

**Contacts:**

Clive Longbottom  
Quocirca Ltd  
Tel +44 118 948 3360  
[Clive.longbottom@quocirca.com](mailto:Clive.longbottom@quocirca.com)

Elaine Axby  
Quocirca Ltd  
Tel +44 20 8874 7442  
[Elaine.axby@quocirca.com](mailto:Elaine.axby@quocirca.com)

# Connectivity and SOA

## Embracing Old Assets, Furthering the New

**Executive Summary**

*Service Oriented Architectures (SOAs) will provide greater flexibility for those utilising them, but will also bring greater issues for data and functional connectivity. This paper contrasts and compares the capabilities of a point-to-point and an enterprise service bus approach.*

**Main Findings**

- **Current trends in technology are leading to application decomposition**  
As adoption of Web Services increases, traditional applications (e.g. ERP, CRM) are being seen as too siloed in their approach to the business needs.
- **The move to SOA will drive connectivity needs**  
The move to a more services-led approach will drive the need for the adaptors at an exponential rate. With services providing discrete functions, a single process may well involve the need for hundreds of these to interoperate and exchange data to facilitate the process needs.
- **Organisations need to look at outwards connectivity**  
“Value chains”, where processes include suppliers and customers, can provide distinct market differentiation where process connectivity is used successfully.
- **“Point-to-point” solutions will rapidly become unusable**  
“Point-to-point” connectivity solutions require adaptors at the rate of the number of data stores/end points squared plus 1 ( $n^2+1$ ), whereas a bus-based approach only requires one adaptor per data store/end points ( $n$ ). This means that managing a point-to-point architecture will rapidly become infeasible.
- **Contextuality of data connectivity is key**  
To maintain business and technical flexibility, data mapping and transformations need to be carried out by adaptors associated with the data store/end point, and abstraction needs to be provided by a bus architecture.
- **Bus architectures cut down on the need for management and maintenance**  
The intelligent abstraction of connectivity means that fewer adaptors need to be managed and maintained, enabling organisations to invest more IT budget in facilitating the business, rather than spending on fire-fighting and maintenance.

**Conclusion**

SOA will provide greater flexibility for organisations looking to automate and facilitate dynamic processes both within their own organisation and between themselves and their suppliers and customers. However, connectivity within an SOA is a major problem, and must be addressed at an early stage. The use of a bus architecture provides a high level of abstraction and minimises the number of adaptors required, while data transformation and intelligent data mapping helps to lower the time spent on retro-testing and maintenance costs.

*An independent report by Quocirca Ltd.*

[www.quocirca.com](http://www.quocirca.com)

Commissioned by IBM

The logo for Quocirca, featuring the word "quocirca" in a lowercase, sans-serif font. The letters "quoc" are in blue, "irca" is in red, and the "i" is in black.

## 1 SOA – Breaking Down the Barriers

A Service Oriented Architecture (SOA) brings a new approach to IT functionality within an organisation. Whereas old approaches had to come from a starting point of an application providing the total solution to specific problems, the idea behind SOA is to provide a far more flexible system, based on discrete pieces of functionality being called as required from any business function to facilitate specific business requirements.

This approach has many benefits – for example, functional redundancy, where multiple applications carry out the same function within their own domain, is minimised, and the optimisation of any function provided in this way is immediately applied to all processes that use that function. The removal of functional redundancy also means that less hardware is required to run the function – no discrete resource is required for each of application a, b, c and d's version of the function; instead, the single instance of the function can be sized to meet the composite requirements of the multiple processes that will be calling this function.

Also, SOA can provide a much greater level of flexibility when it comes to resource utilisation. With average hardware utilisation running around 10-15% in a standard client/server or monolithic web-based application environment, SOA can provide the means to up utilisation to 60% + through the use of virtualisation and dynamic provisioning of functions.

