



6 Ways to Organize Your Architecture Models

Marc Lankhorst



If you have some experience in modeling real-life, full-size architectures for large-scale organizations – preferably in the ArchiMate language, of course – you have likely come across the challenge of organizing your models in logical and manageable ways. In the following pages, we’re going to share our top 6 ways to organize your architecture models. These methods should help you keep your models neat and tidy, while also supporting better outcomes for your strategic initiatives. Let’s see what they are.

1. Organize your model repository by business domains, information domains and technology stacks

In large organizations, you will need some substructure for your integrated current-state model. You can organize your capabilities, business functions and business processes by business domains or high-level business capabilities. That is a natural subdivision of the business world, both easily recognized by the people involved and the typical responsibility and authorization structures in your enterprise. The application and data part of your model repository you can organize according to information domains, consisting of:

- Sets of information items that typically belong together
- Applications that process this information
- Business functions that these applications support

For example, you might have an information domain for Customer Data, comprising the relevant personal data you capture on your customers and the applications used to store and process it, e.g. CRM systems. Another information domain might be Customer Interaction, where you might find, say, call center and social media applications.

For the technology part of the architecture, a business-oriented structure often makes less sense, as much of the technical infrastructure of an enterprise cuts across business or information domains. Instead, organizing by technology stacks makes more sense in many cases. These stack models can then be managed by separate teams with the requisite technology expertise. Setting up a navigation page to help people find their way around this structure is also very useful. Figure 1 shows an example of this.

Two other model categories are 1) external reference models and 2) standards. For instance, the BIAN standard for the banking industry or the ISO/IEC 27001 security standard shown in the figure below describe your way of working, including your:

- Modeling process
- Naming conventions
- Architecture roles and more

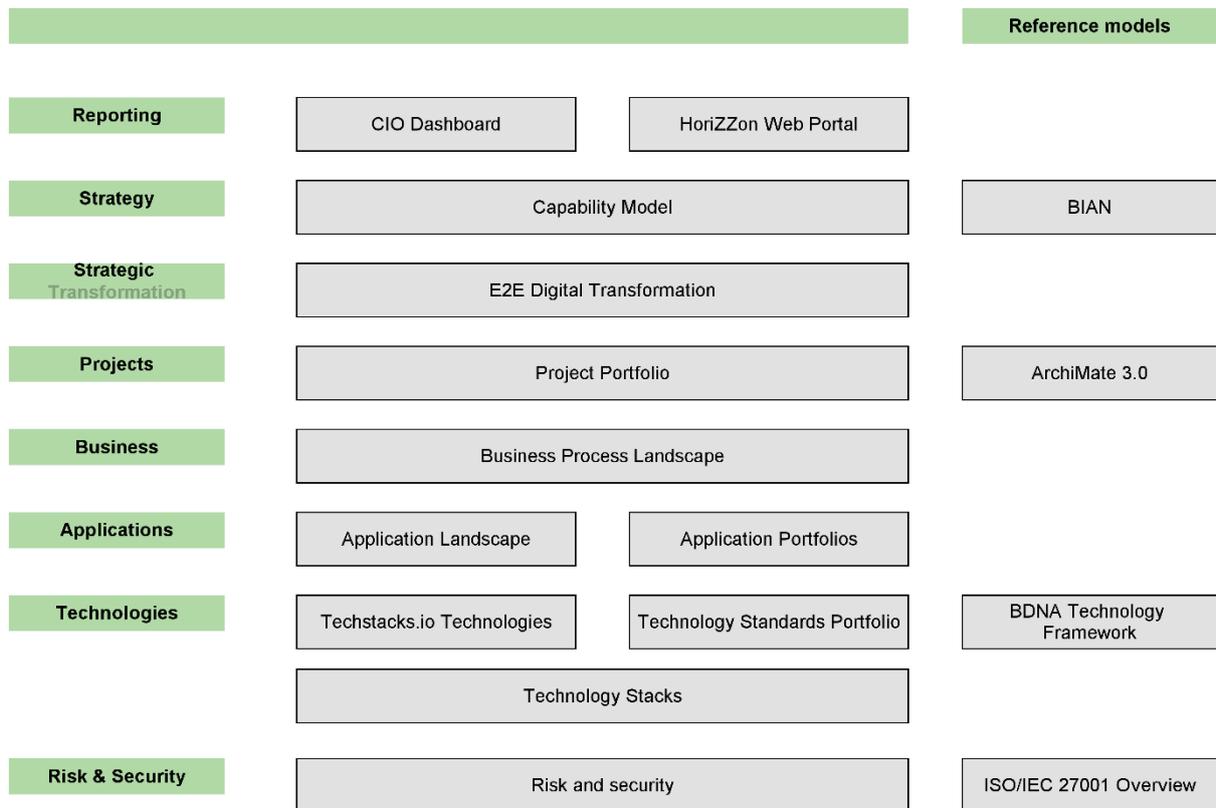


Figure 1: Example navigation page to help stakeholders find their way

Having these accessible in the same repository is very handy. Of course, there may be cross-cutting and overlapping aspects to these domains. To sort these out, you need to set up regular coordination between the experts involved, preferably supported by a collaboration platform that allows you to set the right permissions, create review workflows and otherwise organize the work to be done. Our [Enterprise Studio](#) and [HoriZZon](#) products offer tools that support this aspect of architecture modeling.

