INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

Departamento de Engenharia Informática
Engenharia de Software, Sistemas Distribuídos

Requisitos para a 3ª entrega do projecto

FeaRSe

6 de Maio de 2010

Índice

1 Sumário

Este documento descreve os requisitos para a terceira entrega do projecto conjunto de ES/SD. Os requisitos da terceira entrega incidem sobre aspectos não funcionais de **segurança e transacções distribuídas.** O principal mecanismo a explorar para a implementação destes aspectos são as *STEP* ***Framework Extensions***. As *Extensions*, descritas na secção seguinte, permitem acrescentar código à camada de serviços e à camada de Web Services de forma ortogonal ao código existente, sendo possível a sua activação ou desactivação por simples alteração dos ficheiros de configuração.

Os requisitos a satisfazer são depois descritos neste documento. São também indicados os blocos de requisitos para os quais terão que ser nomeados responsáveis dentro da equipa de desenvolvimento e outras observações relevantes para a realização do trabalho.

Mantêm-se os requisitos do segundo enunciado, aos quais se devem juntar agora os apresentados no resto deste documento (em caso de conflito, deve ser considerado o que consta deste documento).

2 Requisitos

Os requisitos não funcionais são assegurados através da utilização das seguintes soluções tecnológicas: registo dos serviços num servidor UDDI, estabelecimento de canais de comunicação seguros, e implementação de transacções distribuídas.

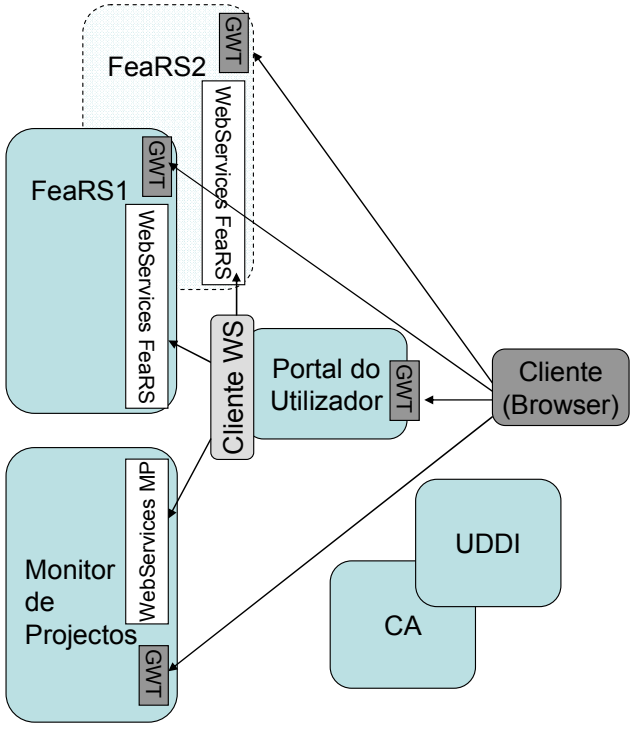

Figura 1 Sistema distribuido para a 3ª entrega

2.1 Registo de serviços

Os Web Services do projecto têm que passar a ser localizados dinamicamente, por intermédio de um **servidor UDDI**. A sua utilização torna a gestão do sistema mais **flexível**, permitindo aos Web Services mudarem de localização.

O servidor de UDDI é central[1], de endereço bem conhecido, definido nos ficheiros de configuração da aplicação.

O registo UDDI tem que ser preenchido com os elementos que caracterizam a *business entity* e os respectivos Web Services, nomeadamente o seu nome e classificação, os Serviços disponibilizados têm de ser registados previamente, indicando o seu endereço (URL) de localização..

A classificação a utilizar para o registo da Certification Authority (CA) deve ser "*Computer or network or internet security: 81111801*".

A classificação a utilizar para os restantes serviços deve ser escolhida de entre a listagem completa disponível na documentação fornecida aos alunos.

2.2 Transacções Distribuidas

Alguns serviços do Portal do Utilizador implicam chamadas a múltiplos web services em diferentes servidores (FeaRS e Monitor de Projectos) para serem completados. Em todos eles, é importante garantir que essas múltiplas invocações remotas são efectuadas como uma única transacção, satisfazendo as propriedades ACID.

Apesar da camada de serviços de cada servidor garantir que cada serviço local é executado no âmbito de uma transacção local na sua base de dados, o sistema actual não suporta transacções distribuídas.

