MM - Proposito da Arquitetura

Propósito da Arquitetura

Após analisar o histórico de projetos já desenvolvidos na empresa, foi possível identificar diversos padrões que poderiam ser aplicados a fim de eliminar as divergências detectadas. Estes trabalhos utilizavam uma arquitetura JEE baseada em frameworks de mercado com a proposta de facilitar o processo de desnvolvimento, agilizando as tarefas comuns no dia-a-dia dos desenvolvedores. Infelizmente, por falta de suporte, documentação e treinamento, erros comuns, muitas vezes básicos, prejudicaram as entregas. Consequentemente, a base de código-fonte que deveria ser sólida e limpa, tornou-se desestruturada e de difícil manutenção.

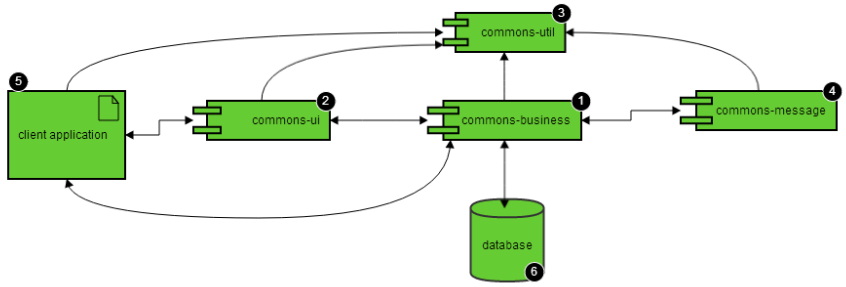
Visando eliminar os problemas previamente citados, uma nova arquitetura, também composta por frameworks de mercado, foi proposta. Com os objetivos traçados, inicia-se a construção de uma nova arquitetura, divididas em três camadas, utilizando o padrão arquitetural MVC para compor as camadas de apresentação, controle e modelo. Estas camadas, por sua vez, também são compostas por padrões de arquitetura e adaptando-os a sua necessidade.

A camada de apresentação utiliza o framework AngularJS, o mais novo lançamento do time de desenvolvedores do Google. Diferentemente de outros frameworks JavaScript, ele adota uma abordagem mais ligada à sintaxe HTML, funcionando como uma espécie de extensão da linguagem. Componentes padrões, como por exemplo laços de repetição (ng-repeat), já estão inclusos no framework. Entretanto, novos componentes customizados foram desenvolvidos para auxiliar nossos desenvolvedores. Estes componentes são classificados no AngularJS como directives, filters, providers.

A camada de serviços faz uso da especificação JSR311 e sua implementação de referência Jersey. Em conjunto, ambos buscam simplificar o desenvolvimento de serviços REST e oferecer uma forma padrão de implementação desta linha de serviços. O Jersey implementa os recursos definidos pela JSR e acrescenta algumas funcionalidades, como o WADL e uma API para clientes REST. Novamente, a arquitetura provê métodos auxiliares, nos quais implementam operações CRUD através de endpoints RESTful.

Por fim, temos a camada de negócios e as diferentes implementações de frameworks que disponibilizam o mapeamento de objeto-relacional, são eles: Hibernate, AO, TopLink. Nesta camada foi possível criar abstrações para que seja transparente as operações nativas destes frameworks, isto é, o desenvolvedor não preocupa-se com detalhes de qual framework está sendo utilizado pela aplicação. Contudo, estas não são as únicas funcionalidades da camada. Validações de modelo e regras de negócio também estão contidas neste terceiro nível.

Assim, após a apresentação formal do propósito da arquitetura, concluimos esta seção com uma ilustração da comunicação e dependência entre as camadas. Nesta ilustração é possível perceber a interação entre as diferentes responsabilidades e componentes.



e

Onde:

Item	Nome	Descrição
1	COMMONS-BUSINESS	É a camada responsável por disponibilizar os métodos padrões de negócio e de serviço.
2	COMMONS-UI	É a camada da arquitetura que provê soluções para as interfaces de usuário.
3	COMMONS-UTIL	
4	COMMONS-MESSAGE	
5	PRODUCTS	São os produtos desenvolvidos pela empresa que consomem a arquitetura proposta. Como exemplo, podemos citar Shield, Catalog Service.

6	DATABASE	