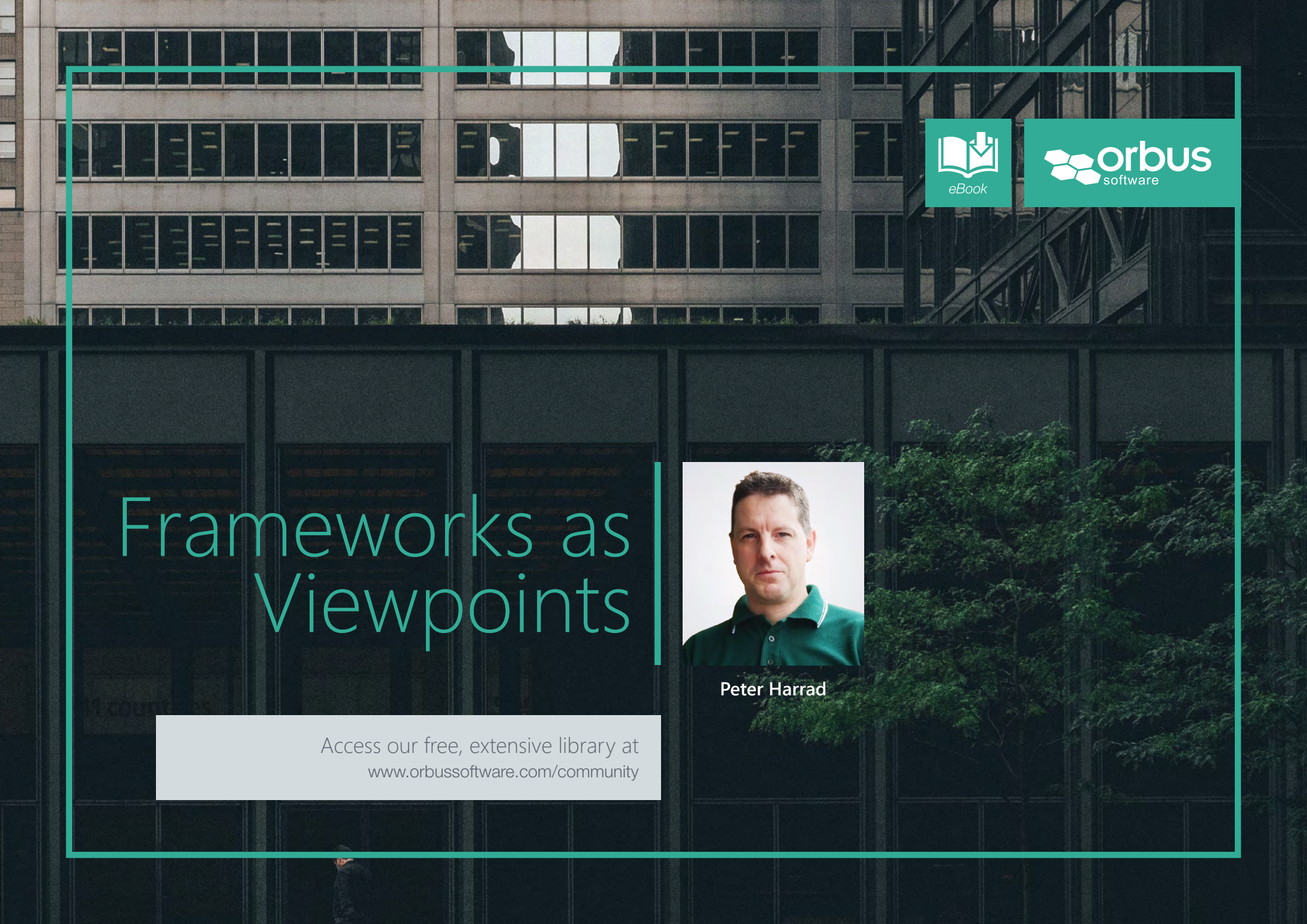


eBook

orbus software

Frameworks as Viewpoints

Peter Harrad

Frameworks as Viewpoints

One of the things about IT is the number of standards, both vendor-specific and independent, that you have to deal with. It gives rise to one of my favorite sayings –

