

The SOA Magazine

Feature Article



Shadow IT: Edge Applications in a Service-Oriented Enterprise

by Paulo Rosado and Rodrigo Castelo

Published: May 16, 2008 (SOA Magazine Issue XVIII: May 2008)

[Download this article as a PDF document.](#)

Abstract: The term "shadow IT" was coined for systems built without corporate approval inside business units, departments and whole subsidiaries. Born from the IT backlogs of every organization, shadow IT can drive innovation and effectiveness without hindering larger IT evolution. While organizations wrestle to centralize their applications, the reality is shadow IT is not going away. As such, it is most effective to ensure the sustainability of shadow IT instead of attempting to avoid it.

To be sustainable, this innovation at the edge must take advantage of central SOA environments and, when the time comes, seamlessly scale into a service-oriented enterprise and become an active part of it. However, for this to be achievable, edge applications must have certain capabilities and satisfy certain conditions.

Introduction

Large enterprises are continually struggling with the problem of deploying the right IT architecture to support their independent business units or remote subsidiaries. From the point of view of an IT organization, centralizing business applications makes sense, since it decreases cost of ownership and relieves maintenance headaches. The flipside of centralization is that branch organizations and business units—the "edge" of the organization—are faced with a lack of flexibility to evolve these same IT systems. In many cases, IT simply does not respond to the fast-paced drive for innovation, process optimization, and evolution demanded by the business side.

From this seeming conflict of interest *edge applications* are born. Also known as "shadow IT", we extend the term to include all systems created outside central IT control inside business units and departments. "Shadow system" or "rogue system" are used with similar connotations. This type of conflict occurs in most companies, from small businesses to Fortune 15 corporations. The good news is that SOA can help companies address shadow IT since it allows the enterprise to sustain "innovation at the edge" in a rational fashion.

In this article, we begin by describing and defining shadow IT and edge applications as they exist in current SOA environments and evaluate their unique business value. We then dig deeper into the top four challenges that edge systems pose to SOA environments with regard to sustainability. Specifically, we take a look at how shadow IT can fit into a service-oriented enterprise (with respect to governance, scalability, and performance requirements) by explaining the top four rules for sustaining edge applications in support of SOA.

Why Should Shadow IT be Sustained?

Even as organizations wrestle with centralization, shadow IT is not going away because it serves a purpose. Misalignment between business and IT has been a reality since the dawn of modern information systems. A bigger slice of budgets now tends to go to business units that are closer to revenue sources and less on the IT side. These business units naturally start to buy or build their own systems, because IT has become unable to support them in such tasks, and because they now have the budget to do so. Thus, the phenomenon of shadow IT emerges in

departments, business units, and subsidiaries, across all industries, supported by the deployment of simple applications in Excel and Access, all the way to fat desktop clients written with technologies like Visual FoxPro.

Another factor is that requirements from business units tend to build up into a very large IT backlog. When these demands increase and the unit's bottom line is directly hit by the lack of appropriate IT response, central IT is pressured to give ground on their ironclad rule of centralizing everything. In many cases, business units take matters into their own hands and redirect portions of their budgets to either buy business applications or commission the building of solutions to solve pressing issues.

Subsequently, they start to struggle with their own shadow IT maintenance cycles and become swamped—just like central IT. Currently, two approaches are available to minimize these issues. On one hand, SaaS appears as a valid outsourcing option for business-line units and their shadow IT. On the other hand, central IT has developed ways of doing more with less, with many IT departments adopting SOA.

With the technology and methodologies now available, edge applications can be consolidated through Central IT and SOA initiatives, as illustrated in Figure 1.

