

What Is Multitenancy?

Table of Contents

[What Is Multitenancy?](#)

[Cloud computing and Multitenancy](#)

[Single-Tenant vs. Multi-Tenant](#)

[What Is Multi-Tenant Architecture?](#)

[What Does a Multi-Tenant Environment Look Like?](#)

[Types of Multitenancy](#)

[Key Features of Multitenancy](#)

[What Is Multi-Tenant Security?](#)

[How To Implement a Multi-Tenant Environment](#)

[How To Scale With Multitenancy](#)

[Multi-Tenant Solution Use Cases](#)

[The Pricing Model for a Multi-Tenant Solution](#)

[Advantages and Disadvantages of Multitenancy](#)

[Examples of Multitenancy](#)

[Multitenancy in GoodData](#)

[Ready To Get Started With Multitenancy?](#)

[Discover More About Multitenancy](#)

What Is Multitenancy?

In today's changing business environment, companies are looking for more efficient and cost-effective ways to manage their data and resources. One solution to these challenges is multitenancy, a concept that enables multiple customers to share a single software instance.

Multitenancy is a software architecture that serves multiple users by delivering a single instance of the software. It enables each user to access the data, configuration, user management, and other specific functionalities of a given instance. These users are often called tenants.

Tenants can be:

Users and user groups who are outside of your company but related to your business (e.g., resellers, agents,

franchise units, etc.).

Customers (e.g., subscribers or client companies).

Users inside your company (e.g., departments, global business units, or single users with specific needs).

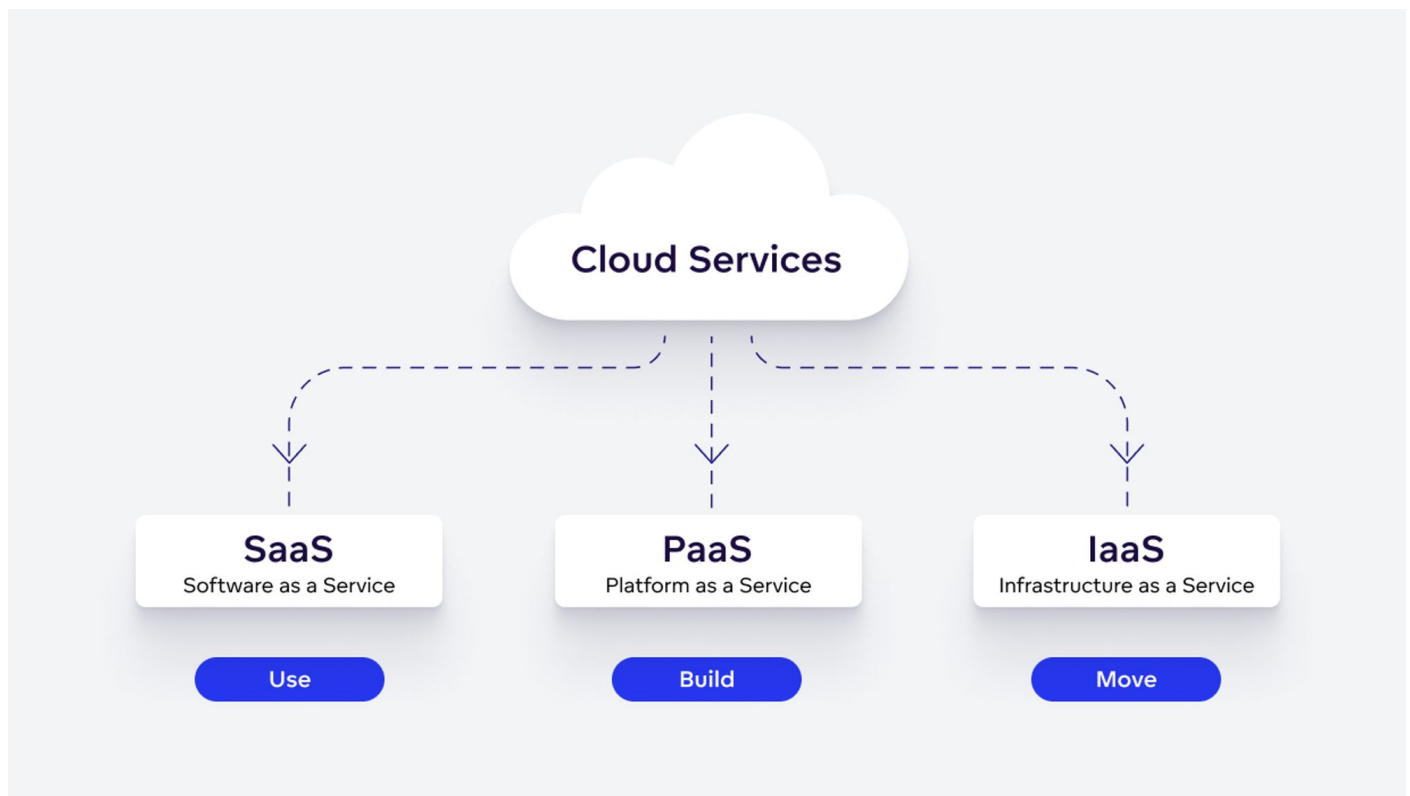
An example of multitenancy can be found in multi-tenant analytics platforms. Multi-tenant analytics performs the same tasks as traditional analytics, i.e., transforming data into meaningful insights and visualizations for business decision-making. However, it also allows multiple customers to use the same analytics tools/platforms while ensuring each user's data and configuration are kept separate and secure from other users.

