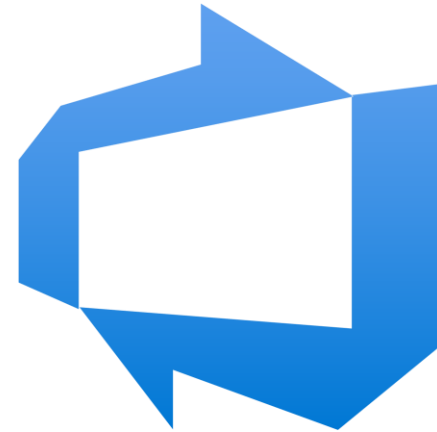


Microsoft Fabric and Azure DevOps – The story so far



Microsoft Fabric



Azure DevOps

Kevin Chant

Managing expectations

- Demos during the session
- All on Trial tenant (no NDA material)
- Includes details about new Microsoft CI/CD workflows document

Agenda

- Bio
- Intro to Microsoft Fabric
- Intro to Azure DevOps
- Configuring Azure Repos for Git integration
- Using Azure DevOps with suggested CI/CD workflow options
- CI/CD for Data Warehouses

Kevin Chant

- Data Engineering Manager in the Netherlands
- Worked in IT since the days of Windows 95
- Experience in various sectors
- Various certifications, Data Platform MVP



- Twitter/Blue Sky: @kevchant
- LI: <https://www.linkedin.com/in/kevin-chant/>
- Blog: <https://www.KevinRChant.com>
- GitHub: <https://github.com/kevchant>





