



---

## **CONTROLLING YOUR DESTINY: ADVANCED DSP INTEGRATION FOR PROVIDERS OF F&I PRODUCTS**

An Online F&I White Paper

---

### **TABLE OF CONTENTS**

Executive Summary .....	2
Advanced DSP Integration .....	2
Service Portfolio.....	3
Service Oriented Architecture .....	6
Concepts .....	7

**THE AUTOMOTIVE B2B MARKET IS USING THE INTERNET FOR MANY OF ITS ACTIVITIES. AT ITS HEART IS THE STRONG INTEGRATION OF DIFFERENT APPLICATIONS. SERVERS, NETWORKS, AND SOFTWARE SHOULD PROVIDE THE INFRASTRUCTURE TO INTEGRATE WEB-BASED APPLICATIONS WITH MAINFRAME AND LEGACY SYSTEMS.**

– DANIEL MENASCÉ, GEORGE MASON UNIVERSITY<sup>1</sup>

---

## EXECUTIVE SUMMARY

Providers today have a new opportunity to deliver F&I products, in real-time, at the dealership. Electronic rating has largely replaced the paper rate guide, and many providers also process contracts and remittance electronically. Menu systems like MenuVantage can price and originate products of all classes, from vehicle service contracts (VSC) to GAP and Credit Life.

Product providers are following the path blazed by automotive finance sources, with electronic credit approval. Today, virtually all lenders are represented in dealerships by at least one “credit aggregation” system, such as Route One. Savvy lenders are on all of them. Similarly, product providers will want to maximize their sales opportunity by integrating with multiple DSPs.

With this new opportunity, however, there are challenges. Providers surveyed by Virag Consulting™ express concern over forms management, IT expenditures, and dealer adoption. This paper will show how service-oriented architecture (SOA) can be used to hold down IT costs, and provide maximum opportunity for integration.

A service-oriented approach, whether limited to DSP integration or promoted throughout the enterprise, saves IT costs by reducing rework and increasing flexibility. The

cost for a single DSP integration may be higher, but it won't have to be repeated.

Industry expert Mark Virag has worked with several leading product providers. This paper presents his vision of a “web service portfolio” specifically for DSP integration, plus a SOA blueprint for F&I providers.

## ADVANCED DSP INTEGRATION

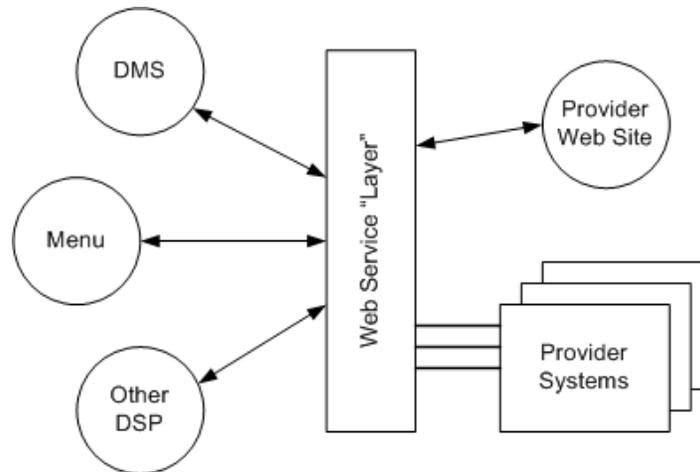
How do you feel about your DSP integration? Most providers we surveyed have at least one – with a DMS, a menu system, or even a credit system. If you are like most, you want your products to be on every DSP platform – but without the programming costs and the forms headache. Most of all, you want to control the process.

We have seen this more than once, and the best approach is to implement your own service-oriented architecture (SOA). If you are talking to a DSP about integration, they should be able to provide a technical specification, and the specification should include web services. A web service is simply a program you support, on the Internet, which allows the DSP to access data – like rates – from your systems.

---

<sup>1</sup> Daniel Menascé and Virgilio Almeida, *Scaling for E-Business* (Prentice Hall, 2000)

FIGURE 1. THE WEB-SERVICE LAYER IS A BUFFER BETWEEN PROVIDER SYSTEMS AND THE DSP



The problem with one-off integration to a DSP is that, sooner or later, you will want to add a second DSP. Adopting a uniform, standards-based approach will reduce your ongoing costs for development, maintenance and support. Here's why:

- You develop the web service once for all DSP integration partners.
- You can use the same web service to drive your own web site.
- You can change the systems behind the service without affecting the DSP.

