

# High Audience Python

Guilherme Vierno



# about:me

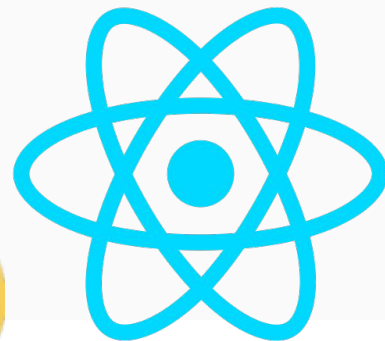
Pythonista desde 2010

Mineiro morando no Rio

Desenvolvedor na globo.com

Cosplayer de JC nas horas vagas

Bacharel em Física



# Python MG



# High Audience Python

Aplicação web com muito acesso

Right now

118698

active users on site

Fact check: Python não escala



# O que eu faço escalar minha aplicação?

*"Premature optimization is the root of all evil"*

-- Donald Knuth

Não otimize, **ainda**.

# Processo

1. Faça sua aplicação funcionar
2. Encontre o gargalo
3. Solucione o problema que gera o gargalo
4. GOTO 2

# A nossa aplicação

```
import tornado.web
```

```
class MainHandler(tornado.web.RequestHandler):
```


```
    def get(self):
```

```
        self.write("Hello, world")
```



# Medindo req/s

```
(benchmark) → tornado_demo wrk -t50 -c100 -d50s http://127.0.0.1:5000/
Running 50s test @ http://127.0.0.1:5000/
50 threads and 100 connections
Thread Stats      Avg      Stdev     Max    +/-  Stdev
  Latency      58.59ms    2.96ms   86.85ms    84.71%
  Req/Sec      33.95      8.05    60.00    79.51%
85363 requests in 50.10s, 16.69MB read
Requests/sec: 1703.99
Transfer/sec:  341.13KB
```



Alternativas ao wrk: ab e locust

# Esse número é bom?

Vamos supor **10M de usuários** em **10 horas**

**5 pageviews** por usuário

$1e7 / 10 / 60 / 60 * 5$

**~ 1500 views/requisições por segundo**

# Adicionando algum processamento

```
import tornado.web

from intensive_render import render

class MainHandler(tornado.web.RequestHandler):
    def get(self):
        html = render()    # 20ms
        self.write(html)
```

