Comunicado Técnico 128

ISSN 1677-8464
Dezembro, 2017
Campinas, SP

Gerenciamento de APIs com o WSO2 API Manager

Daniel Rodrigo de Freitas Apolinário[1]
Jorge Luiz Corrêa[2]
Isaque Vacari[3]
Luis Eduardo Gonzales[4]
Helano Póvoas de Lima[5]
Silvio Roberto Medeiros Evangelista[6]

Nos últimos anos, com a expansão das tecnologias digitais, conduzida principalmente pelo uso intensivo de dispositivos móveis, temos presenciado um aumento significativo na criação e no uso de *Application Programming Interfaces* (APIs) web. O crescimento da economia de APIs tende a acelerar a entrega de software pelas empresas e promover o surgimento de novos negócios. Por isso, as APIs têm sido um dos pilares que sustentam o crescimento na agilidade dos negócios (STURM et al., 2017).

Em recente relatório intitulado *O Estado das APIs no Brasil 2017* (SENSEDIA, 2017), realizado pela Sensedia em parceria com o *International Data Group* (IDG) Now, das empresas entrevistadas, 75% enxergam APIs como uma forma de integrar e agilizar processos internos, 70% a utilizam com o intuito de expandir sua proposta de valor, 55% tem uma estratégia de APIs para criar um ecossistema de parceiros e 50% entendem como uma ferramenta para fomentar a inovação. Ou seja, atualmente, é inegável a importância da utilização de APIs num mundo cada vez mais digital.

É frequente se ouvir que APIs são a "cola" do mundo digital. Isso porque as APIs têm se tornado a principal arquitetura para a comunicação entre sistemas. Na área da computação, desde o surgimento do *Remote Procedure Call* (RPC), passando por *Common Object Broker Architecture* (Corba) e *Service Oriented Architecture* (SOA) tem sido grande a pesquisa e a preocupação na comunicação entre softwares de diferentes plataformas e linguagens. O surgimento do estilo arquitetural *Representational State Transfer* (Rest), das tecnologias de desenvolvimento para

[1] *Cientista da computação, especialista em Plataformas de Desenvolvimento Web, analista da Embrapa Informática Agropecuária, Campinas, SP.*
[2] *Cientista da Computação, mestre em Ciência da Computação, analista da Embrapa Informática Agropecuária, Campinas, SP.*
[3] *Tecnólogo em Processamento de Dados, mestre em Ciência da Computação, analista da Embrapa Informática Agropecuária, Campinas, SP.*
[4] *Engenheiro de computação, analista da Embrapa Informática Agropecuária, Campinas, SP.*
[5] *Cientista da Computação, mestre em Ciência da Computação, analista da Embrapa Informática Agropecuária, Campinas, SP.*
[6] *Estatístico, doutor em Engenharia Elétrica, analista da Embrapa Informática Agropecuária, Campinas, SP.*

Embrapa

aplicativos móveis somados ao avanço e expansão das tecnologias de bandas largas de internet possibilitaram que as APIs Web se tornassem um padrão para conexão de aplicativos, dispositivos e empresas (STURM et. al., 2017).

Uma vez que as APIs Web têm sido essenciais para as empresas e seu número cresce cada vez mais, algumas questões relacionadas ao desenvolvimento e manutenção de APIs começam a surgir, tais como segurança, controle de acesso, controle de tráfego, monetização etc. Há alguns aspectos não funcionais de APIs que são comuns a todas elas e por este motivo foram desenvolvidas ferramentas especializadas no gerenciamento de APIs. Um gerenciador de API é uma parte de uma plataforma de API que, segundo Biehl (2015), é uma infraestrutura para desenvolver, executar e gerenciar APIs. O gerenciamento de APIs garante documentações bem elaboradas, acessibilidade a terceiros, regulação e disponibilidade de tráfego e versionamento (KOPECKÝ et al., 2014).

Este trabalho se propõe a apresentar instalação, configuração e testes da ferramenta WSO2 API Manager (WSO2, 2017a), que é uma das principais ferramentas de gerenciamento de APIs de código aberto disponíveis no mercado. Com o objetivo de ajudar a compreensão do leitor, este trabalho está dividido nas seguintes seções:

Apresentação do WSO2 API Manager

O WSO2 API Manager possui um conjunto de funcionalidades para a gestão de APIs tais como: a) criação; b) publicação; c) gerenciamento do ciclo de vida; d) versionamento; e) monetização; f) governança etc (WSO2, 2017b). Essa ferramenta é composta por vários componentes, cada um com uma responsabilidade definida no processo de gestão das APIs. A Figura 1 apresenta os componentes e seus relacionamentos, sendo que na próxima subseção teremos um resumo das responsabilidades dos principais componentes.

