

SQL database in Microsoft Fabric

Series Scenario – Conference session application

You need:

- to store and serve data for the conference webpage.
- to be able to run analysis over the data.
- an application that allows attendees to easily search for and find sessions of interest to them.
- to follow modern best practices for application lifecycle.
- to be able to monitor and troubleshoot database and query performance.

Course Overview and agenda



Episode 1: Introduction and Overview; Getting started



Episode 2: Dataflows, Notebooks, Reports



Episode 3: GenAI and vector databases



Episode 4: GraphQL and application development



Episode 5: Application lifecycle management



Episode 6: Performance Dashboard, Recap

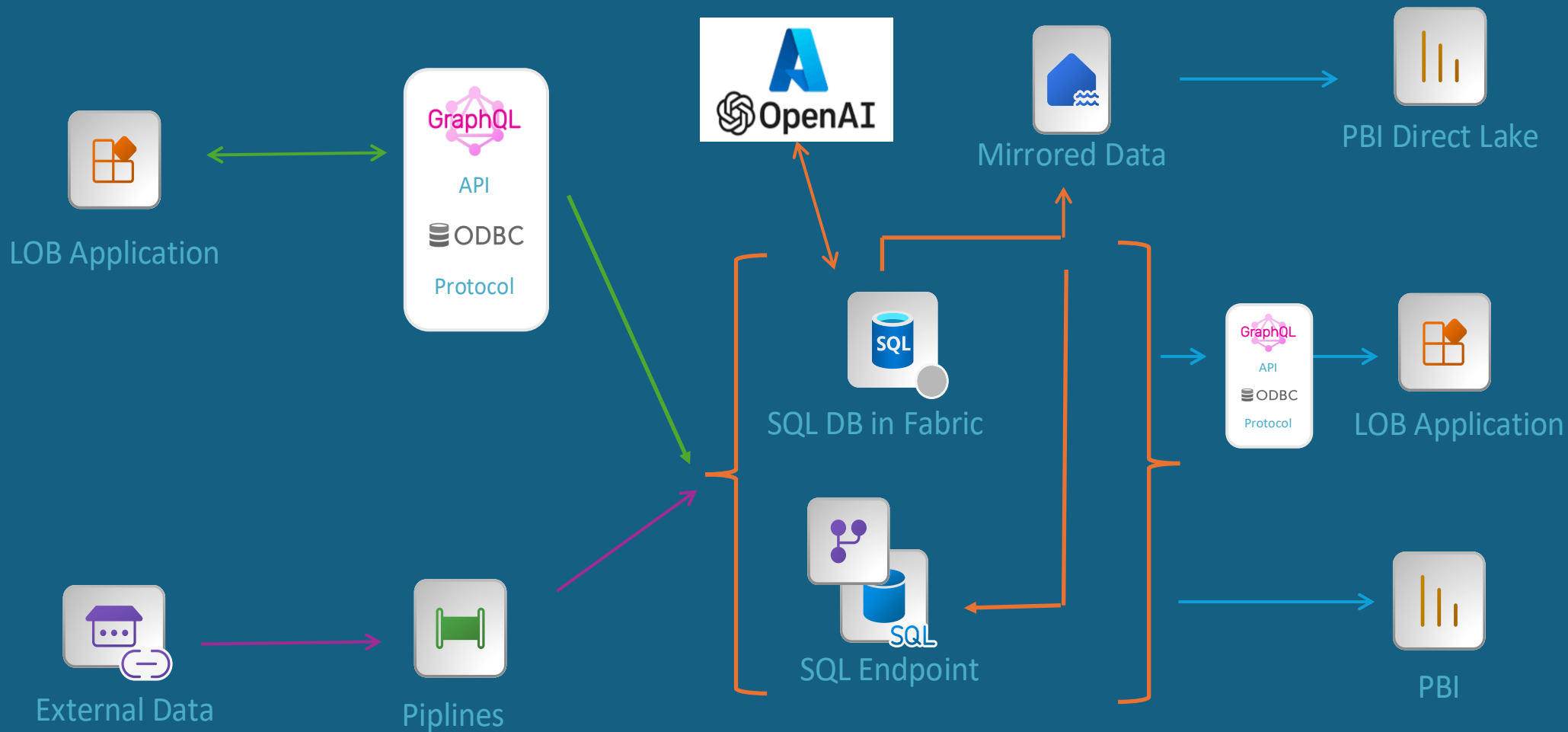
DATA SOURCE

INGESTION

AI

STORE

EXPOSE



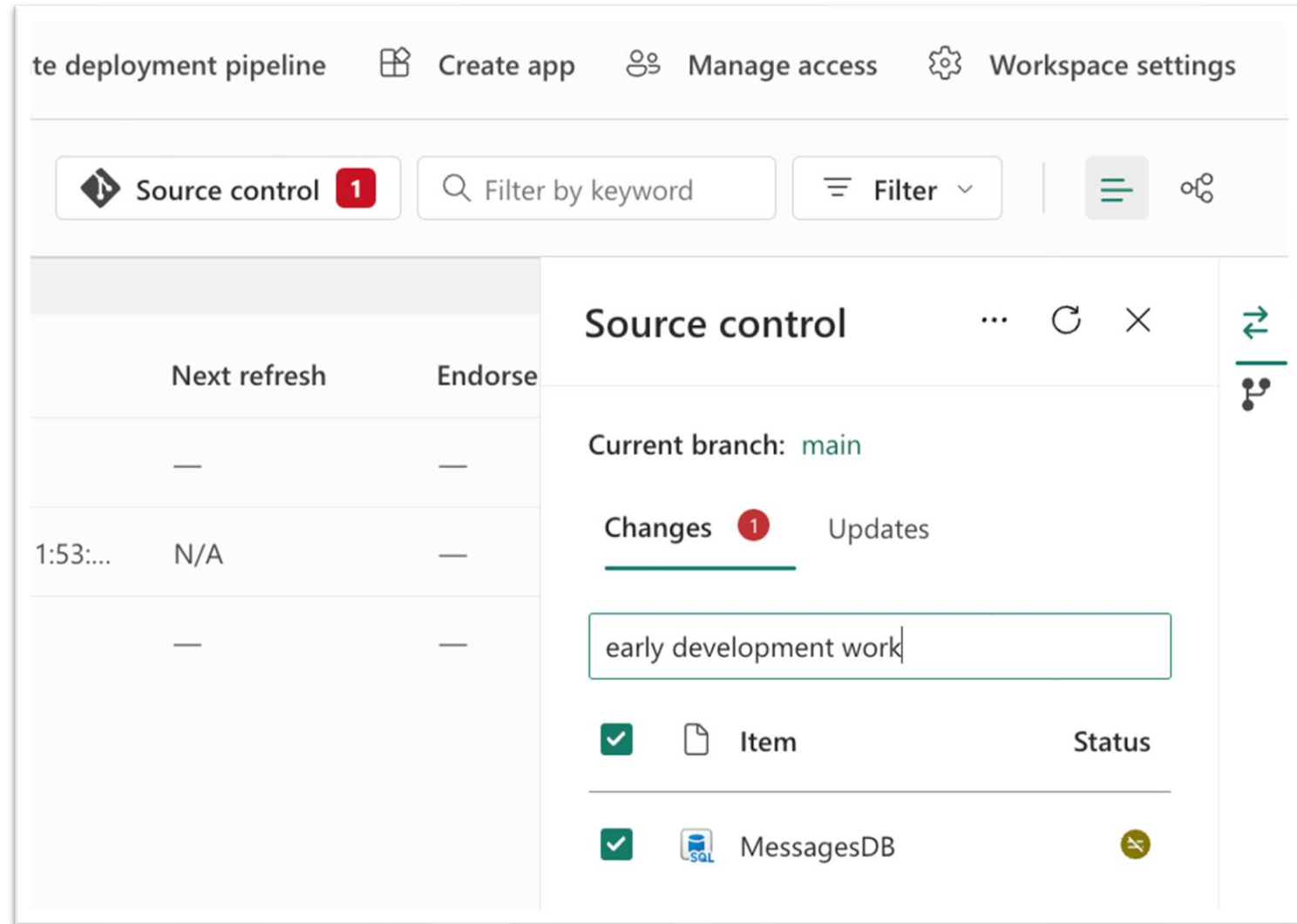
Episode 5: Application Lifecycle Management (ALM)

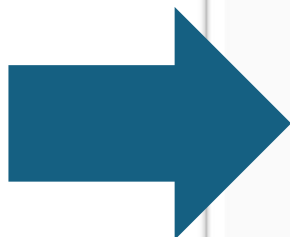
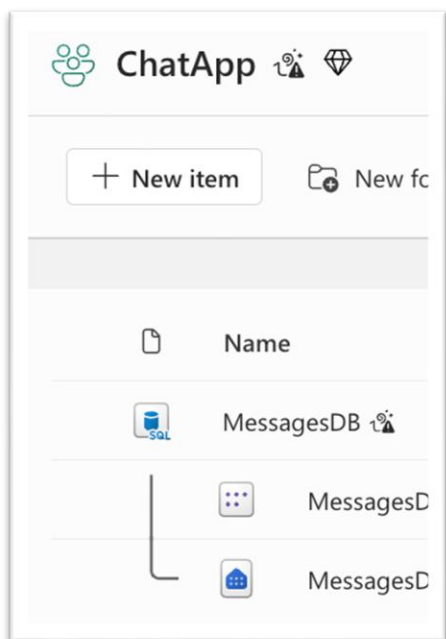
Fabric SQL DB CI/CD goals

