TÉCNICO LISBOA

Instituto Superior Técnico

Universidade de Lisboa

Sistemas Distribuídos 2015-2016
A03

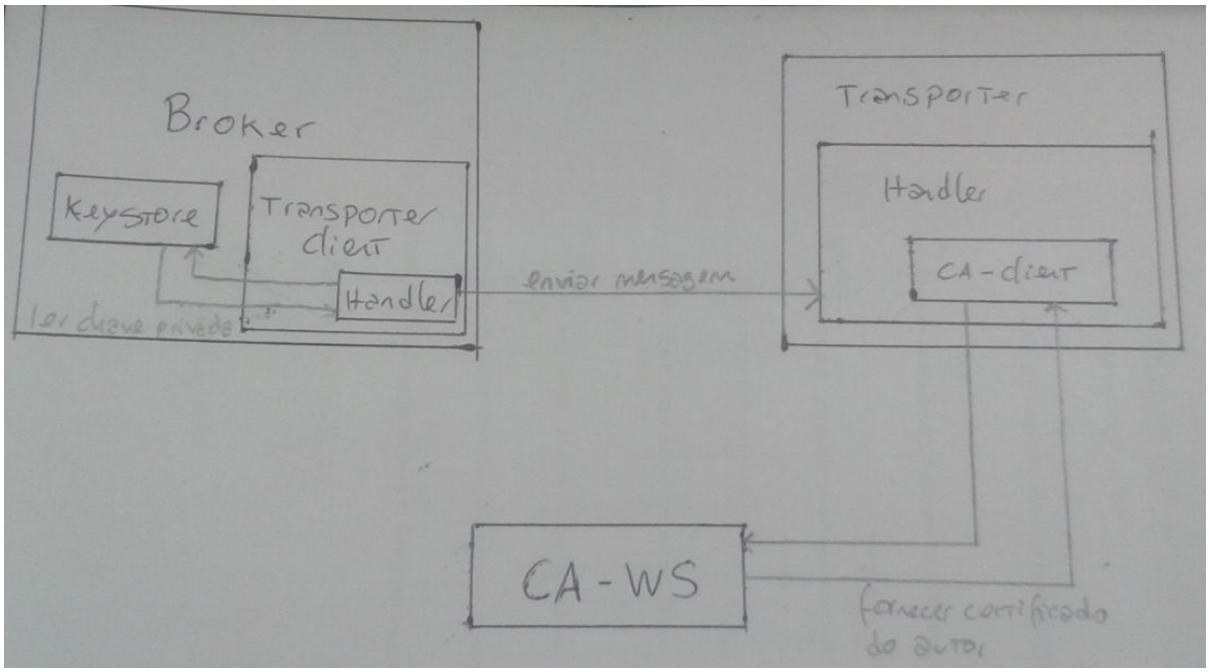
Github: https://github.com/tecnico-distsys/A_03-project

