

White Paper

# w2bill™ REALTIME



CONTEXTO  
ARQUITECTURA  
CAPACIDADES  
ROADMAP

# ÍNDICE

## 03 INTRODUÇÃO

## 04 CONTEXTO

## 05 ARQUITETURA

### 06 MOTOR DE PROCESSAMENTO

### 07 ACESSIBILIDADE DE DADOS

PERSISTENCIA

LOGGING/MÉTRICAS DA PLATAFORMA

MÉTRICAS

LOGGING

SERVIÇO DE DESCOBERTA E SERVIDOR DE CONFIGURAÇÃO

MONITORIZAÇÃO

## 08 INTERFACES

SERVIÇOS DE AUTENTICAÇÃO

TECNOLOGIAS

## 09 CAPACIDADES

## 10 ROADMAP

## 11 ANEXOS

MOTOR DE PROCESSAMENTO

JAVA 8

PIVOTAL SPRING FRAMEWORK

PIVOTAL RABBITMQ

## 12 GESTÃO DE RECURSOS

APACHE MESOS

ACESSIBILIDADE DE DADOS

APACHE IGNITE

PERSISTÊNCIA

APACHE CASSANDRA





# INTRODUÇÃO

Nos últimos anos, as necessidades e expectativas dos clientes cresceram significativamente, impulsionadas pelos avanços tecnológicos e pelo desejo cada vez mais presente de poder fazer tudo, em qualquer lugar e a qualquer momento. Isto levou as empresas a procurar formas de satisfazer estas exigências, oferecendo continuamente produtos e serviços inovadores, ao mesmo tempo que mantêm a capacidade de se adaptarem rapidamente, sem terem de alterar constantemente as suas infraestruturas ou plataformas.

Para responder às expectativas do mercado atual, as organizações devem ser capazes de evoluir, transformar-se e escalar as suas atividades e infraestruturas de suporte — tudo em nome de uma melhor experiência para o cliente, em que todos saem a ganhar.

Todos os aspetos ou dimensões que suportam o modelo de negócio, especialmente a infraestrutura de TI, precisam de ser flexíveis, escaláveis, ágeis, eficientes e com o menor tempo de implementação possível.

As empresas procuram adaptabilidade, modularidade e a capacidade de se prepararem para mudanças futuras. Este facto comprovado tem vindo a criar a necessidade de soluções de integração, capazes de automatizar e escalar o desempenho nas atividades de suporte ao negócio principal, reduzindo significativamente a longa e complexa tarefa de ligar todas as pontas soltas sempre que surge uma inovação, e facilitando a transição dos sistemas centralizados para soluções automatizadas e escaláveis, num curto espaço de tempo.



# CONTEXTO

O foco do mercado está no desenvolvimento de novos serviços e experiências, com base num automatismo mais rápido e profundo, análises preditivas, segurança e níveis de integração e parcerias cada vez mais profundos entre múltiplos intervenientes. Em todos os mercados, em todas as indústrias, tanto as expectativas dos produtores/comerciantes como as dos consumidores estão em constante transformação.

Os consumidores exigem cada vez mais uma entrega mais rápida e eficiente, interações e resultados on-demand, enquanto, aparentemente de forma incoerente, esperam custos e taxas de serviço mais baixos e flexíveis. Desde os consumidores até às grandes corporações, todos parecem concordar que a imediatidade está rapidamente a tornar-se numa capacidade básica, em vez de uma diferenciação. Enquanto o mercado a exige, os reguladores do mercado lutam para a acompanhar, e apenas as empresas mais ágeis e eficientes prosperam.

A tecnologia evolui constantemente, ano após ano, com novos avanços a serem apresentados de forma incessante, mudando a maneira como as pessoas e as empresas fazem as coisas. Cresce continuamente na sua presença em todos os lugares, criando efetivamente uma dependência como nunca antes vista.

As pessoas não vão a lado nenhum sem os seus smartphones, operam simultaneamente em tablets e portáteis, enquanto as empresas dependem de infraestruturas complexas para operar e gerir eficazmente o seu negócio principal.

