Especificação de Requisitos

para

Arquitetura VOD

07/09/2017

0.3

Preparado por:

Equipe DevOps

1. **Introdução 3**
   1. Overview 3
   2. Objetivos 3
   3. Escopo 3
2. **Restrições Gerais de Design 4**
   1. Ambiente de Produção 4
   2. Ambiente de QA 4
   3. Arquitetura Atual 5
   4. Restrições Obrigatórias 7
3. **Proposta de Nova Arquitetura 8**
   1. Arquitetura 8
   2. Requisitos Funcionais 9
   3. Requisitos Não-Funcionais 10
4. **Features 11**
   1. Feature: Publicação de Cue Points 11
      1. Descrição 11
   2. API VOD v3: Disponibilização de Mídias 12
      1. Descrição 12

Histórico de Revisões

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Data** | **Nome** | **Descrição** |
| 0.1 | 06/06 | Júlio | Versão com itens 1 e 2 |
| 0.2 | 07/08 | Júlio | Adição de itens 3 e 4 |
| 0.3 | 07/09 | Júlio | Adição de item 4.2 |

# Introdução

## Overview

Este documento visa descrever da arquitetura geral para a leitura, inserção e distribuição de metadados de mídia VOD pela equipe de DevOps/Plataformas Digitais da Globosat.

Nele estão listados os requisitos básicos, os atores envolvidos e a abordagem geral dos projetos associados. Também endereçamos os principais aspectos de dúvidas e cenários dos quais o projeto se deparará. Não estamos considerando elementos como cronogramas, custo ou metodologias utilizadas no desenvolvimento. Assumimos que estes itens devam ser levantados em outro documento e/ou também durante a implementação do projeto.

## Objetivos

Foram traçados objetivos primários que justificam a adoção de uma nova arquitetura VOD baseados no cenário atual.

O projeto tem os seguintes objetivos:

1. Prover uma arquitetura que disponibiliza mídias com mais confiabilidade de metadados e menor tempo de publicação. Por consequência, maior vazão de títulos disponibilizados.
2. Menor intervenção humana e maior controle no processo de publicação visando melhorar a qualidade das mídias publicadas.
3. Aumento de ferramentas de monitoria de assertividade de qualidade do serviço de publicação.
4. Aumento de documentação e entendimento das atividades da arquitetura.
5. Melhoria na sinergia entre os times envolvidos e terceiros.

## Escopo

Como escopo do projeto, definiremos as principais aplicações, suas formas de interface de informação, seus requisitos técnicos e cenários de problemas. Não definiremos aqui especificações técnicas que não sejam relevantes ao projeto e ao negócio assim como detalhes de implementação.

Utilizaremos ao máximo especificações gráficas em detrimento a descrições textuais.

Também daremos prioridade a abstrações que exponham da melhor forma aspectos relevantes em prol do entendimento geral da arquitetura. As limitações do projeto de arquitetura se limitam a descrição e apontamento de pontos de atenção e não necessariamente a resolução de problemas e propostas de soluções alternativas.

# Restrições Gerais de Design

## Ambiente de Produção

A Arquitetura VOD será implementada no ambiente conhecido como “Colocation Globo.com” onde outras arquiteturas e outras aplicações estão também em funcionamento. Temos também já consolidado padrões de implantação assim como especificações para que a Arquitetura VOD seja colocada em prática. Vamos assumir que as aplicações da Arquitetura serão adaptadas para o uso no “Colocation Globo.com” assim como as necessidades de acesso a outras aplicações externas também serão efetuadas em tempo de implantação. Além disso, temos que considerar adaptações de serviços externos como algo indesejado já que depende de atores terceiros. A seguir a topografia do ambiente de Produção “Colocation Globo.com”

