steeltoe
by Pivotal

a toolkit for building cloud-native .NET microservices

Steeltoe, Cloud-native .NET and Microservices

Pre Bhakta & John Bush
Sr. Platform Architects

Pivotal

Outcomes

Open discussion

How should I really use the cloud?

How can Steeltoe help?

How should I really use the cloud?

> “the Cloud isn’t about **where** computing is done, rather ***how*** it’s done.
>
> – Paul Maritz

Cloud-Native is how

12 Factor App

1. Codebase
2. Dependencies
3. Config
4. Backing services
5. Build, release, run
6. Processes
7. Port binding
8. Concurrency
9. Disposability
10. Dev/prod parity
11. Logs
12. Admin processes

What is Cloud-Native .NET?

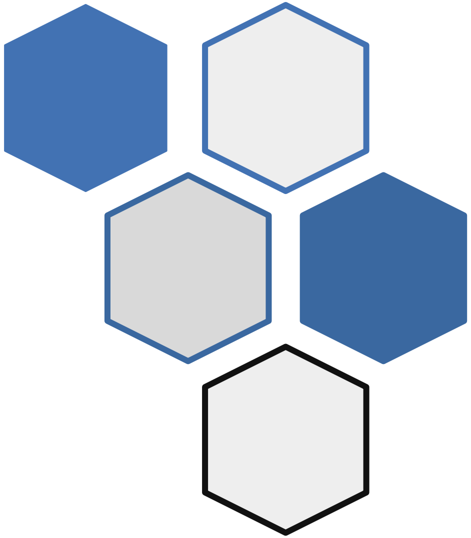
These principles may seem foreign to .NET devs

- Configuration stored in environment variables
- Out of process session state
- No r/w to registry, GAC, local file system
- Loosely coupled backing services (i.e. RDBMS)

Microservices

Microservice architectures have many benefits

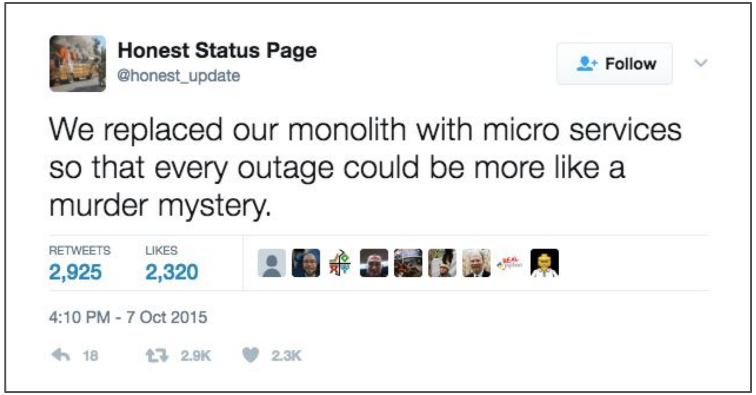
- Independently scalable components
- Quickly iterate and release frequently
- Easy for new devs to join and be productive
- Increased developer velocity
- Shorter test cycles
- Polyglot becomes reality



But Microservices are hard

Microservices = Distributed Systems = Way more complexity

- How to troubleshoot microservices?
- How to set global configuration across the whole application?
- How to look up the dynamic addresses of services you consume?
- What do you do when a service you depend on stops responding?



Honest Status Page
@honest_update

We replaced our monolith with micro services so that every outage could be more like a murder mystery.

RETWEETS 2,925 LIKES 2,320

4:10 PM - 7 Oct 2015

The good news is...

There are some great OSS solutions to these problems available from

NETFLIX OSS and spring

But until now they've been focused on Java and not very accessible to .NET devs…

So ***that's*** why we built Steeltoe

To help *you* build

Cloud native .NET applications

that can leverage Spring Cloud tooling for

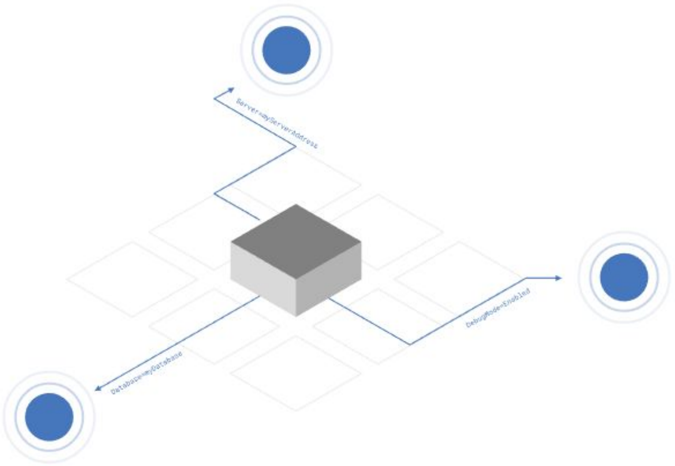
resilient microservices

How can Steeltoe help?

