DEI
DEPARTAMENTO
DE ENGENHARIA INFORMÁTICA
TÉCNICO LISBOA

Sistemas Distribuídos

Grupo 4

75522 Daniel Sil 75714 Miguel Pasadinhas 76012 Carlos Carvalho



Requisito SD-ID.A:

Neste requisito, implementou-se o protocolo Kerberos, de uma forma simplificada, tal como lecionada nas aulas da disciplina de Sistemas Distribuídos.

Num primeiro *round-trip*, um cliente autentica-se através do SD-ID, que serve como servidor Kerberos. Depois dessa autenticação, o cliente recebe um *token* que o habilita a poder usar outros servidores, tais como, o SD-STORE.

Cabe ao SD-STORE verificar se o utilizador está corretamente autenticado (através do *token*). Essa comunicação é feita num segundo *round-trip* do protocolo Kerberos.

Para este requisito, o grupo tomou as seguintes decisões:

- Utilização de cifra AES com CBC, por ser uma solução tida como adequadamente segura e relativamente eficiente.
- Utilização de chaves de 128 bits, por uma questão de simplificação, sendo este um projeto académico, pois, dadas as políticas de segurança do Java, seria necessário instalar ficheiros adicionais para poder chaves de 256 bits, por exemplo, que seriam mais seguras.
- Para criar a chave KC foi usado SHA-1, mais uma vez por uma questão de simplificação. Num projeto "real" (não académico), deveria ter-se usado outra, como por exemplo, SHA-2 ou Blowfish, dado que o SHA-1 tem algumas vulnerabilidades conhecidas, apesar de ainda ser usado.

Sistemas Distribuídos

Grupo 4

75522 75714 76012

Requisito SD-STORE.B:

Neste requisito, implementou-se um sistema de replicação através de *quórums*, com uma garantia relaxada de consistência. Considera-se que não há falhas bizantinas e nunca metade (ou mais) dos servidores do *quórum* falharão ao mesmo tempo.

Foi considerada uma versão mais simples que o protocolo de *quórum consensus* lecionado nas aulas de Sistemas Distribuídos.

Para este requisito, o grupo tomou as seguintes decisões:

- O método que invoca métodos do servidor considera-se bem sucedido se *n + 1* das réplicas de servidores responderem com sucesso, sendo que *n* é resultado da divisão inteira do número total de servidores do *quórum* por 2.
- Caso *n* réplicas de servidores falhem, verifica-se a causa da falha: *timeout* ou exceção de execução. A causa mais frequente é aquela que será apresentada ao cliente, não querendo isto dizer que a outra causa não possa, eventualmente, ter ocorrido. Em caso de igualdade, é dada a exceção de execução. O *timeout* é de 5 segundos.
- No caso de falha, considera-se uma situação de catástrofe, não sendo, por isso, efetuada nenhuma ação de compensação.
- No caso do método para listar documentos receber diferentes listas de documentos das réplicas dos servidores, optámos por fazer um *merge* das mesmas e apresentar o resultado da concatenação ao cliente.