



Never forget another Microsoft Intune administrative task by using low-code solutions

Peter van der Woude & Django Lohn



Peter van der Woude

Principal Consultant @ InSpark

Enterprise Mobility MVP | Windows Insiders MVP

Family | Groningen | Basketball | Gaming



petervanderwoude.nl



@pvanderwoude



peterwoude



pvanderwoude@hotmail.com



cegeka



INSPARK



kpn
Partner Network



PATCH
MY PC



PINK



YDENTIC



Django Lohn

Consultant @ InSpark
Business Applications MVP



knowhere365.space



@LohnDjango



djangolohn



DjangoLohn@outlook.com





Agenda

- Setting the stage
- Addressing the challenge
- Demo the solution
- Summarizing the session





Doing that by merging worlds

- Logic Apps meets Power Automate
- IT Pro meets Citizen Developer
- Peter meets Django

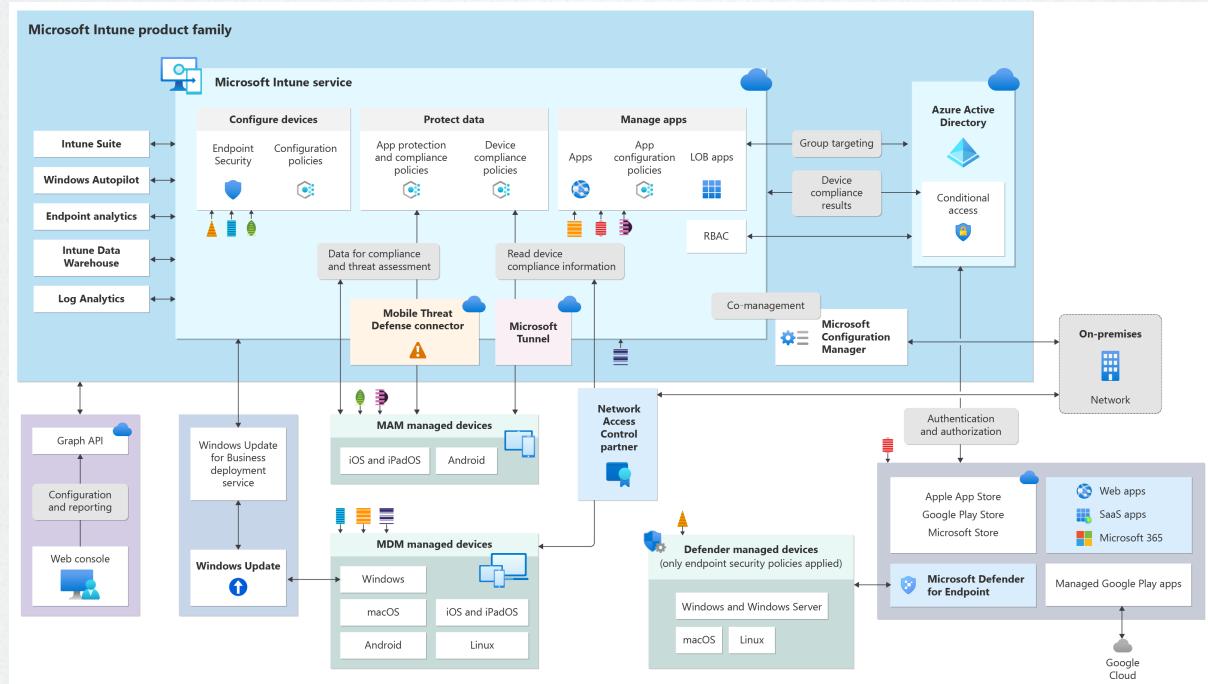




Setting the stage



What's missing in the Microsoft Intune architecture?