## 2 The Real Business Starting Point

With all the benefits around SOAs, it feels as if SOA should have replaced existing architectures already and that old-style, monolithic applications should be ripped out to be replaced with new Web Services-base composite applications built from disparate functional components. As we all know, this is far from the reality, and ongoing business pressures will stop all but the very brave (or foolish) from carrying out a full “rip and replace” approach in the implementation of SOAs.

The main problem is that heavy financial and resource investments have been made in existing systems – and these systems were implemented for all the right reasons when client/server and application centric models were accepted as the norm.

Even solutions sourced in the past couple of years will not be fully SOA-enabled – the majority of applications that have been built to Web Services standards will still be monolithic in approach, and will contain all the functionality required to carry out the set of solutions that the owning vendor is trying to solve. With hard-coded internal connections between functions being replaced with tightly coupled Web Service-based connections, the end users are often little better off than before.

So, the real business problem is the perennial one of how to get to the future from the present; how to ensure that existing investments are protected while being embraced and suitably utilised by the new approach to business-centric computing.

## 3 The Need for Connectivity

Existing applications tend to have been architected to deal with vertical needs – ERP systems deal with specific areas of inventory, business asset management and so on, while CRM systems deal with customer issues. The expansion of the specific Enterprise Resource Planning (ERP), Customer Relationship Management (CRM) and Supply Chain Management (SCM) vendors into a greater range of coverage to try and tie in their own customers to a single vendor solution has forced the vendors into creating connectivity solutions into their overall portfolios to enable their disparate offerings, often built up through mergers and acquisitions, to interact with each other. In many cases, this interaction is carried out through hard coding of discrete adaptors from one part of the solution to the other. In other cases, the vendor will create more “open” connectivity for data exchange using technologies such as the eXtensible Markup Language (XML) for data formatting.

The biggest problems come when we look at the more heterogeneous environments, which may also include the need to connect mainframe and other computing environments into the distributed SOA environment. Here, with less abstraction between the application and the run time environment, hard coded connectivity tends not to work for long, and semi-proprietary approaches to “open” XML are little better.

For many organisations looking to implement SOA, connectivity will be one of the biggest issues. Getting application A and application B to interact has always been a problem, and whole markets have grown up around this – indeed, the term “Systems Integrator” came from the provision of help to link disparate systems and solutions together. In the 1990s, many vendors came to market with Enterprise Application Integration (EAI) solutions, again as a means of creating a more closely integrated set of solutions that would help an organisation be more flexible in its chosen market. The biggest problem has been standards, and a lack of adherence to them – applications were coded in specific languages, using proprietary means of dealing with internal events and data structures. The “easiest” way to enable interaction was often through tapping in to the back end data store – and this immediately lost any business context, and increased the complexity of maintenance.

For many organisations, web services are now being adopted. However, in many cases, this adoption is being done on a project-by-project basis through the use of distinct web service to web service adaptors, so replacing one set of hard-coded connectivity with another. It is Quocirca’s belief that this direct Web Service to Web Service connectivity is not the real way forwards. Through the use of abstraction, Quocirca believes that a much greater degree of ongoing flexibility can be obtained within an SOA, and that businesses will gain the capability to more closely focus on their core competencies, knowing that the infrastructure will be flexible enough to support the changing business needs.

## 4 Abstraction and Connectivity

For a truly flexible SOA environment, it is necessary to remove the dependence of one function directly to another.