2. Separate current- and future-state models

It is important to make a clear separation between what you have described about the world as you know it is today and the future that you envisage. Facts about the present and ideas about the future should be clearly distinguishable in your architecture models.

One way to do this is to create a read-only, integrated reference model of the current state (possibly per domain, as described above) and separate project-level models that describe aspects of the future as it is being designed. The visibility of these future-state models is limited to the specific project team working on them, so people do not get confused by the work being done on future-state designs. Within these future-state models, elements of the current state that carry over into the future can be reused. Once the time has come to put the project results into production, the future-state models will become reality and will transfer into the integrated current-state model.



It's important to note that the process for creating these models is very different. When describing the current state, it is often easiest to start with what is called 'structure' in the ArchiMate modeling language: the organization, applications, software platforms and devices that are readily visible in your enterprise. From there, you can look at their behavior: the processes, services and functions provided by these structure elements. As a next step, you might then look for commonalities and opportunities for synergy or rationalization, just as an example.

When designing a future-state architecture, however, you will often work the other way around: first you define the services you require to solve your business problem, then you outline the processes and functions that need to provide these services. Only once that is complete do you decide on the structure elements that will perform this behavior and deliver these services. This avoids the common trap of constraining your possible design solutions by the structures you know (e.g. components, systems, products etc.), especially early on in the design process.

3. Separate model content from views of that content

In architecture modeling, it is important to distinguish the content of your architecture from views of this content that address specific stakeholder concerns, by visualizing part of the content in a suitable manner. The same model elements may appear in different views; vice versa, all views share the same underlying content.

Typically, you organize the model content according to the types of elements you have modeled, for instance using the layers of the ArchiMate language (motivation, strategy, business, application, technology, implementation & migration), or even more fine-grained by defining separate collections of business processes, application components et cetera. The views on that content are best organized according to the types of stakeholders you address, with views for groups like business managers, process designers, application owners, data stewards, project managers, system administrators, and so on.

4. Define naming and other modeling conventions

To be able to find anything in a large-scale model, you must know what to look for. In plain language, you will need to know what things are called – and how they're labeled – in your model. This is why defining clear naming conventions is essential. Different organizations will have their own specific naming schemes, so no universal convention applies to all situations. Nevertheless, here are common tips that may help you create your naming process:

- Let business experts 'own' the names and definitions of capabilities, business processes, functions and the like. If IT people invent new names for those, the business will often fail to recognize 'their business' in it, leading to misinterpretations and misunderstandings.
- Define specific naming conventions for specific types of elements. For example, use a present-tense verb plus a noun for process names (e.g. 'Handle Claim'). That way, the name of an element already gives you information on the type of thing you model.



- In large organizations, use pre- or postfixes for names, such the business domain of the element. This is particularly important if these domains use the same names. For example, in an insurance company, different domains like ‘health,’ ‘life’ or ‘property’ may each have a ‘Handle Claim’ process, but these processes will differ. Calling them ‘Handle Claim – Health’ or ‘Handle Claim – Property’ helps distinguish between them.
- Make a clear separation between different aspects of the same real-life entity being modeled. If you model a customer who takes part in a process as an actor, don’t use the same name for the information you capture on that customer. You should also have a different name from the customer file or database in which you store that information. Calling all these elements ‘Customer’ will surely confuse the hell out of anyone using your model.
- Be selective in your use of modeling concepts. Languages like ArchiMate are very rich and powerful, but you will seldom need all their concepts. Choosing which concepts to use for which purpose and limiting yourself to the essentials is key to keeping your architectures easy to understand.