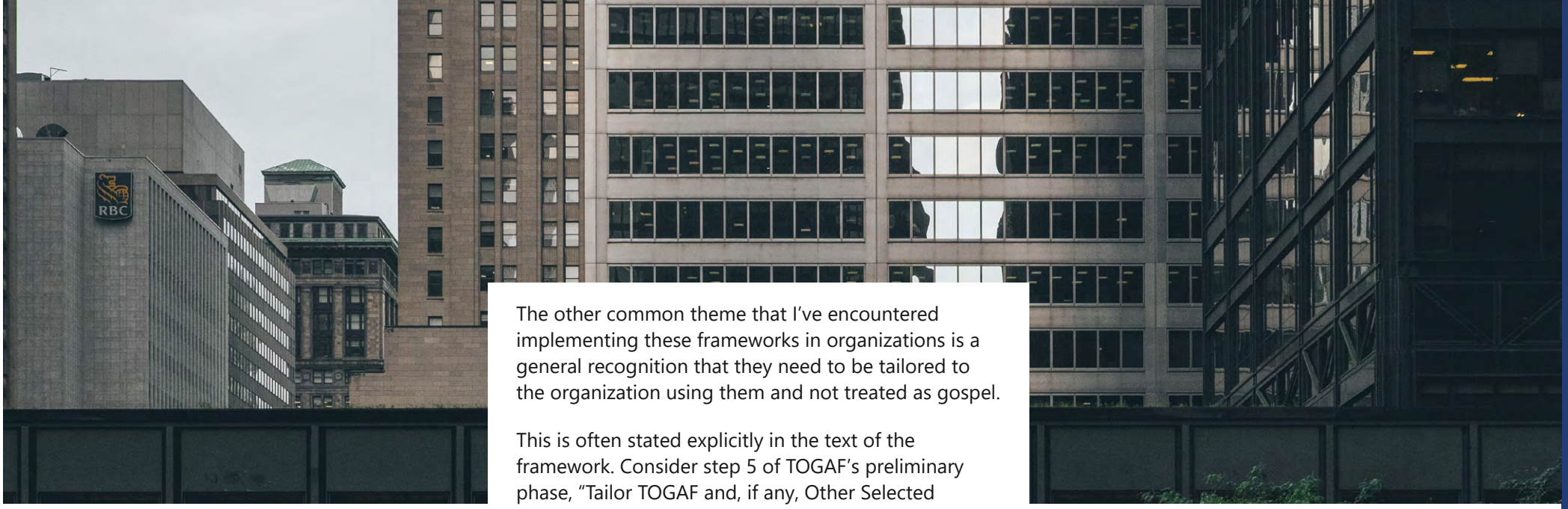
– "Standards, I love 'em. Let's have lots of them."

As regards to IT management, the same is true for frameworks, there's an impressive number of overlapping frameworks that come into play from a number of different sources.

Frameworks exist and operate at a higher level than standards, and they are not as immediately critical. Two servers that fail to communicate and so lose a bank's backup data is one thing, but a failure to tailor the IT governance and enterprise architecture efforts of the same bank won't get as many headlines, even if the long term effects will be much more severe.

You have Axelos's ITIL dealing with IT service management. You have COBIT from ISACA for IT governance. You have good old TOGAF from the Open Group (or DODAF or MODAF or FEAF or FSAM from their various government sponsors).

All of these frameworks are supplemented by industry-level reference models such as BIAN, from… BIAN… and then there are vendor-level models such as IBM's Information Framework (IFW), with all of its subdivisions.



Now, a question that comes up again and again is "how does framework X align with framework Y?" The all-time classic in this category being "how does TOGAF align with ITIL?", and if I had a hundred pounds for every time I've been asked that specific question I could probably pay for a round-the-world holiday with the accumulated proceeds. To be fair,

I've noticed a movement in recent years where standards try to answer this question – for example, appendix E of COBIT considers mapping to ITIL, TOGAF, CMMI and PRINCE, among others – but such efforts are both high level and rare at this point.

The other common theme that I've encountered implementing these frameworks in organizations is a general recognition that they need to be tailored to the organization using them and not treated as gospel.