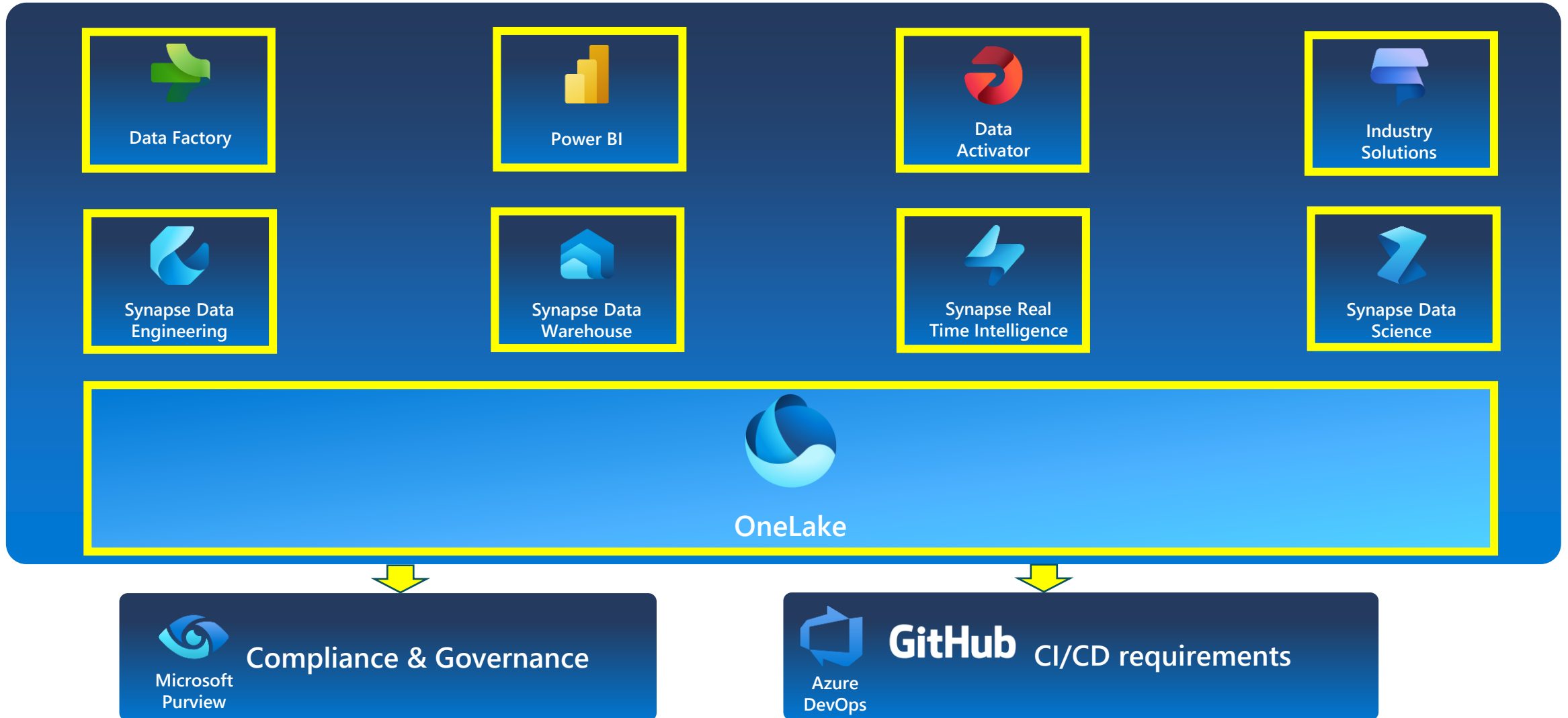
Microsoft Fabric

Comp
analytics
platform

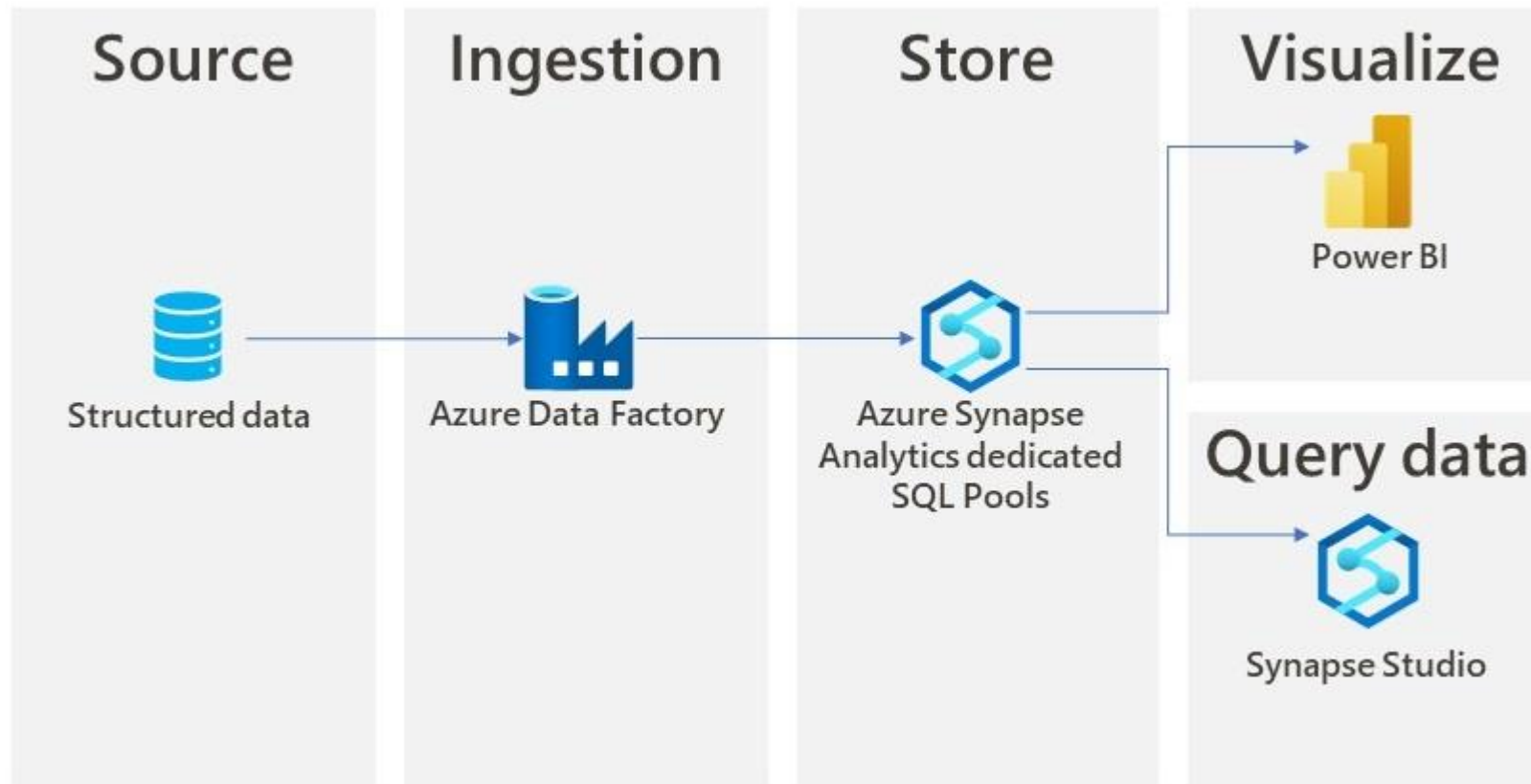
Lake
centric
and open

Empower
every
Business
user

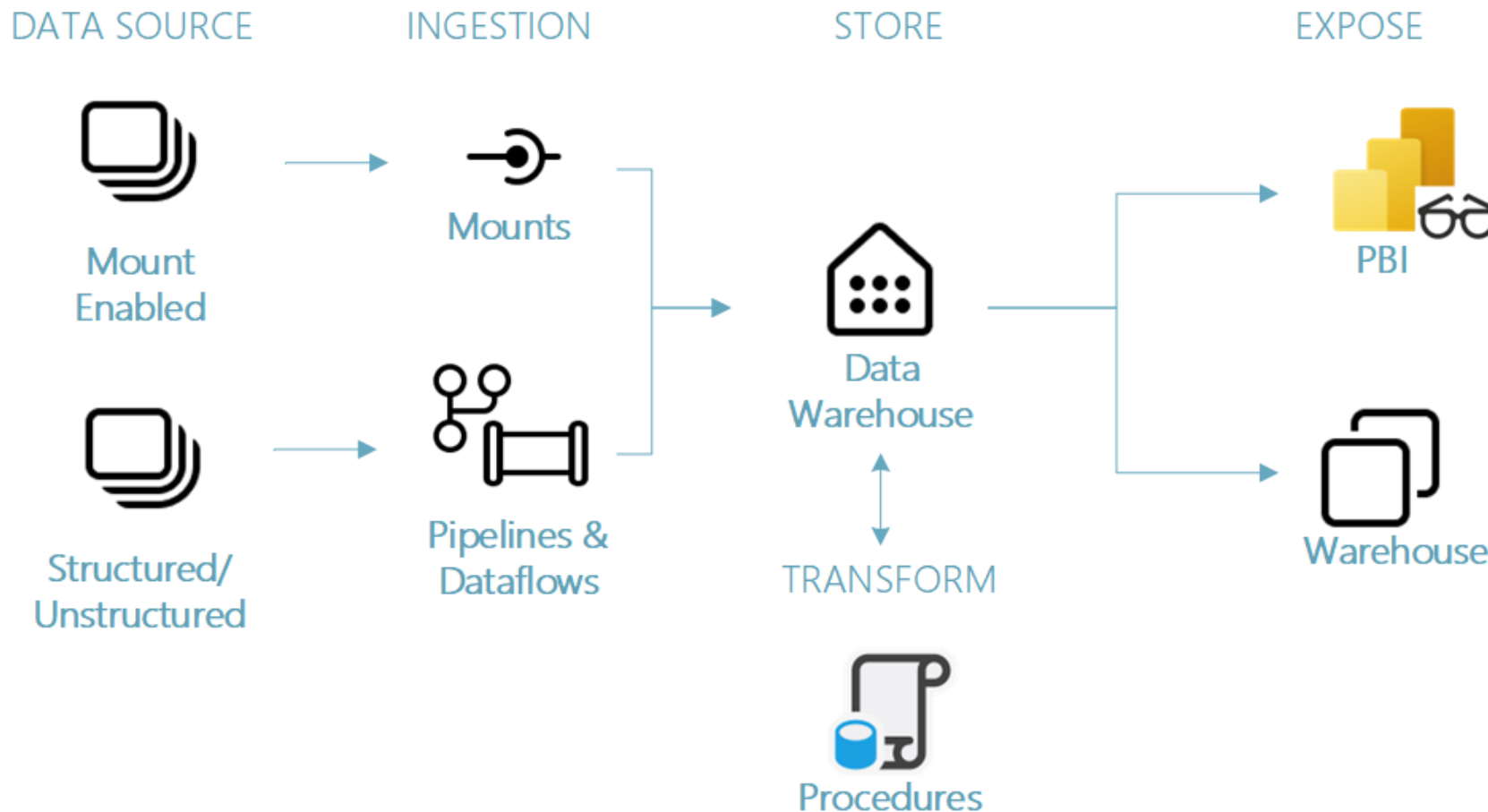
AI Powered



Integrating various services



Within Microsoft Fabric



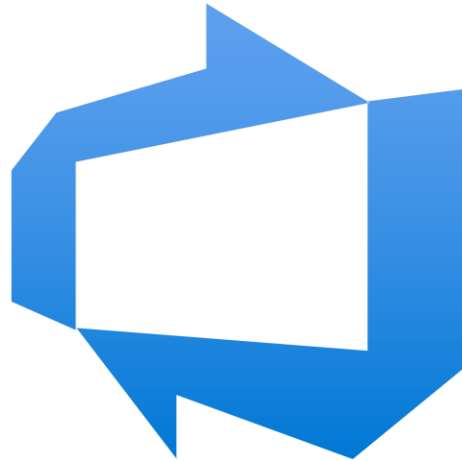
Fabric Demo



Azure DevOps

- Manages Application Lifecycle Management
- Collection of services
- Two main versions

