Arquitetura do Google Wave

Anderson Silva, Bruno Montenegro

{abns, bms2}@cin.ufpe.br

Disciplina: Arquitetura de Software - in0979

Universidade Federal de Pernambuco

Recife, Novembro/2009

HISTÓRICO DE REVISÕES

Data	Descrição	Responsável
25/11/2009	Primeira versão	Anderson e Bruno

SUMÁRIO

1. INTRODUÇÃO

Este documento representa a arquitetura do sistema Google Wave. O Google Wave consiste em uma ferramenta online de comunicação e colaboração em tempo real. Os usuários desta ferramenta utilizam para comunicação um elemento chamado *wave*, que permite, inclusive, que alguns *gadgets* e robôs sejam adicionados e passem a colaborar neste novo meio de interação, assim como qualquer outro usuário. Um *wave* pode ser considerado um documento, o meio utilizado pelos participantes para se comunicar.

Para criar, editar ou participar de uma *wave*, um usuário precisa estar cadastrado em um provedor. Este provedor é responsável por verificar a identidade do usuário e por compartilhar as informações das *waves* com os usuários de outros provedores. Estas alterações devem ser visualizadas à medida que estão sendo digitadas, o que torna o Google Wave uma alternativa interessante à necessidade de utilização de uma aplicação com características de comunicação em tempo-real.

Os provedores são responsáveis, também, por garantir que as informações digitadas por seus usuários serão repassadas para os outros provedores, mesmo que algum destes esteja indisponível no momento.

2. REQUISITOS FUNCIONAIS

[RF001] Um usuário precisa autenticar-se para utilizar o sistema

[RF002] Um usuário deve estar associado a um provedor

[RF003] Um usuário pode criar novas *waves*

[RF004] Um usuário pode convidar outros usuários para compor uma *wave*, sejam estes do mesmo provedor ou não

[RF005] Um usuário pode editar uma *wave* da qual faz parte, independente de quem a criou

[RF006] Os usuários devem visualizar as edições de uma *wave* o mais rápido possível

[RF007] Um usuário pode rever todo o histórico de edição de uma *wave* da qual ele faz parte

[RF008] Um usuário pode enviar mensagens privadas para um outro participante da *wave*

[RF009] A aplicação cliente pode aceitar extensões na forma de *gadgets*

[RF010] Um robô pode ser utilizado como um usuário comum da *wave*

[RF011] Os usuários podem modificar o mesmo documento simultaneamente

[RF012] Um usuário pode manter sua lista de contatos

[RF013] Um usuário pode verificar quais os usuários de sua lista de contatos estão conectados

3. REQUISITOS NÃO-FUNCIONAIS

[RNF001] A aplicação deve estar sempre disponível

[RNF002] A aplicação deve ser tolerante a falhas de comunicação entre os provedores

[RNF003] A aplicação deve ter desempenho aceitável para que a comunicação em tempo real seja possível

[RNF004] A aplicação deve permitir que vários usuários estejam conectados ao mesmo tempo

[RNF005] A aplicação deve garantir um tráfego de dados com segurança

[RNF006] A aplicação cliente deve ser portável em diferentes plataformas

[RNF007] A aplicação servidora deve ser independente de ambiente

4. MODELO DE DADOS

Esta seção descreve os principais conceitos envolvidos com a construção e utilização do Google Wave e serve como guia para entender como os dados são representados. A Figura 1 apresenta a relação entre uma *wave* e seus componentes internos.

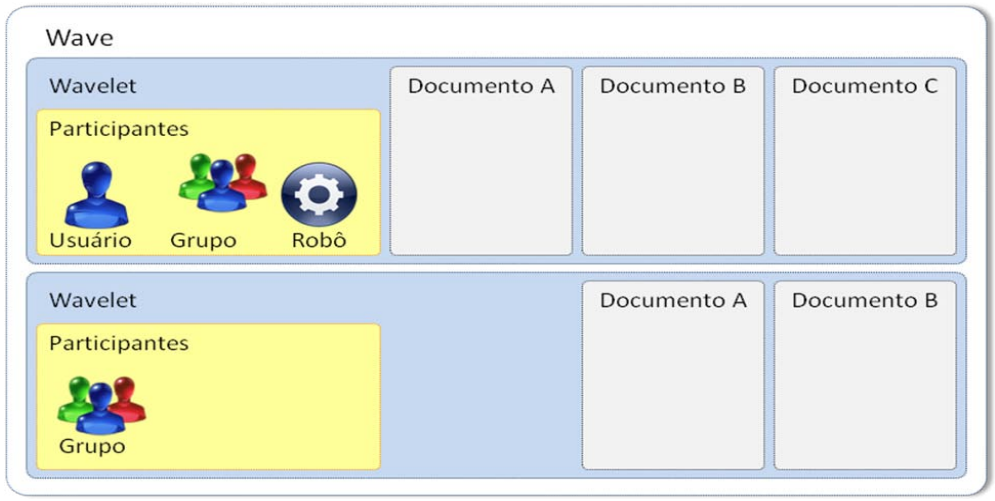

Figura 1: *Wave* e componentes internos

4.1. Wave

Cada *wave* tem um ID globalmente único e consiste de um conjunto de *wavelets*.

