**A simple method to create useful solution architecture diagrams**

[[Tuan Le](https://medium.com/@tuanvietle?source=post_page-----6a127b93a122--------------------------------)](https://medium.com/@tuanvietle?source=post_page-----6a127b93a122--------------------------------)

[Tuan Le](https://medium.com/@tuanvietle?source=post_page-----6a127b93a122--------------------------------)

[Jul 19, 2020·7 min read](https://medium.com/@tuanvietle/a-simple-method-to-create-solution-architecture-diagrams-6a127b93a122?source=post_page-----6a127b93a122--------------------------------)

Being a solutions architect for the past 15 years, I truly understand the value of a useful solution architecture diagram as an effective visual aid to define and describe an architecture of a system delivered in the context of a specific solution as per [Wikipedia](https://en.wikipedia.org/wiki/Solution_architecture).

In my opinion, a persuasive solution architecture diagram must be able to convey the business values to key stakeholders and the technical details to developers. Therefore, it should not be too technical so that business stakeholders can understand; however, it should still provide a practical high-level solution blueprint for later development. In other words, you should try to strike the right balance between technicality and business context when creating a solution architecture diagram.

Having seen and created quite a few solution architecture diagrams in my line of work, I have formulated a simple method to create solution architecture diagrams as outlined in this article. By no means, I would claim that I devised this method. I just synthesized my observations and improvisions into a set of conventions to form a repeatable process that others can follow.

Which tool should I use to create a solution architecture diagram?

There are several proper diagramming tools out there, both online and offline, Visio, LucidChart, Gliffy, Creately, just to name a few. However, I would recommend using Microsoft PowerPoint as your primary tool for the following reasons:

* **Reusability**: If you create a useful diagram, others would want to re-use it in the same project for different purposes or in other projects. Sharing your solution architecture in PowerPoint format will make it a whole lot easier for others to edit as PowerPoint is a universal productive software that everyone would have.
* **Conciseness**: Instead of having unlimited canvas to draw, you will need to confine your diagram within a single slide with PowerPoint. As such, you will need to choose the right granular level for the solution architecture hence forcing you to focus on the essence of your solution.
* **Flexibility**: PowerPoint gives you the flexibility to use whatever shapes you want, whatever images you need. I typically use PowerPoint standard shapes together with icons/logos downloaded from the Internet. These images can comfortably fit into a diagram if they have a transparent background which can be done quickly with the magic wand function in [Paint.NET](https://www.getpaint.net/) (free image editor).

What should I include in a solution architecture diagram?

Essentially, the job of a solutions architect is to analyze business requirements and design a technology solution to meet those requirements. It’s inarguable that solution architecture lays a solid foundation for all successful development projects.

Solution architecture helps bring to life how different aspects of **business**, **information,**and **technology**come together in a particular solution. Therefore, a solution architecture diagram should visualize above three critical elements in a way that is useful for both business stakeholders and developers.

In this section, I will walk you through each element using a B2C e-commerce solution architecture. I picked e-commerce as an example because it is a more straightforward solution compared to an enterprise solution. Still, it is complex enough to prove the importance of a useful solution architecture diagram.

Business requirements for a B2C e-commerce solution are relatively simple. The client just wants to have an e-commerce site where consumers can see their products and make the purchase online. They also want to make products available in Google search and the ability to remarket consumers via Criteo platform to attract more traffics to the site. And of course, they insist that the new e-commerce site must integrate with existing ERP, CRM and marketing automation system.

In the context of an e-commerce solution, let’s look at three critical elements for a solution architecture: business, information and technology.

**Business**

When talking about business aspects, a solutions architect should explore two things: users and functionalities.

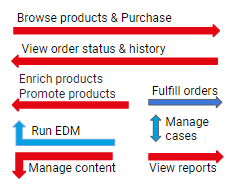
User icon



User icons

Users of an e-commerce solution are not just consumers who shop online but also all business users behind the scene that operate the e-commerce site. For example, the e-commerce manager who oversees the whole operation, the merchandisers who make the products look good with attractive images and run promotions or customer service agents who answer consumers’ inquires. In the solution architecture diagram, you can use different icons found on the Internet to denote these users.

