

Learn to code — free 3,000-hour curriculum

AUGUST 21, 2023 / #ARCHITECTURE

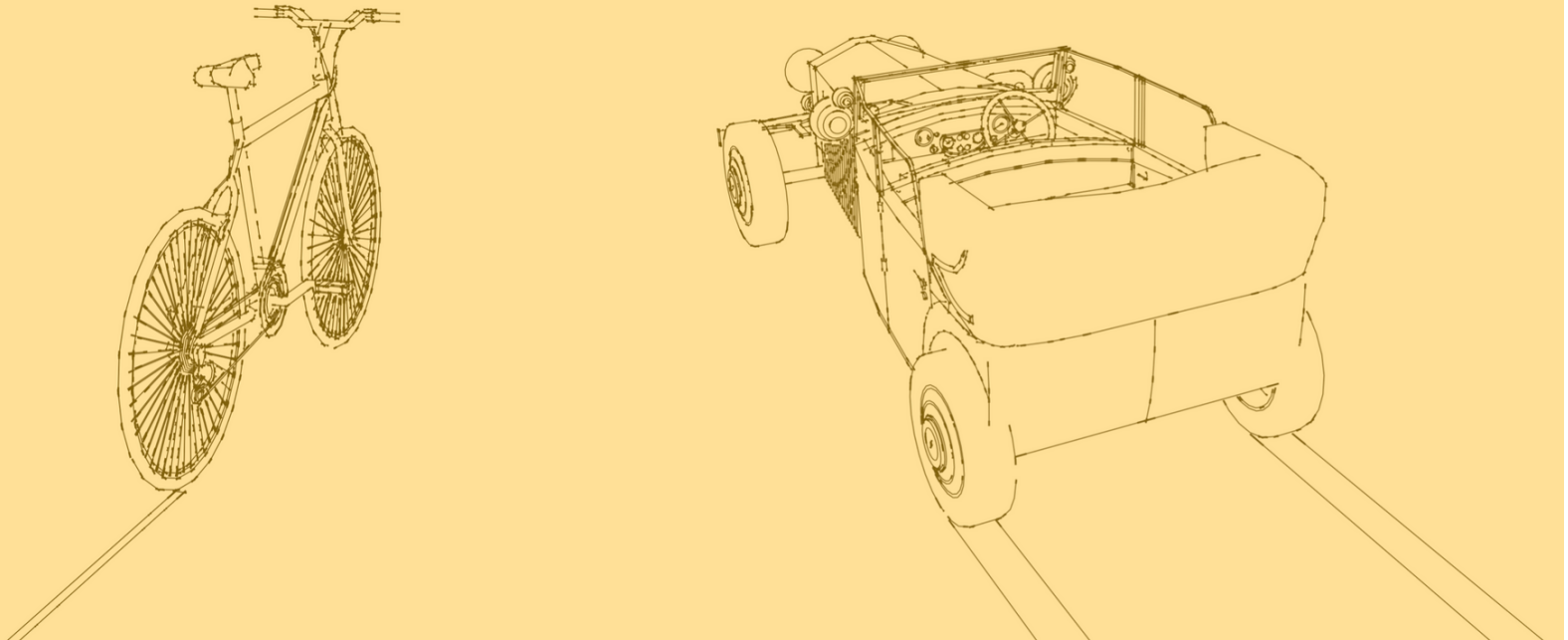
Stateful vs Stateless Architecture – Explained for Beginners



Daniel Adetunji

Learn to code — free 3,000-hour curriculum

architectures explained



In programming, "state" refers to the condition of a system, component, or

Learn to code — free 3,000-hour curriculum

As a simple example, if you are shopping on amazon.com, whether you are currently logged into the site or if you have anything stored in your cart are some examples of state.

State represents the data that is stored and used to keep track of the current status of the application. Understanding and managing state is crucial for building interactive and dynamic web applications.

The concept of a “state” crosses many boundaries in architecture. Design patterns

Learn to code — free 3,000-hour curriculum

stateful or stateless. But the underlying principle of “state” cutting across all of these domains remains the same.