4.2. Wavelet

Uma *wavelet* tem um ID que é único dentro da *wave* que a contém e é composta de uma lista de participantes e um conjunto de documentos. A *wavelet* é a entidade sobre a qual o controle de concorrência / transformações operacionais se aplicam.

4.3. Participante

Um participante é identificado por um endereço *wave* que é uma *string* no mesmo formato de um endereço de email (parte-local@domínio). Um participante pode ser um usuário, um grupo ou um robô.

4.4. Documento

Um documento tem um ID único dentro da *wavelet* que o contém e é composto por um documento XML e um conjunto anotações separadas. Anotações separadas são ponteiros para pontos do documento XML e são independentes da estrutura XML. Elas são usadas para representar a formatação do texto, sugestões ortográficas, links. Documentos formam uma árvore dentro da *wavelet*.

4.5. Wave view

Uma *wave view* é um subconjunto de *wavelets* em uma *wave* às quais um usuário particular tem acesso. Um usuário ganha acesso à *wavelet* ou sendo um participante da mesma ou pertencendo a um grupo que é participante.

5. DECISÕES ARQUITETURAIS

Esta seção apresenta as decisões de projeto tomadas para atender as necessidades dos usuários. As decisões foram baseadas nos requisitos funcionais e não-funcionais apresentados anteriormente.

5.1. Linguagem de programação

Escolha: Java

Justificativa: Linguagem de programação portável que permite que a aplicação seja independente de plataforma/sistema operacional. Amplamente conhecida e utilizada pelos desenvolvedores, facilita e acelerando a implementação.

Requisitos: RNF007

5.2. Protocolo de comunicação cliente/servidor

Escolha: HTTP

Justificativa: Padrão na web, onde o sistema vai funcionar. Bastante comum em qualquer máquina que acesse a WWW, permite que se utilize qualquer navegador para rodar o cliente.

Requisitos: RNF006

5.3. Ambiente de execução para o cliente

Escolha: Web

Justificativa: A utilização de um cliente web torna desnecessária a instalação de um cliente nas máquinas que utilizarão o sistema, aumentando a portabilidade do cliente.

Requisitos: RNF006

5.4. Protocolo de comunicação servidor/servidor

Escolha: XMPP

Justificativa: O protocolo atende as necessidades de comunicação em tempo real e detecção da presença de outros usuários conectados, além de suportar criptografia e já vir sendo utilizado em outros serviços Google, como o Google Talk.

Requisitos: RF013, RNF005

5.5. Infraestrutura

Escolha: Organizacional

Justificativa: Já disponível e bastante conhecida, o que facilita e agiliza a implantação do sistema. Google Accounts é usado para cadastrar e autenticar os usuários, além de manter a lista de contatos. Google Clusters e Big Table garantem os requisitos de desempenho, tolerância a falhas, disponibilidade.

Requisitos: RF001, RF002, RF011, RF0012, RNF001, RNF002, RNF003, RNF004

5.6. Controle de concorrência

Escolha: *Operation Transformation*

Justificativa: Da necessidade de controlar a concorrência das modificações realizadas na *waves* deriva a escolha de utilização de OT's, já que com elas todas as alterações são consideradas, não há problemas de inconsistência e não há bloqueio nos documentos.

Alternativas analisadas: Controle de concorrência otimista – desfaz as alterações em casos de conflitos; *Lock* – é necessário bloquear o documento que será alterado, aumenta o tempo de resposta e existe a possibilidade de *deadlock*; e *Timestamp* – possui problemas de consistência;

Requisitos: RF011

6. DIAGRAMA DE REDES

6.1. Representação

O diagrama apresentado na Figura 2 demonstra a representação dos elementos físicos que compõem o sistema do Google Wave.