<inserir imagem>

## Ambiente de QA

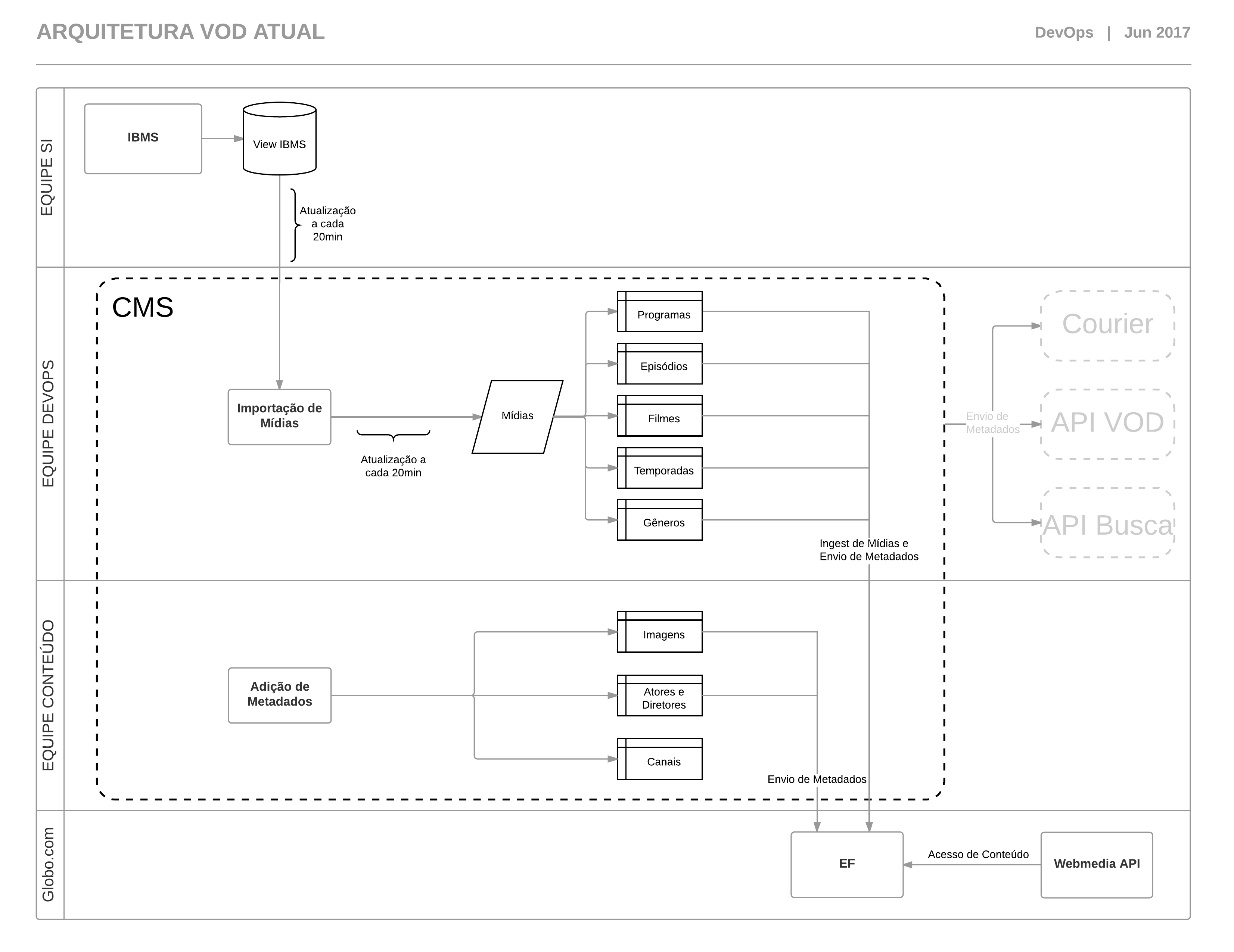
Assim como o ambiente de produção, temos um 2º ambiente destinado a disponibilização de aplicações para testes de aceitação e novas funcionalidades. Este ambiente visa replicar a topografia do ambiente de Produção assim como a dinâmica de implantação das aplicações. O diferença em relação ao ambiente de produção se refere a forma de acesso a aplicação externas e perfomance reduzida.

A seguir a topografia do ambiente de QA “Colocation Globo.com”