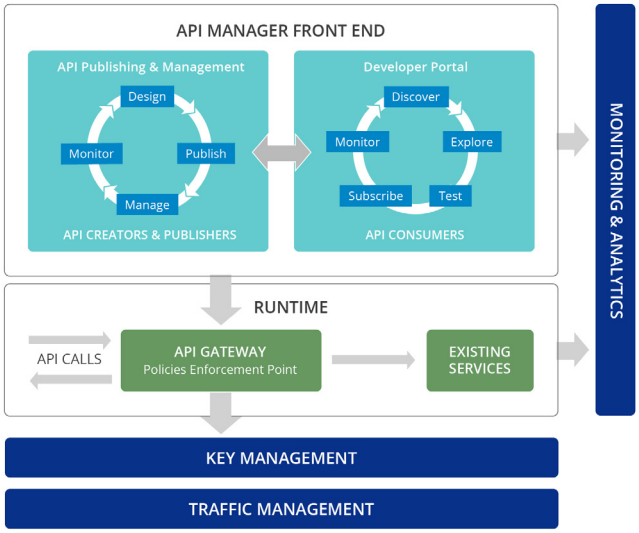

Figura 1. Visão de alto nível dos componentes do WSO2. Fonte: WSO2 (2017c).

Principais Componentes

- **API Gateway**: O componente de Gateway é o único ponto de acesso a todas as APIs, no qual são interceptadas todas as chamadas às APIs, geradas todas as estatísticas e aplicadas às políticas de *throttling* (regulação de tráfego) e de segurança. É um componente de *backend*, ou seja, não possui uma interface gráfica para acesso. Ele recebe as requisições de um consumidor externo, realiza as chamadas para os *endpoints* do *backend* que tratará de fato a requisição e depois retorna ao consumidor o resultado da execução da API.

- **API Publisher**: Este componente provê suporte ao desenvolvimento de APIs, possuindo uma interface web gráfica que permite que várias formas de documentação e de cadastro de metadados das APIs sejam realizadas. Os provedores de APIs podem utilizar este componente para criar, publicar e gerenciar o ciclo de vida das APIs. Algumas estatísticas de uso também estão disponíveis neste componente para acompanhamento de utilização das APIs pelos consumidores.

- **API Store**: Também pode ser conhecido ou chamado de Portal do Desenvolvedor, este componente é uma interface web gráfica responsável pela hospedagem e divulgação de APIs. Por meio deste componente, os consumidores (desenvolvedores) se registram, descobrem, se inscrevem e podem testar e avaliar as APIs. É também neste componente que o usuário gerencia a autenticação com os tokens de acesso às APIs.

- **Key Management**: Componente responsável por gerenciar a segurança e todas as operações relacionadas aos *tokens* de acesso às APIs. Tanto a geração quanto a validação de *tokens*, enviados pelos clientes de APIs, são realizadas por este componente. Todos os *tokens* são baseados no protocolo OAuth 2.0.0. Portanto, a comunicação entre os componentes *Key Management* e *API Gateway* é de extrema importância por causa das políticas de segurança.

Visão Geral do funcionamento da ferramenta

A Figura 2 apresenta uma visão de como funciona o atendimento de uma requisição de uma API que foi publicada pelo WSO2 *API Manager*. A requisição pode ser tanto para o ambiente de produção quanto para o ambiente de *sandbox*[7]. Na arquitetura single node do WSO2, a instância implantada é do tipo *all-in-one* e, nesse caso, há um só gateway que atende tanto as requisições de produção quanto as de sandbox. No entanto, a instalação exibida na Figura 2 segue uma arquitetura de implantação dos componentes um pouco diferente. Nessa implantação as chamadas dos ambientes de produção e *sandbox* são separadas, que é uma configuração possível e que consta na documentação da própria ferramenta. Neste caso, o API Publisher publica as APIs no respectivo gateway, de acordo com o ambiente selecionado no momento da publicação.