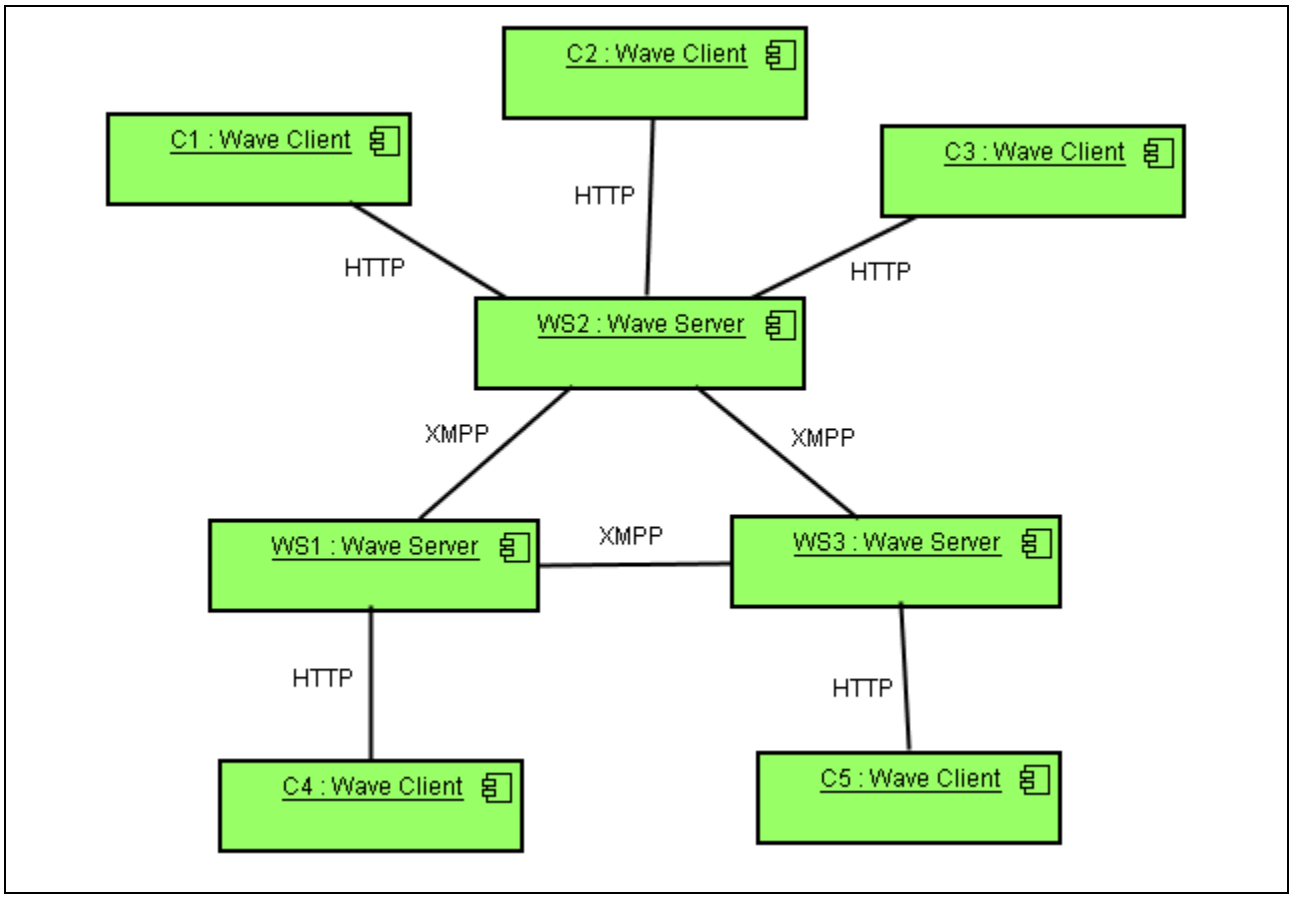

Figura 2: Diagrama de rede

6.2. Catálogo de elementos

Elemento	Descrição
Wave Server	Provedor de serviços responsável por disponibilizar as funcionalidades para os clientes
Wave Client	Utilizado pelo usuário para ter acesso as funcionalidades

6.3. Contexto

Situação hipotética onde existem três provedores de serviços, que se comunicam entre si, além de diversos clientes ligados a estes provedores.

6.4. Plano arquitetural

Entre o usuário e o provedor utiliza-se o modelo cliente-servidor. A comunicação é realizada através do protocolo HTTP. A comunicação entre os servidores é ponto-a-ponto, através do protocolo XMPP. As escolhas deste modelo e protocolos são explicadas na seção *Decisões arquiteturais*.

7. VISÃO DE COMPONENTES E CONECTORES

7.1. Representação

A Figura 3 apresenta os relacionamentos entre os dois principais componentes do sistema, assim como os conectores utilizados nas interações entre estes componentes.

