



## Position Paper

# Architecture for De-perimeterisation

### Problem

De-perimeterisation is the term coined by the Jericho Forum to describe the erosion of the traditional 'secure' perimeters, or 'network boundaries', as a mediators of trust and security. Such network boundaries are more than just physical perimeters – they often mark the conceptual beginning and end of an organisation, entity or enterprise.

Traditionally, 'architecture' at the enterprise level is "the fundamental organisation of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution<sup>1</sup>". It is expressed as various viewpoints to understand the information and human activity systems needed to fulfil its purpose. Through architecture frameworks such as the Zachman Framework<sup>2</sup> and TOGAF<sup>3</sup> these viewpoints are then used to select technologies, define and develop the applications and systems required.

But if boundaries are eroding, and one organisation merges into another, how should we approach 'architecture'? Are existing approaches invalidated by de-perimeterisation? Will costly re-engineering be needed?

### Why Should I Care

Today's successful enterprises are structured to be adaptable to market changes with regard to people, process and technology. If the information systems and processes that support the enterprise cannot adapt easily, in order to enable the enterprise to adapt, then the enterprise will lose its position in the market. Rigid architectures can be adapted, but at a cost of time, capability and money.

By implementing an adaptive de-perimeterised architecture, the enterprise can quickly and safely adapt to the structure required to benefit from new opportunities.

---

<sup>1</sup> ANSI/IEEE Std 1471-2000

<sup>2</sup> <http://www.zifa.com>

<sup>3</sup> <http://www.opengroup.org/architecture>

## The Jericho Forum Response

The Jericho Forum's vision is for enterprises and individuals ultimately to be able to sustain their information and communication technology needs on the Internet itself (JFC#5<sup>4</sup>) – through a combination of appropriate trust models (JFC#6 & 7), inherently secure protocols (JFC#4), endpoint security and so on (JFC#2). This vision emphasizes application autonomy, platform survivability and network transparency.

The vision takes the idea of the 'extended' or 'virtual' enterprise to its ultimate conclusion. While it is perfectly possible to model any type of enterprise, process, or system using existing architectural techniques, these tend to guide practitioners towards established computational and design models such as centralised information storage and client-server configurations. They feature layering of services such that common functions are factored into lower layers so that upper layers can be specialized to the needs of particular users and business processes.

In reality this implicitly optimizes the architecture for one set of users/applications while substantially INCREASING its cost for a future set of valuable usages that may be unknown, or unpredictable, at design time through being too rigid. Even the ubiquitous Web is no more than an implementation of multiple interlinked client-server systems. A contemporary example that tries to address de-perimeterisation with minimal impact on existing systems is to introduce a series of Enterprise Portals<sup>5</sup> corresponding to the various elements of inter-organisational business processes as depicted below.

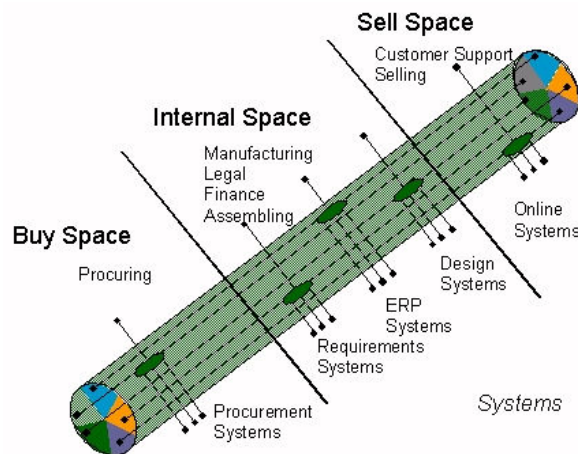


Figure 1 TOGAF Portal Reference Model for Interoperable Enterprise Business Scenario<sup>6</sup>

What analysts and developers need is a way of translating new business requirements that reflect de-perimeterisation into designs that remove hidden dependencies on existing design models, and make explicit how dependencies on these models may evolve and transform.

<sup>4</sup> The term JFC#n refers to the relevant Jericho Forum Commandment number. See [www.jerichoforum.org](http://www.jerichoforum.org)

<sup>5</sup> See [http://en.wikipedia.org/wiki/Enterprise\\_portals](http://en.wikipedia.org/wiki/Enterprise_portals)

<sup>6</sup> See <http://www.opengroup.org/architecture/togaf8-doc/arch/p3/iii-rm/concepts.htm> for explanation.

## Background & Rationale

In a traditional perimeterised architecture, the drive is towards centralized and hierarchical organisation and management, and trust models are strongly coupled to this. Policy-based security seeks to establish interfaces and protocols to allow the propagation of policy rules among multiple architectural components (platforms, applications, network boxes etc.). While this approach caters for distributed components, it is predicated on a single point of ultimate authority (one true way) for the policy rules themselves, and by extension the organisation of detection and enforcement mechanisms needed to underpin other aspects of security. Where the organisational perimeter has disappeared, the single point of authority is lost, and a different approach is required.

