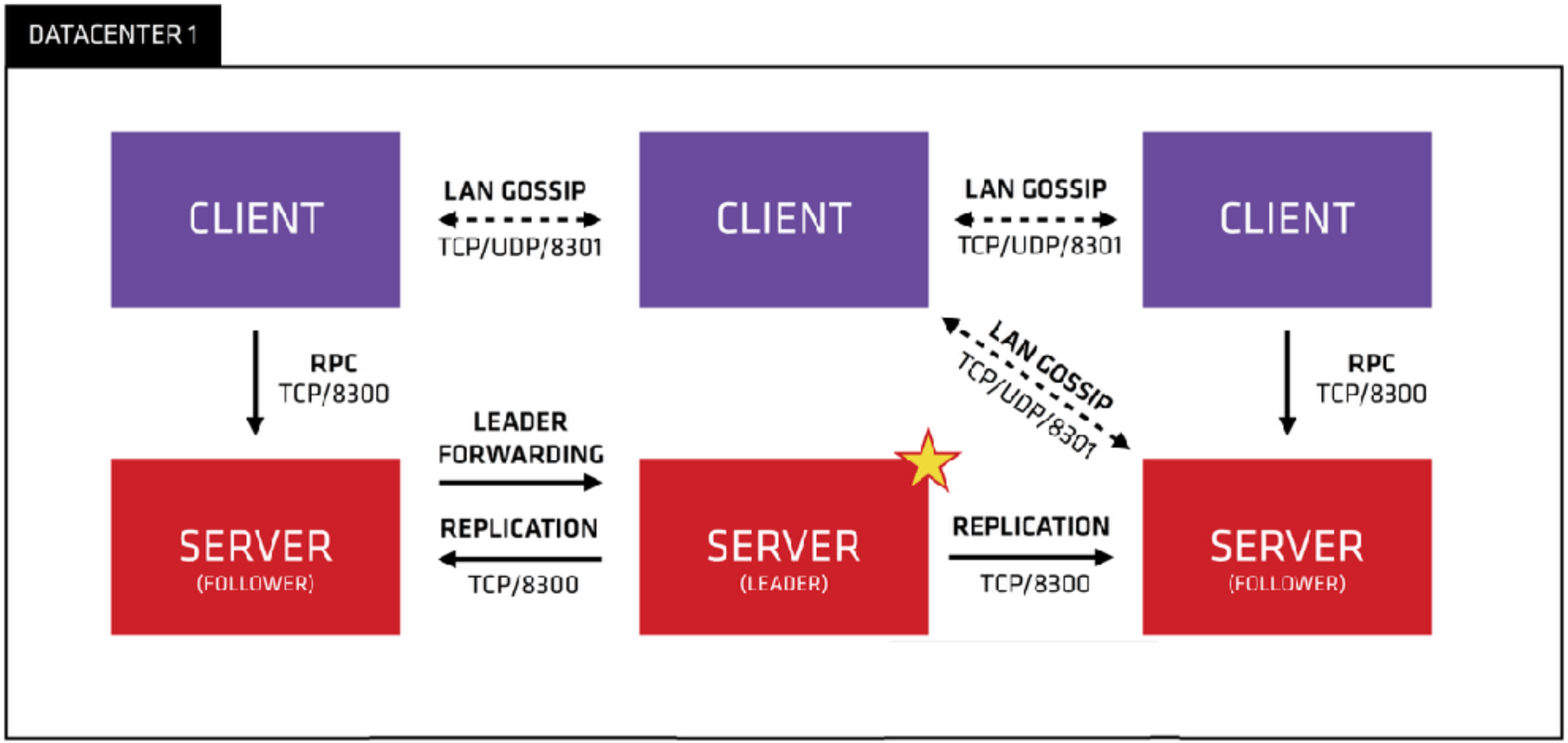


DATACENTER 1



HashiCorp


DATACENTER 1
CLIENT
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)