These are huge benefits, in terms of flexibility and cost saving. We worked with one provider who was able to replace his admin system, without affecting his MenuVantage interface. Another was able to reuse his contracting service to support a data migration. The web services fit between your systems and the DSP, as shown in Figure 1, above.

Any problem with integration among the DSPs stays on their side of the interface – along with other issues, like pending or incomplete contracts, and electronic signature. The web service layer acts as a central switchboard for all external systems and – in a full SOA model – internal ones, too.

Of course, the handful of services that support DSP integration does not constitute a full SOA<sup>2</sup> – but it's a good start. We'll refer to the SOA textbook later, but first let's identify the "portfolio" of web services you will need.

## SERVICE PORTFOLIO

The sample product for this paper is a vehicle service contract, but any product will do. GAP often requires a rate table, as does Credit Life. Services to expose are:

- Rating
- Contracting
- Forms
- Quoting
- Claims

The basic idea is that your key business functions should be accessible via web service. The more that the DSPs are able to serve themselves, the more volume you can handle – with less overhead.

Technically, these functions can be delivered on a single web service but, according to the SOA Reference Model<sup>3</sup>, it's best to keep them separate.

<sup>2</sup> Darmawan, et al., *Power Systems and SOA Synergy* (IBM Red Books, 2008)

<sup>3</sup> Bieberstein, et al., *Executing SOA, A Practical Guide* (IBM Press, 2008)

---

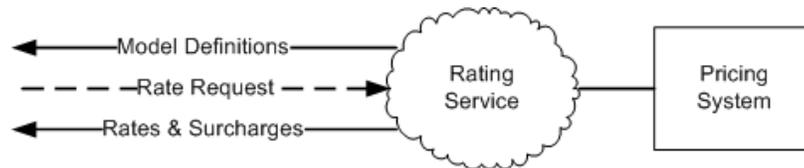
## SERVICE DEFINITIONS

This section gives a brief description of each service, along with a flow diagram and some implementation tips. Control flows are shown using dotted lines, and data flows using solid ones. For each diagram, imagine your DSP partner – say, a menu system – on the left.

---

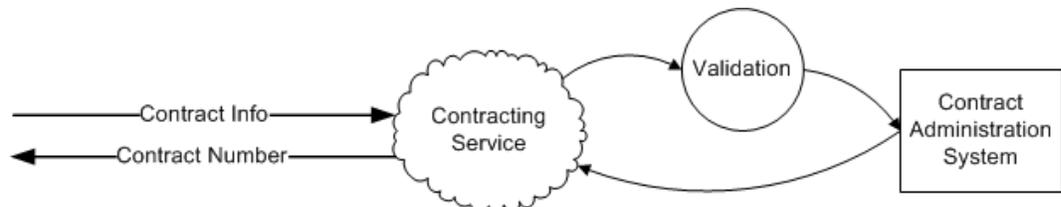
### RATING SERVICE

The DSP requests rates for a given vehicle, and the service transmits a customized “electronic rate card.” If you’d like to rate by straight models as well as VINs, the service should also transmit your model table.



### CONTRACTING SERVICE

The DSP sends complete information for a contract. This service validates the information, boards the contract, and returns a contract number. This indicates to the DSP that the contract was accepted. Otherwise, the service returns an error code. Note that this technique avoids assigning blocks of contract numbers to each DSP.

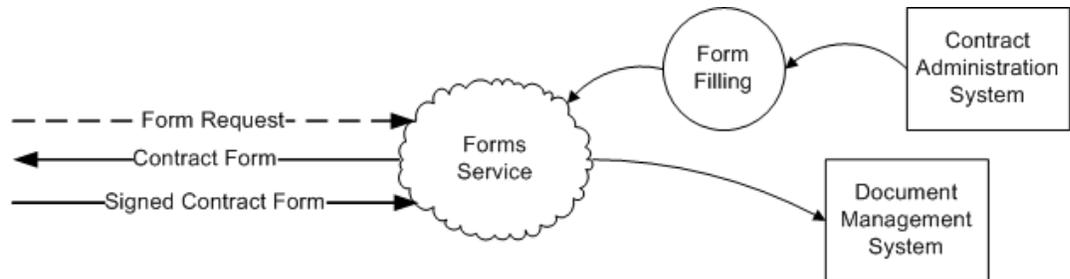


You may want to support a validate-only method, and you may also want to support dummy contracts (see *Quoting Service*, below). Providers vary widely on this point. We favor an “all or nothing” approach that puts the onus on the DSP to deliver a good contract.

