

Aprendizados na criação de uma arquitetura de dados em Go





Olá!



Kamila Hinckel

Desenvolvedora na Neoway

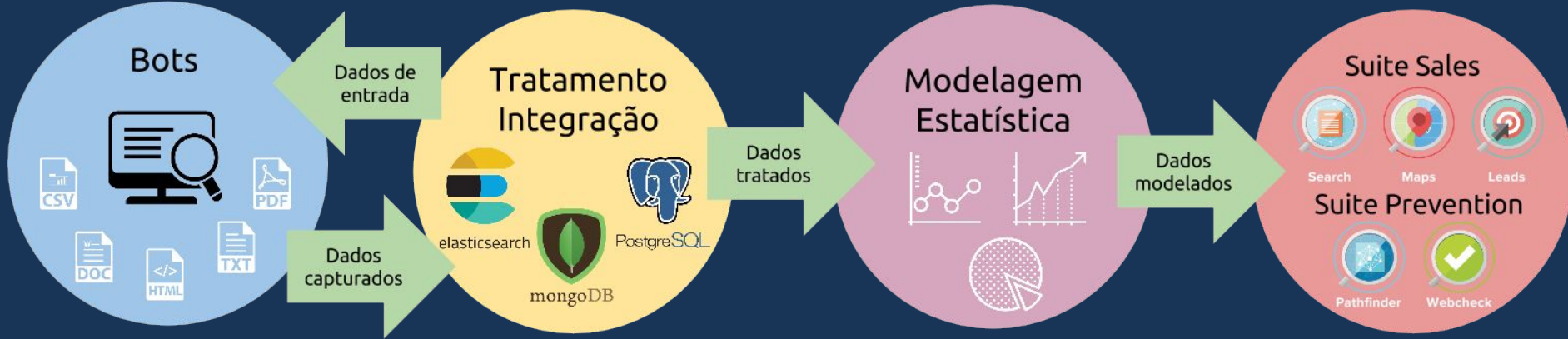
Matheus Vill

Desenvolvedor na Neoway



Neoway
BUSINESS SOLUTIONS

O que a NEOWAY faz?



Data Science

O que fazemos?



Scheduling de inputs para os BOTs

EU PROCURO

Por exemplo um restaurante, um bar, uma loja...

EM

Um endereço específico, bairro ou cidade.



O que fazemos?



Serviços de tratamento e enriquecimento de dados

Exemplos:

Apresentação Neoway -> APRESENTACAO NEOWAY

FULANO DE TAL -> 111.111.111-11

O que fazemos?



Integração e disponibilização de dados

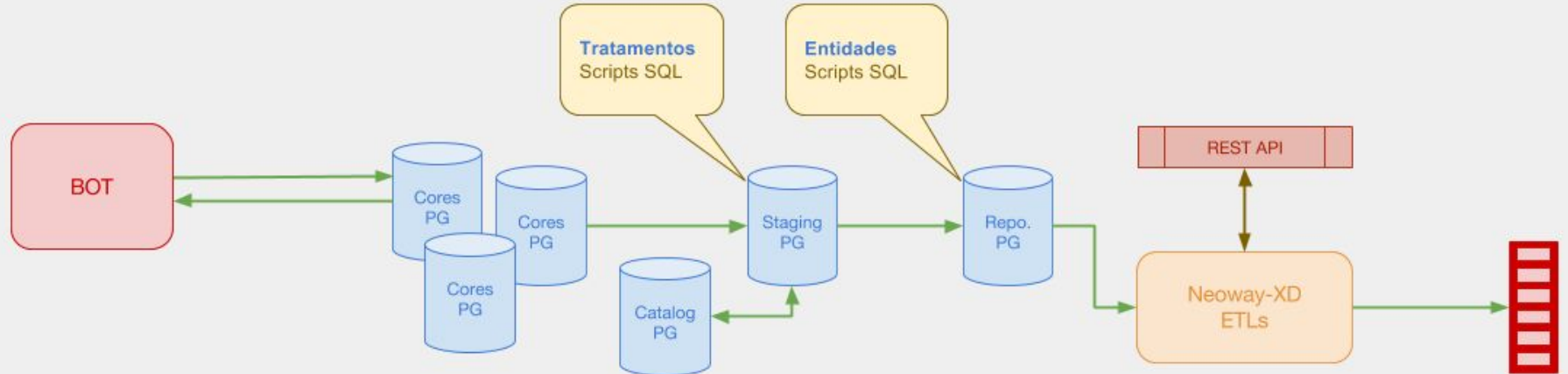
Exemplo:

Informações das empresas da Receita Federal

+

Informações dos sócios das Juntas Comerciais

Arquitetura 1.0



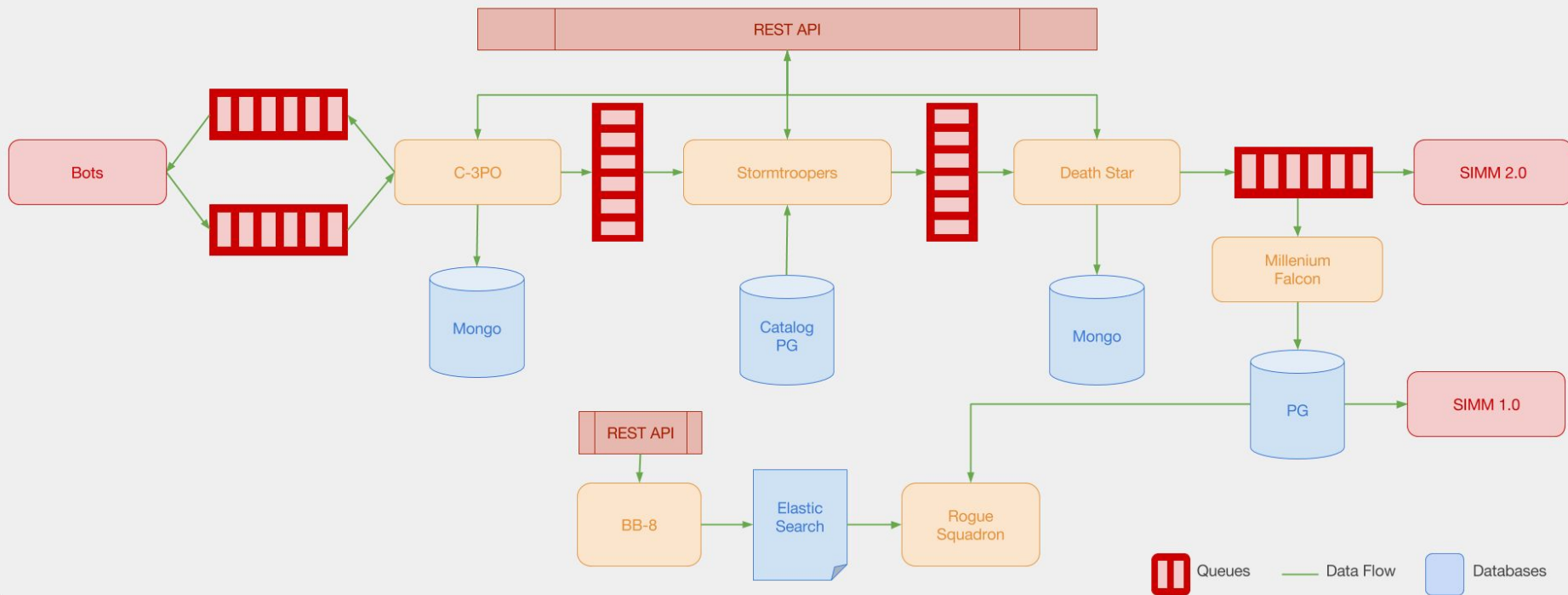
Problemas

- Custo elevado e dificuldades para escalar;
- Atraso na disponibilização dos dados para as aplicações;
- Dificuldades para simular o ambiente existente;
- Dificuldade de mapear os impactos de alterações.

