Benefits of Cloud Thinking

OWASP Meetup, Omegapoint

October 21, 2021

Daniel Deogun & Dan Bergh Johnsson

omega point.

About us

Daniel Deogun
Coder and Quality Defender

Omegapoint

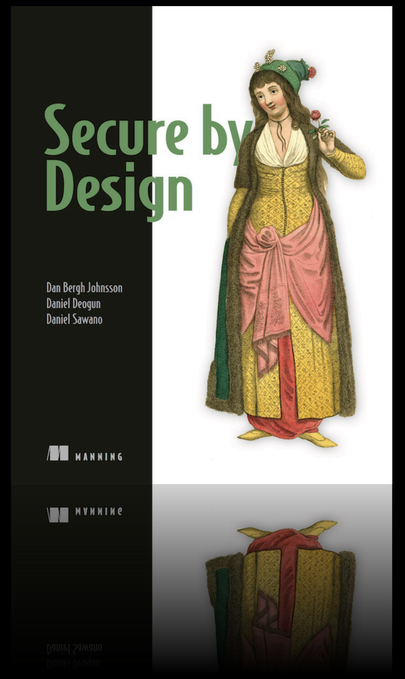
Dan Bergh Johnsson
Secure Domain Philosopher

@DanielDeogun @danbjson #SecureByDesign

omega point.

Take-aways

- By designing for change, you reduce the risk of Advanced Persistent Threats.
- Several design patterns used in the cloud can be used in a non-cloud environment.



Advanced Persistent Threat (APT)

- Not your backyard script-kiddy drive-by-intrusion

- Advanced = several vulnerabilities in combination
- Persistent = patiently over long time
- Threat = significant data loss or damage
  - Baltimore City Ransomware (?, 2019)
  - NotPetya attack on Ukraine (Russia? ,2017)
  - Belgacom operation (GCHQ?, 2010-13)
  - ISIS Media (NSA/US Cyber Command, 2015)
- Exploiting the fact that things seldom change
  - servers
  - credentials
  - ip-addresses
  - non-patched bugs tend to open up for attacks

Richard Patterson https://flic.kr/p/24GcRZQ

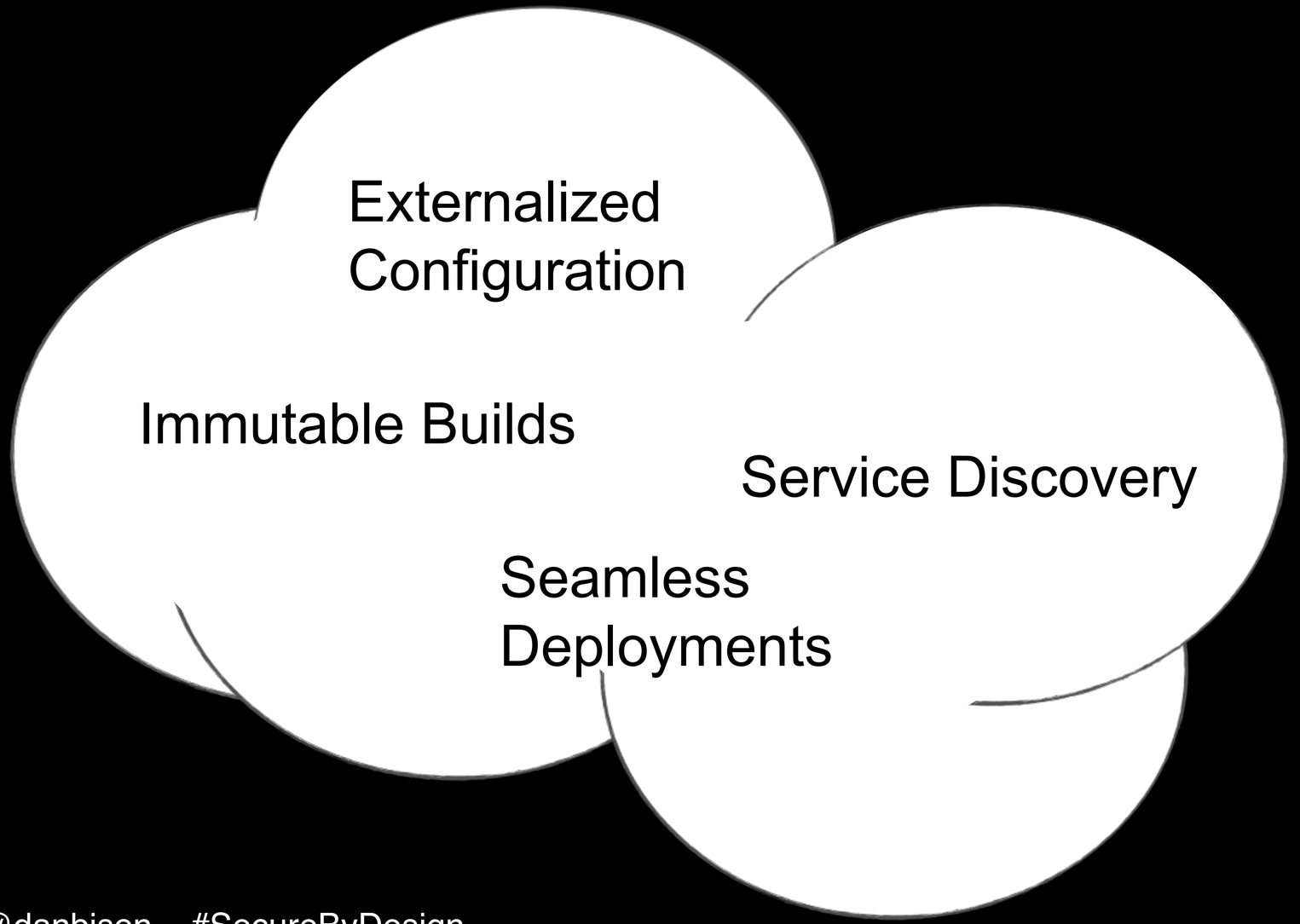
Why do we have stale environments?

- Culture
- Organizational Silos
- Budgets
- Fear
- Ceremony
- Myths & Misconceptions
- Application Design