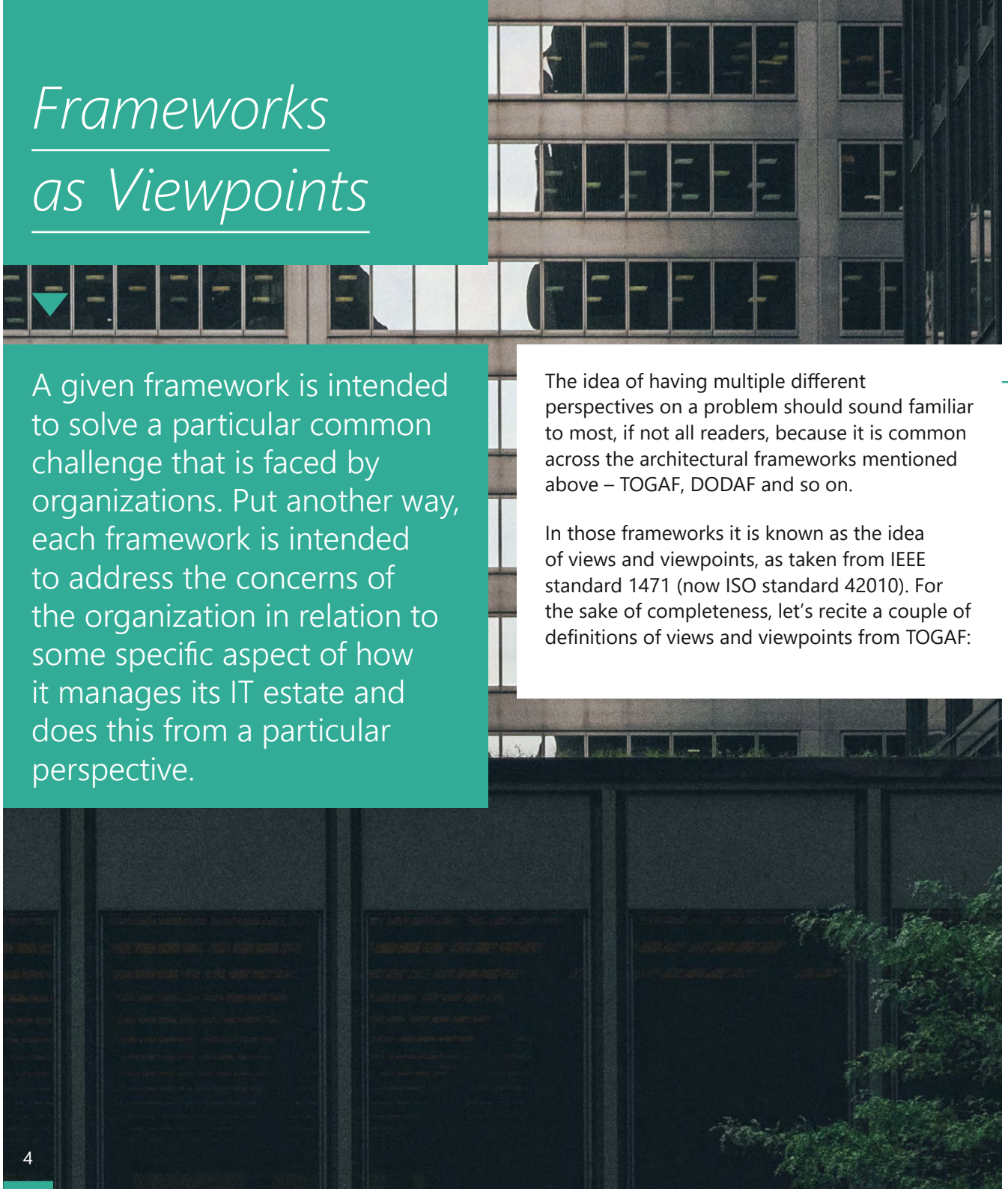
This is often stated explicitly in the text of the framework. Consider step 5 of TOGAF's preliminary phase, "Tailor TOGAF and, if any, Other Selected Architecture Framework(s)." Or consider where chapter 7 of COBIT states that a key factor for successful implementation includes "Tailoring COBIT and other supporting good practices and standards to fit the unique context of the enterprise."

But this fact can be very frustrating to the organization in that the obvious question is "How do we tailor the framework or frameworks? What should we change? What should we adapt?" I've even seen organizations that mandate that TOGAF or COBIT be adopted wholesale, with the organization adapting itself to the text of each standard, in an apparent attempt to avoid this question.

In this paper I'm going to outline an alternative way of thinking about these frameworks that can be helpful in addressing both sets of questions – how a set of frameworks line up to one another, and how frameworks should be tailored for the needs of a given organization.

The root of the problem is that people have nothing to relate these questions. However, there is an analogy we can turn to from within a subject area and the same problem space.

Frameworks as Viewpoints



A given framework is intended to solve a particular common challenge that is faced by organizations. Put another way, each framework is intended to address the concerns of the organization in relation to some specific aspect of how it manages its IT estate and does this from a particular perspective.

The idea of having multiple different perspectives on a problem should sound familiar to most, if not all readers, because it is common across the architectural frameworks mentioned above – TOGAF, DODAF and so on.

In those frameworks it is known as the idea of views and viewpoints, as taken from IEEE standard 1471 (now ISO standard 42010). For the sake of completeness, let's recite a couple of definitions of views and viewpoints from TOGAF:

View: The representation of a related set of concerns. A view is what is seen from a viewpoint. An architecture view may be represented by a model to demonstrate to stakeholders their areas of interest in the architecture. A view does not have to be visual or graphical in nature.