Nova arquitetura!



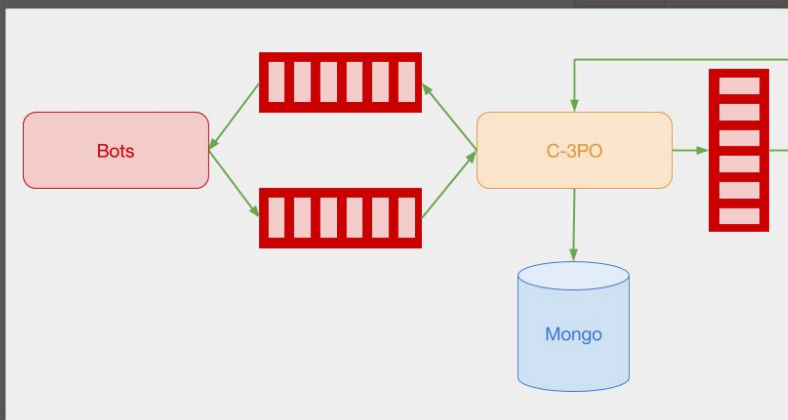
Nova Arquitetura



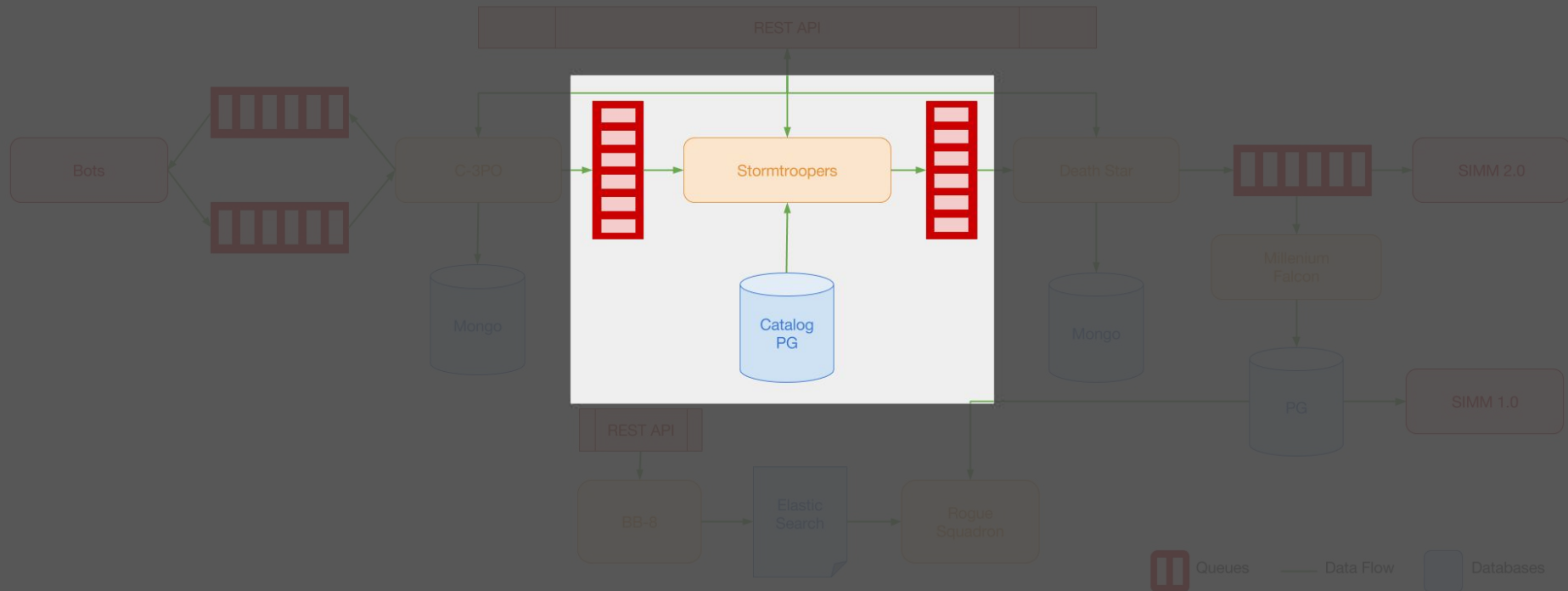
Go, Rabbit e Mongo



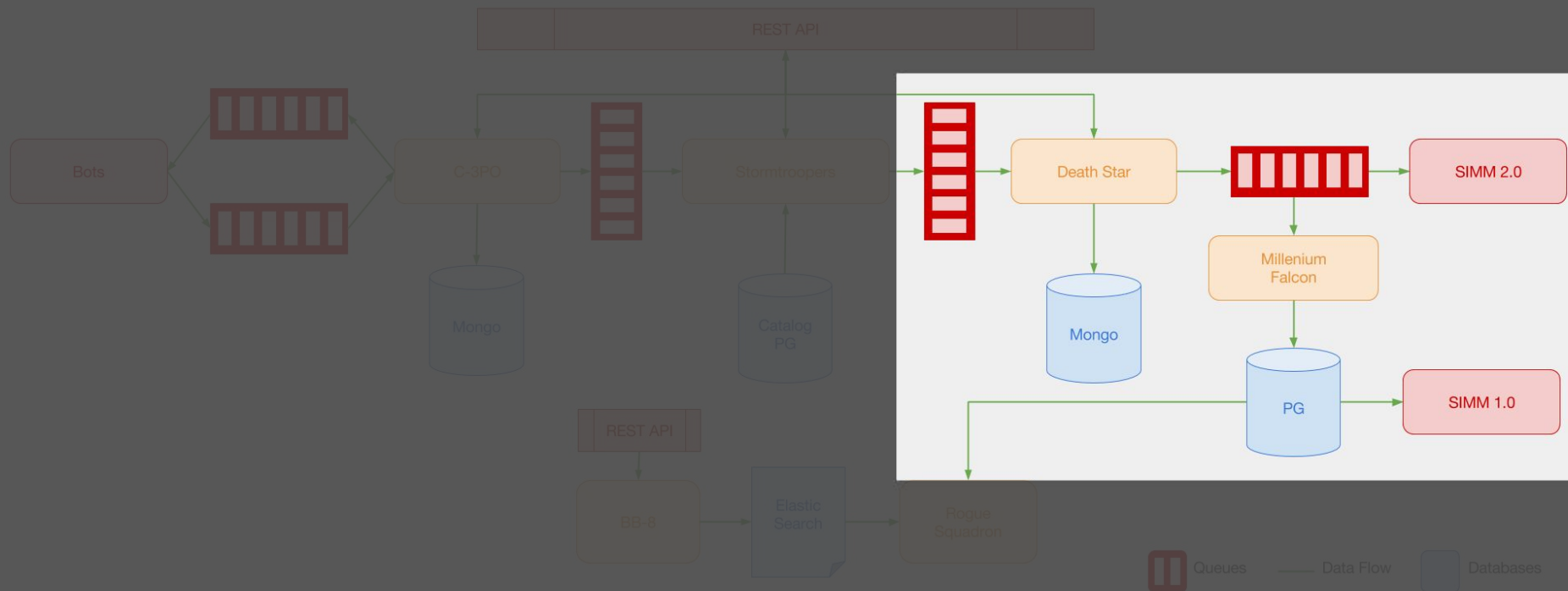
Scheduling de Inputs



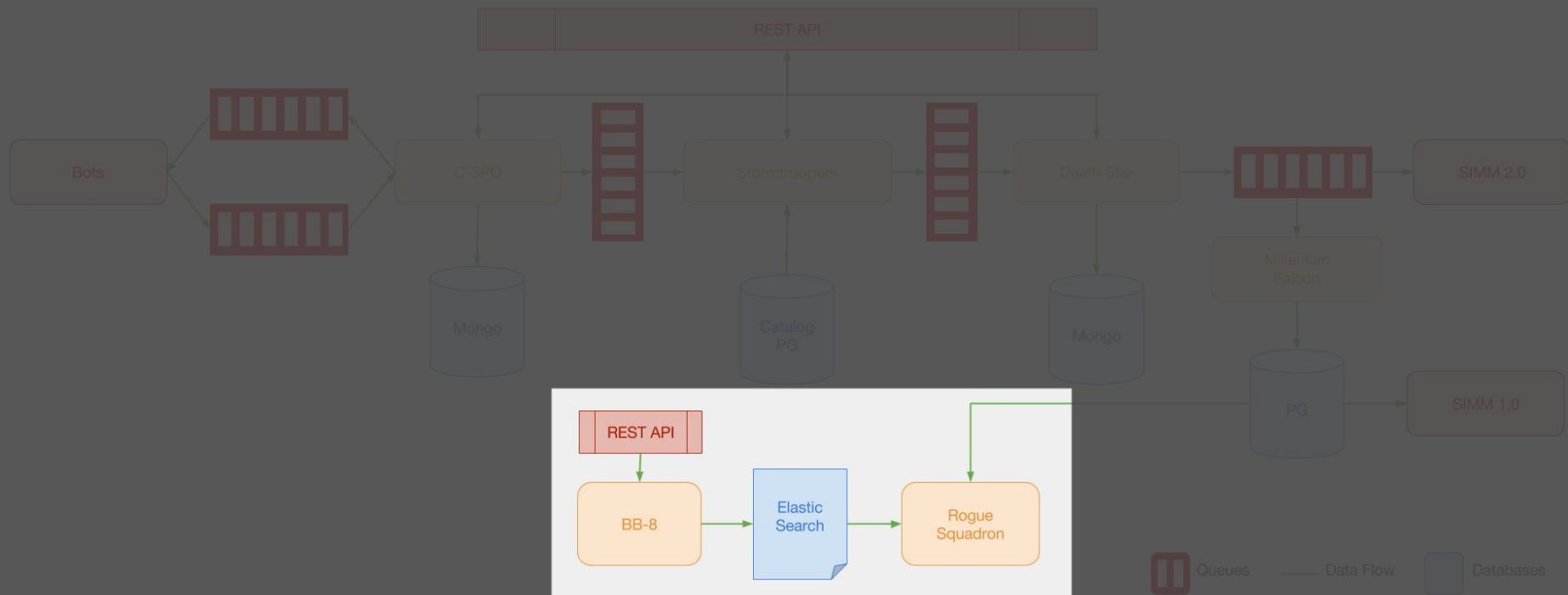
Tratamento e enriquecimento



Integração e disponibilização



Disponibilização do histórico capturado



O que ganhamos?



Tempo: Disponibilização dos dados de forma contínua e integrada entre todas as aplicações.



Escalabilidade: Necessidade de **escalar horizontalmente** devido ao aumento do número de fontes capturadas.



Infraestrutura imutável: Scripts automatizados, evitando máquinas “flocos de neve”.



Qualidade: Arquitetura orientada à testes automatizados.



Custo: Possibilidade de redução de custos de infra.



TESTING

I FIND YOUR LACK OF TESTS DISTURBING.

