dito

Trilha Go

Uma abordagem simples para autenticação de uma API usando Go

dito

$ whoami

Raí Tamarindo

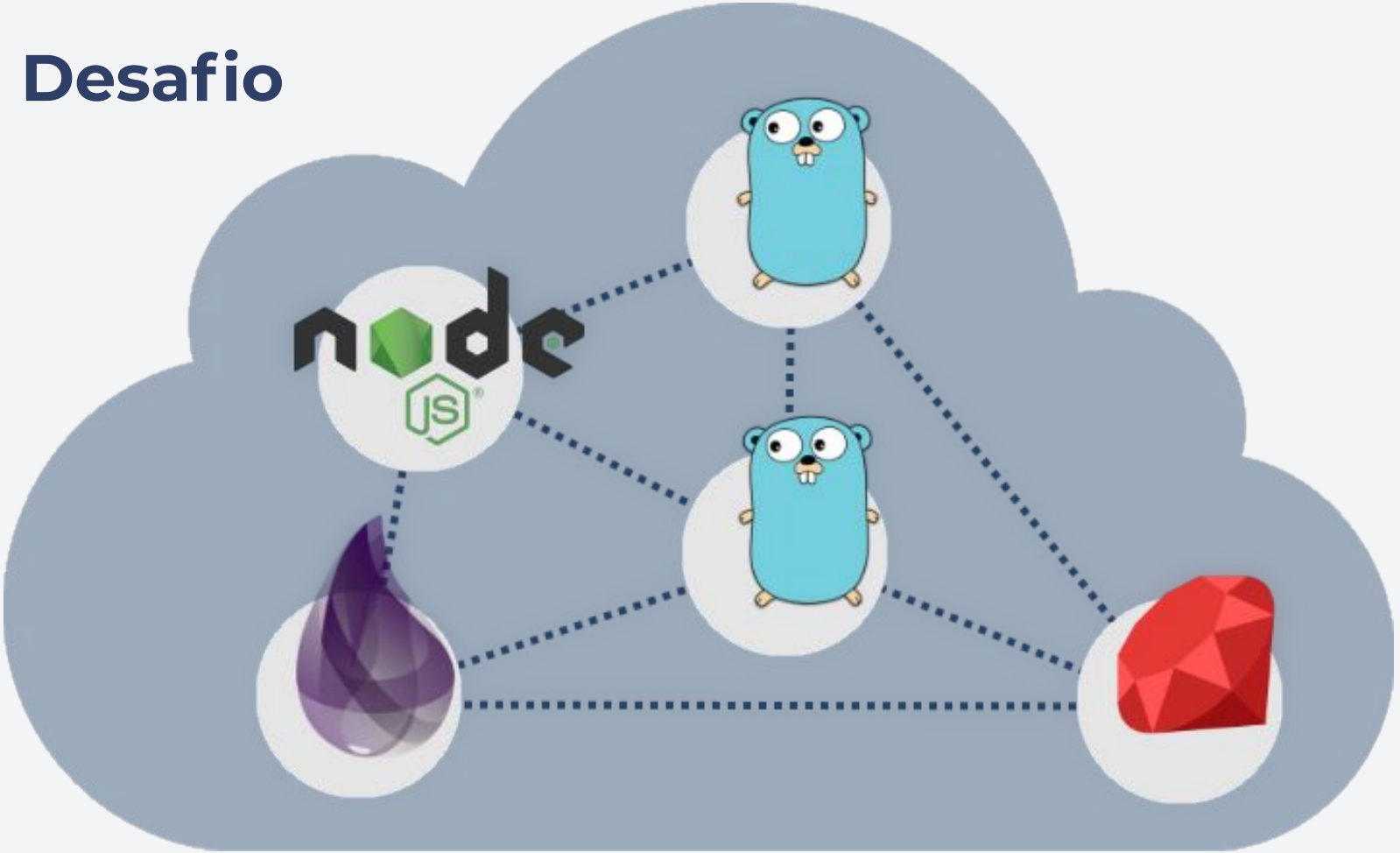
Software Engineer
rai.tamarindo@dito.com.br

/raitamarindo /rtamarindo

dito.com.br

AUTENTICAÇÃO X AUTORIZAÇÃO

O Desafio



O Desafio

60+ microservices
Quem serão os usuários?
Uso back-end e front-end?

