COMMUNITY
INSIGHT REPORT

Evolution of Dynamic IT

A comprehensive primer for medium sized businesses

Dale Vile and Tony Lock, October 2009

Against the backdrop of an increasingly fast moving and interconnected trading environment, with renewed pressure on operational efficiency as a result of economic uncertainty, the need for robust yet flexible IT support has never been greater within the business. Yet many IT departments are constrained in their ability to respond to changing business requirements, mainly because of the relatively static way in which IT has traditionally had to be implemented. The latest ideas and solutions from the IT industry, however, promise to help organisations develop a more dynamic approach to IT service delivery. But, how does this translate to practicality in a medium sized business environment in which resources and time are scarce and the opportunity to investigate and experiment is limited? The aim of this paper is to provide an overview of the most relevant developments that have emerged from IT suppliers and, as importantly, to look at how they fit together to enable the delivery of IT in a more efficient, flexible and business centric manner.

MAIN POINTS

The time and cost associated with IT delivery can translate to a direct business constraint

Many factors conspire to hamper IT responsiveness, including, quite often, the behaviour of business users and stakeholders themselves. That said, most IT landscapes have a degree of inherent rigidity that extends the time and cost of getting things done. As a result, IT delivery is frequently on the critical path for business change initiatives, constraining the rate of progress.

The historical nature of IT, and how investments are funded, lies at the root of the problem

The hardwiring of hardware to software and the monolithic black box nature of business applications have conspired to create a fragmented and inflexible IT landscape in many organisations. This inflexibility has been further reinforced by a ‘systems ownership culture’ in which the way IT is funded creates artificial constraints on making the optimum use of IT assets.

Embracing the latest ideas for dynamic IT delivery can break the inflexibility cycle

Three key ideas and solution areas exist that form the pillars of a more flexible, efficient and dynamic IT environment - infrastructure virtualisation, service oriented architecture (SOA) and a blended approach to resourcing of both manpower and systems capability, all underpinned by the solid foundation of business service management from a delivery and operations perspective. While valuable in their own right, it is when dynamic IT concepts are applied together that significant benefits are achieved in terms of efficiency and flexibility.

Driving improvement isn’t hard, but dynamic IT is more of a direction than a goal

Many of the capabilities that enable dynamic IT are already working their way into the products and services in common use today. The trick to driving improvements is therefore to understand what’s possible, and implement ideas naturally as opportunities arise. Our advice is to adopt a business centric approach, and regard dynamic IT as a direction rather than a goal.

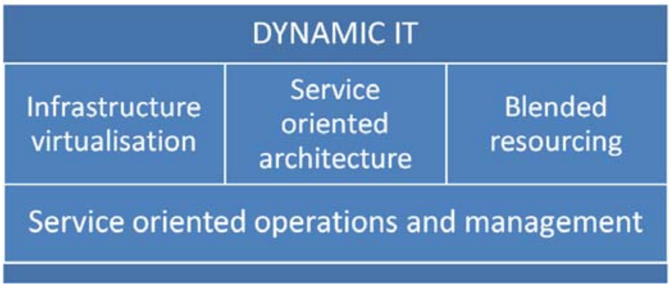
This report was commissioned by IBM and authored on an independent basis by Freeform Dynamics Ltd based on input from a range of IT vendors and services providers, coupled with intelligence from large scale primary research studies into the investment and use of IT in a business context.

Management Summary

While IT professionals generally do a pretty good job of delivering services to the business, the very nature of technology itself has historically imposed a range of constraints that have limited their ability to optimise efficiency. The rigidity and fragmentation that has arisen from years implementing applications as discrete systems with software tightly coupled to hardware has so often led to high overheads and elongated implementation times when new capability is required or existing capability needs to be modified. Indeed, two of the most common complaints heard from business stakeholders are that IT always seems to cost too much and take too long to deliver.

Fortunately, developments over the past few years have opened up opportunities to create a more dynamic IT environment that is naturally more flexible and efficient. These developments include a range of ideas as well as enabling technologies that together allow an IT department to both reduce costs and increase service levels and responsiveness to the business.

The keys to enabling this more dynamic approach to IT can be summarised as follows:


DYNAMIC IT
Infrastructure virtualisation
Service oriented architecture
Blended resourcing
Service oriented operations and management

Infrastructure virtualisation allows the de-coupling of hardware and software. This in turn leads to better utilisation of systems assets such as servers and storage devices, as applications can share resources in ways that have historically been difficult or impossible due to hardwired dependencies. Whether this translates to doing more work with the same number of assets, or the same amount of work with fewer servers and other devices depends on the context. Either way, flexibility and responsiveness are enhanced, and operational overheads and capital costs are lowered.

Developments in the software arena, particularly around componentisation and standards-based web services interfaces, have enabled software to be built and integrated more flexibly and efficiently. Together with appropriate middleware to facilitate and even orchestrate the free flow of transactions between systems, these trends have led to the evolution of service oriented architecture (SOA). Implemented appropriately, SOA can reduce duplication and inconsistency between systems, as well as allowing changes to be implemented more quickly and cost-effectively.

One of the big benefits of the openness and flexibility resulting from the adoption of virtualisation and SOA is the freedom created to tap into external resources. Whether it’s the skills and experience offered by professional services organisations, or resources and functionality delivered ‘over the wire’ by cloud service providers, the dynamic IT approach makes it easier to blend internal and external capability to optimise the IT delivery process. As things become more fluid and dynamic, with activity crossing organisational boundaries, however, proper attention needs to paid to operations and management. The recommendation here is to adopt a business service view of the world, focusing on the performance of what’s delivered rather that what is used to deliver it.

Dynamic IT, based on the three pillars of infrastructure virtualisation, SOA and a blended approach to resourcing, all underpinned by a service oriented approach to operations and management, thus has a lot to offer in terms of removing traditional constraints, reducing costs and enhancing service delivery. The principles underlying the approach, the practicalities of which are explained and explored in this paper, can be used to specify service and technology requirements and to qualify potential suppliers, as well as providing guidance on the optimisation of internal IT activity.

It is important to appreciate, however, that the best results will be achieved by working dynamic IT principles into ongoing activities. The most practical approach is evolution, not revolution.

Contents

Introduction

Information technology (IT) is advancing constantly. There is no let up in the rate of innovation among suppliers, from large IT vendors with rich R&D budgets to small creative start-ups. Throw into the mix the work taking place within the open source community and academia, and it can be quite a challenge keeping up with developments in new ideas and technologies, let alone figuring out where they fit into your plans and activities.

Larger organisations have an advantage, however. They have the economy of scale to justify strategic planning groups, highly specialised IT teams, exploratory testing activity, proof of concept studies, and so on. These factors, together with the presence of significant IT budgets, are very attractive to suppliers looking to drive something new into the market. Products and services in emerging areas are therefore often packaged and presented with the 'corporate need' in mind. As part of their sales and marketing process, IT vendors and consultancy firms then spend a great deal of time educating the 'corporate segment' on the nature and rationale for new solutions. This is backed up by the services of large industry analyst firms such as Gartner and Forrester who sell advice and guidance to large organisations on what to do and what to buy.

Meanwhile, the relevance of the latest ideas and technologies to smaller organisations is often overlooked. Yet if we consider the IT needs of a medium sized business of, say, between 100 and 1000 employees, they are fundamentally not that that different to a large corporate. Sure, things operate on a smaller scale, but whether it's supporting day to day operations such as sales, accounting, logistics and customer services, or allowing the workforce to communicate and collaborate as they manage and drive different aspects of the business, comprehensive IT solutions are still an integral part of the mix.

Furthermore organisations of all sizes compete in the same globally connected business environment in which ever increasing automation is not only speeding the pace at which everything happens, but is also generating ever increasing volumes of data along the way. For those geared up to take advantage of this through efficient and responsive business practices supported by equally efficient and responsive IT systems, the opportunities are significant. For those constrained by inefficiency and inflexibility, though, the world is becoming a riskier place.

Against this background, this paper considers advances in technology and related ideas that have emerged over the past few years to enhance the efficiency and flexibility of IT infrastructures. While many of these 'dynamic IT' concepts have already been embraced by many larger corporates, our treatment here is very much geared to IT professionals working in a medium sized business environment. In line with this, we will be taking a very pragmatic look at what lies behind some of the buzzwords and jargon for which our industry is so renowned – virtualisation, service oriented architecture, cloud computing, business service management, and so on – providing both a basic grounding in each area, and a view of how and why approaches and solutions might be relevant.

While reading this paper, you may find that you are already familiar with some of the areas covered. We make no apologies for this as an important prerequisite for the dynamic IT discussion is a high level understanding of developments across a number of domains, so we are making no assumptions about the completeness of readers' knowledge and experience. Furthermore, even if you are an expert in a particular area, you may find that we look at it from a different perspective in the context of the discussion we are having, so it still may be worth reading the section concerned.

Inputs into this paper

This paper has been put together by Freeform Dynamics, based on in-depth briefings from a range of relevant IT vendors and professional service providers, coupled with the findings of large scale research studies during which views and experiences are gathered from IT and business professionals on their experiences of deploying IT solutions in a business context. As such, we have taken into account both the supplier and the customer sides of the equation, allowing us to explore objectively how the capabilities of the former can be applied to meet the needs of the latter.

For more information on Freeform Dynamics and its work, please visit www.freeformdynamics.com.

