



Access Management with Flexess

White Paper



JresearchSoftwares.r.o.

Dolnokrčská 2019/43a

Praha 4, Krč

Prague, 140 00

Czech Republic.

January 2008

Introduction

The name *Flexess* comes from: FLEXible accESS. Flexess is a component-based product, which adds access management capabilities to the business application. The product provides the entire infrastructure starting from a tool, which allows creating an application security model to the administrative web application, which manages the assignments of the different access privileges to the users.

Problem Statement

Almost every application requires some protection of the unauthorized disclosure. Clients of the bank must perform operations only with their accounts; some movies must be inaccessible for people under eighteen; managers may increase the salary only for the employees of their department. All these activities require checking if the user is authorized to perform this operation.

Although the problem of access management is a key factor in the development of the enterprise applications, even experienced architects leave the process of the designing the access rules to the last step. One problem is that it is often not clear how to separate access and business logic rules. Another problem is that each application level (database, business logic, and web-tier) has its own access management model defined in its own terms. For example, web-tier protects resources (URLs), business logic requires a protection for Java objects and database defines permissions for the tables and columns. All these circumstances make difficult to create a consistent access control model for the whole application.

Solution

The solution is to create the product, which will automate most of the tasks related to the development of access management features. The product must give possibility to define under which conditions the objects are accessed. There should be a uniform way to model access rules for the all types of objects like resources, attributes, database tables. There must be also a ready-to-use application with a user-friendly web interface that allows to the administrator managing the security aspects, for example, create new roles or assign the roles to the users.

Flexess product provides a solution, which allows to almost completely separate the logic of access management from the business logic of the application.

Main features of Flexess are:

- **Security Modeling** - Gives you possibility to create a model of the security features of the application. You can create a protected object *Order*, specify operations like *create*, *read*, *update*, *delete* and provide constraints under which these operations are accessible.
- **Role-Based Access Control** - All the security aspects are modeled in terms of *users*, *roles* and *permissions*.
- **Parameterized Permissions** – It is possible to specify constraints under which the objects are accessed. For example, '*the users are authorized to modify the objects, which were created by themselves*'.
- **Authorization** - The minimal code can be done for authorization. To check if the user is authorized to create an order, you will need to invoke `check(userId, order, "create")`.
- **Authentication** - Flexess provides an authentication module and can be integrated with existing authentication frameworks.
- **Web Administration** - Allows managing users, roles and permissions through the web interfaces.

