

Multi-Tenant Architecture: What You Need To Know

Written by Sandra Suszterova | Jul 27, 2023

Share [in](#) [Twitter](#) [Facebook](#)

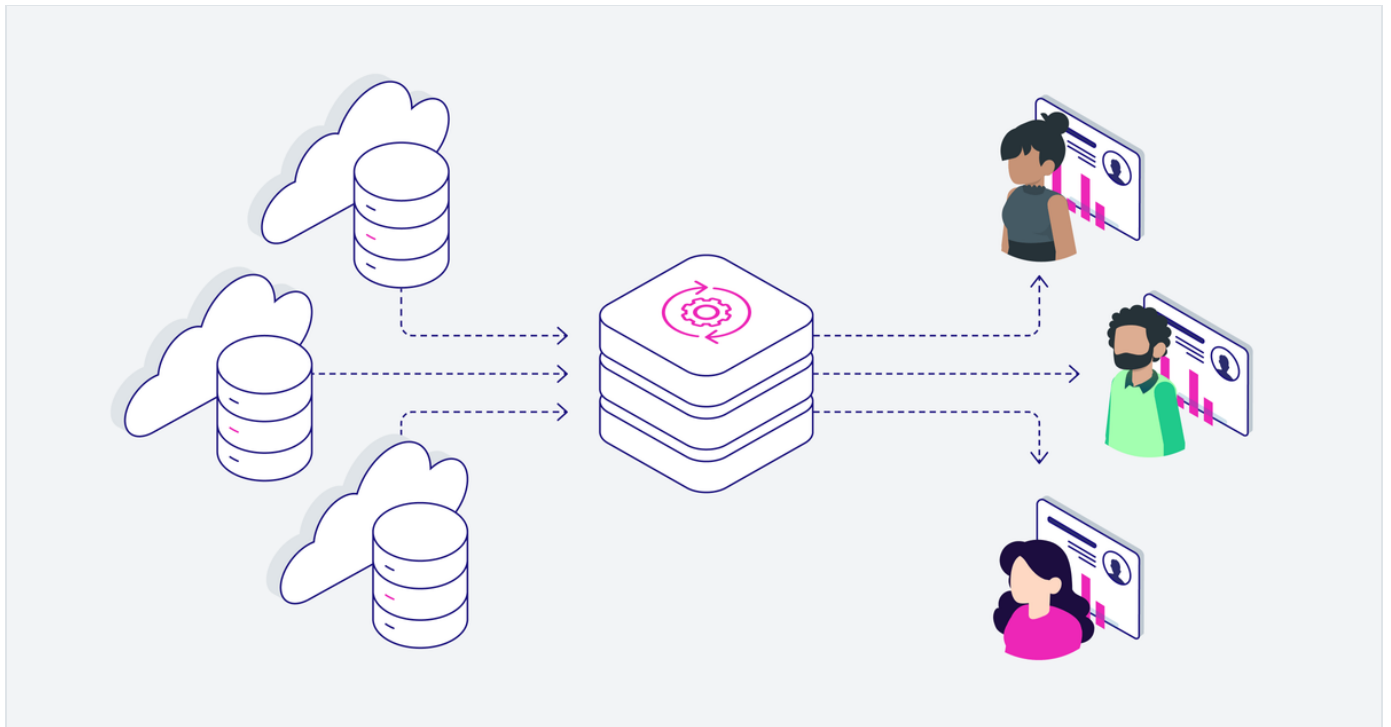


Table of Contents

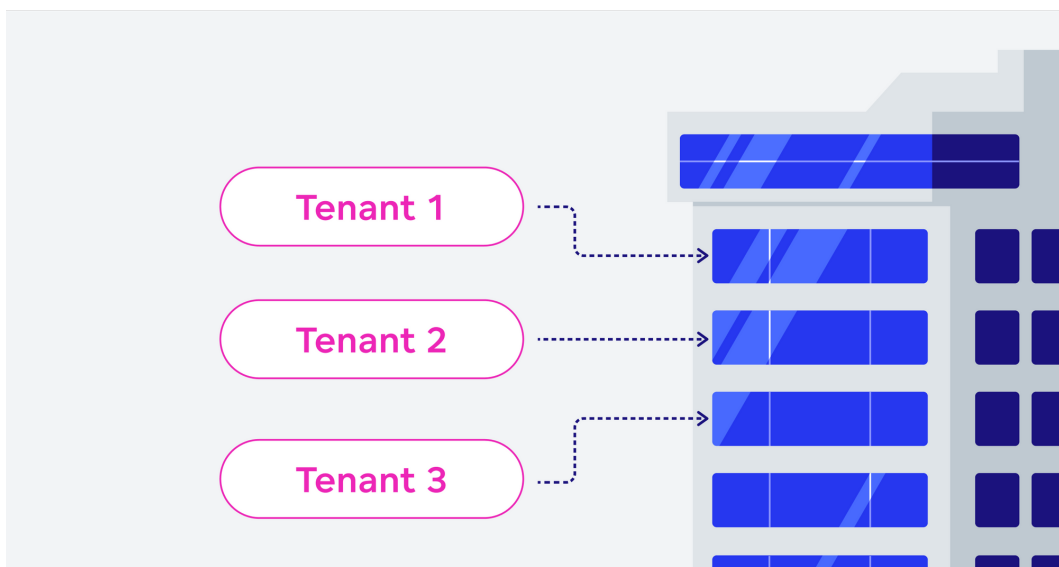
1. [What Is Multi-Tenant Architecture?](#)
2. [How Does Multi-Tenant Architecture Work?](#)
3. [How Is Multi-Tenant Architecture Different From Single-Tenant Architecture?](#)
4. [What Are the Benefits of Multi-tenant Architecture?](#)
5. [What Are the Disadvantages of Multi-tenant Architecture?](#)
6. [Multi-Tenant Architecture Examples](#)
7. [Multi-Tenant Architecture in Analytics](#)
8. [Multi-Tenant Architecture and GoodData](#)
9. [Want To Try It Yourself?](#)

Multi-tenant architecture, also known as [multitenancy](#), has become an increasingly important concept in today's application development paradigm, largely due to the boom in cloud computing and software-as-a-service (SaaS) business models. Where SaaS allows us to deliver applications over the internet as a service, multitenancy enables companies to develop and scale applications faster and more efficiently by allowing them to share resources among different users.

But what exactly is multitenancy and why are companies building their applications using multi-tenant architecture? Here we answer these questions and introduce the main benefits of multi-tenant architecture.

