

The SOA Gateway & Limited Integration Projects

This document is intended to give technical architects and project managers a detailed view of how the SOA Gateway can help them where limited integration with existing data and business logic is required in their projects.

This document is distributed for information purposes only and does not form part of or constitute an agreement with Risaris Ltd. Although Risaris Ltd. uses reasonable efforts to include accurate and up-to-date information in this document, Risaris makes no warranties or representations as to its accuracy. Risaris Ltd. may also make improvements and/or changes to this document at any time without notice. The various approaches outlined in this document are put forward in good faith, but it remains possible that individual results may vary. For that reason and in accordance with standard practice, readers are encouraged to test any materials developed on the basis of this paper before putting them into productive use.

INDEX

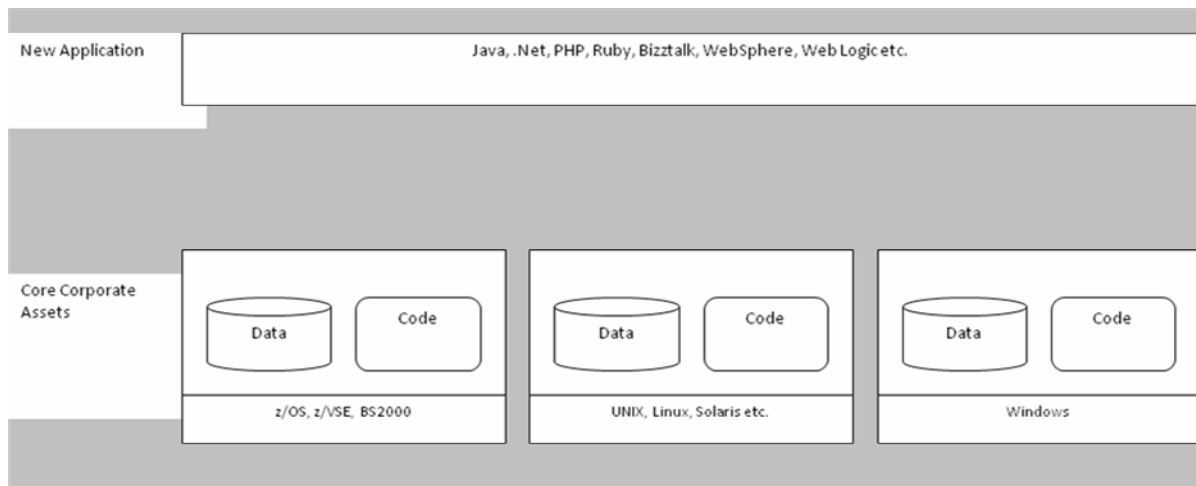
1.	Introduction.....	3
2.	The Business Case.....	3
2.1.	Using the Traditional Approach.....	3
2.2.	Using the SOA Gateway.....	6
2.3.	The Benefits to this and future projects.....	7
3.	Implementation and Using the SOA Gateway.....	8
3.1.	Installation of the SOA Gateway.....	8
3.1.1.	Registering to use the SOA Gateway.....	8
3.1.2.	After the download.....	10
3.1.3.	Installing and Configuring the SOA Gateway Server.....	10
	Windows Installation.....	10
	Other Platform Installation.....	11
	Common Configuration Steps.....	12
	Supporting Multiple Platforms and Operating Systems.....	12
3.2.	Creating the Services.....	13
3.2.1.	Creating Database Services.....	13
3.2.2.	Creating Business Logic Services.....	14
3.2.3.	Registering Services in a UDDI Server.....	16
3.3.	Using the Services in your project.....	16
3.3.1.	Using the Services Directly (without a UDDI Server).....	16
3.3.2.	Using the Services from a UDDI Server.....	17

1. Introduction

There are many projects that need to be completed in organizations with existing data and business logic where the new project could use limited access to this data and business logic. In such cases it is generally not very cost effective with traditional integration tools and technologies to make this available directly. This leads to short term solutions to sometimes leading to copies of the data being made or the implementation the business logic a second time in the new language. The SOA Gateway technology can help an organization to avoid such implementations by making the data and business logic available quickly, simply and in a cost effective manner such that the new project can reuse these core assets.

2. The Business Case

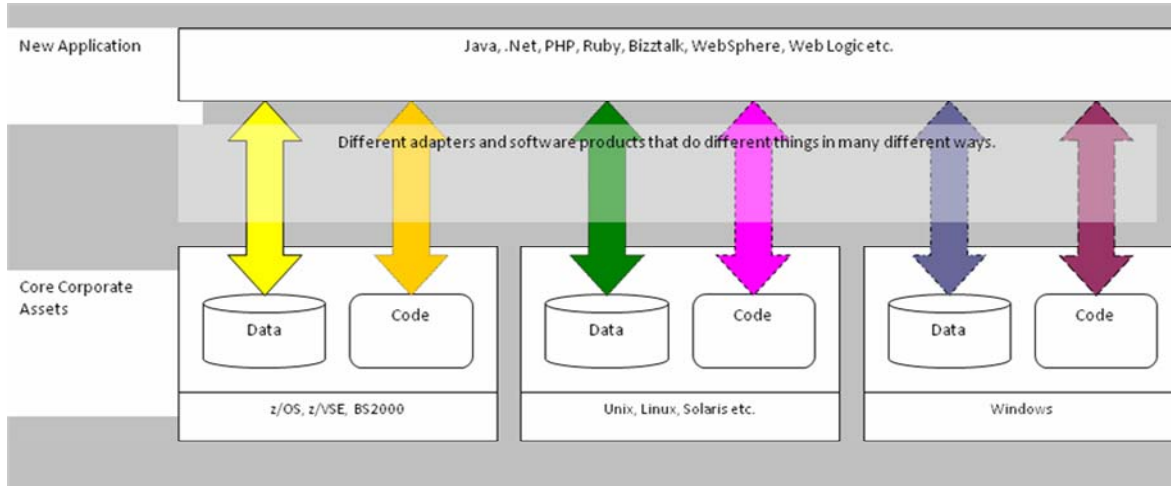
Very few projects are approved and implemented without a cost benefit analysis and a view of the return on investment (ROI) from the project. This leaves many projects extremely sensitive to the cost of implementation. There are many new platforms, languages and technologies that allow for the cost effective implementation and execution of such projects, however, getting at existing data and business logic is still the Achilles heel of most projects. Ultimately the challenge is illustrated in the following diagram: how can new applications get to this data and business logic?