Before we explore how multitenancy works and how you can implement it within your environment, let's first look at why multitenancy is so often described as a key feature.

Cloud computing and Multitenancy

Cloud computing offers companies many benefits, including scalability of resources and cost-effectiveness. With [cloud computing](#), you can deliver infrastructure to network architects, platforms to software developers, and software to end users.

Multitenancy is a key feature of cloud computing. In cloud computing, multitenancy allows multiple customers of a cloud provider to use the same computing resources while keeping their data isolated and inaccessible to other customers.



Cloud computing services

Today, most applications are delivered to customers and end users via the cloud as software as a service (SaaS). This allows many end users to access this software no matter their location or type of device. SaaS is

supported by multitenancy — it guarantees a user access to a single application by allowing them to manage their interface without affecting other users.

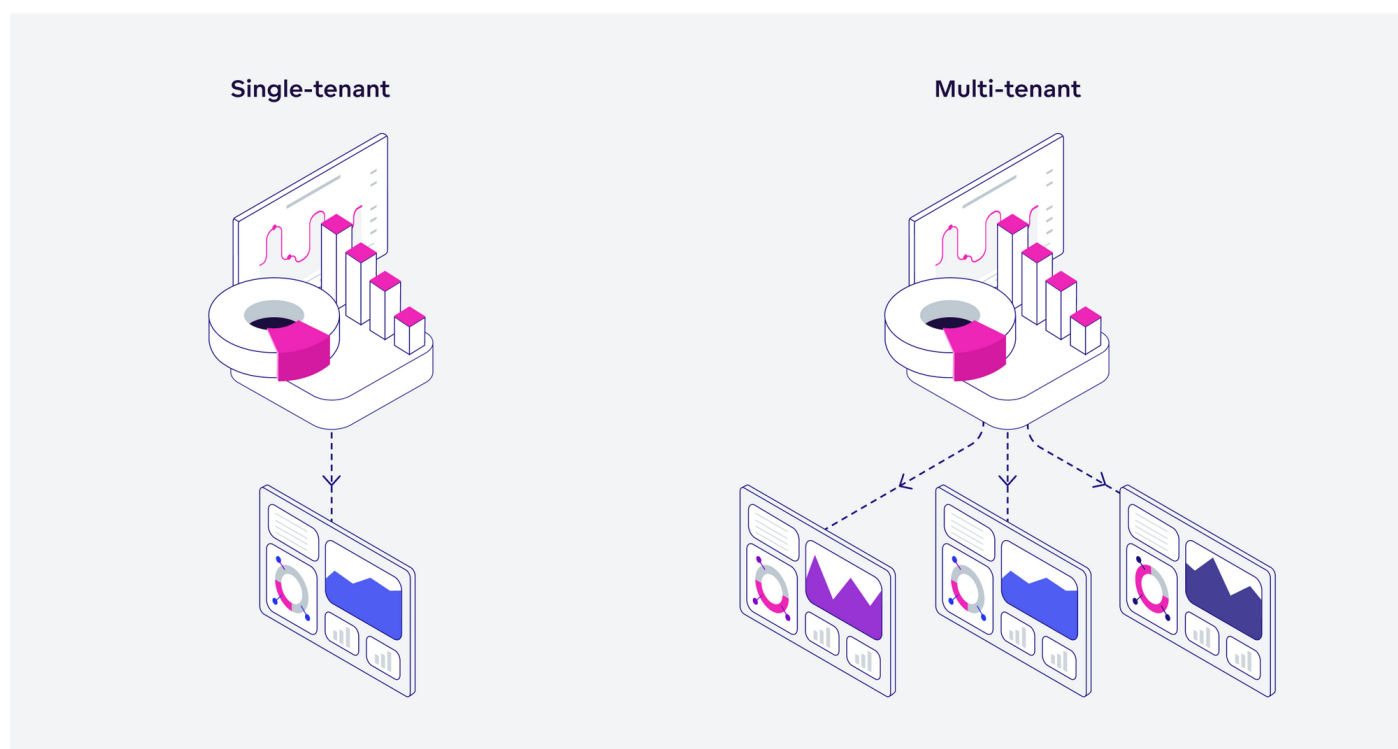
Now we understand why multitenancy is considered a key feature in cloud computing, let's further explore the mechanics of multitenancy by discussing how it differs from single tenancy.