What Is Multi-Tenant Architecture?

Multi-tenant architecture is a design approach that enables multiple user groups — referred to as tenants — access to one instance of an application or system. This is much like an apartment building, where various tenants have their own apartments. All the apartments are private and secure but share the same infrastructure.



An apartment block as an example of multitenancy

This type of architecture enables companies to save money on infrastructure costs by using the same hardware resources for several instances of the same application for multiple tenants. It is especially beneficial for SaaS companies because it allows them to scale their

applications and remain cost-effective.

How Does Multi-Tenant Architecture Work?

Multi-tenant architecture creates distinct, isolated environments within a single physical infrastructure, such as a virtual machine, server, or cloud platform. This is accomplished by partitioning the data storage and processing; providing each tenant with their own dedicated space in the system. A tenant interacts with the application and can access their own data.

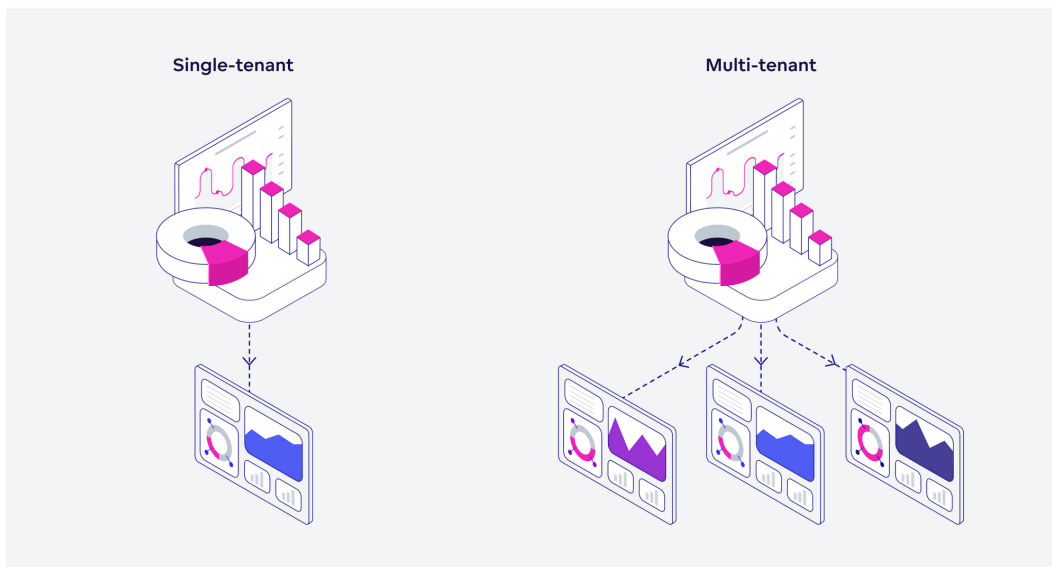
Tenants refers to:

- Users and user groups inside your company (departments, teams, individual employees)
- Users and user groups outside your company (vendors, business partners)
- Customers (subscribers or clients) who purchase your product

Because every multi-tenant environment is separated from one another, they can be customized to meet the needs of each individual tenant without impacting other environments. Tenants can personalize features such as user interface design and data security settings for their environment. Additionally, different sets of rules can be applied to each domain in terms of access control, resource allocation, and feature availability.

How Is Multi-Tenant Architecture Different From Single-Tenant Architecture?

The main difference between multitenancy and single-tenancy is how they handle resource sharing across multiple users or tenants. In single-tenant architecture, each tenant has a dedicated instance with their own set of hardware and software resources. This means each user group must pay for their own hardware and software costs, as well as any maintenance costs associated with running a dedicated instance for each tenant.



Single-tenant vs. multi-tenant architecture

In multi-tenant architecture, tenants share a single set of hardware and software resources. This results in lower overhead costs since maintenance fees are shared across all user groups who utilize those shared resources.

The table below shows the main differences between single-tenant and multi-tenant architecture.

Single-tenant architecture	Multi-tenant architecture
Each tenant has a dedicated instance of the application and its resources	Multiple tenants share a single instance of the application and its resources
Scaling requires provisioning and managing individual instances for each tenant	Easier scalability as multiple tenants can be accommodated within a single instance
More secure due to isolated environments	Security measures involve ensuring data isolation and preventing unauthorized access
Less cost-efficient: hardware and software for each user is paid	More cost-efficient thanks to shared infrastructure and platform

Single-tenant architecture	Multi-tenant architecture
separately	
Independent maintenance and upgrades for each tenant's instance	Centralized maintenance, updates, and upgrades that apply to all tenants
Slower deployment: each tenant requires individual setup and configuration	Faster deployment: new tenants can be added within the existing setup

Note that with single-tenancy there are often additional complexities related to managing different versions of code across multiple instances. Companies can face a situation where different tenants use different versions of the application. Whereas with multitenancy, all users use a single version of code, which considerably simplifies management over time — updates can be deployed across all users automatically instead of having them applied separately.

What Are the Benefits of Multi-tenant Architecture?

Multi-tenant architecture offers several benefits for businesses looking to move their applications to the cloud or deploy SaaS solutions. Some of the most notable advantages include:

Scalability

With one application being able to serve multiple tenants, companies can easily scale up or down quickly and efficiently without managing multiple dedicated systems. This [scalability](#) makes it much easier to accommodate seasonal spikes in demand or adjust usage based on customer needs.

Cost Savings

Multitenancy allows companies to reduce costs by leveraging shared resources instead of having separate systems dedicated to one tenant at a time. This means

businesses needn't invest in additional servers or hardware to meet customer demands, helping to save money in the long run.

Increased Efficiency

Multi-tenant architecture increases efficiency by allowing multiple tenants to share resources such as compute power and storage space without sacrificing performance. This makes it easier for businesses to manage large workloads without having to invest in additional hardware or software licenses.

Easy Maintenance/Management

Multi-tenant architecture is much easier to maintain and manage than single-tenant architecture because all tenants share the same underlying codebase and platform. This means businesses don't have to worry about managing multiple versions of their application across different platforms — they can focus on managing one version across all tenants.

Customization

In multi-tenant architecture, tenants can customize their assigned environment to meet their individual needs and preferences by setting up unique configurations, options, and branding. This allows tenants to effectively use the shared multi-tenant application for their specific use cases, such as filtering data by different branches or locations and creating specified metrics and dashboards.

Tenant Privacy

Each tenant receives access to a dedicated space where they can analyze their data and view their dashboards. This space can either be separated from other tenants' access or accessible to an authorized user group; where each tenant is restricted by their assigned role and dedicated permission in order to manage the analytics environment.

What Are the Disadvantages of Multi-tenant Architecture?