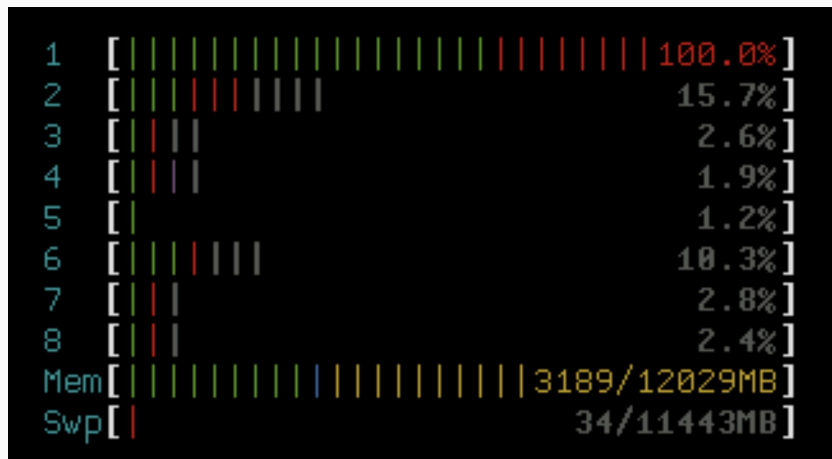
# Adicionando algum processamento

Requests/seg: **39.60**

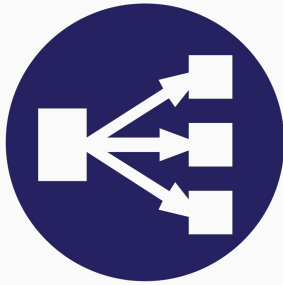
Máximo: **50 req/s**

**Solução:** Eu otimizoo o processo  
ou eu escalo com mais processos

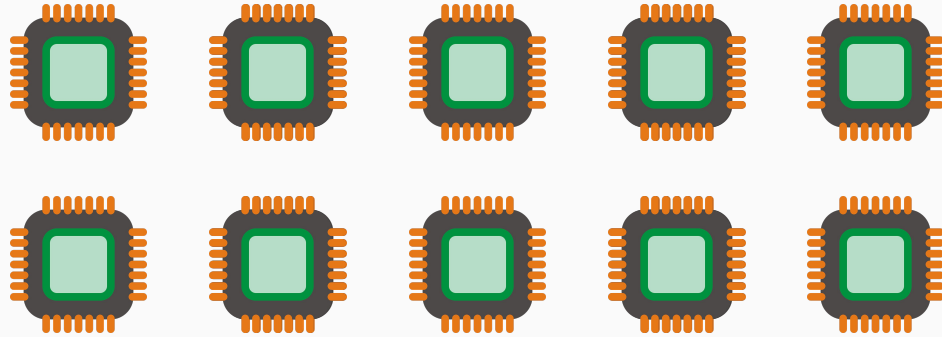
Com 8 processos: **314.16 req/s**



# Escalabilidade horizontal



Load balancer



# Autoscale



CPU 30%



CPU 85%



CPU 70%



CPU 50%



CPU 30%



**kubernetes**

# E se o serviço faz uma consulta externa?

```
from external import external_call
```

```
class MainHandler(tornado.web.RequestHandler):  
    def get(self):  
        html = render()    # 20ms  
        external_call()  
        self.write(html)
```

# E se o serviço faz uma consulta externa?

Taxa de requisições caiu para **9req/s**





# Profiling

	Time	Per Hit	% Time	Line Contents
=====				
no				@profile
che				def get(self):
	21785.0	21785.0	17.6	html = render() # 20ms
	100286.0	100286.0	81.1	external_call()
dem	1602.0	1602.0	1.3	self.write(html)
che				

[https://github.com/rkern/line\\_profiler](https://github.com/rkern/line_profiler)

# Async aio resgate

```
from aioexternal import external_call
```

```
class MainHandler(tornado.web.RequestHandler):
```

```
    async def get(self):
```

```
        html = render()    # 20ms
```

```
        await external_call()    # 100ms but async
```

```
        self.write(html)
```

300 req/s

# Atenção às dependências

- **asyncio:**

- <https://github.com/aio-libs/>
- aiohttp
- aioredis
- aiomysql
- aiopg

- **tornado:**

- toredis
- tornado-alf

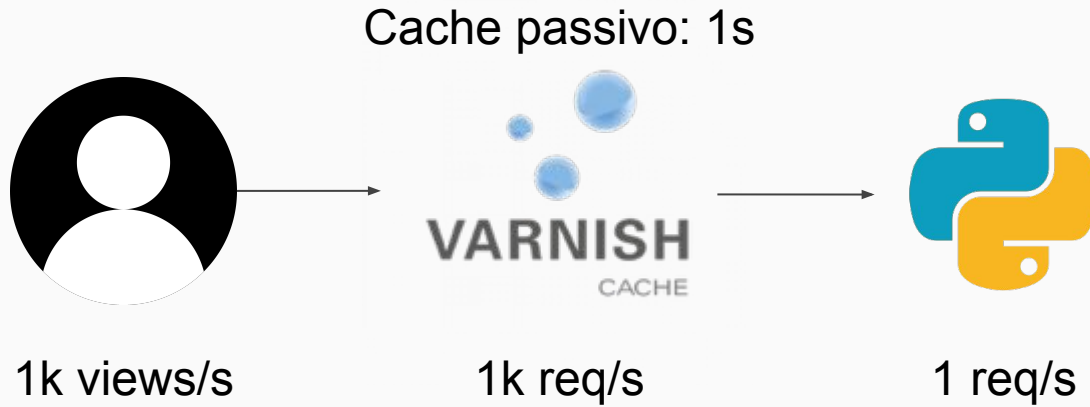
# Escalamos nosso código, e agora?

