The Power Platform Governance

Microsoft Discover: Power Platform Club · 27. August 2020

Peter Hödl, Toni Pohl

www.atwork.at

What we do...

Microsoft
Cloud Solution Provider

atwork
We get you in the cloud!

Gold
Microsoft
Partner
Microsoft

Collaboration

Consulting

Security &
Governance

Power
Platform

Migrations

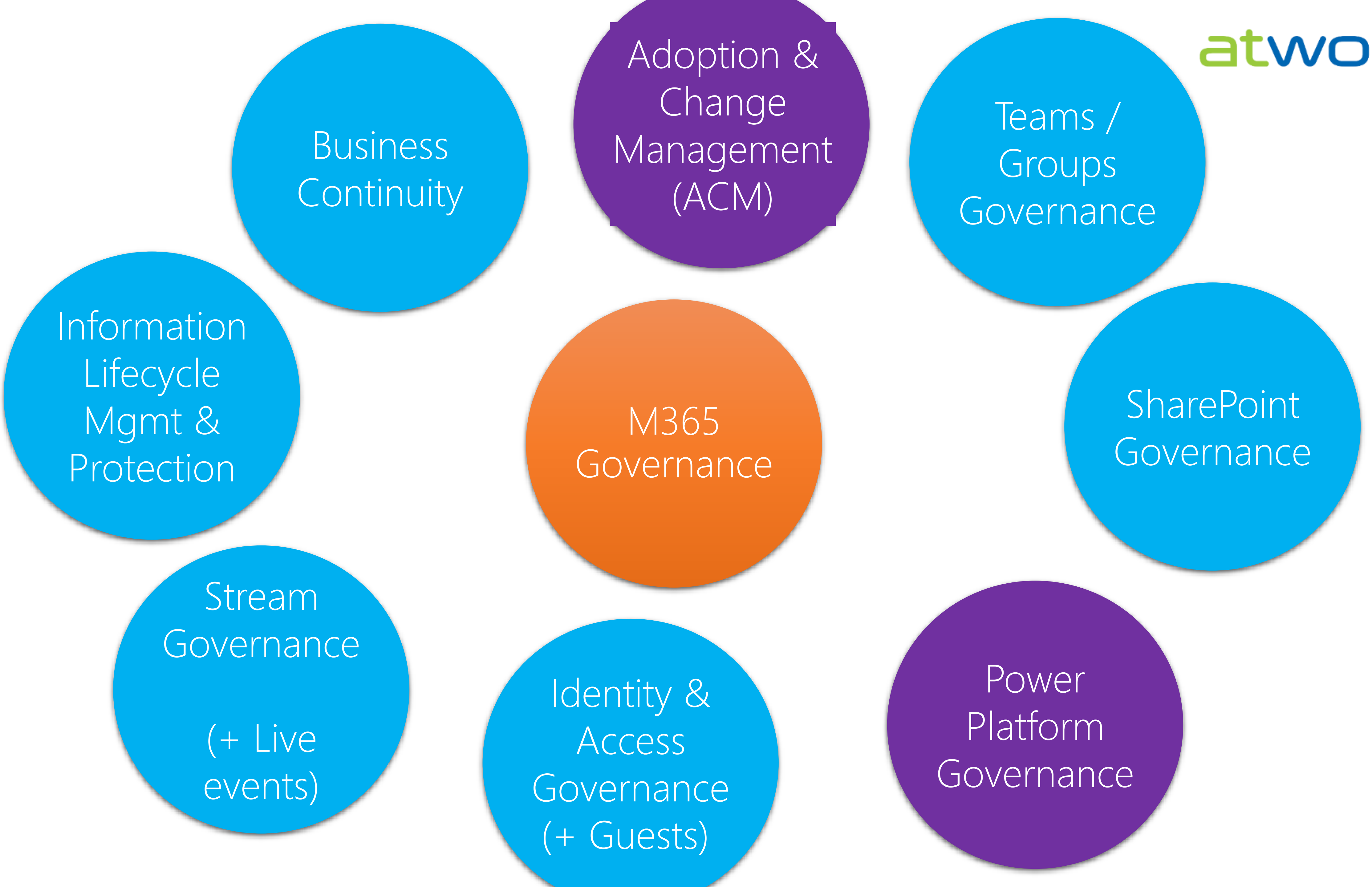
Apps

Development

What is IT governance?

