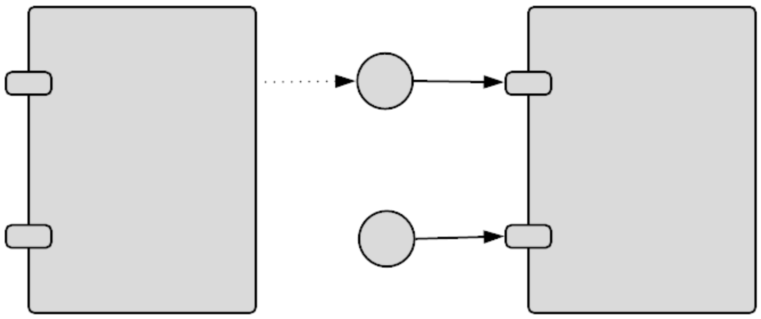
MENU



Component Diagram Tutorial

Contents

Component diagrams are integral to building your software system. Drawn out with UML diagramming software, they help your team understand the structure of existing systems and then build new ones. Keep reading to learn more about component diagrams.

Want to make a Diagram of your own? Try Lucidchart. It's quick, easy, and completely free.

Enter your email address

Make a component diagram

By signing up you agree to our terms & conditions.

What is a UML component diagram?

The purpose of a component diagram is to show the relationship between different components in a system. For the purpose of UML 2.0, the term "component" refers to a module of classes that represent independent systems or subsystems with the ability to interface with the rest of the system.

There exists a whole development approach that revolves around components: component-based development (CBD). In this approach, component diagrams allow the planner to identify the different components so the whole system does what it's supposed to do.

More commonly, in an OO programming approach, the component diagram allows a senior developer to group classes together based on common purpose so that the developer and others can look at a software development project at a high level.

Benefits of component diagrams

Though component diagrams may seem complex at first glance, they are invaluable when it comes to building your system. Component diagrams can help your team:

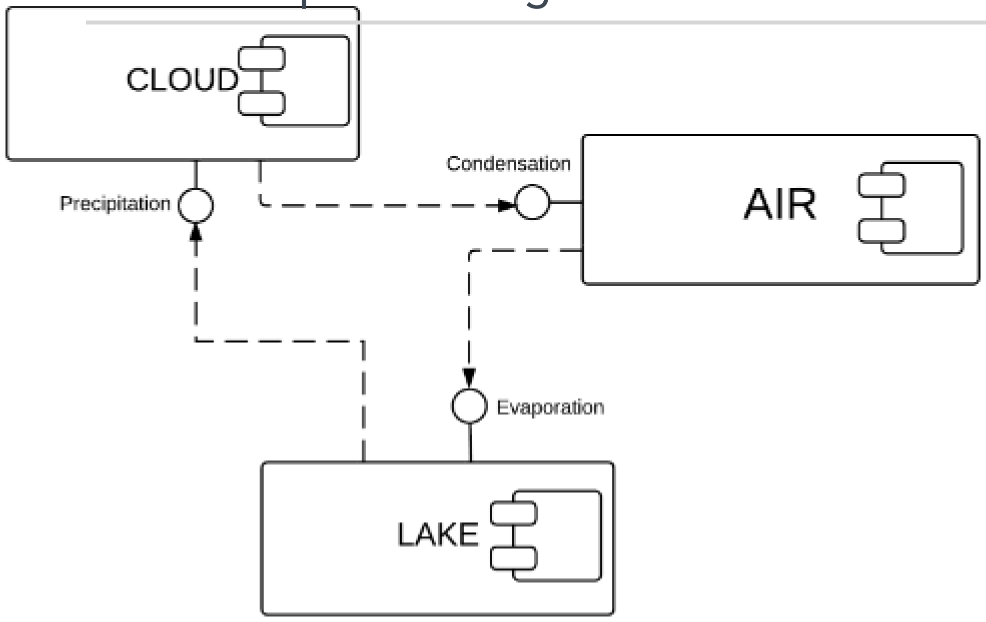
- Imagine the system's physical structure.
- Pay attention to the system's components and how they relate.
- Emphasize the service behavior as it relates to the interface.

How to use component diagrams

A component diagram in UML gives a bird's-eye view of your software system. Understanding the exact service behavior that each piece of your software provides will make you a better developer. Component diagrams can describe software systems that are implemented in any programming language or style.

