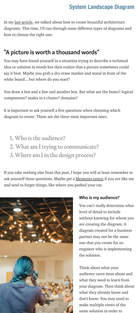
Beyond Senior Engineer: Even More Architecture Diagrams

Think Like an Architect — Part 6 — Which type of diagram do I need?

amber rollwec FOLLOW
Apr 12 · 14min read ★



In my last article, we talked about how to create beautiful architecture diagrams. This time, I'll run through some different types of diagrams and how to choose the right one.

"A picture is worth a thousand words"

You may have found yourself in a situation trying to describe a technical idea or solution in words but then realize that a picture sometimes could say it best. Maybe you grab a dry erase marker and stand in front of the white board… but where do you start?

You draw a box and a line and another box. But what are the boxes? logical components? nodes in a cluster? domains?

It is important to ask yourself a few questions when choosing which diagram to create. These are the three most important ones:

1. Who is the audience?
2. What am I trying to communicate?
3. Where am I in the design process?

If you take nothing else from this post, I hope you will at least remember to ask yourself these questions. Maybe put a Post-it Note on your screen if you are like me and tend to forget things, like where you parked your car.

Photo by Christin Hume on Unsplash

Who is my audience?

You can't really determine what level of detail to include without knowing for whom you are creating the diagram. A diagram created for a business partner may not be the same one that you create for an engineer who is implementing the solution.

Think about what your audience cares more about and what they need to learn from your diagram. Then think about what they already know and what they don't. Sometimes I make multiple views of the same solution in order to communicate to different audiences.

What am I trying to communicate?

In addition to understanding your audience, ask yourself,

> "What is the real purpose of this diagram?"

Do you want to show another team how to integrate with your service? Or do you want to show how your services are deployed for a new team member? Or do you want to show a high level overview of services within a domain? Each of these diagrams will look very different from one another, but they are all valuable.

Where am I in the design process?
(e.g. discovery brainstorming, post-deployment)

Depending on how much info you have or where in the lifecycle of your software you are, you need to show different things.

Let's say you have a current system and are working towards a new architecture. Sad need to take several steps to get there. How do you show that in a way that highlights the things that need to change with each phase? You probably can't do that with a single diagram, rather, you need a set of diagrams that show the phases and the changes needed.

What if you are very early in the design phase, and there are many open questions? The diagram might be super high level and should aim to highlight the unknowns, so that the team knows where further research and discovery are needed.

Photo by Nick Karvounis on Unsplash

Overview of Diagram Types

There are many types of diagrams, and choosing the right tool for the right job takes practice, experience and knowing the answer to the three above questions. This chart should help get you started in making that decision.

HA = high availability
DR = disaster recovery

Standardizing Designs

The trap that many people fall in when they first start creating designs is to try and put everything on one diagram. Everything from data flow to physical components like datacenters to organizations team ownership. True me, we have all done this.

Instead, I suggest we break up these into different diagrams to communicate our ideas.

There are many existing standards to help guide you through the process of documenting your designs. In fact, this is a great list of International Organization for Standardization (ISO) frameworks: http://www.iso-architecture.org/ARES/ufs-frameworks/table.html

It's a long list so I will detail three options that I'm most familiar with, the first two being ones I use most often. Feel free to use the one that works best for you — just avoid the trap of design overload!

Example 1 — Start high level, then drill down

Diagram Backed by Wing Journal
https://commons.wikimedia.org/wiki/File:Wing_and_Bird_icon_white_on_blue.svg

Conceptual — Represents an idea, but not the technology

Logical — Could show processes, inter-system communication, one box per system usually, technology included

Physical — Would include deployment details, clustering, data center/cloud, etc

Example 2 — C4 Model

This model defines four hierarchies or levels abstraction that all begin with C1 Set C4! Here are the four C's followed by a brief description of each one, taken from https://c4model.com/:

Context
Containers
Components
Code

Visit the website to see more detailed descriptions, including intended audiences and even more examples.

In my example diagram, I have created an imaginary magical Unibow (🌈🏹) system for users to upload unicorn and rainbow images, then images are processed to give every unicorn a rainbow and every rainbow a unicorn (as nature intended). I used AWS services in the example, but the other major cloud providers have equivalent offerings.

Level 1: System Context diagram

Context Diagram Example

The highest level view, the context diagram, "where you show your system as a box in the center, surrounded by its users and the other systems that it interacts with." We also show user and external interaction with the system.

Level 2: Container diagram

Container Diagram Example

Take one step down to the container diagram which "shows the overall shape of the software architecture and how responsibilities are distributed across it." Here you will also begin to diagram the logical interactions inside the system.

Level 3: Component diagram

Component Diagram Example

"Zoom in, and decompose each container further to identify the major structural building blocks and their interactions."

Level 4: Code (OPTIONAL)

"Show how it is implemented as code", using UML class diagrams, entity relationship diagrams or similar. "This level of detail is not recommended for anything but the most important or complex components."

Supplementary Diagrams

System Landscape diagram — Even higher level overview showing many systems and their interactions
Dynamic diagram — "Useful when you want to show how elements in a static model collaborate at runtime to implement a use story, use case, feature, etc."
Deployment diagram — Shows the physical deployment of the system…. would include things such as clustering, compute, network and storage specifics.

Deployment Diagram Example

More info here: https://c4model.com/#FourDiagrams and http://www.softwarearchitecture.zone/

Example 3 — "4+1" model [KRUN]

From: https://en.wikipedia.org/wiki/4%2B1_architectural_view_model
The views using the RUP (Rational Unified Process) naming convention:

Logical view
Process view
Implementation view
Deployment view
Use Case view
Data view (optional)

Conclusion

Photo by Nicolas Tissoux on Unsplash

We went through lots of different types of diagrams, a few design standards, and practiced drawing C4 model diagrams using the magical Unibow system as our example.

You can practice drawing diagrams using a system you know. Think about which type you would use in different scenarios. And remember our three important questions:

1. Who is the audience?
2. What am I trying to communicate?
3. Where am I in the design process?

Have ideas for future topics? Tweet @ me or drop a comment below. See you next time!

Software Architecture Software Development Software Engineering Diagrams

10 claps

WRITTEN BY
amber rollwec
Principal Engineer | Architect at Skilify, Formerly Sr Principal Engineer/Architect at Amazon Web Services (AWS), Microsoft and Confluence. Talks at Scales days.

FOLLOW

Write the first response

Discover Medium
Welcome to a place where words matter. On Medium, smart voices and original ideas take center stage - with no ads in sight. About

Make Medium yours
Follow all the topics you care about, and we'll deliver the best stories for you to your homepage and inbox. Explore

Explore your membership
Upgrade for ad-free reading, unlimited articles, the ability to download stories for intelligent reading, fancy stamps, exclusive, and competitive features. Explore

Medium About Help Legal