Proxy Reverso

JSON Web Tokens

1

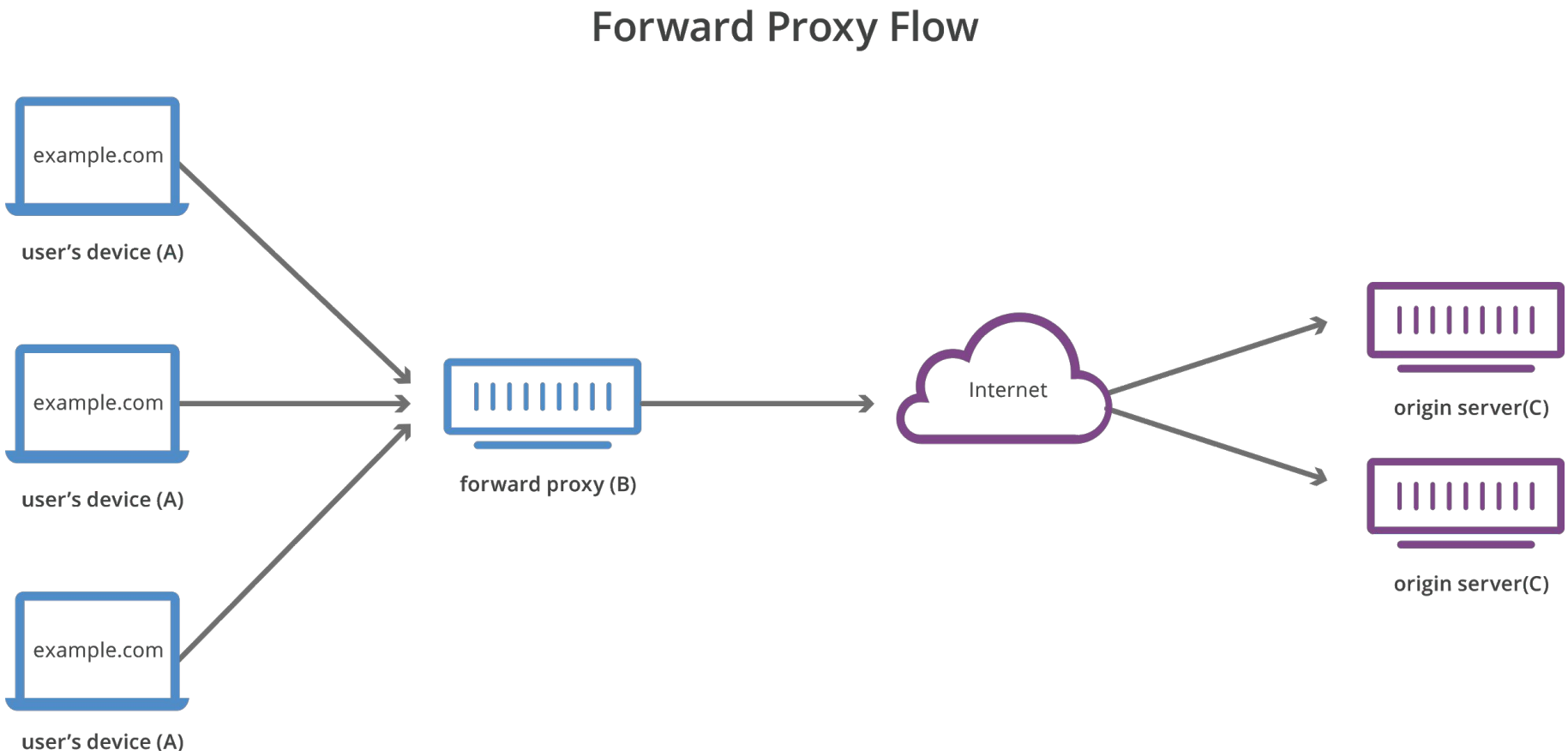
Proxy Reverso

PROXY REVERSO

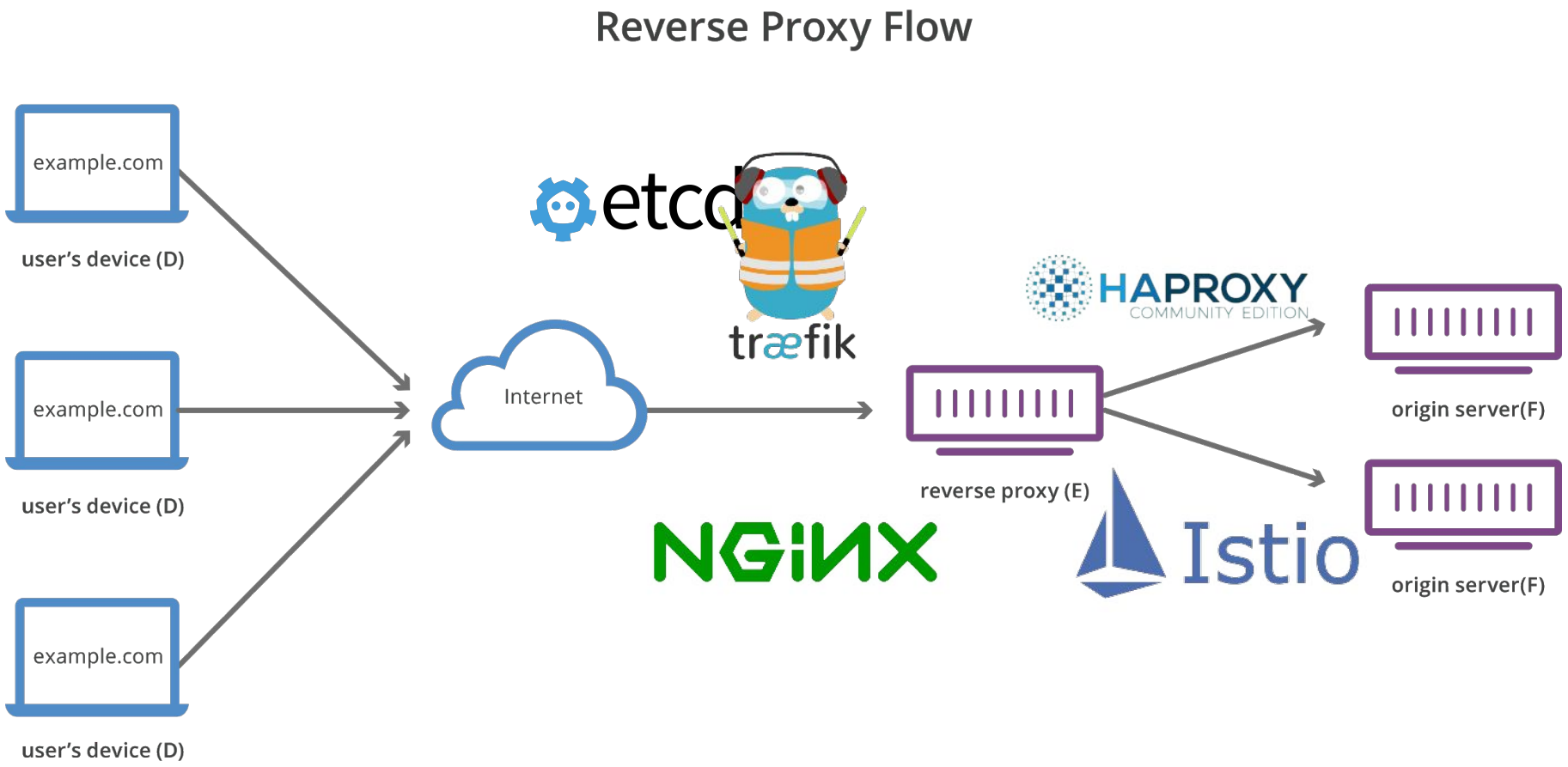
"A **reverse proxy** is a server that sits in front of web servers and forwards client requests to those web server."