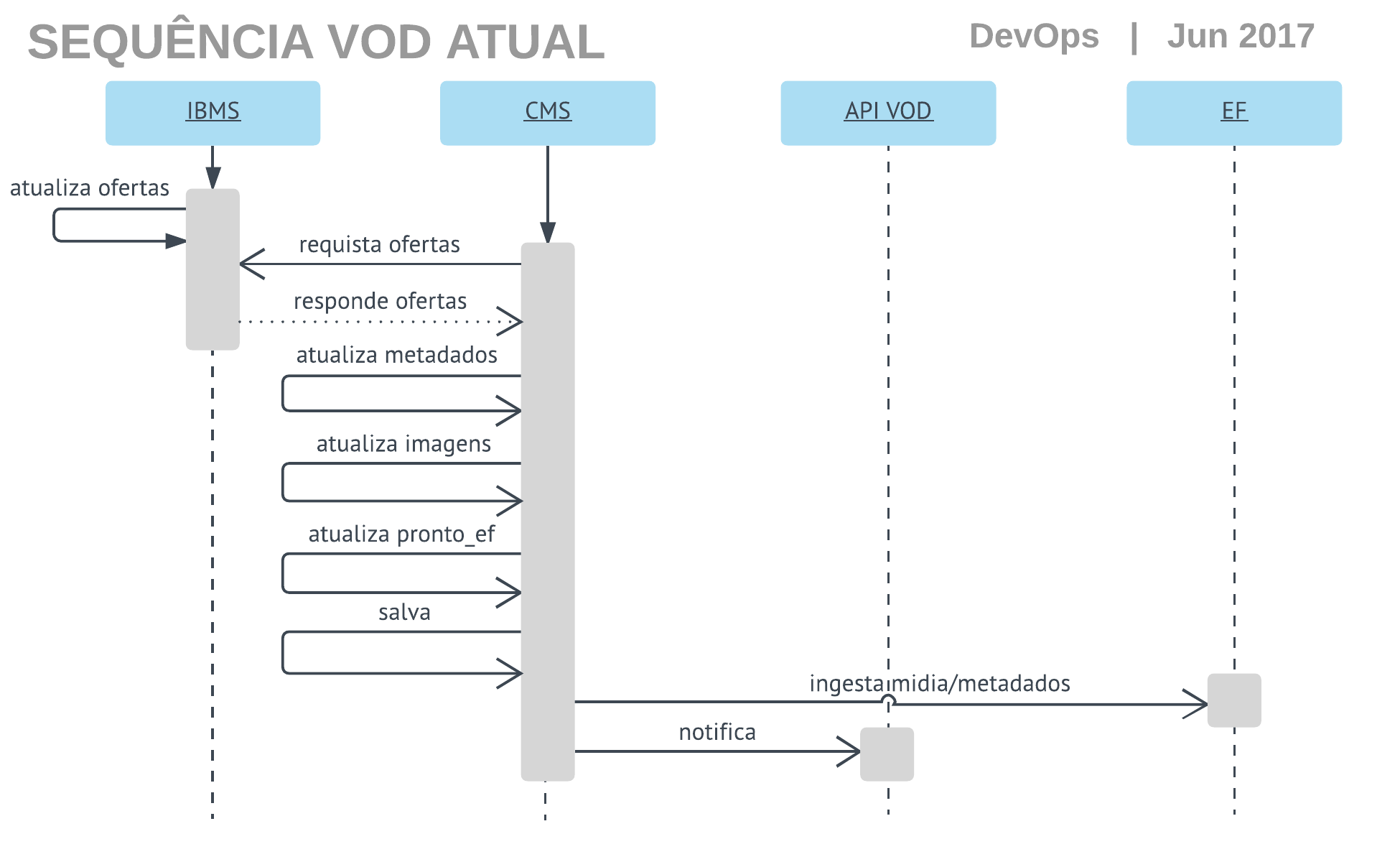
<inserir imagem>

## Arquitetura Atual

Segue abaixo a disposição atual da Arquitetura VOD disposta com os seguintes itens:

* Faixa com os atores responsáveis por cada etapa.
* Estruturas de persistência de dados:
  + Ofertas
  + Programas
  + Episódios
  + Filmes
  + Temporadas
  + Gêneros
  + Imagens
  + Atores e Diretores
  + Canais
* Temporização das atividades de atualização de dados

A responsabilidade do CMS se estende da importação de dados do IBMS, edição de metadados, enriquecimento com imagens (cards e background) e agendamento de publicação na plataforma da EF. É importante perceber que somente uma aplicação deter diversas responsabilidades causa problemas que a Arquitetura VOD sofre hoje. Estas características de acoplamento fazem com que qualquer tipo de monitoramento, relatório e futuras melhorias sejam custosas e passíveis de erro.

