MICROSOFT POWER PLATFORM · PORTFOLIO CASE STUDY

IT Helpdesk **Assistant**

Authenticated ticket management - conversational intake, idempotent notifications, SLA watcher, AI summarisation, and Teams Adaptive Card write-back.

Leila Marchant
Power Platform & Copilot Studio Developer

AZ-900 · AI-900 · PL-900
Winchester, UK · Remote / Hybrid

4	2	<15 min	FLS
Power Automate · Flows	Auth · Modes	MTTA · Target	Field-Level · Security

ES EXECUTIVE SUMMARY

IT helpdesk operations fail in predictable ways. High volumes of low-priority tickets bury critical issues. Managers are notified multiple times for the same incident. SLA breaches go undetected until a user escalates. There is no reliable audit trail of who triaged what, or when.

This project replaces a manual, reactive helpdesk process with a structured, authenticated, SLA-aware business application built on the Microsoft Power Platform. Copilot Studio handles conversational ticket intake. Four decoupled Power Automate flows manage creation, manager notification, status enquiry, and escalation. Dataverse is the single system of record. A scheduled SLA watcher monitors open tickets autonomously - escalating to a director when a threshold is breached. Teams Adaptive Cards allow managers to act on tickets without leaving their channel.

Version 2 of the solution adds full Entra ID authentication - capturing the user's verified identity at the point of submission rather than relying on typed data. Field-Level Security protects sensitive decision fields so only authorised roles can write resolution outcomes. An AI summarisation child flow generates a concise ticket summary using Azure OpenAI via the BYOM pattern, with a safe fallback so the main workflow is never blocked by a model failure.

> **KEY DIFFERENTIATOR**
> The architecture is decoupled by design. The public intake layer (Copilot Studio and ticket creation) is completely separated from the private operations layer (manager notifications, SLA watcher, director escalation). Each layer can fail, retry, or be upgraded independently - the way production enterprise systems are built.

Technology Stack

Copilot Studio	Power Automate	Dataverse	Teams Adaptive Cards	Entra ID	Azure OpenAI	Field-Level Security	PAC CLI

01 BUSINESS PROBLEM & DESIGN GOALS

The core problem is not volume - it is signal-to-noise. When every ticket looks the same in an inbox, critical issues do not get treated differently from password resets. The result is SLA breaches, frustrated users, and managers who cannot see what is actually open and overdue.

- Unstructured intake - tickets submitted by email or chat with inconsistent detail
- No triage logic - critical and low-priority requests handled in the same queue
- Duplicate manager notifications - the same ticket triggers multiple messages when flows retrigger
- No SLA visibility - overdue tickets are only identified when users chase manually
- No audit trail - no record of who was notified, when escalation happened, or what decision was made
- Identity risk - user identity in the ticket is whatever the user typed, not a verified source

The design goal was to build a system that routes structured, identity-bound tickets to the right person immediately, maintains itself autonomously against SLA targets, and leaves a complete audit trail - without requiring process change from the end user.

Design Principles

Principle	What It Means in Practice
Decoupled layers	Public intake layer (Copilot + create flow) is independent from the private operations layer (watcher, escalation). Each has a single responsibility.
Idempotency throughout	NotificationSent and AlreadyEscalated flags prevent duplicate notifications at every stage. Retriggered flows check state before acting.
Identity-aware	Version 2 binds every ticket to an Entra ID object - not a typed email address. Escalations reference a verified identity.
Autonomous operations	The SLA watcher runs on a schedule without being prompted. Escalation is not a manual decision - it is a timed, automated consequence.
AI as enhancement, not risk	The BYOM summarisation child flow has an explicit fail-safe fallback. The main workflow completes even if the model is unavailable.

02 ARCHITECTURE OVERVIEW

The solution separates intake from operations deliberately. The agent and intake flow are designed to be fast and user-facing. The watcher flows operate asynchronously in the background, giving the solution persistent operational behaviour after the conversation ends.

