Spring Cloud Config

Pivotal

Spring Config Context

Configuration for Spring can generally be described as values which wire up Spring Beans

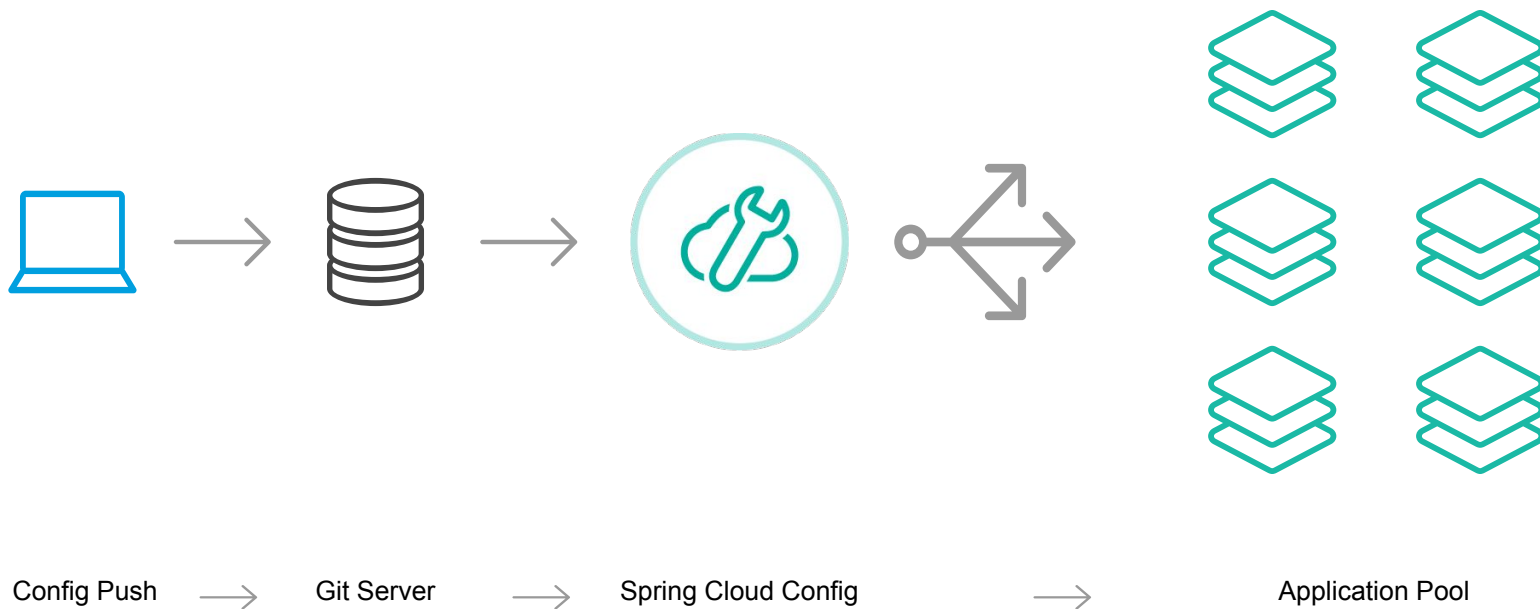- Externalizing Configurations
    - Command Line
    - Environment Variables
    - Spring Config Server
- Challenges
    - Changes Require Restarts
    - No Auditing

HTTP-based API

Asymmetric & Symmetric Encryption Options

Spring Boot Embeddable

Pivotal.

Configuration Workflow

Config Push → Git Server → Spring Cloud Config → Application Pool

Pivotal

Spring Config Server

```java
@Configuration
@EnableAutoConfiguration
@RestController
public class Application {

  @Value("${config.name}")
  String name = "World";

  @RequestMapping("/")
  public String home() {
    return "Hello " + name;
  }

  public static void main(String[] args) {
    SpringApplication.run(Application.class, args);
  }
}
```

application.yml

```yaml
spring:
  cloud:
    config:
      server:
        git:
          uri: <repo>
```