Azure DevOps Demo



About Microsoft Fabric Git Integration

- Allows supported items in a workspace to have metadata synchronized with a Git repository.
 - To be more precise a workspace synchronizes with a branch.
- Items supported at various levels.
- Supports Azure DevOps and GitHub cloud-based offerings.
- Requires Fabric or Power BI Premium capacity.

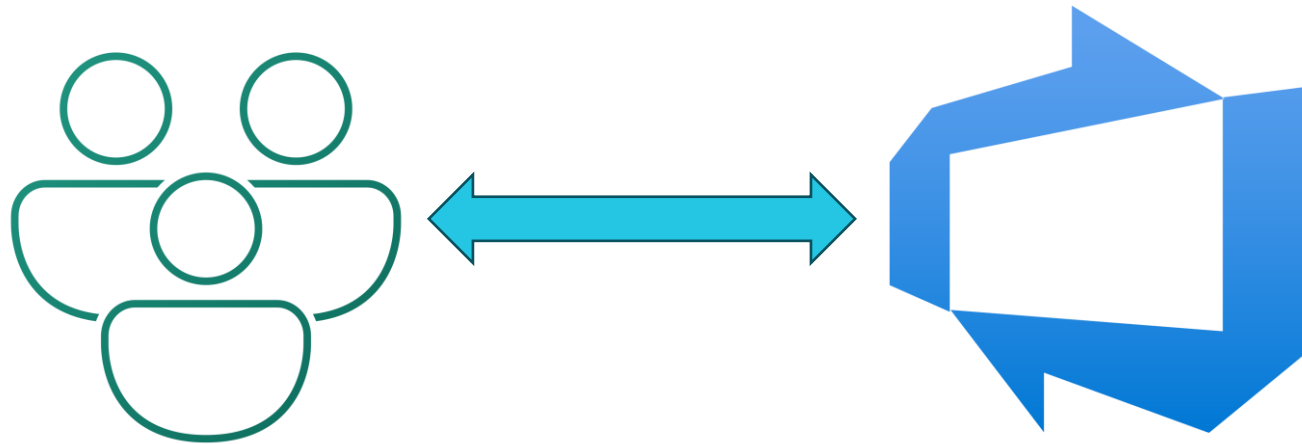
Configuring AzDo for Microsoft Fabric Git Integration

- Utilizes Microsoft Entra ID authentication.
- Both workspace and repository require access.
- Entra user needs to be on same tenant, Azure DevOps org **does not!**

Security considerations

- Keep organization & repository in relevant region.
- Keep organization private.
- Consider GitHub Advanced Security for Azure DevOps.