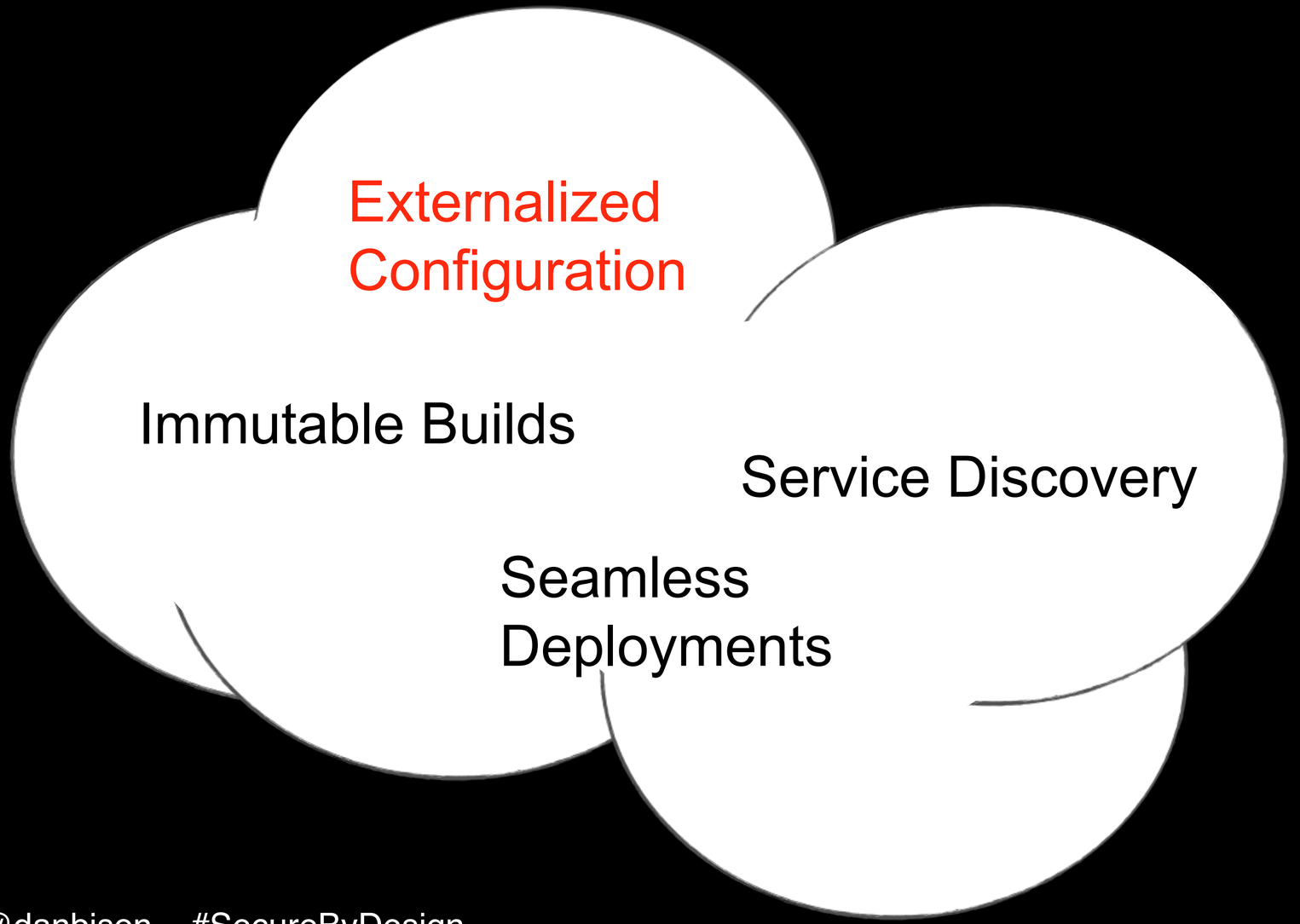
George Fryd https://flic.kr/p/jV1L6C

What if we could have an environment that embrace change?

Patterns used in the Cloud



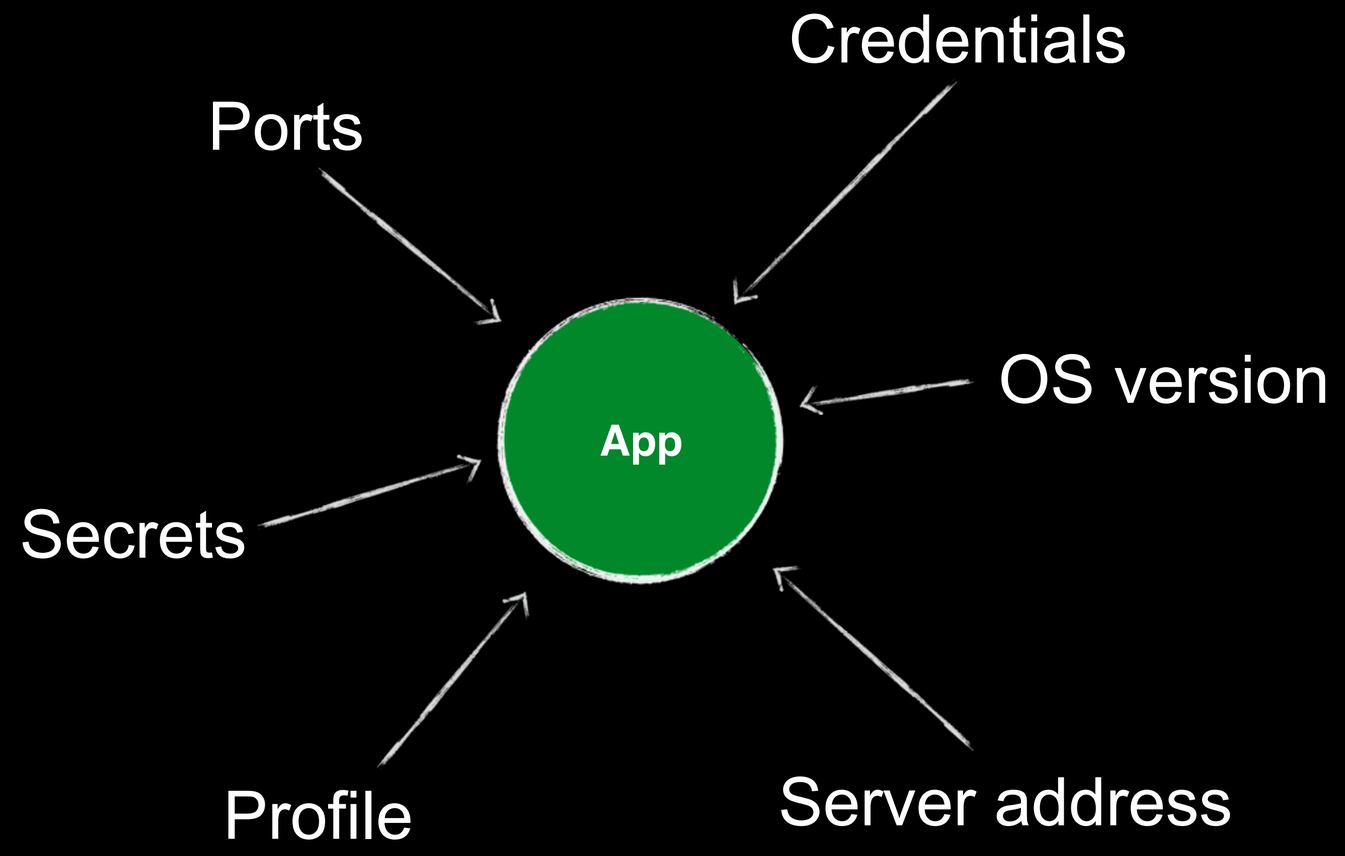
Patterns used in the Cloud



Externalized configuration

THE TWELVE-FACTOR APP

- Configuration that changes between environments belongs to the environment.
- Makes your application environment agnostic.



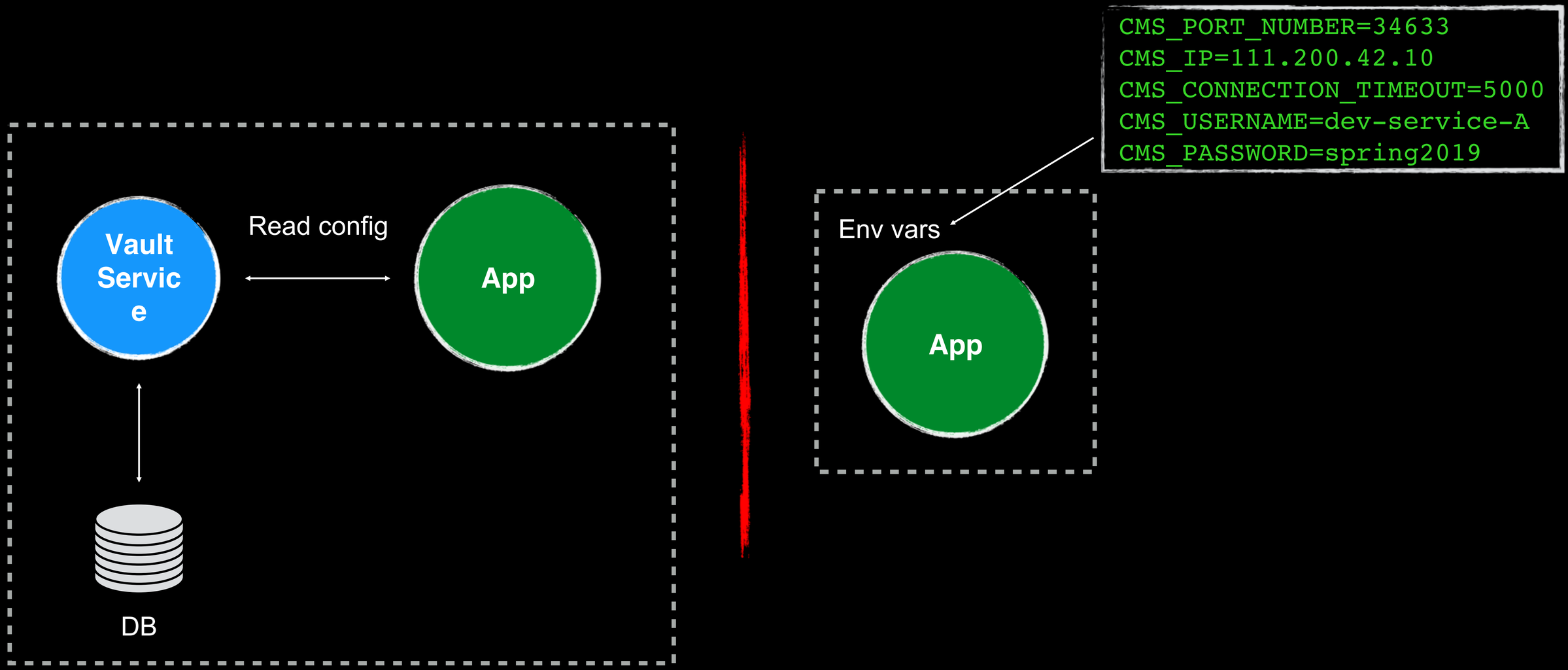
Quick & Dirty: Constants in Code

```
public class CMSConnector {
    private static final int PORT_NUMBER = 34633
    private static final String IP = "111.200.42.10";
    private static final long CONNECTION_TIMEOUT = 5000;
    private static final String USERNAME = "dev-service-A";
    private static final String PASSWORD = "spring2019";
    …
}
```

