Deploy Power BI as Code.

Paulien van Eijk

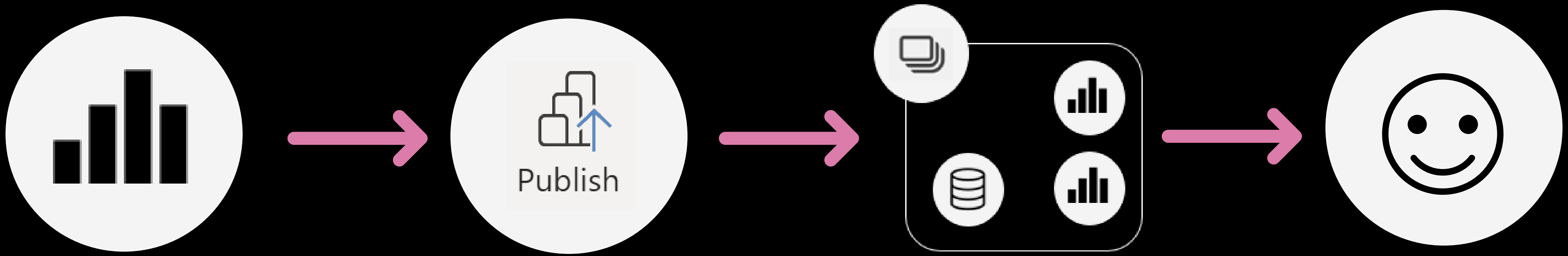
powerbiprincess.com

September 2023

About me

- Paulien van Eijk
- Gouda - Netherlands
- Data & Analytics consultant at Macaw
- Focus on Power BI for the past 5 years
- Data engineer in the making

Publishing your report online



Build dataset and report

Publish to service

Power BI Service

Everybody happy

Things you might have encountered

- Collaboration is difficult
- Keeping track of changes is (almost) impossible
- Download report from service to get latest version
- Publishing a previous version

When using DTAP:

- Accidently deploying to production instead of development.. whoops
- Forgot to change data source connection from dev to prod
- Overwriting data in production

What improved / can be avoided?

- Collaboration is difficult
- Keeping track of changes is (almost) impossible
- Download report from service to get latest version
- Publishing a previous version

When using DTAP:

- Accidently deploying to production instead of development.. whoops
- Forgot to change data source connection from dev to prod
- Overwriting data in production

What about these?

- Collaboration is difficult
- Keeping track of changes is (almost) impossible
- Download report from service to get latest version
- Publishing a previous version

New file format: .pbip

New file format: .pbip

- Power BI Project file
- Saving report and dataset artifacts in separate plain text files in a clear folder structure
- Introduced in June 2023, but still in preview

Why should we care?

Enables capabilities, such as:

- Editable format: Easily make changes using code editors
- Source Control: Track version history, compare versions, revert to previous versions
- CI / CD: Quality controls (review, testing) before deployment to production

Demo

How do we enable the other benefits?

- Editable format: Easily make changes using code editors
- Source Control: Track version history, compare versions, revert to previous versions (e.g. Git.. more about that later ❤️)
- CI / CD: Quality controls (review, testing) before deployment to production

Using Git with Power BI

Who has used Git before?

Who has used Git in combination with Power BI?

Git, what is it?

- Git is a version control system to track and manage changes
- It provides functionalities for:
  - Version control
  - Branching
  - Merging
  - Collaboration
  - Tracking changes
  - Compare versions

Wait, what were our problems again?

- Collaboration is difficult
- Keeping track of changes is (almost) impossible
- Download report from service to get latest version
- Publishing a previous version

Sounds like it is exactly what we need

Git, what is it?

- Git is a version control system to track and manage changes
- It provides functionalities for:
  - Version control
  - Branching
  - Merging
  - Collaboration
  - Tracking changes
  - Compare versions

Demo

Git, what is it?

- Git is a version control system to track and manage changes
- It provides functionalities for:
  - Version control
  - Branching
  - Merging
  - Collaboration
  - Tracking changes
  - Compare versions

Who is familiar with the concept of branching and merging?