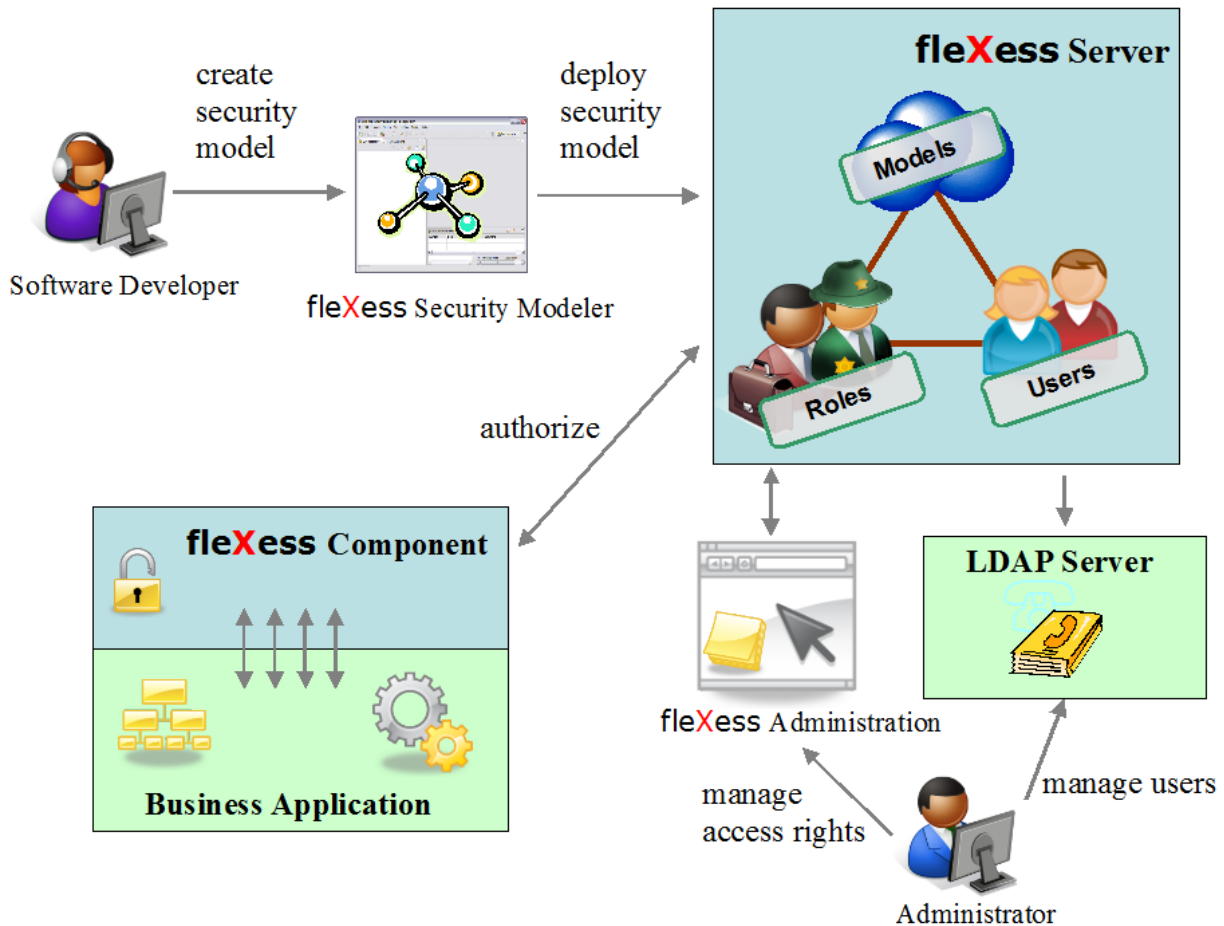
- **User Management** - Maintains a database of users with customized user profiles.

Flexess also provides a feature, which makes this product unique:

- **Parameterized Permissions.** Permissions in Flexess are not strings, but objects, which can be parameterized with the attributes. For example, *RegionalPermission* has an attribute *region* and a constraint „*Order.region=RegionalPermission.region*“. It means that this permission will be satisfied, when the *region* attribute of *Order* matches the *region* attribute of the permission.

Working with Flexess

The scenario of working with the Flexess is shown in the picture below:



1. Software developer create a security model of the business application in Flexess Security Modeler.
2. The model is deployed to Flexess server.
3. After the model has been deployed administrator can manage access rights in run-time, for example, assign roles to the users.
4. When the user accesses the business application, this application invokes Flexess Client Component to check if the user is authorized. Flexess Client Component connects to Flexess server, retrieves the roles of the user, caches this information on the client side and checks the user authorization.

Competitors

Though security is a very hot theme there are not so many products, which provide solutions in this area.

Acegi (http://acegisecurity.org/)	Acegi is a popular open source security framework for Java enterprise applications. The authorization model allows specifying security only on the method level. Acegi does not provide any tools, which facilitate administering the access privileges.
PortSight Security Access (http://www.portsight.com/)	PortSight Secure Access is a component for .NET developers. The authorization model is very simple, permissions are just strings, which can not be parameterized. PortSight Secure Access also has a web-based tool, which allows administrating the security features. This product is the most similar to Flexess, but it is available only for .NET platform.
IBM Tivoli Access Manager http://www-306.ibm.com/software/tivoli/	IBM Tivoli Access Manager enables the integration of Access Manager applications that provide a wide range of authorization and management solutions. Though this line of products is very powerful, it is also very heavy and expensive.