PBIG
POWER BI GEBRUIKERSGROEP

Applying DevOps practices to your Power BI deployments in Microsoft Fabric

Kevin Chant

A big thank you to our amazing partners

Thanks

Agenda

- Bio
- DevOps introduction
- Power BI Desktop projects
- Microsoft Fabric Git integration
- Suggested release/deployment options

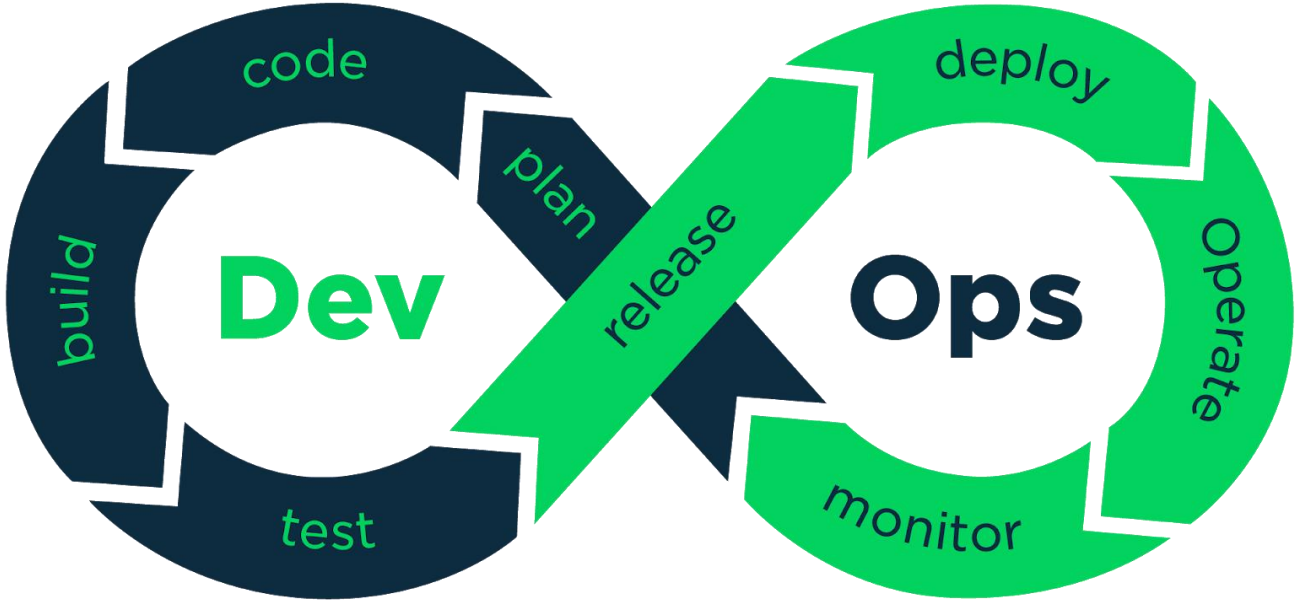
Kevin Chant

- Lead Technology Advocate 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

DevOps introduction

- DevOps is practice that that combines **Dev**elopment and **Op**erational processes together to improve the development lifecycle.



Terminology used

- CI/CD stands for Continuous Integration/Continuous Deployment.
- Git repository is a folder/directory where you store code/files you work with. It contains a hidden database in a .Git subfolder.
- Branches allow you to work with files in a new area in Git repository. It seems like a copy, but actually it is pointers.
- More covered during the session.

Advantages of DevOps

- Audit trail of changes
- Faster updates
- Reliable and more consistent deployments
- Introduces new industry innovations

Version control

- Version control allows you to track and manage changes to files.
- Reduces conflicts and allows history of changes.
- Essential when looking to perform CI/CD with Fabric items.
- Power BI reports are no exception. However, in past has been complex.

Power BI Desktop projects

- Still in preview.
- Stores metadata in various files.
- Allows easy integration with version control.
- Supports two newer file formats:
  - Tabular Model Definition Language (TMDL)
  - Power BI Enhanced Report Format (PBIR)

TMDL

- New(ish) file format for the semantic model metadata.
- Aim is to replace ‘Model.bim’ file.
- Uses a YAML-like syntax for easier reading.
- Breaks down into each object into a separate file.
- Integrates better with version control.

PBIR

- New(ish) format for report metadata.
- Uses a JSON syntax.
- Breaks down report items into a separate files.
- Integrates better with version control.
- Allows quick external editing in other tools.

