

Leveraging an On-Demand Platform for Enterprise Architecture...Preparing for the Change

David S. Linthicum

david@linthicumgroup.com

The notion of enterprise architecture is changing quickly. What was once a concept of technologies and interconnections existing within the firewall is morphing into something that's more dynamic and valuable to those building enterprise architectures. This technology is known as an **on-demand platform**, a way of doing architecture that will both increase efficiencies and innovation, while at the same time reducing costs.

So what's new? An on-demand platform supports multi-tenancy. In contrast to single-tenant counterparts, multi-tenant platforms share a single, common infrastructure and code base that is centrally maintained. Individual customer deployments are unique, separate, and secure within this shared multi-tenant platform, and run a single code base that is shared by all users and upgraded simultaneously – thus delivering all the benefits of innovative new releases, with none of the painful legacy of upgrades.

The use of an on-demand platform has emerged with the notion of Web 2.0. Indeed, we are seeing a quick shift in the use of Web resources within the enterprise. This is largely driven by the use of service-oriented technologies and approaches that allow existing systems to both consume and manage services that are Internet-delivered, as well as emerging on-demand platforms for hosting and managing enterprise processes, integration, and development services. Moreover, these systems are also able to expose services to other consumers on the Web.

Clearly, the “consumer Web” is influencing the way we do enterprise architecture going forward, changing the rules of engagement for the Global 2000 as they approach both SOA and Enterprise Architecture. Indeed, as we learn to leverage services and applications we either host or maintain, the opportunity exists to take the same value proposition of both economies of scale and best of breed technology, to address enterprise architecture challenges going forward. The ROI is compelling as are the quality of platforms that are available, and the days are clearly numbered for complex software-driven enterprise architectures that are both difficult to maintain and reuse.

The notion of building bridges to service providers and managing the interaction among these providers will become more commonplace in 2007 and 2008 as we learn to accept that many services we leverage within an enterprise are services and development technology we may not own or host. The technology exists today. We need to define and refine our approaches now, including architectures, enabling technology, and use of standards.

Indeed, we are moving toward a day when most of our enterprise applications and enabling architecture may be delivered as services, and thus provide a more economical way to approach information technology management. This is also the great equalizer since businesses, large and small, will have access to the same number and quality of services and platforms, much like they do with Web sites today. Shared services will create many opportunities, including better agility and the ability to operate a business with fewer IT resources. In essence, we're moving to the new Web where service delivery over the Internet will be added to information delivery as the key strategic value of the Web to businesses, as well as extending the Web as a true enterprise platform.

In essence, what we are talking about here is a revolution in the way we approach development, integration, and enterprise architecture, understanding that the best could be within the on-demand world. What was once right for application deployment is now right for development, and the integration of those applications into unified solutions. So, we are indeed moving to on-demand architecture, an approach and technology solution that will change the game in terms of how we think about enterprise computing going forward.

Emerging Web-Delivered Platform

We are moving in three clear directions:

- First, the movement from visual to service-based interfaces.
- Second, the movement to outsourced or virtualized business processes.
- Finally, the acceptance of an on-demand platform for applications, services, and now development and enterprise architecture.

The movement from visual to service-based interfaces is changing the way we think about computing. Where once we leveraged on-demand applications through user interfaces driven by Web browsers, today we are quickly moving toward leveraging remote services that are delivered on-demand. This provides the architect with the ability to mix and match services, or application behaviors, allowing him or her to extend the value of those applications to other systems within the architecture. In essence, this is the notion of SOA, or, creating the infrastructure to take advantage of these services and turn them into solutions, something that is moving to an on-demand platform/architecture as well.

The movement toward outsourced or virtualized business processes is basically building on the movement toward service-based interfaces. We are seeing business processes, once the exclusive home of enterprise applications housed in corporate data centers, created, maintained, and leveraged using an on-demand infrastructure. Thus, the business processes are virtual, sharable, and benefit from both economies of scale, and the agility that this approach provides. For instance, no longer are architects bound by processes housed within their enterprise, but they can access thousands upon thousands of processes on-demand, and mix and match those processes to form critical business solutions, or more importantly, remix them as the needs of the business changes.

