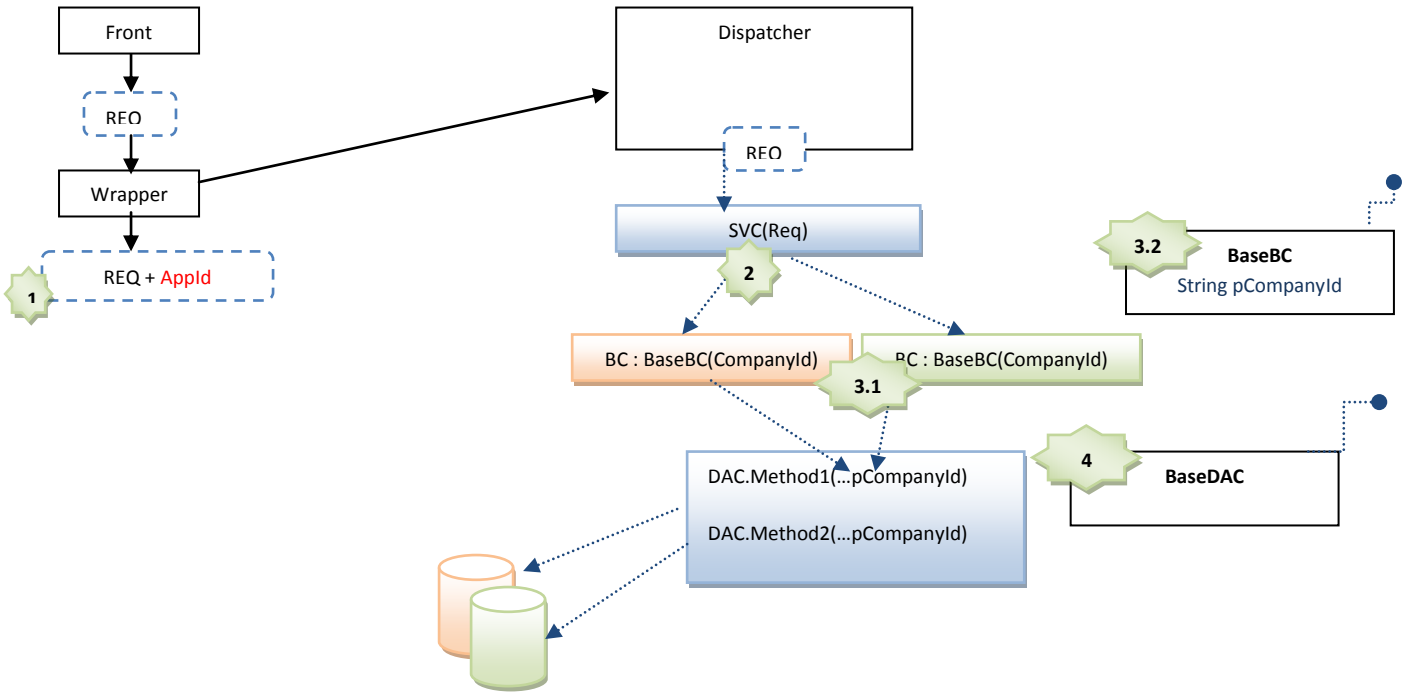
Modelo para identificar empresa en la arquitectura

La identificación de la empresa en los componentes de servicios será denotada por el atributo AppId.

El siguiente grafico muestra el modelo de todo el circuito por el que pasa AppId para desde el frontEnd hasta su destino final.-



Front End (1)

El frontEnd crea una instancia del Request. El request dado sus herencias tiene la capacidad de obtener automáticamente la información de contexto, esta información se aloja en una clase llamada ContextInformation la cual contiene el campo CompanyId.

Para obtener AppId el framework lo hace a través de la configuración del wrapper correspondiente y del proveedor de wrapper que se le pace .-

```
<FwkWrapper defaultProviderName="bigbang">
```

```
  <Providers>
       <add name="bigbang"
              securityProviderName ="PROV_NAME"
              appId ="NOMBRE_EMPRESA"

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

Back End

Servicios (SVC)

Los servicios en la capa SVC deberán pasar esta información a la BC o DAC según que se esté utilizando directamente.- (2)

```
CreateUsersService : BusinessService<CreateUsersRequest, CreateUsersResponse>
    {
        public override CreateUsersResponse Execute(CreateUsersRequest pServiceRequest)
        {
            CreateUsersResponse wRes = new CreateUsersResponse();

            UserBC wUserBC = new UserBC(pServiceRequest.ContextInformation.AppId);

        }
```

Componenetes de negocio (BC)

Las clase BC heredan de BaseBC la cual exige una instanciación con un parámetro mínimo. Este es el CompanyId.- (3.2)

La clase BC se instancia de la siguiente manera como ejemplo:

(3.2)

```
UserBC wUserBC = new UserBC(pServiceRequest.ContextInformation.AppId);
```

Dentro de la clase BC existene metodos que seguramente llamaran a componentes DAC, si estos componetes lo exigen en la firma de sus metodos, las BC seran responsables de pasarle su propiedad CompanyId. (3.1)

Componente de acceso a datos (DAC) (4)

En la DAC el CompanyId puede ser utilizado de dos maneras una directa y otra indirecta

o

- **Directa** para obtener directamente la cadena de conexión

  El CompanyId en la DAC es utilizarlo directamente como clave de cadena de conexión. Esta clave está configurada en el archivo de configuración y la sección connectionStrings

```
Database wDataBase = DatabaseFactory.CreateDatabase(pCompanyId);
```

  Este mecanismo es el utilizado cuando los componentes de acceso a datos son de las Microsoft.Practices.EnterpriseLibrary.Data.

- **Indirecta**
  Utilizada como clave para identificar la cadena en un archivo de configuración.
  Las clases DAC heredan de la clase BaseDAC la cual tiene la responsabilidad de obtener la cadena de conexión a través del método GetCnnstring.

  Pasos para obtener la cadena de conexión por medio de GetCnnstring

  1 Retorna la cadena con el nombre de ***CompanyId*** configurada en la sección connectionStrings. Si no lo encuentra va al paso 2.-
  2 Llama componentes Fwk.Configuration y busca un grupo llamado ***BackFwkSettings*** y la clave con nombre **p*CompanyId*** que se pasa como parámetro. Si existe la clave y grupo en el archivo de configuración retorna este valor si no va al paso 3
  3 Se retorna String.Emty y opcionalmente el desarrollador puede establecer una forma personalizada para rellenar este valor.-

```
public static List<Clients> SearchAll(string pCompanyId)
{
 using (ClientsDataContext dc = new ClientsDataContext(GetCnnstring(pCompanyId)))
 {
       IEnumerable< Clients > list = from s in dc. Clients select s;
       return list.ToList<Clients>
 }
}
```