1. Get every database under source control
2. Make development databases self-service and disposable
3. Offer a low-friction mechanism to perform deployments and code quality checks without learning a new tool
4. Keep integration points open to more advanced CI/CD tools and workflows

Fabric - commit to source control

- Develop directly in the database
- Quickly iterate in early development
- Keep track of changes (commit along the way)





t App Demo / Repos / Files / Chat App Demo

Chat App Demo

- DatastoreWorkspace
 - MessagesDB.SQLDatabase
 - chatter
 - StoredProcedures
 - checkUserToken.sql
 - GetCurrentlyActivePlaces.sql
 - getLatestMessages.sql
 - GetUserInfo.sql
 - joinPlace.sql
 - newMessage.sql
 - newUser.sql
 - RevokeMessage.sql
 - updateUser.sql
 - Tables
 - Messages.sql**
 - PlaceMembership.sql
 - Places.sql
 - TextAnalysis.sql
 - Users.sql
 - Security
 - .platform
 - MessagesDB.sqlproj

Messages.sql

Contents History Compare Blame

```
1 CREATE TABLE [chatter].[Messages] (  
2     [MessageId] INT IDENTITY (1, 1) NOT NULL,  
3     [UserId] UNIQUEIDENTIFIER NOT NULL,  
4     [PlaceId] INT NOT NULL,  
5     [CreatedDateTime] DATETIME2 (7) NOT NULL,  
6     [IsActive] BIT NOT NULL,  
7     [MessageContents] NVARCHAR (MAX) NOT NULL,  
8     PRIMARY KEY CLUSTERED ([MessageId] ASC),  
9     CONSTRAINT [FK_Messages_PlaceId] FOREIGN KEY ([PlaceId]) REFERENCES [chatter].[Places] ([PlaceId]),  
10    CONSTRAINT [FK_Messages_UserId] FOREIGN KEY ([UserId]) REFERENCES [chatter].[Users] ([UserId])  
11 );  
12  
13  
14 GO  
15  
16 CREATE NONCLUSTERED INDEX [IX_Messages_PlaceId_IsActive_CreatedDateTime]  
17 ON [chatter].[Messages]([PlaceId] ASC, [IsActive] ASC, [CreatedDateTime] ASC);  
18  
19  
20 GO  
21  
22
```