Single-Tenant vs. Multi-Tenant

Multitenancy is a new concept for distributing analytics to customers that replaces traditional single-tenant analytics.

With single tenancy, an analytics platform is distributed to each tenant individually: Separate analytics instances are installed on each tenant's device, and individual, customized analytics are set up. This includes creating user accounts, connecting data sources, and creating metrics and dashboards which cannot be accessed by other tenants. The result is an isolated environment that ensures high data security and privacy. Each tenant is responsible for managing their own analytics on their dedicated device as they desire.

With multitenancy, each tenant shares the same analytics platform, but they have defined, specific user roles and access rights in order to access different working environments through a single deployment. One working environment can be accessed by an individual tenant or group of tenants to manage analytics, create metrics and dashboards, or simply view pre-created dashboards to gain relevant information to inform their business decisions.



Single-tenant architecture vs. multi-tenant architecture for analytics

When it comes to analytics, the main difference between single tenancy and multitenancy is:

Single tenancy provides an isolated environment deployed separately for each tenant.

Multitenancy uses a shared platform to serve multiple tenants; these tenants receive defined user roles and access rights to their dedicated environments to manage their analytics interface.

What Is Multi-Tenant Architecture?

Any discussion of multitenancy would be incomplete without touching upon multi-tenant architecture. While multitenancy is a concept for implementing a multi-tenant solution, [multi-tenant architecture](#) refers to a specific design of software architecture that can serve multiple users within one software instance.

Multi-tenant architecture helps with the creation of an efficient multi-tenant environment. In the case of [multi-tenant analytics](#), one platform utilizes a shared environment that enables centralized data analysis and reporting. This allows multiple users and user groups to access a shared data model, metrics, visualizations, and dashboards while maintaining their data privacy.

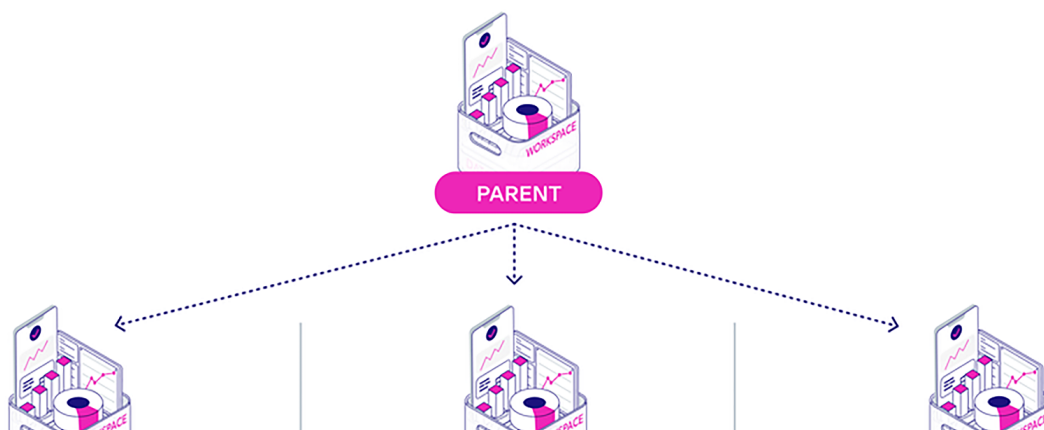
What Does a Multi-Tenant Environment Look Like?

