

# Data Application Access: The SOA Gateway - A Faster Way to Create Business Data Views



## Summary

Many databases don't give users the view of information that they want. They simply are not structured that way.

However the [SOA Gateway](#) enables a tailored business view of data and business logic, without additional client side software, or code development.

From an intuitive GUI, users can create custom data views accessible using standard SOAP, or REST services that are both easily-defined and re-usable.

These services can also be created to enable CRUD access, while preserving and strengthening data integrity and security.



### About the Author

John Power is a driving force behind the development of the [SOA Gateway](#) and Managing Director of Risaris Limited. With over 25 years experience in the software industry, John has delivered complex integration projects in Software AG, Delta Airlines, Boston University and Morgan Stanley.

## Contents

1.	Introduction	4
2.	A Simple Example	4
3.	The Traditional Approach	5
4.	The SOA Gateway Approach	6
5.	Creating a Business Data View Service	7
6.	Using the Business Data View Service	8
7.	Working with Business Data View Object	9
8.	Business Data Views & Multiple Databases	10
9.	Business Data Views Including Business Logic	11
10.	Conclusions	12
11.	About The SOA Gateway	13

V.1.122/01/09

## 1. Introduction

Businesses have enormous amounts of data and employ a variety of interfaces and applications to access, or expose it.

Most, however, don't have one; application, database table, or file that presents a complete business data view. That is because the data model for most IT projects is generally driven by technical issues, such as; efficiency, load or legacy hardware and software choices.

For the business user that traditionally posed a problem. Many IT driven data models on multiple different platforms are hard to understand, or cumbersome to use.

## 2. A Simple Example

Taking a simple example with one request get owner information and vehicle id (called OID); two tables, or files, containing information about the a vehicle and its owner. The data model is as follows:

1. The Owner Table, or File
  - a. This contains details about the owners of a vehicle, as follows:
    - i. The owner's unique ID, labelled 'OID' and the primary key for this table/file.
    - ii. The owner's first name, labelled 'Name'
    - iii. The owner's surname, labelled 'Surname'.
2. The Vehicle Table, or File
  - a. This will contain details about the vehicles, as follows:
    - i. The vehicle's registration number, labelled 'Regnum' and the primary key for this table/file.
    - ii. The vehicle's manufacturer labelled 'Make'.
    - iii. The vehicle's model labelled 'Model'.

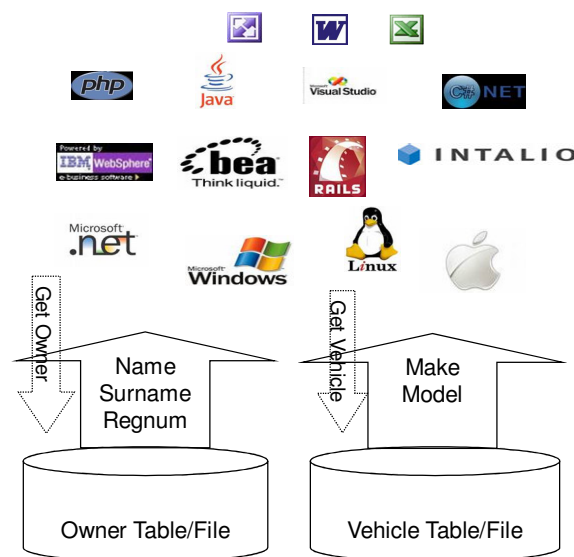
For the business user many IT driven data models are hard to understand...

### 3. The Traditional Approach

In order to access and update Vehicle and / or Owner data, applications must understand that there are two different but associated tables and keep these files consistent. For example, when an owner is added with a registration number, the vehicle file must have an associated entry with that particular registration number.

In the traditional approach access to the data will normally be direct as per the diagram:

Diagram 1: Traditional Implementation



Synchronisation can be an issue as there are multiple databases...