Viewpoint: definition of the perspective from which a view is taken. It is a specification of the conventions for constructing and using a view(often by means of an appropriate schema or template).

A view is what you see; a viewpoint is where you are looking from - the vantage point or perspective that determines what you see.

So, a viewpoint is way of looking at an architecture to address the concerns of a set of stakeholders. A viewpoint is really a kind of filter removes the aspects of the architecture that are not relevant for the stakeholders in question, and you need viewpoints for several reasons.

First of all, it's almost impossible to hold every aspect of a large organization's architecture in one person's mind at once. Seeing it from a variety of different perspectives, one after another is a form of segmentation that makes the overall problem manageable.

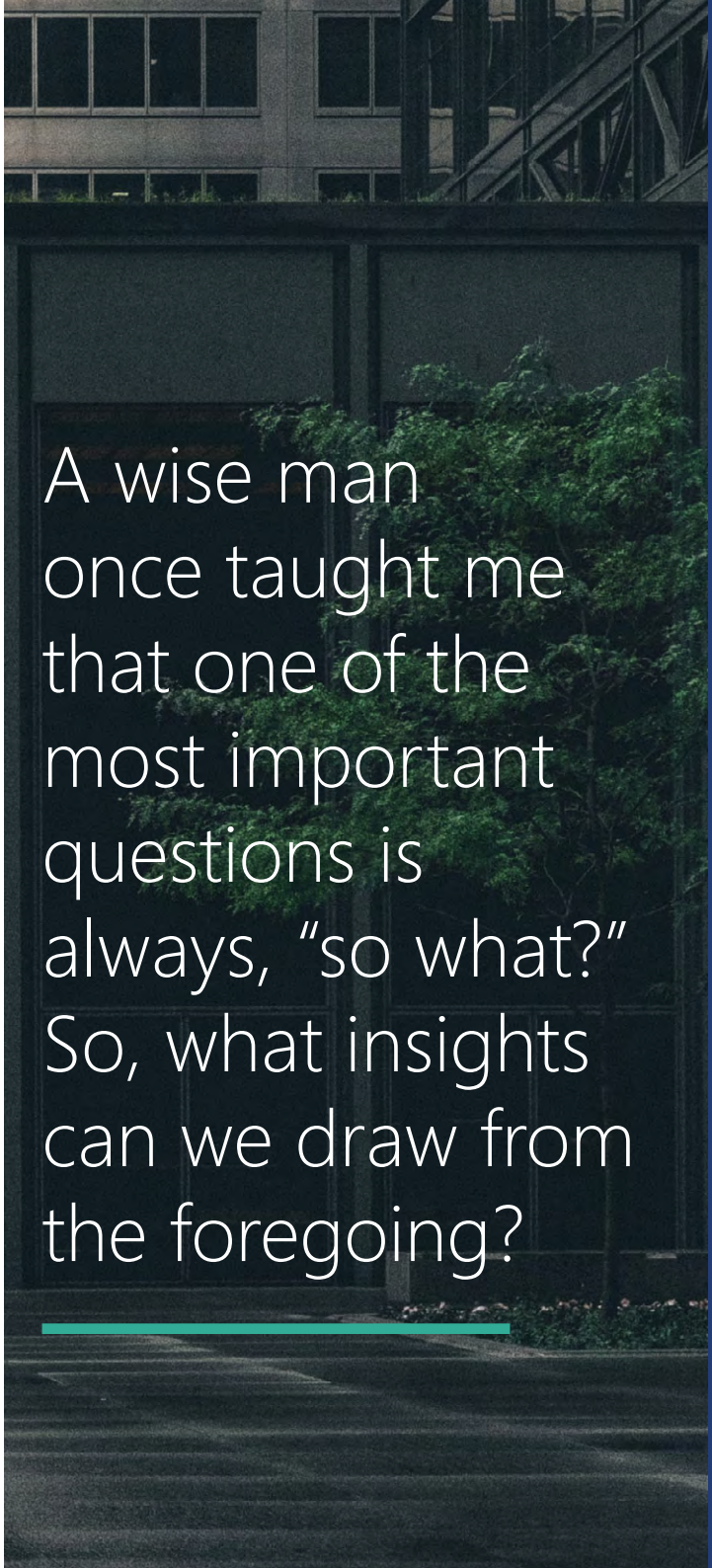
Second, an organization's staff have a finite amount of time available with which to consider the aspects of an architecture that are relevant to them.

The same is also true of frameworks; they are deliberately restricted in scope and coverage. Partly this is because of how much work is involved in getting a group of strongly opinioned experts to agree on subjects in a reasonable timeframe.

But it's also just as much because the staff concerned with, say, service management, have neither the bandwidth nor the background to concern themselves with enterprise architecture – their specialization lies elsewhere (i.e. in service management).