Ways to create Power BI Projects

- Within Power BI Desktop
- When creating in a workspace with Git integration configured.
- Demo

Microsoft Fabric Git integration

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

Currently supported items

- Data pipelines (preview)
- Dataflows gen2 (preview)
- Eventhouse and KQL database (preview)
- EventStream (preview)
- Lakehouse (preview)
- Mirrored database (preview)
- Notebooks
- Paginated reports (preview)
- Reflex (preview)
- Warehouses (preview)
- Reports (except reports connected to semantic models hosted in Azure Analysis Services, SQL Server Analysis Services, or reports exported by Power BI Desktop that depend on semantic models hosted in MyWorkspace) (preview)
- Semantic models (except push datasets, live connections to Analysis Services, model v1) (preview)
- Spark Job Definitions (preview)
- Spark environment (preview)
- SQL database (preview)

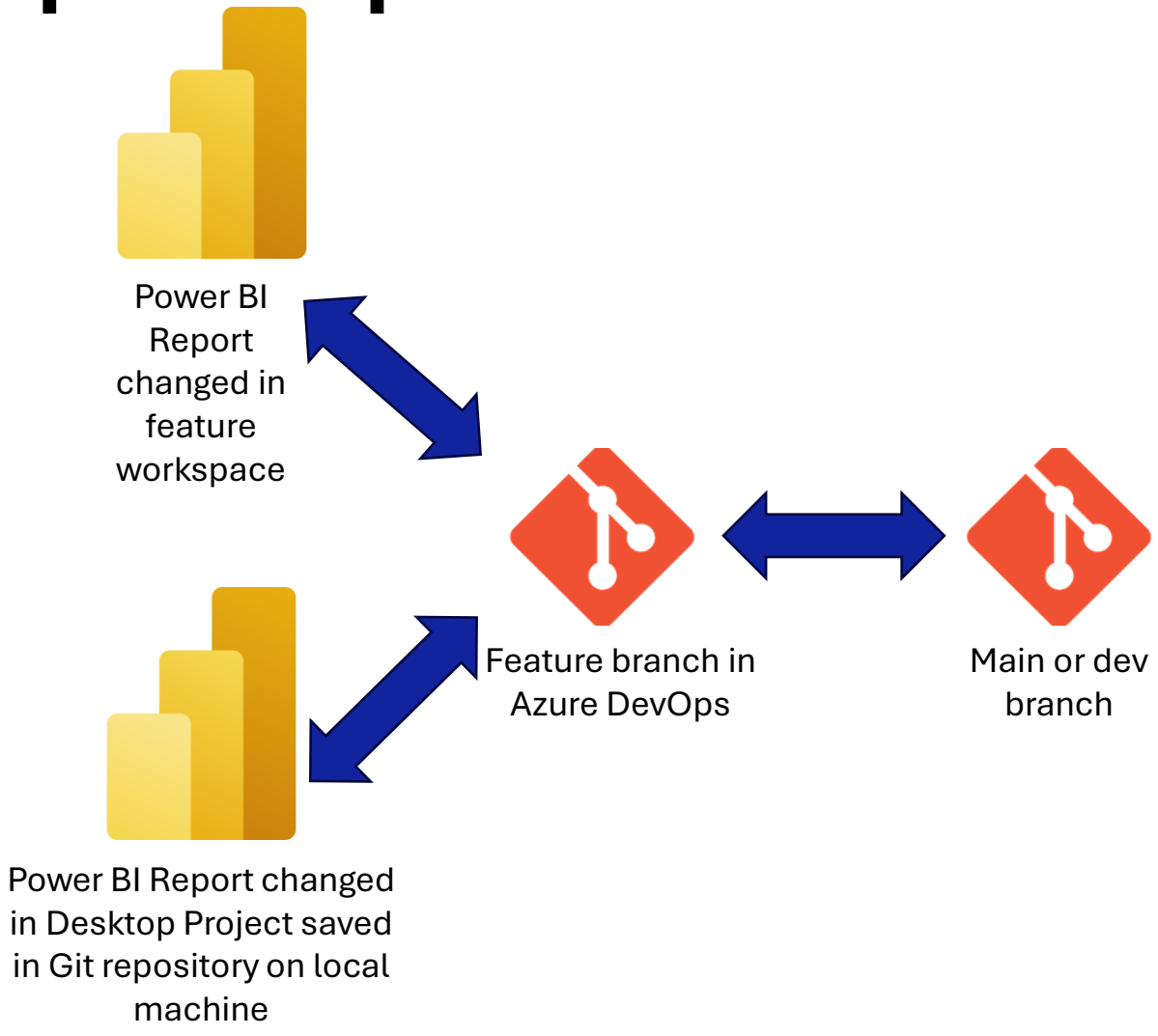
Git integration features

- Can change items in workspace and commit to repository.
- Can also update from repository.
- Can branch out to new workspace to create “feature” workspaces.