There are a number of issues with this scenario:

1. A 'OID' request is in fact two databases accesses generating extra **network traffic**.
2. Although the vehicle data 'Regnum' is the key used by the system to get the associated entry in the Vehicle's table/file, it **may not be relevant** to the client's OID search and may be hidden.
3. For traditional access, **software must be installed** on the client system to access the back end databases.
4. **Synchronisation** can be an issue where the data is in multiple databases.

5. Ensuring **consistency** across the databases requires a clear data model.

#### 4. The SOA Gateway Approach

The [SOA Gateway](#) tackles the 5 issues (listed on page 4) so often associated with the traditional approach to data access. It does this by creating a business service for the data as follows:

Diagram 2: SOA Gateway Implementation



This configuration offers the following advantages:

No additional software is required client side...

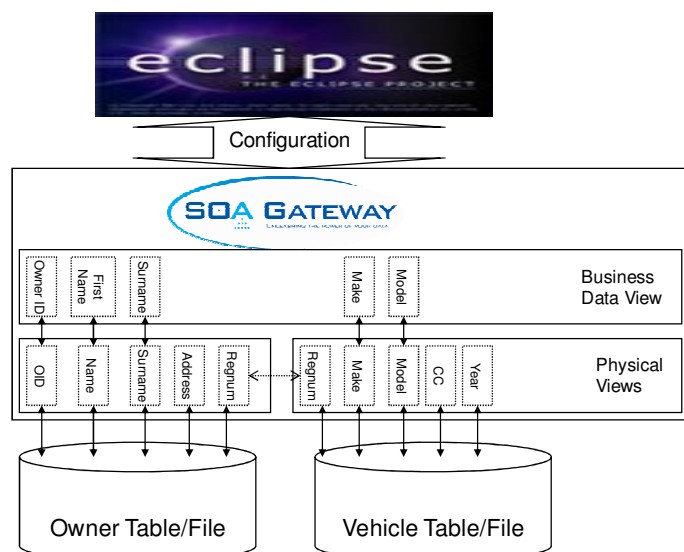
1. **Only one trip** is required to the back-end system to retrieve, or update the composite object.
2. The programmer **does not need to know**, or understand internal links between tables, or files and just deals with the fields and columns of relevance.
3. The SOA Gateway must only be installed on the server where the data is resident – **no additional software** is required client side.
4. The owner of the data defines the model and thus programmers **always deal with the business data view**.

- The SOA Gateway can take care of any **synchronisation** issues, with data from different databases using the transactional capabilities of the back office system.

## 5. Creating a Business Data View Service

Using the [SOA Gateway](#), the Business Data View is built to reflect the business entity and is then linked to the physical data views - all using the intuitive GUI of the Gateway's Eclipse Control Centre.

Diagram 3: Creating a Business Data View



Here are some of the advantages of this approach:

- The users can **select just those fields, or columns required** in the business data view. In the diagram, 'Address' (in the Owner table/file), 'CC' and 'Year' (in the Vehicles table/file) are not used in the Business Data View.
- Fields, or columns (such as registration number in the above example) in the physical view may be **passed and parsed** internally, though not form part of the actual business data view. This is useful for passing internal keys to data which may be meaningless to a user.

A business data view service is created to directly expose a table, file....

3. A Business Data View service is created in the SOA Gateway (based on existing physical services) to directly expose a table, file, or piece of business logic.
4. **Existing services may be re-used** in one, or more Business Data Views.

## 6. Using the Business Data View Service

Once the Business Data View Service has been created, it can be used in precisely the same way as any other [SOA Gateway](#) service. If the service is called 'OwnerVehicle' for example, the WSDL for the SOAP requests may be returned using the following URL:

```
http://<host>:<port>/OwnerVehicle?WSDL
```

This WSDL will offer the standard database CRUD services, namely add, delete, update, get and list, as well as various select methods.