Cloudflare

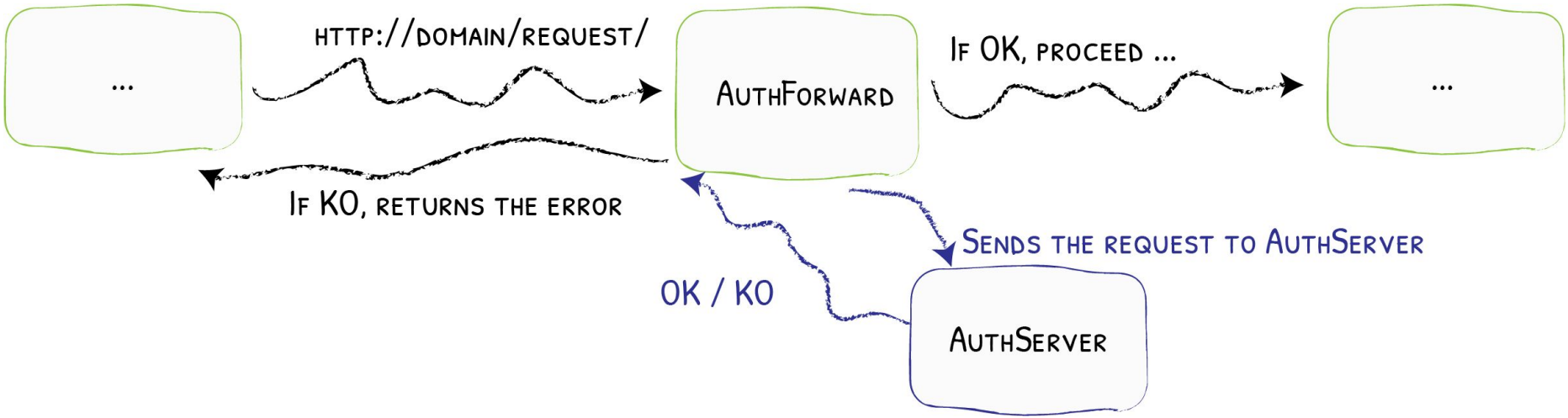
PROXY REVERSO



PROXY REVERSO



Forward Authentication



Configurações necessárias

TOML:

```
[entryPoints]
  [entryPoints.https]
    [entryPoints.https.auth.forward]
      address = "http://auth.local/validate"
```

ETCD:

```
/traefik/entrypoints/https/auth/forward/address
=http://auth.local/validate
```

2

JSON Web Tokens

JWT - JSON Web Tokens (RFC 7519)

```
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR
5cCI6IkpXVCJ9.eyJzdWIiOiJ0ZXN0IiwiaWF0IjoxNTE
2MjM5MDIyfQ.ZGWbu4-T4w7wSMIHdMrfYI5FVntFciRUu
pCZiuI_Cvo
```

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

```
{
  "sub": "test",
  "iat": 1516239022
}
```