"IT Governance is a process used to monitor and control key information technology capability decisions - in an attempt - to ensure the delivery of value to key stakeholders in an organization."

atwork



M365 Governance

Adoption & Change Management (ACM)

Teams / Groups Governance

SharePoint Governance

Power Platform Governance

Identity & Access Governance (+ Guests)

Stream Governance

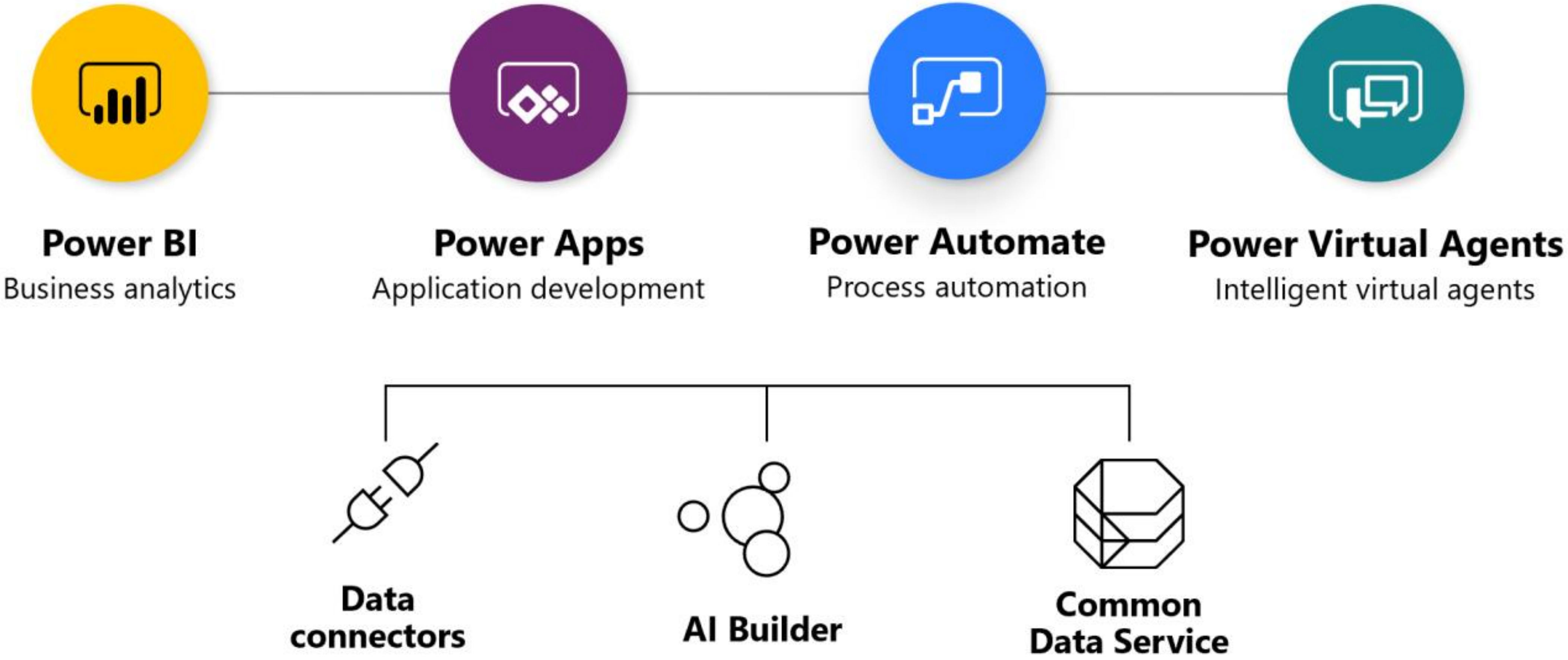
(+ Live events)

Information Lifecycle Mgmt & Protection

Business Continuity

Microsoft Power Platform

The low-code platform that spans Office 365, Azure, Dynamics 365, and standalone applications



