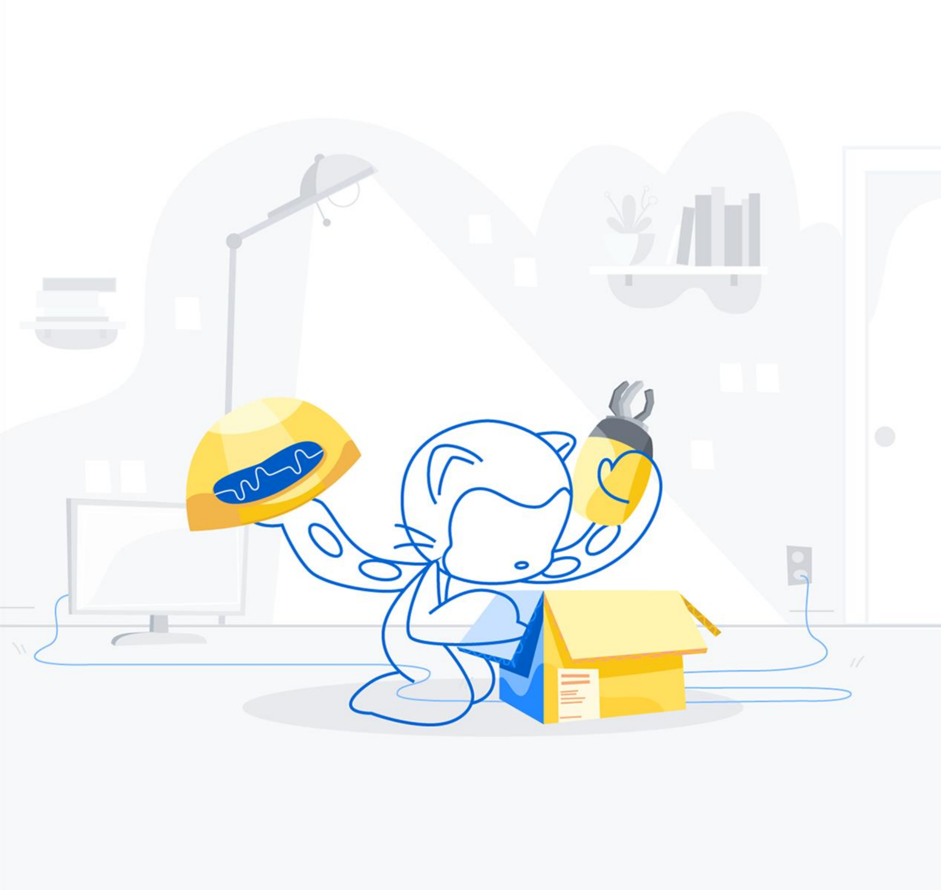
GitHub

GitHub Administration

Andrew Scoppa

Agenda

- GitHub Enterprise overview
  - Platforms
  - Permission flow
  - Enterprise administration
- Organization
  - Overview
  - Administration and settings
- Repository
  - Repository overview
  - Repository administration
  - Branch protections
  - CODEOWNERS



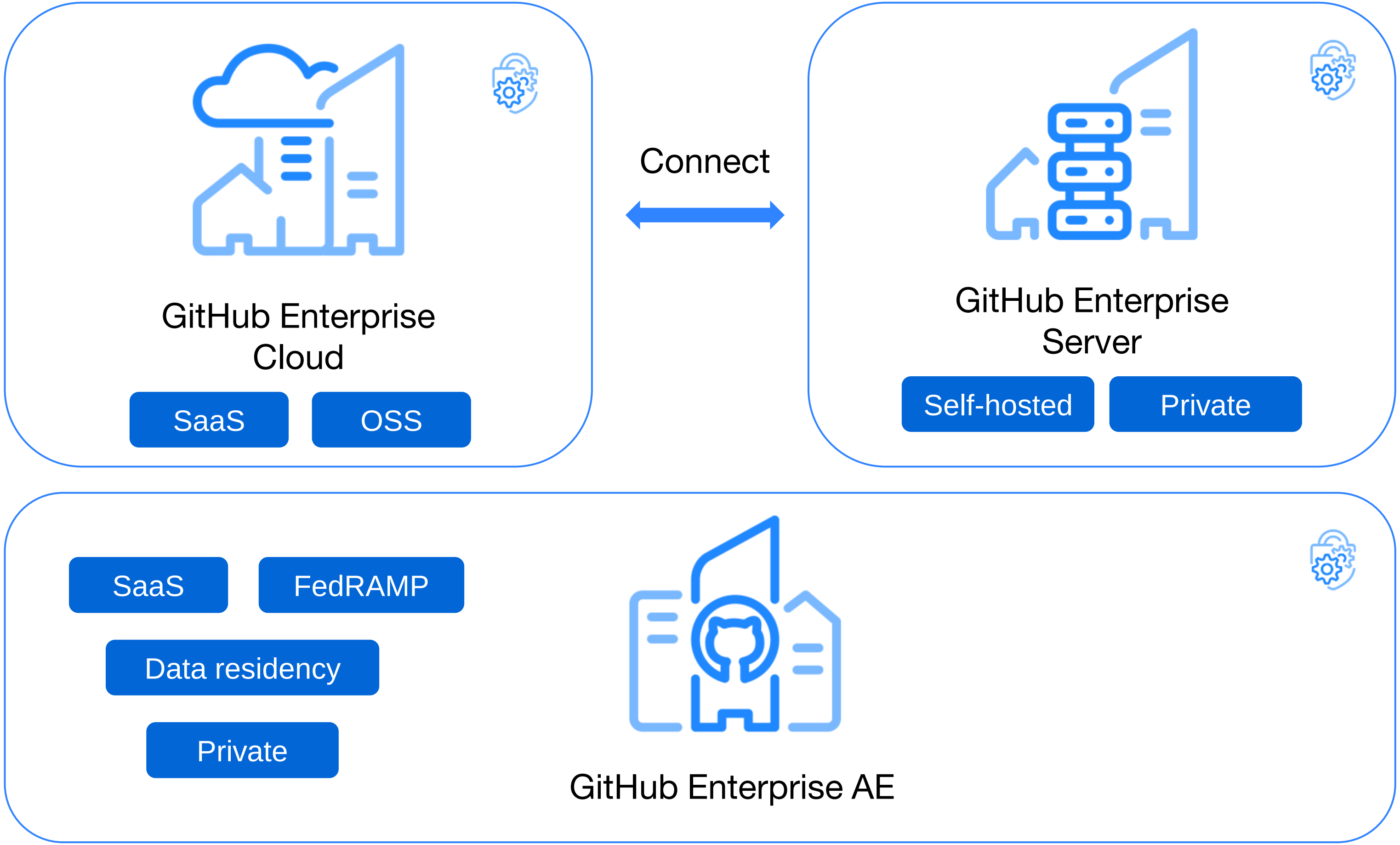
Objectives

- Differences between the **different GitHub Enterprise family** of products (Cloud, Server, and AE)
- Can decide which **permissions** should be applied
- Know all the **policies** at the enterprise level
- Can explain how **security access features** works in GitHub (SSO, SCIM, IP allow-lists...)
- Learn the **authentication methods** for integrations
- Understand how **licensing** works
- Understand what an **organization** is and decide the right **settings** for it
- Manage **team structures** of different sizes and handle permissions to the **protected resources**
- Have an idea on how to organize the **repositories in the organization** and it's **settings**
- Have knowledge about **auditing** your enterprise account
- Find and assess **incorrect usages** in the organization to notify your users

GitHub Enterprise Overview

Platforms ○ Permission flow ○ Enterprise administration

Platforms



GitHub Enterprise Cloud

SaaS

OSS

Connect

GitHub Enterprise Server

Self-hosted

Private

SaaS

FedRAMP

Data residency

Private

GitHub Enterprise AE

GitHub Enterprise Cloud

- GitHub Enterprise Cloud
- Software as a Service offering
- **Security** and policy features synonymous to GitHub Enterprise
- Fast onboarding of new **collaborators**
- Reduced operations overhead
- **Public repositories** on GitHub.com are viewable by anyone on the internet
- Repositories that are **private** on GitHub.com are not accessible to everyone
- Privacy is configured by **enterprise**, organization, team, or individual level

GitHub Enterprise Server

- GitHub Enterprise Server
- **Self-hosted** on customers network
- Can be hosted on many platforms
- Additional **infrastructure configuration** may be required
- Support subdomain isolation
- Outbound web proxy for added layer of security
- **Isolated user accounts**
- Has many of the same features found in Cloud
- Can be connected to an Enterprise Cloud account for vulnerability updates and unified search

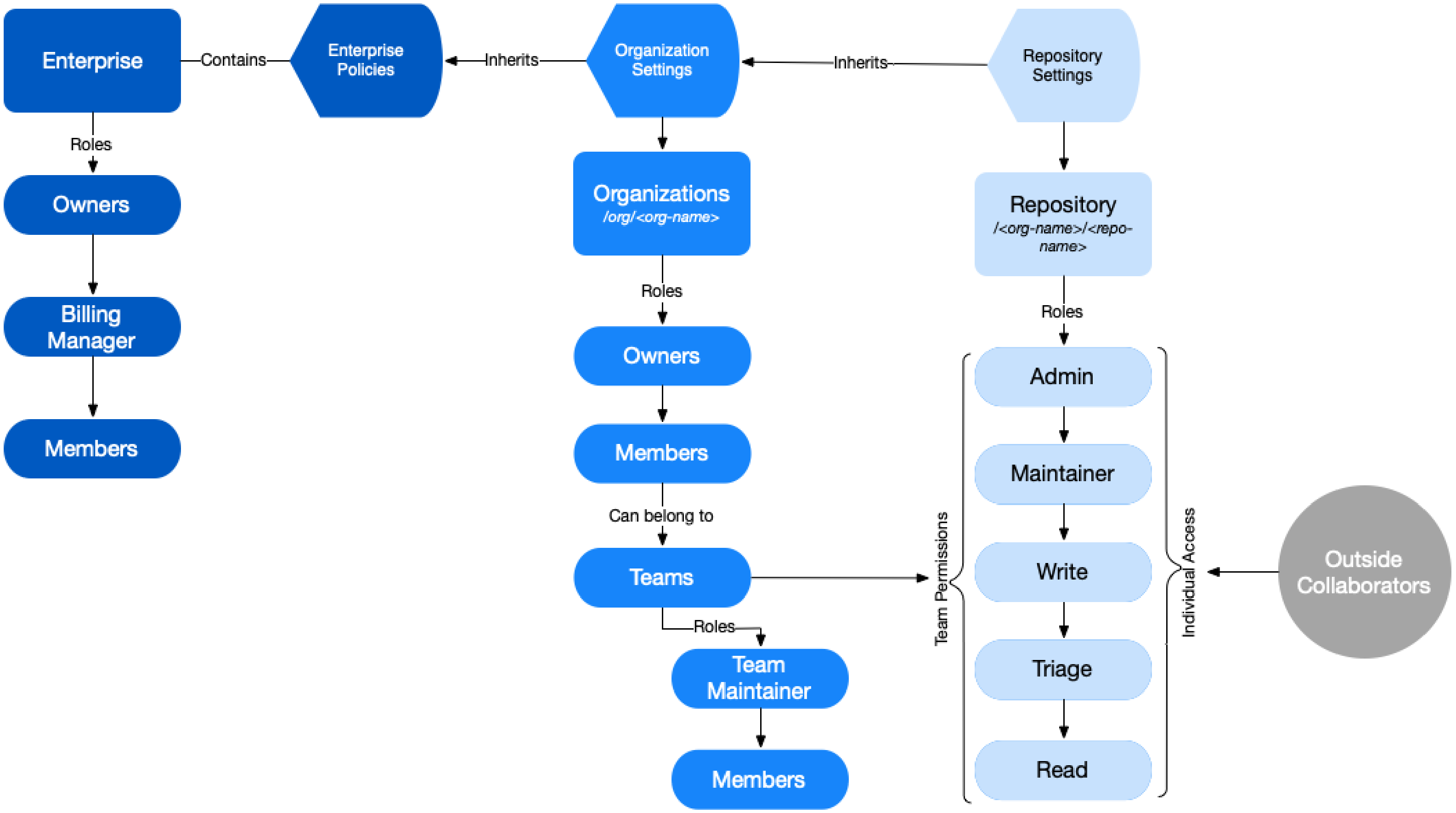
GitHub Enterprise AE

- Software as a Service offering
- Data encryption and **data residency**
- Subdomain isolation
- **Isolated** user accounts
- **Hosted in Azure** Government Cloud or Commercial Cloud
- Meets stringent **security**/compliance requirements including FedRAMP

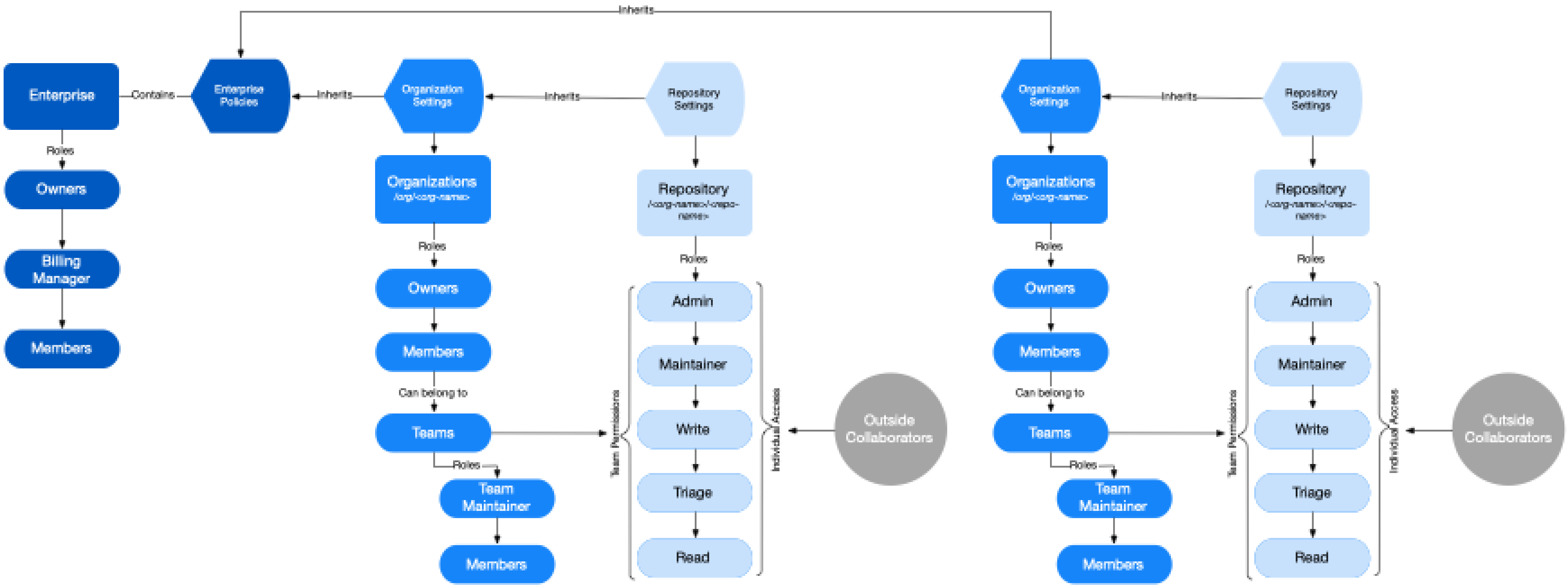
GitHub Enterprise Overview

Platforms ○ Permission flow ○ Enterprise administration

Flow of permissions

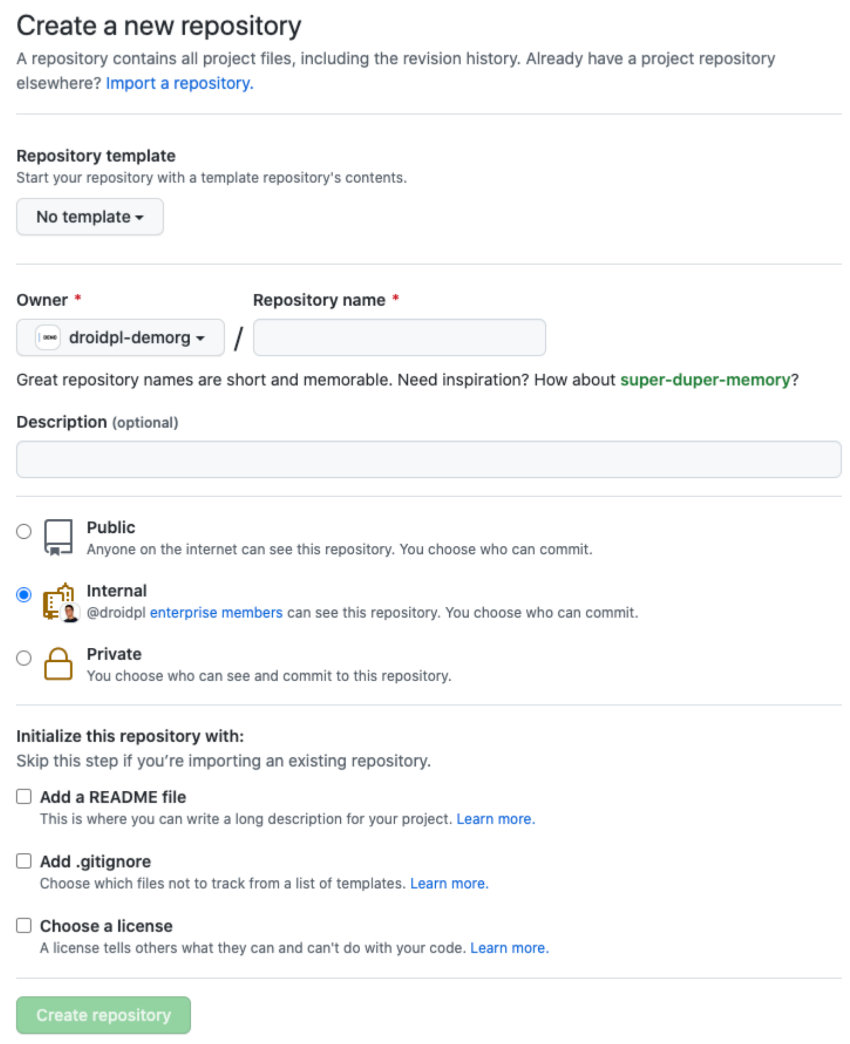


Flow of permissions - multiple orgs



Repository visibility

- **Public** - Anyone on the internet can access (GHEC)
- **Internal** - Organization members in the enterprise can access
- **Private** - Only people with explicit access



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Repository template

Start your repository with a template repository's contents.

