

White Paper on

w2bill™ PAYMENTS



CONTEXTO
ARQUITECTURA
CAPACIDADES
ROADMAP

ÍNDICE

02 INTRODUÇÃO

03 CONTEXTO

04 ARQUITETURA

MOTOR DE PROCESSAMENTO

PLATAFORMA

MÉTRICAS

REGISTO

SERVIÇO DE DESCOBERTA E CONFIGURAÇÃO DE SERVIDOR

MONITORIZAÇÃO

INTERFACE DE UTILIZADOR

SERVIÇO DE AUTENTICAÇÃO

TECNOLOGIAS

PERSISTÊNCIA DE DADOS

INTEGRAÇÃO DE CANAIS

09 CAPACIDADES

10 ROADMAP

11 ANEXOS

PERSISTÊNCIA

APACHE CASSANDRA

ACESSIBILIDADE DE DADOS

APACHE IGNITE

MOTOR DE PROCESSAMENTO

JAVA 8

PIVOTAL SPRING FRAMEWORK

GESTÃO DE RECURSOS

APACHE MESOS

INTRODUÇÃO

À medida que surgem novos avanços a uma velocidade acelerada, impulsionados pela transformação desencadeada por smartphones e tablets, novas tecnologias, incluindo carteiras móveis e aplicativos on-demand, uma nova era de assistentes digitais e tecnologias de conectividade aprimoradas – como NFC e Bluetooth – estão a tornar-se uma parte intrínseca da maneira como os consumidores interagem com os produtos e serviços dos quais dependem todos os dias.

Os pagamentos móveis, o consumo on-demand, o aumento de tecnologias de 'near-field' e a economia partilhada estão, atualmente, a ajudar a moldar a maneira como usamos e pagamos no nosso dia a dia. E com esse status quo, os clientes esperam uma ampla variedade de opções quando estão prestes a pagar, com pouca ou nenhuma fricção.

Grandes marcas e plataformas de e-commerce têm acostumado os clientes a uma experiência de pagamento sem fricções, armazenando as suas credenciais, tais como dados de cartão de crédito e outras formas semelhantes de pagamento, como uma forma de acelerar toda a experiência de compra e garantir continuidade e comodidade a cada interação.

Com as credenciais armazenadas, tais como o método de pagamento preferido, resultam numa concorrência feroz entre startups de fintech, provedores de cartões de crédito e outros intermediários de pagamento para alcançar todo o tecido demográfico – especialmente o público mais jovem, lucrativo e mais apto digitalmente – e conquistar a sua preferência de método de pagamento para uso contínuo.

Quase todos os players de cada mercado sentem a necessidade de ajustar os seus canais de pagamento às necessidades e hábitos dos consumidores, de forma a garantir um tipo de experiência de pagamento sem fricções e em tempo real com que o mercado se tem vindo a habituar. Na era da "big data", o sucesso nesta área é amplamente medido pela capacidade de capturar pagamentos rápidos de contra-consumo, mais adequados ao cliente – loja física, mobile, desktop, tablet ou dispositivos wearable como smart watches.

A boa experiência que, por exemplo, as carteiras móveis conseguem oferecer em termos de velocidade e recompensa – a gamificação das compras – parece ser uma chave importante para impulsionar um consumo maior, acelerar a cobrança e gerar uma satisfação mais completa, já que o ato de pagar – ou o "check-out" – é mais fluido na experiência de compra.

CONTEXTO

A tecnologia supera-se constantemente ano após ano, com novos avanços a serem apresentados de forma incessante, alterando a maneira como as pessoas e as empresas fazem as coisas. A presença crescente da tecnologia em todo o lado, gera efetivamente uma dependência como nunca antes vista. As pessoas não vão a lado nenhum sem os seus smartphones, operam simultaneamente os seus tablets e portáteis, enquanto as empresas, ao mesmo tempo, dependem de infraestruturas complexas para operar e gerir efetivamente o seu negócio central. Pouco pode ser alcançado sem tecnologia, e qualquer empresa sem uma presença digital forte está condenada ao esquecimento.

Com tanta tecnologia, tantos utilizadores diferentes e uma grande diversidade de serviços e ofertas, a quantidade de dados e a complexidade da sua gestão, do ponto de vista de um fornecedor, tornam-se cada vez mais evidentes. A evolução tem sido marcada pela transição de aplicações monolíticas para aplicações modulares, e a partir daí, para sistemas discretos, específicos em cada domínio, interconectados por arquiteturas de integração complexas.

As empresas agora moldam os seus modelos de negócio flexíveis, inovadores e modernos, especialmente em função da pressão que colocam nos custos operacionais.

Desde start-ups até às grandes empresas, a chave para a prosperidade parece estar na capacidade de acompanhar as tendências de mercado, resistir às constantes mudanças e competir com a maior simplicidade possível.