Assim, mais concretamente, a *interface Web* do Portal do Utilizador deverá permitir a um gestor de projecto alterar o estado de forma atómica dos servidores do sistema FEARS*e* como uma única transacção, nomeadamente:

- A alteração do estado de uma sugestão no Portal de utilizador actualizando o estado no FeaRS e no Monitor de Projectos
- O portal do utilizador permite alterar os dados de autenticação de um utilizador de forma agregada envolvendo os vários servidores do sistema FEARS*e*. Assim, a criação de contas de utilizador involve diversos passos para as actualizações nas bases de dados dos vários servidores.

A transacção tem que ser assegurada pelo protocolo de confirmação atómica Two-Phase Commit (2PC) Neste protocolo, existe um coordenador da transacção distribuída (um Web Service autónomo) e existem vários participantes que gerem as transacções locais. Deve considerar-se o 2PC conforme ensinado nas aulas teóricas e descrito no livro base da disciplina de SD (Couloris et al.). Para detectar falhas de paragem de servidores remotos, pode assumir-se um sistema síncrono, em que o tempo de resposta de qualquer servidor tem um limite razoável, definido num ficheiro de configuração.

[1] Por razões de gestão, cada grupo deverá usar um servidor UDDI próprio. Em http://disciplinas.ist.utl.pt/leic-sod/2009-2010/software/uddi-registry-server.zip está disponível um servidor UDDI autónomo que depois de instalado recebe pedidos no endereço http://localhost:9090/RegistryServer/e que pode ser consultado com o utilitário jaxr-browser.

Pretende-se uma solução de transacções distribuídas com uma boa estruturação do programa, separando adequadamente a implementação das transacções da implementação da restante lógica da aplicação. A plataforma transaccional deve ser genérica e reutilizável o que implica que o coordenador, protocolo e *extensions* sejam desenhados de forma independente da aplicação.

- **Nota importante:** a Fenix framework oferece suporte a transacções locais, através de um sistema transaccional (local) chamado JVSTM , que usa controlo de concorrência optimista. O uso do 2PC assume que cada sistema transaccional local oferece uma interface com as operações begin, prepare[2], commit, abort. No entanto, a JVSTM não suporta a segunda operação (prepare). Por essa razão, fará parte do trabalho implementar uma solução adaptada que permita o 2PC correr com a interface limitada do JVSTM.

A implementação das transacções distribuídas **não** pode fazer uso de um motor *Java Transaction Service*, *WS-AtomicTransaction* ou equivalente. Estas normas podem, no entanto, ser usadas como referência.

2.3 Segurança

Os serviços do FeaRS devem ser disponibilizados de forma segura.

Uma vez que os componentes do FeaRS (clientes e servidores) se encontram ligados pela Internet, há várias ameaças sobre o sistema. Nomeadamente, escuta, modificação ou inserção de mensagens, repetição de mensagens antigas, ou servidores fictícios que falsamente assumem a identidade dos servidores legítimos. Pretende-se assegurar a **autenticidade**, **integridade, confidencialidade** e **frescura** dos dados trocados, por forma a proteger o sistema das ameaças acima enumeradas.[3]

2.3.1 Segurança entre browser e servidores

Para a interacção directa entre o *browser* e os servidores web (nomeadamente, Portal do Utilizador, Gestor de Projectos e Fears), deverá passar a usar-se o suporte HTTPS do Tomcat (se estiver correctamente configurado, basta aceder com o protocolo HTTPS no porto 8443, em vez de HTTP no porto 8080).

2.3.2 Segurança entre servidores do FeaRS

Pretende-se implementar canais seguros entre os servidores do FeaRSe. Tal deverá ser assegurado por um protocolo de autenticação de servidores e de distribuição de chaves certificadas aos servidores baseado em cifra assimétrica e usando uma entidade certificadora. Tais canais seguros devem assegurar a autenticidade, integridade, e frescura das mensagens SOAP trocadas nos pedidos e respostas aos Web Services dos servidores.

A solução de comunicação segura deverá separar adequadamente a implementação da segurança da implementação da restante lógica. Deverá também consistir em código genérico e reutilizável. A solução terá que ter em conta os diversos tipos de ataques descritos no livro base da disciplina de SD, como ataques por repetição de

[2] Também chamado *precommit* ou *canCommit*.

[3] A confidencialidade dos dados só é exigida entre browsers e servidores, e não entre servidores – por forma a simplificar o projecto.

mensagens, por exemplo. Por outro lado, as opções tomadas pelos alunos deverão também ter em conta o factor desempenho e os requisitos de segurança das mensagens trocadas entre os servidores do FeaRSe.

Em ambos os componentes de segurança descritos de seguida, os algoritmos de criptografia a utilizar são os implementados na biblioteca JCE do Java.