Branching – General concept

- Safely create new feature / fix bug
- Isolate workflow
- Copy of code, without modifying “production”,
- Test before saving to “production”

Demo

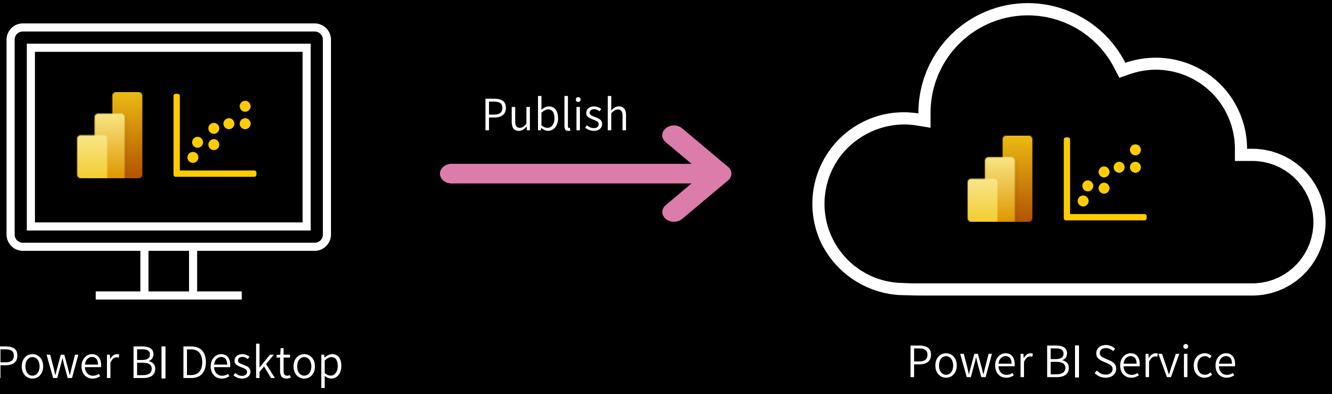
Wasn’t this already possible?

- Yes! Branching and merging was already possible before.
- But, it was in an relatively unreadable file format, called .pbix.
- Dataset and reports were not seperated => one big file
- Dataset contained data
    - Except if you seperately upload a xmla

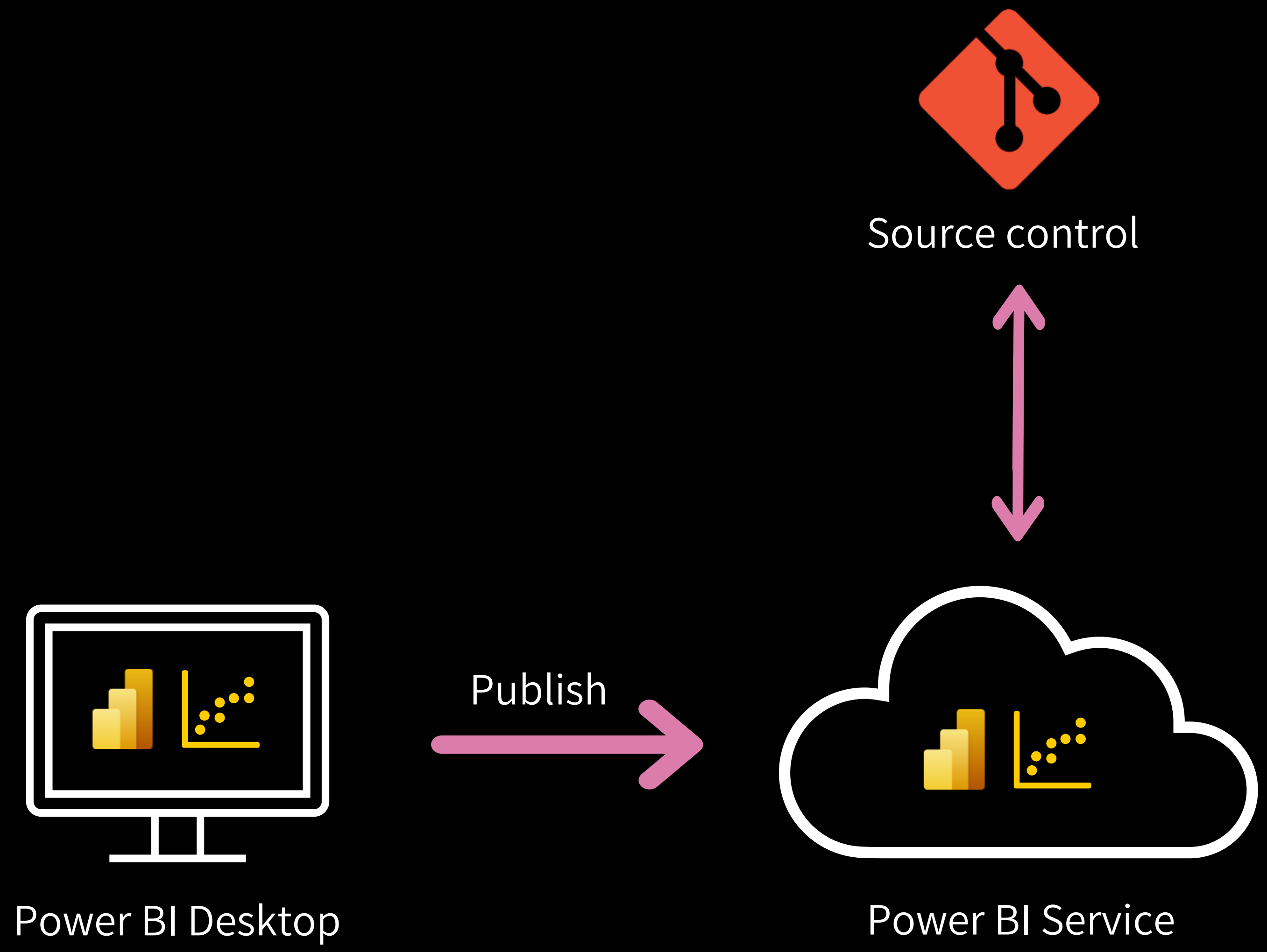
Two “types” of Git Integration to distinguish