Para cumprir com todas as expectativas reais do mercado, as empresas devem ser capazes de mudar, transformar-se e escalar as suas atividades e infraestruturas de suporte – tudo em nome da experiência do utilizador e do benefício mútuo.

Cada aspecto ou dimensão que suporta o modelo de negócio, especialmente a infraestrutura de TI, precisa de ser flexível, escalável, ágil, eficiente e com um tempo de implementação o mais reduzido possível. As empresas procuram adaptabilidade, modularidade e a capacidade de acomodar mudanças futuras.

Este facto comprovado tem vindo a criar a necessidade de soluções de integração que consigam automatizar e escalar o desempenho nas atividades de suporte ao negócio principal, reduzindo a longa e exaustiva dor de cabeça que é ligar todas as pontas soltas a cada inovação, e passar dos sistemas centrais para a automatização e escalabilidade, com muito pouco tempo disponível para o fazer.

ARQUITECTURA

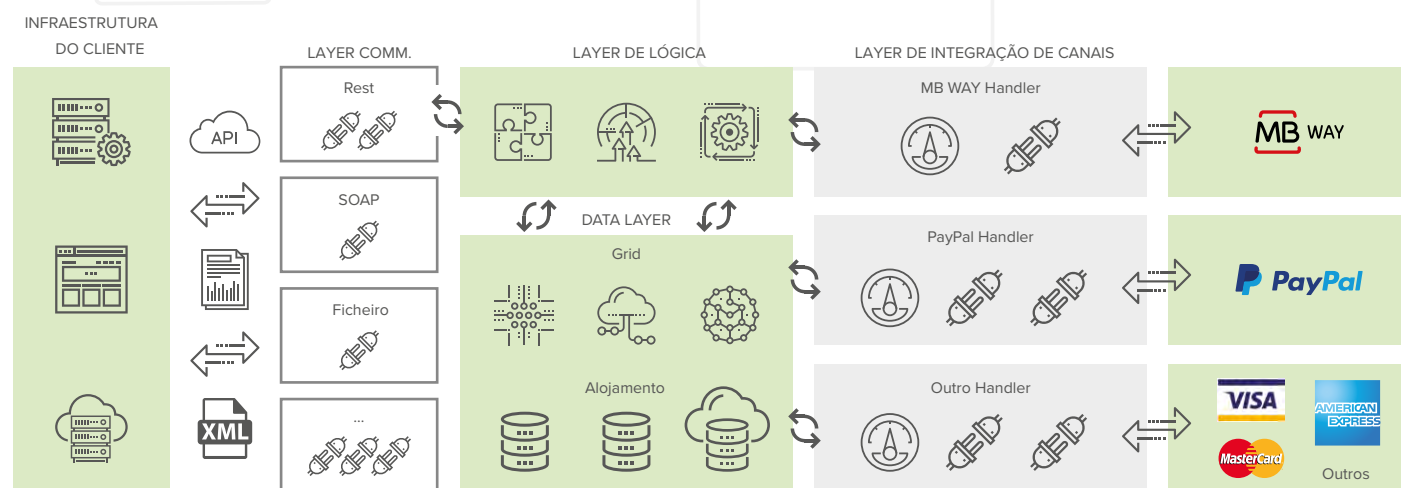


Figura 1. w2bill Arqitectura geral

O W2bill Payments foi concebido, desde as suas fases iniciais, com o futuro em mente. Seguir caminhos antigos fazia pouco sentido num mundo marcado por uma evolução constante e pela crescente certeza de que a solução de hoje estará, muito provavelmente, desatualizada num curto espaço de tempo. Com estas preocupações em mente, foram consideradas diferentes formas de alcançar os objetivos definidos, onde a escalabilidade, tolerância a falhas, confiabilidade, resiliência e configurabilidade foram elementos-chave.

O que é verdade hoje, amanhã estará ultrapassado. A quantidade de dados atualmente considerada aceitável será em breve ultrapassada.

Mais poder de processamento, em diferentes geografias, através de Clouds públicas ou Centrais de Dados privados, tem de ser suportado, tendo em consideração todos os desafios daí decorrentes.

Do ponto de vista funcional, o w2bill Payments incorpora a noção de Ordens de Pagamento, que permitem ao cliente simplificar a gestão das transações financeiras. Por exemplo, ao solicitar que um pagamento seja efetuado, o cliente cria uma Ordem de Pagamento dentro da plataforma, indicando não apenas o montante, mas também os canais nos quais pode ser pago, juntamente com qualquer informação adicional relevante.

À medida que as respostas são processadas, o status da Ordem de Pagamento evolui. Cada resposta é tratada de forma individual e não implica necessariamente que o seu valor corresponda ao valor original, ou seja, os pagamentos parciais para qualquer Ordem de Pagamento são automaticamente geridos, assim como os estornos parciais acionados pelo cliente final.