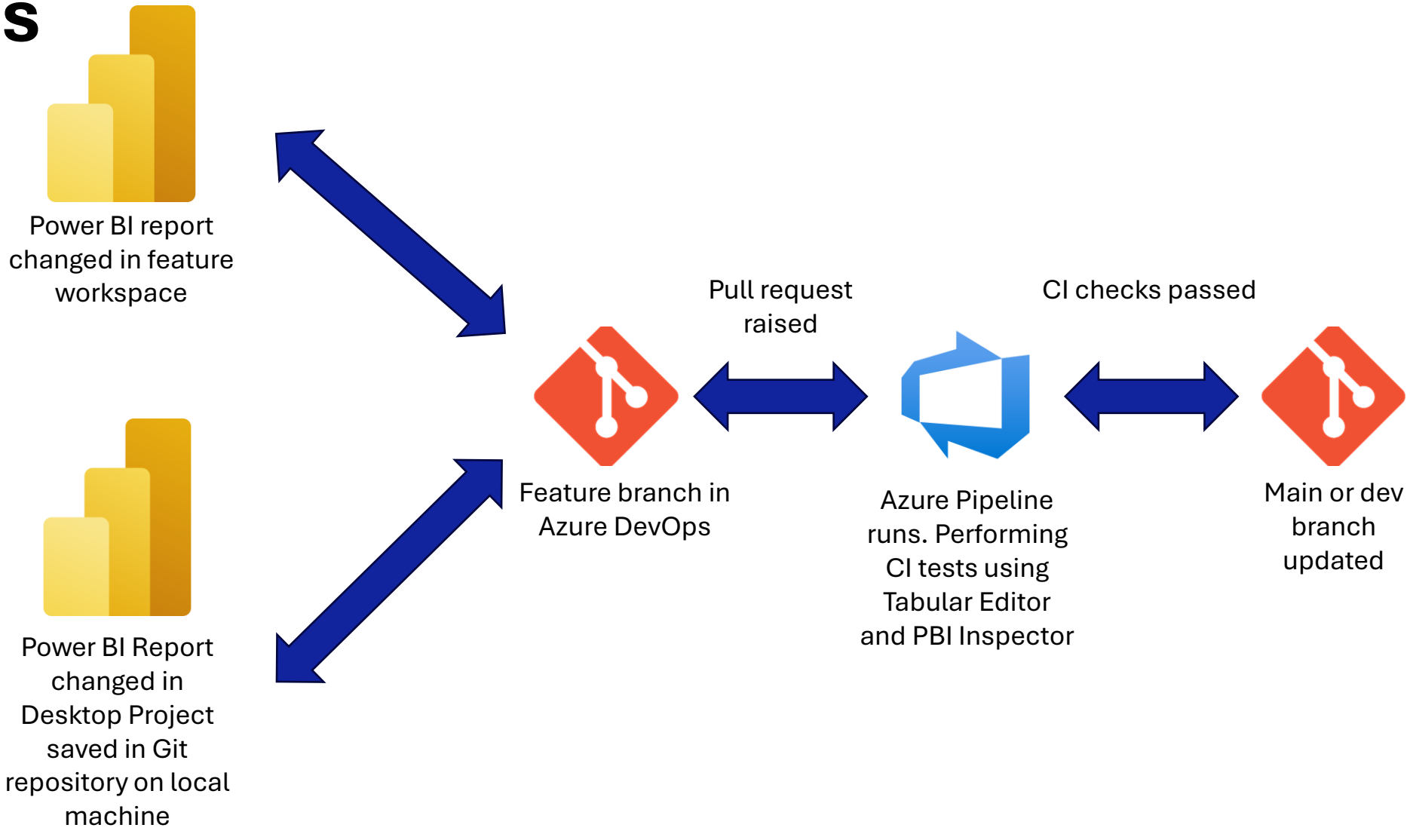
Feature workspaces

- Workspace that represents a specific feature.
- Can be achieved with “branch out to new workspace” functionality.
- However, this functionality requires permissions to create workspaces.
- Alternative is to create separate workspace for each developer.