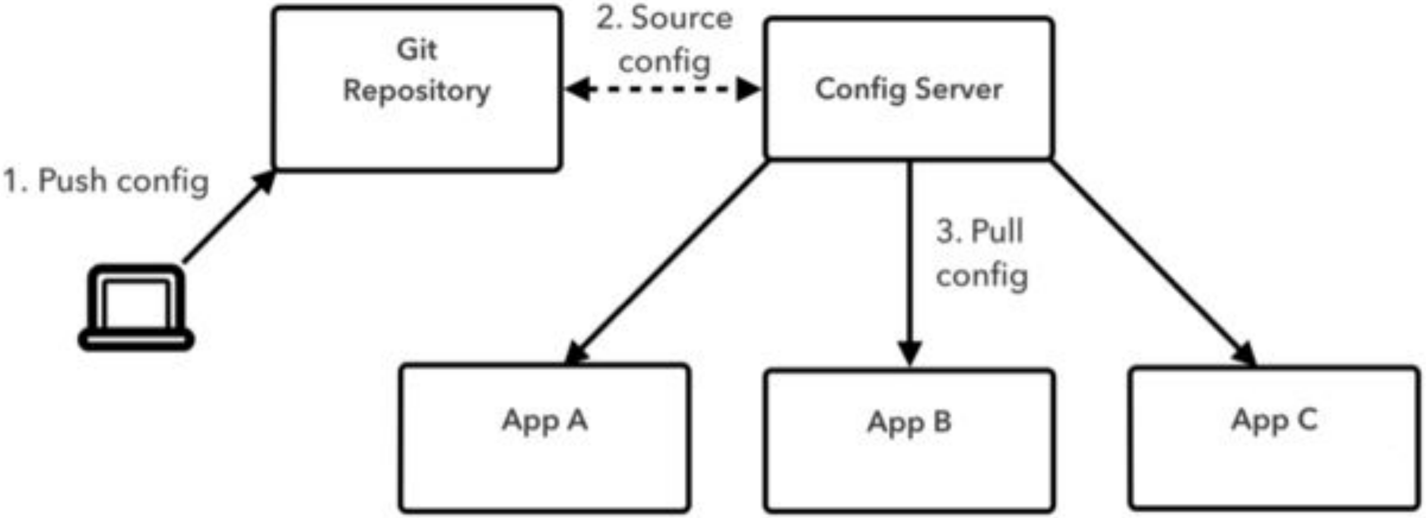
Configuration Providers

Config Server

- Access config stored in Spring Cloud Config Server (backed by Git, Vault, local filesystem)
- Across all instances, all apps, all environments



