Servicios de seguridad ASPNet

El bloque Fwk.Security provee de ciertos servicios que ayudaran al desarrollador implementar el modelo de seguridad de una manera más estándar.

El siguiente gráfico muestra la interconeccion de componentes que definen la arquitectura de los servicios de Fwk.Security. El gráfico nos permitirá tener una visión mas global de donde se ubican y como se consumen.-

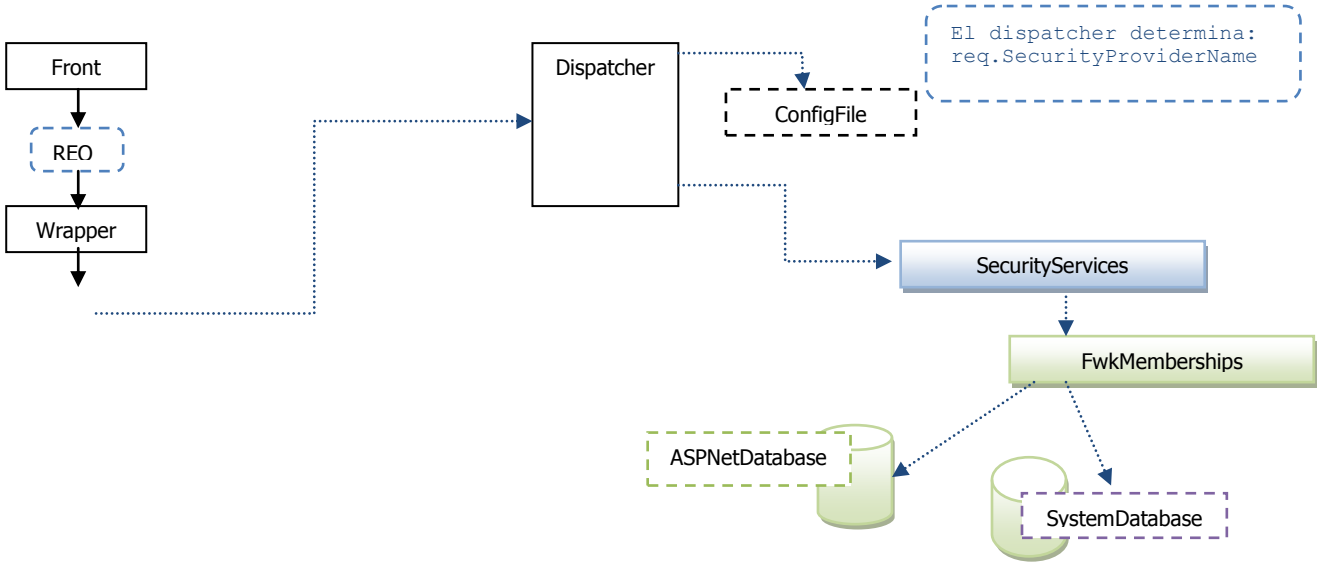


Figura 1.0 Modelo de de servicios de seguridad en la arquitectura

Los servicios de seguridad se encuentran alojados en un dispatcher el documento *Arquitectura Tecnológica.Security.doc* explica bien que proveedores de seguridad (Membership y Roles) deben estar configuradosen el .config

La siguiente tabla muestra una lista de los servicios que existen en el bloque de seguridad.-

Usuarios

1. AuthenticateUser
2. CreateUsers
3. GetUserInfoByParams
4. RemoveUserFromRole
5. ResetUserPassword
6. SearchPelsofters
7. SearchUsersByParam
8. UpdateUser
9. ValidateUserExist

Reglas y categorías de reglas de usuario

10. CreateRule
11. SearchAllRules
12. UpdateRulesService
13. CreateRulesCategory
14. DeleteRulesCategory
15. SearchAllRulesCategory
16. SearchRulesCategoryByParam

17. UpdateRulesCategory
18. AssignRolesToUser

Roles de usuario

19. CreateRole
20. DeleteRole
21. SearchAllRoles
22. SearchRolesForUser

Que configurar para poder utilizar un proveedor de seguridad ?

Indicador de Proveedores de seguridad en el dispatcher

Lo primero que debemos configurar es el dispatcher, debido que generalmente se utilizaran los componentes de seguridad desde un despachador de servicios (win service o IIS) y estos pueden alojar servicios de **diferentes aplicaciones o** de una aplicación pero para **diferentes empresas,** es necesario definir un conjunto de proveedores para cada independencia de aplicación y/o empresa.-

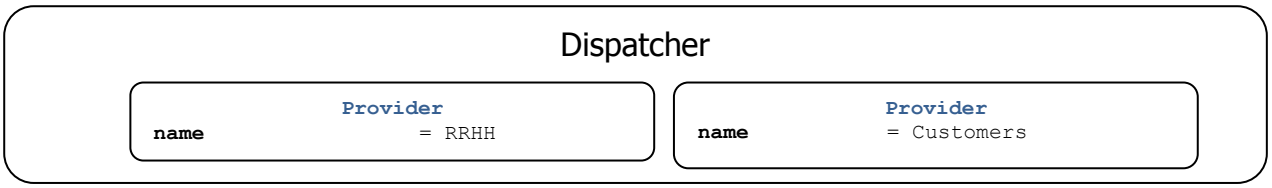


Figura 2.0

En la figura anterior tenemos un despachador con dos proveedores, ”RRHH” y “Customers”. Lo que nos indica que cada uno tiene su seguridad particular.