UML is a set of conventions for object-oriented diagrams that has a wide variety of applications. In component diagrams, the Unified Modeling Language dictates that components and packages are wired together with lines representing assembly connectors and delegation connectors. To learn more about UML and its uses, check out our guide, "What Is UML?"

Component Diagram Tutorial



Diagramming is quick and easy with Lucidchart. Start a free trial today to start creating and collaborating.

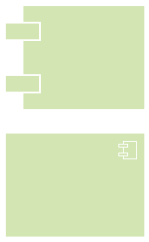
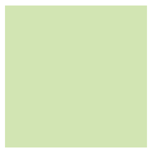
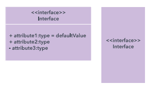
Enter your email address

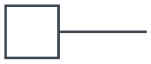
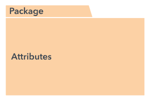
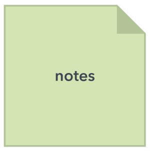
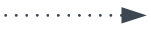
Make a component diagram

By signing up you agree to our terms & conditions.

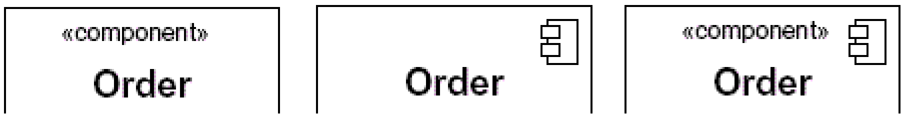
Component diagram shapes and symbols

Component diagrams range from simple and high level to detailed and complex. Either way, you'll want to familiarize yourself with the appropriate UML symbols. The following are shape types that you will commonly encounter when reading and building component diagrams:

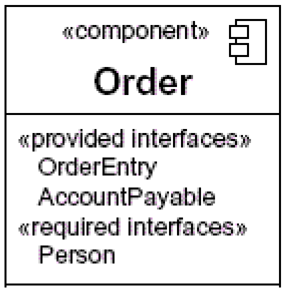
Symbol	Name	Description
	Component symbol	An entity required to execute a stereotype function. A component provides and consumes behavior through interfaces, as well as through other components. Think of components as a type of class. In UML 1.0, a component is modeled as a rectangular block with two smaller rectangles protruding from the side. In UML 2.0, a component is modeled as a rectangular block with a small image of the old component diagram shape.
	Node symbol	Represents hardware or software objects, which are of a higher level than components.
 <<interface>> Interface + attribute1:type = defaultValue + attribute2:type - attribute3:type <<interface>> Interface	Interface symbol	Shows input or materials that a component either receives or provides. Interfaces can be represented with textual notes or symbols, such as the lollipop, socket, and ball-and-socket

Symbol	Name	Description
	Port symbol	Specifies a separate interaction point between the component and the environment. Ports are symbolized with a small square.
	Package symbol	Groups together multiple elements of the system and is represented by file folders in Lucidchart. Just as file folders group together multiple sheets, packages can be drawn around several components.
	Note symbol	Allows developers to affix a meta-analysis to the component diagram.
	Dependency symbol	Shows that one part of your system depends on another. Dependencies are represented by dashed lines linking one component (or element) to another.

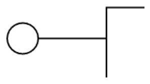
How to use component shapes and symbols



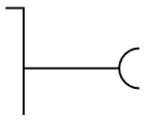
There are three popular ways to create a component's name compartment. You always need to include the component text inside the double angle brackets and/or the component logo. The distinction is important because a rectangle with just a name inside of it is reserved for classifiers (class elements).



As with the class notation, components also have an optional space to list interfaces, similar to the way you add attributes and methods to class notation. Interfaces represent the places where the groups of classes in the component communicate with other system components. An alternative way to represent interfaces is by extending symbols from the component box. Here is a quick rundown of the most commonly used symbols.



Provided interfaces: A straight line from the component box with an attached circle. These symbols represent the interfaces where a component produces information used by the required interface of another component.