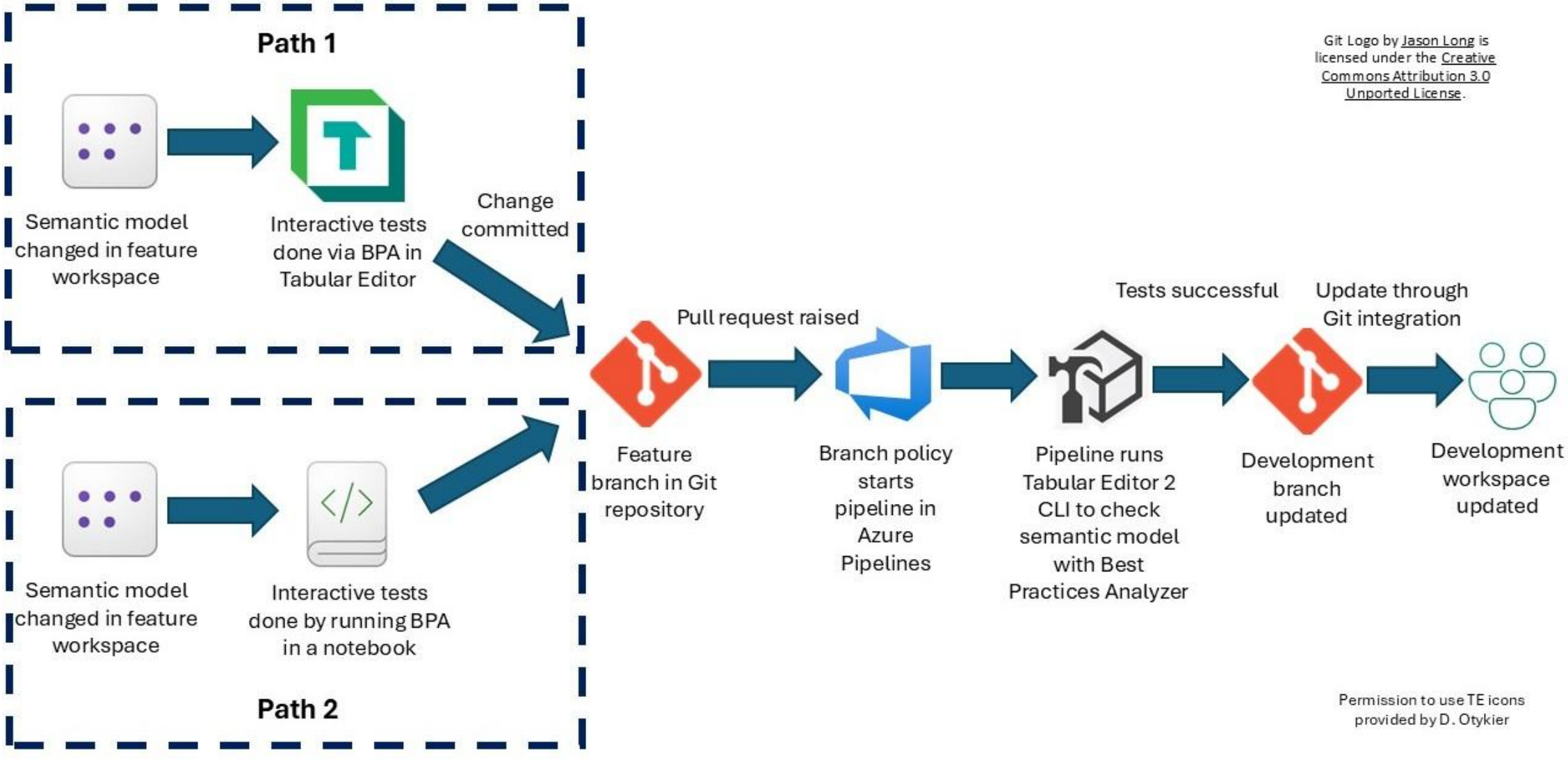
How it aligns with recommended development process for Fabric