Além disso, como as transações financeiras podem ser geridas pelas Ordens de Pagamento, existe um controlo incorporado para garantir que a mesma transação não seja solicitada duas vezes enquanto está em progresso, reforçando ainda mais as validações para evitar cenários erróneos.

Associada à Ordem de Pagamento está a noção de validade, que permite ao cliente gerir a forma como deseja que as suas transações financeiras sejam tratadas.

Como exemplo, é possível definir que uma dada Ordem de Pagamento deve ser paga nas primeiras 5 tentativas de pagamento; caso contrário, é automaticamente cancelada.

Os três conceitos fundamentais por trás do W2bill Payments são:

1. O MOTOR DE PROCESSAMENTO

O W2bill Payments incorpora um Motor de Processamento, que permite a gestão de tarefas complexas.

O Motor de Processamento é um pacote de micro-serviços orientado para a execução de fluxos de tarefas em diversas plataformas. Cada fluxo pode ser totalmente configurado, permitindo que tarefas sejam executadas de forma sequencial ou paralela. Apresenta três classes principais de componentes: Receptor, Orquestrador e Executor.

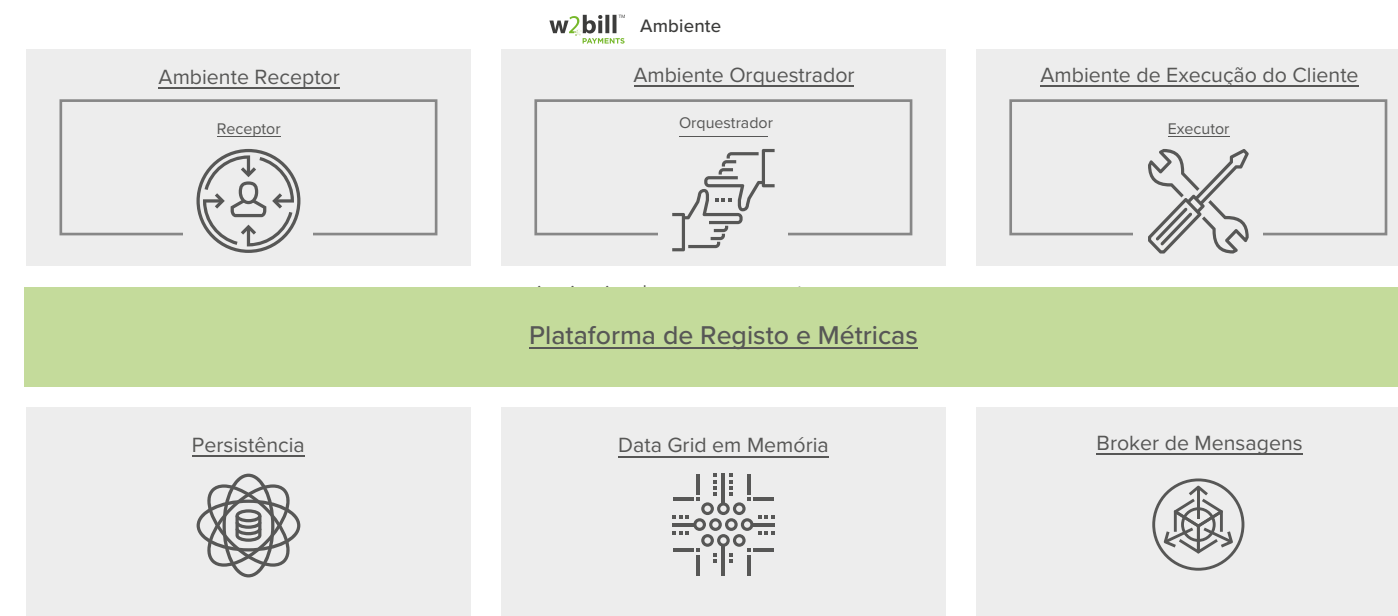


Figura 2. w2bill Visão Geral do Motor de Processamento

1.1 RECEPTOR

O componente Receptor é o principal ponto de entrada de qualquer pedido, permitindo o seu processamento inicial, o qual inclui validações técnicas bem como regras de negócio configuráveis, aplicadas com o objetivo de garantir a coerência da informação e determinar qual o fluxo a executar com base nos dados recebidos.

Caso todos os passos sejam validados com sucesso, uma ordem é colocada no sistema de filas e é devolvida uma resposta à plataforma que fez o pedido.

Por outro lado, caso ocorra uma falha, a plataforma de origem é informada do erro, de forma a poder proceder à sua correção e eventual novo envio.

O Receptor pode disponibilizar vários serviços (endpoints) para serem invocados, juntamente com múltiplas versões de cada endpoint, que são geradas e expostas dinamicamente com base na configuração definida pelo utilizador.

Adicionalmente, suporta múltiplas instâncias em execução em diferentes máquinas, cada uma com o seu próprio conjunto de serviços, permitindo uma granularidade e controlo mais apurados.