- Git as your source control and versioning system – either locally or in the cloud
- Git Integration **in Power BI service** with Azure DevOps as our source control and versioning system

Git and Power BI



Git and Power BI



Source control

Publish

Power BI Desktop

Power BI Service

Demo

Keep in mind that

- DevOps and Power BI in same region
- Power BI Premium capacity / Premium per User / Fabric capacity

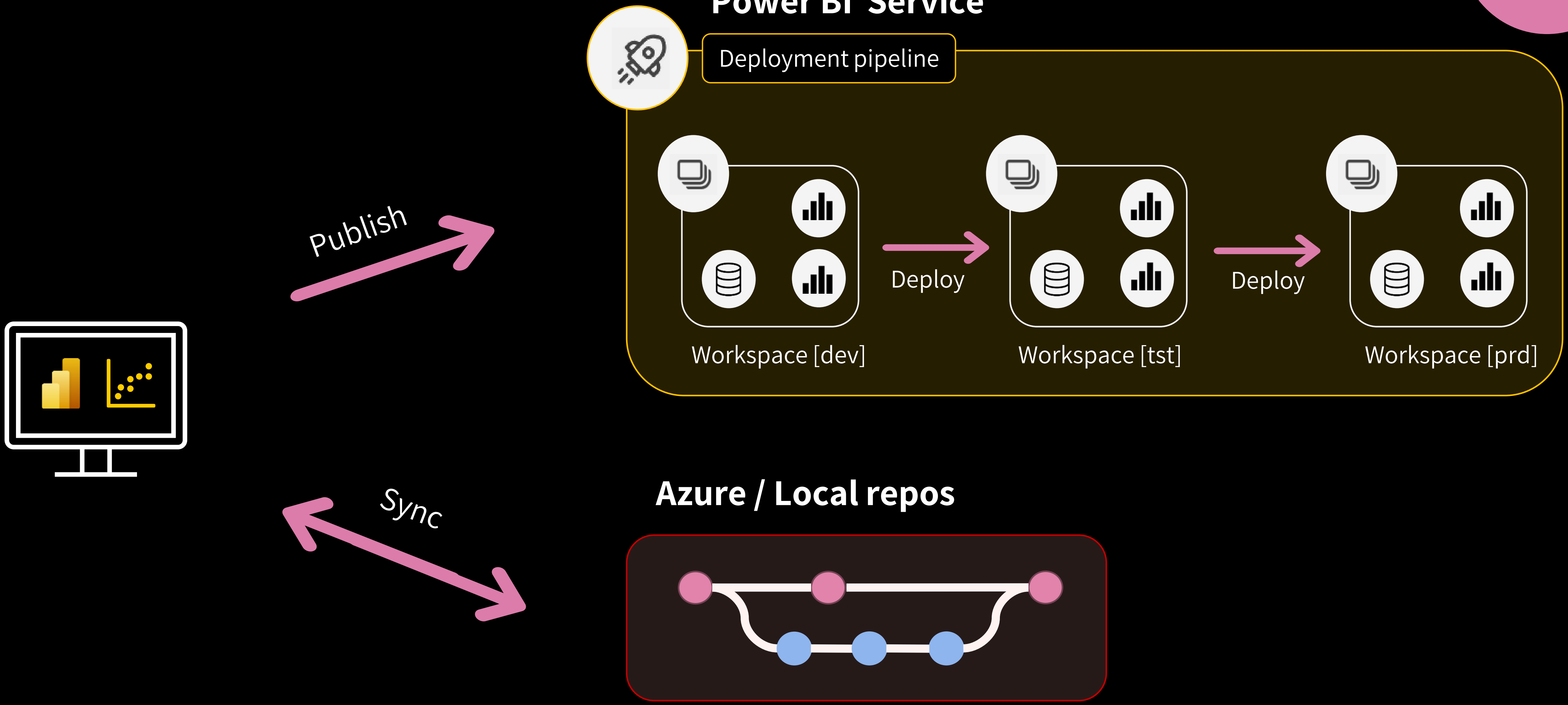
How can we utilize:

Git Integration

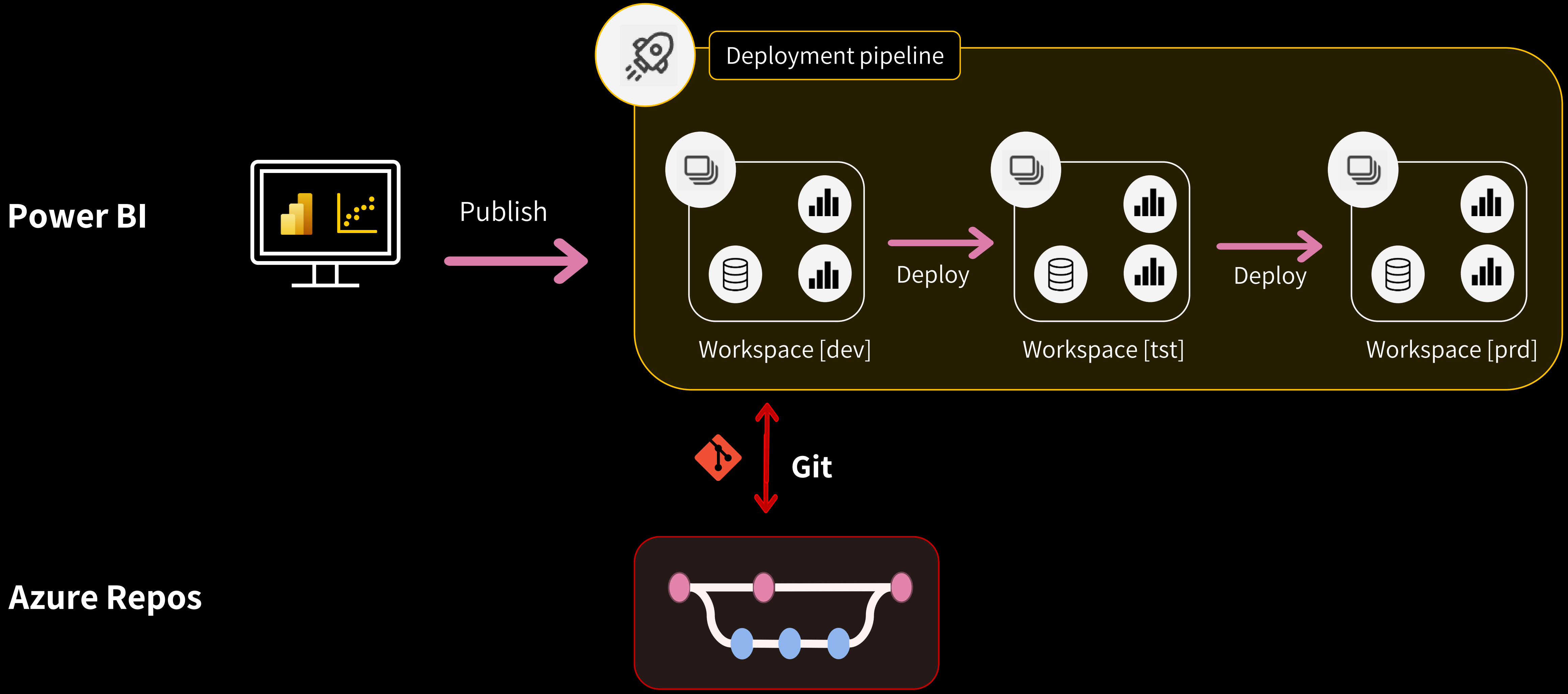
AND / OR

The Power BI Service Git integration

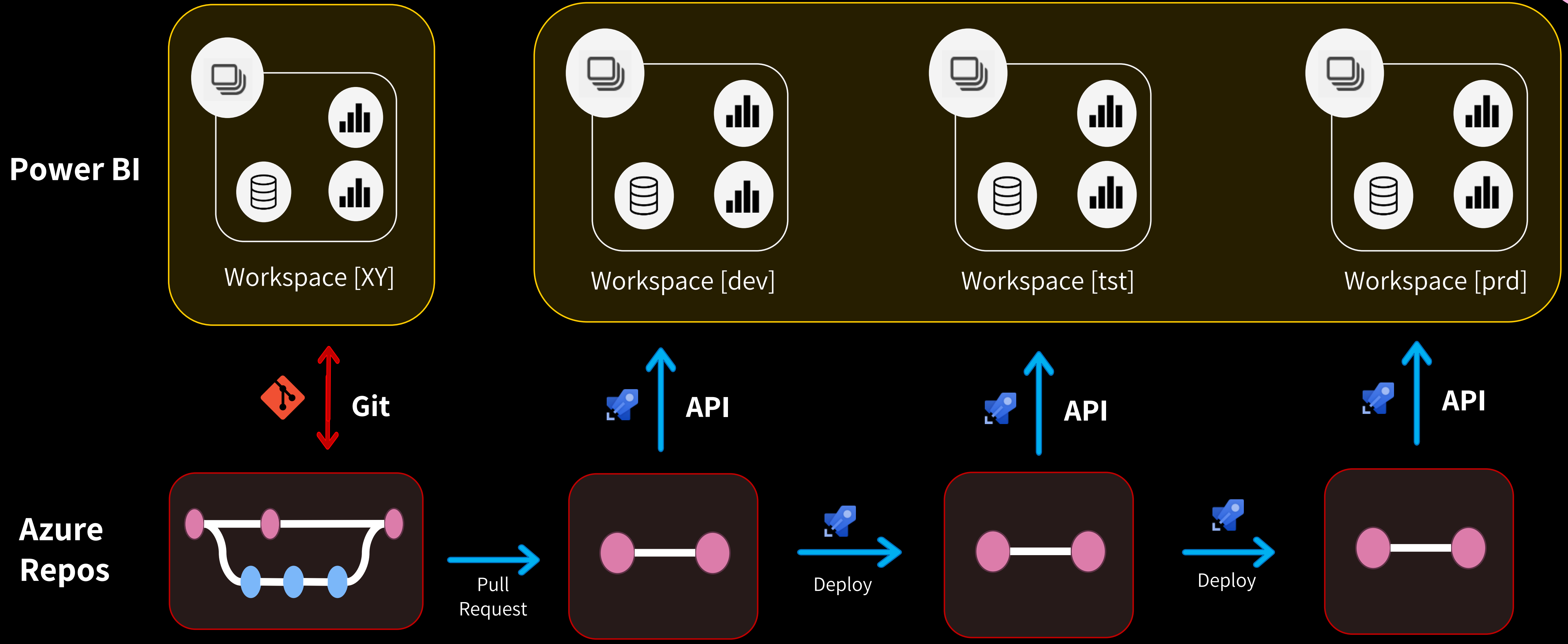
‘Local’ developer



Connect dev workspace to main branch