In the meantime, let's begin our discussion by taking a closer look at why 'dynamic IT' matters.

The business rationale for a more dynamic approach

Given the level of dependency on technology in most organisations nowadays, many significant changes or developments at a business level will require some kind of action or response by the IT department. In an ideal world, business people would define what they need, and the response by IT would be quick and efficient. But how often is it as simple as this?

The reality is, of course, that many factors conspire to get in the way. Business people are not always clear in articulating their requirements, and even when they are, or IT works with them to crystallise what's needed, they often have preconceived notions about costs, timescales and practicalities that are unrealistic and need to be worked through. Then, of course, their priorities and objectives can evolve over the course of a project, so requirements become a moving target.

It is beyond the scope of this paper to look in depth at ways of tackling these perennial issues. Suffice it to say that effective expectation and change management are generally the keys to keeping things under control, which in turn depends on good dialogue and discipline to be accepted in both the IT and business camps. Easier said than done, perhaps, but undeniably true, and something which business stakeholders, who are frequently the weakest link in the equation, would ideally take more responsibility for in many organisations if IT had its way.

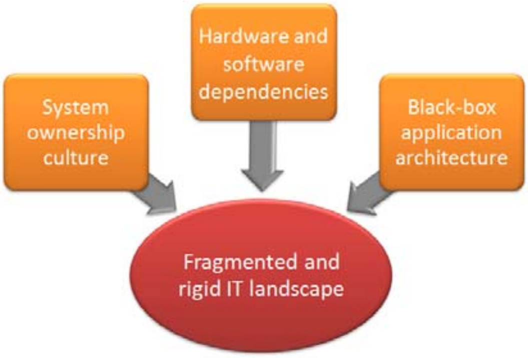
Having said this, if most IT departments are honest with themselves, they will admit that the way in which IT infrastructures have evolved in the past, and the way they have historically been operated, could be improved upon also. All too often, despite expectation misalignments, the business is genuinely constrained in how quickly it can move because of IT restrictions. Sometimes it is a case of having to compromise on plans and activities because the cost of supporting the ideal way forward in IT terms is prohibitive. On other occasions it might not be the cost *per se* but the time it would take to implement systems related requirements. Either way, this can lead to undesired constraint or exposure.

High Delivery Costs
+
Long Delivery Times
=
Business Constraint or Exposure

With this in mind, while IT professionals cannot be held completely responsible for how effectively IT enabled change is implemented in the organisation, there are some things within their control that they can pay attention to in order to improve this situation, i.e. to make IT less of a constraint on the business, and indeed to create an environment where IT is a positive and proactive force in the organisation. This, in a nutshell, is the rationale for looking at ways of creating a more dynamic IT environment. Before getting into what this translates to in terms of ideas, technology and actions, however, we must first take a look at some of the common constraints under which the IT department itself is working.

Understanding the constraints on IT

When we consider some of the constraints placed on IT professionals, it becomes clear that performance and delivery challenges are generally down to factors that have historically been beyond their direct control. In particular, there are three significant constraints that conspire to create a combination of fragmentation and rigidity in the IT infrastructure which lies at the root of so many challenges:


Hardware and software dependencies
System ownership culture
Black-box application architecture
Fragmented and rigid IT landscape

Systems ownership culture

The first constraint is nothing to do with the physical aspects of IT, but with the way in which organisations often fund and account for investments and operational costs. While IT professionals think quite naturally in terms of applications, infrastructure, integration, maintenance, support, etc., i.e. breaking the IT landscape and its operation into the various horizontal layers, business people tend to think more in terms of discrete 'systems' in a top to bottom manner. It is common to hear users refer to 'the accounting system' or 'the sales system', for example, because regardless of what's going on behind the scenes, this is how IT capability manifests itself to them.

This way of looking at things is then reflected in the ownership culture that prevails in most businesses, in which a specific IT system is typically regarded as the 'property' of a specific group, department or division. Investment and ownership then ends up being associated with everything required to deliver and operate a system – not just the application, but the servers, storage and any other infrastructure required to enable the solution to work. This in turn places artificial constraints on the IT department when it comes to making optimum use of the organisation's assets.

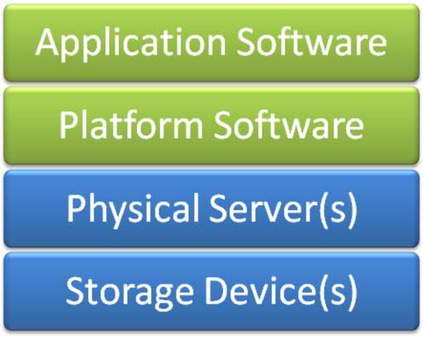
One way of solving a capacity and performance problem for a given application, for example, might be to swap servers with another application that is sitting on a bigger box that is significantly under-utilised. But will the department that 'owns' the other box, having funded it as part of last year's investment, be happy with this? Then there is the question of squaring things in the books from an asset/cost centre allocation perspective. When you stand back and consider such questions dispassionately, it all sounds very trivial, but the politics alone mean it is often easier for IT to just live with the way things are, rather than challenge cultural and administrative constraints.

This is one of the reasons why servers have proliferated over the years in many computer rooms and data centres. The system centric ownership and investment culture encourages the approach of putting each new application onto its own dedicated server, with the hardware and software remaining wedded to each thereafter for administrative, financial and political reasons.

We'll examine some of the consequences of this shortly, but there's also another reason, to do with more practical dependencies, for applications ending up tied to specific physical servers.

Hardware and software dependencies

Most IT professionals are familiar with the concept of the 'systems stack', which allows us to consider the various layers involved in constructing a solution and how they relate to each other. Here, for example, is a very simple schematic of a generic stack that may not be representative of all real-world systems (there are lots of variations in how things can be put together), but is a useful starting point for discussing hardware and software dependencies.



The bottom two layers obviously represent the hardware platform, and sitting on top of this is typically a collection of 'enabling' software components such as operating systems, application servers, web servers, database management systems, etc. These together create a 'software platform' upon which the business application executes, providing all of the necessary runtime services required.

While in an ideal world, the components residing in the various layers should be substitutable, in practice, it has historically not been as easy as that. Not only will each layer expect the next layer down to be of a certain type (.Net versus Java for application serving, Oracle versus DB2 for database management, etc.), but the underlying layer will also generally need to be configured and tuned appropriately for things to run optimally. If you set up the physical server and software platform components to optimise one application (software versions, patch levels, memory, disk configuration, etc.), there is likely be a mismatch with the requirements of other applications.

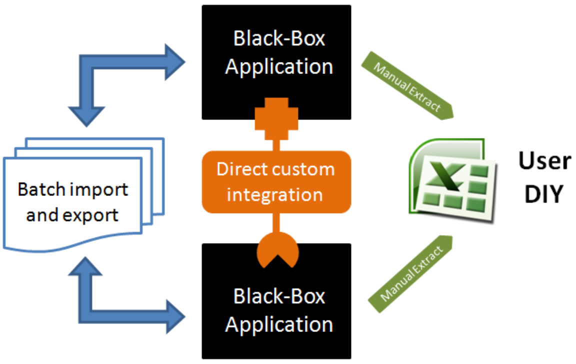
The key point here is that applications often conflict with each other in terms of their platform requirements, which is another factor that has encouraged the practice of implementing applications with dedicated servers, and often dedicated storage too.

Proprietary black-box applications

Focusing on the top layer of our systems stack, the applications that reside here are primarily concerned with business functionality. From a design and construction point of view, such applications have typically been put together to be largely self-contained. This is particularly true of packages that are built with a specific purpose in mind and are bought off the shelf from a software vendor – ‘enterprise resource planning’ (ERP), ‘customer relationship management’ (CRM), etc.

Generally speaking, the suppliers of such applications do not want customers tampering with the way they work internally. The danger is that as soon as you go under the covers and start changing things, unless you're very familiar with the design, code, data structures, and so on, you may disrupt the stability and/or functional integrity of the system. The same, indeed, can be said for custom applications built in-house or by a systems integrator that have any level of sophistication or complexity. The chances are the solution was built with a particular functional requirement in mind, and engineered with everything necessary included in it to meet this need.

The upshot is that the majority of applications deployed in business today are 'black box' in nature, and at the risk of stating the obvious, this means the inevitable integration between systems is typically achieved in systems terms through a combination of custom integration software built around proprietary ‘application programming interfaces’ (APIs) or cumbersome data import/export mechanisms. Beyond this, we then have the user DIY integration approach that is all too familiar, based on extracting and merging data from multiple sources using desktop tools.



None of these approaches is ideal from an implementation cost and time perspective. There is then work involved in keeping everything in sync as application upgrades and modifications disrupt the way in which interfaces operate.

Beyond integration mechanics, there are a couple of other issues that arise. The self contained nature of traditional applications means that functions and data are often duplicated between them. How many applications include functionality to enter or update a customer address or part description? How many versions of the same customer or part record exist across your systems, for example? These are simple illustrations of the issue but ‘redundancy’ of data and functionality between systems means that change requests from the business often translate to a need for equivalent modifications to be coordinated across multiple applications.

Lastly, the relatively fixed nature of traditional applications means users with exceptional or ad hoc needs typically need to come to IT for help. Whether it is pulling together information from different places to form a coherent view of a customer situation, or analysing data from multiple sources to support a key business decision, it is either a request to IT or the user doing it themselves through extracting and manipulating data offline, which risks all kinds of quality and consistency issues, not to mention the hit in terms of distraction and productivity.