Power Platform as the missing link



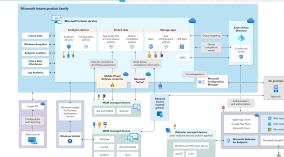
Microsoft 365



Microsoft Power Platform



Microsoft Azure



Identity, security, management and compliance by design





Importance of administrative tasks

Prevent stuff from breaking



Prevent stuff from becoming a mess



Administrative tasks to think about

The image displays a collage of five screenshots from the Microsoft Intune admin center, illustrating various administrative tasks:

- Connectors:** Shows the connectors section with options for Apple VPP Tokens, Android and Chrome OS, Managed Google Play, Chrome Enterprise, Cross platform, Microsoft Defender for Endpoint, Mobile Threat Defense, Partner device management, Partner compliance management, TeamViewer connector, ServiceNow connector, Certificate connectors, Telecom expense management, and Derived Credentials.
- Devices | Enroll:** Shows the devices enrollment section with options for Overview, All devices, Device onboarding, Cloud PC creation, Enrollment, Manage devices, Configuration, Compliance, Conditional access, Scripts, Windows 10 and later updates, Apple updates, Group Policy analytics, eSIM cellular profiles, Policy sets, and Device clean-up rules.
- Devices | Comply:** Shows the devices compliance section with options for Overview, All devices, Device onboarding, Cloud PC creation, Enrollment, Manage devices, Configuration, Compliance, Conditional access, Scripts, Windows 10 and later updates, Apple updates, Group Policy analytics, eSIM cellular profiles, Policy sets, and Device clean-up rules.
- Groups | All:** Shows the groups section with options for All groups, Deleted groups, Diagnose and solve problems, Settings, General, Expiration, Naming policy, Activity, Access reviews, Audit logs, Bulk operation results, Troubleshooting + Support, and New support request.
- Autopilot deploy:** Shows the autopilot deployment section with options for Tenant status, Remote Help, Microsoft Tunnel Gateway, Connectors and tokens, Filters, Roles, Azure AD Privileged Identity Management, Diagnostics settings, Audit logs, Device diagnostics, Multi Admin Approval, Alerts (preview), Intune add-ons, and End user experiences.

Tenant admin | Multi Admin Approval

Received requests

Requested on	Resource type	Operation	Business Justification	Requested by
2/17/2023, 11:48:30 AM	Remediation script	Delete	Start over.	admin@pvdw.onmicrosoft.com
1/20/2023, 5:23:46 PM	Remediation script	Update	Wrong settings configuration.	admin@pvdw.onmicrosoft.com
1/20/2023, 5:17:45 PM	Remediation script	Update	Better name!	pvdw.vanderwoude_a@petervanderwoude.nl
1/19/2023, 8:39:10 PM	Remediation script	Update	Minor adjustment	pvdw.vanderwoude_a@petervanderwoude.nl
12/19/2022, 8:42:01 PM	Remediation script	Assign	Need it	admin@pvdw.onmicrosoft.com
12/19/2022, 8:40:01 PM	Powershell script	Update	Test	admin@pvdw.onmicrosoft.com
12/19/2022, 8:39:25 PM	Powershell script	Update	Test	admin@pvdw.onmicrosoft.com





Effect of never forgetting Microsoft Intune administrative tasks

- Proactively addressing issues
- More secure environment
- A cleaner environment
- More insights

Importance of citizen developers

Citizen Developer (End-User)	Citizen Developer (Power User)	Business-Led Pro Developer	Enterprise IT Pro Developer
			
Full-Time Developer	No	No	Yes
Preferred Tools	"No-code" (configuration)	Low-code	Low-code and Pro-code
Typical Apps	Individual and workgroup	Workgroup	Departmental
			Enterprise