---

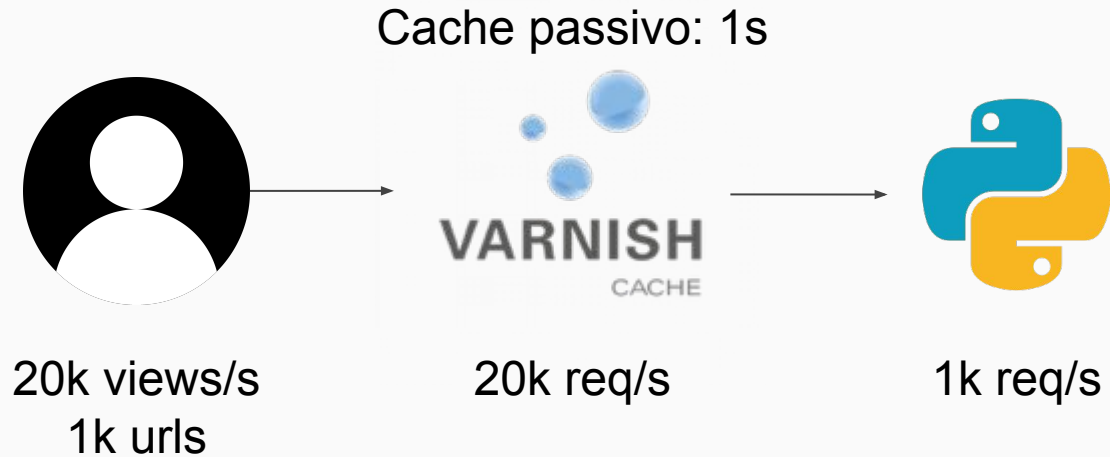
Escalabilidade horizontal pode sair caro...

Hora de pensar em arquitetura!

