TÉCNICO LISBOA

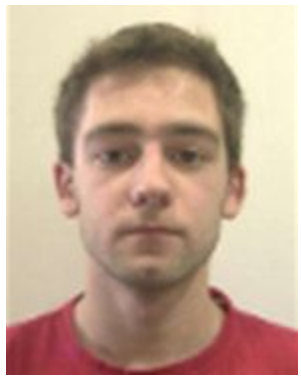
Sistemas Distribuídos 2015-2016

Relatório

LEIC-A

Grupo A45

URL do Repositório: https://github.com/tecnico-distsys/A_45-project.git

70171	75657	75694
		
João Miguel Neves	Paulo Jorge Gouveia	Daniel Machado Figueira

Segurança

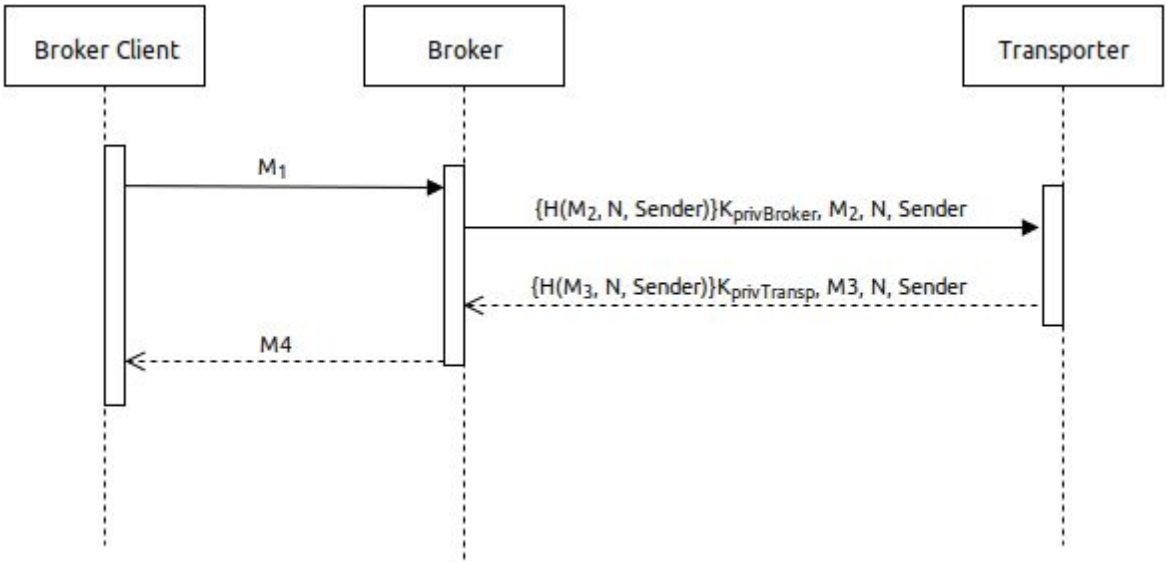


Fig.1: Exemplo do conteúdo das mensagens SOAP entre as entidades principais. (H - função de hash; N - nonce; Sender - nome do servidor que enviou o pedido)

Para a troca de mensagens entre o broker e as transportadoras usou-se *nonces,* para garantir a frescura, e um *digest* encriptado por chave assimétrica (assinatura), para garantir autenticidade, não-repudio e integridade. Não sendo um requisito do projeto garantir a privacidade das mensagens, estas são enviadas em descoberto. O *digest* enviado é feito a partir de toda a mensagem e cifrado com a chave privada de quem a envia. Quem recebe computa o *digest* da mensagem, obtém do certificado a chave pública de quem enviou, decifra a assinatura e compara os valores. Caso a mensagem seja válida, é verificado se o *nonce* é duplicado, caso em que a mensagem é rejeitada.

Para resolver o problema da distribuição de chaves foram utilizados certificados X.509 emitidos e assinados por uma CA. Durante a instalação cada entidade recebe uma cópia do certificado da CA e uma keystore do java que contém a sua chave privada. Quando uma entidade precisa de validar uma assinatura de outra entidade pode pedir o respectivo certificado à CA através de um web service dedicado para o efeito.