Multiple Environments: Resource Files

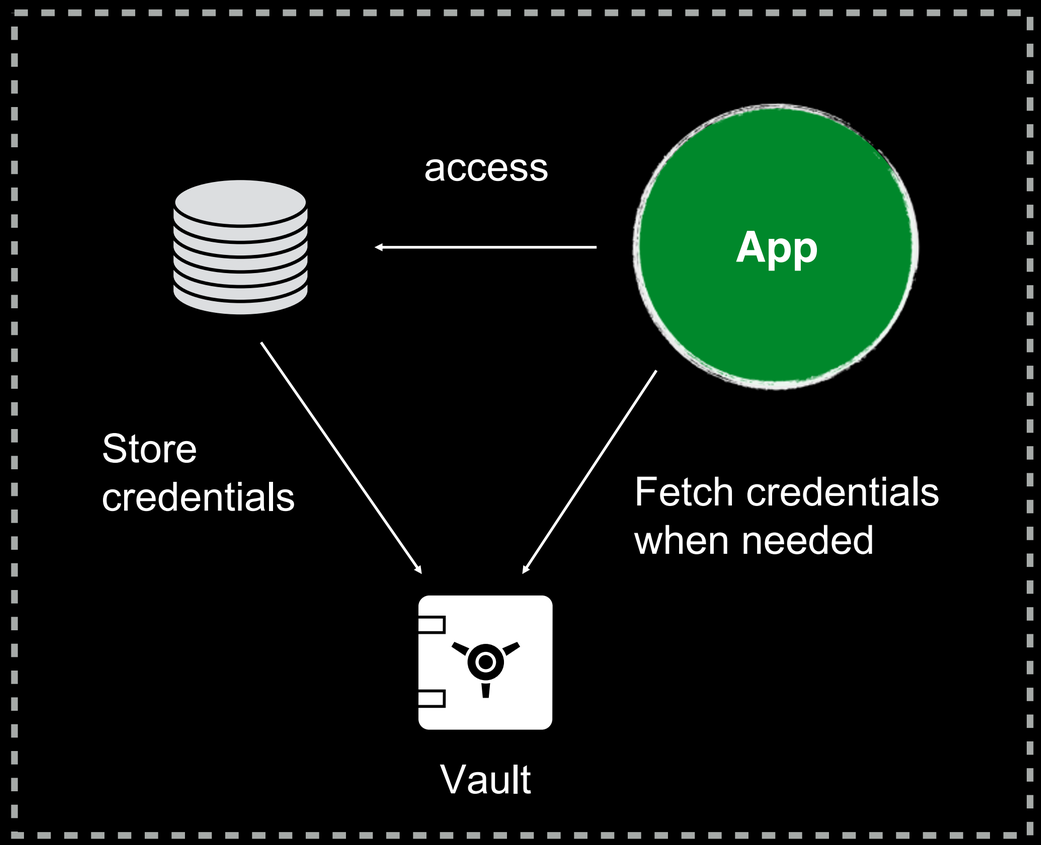
```
environments:
    dev:
        cms:
            port: 34633
            ip: 111.200.42.10
            connection-timeout: 5000
            username: dev-service-A
            password: spring2019
    prod:
        cms:
            port: 34633
            ip: 198.441.11.03
            connection-timeout: 1000
            username: service-A
            password: yC6@SX50
```

Cloud: Externalizing Configuration

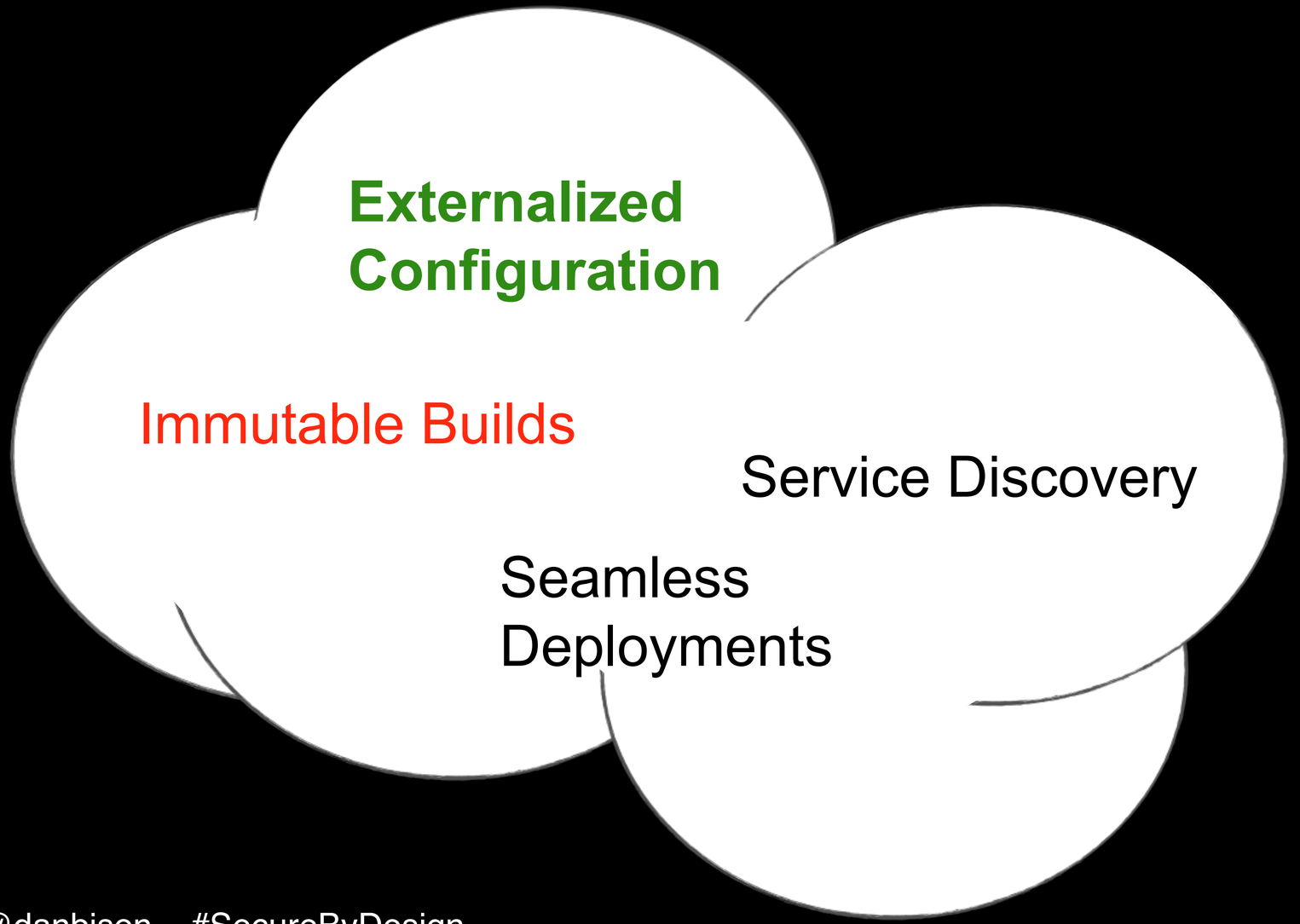


Fetch Secrets on Demand

- But providing secrets at deploy time can be problematic since it requires redeployment after updating a secret
- Secrets should be fetched / provided dynamically
- But can this be done in a non-cloud environment?



