

GO E MICROSERVIÇOS

NASCIDOS UM PARA O OUTRO

<https://eltonminetto.dev>

MICROSERVIÇOS, O QUÊ É?

[...] um estilo arquitetônico que estrutura uma aplicação como uma coleção de serviços que são:

- altamente manuteníveis e testáveis
 - fracamente acoplados
 - possíveis de serem feitos deploys independentes
- organizados em torno das necessidades de negócios

[...] permite a entrega/deploy contínua de aplicativos grandes e complexos. Também permite que uma organização evolua sua pilha de tecnologias.

O QUE NÃO É?

Uma bala de prata

Monolithic vs Microservices



Monolithic



Microservices

Desculpa para usar todas as tecnologias hipster
do mercado ;)

COMO IMPLEMENTAR?

- Começar do zero com a arquitetura
- Começar como monolito e migrar

Quase todos os casos em que ouvi falar de um sistema que foi construído como um sistema de microserviços do zero, acabaram em sérios problemas.

— Martin Fowler

Quase todas as histórias
de microserviços bem-
sucedidas começaram
como um monolito muito
grande que foi dividido

— Martin Fowler

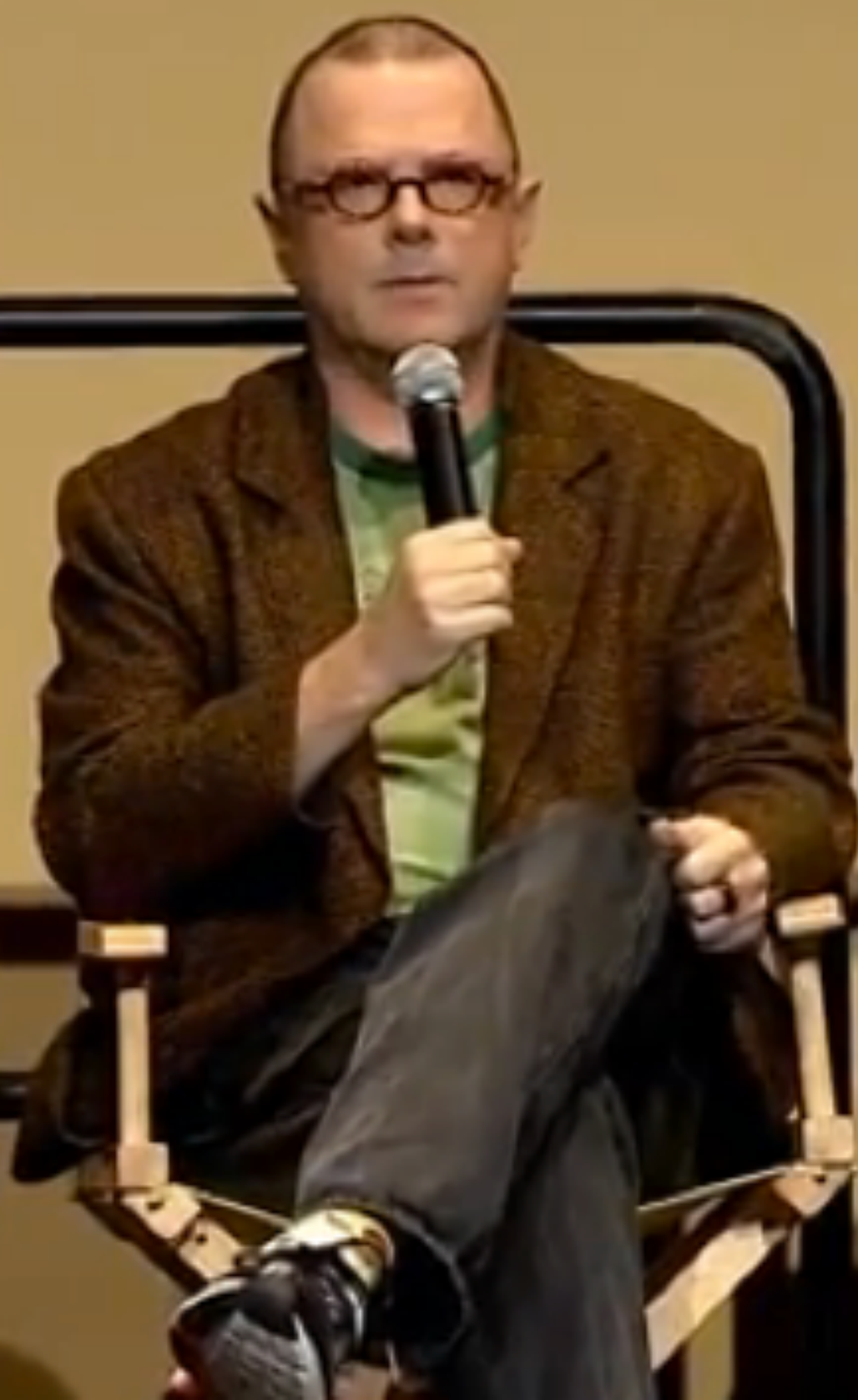
GO, O QUE É?

Uma linguagem open source

Robert Griesemer
(dev Google)

Rob Pike (Unix)

Ken Thompson
(Unix, B, C)



**POR QUE UMA NOVA
LINGUAGEM?**

Muitos problemas com software em grande
escala

Velocidade de compilação

Sistemas distribuídos Multicore

OBJETIVOS

Semântica simples

Programação concorrente

```
package main

import (
    "fmt"
    "time"
)

func say(s string) {} //código oculto para ficar legal no slide ;)

func main() {
    go say("world")
    say("hello")
}
```

Divertida!