Deverlopers’ own workspace sync

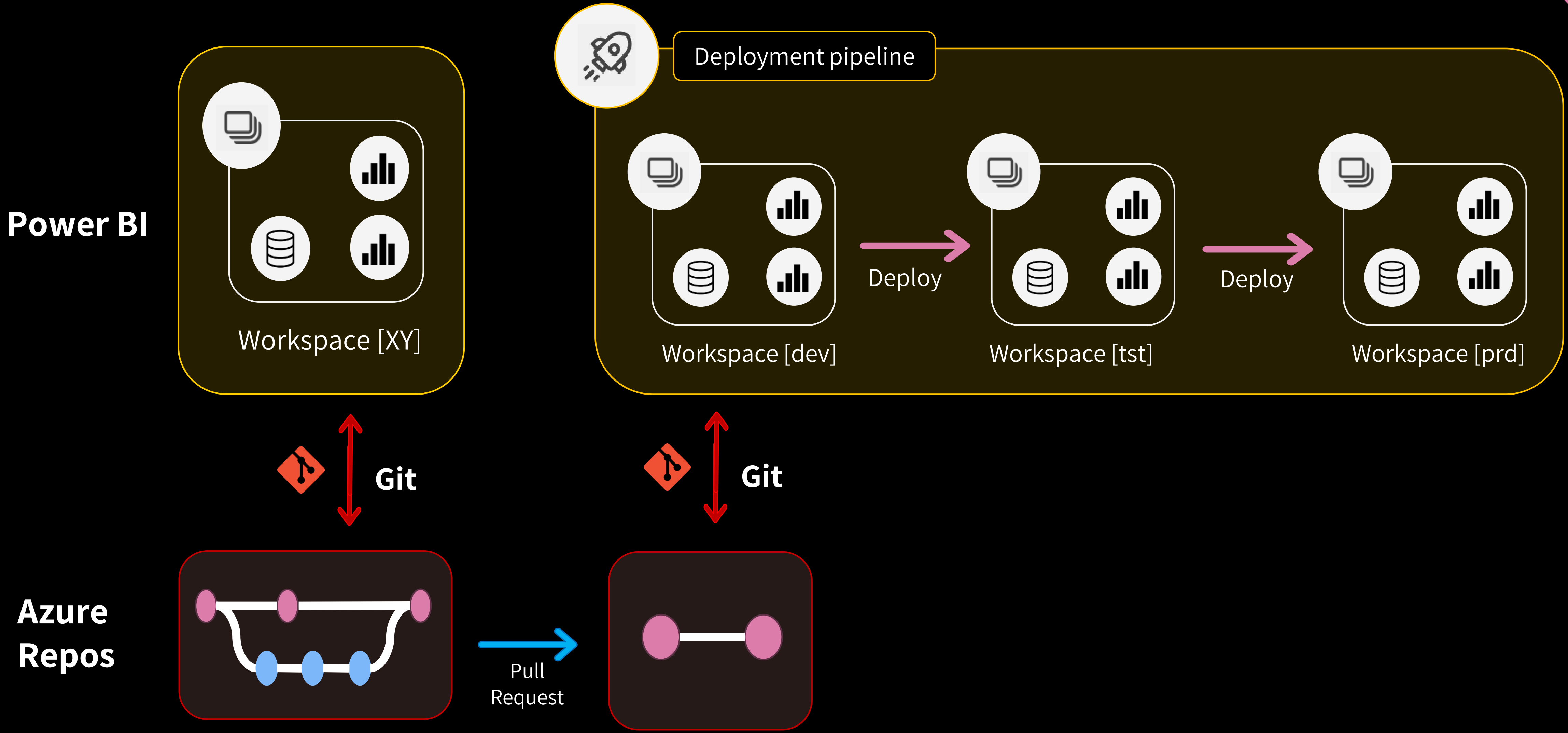


powerbiprincess.com

Combine scenarios

- It’s not as black or white as the solutions presented
- Combining the last to options creates an interesting fourth scenario