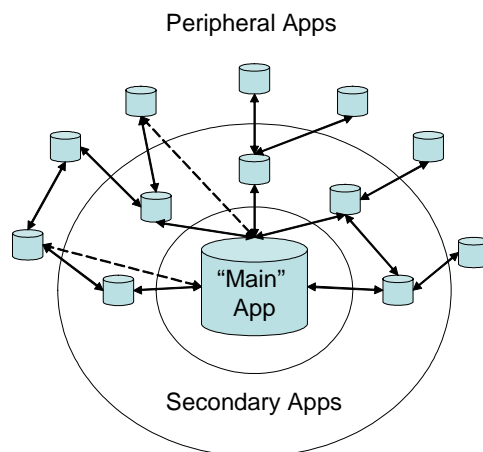
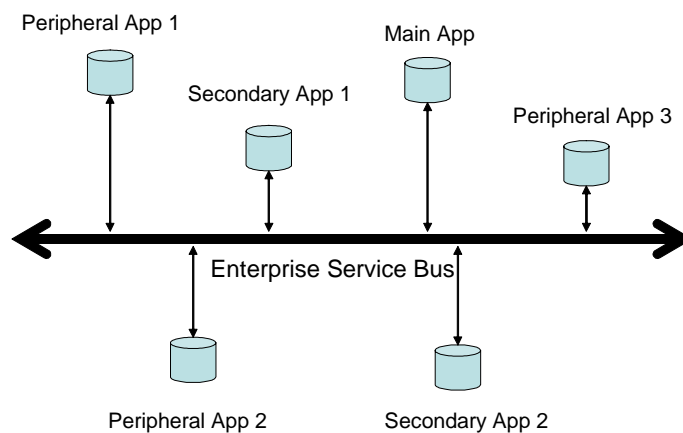


Figure 1

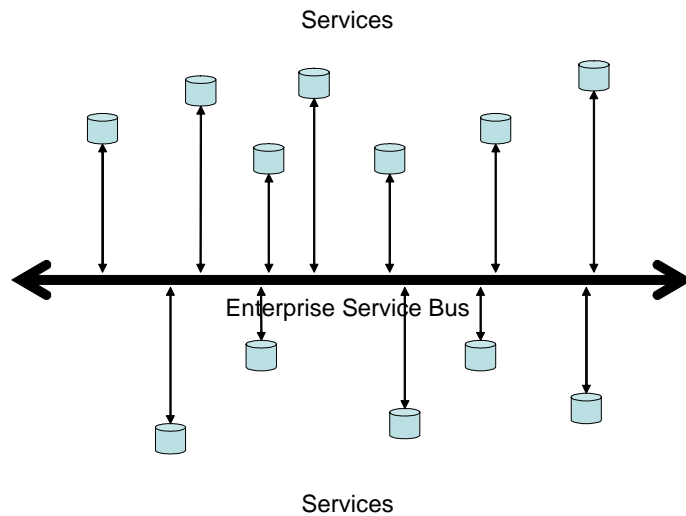
If we look at figure 1, we see the problems with an application centric view of the world when it comes to connectivity. This figure only shows a partial map of the possible connections, showing what may be the currently utilised connections. Any actions that take place purely within the main application will have no problems, and any actions that involve connectivity to “preferred” external applications will generally have defined adaptors provided by one or other of the application vendors. However, once we move through to non-preferred applications, or applications that are more than once removed from the main application, we may well come up against problems. To gain connectivity, we may have to use one of the secondary applications as a route to the peripheral application, or adaptors may only be provided by third parties and may need changing when either of the applications dependent on the adaptor are updated or patched. The costs of testing all such adaptors after every patch and upgrade can be horrendous, and the application topology can become extremely complex as the number of applications and connections increases. If new connectivity is required between two applications, we are looking at two new adaptors.

However, if we take an Enterprise Service Bus (ESB) approach as in Figure 2, we raise the level of abstraction to make it that any adaptor is only dependent on a single application, rather than paired between connected application instances. The use of an ESB means that all applications can be treated as peers, and each adaptor only has to deal with a single application, as the ESB itself deals with one end of the adaptor. This lends extra flexibility – any application can change without impacting any other application – if the change to the application requires a change to the adaptor, this can be easily applied, and retro-testing becomes far easier and more rapid.



**Figure 2**

Finally, we need to look at the emergence of SOA as a force in the market. If we look at Figure 3, we see that an SOA has to be dealt with in a way that is very similar to an application-centric ESB. The main difference here is in the number of items that the ESB has to deal with – the decomposition of applications into services as part of an SOA requires far more connectivity to be provided, and requires greater flexibility, greater scalability and reliability. If you try to imagine this being dealt with through a direct application connectivity approach, the number of dependent adaptors rapidly becomes unmanageable, and the amount of network “chat” could well become appreciable, slowing down overall response.