Patterns used in the Cloud



Immutable Builds

- Build once run anywhere
- Never modify an artifact – always build new
- Authoritative build
  - What's tested is what you run
  - Fully defined by repository & artifactory content
- Automatic Pipelines
  - Minimizes risk of human error
  - Repeatability

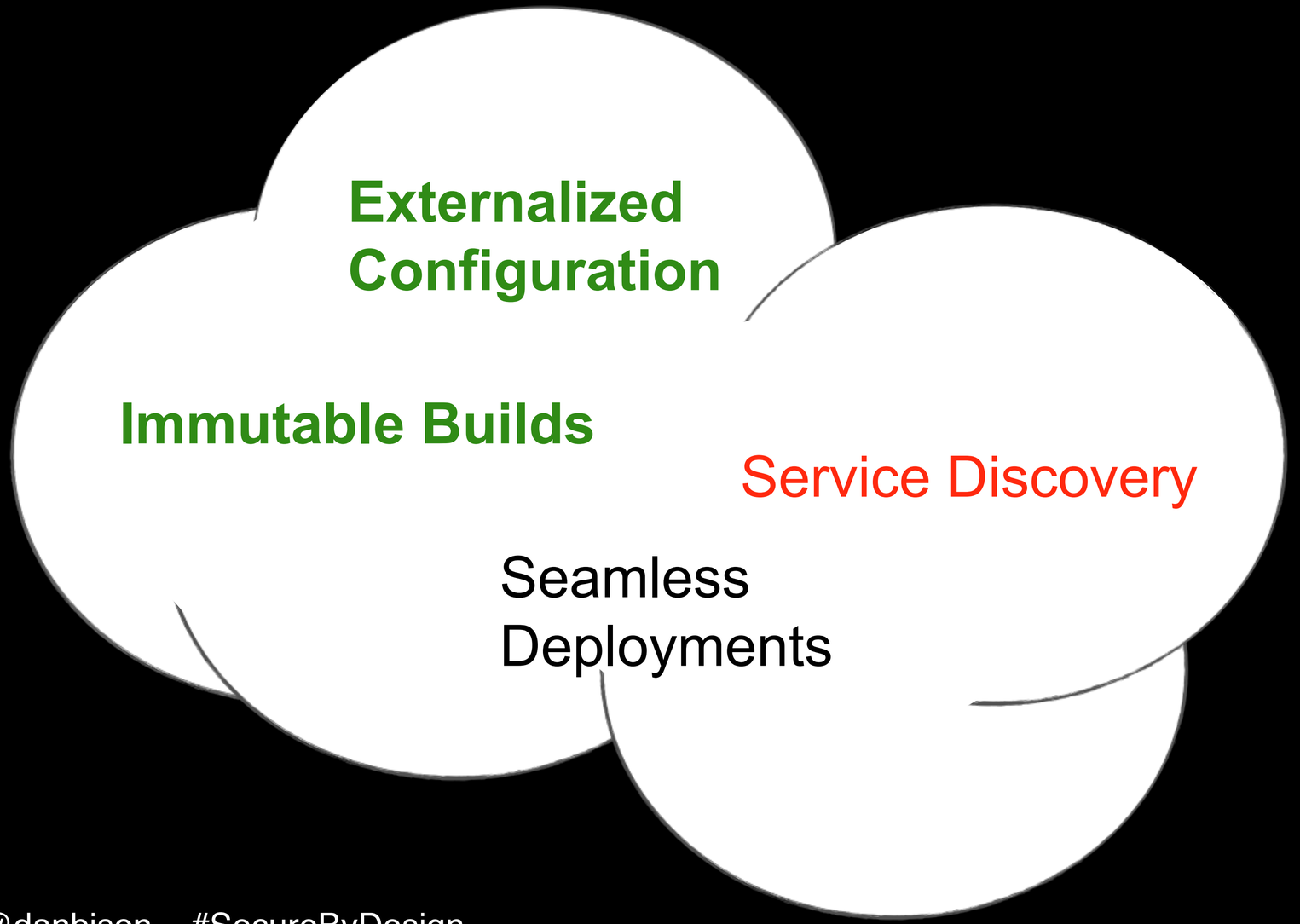
Anna Wolf https://flic.kr/p/8JUKED

Ryan McFarland www.zieak.com https://flic.kr/p/4scuyw

Spencer Cooper https://flic.kr/p/cp5WgS

Can this be done in a non-cloud environment?