Today the use of on-demand technology has proven to be a valuable asset within many enterprises who had not considered such an approach only a few years ago. As a result, the use of an on-demand application delivery mechanism has wide acceptance at the application level, and is quickly moving to the platform and architecture levels as well. Indeed, many of the same benefits realized when leveraging on-demand applications are just as apparent, perhaps more so when considering platform and architecture. Benefits such shared services and processes, shared design and architecture patterns, built-in integration capabilities, and the huge cost savings, considering that the infrastructure is hosted and maintained outside of the enterprise, typically with lowered costs and reduced downtime.

The Changing Game

So, how does an on-demand platform fit into the existing world of enterprise architecture? When considering this question, you have to consider both the dimensions of the technology and business drivers.

When looking at the technology we are clearly moving from something that's more complex and monolithic, to something that's more efficient and dynamic. In short, the use of an on-demand platform provides the enterprise architect with the ability to:

“Why reinvent the wheel by having your staff spend time building service components, when you can quickly subscribe to a component, that's been tested and uptime certified, and pay for it on as-used basis?”

– Joe McKendrick,
WebServices.org

- Define, build, test, and deploy critical business applications in a fraction of the time by leveraging centralized and highly available platform resources.
- Leverage a community that shares architectures, applications, services, and information, leveraging the centralized nature of the on-demand platform.
- Leverage on-demand 3rd party applications, tools, and services that live within the on-demand platform, and are tested to work and play well with your applications and services.
- Have a complete directory of enterprise resources, including data, services, processes, applications, and security components.
- Leverage best-of-breed centralized resources, including state-of-the-art development tools and application components. All up-to-date, and available on-demand.
- Enjoy built-in infrastructure that you neither host nor support, but is always up-to-date.
- Provide built-in support for partner and customer integration that includes hosting shared applications, and information sharing mechanisms, using the on-demand platform.

For the business, the value is apparent as well, including the ability to:

- Provide a more agile architecture, allowing IT to quickly align with the business, such as retooling for a merger or changing major business processes.

- Define, build, test, and deploy critical business applications at a fraction of the cost by avoiding both the acquisition of expensive hardware and software, and, most of all, avoiding costly maintenance over the lifecycle of the application or architecture.
- Reuse of core enterprise assets, either built by the enterprise or a 3rd party, thus avoiding redundant development costs.
- Optimize uptime, supporting all critical business processes, efficiencies that go right to the bottom line.

The Changing World

We know by now that the world is changing, so let's examine the drivers behind that change. Web 2.0 and its technology have been changing the way in which we think about computing. While Web 2.0 does relate to social networking, the true value of the emerging Web has been the acceptance of the Web as a true platform.

Moreover, the notion of SOA provides the infrastructure for the use of the emerging Web, and on-demand platforms within the enterprise. SOA means that existing enterprise applications have the ability to leverage services existing inside or outside of the firewall. Additionally, SOA means that existing enterprise applications produce services as well, for consumption by on-demand systems. In essence, SOA creates the infrastructure for quick and easy integration with the emerging Web, on-demand applications, and, now, on-demand platforms and architecture.

The critical need for most Global 2000 enterprises today is agility, or the ability to change their business processes to accommodate changes in the business. Existing enterprise architectures typically fall short in providing agility because layers upon layers of monolithic enterprise applications have hindered the ability to easily alter business processes. Indeed, major changes to business processes, such as a merger with another company, or the adding of a new product line, are in many instances redevelopment efforts, and can take months, or, in some cases, over a year to accomplish. As such, the lack of agility has hindered or hurt many businesses, and thus has impacted the bottom line perhaps more than is currently measured.

“Why reinvent the wheel by having your staff spend time building service components, when you can quickly subscribe to a component, that's been tested and uptime certified, and pay for it on as-used basis?”