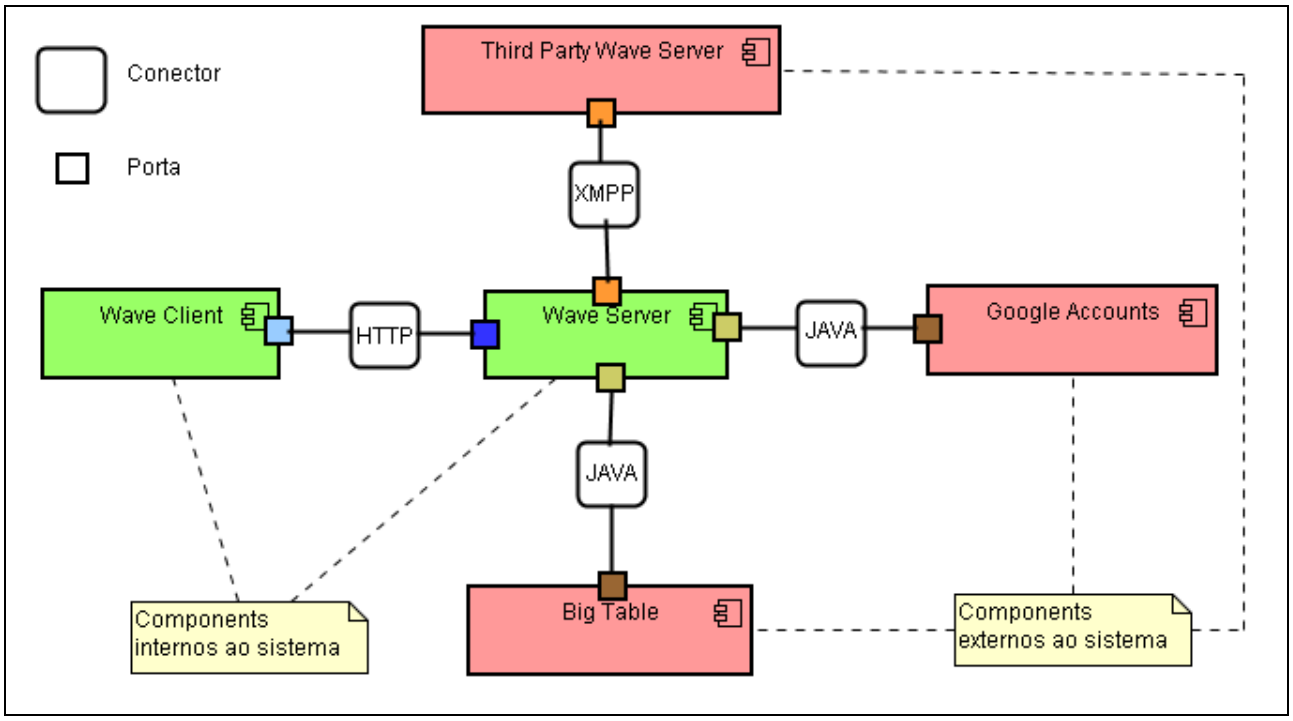

Figura 3: Visão de components e conectores

7.2. Catálogo de elementos

Elemento	Descrição
Wave Server	Provedor de serviços responsável por disponibilizar as funcionalidades para os clientes
Wave Client	Utilizado pelo usuário para ter acesso as funcionalidades
Third Party Wave Server	Representação de qualquer outro provedor de serviços *wave* que irá interagir com *wave server* apresentado.
Google Accounts	Componente externo, utilizado para gerenciar as contas dos usuários. Já disponível na infraestrutura Google.
Big Table	Componente externo, que teve sua utilização influenciada pela infraestrutura da ogranização.

7.3. Contexto

Comunicação entre os principais componentes do sistema e seu relacionamento com componentes externos.

7.4. Plano arquitetural

Utilização do modelo cliente-servidor e ponto-a-ponto, além da comunicação e do relacioamento entre estes e os componentes externos.

8. VISÃO DE MÓDULOS – WAVE SERVER

8.1. Representação

A Figura 4 apresenta em detalhes o componente Wave Server.

