White Paper

August 2010 v.3

Arjuna Agility™:

Removing the Barriers to Business Agility

Moving the enterprise
from the silo to
the clouds

Contents

Executive Summary

In the future, competitive success will belong to organisations that are able to cope with rapid external and internal changes – organisations that can demonstrate 'Business Agility'.

Business Agility: *"An organisation's ability to sense environmental change and respond efficiently and effectively to that change."* Gartner

Whilst it is widely recognised that IT has a key role to play in enabling the 'sense' and 'responding' components of business agility it is also clear that IT can hinder agility through monolithic architectures and siloed applications and information.

Arjuna's product, Arjuna Agility™ (*Agility*) enables business agility by creating an 'internal cloud' of computing resource within the enterprise. In particular, *Agility*:

- De-risks the enterprise from over and under capacity problems by dynamically provisioning IT resources to meet changing business demand
- Improves utilisation of existing IT by supporting the policy-controlled sharing of all IT resources (from bare metal to applications)
- Bridges the semantic gap between the way business people specify service requirements and the way IT departments interpret them by using an innovative Service Description Language
- Supports strong corporate governance by using Service Agreements to create a full audit trail of activities which may be useful for regulatory compliance (e.g. Sarbanes-Oxley) as well as operational efficiency (e.g. tracking usage of software and hardware resources)
- Seamlessly extends the internal cloud across departmental or organisational boundaries if required

Agility provides these advantages without requiring existing IT infrastructure to be restructured or applications changed in any way. *Agility* maximizes the return on IT investment and, at the same time, provides the flexibility and speed of delivery that enterprise IT needs in order to effectively respond to business opportunities and threats when they arise.

In this increasingly turbulent market executives cannot rely on long-range forecasting and must instead create an organisation that can quickly respond to change. This white paper highlights the need for an agile IT infrastructure to support the move towards business agility. It describes how traditional IT infrastructures present a real barrier, illustrating the problems with a set of scenarios. The paper then describes in detail how these problems are solved by *Agility*, revisiting the scenarios to illustrate the business benefits.

Introduction

The concept of business agility arose in the early 90's from the observation that the increasing rate of change in the business environment was outpacing the ability of manufacturing organisations to keep up1. Previously dominant organisations were failing as they were unable to satisfy continually changing markets and customer

1 R. Dove, The Meaning of Life and the Meaning of Agility. Paradigm Shift International www.parshift.com/library.htm., 1997.

interests. Many organisations knew what they had to do but simply couldn't institute internal change quickly enough. *Agility* was the term used to describe the elusive characteristics necessary to keep pace with the changing environment.

Initially, 'lean manufacturing techniques' were used to introduce agility into the enterprise, eliminating waste, simplifying procedures and speeding up production. Many of these techniques have since been applied to service industries allowing improvements in operational effectiveness. As Porter2 has noted however, competing on operational effectiveness alone is not sufficient to gain competitive advantage. The rapid spread of best practice allows competitors to quickly imitate efficient techniques; competitors soon begin to look identical and a war of attrition can result in a race to the bottom. He concludes that "a company can outperform rivals only if it can establish a difference that it can preserve."

As the business environment becomes ever more dynamic and turbulent, establishing a position of difference requires the business to be agile. Microsoft3 describes the four main drivers for change as:

- Increased interconnectedness – the Internet connects both customers and the supply chain directly to the enterprise in real time through web interfaces, e-mail and mobile technology
- Economic and political changes – new regulations and political decisions in all parts of the world affect business
- Improved information collection and delivery – information technology has the potential to improve productivity through the provision of more and better information
- The bottom line – the market continues to demand profitability and in many cases improved profitability

In the future, gaining and sustaining competitive advantage will require a shift away from erecting barriers (vertical integration, proprietary technology and accepting high fixed costs to create scale, etc.) to overcoming or ignoring barriers (through outsourcing, alliances, adaptable technology and the elimination of high fixed costs). The strategic focus will be on changing the rules of the game to destabilize entrenched players. The enterprise therefore must be able to sense threats and opportunities and respond to them quickly – it must become agile.

The purpose of this white paper is to highlight the need for an agile IT infrastructure to support the drive towards business agility. The paper will describe how traditional IT infrastructures present a real barrier to effecting business agility and illustrate the problems with a set of real-world scenarios. It will then describe how these problems are being solved by current work at Arjuna, revisiting the scenarios to illustrate the business benefits.

