Microsoft Certified: DevOps Engineer Expert (AZ-400): Security and Compliance

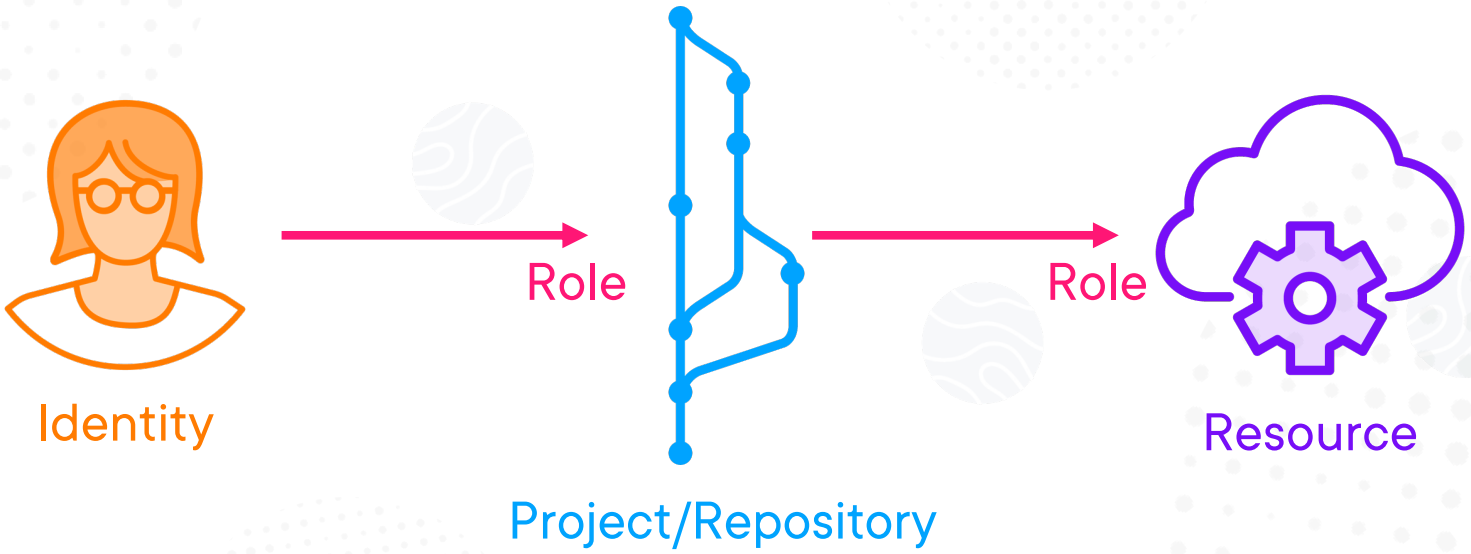
Design and Implement Authentication and Authorization Methods

John Savill
Chief Architect

@ntfaqguy | onboardtoazure.com

Identity Provider Integration for ADO and GitHub

Identity and DevOps



Identity Provider Foundation

Identity Providers (IdP) play a key role in security for IT systems

Cloud IdPs provide services aimed at cloud-based services and protocols

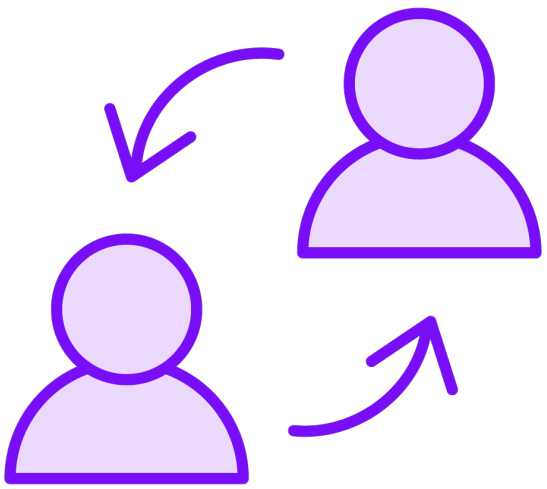
Often a single IdP is used by an organization across all or most of their cloud applications

