

Conceptual Architecture Diagram

HOT Architecture Documentation



Humanitarian
OpenStreetMap
Team

This document provides an overview of a conceptual diagram, then digs into the mechanics of creating one in LibreOffice. For a general overview of the technical documentation approach for HOT, check out the technical documentation Wiki at <https://github.com/hotosm/techdoc/wiki>.

When it comes to solution architecture, a conceptual model is an abstract or high-level design view which includes only the most important components and entities. The goal of this conceptual model is to provide an understandable picture of the high-level design. A conceptual model is often modeled using a block diagram (“boxes and lines”)¹.

Table of Contents

Conceptual Architecture Diagram HOT Architecture Documentation.....	1
What is a conceptual model?.....	1
Boxes and Lines.....	2
Elements of the diagram!.....	2
Some Tips.....	3
Drawing with LibreOffice.....	3
Getting Started.....	3
Page Setup.....	4
Connecting Components.....	5

What is a conceptual model?

From our friends at Wikipedia:

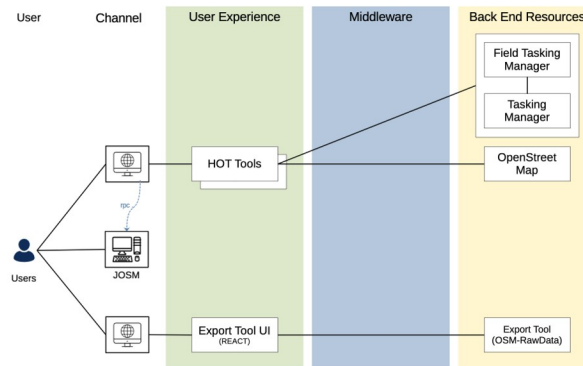
A conceptual model's primary objective is to convey the fundamental principles and #basic functionality of the system which it represents. Also, a conceptual model must be #developed in such a way as to provide an easily understood system interpretation for the #model’s users. A conceptual model, when implemented properly, should satisfy four #fundamental objectives.

- Enhance an individual's understanding of the representative system
- Facilitate efficient conveyance of system details between stakeholders
- Provide a point of reference for system designers to extract system specifications
- Document the system for future reference and provide a means for collaboration

Boxes and Lines

¹ The Carnegie Mellon University Software Engineering Institute refers to this kind of diagram as a Component and Connector view in their approach for documenting software architectures (“Views and Beyond”).
<https://resources.sei.cmu.edu/library/asset-view.cfm?assetID=484159>

Given the high level of abstraction for a conceptual diagram, it is often modeled using a block diagram. A block diagram is a diagram of a system in which the principal parts or functions are represented by blocks (or boxes) connected by lines that show the relationships of the blocks.



Elements of the diagram!

Here is a quick review of how to think about each element on the diagram:

- Each box on the diagram represents an architecturally significant component.
- Each line indicates a connection between two components.
 - Solid lines for realtime (transactional) connections. Realtime connections are very chatty so we don't show any directionality.
 - Dashed lines for batch (bulk) connections. For batch, we show an arrow on either or both end to show the direction the information is flowing.
- The vertical swimlanes represent tiers of the architecture. We use the 5 tiers you see depicted:
 - **Users.** These are the roles of the people using the solution. Not the security roles within the application, but the business roles (usually taken from the Solution User diagram). If all users engage with the architecture in the same manner, you can label them "Users."
 - **Channels.** Methods through which users can interface with Front End Experience components. We use icons for Channels. *Common channels include Web Browser, Desktop Apps, Telephone, Mobile Device, but can also be Fax Machine, GPS Device, etc.*
 - **Front End Experience.** Components providing the user experience. A component can be a "front end experience" in one architecture and a "back-end resource" in another, depending on how it is used in the architecture.
 - **Middleware.** Most typically shared software components loosely coupled through interfaces (e.g., web services), but can also represent other service-like software components, like ETL jobs or file transfer hubs.

- **Back End Resources.** Back end applications or data stores involved in the architecture with which the end user is not directly interfacing. The distinction between “front end” and “back end” is somewhat arbitrary, but often assists in producing a clear, simple diagram

Some Tips

Before we get started on the mechanics, here are a few tips²:

- This is intended to be a conceptual level and “presentation friendly” view of the architecture.
- Tradeoff completeness and precision for clarity. Get it to where it gives someone (technical or non technical) a high level understanding of the architecture, then stop there.
- Save technical details for the more technical views, like a component model.
- You can stack boxes to indicate “there are bunch of components here, but we don’t care about the details.”
- The middleware column is for services shared across the ecosystem and not services specific to a single component or solution.
- You can use a dotted “bounding box” to add a logical grouping to components (like “these are in the cloud”), but don’t ever connect lines to it.
- Keep all the boxes the same size. It makes your diagram look cleaner and more professional. If you show nested components, the internal components should be the same size as other boxes on the diagram.
- Align boxes vertically and horizontally as much as feasible. It makes your diagram look cleaner and more professional.