The Service Oriented Architecture (SOA)<sup>7</sup> approach seeks to provide a unifying view of distributed systems and enterprise architecture focused on the notion of ‘services’. SOA is essentially an evolutionary approach that seeks to build on the tradition of client-server and multi-tier architectures. This is complementary to the Peer-to-Peer (P2P) and decentralised trust models that, in the Jericho Forum’s view, are necessary to address de-perimeterisation.

## Trust frameworks and management

Electronic commerce explicitly recognises that there are genuinely multiple authorities engaged in interaction and no one true way to derive policy rules. As a result, analysts and designers have had to start to consider issues of ‘trust’ and its management. After initial enthusiasm at the end of the last decade that existing technology was up to implementing ‘trust’, the realisation dawned that this was not going to be quite so easy, except to support particular types of entity, interaction and business process; and a single notion of trust.

Following this false dawn, researchers have started to develop a much more sophisticated view of trust in all its guises. Models such as Poblano, the JXTA trust framework for P2P<sup>8</sup> first introduced in 2001; the SULTAN Trust Management Framework<sup>9</sup>; and the seminal KeyNote framework<sup>10</sup>, all allow for dynamic viewpoints and flexible interaction models that can be tailored to the needs of applications and business processes. The frameworks are either entirely endpoint based or require brokers to be established to maintain common knowledge about trustworthiness/reputation and related information. The good news is that they are all substantially open and unencumbered by patents. Real-world applications are starting to appear – JXTA for example has over 12,000 registered developers, and is now in its second release, capable of deployment in large-scale applications.

## Challenges to the industry

1. Companies need to start to understand the possibilities offered by P2P trust management frameworks as the new target model for enterprise architecture. Software such as JXTA is freely available. The Jericho Forum can assist by developing business scenarios that articulate the trust models for particular patterns of interactions and entity types of business value to Jericho Forum members. This will then assist the selection and configuration of trust frameworks suitable for particular applications. Architecture frameworks such as TOGAF can be readily extended to retarget to the new models without disturbing the frameworks themselves.

<sup>7</sup> See [http://en.wikipedia.org/wiki/Service-Oriented\\_Architecture](http://en.wikipedia.org/wiki/Service-Oriented_Architecture)

<sup>8</sup> <http://www.jxta.org/docs/trust.pdf>

<sup>9</sup> <http://www.doc.ic.ac.uk/~tgrand/faq.htm#What%20is%20Trust%20Analysis?>

<sup>10</sup> <http://www.crypto.com/trustmgt/kn.html>

2. There is a need to develop new trust models for specific applications of P2P. For example, in grid computing we still don't know how to deal satisfactorily with potentially hostile processing elements, nor how to preserve confidentiality of the intellectual property associated with grid applications in the case where the grid spans the Internet or multiple companies. The current Open Grid Security Architecture (OGSA)<sup>11</sup> is based on a centralised hierarchical notion of trust, which ties control and ownership of grid resources and users back to a common or perhaps federated authority. The model requires all users and grid elements to be known, registered and certified so they can be strongly authenticated before computation starts, i.e. dynamic acquisition of components does not sit well with the trust model. Given the computational overhead of Web Services this security model is also ill-suited to computational grids that need relatively lightweight communication sub-systems to achieve massive parallelism efficiently. As a result, while the computational grids are a cost effective way to build yourself a supercomputer, if your application needs to apply several supercomputers and you don't trust other people's supercomputers to process your computations correctly and confidentially, you just have to buy more yourself. Xeerkat<sup>12</sup>, a JXTA based computational grid framework, may provide a starting point as depicted below.

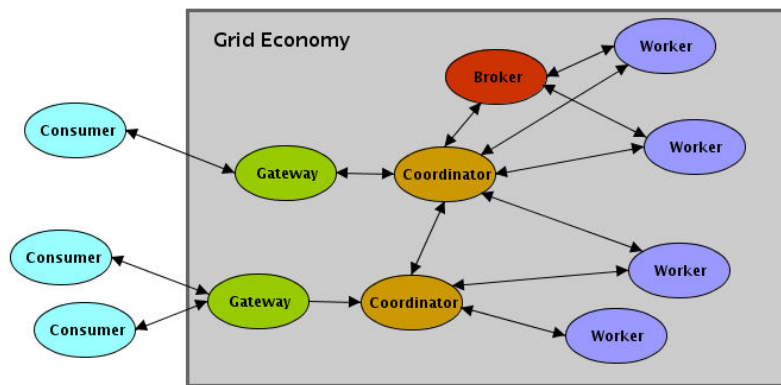


Figure 2 Xeerkat P2P model

3. The frameworks themselves can be used to underpin the management of secure credentials, for example the Jericho Forum blind public key proposals to adapt standard PKI technology to non-hierarchical business relationships.

## The way forward

The Jericho Forum believes that it is realistic for organisations to start trialling and adopting architectures that support decentralised trust frameworks and P2P applications based around them. This will represent a concrete step towards addressing de-perimeterisation cost-effectively.

<sup>11</sup> See <http://www.cs.virginia.edu/~humphrey/ogsa-sec-wg/>

<sup>12</sup> <https://xeerkat.dev.java.net/>