In the same way, to use the REST based interface, the following request will return the Business Data View for the Owner id '12345678':

```
http://<host>:<port>/OwnerVehicle?F=GET&OwnerID=12345678
```

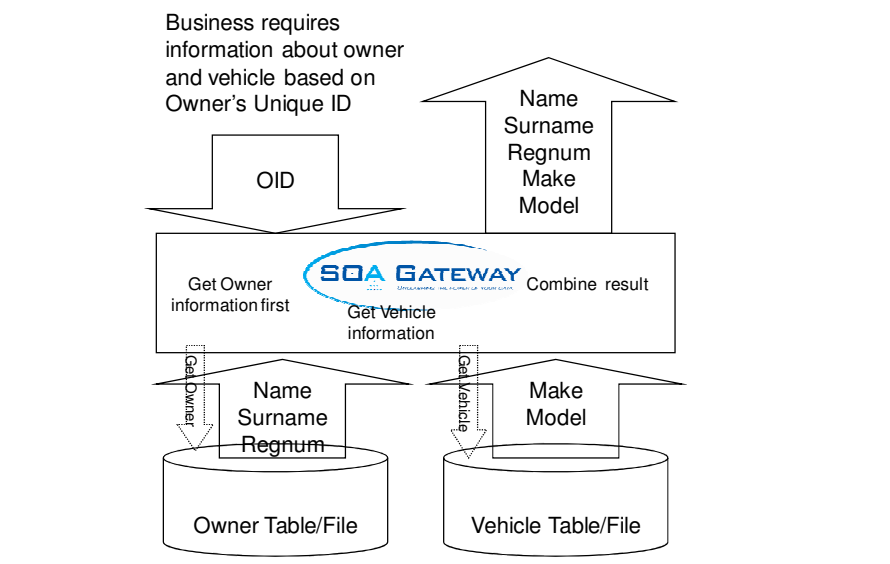
All owners with an ID starting 1234 could be listed using the following URL:

```
http://<host>:<port>/OwnerVehicle?F=LIST&OwnerID=1234*
```

## 7. Working with Business Data View Object

The [SOA Gateway](#) can make any Business Data View object available using the standard REST, or SOAP based access, as in the following example:

Diagram 4 Get a Business Data View Object



Some points to note about the above:

1. The key for the vehicle file is derived from the data on the owner's table/file.
2. The 'Regnum' only appears once in the output but it doesn't have to appear at all.
3. Any data from either table/file may be returned as part of the object.
4. Using the SOA Gateway list functionality, it would be possible to get a list of owners and their vehicles using this service.

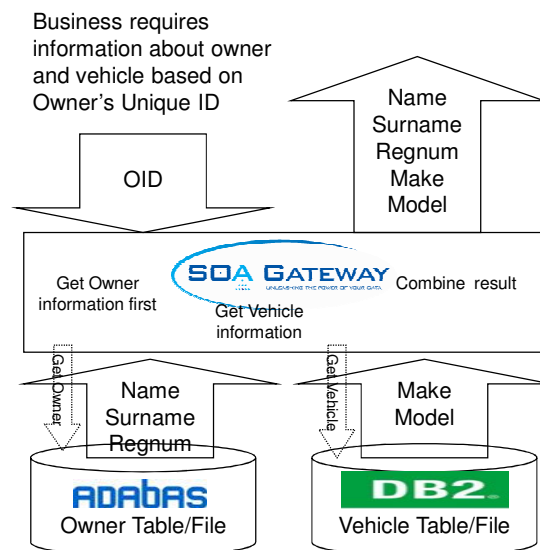
The same straight-forward approach is adopted in respect of; ADDING / UPDATING, or DELETING a business data view object using REST, or SOAP access via the SOA Gateway.

## 8. Business Data Views & Multiple Databases

The SOA Gateway can generate a Business Data View of data residing in multiple database, or even different database technologies.

