

Coerência de cache de granularidade fina, usando django-signals

Helber Maciel Guerra

- Analista de Sistemas
- Computação de alto desempenho (HPC)
 - Clusters, Cloud
- Desenvolvedor Middleware IPTV

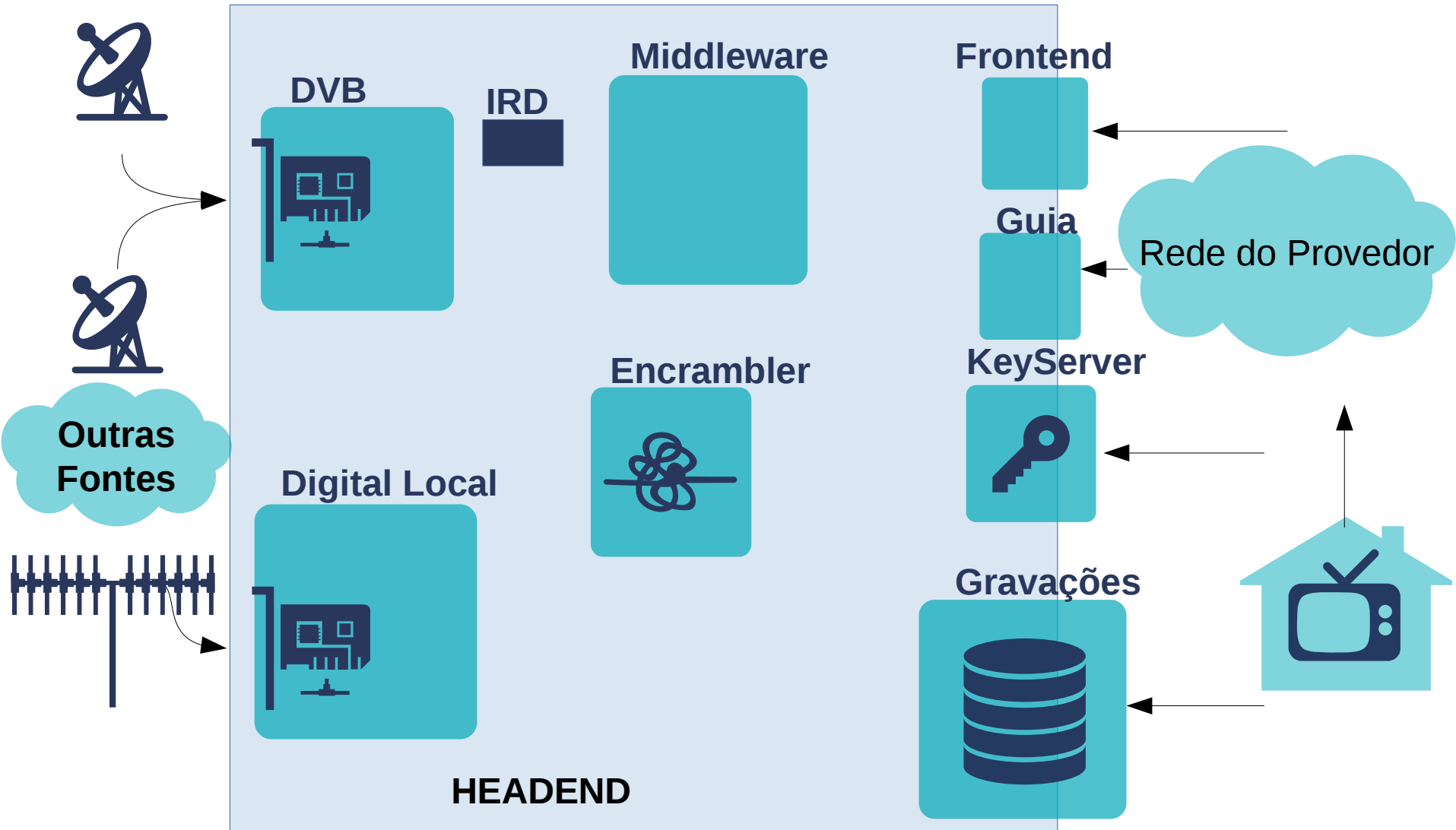
Agenda

- De onde veio a necessidade?
 - Sistema de TV por assinatura
 - O que é?
 - Funcionamento
 - Arquitetura
 - Problema a ser atacado
 - Muito acesso concorrente simultâneo
 - Passos percorridos
 - Entendendo o cache no nginx
 - Como e porque coerência de cache?
 - Entendendo as requisições de APIs (usuários, rest)
 - Usando Signals
 - Resultados