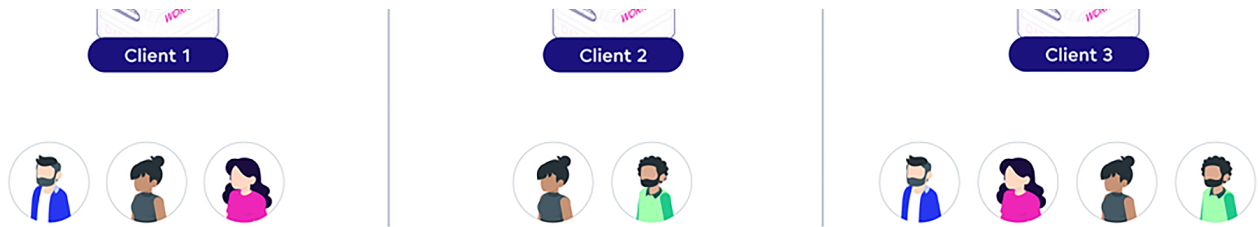
The main element of a multi-tenant environment is a [workspace](#). This is a dedicated space where tenants can manage, analyze, and visualize their data. It contains data sources, logical data models, metrics, visualizations, and dashboards. To achieve a multi-tenant environment, workspaces need to be organized in a so-called workspace hierarchy. A workspace hierarchy contains two types of workspace:

The parent workspace serves as a template for the child workspaces. This workspace contains a data model, all of the metrics, and dashboards that can be shared with the connected child workspaces. No end users are given access to this master workspace.

The child workspace is tenant-facing. It is the space where end users, customers, or clients work with their dashboards. The child workspace contains the data model, core metrics, and dashboards inherited from the master workspace, but each child workspace contains different data. End users in the child workspace may create their own metrics (on top of those provided by the master workspace) and their own dashboards, all without affecting the master workspace.

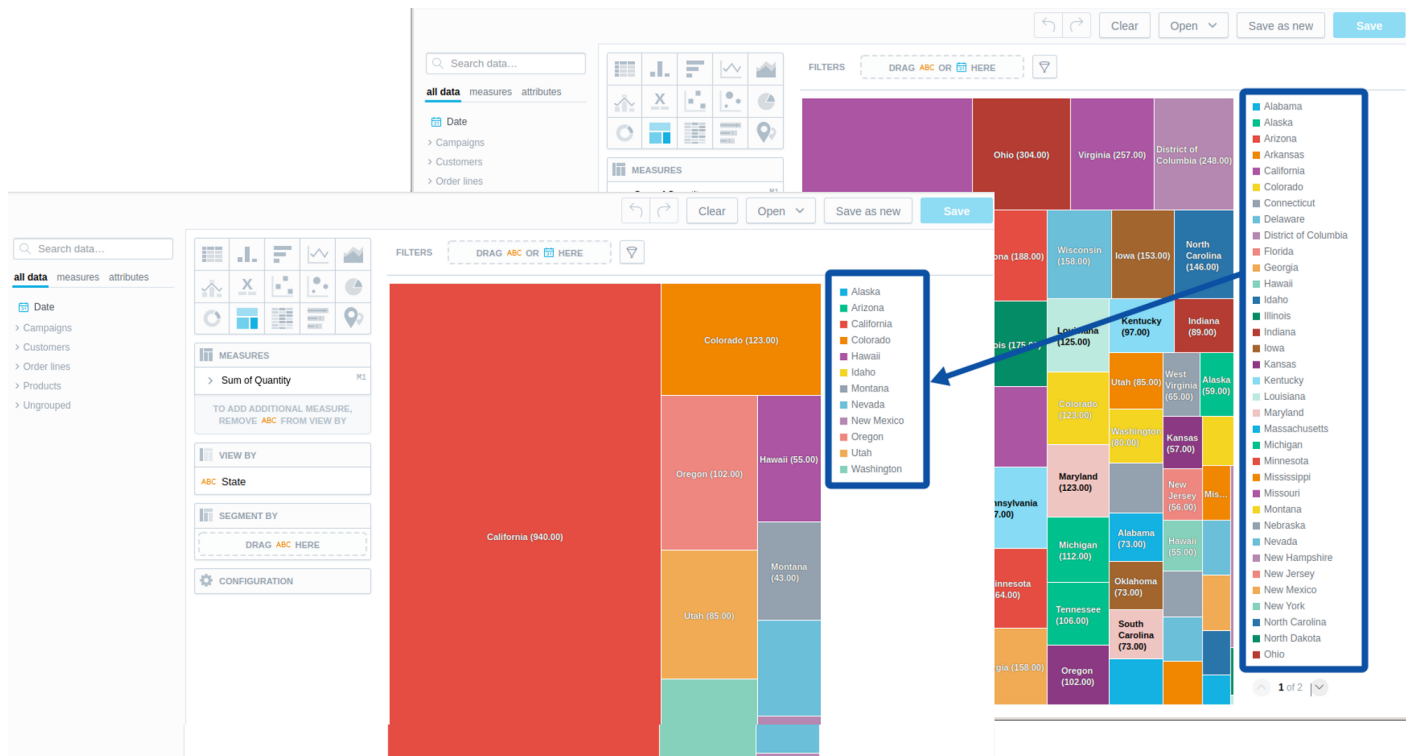
This multi-tenant architecture allows you to push updates from the master workspace to any of its child workspaces. Additionally, you can easily reuse workspaces, create copies of workspaces, and modify them to accommodate the needs of different teams, business partners, customers, or clients.