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

- 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

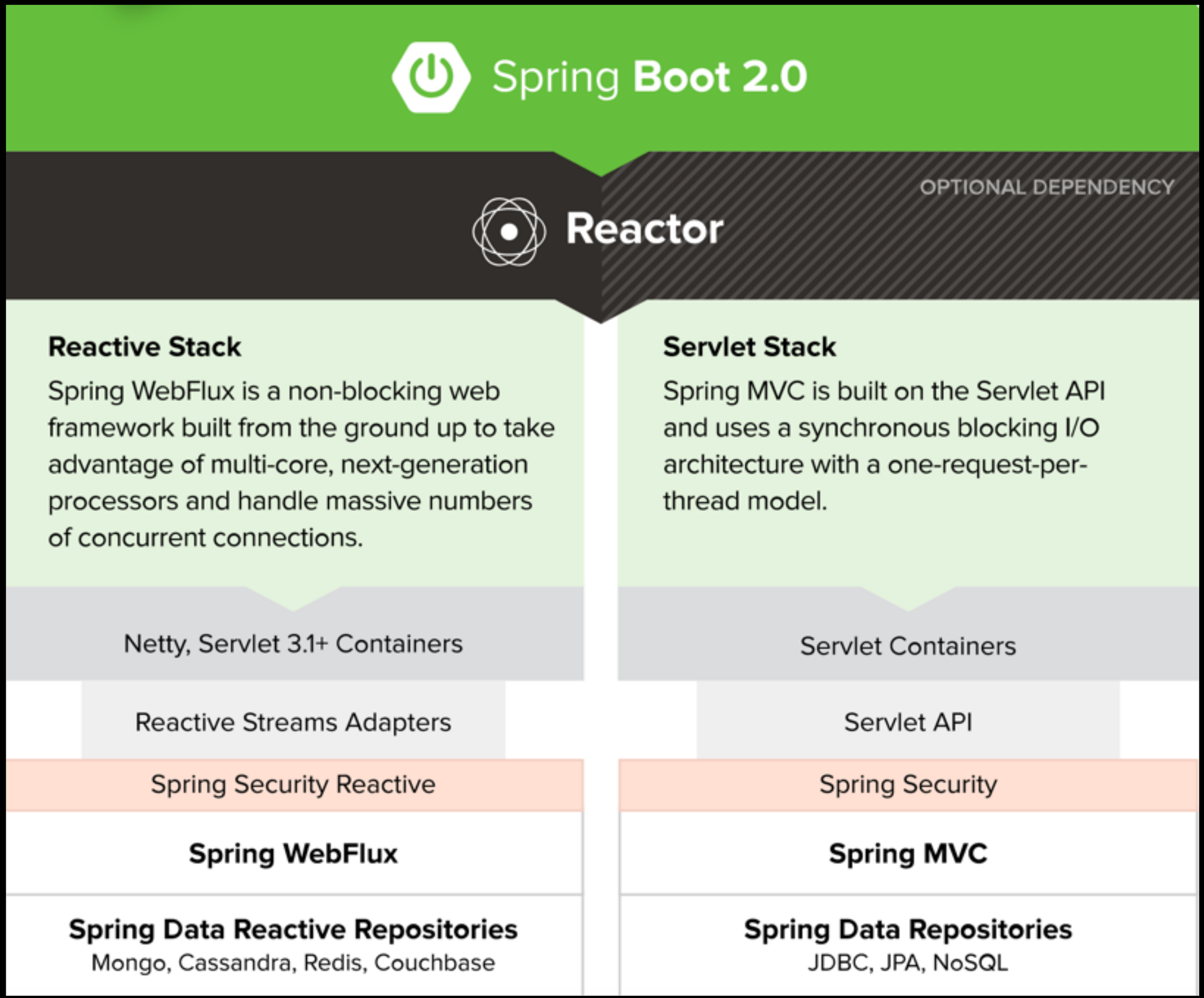
  dig $HOST.node.consul @localhost -p 8600
  dig $SERVICE.service.consul @localhost -p 8600