Git integration Demo



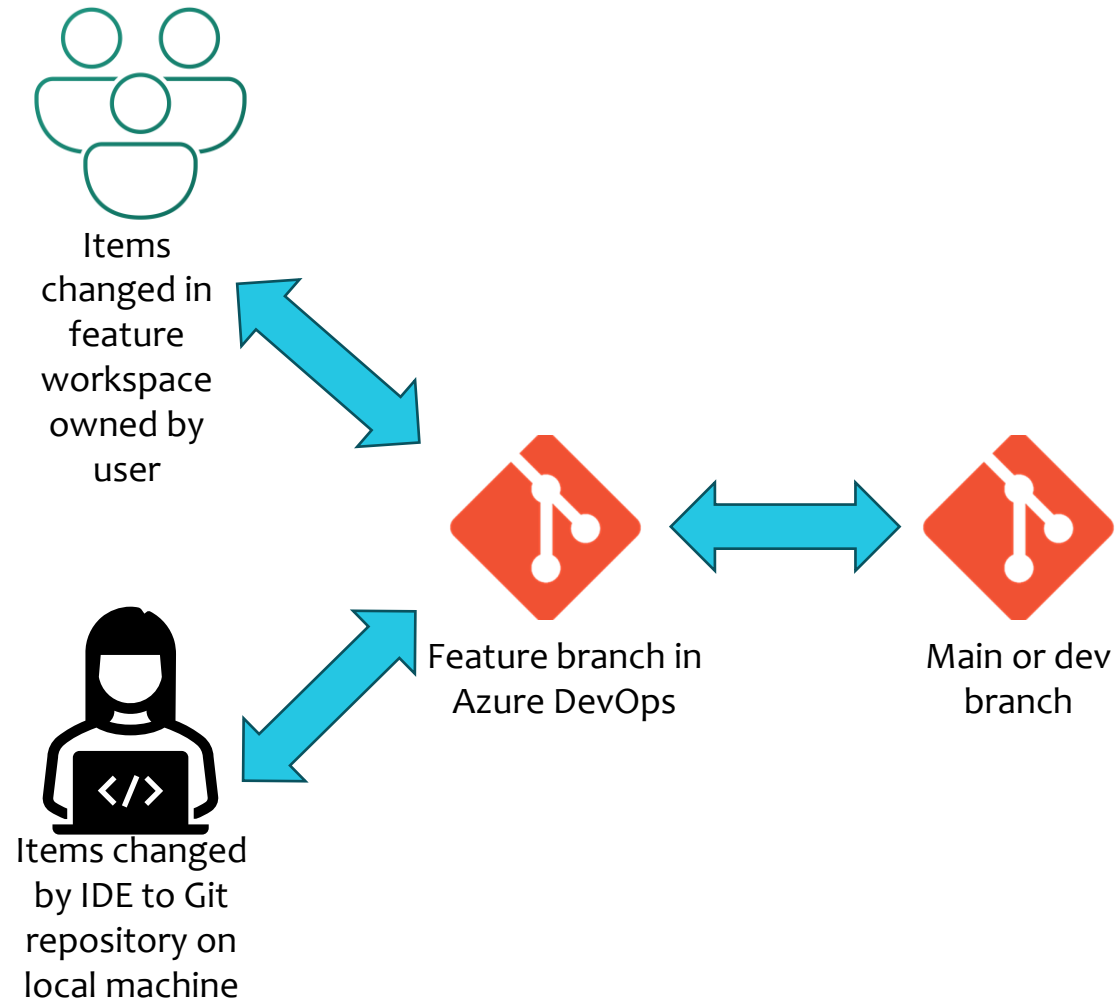
Microsoft Fabric workspace

Azure DevOps

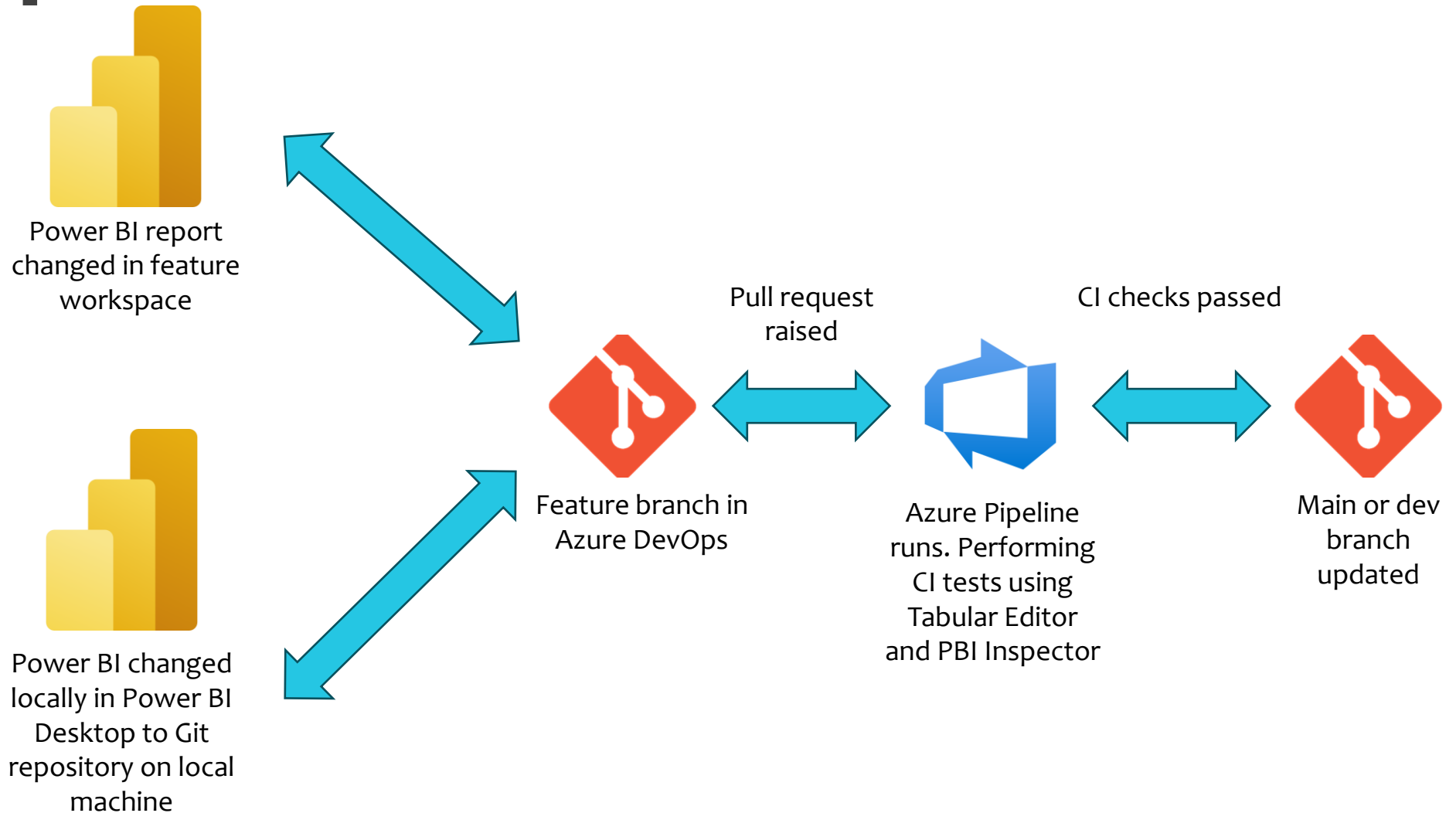
Using Azure DevOps with suggested CI/CD workflow options

- Microsoft released document last month.
- Highlight how Azure DevOps fits into suggested CI/CD workflows.