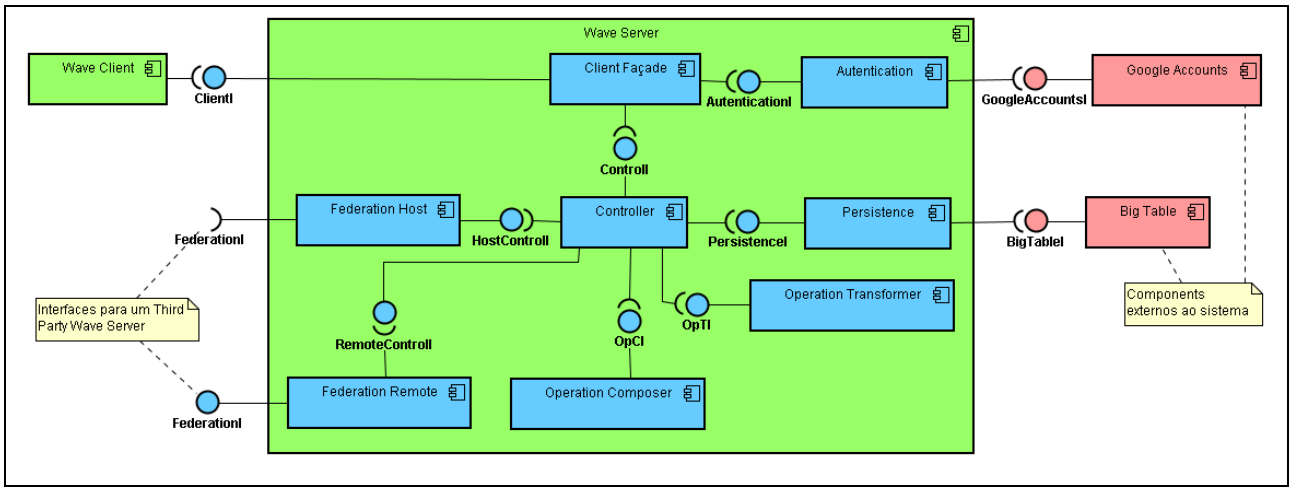

Figura 4: Diagrama de módulos – Wave Server

8.2. Catálogo de elementos

Elemento	Descrição
Client Façade	Responsável por intermediar a comunicação entre o cliente e o Controller.
Autentication	Responsável por intermediar a comunicação entre o sistema e o componente de contas do usuário.
Controller	Componente principal do sistema. Controla as funcionalidades solicitadas.
Persistence	Responsável por intermediar a comunicação entre o Controller e o componente de banco de dados.
Operation Transformer	Resolve os conflitos derivados das transformações.
Operation Composer	Compõe duas operações em uma, mantendo o mesmo resultado.
Federation Host	Comunica as alterações das *wavelets* locais para o componente Federation Remote de um outro provedor.
Federation Remote	Recebe as alterações de *wavelets* remotas e que foram enviadas através do componente Federation Host de um outro provedor.

8.3. Contexto

Esta visão apresenta os detalhes do principal componente do sistema em tempo de desenvolvimento. Cada elemento representado tem um papal importante no sucesso do comportamento do sistema.

8.4. Plano arquitetural

A separação do sistema em módulos facilita o desenvolvimento e a manutenção posterior. Com a divisão de responsabilidade apresentada nesta visão, percebe-se que o Controller passa a representar um componente mais simples.

9. VISÃO DE MÓDULOS – WAVE CLIENT

9.1. Representação

A Figura 5 apresenta a divisão interna do componente Wave Client, facilitando sua compreensão.

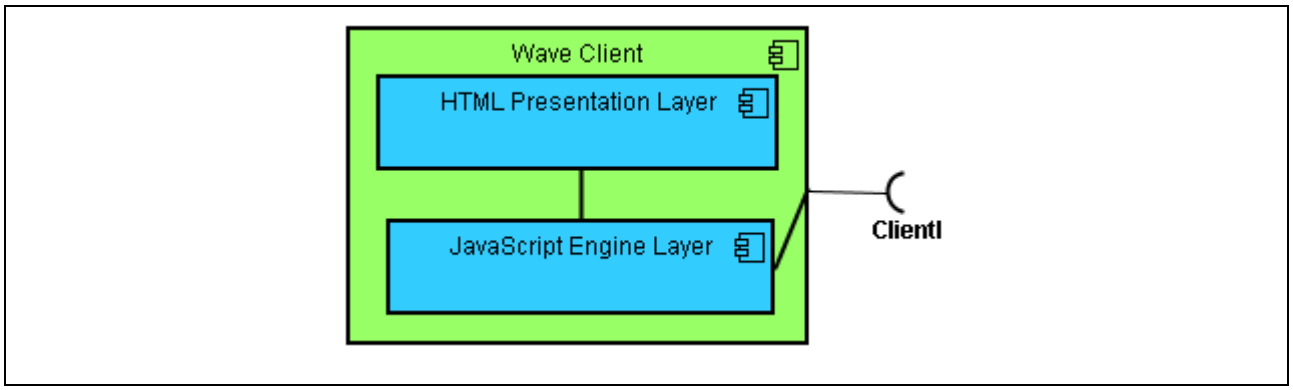

Figura 5: Diagrama de módulos – Wave Client

9.2. Catálogo de elementos

Elemento	Descrição
HTML Presentation Layer	Responsável por construir a interface que será apresentada ao usuário.
JavaScript Engine Layer	Responsável pela comunicação com o servidor.

9.3. Contexto

Detalhes internos do componente cliente do Google Wave.

9.4. Plano arquitetural

Utilização de duas camadas para melhor distribuir as responsabilidades do componente.

10. VISÃO DE INTERAÇÃO

10.1. Representação

As Figuras 6 e 7 apresentam o comportamento dinâmico do sistema quando é realizada a atualização de um documento (*wavelet*) local ou remoto, respectivamente.