Consul

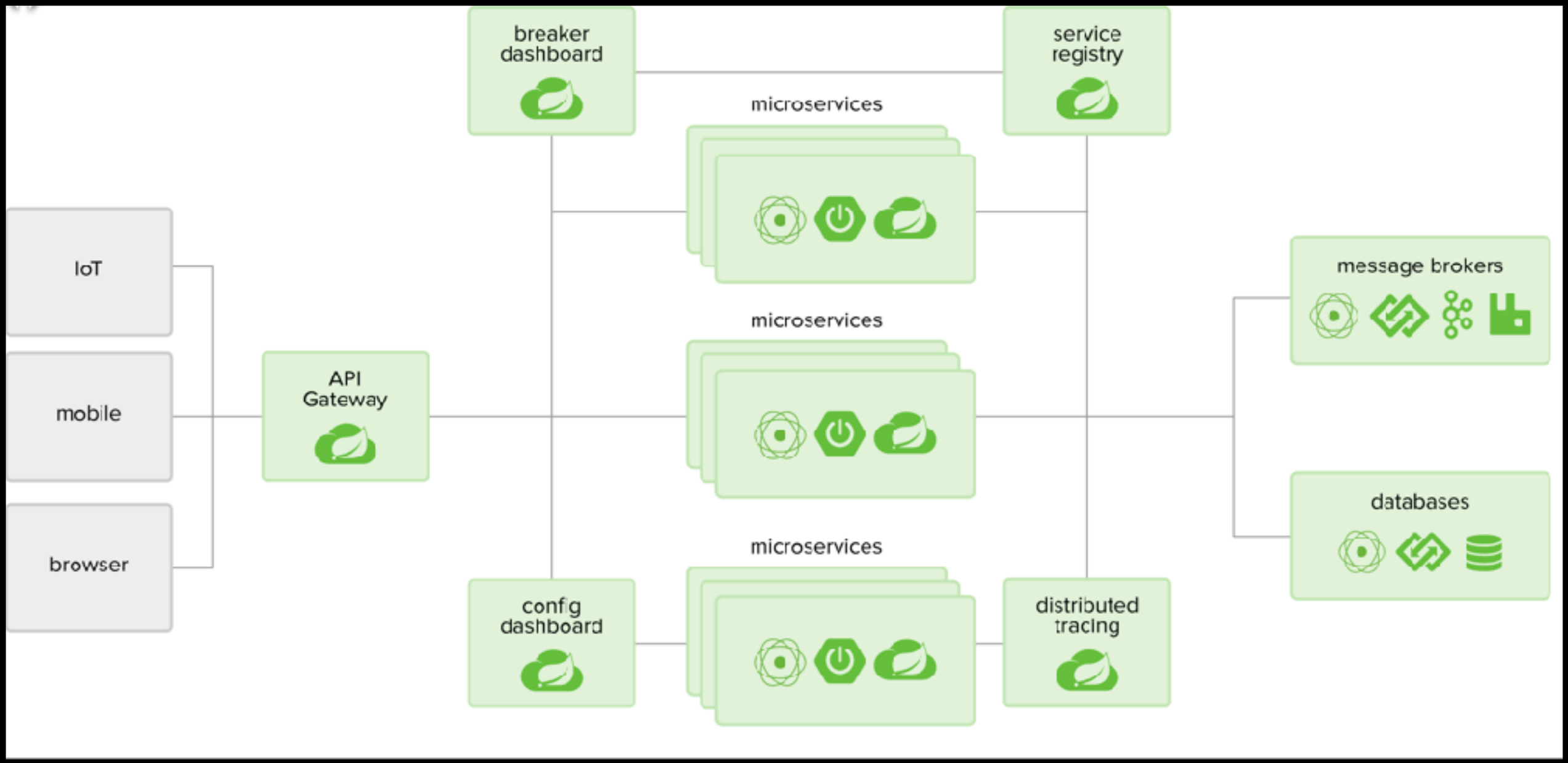
DEMO Setup

Spring Platform

Spring Platform



Spring Platform



Spring Cloud Consul

- Simple Service
- @RefreshScope / @ConfirgurationProperties
- Include Dependencies (version 2.0.0.M7)
  - org.springframework.cloud:spring-cloud-starter-consul-all
    Includes:
    spring-cloud-starter-consul-config
    spring-cloud-starter-consul-discovery
    spring-cloud-starter-consul-bus
  - org.springframework.boot:spring-boot-starter-actuator
- Zuul:
  - org.springframework.cloud:spring-cloud-starter-zuul