Power BI
Business analytics

Power Apps
Application development

Power Automate
Process automation

Power Virtual Agents
Intelligent virtual agents

Data connectors

AI Builder

Common Data Service

atwork

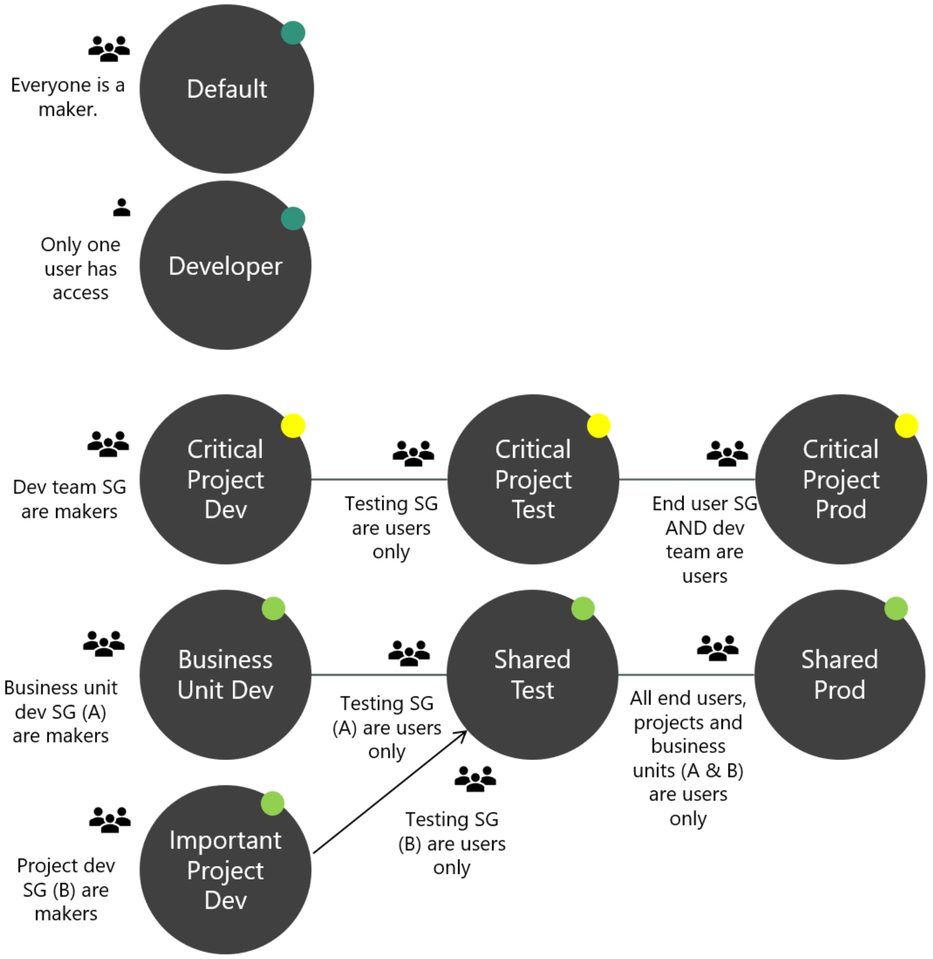
PPAC

The new admin portal

Best Practices

Environments Strategy	Types of Environments Dev / Test / Production Segregation
Policies for Connectors	Data Allowed / Blocked Standard / Premium
Activity Logs and Analytics	Usage Analysis
Automatic Audit processes	Outdated / Orphaned Flows and Apps App Classification
Makers Onboarding	Trainings & Templates

Example environment strategy diagram



Everyone is a maker.

Default

Only one user has access

Developer

Dev team SG are makers

Critical Project Dev

Testing SG are users only

Critical Project Test

End user SG AND dev team are users

Critical Project Prod

Business unit dev SG (A) are makers

Business Unit Dev

Testing SG (A) are users only

Shared Test

All end users, projects and business units (A & B) are users only

Shared Prod

Testing SG (B) are users only

Project dev SG (B) are makers

Important Project Dev

Communicate with everyone that **Default** is not for development of critical apps.

Developer environments are completely locked for any other user except the user who subscribed to the community plan. Applications can be moved out of the environment if needed.

Dedicated dev/test/prod environments for each critical application. Developers have Environment Maker access in the dev environment, but only user access in test and prod. End users only have end user access to the production solution so no one can modify the applications.

Shared test/prod environments for important but medium complex apps can be shared between multiple projects or business units.
Individual projects and business units should always have their own development environment to protect data. End users only have basic user access to solutions and data in production environments.