Patterns used in the Cloud

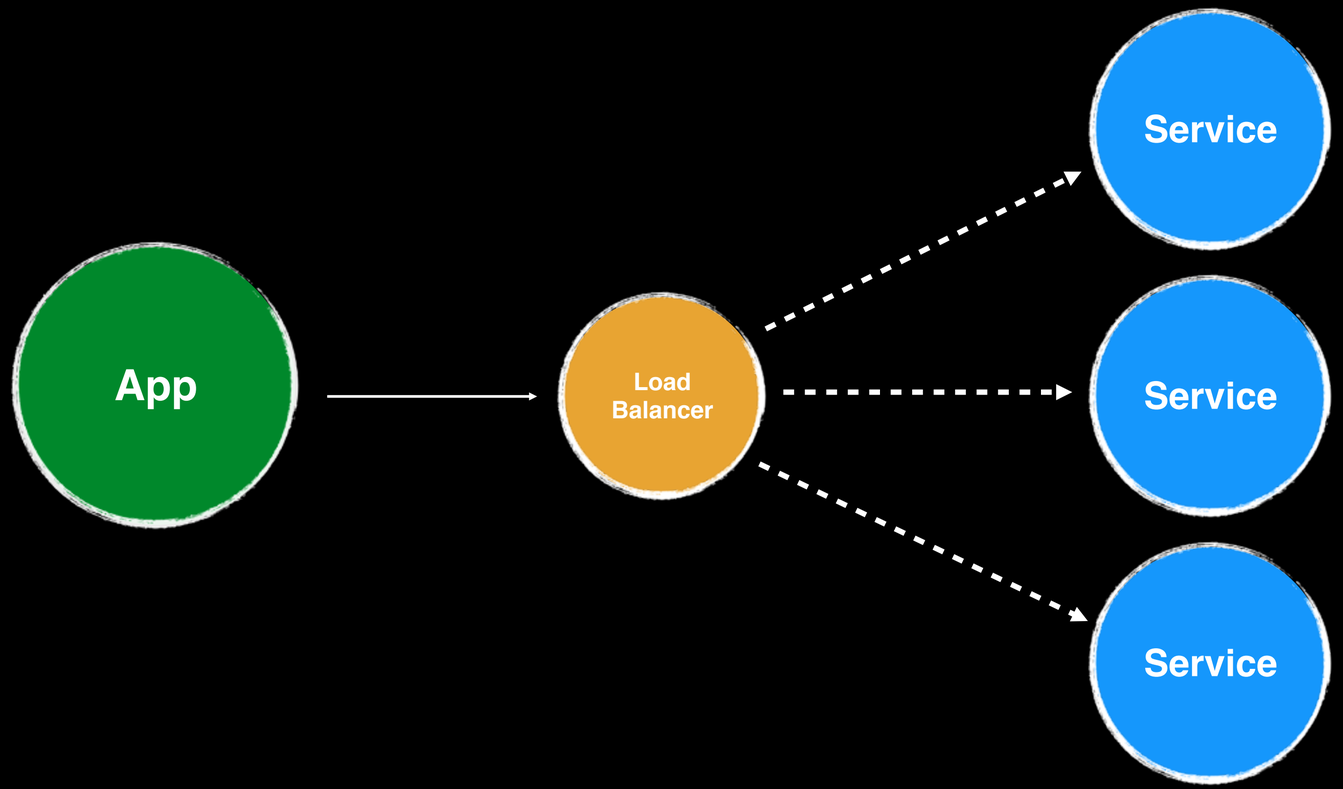


Service Discovery

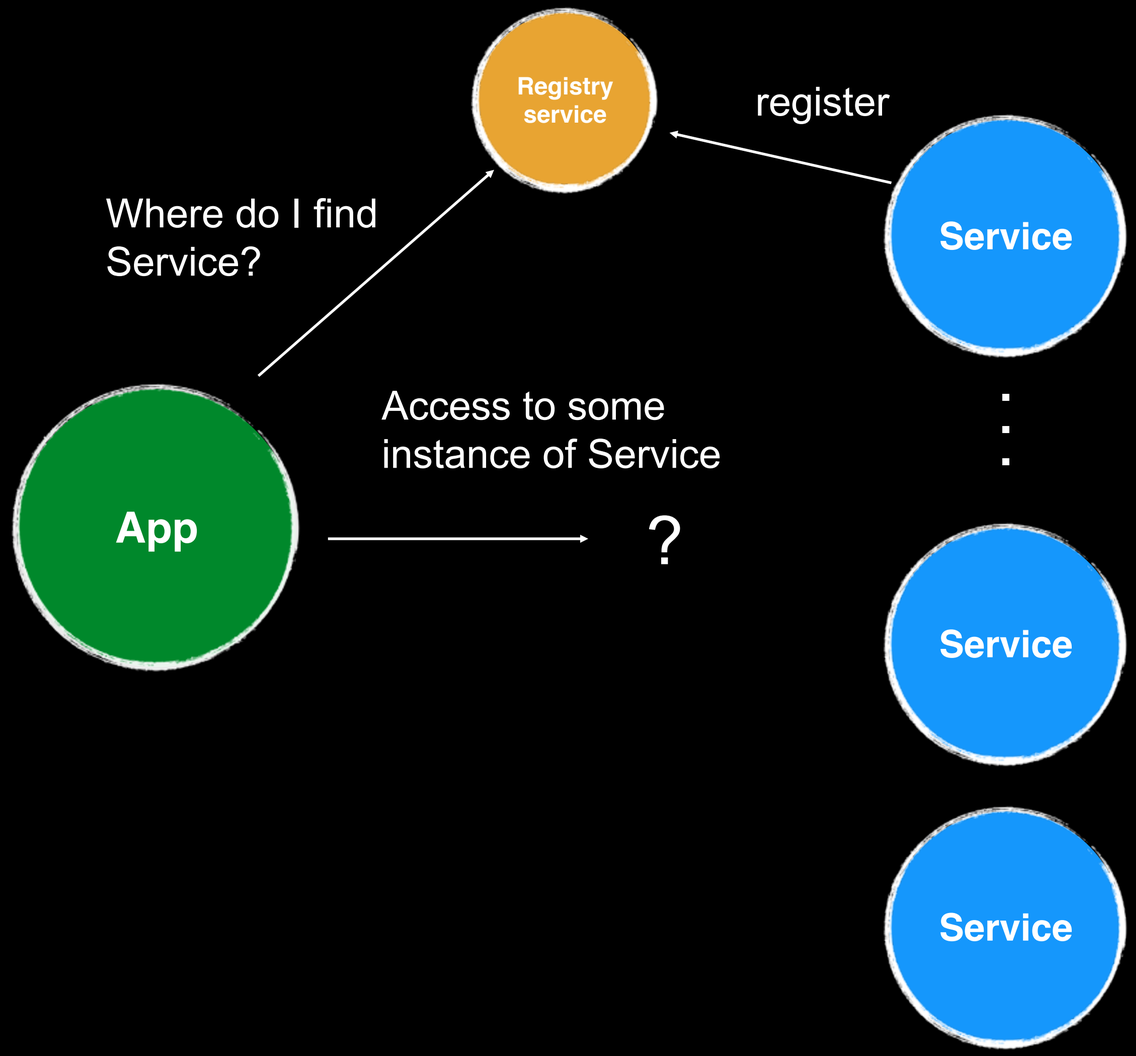
- The process of automatically discovering services in a network.
- Allows services to be loosely coupled.
- Let’s have a look at two ways of achieving service discovery

Nik Gaffney https://flic.kr/p/eb8DvH

Service Discovery using a load balancer

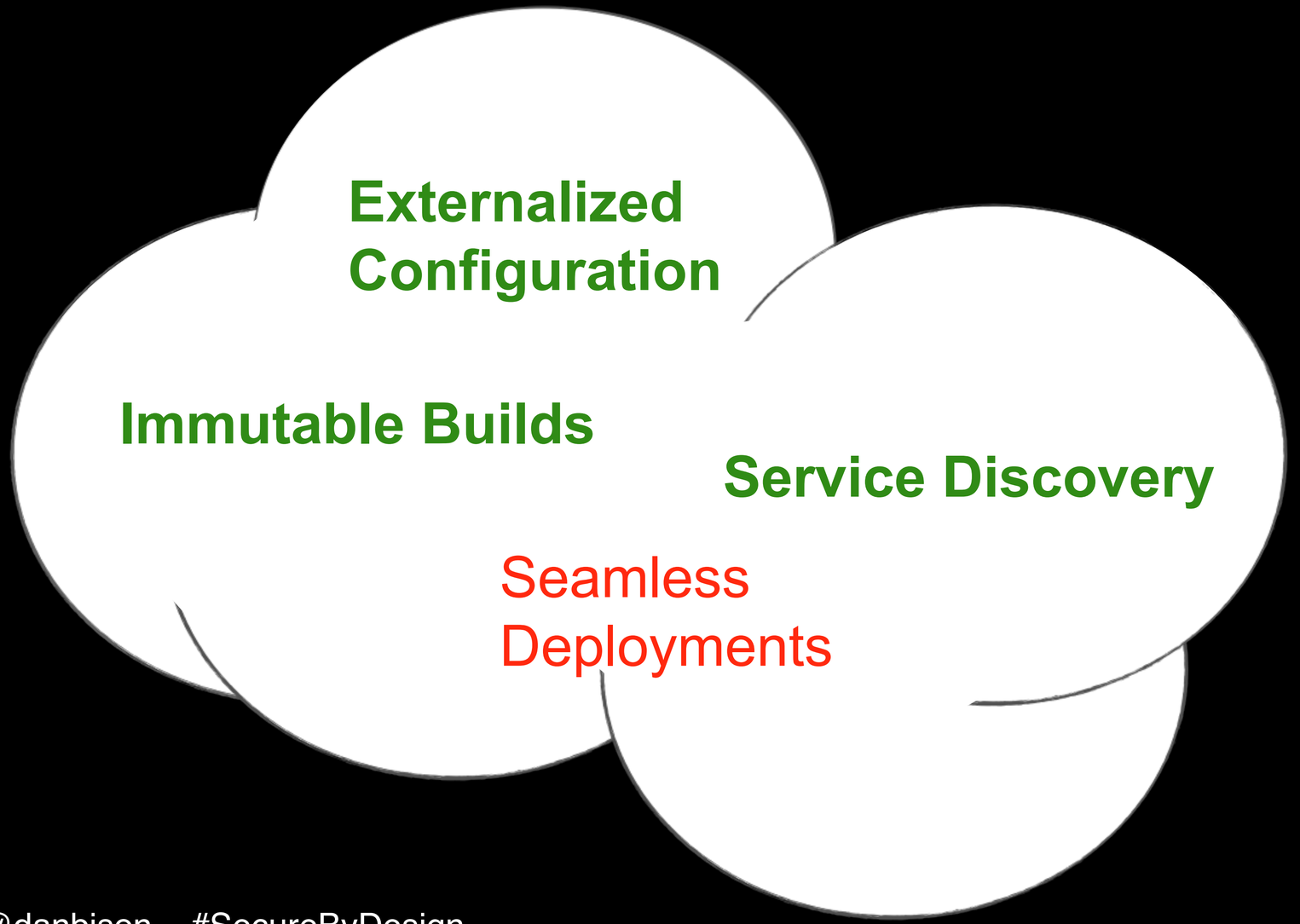


Service Discovery using a Registry



Can this be done in a non-cloud environment?