Biblioteca

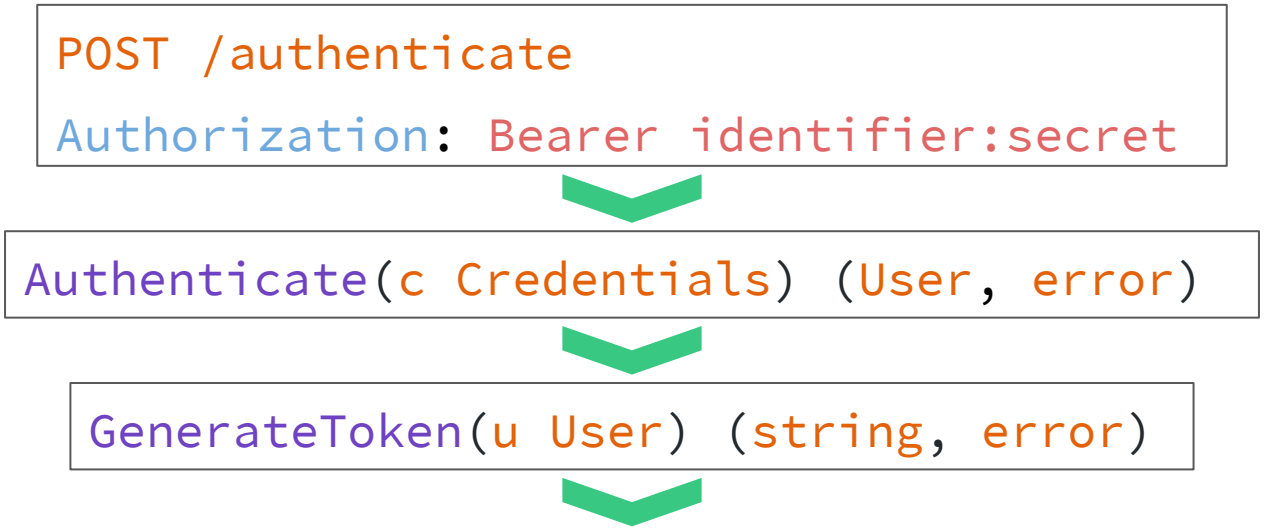
github.com/dgrijalva/jwt-go

- Sign
- Verify
- `iss` check
- `sub` check
- `aud` check
- `exp` check
- `nbf` check
- `iat` check
- `jti` check
- HS256
- HS384
- HS512
- RS256
- RS384
- RS512
- ES256
- ES384
- ES512
- PS256
- PS384
- PS512
- EdDSA

```
func (a Service) GenerateToken(u User) (string, error) {
    now := time.Now().Unix()
    expiration := now + int64(a.tokenPeriod)
    token := jwt.NewWithClaims(jwt.SigningMethodHS256,
        jwt.MapClaims{
            "sub": user.ID,
            "iat": now,
            "exp": expiration,
        })
    return token.SignedString([]byte(a.privateKey)) }
```

```
func (a Service) ValidateToken(t string) error {
    _, err := jwt.Parse(t, a.validationCallback)
    if err != nil {
        return errors.New("invalid access token")
    }
    return nil
}
```

```
func (a Service) validationCallback(token *jwt.Token)
(interface{}, error) {
    if _, ok := token.Method.(*jwt.SigningMethodHMAC); !ok {
        return nil, errors.New("unexpected signing method")
    }
    return []byte(a.privateKey), nil
}
```




```
200 OK
Content-Type: application/json
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJ0ZXN0IiwiaWF0
IjoxNTE2MjM5MDIyfQ.ZGWbu4-T4w7wSMIHdMrfYI5FVntFciRUupCZiuI_Cv"
```

```
GET /validate
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ
zdWIiOiJ0ZXN0IiwiaWF0IjoxNTE2MjM5MDIyfQ.ZGWbu4-T4w7wSMIHdMrfYI
5FVntFciRUupCZiuI_Cv
```

```
ValidateToken(t string) error
```

```
200 OK
```