Multi-tenant architecture continues to gain popularity among SaaS companies. But, it is important to note that

alongside its numerous advantages, there are some disadvantages that require careful attention.

Security Issues If Incorrectly Implemented

Poorly implemented multitenancy can lead to issues such as unauthorized access and data misuse. To nullify such threats, these applications require careful design and management to ensure tenant isolation and secure data storage.

The most important thing is to authorize the access of end users. Applications that can be used for this purpose include Okta, Auth0, and Azure Active Directory. Such applications enable organizations to manage authentication, authorization, and identity management capabilities.

Additional Knowledge Requirements

Serving multiple clients in a single instance of an application/database means implementing additional logic for tenant separation, data filtering, and tenant identification to ensure proper data isolation and security. If the setup of a multi-tenant environment is well-documented and the necessary resources are made available to customers, it can potentially help reduce the knowledge required during the setup process. However, if customers encounter any difficulties or have questions, it is important that the application providers have a team of professionals ready to assist and address customer issues.

System Outages

Multi-tenant applications rely heavily on the application provider for maintenance and upgrades. Any technical issues that arise on the provider's side can potentially impact all tenants, affecting aspects such as service availability, system upgrades, and other global processes.

To mitigate issues, many providers choose to build their applications as a set of microservices that are deployed, for example, in Kubernetes — an open source platform that automates the deployment, scaling, and management of containerized applications. Microservice architecture can bring benefits such as scalability, fault

isolation, and independent deployment. This can help minimize outages that affect end users and ensure the smooth operation of the application.

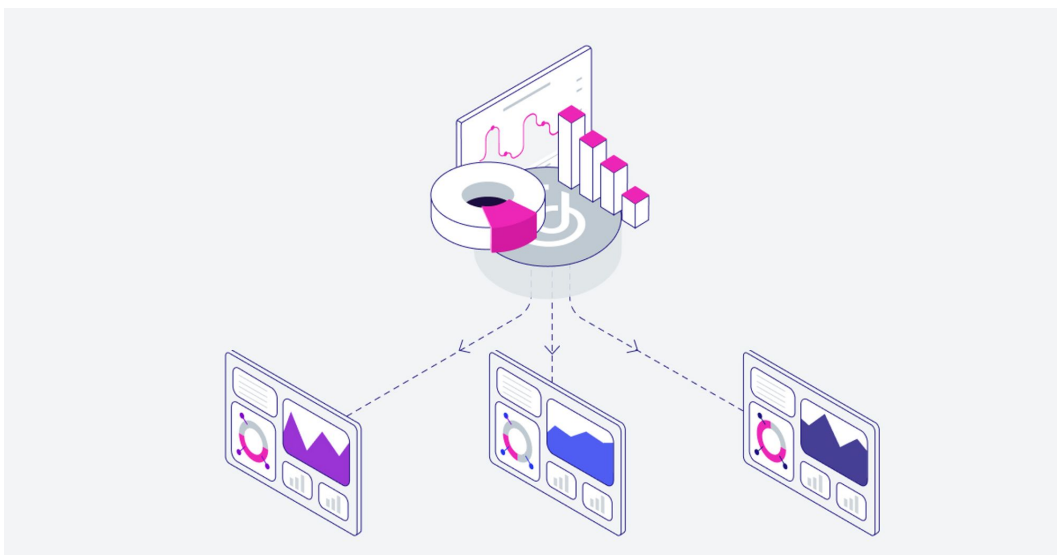
Multi-Tenant Architecture Examples

A good example of multi-tenant architecture is in cloud computing and software-as-a-service applications. Public clouds such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP) provide the same infrastructure for multiple different users simultaneously. Each customer can manage their own virtual machines, storage space, databases, etc., in their cloud instance, but cannot access other customers' services or data.

Similarly, SaaS applications like Gmail, Dropbox, and Netflix allow multiple customers to use a single instance without compromising other users' experiences. When we read emails in Gmail, save files in Dropbox, or watch our favorite show on Netflix, countless other users are doing the same thing at the same time. Yet they remain invisible to us and vice versa — this is the power of multitenancy.

Multi-Tenant Architecture in Analytics

[Multi-tenant analytics](#) is a type of analytics use case that leverages multi-tenant architecture. Analytics and business intelligence platforms that support multitenancy allow companies to share dashboards and report capabilities to different user groups in a secure and cost-efficient way while utilizing the same resources.