Two-Layer Architecture

Layer	Responsibilities
Layer 1 - Public intake	Copilot Studio topic collects structured ticket details. V2 captures Entra ID identity at sign-in. The Helper-CreateTicket flow validates inputs, calls the AI summarisation child flow, writes the Dataverse record, and returns a confirmation.
Layer 2 - Private operations	Automation-NotifyManager watcher monitors Dataverse for new tickets where the NotificationSent flag is false. Posts a Teams Adaptive Card to the assigned manager. Manager actions write directly back to Dataverse. The SLA watcher scans for breached tickets every 15 minutes, escalating to a director with AlreadyEscalated idempotency.

Component Responsibilities

Component	Role	Layer
Copilot Studio	Conversational intake, input validation, status check, Entra ID auth (V2)	User-facing
Power Automate	Four focused flows - create, notify, check-status, escalate - plus AI summarisation child flow	Orchestration
Dataverse	Single system of record - ticket state, identity, timestamps, decisions, audit trail	Data layer
Teams Adaptive Cards	Structured manager notifications with write-back actions - decisions captured as data	Notification
Azure OpenAI (BYOM)	Ticket summarisation child flow - env var config, Key Vault secret, fail-safe fallback	AI layer
Entra ID	Verified identity capture - AadObjectId and UPN stored at ticket creation in V2	Identity

Ticket Lifecycle

1. User opens Copilot Studio. V2: authenticates via Entra ID. V1: provides email in conversation.
2. Agent collects category, priority, description, and affected system.
3. Helper-CreateTicket flow validates inputs, calls SummariseTicket_BYOM child flow, creates Dataverse record with TicketId and initial state.
4. Automation-NotifyManager watcher detects the new ticket (NotificationSent = false), posts Teams Adaptive Card to the assigned manager.

5. Manager acts on the card - resolution or escalation action writes directly back to Dataverse. NotificationSent is set to true.

6. SLA watcher runs every 15 minutes. Tickets breaching the SLA threshold with no resolution trigger director escalation (AlreadyEscalated prevents duplicate escalations).

7. User can query ticket status at any time via the Helper-CheckStatus flow.

03 DATAVERSE SCHEMA: STATE AS AN AUDIT TRAIL

The data model is defined before any flow is built. Every ticket event - creation, notification, manager decision, SLA breach, escalation, resolution - is represented by a dedicated Dataverse column. This makes the entire system observable and supportable without needing flow admin access.

DESIGN DECISION

Using dedicated columns for idempotency flags (NotificationSent, AlreadyEscalated) instead of relying on flow state or run history is the pattern that makes multi-trigger architectures safe. It costs two fields per concern and eliminates a class of duplicate-notification bugs entirely.

Key Fields by Category

Field	Purpose	Category
TicketId	Unique correlation ID generated at creation - ties all events to one record	Identity
SubmitterEmail / UPN	User-provided email (V1) or Entra UPN from signed-in context (V2)	Identity
SubmitterAadObjectId	Entra object ID for verified identity binding - V2 only	Identity
Category / Priority	Structured intake fields - required for routing and SLA calculation	Intake
Description / AiSummary	Raw description and AI-generated operational summary from SummariseTicket_BYOM	Intake
Status / StatusReason	Authoritative lifecycle state driving all watcher logic	Lifecycle
SubmittedOn / ResolvedOn	Audit timestamps for submission and resolution	Lifecycle
NotificationSent / SentOn	Idempotency flag - ensures exactly one manager notification per ticket	Idempotency
AlreadyEscalated / EscalatedOn	Escalation idempotency - prevents duplicate director escalations	Idempotency
ManagerDecision / DecisionNotes	Structured manager action - captured as data, not as a chat message	Decision
SlaBreached / SlaBreachedOn	SLA breach detection flags written by the watcher at point of detection	SLA
EscalationError / NotifyError	Structured error capture - failures written to Dataverse, not lost in run history	Error Handling

04 COPILOT STUDIO: STRUCTURED INTAKE & AUTHENTICATION

The agent topic provides a consistent, structured intake experience. The goal is to collect exactly the information needed to route and prioritise the ticket - nothing more, nothing less. Free-text ticket descriptions submitted over email are replaced with a validated, categorised record.

