

The GitHub Guide to Organizations

This guide will explore the relationship between Organizations, Teams, and Repositories, so that you're able to make the best architectural decision for your company. We'll also introduce you to teams and resources within GitHub that can provide in-depth explanations of Organizations.

Why does Organization structure matter?

Your Organization structure can facilitate collaboration and discoverability — or it can make communication more difficult for developers. When setting up your GitHub Enterprise instance, the immediate instinct may be to create an Organization for every project or department at your company, leading to many divided groups that function in GitHub as siloes. This may seem like a good way to manage permissions and reduce noise, but it's not always the ideal strategy. In fact, it is detrimental to cross-team collaboration and can result in administrative headaches down the line. We want you to get started with an architecture that will help your team work together seamlessly, creatively, and transparently without bogging you down in unnecessary overhead.

Instead of creating many Organizations and siloing users, we suggest using one or few Organizations for shared ownership of repositories and making use of Teams to segment users within those Organizations.

ORGANIZATION

Organizations are a group of two or more users that typically mirror real-world organizations. They are administered by Organization members and can contain both repositories and teams.

REPOSITORY

A repository is the most basic element of GitHub. Think of it as the main folder for each of your projects. It contains all of the project files (including documentation and issues), and stores each file's revision history. Repositories can have multiple collaborators and can be either public or private.

TEAM

Teams give you the ability to create groups of organization members with read, write, or admin permissions to repositories that belong to the organization. Teams are also central to many of GitHub's collaborative features, such as team @mentions, which notify appropriate groups of people that you'd like their input or attention. Teams can be both project or subject-matter focused, and can relate to job titles or interest groups within your company as well. Note: It is not possible to @mention teams in a different organization.

Pro tips for configuring Organizations on GitHub Enterprise

HAVE AS FEW ORGANIZATIONS AS POSSIBLE TO AVOID CHALLENGING SITUATIONS IN THE FUTURE.

This makes discoverability easier and decreases administrative headaches. Instead of managing permissions across many Organizations, having one or few Organizations can help you build a cohesive and nimble permissions strategy.

TEAMS ARE THE BEST WAY TO GROUP USERS AND PROVIDE ACCESS TO REPOSITORIES.

Teams allow you to create groups with separate levels of access and visibility within an Organization. Relying on Teams as opposed to Organizations can help increase collaboration without compromising access controls.

ORGANIZATION OWNERS ARE THE ADMINISTRATORS AT THE ORGANIZATION LEVEL.

Administrators who familiarize themselves with the Organization settings page have a veritable arsenal of tools at their disposal. In addition to features like configuring Organization-wide integrations, organization administrators can also set default member privileges, which is a powerful tool for making repositories visible to Organization members without opening them up to your entire GitHub instance.

If you'd like a technical deep dive on the Organization settings page, just let your Account Manager know, and we'll get one scheduled.

AS A SITE ADMINISTRATOR YOU CAN ENABLE ALL USERS — OR ONLY FELLOW SITE ADMINISTRATORS — TO CREATE ORGANIZATIONS.

While it's always wonderful to see users embracing a product, as a site administrator, you may want to leave Organization creation to the team or people responsible for maintaining a clear and useful architecture — otherwise, you run the risk of superfluous Organizations.

IF NECESSARY, SCHEDULE AN ARCHITECTURE MEETING TO DISCUSS CURRENT AND FUTURE NEEDS WITH YOUR ACCOUNT TEAM.

At GitHub, we use GitHub everyday. It allows us to understand different use cases and practice the best strategies for using GitHub effectively. Our teams can help make sure that you receive the full benefit of that knowledge by walking you through the advantages of different architecture options.

Which Organization architecture is best for your company?

Based on our own use of GitHub, as well as conversations with users, we've created a general overview of Organizational archetypes to help you get started. Use this chart to review the common architecture types.

| Archetype | Company Size | Benefits |
|---|---|---|
| A single Organization with direct organization membership for repository access | Small, potentially medium | <ul style="list-style-type: none">• Ideal for startup teams• Highly collaborative• Teams and team members can be @mentioned across all projects |
| A single Organization with multiple teams to manage repository access | Medium or small companies with strict security needs | <ul style="list-style-type: none">• Works well in smaller companies with security restrictions• More granular repository access |
| Multiple Organizations with multiple teams to manage repository access | Large companies that require restricted access to specific repositories | <ul style="list-style-type: none">• Higher level of separation• Granular access control for repositories• Best for companies with >500 GitHub Enterprise users |

A SINGLE ORGANIZATION WITH DIRECT MEMBERSHIP FOR REPOSITORY ACCESS

In this first example, we see a single Organization on your GitHub instance to which all users belong. All users have write access to Organizational repositories, which is granted by default Organization permission with membership. Teams are used to group users by role or area of expertise, not for granting them access to repositories.

This approach works best in small companies where everyone contributes to multiple projects — startups are a great example of this work model. The ideal candidate for this structure also has a high-trust environment and a highly collaborative work model, as one of the biggest benefits of this architecture is that all members and teams can be @mentioned to get their input on all projects. As this is a loose organizational structure, it doesn't work well for companies with siloed business divisions.

A SINGLE ORGANIZATION WITH MULTIPLE TEAMS TO MANAGE REPOSITORY ACCESS

Though this example is similar to the direct membership scenario, the main difference is that Teams are used to control repository access with more granularity. This is ideal for most mid-size companies — or small companies that are subject to regulations requiring stricter attention to access controls. However, all teams are still able to @

mention each other, as they are in the same organization.

This can potentially create a substantial work load for administrators in larger companies with many teams or smaller companies where users may create teams unnecessarily. This method should be avoided by large companies where no single owner should have access to all repositories managed in your own, secure environment.

MULTIPLE ORGANIZATIONS WITH MULTIPLE TEAMS TO MANAGE REPOSITORY ACCESS

This method advocates the creation of separate Organizations that map to business units, products, or any other silo, as well as many repositories that can largely remain separate. That being said, if cross-team collaboration is important, teams in different Organizations will not be able to @mention each other, and employees may have less of a chance of discovering projects within the company — something to keep in mind culturally as you build out your GitHub instance.

If you'd like more assistance determining which structure works best for your company, ask your Account Manager to schedule a meeting with the GitHub Services team.