



White Paper

Loosely Coupled Sourcing & Closely Collaborative Sourcing

Some of the problems after sourcing IT are caused by overestimating the ability to formalise the division of labour between the various internal and external parties involved. This paper explores this domain and emphasizes the need for mutually closely collaborating sourcing partners to effectively deal with the inevitable gray areas in your information system landscape. An effective sourcing strategy is 'loosely coupled if possible, and closely collaborative if not'.

Mark Smalley, 20 November 2010



The Chain has changed

In my IT Management Consultancy practice I've recently noticed that organizations are struggling with dividing the IT Service Supply Chain into parts that can either be outsourced or given to an internal department to perform. This paper explores the underlying sourcing principles.

The difficulty is in how activities can be divided into packets of activities that can then be delegated to internal or external departments and how these departments are going to interact. This didn't used to be much of a problem – you'd usually settle for a division between Business Information Management on the demand side and Application Management on the supply side, with Infrastructure Management behind Application Management, supplying the environments (production, acceptance, test and development/maintenance) that Application Management had specified. So why are people currently struggling, what's changed? I'm going to explore the demand-supply divide another time and now focus on the division of labour within the IT Service Supply Chain.

“It's a SLA Jim, but not as we know it”

My first comment is that we should be calling it the IT Service Supply Network, because that's one of the changes. Where we often had one apps department and one data center with an Operational Level Agreement or an Underpinning Contract between the two parties, we now seem to have a more complex network of parties with differing kinds of relationships. Take a retailer who uses Google Maps as part of an application that tells their customers where their shops are located. Google executes Application Management and Infrastructure Management and supplies Software as a Service. Have you got a traditional Service Level Agreement with Google? Well, you've clicked on “I Agree” so I guess that's an agreement but it's not what you're used to dealing with. And the relationship with Google is different. You're used to telling the supplier what to do but with Google it's different. You either take it on their terms or not. Can you live with that? Can you explain to your customer that you've contracted a party that won't guarantee much in the way of availability and performance? Think about the relationship with Google in terms of trust. Sure, it's bad luck for the retailer if Google Maps isn't available but Google probably has more to lose in terms of reputation. So you take up the offer.



Close Cooperation and Mutual Adjustment

Another reassurance in this kind of relationship is the ability to replace a component by another component. If Google fails to deliver, it's relatively easy to replace their Maps by an alternative. The interfaces are standardised. I'll repeat that. The interfaces are standardised. Remember the good old design principles of tight binding and loose coupling? Yep, it still works. So we're moving towards a world in which we compose information systems out of loosely coupled components with loosely couples organizations that deliver and support the components. But it's not as simple as that because there are still plenty of areas that are tightly coupled, both in terms of system components that are interwoven and similarly, organizations that support the components. Where some activities (like programming) can be performed in splendid isolation from other organizations, other activities require close cooperation and mutual adjustment between organizations. To take a common example, it's just not feasible (unless it's a very simple technical environment) to think that you can draw a simple line between applications and infrastructure and give responsibility for the apps to one party and infrastructure to another party and expect them to work effectively by executing their mutual contractual responsibilities.

Agnostic Artists and Intelligent Integrators

Let's take a side step and have another look at IT organizations. I think they are dividing themselves up into two groups. We're all aware of the standard solutions that the clouds are starting to provide, e.g. Google Docs and Salesforce.com, and a multitude of apps for smartphones.

These solutions are provided by clever companies that do not know their individual customers. The market, yes; customers, no. I call them agnostic artists. Agnostic because they don't know their individual customers; artists because they make great products. If you buy one of these products it's up to you to fit it into your application landscape and provide the necessary interfacing with other systems. You've also got to monitor – particularly with smaller companies – the supplier's continuity. And whether their product isn't being overtaken by another agnostic artist's product.

Another challenge is managing agnostic artists. The familiar 'command & control' management paradigm won't work. 'Communicate & collaborate' will. For instance using Twitter to muster up other users of Google Maps and apply community pressure. That'll work. The roles that I've just described – architect, interface manager, supplier manager, market monitor, contractor, coordinator – are part of the other kind of IT organization that I see. These I call them intelligent integrators. Where agnostic artists have a one to very many relationship with their clients, the intelligent integrators have one to one relationships. They're often slimmed down internal IT departments.



Ignore the gray area at your peril

With sourcing there's always a gray area and you ignore it at your peril. The bigger the gray area and the more critical the applications, the more attention you'll have to pay to engaging organizations that not only are competent in executing their core responsibilities but also proficient in interacting with other organizations. If you engage two organizations and just expect them to get along with each other, you might get more than you expected. And I don't mean quality of service.