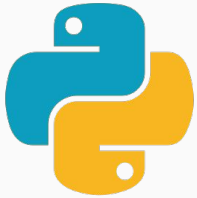
# Cache passivo



# Cache passivo por url



# Adicionando um banco de dados



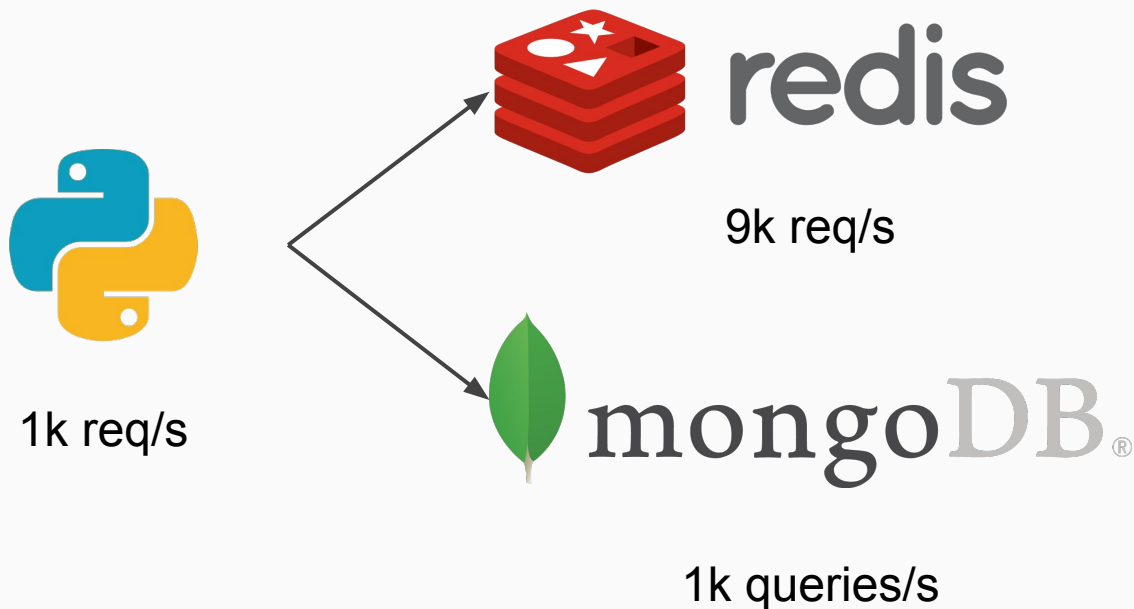
1k req/s



mongoDB®

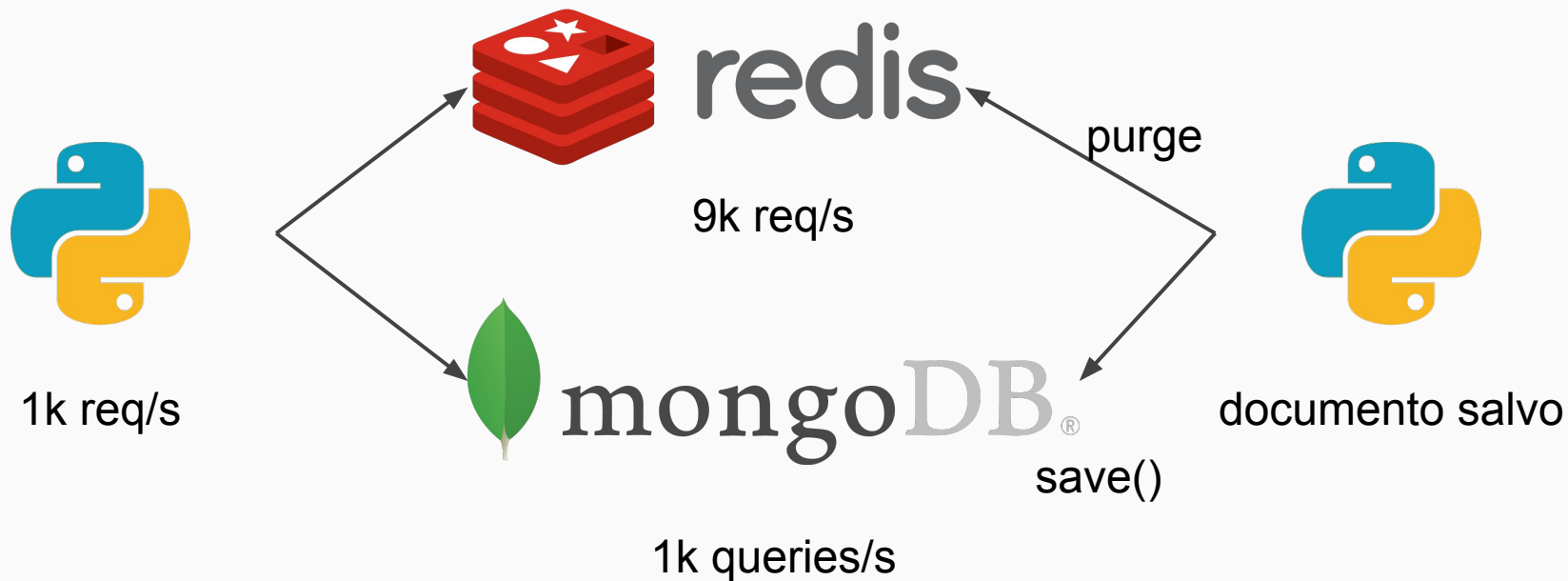
10k queries/s

# Mais cache em memória



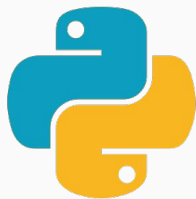