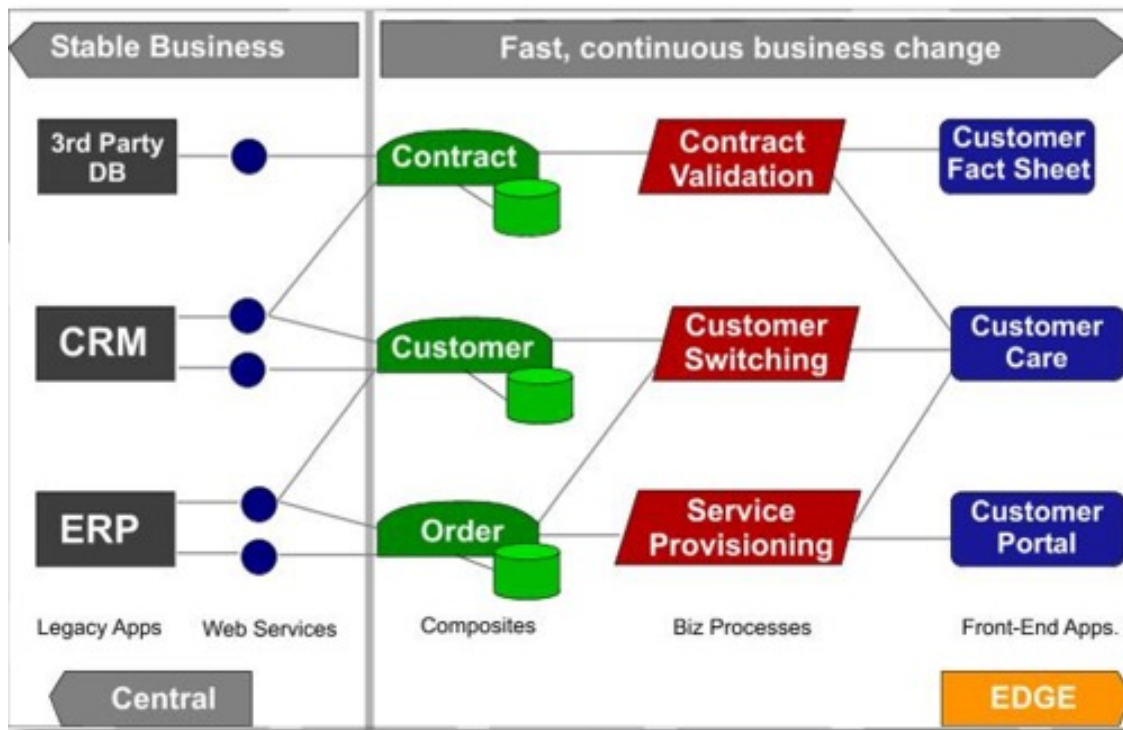


Figure 1: An overview of central IT with edge applications.

Three Ways to Support IT at the Edge of an Organization

Forrester Research estimates that shadow IT expenditures account for 5% to 15% of the average IT budget. CIOs acknowledge that they lack the time, resources and business proximity to fully automate all core business processes that involve departments, business units, and subsidiaries, or that span partners and customers.

Faced with these realities, IT departments are left with three alternatives:

Option 1: IT Decentralization

Some organizations take the route of full decentralization. In a specific case, a corporation with 20 subsidiaries adopted a fully decentralized IT governance model. Each subsidiary has a full-fledged IT department, a fat local budget, and a fair amount of leeway in enterprise "buy vs. build" decisions. Central IT retains veto rights for large projects and ensures a certain degree of company-wide adoption of standards and best practices.

In spite of the fact that subsidiaries require more or less the same collection of business applications, basic fundamental differences in operating procedures and business processes have made these systems useless to other subsidiaries. In most cases, long time-to-market, high cost of adaptation, deficient knowledge transfer processes, and

the reluctance of IT departments to inherit large, opaque systems they haven't built or maintained, results in a decision to buy or build a fresh local system.

As a result, subsidiaries have a high degree of flexibility, but the company is plagued by a tremendous "redundancy" cost.

Option 2: IT Centralization

In other corporations, central IT tightens its iron fist on edge IT spending and the type of projects they are allowed to start independently. This creates a complex problem for local business units as central ERP processes fail to account for a large number of specific needs that should be solved through IT. The outcome is problematic for business and gives rise to true rogue projects financed clandestinely by some business units.

Option 3: The Pragmatic Middle Ground

Some organizations adopt a middle ground. Central IT purchases and controls large core packages, while departmental IT units immersed in business units create small tactical projects to solve specific needs and problems. Projects deemed small (typically less than three man-months) are developed locally in technologies like Excel or Access, while larger projects get added to the central IT backlog. Such a model keeps business users relatively happy and eases the pressure on an understaffed central IT department. However, this huge assortment of "mushroom" applications results in ever-rising maintenance and support expenditures. Without prior planning, business units end up spending up to five times more money than initially planned for supporting and evolving their applications.