Deverlopers' own workspace sync

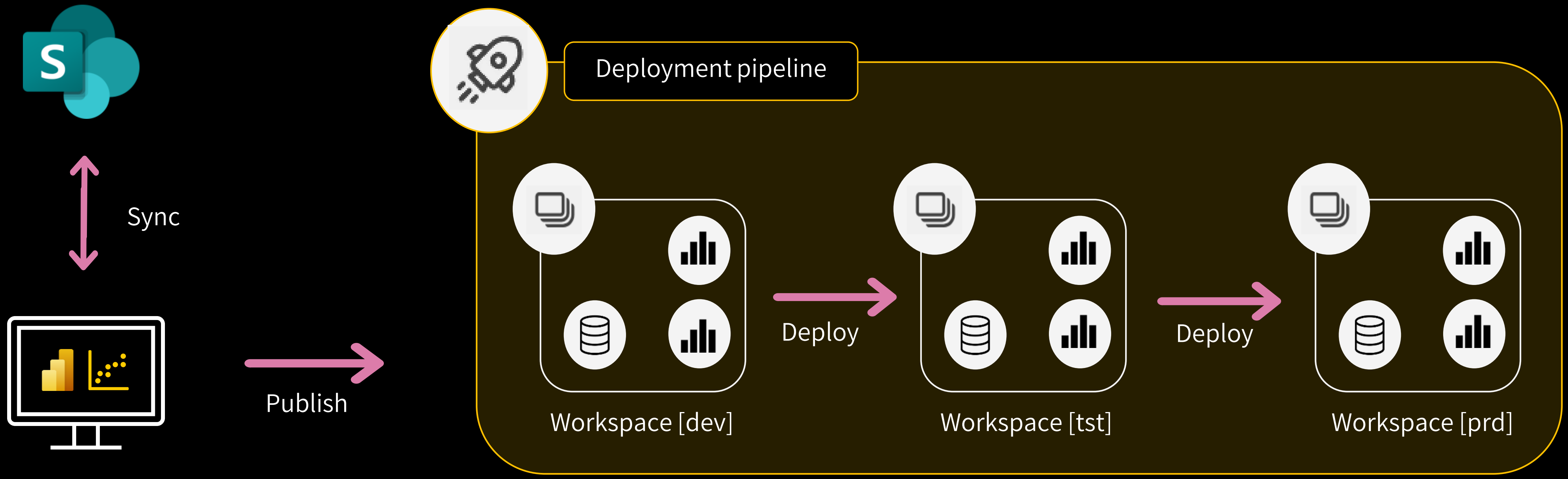


Combine scenarios

- Feels like a Fabric scenario
  - But.. no access to code base on test and prod? No rollback?