**Figure 3**

Historically, the main reason why an ESB approach has not been viable has been that each application vendor has worked to their own set of proprietary “standards”, and as such adaptors have had to be hand-crafted and maintained at high cost. We are now at a position where standards have emerged that are reasonably well adopted and adhered to by the majority of vendors. J2EE and .NET are the main architectures of choice, and Web Services are rapidly emerging as the main means of making functionality visible to the external world. Even for older applications that do not directly support Web Services and for applications outside of the main distributed environment, tooling is available to uncover functional parts of the overall application to make the function available to other parts of the infrastructure. The need for optimisation of existing investments in applications leads to the need to connect all applications to an ESB, and not just new web services enabled applications. This recognition is leading to an extension in ESB capabilities for projects requiring this capability, with off-the-shelf adaptors through to specific functionality in the more widespread enterprise applications, enabling these functions to be exposed as web services to the rest of the environment.

Through the use of an ESB, SOA becomes a more achievable aim. Whereas many organisations have been put off from moving to SOA due to a (misplaced) perception as a “rip and replace” project, an ESB combined with suitable tooling enables companies to maintain existing investments while implementing new functionality as discrete services.

## 5 Contextuality and Data Abstraction

Maintaining the context of data within connectivity is key. Tapping into data stores at the database level is relatively simple, but maintaining and understanding data maps along a connectivity chain is not. Direct, point-to-point data maps breed a high degree of extra adaptors, and any change to a data store (e.g. the addition of a new column or field) may result in the need for changes to all adaptors associated with that data store. Commonly known as an “ $n^2 + 1$ ” issue, the number of direct adaptors where contextuality is not maintained grows at the square of the number of data stores/end points, plus 1 – for example, full connectivity for a solution involving 5 data stores/end points would require 26 discrete adaptors.

With a bus style approach, full contextuality can be more easily maintained through the bus. Intelligence built in to the adaptors provides the capability to transform data from the

underlying data stores so that data maps can be initially inferred and then rapidly checked manually for veracity and altered as required. These data maps can then more easily be maintained, as the connection points are then only between the one end point/data source and the bus. A change in one data store will therefore only have a direct impact on the adaptor associated with it – other adaptors will maintain the data context through their association with the ESB. This then means that full connectivity for a solution involving 5 data stores/end points would only require 5 adaptors (an “n” approach – the need for adaptors grows linearly with the data stores/end points themselves). This also leads to greater flexibility, and in the capacity for functional reuse of specific service components through the adaptor itself – a key requirement for SOA.

If we scale this to a more complex environment, involving, say, 100 data stores/end points, full connectivity in an ESB environment with data maps and transformations being dealt with by the adaptors requires 100 discrete adaptors for complete interoperability of data across the whole system. Compare this to a direct point-to-point solution – there would be a requirement for 10,001 adaptors – a completely unmanageable solution. Even assuming that the majority of these connectors would not be required (that is, many data stores/end points do not have the necessity to interoperate), we would need to go down to a 1% overall connectivity to get down to the simplicity of an ESB approach. In the case of such a point-to-point environment, Quocirca finds that connectivity is therefore only provided where absolutely necessary, minimising flexibility and constraining the organisation’s capabilities in the market.