– Joe McKendrick,
WebServices.org

Reuse has a long history in computing, and something that has yet to be effective in most enterprises. Truth be told, we just have not had the infrastructure to create, register, discover, and leverage code or services inter-application. What has been missing is a common sharable platform for the creation and management of services, and the ability to discover and leverage those services from application to application. What's more, we were missing the ability to share services intra-enterprise and inter-enterprise, as well as a common metadata and database layer.

Changing Role of SaaS

As mentioned above, the use of Software as a Service (SaaS)-delivered applications is now widely accepted, and most Global 2000 organizations have some sort of subscription-based SaaS-delivered application operational today. Typically these on-demand applications deliver services using visual user interfaces, but they are also moving to true service (e.g., Web service)-delivered systems as well, allowing enterprise architects to leverage SaaS-delivered services as enterprise services, getting additional value beyond the user interface.

Considering that SaaS is accepted and indeed a critical part of many enterprise architectures, the next logical step is the use of the SaaS model to provide platform services, including design, development, testing, deployment, database, and integration. While clearly a huge leap in thinking for many traditional developers and architects, the use of an on-demand platforms makes logical sense when considering the advantages of this model, including:

- The cost of the platform is always going to be less expensive than more traditional platforms, and also provide more value.
- The shareable nature of this platform allows designers, developers, and architects to leverage best practices, reusing existing design, code, configurations, metadata, and application architectures.
- Services deployed on this platform are immediately sharable, intra- or inter-enterprise, for any business purpose, including B2B partner integration.
- Services may be layered into an orchestration mechanism or process layer for configuration into solutions.

Considering an On-Demand Platform and Enterprise Architecture

Beyond the notion of an on-demand platform, we have *on-demand enterprise architecture* as the next logical step in the maturation of this technology. As alluded to above, this is both evolutionary and revolutionary, and has the potential to save the Global 2000 huge amounts of money when considering the value of agility, reuse, and the cost savings of not having to maintain costly infrastructures.

As with on-demand platforms, the core notion of on-demand enterprise architecture is to take traditional enterprise architecture assets and resources, and leverage them as shared resources within a virtual private environment. This provides the architects with the opportunity to leverage this new model, which is much more cost effective, but can also reinvent their architecture on an entirely new platform, supporting all core services including design, development, testing, deployment, integration, database services, management services, and governance, as well as managed and supported environments that have superior up time.

The use of this approach to enterprise architecture as all of the advantages as previously stated, as well as the added advantage of staged migration. Since this is a service, and not a huge procurement of both technology and the platform, the use of the on-demand platform can take on specific processes and enterprise data, as desired by the architect, and be given more and more platform responsibilities over time. For example, an architect could move the inventory processing system to the on-demand platform for processes, persistence of data, and integration

to systems within the enterprise or perhaps located within a partner's data center. In any event, the use of this architecture can occur at the pace of the architect without incurring excessive costs. We'll discuss more in the steps to an on-demand platform later in this paper.

Considering the Business

Using this new approach there are numerous business advantages, including the more economical subscription-based model, shared infrastructure, and the ability to avoid expensive platform costs. For instance, with license costs for databases running \$300,000 per year, and hardware and development tools costing about the same, the use of a subscription-based platform will run only a fraction of those costs. Moreover, you're able to leverage as much or as little of the platform as you need.

Also, you need to consider the value of the platform results, as well as the platform itself. The two primary drivers here are agility, which we already began the discussion of above, and reuse of services and information.

Under the concept of *service reuse*, we have a few things we need to determine to better define the value. These include:

- The number of services that are reusable.
- Complexity of the services.
- The degree of reuse from system to system.

The number of reusable services is the actual number of new services created, or, existing services abstracted, that are potentially reusable from system to system. The complexity of the services is the number of functions or object points that make up the service. We use traditional functions or object points as a common means of expressing complexity in terms of the types of behaviors the service offers. Finally, the degree of reuse from system to system is the number of times you actually reuse the services. We look at this number as a percentage.