DatastoreWorkspace

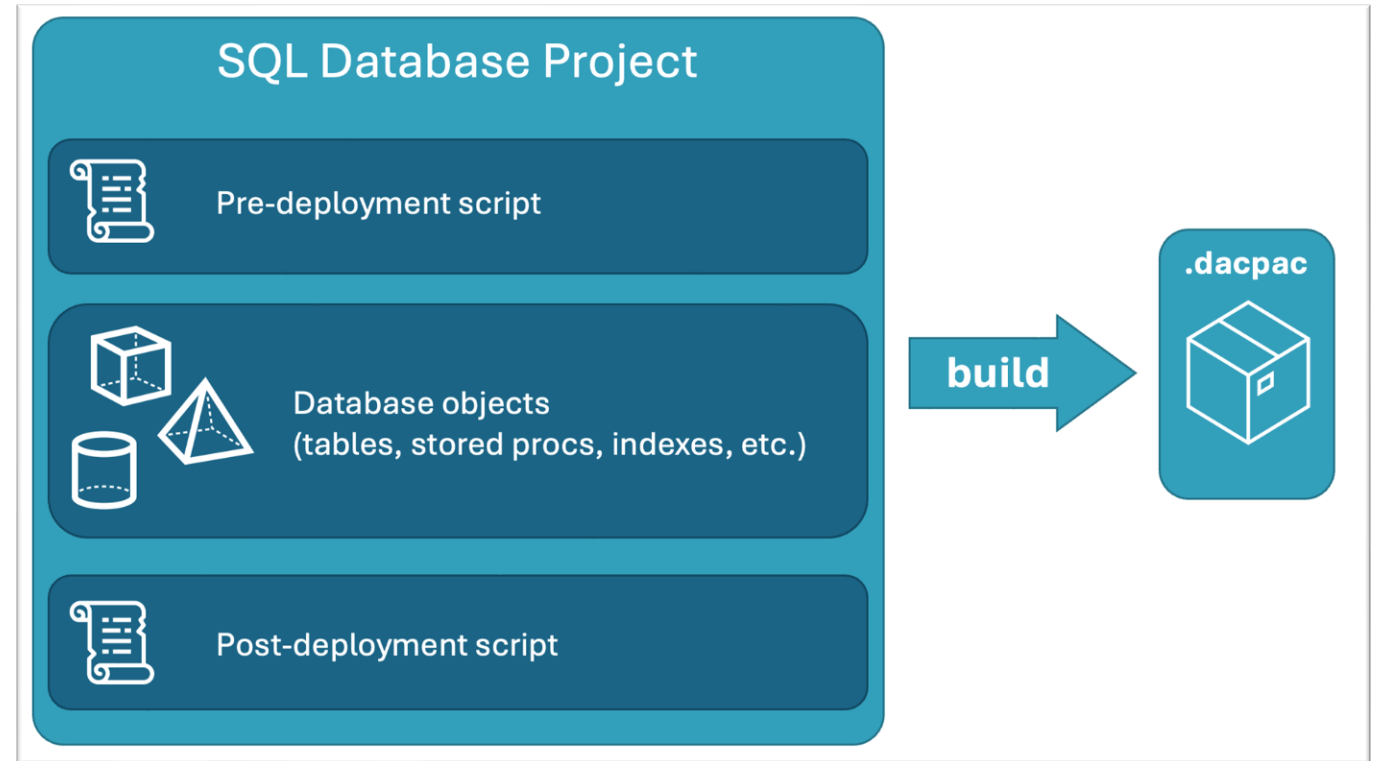
Contents History

Graph Commit

- early development work
78615567 Drew Skwiers-Koballa Today at 12:40 PM
- Committing 1 item from workspace 164f249e-b429-4351-8fd5-3c83ef37ad76
9bb0e9ba Drew Skwiers-Koballa Today at 12:14 PM
- Creating directory DatastoreWorkspace
42a5ede8 Drew Skwiers-Koballa Today at 12:14 PM

What's in the database code?

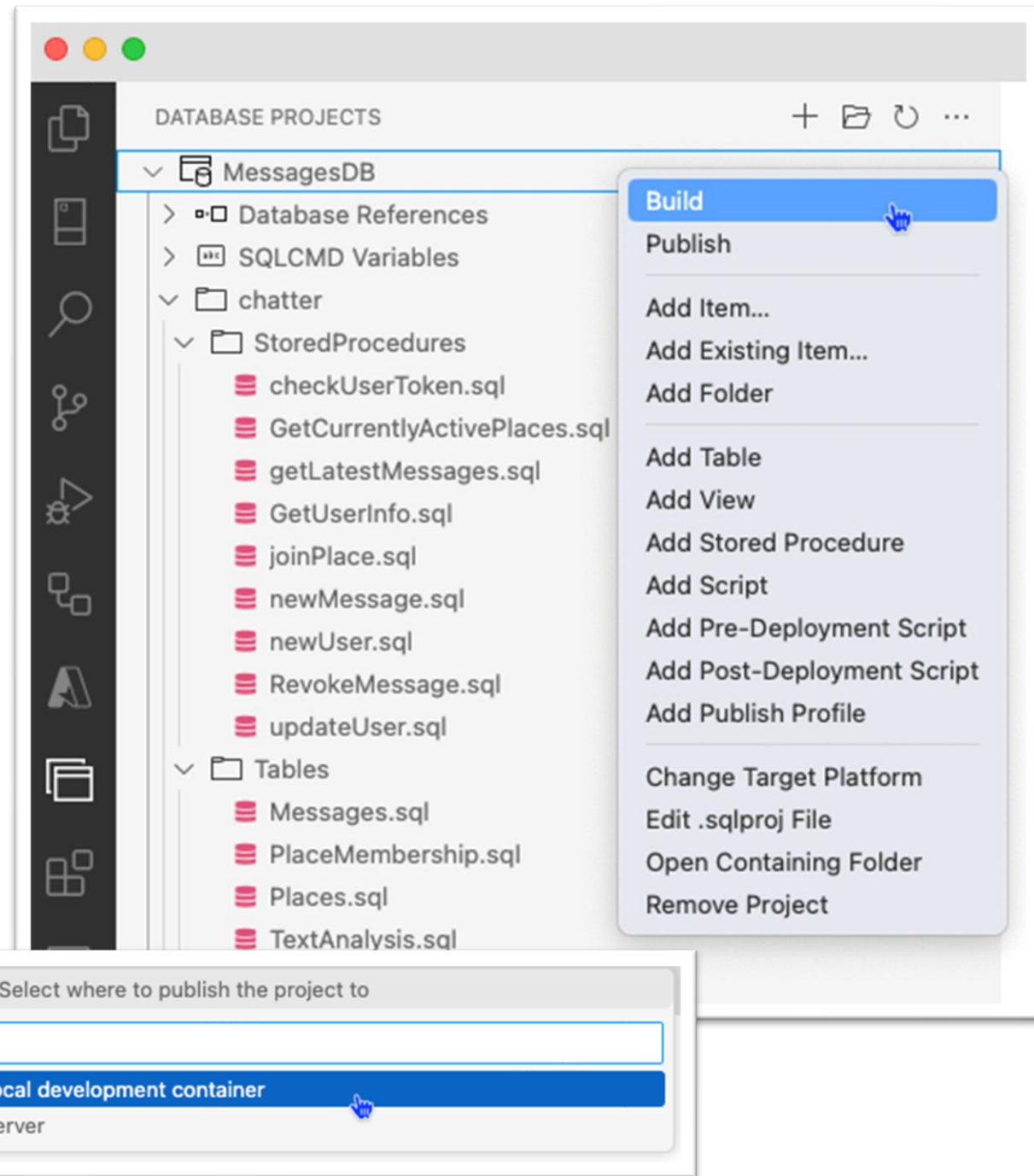
- SQL database project
- 1 file for each object
- Definitions as CREATE T-SQL statements
- Build process validates database model
- Portable and reusable artifact of database state