IT: a Barrier to Business Agility

Business Agility applies to both strategic and operational issues. Enterprises need to be agile to handle strategic issues such as competitor moves or changing customer requirements. They also need agility to handle the operational issues which will result from their strategy. This can mean offering a wider variety of products and services, increasing customisation, developing new products and services, axing old ones, and coping with

2 Porter, M. (1996). 'What is Strategy', Harvard Business Review Nov-Dec, 61-78

3 Parthasarathy, S. (2005), 'Ask the Expert. Business Agility: Responding to Change', Retrieved from http://www.unisys.com/services/insights/outside__opinions/opinions.htm?insightsID=49847

fluctuating volumes. A business must be able to make these changes quickly and at low cost; it must therefore design agility into its operations. In this respect, agility is a combination of the basic operational performance objectives of quality, speed, dependability, flexibility and cost4 – where cost can be reduced by making improvements in any of the other objectives.

Being agile requires the people, processes, strategy and technology of an enterprise to be inextricably linked so that there is a continual, dynamic response to change. Building this agility into the enterprise requires real investment in time, technology and organisational change. The cultural issues that accompany organisational change require long-term commitment and investment in employee training and incentives to achieve results. Yet the benefit of business flexibility arising from reorganisation is often limited by IT. Whilst it is widely recognised that IT has a key role to play in enabling the 'sense' and 'responding' components of business agility5 it is also clear that IT can hinder agility through monolithic architectures and siloed applications and information6. If IT is to support dynamic business systems where change, information flows and decision making are continuous, an agile IT infrastructure is needed.

> *"Business agility and virtualisation will become the primary measures of datacenter excellence by 2012."* Gartner

The Problem of Silos

Traditional IT environments have typically been configured as silos in which resources are aligned around an application or business function. Such environments are characterised by silos of IT resources – a tightly-coupled package of application, data and hardware. Over time, whether through organic growth or though merger and acquisition, much of the organisation's IT infrastructure will be made up of a set of these independent silos.

This multi-silo approach has some advantages. Firstly, related applications can have their own tailored environment with the most appropriate hardware, operating system and support software. Secondly, departmental applications are kept apart and so do not interfere with each other – an increase in load on one application does not effect applications in other departments, nor can data locking by an application prevent others from making progress.

Unfortunately, the multi-silo approach raises serious problems when it comes to supporting an agile business. These can be seen when we consider the implications of silos on each of the performance objectives for an agile business introduced above.

Flexibility: The lack of flexibility to change the capacity of a service is perhaps the major problem with silos. When a new silo is deployed, a decision must be made about its capacity. This is difficult as it involves predicting the initial and future usage. In common cases, where there is a regular peak load that is significantly greater than the norm (e.g. at end of day, week or month), the silo must be sized to handle that peak. Many organisations err on the side of caution when predicting capacity for new services, particularly those that are "customer-facing", leading to even greater overcapacity. Even then, unforeseen spikes may occur, outstripping capacity and the system will deliver either a much degraded service, or even fail completely. To compound this problem, unpredictable spikes often reflect events that are of great significance to the enterprise's customers. In such cases the consequences of service failure will not only be lost revenue but also damage to the enterprise's reputation from which it may struggle to recover.

4 Slack, N., Chambers, S., Johnston, R. (2007) 'Operations Management' Prentice Hall, Harlow, England

5 Bradley, S.P. & Nolan, R.L. (1998), 'Sense & Respond: Capturing Value in the Network Era', Boston, MA: Harvard Business School Press.

6 Daniel, E.M. & Wilson, H.N. (2003), 'The role of dynamic capabilities in e-business transformation', European Journal of Information Systems, 12(4), p.282.

This traditional approach therefore builds in overcapacity for the normal predictable scenario yet remains totally inflexible if unpredictable spikes occur. The static nature of the silo prevents any spare capacity in other underutilised silos being appropriated to address the problem flexibly. In many cases this is resolved by throwing even more hardware at the problem yet this is very costly and highly inefficient and worst of all only works "until the next time."

Speed: If a new business product or service is required then this will require the provision of new IT resource within the silo. This limits business agility by introducing delays - it takes time to manually procure, deploy and configure new hardware and software. This problem becomes acute when organisations need to quickly deploy new products and services to cope with the increased pace of modern business. It is also avoidable, because other departments may have precisely the required resources lying idle.

Dependability: In a siloed IT infrastructure, each silo must manage its own dependability. The silos must be designed and configured such that the required quality of a business service can be delivered to customers even in the presence of hardware or software failure. A common technique for achieving this is replication – having spare systems ready to take over if a failure occurs. This adds cost, and the under-pressure CIO may again regret the fact that the siloed infrastructure requires having expensive but unused hardware "on-call" in each silo when a common pool of hardware on stand-by would be much more cost effective.