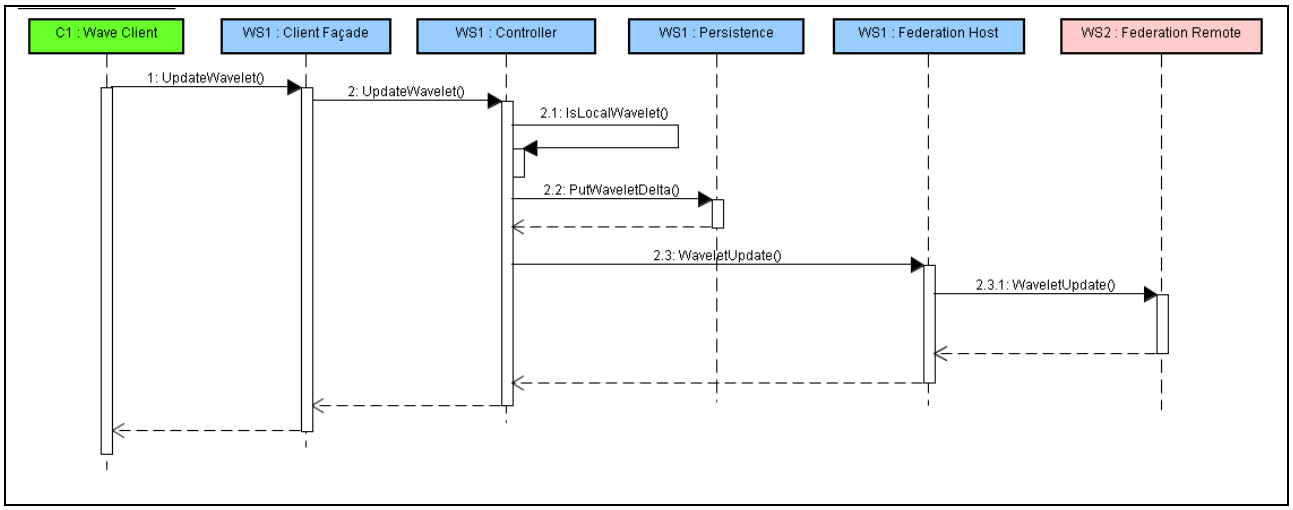

Figura 6: Diagrama de sequência – Local Wavelet Update

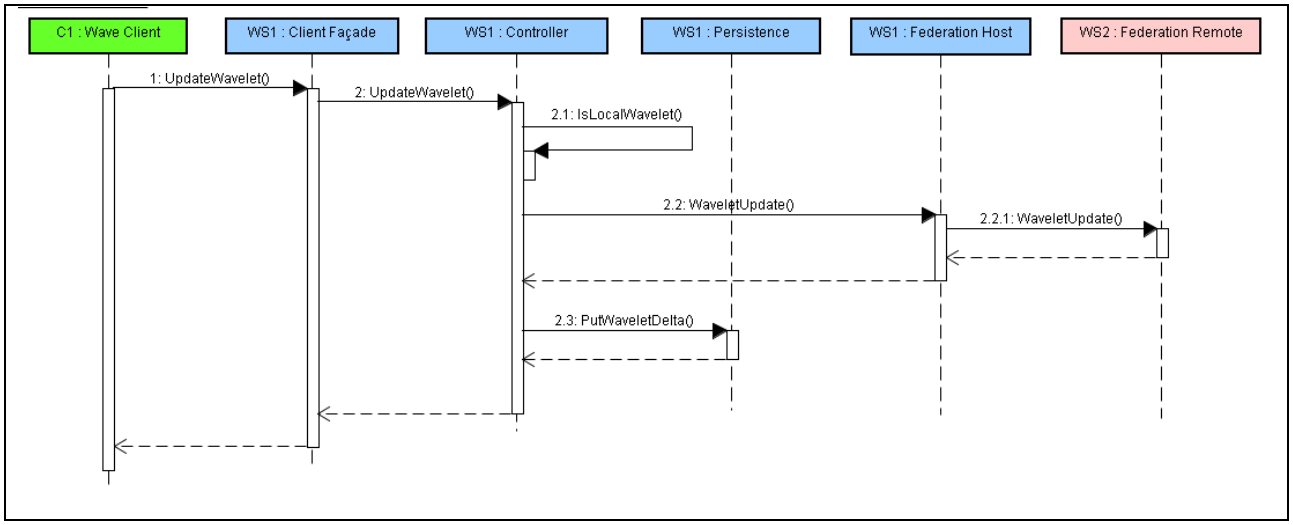

Figura 7: Diagrama de sequência – Remote Wavelet Update

10.2. Catálogo de elementos

Os componentes desta visão já foram descritos em seções anteriores.

10.3. Contexto

Esta visão evidencia o comportamento dinâmico do sistema durante a atualização de um documento, bem como, as interações que ocorrem entre os componentes para poder realizar esta atualização.

10.4. Plano arquitetural

Esta representação permite a visualização do modelo cliente-servidor, onde o uso deste modelo já foi em visões descritas anteriormente, além da separação de responsabilidades que foi utilizada com o propósito de modularizar o sistema como um todo, facilitando a manutenção e a escalabilidade do sistema.