This provides benefits for the organization and the users

Identity Providers Supported

Azure DevOps	GitHub
Microsoft Entra ID	Microsoft Entra ID
Microsoft Account (MSA)	Active Directory Federation Services (ADFS)
	Okta
	OneLogin
	PingOne
	Shibboleth

Choosing an Identity Provider



If the DevOps solution is for your organization, you should use your organizations IdP

Minimize complexity for users

Your organization needs governance and oversight on all identities that engage with corporate resources including DevOps

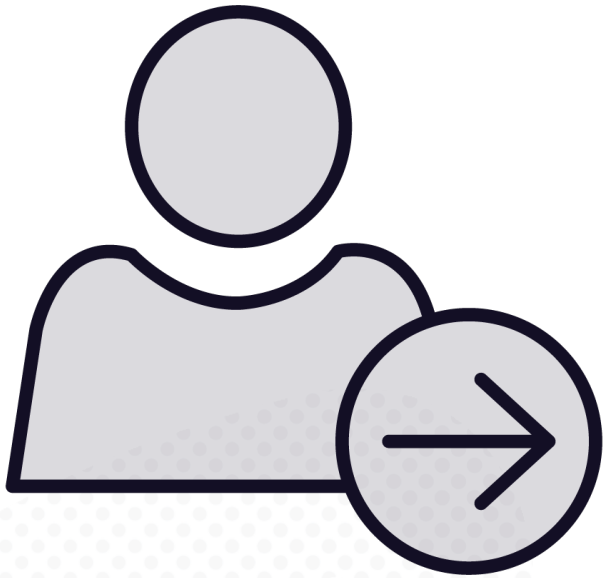
Personal accounts should be avoided

Collaboration with Partners

You will often collaborate with people from other organizations

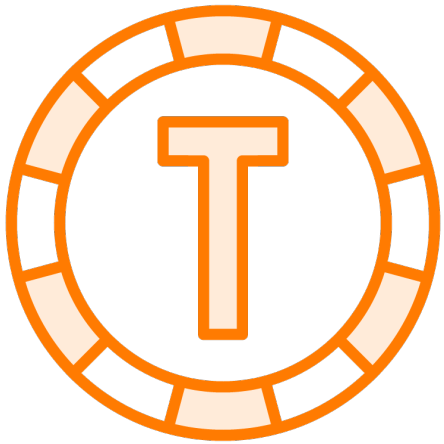
Where possible avoid creating an account in your tenant

Invite their existing account as a guest



Using Personal Access Tokens and Other Auth Methods to ADO and GitHub

Why Do We Need PAT?



Using regular credential with ADO or GitHub may not be desired

You may not be able to use your normal account

You may not wish to expose all your permissions

Benefits of Personal Access Tokens

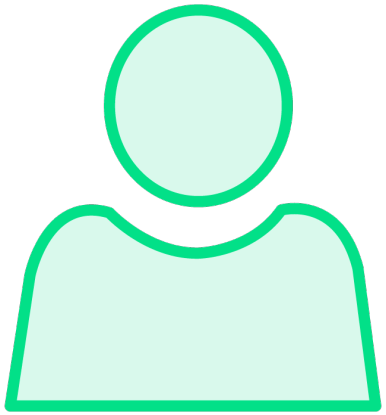
They can be used in place of passwords

They can be constrained to specific permissions and times

They can be revoked

PATs should be safeguarded with the same vigilance as a password

Creating and Revoking PATs



Azure DevOps

- Via User settings – Personal access tokens

GitHub

- Profile – Settings – Developer settings – Personal access tokens – Fine-grained tokens

Using GITHUB_TOKEN

Each workflow job has an automatic GITHUB_TOKEN created

Its permissions are limited to the local repository

It can be used in workflows like any other secret

```
- ${{ secrets.GITHUB_TOKEN }}
```