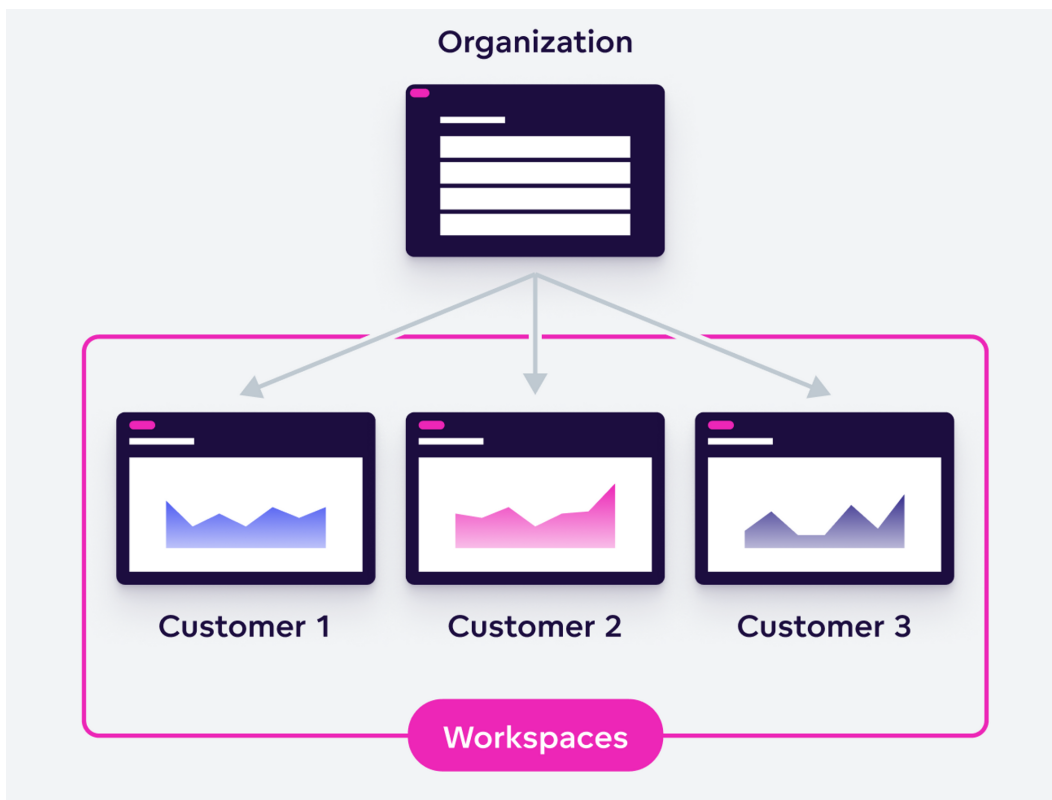


A good example of this is a B2B company that provides analytics to all of its customers using a multi-tenant analytics platform. The company can provide each customer with their own isolated instance of the analytics — with data only relevant to them. All customers share the same platform and its resources, but cannot access each other's data. The company can easily create a new analytics environment whenever it needs to onboard a new customer. Updates like new metrics, visualizations, or dashboard designs can be pushed out quickly and easily across all customer instances, streamlining the management and scaling of the analytics.

Multi-Tenant Architecture and GoodData

GoodData's analytics and business intelligence platform is designed for building multi-tenant analytics applications. The platform utilizes workspace architecture to ensure and enable the multi-tenant use case.

A workspace is a logical container for user groups to analyze and visualize data. Each user group, such as different departments, partners, or customers, has its own dedicated workspace containing all of the necessary components — a logical data model, user-access management, metrics, charts, dashboards, and more. This allows them to quickly and easily access, analyze, and deliver actionable insights from their data.



Workspace hierarchy in GoodData

These analytical workspaces are completely isolated from one another; users in one workspace cannot view the data and analytics of other workspaces unless they are granted permission. Additionally, GoodData's workspaces can support unlimited users without incurring extra costs, making it easy to scale analytics in a secure and cost-effective fashion.

Want To Try It Yourself?

If you are looking for a multi-tenant solution for your analytics, why not register for the free [GoodData trial](#) and see how easy it is to [create a multi-tenant environment](#) yourself?

Read more

If you are interested in [GoodData.CN](#), please contact us.



What Is Multi-tenant Analytics?

ely, sign
[tps://www](https://www)



What Is Multitenancy?

GoodD



Building Real-time, Multi-Tenant Analytics: 10

Subscribe to our newsletter

Get your dose of interesting facts on analytics in your inbox every month.

Subscribe