Patterns used in the Cloud



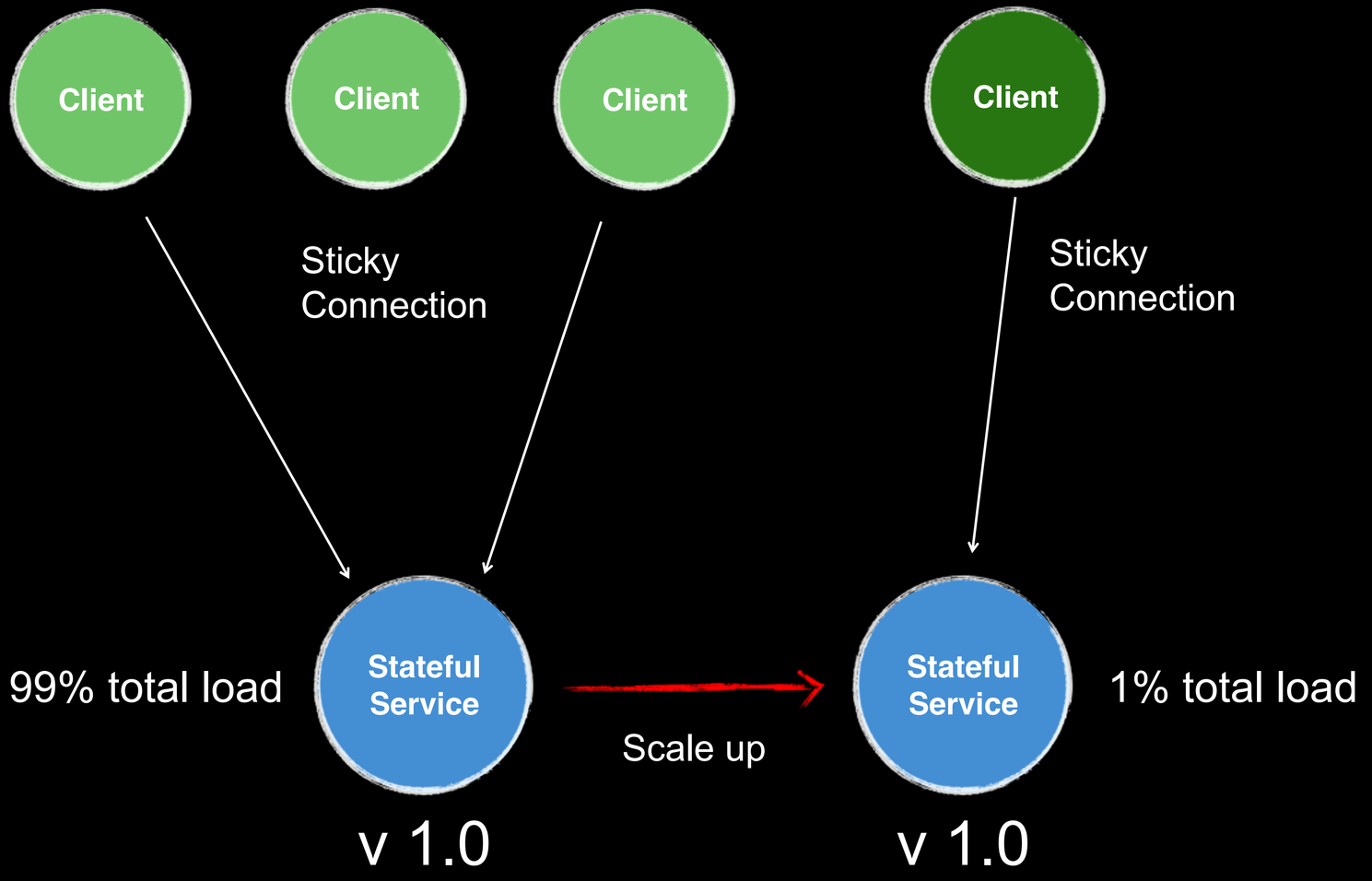
Seamless Deployments

- No downtime or interruptions when deploying a new instance of an application into the ecosystem.
- Enables elasticity, scalability, and great flexibility.
- Challenges: stateful applications and API changes

Paul Bica https://flic.kr/p/7iDwbf

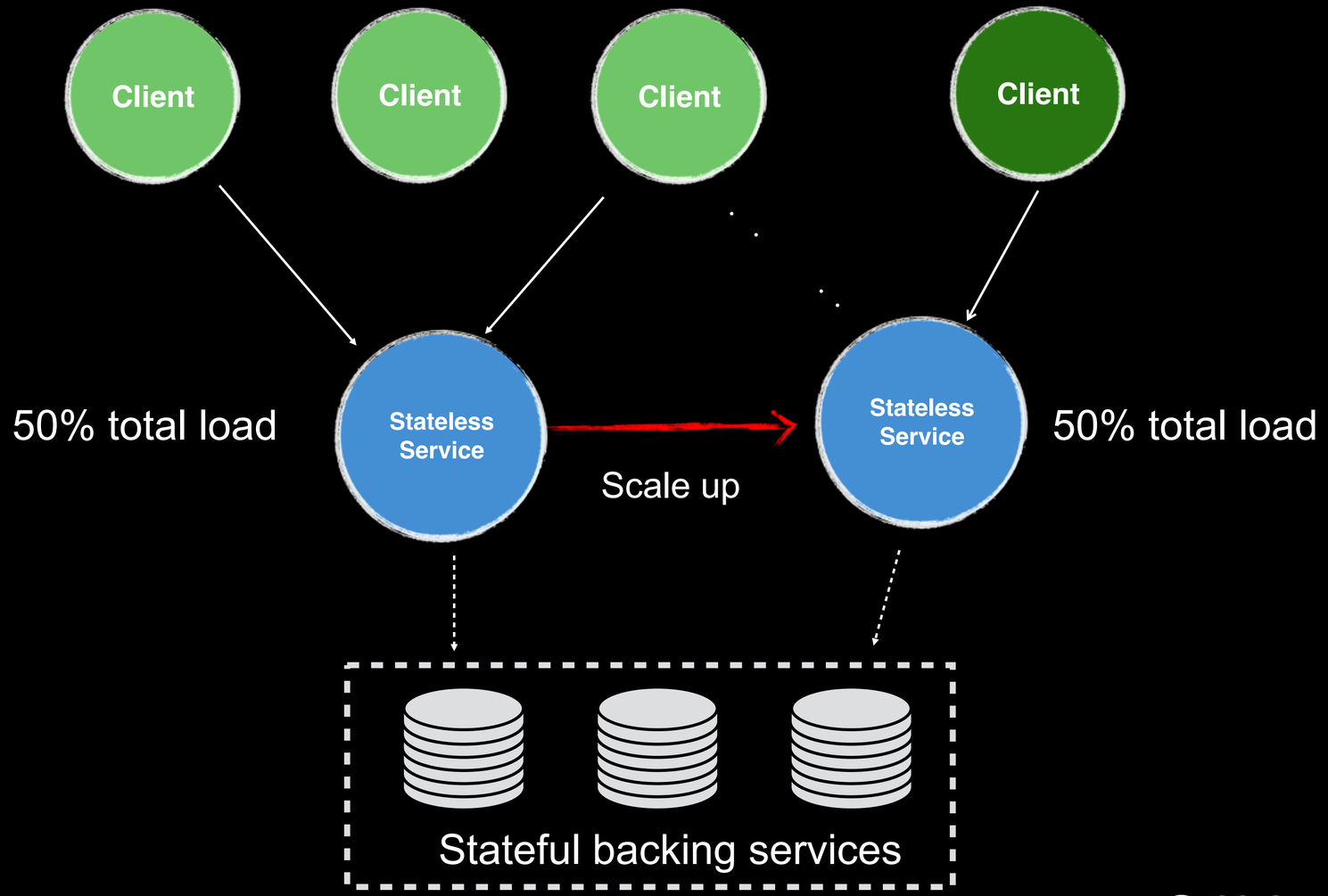
Stateful Services: Non-elastic Scaling

- Deployments can be made without downtime
- Clients must have sticky connections due to service state
- Unbalanced system load
- But what if we move state into backing services?