No template ▾

Owner * droidpl-demorg ▾ / **Repository name** *

Great repository names are short and memorable. Need inspiration? How about super-duper-memory?

Description (optional)

○ **Public**
Anyone on the internet can see this repository. You choose who can commit.

● **Internal**
@droidpl enterprise members can see this repository. You choose who can commit.

○ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

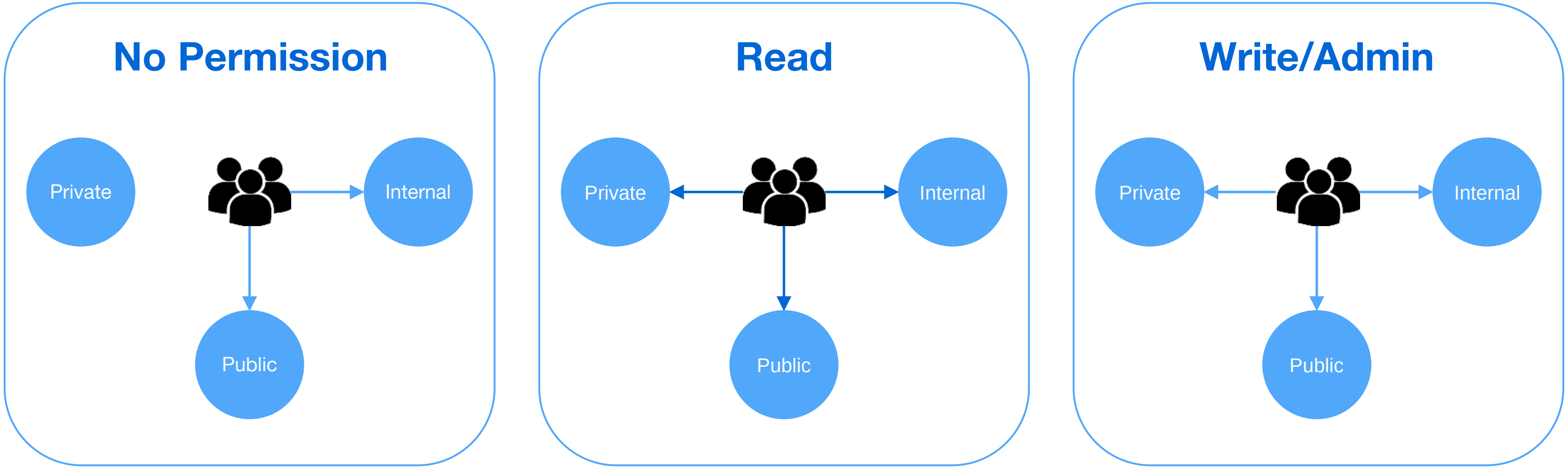
Skip this step if you're importing an existing repository.

□ **Add a README file**
This is where you can write a long description for your project. Learn more.

□ **Add .gitignore**
Choose which files not to track from a list of templates. Learn more.

□ **Choose a license**
A license tells others what they can and can't do with your code. Learn more.

Repository base permissions



***Users being added to an organization with **NO** other special access other than being added as a MEMBER to the organization.*

Permission Levels

Repository Role	Description
`Read`	Read-only access to Code and Actions. Can submit and comment on issues, pull requests, and discussions
`Triage`	Read-only permissions with the additional ability to manage issues, pull requests, discussions, assignments, and labels
`Write`	Gives write access to all parts of a repository project with the exception of the repository settings
`Maintain`	Ability to modify some settings of a repository including topics, enabling repository features, configuring merges and GitHub pages, pushing to protected branches
`Admin`	Has full administrative access to all features, settings and configurations of the repository project

GitHub Enterprise Overview

Platforms ○ Permission flow ○ Enterprise administration

Share screen

Organization

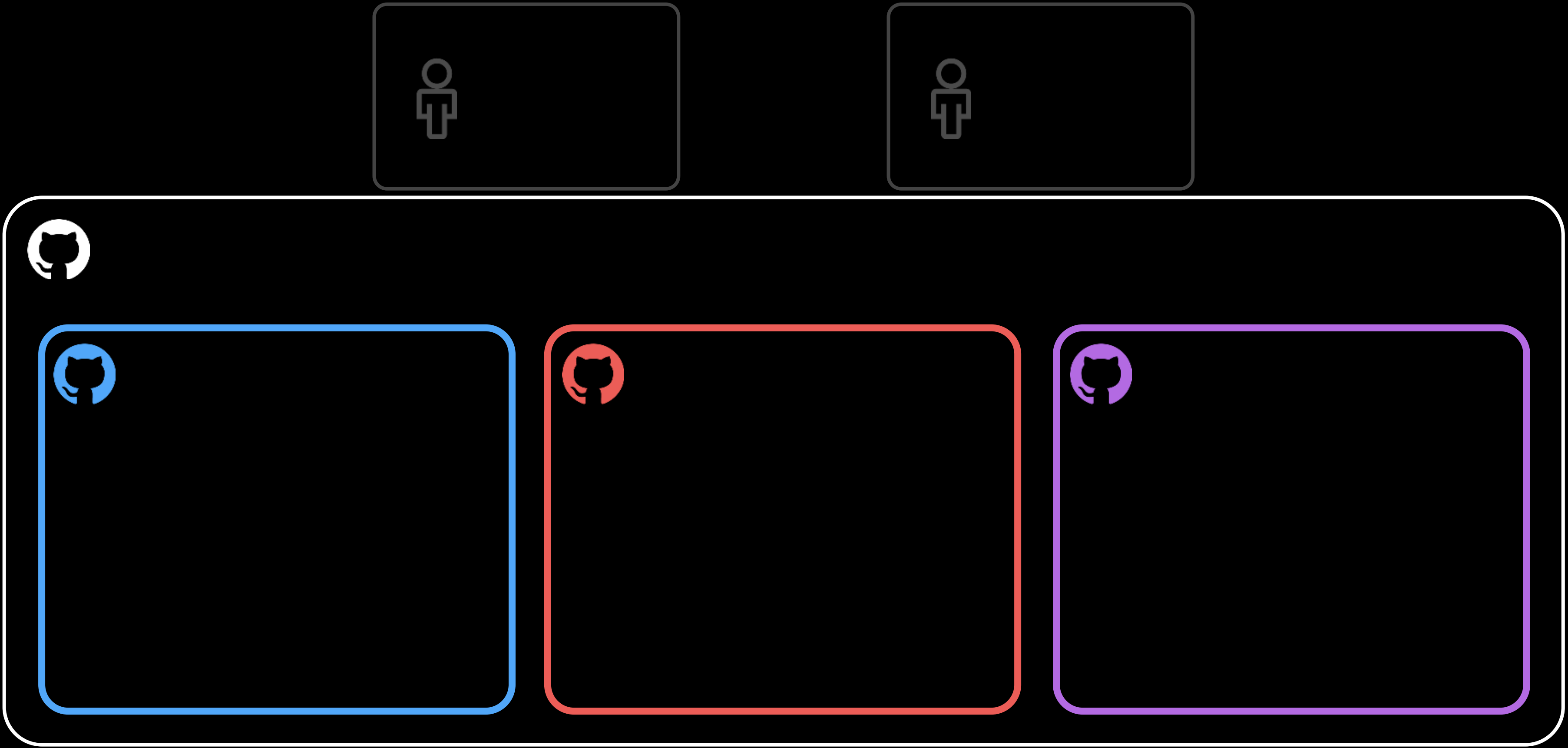
Overview

Administration and settings

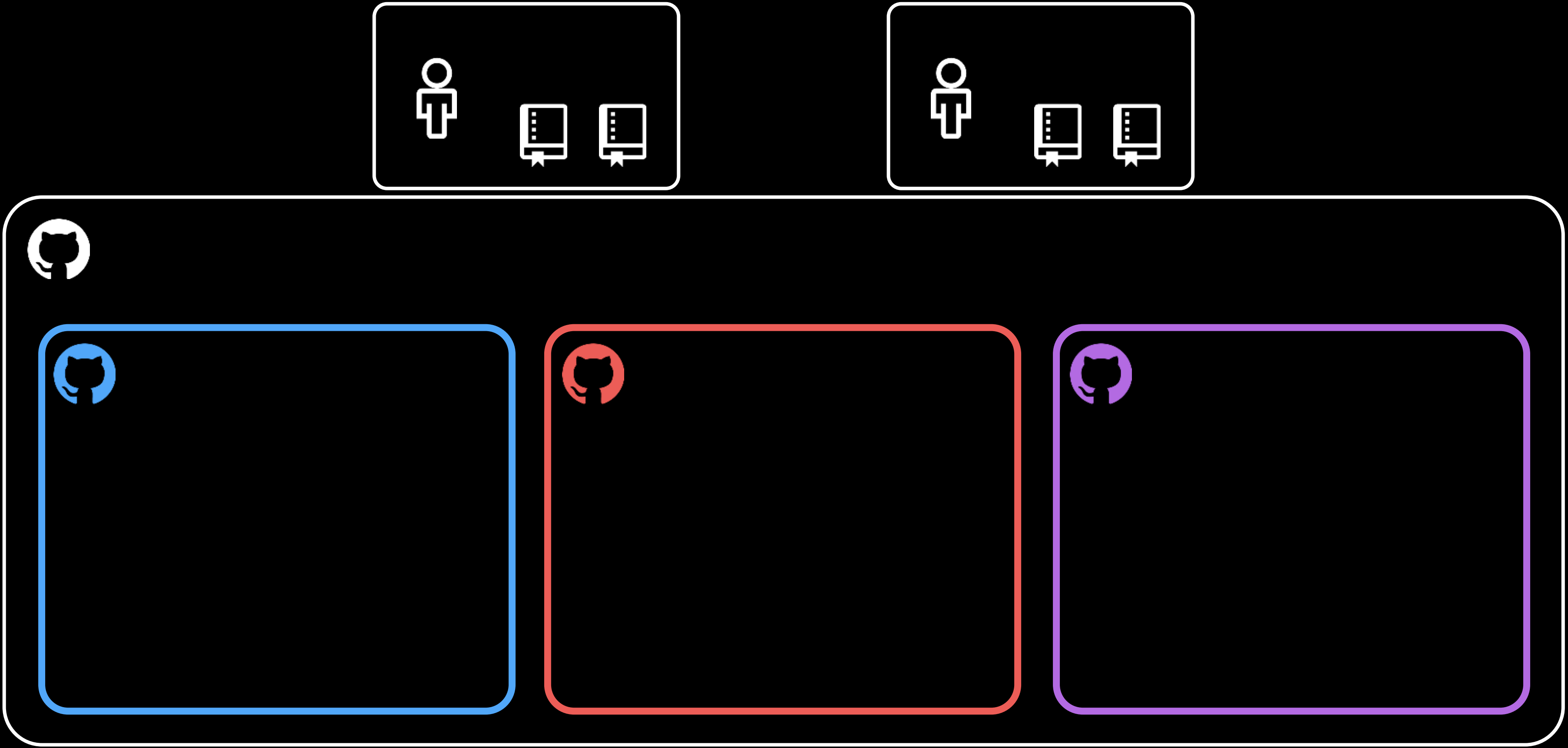
Everything **exists within GitHub (Enterprise)**

Users belong to individuals with their **own namespace**

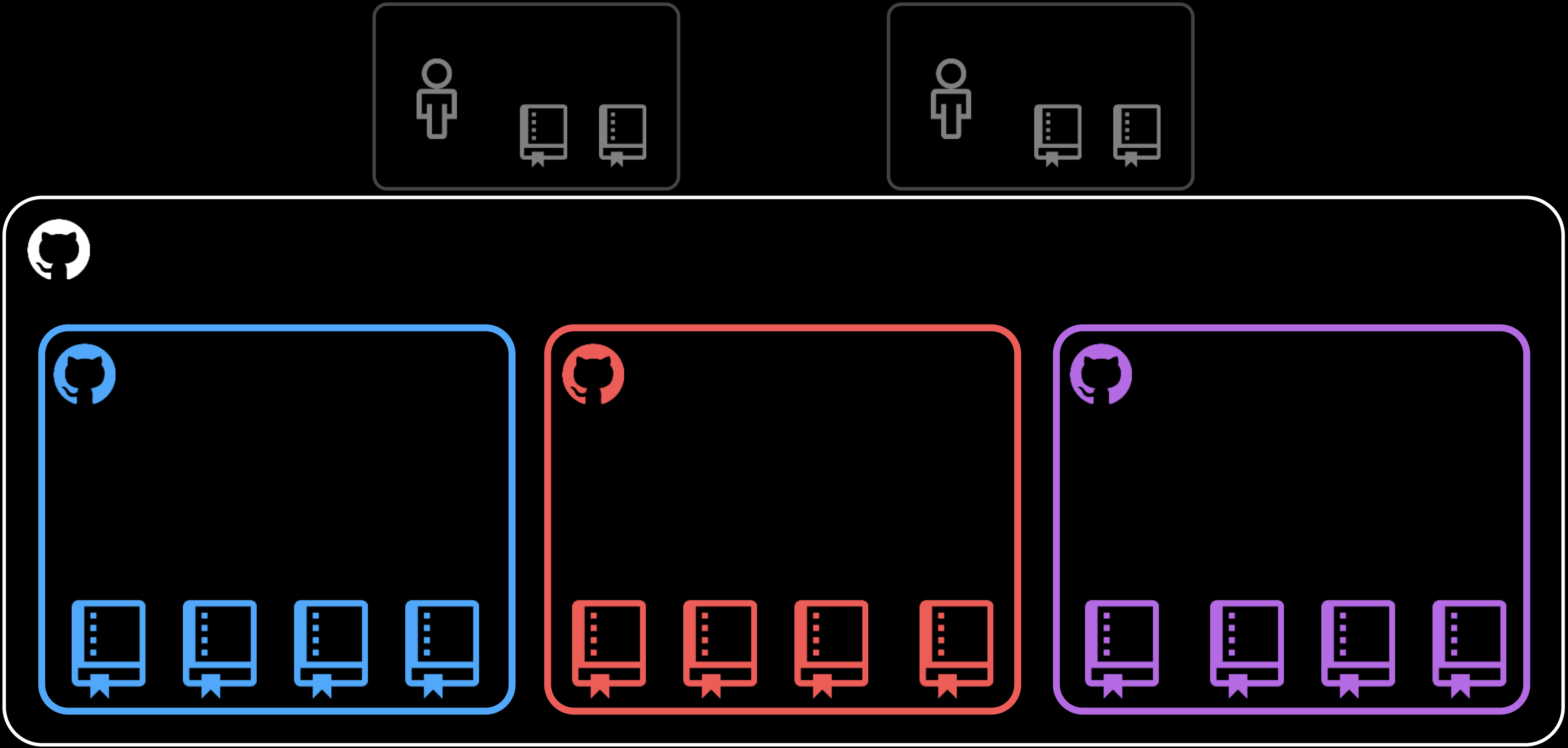
Organizations have **their own namespace**