    → Solution: add branch to each workspace
  - Both actions in Azure and Power BI.. Confusing?
  - No additional code checks between environments
- High potential, but not yet there (in my opinion..)
- Therefore, for now not referred to as a fourth option (but definitely keep an eye out on the latest deverlopments)

So, how would this look like in practice?



What options do we have?

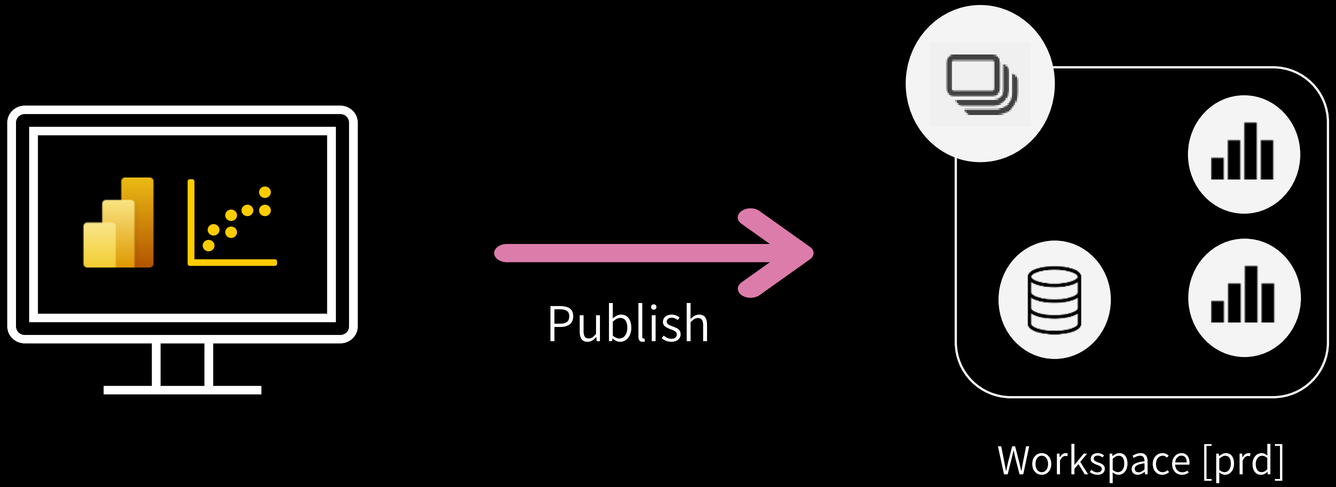
1. **Local developer:** Use the .pbip extension combined with a local/cloud git repository to quickly identify changes and revert if necessary.
2. **Connect development workspace to Git:** Use the Git Integration in Power BI Service for development environment and release via deployment pipeline
3. **Connect personal dev workspace to Git and release via Azure:** Starting using branching + pull requests + Azure pipelines

What options do we have?

1. **Local developer:** Use the .pbip extension combined with a local/cloud git repository to quickly identify changes and revert if necessary.
2. **Connect development workspace to Git:** Use the Git Integration in Power BI Service for development environment and release via deployment pipeline
3. **Connect personal dev workspace to Git and release via Azure:** Starting using branching + pull requests + Azure pipelines

So, how would this look like in practice?