Notations for functionalities

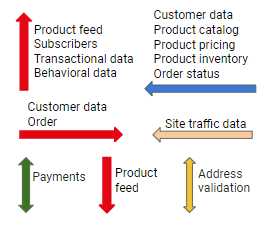


Notations for functionalities

Functionalities included in a solution architecture diagram should be the critical use cases of your solution as it’s not possible to include every single functional requirement. Therefore, you will need to pick the right ones to showcase that the proposed solution does provide the required functionalities to meet business requirements. You can denote these functionalities as arrows connecting users and respective systems. Arrows should point outward from users for use cases where users actively perform something. And arrows should run the other way when users passively consume something from the systems. You should also annotate each use case with a short text starting with a verb.

**Information**

Data flow notations

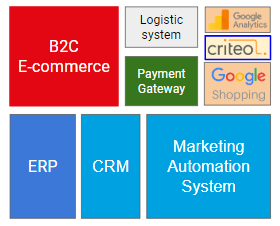


Data flow notations

In solution architecture, information refers to data flowing between different building blocks. At first, you need to identify significant data entities in your solution. For example, in an e-commerce solution, they are product catalogue, product pricing, inventory, customer data and order. Again, you can use arrows denote data follows with annotated data entity names. These arrows should indicate the directions of data flows from one system to another. These data movements will help you identify the integration requirements.

**Technology**

System notations



System notations

Last but not least, you need to include technology building blocks of your solution in the solution architecture diagram. These building blocks include both new and inexistent components. In our example, an e-commerce solution will consist of

* the B2C e-commerce system itself.
* Internal ERP, CRM and Marketing Automation System as the client has indicated.
* And other external systems including Logistics System, Payment Gateway and Google Analytics, Google Shopping and Criteo.

You should denote these technology building blocks as rectangles. You can leverage logos of known systems to make your diagram more visually appealing.

How do I put them together?

Now that we have identified what to include in a solution architecture diagram let’s discuss how to put them together.

**Steps**

I usually draw all systems first followed by data flows the user icons and use cases. It’s virtually impossible to put each element at the right place on the first try. Therefore, you should just draw what you need first then work your way to rearrange them to suitable locations. It can be a bit tedious at first, but once you have a hang of it, creating your next diagram will be a lot easier.

**Layout**

A rule of thumb is the most critical systems should be at the centre as most data flows and use cases will touch it. Another one is that you should place highly inter-dependent building blocks close to each other. As such, you can save a bit of time trying to avoid drawing arrows running across each other.

Besides, you should try to place all internal users on one side and external users on the other side instead of putting user icons all over the place. This convention will make it easier for your audiences to follow your diagram as key stakeholders usually pay more attention to the business context.