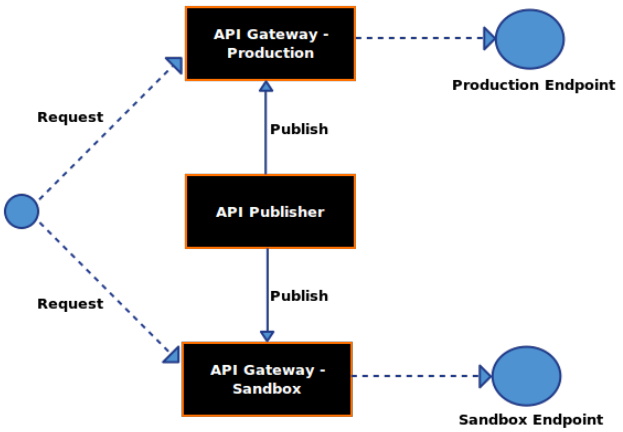

Figura 2. Visão geral de uma requisição a uma API utilizando o WSO2.
Fonte: WSO2 (2017d).

Instalação e configuração do WSO2 API Manager no projeto AgroAPI

Motivações

O projeto AgroAPI é uma iniciativa da Embrapa Informática Agropecuária de criação de uma plataforma para a disponibilização de dados e serviços em forma de APIs. O intuito é de promover a inserção da Empresa Brasileira de Pesquisa Agropecuária (Embrapa) na economia de APIs. Constitui-se numa ação para o estabelecimento de uma estratégia de APIs para a Empresa. Para tanto, uma parte essencial dessa plataforma é um software gerenciador de APIs. Devido ao fato de ser de código aberto e de ser considerado uma das principais e melhores ferramentas desse tipo por instituições como Gartner (MALINVERNO; O'NEILL, 2016) e Forrester (HEFFNER, 2014), o WSO2 API Manager foi a ferramenta escolhida pela equipe do projeto para ser o gerenciador de APIs dessa plataforma.

Arquitetura de implantação

Visando à garantia do fornecimento de um serviço com alta disponibilidade, uma série de tecnologias foi utilizada no estabelecimento da infraestrutura computacional onde o WSO2 é executado. Dentre elas estão o uso de virtualização em *cluster* com Xen[8], *proxy* reverso para um primeiro nível de controle de acesso, geração de estatísticas e gerência de certificados SSL e servidores de banco de dados com replicação assíncrona.

A Figura 3 ilustra essa arquitetura. Os acessos a partir da internet passam primeiramente por um sistema de *proxy* reverso em alta disponibilidade. Esse sistema é responsável por servir todas as URLs de serviços: a)*Store*; b)publicador; c) parte de administração; d) APIs em produção; e)parte de *sandbox*, redirecionando o tráfego corretamente para as máquinas virtuais correspondentes. Algumas dessas *Uniform Resource Locator* (URLs) são controladas, como a parte administrativa, por exemplo. O *proxy* também é responsável pela certificação *Secure Socket Layer* (SSL) de todas as URLs, garantindo que todo tráfego seja transferido criptografado com certificados válidos. Por fim, o

[7] *Ambiente sandbox é um ambiente idêntico ao de produção, porém com dados totalmente separados. Utilizado pelos desenvolvedores consumidores para testar o retorno das APIs*
[8] *Software livre de virtualização para as arquiteturas x86, x86-64, IA-32, IA-64 e PowerPC. O Xen permite a execução de diferentes sistemas operacionais, simultaneamente, em um mesmo hardware.*

proxy gera indicadores de acessos em um sistema de estatísticas web para fins administrativos e de negócios.

A infraestrutura de *cluster* Xen é um conjunto de servidores que compõem um *cluster* de virtualização capaz de executar máquinas virtuais. Este *cluster* permite que, na ocorrência de falhas de hardware, as máquinas virtuais do hardware que falhou sejam migradas automaticamente para nós com o funcionamento correto, conferindo alta disponibilidade do sistema. Ele é responsável por hospedar os três servidores virtuais do WSO2.

Esses servidores virtuais do WSO2 foram organizados da seguinte forma: a) o primeiro é responsável pela execução dos componentes Loja (*Store*), Publicador (*Publisher*), e Admin (Administração); b) o segundo executa o Gateway de APIs, sendo responsável por receber acessos às APIs que estejam em produção; c) o terceiro também executa um Gateway de APIs, porém fica responsável por receber acessos no sistema de *sandbox*, ou seja, APIs para a realização de testes. Essa arquitetura prevê que as requisições realizadas para o ambiente de testes das APIs não interfiram na performance das APIs já em produção.