Pouco pode ser alcançado sem tecnologia, e qualquer empresa sem uma forte presença digital está condenada ao esquecimento. Com tanta tecnologia em funcionamento, tantos utilizadores diferentes e uma tal diversidade de serviços e ofertas, a quantidade de dados e a complexidade da sua gestão, do ponto de vista do fornecedor, tornam-se cada vez mais óbvias.

A evolução tem sido marcada pela transição de aplicações monolíticas para aplicações modulares; daí para sistemas discretos específicos de cada domínio, interligados por arquiteturas complexas de integração.

O modelo para fazer negócios hoje em dia tem de ser flexível, adaptável, inovador e moderno, especialmente perante a pressão sobre os custos operacionais.

Desde a empresa startup até às grandes empresas, a chave para a prosperidade parece ser a capacidade de acompanhar as tendências de mercado, suportar as constantes mudanças e competir da forma mais ágil possível.

# ARQUITETURA

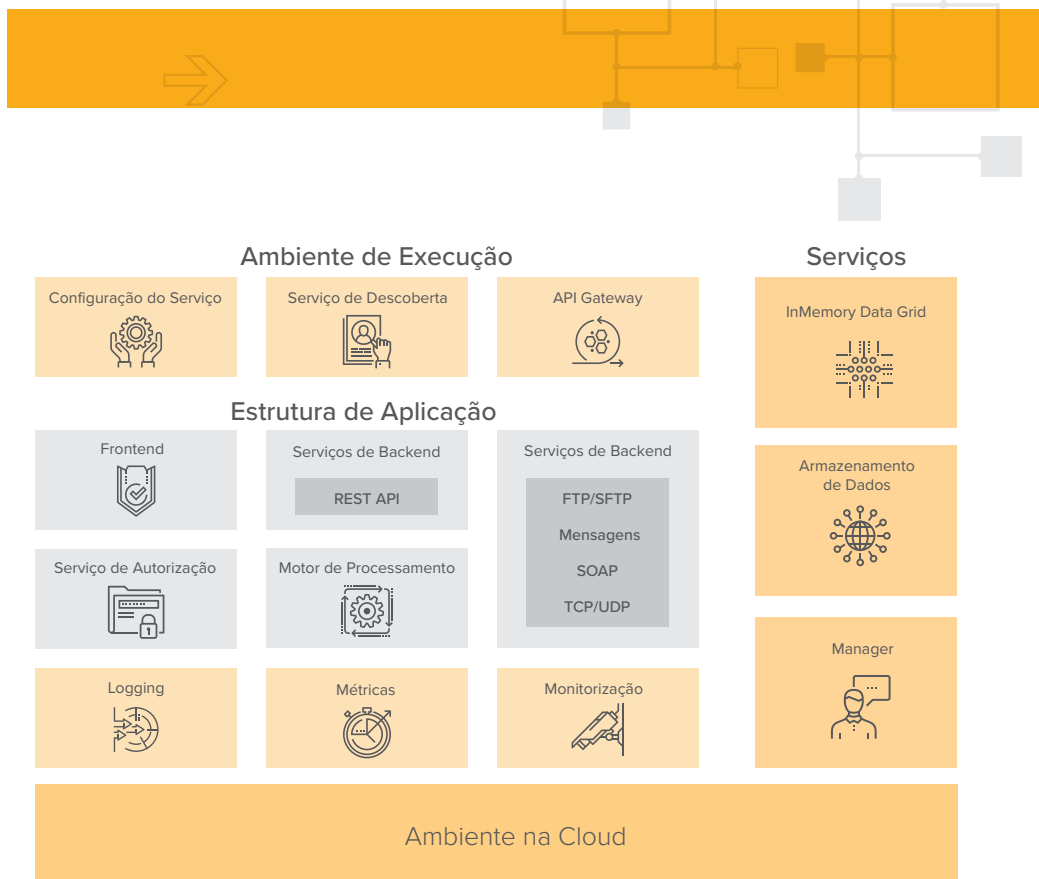


Figura 1. w2bill Visão Geral da Arquitetura

O Framework de Gestão de Tarefas em Tempo Real, w2bill Realtime, foi projetado, desde a sua fase inicial, tendo o futuro em mente. Seguir o caminho tradicional não fazia sentido num mundo marcado por uma constante evolução e com a crescente certeza de que as soluções de hoje provavelmente estarão desatualizadas cada vez mais rápido.