Required interfaces: A straight line from the component box with an attached half circle (also represented as a dashed arrow with an open arrow). These symbols represent the interfaces where a

Component Diagram Tutorial

In UML, a component diagram visually represents how the components of a software system relate to one another. To build one, try using Lucidchart's custom component diagram shape library. Component diagrams should communicate:

- The scope of your system
- The overall structure of your software system
- Goals that the system helps human or non-human entities (known as actors) achieve

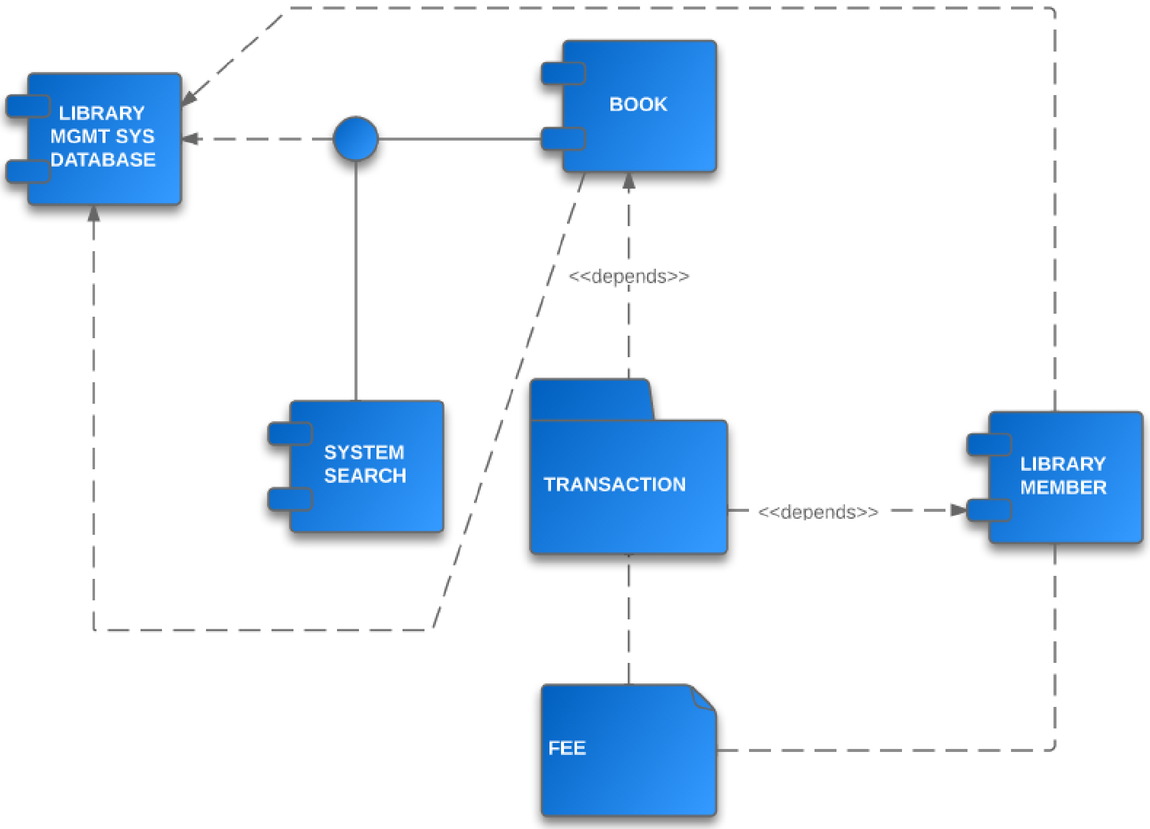
Component diagram examples

UML component diagrams bring simplicity to even the most complex processes. Take a look at the examples below to see how you can map the behaviors of specific processes with component diagrams in UML.

Component diagram for a library management system

Library systems were some of the first systems in the world to become widely run by computers. Today, many of these systems are managed in the cloud by third-party services, rather than internally. Though the term "library system" typically calls to mind a way to monitor printed books, library systems today organize all kinds of data checked in and checked out by users.

These transactions create a network of relationships between the components system functions overall, examine the UML diagram below. You or your team can also use this diagram as a template.