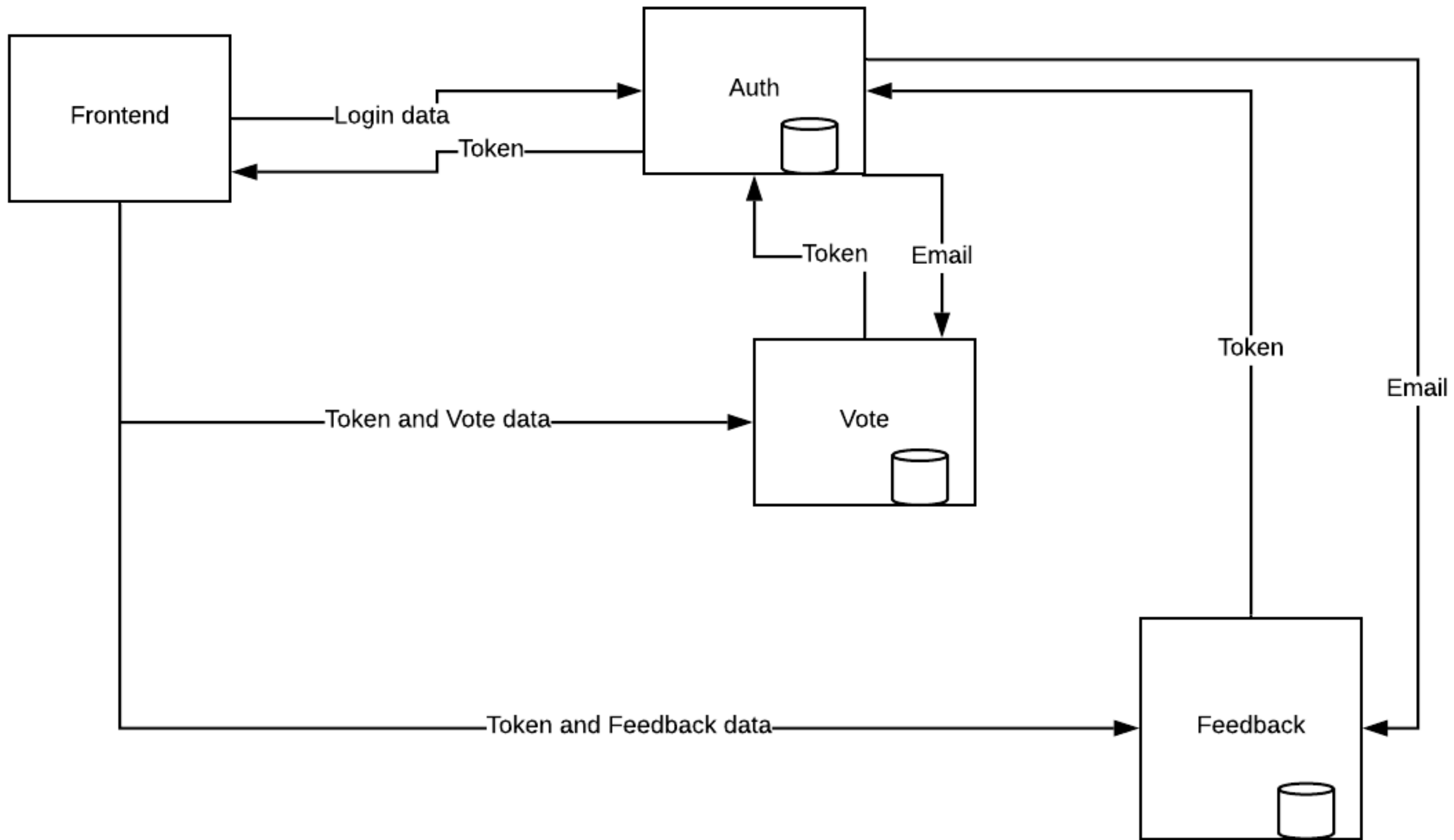
QUEM ESTÁ USANDO?

Google, Basecamp, Globo.com, Canonical, DigitalOcean, Dropbox, Github, Heroku, Medium, Docker, MongoDB, Mozilla, Netflix, New Relic, Mercado Livre, Cabify, New York Times, Resultados Digitais, Moip, Neoway, Walmart, Codenation, ContaAzul, TOTVS, etc

<https://github.com/golang/go/wiki/GoUsers>

TALK IS CHEAP, SHOW

ME THE CODE!



[https://github.com/eminetto/talk-
microservices-go](https://github.com/eminetto/talk-microservices-go)

MELHORIAS

- gRPC
- Containers
- Integração contínua/deploy contínuo
 - Kubernetes

**ISSO DEVERIA SER UM
MICROSERVIÇO?**

- Múltiplas taxas de mudança
- Ciclos de vida independentes
- Escalabilidade independente
 - Falha isolada
- Simplificar interações com dependências externas
- A liberdade de escolher a tecnologia certa para o trabalho

DICA EXTRA

Culture Check

REFERÊNCIAS

What are microservices?

MonolithFirst

Should that be a Microservice? Keep These Six

Factors in Mind

Awesome Go

<https://eltonminetto.dev>

<https://betrybe.com>

<http://asemanago.com.br>

<https://twitter.com/eminetto>