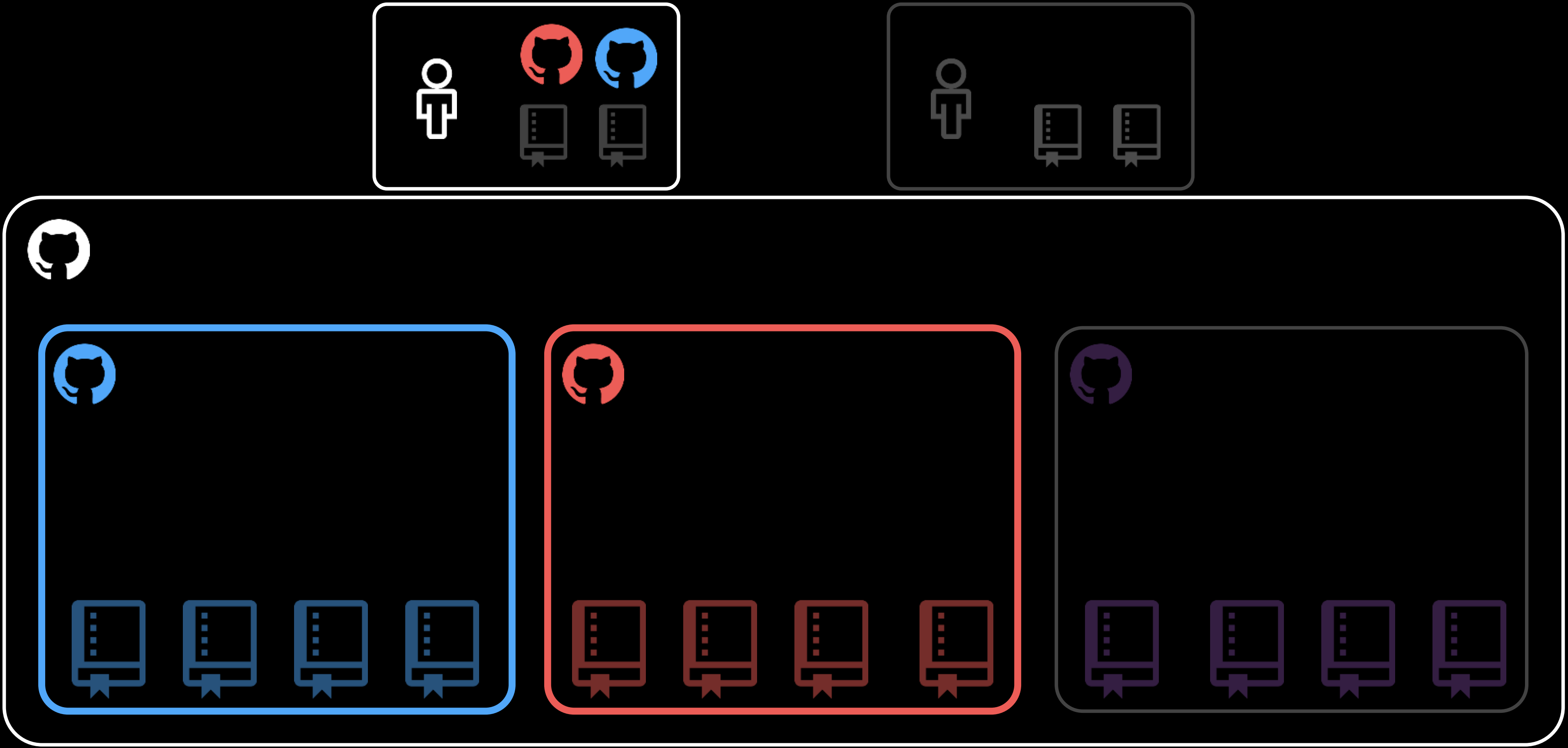
Users can **own repositories**



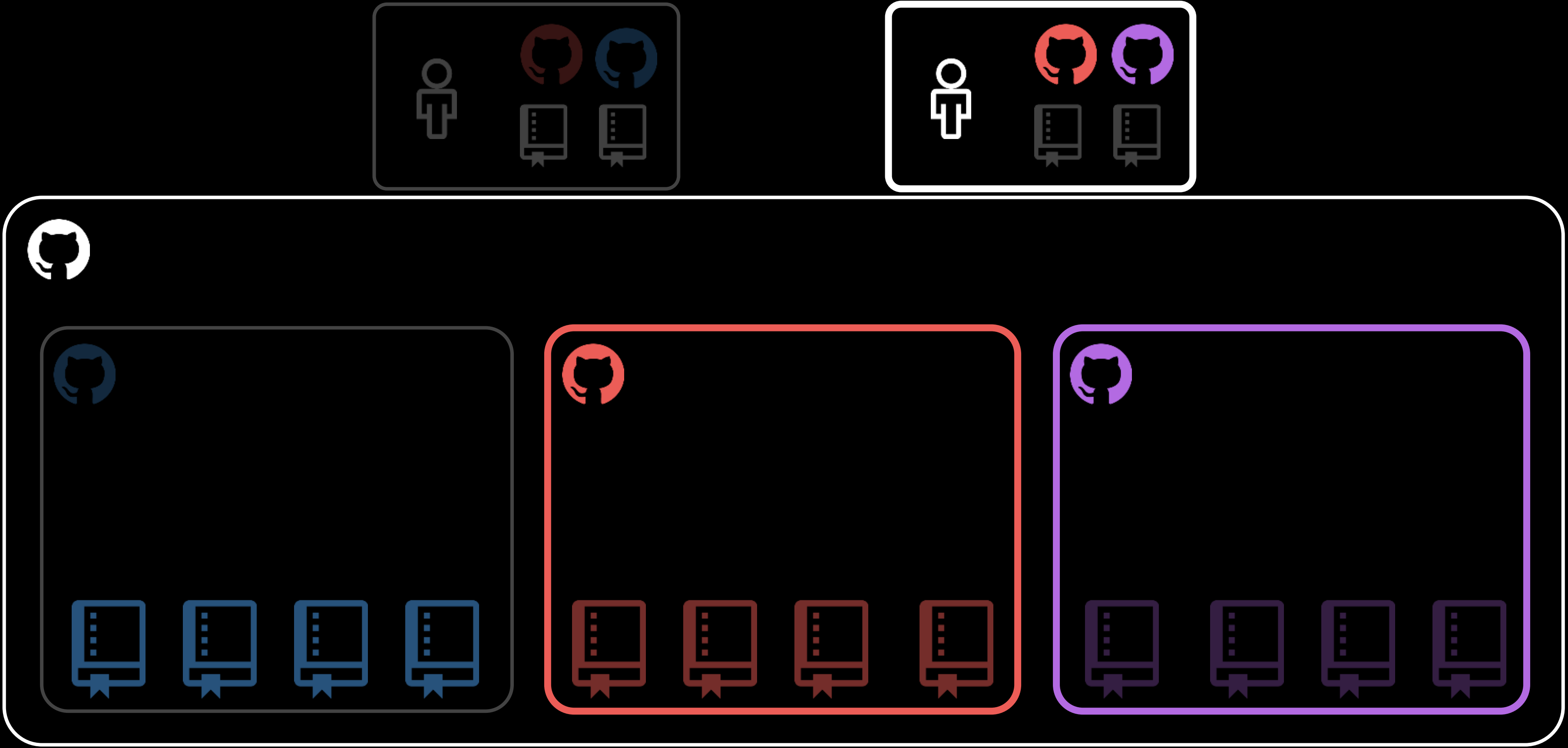
Organizations can **own repositories**



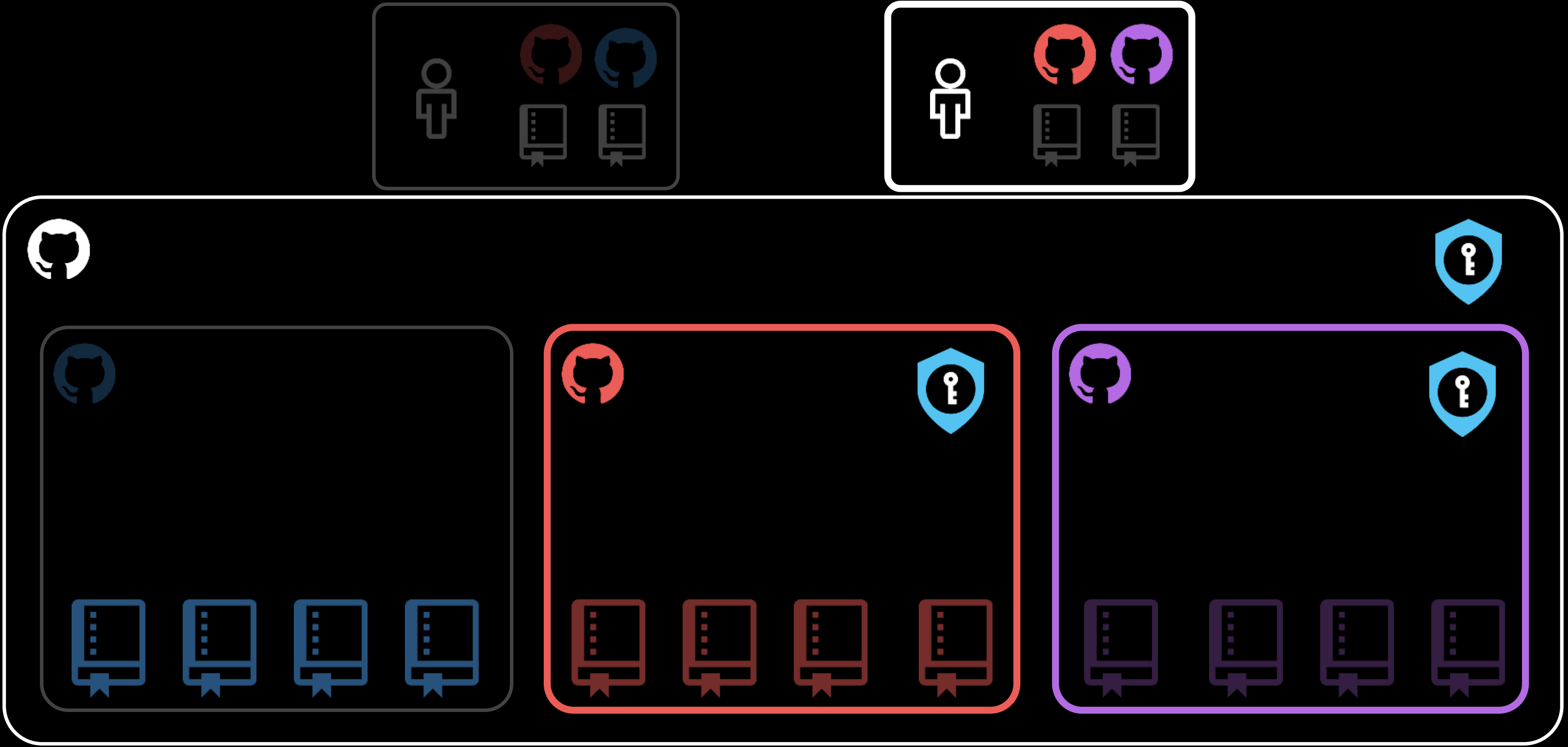
Users can be **members of organizations**



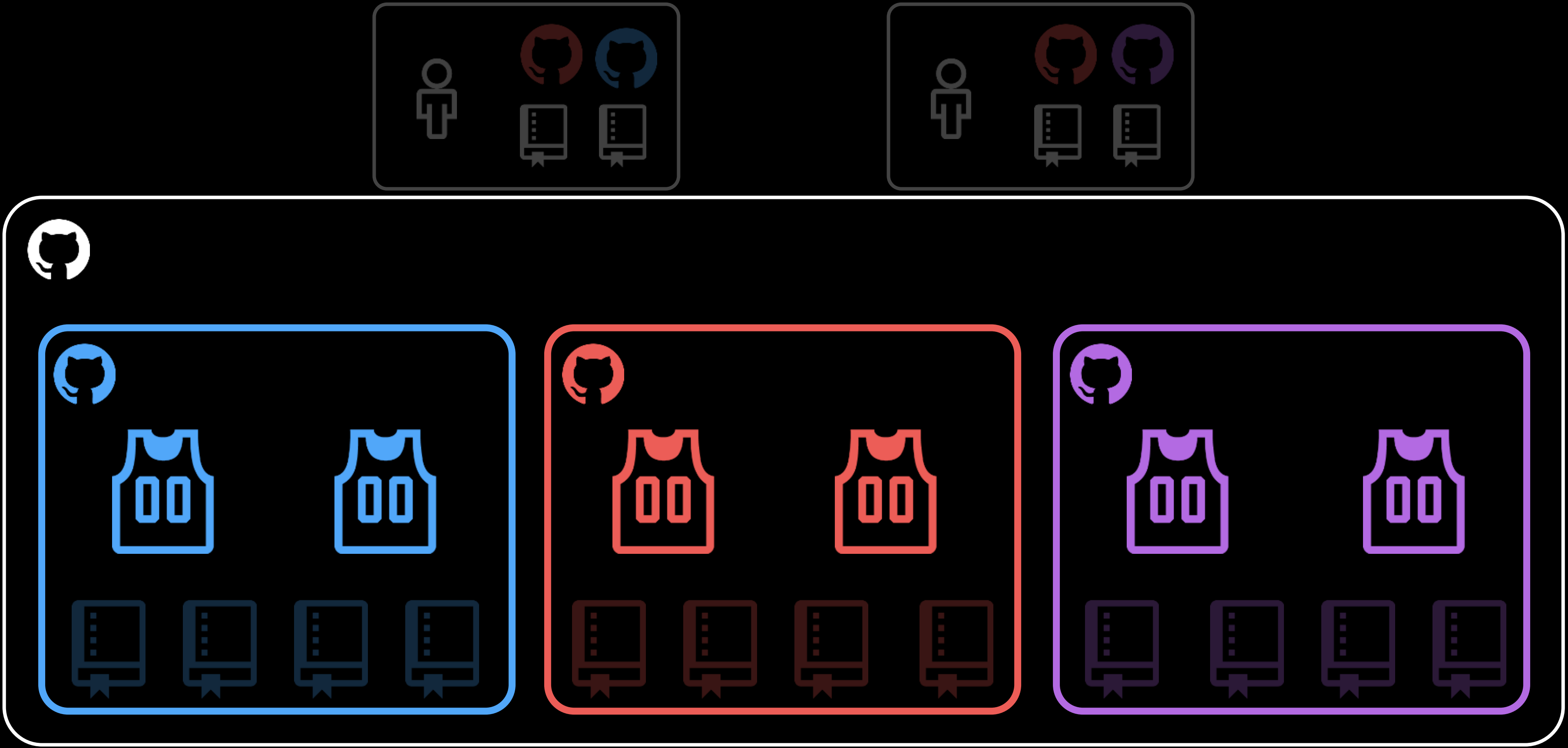
Users can be **members of organizations**