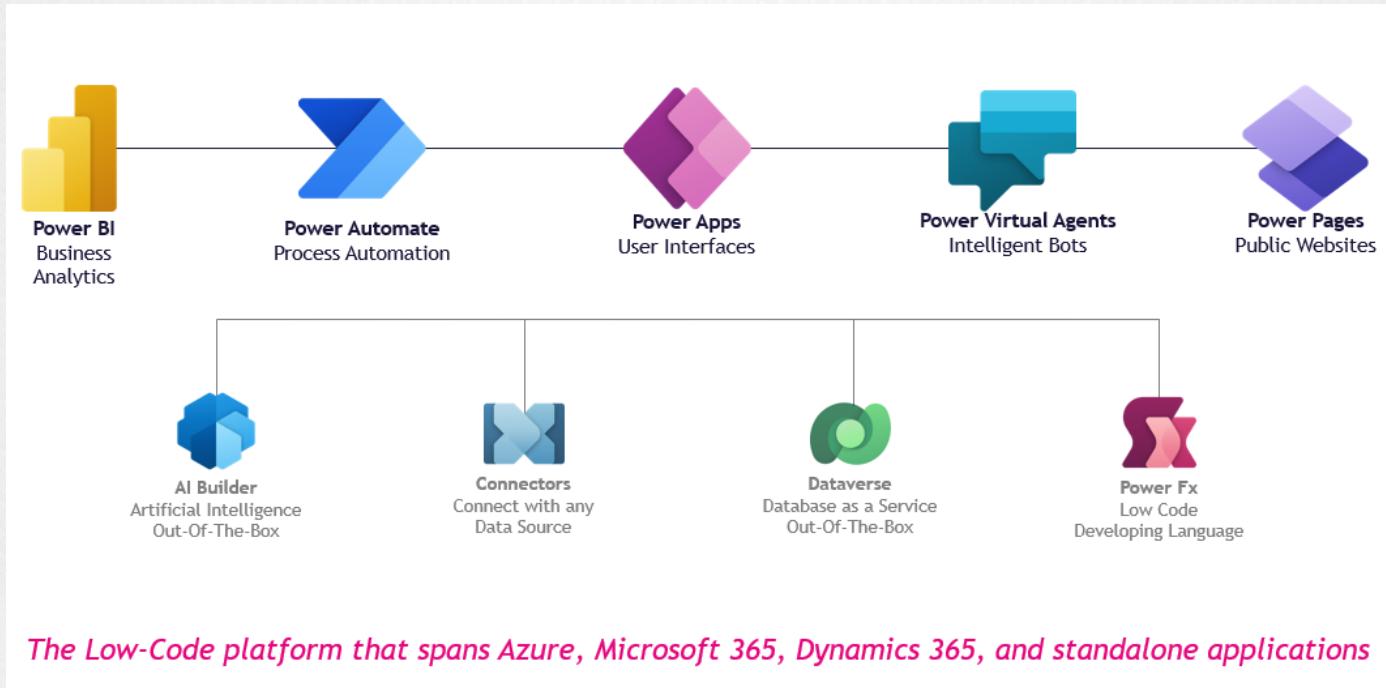
Source: <https://blogs.gartner.com/jason-wong/importance-citizen-development-citizen/>



Addressing the challenge



Power Platform to the rescue





Why this magical combination?

- Some processes / connections / end-points require advanced (centralized) management
- IT versus Business → IT and Business
"Do more with less..."
- Own Logging Logic
- Application Lifecycle Management (ALM)



Possible tools for our challenge



Microsoft 365



Microsoft Power Platform



Microsoft Azure



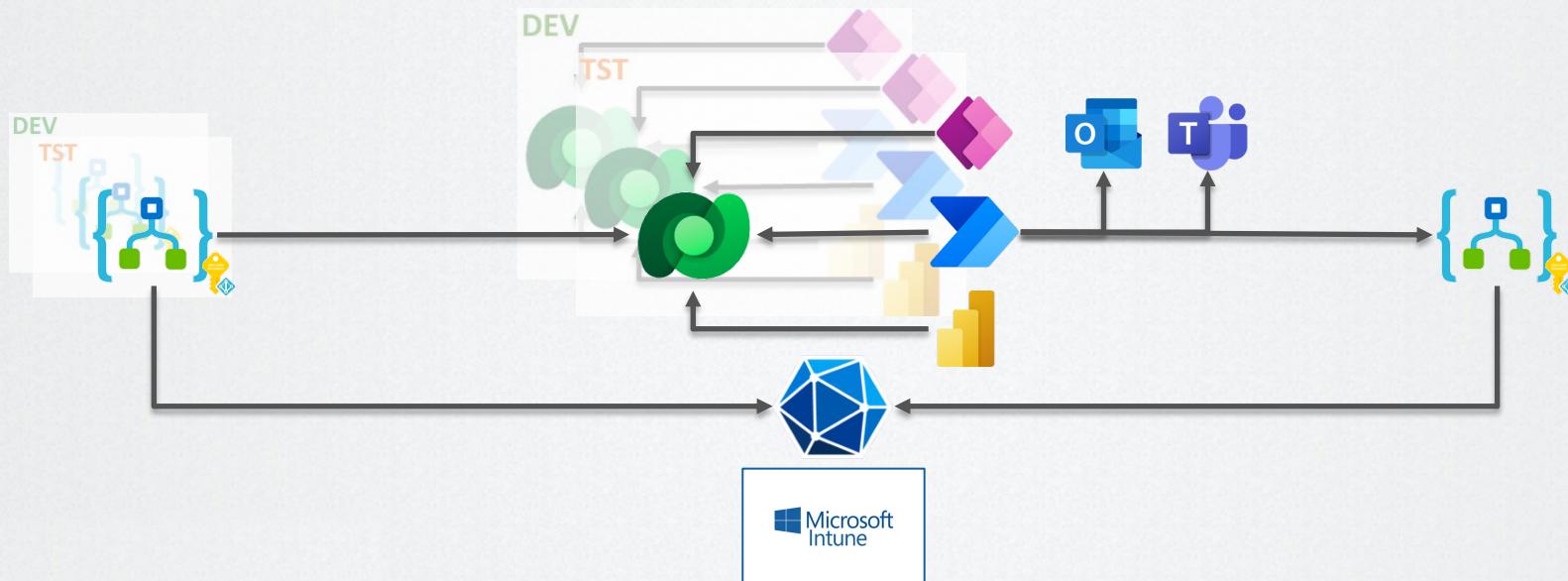
Identity, security, management and compliance by design

