# Sistemas Distribuídos

# 2016/17

## Relatório 3ª Entrega do Projeto

# Grupo A58


Rui Ventura

nº 81045

Pedro Cerejo

nº 81338

João Oliveira

nº 81670


**URL Repositório GitHub:** https://github.com/tecnico-distsys/A58-Komparator

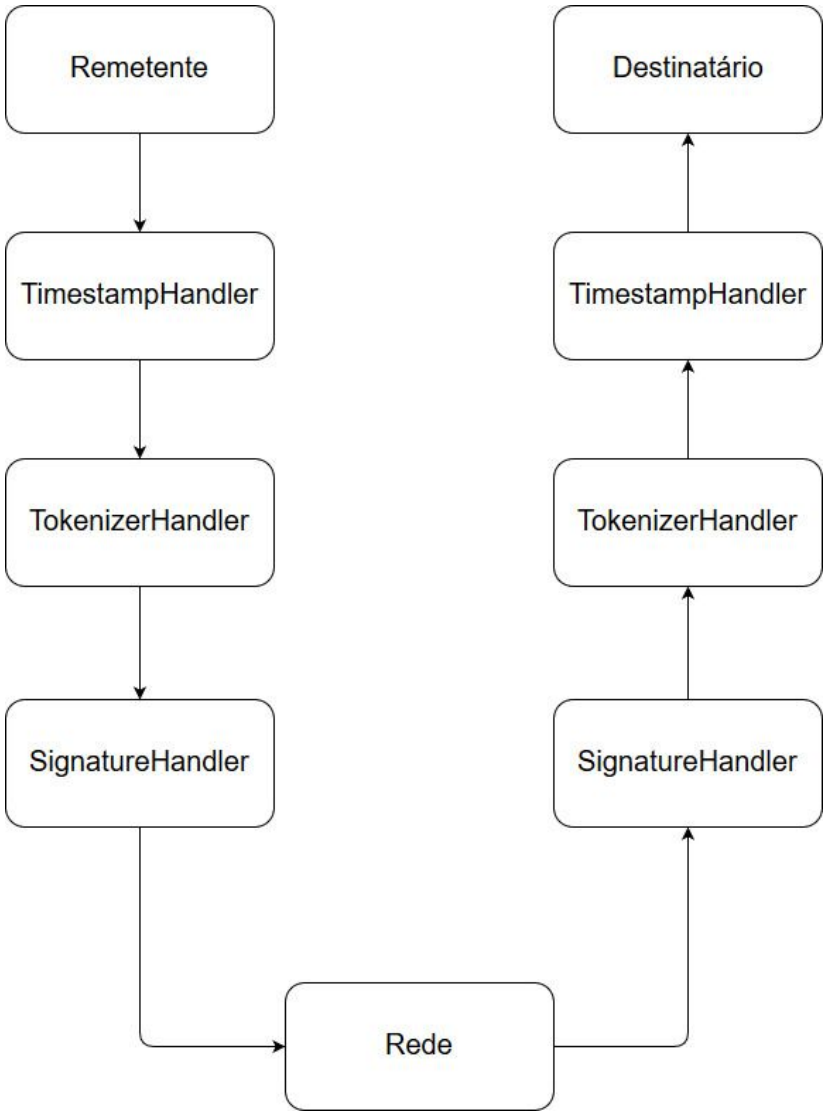

Fig. 1 - Esquema das mensagens SOAP trocadas entre Supplier e Mediator e vice-versa

# Autenticidade

O *handler* SignatureHandler é o responsável por assegurar a autenticidade das entidades envolvidas, nomeadamente o Mediator e os Suppliers, assinando e verificando a assinatura digital das mensagens trocadas entre estes.

Consegue isto seguindo os seguintes passos:

1. Pede o Certificado do emissor da mensagem à Certificate Authority (CA);
2. Verifica se o Certificado recebido foi emitido pela CA (utilizando a sua chave pública para autenticar a sua assinatura neste);
3. Se o Certificado for legítimo, utilizar a chave pública fornecida no Certificado para verificar se a assinatura presente na mensagem corresponde ao digest da mensagem com a chave privada associada à entidade à qual corresponde o Certificado;

Se todos os passos anteriores se verificarem, então o emissor da mensagem está autenticado.

# Frescura

A frescura é assegurada pelos *handlers* TimestampHandler e TokenizerHandler. O primeiro adiciona o timestamp do momento de envio ao header da mensagem quando estas são *outbound* e verifica se a diferença entre esse timestamp e o momento atual é superior a 3 segundos (se verificar esta condição, rejeita a mensagem) se a mensagem for *incoming*.

O TokenizerHandler gera um *token* aleatório utilizando o *SecureRandom*, adicionando o mesmo a uma mensagem *outbound*. Quando é recebida uma mensagem com o dito *token* (*inbound*), este é verificado e, não existindo num conjunto onde os vários *tokens*, é adicionado ao mesmo e a mensagem segue para o próximo *handler*. Se o *token* já existir por já ter sido recebida uma mensagem com o mesmo, a mensagem é rejeitada.

# Encriptação

A encriptação, que cobre o canal *mediator-client*, é aplicada sobre o número do cartão de crédito no acto de verificação do mesmo com o serviço de validação através do *Credit Card client*. Na compra de um *cart*, é feito o pedido ao serviço, sendo a mensagem interceptada pelo EncryptionHandler que encripta o número do cartão de crédito, quando no lado do cliente (*outgoing*) e desencripta o número no lado do servidor (*inbound*).