

# Tom Graves / Tetradian

Random ramblings over the metaphoric edge

[Home](#) [About](#) [Tom Graves](#) [Tetradian](#) [Publishing](#) [\(admin\)](#)

[Home](#) › [Business](#) › Towards a whole-enterprise architecture standard – 2: Core

## Towards a whole-enterprise architecture standard – 2: Core

Posted on [25 April 2016](#) by [Tom G](#) — [2 Comments](#) ↓

For a **viable enterprise architecture** [EA], now and into the future, *we need frameworks, methods and tools that can support the EA discipline's needs.*

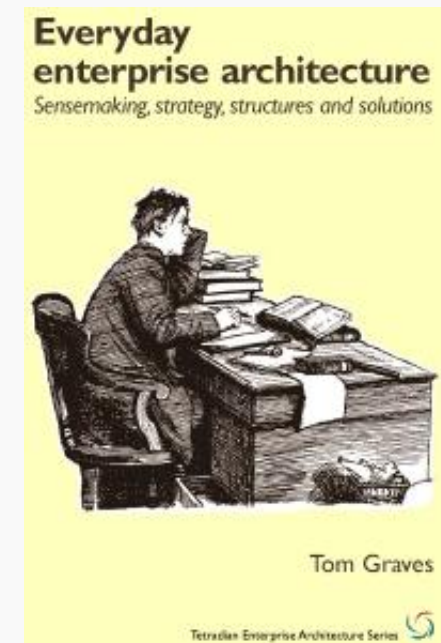
This is Part 2 of a six-part series on proposals towards an enterprise-architecture framework and standard for whole-of-enterprise architecture:

1. [Introduction](#)
2. **Core concepts**
3. Core method
4. Content-frameworks and add-in methods
5. Practices and toolsets
6. Training and education

This part explores core concepts for whole-enterprise architecture, and the implied support required within frameworks and methods.

Go

### Books by Tom Graves



Mapping the Enterprise

# Rationale

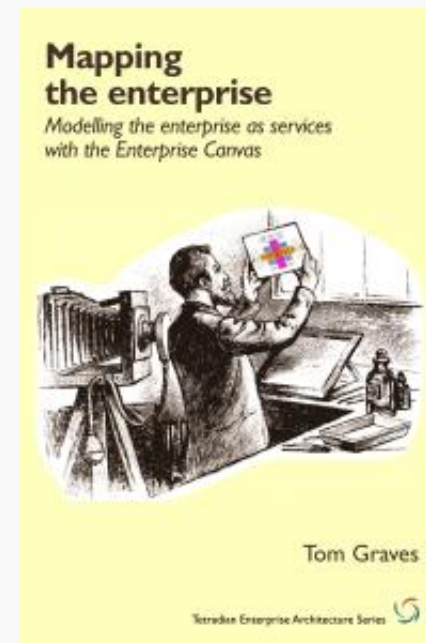
On **frameworks**, perhaps the best place to begin this would be a longish quote from Chris Lockhart's brilliant '[Frankenframeworks](#)' post, back in 2012:

*Frameworks should be easy to interpret, fast to deploy, amenable to adaptation. They can be picked up and used or tossed aside in favor of something that is a better fit. The framework shouldn't fall apart if you decide to remove part of it because it doesn't suit the problem at hand. The framework can be simple or sophisticated, but either way it should be extensible. Multiple frameworks can be put together or merged to become the methodology for a given situation. These Frankenframeworks can be reused. I myself have dozens of them that I reuse or partially reuse on every project I'm on. Keeping this mini-library of frameworks enables me to be agile and flexible.*

*... The business wants architecture to help it deliver value. Responsiveness, clarity and agility are more important to delivering value than spending a year modeling the world alone in a dark closet someplace.*

*To that end, a focus on applying the right framework to the right problem is pretty critical. And I don't think there's a single "right" answer.*

## Ebooks by Tom Graves



Ebook: Enterprise as Story

## Recent Posts

- [Towards a whole-enterprise architecture standard – 6: Training](#)
- [Towards a whole-enterprise architecture standard – 5: Practices and toolsets](#)
- [Towards a whole-enterprise architecture standard – 4:](#)

Chris is right: there isn't a single "right answer", or, as he puts it elsewhere, the perfect SUFFE™ or 'Single Unified Framework For Everything' in enterprise-architecture. Not in terms of *content*, anyway – the 'content-first' approach taken by just about every current mainstream EA framework.