1.2 ORQUESTRADOR

Para os pedidos aceites com sucesso, o componente Orquestrador supervisiona o controlo do fluxo de tarefas a serem executadas, desde o início até ao fim – seja com sucesso ou em erro. Ele acompanha a evolução de cada tarefa, agendando o seu trabalho dentro dos componentes Executor disponíveis, com base nas suas capacidades e no trabalho necessário a ser realizado.

A abordagem selecionada foi de um serviço sem estado e descentralizado, sendo capaz de dividir a gestão de cada fluxo entre todas as instâncias em execução.

Todas as interações entre os componentes ocorrem através de mensagens. A tecnologia de broker de suporte garante a persistência das mensagens, alta disponibilidade, confirmação de entrega (entrega pelo menos uma vez), tolerância a falhas, agrupamento de mensagens, entre outros. Esta abordagem permite que o motor escale verticalmente, aproveitando múltiplos núcleos por máquina, e escale horizontalmente, lançando múltiplas instâncias num cluster de máquinas.

1.3 EXECUTOR

O componente Executor é responsável pela execução real das tarefas dentro de cada fluxo de trabalho. Este componente foi projetado para auto-escalabilidade dentro dos recursos disponíveis, além de ser implantável em múltiplas instâncias para gestão e balanceamento de carga. Cada componente é dedicado a um domínio de trabalho, como interações com RDBMS, integrações com sistemas de arquivos, ou outras funcionalidades mais especializadas dependendo dos requisitos e restrições do cliente.

Imaginando um cenário onde um cliente já está a operar com um conjunto limitado de operações em lote, onde as transações financeiras são realizadas através do envio e obtenção diária de arquivos para e a partir dos bancos (ou outras instituições financeiras). O cliente identifica como crucial a implementação de um canal em tempo real, como o PayPal. O novo canal é implementado por meio de um conjunto dedicado de componentes, os quais não afetam a operação existente. Novos fluxos são configurados, novas funcionalidades são ativadas, mas a plataforma continua a executar normalmente.

2 PLATAFORMA

2.1 MÉTRICAS

Todos os componentes são implementados para produzir métricas de execução e armazená-las num repositório centralizado, onde o desempenho do sistema pode ser medido e monitorizado. Com essas métricas, é possível saber quanto tempo uma ordem leva para ser processada, assim como o tempo que cada componente leva para executar a operação requerida.

Com essas capacidades, é possível detectar anomalias nas instâncias dos componentes, monitorizando os tempos de execução de cada um no sistema, juntamente com outras informações relevantes necessárias para agendar ações de melhoria de desempenho, detecção de bottlenecks ou decisões de escalabilidade.

As métricas são enriquecidas com dados contextuais. É possível saber o host onde o componente está a ser executado, o número de núcleos e memória disponíveis, a que grupo em pertence em tempo real, além de outras informações relevantes.

Como exemplo, um cliente começa a perceber que o processamento de solicitações em tempo real frequentemente está a atingir timeouts, o que causa a necessidade de reprocessamento das solicitações ou até mesmo a insatisfação por parte dos clientes finais.

Ao analisar as métricas, é possível identificar que uma máquina foi modificada e está agora a operar além da capacidade planeada de 30%. A partir disso, podem ser realizadas ações para adotar uma abordagem diferente de escalabilidade, permitindo que os problemas sejam resolvidos.

2.2 REGISTOS

Em implementações distribuídas, as instâncias dos componentes serão espalhadas por várias máquinas de um cluster. Cada instância irá gerar registos (logs) que devem ser armazenados e agrupados com os registos de todos os componentes do sistema, permitindo a análise do comportamento do sistema. Todos os componentes do w2bill Payments estão preparados para gravar os registos da aplicação num ficheiro – na máquina local – e para enviar uma mensagem de registo com dados adicionais, permitindo aos utilizadores correlacionar todos os registos para uma ordem específica ou execução de fluxo de trabalho. Isto proporciona uma visão mais clara do sistema.

Esses registos são centralizados num índice do ElasticSearch com um esquema pré-definido. É possível pesquisar os registos por um id de ordem, um id de correlação ou por mensagem de registo. Os registos são publicados de forma assíncrona no broker, melhorando o desempenho do sistema.

2.3 SERVIÇO DE DESCOBERTA & SERVIDOR DE CONFIGURAÇÃO

As instâncias dos componentes podem ser iniciadas em qualquer host disponível no cluster, manualmente ou automaticamente, com base na carga de trabalho. Para ser possível gerir e controlar todas as instâncias, foi implementado um Registo de Serviços e um Servidor de Configuração.

Todos os componentes retiram a sua configuração do servidor de configuração central e devem criar um registo – com o seu endereço IP e porta HTTP – no Registo de Serviços. Cada instância de serviço é responsável por registar-se no Registo de Serviços durante a inicialização e anular de registo durante o fecho.