Tangible impact of constraints

When considered together, the constraints we have been discussing have a significant impact on both business and IT performance. So let’s net this out in order to focus our minds.

Whether as a result of the systems ownership culture or practical dependencies, the tying of applications to physical resources has some undesirable consequences:

- Fragmentation of the IT landscape, which in turn leads to procedural complexity and duplication of effort. This results in high operational overhead which consumes a lot of IT time, budget and resource that would be better used for activities that enhance service levels or otherwise add value to the business.

- Poor utilisation of hardware as boxes tend to be sized for exceptional peak usage and may do little more than idle for much of the rest of the time. This translates to poor return on capital investments in IT equipment as well as elevating energy bills. Related to this, under-utilisation of software licences can also be an issue, as fees are often determined by the capacity of a server rather than how much of that capacity is actually used.
- Lack of flexibility, as reallocating, sharing and pooling physical resources to deal with new and evolving requirements is so hard. This makes responding to changing business or operational requirements, e.g. to bring new capability on board or catering for changing demands, both costly and time consuming.

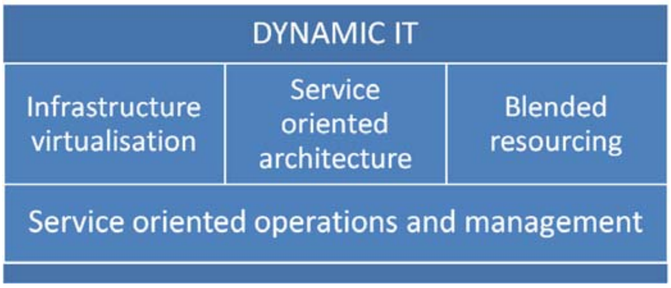
Turning to the consequences of proprietary black-box software, these net out to:

- Complexity of integration mechanics, which not only elevates costs, but has a direct impact on how quickly IT can implement new and changing business requirements. It is not uncommon for the bulk of the development and maintenance effort spent by IT departments to be concerned with integration related activities rather than on the implementation of core business functionality itself. This unfavourably distorts the cost/value equation associated with the use of IT budget and resources.
- Duplication of the same or similar functionality between systems increases the scope of development and maintenance work, again elevating costs and elongating delivery times. There is also the inevitable 'drifting apart' of systems over time, which leads to inconsistencies in both information and processes. This in turn causes inefficiency and risk at a business level.
- The self contained, some might say 'monolithic', nature of traditional applications limits the degree to which functionality can be mixed and matched between systems. As an example, if two applications are in place that both provide a supplier scoring mechanism, it is typically very difficult or impossible to get one application to ignore its 'embedded' capability and use the equivalent functionality in the other system instead. The result is having to live with inconsistencies and compromises that hamper the optimisation of systems.

Introducing dynamic IT

So much for where we have come from. Now let's switch tack and look at where we are going, and the one thing for certain is that we are headed in the direction of pretty much every aspect of IT becoming more fluid and dynamic.

In specific terms, there are three key pillars underpinning the practical side of dynamic IT. These are infrastructure virtualisation, service oriented architecture (SOA), and the idea of blended resourcing, which as we shall see, can relate to both manpower and systems. Experience has shown, however, that for everything to hang together, the pillars need to sit on a solid operational and management foundation which is geared to effective service delivery.


DYNAMIC IT
Infrastructure virtualisation
Service oriented architecture
Blended resourcing
Service oriented operations and management

Let's take a look at these areas a little further. Given that each of them is a specialist domain in its own right, and libraries of books and papers have been written about them covering the detail and richness of ideas and options that exist, we will restrict our treatment here to some of the basic principles and the ways in which they most commonly surface in practical terms. Please see the *Further Reading* section at the end of this document if you want to go deeper.

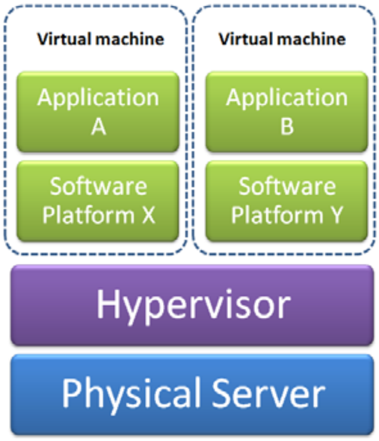
Infrastructure virtualisation

Virtualisation is a hot topic in the IT industry at the moment, and many IT professionals are already familiar with the concept in one form or another. The principle of virtualisation, however, has been applied to a number of aspects of IT, and it is important to understand the differences.

Server partitioning

In this context, the aim is to break that highly constraining marriage between hardware and software to unlock the full potential of servers. This is achieved by introducing an isolation layer known as a 'hypervisor' between the physical server and software that would normally run directly on it.

Using this hypervisor approach we are able to create multiple 'virtual machines' on a single box, which are essentially self-contained runtime environments that mimic the characteristics of a dedicated physical server, within which applications can execute, along with all of their supporting platform software (operating system, application server, etc.). Subject to the capability of the hardware, it is possible to 'partition' a single physical server into many virtual machines, and have it simultaneously support a mix of platform software and applications.



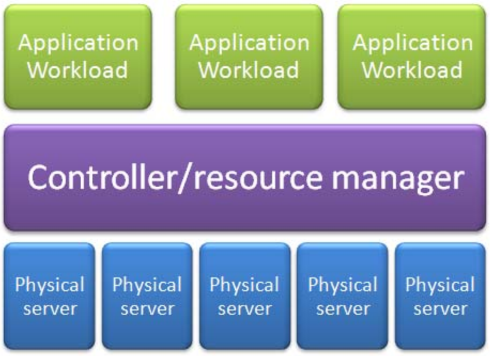
In practical terms, an important attribute of a virtual machine (VM) is that it is self-contained and exists separately from other virtual machines running on the same physical server. This provides the ability for each virtual machine to be configured and tuned for the application it is hosting. As an example, a single server may be partitioned into three virtual machines, one running Windows Server configured to support Microsoft Exchange, a second running Windows Server, but this time configured differently to run an old CRM application, and a third running Linux, optimised for Web serving. The point is that there are few restrictions on the degree to which different software stacks can be co-hosted, provided the server concerned can support the combined load.

On that note, virtualisation solutions allow various approaches to be taken to resource allocation. At one extreme, you can simply set things up to allow virtual machines to compete with each other for processor time, memory and network bandwidth. At the other extreme, it is possible to allocate 'slices' of processor capacity to virtual machines on a fixed basis. On a four processor box, for example, you might choose to dedicate two CPUs to a virtual machine running a particularly critical application to ensure adequate performance regardless of what else is executing on the server.

The advantages of server partitioning as described here are significant. Firstly, from a pure efficiency and cost perspective, we can get much more out of our physical hardware assets. Rather than a server sitting there idling or running at low utilisation for the majority of the time because it is restricted to supporting a single function, the available capacity can be exploited across a number of applications. As importantly, however, IT can respond much more rapidly and flexibly to changing requirements and imperatives as deploying, moving or updating a virtual machine, whether manually or automatically through policy driven triggers that are available in some solutions, is much quicker than working with physical servers.

Server pooling

This is a bit like turning the above server partitioning approach on its head, which is relevant for more demanding environments in which an application may need the horse power of multiple servers in order to run acceptably. Large consumer facing websites are an example of this, or perhaps core applications that serve a population of intense users, such as a call centre system. In some situations, an email system may fall into this category too, as might emerging 'media heavy' applications such as unified communications.

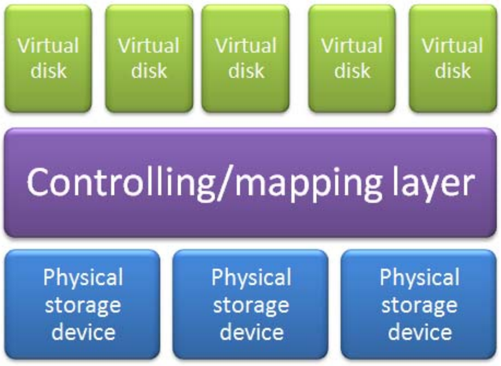


In implementation terms, there is again a controlling/mapping layer that in this case distributes the work of applications across multiple servers. The mapping that drives this is typically relatively static and managed on a manual basis, i.e. by pre-allocating a number of servers to each application supported by the pool. Even so, the management software available nowadays to enable this process takes a lot of the pain out of systems administration by easing the server 'provisioning' (set up) work required to prepare a server to run a particular application. While this assisted manual administration approach is the norm, however, in more sophisticated systems the provisioning and management process can be automated further, with the controlling layer allocating and deallocating resources dynamically as demand goes up and down.

Compared to server partitioning, which is 'mainstream mature' and very accessible to smaller businesses, solutions in this area are still evolving, and more commonly deployed in a larger corporate data centre environment. However, the general approach is one that that underpins the concept of 'cloud' or 'utility' computing, which is an emerging option for smaller organisations to tap into flexible pools of computing resource. We'll be discussing this in more detail a little later.

