Dajeil

Web Services Infrastructure

Whitepaper: The SOA Appliance

The required steps to deliver an ISV's SOA application as an appliance

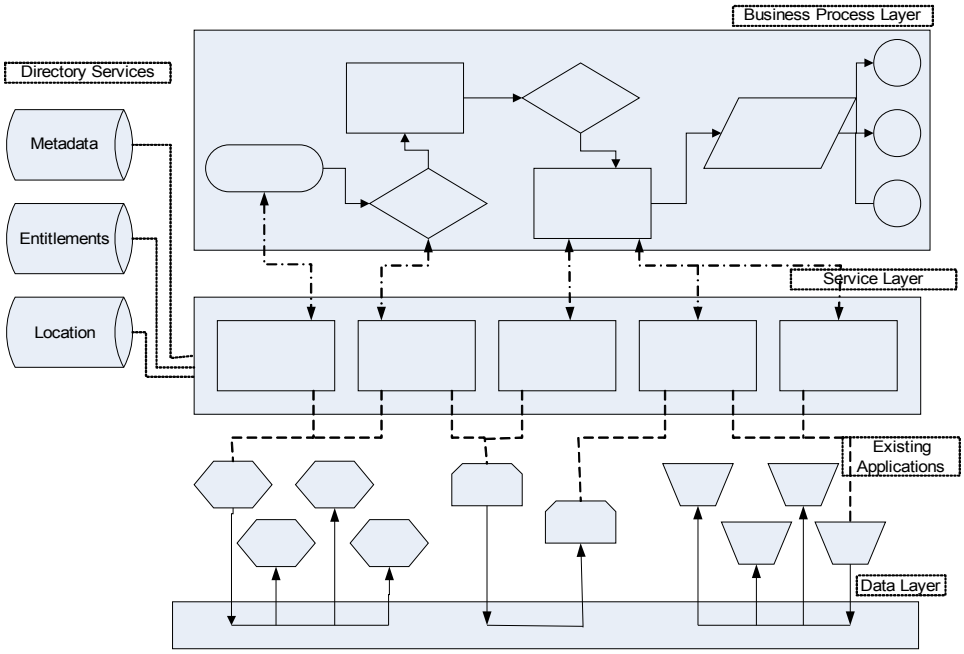
1 The Business Case

1.1 Introduction

Enterprise system professionals have realized that service oriented architectures (SOA) can dramatically increase the complexity of building and maintaining corporate business processes. To achieve maximum benefit from SOAs, it's important to carefully consider architecture and design principles before picking technology and working on practical examples.

In a well-designed system, SOA appliances can greatly reduce cost while simplifying implementation and ongoing maintenance. Appliances can also help insulate companies from the inevitable changes common to any software system by minimizing support activities.

SOA systems can be implemented using many different tools and technologies. The diagram below illustrates the different layers of a typical SOA from higher level business processes to raw data exchange [Ref Chandar Pattabhiram].



The chosen solution must address issues such as scalability, security, availability, and service level agreements. The creation of a service layer populated by SOA appliances which absorb the workload of business processes enablement is a convenient solution to address the issues above.

Even the best planned SOA project has its complexities and requires a broad skill set to implement and maintain. The thinking that originally led to SOA was focused on reducing that complexity. Complexity reduction pays dividends each time a change is made anywhere in a SOA, as the number of moving parts affected by the change is directly related to cost, time, and difficulty of implementation.

1.2 Rationale for SOA Appliances

SOA appliances are useful because they're simple. They require minimal support infrastructure and are highly repeatable and secure. An appliance that can be accessed only via the network can, by definition, be configured, operated, and maintained from anywhere with a secure network connection. Appliances are simple because no infrastructure preparation is required beyond electricity and an IP connection.

Software solutions require an infrastructure composed of host hardware, network hardware, an OS, JVM, patches, RDMS, security software, and management software. Since software

makers are constantly "improving" their products, the odds that any set of infrastructure servers are actually identical, however carefully standardized, is almost zero. Appliances, on the other hand, can easily guarantee that form, fit, and function are identical because all the software and firmware in the box is under a single release control program. They are, in effect, one large software, hardware, and firmware distribution.

Because appliances aren't general-purpose, their components can be optimized for the task they perform. Similarly, there's no need to provide user-level access to internal components, such as RDMS, RAID arrays, OS functions, etc. They come preconfigured and can be managed by the appliance itself. All the user needs to do is configure the device for exactly the task it must perform.

As network devices, appliances can be placed wherever they are needed, and easily managed and upgraded. Local IT presence isn't required. Indeed, many appliance users manage worldwide implementations from a single center of excellence. Appliances fit into many places in the SOA: they are the most logical solution for data integration, security, acceleration, mediation, governance and the enterprise service bus.

At the end of the day, the idea with an appliance is that the clients don't care what's inside. They care about the functions of the device and its costs. Typical TCO calculations indicate that a router costs $100 a month, a server $500 and a SOA appliance $225 a month.

For example, a major global CPG maker recently deployed SOA appliances to accelerate its SAP adoption. A legacy middleware infrastructure was replaced worldwide with a ROI payback of fewer months, reduced total cost of ownership of 73% and project delivery in less than 30 days.