3

Escalabilidade

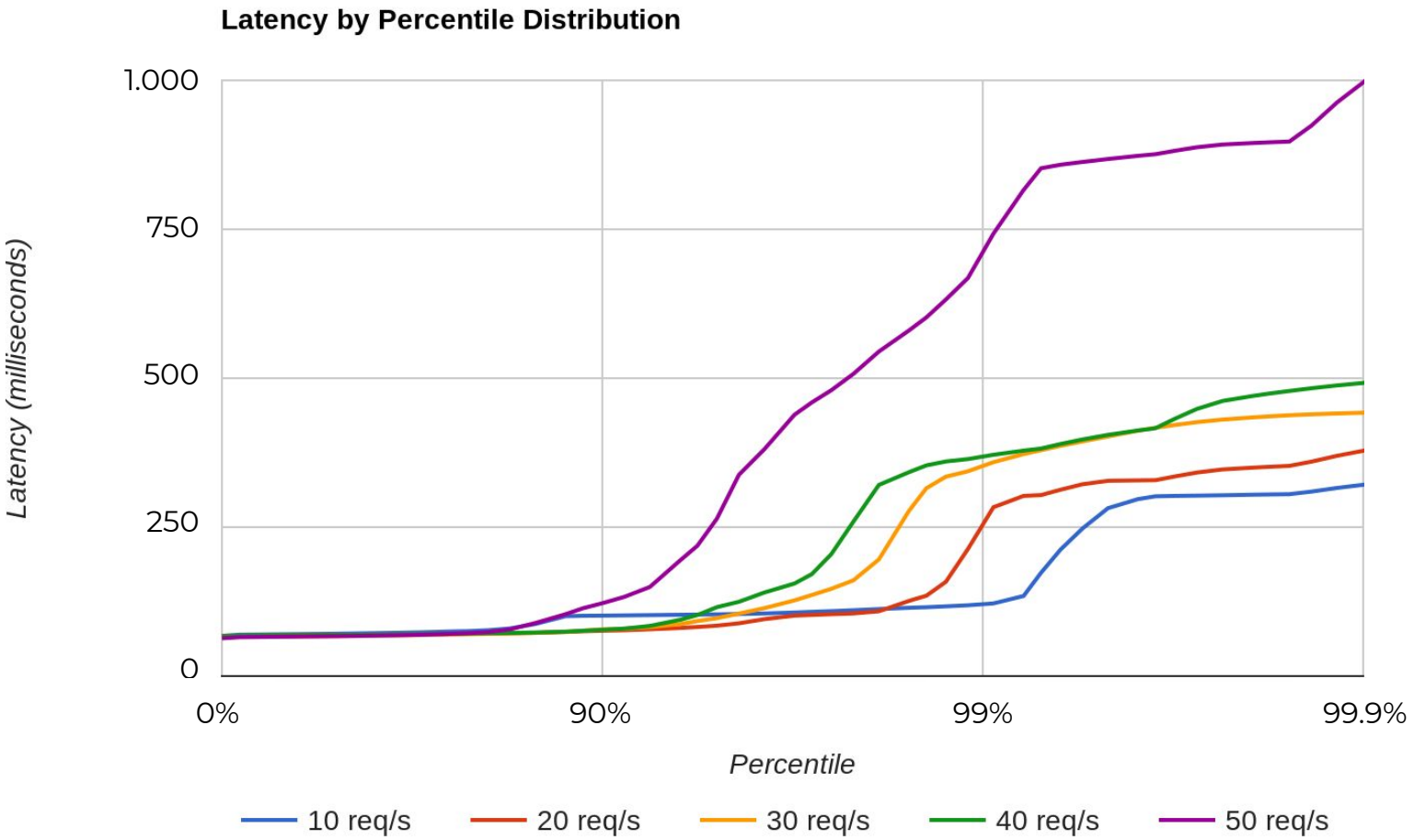
ESCALABILIDADE

Criptografia Simétrica

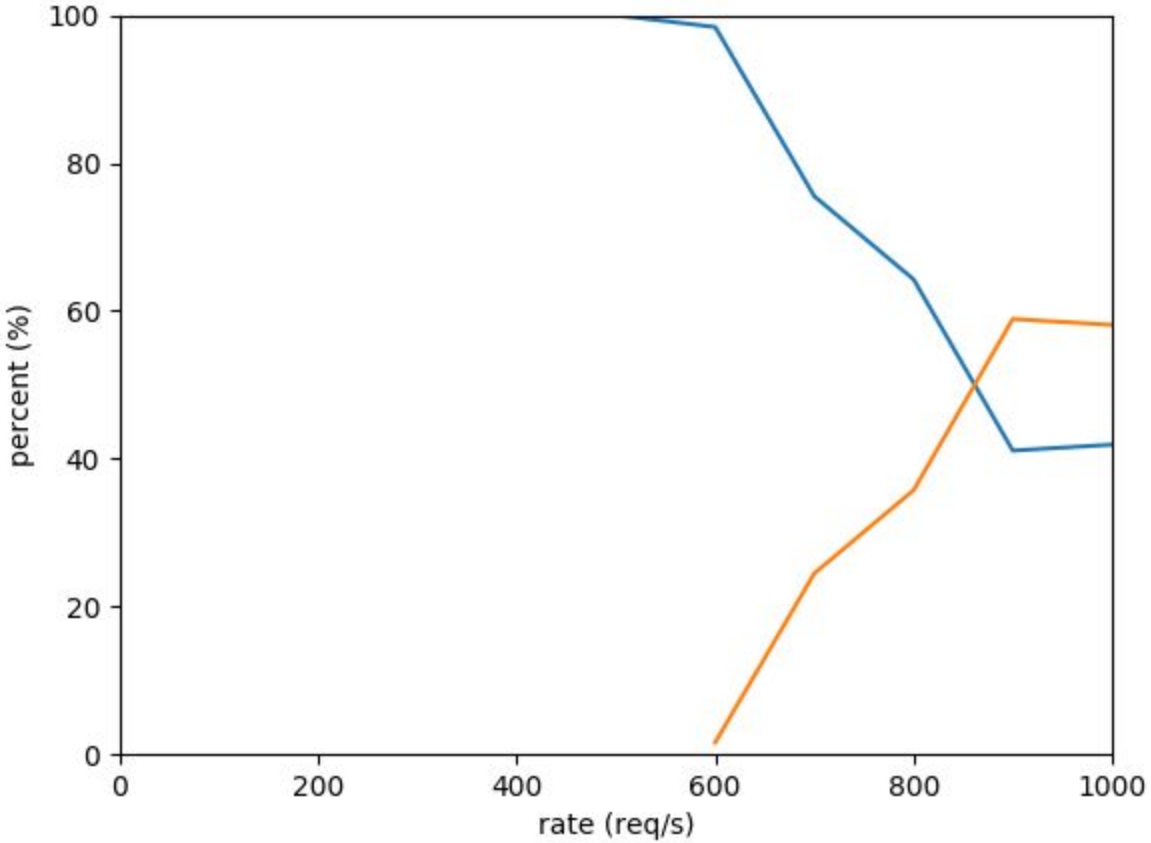
Autenticação Stateless

Sem Operações de IO

ESCALABILIDADE



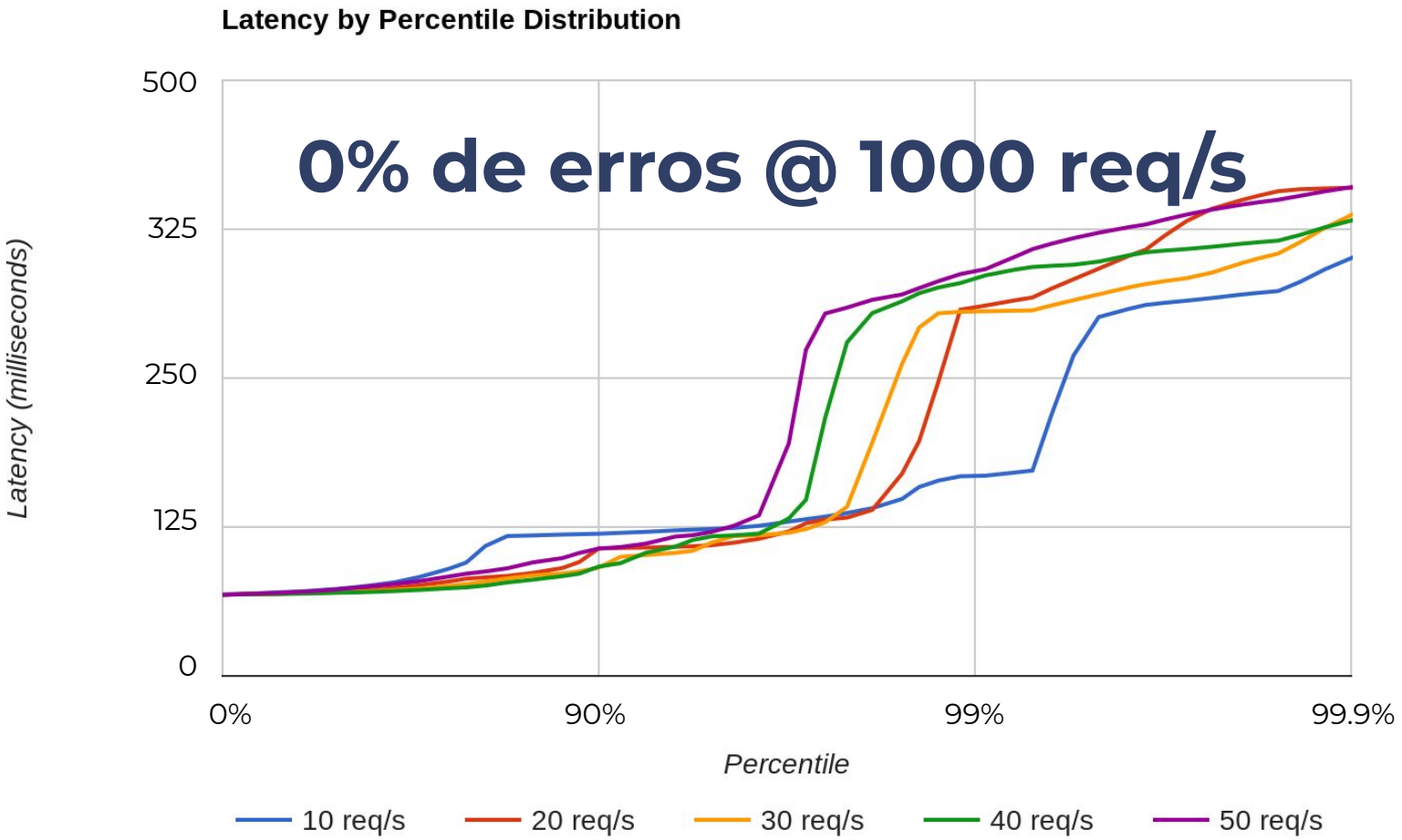
ESCALABILIDADE



RAM: **< 5 MB**
CPU: **< 0.0004 s**
Pico RAM: **< 20 MB**
Pico CPU: **< 0.04 s**

Escalabilidade Horizontal

ESCALABILIDADE



dito

OBRIGADO.

Raí Tamarindo

Software Engineer
rai.tamarindo@dito.com.br

dito.com.br