This section illustrates how the SOA Gateway can now make it possible to use a fully integrated approach for projects which need limited access to existing data and business logic in a cost effective manner.

2.1. Using the Traditional Approach

It is possible today to use integration infrastructures to get at data and business logic, however, each platform and language generally has its own mechanism to access the core asset. For each different type of database or code, or even different version of these, the mechanism for getting at the data can differ and require different approaches from the application that needs to see the data. We end up with architecture like the following:



This leads to the following issues with this approach:

1. Each different type of access requires installation and configuration initially along with a maintenance cycle to keep the software up to date.
2. The format of the data may be different on each database or in different chunks of business logic. For example, a person's nationality may be marked as 'IR', 'UK' or 'DE' on one database whereas it will be 'Ireland', 'United Kingdom' or 'Germany' or even '1', '2' or '3' on another database. This means that the application developer must have knowledge about the database being accessed and the schema in use on that database.
3. Securing architectures such as these is extremely difficult as it adds to the configuration and maintenance to ensure that only those authorized to access the data or business logic can do so.
4. Each mechanism is different so the application developers need to learn different ways to access different systems. As the number of different databases or languages grows, the problem to access them grows exponentially as different technologies can give different results for what looks like the same question.
5. In this case, it is quite clear that having a single, standard technology to access any database or business logic would alleviate all of these problems when it can be done in a cost effective manner.

There is also another issue in that the traditional integration technologies available today to achieve this are prohibitively expensive for this sort of activity. It is difficult to justify the cost of purchasing a traditional integration suite just to get at small amounts of data or existing business logic written in some language other than the language the new project will use. This sometimes leads to the following approaches to enable such projects to proceed:

Where access to existing data is required, in many cases the data is replicated at regular periods to a database which is easily accessible from the new project. In some cases a, simple copy of the data is made and put into a database accessible from the new project. While cost effective, it can lead to many problems down the road such as the following. This leads to the following issues

with this approach:

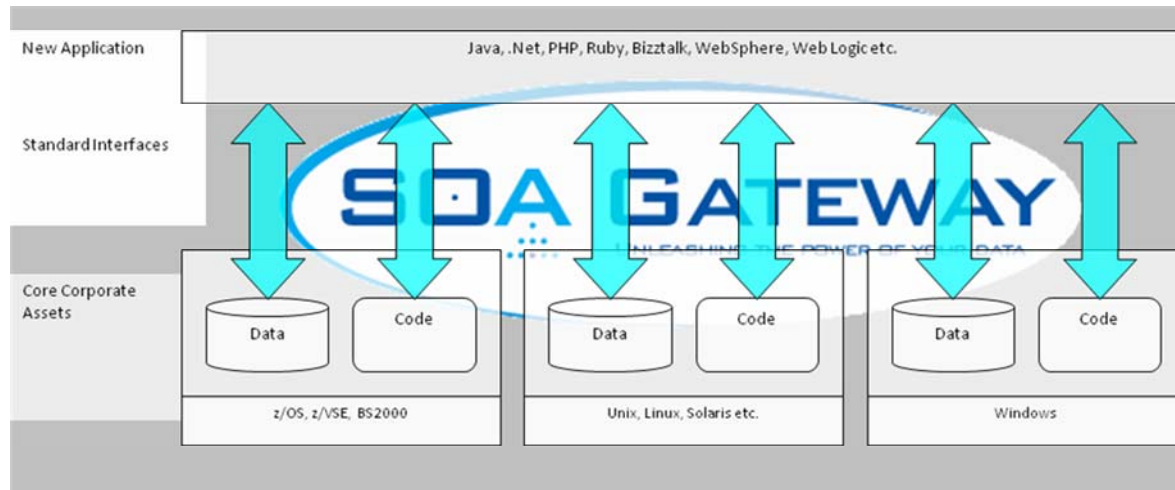
1. Even when data is replicated, unless there is 'real time' replication of updates, the data is out of date as soon as it has been replicated. 'Real time' replication software suites are as expensive if not more so than the integration suites which could not be used in the first place for cost reasons.
2. If data is updated by the new system, it will be necessary to continually sync up the new data with the data in the original system. This type of activity is fraught with danger which can lead to lost data, duplication of data and so on.
3. When the data format of the existing database is changed, it can have a negative impact on the new system which may be totally dependant on the older format.
4. It is clear that the preferred solution would be to get to the data directly if this can be done within the budgetary constraints of the project.

Where existing business logic to implement functionality is available but cannot be reused as the language or location of the code is incompatible with the new system, the logic is generally rewritten in the newer language. This leads to the following problems:

1. If the existing code has a dependency on data in a specific database, this data must also be made available to the new code. This causes issues highlighted in the previous discussion.
2. If the original code is changed, the same change must be implemented in the newly created code so there is more testing required ensuring that both implementations give the same result.
3. It is likely that different skill sets will be required to maintain both copies of the code thus adding to the cost of this solution over the longer term.
4. It is clear that the ultimate solution to this problem would be to access the existing business logic if this can be done within the budgetary constraints of the project.

2.2. Using the SOA Gateway

The SOA Gateway can resolve all of the issues outlined in the previous section. The following illustrates the architecture:



By exposing existing data and business logic using proven industry standards, the SOA Gateway provides a cost effective way to go directly to the existing data and business logic as follows:

1. Once the SOA Gateway is installed, it literally takes minutes to define the services within the SOA Gateway to make your existing data and business logic available as services.
2. Once a service is available, it can be used again and again from any number of systems at no additional cost.
3. Once the service is available, it can be used as a SOAP based Web Service or via a URL (REST) based access which means that the service can be accessed by any language or technology available today such as Excel, Word, InfoPath, Java, vb.net, c#.net and so on.
4. The SOA Gateway requires no software installation on the machine that needs access to the database which saves time in terms of installation and maintenance going forward.
5. The application programmer will use one standard way to get at the data or business logic. Once they have accessed one service, each other service will be identical (except of course for the data) and thus the programmer needs to know nothing about what database or structure has been implemented in the database itself. They will need to know nothing about the language in which a piece of business logic that has been exposed as a service has been written.
6. The SOA Gateway uses standard technologies and can thus use the most up to date security standards available today. This means that the data and business logic can be secured using the Secure Sockets Layer (SSL) technology which is the standard in this area.
7. The SOA Gateway interfaces with existing security systems. This ensures that existing security rules, which have been built up over many years, are still enforced for requests coming through the SOA Gateway. In effect, the SOA Gateway maps an 'internet