Design and Implement Permissions and Roles in GitHub

Personal Accounts vs. Organizations

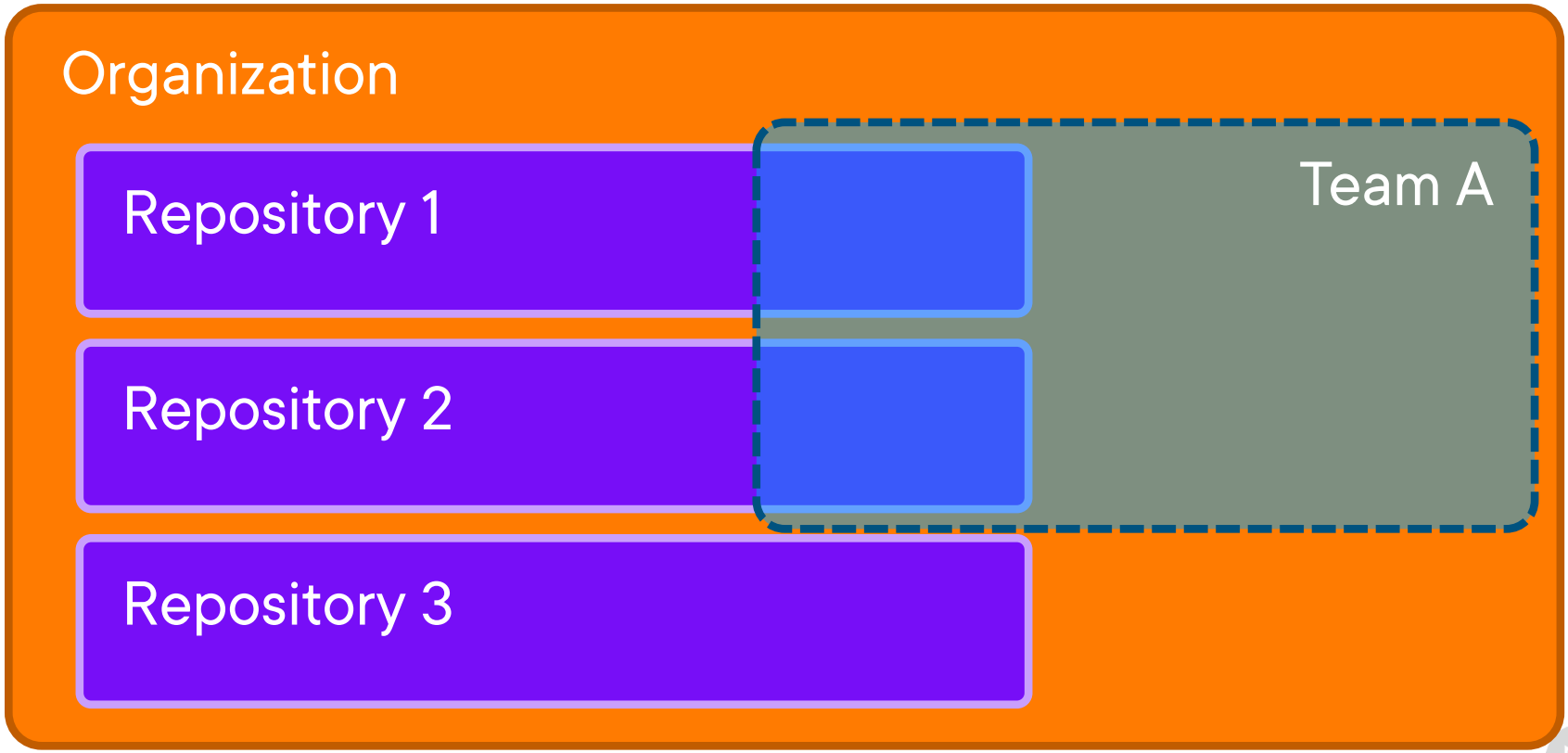
Personal accounts have two roles

Owner and Collaborator

Organizations have many built-in roles

Enterprise Organizations have additional roles and custom repository roles

GitHub Structure



GitHub Organizational Roles

- Organization owners
- Organization members
- Organization moderators
- Billing managers
- Security managers
- GitHub app managers
- Outside collaborators

Organizational Repository Roles

Read	Triage	Write
Maintain	Admin	Custom (Enterprise Only)

Always think least privilege. The lowest role at the lowest scope possible.