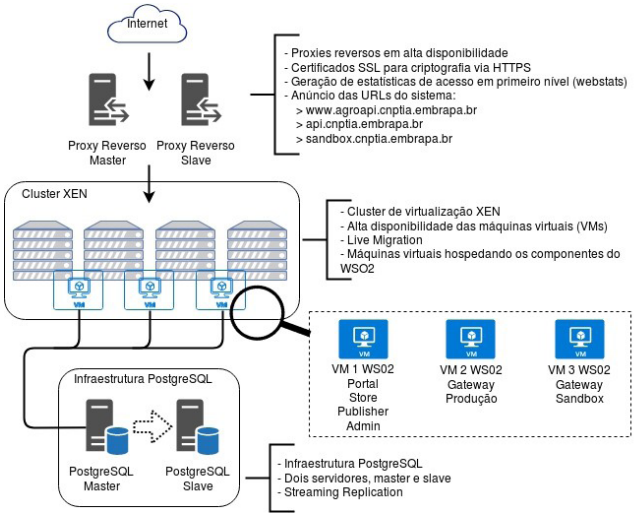

Figura 3. Visão de Arquitetura de Implantação

Instalação e Configuração

Nesta seção serão descritos os principais passos de instalação e configuração de uma maneira resumida. Detalhes de cada passo foram registrados no documento *Guia de construção de ambiente* (em fase de elaboração)[9] no âmbito do projeto AgroAPI.

Primeiramente, foi feito o download da versão 2.1.0 do *WSO2 API Manager*. Por se tratar de uma ferramenta implementada na linguagem Java, é necessária a instalação do *Oracle Java SE Development Kit* (JDK) na versão 1.7 ou superior e ter a variável de ambiente JAVA_HOME configurada. O processo de instalação é muito simples, bastando descompactar o arquivo baixado em alguma pasta na máquina escolhida como servidor. Como o servidor possui o sistema operacional Linux, executamos os passos contidos na documentação para habilitar o WSO2 API Manager como um serviço Linux a fim de que ele possa iniciar automaticamente no boot do sistema operacional.

Por padrão, o *WSO2 API Manager* usa um banco de dados H2 (banco de dados em memória) que já vem pronto e configurado para uso. Porém, em ambiente de produção é altamente aconselhável criar bancos de dados em algum Sistema Gerenciador de Bancos de Dados (SGBD) que seja suportado pela ferramenta. No nosso caso, criamos bancos de dados no servidor PostgreSQL e fizemos as configurações necessárias para que o *WSO2 API Manager* pudesse apontar para esse SGDB.

Dependendo das configurações de rede do servidor que executará o *WSO2 API Manager* deve ser levada em consideração as configurações de *proxy server*. Essas configurações podem ser alteradas no arquivo de configuração *axis2.xml* pelas propriedades *http.proxyHost, http.proxyPort e http.nonProxyHosts*, de acordo com a Figura 4.

[9] *Agroapi - guia de construção de ambiente, de autoria de Daniel Rodrigo de Freitas Apolinário e Jorge Luiz Corrêa, a ser editado pela Embrapa informática Agropecuária, [2018?].*

```
<!-- ================================================= -->
<!--           Transport Outs (Senders)                -->
<!-- ================================================= -->

<transportSender name="http" class="org.apache.synapse.transport.passthru.PassThroughHttpSender">
   <parameter name="non-blocking" locked="false">true</parameter>
   <parameter name="http.proxyHost" locked="false">endereço do servidor de proxy, exemplo:
"proxy.empresa.com.br"</parameter>
   <parameter name="http.proxyPort" locked="false">número da porta do servidor de proxy. Exemplo:
"3128"</parameter>
   <parameter name="http.nonProxyHosts" locked="false">lista de endereços de servidores internos
separados por '|'. Exemplo: "localhost|.*.empresa.com.br|127.0.0.1"</parameter>
</transportSender>
```

Figura 4. Trecho do arquivo <APIM-HOME>/repository/conf/axis2/axis2.xml que contém a configuração de *proxy server*.

Para a configuração de isolamento dos *gateways* de produção e *sandbox*, devem ser alteradas as definições de ambiente que constam no arquivo *api-manager.xml* em todas as instâncias da arquitetura de implantação do *WSO2 API Manager*. No exemplo da AgroAPI, temos três servidores, cada um com uma instância da ferramenta, portanto as duas instâncias responsáveis pelos gateways terão configurações iguais, enquanto a terceira instância receberá configurações um pouco diferentes. As duas primeiras instâncias chamaremos de *gateways* e a terceira instância, que é responsável pela execução dos componentes *Store, Publisher, Key Management* e Admin, chamaremos de instância principal.