2.4 MONITORIZAÇÃO

As instâncias dos componentes são monitorizadas continuamente através de endpoints de verificação de estado fornecidos por cada um deles e pela análise das métricas. Com isso, é possível agir, iniciar e parar instâncias, mantendo o Nível de Serviço desejado e o uso eficaz dos recursos do ambiente em execução.

Esta capacidade está associada à funcionalidade de Métricas para monitorizar continuamente e reagir a mudanças na infraestrutura, planeando ações para mitigar os impactos.

2.5 FRONTEND

O w2bill payments apresenta uma interface gráfica unificada que permite a realização de operações comuns sobre a Plataforma, como diagnóstico, rastreamento, validação de logs e verificações de estado. A abordagem escolhida permite que cada cliente tenha ecrãs ou vistas dedicadas, representando informações relevantes a serem exibidas, em vez de um ecrã estático mostrando dados pré-definidos. Também incorpora perfis e funções configuráveis, associados a cada utilizador, definindo ainda mais o que cada utilizador pode visualizar, criar, modificar ou remover, com base nos requisitos organizacionais.

O Frontend é baseado na tecnologia Angular JS, permitindo um visual e interação atualizados, bem como interatividade responsiva a dispositivos. Tanto as informações exibidas quanto o visual do Frontend são específicos para cada cliente. O racional é que faz mais sentido adaptar o Frontend à imagem corporativa do cliente e à forma de trabalhar, e não o contrário.

Dessa forma, será possível, por exemplo, monitorizar o volume de transações processadas diariamente, por canal, e exibir essa informação no Frontend para os utilizadores com os privilégios adequados para visualizar.

2.6 SERVIÇO DE AUTENTICAÇÃO

Para cumprir com a necessidade de segurança e acesso validado aos serviços, o w2bill payments incorpora um serviço de autenticação que permite que os serviços sejam utilizados apenas por usuários devidamente autenticados.

Com esta funcionalidade, todas as requisições feitas aos serviços expostos devem ser autenticadas através de um servidor de autenticação central – como fornecido pelo w2bill payments, garantindo que apenas os serviços aos quais o usuário tem permissões concedidas sejam acessados.

Este servidor implementa o protocolo OAuth 2.0 e pode autenticar utilizadores numa base de dados relacional ou num servidor LDAP, dependendo das necessidades do cliente.

Esta capacidade fortalece significativamente a organização na proteção de acesso às suas transações financeiras, assegurando que apenas utilizadores válidos e autenticados possam interagir com o sistema.

2.7 TECNOLOGIAS

Toda a solução é baseada no conjunto de tecnologias apresentada abaixo em mais detalhe. Em qualquer caso, dado as decisões da arquitetura tomadas anteriormente, é totalmente possível optar por soluções específicas utilizando tecnologias diferentes, desde que certos aspectos-chave de integração sejam atendidos.

Como exemplo, é possível implementar um componente Executor implementado em C/C++, Microsoft C#, ou até mesmo Python. A escolha no grupo de Persistência pode ser alterada por outras soluções NoSQL, como MongoDB, ou até mesmo sistemas de gestão de base de dados relacionais mais tradicionais, como MySQL, Postgres ou Oracle.