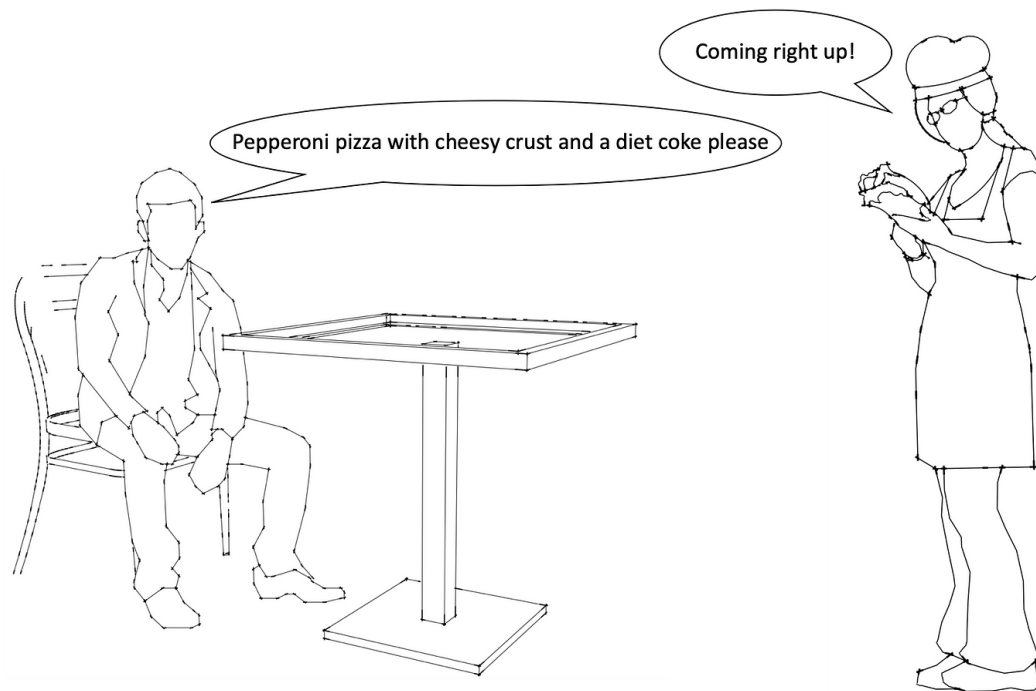
This article will explain what state means. It will also explain stateful and stateless architectures with some analogies and the benefits and tradeoffs of both.

What is Stateful Architecture?

Imagine you go to a pizza restaurant to eat some food. In this restaurant, there is only a

Learn to code — free 3,000-hour curriculum

your preferences based on past orders, like what type of pizza crust you like or toppings you are allergic to, and so on.



Learn to code — free 3,000-hour curriculum

All of these pieces of information that the waiter writes down in their notepad are the customer's state. Only the waiter serving you has access to this information. If you want to make a change to your order or check how its coming along, you need to speak to the same waiter that took your order. But since there is only one waiter, that is not a problem.

Now, suppose the restaurant starts to get busier. Your waiter has to respond to other guests so more waiters are called to work. You now want to check the status of your order and

Learn to code — free 3,000-hour curriculum

different from the one who initially took your order.