## FORMS SERVICE

We recommend separating the Forms service from the Contracting service. Here, the DSP requests a contract – which is already boarded – and the service generates the contract in Adobe™ PDF format. The DSP adds an electronic signature and returns an updated PDF. The DSP is responsible for the digital seal on this document.

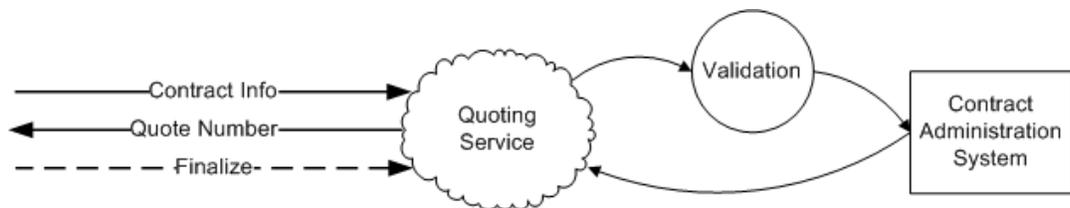
YOU MAY WISH TO PROVIDE A VOID FUNCTION, IN CASE THE DEAL UNWINDS.



The Forms service can dynamically generate a new contract at any time but, for compliance purposes, always transmit the copy from your Document Management System.

## OPTIONAL: QUOTING SERVICE

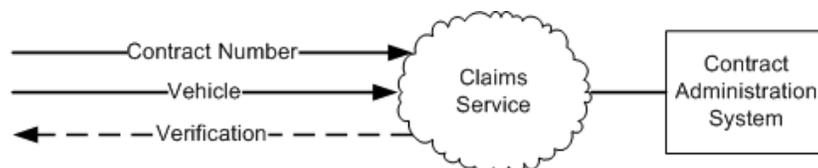
This allows the contract form to be used as a selling tool by creating a sample contract. Your system will need a way to identify the contract as a sample, and then a way to promote it when it becomes active. To avoid confusion in the dealership, make sure these contracts are clearly identified as samples.



Another optional service, not uniformly implemented, is one that allows the DSP to void or cancel the contract after it has been accepted.

## CLAIMS SERVICE

Shown here is a simple method for coverage verification. The Claims service is potentially much more complex, if you wish to support a DSP-driven authorization process.



# SERVICE ORIENTED ARCHITECTURE

As a practical matter, the first service you want to develop is one that integrates with today's DSP. This means deciding on the scope of services, and certain business rules – like those mentioned above. Before starting to design the services, consider these questions:

- Who are the current and future integration partners?
- What are the business goals?
- What metrics will define a successful integration?
- Which business processes should be exposed?
- How many internal systems are involved?

A good design begins with business goals, and then adds interaction patterns, data representation, data semantics, access methods, and operation semantics.<sup>4</sup>

So far, we have talked about individual web services to meet specific interface needs. Beyond that, the goal is an

architecture in which all business processes are supported by standardized components. You can choose to expose any business process, through a DSP or some other system – even a business-to-consumer (B2C) web site.

The SOA Reference Model defines a layered systems architecture. Business processes float on top of the web services, so that you can evolve the business without disturbing the lower layers. The web services, which we have been discussing, rest on software programs called components.