Workspace hierarchy

Besides metrics and dashboards, another crucial aspect of workspaces is data. Displaying what data belongs to which child workspace is achieved via data filters. Data filters allow you to limit the data available in child workspaces. By setting a data filter, the data visibility can be defined for each child workspace.



Setting data filters for a child workspace

Data in each workspace is completely isolated from workspaces that belong to other tenants, ensuring data security and privacy for each tenant.

Types of Multitenancy

Because there are different use cases associated with user requirements, multitenancy can take different forms. There are several possibilities available to multi-tenant solution providers in terms of how they can implement a multi-tenant environment in their product.

Within the multi-tenant environment of an analytics platform, tenants can access data sources and workspaces to analyze data. Multi-tenant architecture can be implemented in the following ways:

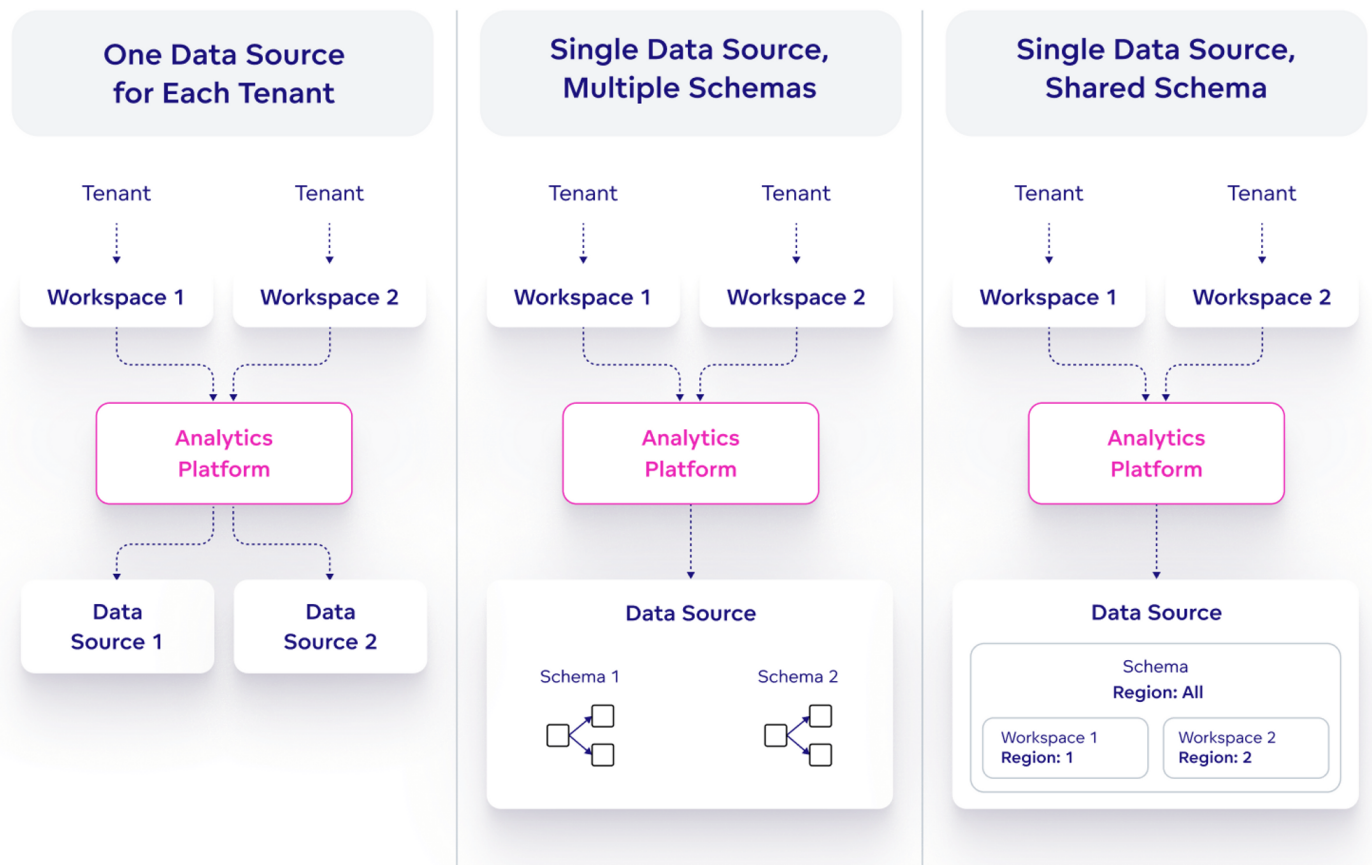
One data source for each tenant: Each tenant has a dedicated workspace where they can connect to their separate data source and manage their analytics interface as required.