RRHH podría apuntar a la misma base de datos que Customers pero con un nombre de aplicación distinto.

El archivo de configuración entonces debería contar con estas dos configuraciones:

```
<roleManager defaultProvider="Customers" enabled="true" cacheRolesInCookie="true" cookieName=".ASPROLES" cookieTimeout="30"
cookiePath="/" cookieRequireSSL="true" cookieSlidingExpiration="true" cookieProtection="All">
                <providers>

                <add name="customers" type="System.Web.Security.SqlRoleProvider"
                        connectionStringName="connectionStringSec"
                        applicationName="ventas"/>

                  <add name="RRHH" type="System.Web.Security.SqlRoleProvider"
                        connectionStringName="connectionStringSec"
                        applicationName="crm"/>
                </providers>


        </roleManager>
        <membership>
                <providers>
                        <add name=" customers " type="System.Web.Security.SqlMembershipProvider, ..."
                        connectionStringName="connectionStringSec"
                        enablePasswordRetrieval="false"
                        enablePasswordReset="true" requiresQuestionAndAnswer="true"
                        applicationName="ventas"
```

```
                    requiresUniqueEmail="false"
                    passwordFormat="Hashed" maxInvalidPasswordAttempts="5"
                    minRequiredPasswordLength="7" minRequiredNonalphanumericCharacters="1"
                    passwordAttemptWindow="10"    passwordStrengthRegularExpression=""/>

                <add name="rrhh" type="System.Web.Security.SqlMembershipProvider, ..."
                    connectionStringName="connectionStringSec"
                    enablePasswordRetrieval="false"
                    enablePasswordReset="true"
                     requiresQuestionAndAnswer="true"
                    applicationName="crm"
                    requiresUniqueEmail="false"
                    passwordFormat="Hashed" maxInvalidPasswordAttempts="5"
                    minRequiredPasswordLength="7" minRequiredNonalphanumericCharacters="1"
                    passwordAttemptWindow="10"    passwordStrengthRegularExpression=""/>
            </providers>
        </membership>
</system.web>
```

Dada la flexibilidad de configuración de los proveedores de seguridad podríamos llegar a disponer en nuestro host de servicios las siguientes situaciones:

1) Dos **aplicaciones** diferentes, Customers y Sales que comparten la seguridad.

- Cadenas de conexión:

```
<add name="CommonSecurity" connectionString="Data Source=Server2;Initial Catalog=
    PelsoftSecurity …
<add name="Sales" connectionString="Data Source=Server2;Initial Catalog=SALES
<add name="Customers" connectionString="Data Source=Server1;Initial Catalog=Customers …
```

- Proveedores de seguridad: *Customers y Sales comparten el mismo proveedor*

Name	**CommonSecurity**
ApplicationName	**crm**
connectionStringName	**CommonSecurity**

2) Dos **empresas** diferentes que utilizan la misma aplicación "Customers" y no desean compartir la seguridad

```
<connectionStrings>
<add name="Personal" connectionString="Data Source= server;Initial Catalog=Micro
<add name="Datacom" connectionString="Data Source= server;Initial Catalog=Customers …
```