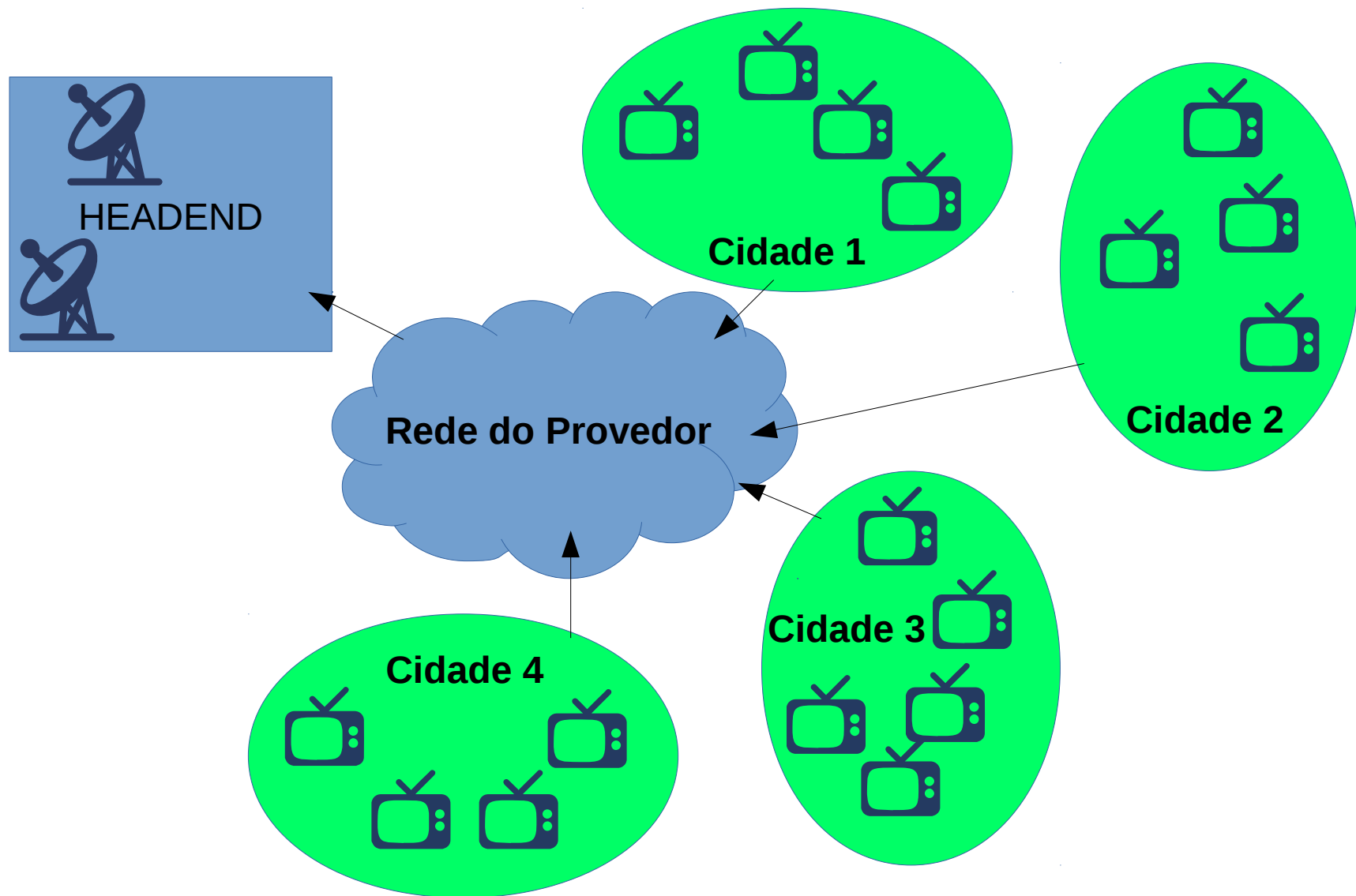
TV por assinatura

Vídeos

HEADEND - IPTV



HEADEND - IPTV



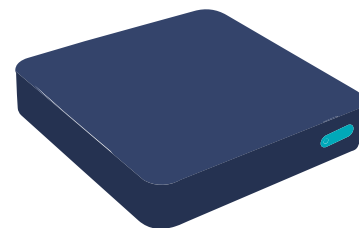
SETTOPBOX

- Linux HTML5 (Framebuffer, Qt, GTK)
- Android com HTML5 (aproveitar aplicações de terceiros - netflix, email, jogos, ...)
- Aceleração em hardware para decode de vídeo
- Aceleração em hardware para decrypt, descramble
- Sem estado no dispositivo ("TUDO" está no middleware)



Inicialização STB

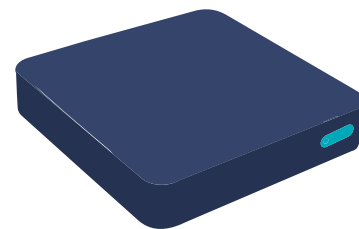
- Boot do sistema
- Provisionamento
- Sincronização de horário (NTP)
- Atualizações de Firmware
- Conversa com o CAS (Chaves de acesso aos conteúdos)
- Carrega o Browser (Geralmente Webkit)
- Aponta para o middleware
- Carrega interface GUI (html, css, javascript, fontes, ...)
- Autenticação
- Carrega, informações:
 - Canais
 - Guia de programação
 - Configs (Audio, volumes, agendamentos, gravações, controle parental, perfis)
 - APIs externas (Ex. Youtube, facebook, twitter, imdb, telefone, e-mail, ...)
 - Plano (Canais que tem acesso)
 - Configuração dos canais
 - Multicast
 - OTT
 - Rádios
 - ...
- Canal de comunicação
 - Controle remoto (Remoto)
 - Mensagens
 - Gerencia remota
 - Qualidades
 - ...