Single data source, multiple schemas: Tenants access the same data source, but each tenant has their own

workspace where they have a separate schema — which is a relevant part of the overall data model — and can work with it.

Single data source, shared schema: All tenants access the same data source and use the same logical data model. However, in their workspace, they only manage relevant data that is filtered from the entire data source — for example, based on region, branch, or department.

The diagram below illustrates the three different methods of multi-tenant architecture implementation. Note that a tenant can be an individual user or a user group.



Types of multitenancy

Key Features of Multitenancy

Each multi-tenant solution that is based on the concept of multitenancy should have the following features:

Shared infrastructure: Serves multiple customers with a single instance of an analytics platform, while maintaining tenant isolation. This includes sharing the same data sources connected to the analytics platform, as well as the same metrics, visualization elements, and dashboards.

Tenant isolation: Each tenant is provided with 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 in order to manage the analytics environment.

Customization: Tenants can customize the analytics environment to meet their specific needs and preferences by setting up unique configurations, options, and branding. This is an important feature as it allows tenants to effectively use the shared multi-tenant analytics platform for their specific use cases, such as filtering data by different branches or locations and creating specified metrics and dashboards.

Scalability: Ensures that the analytics platform can accommodate growth without becoming overwhelmed or experiencing degradation in performance. Multi-tenant architecture for analytics should be able to handle increasing numbers of tenants and automatically allocate resources as needed.

Centralized management: Helps to manage multiple tenants and analytics setups from a single point of control via a single analytics interface. Changes made to data sources, data, metrics, and dashboards in the master workspace are automatically reflected in all child workspaces. This process simplifies maintenance and reduces the need for manual setups for each tenant, making it easier for the provider to manage the entire multi-tenant environment and remotely assign access and roles to tenants.

What Is Multi-Tenant Security?

As you can imagine, with multiple tenants accessing various workspaces, security becomes paramount. Multi-tenant security refers to the measures put in place to ensure the security and privacy of each tenant's data and resources within one workspace in a multi-tenant environment.

Multi-tenant security includes:

Managing users and user groups: This involves ensuring that each user or user group only has access to the data and resources relevant to them. Assigning dedicated credentials to each user or user group ensures that only authenticated and authorized users can access their designated workspace.

Managing permissions: This involves setting permissions for different users and user groups. Permissions can be set for objects such as data sources and workspaces using specific permission types, such as "manage," "view," "analyze," or "use."

How To Implement a Multi-Tenant Environment

Setting up a [multi-tenant environment](#) can feel like a complex process, consisting of implementing a multi-tenant architecture, creating a workspace hierarchy, user and user group management, and more. To simplify the steps required, we have created the guide below:

Choose an analytics platform with built-in multi-tenant capability: Ensure the platform supports multiple tenants, data isolation, user group management, and other user requirements.

Create a master workspace: Connect data sources to the master workspace, generate logical data models, and create the main metrics, visualizations, and dashboards.

Build a workspace hierarchy: Create child workspaces. This also defines the relationships between individual users and user groups.

Create data filters: From the master workspace, set up data filters to restrict access to specific data within child workspaces.

Manage users and user groups: Assign users to the appropriate user group, enabling them to access specific workspaces.

Manage permissions to these users and user groups: Define the level of access each user and user group has to specific data sources and workspaces. Set up permissions for different objects such as data sources and workspaces.

How To Scale With Multitenancy

Once a multi-tenant environment has been implemented it can always be modified in the future. As data amount increases, the number of users grows, and new use cases are revealed, multitenancy allows you to easily scale your analytics platform to accommodate these changes.

Scaling opportunities within a multi-tenant architecture include:

Scaling data storage: Increasing the storage capacity, either by upgrading hardware components in an on-premises system through vertical scaling, or by adding more nodes to a cloud storage system through horizontal scaling. It is important to ensure that the application is capable of handling the increased data volume.

Scaling user base: Scaling the application interface by adding or removing separate users or user groups, regardless of their geographic location.