Design and Implement Permissions and Security Groups in Azure DevOps

Azure DevOps Entra ID Integration

Azure DevOps was built on Entra ID (fka Azure AD)

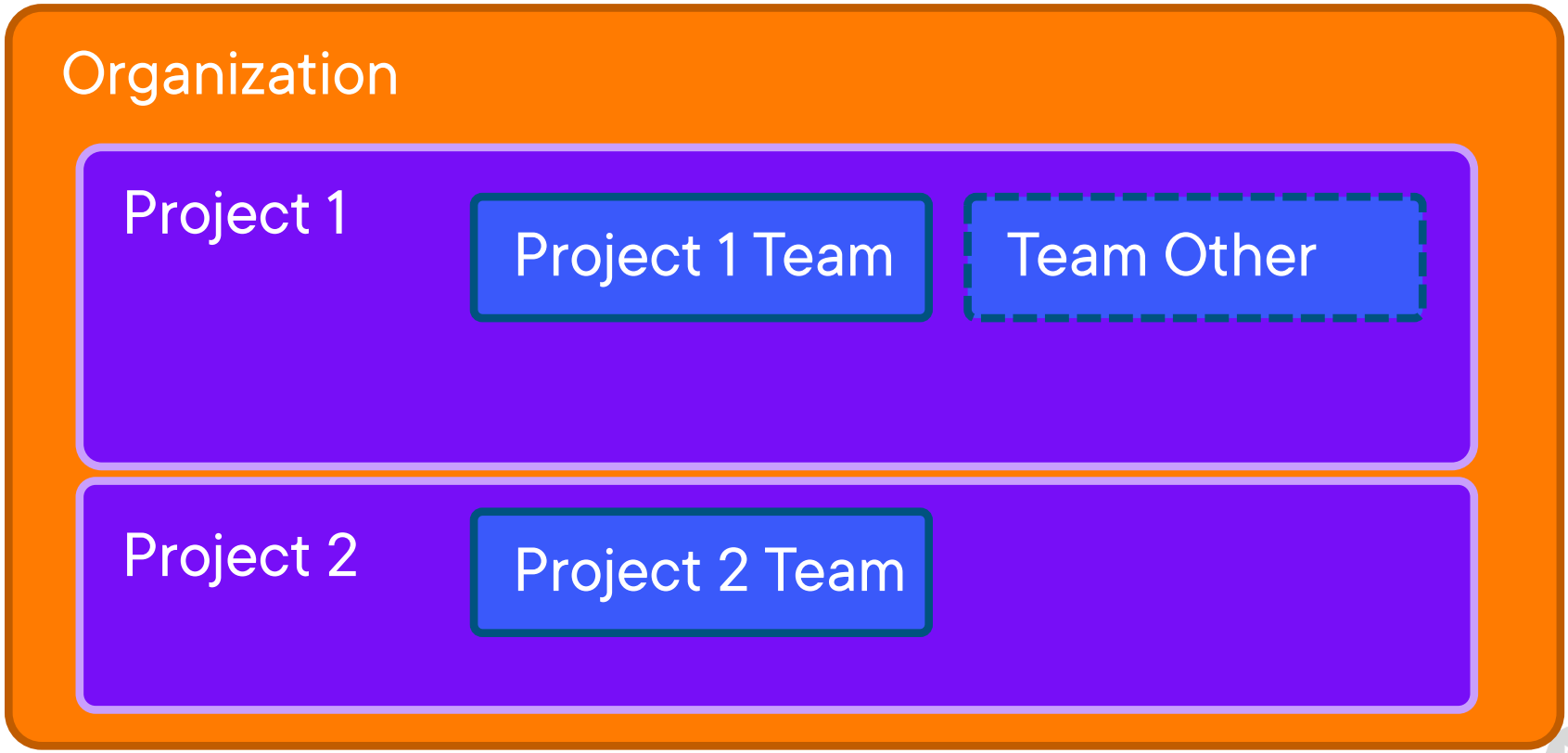
It therefore has support for Entra users and groups

Where possible leverage groups for role assignments instead of users

There are several built-in ADO groups that have specific ADO roles assigned

Add Entra groups to ADO groups

Azure DevOps Structure



Default Azure DevOps Groups

Organization level:

Project Collection Administrators

Project Collection Build Administrators

Security Service Group

Many Service Accounts groups

Project level:

Build Administrators

Contributors

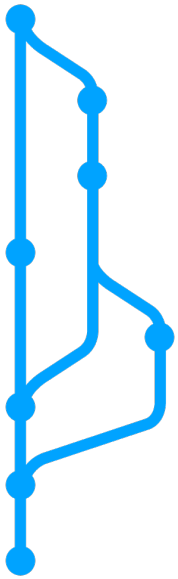
Project Administrators

Readers

...