So, for COBIT, governance of the organizational IT estate should include some consideration of enterprise architecture and service management. But COBIT does not attempt to propose how to engage in an enterprise architecture effort or a service management effort – those topics are dealt with in more detail by TOGAF et al, and ITIL respectively.

To bring this section to a conclusion, then, we can think of the various interlocking frameworks that exist in IT management as effectively being different 'viewpoints' onto the subject of IT management in that they focus on different but complimentary subject areas, in order to break the subject into manageable chunks.



Framework alignment may not be relevant

As anyone who's worked with the viewpoints in a framework like TOGAF or ArchiMate will know, two given viewpoints may cover the same entities, some of the same entities or have no overlapping coverage whatsoever.

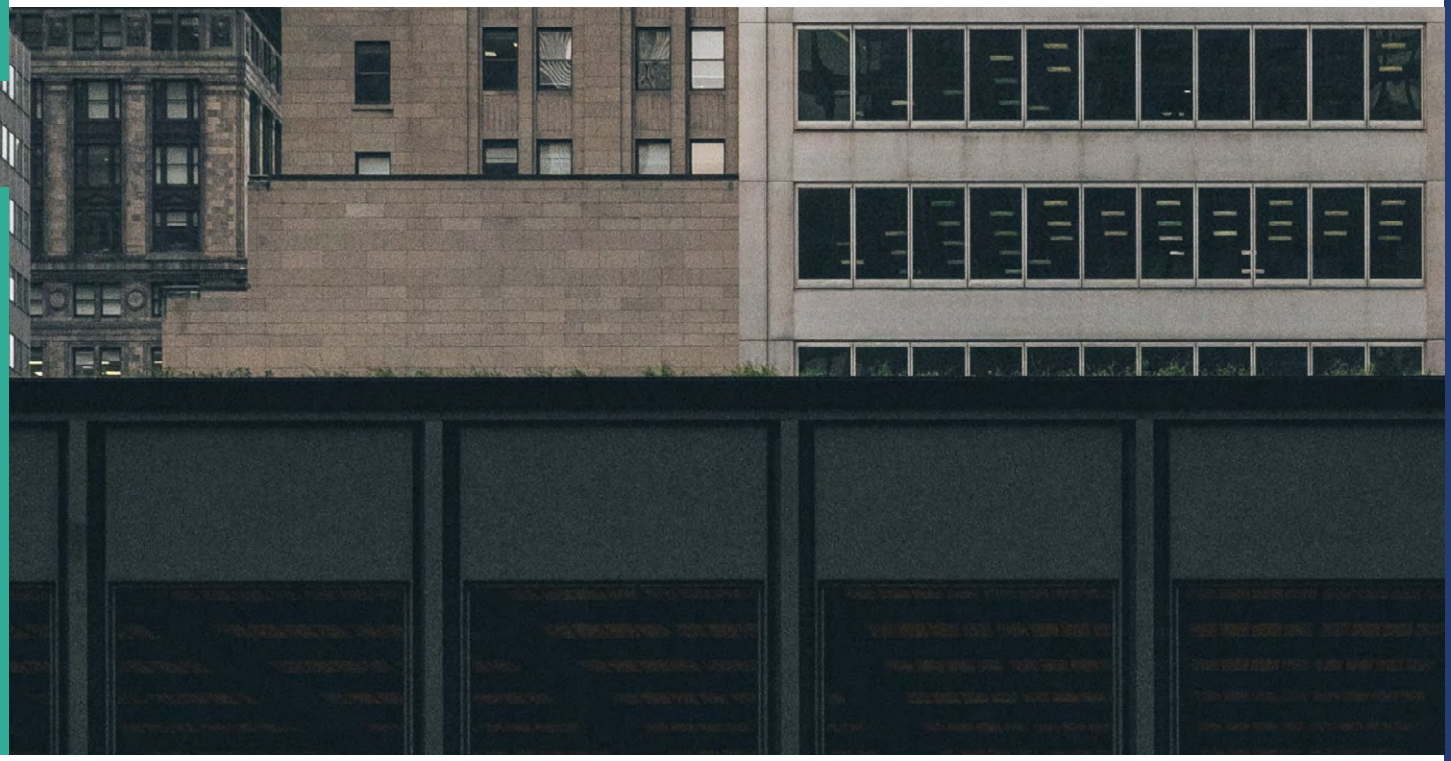
For example, a business function diagram and an infrastructure deployment diagram will have no overlap at all in terms of containing the same information. Now, it's possible to argue about the usefulness of those two viewpoints but as defined they don't overlap.