For true CI I recommend introducing unit tests



Happy paths to test semantic models



Git integration demos

- Microsoft Fabric Git integration.
- Continuous integration tests.
- BPA bulk

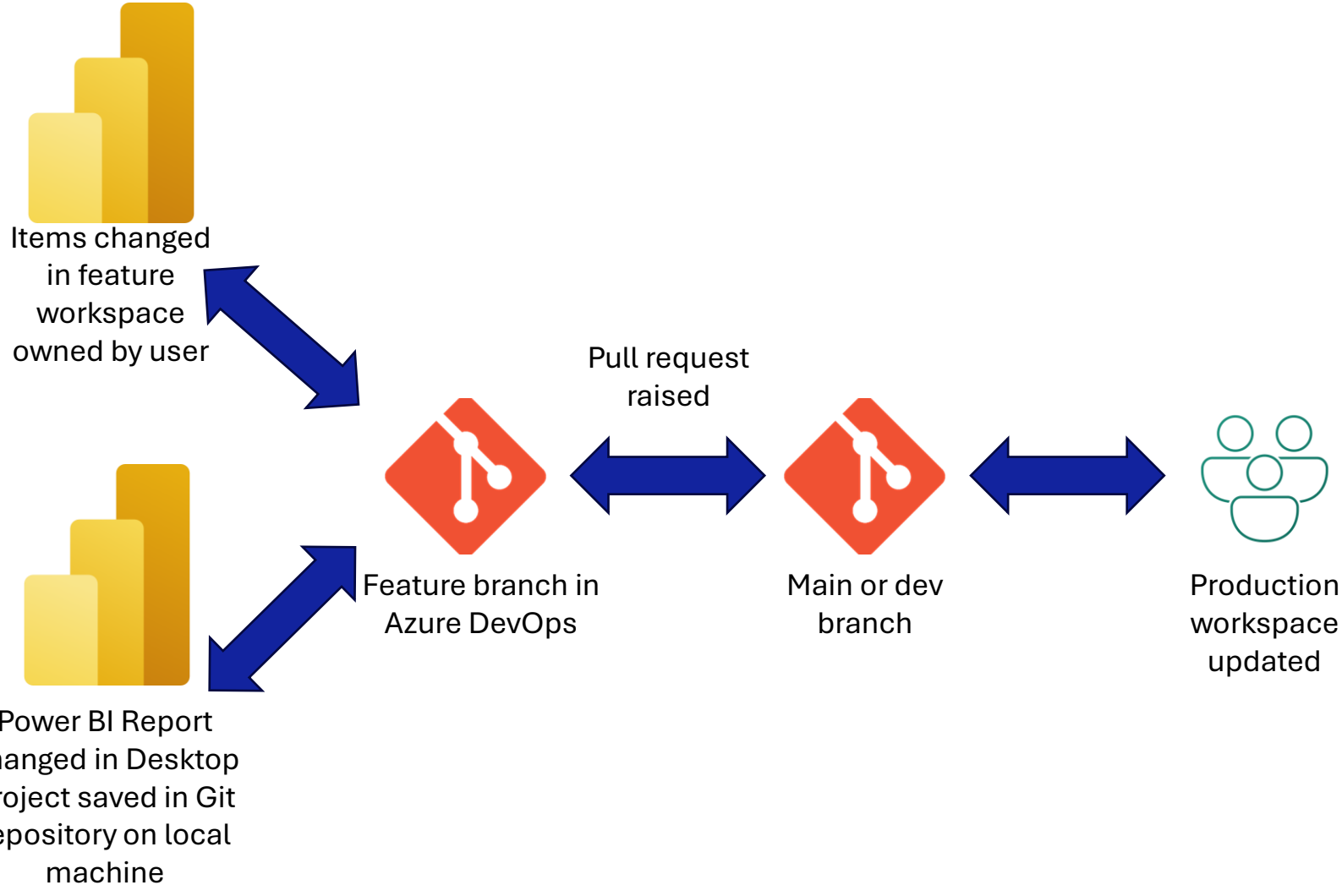
Recommended release options

- Microsoft released article last September.
- Includes previously shown recommended development process (CI) and release options (CD)
- I provide additional guidance and advice in context of Power BI.

Option 1 – Git-based deployments

- Have workspaces connected to different branches in the same Git repository.
- Update different workspaces via pull requests.
- Can be done with Fabric Git APIs supplemented by other APIs.
- Ideal for scenarios that require only minor source changes.
- Requires Git knowledge.

Option 1 diagram



Option 2 – Git-based deployments using build pipeline

- Deploy to different workspaces through a suitable pipeline deployment service (Azure Pipelines, GitHub Actions). Typically with APIs.