Configure Projects and Teams in Azure DevOps and GitHub

Why Teams and Projects?



We care about the capabilities of the tools

Repositories, boards, build capabilities etc.

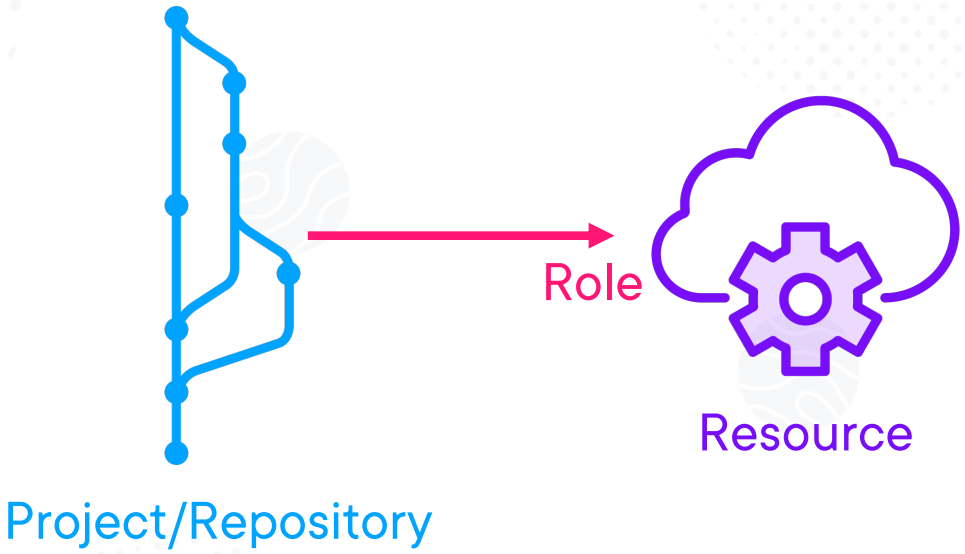
In any medium or larger organization, we will need more than one collection

We therefore group objects to increase usability and simplify permission management

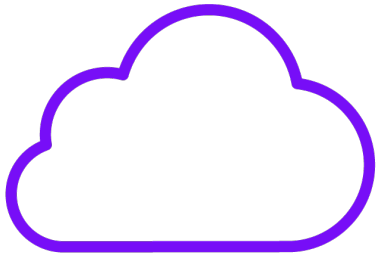
ADO and GitHub use the Projects/Repository and Teams concepts

Review of ADO and GitHub Interaction with Azure and Other Services

DevOps to Resource Authentication



Azure Role Based Access Control



Azure resources live in a subscription

A subscription trusts a specific Entra tenant for all identities

Identities include users, groups and service principals

An identity is granted a role at a specific scope

Authenticating to Azure

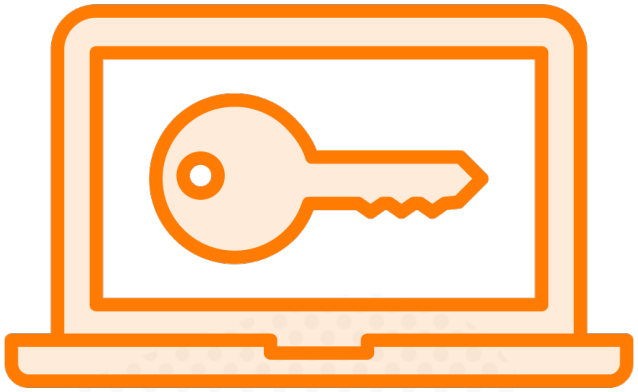
Your pipeline/actions needs an identity to utilize

Historically a service principal is created

The secret for the identity is stored as a secret object within the tooling

Maintaining secrets is never desirable

NEVER store plain text!



Federation Alternative