Inicialização STB

- Boot do sistema
- Provisionamento
- Sincronização de horário (NTP)
- **Atualizações de Firmware**
- Conversa com o CAS (Chaves de acesso aos conteúdos)
- Carrega o Browser (Geralmente Webkit)
- Aponta para o middleware
- **Carrega interface GUI (html, css, javascript, fontes, ...)**
- **Autenticação**
- Carrega, informações:
 - **Canais**
 - **Guia de programação**
 - **Configs (Audio, volumes, agendamentos, gravações, controle parental, perfis)**
 - **APIs externas (Ex. Youtube, facebook, twitter, imdb, telefone, e-mail, ...)**
 - **Plano (Canais que tem acesso)**
 - Configuração dos canais
 - **Multicast**
 - **OTT**
 - **Rádios**
 - ...
- **Canal de comunicação**
 - **Controle remoto (Remoto)**
 - **Mensagens**
 - **Gerencia remota**
 - **Qualidades**
 - ...



Inicialização STB

- Atualizações de Firmware

Conteúdo estático

- Carrega interface GUI (html, css, javascript, fontes, ...)

- Autenticação

Conteúdo Dinamico (APIs)

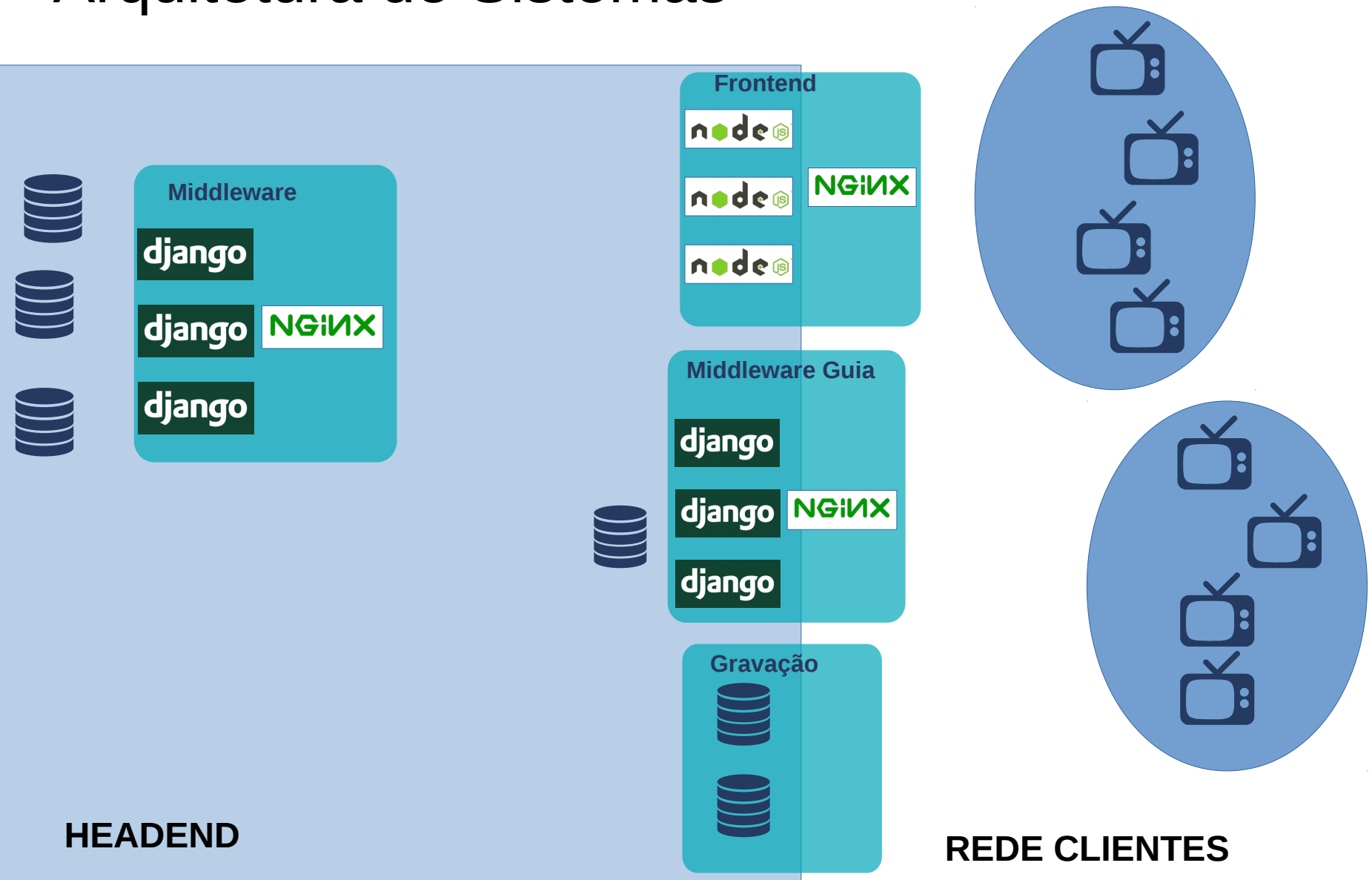
- Canais
- Guia de programação
- Configs (Audio, volumes, agendamentos, gravações, controle parental, perfis)
- APIs externas (Ex. Youtube, facebook, twitter, imdb, telefone, e-mail, ...)
- Plano (Canais que tem acesso)