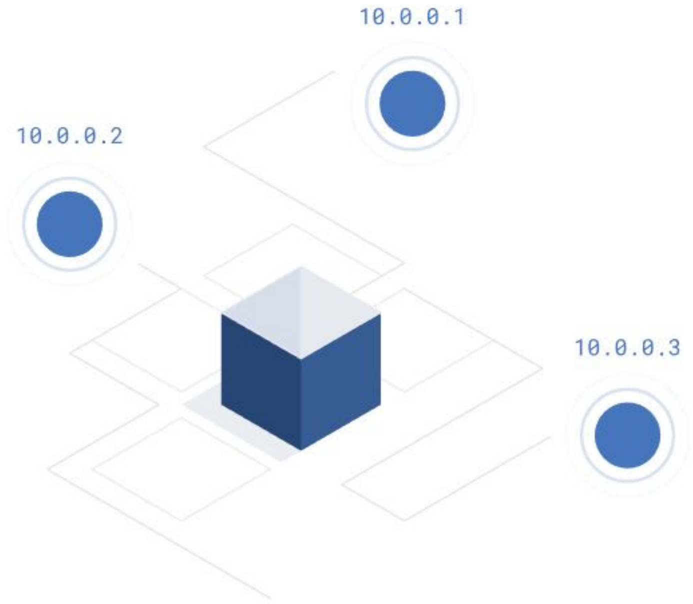
About Spring Cloud Config Server



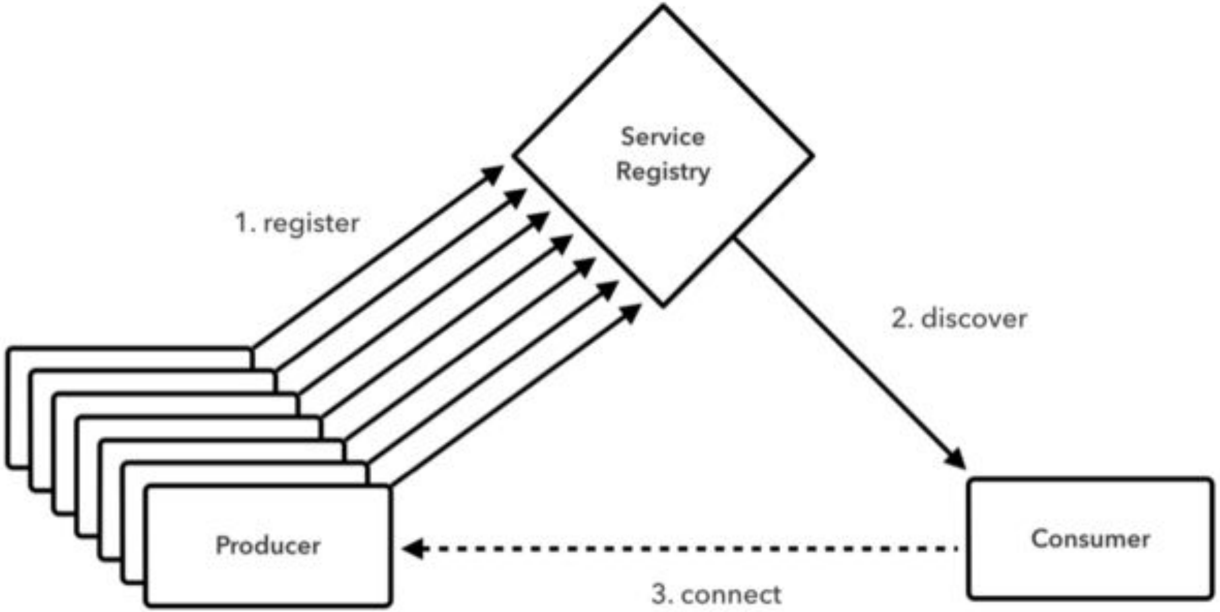
Service Discovery

Service Discovery Client

- .NET client for Netflix Eureka
- Implements Service Discovery design pattern
- Dynamically discover and call registered services



About Eureka Server

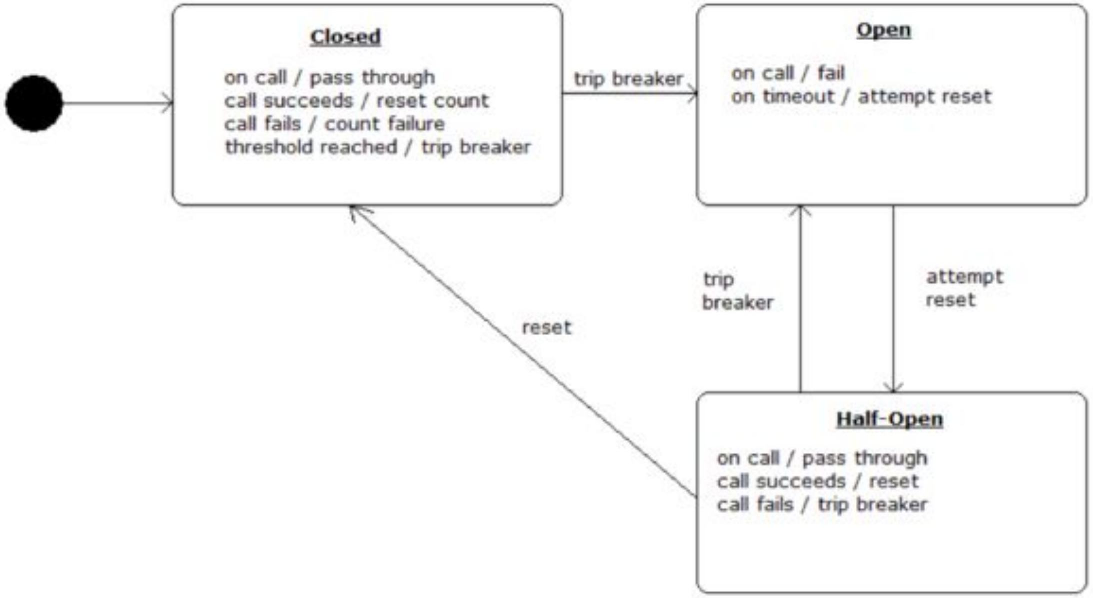


Circuit Breaker

Circuit Breaker Client

- .NET implementation of Netflix Hystrix
- Bypass failing services with elegant fall-back behavior (so your users don’t see nasty error messages)
- Rich metrics and monitoring

About Hystrix



Resources

How to get started

The best way to get started is to take a look at our docs, clone a sample from the GitHub repo, and contact the team in Slack with questions and feedback.

- Learn more → http://steeltoe.io
- Check out the samples → https://github.com/steeltoeoss
- Get the bits → https://www.nuget.org/profiles/steeltoe
- Talk to us → http://slack.steeltoe.io/
- Tell your friends → @SteeltoeOSS

Other Resources

12 Factor App

https://12factor.net

Polyglot microservices and Eureka

https://seroter.wordpress.com/2017/03/27/yes-you-can-use-a-single-service-registry-for-net-and-java-microservices/

Microservices, .NET, Cloud Foundry and Microsoft's Face API

https://www.altoros.com/blog/microservices-with-steeltoe-and-cloud-foundry-a-dotnet-app-using-microsoft-face-api/

.NET Core Microservices and Steeltoe

https://www.altoros.com/blog/enabling-dotnet-core-microservices-with-steeltoe-and-pivotal-cloud-foundry/

Contact Us

https://www.linkedin.com/in/prebhakta & https://www.linkedin.com/in/jbush

Demo time!