Storage virtualisation

With data volumes increasing almost exponentially in many organisations at the moment, managing the growth and operation of storage devices has become quite a challenge. While 'network storage' solutions, have alleviated some of the complexity associated with management of storage devices embedded in servers and desktops, the physical layer in the infrastructure can still be very prominent. This brings with it a lot of the same constraints on efficiency and flexibility as discussed in relation to servers.



Through virtualisation, again achieved by introducing a controlling and mapping layer into the equation, we can separate our 'logical' view of storage from the physical side of things. The end result is a set of 'virtual disks' being presented to applications running on servers or desktops, to which policies can be applied, just as you would with a physical device. The advantage, of course, is that behind the scenes, a logical device can map onto anything – a segment of a physical device, multiple whole devices, segments from multiple devices, or any other blend of storage that is appropriate.

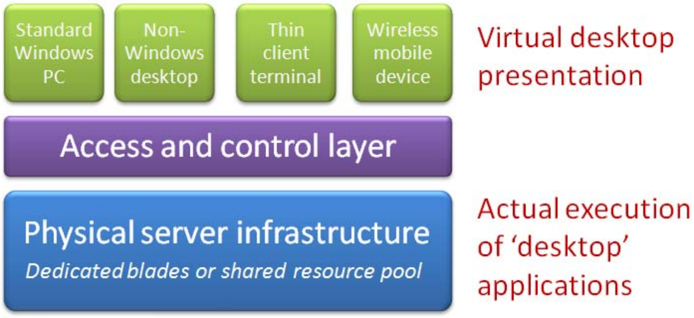
This allows far more freedom in terms of how resources are managed and how changes are made to deal with varying requirements. If an application needs more space, allocate more physical resource to the virtual disk it is using. If it evolves in a way that requires faster or more fault tolerant storage, a data migration and remapping exercise can be carried out without worrying about how the application will be affected. With all such changes, the application can continue using the same virtual disk, even though the nature and/or amount of underlying storage may have changed dramatically. Again, this can make the IT department much more responsive to changing business and/or operational requirements and help maximise the use made of IT storage investments.

Thin client computing

With the majority of organisations using traditional Windows based PCs for client-side computing, the complexity, cost and challenges associated with maintaining and supporting the desktop environment are all too familiar. Against this background, many IT vendors are starting to promote what is increasingly being referred to as virtual desktop infrastructure (VDI), the idea being to get away from having to deal with the huge variability and inconsistency that characterises most desktop and laptop computing estates, not to mention minimising the impact of user-initiated service interruptions.

The most established form of desktop virtualisation, the so called 'thin client' approach popularised by Citrix in the 90's, is nothing new. In this model, both the operating system (typically Windows) and applications run on the server rather than the desktop, and the client machine acts as a graphical display device. The client device itself can either be a specially designed thin client 'terminal' with limited local processing capability (not required as applications are running on the server) and no local storage, or a standard PC running thin client terminal software. In the case of the latter, because the desktop PC is only required to deal with the user interface part of the equation, it doesn't need to be that powerful. This can give a new lease of life to older kit that would otherwise be retired as the operating systems and applications become ever more demanding.

Either way, there are some other big advantages to the thin client approach. The operating system, applications, data and associated policies can be managed and deployed centrally. This not only reduces management overheads and security risks, but also allows new and updated applications to be rolled out more quickly and reliably. No longer do you need to worry about getting new capability running acceptably across two or three generations of Windows, with machine specs varying from the latest multi-core powered beasts with gigabytes of memory, to four year old work horses. From a user perspective, there is then the benefit of being able to run their 'virtual PC' from any desktop with access to the server. This, provides general flexibility in terms of where people can work, and can also form the foundation for more formal hot-desking, home working and branch office computing setups.


Standard Windows PC
Non-Windows desktop
Thin client terminal
Wireless mobile device
Virtual desktop presentation
Access and control layer
Physical server infrastructure
Dedicated blades or shared resource pool
Actual execution of 'desktop' applications

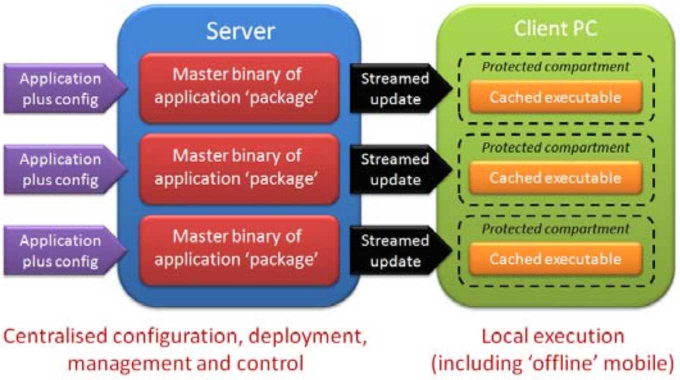
While the thin client model has been around for a while, it is still very much evolving. The traditional software based solutions are becoming more efficient, but we have also recently seen the emergence of a more hardware centric approach. This leverages some of the developments we have seen in server form factors, specifically around the use of 'blade architectures'. These were originally designed to allow lots of servers, each in the form of a 'blade' (processor, memory and other essentials on a card) to be packed into a tight space (a blade 'chassis' that provides shared power, cooling, networking, access to storage, etc). However, similar technology can also be used to move PC hardware from the desktop to the computer room. The advantage of this approach is that the user is running a dedicated 'PC blade' behind the scenes, so rather than relying on a software controller to share resources between many users on demand, which can lead to conflicts and performance issues in some scenarios, the user is guaranteed the same amount of processor and memory capacity as they would have with a traditional desktop. This is important for some users who rely on particularly demanding applications.

Of course the one big disadvantage of the thin client model is that the user needs to be connected to the server infrastructure for it to work – not a problem for desk based workers, but what about those that are mobile, or for whom connectivity is poor, expensive or cannot be relied on?

Desktop application streaming

A compromise between the traditional desktop computing and thin-client approach is desktop application streaming. The idea here is to centralise the management and deployment of PC software, but allow applications to run locally on either a desktop or a laptop. In effect, the model is based on a master copy of each application residing on the server which is 'streamed' (i.e. downloaded) to the client machine on demand, i.e. when the user wants to run it.

This may sound very slow and impractical given the size of modern applications, for example, and the bandwidth generally available from the network, but solutions in this space are based on intelligent caching and update of local application images. Once an application has been initially downloaded, there is therefore only need to stream changes and updates as required, something which is done in a highly optimised manner at a binary level.



The upshot is that the IT team only needs to maintain the master copy of each application. If an update is applied, the next time the user attempts to run the application when they are connected to the server, the difference will be detected and the local application cache updated accordingly. This means less work for IT, and an assurance that users are (subject to them connecting) always running the correct version of an application with the latest policies applied.

The other big benefit of desktop streaming is that each application runs on the local machine in its own ‘compartment’, similar in principle to the virtual machine concept previously described. The difference is that while applications are prevented from interfering with each other (via registry conflicts, memory conflicts, shared library incompatibilities, etc.), they can still work together as normal. As an example, features such as cutting and pasting, object embedding, object linking, and so on, behave exactly the same. The big difference, of course, is a significant reduction in support costs, as issues caused by the complexity of the user PC environment are at the root of so many helpdesk calls, particularly when users are installing software and changing the machine configuration themselves. With this in mind, some solutions go one step further and allow applications and data on the mobile machine to be “cleaned” when required, removing undesirable objects according to policy; an approach with great appeal from a security perspective.

The application streaming approach thus has many of the same operational and support benefits of thin client computing, but also caters for the needs of mobile workers. It’s still relatively new, however, so it is important to check out the current capability of specific offerings.

Other forms of virtualisation

The forms of virtualisation described above are those that are either most commonly deployed or most commonly discussed by IT vendors. When you start looking in detail, you will discover that there are very many variants on all of the basic themes we have discussed.

Beyond this, an area we haven’t dwelled on here is ‘client partitioning’, in which a desktop or laptop machine is set up to run multiple operating system instances. This is an approach predominantly used by two constituencies at the moment – IT professionals who want the convenience of being able to develop and test in multiple environments on the same machine, and Apple users who have a need to run Windows applications on their Mac. Client partitioning, however, is a rapidly evolving area, and we can expect some interesting solutions to emerge over time to deal with the perennial problem of managing the conflict between personal and business computing taking place on the same physical device, which many organisations are now grappling with.

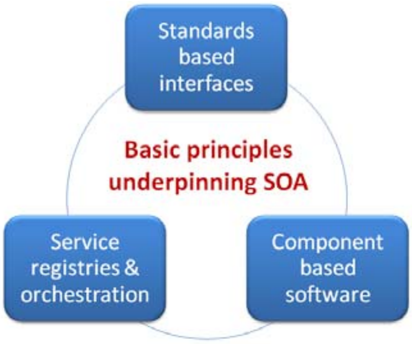
Service orientated architecture (SOA)

Turning our attention to the application software layer, one of the most confusing areas of IT at the moment for many is the meaning and significance of ‘service oriented architecture’ – commonly referred to as ‘SOA’.

On the one hand, we have deep thinking academics, ideological industry analysts, and high-powered architects all discussing the theory of service orientation in a very purist manner. If you take your lead from these constituencies, you could easily end up with the impression that SOA is something that requires a complete upheaval of the way IT is done, and that the only way to realise the benefit is if you become totally committed to a top down all-encompassing business-oriented initiative. All potentially very scary. We then have IT vendors, who each ‘bend’ the description of SOA and what’s most important about it according to what they are trying to sell you. Listen to half a dozen sales pitches, and you are likely to end up with the same number of SOA definitions. No wonder, then, that many IT professionals working in medium sized organisations conclude that SOA is either not relevant or not practical in their environment.