- Multicast
- OTT
- Rádios

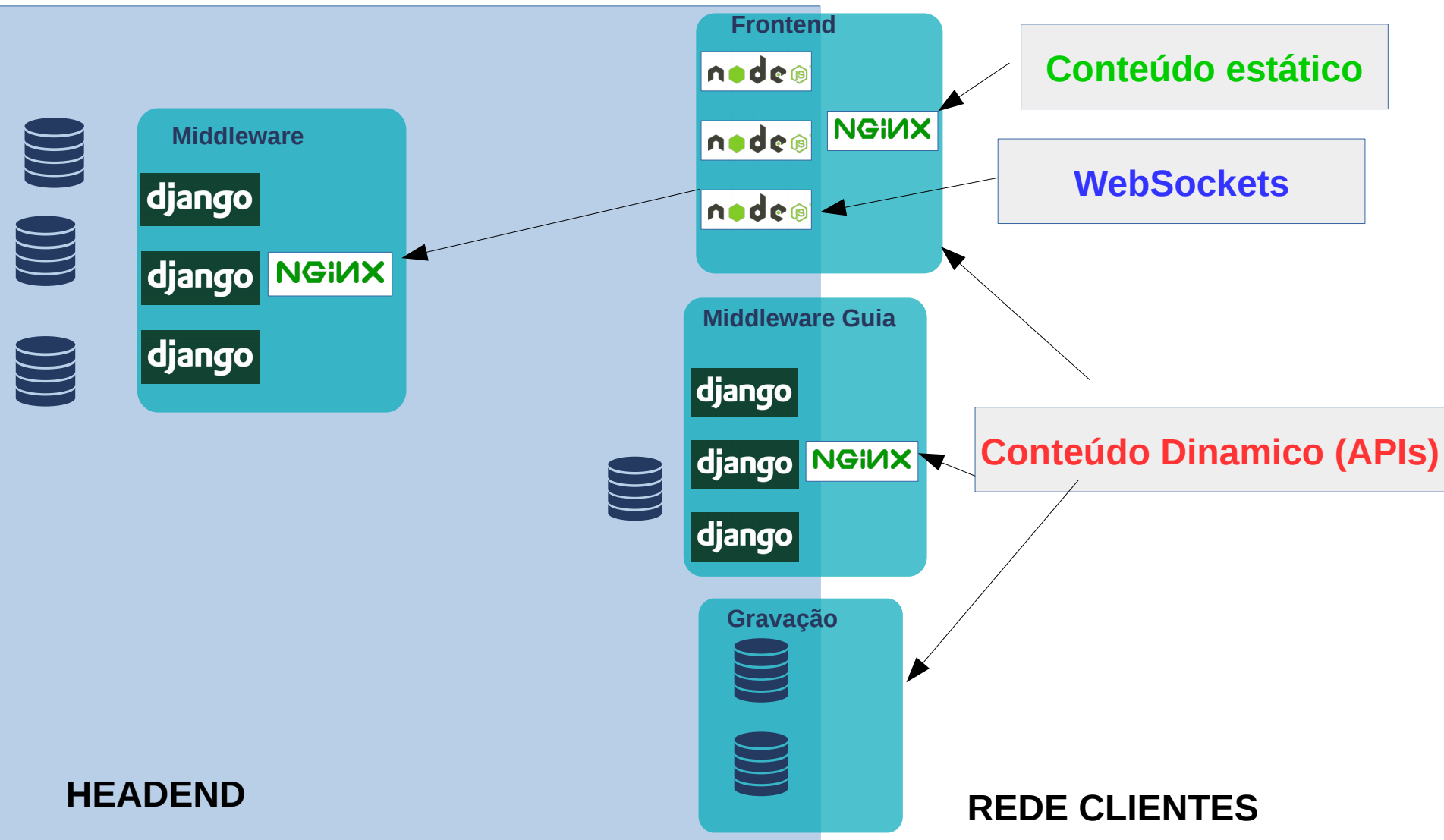
- Controle remoto (Remoto)
- Mensagens
- Gerencia remota
- Qualidades

WebSockets

Arquitetura de Sistemas



Arquitetura de Sistemas





Requisições HTTP (APIs)

- Atualizações de Firmware

Conteúdo estático

- Carrega interface GUI (html, css, javascript, fontes, ...)

- Autenticação

- Canais

- Guia de programação

- Configs (Audio, volumes, agendamentos, gravações, controle parental, perfis)

- APIs externas (Ex. Youtube, facebook, twitter, imdb, telefone, e-mail, ...)

- Plano (Canais que tem acesso)

- Multicast

- OTT

- Rádios

Conteúdo Dinamico (APIs)

- Controle remoto (Remoto)

- Mensagens

- Gerencia remota

- Qualidades

WebSockets



Requisições HTTP (APIs)

TOTAL DE CHAMADAS = 76

- Atualizações de Firmware

- Carrega interface GUI (html, css, javascript, fontes, ...)

- Autenticação

- Canais

- Guia de programação

- Configs (Audio, volumes, agendamentos, gravações, controle parental, perfis)

- APIs externas (Ex. Youtube, facebook, twitter, imdb, telefone, e-mail, ...)