In the same way, the reason why the overlap or touchpoints between two different frameworks is often not clear may actually be because the overlap is slight at best. In other words, the question is hard to answer because the question does not apply.

For example, for all the interest in the overlap between TOGAF and ITIL I have yet to see a good mapping, but I have come to the conclusion that there is no useful, meaningful mapping to made between these two standards – they cover areas that are too different. This does not invalidate either, it merely means that there is no synergy between them.

Objects may not be perfectly aligned

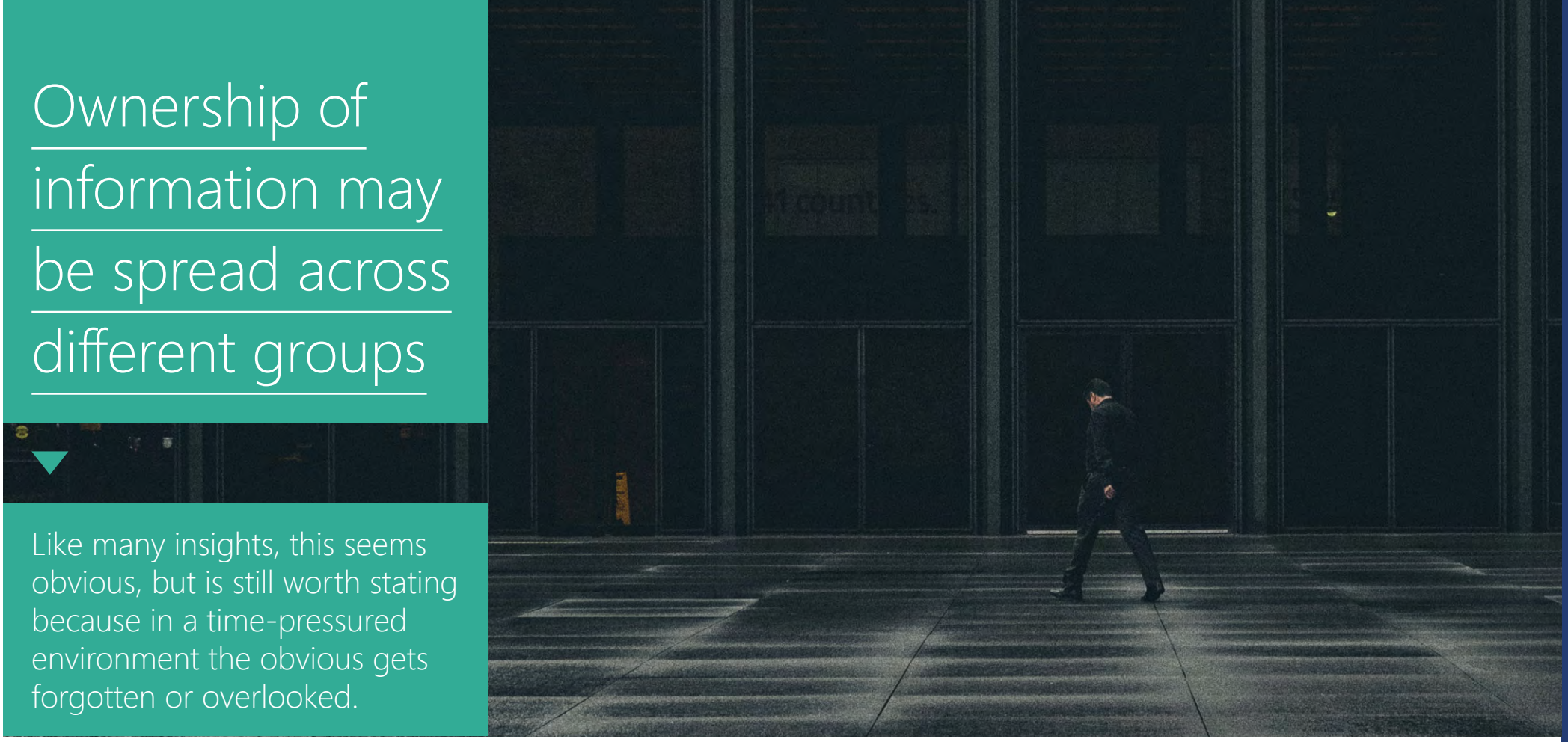
It can sometimes be difficult to reconcile the precise definitions of types such as ‘business function’ between different frameworks, and this can lead to frustration – is the definition in one framework incomplete? Should the entity in one framework be represented in another one?



The fallacy here is that every classification is arbitrary to some extent, and there is no reason that, say, a business function in one framework should exactly correspond to a business function in another framework.