SQL projects in VS Code and Visual Studio

Develop in client tools

- Create new branch in source control
- Clone (copy) code to local machine
- Edit and build SQL project in VS Code with mssql and SQL Database Projects extension
- Connect to Fabric or use a local instance for testing



Workspace check
into and update
from source
control




Source control setup

1. Primary Fabric workspace where proof of concept (POC) work is taking place
2. Create an empty git repository (Azure DevOps or GitHub)
3. Connect workspace to main* branch of repo

Git integration (Preview)

Connect to Git to manage your code and back up your work. [Learn more](#)


 Connect Git provider and account


Provider


Azure DevOps

AAD account

drskwier@microsoft.com

 Log out

Manage all accounts 


 Connect Git repository and branch

Organization *


drskwier

Project *


Chat App Demo

Git repository * 

Chat App Demo

Branch * 

main

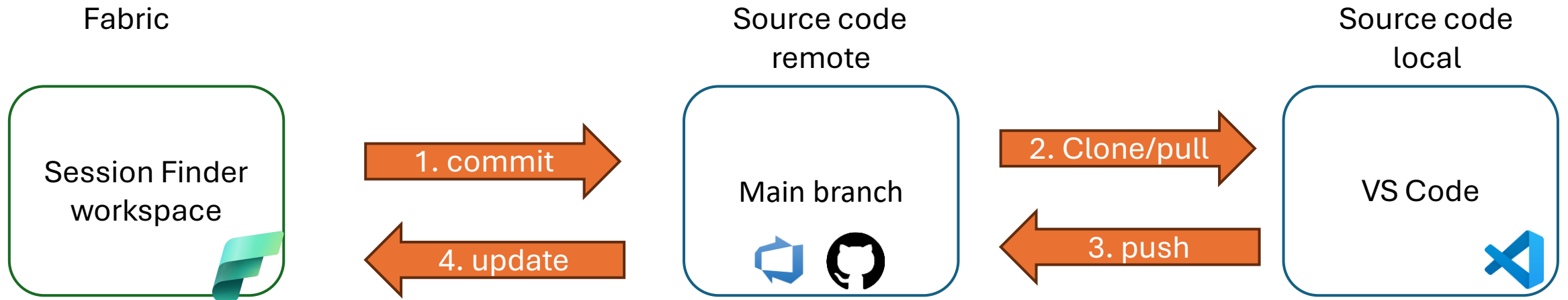
Git folder 

Enter name of folder

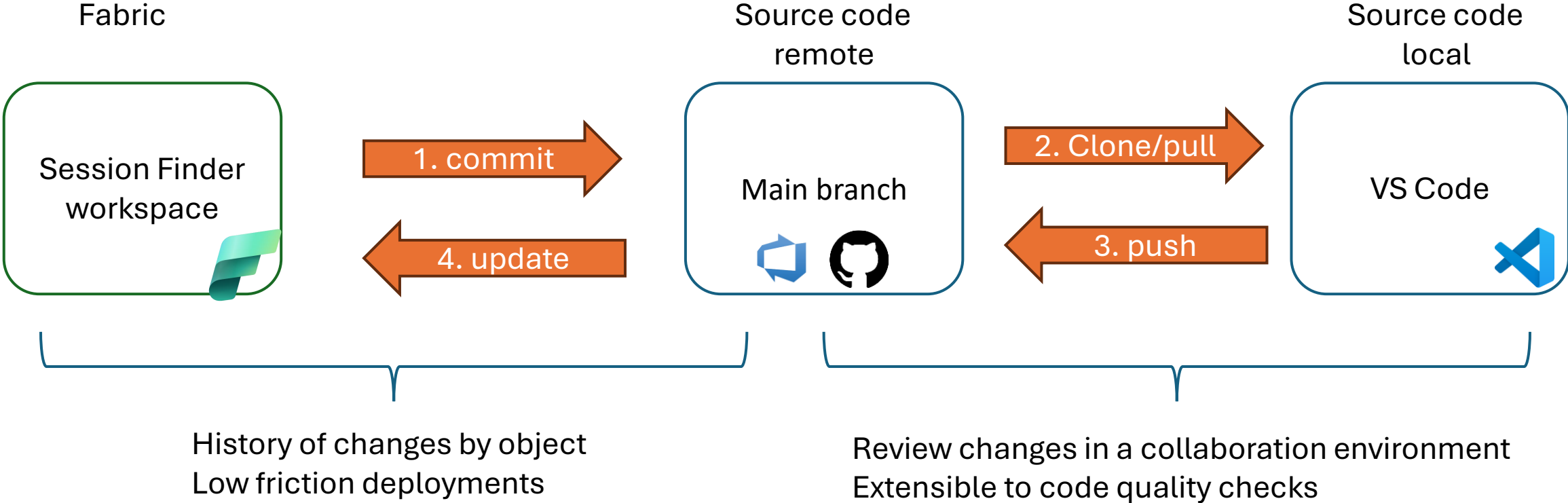
Connect and sync

Cancel

Source control integration in action (activity 5.1)



Benefits from source control integration

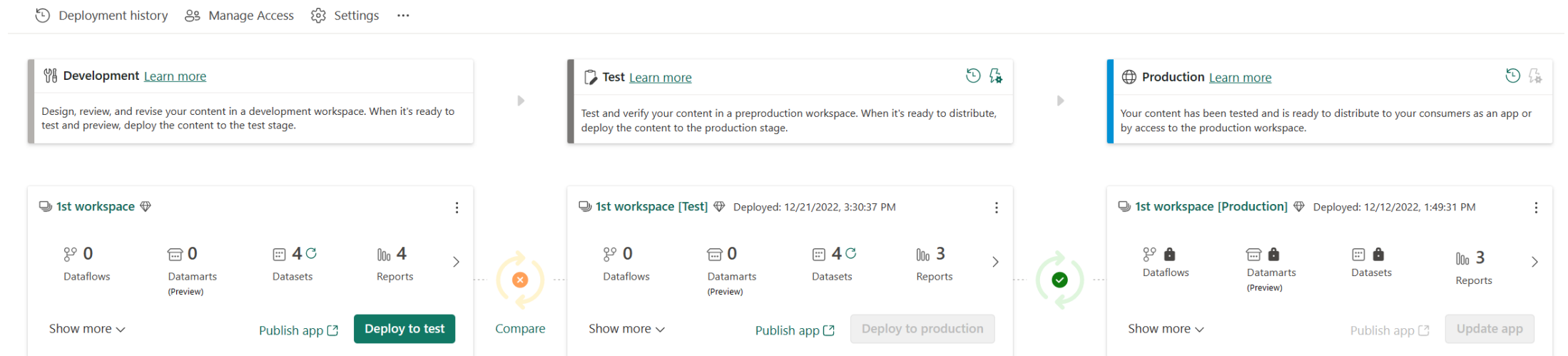


Multi-workspace deployment pipelines

Data pipelines != deployment pipelines

Deployment pipelines in Fabric

- Multi-workspace deployment process
- Dev/Test/Prod environment automation
- Repeatable deployments



SqlPackage publish