Quality: Customers judge the quality of a service not only by the results it produces, but also according to its performance and dependability. As we have seen, a siloed infrastructure can struggle to live up to expectations due to performance and dependability problems. A further problem is that quality expectations are not usually formally captured and agreed beforehand – infrastructure providers will do their best to meet demand, but this may not be enough to satisfy customers. Without any up-front agreement there is no way to ensure that sufficient resources will be provided to meet customers' quality requirements.

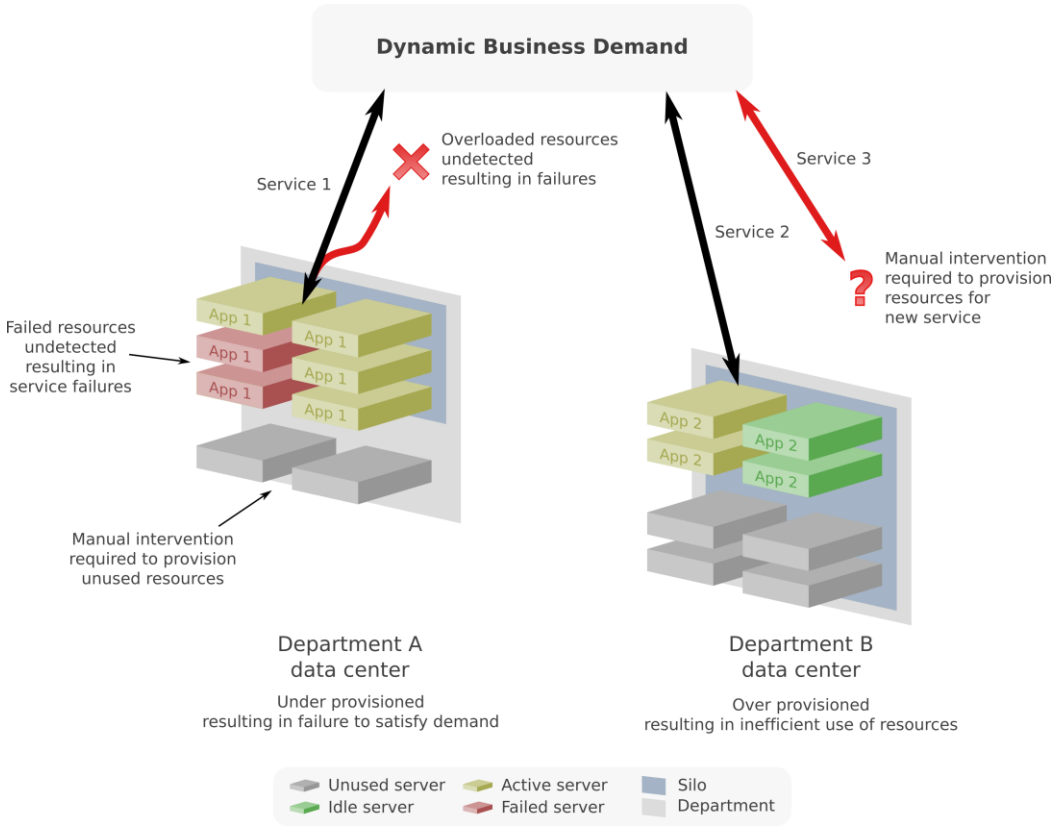

Figure 1. IT as a Barrier to Business Agility

Figure 1 illustrates how traditional siloed infrastructure fails to deal with dynamic and rapidly changing business demand. Resources are statically provisioned (within silos) to handle particular service demands. If the demand reaches levels which exceed the capacity of the provisioned silo then some of the demand will not be satisfied. If demand remains high, or if entirely new services are requested, then significant manual intervention is required before matching provision can be made. Additionally, if demand reduces the original provision can remain locked into the silo worthlessly absorbing valuable capacity.

Scenarios

We have described ways in which IT can act as a barrier to business agility. It is worth reinforcing this position through a set of scenarios that tell the story of two Trading Inc. employees - Joe, the IT Manager of the Trading department and Greg, the IT Manager of the Analysis department.

Under-utilisation of IT resource

Joe has previously argued for significant IT budget and, as his department's function is front-end and revenue-generating, he has persuaded management to over-provision in order to cope with peaks of demand.

As a consequence Joe has been able to maintain generally satisfactory service. Unfortunately, during the business day his servers operate at an average of only 10% and even under peak loads at around 50% of capacity. At night they are completely idle. Joe knows that he can't predict the future and that some day he may

well need the entire 100%, but on the other hand his high IT costs are being questioned by management – maintenance and power consumption costs are almost as expensive when machines are idling as when they are busy. Moreover, Joe is running out of physical space to house additional machines.