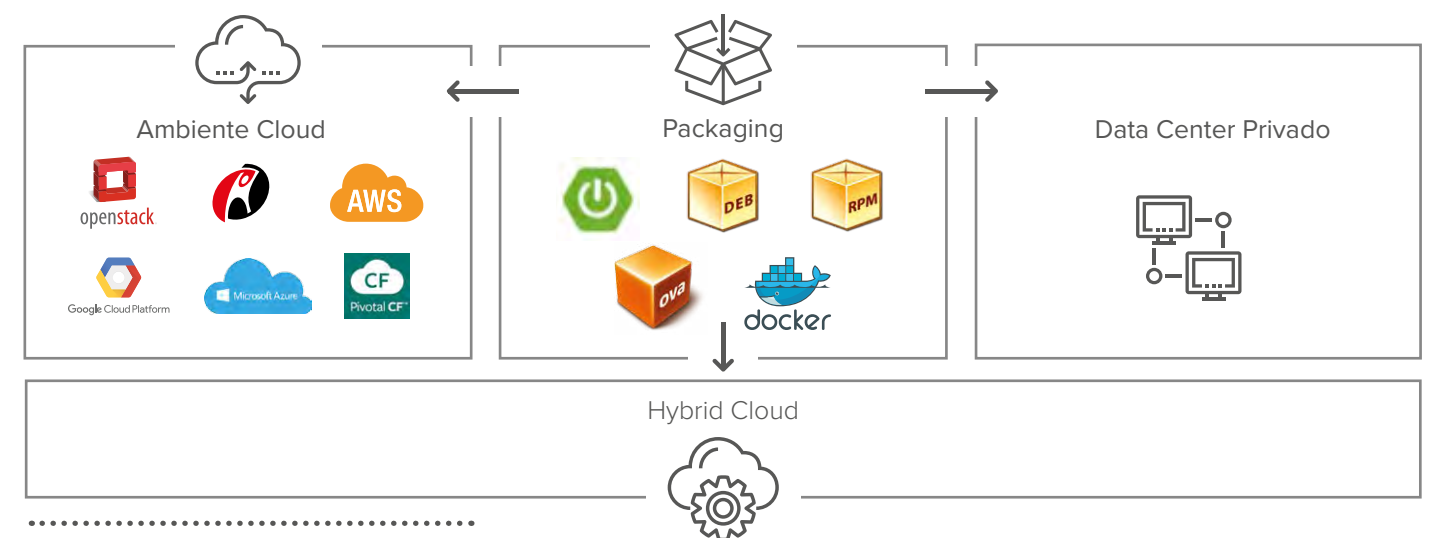


Figura 3. w2bill Packaging para múltiplos deployments

É espectável que certos fornecedores prefiram tecnologias com protocolos mais rigorosos, focadas no desempenho a nível de bits, enquanto outros preferem abordagens mais leves, completamente diferentes entre si. Soluções herdadas não são prejudicadas, nem prejudicam as mais recentes, quando a integração com o w2bill payments é equacionada.

Como mencionado anteriormente, esta capacidade pode ser utilizada para o cliente usar a sua própria implementação de canais com lógica proprietária, que não deve ser implementada diretamente na plataforma, mas beneficia da sua utilização.

Em toda a solução, a escolha da tecnologia foi focada na sua adequação (para o conjunto específico de responsabilidades), bem como no seu histórico comprovado, adoção generalizada na indústria, suporte empresarial fiável e roadmap de evolução.

Com a crescente adoção de abordagens baseadas em Cloud, em vez de soluções on-premise, certos aspectos da arquitetura, design e implementação subjacente foram considerados para garantir que a solução estivesse preparada sem a necessidade de personalização específica. Ao abordar estas preocupações desde o início, a seleção das tecnologias foi refinada, ficando apenas as que apresentavam níveis apropriados de conformidade elegíveis.

3. PERSISTÊNCIA DE DADOS

É sabido que os dados devem ser mantidos de forma segura, consistente e protegida, para serem utilizados. Dado que o âmbito da solução envolve o manuseio de transações financeiras e a interação com plataformas internas

para reconciliações de capital, é fundamental que nenhuma informação seja perdida. Os dados devem ser monitorizados e protegidos contra manipulações indesejadas.

Além disso, devem ser facilmente escaláveis e ter o menor número de restrições possível na sua gestão. Devido à complexidade de critérios, não existe uma solução direta ou abordagem clara sobre que tecnologia será melhor para garantir a persistência.

Plataformas tradicionais de RDBMS são soluções comprovadas, que oferecem uma grande variedade de capacidades que a maioria das pessoas considera garantidas, mas apresentam falhas em termos de custo, escalabilidade ou desempenho. Soluções mais recentes de NoSQL têm uma adoção menos ampla em grandes empresas, mas normalmente são escaláveis e eficientes, a uma fração do custo. Além disso, existem abordagens mais próximas ao armazenamento físico real, como HDFS, ZFS, entre outras.

Para lidar com a persistência, o w2bill payments adotou uma abordagem de "escolha qualquer uma". Em outras palavras, seja qual for a tecnologia mais adequada para um cliente, ou para um domínio específico dentro de um cliente, a solução irá suportá-la, incluindo estratégias mistas com dados segregados entre plataformas. Para compreender melhor os seguintes conceitos, é relevante entender como o w2bill é projetado, particularmente como os seus componentes interagem. O w2bill é estruturado como um conjunto de micro-serviços, cada um responsável por um determinado domínio de ações. Com esta abordagem, torna-se possível adicionar novos componentes ou remover antigos, sem comprometer a consistência de toda a solução.

Esta arquitetura modular é alcançada, pelo modelo dos componentes como micro-serviços independentes – ou aplicações – capazes de interagir entre si, com vários serviços de integração fornecendo funcionalidade para esse fim.

O Serviço de Configuração e o Serviço de Descoberta permitem que os componentes se registem e configurem a si mesmos, possibilitando a sua interação com toda a plataforma quando estes forem 'conectados'.

4. INTEGRAÇÃO DE CANAIS

A estratégia anterior é um passo importante para a integração rápida e fácil de múltiplos canais. Seguindo estas diretrizes de integração da plataforma, novos micro-serviços direcionados à lógica e gestão de canais específicos podem ser incorporados sem prejudicar a plataforma existente ou exigir grandes alterações na mesma.