Among the available alternatives, the third option appears the most balanced and promising. In order to achieve a pragmatic middle ground requires that you establish an environment that has certain characteristics in support of edge applications:

- *Rapid development based on Web standards* - The style of development needed for edge applications requires that solutions be easy to learn, fast to deliver, and cheap to maintain. Solutions are required to foster an iterative development model and be easy to tune, change, and redeploy. At the same time, the use of Web Services and the delivery of highly usable UIs over-the-web are mandatory requirements.
- *Consolidation of corporate data* - Independently of the type of user interface or interaction device used to access the edge applications, data must be kept in an adequate repository and supported by a centralized database server engine. This way, older data that is spread across departmental databases, spreadsheets, and mailboxes will be centralized and consolidated. Moreover, IT administrators must be able to control access and resource usage in such a way that the multiple islands of data can coexist and thereafter be integrated and consolidated.
- *Scalable, robust infrastructure* - Fast and iterative development must be accommodated, without compromising the future scalability and robustness of the applications. This needs to include the ability to configure, isolate, and distribute load among runtime components and databases, without having to re-architect and rewrite the applications.
- *Transparent configuration management* - The process of deploying an edge application or service must be as easy as sending a spreadsheet by email. At the same time, applications must be deployed in a properly configured server running at a data center and the application sources must be stored in a configuration management repository for future update, management, traceability, and eventual knowledge transfer.
- *Multi-tenancy remote development and deployment* - Push-button deployment technology must be available to departmental developers and application managers so that they can safely deploy applications into a production environment, without compromising other applications, or requiring detailed IT supervision. In the same way, IT must be able to rearrange, migrate, and distribute edge applications with no downtime.

Understanding the benefits of shadow IT and the value of its resulting edge applications can bring an organization to the next level in its IT evolution. It has a positive impact on the bottom line, especially when these applications can be incorporated as part of an SOA infrastructure. Such understanding allows the entire organization to leverage application innovation whether it forms in central IT or at the edges.

On the other hand, ignoring shadow IT incurs the risk of hindering your SOA evolution, overlooking potential application innovations that could streamline business process and increase revenue, and increasing expenditures in

the long run due to retrofitting, and also potential security and governance breaches.

Once you determine that shadow IT is a reality in your organization, a common question arises: How do you leverage your central IT SOA environment to facilitate these edge applications without bringing your central IT infrastructure to its knees? There are four key rules that need to be followed, as explained in the remainder of this article.

Rule 1: Provide Functionality Needed by Business Managers ASAP

Your SOA environment has to be ready to provide new functionalities for line of business users through the delivery of new services, while not becoming a bottleneck. Therefore, the very first challenge relates to the time-to-market of new services.

Edge applications that tie into a larger SOA infrastructure will require central IT to deliver services. This can cause friction as it makes your business-line units dependent on central IT. Therefore central IT will take the blame if its SOA environment is not ready to expose new services quickly and cost effectively. More importantly, if they can't make it happen, shadow IT will find a work around and create yet another set of rogue applications. This is the start of a vicious circle that you don't want to be pulled into.

How fast can you deploy new services in your SOA environment, and at what cost?

Here's a good way to find out how agile your deployment is: measure the cost of exposing new services for a business rule (from a process already implemented in one of your core systems) or a service for information you already have in a database.

For instance, SAP allows you to expose any BAPI - that is, any business object or piece of logic already available in the system - as a Web service, without requiring any change to the system. Other systems, such as BEA's AquaLogic Data Services Platform, allow you to expose an existing data service as a Web service.

Creating these services is possible, but you need to figure in extra budget and time to get it done. For SAP, evaluate how fast you can build a custom BAPI to answer a new specific need, and make sure you have skilled resources to do so (and how much this will cost). In BEA's case, consider how quickly you can create a new data service that queries a specific business object, and factor in available resources and time needed.

Rule 2: Don't Stress Core Systems Until They Collapse

Once you provide shadow IT the requested services, you need to assure their quality of service and the SLAs of your core systems, which are now serving requests to the outside.

The reason is that shadow IT systems may initially be used by only a couple of end-users but can then be rapidly adopted by an entire department. This means that if the services access your core systems directly, your central system may suddenly start receiving hundreds of requests per second. At best, an onslaught of requests will decrease your core systems' responsiveness, making them unusable by your shadow IT and any other systems dependent on them. In the worst case, this can overload your core systems and cause them to collapse.

Returning to the SAP scenario, assuming you expose any existing BAPI through a Web service, each service invocation will open a transaction inside SAP. With a large number of requests you may simply run out of transactions. For example, the system would be unable to run the transactions needed to bill your customers!

Network traffic is another aspect that can seriously affect your SLAs, depending on the way these services are built, exposed and used. If your shadow IT systems do not cache answers from your central SOA, they probably will flood your network with bulk information returned from core systems, and eventually render it unusable to other systems.

Is your SOA environment ready to deliver new services with no SLA impacts on your core systems?

If you don't know the answer to this question, you will need to perform some tests over your infrastructure to see how it scales. It is advisable to create architecture to cope with this performance problem.