Over-utilisation of IT resource

The Analysis department handles back-office, largely jobs-based, processing (primarily in the form of analysis and reconciliation of the data collected during the previous day's trading) and Greg has been squeezed by recent IT cuts. As a consequence his IT infrastructure is frequently unable to cope with peak demands.

This means delays in the processing of jobs which cause the traders to constantly grumble about the performance of his team, and forces the team to put in extra hours in order to clear the backlogs. Morale is not high. Greg frequently complains that Joe should give up some of his under-utilised resource to assist his team.

Sudden and unexpected spike in demand for IT capacity

One day, following an extremely gloomy economic forecast which causes turmoil in the market, the load on the Trading department's IT infrastructure begins to rise dramatically.

Unfortunately, Joe is at lunch and by the time he is found applications have started to fail as they are unable to meet their real-time requirements. Each failure means lost revenue and damaged reputation for Trading Inc. Joe's boss is irate! Joe knows that if he can't solve the problem quickly then all applications will eventually fail completely. He gathers his team together to discuss which extra resources could be obtained. They approach other departments who begin to shut down some of their lower priority applications in order to free up servers. This takes time as those departments need to assess the current workloads and then wait for active work to complete. Once the servers are available Joe and his team begin to configure them. Amongst many other things, this requires the installation of the required application and its entire environment, which takes a considerable amount of time. By the time the team is ready to make the new resources available for the department, the crisis has passed. Unfortunately, for Trading Inc. many failures have already occurred and they have many dissatisfied customers.

Trading Inc.'s reputation has been badly damaged and Joe is held responsible.

Quality of Service degradation due to poor configuration

Joe has been managing the department's IT for some considerable time and he's noticed the pace of change accelerate over the years. New applications are deployed and decommissioned at increasing rates.

One day a business manager complains of degraded Quality of Service (QoS) from a particular application and Joe and his team have to try to track down the cause. It's hard to map QoS to the use of the underlying infrastructure but, after significant research, involving complex system monitoring and network probing, the team identifies the problem as poor configuration which they believe can be solved by co-locating two components which have close dependencies upon one another. Overnight, when the application is not in use, Joe’s team reconfigures the system.

In the morning Joe arrives to find the department in a panic. Another, seemingly unrelated application has been unable to meet its required QoS and high value orders are being lost! Joe‘s team are forced to bring down the entire system and restart it with the previous configuration. A week later, after further extensive research, Joe’s team works out that the application which failed had a dependency upon something, which had a dependency

upon something else, which had a dependency upon one of the components which was moved during the overnight reconfiguration. Joe and his team work out a new configuration and that night try again with their fingers crossed.

Catastrophic failure

One night an electrical fault causes a fire and the entire building is burnt to the ground.

The complete loss of the Analysis department hasn't been considered fully in the enterprise's planning and it takes many weeks for new infrastructure to be purchased and off-site backups restored. Unable to fulfill its orders the enterprise suffers a huge slump in its cash flow.

Arjuna Agility™: Removing the Barriers to Business Agility

Arjuna *Agility* is a 'federated cloud computing platform' designed to improve business agility through a more flexible infrastructure approach. Federated, because *Agility* is constructed from IT resources that are assigned by autonomous, cooperating, business parties within and beyond the enterprise.

Whilst trust and power are real organisational barriers to resource sharing, *Agility* provides a controlled way for the enterprise to organically grow its 'internal cloud'. Resource owners retain control by attaching policy to the assigned resource which describes the conditions under which it can be shared by *Agility*. Once assigned and subject to policy, *Agility* dynamically provisions the resource pool to meet the changing IT demands of the business.

Agility maximises the return on existing IT investment and provides the flexibility and speed of delivery that enterprises need in order to effectively respond to business threats and opportunities as they arise.

Features

Agility is realised as a set of services and tools which execute in parallel with existing IT infrastructure. Unlike other approaches, *Agility* does not force “big bang” changes onto the enterprise’s IT infrastructure and applications. Instead it can be introduced incrementally, as needed. *Agility*:

- Does not require existing middleware investments to be written-off
- Does not require existing IT infrastructure to be restructured or changed in any way
- Does not require resources to be shared, except (as described below) with the permission of the owner of the resources and under conditions explicitly specified by them

Agility supports controlled sharing of IT resources: IT administrators may add, or remove, some or all of the resources under their control to, or from, Agility. In doing so, the administrator defines his policy with respect to Agility’s use of those resources. The policy defines the conditions under which, and by whom, the resources may be utilised.