Com isso em mente, foram consideradas diferentes formas de fazer as coisas, onde escalabilidade, tolerância a

falhas, fiabilidade, resiliência e configurabilidade foram fundamentais. O que é verdade hoje em dia em breve estará ultrapassado e desatualizado. A quantidade de dados gerida atualmente será em breve ultrapassada.

Deve ser possível suportar mais poder de processamento em diferentes geografias, através de Clouds públicas ou Data Centers privados, com todos os desafios inerentes aos mesmos.

# Os três conceitos fundamentais do w2bill Realtime

## 1. MOTOR DE PROCESSAMENTO

O Motor de Processamento é o coração da solução w2bill realtime. É um conjunto de micro-serviços destinados a executar fluxos de tarefas através de uma variedade de plataformas.

Cada fluxo pode ser totalmente configurado, permitindo a execução de tarefas sequenciais ou paralelas.

Ele possui três classes principais de componentes: **Receptor, Orquestrador e Executor.**

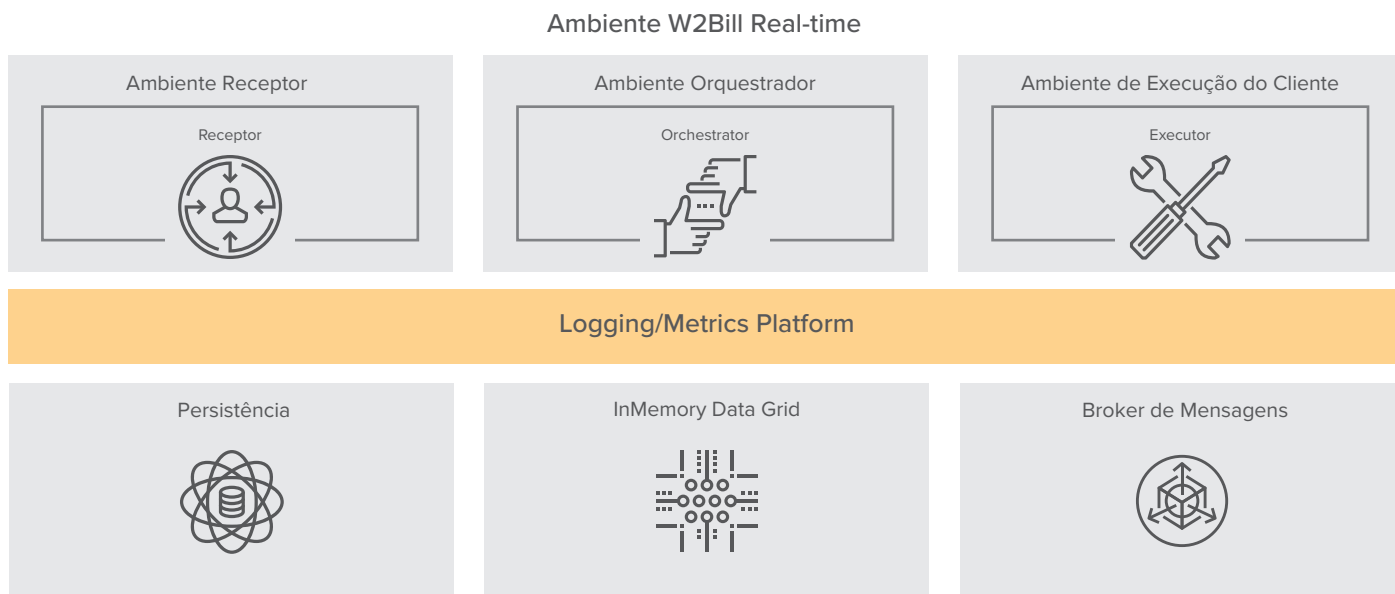


Figura 2. w2bill™ Visão Geral do Motor de Processamento

### 1.1 RECEPTOR

O componente Receptor é o ponto de entrada principal de qualquer solicitação, permitindo o seu processamento inicial, o que inclui validações técnicas, bem como regras de negócios configuráveis a serem aplicadas para verificar a coerência da informação e determinar que fluxo deve ser executado com base nos dados de entrada. Caso todas as etapas sejam válidas, uma ordem é colocada na lista de espera e uma resposta é devolvida à plataforma que fez o pedido. Por outro lado, caso haja uma falha, a plataforma que fez o pedido é informada do erro para correção e eventual reenvio. Um Receptor pode fornecer vários serviços (endpoints) a serem

invocados, juntamente com várias versões de cada endpoint, que são geradas dinamicamente e expostas com base na configuração do usuário.

### 1.2 ORQUESTRADOR

Para pedidos aceites com sucesso, o componente Orquestrador supervisiona o controlo do fluxo de tarefas a serem executadas, do início ao fim – com sucesso ou com erro. Acompanha a evolução de cada tarefa agendando o seu trabalho dentro dos componentes do Executor disponíveis, com base nas suas capacidades e no trabalho necessário a realizar. A abordagem selecionada foi a 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 componentes ocorrem por via de mensagens. A tecnologia de broker de suporte assegura persistência de mensagens, alta disponibilidade, confirmação na entrega (pelo menos uma entrega), tolerância a falhas, agrupamento de mensagens, entre muitas outras.