DEMO

Consul

8500/tcp

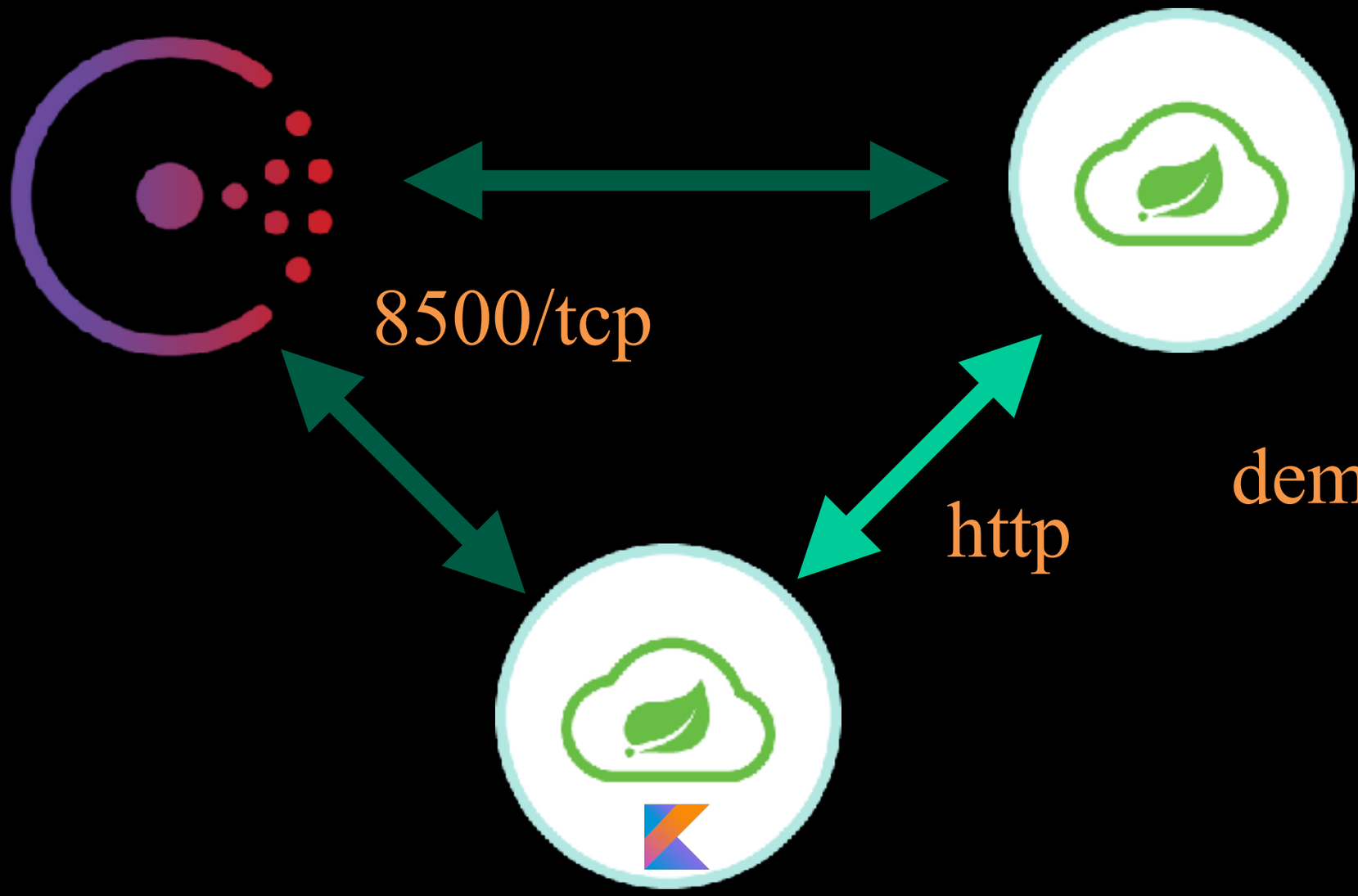
demo-service

Spring Cloud Consul

DEMO Config

Spring Cloud Consul

DEMO



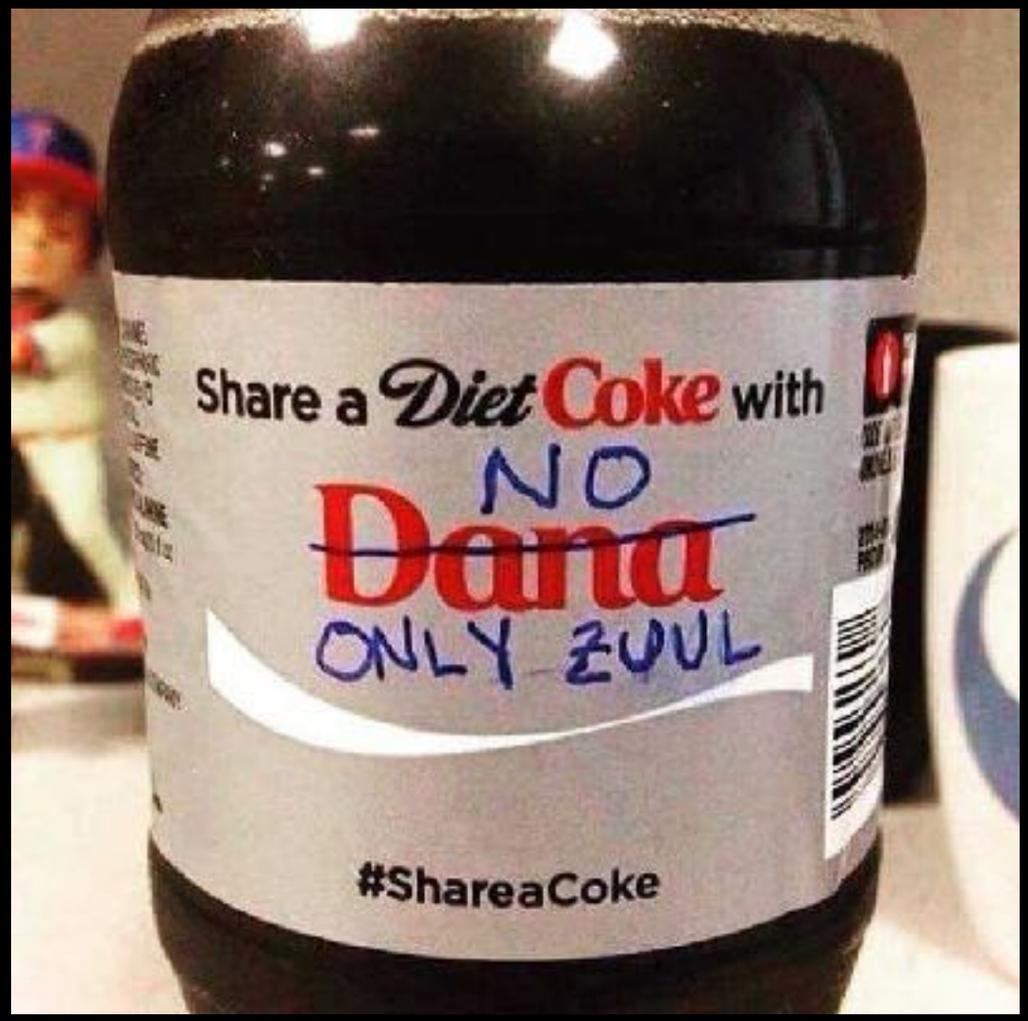
Spring Cloud Consul

DEMO Discovery

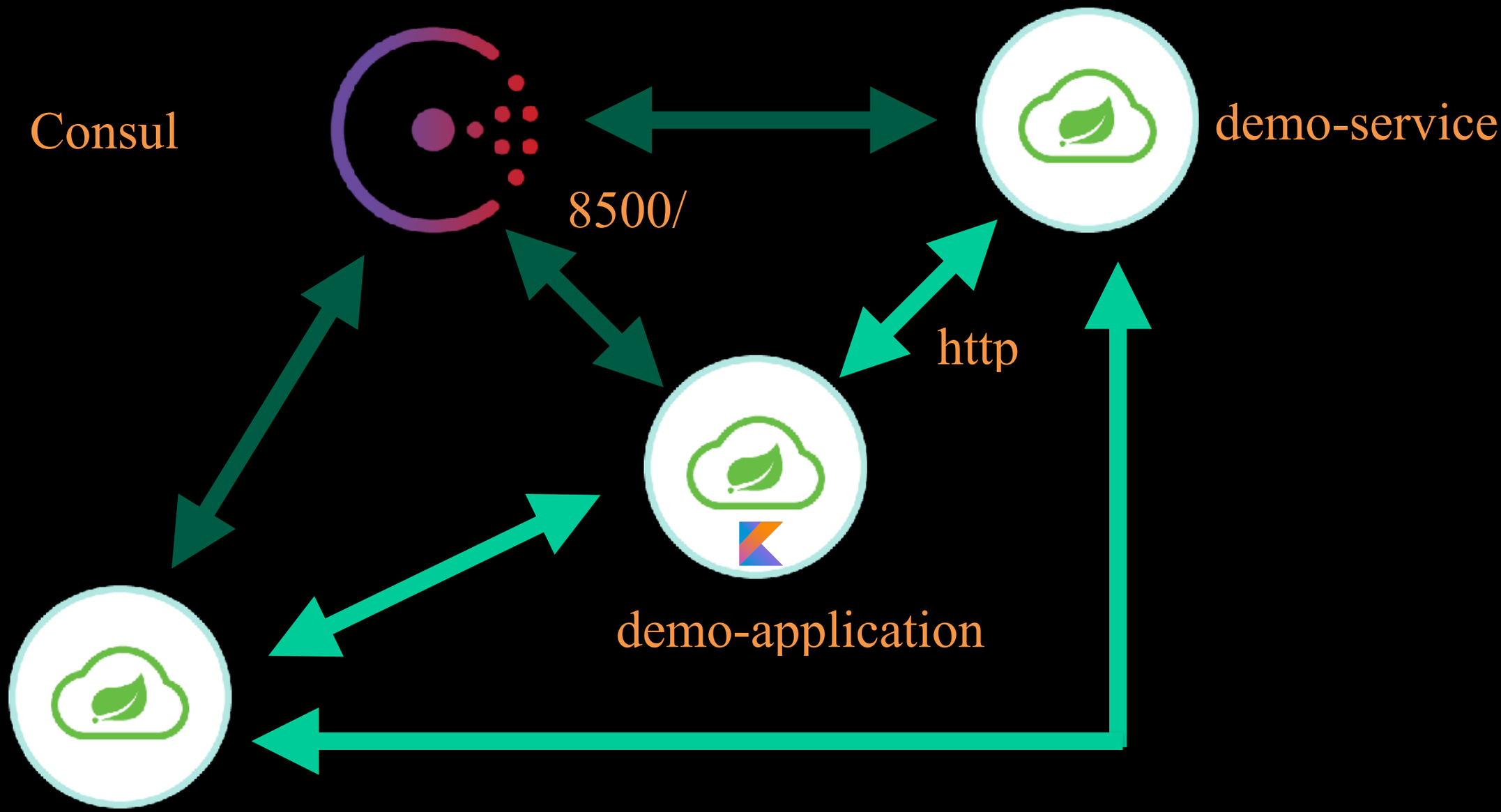
@DiscoveryClient
@LoadBalanced

Spring Cloud Consul

There is…



DEMO


Consul
8500/
demo-service
http
demo-application
gateway-server (zuul)

Spring Cloud Consul

DEMO Gateway

@EnableZuulProxy

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

WARNING: SECURE ACTUATOR
WARNING: Spring Boot 2.0 Release

Spring Cloud Consul: Other

- Consul Templates for File System: https://github.com/hashicorp/consul-template
- ENVConsul: https://github.com/hashicorp/envconsul
- Consul-Replicate (across data-centers) https://github.com/hashicorp/consul-replicate
- Consul Guides (how to setup with Vault) https://github.com/hashicorp/consul-guides
- git2consul https://github.com/breser/git2consul

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
- Consul API Docs
  https://github.com/api-stack/hashicorp-consul-api

  https://consul.docs.apiary.io

Questions ?

Slides/Code: https://goo.gl/ev36Zf

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

Thanks Everyone !!!

Don’t forget surveys!

L S E

LSE