Agility virtualises all resources, from bare metal to applications. Further, any resource which can contain other resources (e.g. computer, operating system, application server) is modeled as a container. *Agility* can deploy virtualised resources (and any required containers) at any suitable location, and then manage them through a

common lifecycle. This does not impact upon any existing use of virtualisation within the enterprise, and extends the benefits of virtualisation to all resources within the enterprise.

Agility captures service requirements: *Agility* provides an innovative Service Description Language that bridges the semantic gap between the way business people (service users in a Service-Oriented infrastructure) specify their service requirements, and the way IT configures and deploys these solutions. The language allows service requirements to be expressed, focusing only on the concerns of the service user i.e. the type of service required and the quality of that service, without any direct tie-in to the underlying IT infrastructure.

Agility dynamically provisions: *Agility* matches IT resource to meet service requirements and changing demand. It achieves this with reference to the policies defined by the administrators who have assigned resource to it. Once provisioned *Agility* continues to monitor the performance of the system with regard to the service requirements and if it identifies that the requirements are not being met will take action to reassign resources in order to rectify the situation.

Arjuna's Service Description Language is also used within *Agility* to express concrete descriptions of shared resources, the environment they require (i.e. their container) and the connections between them. *Agility* maps descriptions of abstract service requirements onto descriptions of the infrastructure supporting those service requirements.

Agility creates an internal cloud: *Agility* takes the services and resources assigned to it and, with them, creates an agile, internal cloud for the enterprise. Resources and services may be added, removed and modified, and *Agility* dynamically reconfigures resource usage to take account of the changes, all the while endeavoring to maintain existing service agreements. If unable to do so (because sufficient resource is not available) *Agility* will inform the relevant parties and provide a record of the events covered by the agreement, allowing them to diagnose the cause of the problem.

Agility is hidden to its users: *Agility's* service users, including applications which are utilising services, need not be aware of *Agility's* existence. This is because *Agility*:

- Does not require a new programming model to be introduced
- Does not require any modification of legacy applications
- Does not require any modification to the clients of legacy applications

Agility is fully exposed to its administrators: All relationships within *Agility* (e.g. between IT administrators and *Agility*, service administrators and *Agility*, service users and the resources assigned to meet their service requirements) are realised as service agreements. These service agreements are created and maintained with regard to the policies of the parties concerned. Details of the service agreements created, the events which occur within them, and the relationships between service agreements, are recorded for audit, capacity planning and real-time monitoring. The information recorded may be minimal or extensive depending upon the form of the service agreement and the policies involved. Administrators may dynamically modify policies with regard to specific service agreements in order to change the amount of information recorded e.g. in order to obtain an extra level of detail concerning a service agreement during some troublesome period.

Service agreements range from the long term (e.g. an agreement defining under which conditions one department may utilise resource owned by another) to the extremely short-term (e.g. *Agility* treats an individual invocation upon a service as a service agreement). Service agreements may also be nested and recursive. A simple example would be a long-term service agreement between departments containing agreements

representing the invocations of services. Capturing service agreements in this way allows the system administrator to discover the relationships between events of interest.

Agility seamlessly extends the internal computing utility across organisations: *Agility* is designed to be utilised by multiple, federated organisations which may have their own services, resources and policies. Initially, these organisations are likely to be semi-independent departments within a single enterprise, but *Agility* does allow the federation of separate enterprises, e.g. within a supply chain. Ultimately, *Agility* can provide seamless access to external utility/cloud services and resources without the need for clients to be aware of whether these suppliers are hosted within, or beyond, the enterprise.

Agility supports policy: *Agility* supports but does not control policy. It comes with a set of useful policies that can be immediately deployed, including those that provide load-balancing and fault-tolerance options. However, policy is completely configurable as *Agility* allows organisations to plug in their own custom policies or even connect to their existing policy engines. Policy is therefore completely under the control of the Enterprise.

Agility can manage all types of resources: Although *Agility* provides a container model and can control resource and container lifecycles, the means of driving lifecycle transitions e.g. deployment and undeployment, is entirely specific to the resource type in question. *Agility* comes with support for common resource and container types, and also provides an interface so other types can be supported by user-defined plug-ins. *Agility* also recognises that not all deployment can be scripted and so, when presented with a resource which cannot be automatically deployed, *Agility* will inform the resource's IT Administrator and await manual intervention. This gives complete flexibility to administrators over how to deploy a resource.