Também é um factor facilitador chave para a maioria dos negócios e especificações técnicas frequentes. Um cliente pode ter os seus próprios canais personalizados ou querer realizar uma incursão rápida em novos mercados com diferentes regulamentos, ou até mesmo abraçar mudanças nos seus próprios mercados operacionais – novas tecnologias, novas leis, novas oportunidades de negócios.

Para qualquer um desses cenários, o processo de construção de micro-serviços pode variar desde uma simples configuração de micro-serviços existentes que oferecem uma vasta gama de capacidades adequadas para o novo propósito específico, até melhorias de serviços já existentes, ou, finalmente, implementação personalizada de lógica dedicada com novos micro-serviços.

A plataforma lida com qualquer uma dessas abordagens com perturbações limitadas. Os clientes podem crescer e adaptar-se sem temer a perda de serviço. Também está garantido que integrações específicas do cliente, para protocolos privados ou proprietários, podem ser integradas com toda a solução sem perder o controlo do cliente. É possível, como exemplo, que um cliente tenha a sua própria implementação de um canal integrado com a solução w2bill payments. Ela aproveita as capacidades da plataforma, sem publicar ou abdicar do controlo da sua implementação privada.

CAPACIDADES

Resumo do que é fornecido com o w2bill:

- Processa ficheiros para processamento em lote offline, recebidos de instituições financeiras
- Gera ficheiros para processamento em lote offline, enviados para instituições financeiras
- Suporta pedidos de pagamento em tempo real e respetivo cancelamento
- Permite a confirmação e rejeição de pagamentos em tempo real
- Suporta modelos de comunicação síncronos e assíncronos, em tempo real
- Gera relatórios com base nas transações financeiras processadas
- Integra novos canais de pagamento (diferentes protocolos)
- Incorpora canais transacionais proprietários ou privados do cliente
- Suporta relatórios para contabilidade geral (General Ledger)
- Suporte a táticas de reconciliação de inserção-se-em-falta ou reporte-de-em-falta (em tempo real com processamento em lote)
- Tarefas e fluxos configuráveis
- Escalabilidade automática transversal (interna, vertical e horizontal)
- Design tolerante a falhas através de clusterização de componentes
- Sem ponto central de falha
- Registo (logging) com métricas de desempenho
- Verificação e monitorização do estado dos componentes
- Grelha de dados (data grid) para acesso e manipulação de dados com alto desempenho e suporte a ACID
- Persistência de dados em cluster, resiliente e distribuída
- Suporte a múltiplas soluções de persistência de dados (RDBMS, NoSQL)
- Comunicação entre componentes através de mensagens com garantia de entrega pelo menos uma vez (at-least-once delivery)

ROADMAP

ANEXOS

Alguns insights técnicos

Plenamente ciente de que o produto de hoje não será o produto de amanhã - uma vez que as necessidades e tecnologias atuais certamente serão substituídas no futuro — a solução w2bill payments possui um roadmap de funcionalidades em constante evolução. Essas novas capacidades surgem a partir da experiência acumulada com os nossos clientes, bem como os seus insights, conselhos e sugestões.

O objectivo não é manter ramificações específicas por cliente, paradas no tempo a cada implantação, mas sim

garantir uma evolução recorrente, voltada a oferecer novas e melhores formas de apoiar a tecnologia e os negócios.

Ao investigar constantemente o que está a acontecer e o que está para vir — além de agregarmos experiências entre mercados, clientes e mentalidades —, vislumbramos o caminho da melhoria contínua a ser disponibilizado. A parceria com os nossos clientes é essencial para a construção dessa visão e definição do roadmap.

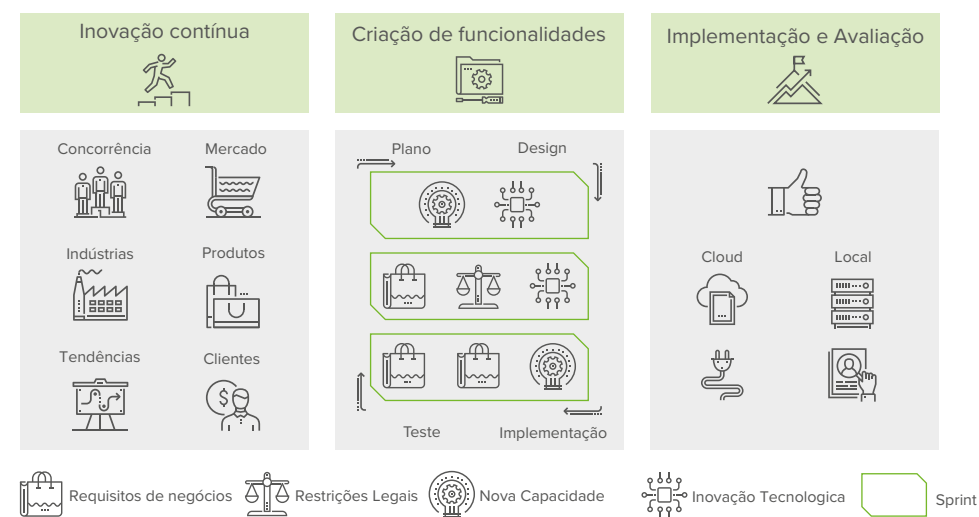


Figura 4. w2bill Abordagem do Roadmap

PERSISTÊNCIA

APACHE CASSANDRA

A escolha do Apache Cassandra para armazenamento foi baseada em sua multidão de capacidades, em especial:

- Natureza descentralizada, evitando pontos únicos de falha ou bottlenecks com origem na rede
- Tolerância a falhas, por meio da replicação de dados entre nós e data centers, além da capacidade de substituir nós com falha sem tempo de inatividade
- Escalabilidade, utilizando múltiplos nós "quase infinitos" para suportar o crescimento do processamento e armazenamento
- Elasticidade, baseando-se no aumento de nós para melhorar a taxa de leitura/gravação
- Suporte profissional e uso comprovado em grandes empresas globais (CERN, eBay, GitHub, Apple, Netflix, para citar algumas)

O framework w2bill payments depende da solução Apache Cassandra para armazenar suas informações entre seus componentes. O uso é indireto, no entanto, pois a camada de acessibilidade é realizada através do Data Fabric Apache Ignite.

É responsabilidade dessa camada comunicar efetivamente com o Apache Cassandra para o armazenamento eficaz (gravação) e leitura de dados.

Independentemente dessa escolha, como mencionado, a solução é projetada para interagir com múltiplas soluções de persistência de dados, com base nos requisitos específicos de cada cliente.

ACESSIBILIDADES DE DADOS

O Apache Ignite é uma *In-memory Data Fabric* que oferece grids de dados, computação e serviços de alto desempenho. Ele suporta transações distribuídas totalmente compatíveis com o padrão ACID, garantindo consistência em todos os dados e oferecendo suporte à sintaxe SQL padrão para consultar os objetos armazenados na grade de dados.

Atualmente, o w2bill payments utiliza a grade de dados como uma camada para interagir com a solução de persistência, utilizando abordagens de leitura e gravação (read-through ou write-through). A gravação efetiva é executada de forma assíncrona, para agilizar o desempenho.

Outra característica relevante é a avançada capacidade de clustering, que permite atender aos requisitos de escalabilidade, tolerância a falhas e alto desempenho.

Atualmente, o w2bill payments utiliza a grade de dados como uma camada para interagir com a solução de persistência, utilizando abordagens de leitura e gravação (read-through ou write-through). A gravação efetiva é executada de forma assíncrona, para agilizar o desempenho.

MOTOR DE PROCESSAMENTO

JAVA 8

O w2bill payments é desenvolvido utilizando a versão mais recente da linguagem Java. Isso permite aproveitar todos os novos recursos e melhorias de desempenho introduzidos com a versão 8.

O Java foi selecionado como linguagem de referência devido à sua ampla adoção na indústria, à simplicidade de implantação em múltiplas plataformas e à portabilidade do código.

O Java conta com um vasto número de plugins e frameworks open source, uma comunidade de developers extremamente ativa, um amplo roadmap de evolução e um sólido grupo de suporte.

PIVOTAL SPRING FRAMEWORK

Esta framework é utilizada como base para a construção dos serviços do w2bill payments.

Oferece uma ampla infraestrutura para desenvolvimento de aplicações Java corporativas, promovendo boas práticas de programação, como injeção de dependências, inversão de controle e programação orientada a aspectos.

O uso do Spring permite acelerar o desenvolvimento, garantir maior modularidade e facilitar a integração com outros componentes da plataforma, além de oferecer suporte nativo para desenvolvimento de aplicações baseadas em micro-serviços.

GESTÃO DE RECURSOS

APACHE MESOS

O w2bill Payments é um sistema naturalmente distribuído. Qualquer um de seus componentes e respectivas instâncias pode ser executado em múltiplas máquinas, contribuindo assim para um melhor desempenho por meio da escalabilidade horizontal, além de evitar pontos centrais de falha, ao utilizar técnicas de clustering e tolerância a falhas.

A implementação foi projetada para utilizar não apenas a virtualização comum de máquinas, mas também abordagens de containerização, como Docker ou Kubernetes.

A distribuição e o gerenciamento de recursos, portanto, são preocupações que foram devidamente tratadas por meio do uso do Apache Mesos, que oferece um controle centralizado para implantação e escalonamento dos componentes do w2bill Payments em qualquer tipo de instalação, seja on-premise, em nuvem ou em ambientes híbridos.



CMAS - Systems Consultants, Lda
Edifício Escritórios do Tejo
Rua do Pólo sul , N°2 , 1ªA
Parque das Nações
1990-273 Lisboa

mail@cmas-systems.com

www.cmas-systems.com