Pedro Bucho 69537

Miguel Amaral 78865

João Figueiredo 75741

Segurança



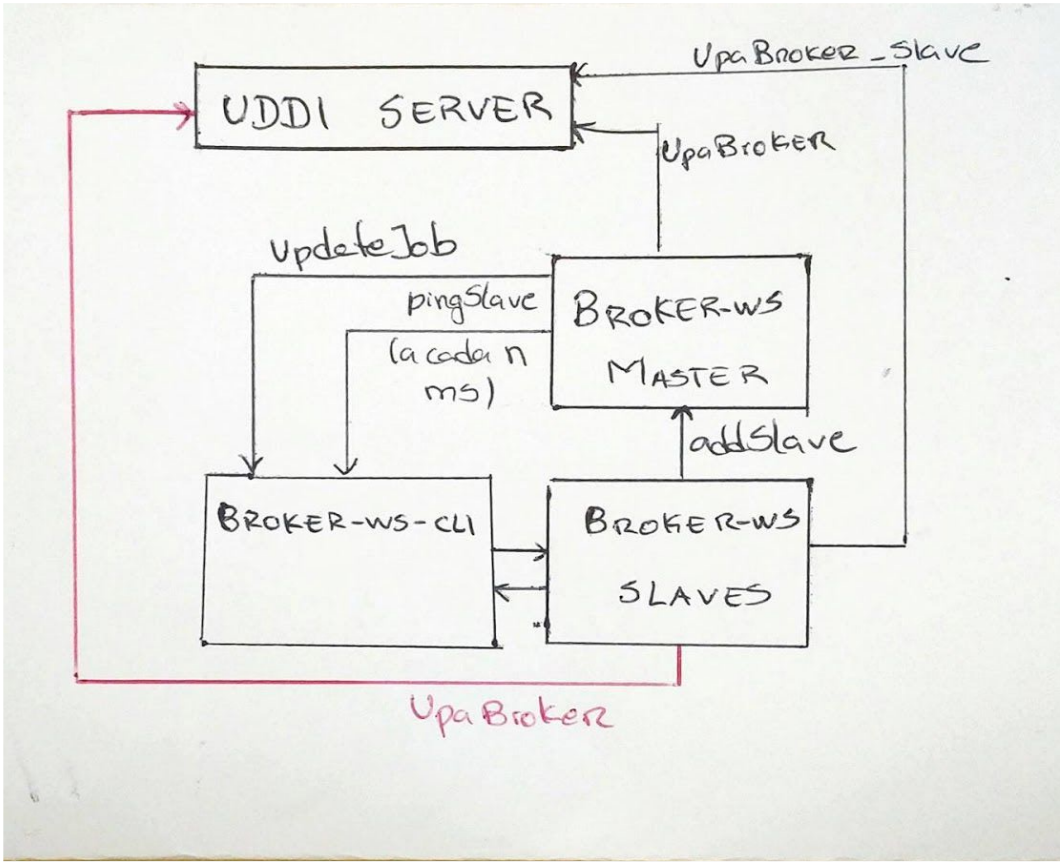
Sempre que é necessário enviar uma mensagem o processo de assinar é igual. (Em cima está representado o processo de envio de uma mensagem do Broker a um dos Transporters).

O Broker instancia um Transporter client que assina (com a chave privada que está numa keystore em disco), o *digest* (SHA-512) do corpo da mensagem concatenado com o *nounce* e *autor* da mensagem. Em seguida, envia-a para o Transporter. O Transporter, através de um CA-client, pede à CA-ws o certificado com a chave pública do autor da mensagem. O Transporter valida o certificado através do certificado da CA (já existente à partida em disco). Após isto faz o *digest* da mensagem da mesma maneira que o Broker fez e compara o resultado com o *digest* que desencriptou (com a chave publica), confirmando assim a assinatura do remetente.

O nounce é composto por um timestamp (número de milisegundos desde a epoch time) aliado a um contador único. Caso a mensagem seja recebida com mais de 30 segundos de diferença é automaticamente descartada. Desta forma é garantida a frescura da mensagem.

A assinatura garante autenticidade e o facto de ser feito com chave assimétrica garante o não repúdio.

Replicação



Quando o UpaBroker se inicia, procura no servidor UDDI por outros Broker. Caso não encontre um registo (ou encontre, mas não consiga fazer “ping” com sucesso), este ativa-se como Master, e regista-se no UDDI como “UpaBroker”.

Caso encontre um registo válido, então regista-se no UDDI como “UpaBroker_Slave”, e chama o WebRequest “addSlave” no UpaBroker master. Assim o master tem conhecimento do Broker slave, e consegue atualizá-lo sempre que necessário.

O Broker master faz “pingSlave” a todos os slaves registados (neste caso, apenas um) em intervalos de tempo constantes.
Caso o BrokerSlave não receba a operação de “pingSlave” durante 5 intervalos de tempo consecutivos, então assume que o BrokerMaster falhou e regista-se no UDDI como UpaBroker (Master).

O UpaBroker Master também atualiza a lista de transportes existentes sempre que é pedido um novo transporte. O estado atual dos transportes não é propagado para o BrokerSlave, porque este consegue obter essa informação consultando os UpaTransporters correspondentes.