Esta abordagem permite ao motor escalar verticalmente, fazendo uso de múltiplos núcleos por máquina, e escalar horizontalmente, lançando múltiplas instâncias num cluster de máquinas.

### 1.3 EXECUTOR

O componente Executor é responsável pela execução real das tarefas de trabalho dentro de cada fluxo. Este componente foi concebido para autoescalar dentro dos seus recursos disponíveis, bem como, ser implementado em múltiplas instâncias para gestão e balanceamento de esforço.

Cada componente é dedicado a um domínio de trabalho, como interações com RDBMS, integrações com sistemas de ficheiros ou outras funcionalidades mais especializadas, dependentes dos requisitos e restrições do cliente.

### 2. ACESSIBILIDADE DE DADOS

Dado que os dados são armazenados, é necessário que sejam recuperados para processamento e depois armazenados novamente quando alterados (seja através de atualização ou remoção total).

Sabendo de antemão que um determinado cliente pode optar por várias abordagens, o w2bill Realtime implementou uma solução de Dados (Data Grid) para lidar com todos os aspetos da manipulação de dados.

Esta camada permite que todos os Serviços que interagem se abstraíam completamente de como e onde os dados são geridos, acedendo-lhes apenas de forma eficiente e fiável.

### 3. PERSISTÊNCIA

É imediatamente reconhecido que os dados devem ser mantidos de forma segura, consistente e protegida, para que possam ser utilizados. Devem também escalar facilmente e ter o mínimo de restrições possíveis na sua gestão.

Por essa razão, não existe uma solução direta ou abordagem clara sobre qual a melhor tecnologia a utilizar.

As bases de dados relacionais (RDBMS) são plataformas com resultados comprovados, pela sua fiabilidade e eficácia, e oferecem uma multiplicidade de funcionalidades que a maioria das pessoas dá como garantidas, mas apresentam limitações em termos de custo, escalabilidade e desempenho.

As soluções NoSQL mais recentes têm uma adoção menos generalizada em grandes empresas, mas são, na sua generalidade, intrinsecamente escaláveis e eficientes, a uma fração do custo. E depois existem abordagens mais próximas do armazenamento físico real, como HDFS, ZFS e outras.

Para lidar com a persistência, o w2bill Realtime adotou uma abordagem de “escolha qualquer uma”. Ou seja, seja qual for a solução mais adequada para um cliente ou para um domínio específico dentro de um cliente, a plataforma oferecerá suporte.

Dado que os dados são armazenados, é necessário recuperá-los para processamento e, em seguida, armazená-los novamente quando forem alterados (seja atualizados ou removidos por completo).

