

# Two Tier Applications

In this lesson, we will learn about the Two Tier applications.

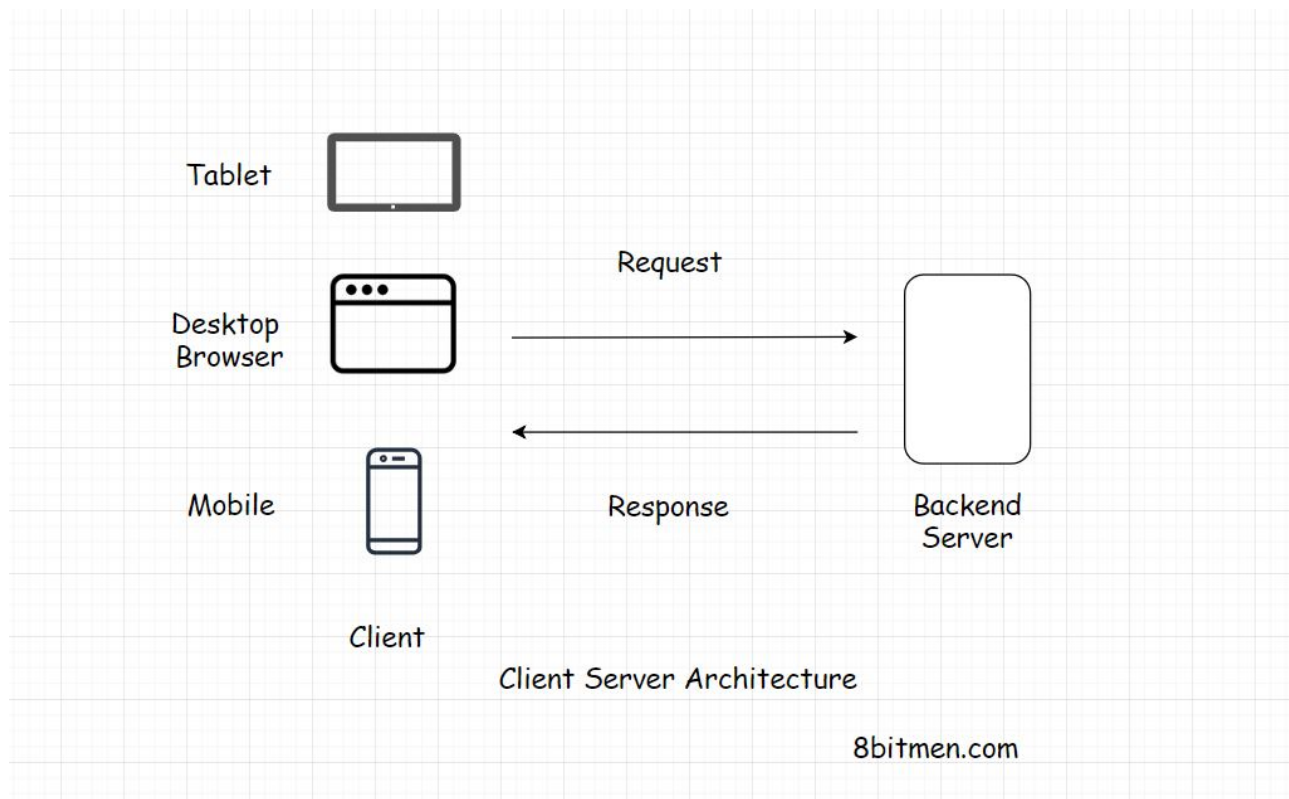
We'll cover the following



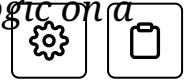
- Two Tier Application
- The Need For Two Tier Application

## Two Tier Application #

A **Two-tier** application involves a *client* and a *server*. The *client* would contain the *user interface* & the *business logic* in one machine. And the backend *server* would be the *database* running on a different machine. The database server is hosted by the business & has control over it.



*Why the need for two-tier applications? Why not host the business logic on a different machine & have control over it?*



*Also, again isn't the application code vulnerable to being accessed by a third person?*

## The Need For Two Tier Application #

Well, yes!! But there are use cases where *two-tier* applications come in handy, for instance, a to-do list app or a similar planner or a productivity app.

In these scenarios, it won't cause the business significant harm, even if the code is accessed by a third person. On the contrary, the upside is since the code & the user interface reside in the same machine, there are fewer network calls to the backend server which keeps the latency of the application low.

The application makes a call to the *database* server, only when the user has finished creating his to-do list & wants to persist the changes.

Another good example of this is the online browser & app-based games. The game files are pretty heavy, they get downloaded on the client just once when the user uses the application for the first time. Moreover, they make the network calls only to keep the game state persistent.

Also, fewer server calls mean less money to be spent on the servers which is naturally economical.



Though, it largely depends on our business requirements & the use case if we want to pick this type of tier when writing our service.



We can either keep the *user interface* and the *business logic* on the *client* or move the *business logic* to a dedicated *backend server*, which would make it a *three-tier* application. Which I am going to discuss up next.

Single Tier Applications

Three Tier Applications

☒ Mark as Completed



-  Report an Issue
-  Ask a Question  
([https://discuss.educative.io/tag/two-tier-applications\\_\\_different-tiers-in-software-architecture\\_\\_web-application--software-architecture-101](https://discuss.educative.io/tag/two-tier-applications__different-tiers-in-software-architecture__web-application--software-architecture-101))