11. VISÃO DE IMPLANTAÇÃO

11.1. Representação

As Figuras 8 e 9 apresentam duas visões de implantação para o sistema Google Wave.

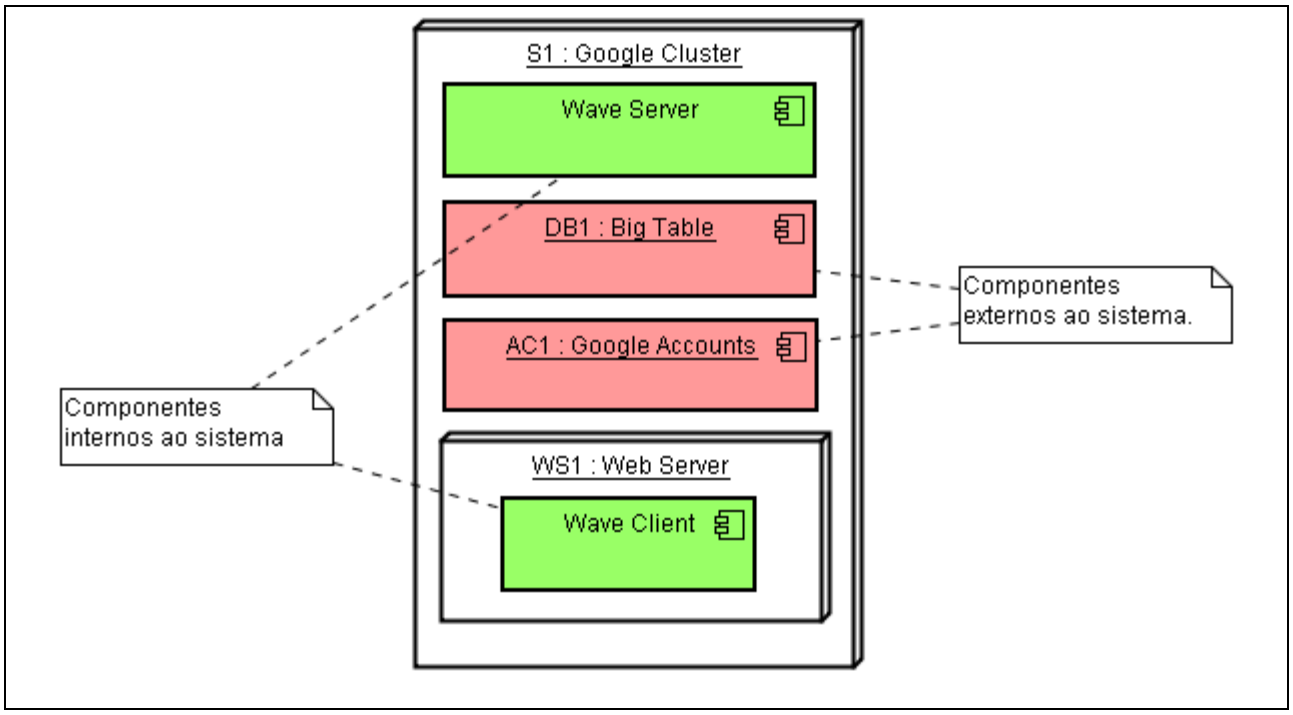

Figura 8: Diagrama de implantação - Instalação

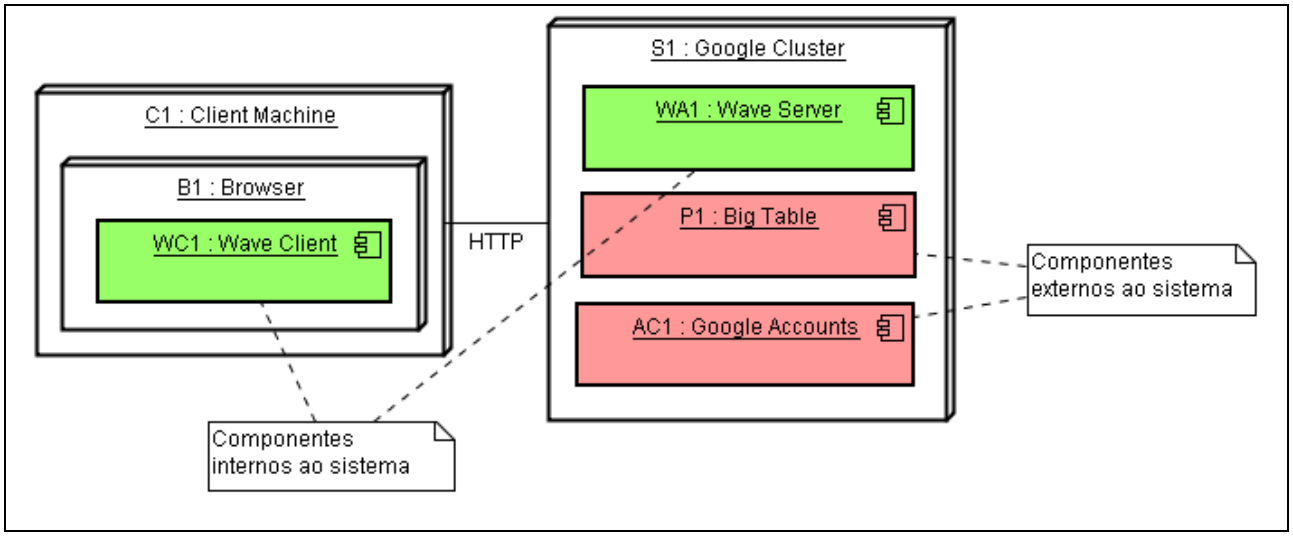

Figura 9: Diagrama de implantação – Execução

11.2. Catálogo de elementos

Os componentes desta visão já foram descritos em seções anteriores.

11.3. Contexto

Esta representação evidencia a localização dos componentes do sistema durante as fases de instalação e execução.

11.4. Plano arquitetural

Esta representação permite a visualização do modelo cliente-servidor, além da disposição dos componentes no que se pode chamar de Google Cluster.

12. INTERFACES

Esta seção descreve as interfaces utilizadas na interação entre os componentes internos ao *wave server*.

12.1. AutenticationI

Serviço	Parâmetros	Resposta
Login	UserName, Password	Ok

12.2. ClientI

Serviço	Parâmetros	Resposta
Login	UserName, Password	True, False
GetWaveView	Wave ID	List of Wavelet IDs
OpenWavelet	Wave ID, Wavelet ID	Snapshot of wavelet, Hash
UpdateWavelet	Wave ID, Wavelet ID, Version, Delta	New version, Hash
GetWaveList		List of Wave IDs
CreateWave		Wave ID
GetWaveHistory	Wave ID, Hash	List of operations from hash
RequestUpdate		Wavelet ID, Delta, Version, Hash
AddUserToWave	Wave ID, Username	Ok
AddUserToWavelet	Wave ID, Wavelet ID, Username	Ok

12.3. ControlI

Serviço	Parâmetros	Resposta
GetWaveView	Wave ID	List of Wavelet IDs
OpenWavelet	Wave ID, Wavelet ID	Snapshot of wavelet, History hash
UpdateWavelet	Wave ID, Wavelet ID, Version, Delta	New version, Hash