Scaling among use cases: Adapting the multi-tenant application to handle a variety of situations and use cases, such as embedding its parts into custom applications or web applications, integrating it with different communication tools, or accessing it through APIs or developer interfaces.

A multi-tenant solution is a great option for companies that operate in a dynamic environment, as the architecture is highly adaptable to changes in the number and needs of users, offering both flexibility and scalability. This flexibility also extends to the types of companies that can take advantage of multitenancy, as you'll see below.

Multi-Tenant Solution Use Cases

To serve the needs of their customers, providers offer multi-tenant environments built on a multitenancy concept within a single analytics platform.

End users of a multi-tenant analytics solution can be:

Companies with multiple branches and stakeholders from different locations, allowing for centralized data management and access to shared resources.

Small to medium-sized companies from the same location.

Data-driven companies that require centralized data storage connected to their analytics platform, distributed to individual users and groups.

Companies that operate in the B2B market, as it enables them to offer a customized version of their pre-purchase solution from other vendors for each customer.

A multi-tenant analytics solution can be used in any type of industry, but it is especially appropriate for industries such as e-commerce, financial services, healthcare, insurance, software companies, manufacturing, and logistics.

The Pricing Model for a Multi-Tenant Solution

In analytics with single-architecture deployment, companies pay fees based on the number of end users. However, as the business environment is constantly changing, companies often find it difficult to predict their future user base. For this reason, the traditional 'per-user' pricing model may not be effective.

In contrast, a multi-tenant analytics solution uses the per-workspace pricing model. In addition to the regular fees for renting the analytics platform, the per-workspace model charges a fixed fee for each workspace used, regardless of the number of users or activity within that workspace. This model is a good fit for:

Multi-tenant solutions designed to have separate workspaces for each team/department.

Companies in the B2B market that need to deliver analytics to customers but can't predict their user base.

Advantages and Disadvantages of Multitenancy

Multitenancy in analytics can be the right solution for companies to meet their demands. When properly planned, designed, and managed, a multi-tenant solution can bring numerous advantages to organizations and help them to achieve their goals. These advantages include:

Lower costs compared to single-tenant analytics platforms (as there is no need to pay for separate versions for each user).

Streamlined onboarding and offboarding of tenants to and from their respective workspaces within a shared analytics deployment, with tenant isolation and analytics performance preserved.

Improved security through isolated tenant data and controlled access, managed through permission management and user access.

Simplified maintenance, with a single codebase and a parent-child relationship between workspaces.

Streamlined deployment through reduced manual setup, making it easier and faster to add new users by creating a separate workspace or adding a new user to an existing user group.

Note, however, that multitenancy does have its disadvantages:

Elevated system complexity with the need for additional logic for tenant separation, data filtering, and tenant identification.

Increased attention to careful design and management required to ensure tenant isolation and secure data storage.