2.3.3 Comunicação Segura

Os alunos deverão implementar o *estabelecimento de um canal seguro entre servidores, baseado na utilização de certificados digitais de chave pública.*

- Assume-se, por isso, que existe uma Autoridade de Certificação (CA), que deve ser implementada como um servidor autónomo, acessível por Web Services. Deverão ser implementadas funcionalidades na CA, disponibilizadas através de uma interface de Web Services, para assinar certificados de chave pública dos servidores do FeaRS*e*
- Os servidores do FEARS*e*, nomeadamente cada servidor FeaRS, assim como o Monitor de Projectos e o Portal do Utilizador, deverão inicialmente (i) gerar e guardar o seu par de chaves assimétricas (chave privada, chave pública); (ii) ter acesso à chave pública da CA; (iii) gerar o seu certificado de chave pública. A CA, por seu lado, conhecerá a chave pública de cada servidor (para além da chave privada da CA).
- Cada servidor deverá, numa situação inicial, solicitar à CA que assine o certificado da chave pública desse servidor.

Cada servidor FeaRS, o Monitor de Projectos, e o Portal do Utilizador deverão garantir autenticidade, integridade, e frescura dos dados trocados em pedidos e respostas aos Web Services respectivos.

- Sempre que um servidor envie uma mensagem a outro, deverá anexar a essa mensagem o certificado da chave pública do servidor que envia a mensagem. Usando esse certificado, o servidor receptor deverá validar a mensagem.
- Para simplificação, **não será requerida a confidencialidade** das mensagens trocadas entre servidores. Assume-se que tal requisito será implementado, mas não no âmbito deste projecto.

2.3.4 Revogação de certificados

Quando o administrador de um servidor suspeitar que a respectiva chave privada está em risco de ser descoberta, o servidor deve gerar um novo par de (chave privada, chave publica), e solicitar à CA que assine o novo certificado de chave publica desse servidor. A CA deverá, após assinar o certificado, distribui-lo assinado ao servidor[4]. Complementarmente, a CA deverá revogar o certificado da chave pública anterior.

Assim sendo, cada servidor que receba uma mensagem e o certificado associado, para além de testar a validade da chave pública contida num certificado assinado pela CA, deverá também verificar a validade dos certificados.

[4] Mais uma vez, a distribuição da chave deveria ser feita de forma confidencial. No entanto, para simplificação do projecto, ignore o requisito de confidencialidade.

Deverão ser implementadas funcionalidades na CA para revogar certificados, e disponibilizar uma interface para requerer a revogação.

Para implementar a revogação dos certificados, os alunos deverão usar o OCSP – Online Certificate Status:

- A CA tem a capacidade de responder a perguntas feitas on-line sobre a validação de certificados específicos
    - As respostas da CA devem ser transmitidas de forma segura, o que exige que estas respostas têm de ser assinadas digitalmente
- A CA deverá disponibilizar uma interface para um cliente aceder a esta funcionalidade
- Um servidor, quando recebe uma mensagem, deverá não só verificar a validade do certificado enviado com essa mensagem, como também questionar a CA sobre se o certificado já foi revogado.

2.4 Integração

Os alunos deverão integrar o trabalho de forma a funcionar no final com transacções distribuidas e segurança.

3 Blocos de requisitos

A divisão dos membros do grupo pelos blocos de requisitos de que são responsáveis tem que ser equilibrada.

Para começar, os grupos da entrega anterior serão também responsáveis pela concretização do registo e pesquisa no UDDI do seu Web Service: FEARS e Monitor de Projectos.

Para esta entrega os seguintes blocos de requisitos terão que ter responsabilidade explícita, decidida dentro do grupo e definidas as respectivas equipas na ferramenta de gestão de projecto, até à data da próxima reunião de gestão de projecto com o docente de laboratório de ES:

1. Implementar transacções distribuidas
   equipa `tran`
2. Implementar a segurança, assim como a assinatura e revogação de certificados
   equipa `sec`
3. Integração
   Ambas as equipas

Cada grupo de responsáveis terá que elaborar um breve **relatório** (com 4 páginas no máximo, excluindo índices) explicitando em detalhe qual o problema que resolveram e apresentando a solução produzida, indicando as suas principais virtudes, limitações e quais os aspectos não implementados. Os alunos deverão indicar no relatório de forma clara todas as opções tomadas.

O relatório terá de seguir uma *template* que será fornecida posteriormente.

Cada grupo terá também que conhecer a solução implementada pelos restantes grupos para os outros blocos de requisitos. Não se exige aos elementos de um grupo que influenciem ou defendam as opções tomadas por outro grupo, mas sim que tenham uma opinião crítica sobre as mesmas.