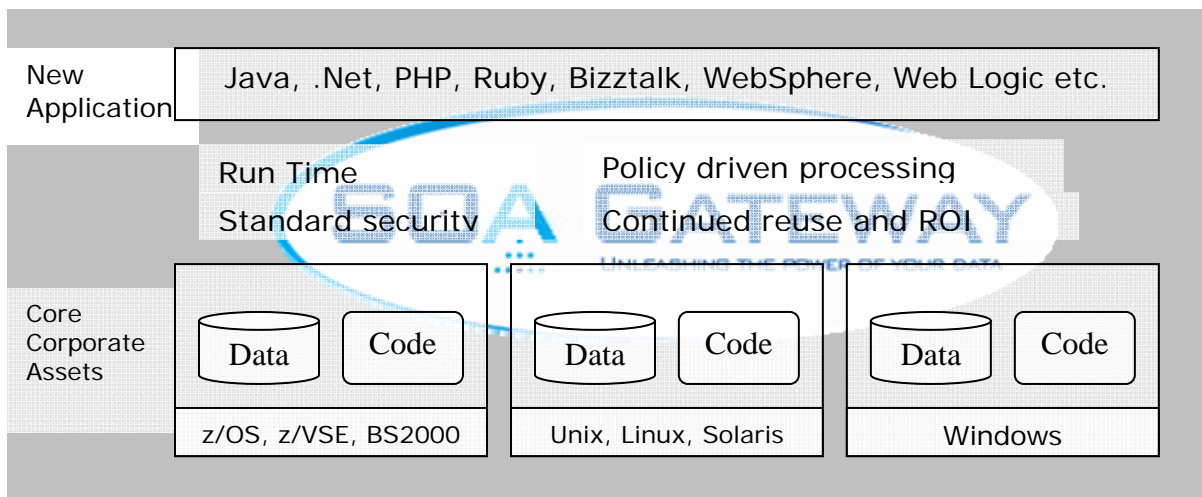
credential' to something the local system where the database or business logic is running can understand

8. The SOA Gateway can map database or language internal types to well defined external types. This means that the data can be normalized so that application developers are dealing with the same type of data no matter what database or language is being accessed. For example, taking the nationality field again, the SOA Gateway can map the internal representation into an agreed external format and ensure that the nationality field from each different core asset uses the same external format no matter how the internal format looks.
9. Once created, a service may then be registered with a UDDI server. Examples of such servers are CentraSite from Software AG, the Systinet registry from HP and the OpenUDDI server available as open source. This means that a central repository of all available services can be maintained electronically and future requirements for these services may be satisfied using that repository.
10. The SOA Gateway license model is based on usage so even though the full power and facilities of the technology are available, you only pay for what you use. It's possible to pay for units of 5 services per year. For example for a cost of 400 Euro per year, it would be possible to expose 5 database tables, 5 applications programs or a mixture of both as services.

2.3. The Benefits to this and future projects

It is clear that projects requiring limited access to existing data and business logic benefit through the simple, quickly and cost effective reuse of existing assets. This can occur directly with no requirement to make copies of data or business logic.

Once completed, the organization can continue to get benefit from the infrastructure as can be seen from the following architecture:



The services created for an initial project can be reused again and again in later projects no matter what language they are being developed in or what technology is being used. The services are future proofed in that they will interface with any technology on the horizon at the moment.

As the infrastructure is 100% based on standards, your installation will also benefit from run time governance, policy driven processing and standards based security out of the box. As new standards or best practices emerge, these will be added to the SOA Gateway without any changes to your applications.

In addition, the license model ensures that licensed customers are entitled to all of the new developments and improvements that are made in later versions of the SOA Gateway. This means that the full power of the technology is available if you continue to use it with a small number of services or if it is used on an extensive basis in your projects.

3. Implementation and Using the SOA Gateway

The SOA Gateway has been designed to be as simple as possible to license, install and use in your projects. The following steps can generally be completed in a half a day or less at which point, it is possible to continue creating services from your existing core assets in minutes. In general, the following time is required to start working with the SOA Gateway:

- Registration and download: 30 minutes (depending on the speed of your connection to the Internet)
- Installation of the SOA Gateway Control Centre and one SOA Gateway Server: 1 hour (depending on target platform and speed of the link to that target platform)
- Creation of services: <1 Minute per service

3.1. Installation of the SOA Gateway

The SOA Gateway consists of two distinct pieces of software.

1. The SOA Gateway Control Centre is an Eclipse plug-in that runs in Eclipse. Eclipse and this plug-in are contained in the package downloaded after you register.
2. The SOA Gateway Server is installed as a stand alone component on the platform where the data or business logic that you wish to expose is running. The server implementation for all available platforms is also included in the package downloaded after you register. The SOA Gateway Server can then be deployed to each target server using the SOA Gateway Control Centre.

The installation of the SOA Gateway requires that you register for a license for the SOA gateway and download the installation materials. There are free licenses available for most uses with some limitations on commercial usage.

3.1.1. Registering to use the SOA Gateway

You must register to use the SOA Gateway [here](#) as illustrated by the following screenshot.

SOA GATEWAY
Understand the power of your data

Home | Overview | Data Access Fast | Re-use Business Logic | Free Trial | Contact Us | About Us | Login

Please fill in the registration form below to get access to the SOA Gateway software and much more.

In moments your free trial will begin. To get started please complete this Quick Registration form:

Registration form: * Mandatory Fields

First Name: * [] Surname: * []

Company: []

Country: [Please Select v]

Mob/Tel: []

Email: * []

Confirm your email: * []

Set Password: * [] - passwords should be 6 characters minimum, preferably alpha & numeric.