Replicação

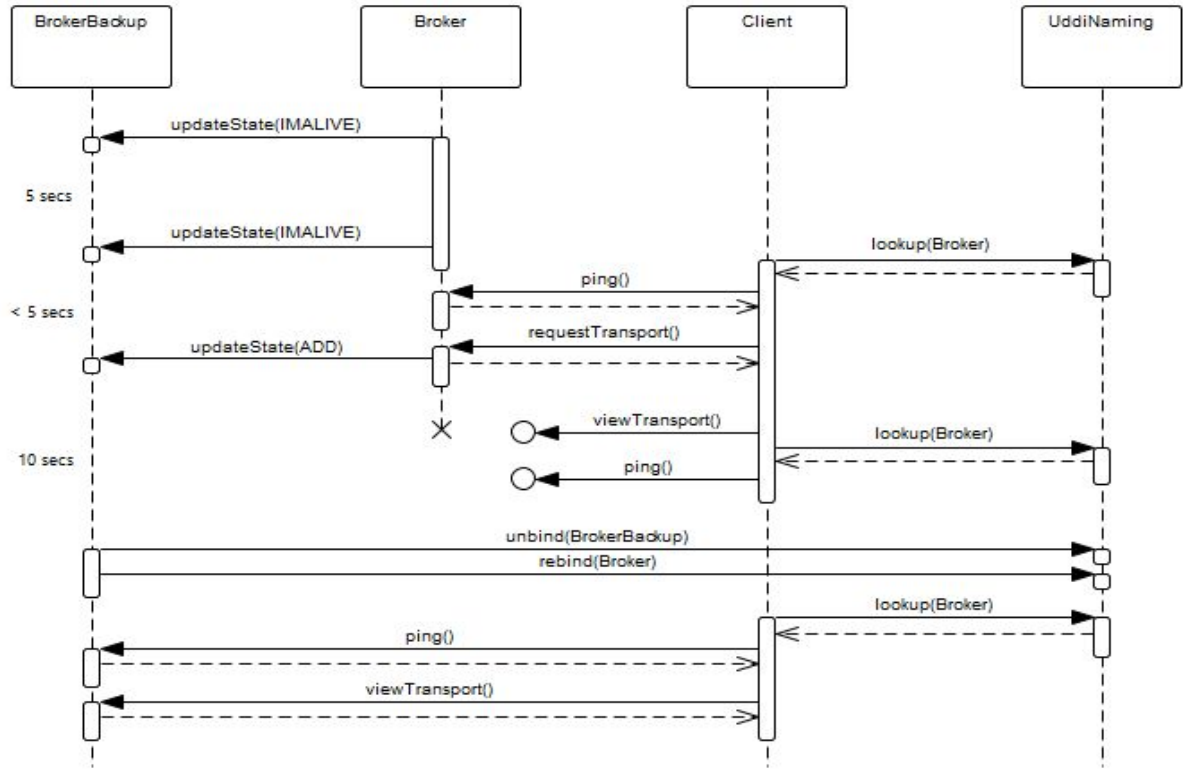


Fig.2: Exemplo de recuperação de falhas do broker

De forma a garantir a tolerância a faltas recorremos a um segundo broker que funciona como backup do broker principal. Tal como demonstrado no diagrama acima, o servidor principal mantém o servidor secundário atualizado através da função *updateState*() que foi adicionada ao contrato WSDL. Esta função permite adicionar um novo transporte, atualizar um transporte existente, limpar a lista de transportes ou simplesmente comunicar ao servidor secundário que o servidor principal continua vivo (IMALIVE). O envio do sinal IMALIVE ocorre automaticamente após terem passado 5 segundos da última comunicação com o servidor secundário. Caso passem 10 segundos sem que o servidor secundário receba sinais de vida do servidor principal este dá-lo como morto e assume o seu lugar.

Como a mudança de servidores não é instantânea adicionámos do lado cliente do broker uma camada de lógica adicional (*BrokerClientFrontEnd*) que lida com as chamadas aos métodos do broker. Assim, caso a chamada a um método do broker falhe o cliente volta a procurar o servidor no servidor de nomes, tenta estabelecer contato (através de *ping*) e repete a chamada ao método mais uma vez. Se o servidor não estiver registado ou não obtiver resposta ao *ping()* então espera 2 segundos e tenta outra vez, até um máximo de 10 tentativas. Desta forma mesmo que o servidor principal vá abaixo durante a execução de um teste é dado ao servidor de backup tempo suficiente para assumir o seu lugar e resumir a execução do teste com sucesso.

Nota: Apesar de não estar representado no diagrama também foi adicionada ao WSDL a função *updateNounce*() que permite atualizar a lista de *nonces* do servidor de backup.