Como a abordagem relativa à persistência tem sido de “escolha qualquer uma”, e sabendo de antemão que um determinado cliente pode optar por várias soluções, o w2bill Realtime implementou uma solução de Data Grid para lidar com todos os aspetos da manipulação de dados.

Esta camada permite que todos os Serviços que interagem com os dados abstraíam completamente a forma e o local onde estes são geridos — limitando-se a aceder-lhes de forma eficiente e fiável.

### PLATAFORMA DE LOGGING/ 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 estas métricas, é possível saber quanto tempo uma ordem leva a ser processada, bem como quanto tempo cada componente demora a executar a operação requerida.

Com estas capacidades, é possível detetar anomalias nas instâncias dos componentes ao monitorizar os tempos de execução de cada sistema, juntamente com outras informações relevantes necessárias para o agendamento de ações de melhoria de desempenho, deteção de bottlenecks ou decisões de escalabilidade.

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

### LOGGING

Em implementações distribuídas, as instâncias dos componentes estarão espalhadas por várias máquinas de um cluster. Cada instância irá produzir logs que devem ser armazenados e agrupados com os logs de todos os componentes do sistema, permitindo a análise do comportamento do sistema.

Todos os componentes do w2bill Realtime Framework estão preparados para escrever logs da aplicação num ficheiro – na máquina local – e para enviar uma mensagem de logging com dados adicionais, permitindo aos utilizadores relacionar todos os logs de uma ordem ou execução de workflow específicos.



Isto permite obter uma visão mais clara e abrangente do sistema. Estes logs são centralizados num índice do Elasticsearch com um esquema pré-definido. É possível pesquisar logs por id da ordem, relação do id ou por mensagem de log. Os logs são publicados de forma assíncrona para o broker, melhorando o desempenho do sistema.

### SERVIÇO DE DESCOBERTA E SERVIDOR DE CONFIGURAÇÃO

As instâncias dos componentes podem ser iniciadas em qualquer host disponível do cluster, de forma manual ou automática com base no esforço de trabalho. Para conseguir gerir e controlar todas as instâncias, foi implementado um Service Registry e um Configuration Server. Todos os componentes obtêm a sua configuração a partir do servidor central de configuração, e devem criar um registo – com o seu endereço IP e porta HTTP – no Service Registry.

Cada instância de serviço é responsável por se registar no Service Registry durante o arranque e por se desregistar no encerramento.

### MONITORIZAÇÃO

As instâncias dos componentes são monitorizadas continuamente utilizando os endpoints de verificação de integridade fornecidos por cada um deles, bem como pela análise de métricas. Com isso, é possível atuar, iniciando e parando instâncias, mantendo o nível de serviço desejado e o uso efetivo dos recursos no ambiente em execução.