- Plano (Canais que tem acesso)

- Multicast

- OTT

- Rádios

- Controle remoto (Remoto)

- Mensagens

- Gerencia remota

- Qualidades

Conteúdo estático:

Firmware = 1

HTML = 1

Font = 1

Img = 7

CSS = 11

JS = 6

Conteúdo Dinamico (APIs):

XHR (APIs) = 41

WebSockets:

WS = 1 (Mas com muitos frames)



Anatomia do Request

Request URL: <http://127.0.0.1/tv/api/tv/v2/channel/>

Request Method: GET

Status Code: 200 OK

Request Headers

[view source](#)

Accept: */*

Accept-Encoding: gzip, deflate, sdch

Accept-Language: pt-BR,pt;q=0.8,en-US;q=0.6,en;q=0.4

AUTHORIZATION: ApiKey STB_FF2130706433:faa8695e31c0d15637beadadcceaab09e0283551

Connection: keep-alive

Cookie: _ga=GA1.1.163808754.1438824852; csrftoken=6jBl4fOVQwYSxCfodHr8IyFGWQ2PsM0x; sessionid=skxpyzrn2627

Host: 127.0.0.1

MAC: FF2130706433

Referer: <http://127.0.0.1/menu>

User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/50.0.2661.94 Safari/537.36

X-Requested-With: XMLHttpRequest

Response Headers

[view source](#)

Cache-Control: no-cache

Connection: keep-alive

Content-Type: application/json

Date: Tue, 10 May 2016 17:23:43 GMT

Server: nginx/1.8.1

Set-Cookie: csrftoken=eBinGqowsfpsLb6LTD1lwoNsdADZnaMz; expires=Tue, 09-May-2017 17:23:29 GMT; Max-Age=31449

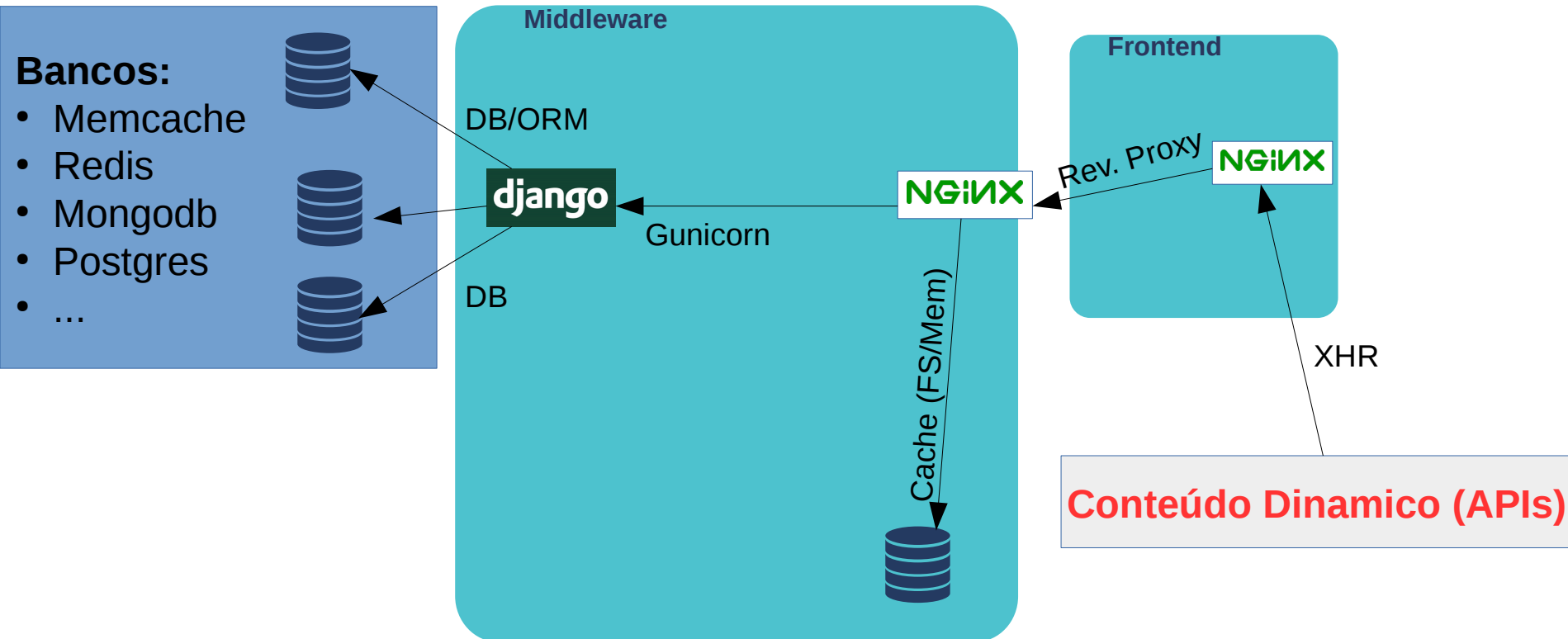
Set-Cookie: sessionid=skxpyzrn2627ftpmqjz4c7l0ldjwxmk; expires=Tue, 24-May-2016 17:23:29 GMT; httponly; Max-Age=1