Resolving the Problem of Silos

Arjuna *Agility*™ resolves the problems described with the multi-silo approach by dynamically managing the deployment of applications on the underlying compute resources, in order to match business needs. As a result, all the performance objectives of an agile business can be improved:

Flexibility: With *Agility*, flexibility can be improved by decoupling service requirements from the specific infrastructure required to satisfy those requirements. *Agility* enables resource sharing between silos so that load can be dynamically redistributed across an organisation and potentially beyond, utilising any available capacity in a controlled fashion so as to maintain the necessary service requirements.

Speed: With *Agility*, speed of delivery for new and modified services can be improved by automating the time-consuming and error-prone process of manual system reconfiguration. Service requirements are expressed using Arjuna's innovative Service Description Language, additional resources required for the new and modified service are registered, and necessary changes to policy are made. *Agility* automatically deploys the new and modified service ensuring that all pre-existing service requirements continue to be met.

Dependability: With *Agility*, dependability can be improved by supporting the dynamic redeployment of resources whenever failures are detected. After failure, as service requirements are decoupled from the IT infrastructure, *Agility* is able to identify alternative resources that could be used to satisfy the service requirements and then reconfigure the system to use those resources in order to ensure continued fulfillment of service requirements.

Quality: With *Agility*, quality can be improved by measuring the system's performance directly against service requirements, expressed using Arjuna's Service Description Language, rather than against IT infrastructure measurements that do not directly relate to customer needs. Quality is also improved by automating the, normally manual and consequently error-prone, processes for reconfiguring the system when quality of service failures are detected.

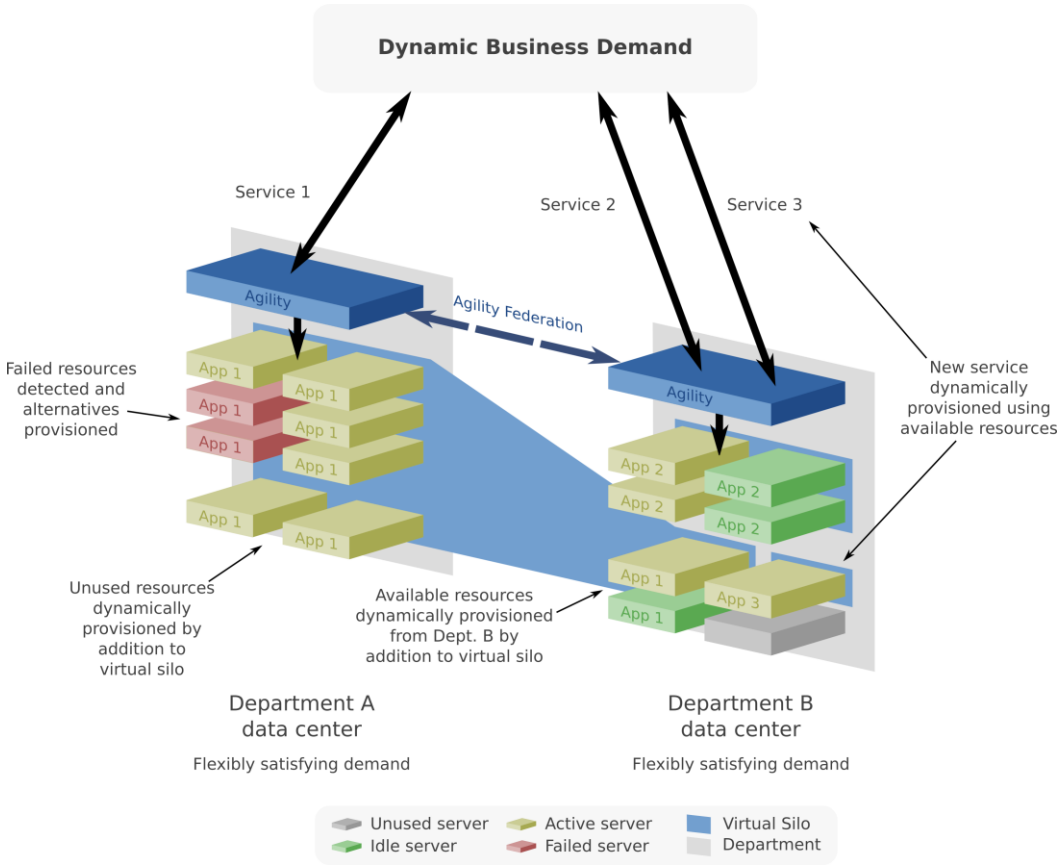

Figure 2. Arjuna *Agility*™ Enables the Agile Enterprise

Figure 2 illustrates the benefits of deploying *Agility* using the same enterprise example as depicted in Figure 1. In this case, each department installs a separate instance of *Agility* retaining control of its own resources through its own *Agility* instance. As demand changes and new services are requested *Agility* enables the lending and borrowing of resources as directed by Policy. In so doing, it transforms traditional silos into an internal cloud of IT resources, dynamically re-provisioning available resource to match changing demand. *Agility* therefore, de-risks the enterprise from under or over-provisioning problems, optimising existing resources and supporting all business requirements in a dynamic and adaptable manner.