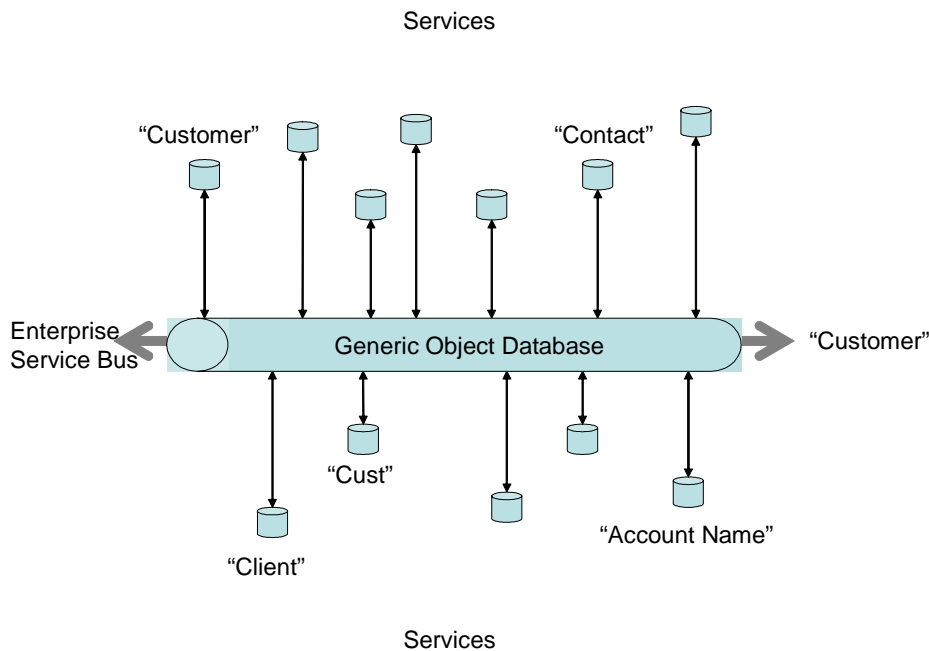
Now move this through to an SOA – we may have tens of thousands of discrete services, each requiring their own connectivity solution so that they can be utilised as start points, through points and end points within the changing needs of business processes. A direct point-to-point solution means that even in an environment where we minimise overall connectivity we will be looking at hundreds of thousands to millions of adaptors – with any change to a single service possibly impacting thousands of other adaptors. Even if you have a relatively simple system at the moment, the continuing move to SOA should make it that ensuring that you have the required flexibility for the future will mean that any new applications or functionality should be implemented utilising the connectivity capabilities provided by the use of an ESB.

## 5.1 Generic Object Database and Common Data Definitions

As we move forwards, the need for a common data definition become more important. Even with an ESB approach, we still run into an overall information visibility issue. One data source may well have a customer field marked as “Customer”, whereas another may have it as “Client”. Solutions that are attempting to provide a “one true source” still have dependences on being able to request the information from the adaptor in a manner that is standardised and common across the whole organisation.

Once way around this is to centralise the data definitions, via a generic object database. Here, a secondary map of the information held within the adaptors is created, showing how the different data sources’ fields map on to each other. Through this means, any request for information can then be made using a common definition – a Customer information request can always refer to a common data definition of “Customer” – the generic object database then handles any translation that are required in talking to any of the dependent adaptors.

A high-level example is shown in Figure 4. Here, we have many different services, many of which have their own different definition of a customer – “Customer”, “Client”, “Account Name”, “Cust”, “Contact” and so on. For any requesting service that requires all the known information about a specific customer, specific requests would have to be written to interrogate each separate service if we were to utilise either a point-to-point or a single-connector ESB solution. However, by creating a single generic object database, the mapping of each of these different definitions is already known. Therefore, we can interrogate the generic object database through a single definition (in this case, “Customer”), and the generic definition then decomposes into the specific definitions as required.



**Figure 4**

Although this requires more work up front in carrying out dual data mapping (one for the adaptor and one for the generic object database), the flexibility provided becomes a major facilitator for market differentiation.

## 6 Flexibility of connectivity and Value Chains

Another area that is often overlooked when organisations look at connectivity is the growth of inter-organisational data exchanges. More often than not, an organisation looks only within its own four walls, neglecting to look to the massive efficiency and effectiveness gains that can be made through the use of connectivity along the value chain.

If we look at a simple value chain, we have a supplier, ourselves and a customer. We can work along this chain as three separate, isolated processes, with each participant looking after their own responsibilities. However, if the customer requests information about the delivery of the end item, for example, this request has to be taken from their solution, passed to our organisation in isolation, and we then have to contact the supplier and find out the status of the delivery to us. With no integration along the chain, we have the probability of errors creeping in to the accuracy of the informational exchanges, as well as the time element of the manual steps.