Personal SA SecurityProvider

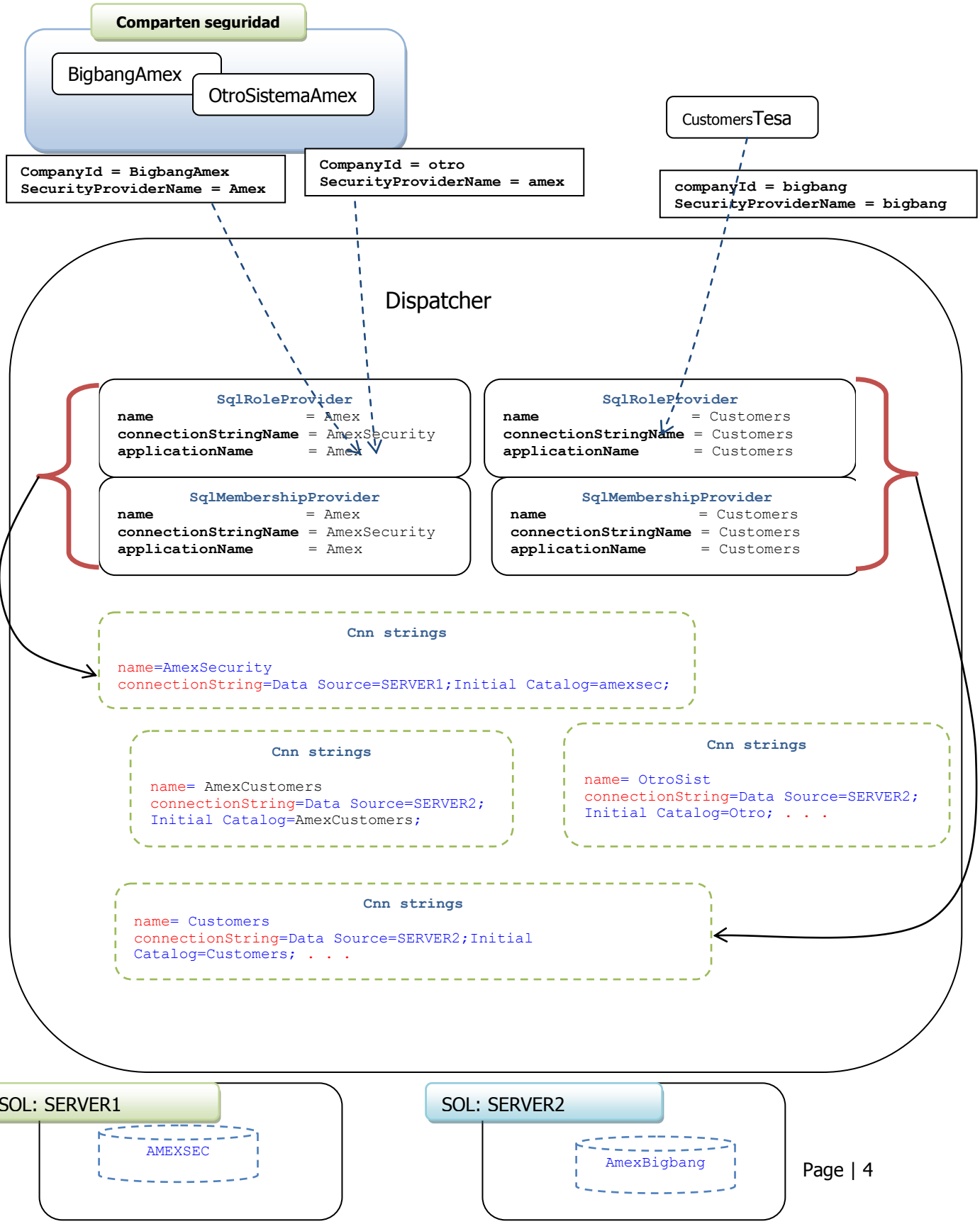
Name	**Personal**
ApplicationName	**PelsoftCustomers**
connectionStringName	**Personal**

Datacom SecurityProvider

Name	Datacom
ApplicationName	DatacomCustomers

connectionStringName	Datacom

La siguiente figura representa gráficamente lo que acabamos de explicar:



Indicador de Proveedores de seguridad en Front-Eend

Las aplicaciones clientes que consumen estos servicios **NO** deberán indicar que proveedor de seguridad utilizaran atreves de los servicios.-

El despachador de servicio es quien identifica cual proveedor de seguridad usar a través de un atributo **securityProviderName** en la configuración de la metadata.

SecurityProviderName*: Permite identificar los proveedores de seguridad de membership provider, rol provider y rule provider.- Es decir, permite identificar el origen de datos estándar que propone el framework Fwk.-*

Por lo tanto las aplicaciones Front-End deben contar con algún mecanismo para poder obtener este valor y establecérselo a los Request que envía el servidor.- Esto se hace mediante la correcta configuración del Wrapper (del lado del cliente) por medio del atributo **serviceMetadataProviderName**

De esta manera podemos tener un cliente configurado de la siguiente manera.-

```
<FwkWrapper defaultProviderName="Customers">
  <Providers>
        <add name="Customers"
               serviceMetadataProviderName ="customersSecurity"
               appId ="pelsoft"

               type="Fwk.Bases.Connector.RemotingWrapper,Fwk.Bases.Connector"
               sourceinfo="remotingcnfgfile.config" />
               </Providers>
</FwkWrapper>
```

Y un dispatcher server de la siguiente:

```
<!--Configuracion del metadata de servicios del frameworck -->
<FwkServiceMetadata defaultProviderName="customersSecurity ">
  <Providers>
         <add name="customersSecurity " type="sqldatabase"  sourceinfo="test" securityProviderName="sec_ customers" />
  </Providers>
</FwkServiceMetadata>
```

Puede entender la lógica de estas configuraciones dirigiéndose al documento: *Arquitectura Tecnológica Configuración Metadata de Servicios.docx*