# Expiração ativa



# E escrita?

Buffer em memória

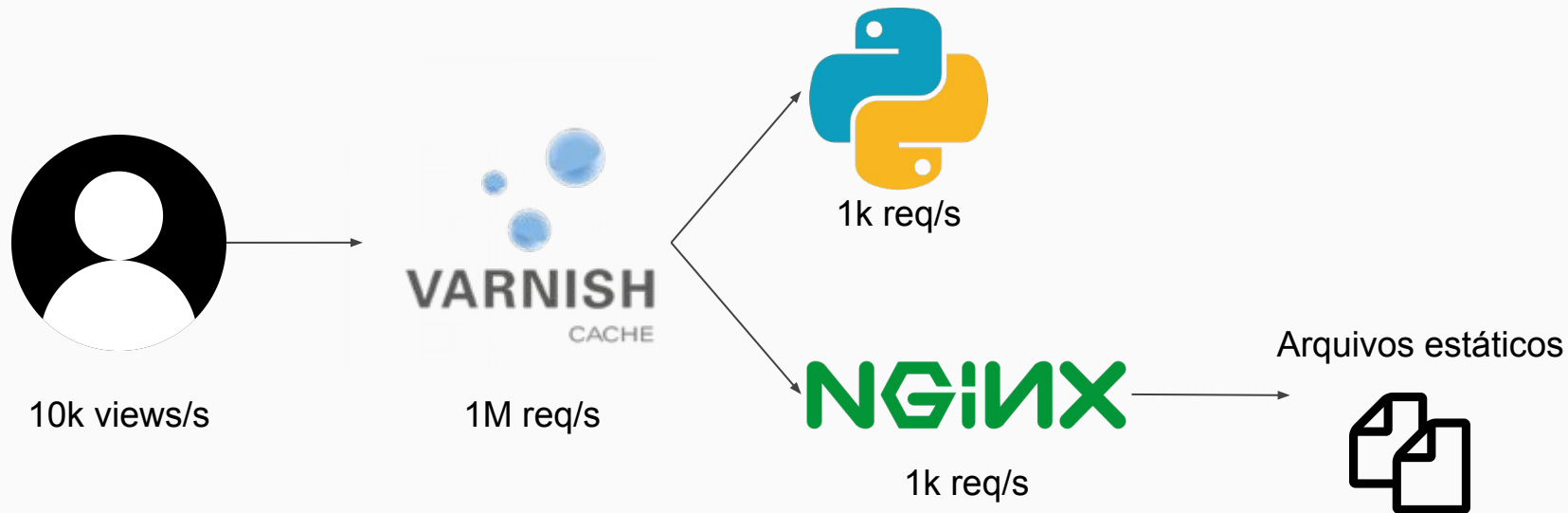


20k docs/s

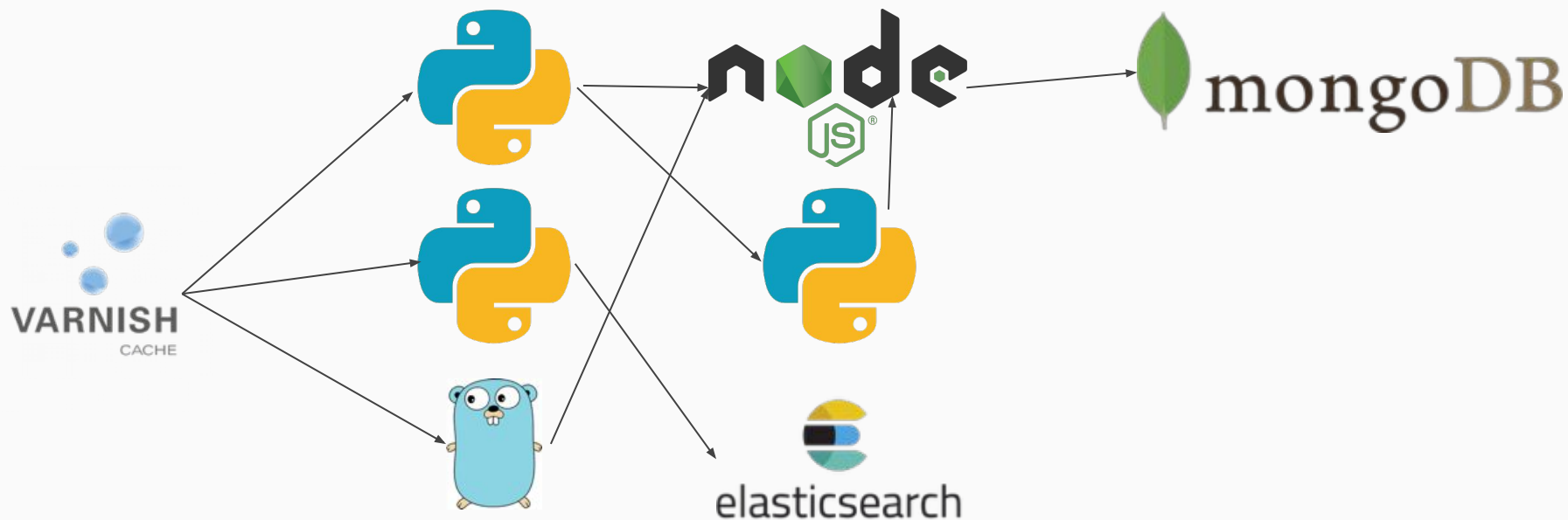


20 bulks/s

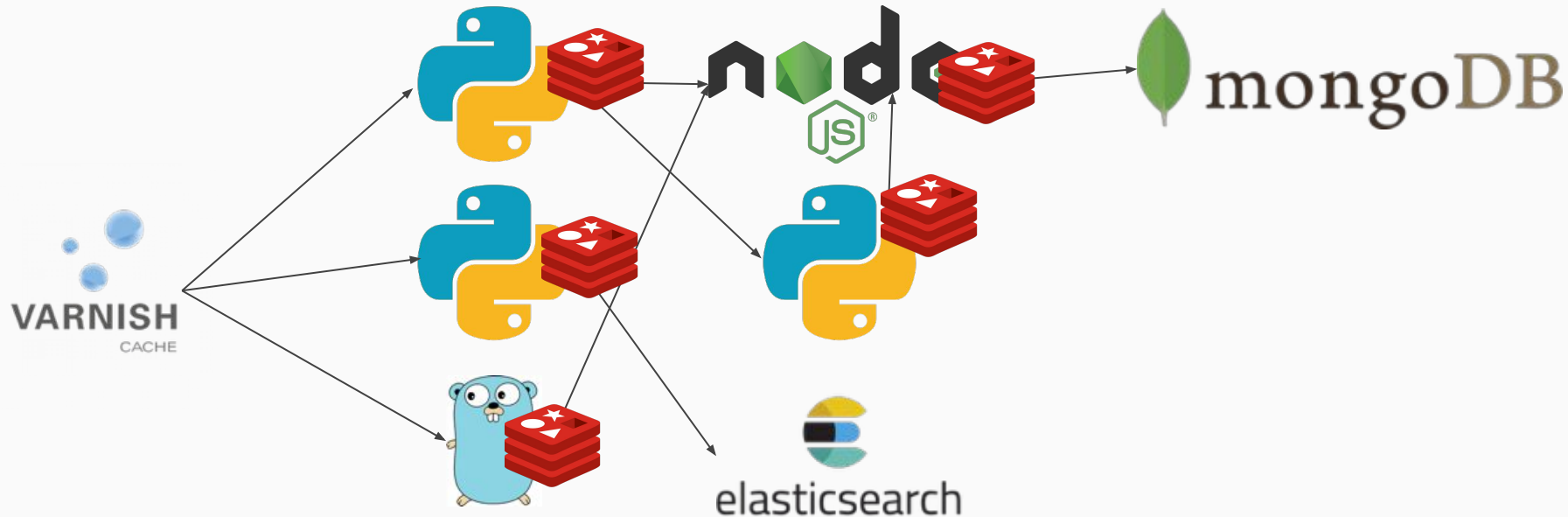
# Uma página é mais que HTML



# Separação em [micro]serviços



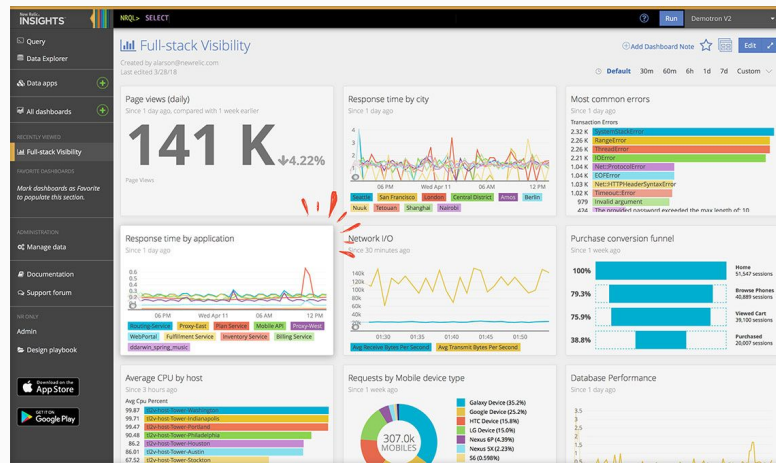