Scenarios revisited

To illustrate the advantages of *Agility* we can revisit the dilemmas of Joe and Greg at Trading Inc.

Joe makes the initial decision to deploy *Agility*. The fact that *Agility* is a platform overlay means that the initial deployment has no impact upon running applications. Joe can take advantage of *Agility's* features one at a time, reducing risk and leaving Joe in total control of the process. Initially, he just uses the management aspects of *Agility's* container-based virtual infrastructure. This enables him to specify application structure, using *Agility's* Service Description Language, and to then manually deploy and undeploy virtual applications in a single step from the *Agility* console. Later however, he begins to specify his users' service requirements, again using the Service Description Language and to map those requirements onto the services offered by his virtual

applications. These virtual applications may now be deployed and undeployed automatically upon demand. Joe is then able to monitor the service agreements created within the system and to manually modify the IT infrastructure accordingly. Eventually, Joe encodes some of these modifications as departmental policy which allows the system to be reconfigured dynamically under certain, common occurring conditions e.g. a sharp increase in load.

Under-utilisation of IT resource

Finally, confident that his service requirements are captured and that *Agility* will automatically maintain the required quality of service, Joe is able to set his departmental policy to allow other departments' access to any unused infrastructure (and to record that usage for chargeback purposes). His policy defines under what conditions the infrastructure is made available, and under what conditions any infrastructure which is being used by others can be seized back.

Now, Joe can continue to offer his high levels of service but his unused resource is serving a useful company-wide function and generating internal income for Joe's department. Joe can even justify his low utilisation levels as benefiting the company as he now offers a pool of resources available to handle other departments' peaks in demand.

Over-utilisation of IT resource

With *Agility* in place within Joe's Trading department, Greg also deploys *Agility* so that he can regularly take advantage of Joe's pool of resources, and reciprocates by allowing his own resource to be shared under certain, clearly specified conditions. As a consequence, Greg can now service all work without delays. The fact that Joe's department retains overall control of the Trading department's resources and (as clearly stated in a Service Agreement) might, at any point, seize them back for higher priority work, is not an issue for Greg as such an occurrence is very rare, and besides most of Greg's department's work is jobs-based and so the jobs can just be rerun later if forced to terminate.

Sudden and unexpected spike in demand for IT capacity

When a sudden and unexpected spike in demand for IT capacity occurs within Joe's department, *Agility* recognises this fact and starts to deploy additional instances of the application onto the departmental infrastructure, as directed by departmental policy. However, despite this, the load continues to grow to the point where additional resource is required if the service requirements are to be maintained. *Agility* identifies that infrastructure allocated to Greg's department is currently unused and that Greg's department's policy allows that infrastructure to be made available to others (if, and only if, the tasks to be run are deemed higher priority by the company). Joe's revenue-generating tasks qualify, and so instances of the required application and its entire environment are deployed automatically on the spare infrastructure. The system records Joe's usage of Greg's infrastructure (as a service agreement between the two departments) enabling Greg to later cross-charge Joe's department at previously agreed prices.

The spike is handled without loss of business (although the running of some of Greg's back-end applications has been delayed) and when the load finally begins to decrease, the instances of the application are removed from Greg's infrastructure. Unless they were monitoring the system in real-time, neither Joe nor Greg were aware of the 'crisis' as it happened (although the entire record of the event may be reviewed by examining the service agreements, and *Agility's* response). Joe has maintained the required service, the company has maintained its

revenue and its reputation, and Greg, although he has some backlog to handle, has some additional income in compensation.

QoS degradation due to poor configuration

One day (before the problem becomes noticeable to a business manager) *Agility* detects that QoS thresholds (defined in policy by Joe) for a particular application have been breached and sends an alert to Joe's admin console. The team examines the relevant service agreements and uses the information to trace dependencies. They modify policy relating to those service agreements in order to obtain more detailed information. Consequently, they are able to identify the cause of the problem as a failure to capture a close dependency that two components have on one another. The service requirements are modified to recognise this dependency. *Agility* then automatically reconfigures the system in order to take this change into account, whilst continuing to maintain all other dependencies that were already defined.