Na Figura 5, apresentamos trechos do arquivo *api-manager.xml* alterados para a instância principal. A configuração realizada nesta instância define os dois gateways disponíveis para a criação e publicação das APIs e também define o endereço do *thrift server*[10] que nesse caso é o endereço da máquina que executa a instância principal.

Já na Figura 6, apresentamos os trechos dos arquivos *api-manager.xml* das instâncias que executam os componentes *gateways* do *WSO2 API Manager*. Basicamente, as configurações alteradas se referem ao apontamento do endereço da instância principal como sendo o gerenciador de autenticação e de validação de *tokens* de acesso (componente Key Management).

Portanto, com essas configurações apresentadas nas Figuras 5 e 6 habilitamos nossa arquitetura para o funcionamento dos *gateways* de maneira independente (produção e *sandbox*) e definimos uma instância principal na qual os demais componentes da solução (P*ublisher, Store, Key Management*, *Admin*) serão executados.

[10] *Thrift é o protocolo de comunicação entre o gateway e o componente key manager para propósitos de validação e autenticação dos tokens de segurança de acesso às APIs.*

```
...

<APIGateway>
   <!-- The environments to which an API will be published -->
   <Environments>
      <!-- Environments can be of different types. Allowed values are 'hybrid', 'production' and
'sandbox'.
      An API deployed on a 'production' type gateway will only support production keys
           An API deployed on a 'sandbox' type gateway will only support sandbox keys
           An API deployed on a 'hybrid' type gateway will support both production and sandbox keys. -->
           <!-- api-console element specifies whether the environment should be listed in API Console or not
-->
           <Environment type="production" api-console="true">
              <Name>Production Gateway</Name>
                  <Description>Production Gateway Environment</Description>

          <ServerURL>https://gateway01.embrapa.br:${mgt.transport.https.port}${carbon.context}services/</S
erverURL>
                  <Username>${admin.username}</Username>
                  <Password>${admin.password}</Password>
         <GatewayEndpoint>http://api.embrapa.br, https://api.embrapa.br</GatewayEndpoint>
           </Environment>
           <Environment type="sandbox" api-console="true">
         <Name>Sandbox Gateway</Name>
                  <Description>Sandbox Gateway Environment</Description>

          <ServerURL>https://gateway02.embrapa.br:${mgt.transport.https.port}${carbon.context}services/</S
erverURL>
                  <Username>${admin.username}</Username>
                  <Password>${admin.password}</Password>
         <GatewayEndpoint>http://sandbox.embrapa.br,
https://sandbox.embrapa.br</GatewayEndpoint>
           </Environment>
   </Environments>
</APIGateway>

'''

<APIKeyValidator>

   '''
   <EnableThriftServer>true</EnableThriftServer>
   <ThriftServerHost>instPrincipal.embrapa.br</ThriftServerHost>
   <!--ThriftServerPort>10397</ThriftServerPort-->

   '''
</APIKeyValidator>
```

Figura 5. Trecho do arquivo <APIM-HOME>/repository/conf/api-manager.xml responsável pelas configurações da instância principal.

```
...
<AuthManager>
   <!-- Server URL of the Authentication service -->

<ServerURL>https://instPrincipal.embrapa.br:${mgt.transport.https.port}${carbon.context}services/</ServerU
RL>
   ...
</AuthManager>
...
<APIKeyValidator>
   <!-- Server URL of the API key manager -->

<ServerURL>https://instPrincipal.embrapa.br:${mgt.transport.https.port}${carbon.context}services/</ServerU
RL>
   ...
   <EnableThriftServer>false</EnableThriftServer>
   <ThriftServerHost>instPrincipal.embrapa.br</ThriftServerHost>
   ...
</APIKeyValidator>
...
<OAuthConfigurations>
   ...
   <!-- This the API URL for revoke API. When we revoke tokens revoke requests should go through this API
deployed in API gateway. Then it will do cache invalidations related to revoked tokens. In distributed
deployment we should configure this property in key manager node by pointing gateway https( /http, we
recommend users to use 'https' endpoints for security purpose) url. Also please note that we should point
gateway revoke service to key manager -->
   <RevokeAPIURL>https://instPrincipal.embrapa.br:${https.nio.port}/revoke</RevokeAPIURL>
   ...
</OAuthConfigurations>
...
```

Figura 6. Trecho do arquivo <APIM-HOME>/repository/conf/api-manager.xml responsável pelas configurações das instâncias *gateways* (produção e *sandbox*).

