**The three tiers in detail**

<https://www.ibm.com/cloud/learn/three-tier-architecture>

**Presentation tier**

The presentation tier is the user interface and communication layer of the application, where the end user interacts with the application. Its main purpose is to display information to and collect information from the user. This top-level tier can run on a web browser, as desktop application, or a graphical user interface (GUI), for example. Web presentation tiers are usually developed using HTML, CSS and JavaScript. Desktop applications can be written in a variety of languages depending on the platform.

**Application tier**

The application tier, also known as the logic tier or middle tier, is the heart of the application. In this tier, information collected in the presentation tier is processed - sometimes against other information in the data tier - using business logic, a specific set of business rules. The application tier can also add, delete or modify data in the data tier.

The application tier is typically developed using Python, Java, Perl, PHP or Ruby, and communicates with the data tier using [API](https://www.ibm.com/cloud/learn/api) calls.

**Data tier**

The data tier, sometimes called database tier, data access tier or back-end, is where the information processed by the application is stored and managed. This can be a [relational database management system](https://www.ibm.com/cloud/learn/relational-databases) such as [PostgreSQL](https://www.ibm.com/cloud/learn/postgresql), MySQL, MariaDB, Oracle, DB2, Informix or Microsoft SQL Server, or in a [NoSQL](https://www.ibm.com/cloud/learn/nosql-databases) Database server such as Cassandra, [CouchDB](https://www.ibm.com/cloud/learn/couchdb) or [MongoDB](https://www.ibm.com/cloud/learn/mongodb).

In a three-tier application, all communication goes through the application tier. The presentation tier and the data tier cannot communicate directly with one another.

**Tier vs. layer**

In discussions of three-tier architecture, *layer* is often used interchangeably – and mistakenly – for *tier*, as in 'presentation layer' or 'business logic layer.'

They aren't the same. A 'layer' refers to a functional division of the software, but a 'tier' refers to a functional division of the software that runs on infrastructure separate from the other divisions. The Contacts app on your phone, for example, is a *three*-*layer* application, but a *single-tier* application, because all three layers run on your phone.

The difference is important, because layers can't offer the same benefits as tiers.