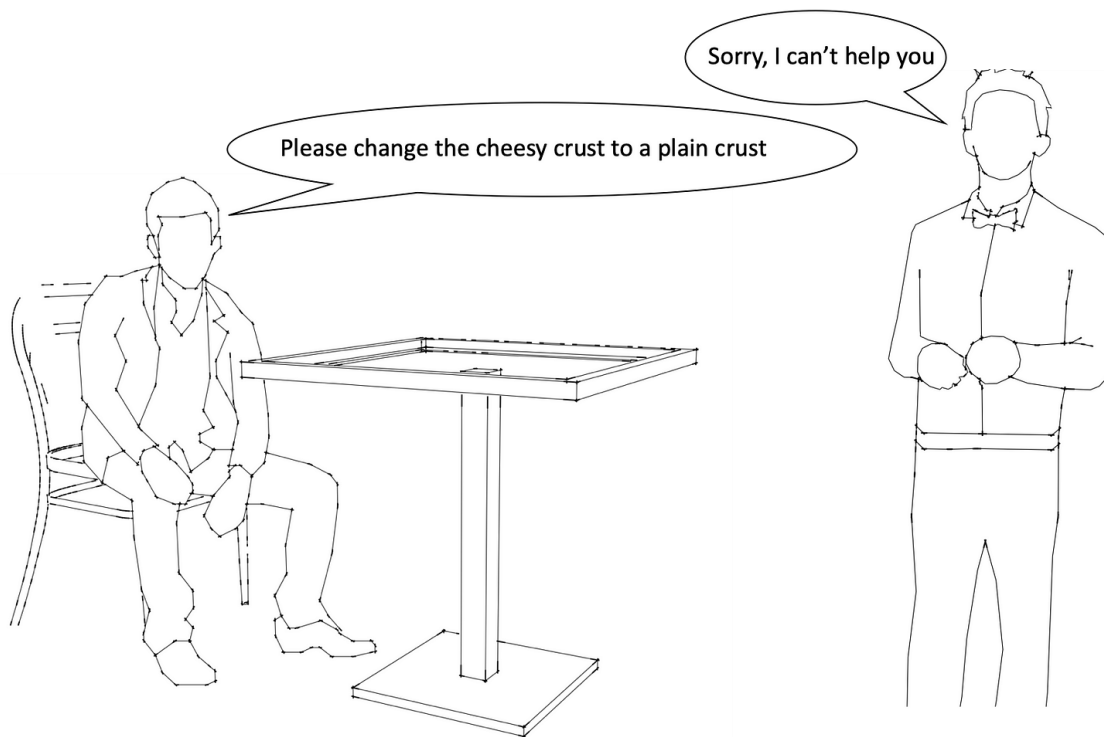


Illustration showing a different waiter being unable to help the customer change their order

Learn to code — free 3,000-hour curriculum

order, that is your state. Naturally, they will not be able to check the status of your order or make changes to it. A restaurant that operates like this, where only the waiter that initially took your order can give you updates about it, or make changes to it, follows a stateful design.

Similarly, a stateful application will have a server that remembers clients' data (that is, their state). All future requests will be routed to the same server using a load balancer with sticky sessions enabled. In this way, the server is always aware of the client.

[Forum](#)[Donate](#)[Learn to code — free 3,000-hour curriculum](#)

balancer. Since the application state is maintained on the servers, the users must always be routed to the same server for every single request in order to preserve state.

Learn to code — free 3,000-hour curriculum

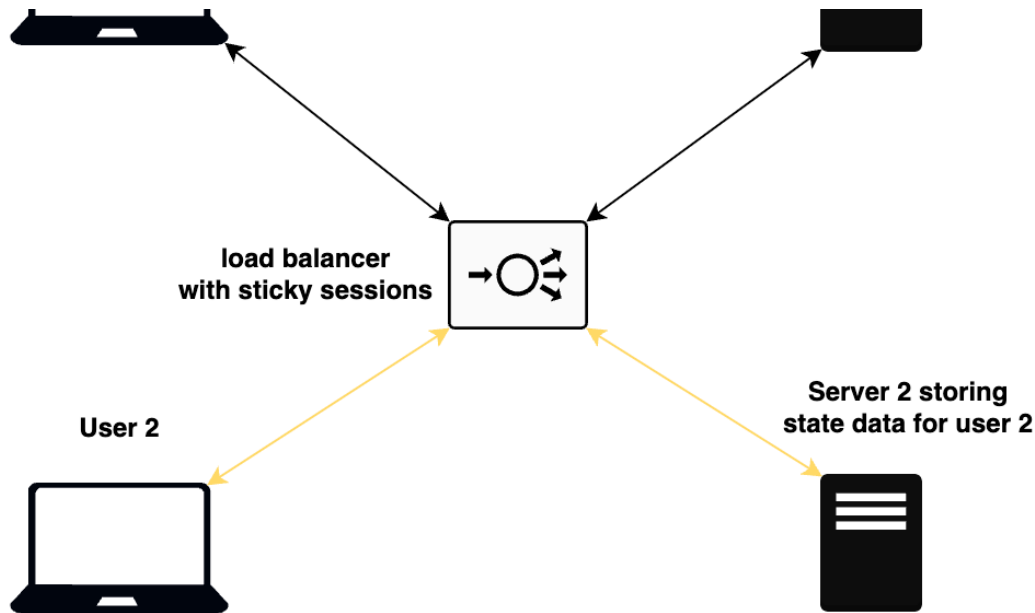


Diagram showing how a stateful application works

Sticky sessions is a configuration that allows the load balancer to route a user's requests consistently to the same backend server for

Learn to code — free 3,000-hour curriculum

from a user can be directed to any available backend server in a round-robin or other load distribution pattern.

What is the problem with a stateful architecture? Imagine a restaurant run in this manner. While it may be ideal and easy to implement for a small, family run restaurant with only a few customers, such a design is **not fault tolerant** and **not scalable**.

What happens if the waiter who took a customer's order has an emergency and needs to leave? All the information regarding that

Learn to code — free 3,000-hour curriculum

new waiter brought in to replace the old one has no knowledge of previous orders. This is a design that is not fault tolerant.

Also, having to distribute requests so that the same customer can only speak to the same waiter means that the load on different waiters is not equally distributed. Some waiters will be overwhelmed with requests if you have a very demanding customer who always modifies or adds things to their order. Some of the other waiters will have nothing to do, and can't step in to help. Again, this is a non scalable design.