### FRONTENDS

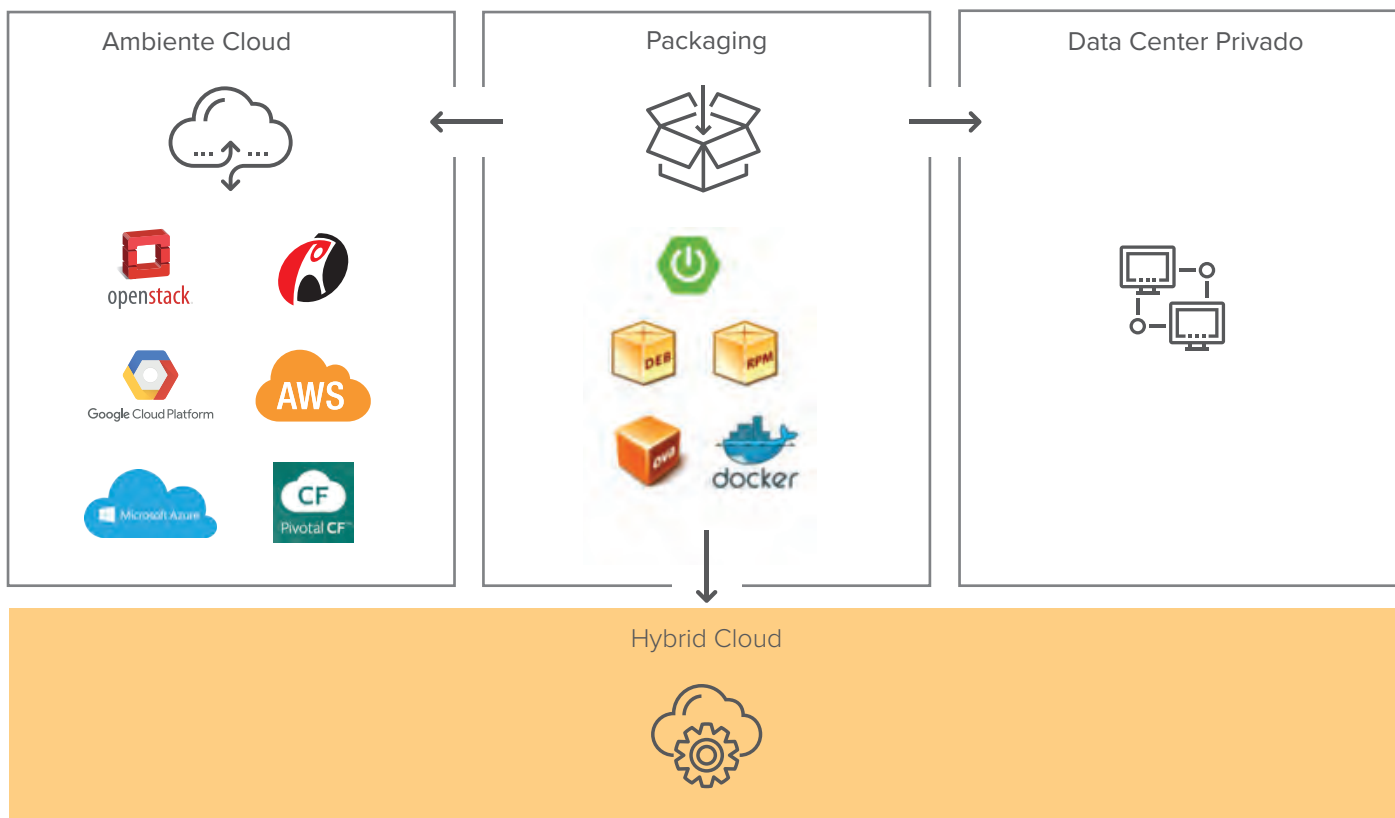
O w2bill apresenta uma interface gráfica unificada que permite realizar operações comuns sobre a Plataforma, como diagnóstico, rastreamento, validação de logs e verificação de status. A abordagem escolhida permite que cada cliente

tenham visualizações específicas, representando informações relevantes a serem exibidas, em vez de um ecrã estático mostrando dados predefinidos.

### SERVIÇO DE AUTENTICAÇÃO

Para atender à necessidade de segurança e acesso validado aos serviços, o w2bill Realtime incorpora um serviço de autenticação que permite que os serviços sejam utilizados apenas por utilizadores devidamente autenticados.

Com esse recurso, todas as requisições feitas aos serviços expostos devem ser autenticadas por meio de um servidor central de autenticação – fornecido pelo w2bill Realtime – garantindo que apenas os serviços aos quais o utilizador possui permissões de acesso sejam acedidos. Este servidor implementa o protocolo OAuth 2.0 e pode autenticar utilizadores numa base de dados relacional ou num servidor LDAP.



.....  
 Figura 3. w2bill™ Packaging para múltiplas implementações

# CAPACIDADES

## TECNOLOGIAS

Toda a solução é baseada no stack tecnológico apresentado abaixo em mais detalhe. De qualquer forma, dadas as decisões de arquitectura tomadas anteriormente, torna-se totalmente possível optar por soluções específicas usando diferentes tecnologias, desde que certos aspectos-chave de integração sejam atendidos. Como exemplo, é possível implementar um componente Executor implementado em C/C++, ou Microsoft C#, ou até mesmo Python.