Bridge to bring your own pipelines

Setup Azure DevOps identity



New service connection

Choose a service or connection type

Search connection types

- ☐ Azure Classic
- ☐ Azure Repos/Team Foundation Server
- ☒ Azure Resource Manager
- ☐ Azure Service Bus
- ☐ Bitbucket Cloud

New Azure service connection

Azure Resource Manager

Authentication method

- ☒ Workload Identity federation (automatic) Recommended
- ☐ Workload Identity federation (manual)
- ☐ Service principal (automatic)
- ☐ Service principal (manual)
- ☐ Managed identity
- ☐ Publish Profile

[Need help choosing a connection type?](#) Back Next

New Azure service connection

Azure Resource Manager using Workload Identity federation with OpenID Connect (automatic)

Scope level

- ☒ Subscription
- ☐ Management Group
- ☐ Machine Learning Workspace

Subscription

Resource group

drskwier-chat-app-hackathon

Details

Service connection name

devops-demo

Description (optional)

[Home](#) >



drskwier-Chat App Demo-d

Search

Delete Endpoints Preview features

Overview

Quickstart

Integration assistant

Diagnose and solve problems

Manage

Support + Troubleshooting

Essentials

Display name : drskwier-Chat App Demo-d51

Application (client) ID :

Object ID :

Directory (tenant) ID :

Supported account types : [My organization only](#)



Setup Azure DevOps identity



🔒 Manage SQL security



Manage access

Add or remove users for this role.