Learn to code — free 3,000-hour curriculum

tolerant and not scalable. A server failure will lead to loss of state data. So, if a user is logged in and about to checkout for a large order on Amazon.com for example, the user will be forced to re-authenticate and the user's basket will be empty. They would have to log in again and fill up their basket from scratch – a poor user experience.

Scalability will also be difficult to achieve during peak times like Black Friday with a stateful design. New servers will be added to the auto scaling group but since sticky sessions are enabled, clients will be routed to the same

Learn to code — free 3,000-hour curriculum

poor user experience.

Stateless architectures solve a lot of these problems.

What is Stateless Architecture?

“Stateless” architecture is a confusing term, as it implies the system is without state. A stateless architecture does not, however, mean that state information is not stored. It simply means that state information is stored outside

Learn to code — free 3,000-hour curriculum

Bringing back the restaurant analogy, waiters in a stateless restaurant can be thought of as having perfectly forgetful memories. They do not recognise old customers, and can't recall what you ordered or how you like your pizza. They will simply take note of customers' orders on a separate system, say a computer, that is accessible by all the waiters. They can then revert back to the computer to get details of an order and make changes to it as required.

Learn to code — free 3,000-hour curriculum

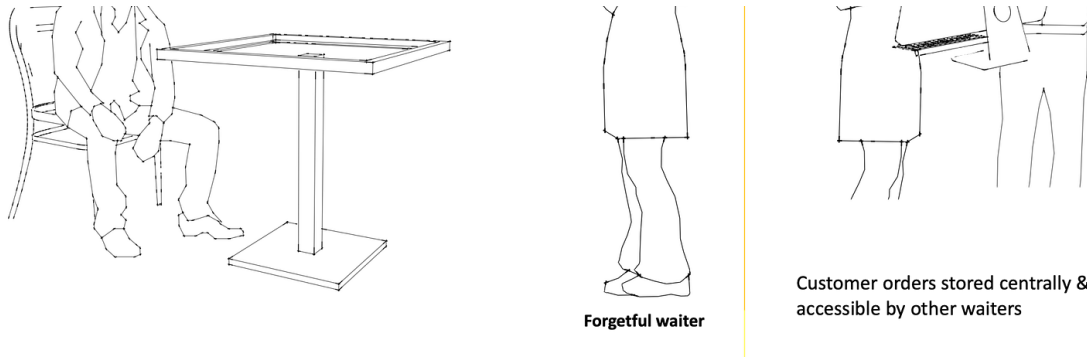


Illustration of a "forgetful" waiter taking an order and then consulting the computer about orders

By storing the 'state' of a customer's order on a central system accessible by other waiters, any waiter can serve any customer.

In a stateless architecture, HTTP requests from a client can be sent to any of the servers.

Learn to code — free 3,000-hour curriculum

creates a fault tolerant and scalable architecture since web servers can be added or removed as needed, without impacting state data.

The load will also be equally distributed across all servers, since the load balancer will not need a sticky session configuration to route the same clients to the same servers.

The diagram below shows two different users trying to access a web server through a load balancer. Since the application state is maintained separately from the servers, the

Learn to code — free 3,000-hour curriculum

an external database accessible by both servers.