Just because we abstract existing systems as services, or create services from scratch, that does not mean that they have value until they are reused by another system. In order to determine their value we must first determine the Number of Services that are available for Reuse (NSR), the Degree of Reuse (DR) from system to system, as well as the Complexity (C) of each service, as described above. The formula to determine value looks much like this:

$$\text{Value} = (\text{NSR} * \text{DR}) * C$$

Thus, if you have 100 services available for reuse (NSR=100), and the degree of reuse at 50 percent (DR=.50), and complexity of each service is, on average, at 300 function points, the value would look like this:

$$\text{Value} = (100 * .5) * 300$$

Or

Value = 15,000, in terms of reuse.

If you apply the same formula across domains, with consistent measurements of NSR, DR, and C, the relative value should be the resulting metrics. In other words, the amount of reuse directly translates into dollars saved. Also, keep in mind that we have to subtract the cost of implementing the services, or, creating the on-demand platform, since that's the investment we made to obtain the value.

Agility is a strategic advantage that is difficult to measure in hard dollars, but not impossible. We first need to determine a few things about the business, including:

- The degree of change over time.
- The ability to adapt to change.
- Relative value of change.

The **degree of change over time** is really the number of times over a particular period that the business reinvents itself to adapt to a market. Thus, while a paper production company may only have a degree of change of 5 percent over a 5 year period, a high technology company may have an 80 percent change over the same period. The **ability to adapt to change** is a number that states the company's ability to react to the need for change over time. Finally, the **relative value of change** is the amount of money made as a direct result of changing the business. For instance, a retail organization's ability to establish a frequent buyer program to react to changing market expectations, and the resulting increases in revenue from making that change.

Clearly, the business advantages match the technical advantages when considering an on-demand platform, and its value for enterprise architecture. The cost of maintaining the platform is very low considering more traditional approaches, and the ROI from the use of that platform, namely the benefits of agility and reuse, is very high.

Salesforce.com's Apex On-Demand Platform

Taking into account both the value of an on-demand platform when considering the next generation enterprise architecture, it's helpful to examine salesforce.com's Apex platform. The Apex platform is the world's first multitenant, on-demand platform, providing a no-software, no-hardware approach to application creation, customization, integration, and management. What's unique about Apex is not the on-demand model it employs, but the value the platform brings to large and small organizations that are looking to reinvent their enterprise architecture into something far more valuable by leveraging the notions and concepts of the emerging Web, including a sharable on-demand infrastructure.

As a platform, Apex provides the following core services (see Figure 1):

- Apex Builder for application construction, testing, and deployment.
- Apex Connect for application integration.
- Apex API for application services from the platform.
- Apex Code for application development.

- Apex DB for database services.
- Apex OS for core operating systems and platform service.

Let's drill down each service in a bit more detail.

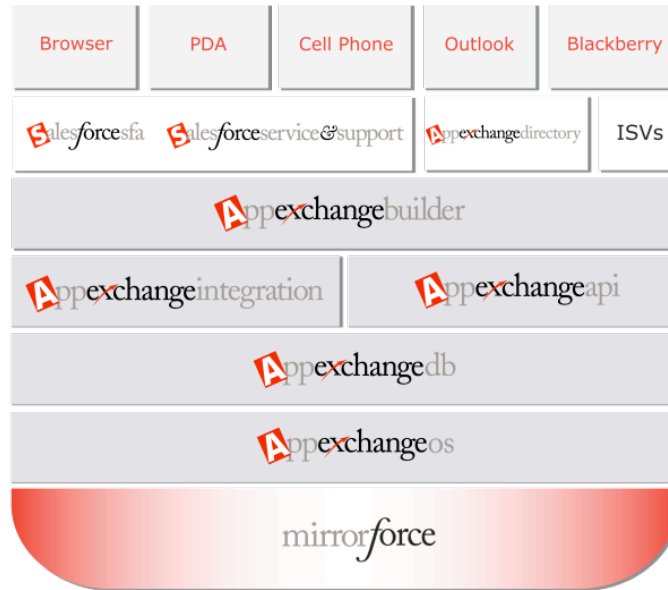


Figure 1: The Apex platform.