The following diagram illustrates this for data spread across ADABAS and DB2:

Diagram 5: Get a Business Data View Object



Some points to note about the above:

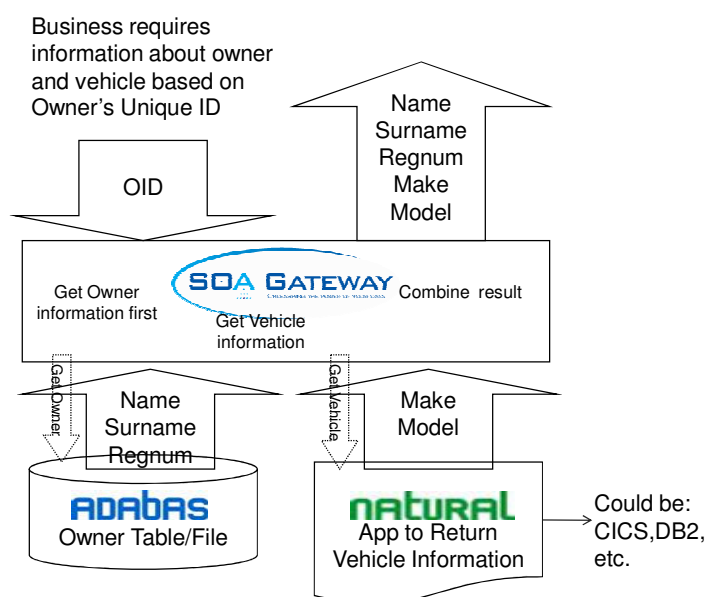
... a business data view of data residing in multiple databases.

1. Additions, updates and deletions employ a transaction manager, such as; RRMS on Z/OS to maintain integrity between the databases.
2. Databases can be replaced and / or added, as appropriate, without the external business logic changing.

## 9. Business Data Views of Business Logic

The SOA Gateway provides a single business view of business logic, as well as data. That means the data can be retrieved from applications, or sub-programs such as a Natural subprogram in the following diagram:

Diagram 6 Business Data View With Business Logic



Some points to note about the above:

1. Additions, updates and deletes are also possible in which case the application will be given an indication of the type of operation when it is called.
2. The business logic may be running in CICS, COBOL, PL1, C etc.
3. Because it is possible to create a Business Data View service based on business logic services, no direct database access is required.

## 10. Conclusions

- Many business systems and databases don't give users the view of information that they want. They simply are not structured that way.
- Furthermore, they often involve data; stored in multiple files / databases, being accessed by complex applications. Accessing data on those systems often generates additional network traffic and requires additional client side software.
- In these cases issues of data integrity, synchronisation and security are to the fore.
- The [SOA Gateway](#) now enables a tailored business view of data, as well as business logic, without programming.
- From an intuitive GUI, users can easily-define SOAP, or REST services to create custom data views from multiple sources, including; different databases and business logic as required.
- Using such web services, the SOA Gateway enables CRUD access, while preserving and strengthening data integrity and security.
- These services can be used again and again with technologies, including; Java, c#.net, vb.net, php, Ruby, Excel, Word, Internet Explorer and so on. No changes required.
- Using the SOA Gateway and messaging systems such as MQ or EntireX, batch updates of your business objects that can be enabled more efficiently than using traditional approaches.
- A higher level of security is possible – being applied at the business view of the data and not just the physical view of the data.

## 11. About The SOA Gateway

The SOA Gateway is a cost effective software tool to:

### **Access data faster...**

It enables access to data from a wide range of database languages (ADABAS, MySQL, DB2, VSAM & Oracle), or applications, without server side code, or expensive middleware.

### **Access business logic easier...**

The SOA Gateway enables easy access and re-use of valuable business logic written in CICS, COBOL, C, NATURAL and many other languages.

The [SOA Gateway](#) is developed by [integration specialists Risaris](#) Limited.

For a free trail of the [SOA Gateway](#), please visit:

[http://www.soagateway.com/html/registration\\_form.php](http://www.soagateway.com/html/registration_form.php)

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.

More whitepapers at:

[http://www.soagateway.com/html/industry\\_papers.html](http://www.soagateway.com/html/industry_papers.html)

See Terms & conditions at [www.soagateway.com](http://www.soagateway.com) © Risaris Limited 2009.