Quick technical comparison

	Power Automate	Logic Apps
Users	Office workers, business users, SharePoint administrators	Pro integrators and developers, IT pros
Scenarios	Self-service	Advanced integrations
Design tool	In-browser and mobile app, UI only	In-browser, Visual Studio Code , and Visual Studio with code view available
Application lifecycle management (ALM)	Design and test in non-production environments, promote to production when ready	Azure DevOps: source control, testing, support, automation, and manageability in Azure Resource Manager
Admin experience	Manage Power Automate environments and data loss prevention (DLP) policies, track licensing: Admin center	Manage resource groups, connections, access management, and logging: Azure portal
Security	Microsoft 365 security audit logs, DLP, encryption at rest for sensitive data	Security assurance of Azure: Azure security , Microsoft Defender for Cloud , audit logs

Source: [Integration and automation platform options in Azure | Microsoft Learn](#)

Possible process for our challenge





Demo

Working with non-compliant devices in Microsoft Intune



Demo prerequisite: Graph call

The screenshot shows the Microsoft Intune admin center interface. On the left, there's a sidebar with various icons and a main pane titled "Devices | Compliance". In the main pane, there are tabs for "Monitor", "Policies", "Notifications", and "Retire noncompliant devices", with "Retire noncompliant devices" being the active tab. Below these tabs are buttons for "Refresh", "Export", "Column", "Clear all devices retire state", and "Retire". A search bar and a filter button are also present. The main list area shows a table with columns for "Device name", "Compliance", "OS", and "Ownership". One row is visible, showing "pvanderwoude_AndroidForWo..." under Device name, "Not comp...", "AndroidW...", and "Personal" under Ownership.

A network traffic capture tool is overlaid on the right side of the screen, showing a single request. The request URL is `https://graph.microsoft.com/beta/deviceManagement/deviceCompliancePolicies/getNoncompliantDevicesToRetire`. The request method is POST, with a status code of 200 OK. The remote address is 40.126.32.99:443. The referer policy is strict-origin-when-cross-origin. The response headers include Access-Control-Allow-Origin: *, Access-Control-Expose-Headers: ETag, Location, Preference-Applied, Content-Range, request-id, client-request-id, ReadWriteConsistencyToken, Retry-After, SdkVersion, WWW-Authenticate, x-ms-client-gcc-tenant, client-request-id: 416b2b23-2841-4879-90bf-7dc3a50286b, Content-Type: application/octet-stream, Date: Mon, 15 May 2023 06:58:03 GMT, and request-id: b59a1c5a-0886-40fc-85c5-53a78893f37.