Target Operating System: * [Please Choose: v]

I agree to be subscribed to the SOA Gateway mailing list from Risaris.
 Yes

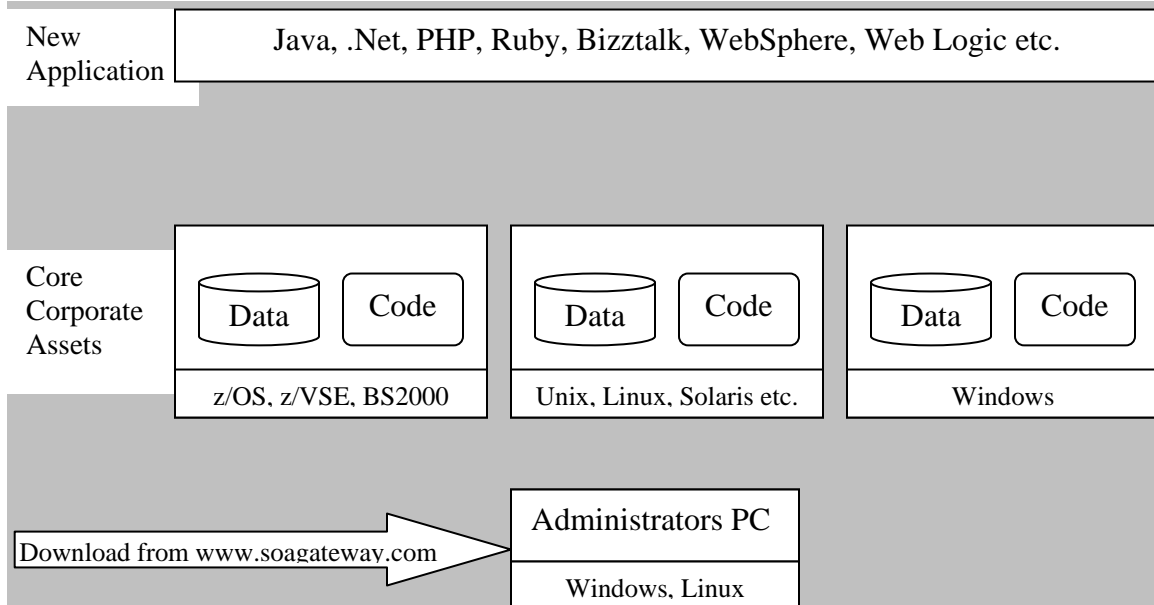
I accept the [licensing terms & conditions](#).
 Yes

[Register]

[Read our privacy policy here >>](#)

Risaris Limited, 6 The Mill Buildings, The Maltings, Bray, Co. Wicklow, Ireland Tel: +353(1)2746840 Fax: +353(1)2745774 info@risaris.com

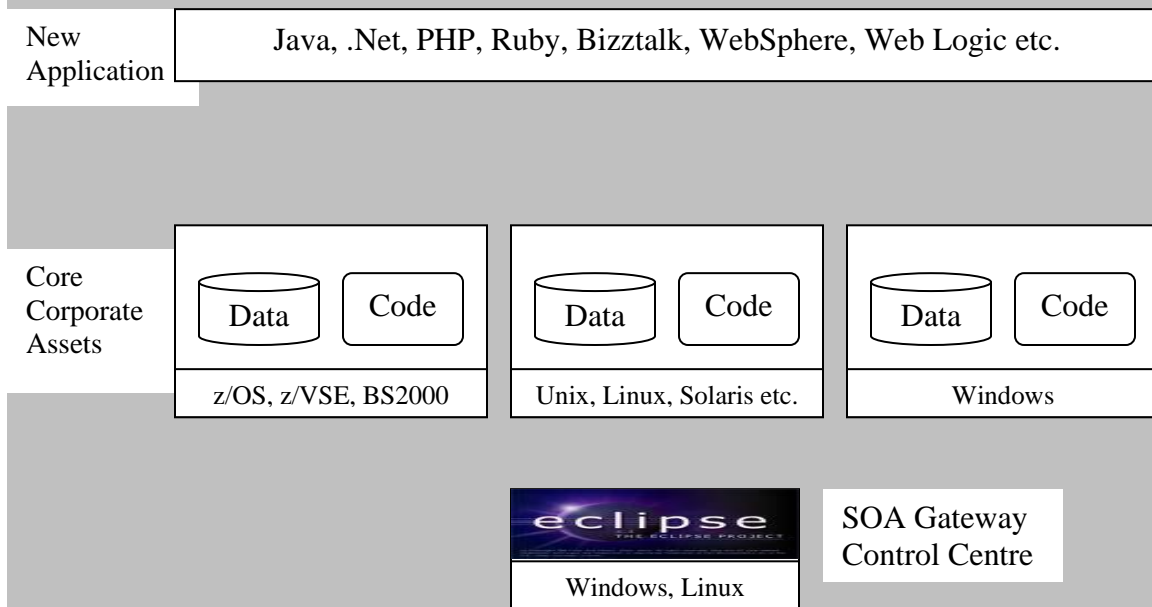
Once you have filled the details in on the above screen and hit the 'Register' button, a confirmation email will be sent to the email account with which you registered. When you have received that email and confirmed the email address, a second email will be sent with a link to continue the process and a license file attached. This link will give you further information about the installation process and will start the download of the SOA Gateway installation materials to your local PC.



This download is approximately 250 Meg and how quickly this downloads will depend on the speed of your link to the Internet.

3.1.2. After the download

Once you have downloaded the package, you are ready to begin installing the SOA Gateway. The next steps are documented in the email sent to you after you have confirmed your registration. Once you have completed those steps, your configuration will look like the following:

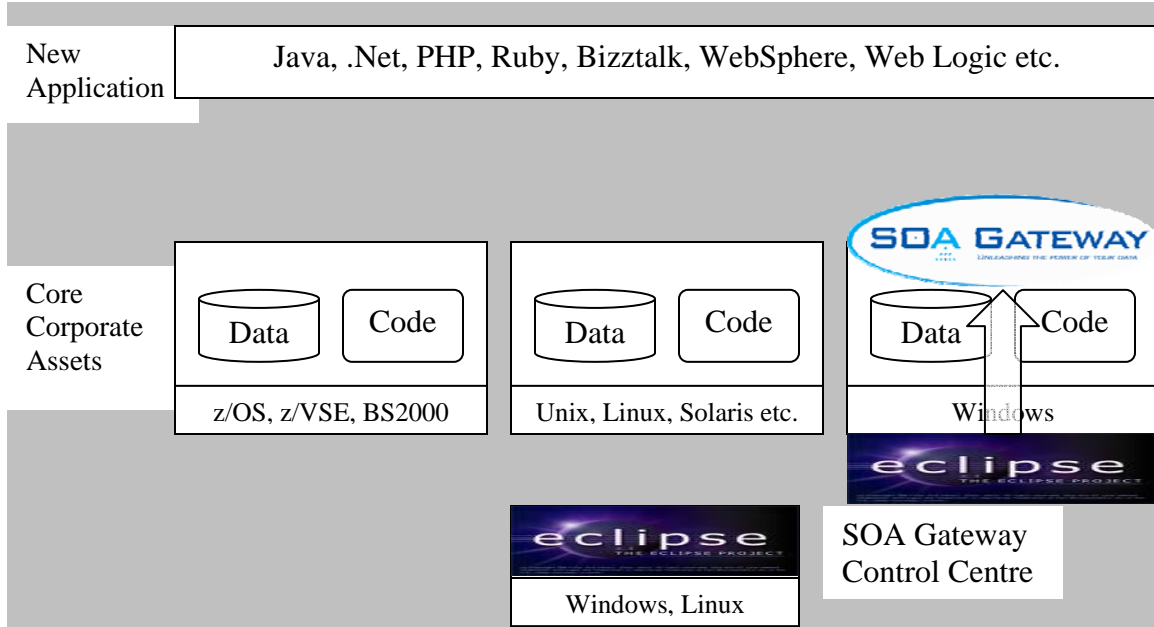


3.1.3. Installing and Configuring the SOA Gateway Server

Once the SOA Gateway Control Centre has been started in Eclipse, you will need to install the SOA Gateway on the target platform. This is a little different depending on your target operating system:

Windows Installation

On Windows, the SOA Gateway Control Centre must be installed on the Windows system where you wish to install the SOA Gateway server. When you select that you wish to install the SOA Gateway server on Windows, the Control Centre will launch the Windows Setup program on the local machine and the SOA Gateway server installation and configuration steps are managed by that setup script.

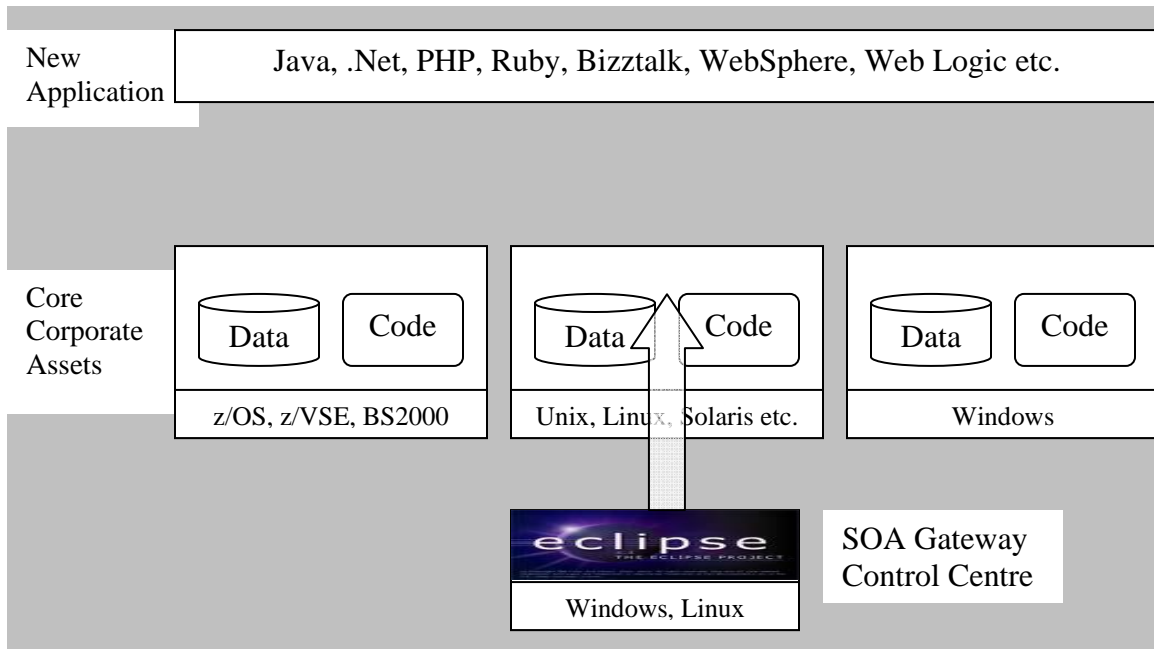


Once the setup script has completed, you will be returned to the Control Centre Deployment Wizard.

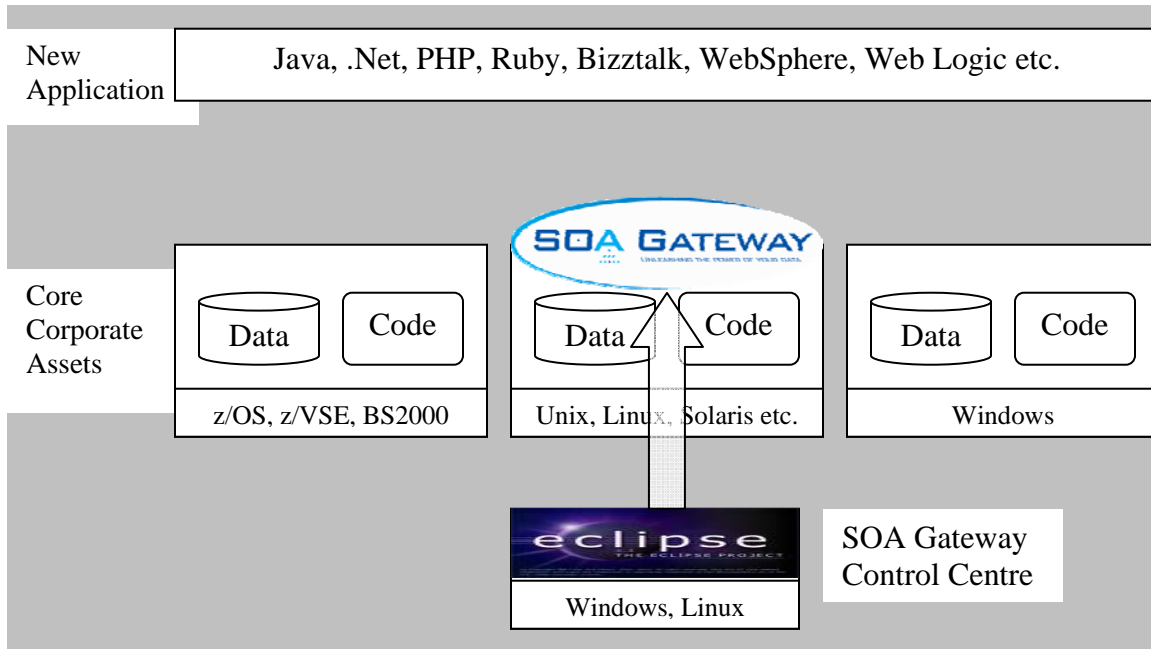
Note that the Eclipse running on the Windows server is only required for installation. This server can subsequently be managed and configured from a remote administrators PC in the same way as other platforms.