Yet we still need some kind of 'just-enough-framework' that can hold everything together – any combination of content-oriented frameworks and methods, all 'smooshed-together' as required, according to context or need, yet still able to connect cleanly and consistently to *any other* 'smooshed-together' framework too. That's how we can make sense of an enterprise-architecture that needs to be able to cover and link between any context, any domain, any scope, any scale. That's what this series is about.

How do we do that? How could we build such a '**metaframework**'? Where do we start?

For enterprise-architecture, the simplest place to start is with the **enterprise** itself, and the **context-space** of an enterprise.

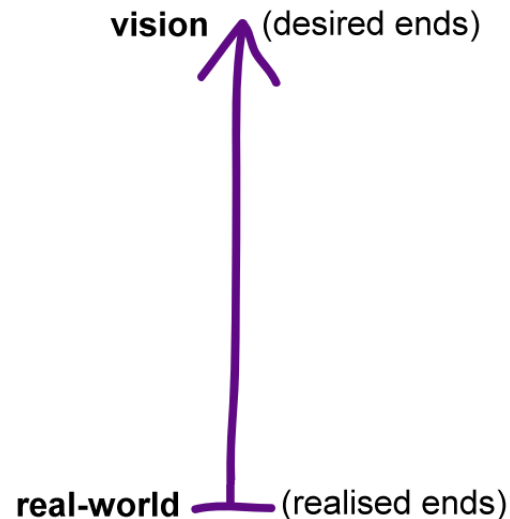
Why does anything happen, in the human world? The short answer is that there's a tension between what exists (the *realised ends*) and what we desire in the future (the *desired ends* – sometimes also described as the *vision*, or *purpose*, or *promise*):

## Content

- [Towards a whole-enterprise architecture standard – 3: Method](#)
- [Towards a whole-enterprise architecture standard – 2: Core](#)

## Recent Comments

- [Doug McDavid](#) on [Towards a whole-enterprise architecture standard – 5: Practices and toolsets](#)
- [4 Layers Of Enterprise Architecture | Architecture Fan](#) on [On layers in enterprise-architecture](#)
- [Tom G](#) on [Towards a whole-enterprise architecture standard – 3: Method](#)
- [Tom G](#) on [Towards a whole-enterprise architecture standard – 3: Method](#)
- [Jerome Abudah](#) on [Towards a whole-enterprise architecture standard – 3: Method](#)



The moment we set out to take some kind of action towards that purpose, we have an **enterprise as a 'bold endeavour'** toward the purpose. (At present, 'enterprise' is still a very *human* matter: at least, I've yet to find any computer-system that is itself either bold or endeavouring...).

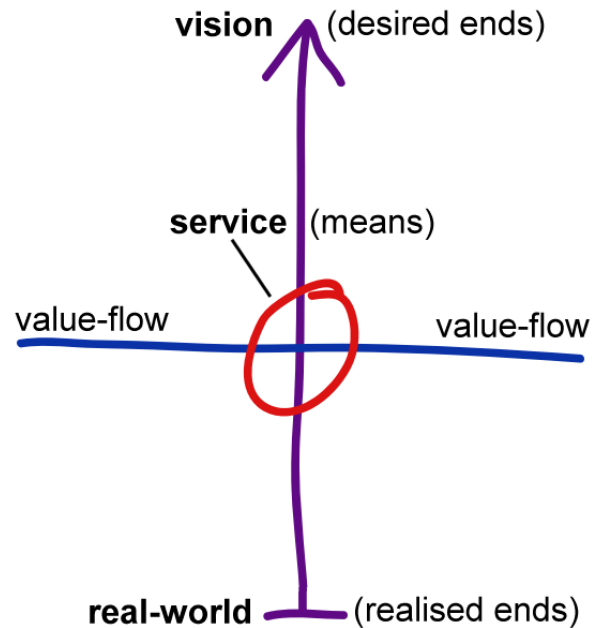
Often, we can't reach those desired-ends on our own: we need help from others. And likewise, we can help others in their endeavours. That's the basis for *shared-enterprise*, and for *services* and *organisations* to serve that shared-enterprise:

## Categories

Select Category ▾

## Tags

anarchist **Business**  
 business-IT divide  
**business**  
**architecture**  
 chaos **complexity**  
 conference culture cynefin  
 decision-making disruption  
 economics  
**effectiveness**  
**enterprise**  
 enterprise 2.0  
**Enterprise**  
**architecture**  
 enterprise canvas **Futures**  
 IT-centrism **Knowledge**  
 metamodel methodology  
 mythquake narrative  
 narrative knowledge Open Group  
**paradigm** power quality  
**RBPEA** responsibility  
**SCAN** Scribbles / writing



## sense-making

service-oriented enterprise skills

social media Society story

strategy taxonomy togaf

twitter values worldview

## Archives

Select Month ▾

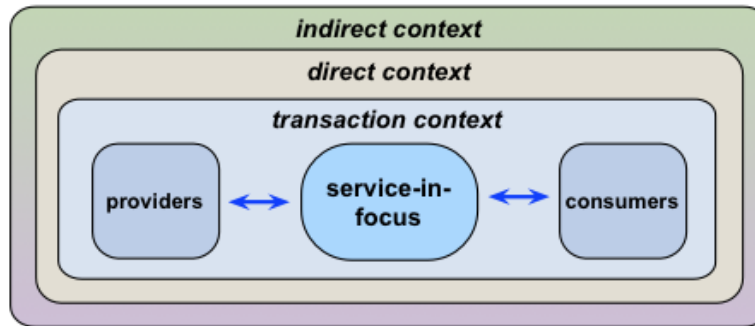
Each organisation or service forms one element within an overall **ecosystem of services**. The crucial difference with a shared-enterprise, though, is that **an enterprise is an ecosystem bound together by a shared-purpose**: we need to keep in mind that point about the centrality of purpose, in all forms of architecture.

Hence, in practice, the core focus for enterprise-architecture is to **understand the context** of the respective service or organisation-as-service, and **create structure and story** to serve the respective need. We develop an architecture *for* something, *in context of* the broader shared-enterprise for that something.

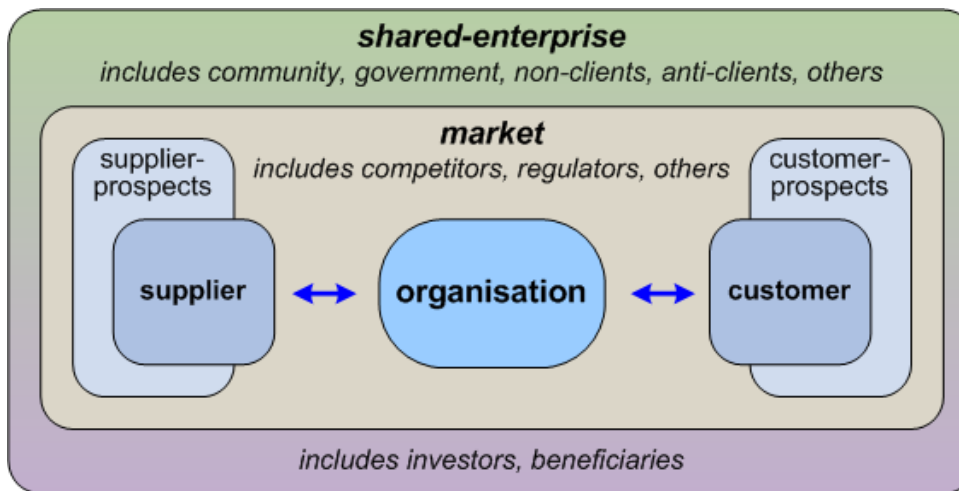
Alternatively, creating support for an enterprise consists of **an iterative dance between architecture and design**, where *architecture focuses more on the abstract or Why*, whilst *design focuses more on the concrete or How and With-What*, all in service of the enterprise as *Who*.