The reality is that SOA, as we shall see, is a common sense way of constructing software that has benefits in terms of both efficiency and flexibility, and essentially represents an unstoppable trend. Even if you do not explicitly elect to ‘adopt SOA’, you will find software creeping into your infrastructure based on this model from application vendors who have re-engineered their packages to take advantage of service orientation. And in the context of this paper, whichever way you look at it, SOA is an implicit part of the dynamic IT approach, so it is worth getting to grips with.

Rather than being drawn into the debate about definitions and what constitutes 'proper' SOA, however, let's take a look at some of the basic principles that underpin the approach and consider their practical relevance.


Standards based interfaces
Basic principles underpinning SOA
Service registries & orchestration
Component based software

Standards based interfaces

An important development in recent years that directly addresses the cost and flexibility constraints represented by proprietary APIs is the emergence of standards-based interfaces. Indeed, a significant turning point occurred when all of the major vendors in the industry got together, some might say quite uncharacteristically, and agreed to collaborate on standards around 'web services'. This was an acknowledgement of how important it was to remove the friction associated with application integration across a network, either internally or over the internet.

While it is beyond the scope of this paper to discuss specific standards (though we recommend looking up some of the basic ones such as XML, WSDL and SOAP if you're not already familiar with them), suffice it to say that the key to success was agreement on one important principle. This was that each application in a web services environment would declare a range of 'services', i.e. functions it would be willing to perform for another application making a request. Furthermore, services would be declared in a 'self describing' manner, meaning that other applications, registry systems, and so on could 'discover' not only the range of services on offer, but also how to invoke them and interpret what they return.

SOME BASIC STANDARDS TO BE AWARE OF

XML: eXtensible Markup Language
Generic markup language used as the foundation for building and declaring services

WSDL: Web Services Description Language
Defines the way in which XML is used for describing services and service related messages

SOAP: Simple Object Access Protocol
Defines the way in which XML is used to exchange structured information between applications

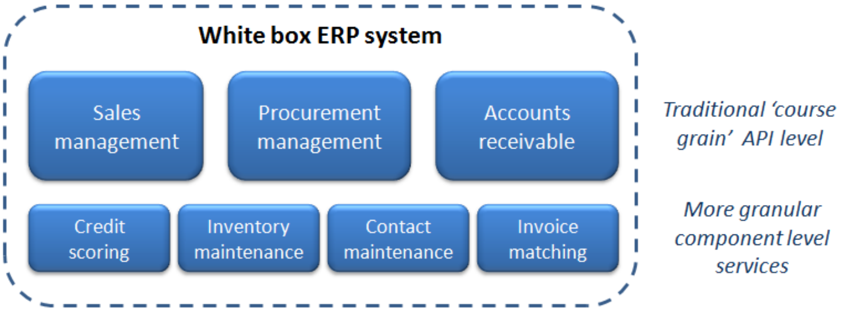
One of the advantages of the approach is that the ways in which services are fulfilled are hidden behind the standard interfaces. This means things can change behind the scenes, such as the manner in which data is stored and retrieved or the logic used to fulfil a request, without the application calling upon those services requiring modification. Such 'decoupling' of applications can significantly reduce hardwired dependencies and therefore greatly increase flexibility and responsiveness.

In practice, of course, it isn't all just out of the box plug and play, and work still typically needs to be done to ensure services are aligned at a business logic level, but this is far easier than building interfaces from technical first principles, and once a set of services is declared and understood, initial work can be reused to deal with other requirements. The other consideration is the dynamic nature of the standards themselves. While those concerned with basic interactions between applications are now pretty settled (such as those mentioned above), other standards that deal with requirements such as advanced security and transaction management are still evolving. This does not, however, detract from the huge leaps forward that have already been made.

Component based software

While standards based interfaces gets us a long way forward, SOA specialists would argue that supporting this approach is not in itself enough to qualify an application to carry the label 'SOA', and they would be right. A common practice to ease the integration of older applications, for example, is to 'wrap' them in a web services layer. What this essentially means is creating an integration wrapper which maps web services onto proprietary API calls so other applications can make requests in a standard manner. Behind the scenes, the original application typically retains its traditional monolithic black box nature. It's an approach that many corporate IT departments have adopted to open up mainframe installations, and it has also been used over the past several years by packaged application vendors as a first step towards allowing genuinely open integration.

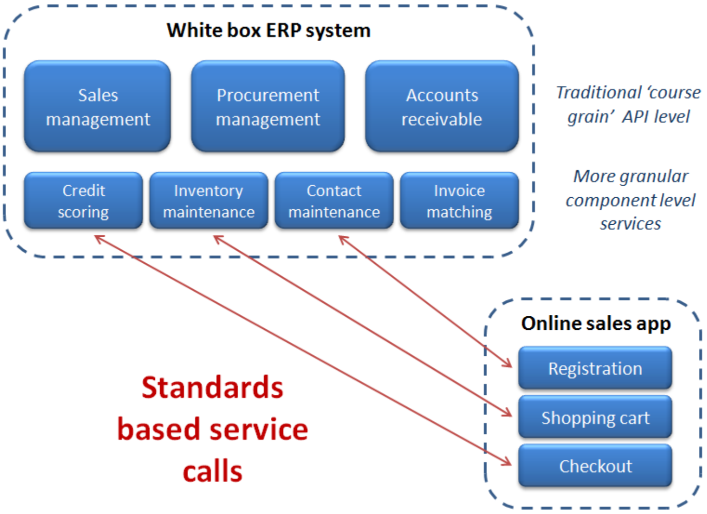
The 'A' in SOA, however, stands for architecture, and one aspect of this is the way applications are engineered. The overwhelming trend in this respect among software vendors and within the development community in general is to move towards a more component-based 'white box' approach. The idea is that application functions are grouped and implemented as smaller semi-autonomous units that are designed to communicate with each through standards-based interfaces rather than proprietary connections that were never meant to be externalised.



As an example, an Enterprise Resource Planning (ERP) system might be broken down into a range of components that between them serve the needs of the application itself, but also declare a range of services externally allowing other applications easy (but properly controlled) access to functions and data. Examples here might include customer record querying and maintenance, or functions concerned with invoicing, credit checking, stock control, and so on.

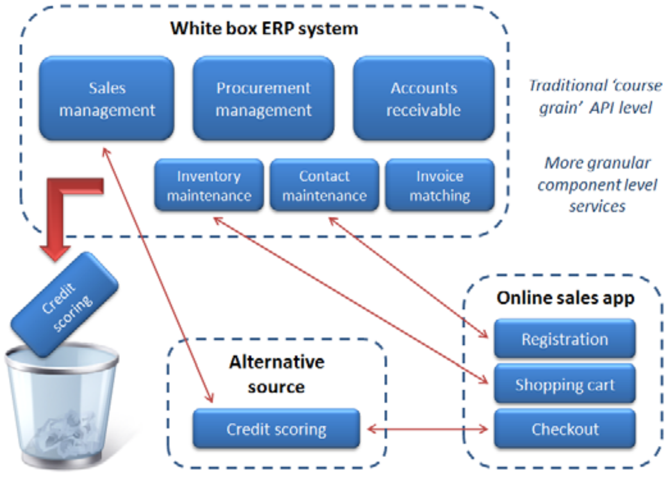
There are several benefits of this component based approach, so let's walk through them.

Firstly, the main application itself becomes inherently more flexible. Unlike older monolithic applications where everything is hard-wired together, it is possible to modify or upgrade one component with far less risk of disrupting the entire system. If it's a software package we are talking about, this potentially provides vendors with the ability to roll out smaller incremental changes, minimising the need for major upgrades that customers have learned to dread over the years. If it's an in-house or custom built system, the same basic principle translates to less resource, cost, risk and time to implement modifications.



Another significant benefit is the ease with which existing functionality may be reused during the development process. Implementation of a new online store application, for example, might call upon the functions already available within the ERP system through standard interfaces. This is clearly going to be quicker and more cost effective than either reinventing the wheel, e.g. by implementing yet another credit scoring system, or handcrafting integration against traditional proprietary APIs or via messy import/export workarounds.

Again, the end result is a faster response to new business requirements, as well as reducing ongoing maintenance and support costs down the line and improving the consistency with which business is done. The latter is achieved by avoiding the need to duplicate data and functionality at the risk of things drifting out of sync over time.



Flexibility from the component based approach also comes in another guise. Because the individual parts of the system are decoupled in the way we have described, substituting one component for another component, or one service for another service, adds yet another dimension to the overall flexibility equation.

If we stick with our ERP and online sales example, one of the moves we might make is to take advantage of an alternative credit scoring service. This might be from another in house system, e.g. a new performance management application brought in with advanced credit management functionality, but equally, it might be a credit scoring service provided by an external bureau.

Either way, provided the service does the equivalent job of the old one that was embedded in the ERP system, everything should continue working as usual. The fact that the alternative functionality now driving the credit scoring process is using superior data or methods behind the scenes, and is thus delivering more robust and accurate results, should be transparent to both the remaining parts of the ERP system and the online sales application.