Request URL: `https://graph.microsoft.com/beta/deviceManagement/deviceCompliancePolicies/getNoncompliantDevicesToRetire`

Request Method: POST

Status Code: 200 OK

Remote Address: 40.126.32.99:443

Referer Policy: strict-origin-when-cross-origin

Response Headers

- Access-Control-Allow-Origin: *
- Access-Control-Expose-Headers: ETag, Location, Preference-Applied, Content-Range, request-id, client-request-id, ReadWriteConsistencyToken, Retry-After, SdkVersion, WWW-Authenticate, x-ms-client-gcc-tenant
- client-request-id: 416b2b23-2841-4879-90bf-7dc3a50286b
- Content-Type: application/octet-stream
- Date: Mon, 15 May 2023 06:58:03 GMT
- request-id: b59a1c5a-0886-40fc-85c5-53a78893f37

Demo prerequisite: Permissions

The screenshot shows two browser windows. The left window is the Microsoft Graph REST API Beta documentation page for 'Permissions'. It lists various permission types (Delegated, Application) and their descriptions. The right window is the Microsoft Graph Explorer tool. In the Explorer, a query is being run: `https://graph.microsoft.com/beta/deviceManagement/deviceCompliancePolicies/getDevicesScheduledToRetire`. The results show that two permissions are required: `DeviceManagement.ReadWrite.All` and `DeviceManagement.Read.All`. Both permissions are listed as having 'Yes' for 'Admin consent required' and 'Unconsent' for 'Consent type'. The response preview shows JSON data related to device retirement.

Permissions

One of the following permissions is required to run the query. If possible, consent to the least privileged permission.

Permission	Description	Admin consent required	Status	Consent type
DeviceManagement.ReadWrite.All	Allows the app to read properties of Microsoft Intune-managed device configuration and device compliance policies and their assignment to groups.	Yes	Unconsent	Principal
DeviceManagement.Read.All	Allows the app to read and write properties of Microsoft Intune-managed device configuration and device compliance policies and their assignment to groups.	Yes	Unconsent	Principal

Request body

Response preview

```
{ "@odata.context": "https://graph.microsoft.com/beta/$metadata#Collection(microsoft.graph.retireScheduledManagedDevice)", "@odata.count": 1, "value": [ { }
```

Demo prerequisite: Apply permissions

The screenshot shows a PowerShell script in a VS Code editor. The script connects to Microsoft Graph, retrieves a managed identity, and assigns a role to it. Two annotations highlight specific parts of the code:

- An annotation with a blue border and the text "Managed Identity of your Logic app" points to the line where a managed identity ID is assigned: `$managedIdentityId = "4d3199b5-a687-4945-bf50-85d0c47acdca"`.
- An annotation with a blue border and the text "Permissions for your Graph call" points to the line where a role is assigned to the service principal: `New-MgServicePrincipalAppRoleAssignment -ServicePrincipalId $managedIdentityId -PrincipalId $managedIdentityId -AppRoleId $role.Id`.

```
Connect-MgGraph Untitled-1
1 Connect-MgGraph
2
3 $managedIdentityId = "4d3199b5-a687-4945-bf50-85d0c47acdca"
4 $roleName = "DeviceManagementConfiguration.Read.All"
5
6 $msGraph = Get-MgServicePrincipal -Filter "AppId eq '00000003-0000-0000-c000-000000000000'"
7 $role = $msGraph.AppRoles | Where-Object {$_ .Value -eq $roleName}
8
9 New-MgServicePrincipalAppRoleAssignment -ServicePrincipalId $managedIdentityId -PrincipalId $managedIdentityId -AppRoleId $role.Id
10
11 Disconnect-MgGraph
```

Below the code editor, the VS Code interface includes:

- PROBLEMS, OUTPUT, TERMINAL, DEBUG CONSOLE tabs.
- A status bar at the bottom showing: PowerShell Extension v2023.2.1, Copyright (c) Microsoft Corporation, https://aka.ms/vscode-powershell, Type 'help' to get help., PS C:\Users\ (redacted), Ln 3, Col 59, Spaces: 4, UTF-8, CRLF, PowerShell.



Summarizing the session





Benefits of using low-code solutions

- Logic apps and Graph: A powerful combination!
- Segregation of duties
- "*Think big, act small*"
- Enterprise worthy Application Lifecycle Management options with Power Platform (DEV – TST – PRD)



To remember

- Automating administrative tasks Intune help with
 - proactively addressing issues,
 - more secure environment,
 - a cleaner environment,
 - and more insights
- Never forget: "*A fool with a tool is still a fool*"
- Don't make the Power Platform your next Microsoft Access, or your next Macro's
- Citizen developers can help with addressing the shortage of IT resources



In the end
we want
everybody
to be
happy







15:30 – 16:20

Building a PAW for and through the cloud

- Michael Van Horenbeeck