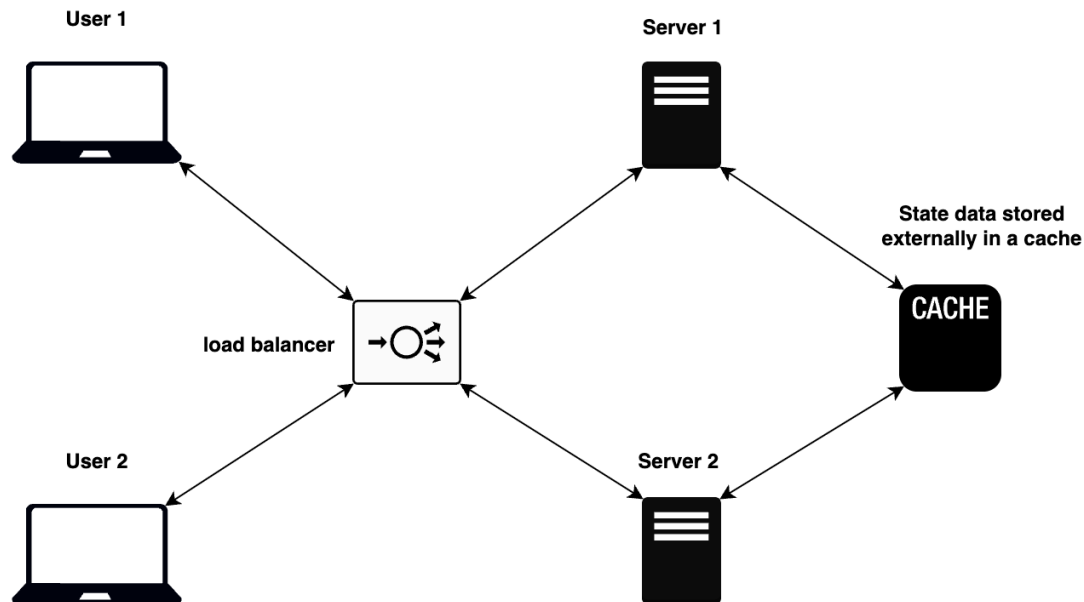


Illustration showing a diagram of stateless architecture

Learn to code — free 3,000-hour curriculum

data in-memory improves read and write times, compared to storing it on disk, as explained [here](#).

Bringing it Together

This article has described how stateful and stateless web applications work and the trade-offs of both. But the principle of statefulness and statelessness apply beyond web applications.

If we look at network protocols as an example, HTTP is a stateless protocol. This means that each HTTP request from a client to a server is

Learn to code — free 3,000-hour curriculum

treats each request as a separate and isolated transaction, and it doesn't inherently maintain information about the state of the client between requests.

State is either maintained on the servers (stateful architecture) or in a separate database outside the servers (stateless architecture). The HTTP protocol itself does not maintain state.

Unlike the stateless nature of HTTP, the TCP protocol is connection-oriented and stateful. It establishes a connection between two devices

Learn to code — free 3,000-hour curriculum

connection is terminated.

The same logic applies to firewalls as well, which can be stateful or stateless.

In AWS, a security group is a virtual firewall that controls inbound and outbound traffic for virtual machines or instances within a cloud environment. Security groups are stateful. When you allow a specific incoming traffic flow, the corresponding outgoing traffic flow is automatically allowed as well. In other words, the state of the connection is tracked.

Learn to code — free 3,000-hour curriculum

at the subnet level in AWS. NACLs are stateless. Being stateless means that you must explicitly define rules for both incoming and outgoing traffic.

Unlike security groups, where response traffic is automatically allowed when you allow incoming traffic, NACLs require you to define separate rules for inbound and outbound traffic.

Functions and design patterns can also be stateful or stateless.

Learn to code — free 3,000-hour curriculum

knowledge or previous calls or requests, while something that is stateless has no memory or knowledge of previous calls or requests.

Hopefully you now have a good grasp of how stateful and stateless applications work and can decide which option is best for your applications.



Daniel Adetunji

Read [more posts](#).

[Forum](#)[Donate](#)[Learn to code — free 3,000-hour curriculum](#)

Learn to code for free. freeCodeCamp's open source curriculum has helped more than 40,000 people get jobs as developers. [Get started](#)

freeCodeCamp is a donor-supported tax-exempt 501(c)(3) charity organization (United States Federal Tax Identification Number: 82-0779546)

Our mission: to help people learn to code for free. We accomplish this by creating thousands of videos, articles, and interactive coding lessons - all freely available to the public. We also have thousands of freeCodeCamp study groups around the world.

Donations to freeCodeCamp go toward our education initiatives, and help pay for servers, services, and staff.

You can [make a tax-deductible donation here](#).

Trending Guides

[JS isEmpty Equivalent](#)[Coalesce SQL](#)[Rename Column in Pandas](#)[Submit a Form with JS](#)[Python join\(\)](#)[Delete a File in Python](#)

[Forum](#)[Donate](#)[Learn to code — free 3,000-hour curriculum](#)[Add to Dict in Python](#)[SOLID Principles](#)[Create a File in Terminal](#)[Java For Loop Example](#)[Sort a List in Java](#)[Clear Formatting in Excel](#)[Matplotlib Figure Size](#)[For Loops in Python](#)[Accounting Num Format
Excel](#)[Database Normalization](#)[JavaScript 2D Array](#)[Check if File Exists Python](#)[Nested Lists in Python](#)[SQL CONVERT Function](#)[Iterate Over Dict in Python](#)

Our Charity

[About](#) [Alumni Network](#) [Open Source](#) [Shop](#) [Support](#) [Sponsors](#) [Academic Honesty](#)[Code of Conduct](#) [Privacy Policy](#) [Terms of Service](#) [Copyright Policy](#)