The use of external services in this way is something we shall be picking up on shortly when we discuss software as a service (SaaS) and cloud computing. To complete our discussion of SOA principles, however, we need to consider how the publication, discovery and execution of services are managed.

Service registries and orchestration

Beyond the way applications are constructed, or perhaps we should say 'how services are implemented', the other main architectural aspect of SOA is how everything fits together. In the examples given so far, we have simply talked about calling or invoking services when they are required. But how is this achieved?

Well the simplest method is direct 'point-to-point' invocation. With modern packaged applications that support web services the vendor usually provides service definitions in XML format (based on the aforementioned standards), often with supporting documentation. With each service having an address on the network, a developer building or integrating another application can then use this information to code the necessary calls. This is really no different to any other type of remote procedure call, except that there is no need to worry about the mechanics at a technology level. Of course if you are building your own applications (or commissioning an integrator to do this for you) then the inclusion of the relevant service declarations would ideally be part of the specification. Either way, in a smaller scale business environment, direct point to point access will probably be adequate to solve many business and system requirements.

As systems and the level of integration between them becomes more sophisticated and dynamic, however, it is useful to introduce a degree of automation to control the way in which services are managed and accessed. This can be achieved through so called service 'registries' and/or SOA middleware. It is beyond the scope of this paper to go into a great deal of detail here, but it is useful to at least review the types of facilities to look out for.

In terms of registries, as the name suggests, these are basically repositories which contain lists of services, along with their purpose, owner and location. Registries can be either external or internal and the idea is that when a suitable service is sought to meet a need, the registry is interrogated and returns the necessary details. There have been attempts to create public registries based on something called the UDDI (Universal Description, Discovery and Integration) specification, though these have not been particularly successful. The original idea here was to collate everything in one place, from generic internet based services to services externalised on a privileged basis by organisations for use by their customers, suppliers, etc. in a supply chain context. As of today, however, registries have overwhelmingly been implemented on a private basis for either purely internal or very controlled external use. Nevertheless, registries have an important part to play in creating visibility of services to allow effective integration and exploitation.

Turning to SOA middleware, most modern application servers are web services aware, so the basic mechanics are supported pretty much as standard by both commercial and open source offerings. Beyond this, we have the ‘enterprise service bus’ (ESB) concept, which in its basic form manages the routing of service requests according to a pre-defined set of rules, with more advanced solutions also dealing with full ‘orchestration’ of the flow of activity and transactions between services. The best way to think of the latter is controlling the sequence in which services from different sources are involved when a business process spans multiple applications. By defining the rules, events and workflows centrally, and with the inherent decoupling represented by the SOA approach, this kind of setup can provide a great deal of flexibility to modify or optimise the behaviour of IT systems, almost on a continual basis, without the need for excessive technical development or integration work.

As solutions have matured in this area, particularly from some of the larger IT vendors, the line between SOA orchestration middleware and ‘business process management’ (BPM) suites, with their roots in traditional workflow, has become quite blurred. The area is also riddled with jargon, its own set of standards (e.g. WSCI, BPEL, BPML), and a range of often conflicting emerging ideas on how things should be done. If you are not familiar with this area, it is therefore worth enlisting the help of a trusted IT vendor or professional services firm to work through the specifics with you.

But is it SOA?

As a word of warning, many of the aforementioned purists would read the above and question whether our description of SOA principles is accurate and complete. They might argue, for example, that we have placed far too much emphasis on web services standards and point out that SOA, strictly speaking, can be implemented based on any messaging mechanism. We make no apologies for focusing on the more commonly encountered pragmatic approaches, however, and would urge you yourself not to get drawn into the debate on the theory or get too hung up about whether what you are doing can be considered SOA ‘proper’. It really doesn’t matter that much if you are realising some of the benefits of the underlying principles and mechanisms we have gone through.

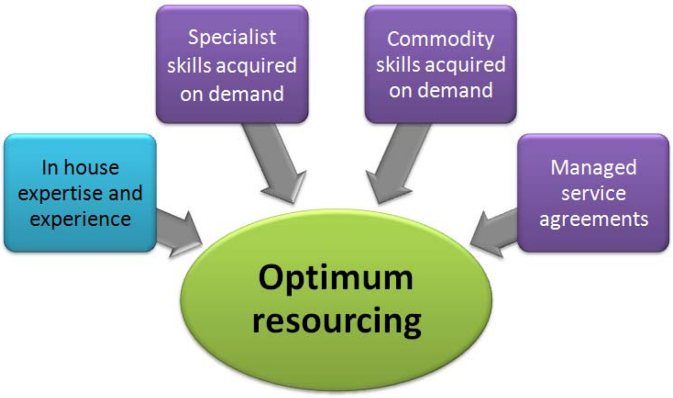
The one thing that is worth adding is that much of the commentary around SOA highlights the IT-business alignment dimension, and many would argue, in fact, that allowing IT to tune into business needs and requirements more effectively is the primary purpose of the SOA approach. The premise behind this is that if you dissect most business processes, the activities that make them up can often be mapped directly onto key IT capabilities that are required to support them, which in turn can be described as services to be provided by IT systems. This has advantages in that it facilitates an objective dialogue between the business and IT on everything from requirements and priorities through to service level expectations on performance, scalability, resilience, etc.. For a further discussion of this, please see our *Further Reading* list at the end of this paper. In the meantime, it is important not to lose sight of the tangible systems level benefits we have discussed.

Blended resourcing

On a few occasions so far, we have hinted at the role of external resources and services, and as we look to create a more dynamic, responsive and efficient IT environment, this is an important part of the equation. But if you think ‘outsourcing’ is a dirty word, don’t worry, we are not about to advocate handing over the running of your IT services completely to a third party (even though some might think this is a good idea). What we do recommend, however, is to blend the right level of external services into your IT activities and infrastructure, so let’s take a look at what we mean by this.

Manpower, skills and experience

In a smaller business environment in particular, there is a limit to the range of skills that can be maintained in-house cost effectively. It may be, for example, that you need access to specialist experience for a short period of time to implement a desktop virtualisation solution or install and configure some SOA middleware. Recruiting a specialist onto the payroll or even cross training an existing member of staff to obtain skills that are only going to be used for few days or weeks is going to be hard to justify in economic terms. There is then the issue of retaining personnel with under-utilised specialist skills that others might be willing to pay a premium for – a recipe for undesired staff turnover, with all of the cost and disruption that comes with that. Yet if you attempt to muddle through without the necessary expertise, there is a risk of creating issues and failing to realise the true potential of IT related investments, as well as an increased likelihood of service interruptions and security exposure. And, of course, things invariably take more time to deliver, bringing us back to the IT responsiveness issue that dynamic IT is trying to get us away from. Interestingly, we can also look at the resourcing question the other way around. While in-house IT staff may not always have the in-depth specialist skills we were referring to above, they typically have critical competencies of a different kind, and a familiarity with the local IT and business environment, both of which are key to effective service delivery. Another form of false economy is tying up such valuable personnel on mundane activities, whether operational or development in nature, that don't make full use of their skills and experience. And while they are occupied in this way, they are obviously not being used in a manner that optimises their contribution to the business.


Specialist skills acquired on demand
Commodity skills acquired on demand
In house expertise and experience
Managed service agreements
Optimum resourcing

The answer is to adopt a blended approach to resourcing manpower requirements, which might go against the grain culturally in some situations, but as we look at the rate of change on both the technology and business sides of the equation, is probably going to be necessary to achieve the kinds of responsiveness and efficiencies we would ideally like to see looking ahead.

From a practical perspective, if you haven't done so already, a good place to start is to review the competencies that exist in-house and the kinds of external resources you are using at the moment against the range of activities and priorities that fall within the IT domain. For project related activity, it is then a case of identifying areas where it makes sense to acquire either specialist expertise or commodity skills (e.g. routine coding) on a short term contract, or even to take advantage of managed service offerings on an ongoing basis (e.g. for desktop support, break/fix maintenance and so on). It's impossible to generalise on what will make sense for any given organisation, but the basic principle of blended resourcing in general in this area does have broad applicability.

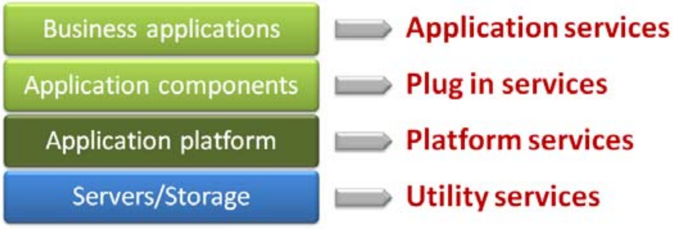
Hosted services

The other resources to which a blended approach can be applied are those that make up the IT systems themselves, and this is where hosted service offerings come into play. The basic premise is that it may not make sense for all of the IT infrastructure required to support the organisation's business requirements to be resident in house, or 'on premise', as some would say.

The most widely accepted form of hosted services at the moment in a smaller business environment is web hosting. While some are running web servers on premise, it is very common practice to let an internet service provider (ISP) take care of things for you in this area, and in many cases, this service extends to include email service provision as well.

The other form of hosted services that is likely to be familiar is when a solution provider takes ownership or control of one of your systems and runs it for you on their premises. This again is a practice that has been going on for years in the smaller business space, and can have advantages in terms of both reducing the internal IT burden and benefiting from the provider's economies of scale. A reputable hoster, for example, can better afford state of the art, highly fault tolerant facilities and the maintenance of specialist skills on the payroll when their investment in these areas pays back across multiple customer contracts. Such bespoke or custom hosting arrangements are therefore still as valid today as they have always been.