● Critical project
● Individual project
● Personal productivity apps

SG = Security Group

https://powerapps.microsoft.com/de-de/blog/establishing-an-environment-strategy-for-microsoft-power-platform/

atwork

Managing environments and apps with PowerShell

Automation of the Power Platform

Install the Power Platform PowerShell modules

```
# PowerShell in this topic requires PowerShell version 5.x.
Install-Module -Name Microsoft.PowerApps.Administration.PowerShell
Install-Module -Name Microsoft.PowerApps.PowerShell -AllowClobber

# Get static analysis checks of solutions against a set of best practice rules
Install-Module Microsoft.PowerApps.Checker.PowerShell

# check the latest installed version
Get-InstalledModule -Name Microsoft.PowerApps.Administration.PowerShell,
 Microsoft.PowerApps.PowerShell
```

Sign-in

```
# Ask for credentials
Add-PowerAppsAccount
```

OR

```
# Alternatively: You can pass in credentials (avoiding opening a prompt)
# This works only without MFA.
$pass = ConvertTo-SecureString "password" -AsPlainText -Force
Add-PowerAppsAccount -Username foo@bar.com -Password $pass
```

Sign-in

```
# Ask for credentials
Add-PowerAppsAccount
```

OR

```
# Alternatively: You can pass in credentials (avoiding opening a prompt)
# This works only without MFA.
$pass = ConvertTo-SecureString "password" -AsPlainText -Force
Add-PowerAppsAccount -Username foo@bar.com -Password $pass
```

Sample to get the environment

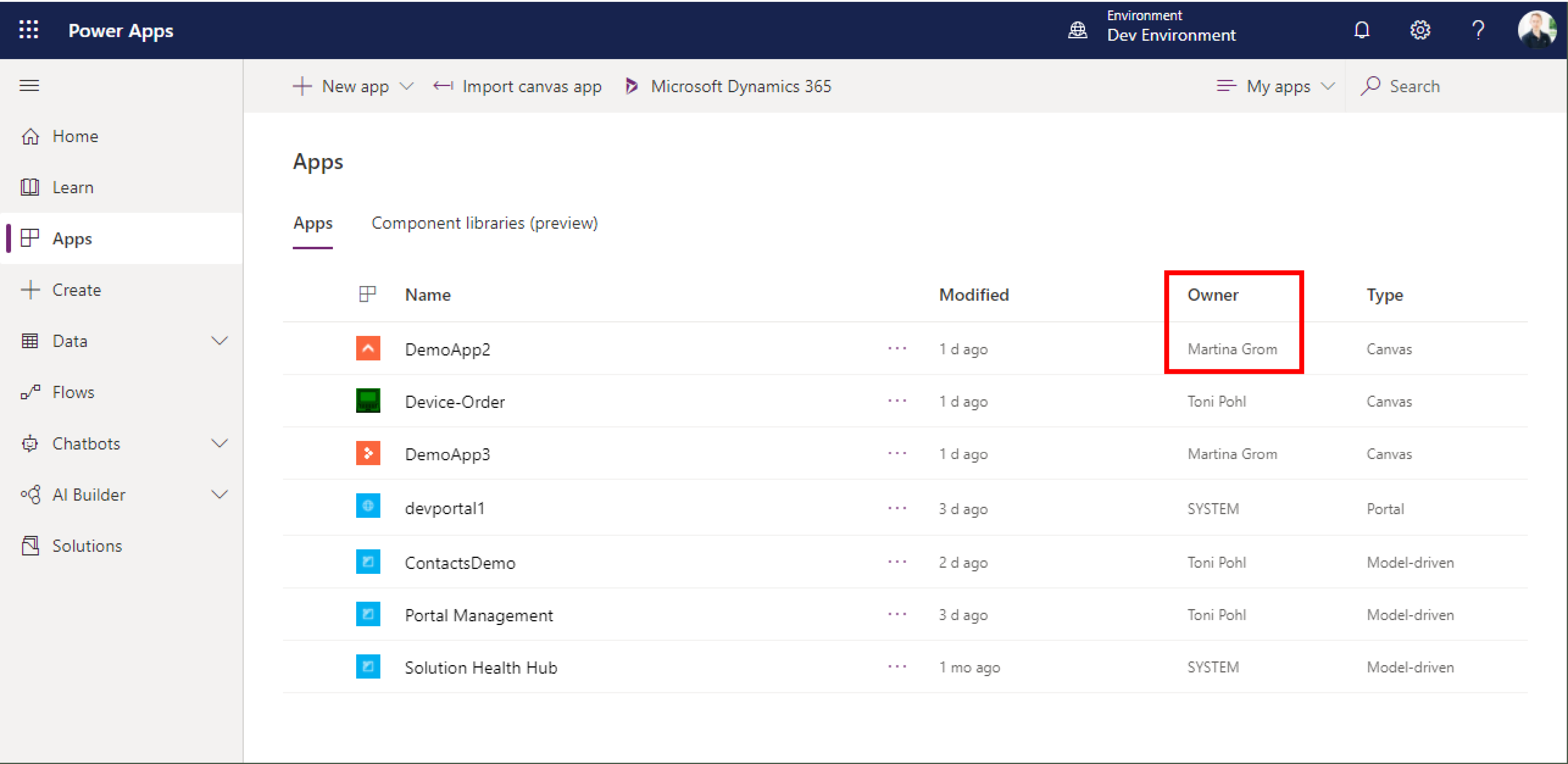
```
# Get the environment
Get-AdminPowerAppEnvironment | ft

$environment = Get-AdminPowerAppEnvironment -Default
OR
$environment = Get-AdminPowerAppEnvironment -EnvironmentName 'a123…'

# $environmentid = $environment.Internal.id
# We need to extract the ID only from
# "/providers/Microsoft.BusinessAppPlatform/scopes/admin/environments/Default-<someid>"
$environmentid = $environment.Internal.id.split("/")[-1]
Write-Output "Environment ID: $environmentid"
```