Add people, groups or apps

drskwier-Chat App Demo-d51



drskwier-Chat App Demo-d51

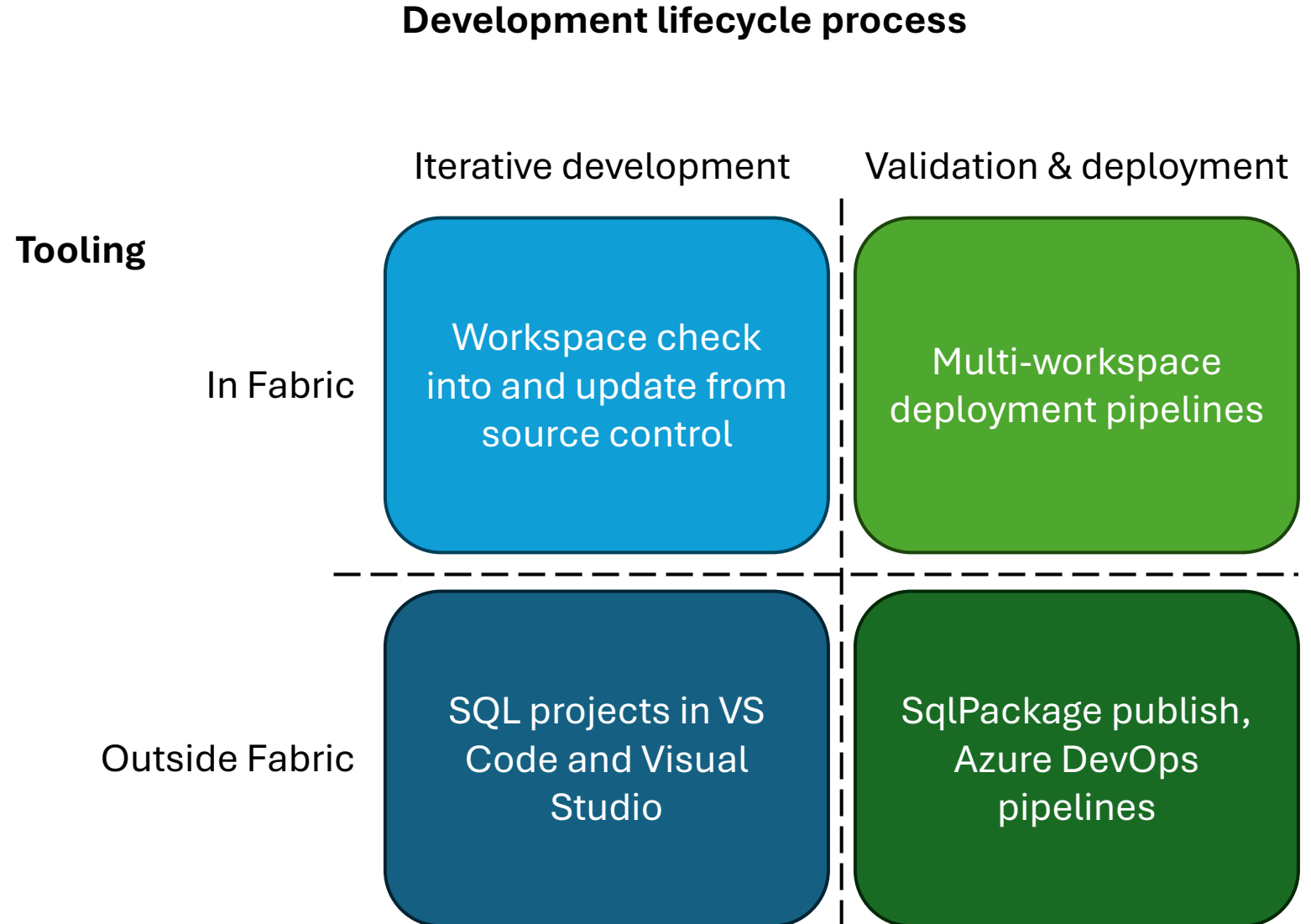
Add

Clear



```
30 # this connection settings would usually be in variables
    Settings
31 - task: SqlAzureDacpacDeployment@1
32   inputs:
33     azureSubscription: 'devops-demo'
34     AuthenticationType: 'aadAuthenticationIntegrated'
35     ServerName: 'fleet-srv-767c0a4b-beec-4d0a-9c10-333efe44de71-a08b9015aef7.database.windows.net'
36     DatabaseName: 'MessagesDB-91784e0c-92f1-4c30-b3c7-2e40acd5be9c'
37     deployType: 'DacpacTask'
38     DeploymentAction: 'Publish'
39     DacpacFile: '$(Pipeline.Workspace)/s/DatastoreWorkspace/MessagesDB.SQLDatabase/bin/Debug/MessagesDB.dacpac'
40     IpDetectionMethod: 'AutoDetect'
41
42
```