Developments in recent years, however, have led to the emergence of a variety of what can best be described as 'on demand' hosting options. The basic idea is that you pay for computing resources and/or functionality based on what you use (or consume) rather than what you own. If it's an application, for example, such a CRM system, you don't buy any software or equipment, you simply pay to access the service provider's system, typically on a fee per user per month basis. This is the model that gave rise to the term 'Software as a Service', 'SaaS' for short.



At the other extreme, if you need access to some server horsepower for a period, e.g. to run a one-off heavy processing job, you might rent some power from a service provider that makes raw computing resources available. This is a model that originally gave rise to the term 'utility computing'. In between these two, we have seen services emerge that fulfil requirements at different levels in the IT stack, ranging from application plug-ins (equivalent to component level services in the previously discussed SOA model) to application platform services, providing everything a developer might need to build and execute custom software solutions – tools, middleware, directory services, security, and so on.

While using the traditional stack as a reference point can help to understand where individual services might fit into your activities, a full review of what's now available reveals quite a few different on-demand service categories that are useful to be aware of:

ON-DEMAND SERVICE CATEGORIES	
Business application services	It is in this area that the term 'Software as a Service' (SaaS) was originally coined. Services at this level are typically focused on the delivery of complete business functionality, e.g. CRM, ERP, etc.
Hosted productivity tools	Services here are more concerned with horizontal capability ranging from desktop suites for end users, through to modelling, development and project management tools for analysts and developers.
Hosted comms/collaboration	Spearheaded initially by hosted email and web conferencing, the number of services offerings in this area has exploded to include full unified communications and/or social media (directories, blogs, wikis, etc).
Trading community services	As supply chain automation has gathered momentum in some industry sectors, services have emerged aimed at facilitating the way in which customers and suppliers collaborate and transact electronically.
Plug-in services	A myriad of services exist which do not provide complete business functionality but 'plug into' existing applications to enhance or extend them. Examples include everything from mapping to credit checking.
Application platform services	As an alternative to consuming pre-built services from external providers, application platform services provide development and runtime environments allowing custom applications to be built and hosted online.
Utility services	Sometimes, you have your software, but simply want somewhere for it to run and store its data. This is the realm of utility services, which are essentially about providing raw compute and storage resources.
Operational services	This often overlooked but highly important category is where we find services concerned with online backup, archiving, security (e.g. email filtering), etc., and even full blown monitoring and management tools.

There are two things we should highlight about the above table. Firstly, it is undoubtedly not exhaustive, as the area of on-demand hosted services is developing so rapidly at the moment. Secondly, the categories we have defined are deliberately descriptive and jargon free.

The reason for avoiding jargon is because at the time of writing, there is a lot of confusion in the industry about what is meant by certain terms. As an example, some use 'cloud computing' as an umbrella term to refer to all of the above described categories, while others associate it purely with what we have referred to above as utility services. Yet more go on to talk about 'private clouds' based on the automated server pooling concept we went through in our virtualisation discussion.

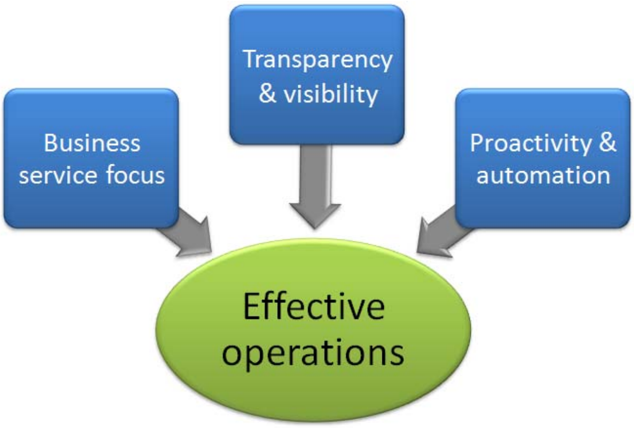
The upshot is that whole conversations and debates can (and often do) take place about cloud computing in which participants are talking about quite different things without even realising it. With similar ambiguity around acronyms like SaaS, which again is used to describe any or all of the above, we thought it best to provide a neutral framework against which you can compare various offerings to understand where they might fit into your environment.

And coming back to the theme of blended resourcing, the consideration of fit is highly important. While some on-demand services can be adopted on a standalone basis, e.g. sales force automation in the CRM space, most will need to work with your existing IT infrastructure and software landscape in one way or another – indeed they may have been designed to do so as part of a 'hybrid' architecture that integrates external services with in-house capability. The significance of our discussion around standards based interfaces and component based software will hopefully be even clearer now as we consider the use of on demand hosted services. But to stress what we said before, you don't need some big SOA or cloud computing initiative to take advantage of developments in this area, as the applications, tools and middleware you use in the natural course of things will introduce the relevant capability and 'hooks' over time if they haven't done so already.

As things continue to develop and mature in the hosted on-demand services arena, more opportunities will open up for IT departments of all sizes to increase their responsiveness to the business by grabbing and integrating the relevant external services rather than building or buying systems in the traditional manner. In adopting a blended approach, IT professionals are also able to pass off some of the operations related burdens and headaches to service providers, which is always welcome in the typically over-stretched medium business environment. Which leads us nicely to the last key dynamic IT enabling concept we need to cover.

Service oriented operations and management

Operating and managing things in a dynamic IT environment that embraces some of the enablers and principles covered in this paper requires a bit of an adjustment in thinking and approach. As we move away from the traditional hardwired systems stack and enable things to be changed more quickly and dynamically, things, well, change more quickly and dynamically. The methods we have traditionally used to take care of static systems therefore need to be adjusted, and in order to understand how, let's look at the three main ingredients of effective dynamic IT operations:



Business services focus

When software has been decoupled from hardware and SOA has disrupted the boundaries between applications, and even between internal and external capability, it becomes more difficult to define what we mean by 'a system', which in turn challenges the whole notion of 'systems management' and especially 'change management. The best way around this is to just look at IT from the other direction – 'business in' rather than 'infrastructure out'. This leads us to what many refer to as 'business service management' (BSM), a term often used by high-powered consultants who surround it with mystique and lofty goals to do with 'IT-business alignment'. BSM, though, is simply about changing your perspective, e.g. from asking “is this system running as it should?” to “are users receiving the service they require?”

Making this shift introduces another form of decoupling. In this case we are separating the ‘what’ is delivered, i.e. the service or capability, from the ‘how’ of delivery, e.g. the way in which IT does things behind the scenes in terms of how hardware and software is allocated and configured, how external services are blended into the mix, and so on. The business alignment benefit then stems from the fact that it helps with the objective discussion with the business about what matters, just as we discussed in relation to SOA. The difference in the management context is that we can extend this to consider things like service level expectations. Whether you take this to the extent of defining formal ‘service level agreements’ (SLAs) around business services depends on the culture you are operating in, but either way, it is useful to distinguish, for example, between genuinely critical services and those for which a more tolerant view can be taken.

The concept of business services is one that is integral to major best practice frameworks for IT delivery such as ITIL and COBIT. While the formality of these for smaller businesses is considered to be overkill by many, the good news is that their strong adoption in the large corporate and public sectors has meant that management tools vendors have encapsulated the ‘spirit’ of BSM in their products. You will therefore probably find facilities and templates within the latest versions of tools that can be used as a guide for defining the business services that matter and mapping underlying IT infrastructure onto them. This is the first step towards creating transparency and visibility.

Transparency and visibility

One of objectives of dynamic IT is to allow flexibility in the way IT assets are deployed, and indeed redeployed as requirements change. The challenge that comes with this is keeping track of things – which applications are being driven by which servers, which software licences are associated with which application instances, which storage devices contain which data, and so on. Ideally this should be coupled with information regarding who is using the service and the business function it supports. Traditional asset management assumes a certain degree of persistency in where things are and what they are being used for. In a more dynamic setup, however, such assumptions cannot be made. The upshot is that management tools need to be more fluid in the way they allow components to be mapped and tracked, and automated ‘discovery’ and ‘interrogation’ (to identify assets on the network and how they are configured) become an important part of the mix.

Again, we are fortunate in that small-footprint solutions are now available for medium sized businesses that can allow such capability to be obtained and commissioned cost effectively. If used properly, these can provide visibility into current configurations and dependencies that can be used as the basis for planning, change management, service performance monitoring, utilisation assessment, maintenance, and support activities. In addition, transparency and visibility is important for that often forgotten area of software licence management. The unfortunate reality is that some software vendors haven’t yet updated their commercial models and legal terms to match the flexibility that modern virtualisation capability permits. Tracking software asset deployment and/or usage is therefore important for compliance. And, of course, it is good practice anyway. It is not uncommon for organisations to discover after implementing software asset tracking that they have unused licences and/or subscriptions that are being paid for unnecessarily. It should also be recognised that IT can exploit the information available through such transparency and visibility’ solutions to communicate more effectively with business users, thereby enhancing the perceived value delivered to the organisation, still today one of IT’s major challenges.