The cleavage point here is to recite our question from earlier: “So what?” In other words, why do you need the definitions to line up perfectly? What insights do you hope to achieve from a perfect alignment between the two frameworks in this case?

Ownership of information may be spread across different groups



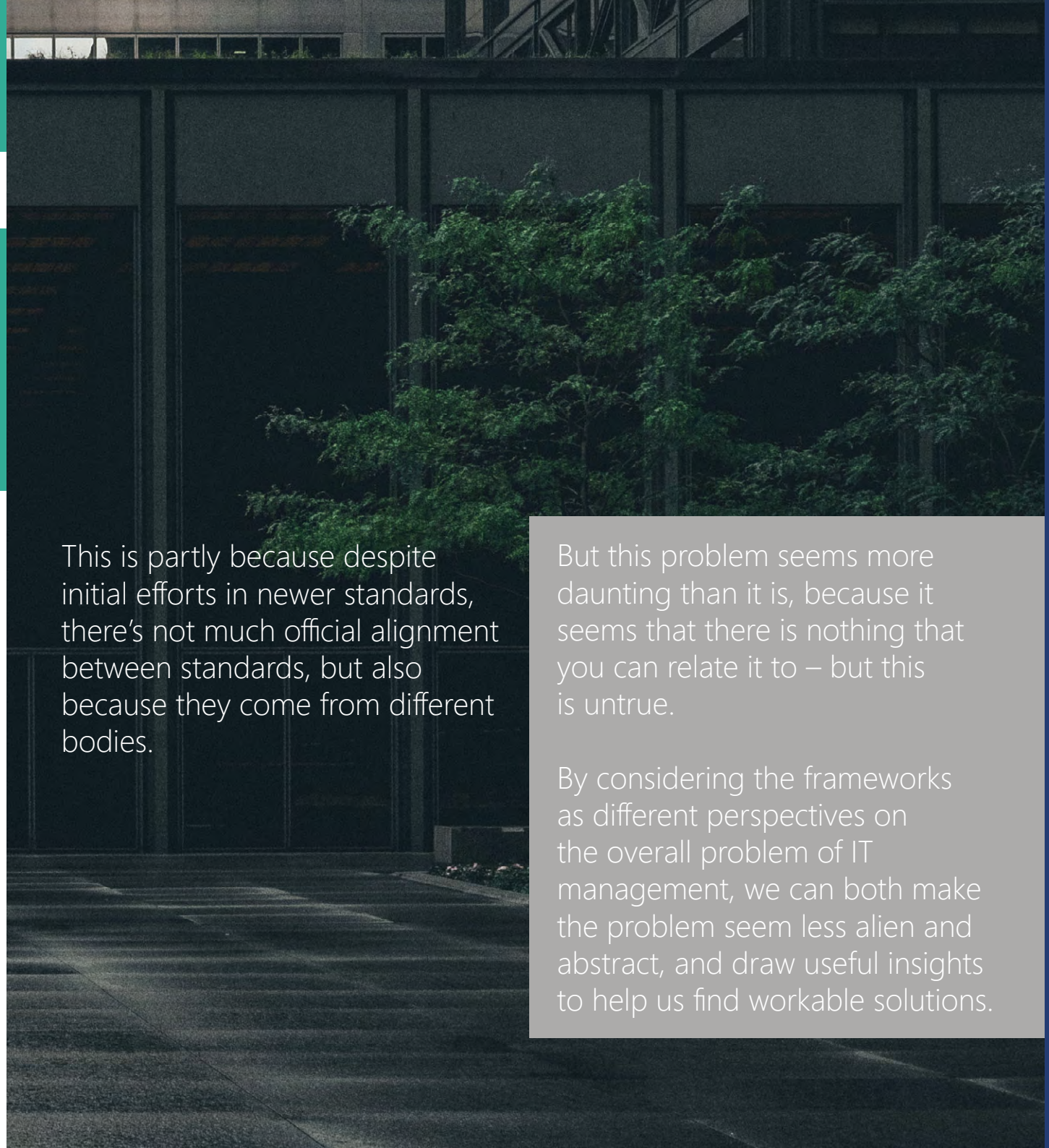
Like many insights, this seems obvious, but is still worth stating because in a time-pressured environment the obvious gets forgotten or overlooked.

In architecture, the information about the infrastructure will be owned by one group and the information about the organization structure will be owned by a different group. Likewise, the information about the organization's governance structure will be owned by a different group to the one that owns information about the service management of the organization.

In other words, the presence of different frameworks implies that there needs to be a mechanism for communication between the 'owners' of the different frameworks to make sure that each is aware of any changes to elements in the sphere of the other.