GetWaveList		List of Wave IDs
CreateWave		Wave ID
GetWaveHistory	Wave ID	List of operations
RequestUpdate		List (Wavelet ID, Delta, Version, Hash)
AddUserToWave	Wave ID, Username	Ok
AddUserToWavelet	Wave ID, Wavelet ID, Username	Ok

12.4. FederationI

Serviço	Parâmetros	Resposta
GetSigner	Wave ID, Wavelet ID, Hash, Version	Certificates
PostSigner	Certificates	Ack
WaveletUpdate	Wave ID, Wavelet ID, Delta	Ack
RequestHistory	Wave ID, Wavelet ID, Start-version, End-version, Start-hash, End-Hash	List of deltas

SubmitRequest	Wave ID, Wavelet ID, Delta	Hash, Version

12.5. HostControlI

Serviço	Parâmetros	Resposta
GetSigner	Wave ID, Wavelet ID, Hash, Version	Certificates
PostSigner	Certificates	Ack
WaveletUpdate	Wave ID, Wavelet ID, Delta	Ack

12.6. OpCI

Serviço	Parâmetros	Resposta
ComposeOperation	Operation1, Operation2	ComposedOperation

12.7. OpTI

Serviço	Parâmetros	Resposta
TransformOperation	Operation1, Operation2	TransformedOperation1, TransformedOperation2

12.8. PersistenceI

Serviço	Parâmetros	Resposta
GetWaveView	Wave ID, Username	List of Wavelet IDs
GetWaveletSnapshot	Wave ID, Wavelet ID	Snapshot, Hash
GetWaveletFromHash	Wave ID, Wavelet ID, Hash	List of deltas from history hash
PutWaveletDelta	Wave ID, Wavelet, ID, Delta	Hash
NewWave	Username	Wave ID
NewWavelet	Wave ID	Wavelet ID
AddUserToWave	Wave ID, Username	Ok
AddUserToWavelet	Wave ID, Wavelet ID, Username	Ok

12.9. RemoteControlI

Serviço	Parâmetros	Resposta
RequestHistory	Wave ID, Wavelet ID, Start-version, End-version, Start-hash, End-Hash	List of deltas
SubmitRequest	Wave ID, Wavelet ID, Delta	Hash, Version