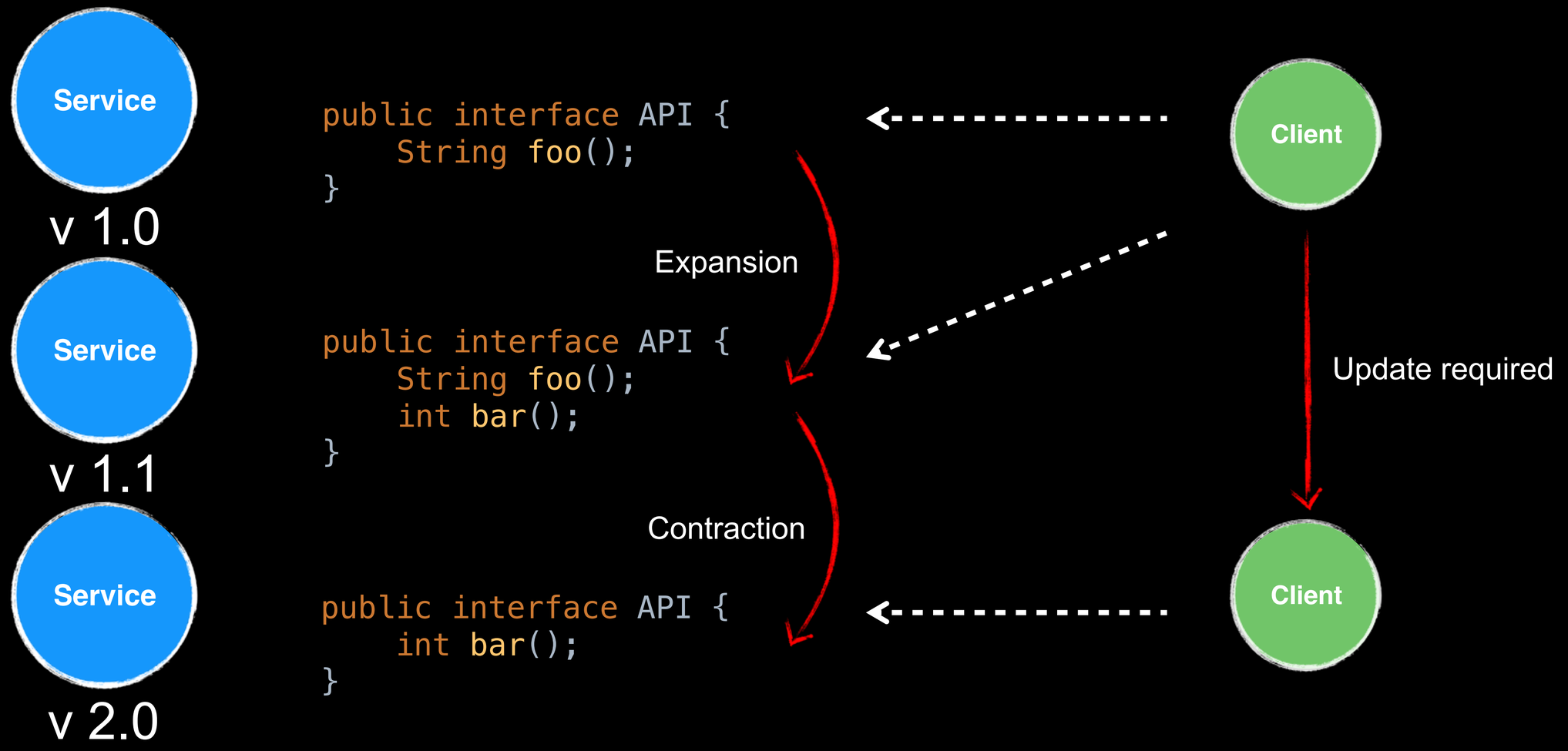


Stateless Services: Elastic Scaling

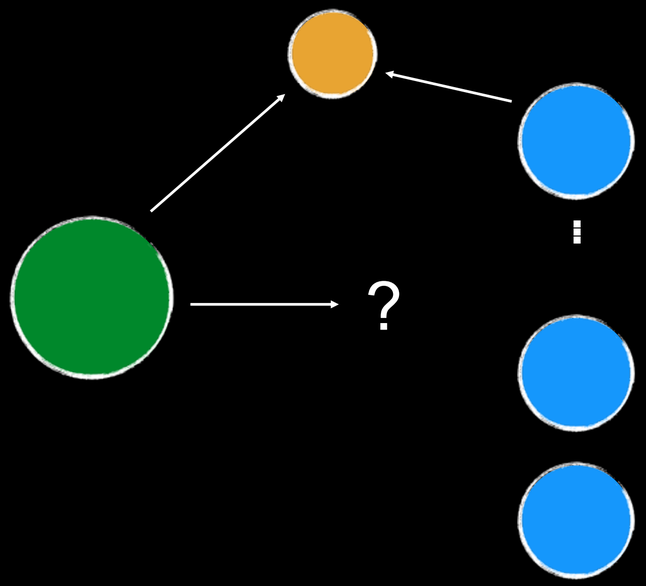
- Clients can use any instance
- Easier to balance system load
- But what about API changes?

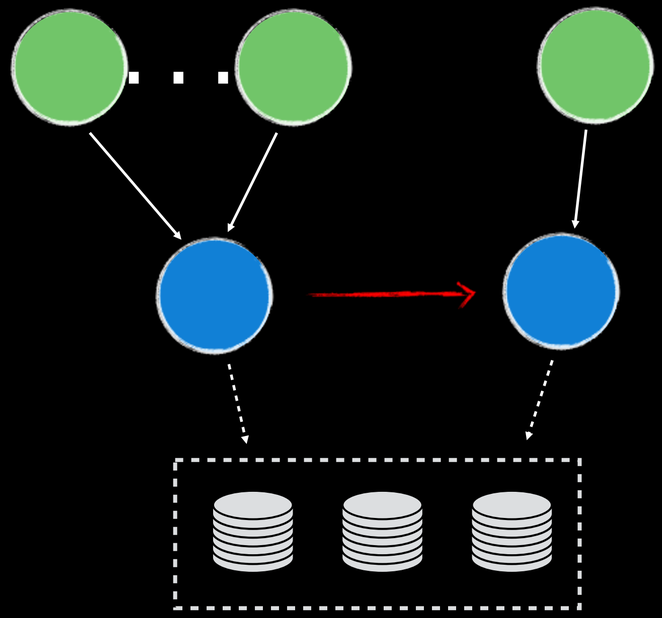


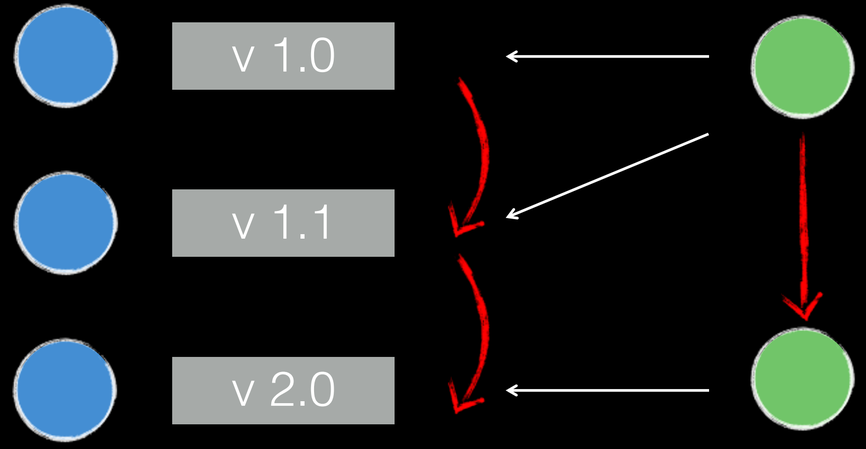
Expand - Contract APIs



Elastic Deployment without Downtime







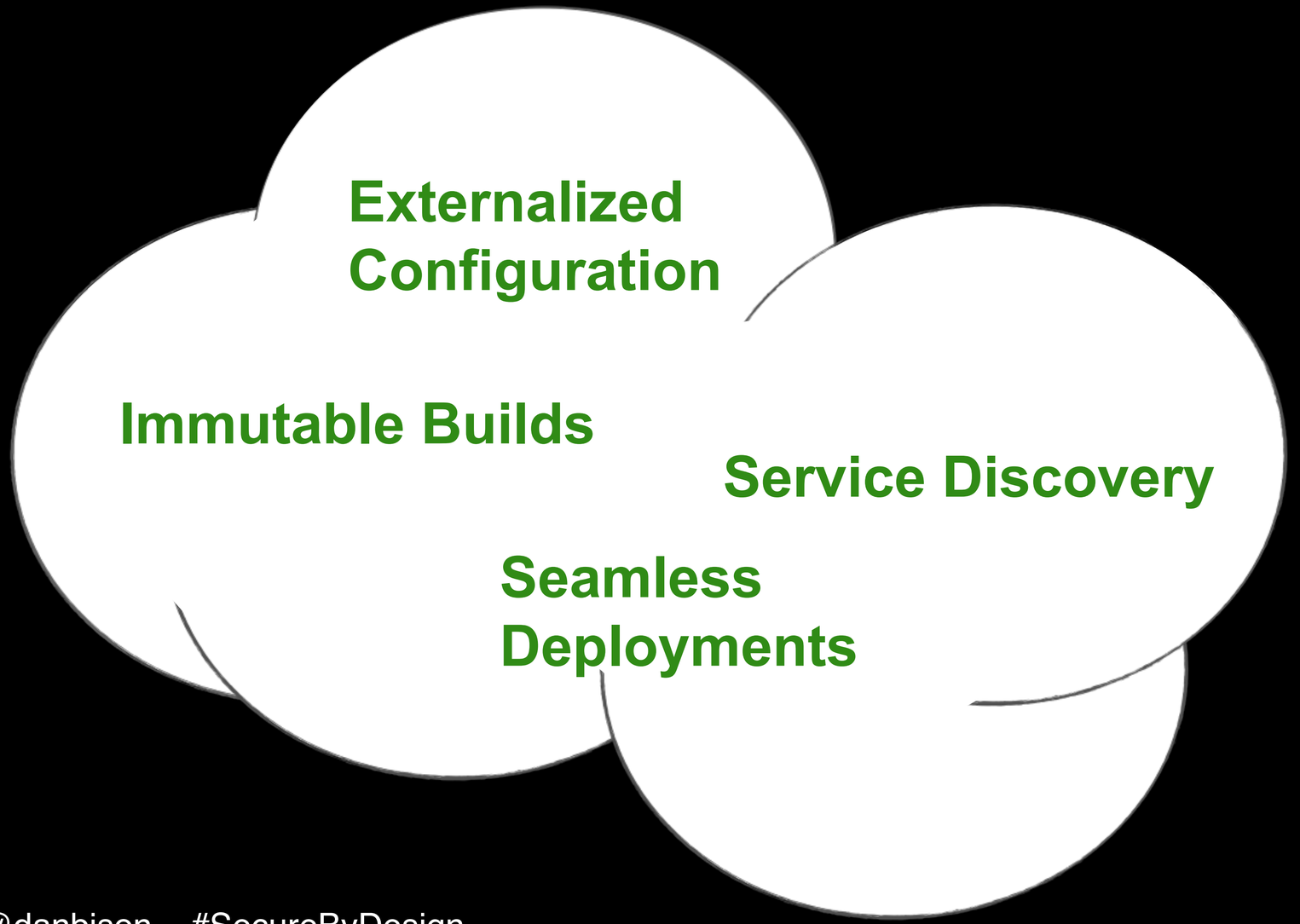
Service Discovery

Stateless Services

Expand / Contract APIs

Can this be done in a non-cloud environment?