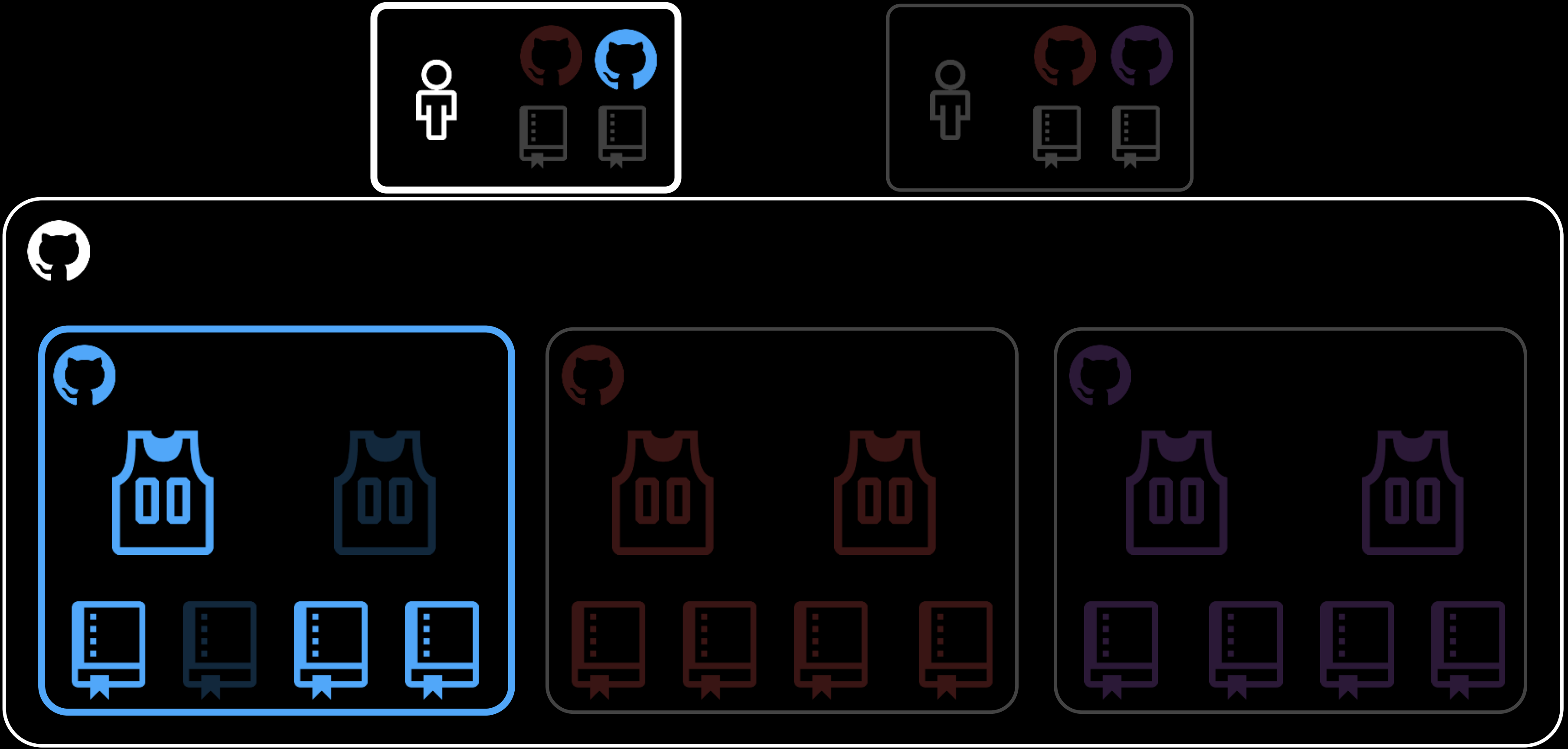
And the org content can be protected by SSO



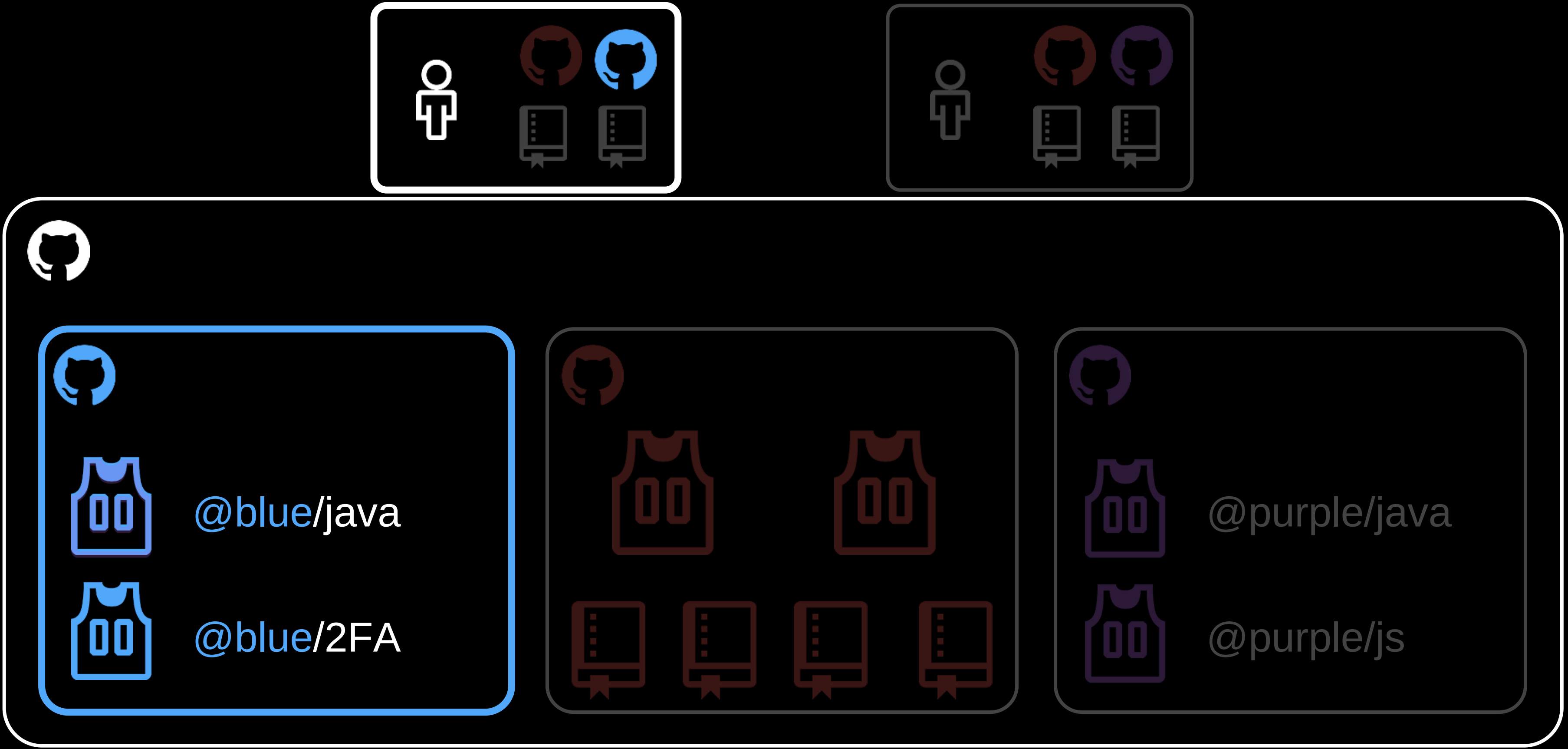
Teams exist **within organizations…**



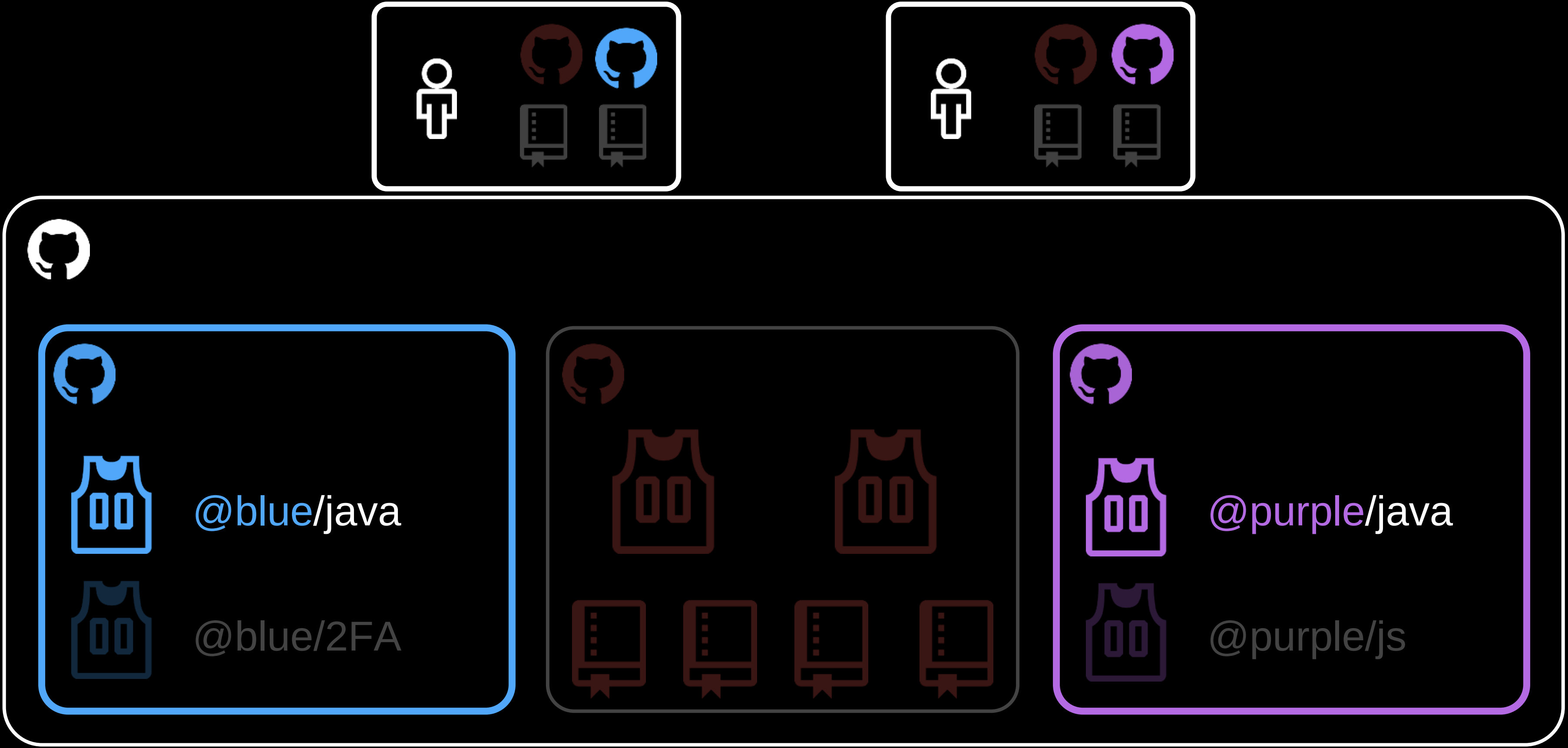
to apply **permissions and group repositories...**



and facilitate **communication**



how can these users participate in java discussions across the enterprise?



Enterprise best practices

Organizations

Have as few organizations as possible. Many have just 1:

- **Insurance company** (3800 users)
- **Drug manufacturer** (2.000+ users)

Having multiple owners

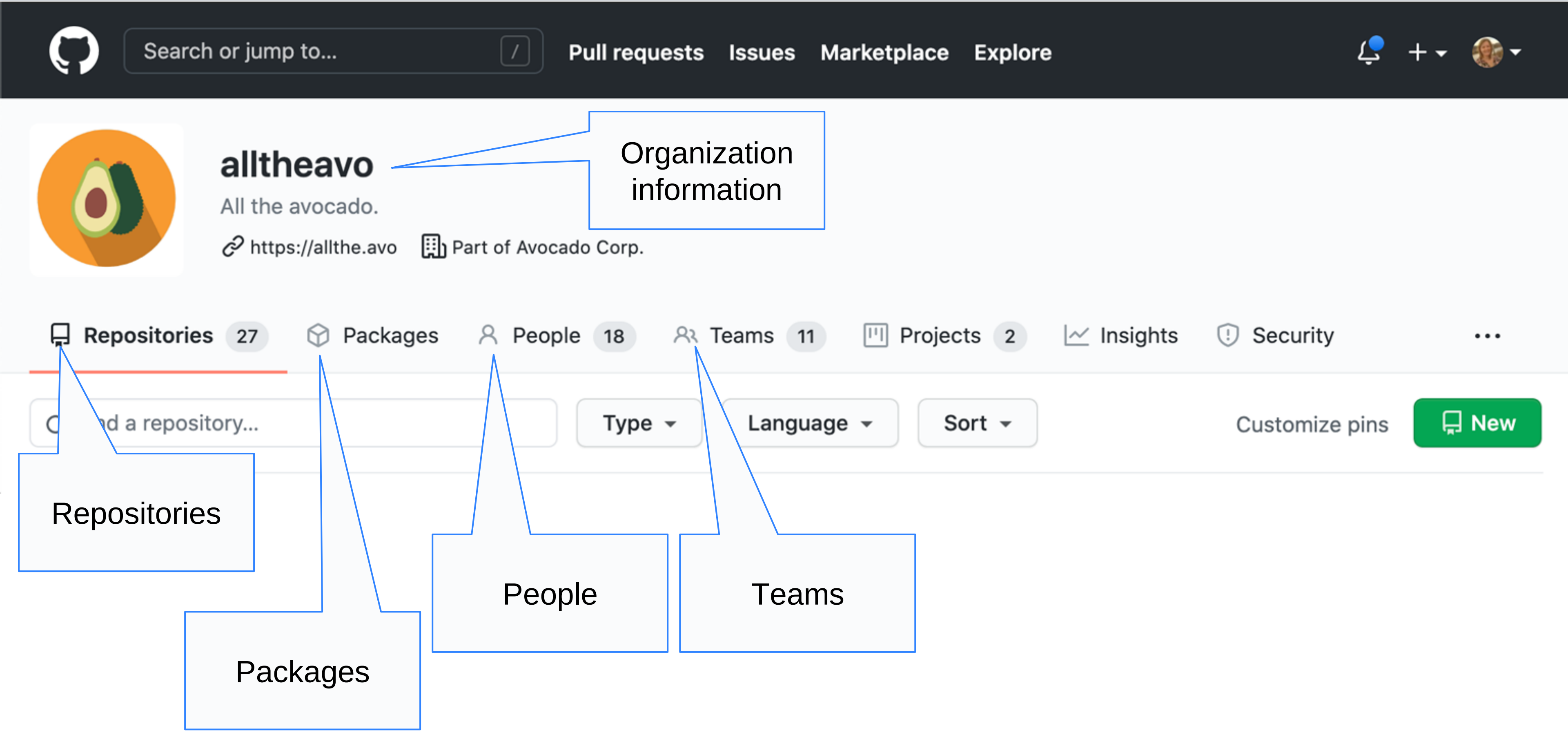
Teams

- Focus on top-level corporate divisions
- Think about who would benefit from:
    - Learning from others
    - Asking questions of others
    - Offer help to others

Organization

Overview ○ Administration and settings

https://github.com/alltheavo



alltheavo

All the avocado.

https://allthe.avo

Part of Avocado Corp.