Guidance - Gray analysis

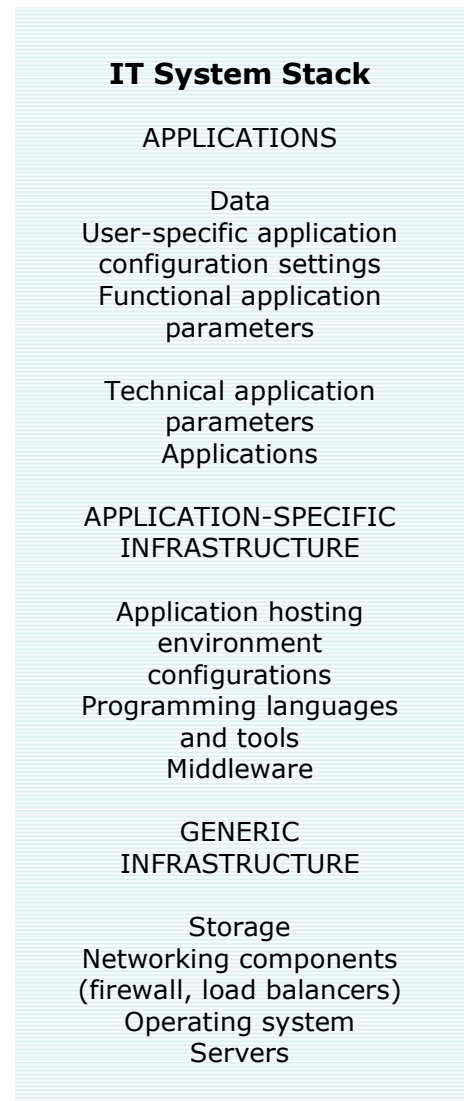
So what guidance can you follow? First take the trouble to analyse the 'stack' of IT components in the information system or systems in terms of who is responsible for what. You'll find areas that are obviously the responsibility of one party (for instance, Application Management is responsible for programming, and shutting down servers is an unquestionable responsibility of Infrastructure Management) but there'll be areas where several parties are involved (for instance, setting the parameters for middleware components like an Enterprise Service Bus). The latter areas are the trouble makers.

Headache prevents stomach ache

RACI matrices are very useful when organizing the IT management function and fathoming out who is responsible for what. A variation that I like to use is a RACI matrix with four columns:

- System component
- Business Information Management (Demand)
- Application management (Supply)
- Infrastructure management (Supply)

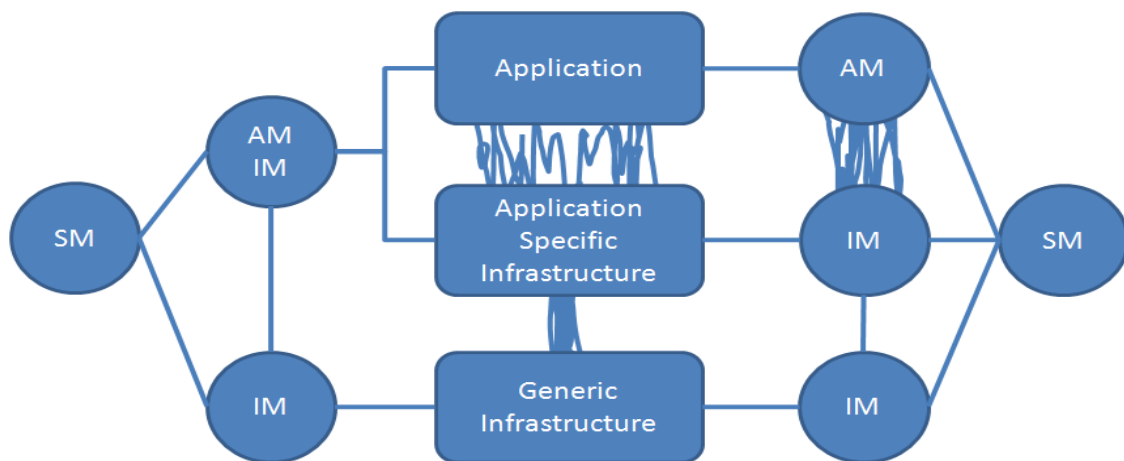
I get people to list the stuff that comprises an information system, like application modules, scripts, databases, user manuals etc. and use the BIM/AM/IM cells to fill in the RACI responsibilities, e.g. R:Roger, A:Arthur, C:Clara, I:Ingrid. You usually end up with a pretty long list and I give people a packet of aspirin because it's usually gives them a headache working out the responsibilities. But it does prevent stomach ache further down the road because sooner or later you're going to have to know who's responsible for what.





Close Collaboration or Loose Coupling

Secondly, don't try and over-formalize the responsibilities in these areas but identify them as gray areas and get both (or more) parties to agree to a participatory approach. And if you think they wouldn't be able to collaborate, either replace them by organizations with more cooperative cultures or redefine the borders so that you've given the gray areas to one organization. A strategy I'm seeing emerging is making a distinction between the generic infrastructure that is loosely coupled with the application layer and the application specific infrastructure. I'd call hardware, operating systems, firewalls etc generic infrastructure and something like a Microsoft .NET Framework and SQL Server application specific. Sure, installing a .NET Framework and a database is formally defined as 'Infrastructure Management' but I'd feel more confident giving this to the same organization that's responsible for Application Management than to a hosting provider who specialises in providing generic platforms. Like most of the Cloud Computing providers.



In the diagram two constructions for organizing IT Management are depicted.