Structured Input Collection

- Ticket category - Hardware, Software, Network, Account, or Other
- Priority level - Low, Medium, High, or Critical
- Description of the issue
- Affected system or application

Priority is collected as a structured choice, not inferred from free text. This eliminates the interpretation error that causes low-priority tickets to be treated as critical (and vice versa) in manual triage.

GetStatus Topic

A dedicated status-check topic allows users to query the current state of their ticket by TicketId at any point. This queries Dataverse directly via the Helper-CheckStatus flow and returns real-time lifecycle state without requiring a support desk interaction.

> **V2: ENTRA ID AUTHENTICATION**
>
> Version 2 requires the user to sign in before the intake topic begins. The user's Entra ID AadObjectId and UPN are captured from the authenticated context - not from what the user types. This means every ticket is legally attributable to a verified organisational identity. The agent does not ask for an email address in the authenticated variant.

05 POWER AUTOMATE: FOUR FOCUSED FLOWS

Rather than a single orchestration flow, the solution uses four focused flows - each with one clearly defined responsibility. A fifth flow handles AI summarisation as a reusable child. This separation makes each flow independently testable, debuggable, and upgradeable.

Flow	Responsibility
Helper-CreateTicket	Called by Copilot Studio on ticket submission. Validates inputs, generates a TicketId, calls SummariseTicket_BYOM child flow, creates the Dataverse record, returns structured confirmation to the agent.
SummariseTicket_BYOM (Child)	Reusable AI summarisation child flow. Calls Azure OpenAI HTTP action with endpoint, deployment name, API version, and system prompt all driven by Dataverse environment variables. Returns a summary string; falls back to a safe default if the model call fails.
Helper-CheckStatus	Called by Copilot Studio when the user requests a status update. Queries Dataverse by TicketId, returns current Status, StatusReason, and SubmittedOn to the agent for display in the conversation.
Helper-EscalateTicket	Available as an explicit escalation trigger. Validates that the ticket exists and is in an escalatable state before writing the escalation fields to Dataverse. Checks AlreadyEscalated flag before acting.
Automation-NotifyManager	Scheduled watcher flow. Scans Dataverse for tickets where NotificationSent = false. Posts a Teams Adaptive Card to the assigned manager. Writes NotificationSent = true and SentOn timestamp after successful delivery. Try/Catch - errors written to NotifyError field.

WATCHER PATTERN

The Automation-NotifyManager flow does not run inline with ticket creation - it runs on a schedule and detects work to do from Dataverse state. This means the notification layer is completely decoupled from the intake layer. If the watcher fails or retriggers, the NotificationSent flag prevents the manager from receiving duplicate cards.

06 BYOM SUMMARISATION: AI WITH FAIL-SAFE

AI ticket summarisation is built as a reusable child flow rather than embedded inline in the create flow. This makes the component independently testable and reusable across other flows in the solution.

Architecture Decisions

- Azure OpenAI endpoint, deployment name, API version, and system prompt are all Dataverse environment variables - configuration, not code
- The Azure OpenAI API key is stored in Azure Key Vault and retrieved at runtime - never hardcoded in the flow
- The HTTP action calls the Azure OpenAI completions endpoint with a structured prompt containing the ticket description and category
- If the model call fails, throttles, or returns an error, the child flow returns a safe fallback summary string
- The parent flow (Helper-CreateTicket) always completes regardless of AI availability - ticket creation is never blocked by a model failure
- The child flow is callable from any other flow in the solution, not just the create flow

ENTERPRISE PATTERN

Externalising the model endpoint and prompt via environment variables means the AI model can be changed, upgraded, or pointed to a different deployment without editing the flow. This is the correct enterprise pattern for managing AI dependencies across dev, UAT, and production environments.