Conclusion

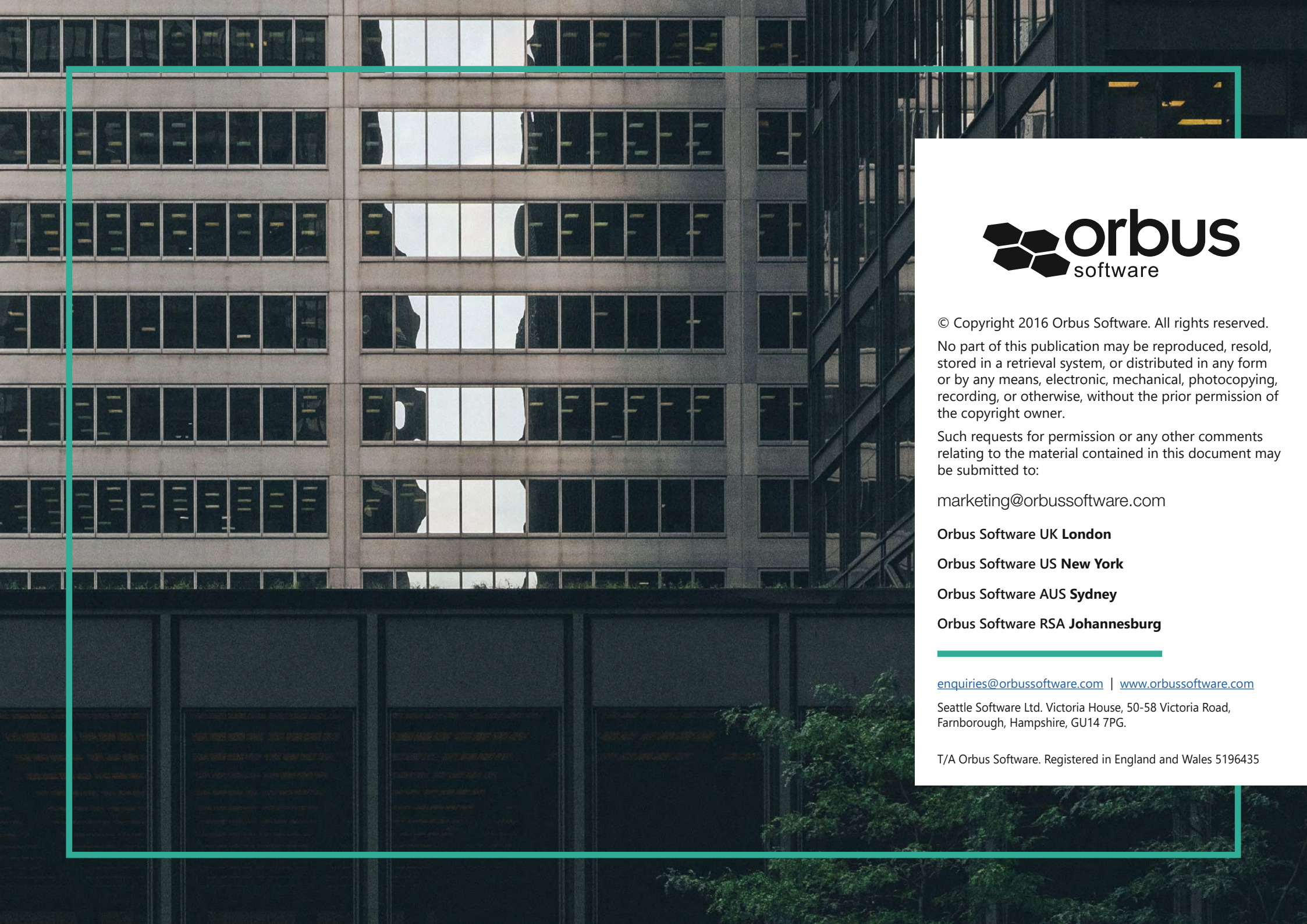
The twin challenges of implementing multiple frameworks and tailoring them to the organization can seem a daunting exercise that requires extensive thought and analysis.



This is partly because despite initial efforts in newer standards, there's not much official alignment between standards, but also because they come from different bodies.

But this problem seems more daunting than it is, because it seems that there is nothing that you can relate it to – but this is untrue.

By considering the frameworks as different perspectives on the overall problem of IT management, we can both make the problem seem less alien and abstract, and draw useful insights to help us find workable solutions.



Orbus Software UK London

Orbus Software US New York

Orbus Software AUS Sydney

Orbus Software RSA Johannesburg

enquiries@orbussoftware.com | www.orbussoftware.com

Seattle Software Ltd. Victoria House, 50-58 Victoria Road, Farnborough, Hampshire, GU14 7PG.

T/A Orbus Software. Registered in England and Wales 5196435