Organization information

Repositories

Packages

People

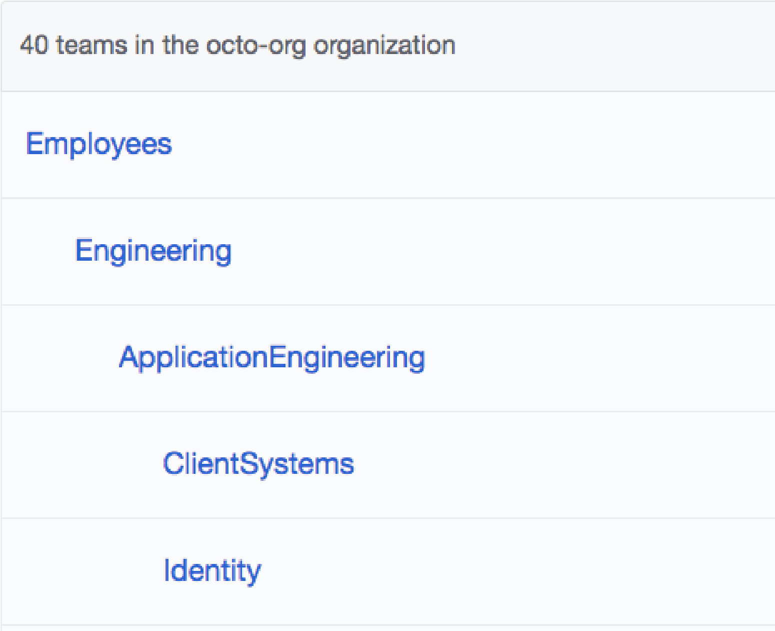
Teams

Reasons to use teams

- Collaboration
- Innersource
- Onboarding and offboarding
- Security

Managing teams

- Nested teams
    - Parents team can have more than one child
    - Child teams inherit parent’s permissions
    - Children receive parent’s notifications
    - Users in a child team belong also to the parent team



40 teams in the octo-org organization

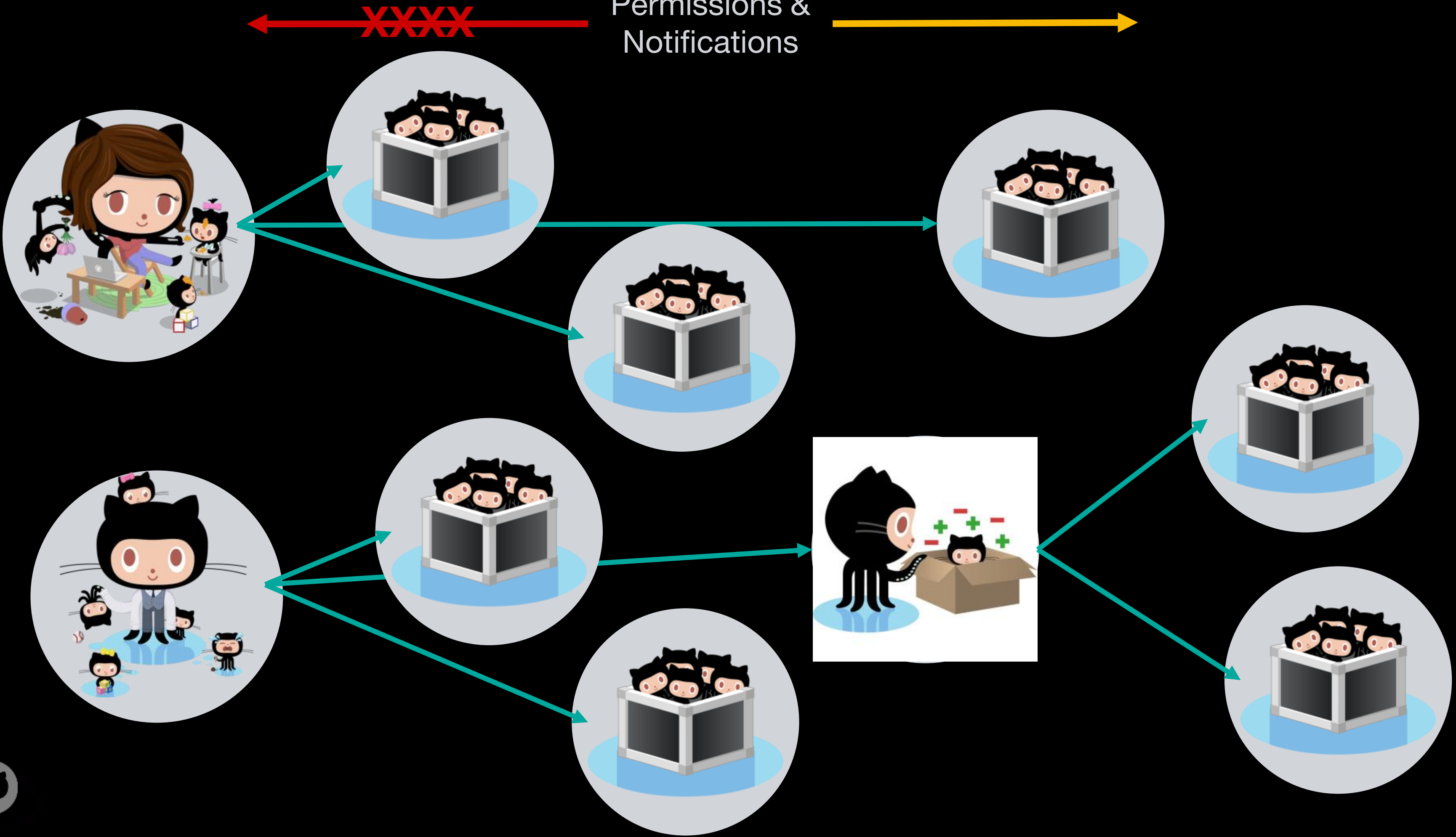
Employees

Engineering

ApplicationEngineering

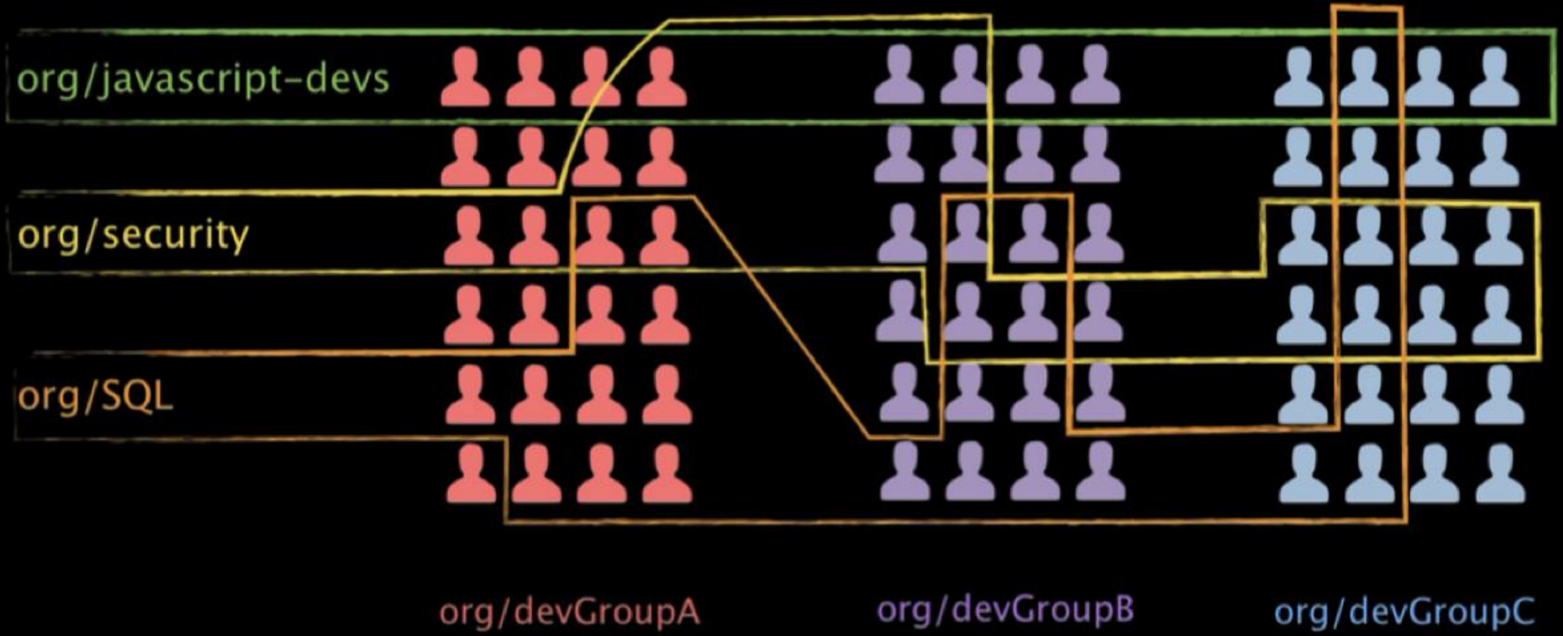
ClientSystems

Identity



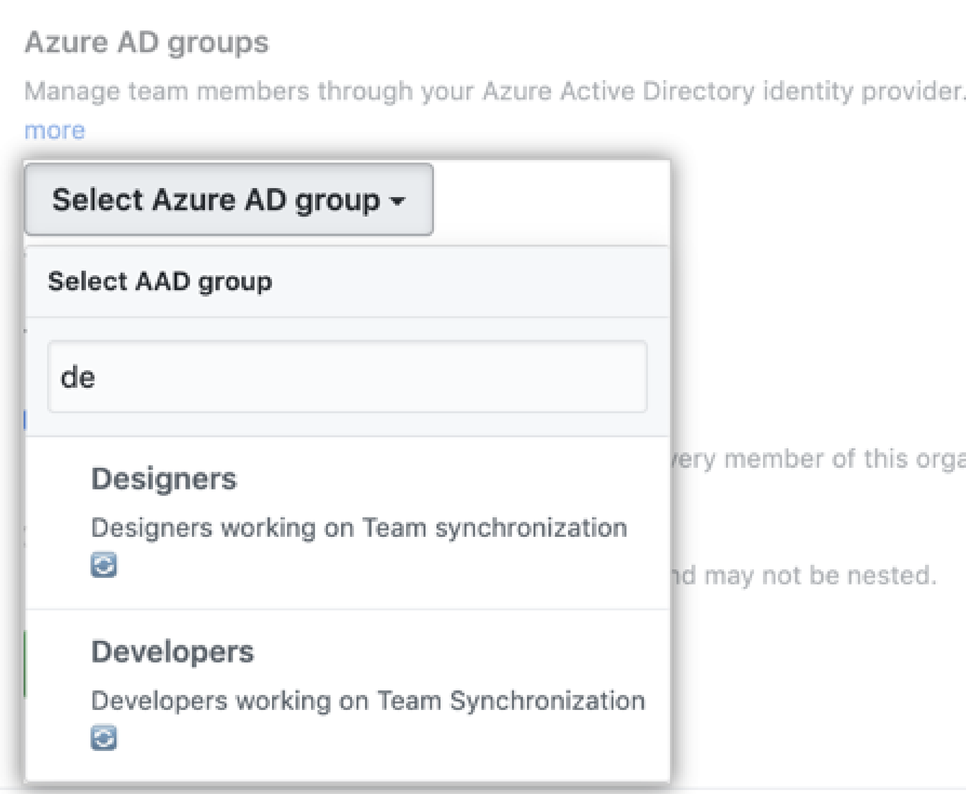
Permissions & Notifications

Team best practices



Managing teams

- Teams managed with Team Sync
  - Integrate IdP to synchronize groups to GitHub Teams
  - Manage rights and permissions in one place
  - Can’t connect to parent team (if using nested teams)
- Teams managed in GitHub
  - Manage membership within GitHub
  - Keep them open, reduce friction



Azure AD groups

Manage team members through your Azure Active Directory identity provider.

more

Select AAD group

de

Designers

Designers working on Team synchronization

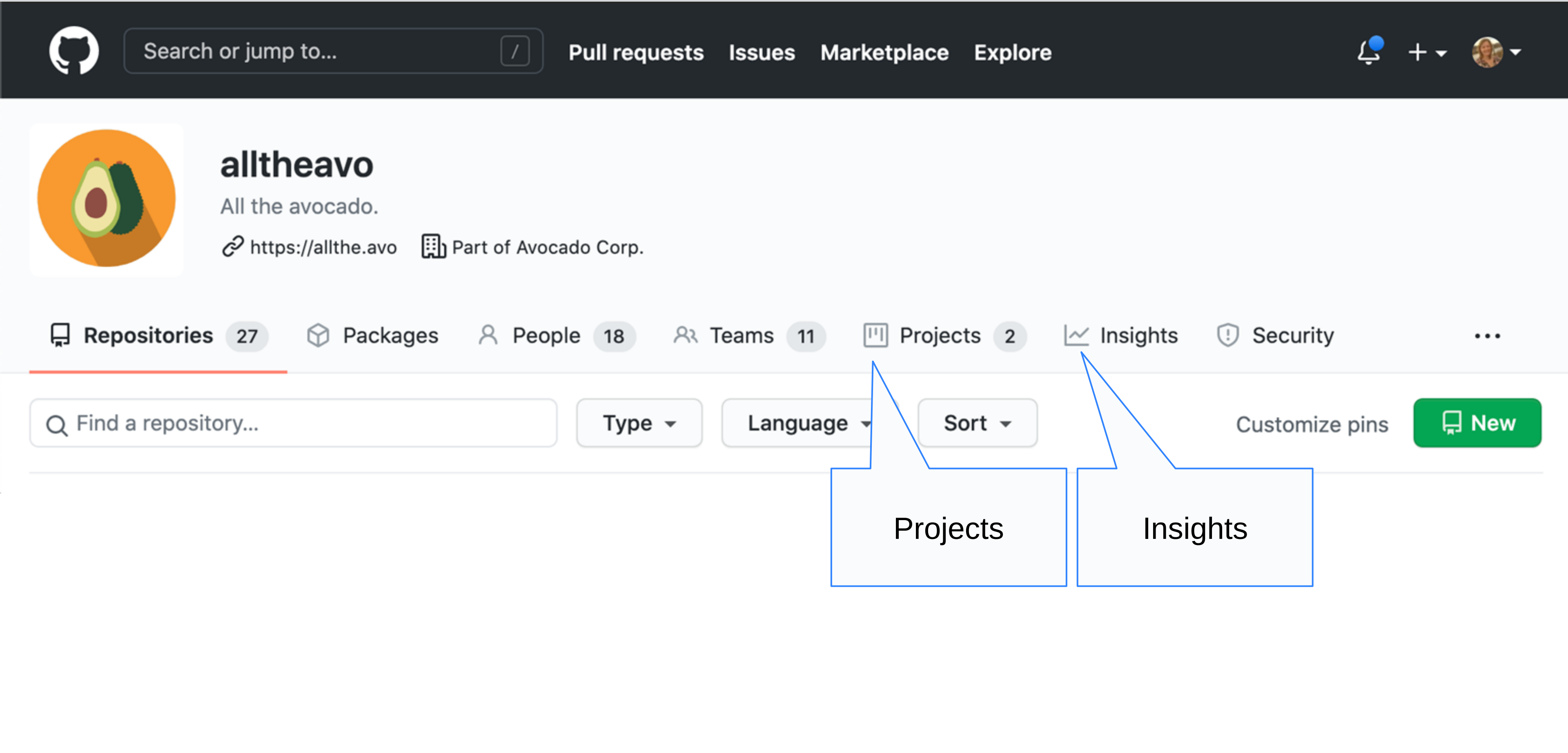
Developers

Developers working on Team Synchronization

very member of this orga

nd may not be nested.

https://github.com/alltheavo



alltheavo

All the avocado.

https://allthe.avo

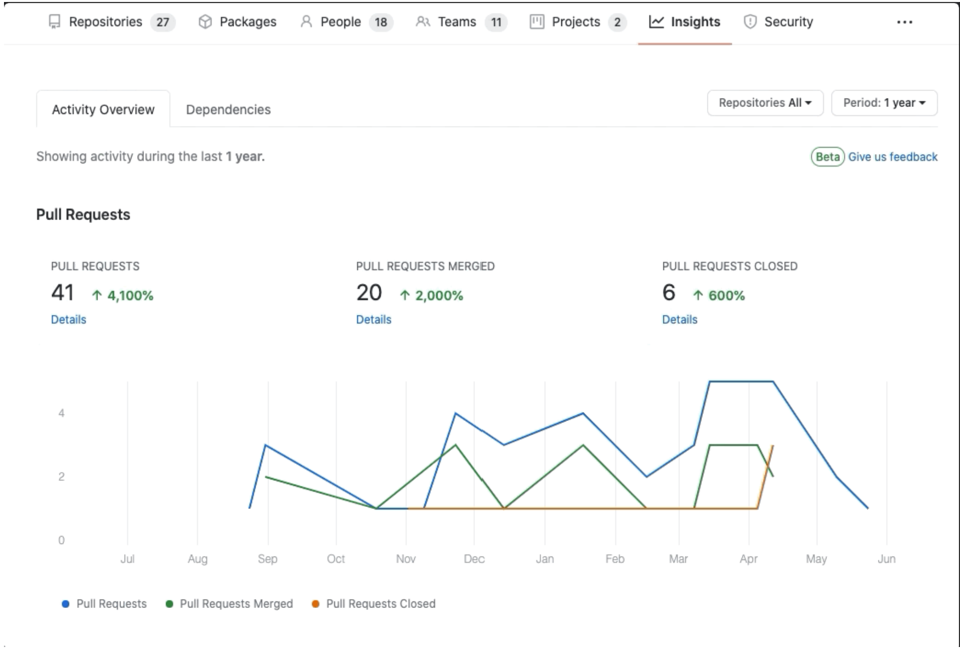
Part of Avocado Corp.