Every context is enclosed within a broader context – or, often, an intersection of many broader contexts. For the architecture of an enterprise, or a service within that

enterprise, we typically need to take into account *a context-scope that is at least three steps larger* than the immediate context of the service-in-focus itself:



Hence, for an organisation, the context-scope for its enterprise-architecture would extend at least to a shared-enterprise like this:

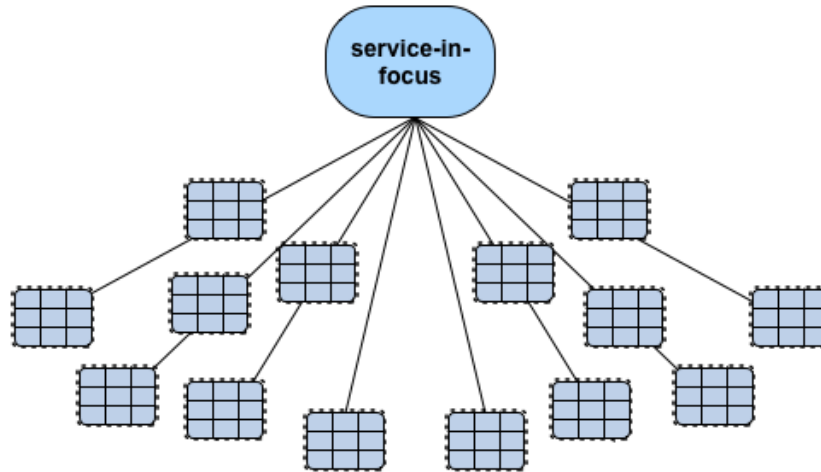


Note that whilst the service or organisation is typically shown as the centre of the respective shared-enterprise, the reality is that it is merely one of many possible or actual entities within that shared-enterprise context-space. A fundamental here is that **everything and nothing is 'the centre' of the enterprise, all at the same time.**

(That principle is also a core reason why **IT-centrism** – or any other form of

‘anything-centrism’ – is so dangerous to the viability of an architecture: it skews architectural awareness and architectural decisions into forms that are often highly dysfunctional, a long way distant from the actual need.)

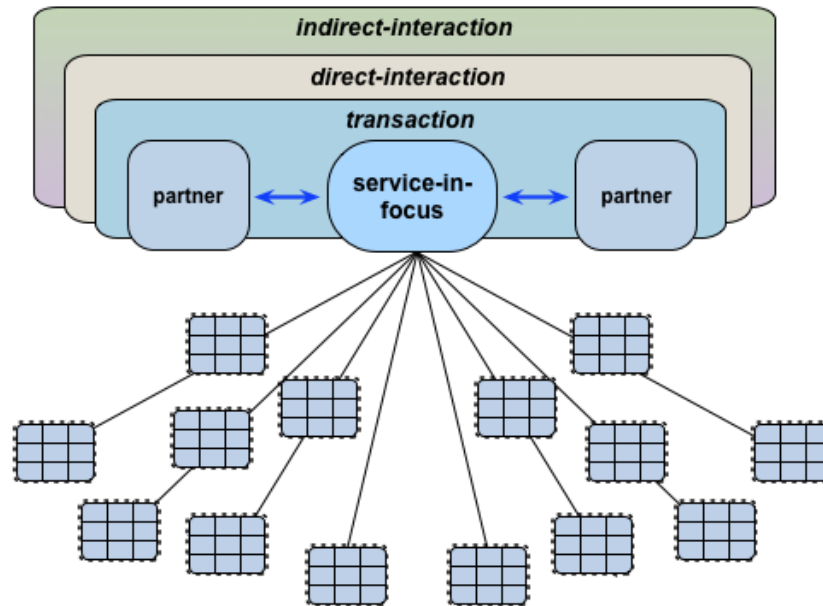
That’s the context-view ‘above’ the *service-in-focus* – the organisation or service that’s the current focus of architectural attention. Looking ‘downward’ from that service-in-focus, we would have a broad array of other services:



These ‘child-services’ may be implemented by **any combination of people, machines or IT**, and may be modelled to any required depth of detail.