1.3 Benefits of SOA Appliance

We have focussed on the benefits of SOA appliances to the end-users however appliances are equally appealing to the independent software vendor (ISV). The benefits to the ISV are:

- Predictable, Secure environment
  The execution environment for the application is predictable and security features not available on a standard operating system can be used to protect from any rogue software.
- Hardware acceleration enhances performance
  Hardware acceleration is available for crypto and XML processing to offload the CPU intensive processing thus improving throughput and latency.
- Application Optimised for Platform
  With a known memory, CPU and disk configuration optiomal selections can me made in respect of no of threads, memory allocation, file sizes,…etc.
- Single HW platform / OS - Minimal Test / Support Overheads
  A single hardware platform greatly reduces time for new releases in QA as they needn’t be tested in multiple hardware and software environments. Support engineers need only be familiar with the appliance platform supported. As appliance platforms are locked down users can not adversely configure system parameters. Application behaviour variances between different operating systems are eliminated. A single point of contact for support issues makes their resolution easier. In summary, firmware updates, new releases and end-user support can be provided in a more timely manner at a lower cost.
- Quick and Easy Installation
  Installation issues are largely power and network connectivity.
- Hardware Logistics outsourced
  For the ISV the option of outsourced hardware logistics is attractive. It can focus on features, functionality and software support and the appliance vendor can take care of warranty and repairs in the field. Box replacement in the case of failures is becoming

more attractive to clients thus easing the burden of maintaining field engineers in each region.

2 The Steps

Once the ISV has confirmed the market demand for an appliance in its product space, the following steps represent the typical delivery of product. These steps can be divided into two phases. A proof-of-concept phase is recommended for the first appliance product of the ISV. The second phase is the completion of the go-to-market developments for the product.

2.1 Proof-of-concept

Step 1 - Specify appliance functionality

The features and performance targets for the appliance should be specified.

Step 2 - Select the software modules for the appliance

Identify the software modules required to delivery the specified functionality.

Step 3 -Characterise the system

The appliance vendor should attain a level of familiarisation with the application in order to identify which parts need modification during porting. This involves performing a standard installation on Linux, configuring, and using the software in a typical configuration.

Step 4 - Port the software

Appliance platforms secure their software by digitally signing modules for execution. There is a unique product key for the signing all firmware releases and upgrades. Porting the software involves moving the modules to a directory structure on the appliance and grouping the modules into logical units for application of digital signatures. Scripts of start, stop and configure the unit must be developed. System and OS parameters are tuned to optimise the performance of the application.

Step 5 - Add User Interface

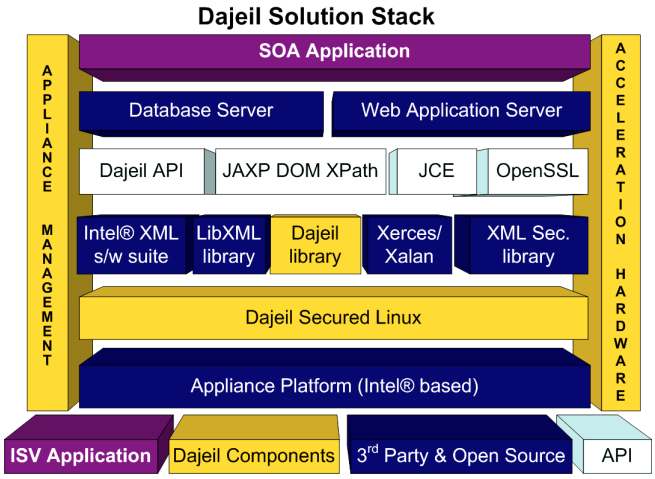
A user interface is required to manage the appliance. This is available in two forms; (1) a command line interface and (2) a web GUI. The basic functionality of these interfaces is required for the proof-of-concept evaluation.

Step 6 - Test Functionality & Performance

Once the development work is complete a suite of QA functional and performance tests are applied to the POC. These tests are executed iteratively as new acceleration functions are added to that the overall system uplift can be measured.

Step 7 - Integration of Acceleration

This activity usually starts with integration of security acceleration. The security libraries are pluggable and provided the application has not hard-coded calls to third-party libraries, the integration process is straightforward. Dajeil's solution stack is illustrated below.



The XML integration is usually more complicated as it is typically used throughout the application. The steps for integration of acceleration are:

- **Profile the Application Processing**
  The application should be profiled to identify the XML and security bottlenecks. This helps identify areas where there is most payback from acceleration.
- **Measure Standalone Acceleration**
  For transforms, schema validation, parsing and xpaths, extract the necessary files and benchmark the performance in a standalone test program. These figures can be used to prepare a good estimate of overall system boost. Such tests will also highlight at an early stage any non-conformances.
- **Identify the Interface Points**
  The interface points to the application should identified to ensure that accelerated libraries are only used when required.
- **Add acceleration**
  Install the selected acceleration libraries and drivers.

Step 9 - Prepare Cost-Benefit Report

After the acceleration has been integrated the performance tests should be run again to measure the actual system performance boost. With this information a cost benefit report can be prepared looking at the different markets for the appliance product.

2.2 Production Appliance

The following activities are involved in the preparation of the go-to-market product: Product name, brochure, branding, pricing, documentation, certifications, packaging, firmware, …etc. delivery to beta customers in advance of full launch is recommended.

3 The Rewards

The rewards associated with an appliance product are easily measurable. Obviously, there is increased sales revenue. The profit margin for an appliance is larger than that of software only solutions because the development, support, and test overheads are lower. Customers are happier with the single point of contact for any issues related to the appliance functions. Market demand for an appliance offering is satisfied by the introduction of the new product. The ISV's lost sales to the market leader (offering a range is appliance products) are stemmed. The effort associated with the development of the first appliance product is far less for any follow on appliance products. In conclusion, the rewards are many and easily measured.