Testes Unitários

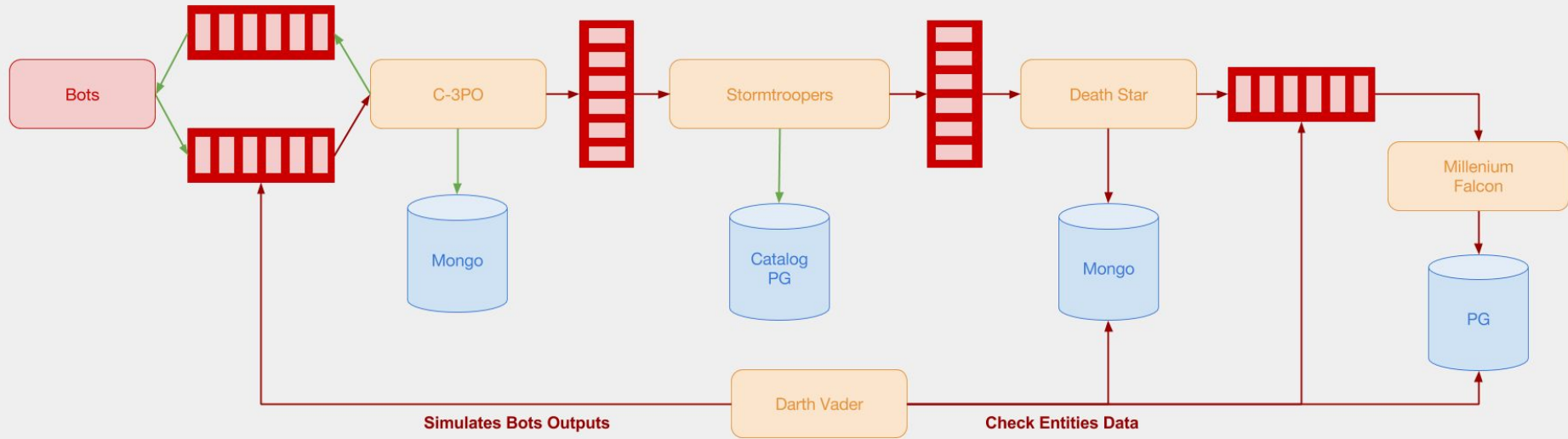
Vantagens:

- Executam em poucos segundos
- Segurança para implementar novas features
- Teste de erros e exceções

Desvantagens:

- Alguns problemas com Mongo e Rabbit não eram percebidos e só foram descobertos nos primeiros deploys

Testes end-to-end



Testes end-to-end

Vantagens:

- Garante que todos os serviços funcionem de forma integrada
- Ideia futura de integrar os bots nesses testes também

Desvantagens:

- Mais lento
- Mais difícil de encontrar onde está o problema

Testes de integração

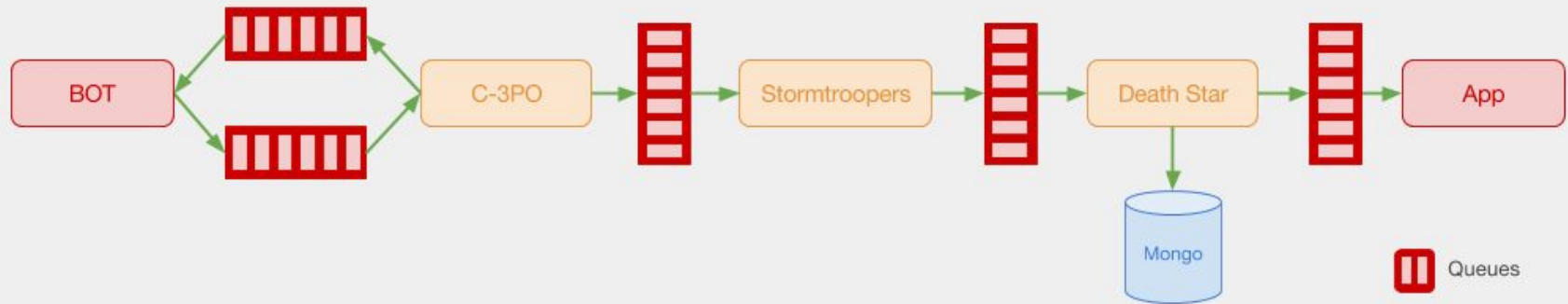
Vantagens:

- Necessidade de ter uma meio termo entre os testes unitários e os testes end-to-end
- Permitiram testes integrados com o Mongo e o Rabbit, porém com cases específicos de cada serviço

Desvantagens:

- Setup desses testes são mais complexos que os testes unitários
- Mais difícil garantir o isolamento dos testes

Problemas com Performance



Benchmark

| | | |
|--|----------|--------------|
| BenchmarkTextCleansingDomain-4 | 3000 | 470369 ns/op |
| BenchmarkReplaceSpacesAndAmpersand-4 | 10000000 | 185 ns/op |
| BenchmarkReplaceOrdinalSymbols-4 | 3000 | 424179 ns/op |
| BenchmarkReplaceDoubleSpace-4 | 50000000 | 24.8 ns/op |
| BenchmarkRemoveSpaceBetweenDocumentNumbers-4 | 50000 | 33780 ns/op |
| BenchmarkRemoveAccent-4 | 500000 | 3291 ns/op |

Função lenta

```
1
2 func ReplaceOrdinalSymbols(text string) string {
3     cleanText := ""
4     for _, character := range text {
5         expression, _ := regexp.Compile("[:alnum:]&ÀÁÂÃÄÅÇÈÉÊËÌÍÎÏÑÒÓÔÕÖÙÚÛÜÝàáâãäåçèéêëìíîïñòóôõöùúûüýÿ ")
6
7         if expression.MatchString(string(character)) {
8             cleanText = cleanText + string(character)
9         } else if string(character) == ` ` {
10             cleanText = cleanText + "0"
11         } else if string(character) == `a` {
12             cleanText = cleanText + "A"
13         } else {
14             cleanText = cleanText + " "
15         }
16     }
17     for strings.Contains(cleanText, " ") {
18         cleanText = strings.Replace(cleanText, " ", "", -1)
19     }
20     return cleanText
21 }
22
```

Benchmark

```
1  
2 func BenchmarkReplaceOrdinalSymbols(b *testing.B) {  
3     t := "adasd -- 21331231 - sads"  
4     for n := 0; n < b.N; n++ {  
5         text.ReplaceOrdinalSymbols(t)  
6     }  
7 }  
8
```

Modificação

```
1
2 func ReplaceOrdinalSymbols(text string) string {
3     text = strings.Replace(text, "0", "0", -1)
4     text = strings.Replace(text, "a", "A", -1)
5
6     re, _ := regexp.Compile("[^[:alnum:]]&ÀÁÂÃÄÅÇÈÉÊËÌÍÎÏÑÒÓÔÕÖÙÚÛÜÝàáâãäåçèéêëìíîïñòóôõöùúûüýÿ ]")
7     text = re.ReplaceAllString(text, " ")
8
9     return text
10 }
11
```

Benchmark

Antes:

| | | |
|--|----------|--------------|
| BenchmarkTextCleansingDomain-4 | 3000 | 470369 ns/op |
| BenchmarkReplaceSpacesAndAmpersand-4 | 10000000 | 185 ns/op |
| BenchmarkReplaceOrdinalSymbols-4 | 3000 | 424179 ns/op |
| BenchmarkReplaceDoubleSpace-4 | 50000000 | 24.8 ns/op |
| BenchmarkRemoveSpaceBetweenDocumentNumbers-4 | 50000 | 33780 ns/op |
| BenchmarkRemoveAccent-4 | 500000 | 3291 ns/op |