Other Platform Installation

On all other platforms, the SOA Gateway Control Centre will FTP the required installation materials to the target system.



Once the FTP has been completed, you must logon to the target system to run a short script or a number of jobs to complete the installation process. These are documented in the installation documentation for the platform where you are installing the SOA Gateway Server. Once this has been completed, return to the control centre to complete the process.



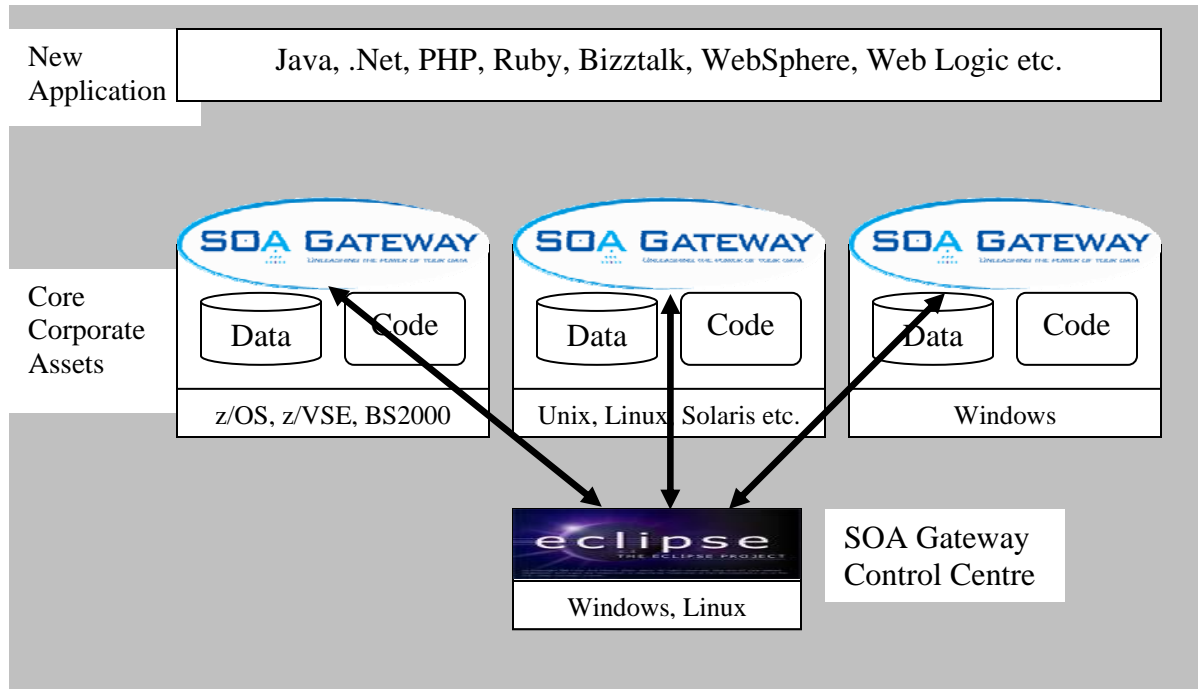
Common Configuration Steps

Once the SOA Gateway Server is running, it is possible to test if the Control Centre can communicate with it from the current screen in the deployment wizard. Once this communication is ok, you just need to hit the configure button. This will install and configure each of the licensed SOA Gateway data source drivers in your server environment. Where additional installation specific information is required, this will be requested during this process. Please refer to the documentation for more details on what may be required for each of the data source drivers.

Once the configuration step has been completed, you are ready to start creating services.

Supporting Multiple Platforms and Operating Systems

It is possible to deploy and install the SOA Gateway server on multiple machines where access is required while monitoring these from the one administrators PC as per the following architecture:



3.2. Creating the Services

The creation of services is achieved from the SOA Gateway Control Centre and is a 3 step process.

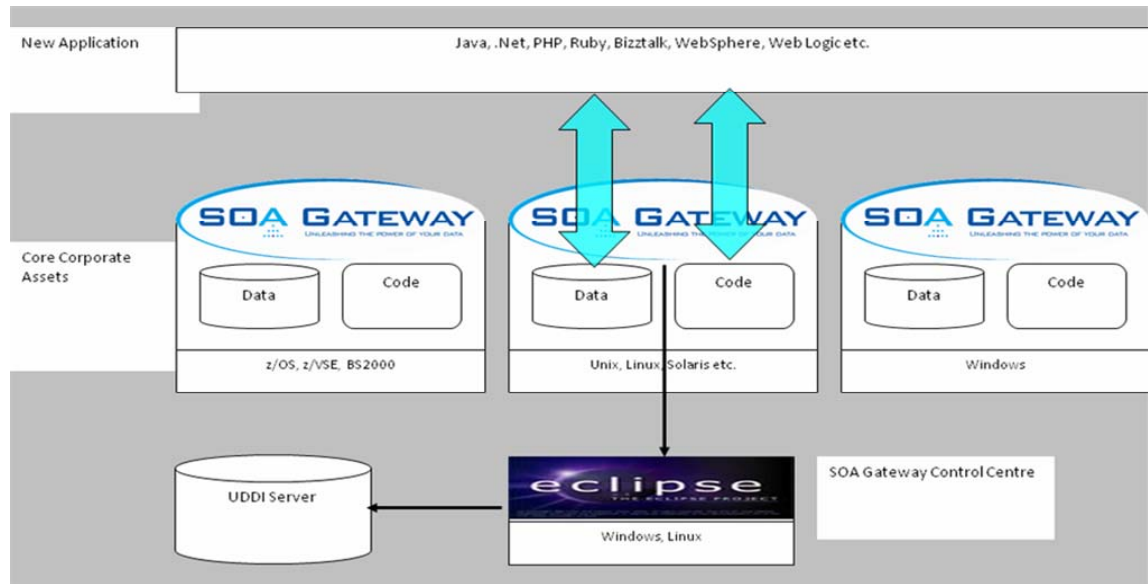
1. The SOA Gateway can discover what resources are available for a specific data source for which services can be created.
2. The Meta data for those resources is identified and used to create what is required by the SOA Gateway.
3. The service definition is deployed to the SOA Gateway Server and is ready for use.

Risaris Limited also recommend an additional step which is the registration of the WSDL in a UDDI server such as CentraSite from Software AG, the Systinet registry from HP or the OpenUDDI server available as open source, however, this is not required to use the service.

