Instituto Superior Técnico

Sistemas Distribuídos 2015-2016

A36

URL do GITHUB: https://github.com/tecnico-distsys/A_36-project.git

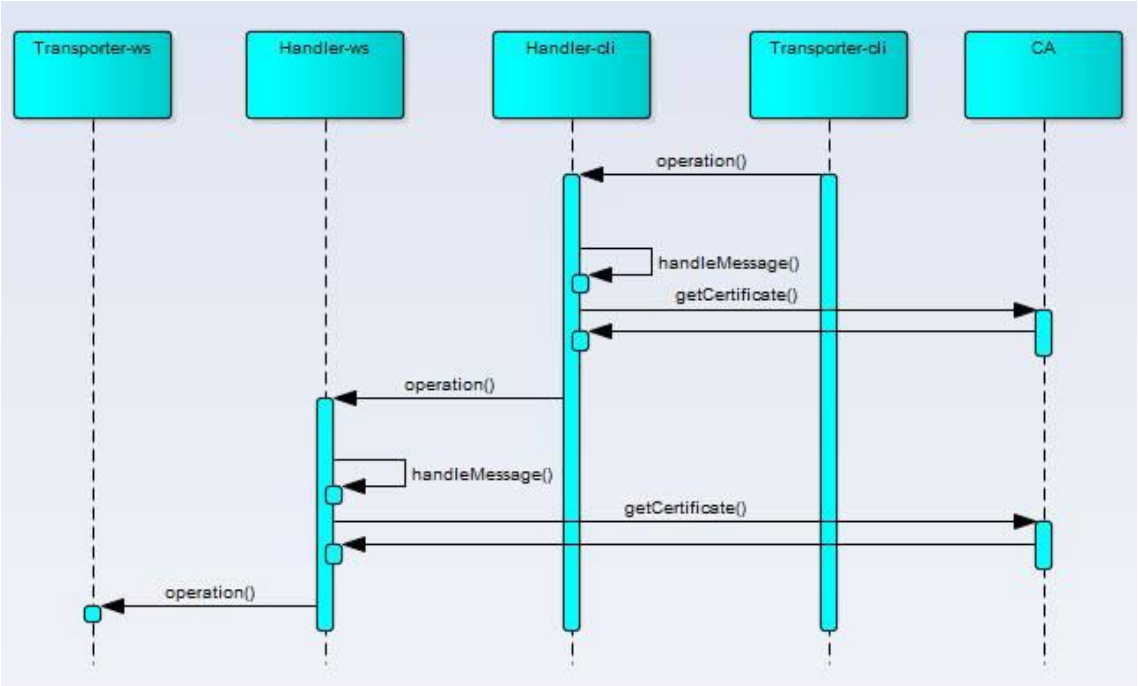
 Luís Borges - 78349

 Diogo Silva - 78517

 Paulo Ritto - 78929

13/5/2016

SEGURANÇA



Descrição/explicação da figura:

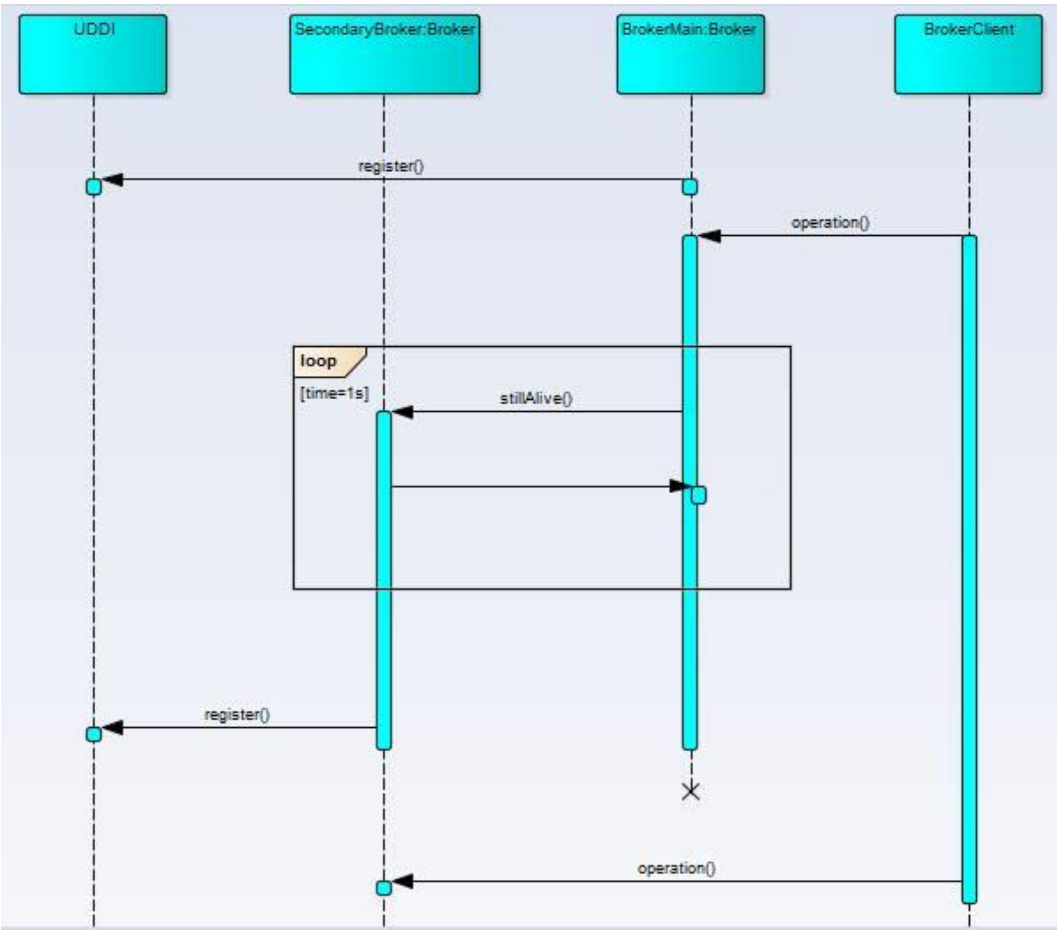
A implementação consiste na criação de 2 handlers, o handler do transporter-client (chamado através do Broker) e o handler do transporter-ws (servidor).

A comunicação entre estas duas entidades consiste no envio de mensagens SOAP, que vão ser intercetadas pelos handlers. Esses handlers (que fazem a distinção entre o envio e a receção de uma mensagem SOAP) vão ser os responsáveis pela assinatura digital das mensagens a enviar.

A assinatura consiste no cálculo da função digest, encriptada com a chave privada da entidade a enviar. De notar que se usa a chave privada da entidade que envia e não a chave pública da entidade que vai receber, visto que não se pretende garantir confidencialidade, mas apenas a integridade, não repúdio, autenticidade e frescura das mensagens.

A entidade que recebe vai contactar a Central de Autenticação para obter o certificado de chave pública da entidade que enviou a mensagem, para verificar que a chave pública é mesmo aquela. Obtido o certificado e a chave pública, usa-se a chave pública para desencriptar o digest recebido, comparando com o cálculo do digest da mensagem recebida. Se forem iguais, obtendo-se a confirmação de autenticidade.

REPLICAÇÃO



Descrição/explicação da figura:

Na figura acima, apresenta-se um diagrama de sequência que explica a implementação da replicação do servidor principal BrokerMain do projeto. Considerou-se as entidades UDDI, SecondaryBroker (servidor secundário), BrokerMain(servidor principal) e BrokerClient(cliente).

Em primeiro lugar, o servidor principal começa por se registar na UDDI (serviço de nomes) e fica à espera que o cliente o contacte para executar uma determinada operação. Assim que começam o contacto, o servidor fica em loop a notificar o servidor secundário de 1 em 1 segundo que ainda está "vivo"/funcional. Quando o servidor principal fica em baixo, um timer dispara e o servidor secundário deixa de receber as tais notificações do servidor principal. Assim, o servidor secundário substitui o servidor principal, regista-se na UDDI, passa a ser o servidor principal e fica à espera de um contacto do cliente.

A replicação do servidor principal implementada neste projeto, tem como objetivo permitir a tolerância a faltas no servidor tornando o projeto mais robusto.