Now let us suppose that we integrate these processes. If we do this through “hard” connectivity (application to application), we become highly dependent on the status of the applications within each of the constituents within the chain. Should one of the constituents change an application, we can no longer be sure that the overall data flows will still work. Previous research by Quocirca has shown that the majority of companies have several thousand suppliers and customers – and managing such a complex connectivity environment becomes impossible. Once we start looking at the use of services, we have a massive increase in the possible problem – rather than silos of application that we need to connect, we now have decomposed services that need to be rapidly aggregated to create the facilitation for the required processes – and hard connectivity just does not allow for this.

Here again, the ESB provides the capability for the required flexibility. Internally, services are connected via the ESB, and external connections are also included. As before, as the ESB provides one end of the connectivity, and changes with services within the other components

of the value chain can be more easily dealt with through changing the ESB adaptor, rather than trying to deal with service-to-service adaptors and the need for dependency checking throughout the whole system.

## 7 What to look for in Connectivity solutions

When looking for solutions offering the best flexibility for connectivity in an SOA architecture, there are several things that should be considered.

- **Connectivity** – Obvious, but important. Check to see what adaptors are available out of the box, and what the level of support for non-standard adaptors via the channel is, to ensure all parts of the business can benefit from enhanced connectivity.
- **Standards** – With standards now maturing rapidly, full support for evolving standards such as Web Services and XML alongside support for older standards is key.
- **Interoperability** – Wherever possible, the chosen solution must be able to understand the applications that it connects to. With this capability, process flows are maintained in context, retro-testing is minimised and flexibility is maximised.
- **Scalability** – The specific needs of a service oriented architecture means that many more calls will be made on adaptors. Therefore, the scalability of the chosen solution will need to cope not only with the immediate need, but also with the future need as existing silo applications are decomposed to services.
- **Flexibility** – How can new adaptors be added and how are existing adaptors upgraded?
- **Manageability** – Your SOA environment will be dependent on connectivity: without it, no composite application or process will be able to work. Therefore, it is imperative that the connectivity solution is fully manageable, preferably through a centralised systems management solution. Alerts, automated actions, root cause identification and the remote management should be available in the chosen solution.
- **Resilience** – It is important to ensure that the chosen SOA connectivity solution has inbuilt capabilities for failover, and for roll-back and resume should anything happen in any dependent services or applications. In these cases, the adaptor will have to be the point of intelligence as to what steps will need to be taken.
- **Future directions** – is the vendor supportive of providing functionality such as a generic object database? Does the vendor have a story for a move through to a fully utility style computing environment, such as Grid and Software as a Service? How does the vendor view value chain connectivity and the associated issues such as data protection and data fidelity?

## 8 Customer Example

GROHE is a company providing water fittings for bathrooms and kitchens. Headquartered in Germany, this 5,600 employee company has 20 subsidiaries and 12 sales offices across 130 countries. Annual sales in 2005 were € 865m (\$1.1bn).

### 8.1 Customer's business issue

GROHE was implementing a new SAP enterprise resource planning system, and needed to figure out how to exchange data between this new SAP implementation and existing legacy applications that were crucial to the company's business. These included duty and plant applications and delivery, invoice and product catalogue systems, as well as bar coding, logistics and inventory management software.

GROHE was also up against tight timescales – the company needed the SAP system to be up and running, complete with data connectivity, within 2 months.

## 8.2 Customer's approach to solving business issue

In total, GROHE identified 14 interfaces that would need to be created to integrate older applications with the newer SAP modules. As the project had to be completed quickly, GROHE needed to determine whether it was more cost-effective to perform hand-coded, point-to-point integration or purchase a packaged solution designed for speedy integration and continued consistency of business processes.

GROHE's software manager, Armin von Dolenga, compared the time, cost and effort that would be needed to manually program 14 SAP interfaces, and then compared this to an approach utilising an Enterprise Service Bus approach. Von Dolenga calculated that it would have taken 6 months to hand-code the required connectors, and rapidly decided that an SOA approach based around an ESB would be the only viable way forwards.