Depois:

| | | |
|--|----------|-------------|
| BenchmarkTextCleansingDomain-4 | 30000 | 58794 ns/op |
| BenchmarkReplaceSpacesAndAmpersand-4 | 10000000 | 185 ns/op |
| BenchmarkReplaceOrdinalSymbols-4 | 100000 | 19988 ns/op |
| BenchmarkReplaceDoubleSpace-4 | 50000000 | 25.4 ns/op |
| BenchmarkRemoveSpaceBetweenDocumentNumbers-4 | 50000 | 33832 ns/op |
| BenchmarkRemoveAccent-4 | 500000 | 3715 ns/op |

Testes de carga

Características:

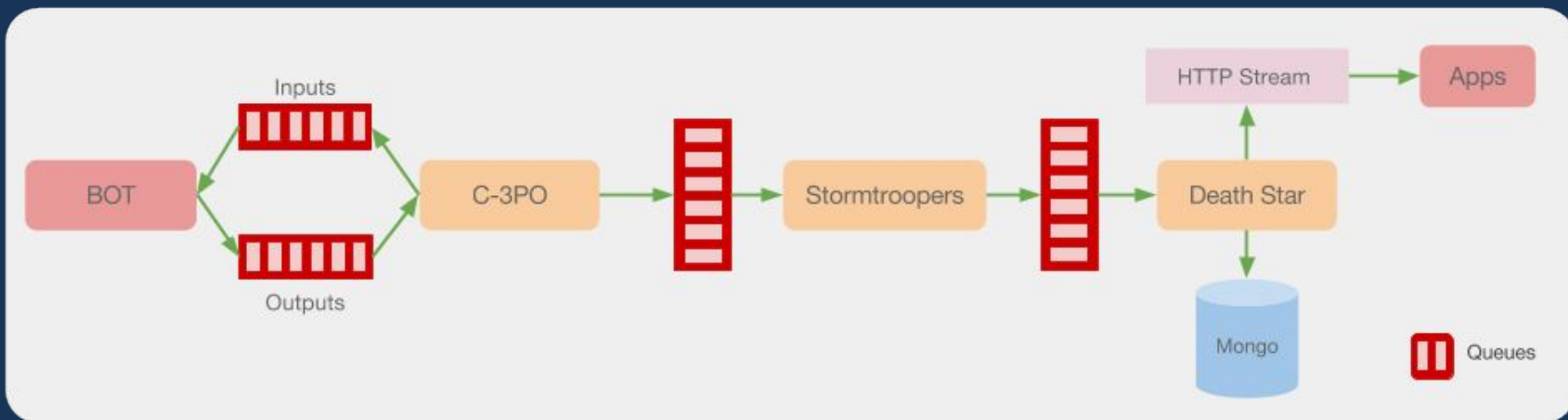
- Possível verificar se novas features não impactaram na performance do stream de dados
- Verificar qual dos microserviços pode estar diminuindo a velocidade do fluxo de dados

Ferramentas:

- rabbit-mq-stress-tester:
<https://github.com/backstop/rabbit-mq-stress-tester>
- boom:
<https://github.com/rakyll/hey>

HTTP Stream

- Dificuldade na integração com outros times;
- Disponibilizar dados em stream para as aplicações;
- HTTP com conexão persistente;



**SHOW ME THE
SOURCE**

YOU MUST

Golang exemplos

- **Neoway**
<https://github.com/NeowayLabs>
- **Terraform**
<https://github.com/hashicorp/terraform>
- **Docker**
<https://github.com/docker/docker>
- **Kubernetes**
<https://github.com/kubernetes/kubernetes>



kubernetes

University of Salzburg
Department of Computer Science

Google Go!

A look behind the scenes

Seminar for Computer Science
Summer 2010

Martin Aigner
Alexander Baumgartner

July 15, 2010

NewStore TechTalk - Advanced Testing with Go by Mitchell Hashimoto



Advanced Testing with Go

With Mitchell Hashimoto

▶ ◀ 🔊 0:03 / 1:17:29



Obrigado!!!