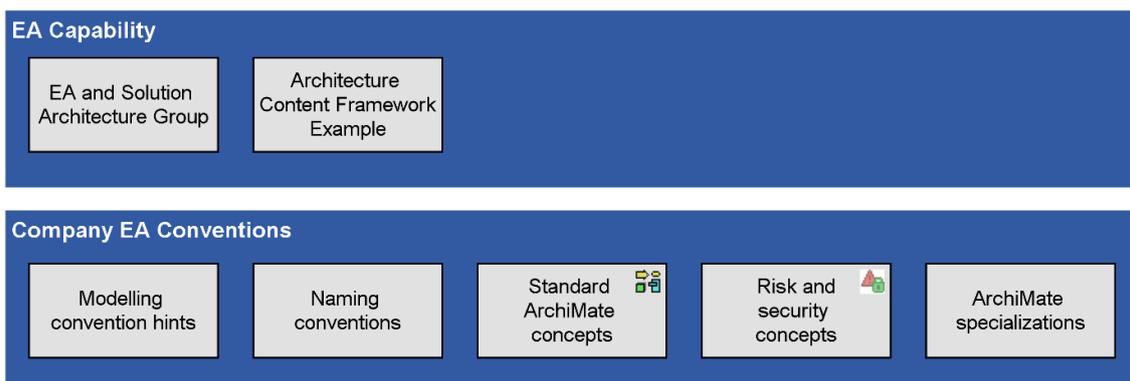


Figure 2: Overview of your EA capability, ways of working, and best practices

5. Set up an ‘editorial board’

To create collaboration in large, federated organizations, you cannot rely on simple top-down control from a central department. There must be room for local variance simply because, in large organizations, no standard will fit all situations. Nevertheless, you do need to coordinate between different departments and domains to ensure their ways of working are compatible, and to share and benefit from each other’s experiences so you can build up your own library of organization-specific best practices.

Working with such large and complex organizations, I have seen that it is often helpful to create a kind of ‘editorial board,’ composed of representative architecture and modeling experts from different areas of the organization. They can find shared ways of working and other commonalities across your architectures (such as the modeling conventions and structuring mentioned previously), and they can coach their less experienced colleagues in applying these practices to promote high-quality models.



However, an ‘editorial board’ is different from your typical architecture board, which decides on the architectural content itself. Rather, this editorial board supervises the way architectures are modeled and described. These are two distinct competencies: A good architecture may be badly documented, and poor architecture may be well-documented. Make sure both are well-done.

6. Define clear responsibilities, access rights, user groups and procedures

In any large-scale organization, you need to be clear about who does what and who has access to which materials. If everyone can edit anything in your architecture models, you can guess what will happen. Things may get moved or entire models may be deleted. In general, I would not advocate hiding parts of the architecture from view (unless for specific security-related reasons). Architects should be trusted to see the work of their peers and make good use of that. However, you will likely need to limit write access to the people who have a defined role in creating or updating models. Otherwise, your models will quickly become a mess.

One way to do this is to organize the work on your future-state architectures in projects, each with their own specific models, as described above. Project teams have the right to change their project models, but their results are only published to the wider architecture repository when they are sufficiently ready, as decided by a lead architect or decision maker within the organization.

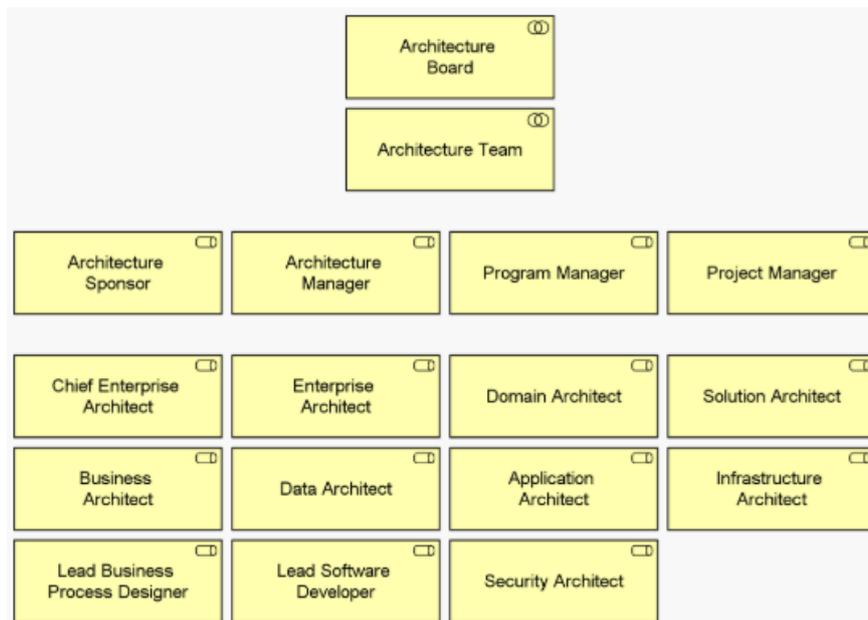


Figure 3: Common architecture-related roles

Enterprise Studio supports your organization’s different projects, roles and rights for designers and lead designers, and it offers user groups that can be organized as project teams. Moreover, the workflow facilities in the HoriZZon portal let you create workflows to organize the architecture development process. Out of the box it supports review and approval workflows, but you can also create your own customized processes. Interested to learn more about the way we support large-scale architecture endeavors? [Watch a webinar](#) or [book a demonstration](#) today!



About BiZZdesign

BiZZdesign is a leading enterprise transformation software vendor based in the Netherlands. Founded in 2000 as the commercial spin-off of an R&D institute, today the company enjoys a global presence and is recognized by industry analysts as a market leader. BiZZdesign's flagship product, Enterprise Studio, is deployed in blue chip companies and government organizations across all continents, where it plays a key role in enabling meaningful business change.

For more information, visit www.BiZZdesign.com.