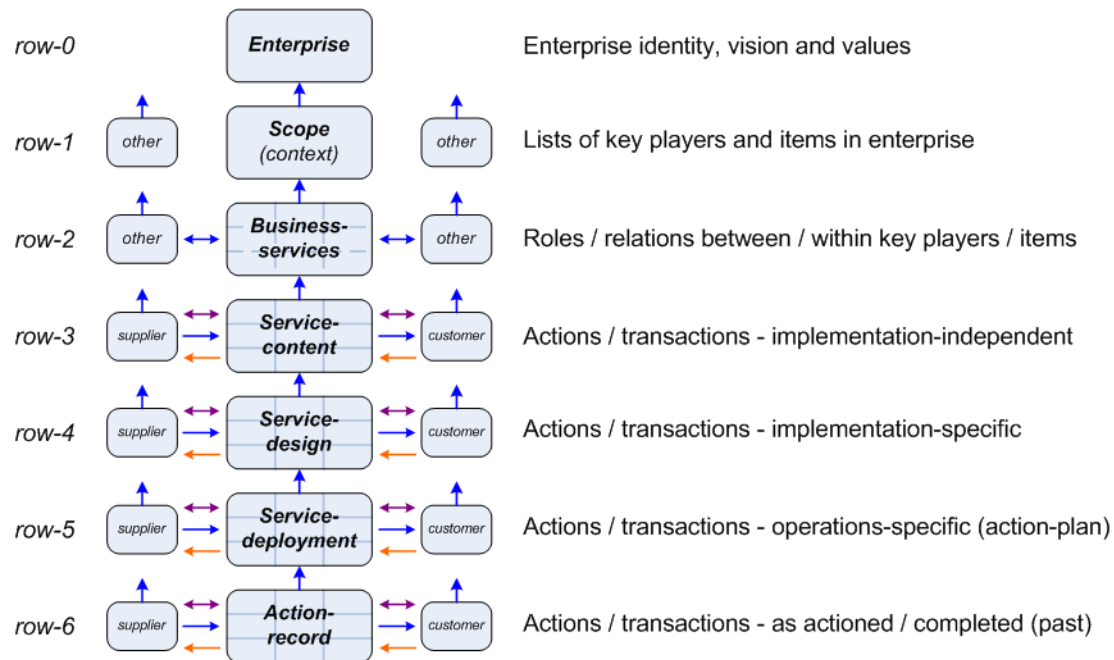
(Note that we must *not* fall into the trap so common in existing mainstream ‘EA’-frameworks, of assuming that only IT-services are relevant to implementation of the service-in-focus. That error, again, almost invariably leads to IT-centrism and increased risk of architecture-failure.)

Linking together the two sets of views – ‘upward’ and ‘downward’ from the service-in-focus – we end up with a kind of ‘jellyfish-pattern’:

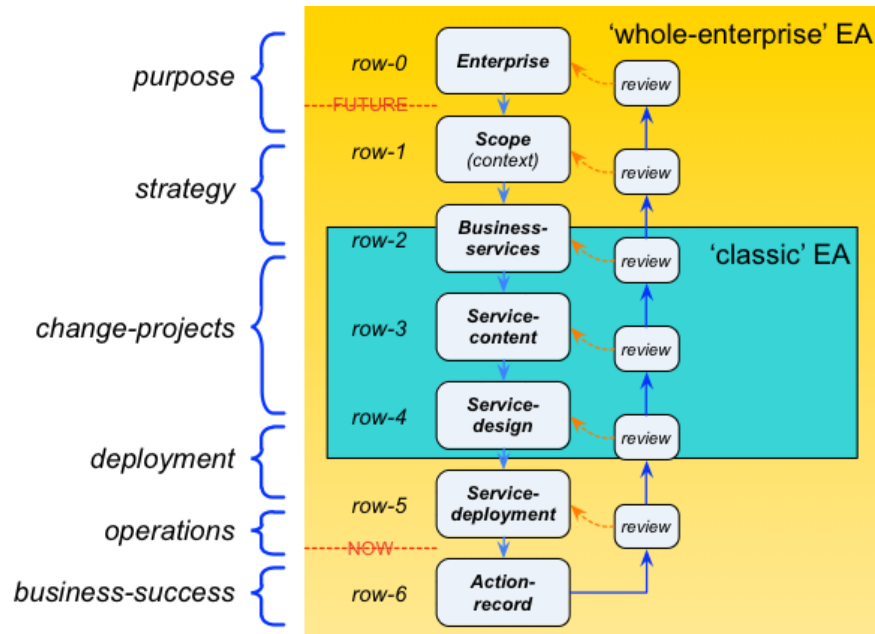


For **architecture-scope**, a whole-enterprise architecture needs to be able to link all the way *from strategy to tactics to operations*, and back again. Or, in terms of **abstract versus concrete**, the architecture needs to be *able* to cover the entire range from most-abstract to most-concrete – from the enterprise-vision right down to deployment, operation and action-records:





Even though architecture may not – and probably should not – dive often into the fine-detail of the more-concrete layers, those details are often crucially important as feedback and feedforward, to guide future architectural decisions. For comparison, ‘classical’ mainstream ‘EA’-frameworks would typically address only the mid-range of that scope, which can leave it at risk of decisions made on the basis of incomplete information both ‘above’ and ‘below’:



What we really want from an enterprise-architecture is support for **enterprise-effectiveness** – which we might summarise by the **tagline** of ***‘things work better when they work together, on purpose’***. Or, to put it another way, things are more *effective* when the work together, on purpose. (We’ll explore the practical meaning of ‘effectiveness’ later on in this series.)

All of this might seem a bit abstract at first. But by starting from the abstract, we can reframe and re instantiate it for *any* ‘language’, *any* context, *any* need. Given the enormous implicit scope of a whole-enterprise architecture, we will *need* that versatility, as a core feature of the framework: it is the means by which we break out of the ‘content-first’ trap so typical of almost all existing mainstream ‘EA’-frameworks.

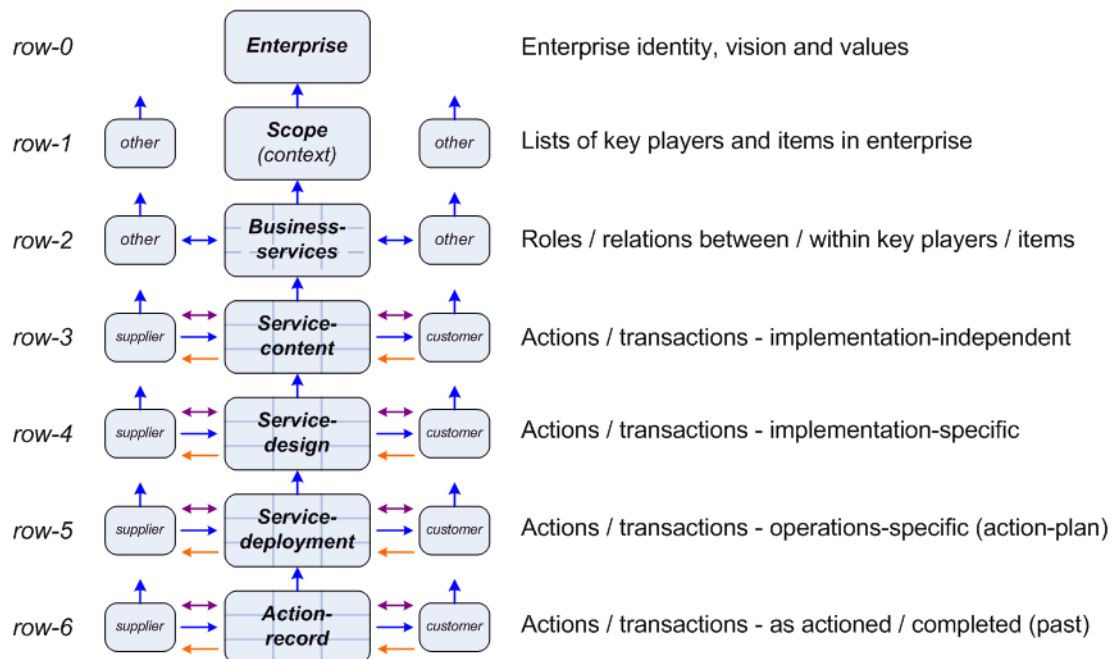
## Application

From the above, we can derive suggested text for the ‘Core concepts’ section of of a possible future standard for whole-enterprise architecture. Some of the graphics above might also be included along with the text.