Projects

Insights

Insights - Activity overview

- High level insights
  - Pull requests
  - Issues
  - Where members contribute
- Use to grasp how people is contributing
- For more metrics use the API and present KPIs (grafana, ELK…)



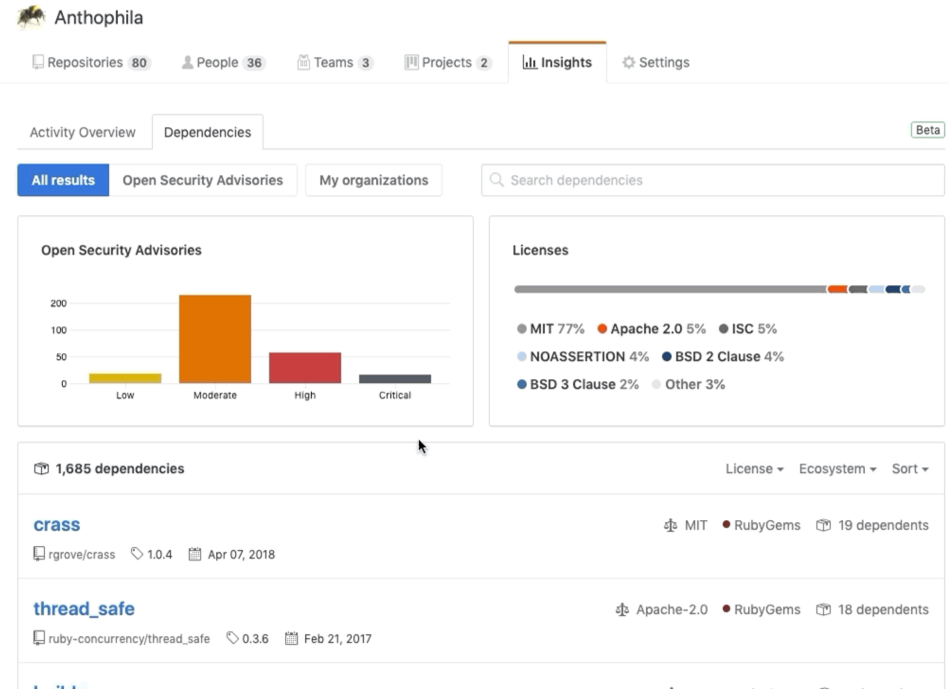
Showing activity during the last **1 year.**

Pull Requests

PULL REQUESTS	PULL REQUESTS MERGED	PULL REQUESTS CLOSED
41 ↑ 4,100%	20 ↑ 2,000%	6 ↑ 600%

Insights - Dependencies

An overview of the dependencies, vulnerabilities, and licenses in your organization



Open Security Advisories

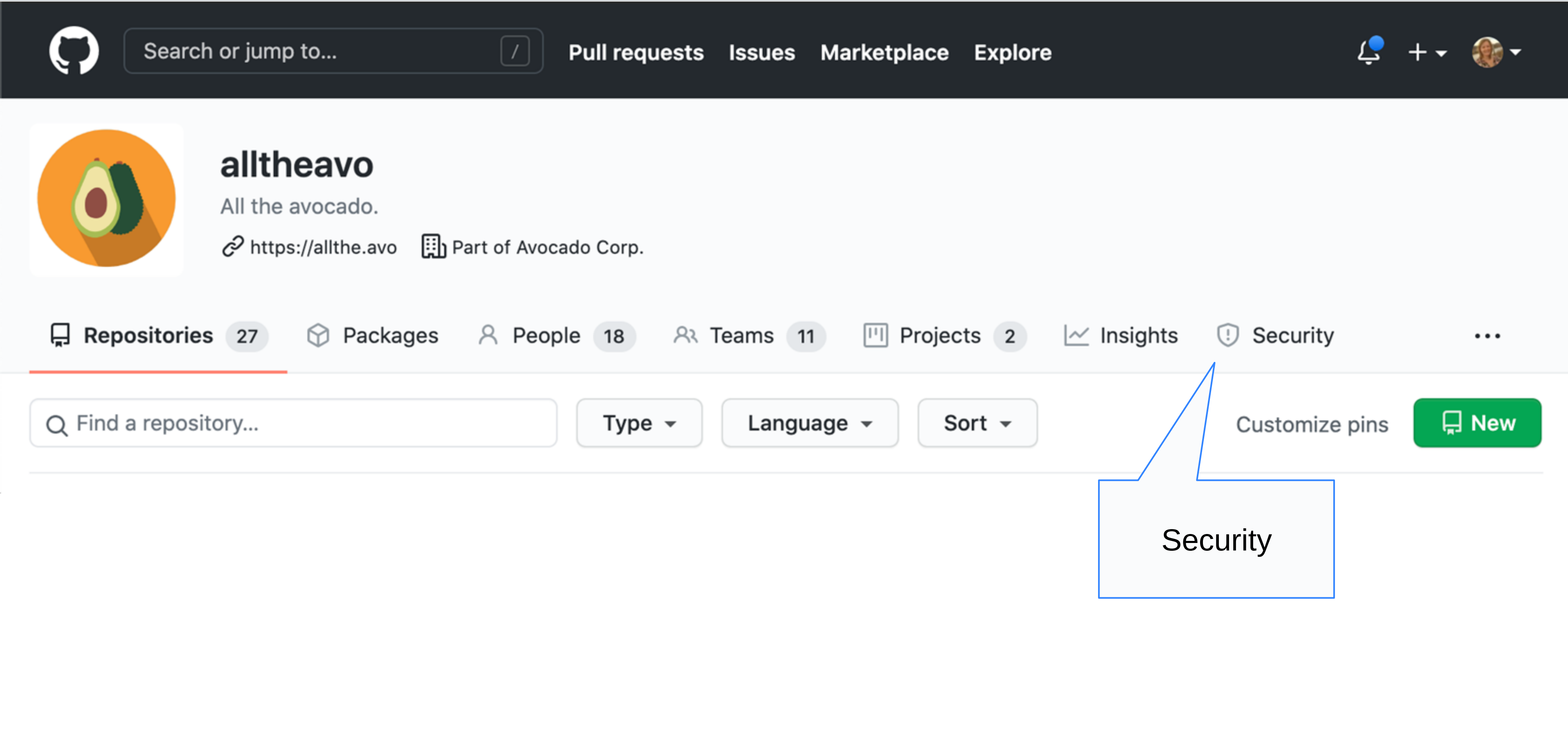
Licenses

● MIT 77% ● Apache 2.0 5% ● ISC 5%
● NOASSERTION 4% ● BSD 2 Clause 4%
● BSD 3 Clause 2% ● Other 3%

1,685 dependencies

crass rgrove/crass 1.0.4 Apr 07, 2018	MIT ● RubyGems 19 dependents
thread_safe ruby-concurrency/thread_safe 0.3.6 Feb 21, 2017	Apache-2.0 ● RubyGems 18 dependents

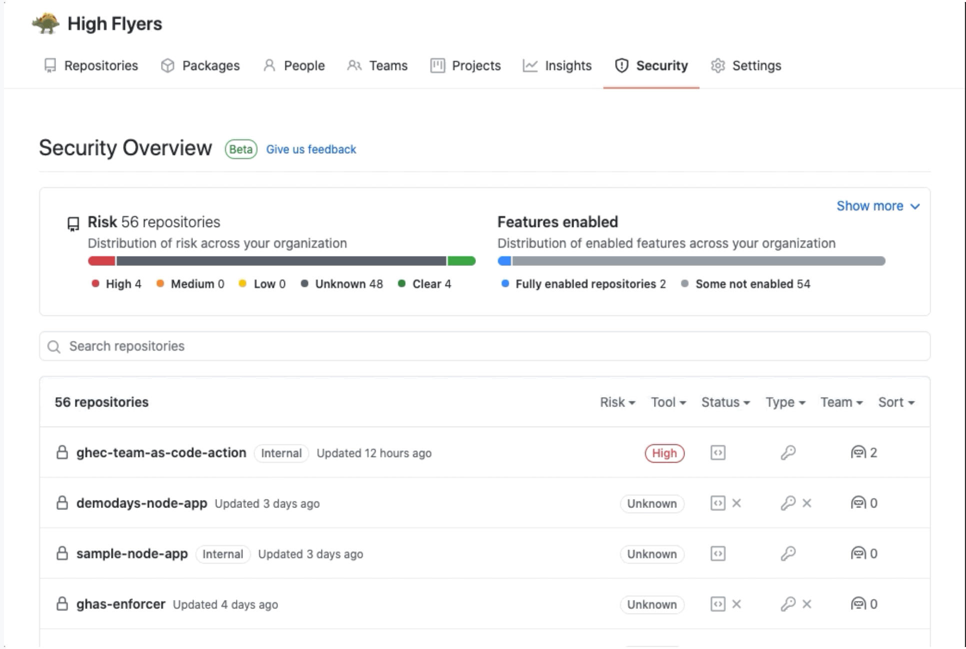
https://github.com/alltheavo



Security

Security

An overview of the security features being utilized within your organization and any alerts that have been identified



Security Overview

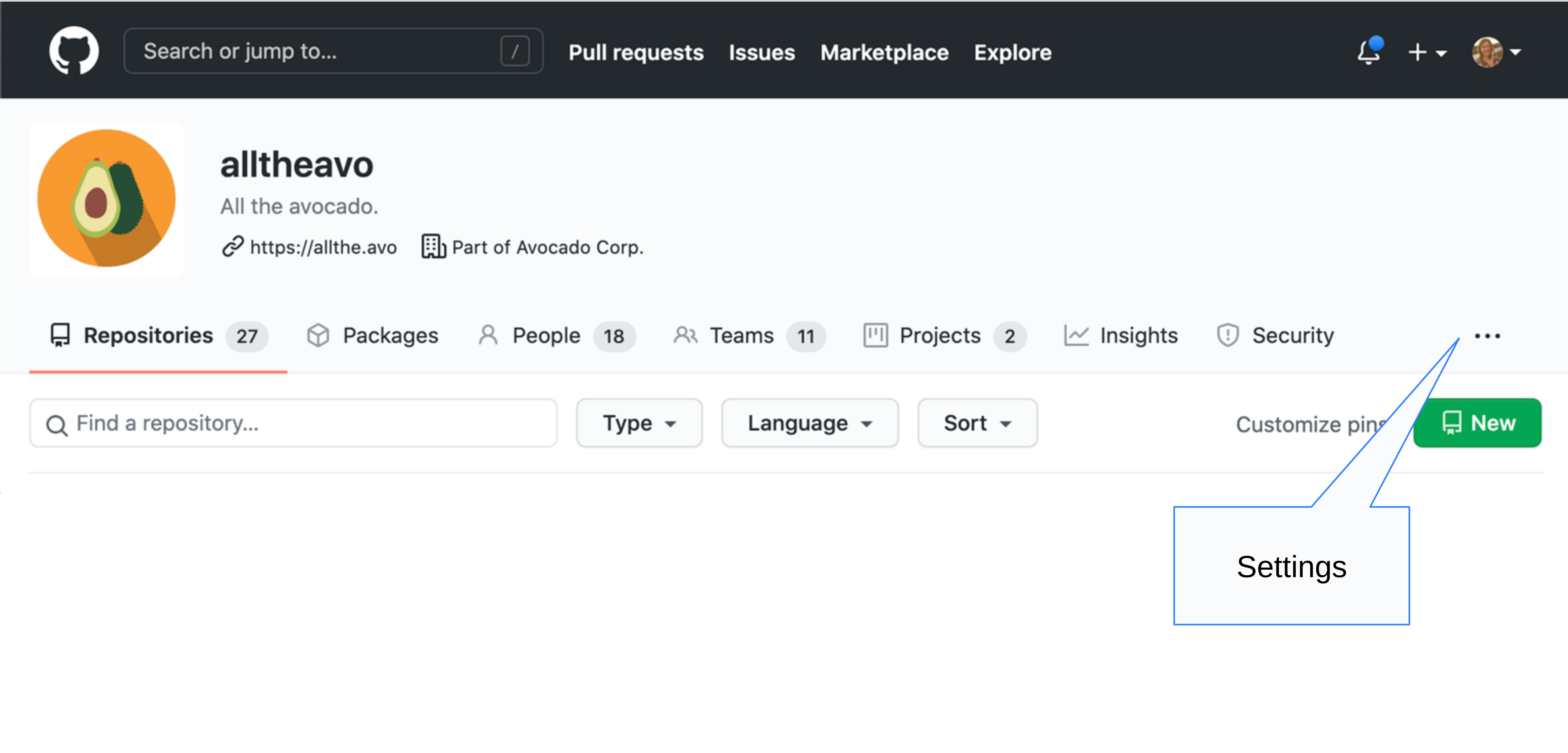
Risk 56 repositories
Distribution of risk across your organization
● High 4 ● Medium 0 ● Low 0 ● Unknown 48 ● Clear 4

Features enabled
Distribution of enabled features across your organization
● Fully enabled repositories 2 ● Some not enabled 54

56 repositories

Repository	Visibility	Updated	Risk	Code scanning	Secret scanning	Alerts
ghec-team-as-code-action	Internal	Updated 12 hours ago	High			2
demodays-node-app		Updated 3 days ago	Unknown	✕	✕	0
sample-node-app	Internal	Updated 3 days ago	Unknown			0
ghas-enforcer		Updated 4 days ago	Unknown	✕	✕	0

https://github.com/alltheavo



alltheavo

All the avocado.

https://allthe.avo

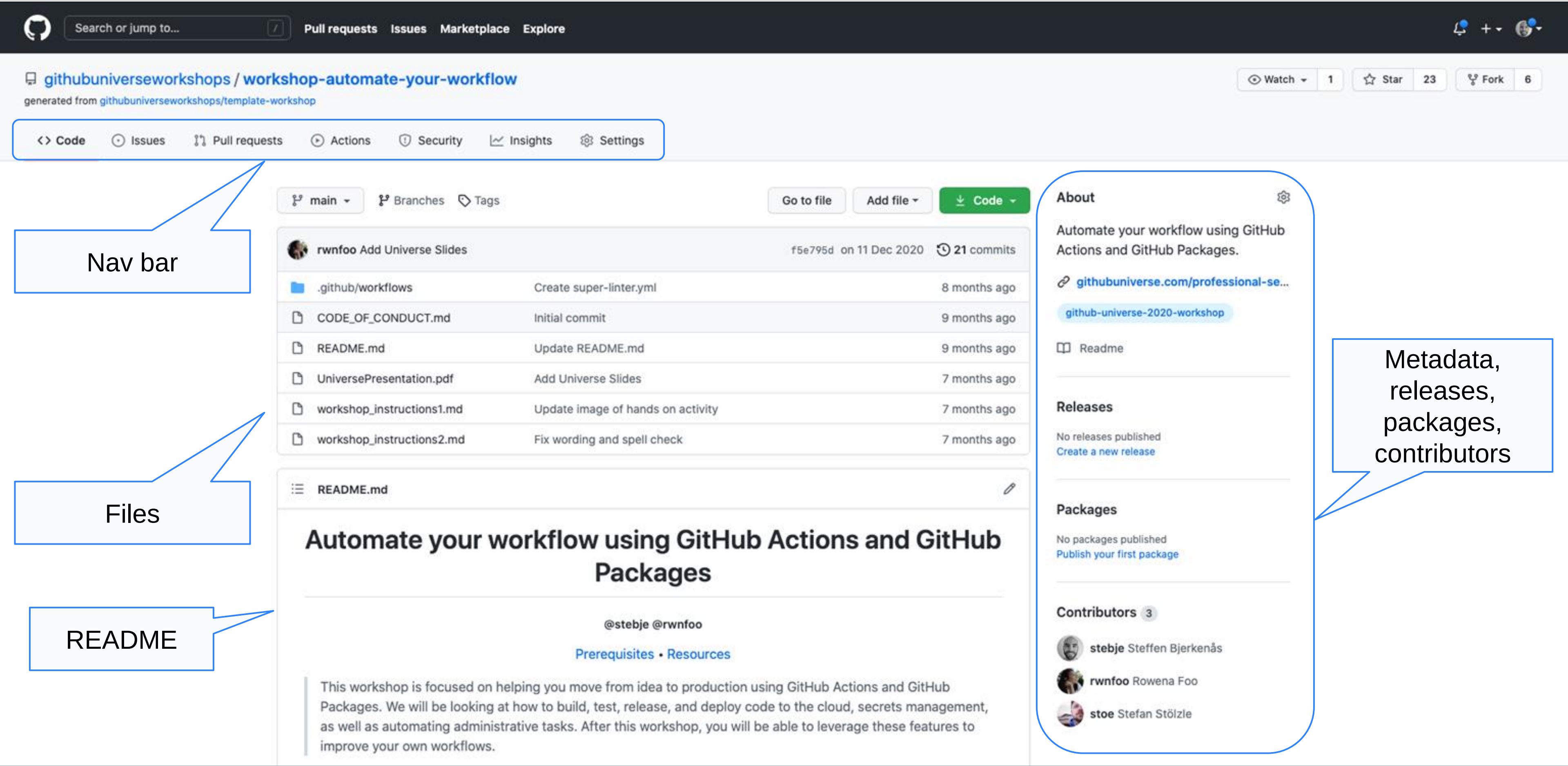
Part of Avocado Corp.