Patterns used in the Cloud



What have we learned so far?

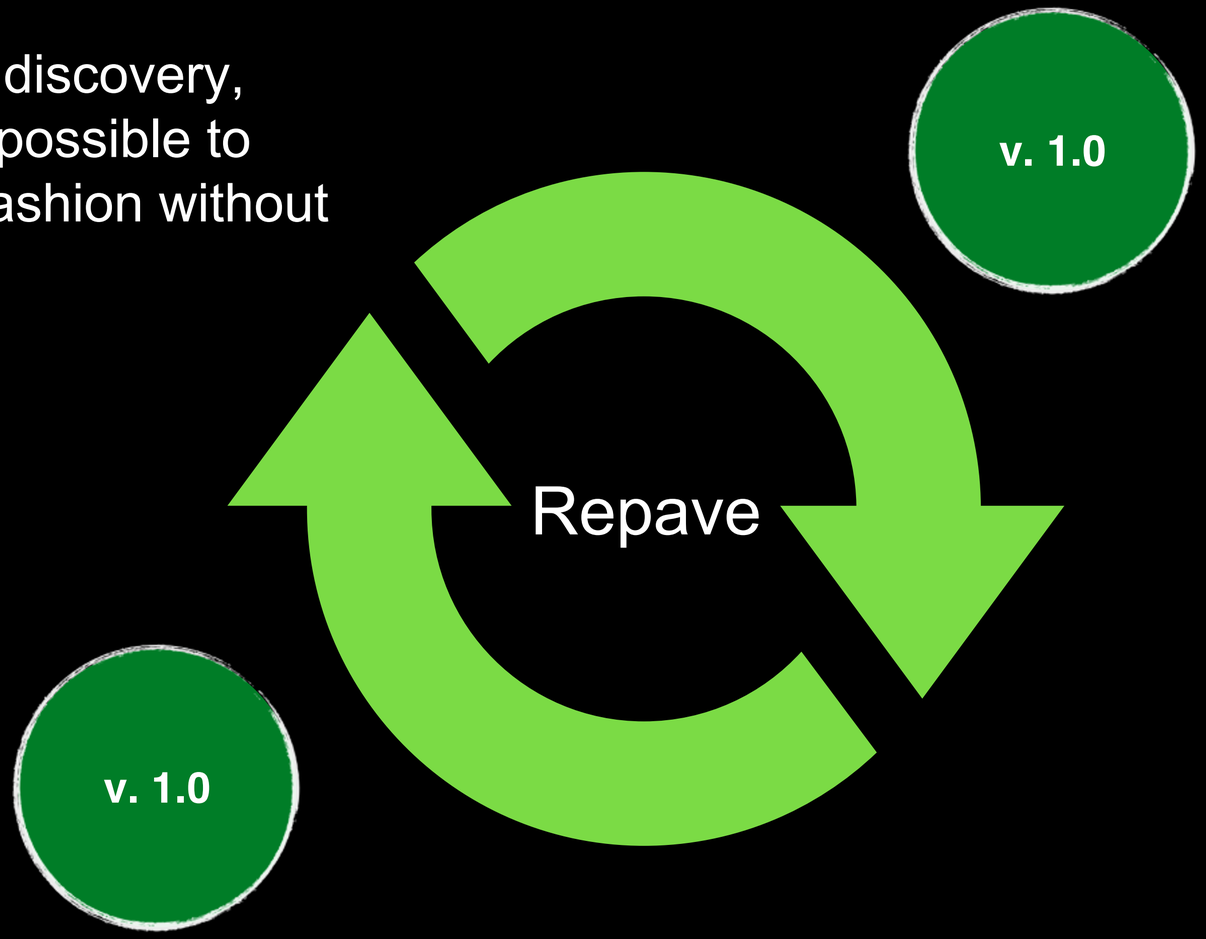
- Externalized configuration
  - makes your application environment agnostic
- Immutable builds
  - build once run anywhere
- Service discovery
  - decouples your application
- Seamless deployments
  - enables zero downtime

Richard Patterson https://flic.kr/p/24GcRZQ

But what about APT?

Automatic Application Repavement

- Combining immutable builds, service discovery, and seamless deployments makes it possible to repave applications in an automatic fashion without downtime.
- But how does this address APT?

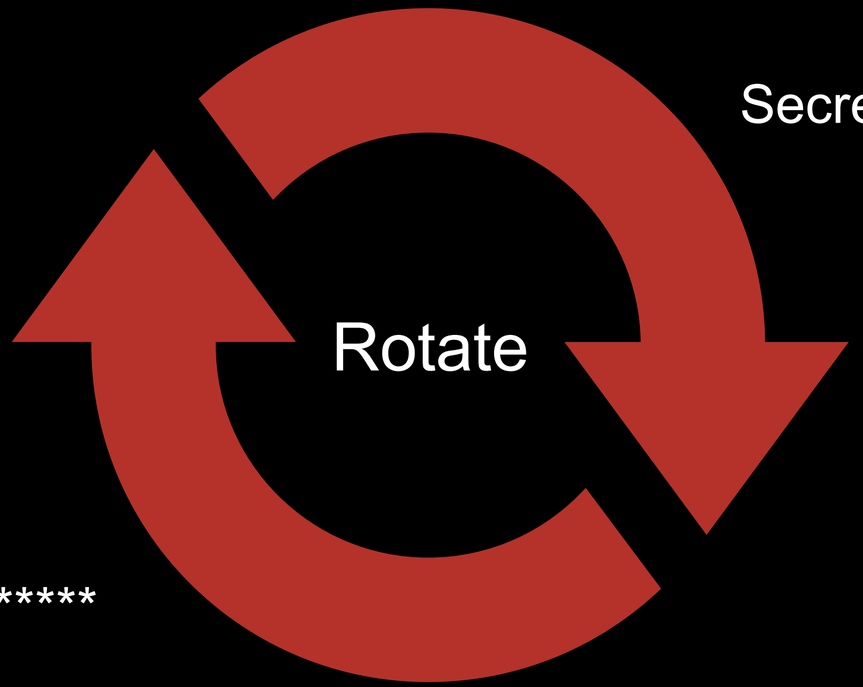


Automatic Rotation of Secrets

- By externalizing configuration, secrets can be placed in a vault that allows automatic rotation of secrets.
- Note: placing secrets in env variables can be dangerous. For example, in most Linux systems, it's possible to inspect environment variables using

```
cat /proc/$PID/environ
```

- But how does this address APT?



Automatic Patching (Repair)

- By combining build pipelines (with automated tests), immutable builds, service discovery, and seamless deployments, it becomes possible to continuously patch the OS and verify that it works with the application(s).
- An updated OS image can then be rolled out during the next repavement phase or on a scheduled basis.
- But how does this address APT?

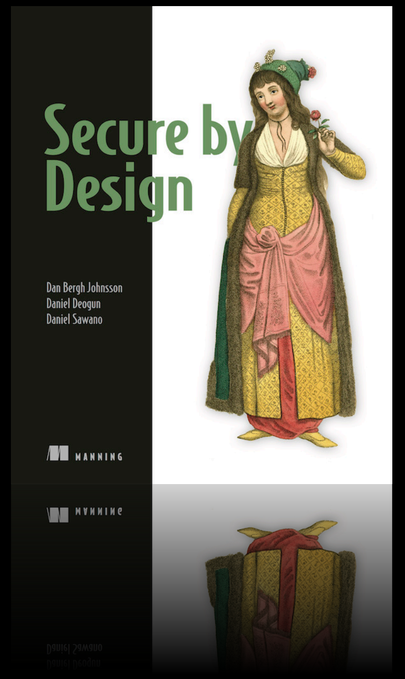
The R's of Enterprise Security

Rotate Repave Repair

Ref: Justin Smith, Pivotal @justinjsmith

Take-aways

- By designing for change, you reduce the risk of Advanced Persistent Threats.
- Several design patterns used in the cloud can be used in a non-cloud environment.



Thanks

 @DanielDeogun @danbjson #SecureByDesign

omega point.