Transfer-Encoding: chunked

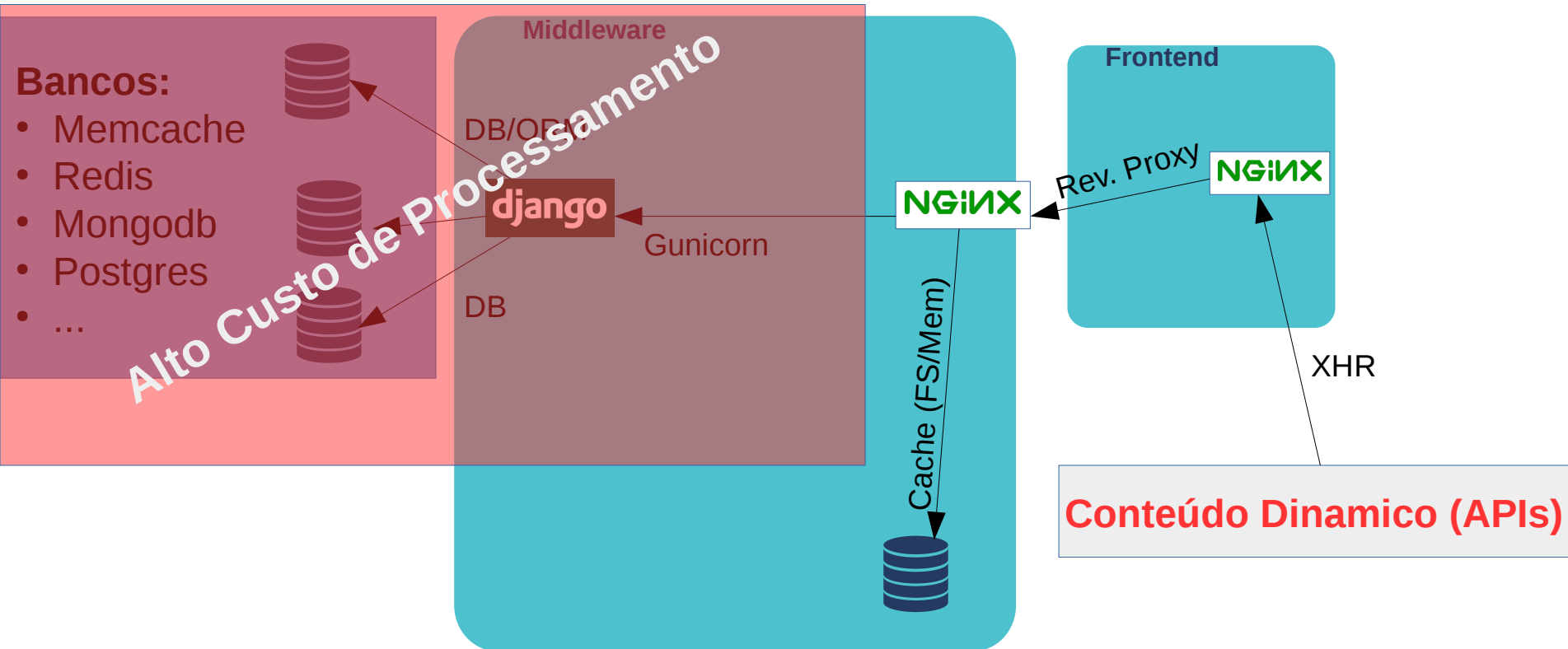
Vary: Accept, Cookie

X-Cache-Status: HIT

Cominho da Request



Cominho da Request



Carga na requisição

Demo TIME

Configurando o Cache

```
# nginx.conf
http {
    ...
    include upstream/*.conf;
    include conf.d/*.conf;
    ...
}
```

```
# conf.d/server.conf
server {
    ...
    include cache/*.conf;
    include plugins/*.conf;
}
```

```
# upstream/middleware.conf
upstream middleware {
    ...
    server unix:/iptv/var/run/site_iptv_gunicorn.socket;
}
```

```
# plugins/channel.conf
proxy_cache_path /var/lib/cache/channel levels=1:2 keys_zone=cache_channel:512M max_size=3000M;
```

```
# cache/channel.conf
location ~ ^/tv/api/tv/v2/channel/ {
    access_log          /iptv/var/log/nginx/server_mw_cache_channel.log cache_key;
    proxy_cache cache_api_tv_v2_channel;
    proxy_cache_key "$scheme$request_uri";
    proxy_cache_valid 200 302 12M; # 12 meses
    proxy_cache_valid 301 12M; # 12 meses
    proxy_cache_valid any 10m; # 10 minutos
    proxy_cache_min_uses 1;
    add_header X-Cache-Status $upstream_cache_status;
    proxy_cache_lock on;
    proxy_cache_use_stale updating error timeout invalid_header http_500;
    proxy_ignore_headers "X-Accel-Expires" "Vary" "Cache-Control" "Expires" "Set-Cookie";
    proxy_pass http://middleware;
    proxy_cache_purge PURGE from 127.0.0.1;
}
```