Apex Builder and Apex Code

The Apex programming language and development platform, Apex Code, is a new tool for developers interested in building applications that are core to the enterprise. When using Apex Code (see Figure 2), you have the potential to create a new breed of on-demand applications, including sophisticated processes and business logic, entirely on-demand and without software. Also, Apex Code will enable many levels of customization that make it possible to modify the behavior of existing features or create entirely new ones.

```

Integer NUM = 10;
Account[] accs;
// Clean up old data
accs = [ select id from account where name like 'test%'];
delete accs;
commit;
accs = new Account[ NUM ];
for (Integer i = 0; i < NUM; i++) {
    accs[i] = new Account(name='test ' + i, outstandingshares_c-1);
}
insert accs;
Contact[] cons = new Contact[ 0 ];
for (Account acc : accs) {
    cons.add(new Contact (lastName=acc.name + '1', accounted=acc.id));
    cons.add(new Contact (lastName=acc.name + '2', accounted=acc.id));
}
insert cons;

```

Figure 2: Apex code is a Java-like development environment.

For developers, the Java-like environment will be immediately familiar, so they can quickly leverage the many features of the Apex language. Because Apex Code runs natively on the server, it can interact with the user interface via buttons and events, so developers can manipulate data, channel transactions, and implement flow controls on the server side. That allows them to do everything from create custom components, customize and modify existing salesforce.com code, and create triggers and stored procedures, all the way to building and executing complex business applications.

Apex Connect

Apex Connect is a set of technologies built on top of the Apex Web Services API that reduces the effort associated with integration. Apex Connect provides five paths to integration by leveraging technologies—including Web services, multitenant architectures, and publicly available APIs—and an extensive integration partner ecosystem.

“Mash-ups portend big changes for software companies, Web sites, and everyone online. No longer just a collection of pages, the Web is morphing into a sort of global operating system ...”

– Business Week

For enterprise architects, Apex Connect means integration with any enterprise application or system, including SAP, Oracle, Microsoft, and other third-party solutions. Apex Connect avoids intricate technology dependencies, or vendor lock in, while providing the flexibility to take advantage of emerging Web services.

Apex API

The Apex API provides programmatic access to core Apex platform services, allowing developers to access the features of the Apex platform for any purpose within any application. This may include, but not be limited too: Mashups, enterprise application development, integration, and mobile applications. The existence of Apex API means that anything is possible when it comes to mixing and matching the capabilities of the platform.

Apex DB

Apex DB (database) provides a data services layer (see Figure 3), allowing the application and service developer to persist information as needed by applications. Instead of hard-coding data tables and page layouts, users define attributes and behaviors as metadata, which functions as the application's logical "blueprint." This means the developer can work from the information up to the services or application, defining the critical metadata first, and building up from there.



Figure 3: Apex DB provides a strong foundation in metadata.

In addition to an application's metadata, virtual partitions leveraged with Apex DB will also store custom code developed with the Apex Code programming language. This ensures that any potential problems with that code will not affect other customers, and preventing bad code associated with one object from affecting any other aspects of an individual's application.

Apex OS

Apex OS is the world's first on-demand operating system in support of an on-demand platform. Apex OS performs all of the duties of a traditional operating system, including resource utilization, storage, memory management, security, peripherals, multitenant services, recovery services, and application platform support.

Approaching Enterprise Architecture Using an On-Demand Platform

So, how does one build an enterprise architecture using an on-demand platform? The approach is actually easier than more traditional notions, and more effective. Consider the following seven steps:

Step 1: Accessing Current Enterprise Architecture Issues

First, you need to understand your own requirements, including what's working and what's not. Take an honest assessment of the architecture's ability to service the current business, including ability to change the enterprise architecture to adapt to new opportunities, reliability, manageability, and reuse of core IT assets, including services and data.

Step 2: Making the Business Case

Next, and using the concepts presented above when considering ROI, it's time to make a business case for the new architecture, specifically looking at the value the architecture will provide over a five year horizon.

Question asked now include:

1. What are the cost savings from reuse?
2. What value will agility bring to the enterprise?
3. What are the cost savings when leveraging an on-demand platform?
4. What about efficiencies gained by the end users when leveraging this new platform?
5. Opportunities for automation and integration?
6. Opportunities for sharing processes with both partners and customers?