Sample to get all apps in one enviroment

```
$apps = Get-AdminPowerApp | Where-Object {$_.EnvironmentName -eq $environmentid }

$apps | Format-Table -Property DisplayName, CreatedTime, EnvironmentName
$apps.count

# Get one specific app by AppName
$myapp = $apps | Where {$_.DisplayName -eq "DMyApp"  }
$myapp.AppName
```

Modify app permissions

```
# Get the permissions for a specific app (prev. Get-AdminAppRoleAssignment)
Get-AdminPowerApp -AppName $myapp.AppName -EnvironmentName $environmentid `
    | Get-AdminPowerAppRoleAssignment  `
    | Select PrincipalDisplayname, PrincipalObjectId, RoleType, AppName `
    | ft

# Add a user. We need the UserID from Azure AD.
$userid = "c345…"
# Add permissions to an app = Share the app.
# The user gets an email if -Notify $true (default)
Set-AdminPowerAppRoleAssignment -PrincipalType Group  `
      -PrincipalObjectId $userid -RoleName CanEdit `
      -AppName $myapp.AppName -EnvironmentName $environmentid -Notify DoNotNotif
```



Name	Modified	Owner	Type
DemoApp2	1 d ago	Martina Grom	Canvas
Device-Order	1 d ago	Toni Pohl	Canvas
DemoApp3	1 d ago	Martina Grom	Canvas
devportal1	3 d ago	SYSTEM	Portal
ContactsDemo	2 d ago	Toni Pohl	Model-driven
Portal Management	3 d ago	Toni Pohl	Model-driven
Solution Health Hub	1 mo ago	SYSTEM	Model-driven