- Current set up at another client



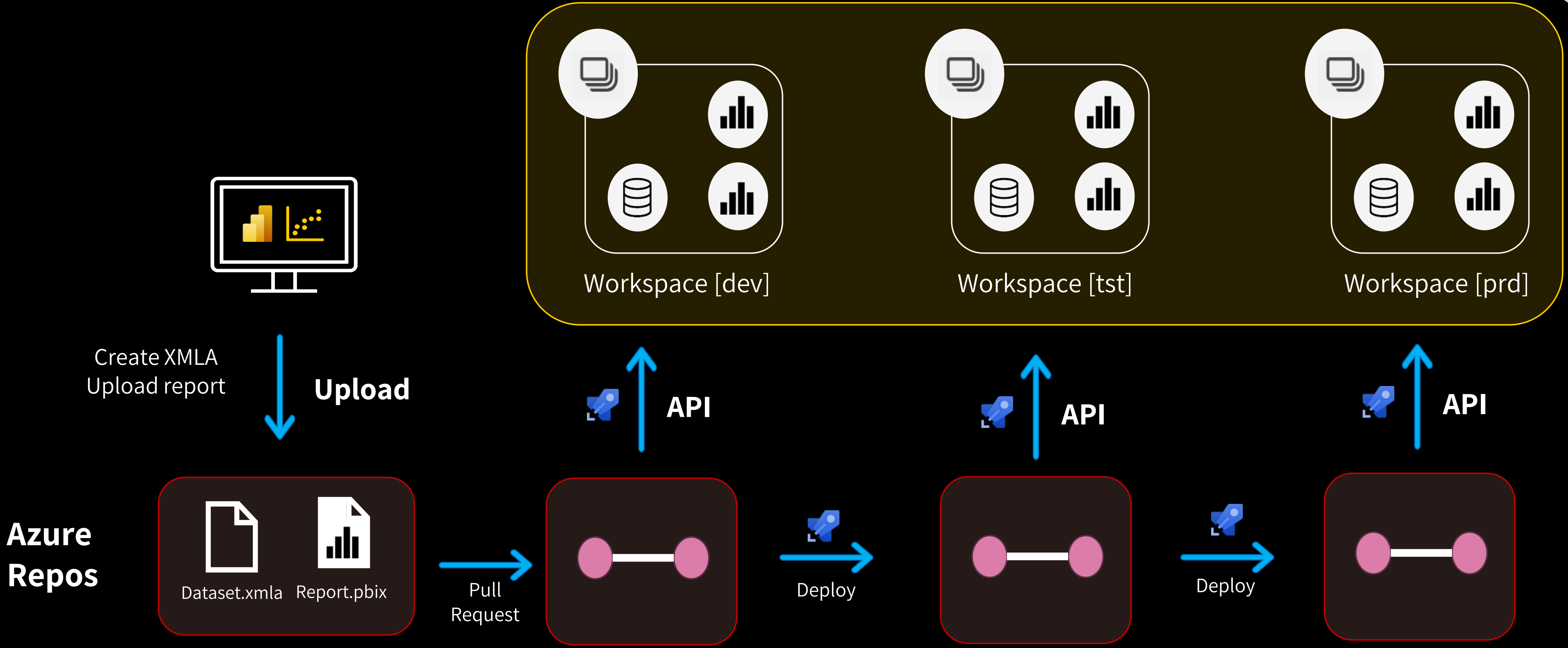
What options do we have?

1. **Local developer:** Use the .pbip extension combined with a local/cloud git repository to quickly identify changes and revert if necessary.
2. **Connect development workspace to Git:** Use the Git Integration in Power BI Service for development environment and release via deployment pipeline
3. **Connect personal dev workspace to Git and release via Azure:** Starting using branching + pull requests + Azure pipelines

What options do we have?

1. **Local developer:** Use the .pbip extension combined with a local/cloud git repository to quickly identify changes and revert if necessary.
2. **Connect development workspace to Git:** Use the Git Integration in Power BI Service for development environment and release via deployment pipeline
3. **Connect personal dev workspace to Git and release via Azure:** Starting using branching + pull requests + Azure pipelines

So, how would this look like in practice?



Workspace [dev]

Workspace [tst]

Workspace [prd]

Create XMLA
Upload report

Upload

API

API

API

Azure Repos

Dataset.xmla

Report.pbix

Pull Request

Deploy

Deploy

What options do we have?

1. **Local developer:** Use the .pbip extension combined with a local/cloud git repository to quickly identify changes and revert if necessary.
2. **Connect development workspace to Git:** Use the Git Integration in Power BI Service for development environment and release via deployment pipeline
3. **Connect personal dev workspace to Git and release via Azure:** Starting using branching + pull requests + Azure pipelines

What options do we have?

1. **Local developer:** Use the .pbip extension combined with a local/cloud git repository to quickly identify changes and revert if necessary.
2. **Connect development workspace to Git:** Use the Git Integration in Power BI Service for development environment and release via deployment pipeline
3. **Connect personal dev workspace to Git and release via Azure:** Starting using branching + pull requests + Azure pipelines

So.. what now?

- Insight in how the new .pbip can help us mature our deployment of solutions
- Insight in how the new Git integration can help us mature our deployment of solutions
- Waiting for the final puzzle pieces:
    - Support of all artifacts
    - Deployment APIs for .pbip files
    - TMDL
    - REDL (?)

Q&A

Big thank you to our great sponsors

Premium

twoday kapacity

Fellowmind

ITPA CONSULTING

ITPA consulting

dataon Power your data

dataon

Standard

KPMG

CatMan® Solution

BI BUSINESS INNOVATION BUSINESS INFORMATION BUSINESS INTELLIGENCE

A|S

enversion

Tabular Tools