SerCon, an IBM company, was chosen as the partner to design and implement an SOA for GROHE using IBM's WebSphere family of products to provide the needed ESB.

## 8.3 Customer's perceived benefits from implemented solution

The IBM WebSphere ESB implementation now handles between 5,000 and 25,000 messages per day, enabling full data transformation and providing a global exchange of information with a string of services between decoupled front and back ends.

This SOA incorporates standardized interfaces that use common message formats such as XML and SAP Intermediate Documents, so GROHE is ensured that its business services can remain stable and well-defined, yet easy to change in order to meet the fluctuating needs of the business. Because of this building-block approach, von Dolenga estimates that his IT group can bring a new service online within two to four weeks.

The new WebSphere software-based solution significantly reduces the time and cost required to integrate older applications with the new SAP modules compared to a hand-coded, point-to-point integration technique. By enabling reuse of existing resources with an open standards-based solution, GROHE is preserving its investment in its existing assets.

Von Dolenga says that the solution has not only allowed him to standardise the SAP and database interfaces, but also to service-enable GROHE's legacy systems, so facilitating future business integration projects.

## 9 Conclusion

Connectivity within and across organisations is a problem that will only get worse in the future. Not only are companies looking to the use of application integration to provide greater flexibility within existing solutions, but the increased move to Service Oriented Architectures means that connectivity requirements will grow exponentially. Point-to-point solutions will increasingly become unsupportable, and those organisations using such solutions will spend increasing amounts of money and time in retro-testing adaptors and processes every time that a single service changes. However, for those who choose an Enterprise Service Bus solution, such retro-testing will be minimised – enabling the organisation to invest IT budget in new functionality and to be more responsive to the markets' needs.

## About Quocirca

Quocirca is a company that carries out world-wide perceptual research and analysis covering the business impact of information technology and communications (ITC). Its analyst team is made up of real-world practitioners with first hand experience of ITC delivery who continuously research and track the industry in the following key areas:

- Business Process Evolution and Enablement
- Enterprise Applications and Integration
- Communications, Collaboration and Mobility
- Infrastructure and IT Systems Management
- Utility Computing and Delivery of IT as a Service
- IT Delivery Channels and Practices
- IT Investment Activity, Behaviour and Planning
- Public sector technology adoption and issues

Through researching perceptions, Quocirca uncovers the real hurdles to technology adoption – the personal and political aspects of a company's environment and the pressures of the need for demonstrable business value in any implementation. This capability to uncover and report back on the end-user perceptions in the market enables Quocirca to advise on the realities of technology adoption, not the promises.

Quocirca research is always pragmatic, business orientated and conducted in the context of the bigger picture. ITC has the ability to transform businesses and the processes that drive them, but often fails to do so. Quocirca's mission is to help organisations improve their success rate in process enablement through the adoption of the correct technologies at the correct time.

Quocirca has a pro-active primary research programme, regularly polling users, purchasers and resellers of ITC products and services on the issues of the day. Over time, Quocirca has built a picture of long term investment trends, providing invaluable information for the whole of the ITC community.

Quocirca works with global and local providers of ITC products and services to help them deliver on the promise that ITC holds for business. Quocirca's clients include Oracle, Microsoft, IBM, Dell, T-Mobile, Vodafone, EMC, Symantec and Cisco, along with other large and medium sized vendors, service providers and more specialist firms.

Sponsorship of specific studies by such organisations allows much of Quocirca's research to be placed into the public domain.

Quocirca's independent culture and the real-world experience of Quocirca's analysts, however, ensure that our research and analysis is always objective, accurate, actionable and challenging.

Quocirca reports are freely available to everyone and may be requested via [www.quocirca.com](http://www.quocirca.com).

**Contact:**

Quocirca Ltd  
Mountbatten House  
Fairacres  
Windsor  
Berkshire  
SL4 4LE  
United Kingdom  
Tel +44 1753 754 838