Settings

Share screen

Repository

Repository overview ○ Repository administration ○ Branch protections ○ CODEOWNERS https://github.com/githubuniverseworkshops/workshop-automate-your-workflow ★



githubuniverseworkshops / workshop-automate-your-workflow

generated from githubuniverseworkshops/template-workshop

rwnfoo Add Universe Slides f5e795d on 11 Dec 2020 21 commits

.github/workflows	Create super-linter.yml	8 months ago
CODE_OF_CONDUCT.md	Initial commit	9 months ago
README.md	Update README.md	9 months ago
UniversePresentation.pdf	Add Universe Slides	7 months ago
workshop_instructions1.md	Update image of hands on activity	7 months ago
workshop_instructions2.md	Fix wording and spell check	7 months ago

README.md

Automate your workflow using GitHub Actions and GitHub Packages

@stebje @rwnfoo

Prerequisites • Resources

> This workshop is focused on helping you move from idea to production using GitHub Actions and GitHub Packages. We will be looking at how to build, test, release, and deploy code to the cloud, secrets management, as well as automating administrative tasks. After this workshop, you will be able to leverage these features to improve your own workflows.

About

Automate your workflow using GitHub Actions and GitHub Packages.

githubuniverse.com/professional-se...

github-universe-2020-workshop

Readme

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Contributors 3

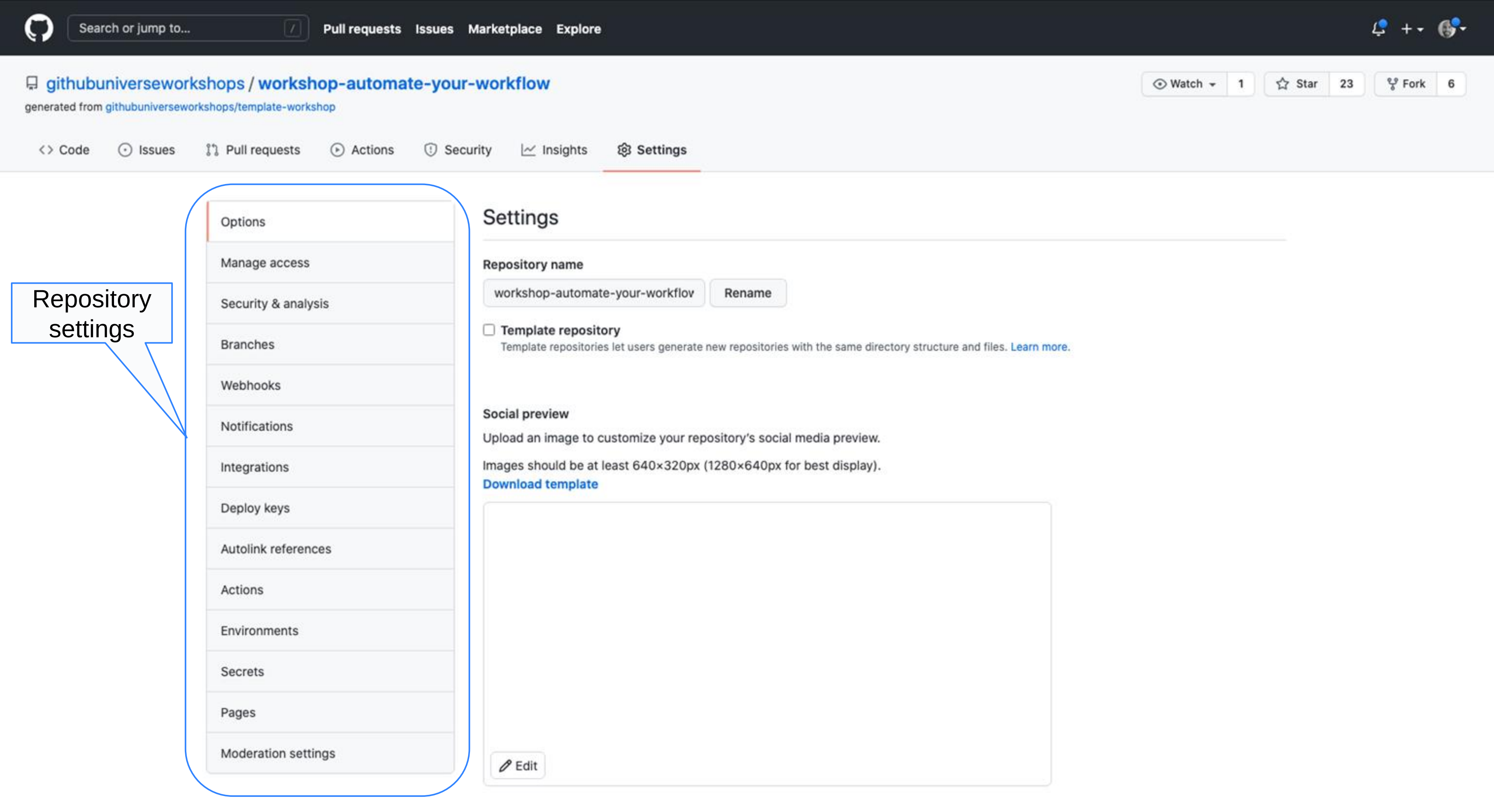
stebje Steffen Bjerkenås
rwnfoo Rowena Foo
stoe Stefan Stölzle

Nav bar

Files

README

Metadata, releases, packages, contributors



githubuniverseworkshops / workshop-automate-your-workflow
generated from githubuniverseworkshops/template-workshop

Repository settings

Settings

Repository name

workshop-automate-your-workflov

☐ **Template repository**
Template repositories let users generate new repositories with the same directory structure and files. Learn more.

Social preview

Upload an image to customize your repository's social media preview.

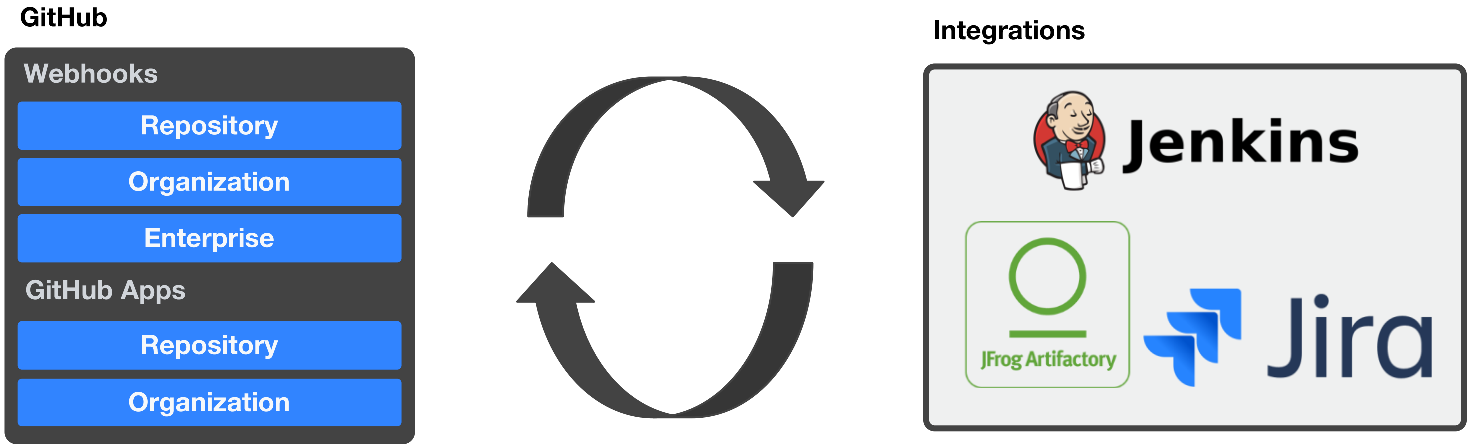
Images should be at least 640×320px (1280×640px for best display).
Download template

Share screen

Additional topics

API Overview ○ Authentication methods ○ Actions overview ○ Marketplace overview

Core integration loop



GitHub

Webhooks

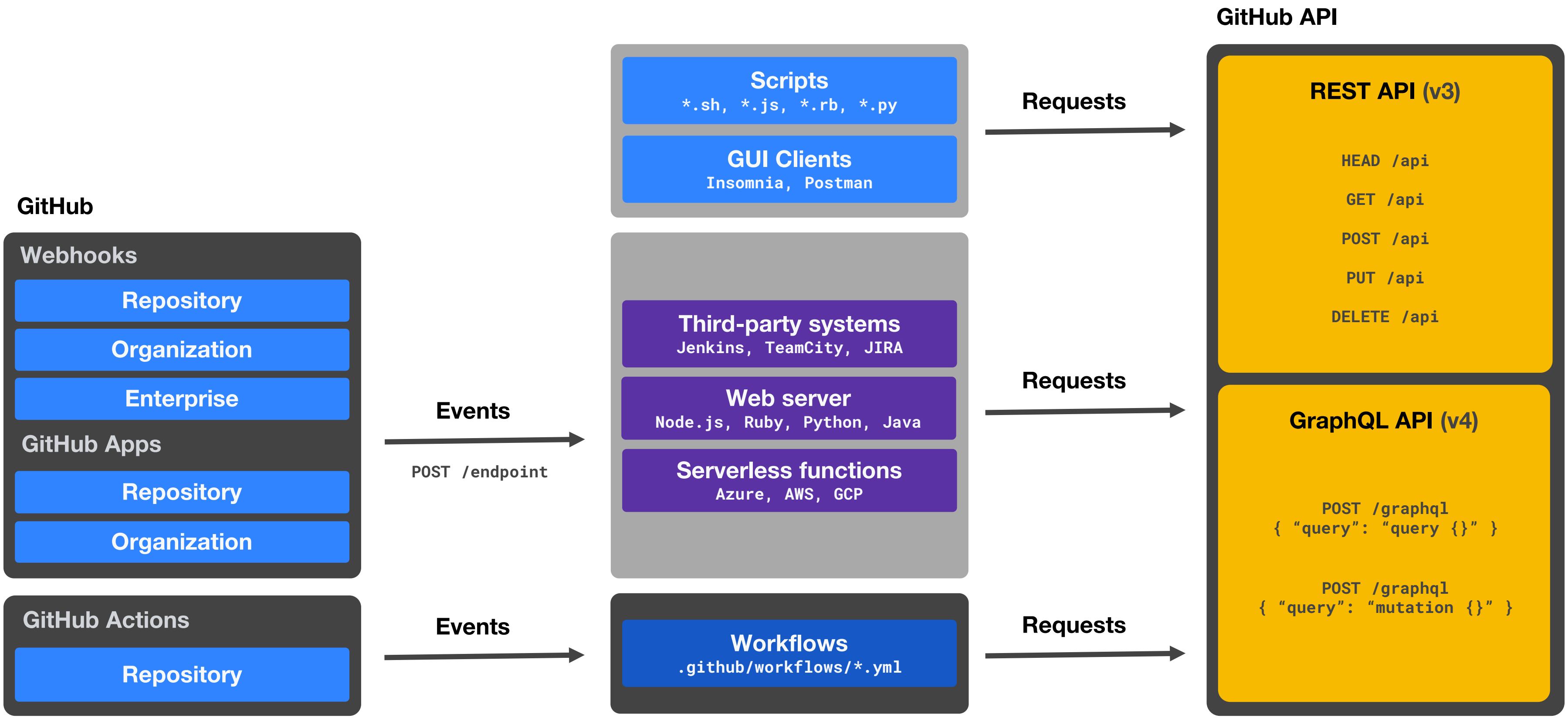
- Repository
- Organization
- Enterprise

GitHub Apps

- Repository
- Organization

Integrations

Core loop overview



GitHub

Webhooks

Repository

Organization

Enterprise

GitHub Apps

Repository

Organization

GitHub Actions

Repository

Events

`POST /endpoint`

Events

Scripts

`*.sh, *.js, *.rb, *.py`

GUI Clients

`Insomnia, Postman`

Third-party systems

`Jenkins, TeamCity, JIRA`

Web server

`Node.js, Ruby, Python, Java`

Serverless functions

`Azure, AWS, GCP`

Workflows

`.github/workflows/*.yml`

Requests

Requests

Requests

GitHub API

REST API (v3)

```
HEAD /api

GET /api

POST /api

PUT /api

DELETE /api
```

GraphQL API (v4)

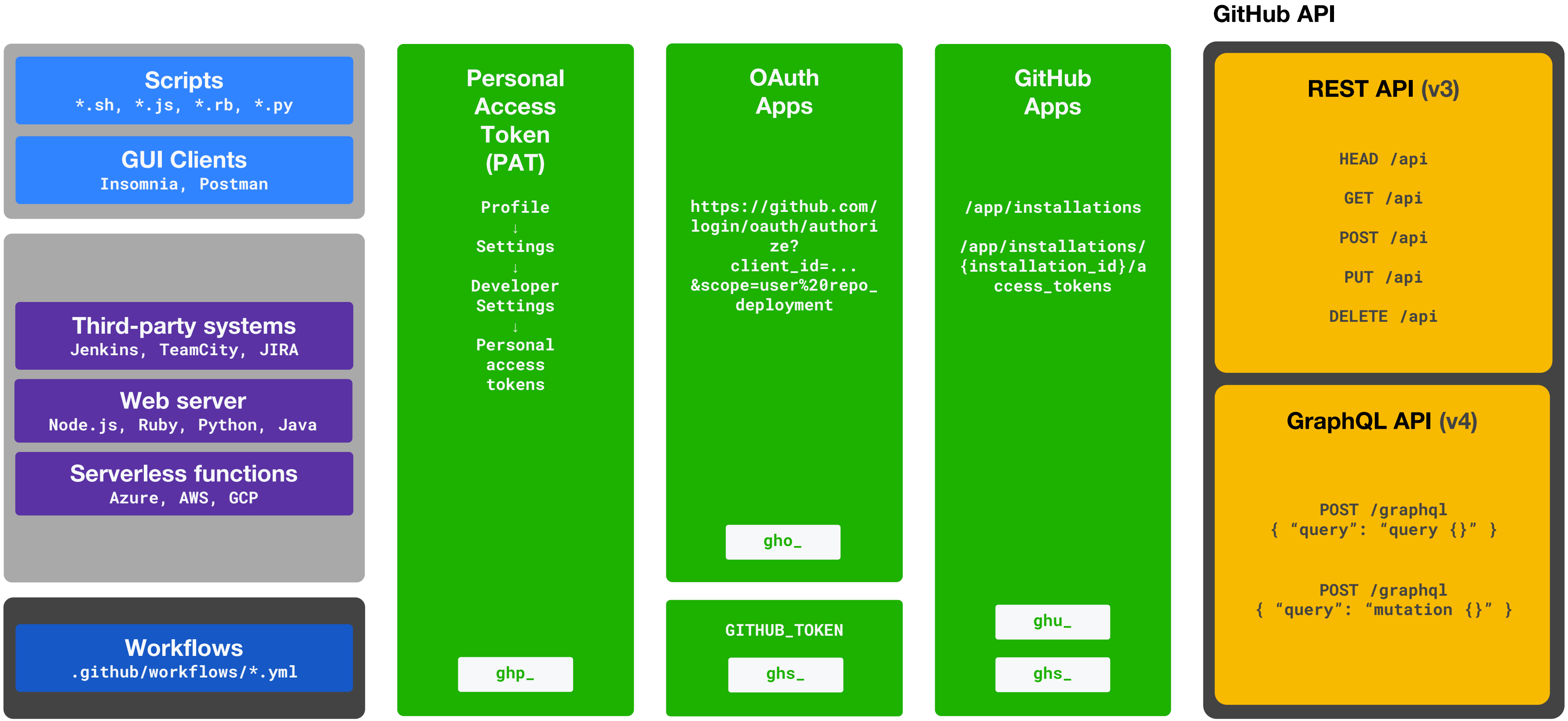
```
POST /graphql
{ “query”: “query {}” }

POST /graphql
{ “query”: “mutation {}” }
```