- Recommendation is that workflow for each stage contains:
  - Build for unit tests
  - Release to perform update

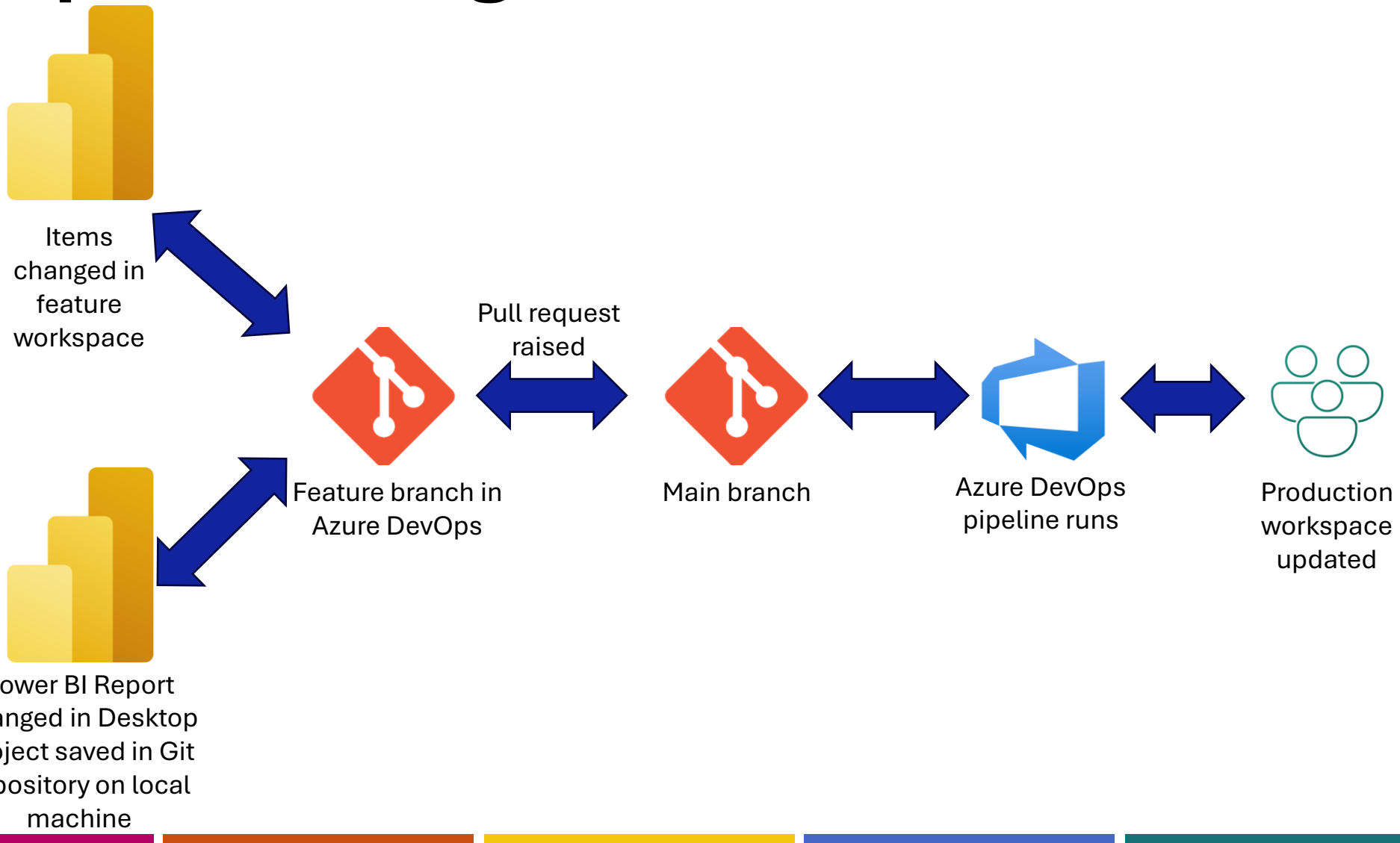
- Ideal for more complex scenarios and when you implement DataOps.