Configurando o Cache

Request URL: **http://127.0.0.1/tv/api/tv/v2/channel/**

Request Method: GET

Status Code: 200 OK

`$scheme $request_uri`

```
# cache/channel.conf
location ~ ^/tv/api/tv/v2/channel/ {
    access_log          /iptv/var/log/nginx/server_mw_cache_channel.log cache_key;
    proxy_cache cache_api_v2_channel;
    proxy_cache_key "$scheme$request_uri";
    proxy_cache_valid 200 302 12M; # 12 meses
    proxy_cache_valid 301 12M; # 12 meses
    proxy_cache_valid any 10m; # 10 minutos
    proxy_cache_min_uses 1;
    add_header X-Cache-Status $upstream_cache_status;
    proxy_cache_lock on;
    proxy_cache_use_stale updating error timeout invalid_header http_500;
    proxy_ignore_headers "X-Accel-Expires" "Vary" "Cache-Control" "Expires" "Set-Cookie";
    proxy_pass http://middleware;
    proxy_cache_purge PURGE from 127.0.0.1;
}
```

Processo de Cache

Request URL: **http://127.0.0.1/tv/api/tv/v2/channel/**

Request Method: GET

Status Code: 200 OK

\$scheme \$request_uri

MD5("http/tv/api/tv/v2/channel/") = 799e5d973283bcfe17b290ca176cc1f5

```
# plugins/channel.conf
proxy_cache_path /var/lib/cache/channel levels=1:2 keys_zone=cache_channel:512M max_size=3000M;
```

/var/lib/cache/channel/5/1f/799e5d973283bcfe17b290ca176cc1f5

```
KEY: http/tv/api/tv/v2/channel/
HTTP/1.0 200 OK
Server: gunicorn/19.3.0
Date: Tue, 10 May 2016 19:16:32 GMT
Connection: close
Vary: Accept, Cookie
Content-Type: application/json
Cache-Control: no-cache
```

```
{"meta": {"limit": 20, "next": "/tv/api/tv/v2/channel/?limit=20&offset=20", "offset": 0, "previous": null,
"total_count": 215}, "objects": [{"channelid": "static_Portal Life HDTV", "description": "Life HDTV - Em breve...",
"name": "Portal Life-815 BBC Earth à partir de 01/09", "next": "/tv/api/tv/v2/channel/192/",
```

O que é um usuário?

- Uma abstração com alguma informação enviada nas requisições
 - Exemplos: Token, ApiKey, SessionId, ... (qualquer coisa)
- Estas informações são passadas nas requisições de várias formas:
 - HTTP-HEADERS, QueryString, Post, ...

O que é um usuário?

- No Django:
 - Middleware de 'django.contrib.auth'
 - Para a request identifica o usuário e injeta no objeto request
 - O programador pega o usuário autenticado na view, CBV, ...
 - Usando o ORM, o programador cria as regras de negócio, filtra conteúdos, etc.



O que é um usuário?

Request URL: http://127.0.0.1/tv/api/tv/v1/userchanneltvod/?api_key=faa8695e31c0d15637beadadcceaab09e0283551

Request Method: GET

Status Code: 200 OK

Request Headers

view source

Accept: */*

Accept-Encoding: gzip, deflate, sdch

Accept-Language: pt-BR,pt;q=0.8,en-US;q=0.6,en;q=0.4

AUTHORIZATION: ApiKey STB_FF2130706433:faa8695e31c0d15637beadadcceaab09e0283551

Connection: keep-alive

Cookie: _ga=GA1.1.163808754.1438824852; csrftoken=eBinGqowsfpsLb6LTD1lwoNsdADZnaMz;

sessionid=skxpyzrn2627ftpmqjz4c7l0ldjwkmk

Host: 127.0.0.1

MAC: FF2130706433

Referer: http://127.0.0.1/menu

User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/50.0.2661.94

Safari/537.36

X-Requested-With: XMLHttpRequest

Cache Key com “Usuário”

```
1 # vim: filetype=nginx:ts=4:sw=4
2 location ~ ^/tv/api/tv/v2/userchannel/ {
3     access_log /iptv/var/log/nginx/server_mw_cache_userchannel.log timed_combined;
4     proxy_cache cache_api_tv_v2_userchannel;
5     proxy_cache_key "$scheme$request_uri$http_api_key$http_authorization";
6     proxy_cache_valid 200 302 12M; # 12 meses
7     proxy_cache_valid 301 12M; # 12 meses
8     proxy_cache_valid any 10m; # 10 minutos
9     proxy_cache_min_uses 1;
10    add_header X-Cache-Status $upstream_cache_status;
11    proxy_cache_lock on;
12    proxy_cache_use_stale updating error timeout invalid_header http_500;
13    proxy_ignore_headers "X-Accel-Expires" "Vary" "Cache-Control" "Expires" "Set-Cookie";
14    proxy_pass http://middleware;
15    proxy_cache_purge PURGE from 127.0.0.1;
16 }
```




Coerência de cache?

- Se alguma informação for **modificada**, **invalidar** ou **refazer** o cache para que o conteúdo do cache esteja **coerente** com o que foi modificado.

Coerência de Cache

- Detectando a modificação da informação no django:
 - Signal:
 - Ajuda a desacoplar o código
 - Deixa seu componente de software “ouvindo” eventos
 - O Django já vem com uma série de signals “built-in” (pre_save, post_save, post_delete, m2m_changed, ...)