Drawing with LibreOffice

Getting Started

This assumes you know how to use LibreOffice Draw and provides additional guidance to help you create this specific diagram in LibreOffice. Please checkout these resources to learn more about using LibreOffice:

- <https://www.libreoffice.org/get-help/install-howto/>
- <https://documentation.libreoffice.org/>
- <https://documentation.libreoffice.org/assets/Uploads/Documentation/en/DG7.5/DG75-DrawGuide.pdf>
- <https://help.libreoffice.org/latest>

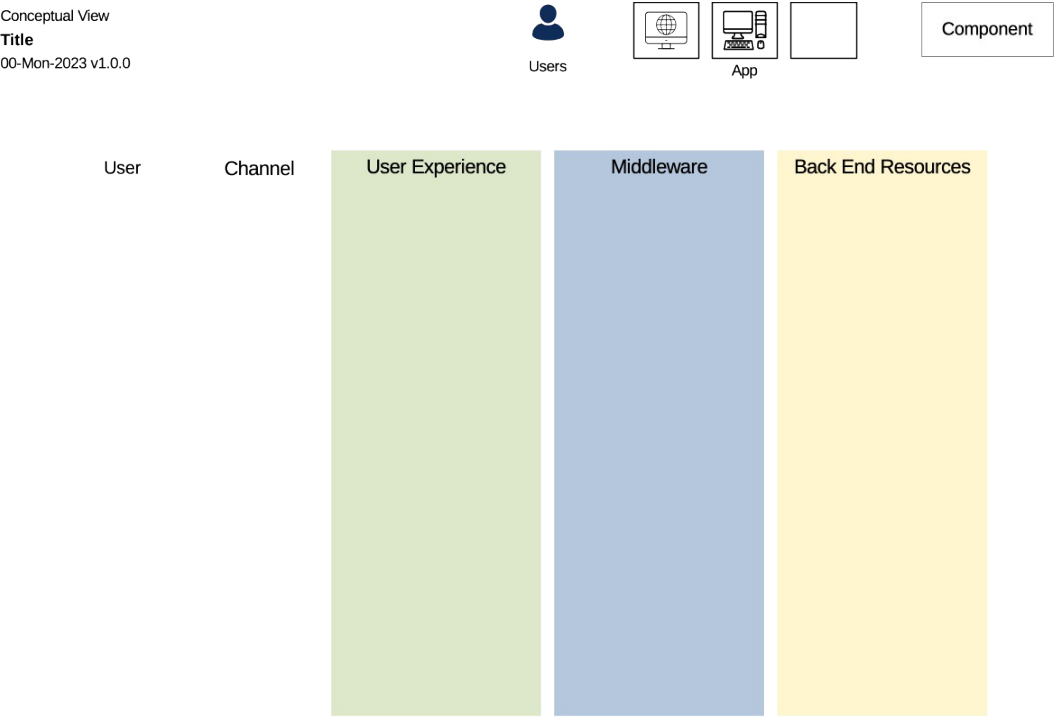
2 Scott Ambler’s book “Elements of UML Style 2.0” has some great general diagramming tips.

It is also often quickest to grab an existing diagram and edit instead of starting from scratch. You can find an existing conceptual diagram here: <https://github.com/hotosm/techdoc/blob/main/overarching-architecture/Overarching%20Conceptual.odg>

Otherwise, start by using the file menu to Create a New Drawing.



Page Setup

If you are not using an existing diagram to start, make a copy of the Conceptual Diagram template and rename it to the name of your solution, e.g. “Tasking Manager Conceptual Diagram.odg.”



In the upper left, is our standard **title block**. Change the word “title” to the title of your diagram. It should describe the scope of your diagram. For example, “Tasking Manager.”

In the upper right are 3 sets of shapes you will use as the “boxes” on the diagram:

	The User icon is used for each User role in scope for the solution.
	<p>The Channel icons identify the method of accessing whatever is providing the user experience.</p> <ul style="list-style-type: none">■ The first icon is the browser channel, used for web based applications■ The second is a desktop application. When using this label the component with the name of the application running on the desktop.■ The final “icon” is a DIY empty box. You put an icon representing your channel inside that box.

Component

The **component** is the most common box on the diagram and is used for all the components in the User Experience, Middleware, and Back End Resource columns.

To use these, simply make a copy, rename and position appropriately. When you are done with the diagram you can delete the shapes off of the top.

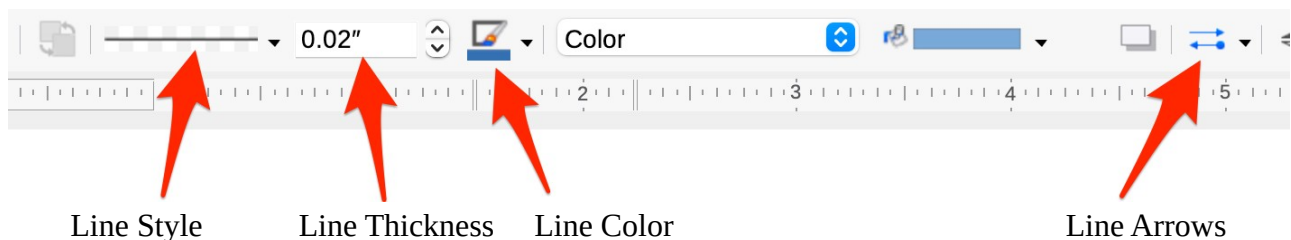
Connecting Components

Once you have some components added to the diagram, you click the connector button on the shape bar on the left, then hover over any component. The connectors will appear on the component you are hovering over.

Click on any connector and holding the mouse button down, move the mouse over the other shape you would like to connect. Release the mouse button when it is hovering over one of the connectors on the other component.

You can use the toolbar at the top to fix the connection style:

- For realtime connections, increase the thickness of the line and change the color to black.
- For batch, increase the thickness of the line, change the style to dashed, change the color to blue, and add arrows to one or both ends depending on the flow of information between the two components.



Pro tip: If you select the right Line attributes with no connector selected, it will become your default for all future connectors.

Clean Up

Once you have your whole diagram laid out, select components and use the right-click “align” function to get them all in alignment accordingly.