



Sistemas Distribuídos 2015-2016

Licenciatura em Engenharia de Telecomunicações e Informática

Relatório

Grupo T44

Pedro Lopes
81988



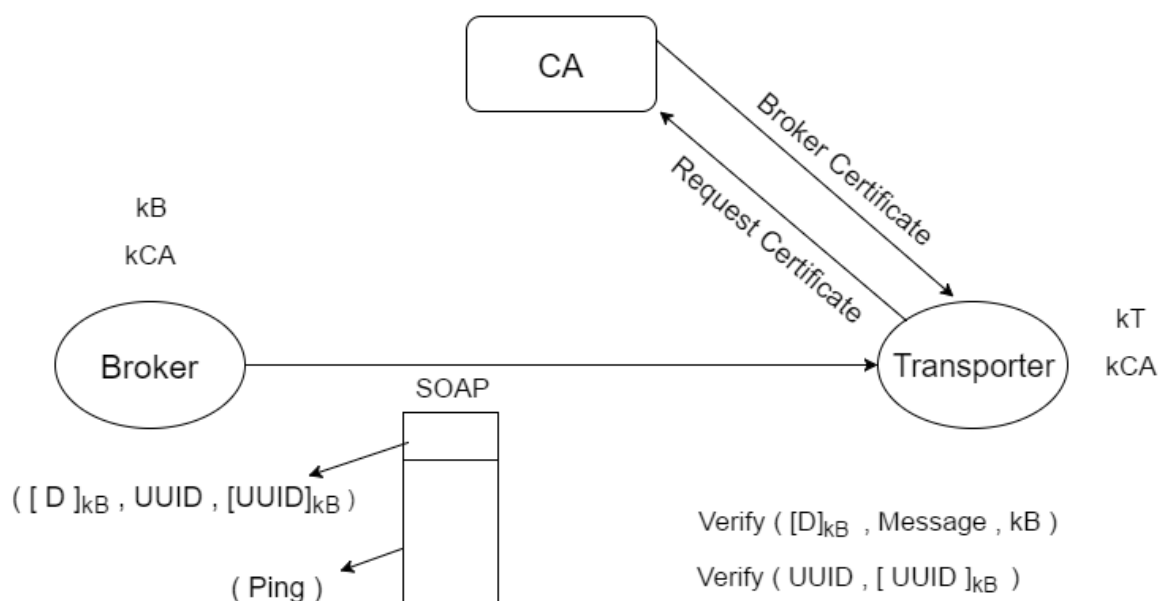
Diogo Barradas
81975



João Correia
81990



Segurança



Quando o Broker faz um Ping para uma Transporter (OutBound), vai adicionar ao Header do SOAP, através do HeaderHandler que interceta as mensagens SOAP, o nome do emissor (Broker), assinatura $([D]_{k_B}$, sendo D o resumo da mensagem SOAP sem header), identificador único (UUID) e a assinatura desse mesmo identificador $([UUID]_{k_B})$ sendo k_B a chave privada obtida através da KeyStore.

Quando a Transporter recebe o Ping (Inbound), vai ler no Header o nome do emissor e vai pedir à CA (Web Service) o certificado do emissor para posteriormente verificar a assinatura do emissor através da chave pública obtida pelo certificado.

Depois de guardar a assinatura, o identificador único e a assinatura do identificador único, são retirados todos os Headers do Soap para proceder à verificação da Integridade da mensagem com a função verify. A frescura é garantida através da comparação do identificador único com a lista dos últimos identificadores únicos guardados e caso já esteja na lista a mensagem é excluída por ser repetida. A autenticidade e o não repúdio são verificados através da assinatura do identificador único que é comparada com uma nova assinatura gerada pela chave pública do emissor. Caso sejam diferentes a mensagem é ignorada e considerada inválida.

Replicação

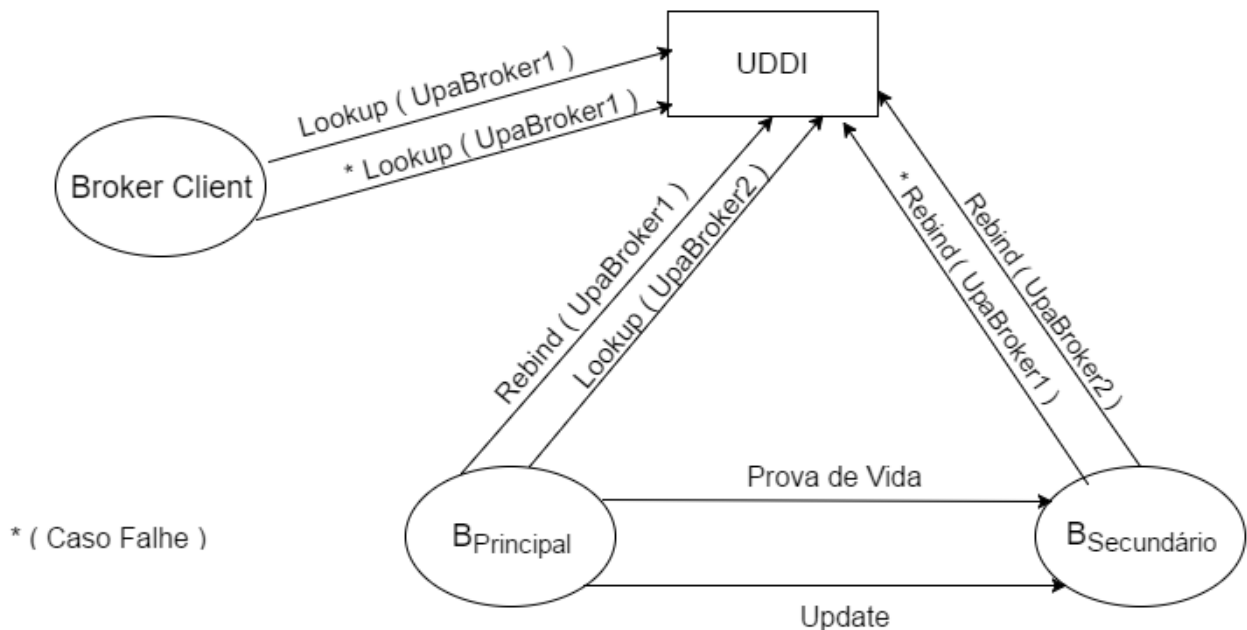


Fig1. -

Em primeiro lugar, lançamos o Broker Secundário, registrando-se no servidor de nomes como UpaBroker2. De seguida, executamos o Broker Principal que se vai registar no servidor de nomes como UpaBroker1 e vai procurar o Broker Secundário para que, em caso de falha, o Broker Secundário saiba que o Broker Principal falhou. Para isto, o Broker Principal envia de 5 em 5 segundos uma prova de vida (Ping) para o Broker Secundário, que está a espera de receber essa confirmação de funcionamento num período de tempo de 10 segundos. Caso o Broker Secundário não receba essa prova de vida, assume o lugar do Broker Principal, registrando-se novamente no servidor de nomes como UpaBroker1. Depois disso, o Broker Cliente, ao tentar executar uma operação deteta uma alteração e volta a contactar-se com o servidores de nomes, procurando novamente por UpaBroker1, para receber um novo endereço.

O Broker Principal mantém sempre o Broker Secundário atualizado, usando para isso a função update sempre que é realizado um RequestTransport . Esta função atualiza a lista de TransportViews no BrokerSecundario para que, caso aconteça uma falha, o Broker Secundário estar preparado para fazer o papel do Principal.