- Knowledge of ALM service required (Azure DevOps, GitHub)

Current Git-based deployment methods

- Direct REST API calls.
- PowerShell modules.
- Fabric-cicd Python library

Option 2 diagram



Advice for option 2 pipelines

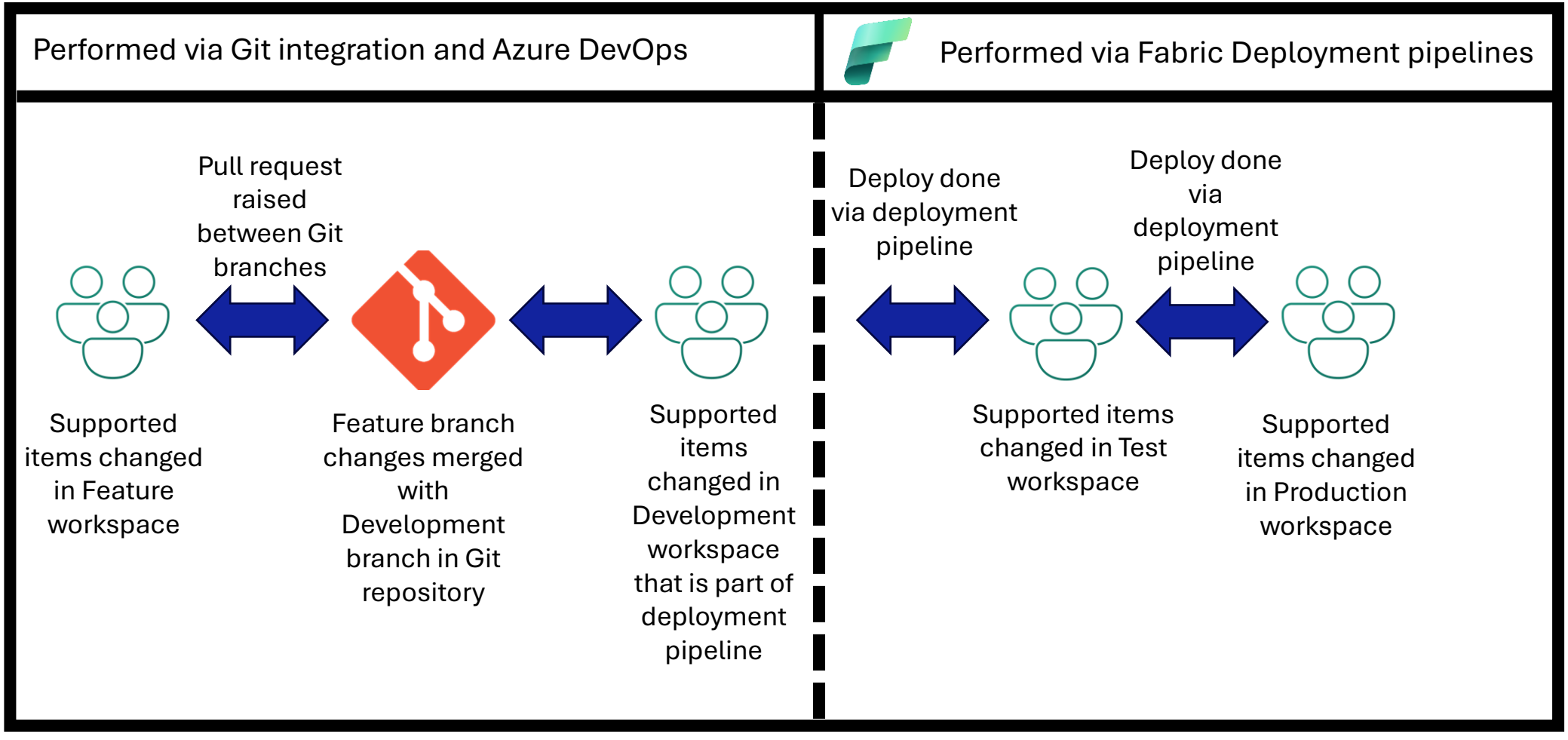
- Consider YAML pipelines for portability.
- Avoid hard-coding sensitive values. Either use the ALM offerings secrets store (variable groups, secrets) or Azure Key Vault.
- Strive to implement approvals process for production workloads.