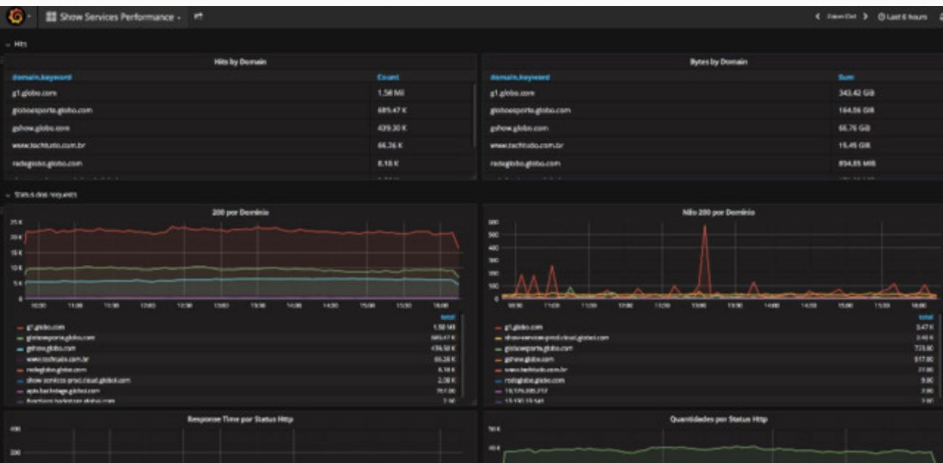
# Separação em [micro]serviços



# Monitoração

Identificar os gargalos

Prevenir problemas de escala



# Resumão

**Não optimize prematuramente!** Você não precisa de Kubernetes ou Microserviços até você precisar.

Identifique o gargalo e se o problema é limitado CPU/MEM ou IO

Pense em escalabilidade através de arquitetura e código

# Obrigado!

Contato: [guilhermevierno@gmail.com](mailto:guilhermevierno@gmail.com)

[opensource.globo.com](https://opensource.globo.com)

[talentos.globo.com](https://talentos.globo.com)