Additional topics

API Overview ○ Authentication methods ○ Actions overview ○ Marketplace overview

API Authentication



Scripts
*.sh, *.js, *.rb, *.py

GUI Clients
Insomnia, Postman

Third-party systems
Jenkins, TeamCity, JIRA

Web server
Node.js, Ruby, Python, Java

Serverless functions
Azure, AWS, GCP

Workflows
.github/workflows/*.yml

Personal Access Token (PAT)

Profile
↓
Settings
↓
Developer Settings
↓
Personal access tokens

ghp_

OAuth Apps

https://github.com/login/oauth/authorize?
client_id=...
&scope=user%20repo_deployment

gho_

GITHUB_TOKEN

ghs_

GitHub Apps

/app/installations

/app/installations/{installation_id}/access_tokens

ghu_

ghs_

GitHub API

REST API (v3)

HEAD /api

GET /api

POST /api

PUT /api

DELETE /api

GraphQL API (v4)

POST /graphql
{ “query”: “query {}” }

POST /graphql
{ “query”: “mutation {}” }

Authentication methods

- **GitHub Apps**
- OAuth Apps
- Personal access tokens
- Deploy keys
- Machine users

- A user or organization can own up to **100 GitHub Apps**
- A GitHub App should take actions **independent** of a user
- The GitHub App be installed in a personal account or an organization
- Don’t expect the GitHub App to know and do everything a user can
- Search for "**Works with GitHub Apps**" in the docs
- Can behave as OAuth apps with more permissions
- Up top 15k requests (enterprise)
- Permission changes require approval

Authentication methods

- **GitHub Apps**
- OAuth Apps
- Personal access tokens
- Deploy keys
- Machine users

Installation (S2S)

App ID + Private key .pem + expiration = JWT

JWT + installation id = API Token

OAuth (U2S)

Client Id + callback url = auth request + code

code + client secret (stored in server) = API Token

Authentication methods

- GitHub Apps
- **OAuth Apps**
- Personal access tokens
- Deploy keys
- Machine users

- A user or organization can own up to **100 OAuth apps**
- An OAuth App should always **act as the authenticated GitHub user** across all of GitHub
- An OAuth App can be used as an **identity provider** by enabling a “Login with GitHub” for the authenticated user
- OAuth Apps can act on all the **authenticated user’s** resources
- Limit of 5k requests
- Requires OAuth flow (client id + auth + code + client secret)

Authentication methods

- GitHub Apps
- OAuth Apps
- **Personal access tokens**
- Deploy keys
- Machine users

- Remember to use this token to represent **yourself only**
- You can perform **one-off cURL requests**
- You can run **personal scripts**
- **Don’t set up a script for your whole team or company to use**
- Use a machine user for authentication
- Limit of 5k requests
- Part of token scanning if leaked publicly
- Removed automatically after one year without use
- Limited permissions by the user

Authentication methods

- GitHub Apps
- OAuth Apps
- Personal access tokens
- **Deploy keys**
- Machine users

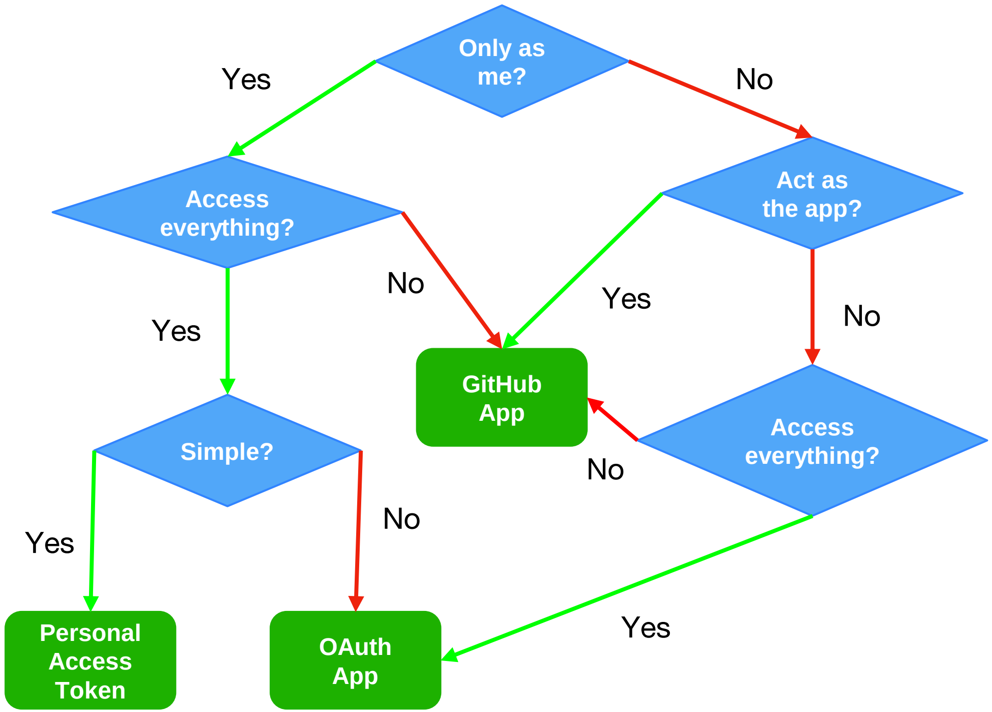
- Anyone with access to the repository and server can deploy the project
- Users don’t have to change their local SSH settings
- Deploy keys are read-only by default
- Deploy keys only grant access to a single repository
- Deploy keys are usually not protected by a passphrase

Authentication methods

- GitHub Apps
- OAuth Apps
- Personal access tokens
- Deploy keys
- **Machine users**

- Anyone with access to the repository and server can deploy the project
- No (human) users need to change their local SSH settings
- Multiple keys are not needed
- Only organizations can restrict machine users to read-only access
- Machine user keys, like deploy keys, are usually not protected by a passphrase

Which should you choose?



Additional topics

API Overview ○ Authentication methods ○ Actions overview ○ Marketplace overview

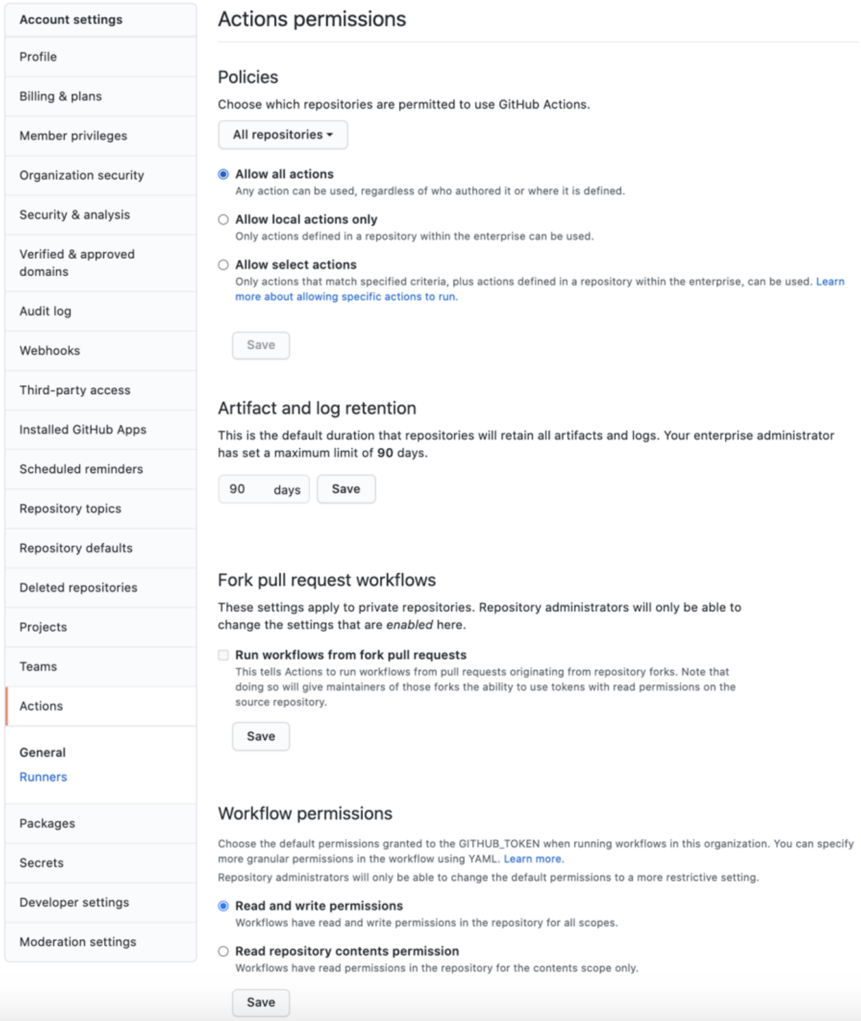
What is GitHub Actions

GitHub Actions is a GitHub product that allows you to **automate your workflows.**

- Workflows stored as `yml` files
- Fully integrated with GitHub
- Respond to GitHub events
- Live logs and visualized workflow execution
- Community-powered workflows
- GitHub-hosted or self-hosted runners
- Built-in secret store

Actions policies

- Configure Actions policies on enterprise / organization / repository level
  - Which Actions are allowed
  - Artifact retention period
  - Running workflows from fork PRs
  - Permissions of `GITHUB_TOKEN`



Actions permissions

Policies

Choose which repositories are permitted to use GitHub Actions.

All repositories ▾

● **Allow all actions**
Any action can be used, regardless of who authored it or where it is defined.

○ **Allow local actions only**
Only actions defined in a repository within the enterprise can be used.

○ **Allow select actions**
Only actions that match specified criteria, plus actions defined in a repository within the enterprise, can be used. Learn more about allowing specific actions to run.

Save

Artifact and log retention

This is the default duration that repositories will retain all artifacts and logs. Your enterprise administrator has set a maximum limit of **90** days.

90 days Save

Fork pull request workflows

These settings apply to private repositories. Repository administrators will only be able to change the settings that are *enabled* here.

□ **Run workflows from fork pull requests**
This tells Actions to run workflows from pull requests originating from repository forks. Note that doing so will give maintainers of those forks the ability to use tokens with read permissions on the source repository.

Save

Workflow permissions

Choose the default permissions granted to the GITHUB_TOKEN when running workflows in this organization. You can specify more granular permissions in the workflow using YAML. Learn more.

Repository administrators will only be able to change the default permissions to a more restrictive setting.

● **Read and write permissions**
Workflows have read and write permissions in the repository for all scopes.

○ **Read repository contents permission**
Workflows have read permissions in the repository for the contents scope only.

Save

Sharing workflows in an organization

- Use GitHub **actions starter templates** from `.github` repository to share workflows
- Use GitHub packages and `ghcr.io` to share actions using docker execution and **package registry** permissions (only for public registries)
- Git Submodules or subtrees (not most recommended option)
- **(Upcoming)** Organization workflow execution. Open source concept: https://github.com/SvanBoxel/organization-workflows

Sharing private actions

Use a **GitHub App** to clone actions from:

- Actions in different repositories
- Actions monorepo
- Actions separate organization

```yaml
jobs:
  do-something:
    runs-on: ubuntu-latest

    steps:
      - name: Generate app installation token
        id: app
        uses: peter-murray/workflow-application-token-action@v1
        with:
          application_id: ${{ secrets.APP_ID }}
          application_private_key: ${{ secrets.PRIV_KEY }}

      - name: Checkout private repository
        id: checkout_repo
        uses: actions/checkout@v2
        with:
          repository: my-org/repo
          path: path/to/privateAction
          token: ${{ steps.app.outputs.token }}
```

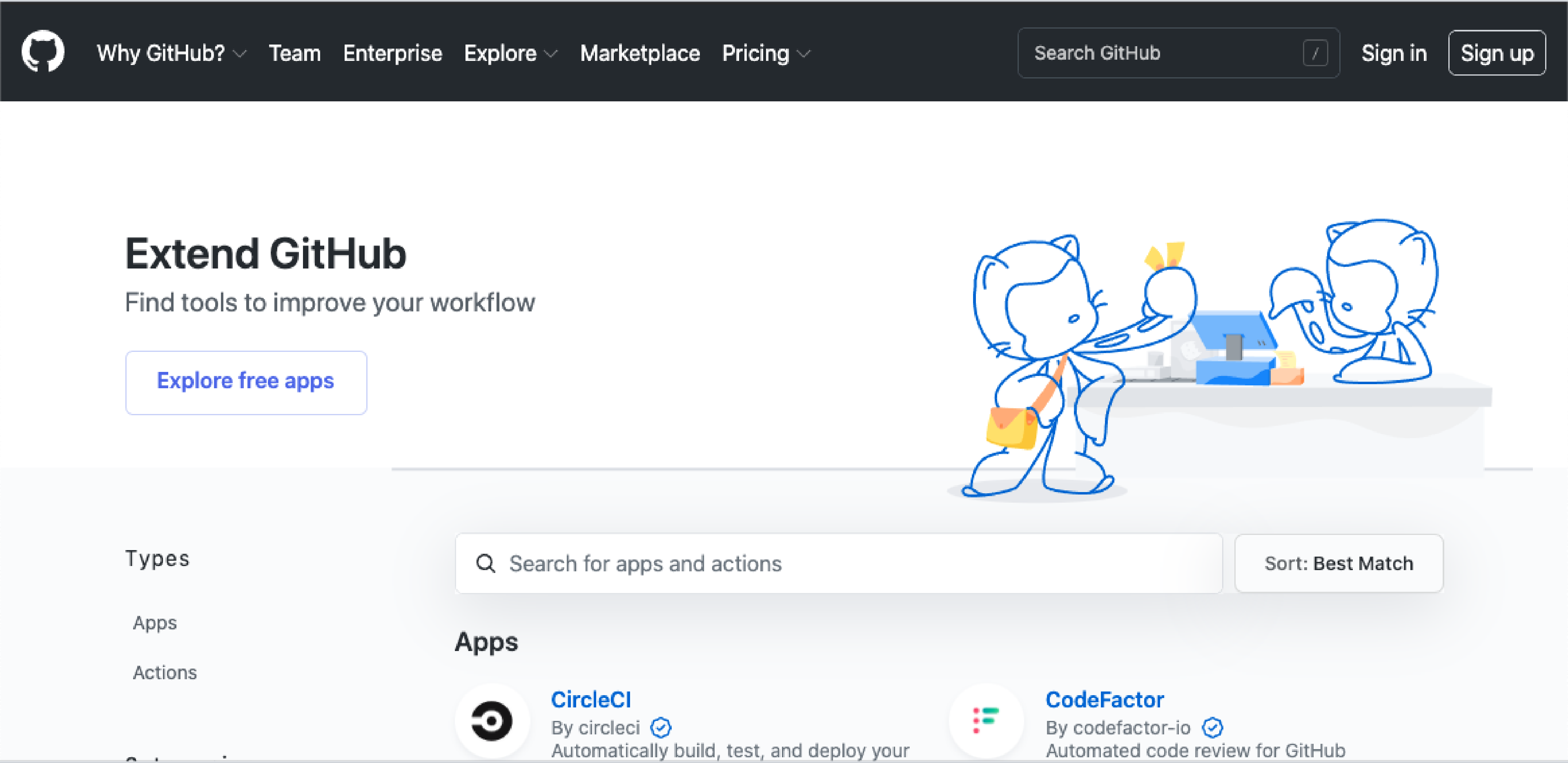
Best practices on Actions in an organization

- Use the `GITHUB_TOKEN` when possible, as a second option GitHub Apps
- **Limit token permissions.** Set organization workflow permissions to read only
- Run only **trusted actions** and provide actions only the secrets that are needed. Pin untrusted actions
- Protect your secrets with **environments**
- Create **starter workflows** for reusability
- Always create meaningful `README` files for your custom actions
- Small and focused actions. Reuse them from the **marketplace**
- Use actions for CI/CD but also ***-ops**

Additional topics

- API Overview
- Authentication methods
- Actions overview
- Marketplace overview

https://github.com/marketplace



Extend GitHub

Find tools to improve your workflow

Types

- Apps
- Actions

Apps

CircleCI
By circleci
Automatically build, test, and deploy your

CodeFactor
By codefactor-io
Automated code review for GitHub

Q&A

Thank you