Recommended development process



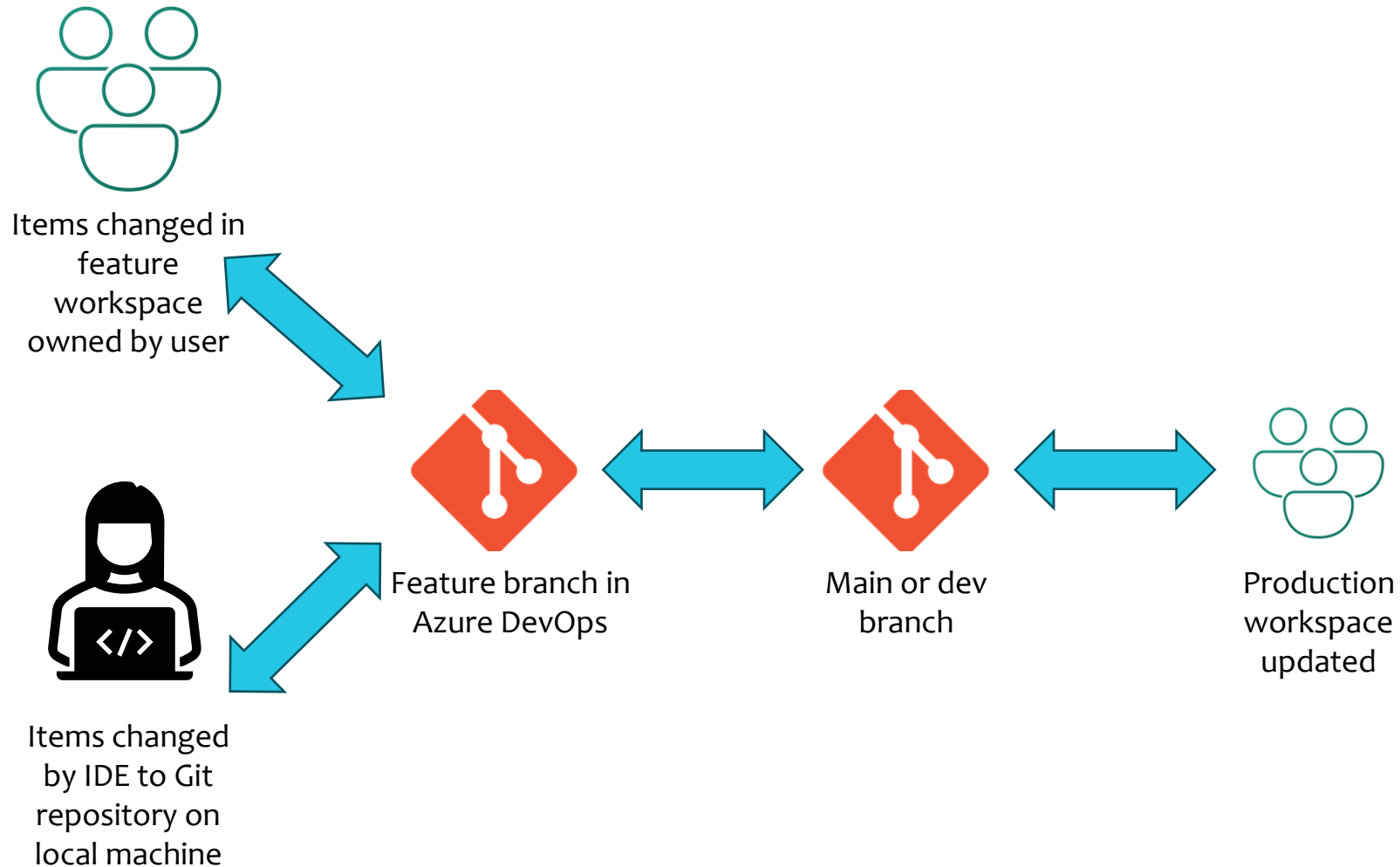
One step further for Power BI reports



Release Option 1 – Git-based deployments

- Deploy to multiple workspaces connected to the same Git repository.
- Achieved with different branches.

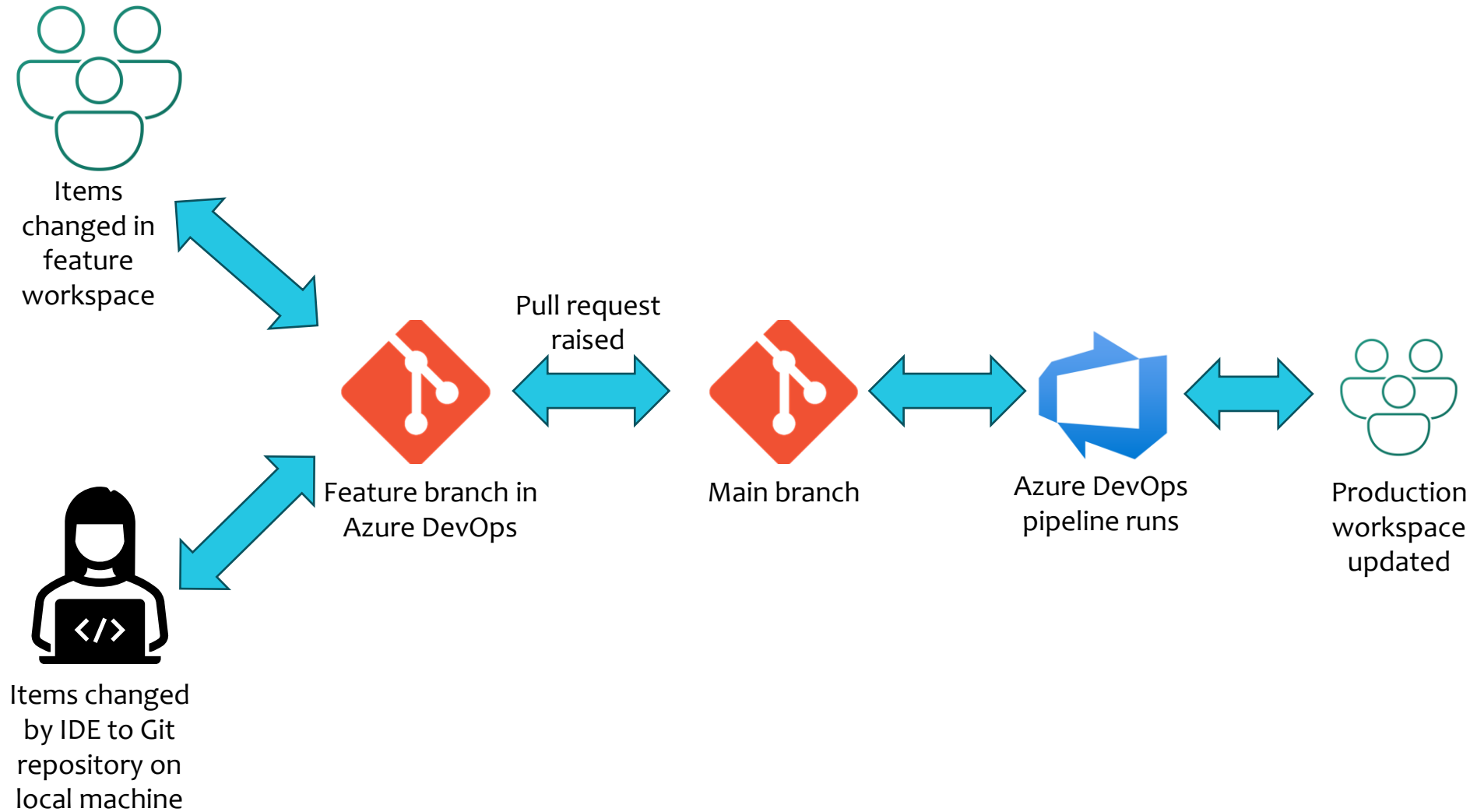
Release Option 1 - Diagram



Release Option 2 – Git-based deployments using build pipeline

- Deploy to different workspaces using Azure Pipelines.
- Recommendation is that each workflow contains a build and release process.
 - Build for unit tests.
 - Release to perform update.
- Note that for APIs only Power BI items are supported by service principals at this moment in time