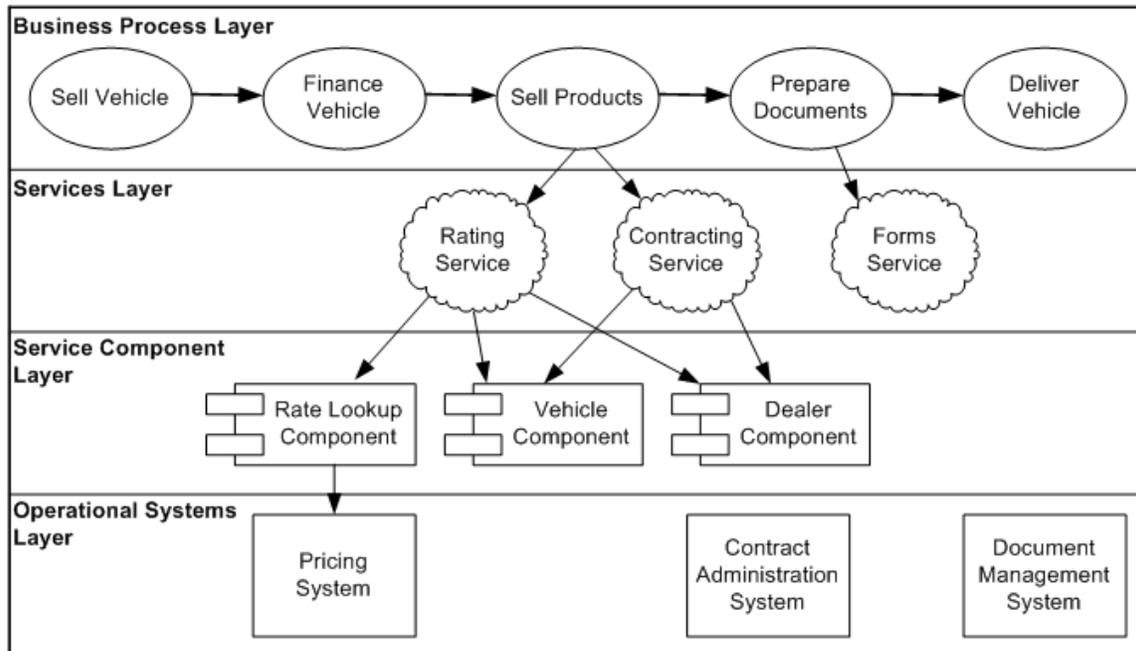
Figure 2, below, is a partial Reference Model customized for the F&I products business. It illustrates how the services in the previous section can be organized to fit your systems.

---

<sup>4</sup> Paul C. Brown, *Implementing SOA: Total Architecture in Practice* (Addison-Wesley, 2008)

---

FIGURE 2. A SAMPLE REFERENCE MODEL FOR THE F&I PRODUCTS BUSINESS



## CONCEPTS

Here is a short list of terms and concepts you will encounter in the SOA world:

- **Consumer** – a web service is said to be “consumed” by the program that binds to it. That is, your integration partner, the DSP.
- **Exposed** – you will be “exposing” your business processes in the form of these web services. In some cases, you may also consume a service of the DSP.
- **Message Syntax** – any agreed-upon coding scheme for the data to be sent by the web service.
- **XML** – Extensible Markup Language. This is a versatile coding scheme for any kind of message. For instance, we can encode a rate card as XML.<sup>5</sup>
- **WSDL** – Web Service Definition Language. This is a computer-readable description of your web service.
- **Object-Oriented** – this is a modern programming practice that supports encapsulation. It is used in the component layer of the SOA model.
- **Encapsulation** – this is the concept that you can convert one of the operational systems without disturbing the service layer.

## SOA DESIGN METHODOLOGY

If you choose to design an enterprise SOA model, you will want to use the SOMA methodology from IBM. At a minimum, you will need these items:

- **Service Portfolio** – A rough outline is given above in section one.
- **Service Hierarchy** – Organization among the services.
- **Service Exposure** – As above, which services to expose and why.
- **Service Dependencies** – Which services depend on which others.
- **Service Composition** – How the services are organized to support the business processes.
- **State Management** – The various states that each service should implement.

This list is straight from the manual, and it’s just for the service layer. The service layer depends on an object model that encapsulates the functions of your legacy systems. If you don’t already have an object-oriented design capability, you will want to acquire one.

---

<sup>5</sup> Mark Virag, *Provider Interface Specification* (MenuVantage, 2005)

---

---

Virag Consulting can help you design and implement web services like the ones presented here, customized to suit your systems and your products. Call today to discuss your web service needs, or mail to [sales@viragconsulting.com](mailto:sales@viragconsulting.com).