SQL database in Fabric CI/CD ecosystem



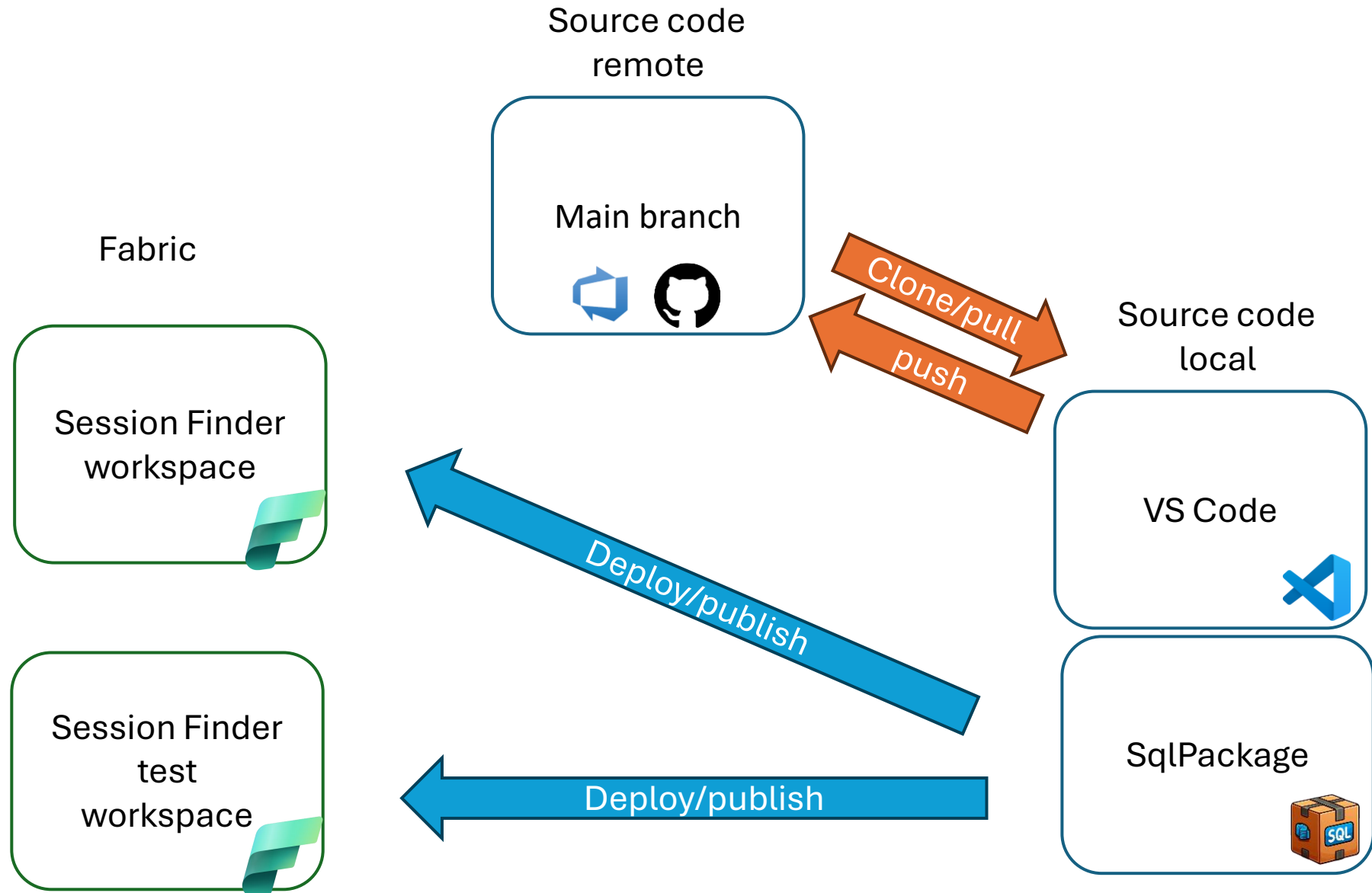
CI/CD



Fabric deployment pipelines (activity 5.2)



SqlPackage deployments (activity 5.2)



End Episode 5

Overview of ALM

Checking database into source control

Database deployments - Fabric pipelines and SqlPackage