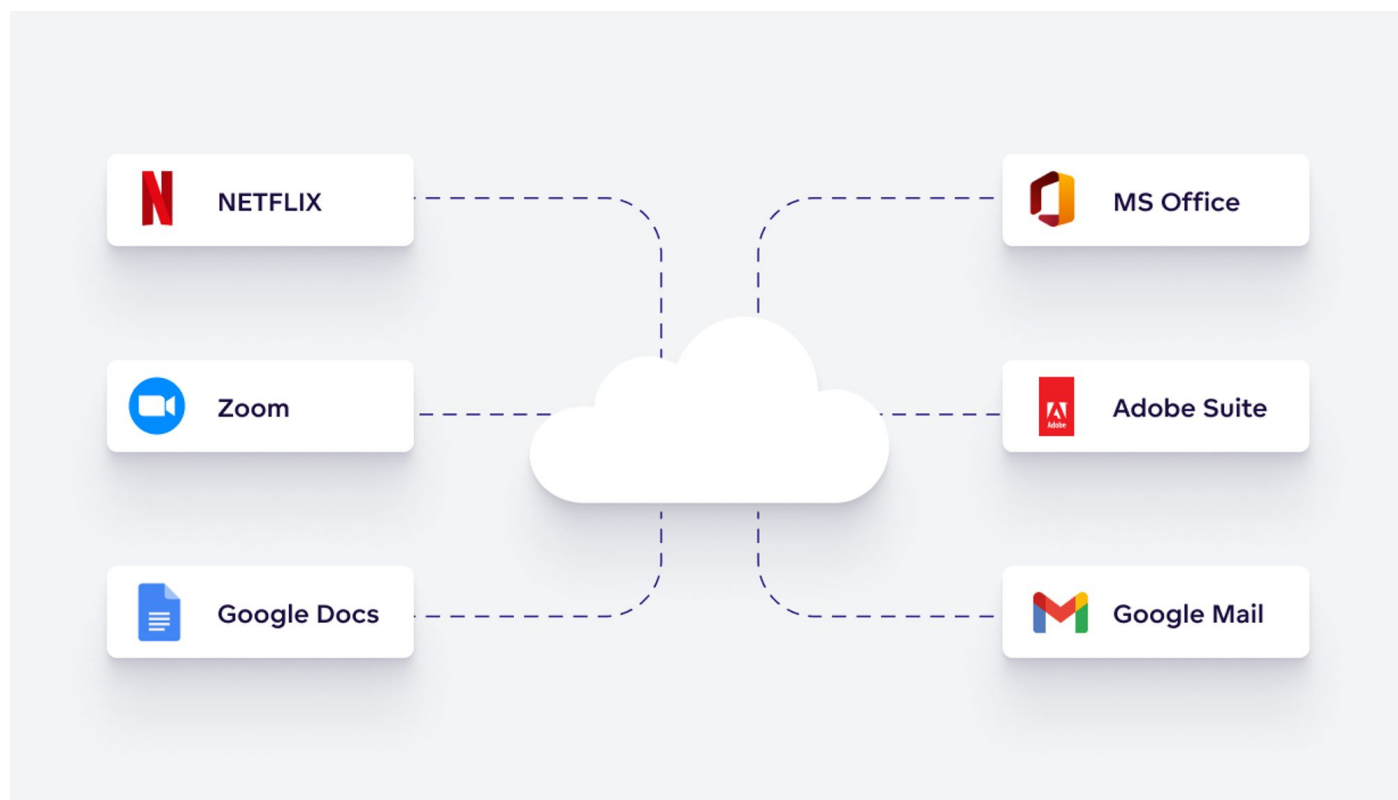
Limited control over the analytics environment due to the shared platform, and potential provider-imposed limitations or restrictions.

Increased dependence on the provider for maintenance and upgrades, requiring tenants to choose a reputable and reliable provider.

Companies must weigh the potential benefits and drawbacks of implementing a multi-tenant solution, ensuring it meets their specific needs. To succeed, it is important to understand the workings of multi-tenant analytics in a corporate environment, as well as the requirements for proper setup and implementation.

Examples of Multitenancy

For a better understanding of multitenancy, it can be helpful to consider some well-known examples that are found in cloud environments and used as Software-as-a-Service (SaaS) applications. These include Gmail for reading and sending emails, Zoom for online meetings, Google Docs for document storage and sharing, Adobe Creative Cloud for graphic design, and Netflix for streaming movies and TV shows.



Examples of multitenancy

An excellent example of multitenancy in analytics is GoodData's analytics platform, which offers a user-friendly multi-tenant environment for managing analytics.

Multitenancy in GoodData

The GoodData analytics platform is designed to meet the needs of multi-tenant analytics applications. This

approach is supported by its ability to connect to various database providers. GoodData promotes the use of APIs to set up your analytics, create metrics and dashboards, and implement a multi-tenant architecture for your purposes, rather than relying on a user interface.

GoodData's multitenancy can benefit customers by:

Providing an analytics interface to separate users and user groups within or outside the company.

Delivering analytics to users quickly and easily; every change in the parent workspace is automatically reflected in the child workspaces

Providing [self-service analytics](#) for users without technical skills, allowing them to make queries and reports without the support of data teams

[Embedding analytics](#) into customers' applications and platforms, providing users with access to data and insights within their preferred tools

Setting up a multi-tenant architecture through a robust [API](#) that allows for automation and streamlined management of the multi-tenant environment

To achieve an effective multi-tenant analytics solution, customers can connect GoodData's platform to various data sources, including the well-known databases [Snowflake](#) and [PostgreSQL](#).

Ultimately, GoodData is a powerful multi-tenant analytics solution, offering scalability, security, data warehousing capabilities, and integration options.

Ready To Get Started With Multitenancy?

Interested in gaining first-hand experience with a multi-tenant environment? To see how GoodData deals with multi-tenant analytics, [start a free trial](#) or [request a demo](#).

Discover More About Multitenancy

Want to learn more about multitenancy? We have several other free resources available. Some of our favorites include:

[Multitenancy in GoodData](#)

[Webinar: How To Build Real-time, Multi-tenant Analytics](#)

[Building Real-time, Multi-tenant Analytics: 10 Questions and Answers](#)