A escolha na stack de Persistência pode ser substituída por outras soluções NoSQL, como o MongoDB, ou até sistemas tradicionais RDBMS, como MySQL, Postgres ou Oracle. Ao longo de toda a solução, a escolha da tecnologia foi focada na sua adequação (para o seu conjunto específico de responsabilidades), bem como o seu histórico comprovado, adoção disseminada na indústria, suporte confiável de nível empresarial e roadmap de evolução.

Com a crescente adoção de alternativas baseadas em Cloud em vez de soluções locais, certos aspectos de arquitetura, design e implementação subjacente foram considerados para garantir que a solução estivesse preparada sem a necessidade de personalização específica. Ao abordar essas preocupações logo no início, a seleção das tecnologias foi refinada, onde apenas aquelas com níveis apropriados de conformidade foram mantidas como elegíveis.

## Uma lista resumida do que a Framework de Gestão de Tarefas em Tempo Real fornece:

- Conceito de pedido, acompanhando cada solicitação do início ao fim
- Conceito de pedido duplicado através de campos-chave configuráveis
- Conceito de serviços configuráveis e versionados
- Conceito de agrupamento de serviços para agregação lógica
- Conceito de usuários de serviços e autenticação respectiva
- Escalabilidade cruzada de plataformas, auto, para cima e para fora
- Regras de validação de solicitações, desde a sintaxe até critérios de negócios
- Fluxos de tarefas geridos por Orquestradores e implementados através de Executores
- Ciclo de vida do pedido com status de pedido diferenciados
- Múltiplas linguagens de integração: SOAP, REST, etc.
- Design tolerante a falhas através do agrupamento de componentes
- Sem ponto central de falha
- Registro de logs com métricas de desempenho
- Verificações de saúde dos componentes e monitorização
- Data Grid para acesso e manipulação de dados de alto desempenho e ACID
- Persistência de dados distribuída, resiliente e em cluster
- Suporte a múltiplas soluções de persistência de dados (RDBMS, NoSQL)
- Abordagem distribuída do Orquestrador
- Executores padrão para operações comuns, com capacidades ajustadas às especificidades do cliente
- Pausa e retomar de fluxo
- Execução de fluxo alternativo para cenários de falha de fluxo padrão
- Comunicação entre componentes através de mensagens para garantia de entrega pelo menos uma vez

# ROADMAP

Plenamente conscientes de que o produto de hoje não será o produto de amanhã, uma vez que as necessidades e a tecnologia atuais serão certamente substituídas no futuro, a solução w2bill Realtime possui um roadmap de funcionalidades em constante evolução.

Estas novas funcionalidades surgiram da experiência adquirida junto dos nossos clientes, bem como dos seus insights, conselhos e sugestões. O objetivo não é manter versões específicas para cada cliente

estagnadas no tempo em cada instalação, mas sim assegurar uma evolução contínua, orientada para oferecer novas formas e melhores de suportar a tecnologia e o negócio.

Ao investigar constantemente o que está a acontecer e o que está para vir, bem como o convergir de experiências entre mercados, clientes e mentalidades, vislumbramos o caminho de melhoria a disponibilizar.

A parceria com os nossos clientes é fundamental na formação desta visão e na definição do roadmap.



Figura 4. w2bill™ Visão Geral da Arquitetura



# ANEXOS

## Alguns insights técnicos sobre o w2bill Realtime

### MOTOR DE PROCESSAMENTO

#### JAVA 8

O w2bill Realtime é desenvolvido utilizando a versão mais recente da linguagem Java. Isto permite tirar partido de todas as novas funcionalidades e melhorias de desempenho introduzidas 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 seu código. O Java conta com um vasto número de plugins e frameworks open source, uma comunidade de programadores extremamente ativa, um extenso roadmap de evolução e também grupos de suporte.

#### PIVOTAL SPRING FRAMEWORK

Pivotal Spring Framework é um framework de código aberto que suporta o desenvolvimento de aplicações Java, fornecendo suporte às necessidades de infraestrutura e oferecendo um modelo de programação consistente sobre diferentes tecnologias.