As solicitações de assinatura de APIs, feitas pelos desenvolvedores consumidores, são automaticamente aprovadas na configuração padrão da ferramenta. Por isso, no projeto AgroAPI, foi realizada uma customização no *workflow* de assinaturas de APIs para permitir que uma solicitação de assinatura para um plano comercial (pago) inicie um fluxo de aprovação conduzido por um usuário com perfil administrativo na ferramenta, ou seja, a assinatura fica "aguardando" por uma aprovação ou rejeição e, enquanto isso não ocorre, o usuário não pode invocar a API pretendida. Esta customização pode ser feita de várias maneiras, inclusive podendo incluir integração com um sistema de pagamento, porém, no caso da AgroAPI optamos por realizar uma customização mais simples. O processo para que um *workflow* seja customizado é descrito na documentação[11] do próprio WSO2 API Manager. Na Figura 7, mostramos a classe Java que implementa a customização realizada no projeto AgroAPI.

[11] *Os passos para a customização do workflow de assinatura de API é disponível em https://docs.wso2.com/display/AM210/Adding+an+API+Subscription+Workflow*

```
...
import org.wso2.carbon.apimgt.impl.workflow.SubscriptionCreationWSWorkflowExecutor;
...

public class SubscriptionAgroAPI extends SubscriptionCreationWSWorkflowExecutor {
...
   @Override
   public WorkflowResponse execute(WorkflowDTO workflowDTO) throws WorkflowException {
      // Obtém o plano de assinatura solicitado pelo usuário
     SubscriptionWorkflowDTO subsCreationWFDTO = (SubscriptionWorkflowDTO)workflowDTO;
     String tierPlan = null;
           try {
              Tier tier = APIUtil.getTierFromCache(subsCreationWFDTO.getTierName(),
subsCreationWFDTO.getTenantDomain());
              if(tier != null){
                 tierPlan = tier.getTierPlan();
              }
     } catch (APIManagementException e) {
              e.printStackTrace();
     }
           // Se o plano da assinatura for do tipo comercial (pago), o workflow
           // de aprovação será acionado. Caso contrário, ou seja, o plano de assinatura
           // for gratuito (free), a assinatura é automaticamente aprovada
     if(APIConstants.COMMERCIAL_TIER_PLAN.equals(tierPlan)){
              super.execute(workflowDTO);
     } eles {
              workflowDTO.setStatus(WorkflowStatus.APPROVED);
              complete(workflowDTO);
           }

           return new GeneralWorkflowResponse();
   }
}
```

Figura 7. Implementação de classe que customiza *workflow* de assinatura de API.

O *WSO2 API Manager* suporta a configuração de *user stores* diferentes para a autenticação em suas ferramentas. Por isso, também fizemos a configuração de adicionar um LDAP *Server* como uma *user store* adicional e, com isso, todos os usuários cadastrados no *Lightweight Directory Access Protocol* (LDAP) da Embrapa possuem acesso para autenticação no *WSO2 API Manager*.

O *WSO2 API Manager* é um software que possibilita uma grande quantidade de customizações, possibilitando que se adapte a vários ambientes e contextos de uso e, por isso, há uma quantidade muito grande de configurações que não foram explorados no exemplo do projeto AgroAPI. No entanto, citamos a seguir algumas outras configurações possíveis que foram testadas e/ou consultadas:

- Criação de grupos de APIs para ajudar na busca e categorização.

- Configuração de *tenants*[12] para que o controle das APIs possa ter algum nível de "departamentalização".

- Customização do layout da interface gráfica da *API Store.*

- Possibilidade de criação de um portal de desenvolvedor, a partir de qualquer outra tecnologia, usando as APIs disponíveis pelo próprio WSO2 para obter as informações disponíveis na API Store.

- Customização de *workflows* para vários tipos de ações como cadastro de usuário, assinatura de APIs, criação de aplicações, geração de tokens de acesso etc.

[12] *Um* tenant *em WSO2 é um nível negocial separado para agrupar papéis, tais como: departamento, grupos ou qualquer outro domínio lógico.*

Para que sejam possíveis o armazenamento e também a consulta de estatísticas relacionadas ao uso das APIs, é necessário a instalação da ferramenta *WSO2 API Manager Analytics*. O processo de instalação[13] dessa ferramenta é semelhante ao do *API Manager*. Além disso, o *WSO2 API Manager* deverá ser habilitado para funcionar em conjunto com o Analytics. São aplicações independentes, mas, quando instaladas e habilitadas, trocam mensagens entre si para que as estatísticas das APIs possam ser armazenadas e consultadas pelos usuários.