demo.yml

```yaml
name: "mike"
```
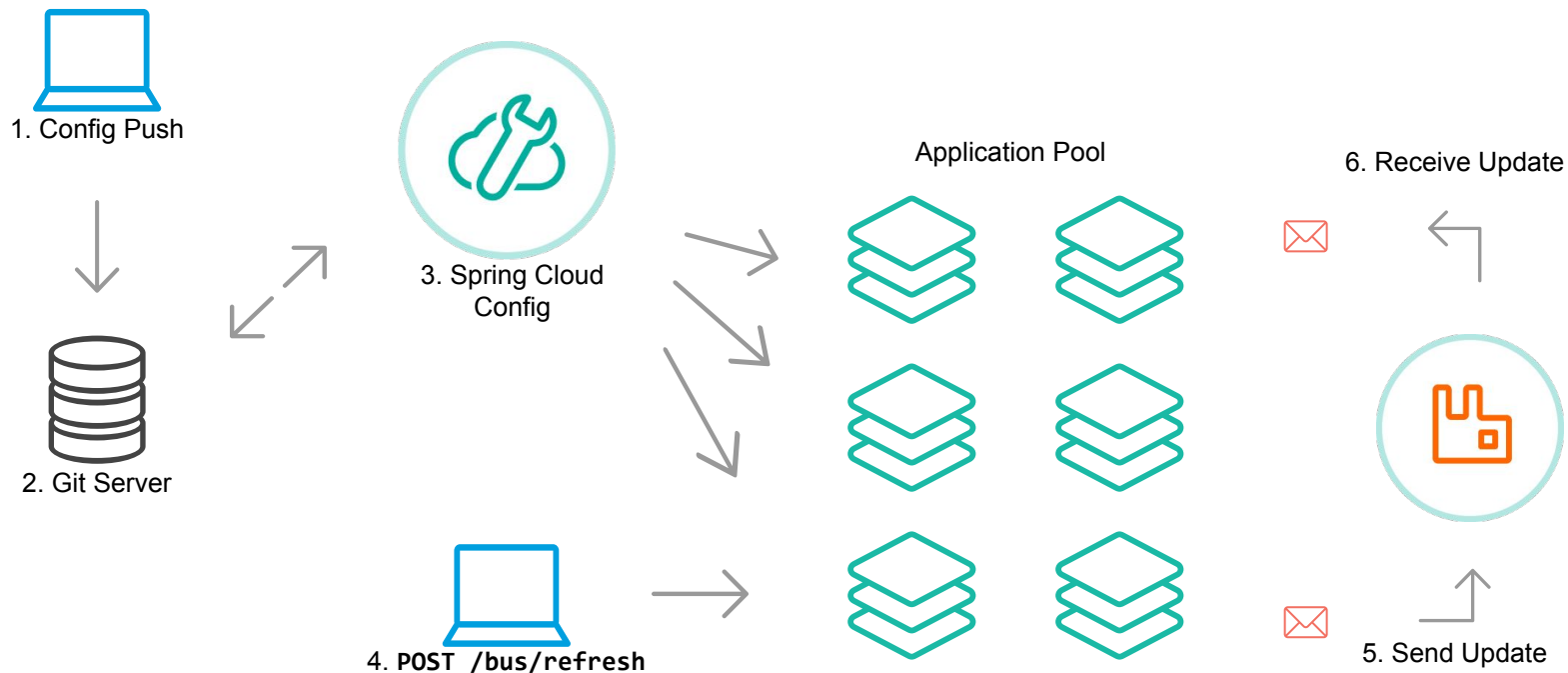
Pivotal.

Refreshing Context

Publisher -> Subscriber

- Update all applications easily

- Spring Cloud Bus

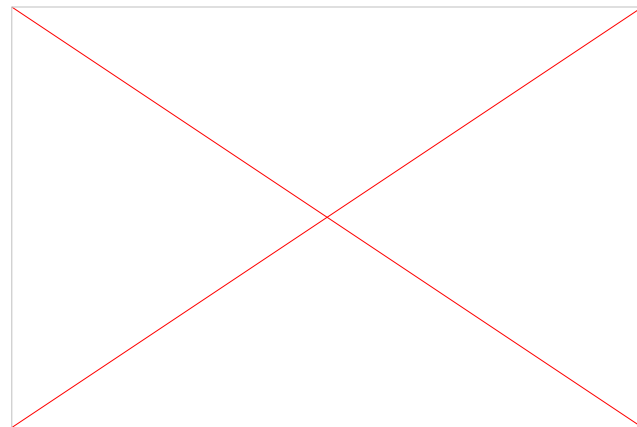- Easy as **POST**
    - One Command
    - Update Everywhere

Pivotal

Spring Cloud Bus Workflow

1. Config Push

2. Git Server

3. Spring Cloud Config

4. **POST** `/bus/refresh`

Application Pool

5. Send Update

6. Receive Update

Pivotal

Spring Cloud Services: Config Server

Cloud Foundry Marketplace

- Automated Config Server Deployment

- Specify Git URL

- Bind onto Cloud Foundry App

- *Optionally* bind in RabbitMQ for Cloud Bus

Lab: Spring Cloud Config