Proactivity and automation

A direction in which operational best practice is moving in general is towards more proactivity and automation. The premise for this is that the traditional monitoring centric approach, in which the role of management tools is primarily to keep tabs on performance and health then flag up exceptions for manual handling, relies too much on human intervention for tasks which could in theory be automated. And the motivation for automation, in line with the fundamental objectives of dynamic IT in general, is to drive both efficiency and responsiveness whilst frequently reducing errors likely to result from manual operator intervention, still a significant factor in service interruption statistics.

While the days of complete ‘lights out’ operation in which systems literally take care of themselves are not with us yet (and would probably make many IT professionals uncomfortable anyway), there is a middle ground between this and having to do everything manually that can be achieved. The rule and policy based resource allocation and usage awareness that is an inherent part of virtualisation technologies plays to our hand here as does the evolution undertaken by many systems management solutions. Essentially, the infrastructure can be considered to be more ‘intelligent’, so not only can thresholds be monitored and exception situations watched for as the basis for alerting, but in many case the alert can be accompanied by a recommended action. Such recommendations can then be easily (and swiftly) implemented by the operator after a quick sanity check, but there is also the possibility of allowing responses to certain situations to be implemented automatically.

In practice, there is clearly a trade-off between response time and risk. On the one hand, if your BSM view of the world tells you that a service is critical, you are likely to want as much automation as possible to be applied to operating the infrastructure underpinning that service. That way, exceptional situations or undesired drift can be identified and corrected as necessary in a preventative manner, avoiding damaging performance issues and/or service interruptions. On the other hand, the last thing you need is an inappropriate auto-response leading to an inappropriate change that actually makes things worse or causes disruption in some other kind of way. The need for good operational skills and judgement, together with the BSM approach, is therefore the key to safety and success.

Steering the right course

From the scope of this paper it will be evident that there is a lot to consider when looking at options for removing some of the traditional constraints that have held IT professionals back from an efficiency and responsiveness perspective. Whether you are in the position of being under explicit pressure to improve performance, or simply interested in the ongoing process of making things better to ensure maximum value and service from IT, the big question is always which of the emerging ideas and technologies are most relevant to implement, how and when.

In this respect, it is important to consider the adoption of dynamic IT related principles and solutions as a general direction, rather than a journey with an end point. The reality is that even if you were to create an initiative to implement everything we have outlined in this paper, by the time you did it, the industry would have moved on and there would be yet more options available for taking things further. With this in mind, here are some tips on steering the right course:

- Ensure that you as IT have an effective, open and honest dialogue and working relationship with stakeholders in the business. In some situations there may be history or some ongoing circumstances creating disharmony, in which case this needs to be acknowledged and addressed as a first step. Unless an effective dialogue is created, moving forward effectively with a dynamic IT initiative will be significantly hampered.

- Spend time with business stakeholders developing an initial BSM view of the world. This need not be done exhaustively and in fact most people in a medium sized business environment find that they have a pretty good idea of what the critical services are before they start. Talking it through, however, is great for focusing minds on what really matters, and getting IT and the business on the same page in terms of priorities and objectives. This provides an important foundation for subsequent planning and review activity, and as a spin-off, can be a great enhancer of IT-business relations.

- Armed with the insight arising from the BSM related analysis, perform an honest appraisal of how well IT is delivering today against business objectives and the expectations of business stakeholders. Some areas of both good and poor performance will already have been flagged up as part of the conversation with the business, but there will be other areas in which IT knows it can do better, even if the business is not explicitly complaining at the moment. As part of the process, it will be useful to use the analysis of common constraints laid out earlier in this paper as a reference to form a crystallised view of what’s holding things back the most.

- Based on the above, it is worth considering putting a short to medium term improvement plan in place to address the most significant areas of constraint or weakness. In the current economic climate, cost savings might form a focal point for prioritisation, but it is common to find that business people often appreciate improvements in responsiveness and alignment of IT activities much more. The ideas and solutions we have discussed that drive improvements in flexibility should therefore not be dismissed, even if they require some investment to implement.

- On a very specific point, if you haven't done so already, it is worth looking at your hardware infrastructure and considering how you might take advantage of server partitioning and storage virtualisation in particular. The reason for saying this is because virtualisation projects in these areas tend to payback very quickly in terms of cost savings (both capital and operational), as well as reducing the burden on IT (thereby freeing up resource), and increasing flexibility and responsiveness. It is no coincidence that one of the most popular types of project in the industry at the moment is server consolidation, and some organisations have been able to achieve an order of magnitude reduction in the number of boxes they need to manage and maintain in their computer rooms, which is a relatively quick win, whichever way you look at it.

- Your approach to taking SOA on board will largely depend on the amount of custom development work you undertake or commission. If you spend a lot of time building and integrating systems in house, it may be worth training your staff on SOA principles and tooling, and perhaps even investing in some of the SOA middleware we were talking through earlier. Finding a good supplier with both the right track record and supporting attitude is worth spending some time on here. And on the subject of suppliers, when engaging with packaged application vendors, make sure you grill them on their commitment to open standards and component based architectures. For new acquisitions, it makes sense to prioritise these as selection criteria. For packages you are already using, the increased flexibility represented by SOA enablement of the latest releases may encourage you to upgrade sooner rather than later.

- Picking up on this last point, while it is something we have not covered as part of this paper, it is worth being aware that SOA enabled packages and other applications can have significant benefits from a user empowerment perspective. Having a library of robust services that can be presented as widgets and add-ins for use in intranet portals and desktop tools can allow users to solve many of their own requirements through dragging and dropping from a palette or clicking on a few options on a menu.

- The degree to which you explore alternative resourcing options, either professional services or on-demand hosted services, will be very much dependent on your environment and how IT operates. We would encourage you to at least consider where these might fit, however, as there is very clear evidence that open-mindedness on the sourcing front correlates strongly with successful IT service delivery. For a fuller discussion of this, and thoughts on the pros and cons of various types of approach, please see our report entitled *"IT on the front foot"* listed in the *Further Reading* section. The main caveat with regard to on-demand hosted services is to beware of the temptation to shortcut due diligence from a security, integration and lock-in perspective because signing up and getting going is so seductively easy. There shouldn't be any issues with the right provider, but it is imperative that you do your homework.

- At some point in your plan, you are likely to need to confront the traditional 'systems ownership' mindset. While we appreciate that changing the way in which investments are appraised and costs are accounted for can't change overnight, it's important to at least come to an understanding with the business about the need for flexibility on asset deployment and allocation. Whatever arrangement you arrive at to make this happen is secondary to the requirement for IT to have the freedom to move resources around when necessary to ensure optimum service delivery.

- One last tip based on looking at the flexibility question from a slightly different angle is to make sure you explore alternative funding options when investments are required. Options are now available from major IT vendors and their partners, plus independent financing firms who specialise in IT related matters, to work around capital constraints via arrangements ranging from traditional leasing to full project financing, including hardware, software and professional services. In an uncertain economic climate, this can really help to keep improvements flowing.

While not exhaustive, we hope these suggestions provide some useful pointers for moving forward in a practical, achievable and beneficial manner.

Further Reading

The following Freeform Dynamics reports and papers expand on some of the areas touched upon in this document, and are all freely downloadable via the links provided.

SOA and Growth Markets
A review of best practice and adoption reality
http://www.freeformdynamics.com/fullarticle.asp?aid=674

Taking stock of the IT environment
Practical steps towards infrastructure optimisation
http://www.freeformdynamics.com/fullarticle.asp?aid=599

IT Delivery in the Downturn
Responding appropriately to economic pressures
http://www.freeformdynamics.com/fullarticle.asp?aid=524

Service Orientation in Business
Harnessing change from the board room to the data centre
http://www.freeformdynamics.com/fullarticle.asp?aid=632

IT on the front foot
Sourcing, architecture and the progressive IT organisation
http://www.freeformdynamics.com/fullarticle.asp?aid=318

The Great Virtualization Debate
Practitioner insights into the where, why and how
http://www.freeformdynamics.com/fullarticle.asp?aid=139

IT Management Checkpoint
The Next 5 Years
http://www.freeformdynamics.com/fullarticle.asp?aid=129

Deploying CMDB Technology
Pragmatism and realism will deliver the benefits
http://www.freeformdynamics.com/fullarticle.asp?aid=113

Service Oriented Architecture (SOA)
Insights from the front line
http://www.freeformdynamics.com/fullarticle.asp?aid=50

Please see www.freeformdynamics.com for a wide range of other papers, reports and research notes on the effective use of IT in a business context.

About Freeform Dynamics

Freeform Dynamics is a research and analysis firm. We track and report on the business impact of developments in the IT and communications sectors.

As part of this, we use an innovative research methodology to gather feedback directly from those involved in ITC strategy, planning, procurement and implementation. Our output is therefore grounded in real-world practicality for use by mainstream IT professionals.

For further information or to subscribe to the Freeform Dynamics free research service, please visit www.freeformdynamics.com or contact us via info@freeformdynamics.com.

About IBM

At IBM, we strive to lead in the invention, development and manufacture of the industry's most advanced information technologies, including computer systems, software, storage systems and microelectronics.

We translate these advanced technologies into value for our customers through our professional solutions, services and consulting businesses worldwide.

For more information on IBM, please visit www.ibm.com.

Terms of Use

This report is provided for your general information and use only. Neither Freeform Dynamics Ltd nor any third parties provide any warranty or guarantee as to the suitability of the information provided within it for any particular purpose.