Beyond a brief increase in response times as components are redeployed, the users do not notice anything.

Catastrophic failure

One night, there is an electrical fault in Greg's Analysis department and a fire starts. The entire building is burnt to the ground.

Even as the fire is raging, Agility notices failures are occurring and begins to redeploy work to the Trading department (which is at another site, and largely unused during the night).

In the morning, although all systems are still functioning, the company executives meet to discuss any necessary modifications to policy, in the light of the significantly reduced IT capacity. These changes are introduced and the system then automatically reconfigures itself to support the new policy. As a result, high priority applications continue to function, and cash keeps flowing through the organisation until new servers can be deployed. The modified policy now becomes a permanent feature of the system so that any reoccurrence of a similar event will be handled automatically.

The Future

Joe and Greg are able to satisfy the CIO's demand for consolidation and rationalisation in their IT departments.

Using *Agility* they incrementally retire old servers and seamlessly transfer the workload to machines with available capacity. Where required, they replace old machines with new machines requiring 10% of the previous power consumption and reallocate workload to these machines using *Agility*. This consolidation significantly reduces the physical space required to house the machines and the overall power consumption required thus contributing to Trading Inc.'s strategic initiative to reduce its carbon footprint.

Additionally, the flexibility within Joe's IT infrastructure has allowed the delivery of new services, faster than before, providing the trading department with a competitive advantage and improving profitability. Greg's department is also able to utilise spare capacity from the Trading department and can now satisfy all of its workload in a timely fashion.

The company also improves their disaster recovery processes by extending their Agility deployment to Amazon's EC2. Now if disaster strikes, suitable applications and their data sources are pre-defined to be automatically deployed on EC2 by Agility, freeing up resource within the enterprise to run the core applications. To Trading

Inc.'s service users and administrators the dynamically assigned EC2 resources will appear no different to resources utilised within their enterprise.

Conclusions

This paper has described how an inflexible IT infrastructure will create a barrier to business agility and has illustrated these problems with all too common business scenarios. In contrast, it has described how Arjuna Agility™ is a solution that can overcome these barriers, reinforcing the fact that an agile business needs an agile IT infrastructure.

> "*[Business] agility will become a major business differentiator in a connected world. Business agility requires agility in the datacenter, which is difficult as many of the technologies for improving the intelligence and self-management of IT are very immature, but they will evolve over the next ten years,*" Tom Bittman, Gartner Analyst7.

Agility offers enormous business and technical benefits. It reduces costs by using the services and resources registered with it to create an agile, internal cloud for the enterprise. Controlled sharing of these resources enables better utilisation of existing IT, improving performance and reducing new hardware, maintenance and support spend. As *Agility* simplifies the IT operation, administration costs can also be reduced. *Agility* bridges the semantic gap between the way business people specify their service requirements, and the way IT configures and deploys these solutions, speeding up the time to service availability. As service demand changes, *Agility* dynamically provisions IT supply to match it, de-risking the enterprise from over- and under-capacity problems. *Agility* allows the IT department to better understand how resources are being used, by which clients and under what conditions. By using service agreements *Agility* creates a full audit trail of activities which may be useful for regulatory compliance (e.g. Sarbanes-Oxley) as well as operational efficiency. Ultimately, *Agility* can seamlessly extend this 'internal utility' to external 'utility computing' services and resources as and when required by the enterprise.

Agility provides these advantages without forcing 'big bang' changes onto the existing IT system. By allowing incremental progress, *Agility* provides a way for the enterprise to move towards a fully agile IT infrastructure in a controlled manner.

Improving business agility today allows the enterprise to compete more effectively in the current turbulent market environment. In the future, competitive success will go increasingly to enterprises that are able to achieve radical change both within the enterprise and externally throughout the whole extended business network.

7 Savvas, A. (2007), 'Business agility and virtualisation will be key measure of datacenter excellence', Retrieved from: http://www.computerweekly.com/Articles/2007/10/24/227703/business-agility-and-virtualisation-will-be-key-measure-of-datacentre.htm

Arjuna Technologies Limited is a world leading innovator in reliable, large-scale distributed systems and has built an international reputation for delivering mission-critical products and consultancy to global businesses. We are proud of our history and are uniquely positioned to enable clients to profit from the most compelling business and technical proposition in IT - Cloud Computing

Web: www.arjuna.com
E-Mail: info@arjuna.com
Tel: +44 191 243 0676
Fax: +44 191 243 0677
Office: Arjuna Technologies Limited
Nanotechnology Centre, Herschel Building
Newcastle upon Tyne, NE1 7RU, UK