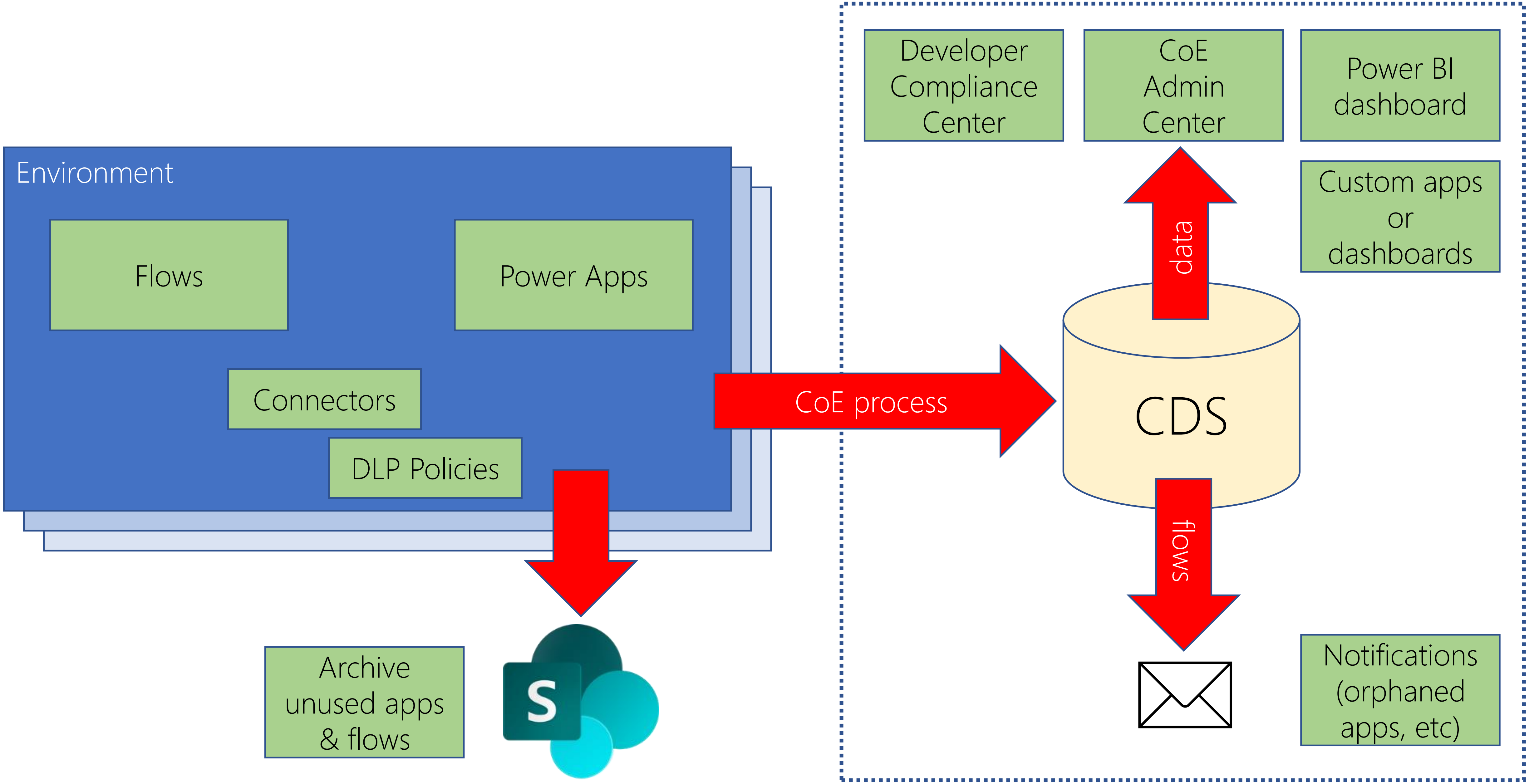
Change the owner of an app

```
# Change owner: Set-AdminPowerAppOwner (previously Set-AdminAppOwner)

Set-AdminPowerAppOwner -AppName $myapp.AppName `

        -AppOwner $userid `

        -EnvironmentName $environmentid
```

Center of Excellence (CoE)

Microsoft Add-On for governance of the Power Platform

CoE Architecture


Environment
Flows
Power Apps
Connectors
DLP Policies
CoE process
CDS
data
flows
Developer Compliance Center
CoE Admin Center
Power BI dashboard
Custom apps or dashboards
Notifications (orphaned apps, etc)
Archive unused apps & flows

atwork

CoE Demo

Components of the starter kit & Walkthrough

Offering

Phase 1: Analyse und Planung

- Architektur, Lizenzierung und Security
- Environments, Connectors, Common Data Services, Data Loss Prevention (DLP)
- Detaillierter Plan für die Verwendung der Komponenten

Phase 2: Aufbau des CoE

- Aufbau des Power Platform Center of Excellence (CoE)
- Monitoring, Alerts und Deployment (DevOpps)
- Übersicht über die bereits verwendeten Power Apps und Kontrolle über neu erstellte Lösungen

Phase 3: End User Adoption

- Wissentransfer und Trainings der End User für Best Practices und Policies im Unternehmen
- Workshops, Community Aufbau, Champions Programm
- End User wissen über die sichere Verwendung der Power Platform Bescheid

Summary

Governance Tools for the Power Platform

- Develop a governance plan & strategy
- Admin Center
- Manage with PowerShell
- Center of Excellence

Call to action!

Contact us and see our
M365 governance solutions
www.atwork.at
www.governancetoolkit365.com
https://github.com/tonipohl/PowerPlatformGovernance