Option 3 – Microsoft Fabric deployment pipelines

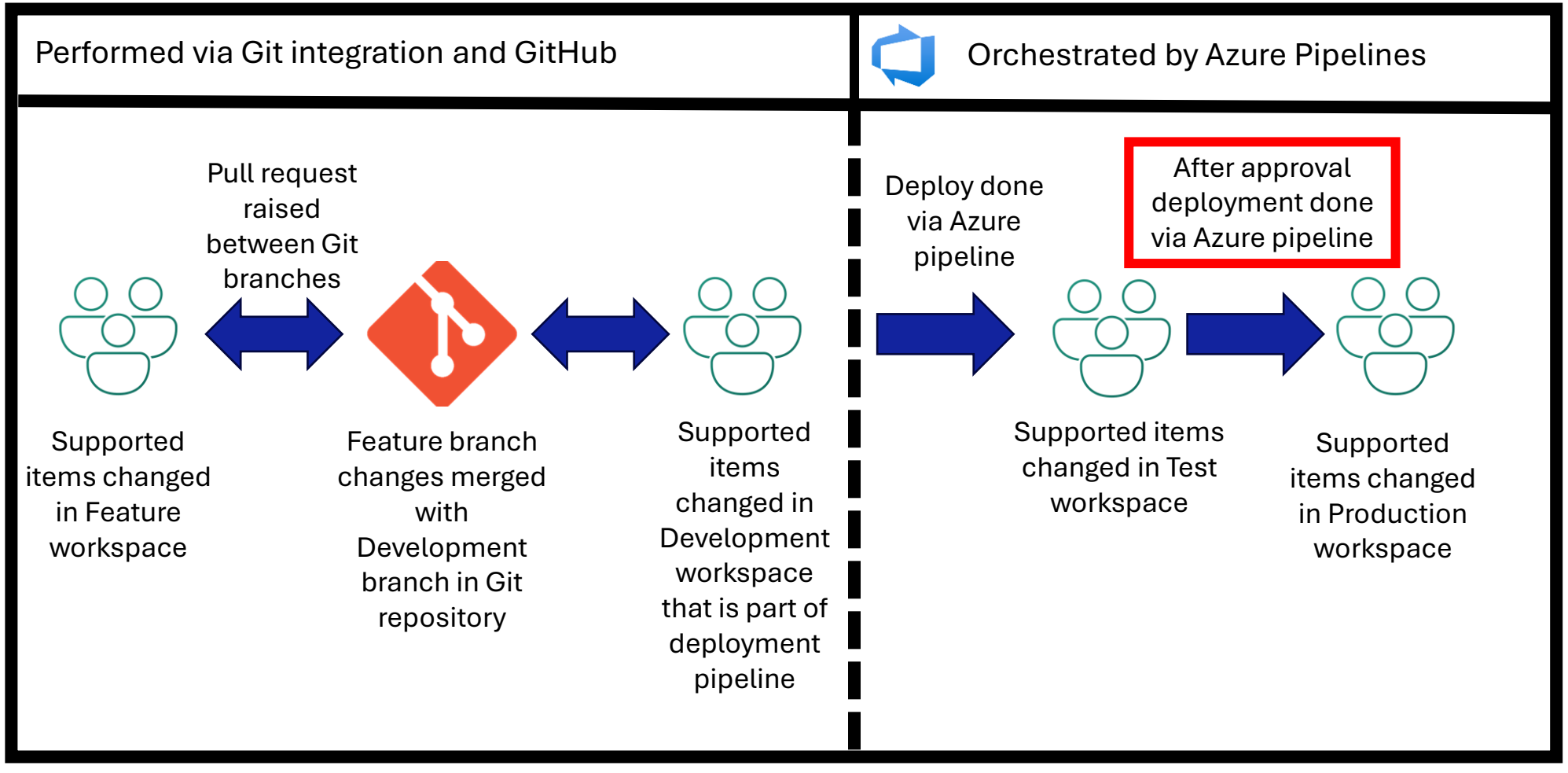
- Use development process to update a workspace that represents development environment/stage.

- From there, orchestrate with Microsoft Fabric deployment pipelines.

Option 3 with deployment pipelines



Deployment pipelines orchestrated by Azure Pipelines



Option 4 – For multiple clients/solutions/tenants

- Similar to option 2.
- All development and test done in same tenant.
- Then deployed to workspaces in other tenants via pipelines.
- Aimed mostly at ISV’s. However, useful for custom reports as well.

Demos

- Option 2 - Fabric-cicd

- Option 3 - Deployment pipelines.

To summarize

- Applying DevOps processes can lead to fast and reliable deployments.
- Testing is essential, ideally at a very early stage.
- Recommend adopting Power BI Desktop projects.
- Experiment to see what release option suits your needs.

PBIG
POWER BI GEBRUIKERSGROEP

Session Feedback

Event Feedback

Thank you

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