O CMS também possui outras funcionalidades de publicação de metadados e mídias para operadores representada pelo agrupamento “Courier”. Há também a API de Busca que utiliza os metadados disponibilizados para o *Elasticsearch*. Por fim a API VOD recebe os dados também do CMS. Tanto a API de Busca quanto a API VOD recebe os dados de forma passiva através de processos em background do CMS chamados de “Notificações” que traduzem o metadado cadastrados para o metadado no formato estabelecido pela API VOD e no formato requerido pelo mapeamento no *Elasticsearch*. Representamos esses projetos satélites apenas pelo agrupamento para focarmos especialmente no agrupamento CMS e na publicação VOD para EF. É importante lembrar que a mudança de arquitetura VOD deve permitir que outras aplicações sejam compatíveis com as estruturas de dados e com o uso das equipes e estes projeto satélites.

A ação de requisição de oferta é feita pelo sistema através de um agendamento de tarefa. Essa tarefa conta a responsabilidade de gerenciar as ofertas já importadas assim como traduzir os dados disponibilizados pelo IBMS para o formato de mídia do CMS. As ações de atualização de metadados, adição de imagens e marcar de flag “Pronto EF” são feitas manualmente no CMS pela equipe de conteúdo.

Ao salvar uma mídia, é responsabilidade do CMS adicionar mais uma tradução do modelo no formato CMS para o modelo API VOD através de uma tarefa gerenciada em background. Também é executado no salvamento da mídia a atualização de metadados, ou ingestão no caso do primeiro salvamento da mídia. Todas essas etapas são realizadas assincronamente e gerenciadas pela equipe de DevOps. A equipe de conteúdo acompanha apenas o andamento da ingestão de mídia/metadados através de uma área separada do CMS administrativo do CMS.

## Restrições Obrigatórias

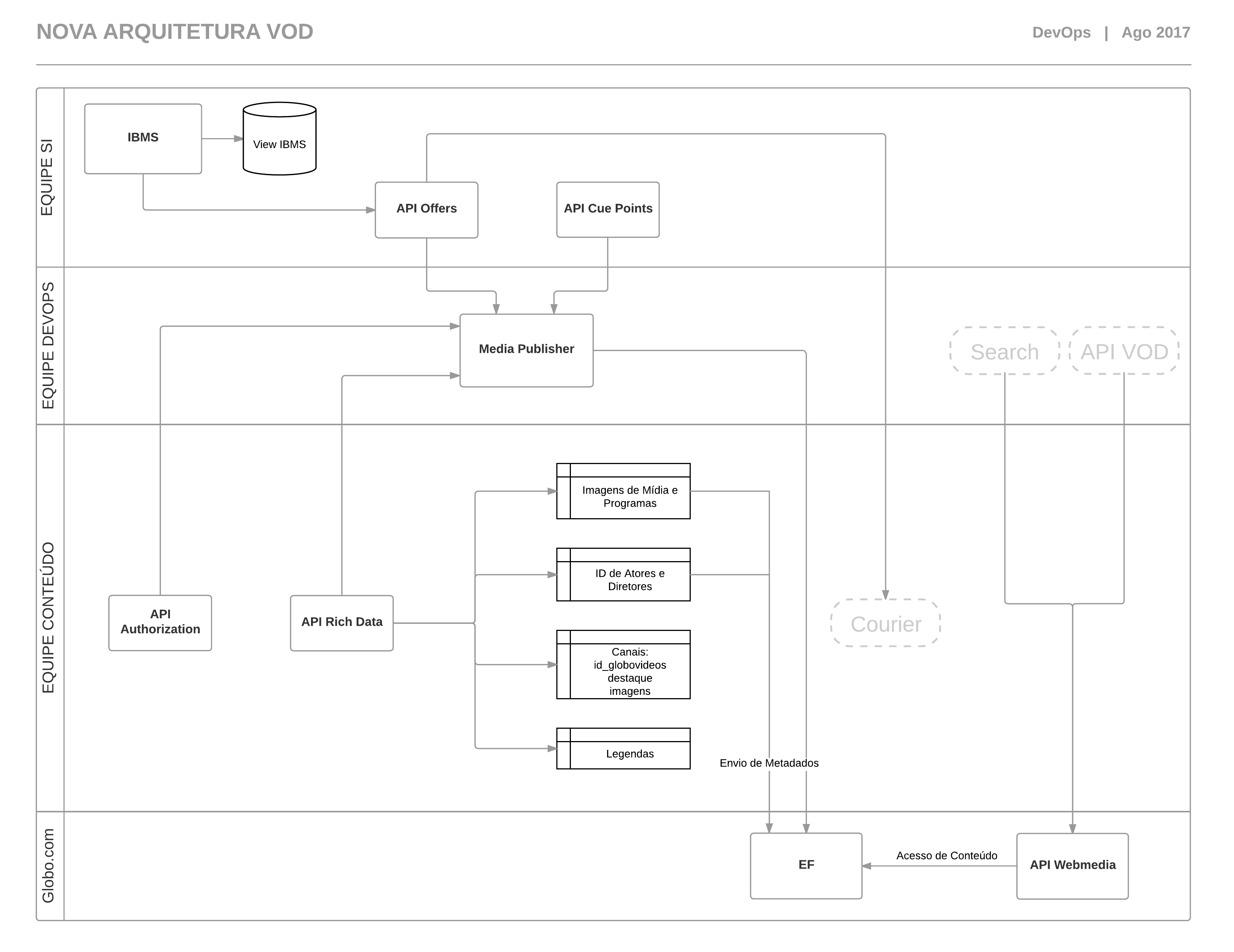
Este projeto visa através de uma arquitetura de microserviços solucionar os problemas do item 1.2. Qualquer outro objetivo deve ser visto como objetivo secundário e não abordado nesse projeto. Seguindo este conceito temos as seguintes restrições definidas:

* Persistência de dados: Hoje o CMS possui dados cadastrados e lógicas associadas e esses dados precisam mater a consistência na nova arquitetura. Esta consistência não pode ser traduzida em persistência. Ou seja, toda solução para a Nova Arquitetura não deve possuir persistência de metadados de mídia. Somente persistência de execução, monitoramento e controle serão aceitas.
* Uso do sistema: Os usuários do CMS precisam manter o mesmo nível de controle de dados que possuem atualmente. Embora algumas atividades sejam automatizadas, todas as atividades devem possuir logs que registram os status e um aparato ferramental de controle.
* Aplicações de terceiros: Temos aplicações de terceiros que precisam manter a interface ou pelo menos acesso a mesma API provida pelo CMS. O aspecto fundamental é a manutenção de contratos da API.

# Proposta de Nova Arquitetura

## Arquitetura

A seguir vamos propor a Nova Arquitetura VOD. Nesta proposta não vamos especificar nenhum requisito de usabilidade, operação e/ou legal.

Nessa arquitetura temos as seguintes aplicações:

* **API Offers**: Aplicação que traduz as ofertas cadastradas no IBMS e disponibiliza os dados através de uma API REST para acesso. É importante lembrar que esta API deve possibilitar filtros de mídias por data de alteração (necessário para o ciclo de publicação de mídias) e ID de Mídia. Não possui interface administrativa.
* **API Cue Points**: Aplicação que retorna os dados de Cue Points por ID de Mídia. Também não possui interface administrativa.
* **API Authorization**: Aplicação de que gerencia os *status* de "aberta"/"fechada" da Mídia. Necessita de interface administrativa para gestão da Equipe de Conteúdo para agendamento. Gerencia o recurso:
  + Agendamento de Mídia:
    - Próxima data de abertura
    - Proxima data de fechamento
* **API Rich Data**: Aplicação de enriquecimento de dados da Mídia. Responsável por adicionar metadados que não são disponibilizados na API de Ofertas pelo IBMS. Necessita de interface para gestão pela Equipe de Conteúdo dos seguintes recursos:
  + Metadados de Programas:
    - Nome
    - Slug
    - Imagem
    - Associação de IDs EF (também conhecido por *id\_globo\_videos*)
  + Metadados de Mídias:
    - Imagens
    - Legendas
    - Categorias
    - Sinopse Alternativa
  + Cadastro de Atores e Diretores:
    - Associação de IDs EF (também conhecido por *id\_globo\_videos*)
  + Metadados de Canais:
    - Associação de IDs EF (também conhecido por *id\_globo\_videos*)
    - Destaque
    - Imagens
* **Media Publisher**: Aplicação de orquestração de dados e envio de Mídias e metadados para a Encoding Factory. Responsável pela criação do fluxo de leitura dos dados do IBMS, enriquecimento de metadados na API Rich Data, adição de Cue Points pela API de Cue Point e a ingestão dos dados para a EF.
  + Metadados de Cue Points:
    - Mídia
    - Estágio

## Requisitos Funcionais

Todas as aplicações deve ter o seguintes requisitos funcionais:

* + **Log de Ações**: As aplicações deve registrar todas as principais etapas de processamento registrando os seguintes valores:
    - Objeto alterado e com referência no sistema
    - Ação realizada pelo processamento
    - Messages que identifica os valores processados
    - Nível hierárquico da mensagem
    - Data de execução da ação
  + **Interfaces:**
    - **Interface do Usuário**: As aplicações que possuem interação de controle ou visualização de dados pelo usuário deve possui uma interface Web.
    - **Interface de Aplicação**: As aplicações que interagem com outras aplicações devem fazê-la de através de uma API pública através de transações de protocolo HTTP utilizando o conceito de REST.
  + **Autenticação de Usuários**: A aplicação deve contar com um registro principal de usuários que permita a criação, edição e remoção de um cadastro de usuário através de login via email e senha.
  + **Controle de Estágio**: Cada aplicação deve conter o registro dos estágios em andamento e visualização do *workflow* de processamento. Ou seja, a cada etapa do processamento, o usuário deve ser capaz de identificar em qual etapa de processamento a atividade está. Esse controle é independente do Log de Ações mas usualmente está associado a ele. É desejável a cada etapa do *workflow* a criação de um registro de Log da Ação e registro do estágio de processamento.

## Requisitos Não-Funcionais

Os requisitos não-funcionais descrevem necessidades da Nova Arquitetura que não são limitadas pelas funcionalidades das Aplicações mas que precisam ser executadas.

Temos os seguintes requisitos não-funcionais:

* + **Monitoramento**: O monitoramento das Aplicações deve ser feita através de ferramentas online como *Newrelic*.
  + **Processamento em *background***: Qualquer interação entre Aplicações da Nova Arquitetura deve ser realizada de forma assíncrona, em *background* e gerenciável através de uma interface que proporcione um controle mínimo das ações. É desejável o uso de filas de processamento para que esse controle seja transparente e efetivo.
  + **Testes automatizados**: Toda aplicação deve conter uma suite de testes automatizados que forneçam confiabilidade das ações de cada aplicação. É desejável a metrificação da cobertura de testes.

# Features

Nesta seção estamos interessados em descrever as atividades e fluxos que caracterizam cada aplicação assim como suas principais *features*. Abordaremos somente as atividades *core* as *features*. Subentende-se que as atividades não-funcionais como descritas em 3.3 não serão descritas.

## Feature: Publicação de Cue Points

Segue abaixo a descrição da atividade de criação da atividade de Publicação de Cue Points.

### Descrição

A aplicação Media Publisher é responsável por gerenciar o envio de Cue Points para a Encoding Factory. Nele há a uma tarefa agendada a cada 10min que requisita para a Offer API as últimas ofertas alteradas. A partir de cada Oferta, Media Publisher requisita o valor *id\_midia* (também conhecido por *id\_globo\_videos*) necessário para a comunicação com a EF. Caso essa mídia não seja encontrada, e por consequência o *id\_globo\_videos*, o fluxo cessa para essa mídia.

O 3º passo da Media Publisher é requisitar na API de Cue Points os dados de Cue Points de fato (tempo e tipo), processar essas informações e montar o payload necessário para o request para o envio à EF.

Caso qualquer uma dessas etapas seja mal sucedida, o fluxo para e o cue point não é enviado. É importante lembrar que o envio de Cue Point das mídias é feito individualmente para cada mídia, de forma assíncrona e em *background*.

## API VOD v3: Disponibilização de Mídias

A API VOD é a API onde as aplicações clientes se conectam para exibir ao usuário final os dados de mídias.

### Descrição

Atualmente temos a versão 2 atendendo as aplicações clientes e a nova versão v3, manterá as funcionalidades básicas;

* + **Listagem**: Lista todos os items de um recurso em particular: *programas*, *episódios*, *filmes*, *shows* e *canais*. É possível listar utilizando filtros disponibilizados pela API Webmedia. A paginação dos dados, quando necessária, terá seu formato definido durante a implementação da API.
  + **Recuperação**: Acesso a um item específico através do id fornecido.

Não iremos manter o contrato de acesso aos *endpoints*. Também não haverá persistência de informações e portanto não haverá área administrativa. Não haverá nenhum tipo de processamento de imagens ou arquivos associados às mídias. A base de dados para *programas*, *episódios*, *filmes*, *shows* e *canais* é a API Webmedia onde os dados são enriquecidos pela API Rich Data.