**Optic**

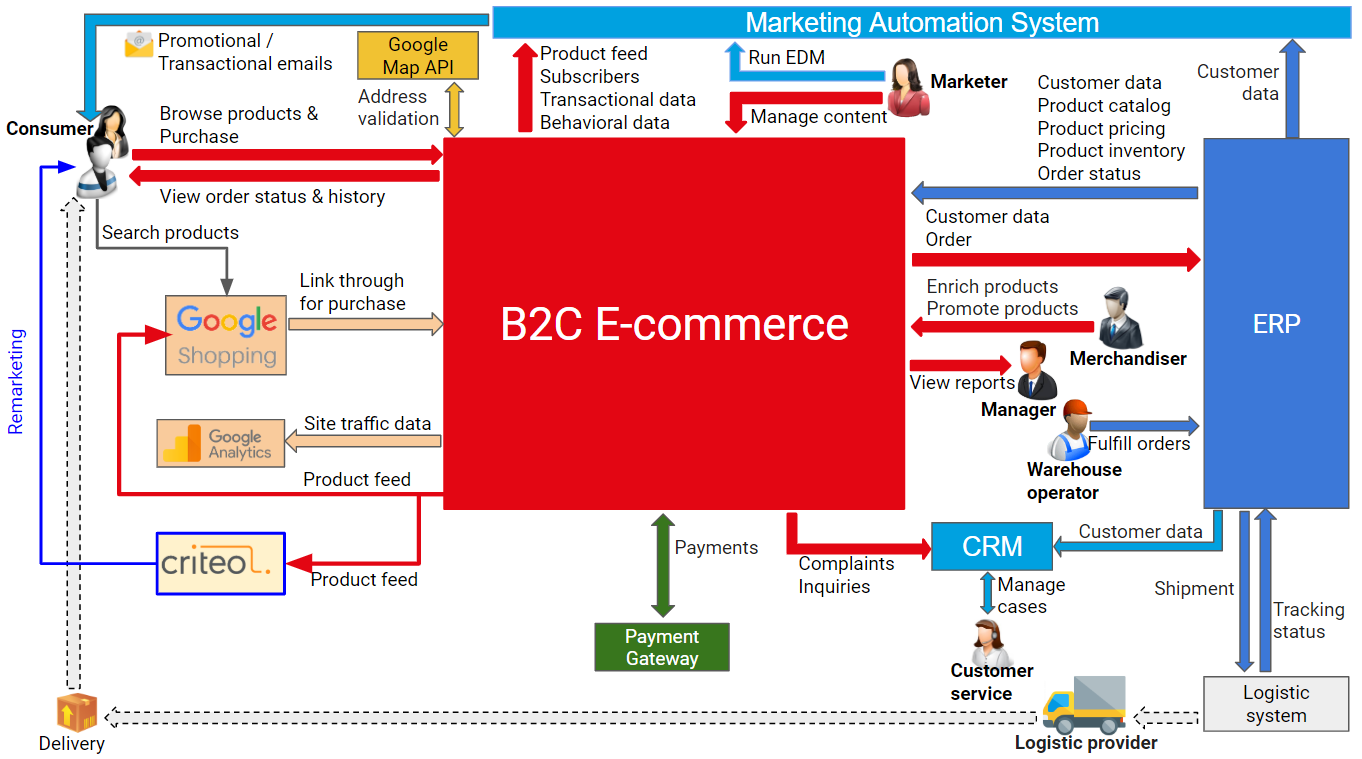
There are quite a few things in your solution architecture diagram; hence if you don’t pay attention to the optic, your solution architecture can look very messy. Here are a few tips:

* **Font**: Use consistent font face and font size across your diagram. Don’t ever mix font-face or use different font-sizes for similar items.
* **Colour**: You should use the most prominent colour such as red for the most critical systems and different colours for other systems. Functionality and data flow arrows should also inherit system colours to enhance readability.
* **Alignment**: You should try to align texts and blocks whenever possible as proper alignments can make your diagram look less messy and more readable.

**The artefact**

Below is a sample solution architecture diagram for our example where business, information and technology elements come together. Solution architecture is a complex subject; hence the solution architecture diagram inherently looks busy, but when you arrange them neatly in the right order, the audience can comprehend what is in your proposed solution architecture.

B2C e-commerce solution architecture diagram



B2C e-commerce solution architecture diagram

Should I use this method to create solution architecture diagrams?

In software engineering, [Unified Modeling Language](https://en.wikipedia.org/wiki/Unified_Modeling_Language) (UML) is a standard way to visualize the design of a system. Software architects and technical architects widely use UML to model a system independent of a platform language.

In the field of solution architecture, however, I have not come across any standardized method to visualize the design of a solution. It seems that each solutions architect has his/her way to draw solution architecture diagrams. Having seen both good and not-so-good ones in the past, I believe that this simple method would help solutions architects produce useful solution architecture diagrams more consistently and efficiently.

To me, as a solutions architect, the most rewarding moments is when a key stakeholder pats me on the back and sincerely says “I believe you have a good handle on this” at the end of a presentation which features a useful solution architecture diagram as the focal point for discussions. I believe this simple method does help me create opportunities to enjoy many of those moments.

***Disclaimer****: This disclaimer informs readers that the views, thoughts, and opinions expressed in the text belong solely to the author, and not necessarily to the author’s employer, organisation, committee or other group or individual.*