Em caso da customização de *workflows* para alterar o comportamento de etapas do gerenciamento de APIs, também é necessário que seja instalada a ferramenta *WSO2 Business Process Server*, que é uma ferramenta de modelagem de processos de negócio e que provê o gerenciamento de *workflows*.

Limitações e Dificuldades

Documentação

A documentação existente disponibilizada na internet pela própria WSO2 cobre os principais aspectos e conceitos da ferramenta assim como as possibilidades de configurações diferentes suportadas pelo WSO2 API Manager.

No entanto, principalmente se considerarmos configurações mais avançadas, podemos notar escassez e até inexistência de exemplos funcionais para alguns cenários de customização. Não conseguimos encontrar um exemplo funcional de uma integração com sistema de billing, embora em algumas outras documentações encontradas na internet tenha sido sugerido que haviam exemplos dessa integração disponíveis para versões anteriores da ferramenta. Também podemos citar que na implantação do projeto AgroAPI, numa arquitetura de separação de *gateways* de produção e *sandbox*, houve alguns problemas cuja solução não constava na documentação da ferramenta. Para as configurações de *proxy server* e também tivemos dificuldade por falta de clareza e exatidão da documentação relacionada a este assunto.

É notável o esforço da empresa em manter uma documentação atualizada por versão de suas ferramentas, no entanto, também se percebe uma certa confusão detectada nas configurações de diferentes versões de um mesmo produto e, para o usuário final, às vezes é difícil distinguir o que é configuração de componentes de código aberto que fazem parte da solução *WSO2 API Manager* das configurações do produto final. Um exemplo disso é que para algumas configurações do *WSO2 API Manager* tivemos que recorrer a documentação do *WSO2 CARBON*, que é a base na qual o API Manager e outras ferramentas da WSO2 são construídas. Os links existentes entre essas documentações não foram claros o suficiente para explicar essas dependências.

***Enpoints* externos à rede Embrapa**

Quando uma API cuja URL do *endpoint* é externa à rede da Embrapa, temos problemas para a execução das APIs. A configuração ocorre normalmente sem problemas, porém ao invocar a API foi detectado um erro na chamada. Depois de alguma investigação, foi descoberto que o WSO2 não consegue "montar" a URL inteira para *endpoints* externos à rede Embrapa e, portanto, o WSO2 tenta invocar uma URL relativa que não existe na web, ocasionando o erro. Entendemos que o erro acontece devido ao cenário de configurações de rede da empresa em conjunto com as configurações próprias do WSO2 e, por isso, a solução para este problema foi difícil de ser encontrada. Conseguimos encontrar em um *blog* de usuário que passou por problema semelhante e a solução foi alterar uma configuração no XML que representa uma API criada pela ferramenta da WSO2. Essa solução não é a ideal, pois é necessário alterar de maneira manual um arquivo XML depois dela ter sido criada na ferramenta WSO2, mas é a única que resolveu o problema detectado.

URLs de endpoints com certificado digital

Devido a aspectos de segurança, é comum que APIs estejam disponíveis no protocolo HTTPS e, por isso, podem existir APIs que estejam se utilizando de certificados digitais que não pertençam às cadeias autorizadoras oficiais. Quando isso acontece em um web site, o browser emite um alerta avisando do ocorrido e pergunta se mesmo assim o usuário deseja continuar acessando aquele

[13] O processo de instalação do WSO2 API Manager Analytics pode ser consultado em: https://docs.wso2.com/display/AM210/Configuring+APIM+Analytics

site. No cenário de gerenciadores de APIs, esse certificado deve ser conhecido no servidor no qual as APIs estão invocadas, no caso o servidor que roda o API Gateway precisa conhecer esse certificado para que possa invocar a API e obter o seu retorno com sucesso. Para isso, é necessário que, caso seja criada uma API cujo *endpoint* tenha um certificado digital de uma cadeia autorizadora não oficial e, sendo esta segura (algo que deve ser garantido pelos criadores e gerenciadores da API), o certificado deve ser importado pelo administrador da plataforma de gerenciamento de APIs, pois será necessário ter acesso *root* nas máquinas servidoras que hospedam o *WSO2 API Manager*. Sem essa configuração, os consumidores da API não conseguirão ter sucesso ao invocá-la.