For instance, IBM endorses the FastSOA approach that relies on a middle-tier cache, based on a relational database and XQuery between your services and their consumers- that is, between your central IT SOA environment and the

shadow IT. There are some products in the market based on this architecture.

And there you have it. Yet another system to fit in your architecture, with additional overheads compounded by an already tight budget and limited IT resources. Whether you buy a product to solve this performance issue or rely on in-house development, you will have additional costs and require additional skills. Either way, keep Rule 1 in mind: "Will this extra layer affect the time-to-market of exposing new services?"

Rule 3: Govern Access to Exposed Services

A highly-debated issue with SOA relates to exercising control over services. Once you expose services to your shadow IT, you must somehow ensure that only granted systems can access them. This particular SOA governance concern can be solved in many ways.

The lower level solution for this problem is to exercise control at the network level, by defining which machines can access which services. This may not be enough due to the simple fact that a departmental server can host multiple systems, each one with different purposes and privileges. Organizations are forced to leave the network level control and move to an application level protection. However, this implies all your services now need an extra layer to authorize the system accessing a service.

If your SOA implementation is based on Web services and you started addressing this concern early, chances are you created your own SOAP header-based control layer and are now struggling with interoperability issues. Even with an upgrade to a WS-Security based framework, it's probable that your time-to-market for new services has increased.

Can you control which systems access which services?

The answer should be a sound yes! At the end of the day, you need this control, so you can either do it yourself or you use auxiliary tools.

Several products in the market can help you in the task of authenticating and authorizing access to your Web services. With such solutions, your services require clients to include authentication and authorization tokens in all requests, pretty much like a WS-Security based control. The advantage these products introduce is providing you with the technical framework to add this extra level of security to your services and manage the access rights. Keep in mind that auxiliary tools will introduce new costs and may affect the time-to-market of exposing new services.

You'll also need to figure out how to integrate this security layer with your SOA. This extra layer of complexity can affect your central IT and take a hit on your shadow IT agility.

Rule 4: Secure Exposed Information

More important than just controlling which systems can access which services, is controlling which entities in your organization can access and change the information that is now accessible through your edge applications. This is exactly the same concern we covered previously, but transposed to the entity-level rather than the system-level.

Another common requirement is to protect your strategic information from other organizations and compliance with industry regulations, like Sarbanes-Oxley. Here, the problem of how to control which employees can access and change which information is particularly relevant in SOA environments. Since such environments thrive on loosely coupled, dispersed services, it is necessary to introduce another layer to your SOA environment to cope with fine-grained governance issues.

Essentially, this layer can replace the one discussed in Rule 3, because with such fine-grained access control there is no need to control which systems access which services, since the entity is what matters. However, most organizations never reach this level of control because it costs too much to deploy and maintain.

The trick is to include this planning from the very beginning to avoid the higher cost to implement it down the road. Cost savings can be factored even when implementation is human-based and not technology-based, and supported by operational procedures and training.

Can you transitively assure authentication and authorization once you lose control over the information?

A good way to address this is to audit your systems and trace a set of accessed or changed pieces of information back to whoever accessed or changed it. Regardless of the answer, you must be aware that there are many different ways to deal with this. The easiest way is to use and integrate specialized technologies. Sun's Access Manager is a good example of a solution that offers this capability. Besides providing a series of Web services to query access control rules, it allows you to control authorization rules over Web applications without changing them at all. This is achieved by controlling which users can access which URLs inside your organization.

One again, keep in mind that the solution you choose may introduce lag into your shadow IT applications, since your central IT controls access to the applications being changed at the edge. Also consider that this layer of security does nothing to secure evidence of compliance or traceability of access/changes to specific information.

Conclusion

Shadow IT serves a purpose, and given that your central IT will likely be adopting SOA, you have the opportunity to avoid the problems of the past and ensure that these two worlds effectively join efforts to deliver the maximum value to your business.

Because of the volatile characteristics of edge applications, shadow IT poses particular challenges to SOA. You should be aware of these challenges and at best solve them proactively to maximize the value delivered to business and minimize future costs.

In summary, your service-oriented enterprise must be able to:

- Sustain short time-to-market of new services;
- Support unexpected and variable heavy load on the services and core systems;
- Control authentication and authorization over services;
- Assure compliance rules on the services usage, as well as transitivity of those rules.

Finally, when choosing the right technology to sustain shadow IT, be sure to look for knowledge transfer features, a scalable and robust architecture, built-to-change characteristics, and inherent support for reusability and refactoring with low cost and risk.

THE PRENTICE HALL SERVICE-ORIENTED COMPUTING SERIES FROM THOMAS ERL