Módulo de Cache purge

https://github.com/FRiCKLE/nginx_cache_purge/

Coerência de Cache

```
11 class TVConfig(AppConfig):
12     name = 'tv'
13     verbose_name = _('TV')
14
15     def ready(self):
16         from . import dispatch # NOQA
17         from .models import cache_clear_user_channel
18         models_signals = getattr(settings, 'TV_CACHE_SIGNALS', [])
19         for model in models_signals:
20             m = apps.get_model(model)
21             signals.pre_save.connect(cache_clear_user_channel, sender=m)
22             signals.pre_delete.connect(cache_clear_user_channel, sender=m)
23
24
```

```
TV_CACHE_SIGNALS = ('tv.Channel', 'client.SetTopBoxChannel')
```

Coerência de Cache

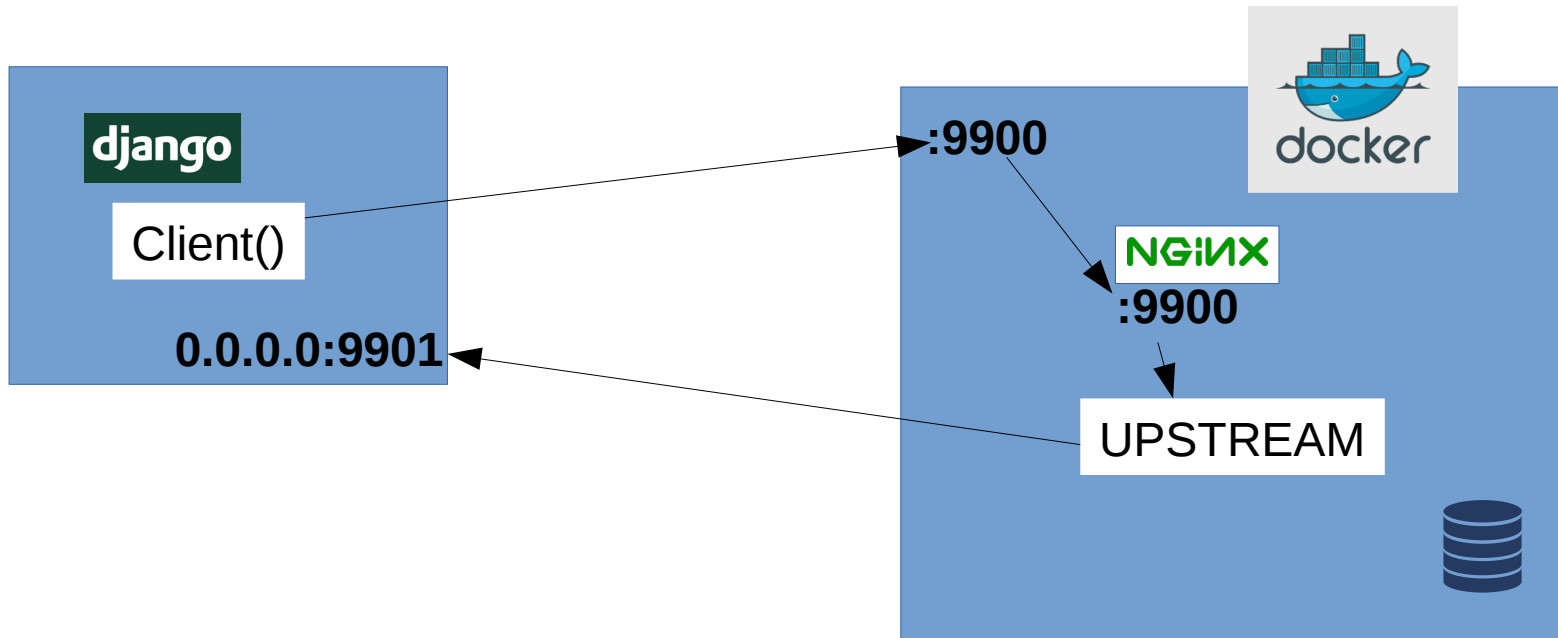
- Limpando
 - Request.get('PURGE', url, headers=data)
 - Tem que ter “TUDO” exatamente como foi feito o request
 - Salvando em algum lugar

```
134 .....url = 'http://{host}:{port}{path}'.format(↵
135 .....|.....host=host, port=port, path=result['path'])↵
136 .....response = requests.request("PURGE", url, headers=headers)↵
137 .....log.info(↵
138 .....|.....'PURGE Url:[%s]%s, Headers:[%s] Data:[%s]',↵
139 .....|.....response.status_code, url, headers, result↵
140 .....|.....)↵
141
```

Teste

- TDD + Integração
 - UnitTest
 - Docker (Garantir limpeza, estado e isolamento)

Teste



Teste

Demo TIME

Conclusão

- É possível fazer cache de APIs com dados de usuário
- Python + Django
 - Não é só para fazer Blog
 - Nem sisteminhas simples
 - Framework full stack ajuda (e muito)
- Independente da sua aplicação, use cache sempre que possível

Obrigado

- Helber Maciel Guerra
 - Twitter: @helbermg (<https://twitter.com/helbermg>)
 - E-mail:
 - helbermg@gmail.com