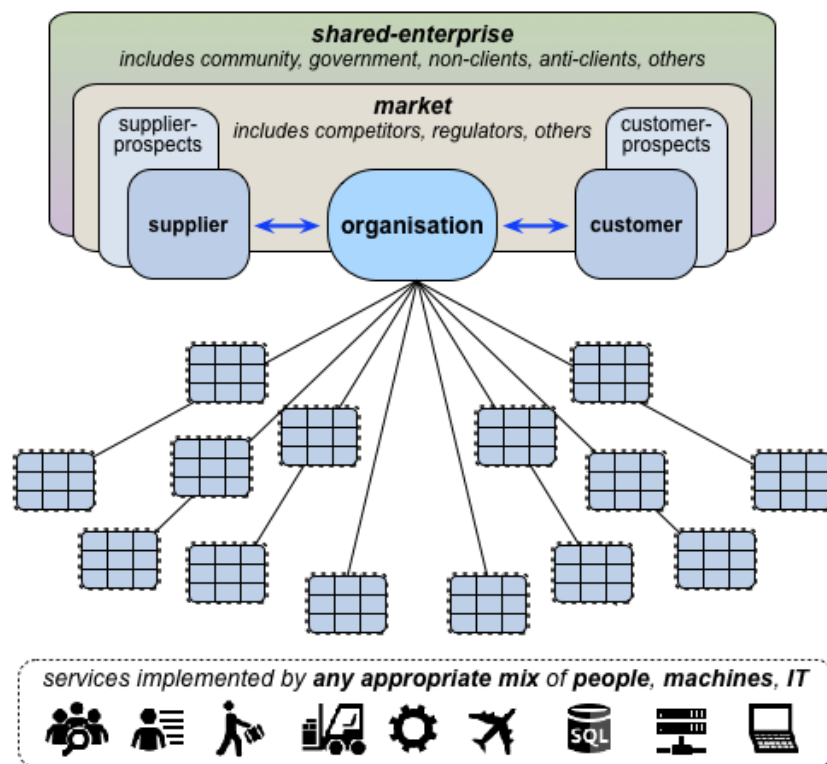
– *Purpose of whole-enterprise architecture*: Continually **enhance effectiveness** across any or all of the respective enterprise. Everything must anchor back to the vision and values of the shared-enterprise, and – for the organisation itself – the organisation's own vision, values and mission(s) in respect of that shared-enterprise vision.

– *Definition of 'enterprise'*: A human construct – a **'bold endeavour'** – around which motivation and action may coalesce. It is typically bounded by a shared-vision, shared-values and mutual commitments. (The term 'enterprise' *may* be used to denote a large commercial organisation, but that usage represents only one very small subset of human enterprises.) A typical – though not mandatory – concept of 'enterprise' in a practical sense is that it is comprised of an ecosystem of interrelated, interdependent services, bound together by a shared-purpose.

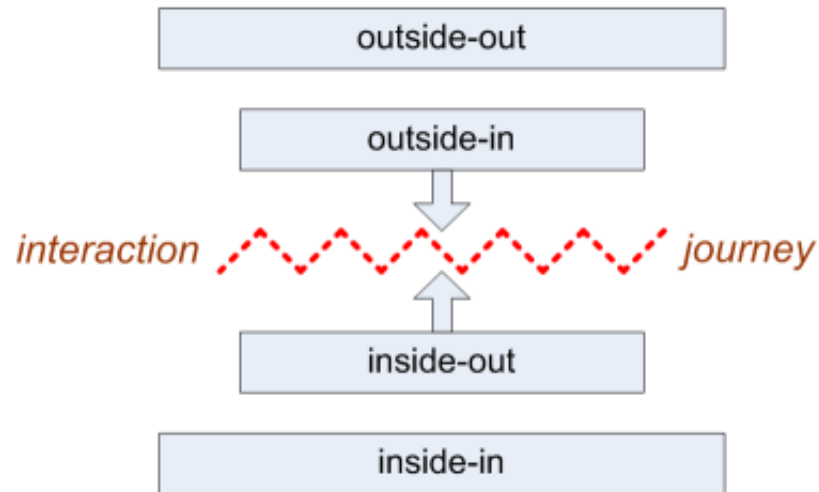
– *Scope of whole-enterprise architecture*: **Any or all of the scope of the respective enterprise**, including any subset and/or relevant intersection-supersets shared with other enterprises, and at any or all levels from most-abstract (enterprise-vision) to most-concrete (deployment, operations and action-records):