4 Gestão do projecto

Nesta terceira fase, a avaliação dos alunos de Engenharia de Software centra-se na capacidade de gestão do projecto da equipa.

Os alunos que apenas estão a realizar a cadeira de Sistemas Distribuídos ficam dispensados da avaliação da componente de gestão do projecto, efectuando apenas uma gestão informal com os docentes do laboratório.

Em resumo, pretende-se que os alunos sejam capazes de efectuar o levantamento de requisitos, especificar os diversos componentes a desenvolver, e estruturar o trabalho necessário em conjuntos de tarefas que demonstrem um evoluir constante do trabalho, ao longo do tempo disponível para a sua realização. É fundamental ser capaz de estruturar e dividir o trabalho por forma a conseguir uma evolução incremental e sustentada do trabalho. O docente do laboratório de ES fará o acompanhamento dessa gestão, em reuniões semanais.

4.1 Aplicação da metodologia SCRUM

O desenvolvimento desta entrega deve seguir o processo de desenvolvimento SCRUM, com *sprints* de duas semanas (14 dias) e uma distribuição de trabalho equilibrada entre os membros das duas equipas. Os procedimentos a seguir estão descritos na página da disciplina, sendo que no início de cada *sprint* deverá ser entregue via Fénix a folha de gestão (http://disciplinas.ist.utl.pt/leic-es/2009-2010/proj/scrum-3.xls) com a execução do *sprint* anterior, e o planeamento do novo *sprint* já preenchidos, compactados num único ficheiro ZIP. Todos estes documentos devem ter o nome *grupo-sprint* (em que *grupo* é o nome do grupo, correspondente ao módulo CVS; e *sprint* é o número do sprint a que o documento diz respeito - exemplo: A0315-2 seria o nome dos ficheiros do grupo A0315 relativos ao 2º sprint).

- A folha de gestão deve ser actualizada diariamente e, apresentada **actualizada e em papel** na reunião de gestão semanal[5] para análise.

A divisão do trabalho pedido é deixada ao critério do grupo, sendo alvo de avaliação na componente de gestão de projecto e mantendo-se a sugestão de seguirem uma abordagem *pair programming*.

5 Grupos só SD

Os grupos só SD (ou de ES/SD com até três elementos), assim como os trabalhadores estudantes a realizar apenas a disciplina de SD e a realizar o projecto individualmente, terão que satisfazer apenas os requisitos de UDDI para o FeaRS e o bloco de requisitos de segurança.

6 Entrega

A segunda entrega tem duração de 20 dias úteis, sendo a hora limite de entrega as 20:00 do dia 2 de Junho. As regras para o processo de entrega do trabalho através do repositório de CVS do grupo estão descritas no documento "Utilização do CVS no projecto" (ver http://disciplinas.ist.utl.pt/leic-es/2009-2010/proj/RegrasCVS.html)

[5] Durante o horário de laboratório em que o grupo está inscrito.

- A etiqueta a colocar para indicar a entrega da terceira fase do projecto é **RELEASE_3. A ausência da etiqueta será interpretada como não tendo sido entregue.**

7 Avaliação

Ao longo do projecto os aspectos de gestão de projecto vão sendo avaliados continuamente. Os aspectos relacionados com o desenvolvimento da aplicação em si são avaliados nos seguintes passos:

1. Avaliação do código desenvolvido;
2. **Apresentação** do projecto
   - cada grupo deve planear a apresentação do projecto para no tempo definido demonstrar o seu funcionamento, o cumprimento de todos os requisitos assim como optimizações de desempenho implementadas
3. **Discussão final** com perguntas individuais a cada elemento do grupo.

A primeira parte da avaliação do trabalho desenvolvido é realizada pelo corpo docente e consiste na avaliação do projecto do ponto de vista da correcção da solução e do cumprimento das normas de utilização da arquitectura.

O calendário de avaliações e mais pormenores sobre o processo serão publicados oportunamente. A nota da discussão final é individual, corresponde à nota final da componente laboratorial e será decidida com base no desempenho individual de cada elemento do grupo na discussão, bem como nas notas ponderadas de cada entrega do grupo.

- O grupo deve preparar a demonstração nos PCs do laboratório, obtendo o projecto **a partir do repositório CVS**, efectuando o *deploy* e exercitando as várias funcionalidades desenvolvidas.[6]
- O grupo deverá distribuir a solução, instalando cada servidor FeaRS, Monitor de Projectos e Portal de Utilizador em máquinas diferentes

FIM DO ENUNCIADO

[6] A utilização do Eclipse é facultativa, pelo que fica ao critério de cada grupo decidir sobre a sua utilização ou não.