Release Option 2 - Diagram



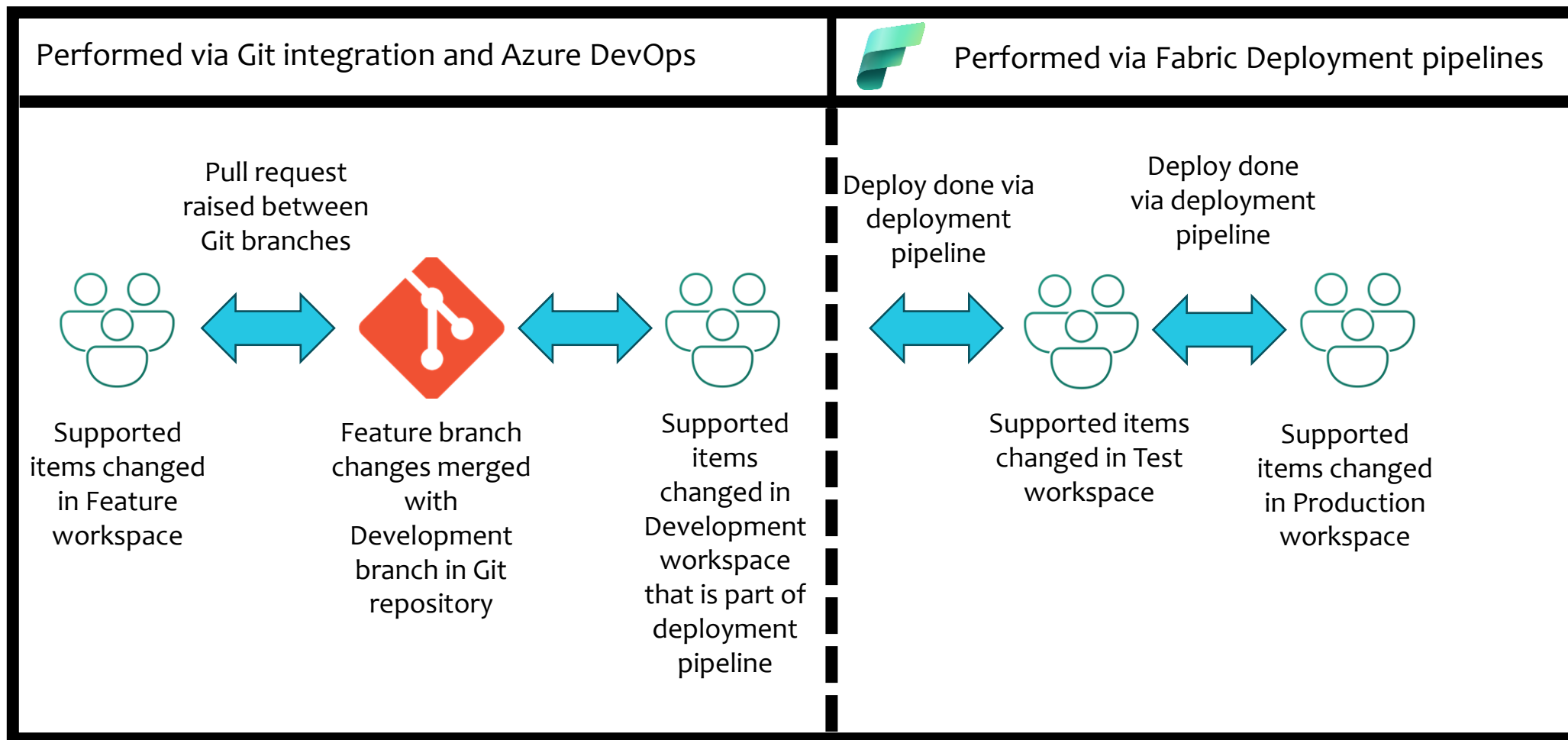
Azure Pipeline recommendations

- Consider YAML Pipelines.
 - You cannot do a PR on a classic pipeline!
- Self-hosted agents to keep workload secure.
- Use variable groups & Azure Key Vault
- Implement approvals process for production workloads via environments.

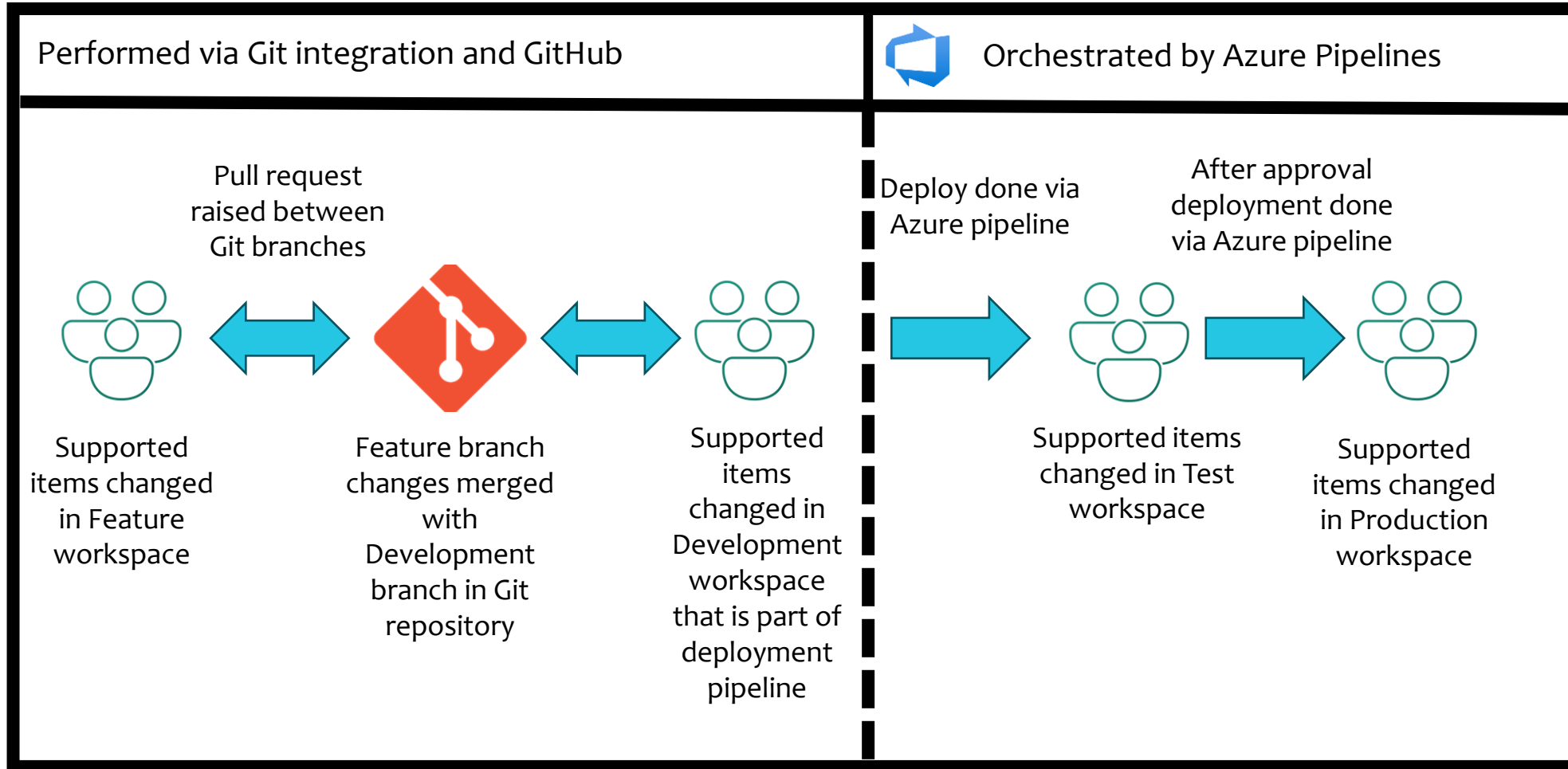
Release Option 3 – Deploy using Microsoft Fabric deployment pipelines