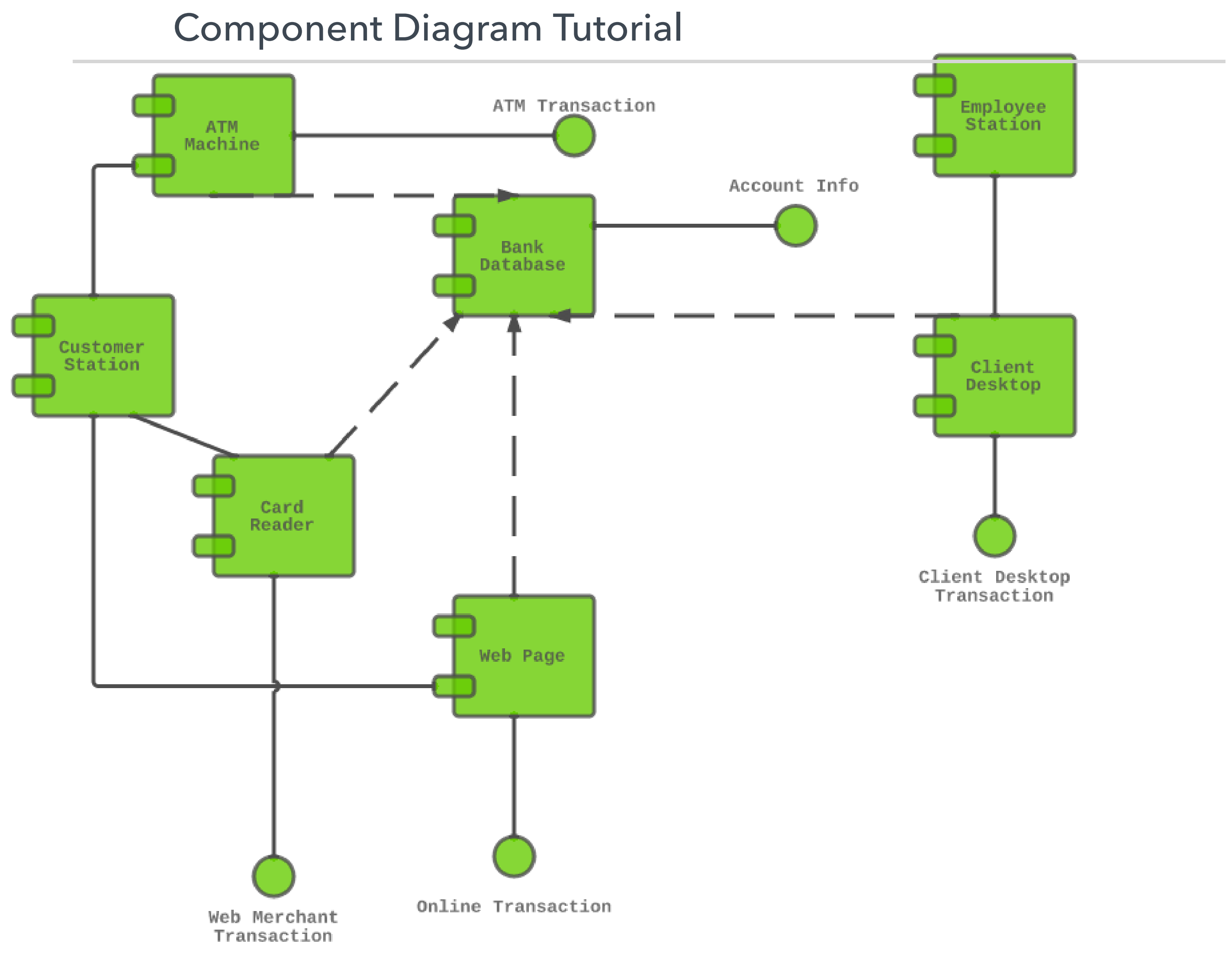


Click here to use this template

Component diagram for an ATM system

A component diagram is similar to a class diagram in that it illustrates how items in a given system relate to each other, but component diagrams show more complex and varied connections that most class diagrams can.

In the diagram below, each component is enclosed in a small box. The dotted lines with arrows show how some components are dependent on others. For example, the card reader, web page, client desktop, and ATM system are all



Click here to use this template

How to make a component diagram

In Lucidchart, you can easily craft an intricate component diagram in UML from scratch. Just follow the steps below:

2. Enable the UML shape library. Click "Shapes" on the left side of the editor, check "UML" in the Shape Library Manager, and click "Save."

3. Select the shape you want from the library you added, and drag the shape from the toolbox to the canvas.

4. Model the process flow by drawing lines between shapes.

If you're looking for a step-by-step breakdown, use our guide on how to draw a component diagram in UML to better assist you.

Additional Resources

UML Diagram Tool

What is UML?

How to Draw a Component Diagram

Other Types of UML Diagrams

UML Templates & Examples

Use Lucidchart to collaborate and create UML diagrams when you start an account for free today! No plugins or download required.

Want to make a Diagram of your own? Try Lucidchart. It's quick, easy, and completely free.

Enter your email address

Make a component diagram

Component Diagram Tutorial