Tem sido amplamente utilizado em toda a indústria de desenvolvimento Java, como uma alternativa ao modelo de Enterprise Java Beans.

A Pivotal, e em particular a sua equipa do Spring, está sempre a planear o futuro e a conduzir o framework para responder a novos requisitos de negócio. Exemplos relevantes são o projeto Cloud Stream e a introdução da programação reativa na próxima versão 5.0.

O w2bill Realtime acompanhará de perto estas evoluções para extrair delas quaisquer melhorias relevantes.

#### PIVOTAL RABBITMQ

O RabbitMQ é um broker de mensagens open source gerido pela Pivotal e é bem suportado pelo Spring Framework. Apresenta fiabilidade, roteamento flexível, clustering e filas altamente disponíveis. Integra-se com várias linguagens de programação, facilitando a integração do padrão de mensagens com diferentes aplicações, permitindo assim que o w2bill Realtime incorpore rápida e facilmente módulos específicos de clientes desenvolvidos em linguagens diferentes da linguagem principal de implementação do w2bill Realtime.

O RabbitMQ é um broker capaz de suportar múltiplos protocolos, o que amplia ainda mais as capacidades de integração do w2bill Realtime, não restringindo a sua abordagem de mensagens a um protocolo específico.

## GESTÃO DE RECURSOS

### APACHE MESOS

w2bill Realtime é um sistema naturalmente distribuído. Qualquer um dos seus componentes e respectivas instâncias pode ser executado em múltiplas máquinas, contribuindo assim para um melhor desempenho através da escalabilidade horizontal, além de evitar pontos centrais de falha, utilizando técnicas de clustering e tolerância a falhas.

A implementação é projetada para usar não só a virtualização comum de máquinas, mas também abordagens de *containerization*, como Docker ou Kubernetes. A distribuição e gestão de recursos são, portanto, uma preocupação que foi devidamente abordada usando o Apache Mesos, que oferece um manuseio centralizado para o deploy e escalabilidade dos componentes do w2bill Realtime em qualquer tipo de instalação, desde instalações locais, na Cloud ou numa mistura de ambos.

### APACHE IGNITE

Apache Ignite é uma Data Fabric em memória que fornece alto desempenho para dados, computação e service grids. Suporta transações distribuídas totalmente compatíveis com ACID, garantindo consistência em todos os dados e suporta a sintaxe SQL padrão para consultar os objetos armazenados na Base de Dados. O acesso a essa base de dados é possível através de múltiplas linguagens de programação.

Outra característica relevante são as avançadas capacidades de clustering, que possibilitam escalabilidade, tolerância a falhas e atendem aos requisitos de alto desempenho.

## PERSISTÊNCIA

### APACHE CASSANDRA

A escolha do Apache Cassandra para armazenamento foi baseada em sua multitude de capacidades, particularmente:

- Natureza descentralizada, evitando pontos únicos de falha ou bottlenecks na rede
- Tolerância a falhas, através da replicação de dados entre nodes e data centers, juntamente com a capacidade de substituição de nodes com falhas, sem downtime
- Escalabilidade, pelo uso de "quase infinitos" múltiplos nodes para suportar o crescimento de processamento e armazenamento
- Elasticidade, baseada no aumento de nodes para aumentar a taxa de leitura/gravação
- Suporte profissional e uso comprovado em grandes empresas globais (CERN, eBay, GitHub, Apple, Netflix, entre outras)

O w2bill Realtime Framework depende da solução Apache Cassandra para armazenar as suas informações entre componentes. No entanto, o uso é indireto, já que a camada de acessibilidade é realizada através do Data Fabric Apache Ignite. É responsabilidade dessa camada comunicar-se efetivamente com o Apache Cassandra, para o armazenamento (gravação) e leitura dos dados.





CMAS - Systems Consultants, Lda  
Edifício Escritorios do Tejo  
Rua do Pólo sul , Nº2 , 1º  
A Parque das Nações  
1990-273 Lisboa

[mail@cmas-systems.com](mailto:mail@cmas-systems.com)

[www.cmas-systems.com](http://www.cmas-systems.com)