Step 3: Understanding Semantics

You can't deal with information you don't understand, including information bound to behavior (services). Thus, it is extremely important for you to identify all application semantics—metadata, if you will—that exist in your domain, thus allowing you to properly deal with that data when moving to this new platform.

The understanding of application semantics establish the way and form in which the particular application refers to properties of the business process. For example, the very same customer number for one application may have a completely different value and meaning in another application. Understanding the semantics of an application guarantees that there will be no contradictory information when the application is integrated with other applications at the information or service levels.

Step 4: Understanding Services

It is important to devote time to validating assumptions about services, including:

1. Where they exist.
2. The purpose of the service.
3. Information bound to the service.
4. Dependencies (e.g., if it's a composite service).
5. Security issues.

The best place to begin with services is with the creation of a services directory. As with other directories, this is a repository for gathered information about available services, along with the documentation for each service, including what it does, information passed to a service, information coming from a service, etc.. As an option, this directory may be created within the on-demand platform.

Step 5: Understanding Processes

You need to define and list all business processes that exist within your domain, either automated or not, for relocation to the on-demand platform. This is important because, now that we know which services and information sources and sinks are available, we must define higher level mechanisms for interaction, including all high-level, mid-level, and low-level processes. In many instance, these processes have yet to become automated or are only partially automated.

You should also consider the notion of shared versus private processes. Some processes are private, and thus not shared with outside entities (or, in some cases, they are not even shared with other parts of the organization). Other processes are shared, meaning that others leverage the same processes in order to automate things inter-enterprise. Private and shared processes can

exist in the same process space with the process integration technology that manages security among the users.

Step 6: Understanding the Technology

Next it's time to understand the new platform, and the underlying details that will support the new architecture. The Apex platform is described above, and it's this step where you map your understanding of your data (application semantics), services, and processes into the on-demand platform.

Step 7: Execution and Assessment

Finally, it's this step where you execute the plans and understanding developed in the previous steps. You move information, services, applications, and integrations to the new platform, making sure to test and evaluate the use of the new on-demand platform.

On-Demand Platform...the Future of Enterprise Architecture

While this is a huge step in the world of enterprise architecture, the value and fit to purpose are clear when considering the core notions. Truth-be-told, enterprise architectures today are in a state of disarray, and are actually hindering the growth of the business, this due to years and years of layering in expensive, static, and monolithic applications that have hindered business agility, costing millions of dollars over the years.

When leveraging an on-demand platform, you can:

- Create a dynamic agile architecture that's also cost effective.
- Build, test, and deploy critical business applications in a fraction of the time by leveraging centralized and highly available resources.
- Leverage best-of-breed centralized resources, including state-of-the-art development tools and application components.
- Leverage a community that shares architectures, applications, services, and information, leveraging the centralized nature of the on-demand platform.
- Leverage on-demand 3rd party applications, add-on, and services that live within the on-demand platform.
- Provide built-in support for partner and customer integration that includes hosting shared applications, and information sharing mechanisms, using the on-demand platform.
- Reuse of core enterprise assets, either built by the enterprise or a 3rd party, thus avoiding redundant development costs.
- Optimize uptime, supporting all critical business processes, efficiencies that go right to the bottom line.

The notion of an enterprise architecture leveraging an on-demand platform provides key technology and business drivers that make this approach compelling for enterprises both large and small. Moreover, this provides the enterprise architects with the ability to migrate over to the on-demand platform, as needed, and without disrupting existing enterprise IT operations. The more processes, data, applications, and services moved to the on-demand platform, the more value the enterprise will realize over time.

About the author:

David S. Linthicum is the CEO of the Linthicum Group, LLC, a consulting and advisory organization dedicated to excellence in both SOA and the influence of the emerging Web within the enterprise. David has authored 3 best selling books on enterprise architecture, SOA, and integration, and keynotes at most major industry conferences. You can reach David at david@linthicumgroup.com.