- Perform your pull request from feature branch to branch connected to a workspace that represents Dev stage of a deployment pipeline.
- From there orchestrate using Microsoft Fabric deployment pipelines.
- Alternatively, orchestrate to different Microsoft Fabric deployment pipeline stages using Azure DevOps.

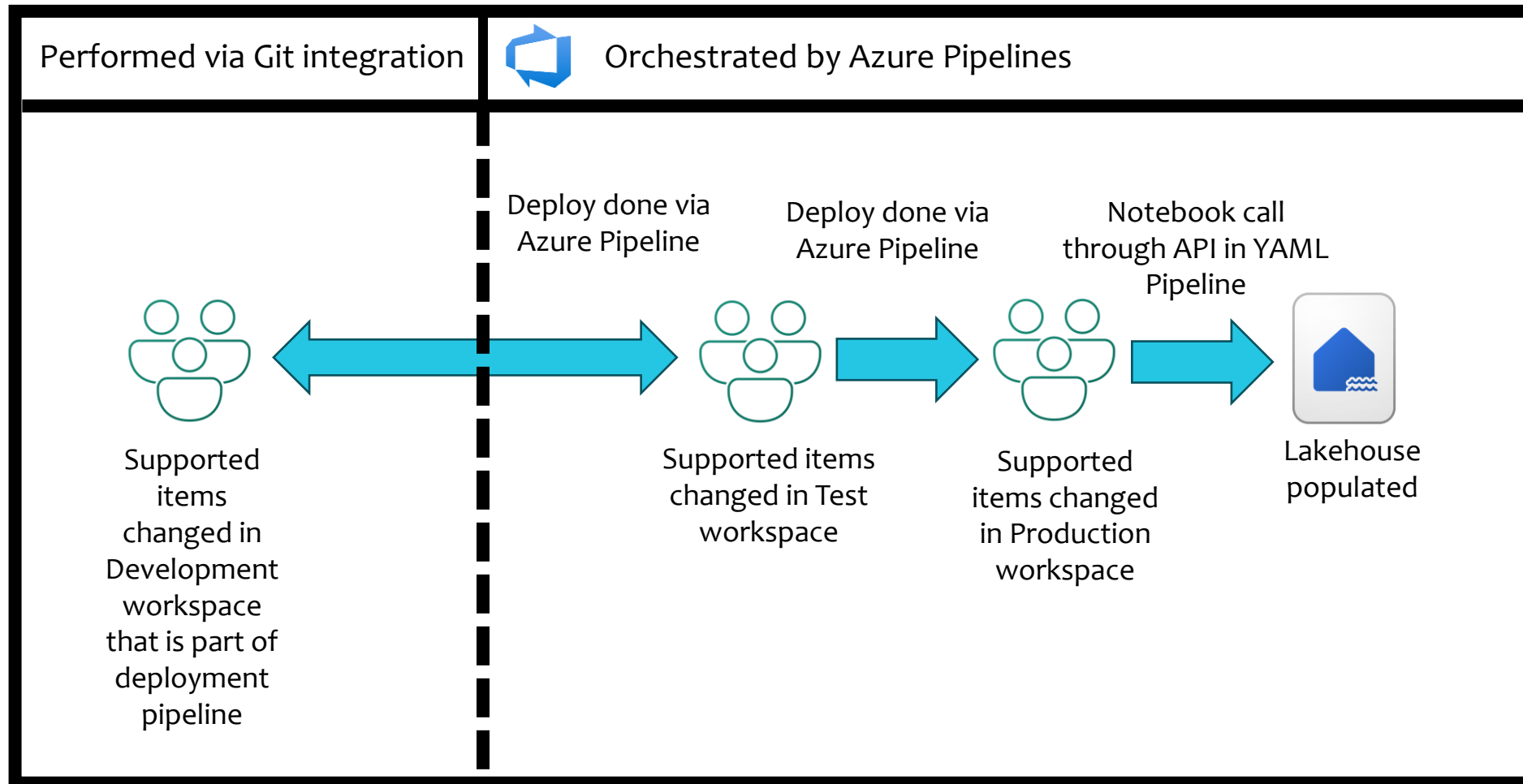
Release Option 3 - Diagram



Orchestrated by Azure Pipelines



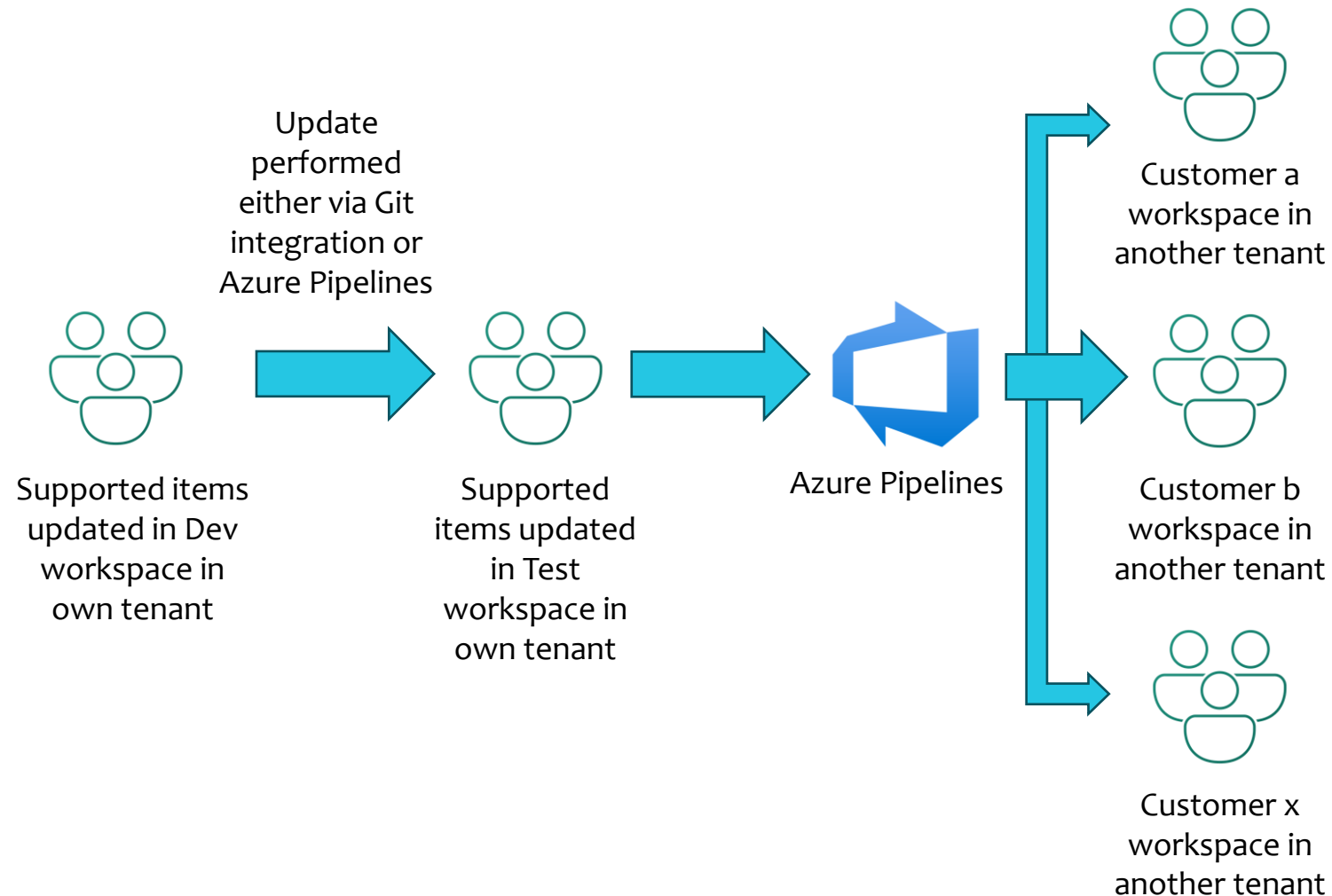
Advantage of Azure Pipelines



Release Option 4 – For multiple customers/solutions/tenants

- Dev and test stages are managed in same Fabric tenant.
- Deployment to Prod stages to workspaces in other tenants using Azure Pipelines

Release Option 4 - Diagram



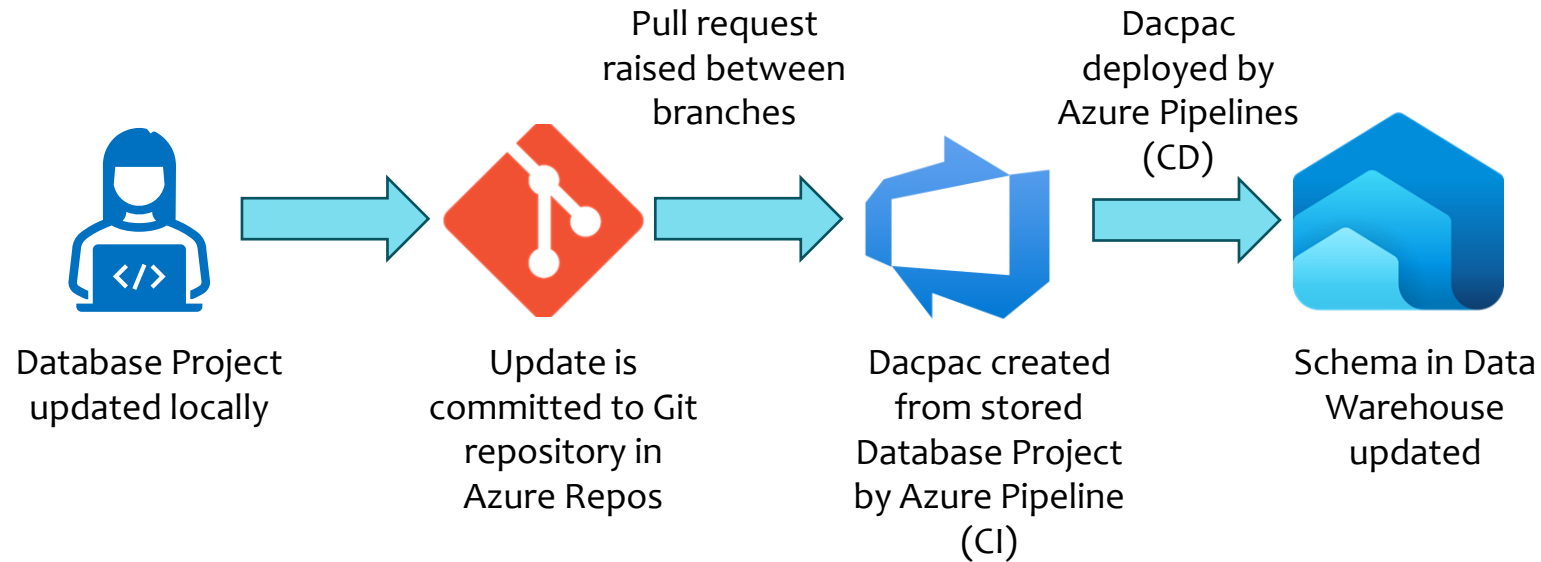
Demos

- Power BI Desktop Projects
- Deploying to multiple workspaces via Git integration
- Deploying via deployment pipelines

CI/CD methods for Data Warehouses

- Can connect to Data Warehouse via connection string
- Allows deployment through traditional CI/CD methods.
 - State-based (dacpac)
 - Migration-based
- Supports Database Projects created in number of places.
- Can be deployed using an Azure DevOps pipeline.

CI/CD for Data Warehouses



CI/CD for Data Warehouse Demo



Questions



Thank you



- Twitter/Bluesky: @kevchant
- LI: <https://www.linkedin.com/in/kevin-chant/>
- Blog: <https://www.KevinRChant.com>
- GitHub: <https://github.com/kevchant>

Links shared

- [Thoughts about disabling classic pipelines in Azure DevOps](#)
- [Introduction to Git integration](#)
- [Power BI Desktop projects](#)
- [Power BI Project \(PBIP\) and Azure DevOps CI performance tests](#)

Additional links shared

- [Working with Microsoft Fabric Git integration and multiple workspaces](#)
- [Initial tests to copy a Direct Lake semantic model to another workspace using Microsoft Fabric Git integration](#)
- [Introduction to deployment pipelines](#)
- [CI/CD for Microsoft Fabric Data Warehouses using Azure DevOps](#)