The system stack in the middle is comprised of generic infrastructure with a 'clean' interface with the application specific infrastructure, which in its turn has complex interdependencies with the application layer.

The organizational construction on the left hand side depicts a Service Management responsibility that has delegated application management and management of the application specific infrastructure to one party and management of the generic infrastructure to another. The complex interface between the application layer and the application specific infrastructure layer is dealt with by the first party, relieving service management of dealing with conflicts between the two subcontractors.

In the other organizational construction, service management has chosen three subcontractors and has selected an application management subcontractor and application specific infrastructure management contractor who can closely collaborate with each other, therefore reducing the need to act as an intermediary.



Governance of the Gray

Let's look in more detail at the gray areas, which are illustrated in the chart below. Just reading this will probably give you a headache but all of this has to be addressed unless you're happy to take a risk and correct things on the fly.

	Develop & Maintain	Deploy	Operate
Application			
Application Specific Infrastructure			
Generic Infrastructure			

The well accepted areas are

- Application Development and Application Maintenance that for the sake of simplicity, although often separate organizations, have been combined into the top left cell. This starts with requirements that have been provided by the user organization, translating them into specifications for applications that are developed or acquired and then set-up for the organization, testing and finally handing over the first version application (development) or the new releases (maintenance) for deployment to production. In the case of maintenance, the releases often comprise solutions for problems that slipped through testing and have occurred during use of the application.
- Infrastructure Management (bottom row), comprising all activities required to provide a generic infrastructure on 'top' of which infrastructural software components are added to create technical environments in which applications can be developed, maintained, tested, accepted and used.

The gray areas are (from top to bottom and left to right):

1. Deploy Application – transition from the development/maintenance environment to production; installation and production testing
2. Operate Application – monitoring availability and performance at an application level and taking operational measures or requesting a change
3. Coordinate changes in Application and changes in Application Specific Infrastructure – synchronizing the autonomous but interrelated lifecycles
4. Coordinate deployment of Application and deployment of Application Specific Infrastructure – touching base and mutual adjustment



5. Coordinate operation of Application and operation of Application Specific Infrastructure – sharing information about operational status
6. Define Application Specific Infrastructure – explicitly specifying the components in this domain
7. Develop & Maintain Application Specific Infrastructure – specification, acquisition, installation, setting-up, dealing with updates
8. Deploy Application Specific Infrastructure – transition from the development/maintenance environment to production; installation and production testing
9. Operate Application Specific Infrastructure – monitoring availability and performance of these components and taking operational measures or requesting a change
10. Coordinate changes in Application Specific Infrastructure and changes in Generic Infrastructure – synchronizing the autonomous but interrelated lifecycles
11. Coordinate deployment of Application Specific Infrastructure and deployment of Generic Infrastructure – touching base and mutual adjustment
12. Coordinate operation of Application Specific Infrastructure and operation of Generic Infrastructure – sharing information about operational status

Sometimes the term Technical Application Management is used to (partially) define the 'gray' activities, but the term does not have a formal definition and is in practice used with different interpretations. To avoid confusion it is therefore prudent to specify which activities and which components are being addressed.

Governing the IT Supply Chain/Network

As IT landscapes are breaking down into more discrete components and IT Service Providers are specializing in their own sweetspots and becoming agnostic artists, organizing the IT Supply Chain (or Network) is becoming increasingly important. This is reflected in the changes that were implemented in the second version of ASL with the introduction of the processes Supplier Definition, Supplier Management and Operations Management, covering governance and management of suppliers; and the repositioning and renaming of Service Level Management into Contract Management, reflecting that an Application Management provider not only can be engaged directly by Business Information Management in the demand side, but also as a subcontractor by another Application Management provider on the supply side.

The guidance in this area is to critically re-examine how the chain is organized and determine whether it is still an effective instrument. So when you're making sourcing decisions about complex landscapes, please analyse the gray areas and decide on the division between labour based on your evaluation of the need for close collaboration and mutual adjustment and the ability of suppliers to perform in this area. Nowadays sourcing strategy is more about creating a network of loosely coupled interactions than anything else. Just like the organizational design principle of 'decentralised if possible, otherwise centralized', here it's 'loosely coupled sourcing if possible, closely collaborative if not'.



Author

Mark Smalley is director of international affairs at not-for-profit ASL BiSL Foundation. He publishes and speaks about Application Management and related topics (ASL, BiSL, IT Governance, Business IT Alignment) on a regular basis and has reached out to several thousand people in four continents.

He works as an IT Management Consultant for Capgemini in the Netherlands and contributes to EXIN certification material. He also lectures in Brussels, Hangzhou and Rotterdam.



Further details, publications and speaking engagements at <http://www.linkedin.com/in/marksmalley>.

CONTACT

aslbisl foundation

Mark Smalley

International Affairs

Goeman Borgesiuslaan 77 / PO Box 9769, 3506 GT Utrecht, The Netherlands
T: +31 (0)30 6632293 / M: +31 (0)6 53464157
mark.smalley@aslbislfoundation.org / www.aslbislfoundation.org