

[Back To Module Home](#)

# Distributed Systems

0% completed

## Introduction to Distributed Systems

Getting Started

Fallacies of Distributed Computing

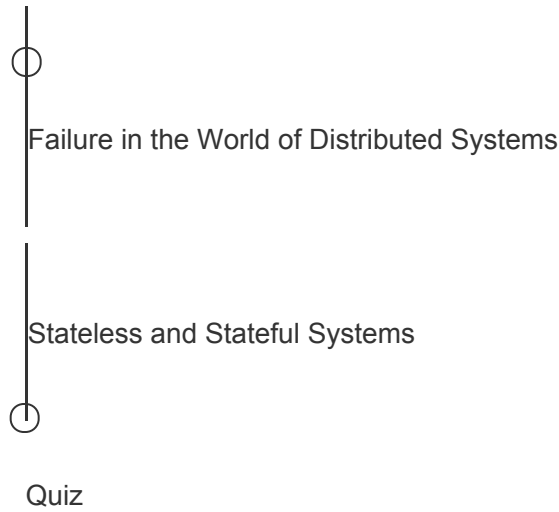
Difficulties Designing Distributed Systems

Measures of Correctness in Distributed Systems

System Models

Types of Failures

The Tale of Exactly-Once Semantics



## Basic Concepts and Theorems

## Conclusion

## Mark Module as Completed

# Stateless and Stateful Systems

Let's see the difference between stateless and stateful systems.

## We'll cover the following

- Stateless system
  - Examples
- Stateful systems
  - Example
- Some interesting observations
  - Benefits of stateless systems over stateful systems

We can say that a system belongs in one of the two following categories:

- Stateless systems
- Stateful systems

## Stateless system#

A stateless system maintains no state of what happened in the past and performs its capabilities purely based on the inputs we provide to it.

## Examples#

A contrived stateless system receives a set of numbers as input, calculates their maximum, and returns it as a result. These inputs are either direct or indirect. *Direct inputs* are inputs that are included in the request, while *indirect inputs* are inputs that are potentially received from other systems to fulfill the request.

Imagine a service that calculates the price of a specific product by retrieving its initial price and any currently available discounts from some other services, and then performing the necessary calculations with this data. This service is still stateless.

Stateless systems

## Stateful systems#

Stateful systems are responsible for maintaining and mutating a state. Their results depend on this state.

# Example#

Imagine a system that stores the ages of all the employees of a company, and we can ask it the maximum age. This system is stateful since the result depends on the employees we register in it.

Stateful systems

# Some interesting observations#

- Stateful systems are beneficial in real life because computers are much more capable than humans of storing and processing data.
- Maintaining state involves additional complexity. For example, we must decide what's the most efficient way to store and process it, how to perform back-ups, etc.
- As a result, it's usually wise to create an architecture that contains clear boundaries between stateless components (which perform business capabilities) and stateful components (which handle data).

# Benefits of stateless systems over stateful systems#

Stateless distributed systems are much easier to design, build and scale, compared to stateful ones.

The main reason for this is that we consider all the nodes (e.g., servers) of a stateless system identical. This makes it a lot easier for us to balance traffic between them, and scale by adding or removing servers.

However, stateful systems present many more challenges. As different nodes can hold

different pieces of data, they require additional work. They need to direct traffic to the right place and ensure each instance is in sync with the others.

As a result, some of the course's examples include stateless systems. However, the most challenging problems we cover in this course mainly concern stateful systems.

**Back**

Failure in the World of Distributed Sys...

**Next**

Quiz

Mark as Completed

---

Report an Issue