– *Content in scope for whole-enterprise architecture*: The scope may or will include services implemented by **any combination of people, machines and IT**, using any requisite types of resources. The applicable context in scope would typically be described in terms of the ‘jellyfish-pattern’ – for example, when the service-in-focus is the entire organisation, then the jellyfish-pattern for applicable scope could be summarised as follows:



– *Core activities of whole-enterprise architecture*: Identify and act on means to **link people and/or systems together** to enhance enterprise effectiveness in the required aspects of the shared-enterprise, both proactively and in response to dynamic changes in the context for the organisation (either ‘inside-in’ or ‘inside-outward’ view) or in the broader shared-enterprise and its interactions with the organisation (either ‘outside-in’ or ‘outside-out’ view):



– *Core focus of activities: **Context-first, not content-first.*** The first focus should always be on *understanding the context*, in respect of business drivers and desired business outcomes. Content-frames such as so-called ‘solutions’ should always be assessed *after* adequate exploration of the context – or, alternatively, in context-first iterative architecture-developments. (In part, the purpose here is mitigate against ‘solutioneering’, ‘vendor-driven architectures’ and similar architectural risks.)

The above provides a quick overview of the emphases that we would need in a whole-enterprise architecture. We’ll move on now to explore some ideas about how this would work in practice – what we would *do* in order to create whole-enterprise architecture.

In the meantime, any comments so far?

---

#### *Related posts:*

1. [Towards a whole-enterprise architecture standard – 1: Introduction](#)
2. [CoIT: another architectural disaster unfolds?](#)
3. [More on that enterprise-architecture 'help needed'](#)
4. [Work-in-progress – two more books](#)

## 5. Making a knowledge-base for whole-enterprise EA more accessible

◀ Towards a whole-enterprise architecture standard – 1: Introduction

Towards a whole-enterprise architecture standard – 3: Method ▶

**Tagged with:** Business, business architecture, complexity, decision-making, effectiveness, enterprise, Enterprise architecture, framework, Knowledge, metaframework, methodology, paradigm, sense-making, standards, strategy

**Posted in** Business, Complexity / Structure, Enterprise architecture, Knowledge

### 2 comments on “Towards a whole-enterprise architecture standard – 2: Core”



Jerome Abudah says:

10 May 2016 at 3:04 pm

Thanks Tom. Another great write up. My concern – “A fundamental here is that everything and nothing is ‘the centre’ of the enterprise, all at the same time.” Where exactly will EA now sit ? avoiding IT-Centrism and “anything Centrism “.

[Reply](#)



Tom G says:

10 May 2016 at 3:31 pm

Thanks, Jerome. Where it sits is kind of parallel to and intersecting with the existing strategy-unit and the PMO (or whoever does strategy and change-management these days). EA builds a picture of the structure and story of the organisation in context of the broader enterprise. Strategy then works with this picture, changing emphases to adapt to the market or whatever; EA assesses the structural and other implications of strategy, and guides change-management in the respective changes to capabilities, services or whatever. If

you take EA out of the equation, you have strategy built without proper knowledge of what the organisation is and does, and change-management built without proper linkage to strategy or awareness of impacts on overall effectiveness – both of which are depressingly common, of course...

[Reply](#)

## Leave a Reply

Your email address will not be published. Required fields are marked \*

**Name \***

**E-mail \***

**Website**

**Comment**

**Post Comment**