Conclusão

De maneira geral, o *WSO2 API Manager* é uma ferramenta muito poderosa para o gerenciamento de APIs. Contém todas as funcionalidades mais importantes que atualmente são requiridas de um software para esta finalidade. O fato de ser uma ferramenta de código aberto é algo importante para empresas que desejam iniciar uma estratégia de APIs mas que não dispõem de muitos recursos para serem investidos. Pode-se dizer que para a atualização de *patches* e suporte da WSO2 à ferramenta, é necessário assinar uma subscrição paga anualmente e que, portanto, para seu uso em produção é altamente recomendado a contratação de suporte. A ferramenta se destaca pelo seu alto grau de customização e flexibilidade, que pode ser determinante para a adoção da ferramenta em cenários diferentes do comum. Grande parte das informações e funcionalidades do WSO2 estão disponíveis no formato de APIs também, o que dá liberdade para criar aplicações clientes diferentes das aplicações web disponíveis na própria ferramenta. Enfim, a instalação e testes no projeto AgroAPI mostrou que a ferramenta atende de maneira bastante satisfatória o gerenciamento de APIs, contribuindo bastante para o fortalecimento de uma estratégia de APIs interna e externamente a uma organização.

Referências

BIEHL, M. **API architecture:** the big picture for building APIs. [S.l.], 2015. p. 3. (API university series, v. 2).

HEFFNER, R. **The forrester wave TM**: API management solutions, Q3 2014. Cambridge: Forrester Research, 2014. Disponível em: <https://www.forrester.com/report/The+Forrester+Wave+API+Management+Solutions+Q3+2014/-/E-RES119266>. Acesso em: 27 out. 2017.

KOPECKÝ, J.; FREMANTLE, P.; BOAKES, R. **A history and future of Web APIs**. Information Technology, v. 56, n. 3, p. 90-97, 2014.

MALINVERNO, P.; O'NEILL, M. **Magic quadrant for full life cycle API management**. Stamford: Gartner, 2016. Disponível em: <https://myleadcorner.files.wordpress.com/2017/01/magic-quadrant-for-full-life-cycle-api-management-oct-2016.pdf>. Acesso em: 27 nov. 2017.

SENSEDIA. O Estado das APIs no Brasil 2017. Disponível em: <https://page.sensedia.com/pesquisa-o-estado-das-apis-brasil-2017/>. Acesso em: 27 out 2017.

STURM, R.; POLLARD, C.; CRAIG, J. Application programming interfaces and connected Systems. In: STURM, R.; POLLARD, C.; CRAIG, J. **Application performance management (APM) in the digital enterprise**: managing applications for cloud, mobile, IoT and ebusiness. Cambridge: Morgan Kaufmann, 2017. p. 137–150. DOI: 10.1016/B978-0-12-804018-8.00011-5>.

WSO2. **API management**. 2017a. Disponível em: <https://wso2.com/api-management/>. Acesso em: 27 out 2017.

WSO2. **API manager documentation**. 2017b. Disponível em: <https://docs.wso2.com/display/AM210>. Acesso em: 27 out 2017.

WSO2. **API manager front end**. 2017c. Disponível em: <https://docs.wso2.com/download/attachments/57743933/APIM-Overview.png?version=1&modificationDate=1479484494000&api=v2>. Acesso em: 27 out. 2017.

WSO2. **API production sandbox**. 2017d. Disponível em: https://docs.wso2.com/download/attachments/57743762/production-sandbox-publish.png?version=1&modificationDate=1479484488000&api=v2>. Acesso em: 27 out. 2017.

Comunicado Técnico,128

Embrapa Informática Agropecuária
Endereço: Av. Dr. André Tosello, 209 - Cidade Universitária, Campinas - SP
Fone: (19) 3211-5700
https://www.embrapa.br/informatica-agropecuaria

1ª edição publicação digital - 2017

Comitê de publicações

Presidente: *Giampaolo Queiroz Pellegrino*
Secretária-Executiva: *Carla Cristiane Osawa*
Membros: *Adriana Farah Gonzalez, Carla Geovana do Nascimento Macário, Flávia Bussaglia Fiorini, Ivo Pierozzi Júnior, Kleber X. Sampaio de Souza, Luiz Antonio Falaguasta Barbosa, Maria Goretti G. Praxedes, Paula Regina K. Falcão, Ricardo Augusto Dante, Sônia Ternes*
Suplentes: *Jayme Barbedo, Michel Yamagishi e Goran Nesic*

Expediente

Supervisão editorial: *Kleber X. Sampaio de Souza*
Revisão de texto: *Adriana Farah Gonzalez*
Normalização bibliográfica: *Maria Goretti G. Praxedes*
Editoração eletrônica: *Tuíra Santana Favarin, sob supervisão de Flávia Bussaglia Fiorini.*

CGPE 14231