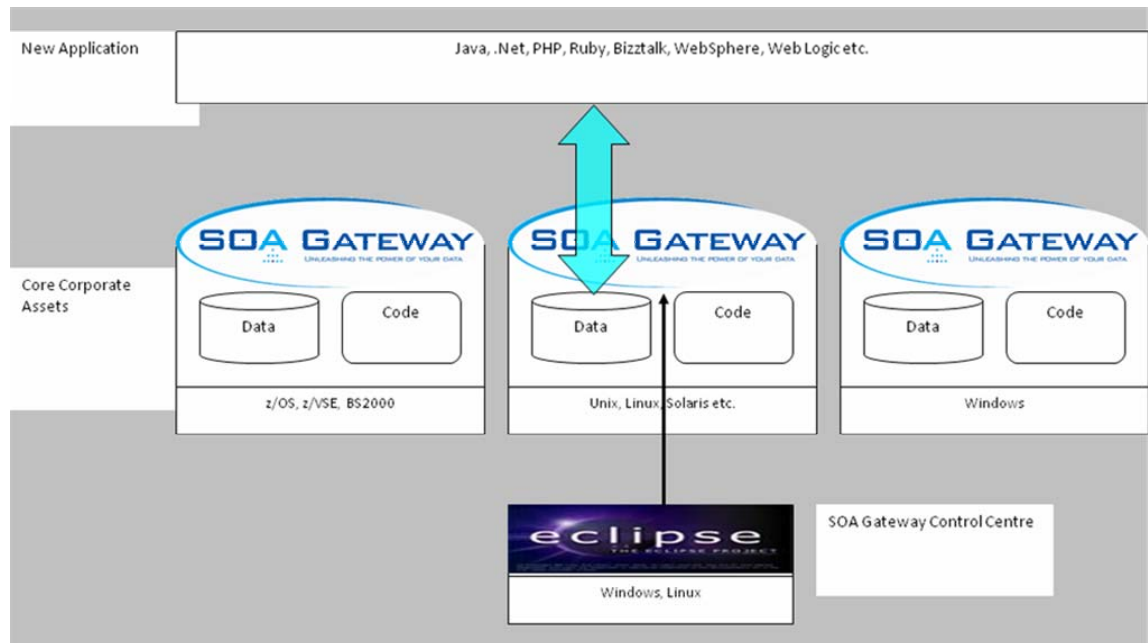
3.2.1. Creating Database Services

Database services are created by simply following these steps:

1. Identify the database from which you wish to create services.
2. Provide the database name.
3. The SOA Gateway Server will return a list of tables or files available on that database as per the following:



4. Select the tables for which you wish services to be created and hit the continue button.
5. The wizard will create services for each of the tables you selected and deploy these services to the SOA Gateway Server. These services are now available for use.

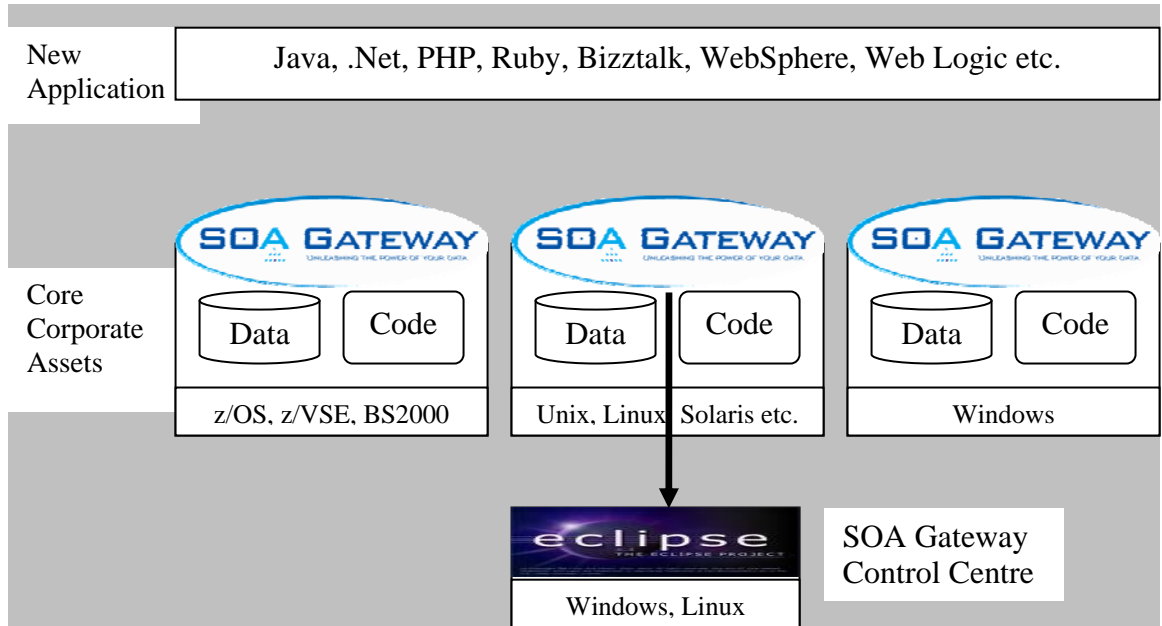


3.2.2. Creating Business Logic Services

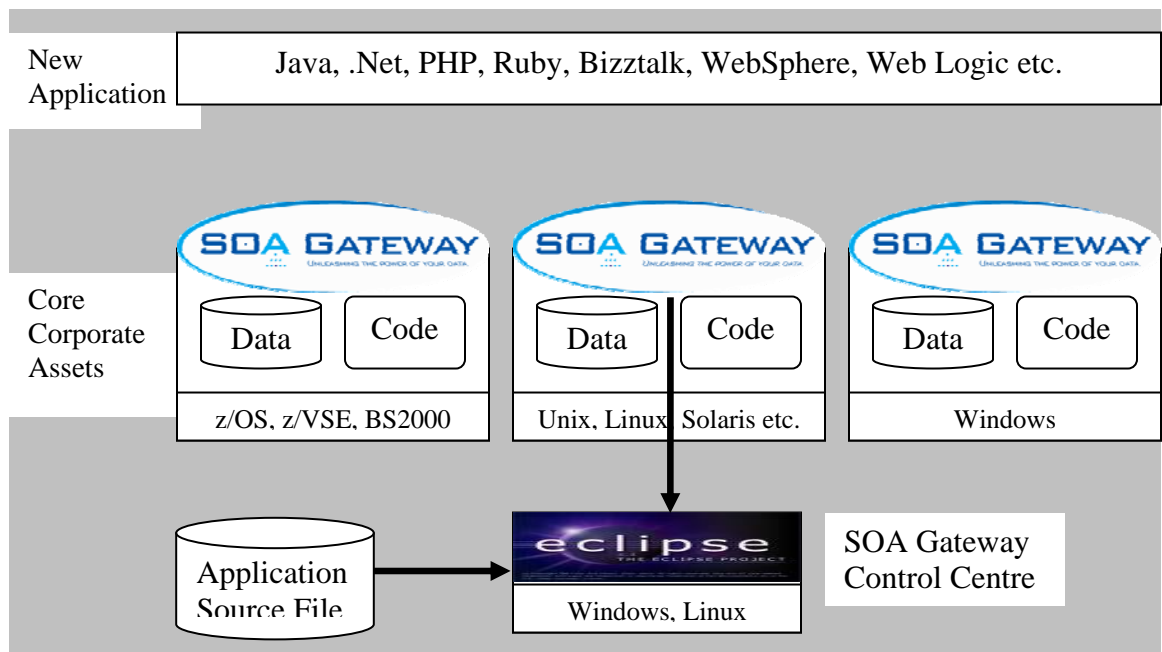
Business logic services are created by simply following these steps:

1. Identify where the application which provides the business logic is implemented (e.g. Natural program, CICS application, Windows DLL etc.)

2. Provide the location of the application.
3. The SOA Gateway Server will return a list of applications available at that location.



4. Select the applications for which you wish services to be created and hit the continue button.
5. Provide the source file for each service to be created.

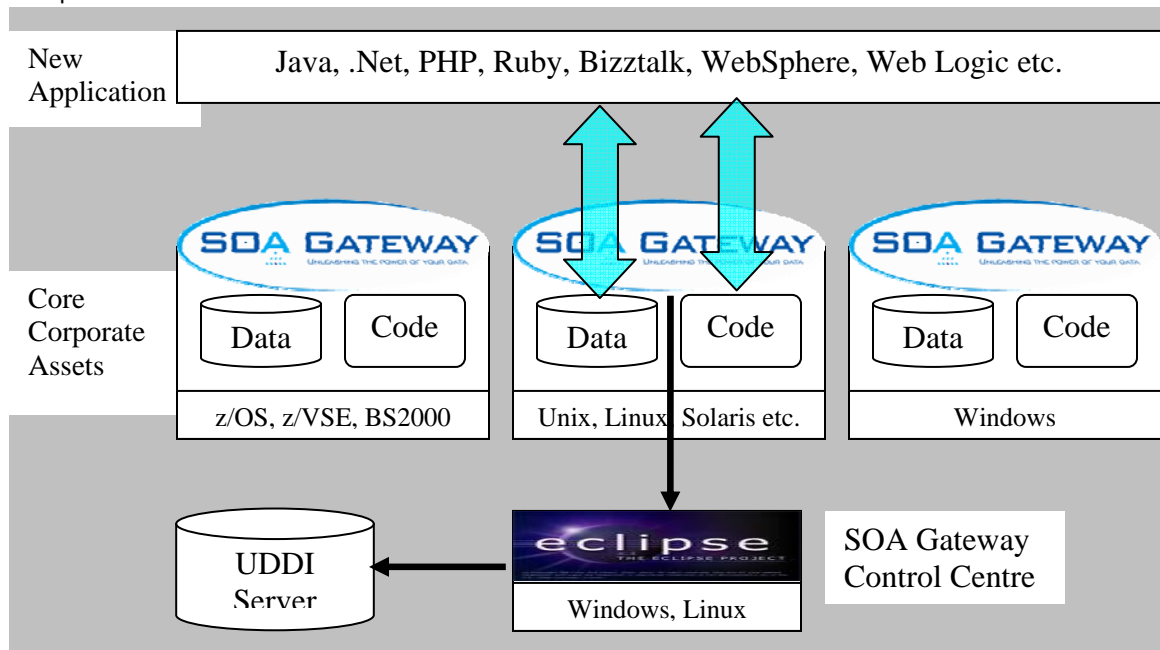


6. The wizard will create services for each of the applications you selected and deploy them to the SOA Gateway Server.
7. Risaris recommend that you review and modify the input only, output only and input/output fields for each service based on your knowledge of the application requirements.
8. These services are now available.

3.2.3. Registering Services in a UDDI Server

The services you have been created can be registered with a UDDI server as follows:

1. Define your UDDI server to the SOA Gateway Control Centre (must only be done once).
2. Select the SOA Gateway services to be registered.
3. Provide the information required by the control centre Wizard until the registration is complete.



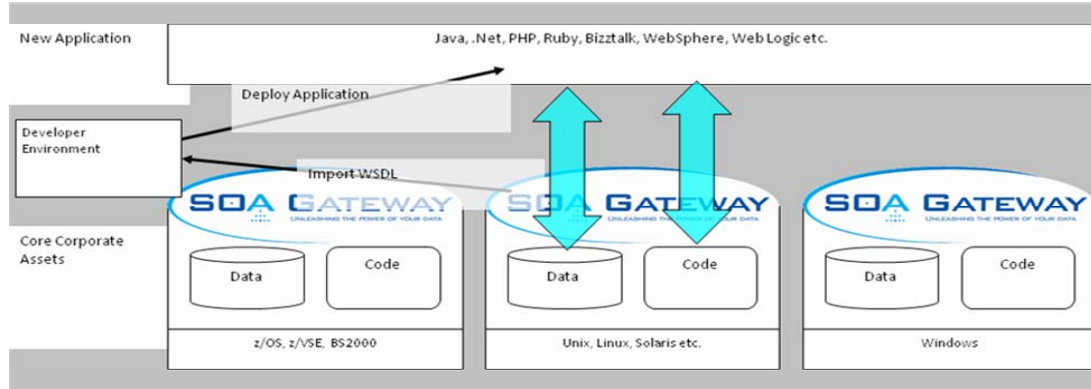
4. This must be repeated for each service you wish to register.

3.3. Using the Services in your project

How the services are used will depend on whether you have used a UDDI server or not.

3.3.1. Using the Services Directly (without a UDDI Server)

1. The application developer is provided with a WSDL location.
2. The application developer will import the WSDL from the SOA Gateway server into their project.



3. The service is now available for use from the project.

3.3.2. Using the Services from a UDDI Server)

1. The application developer is provided with the location of the UDDI server.
2. The application developer selects from the UDDI Server the service that they wish to use.
3. The UDDI server provides the location of the WSDL.
4. The application developer will import the WSDL from the SOA Gateway server into their project.
5. The service is now available for use from the project.