Summary Output

The AiSummary field in Dataverse stores the generated summary. This gives the manager receiving the Teams Adaptive Card an immediate operational context for the ticket - without reading the full description. If the child flow returned the fallback, the field stores a note indicating that AI summarisation was unavailable for this ticket.

07 TEAMS ADAPTIVE CARDS & WRITE-BACK

Manager notifications are delivered as Teams Adaptive Cards rather than plain messages. The card displays structured ticket information - ID, category, priority, AI summary, submitter identity - and provides action buttons the manager can use to respond directly from Teams.

Card Actions

- Acknowledge - marks the ticket as in progress, records the manager action in Dataverse
- Resolve - captures the resolution decision and notes, writes to ManagerDecision and DecisionNotes fields
- Escalate - writes escalation data to Dataverse and triggers the director notification branch

All manager actions write directly back to Dataverse via Power Automate. The decision is captured as structured data - not as a chat message or email thread. This makes the audit trail complete and queryable without any human follow-up to record what was decided.

> **WRITE-BACK PATTERN**
>
> The Adaptive Card posts a response payload back to a Power Automate flow when the manager clicks an action button. That flow writes the decision to Dataverse - keeping the system of record authoritative. A manager does not need to open a canvas app or a model-driven form to progress a ticket; the entire decision loop closes within Teams.

Identity in the Card

In Version 2, the manager card displays the submitter's verified Entra display name alongside their UPN - not a typed email address. This removes ambiguity about who raised the ticket and makes the notification directly actionable without needing to look up the user.

08 SLA WATCHER & DIRECTOR ESCALATION

The SLA watcher is the autonomous operations layer of the solution. It runs on a 15-minute schedule, operates without being prompted, and escalates overdue tickets without human intervention. This is the behaviour that separates an agentic system from a simple notification bot.

What the Watcher Does

- Queries Dataverse for tickets with Status = Open and SubmittedOn earlier than the SLA threshold
- For each overdue ticket where AlreadyEscalated = false, posts a Teams notification to the assigned director with full ticket context
- Sets AlreadyEscalated = true and writes EscalatedOn timestamp - preventing duplicate escalations on subsequent watcher runs
- Sets SlaBreached = true and SlaBreachedOn timestamp at the point of first detection
- Writes any watcher errors to the EscalationError Dataverse field - failures are never silent

The SLA threshold is configured via a Dataverse environment variable - not hardcoded in the flow. Different categories or priorities can be given different thresholds without editing any flow.

Director Escalation Branch

When a ticket is escalated, a separate branch of the watcher flow posts a structured Teams message to the director. The message includes the ticket ID, priority, AI summary, submitter identity (UPN in V2), and the time elapsed since submission. The director receives full context without needing to query Dataverse directly.

> **WHY THIS MATTERS**
>
> Most portfolio builds show only the happy path - a ticket is raised and resolved. The SLA watcher proves the system continues to operate correctly over time, without manual intervention. It demonstrates that the architecture can maintain SLA compliance autonomously - the expectation for any enterprise-grade helpdesk implementation.

09 SECURITY & GOVERNANCE

Security is implemented in layers. Each control is independently visible in platform screenshots and recordings - making the governance posture demonstrable, not just described.

Role-Based Access Control

Role	Privileges
End User	Create new tickets and read their own records only - cannot view other users' tickets
Manager	Read assigned open tickets and write resolution decisions - cannot create or delete records
Director	Read escalated tickets and view operational dashboard - no edit access to ticket fields
Admin / Operations	Full operational visibility, record management, and environment configuration
Automation (Service)	Least-privilege service identity for flow operations - scoped to required tables only

Field-Level Security

Version 2 applies Field-Level Security (FLS) profiles to the most sensitive decision fields - ManagerDecision, DecisionNotes, EscalationError, and AiSummary. These fields can only be written by the Automation service principal and the Manager role. End users cannot read or modify resolution decisions.

FLS is configured at the Dataverse column level via Security Profiles - independent of table-level permissions. This provides a fine-grained governance layer that RBAC alone cannot enforce.

Idempotency Controls

NotificationSent and AlreadyEscalated are explicit boolean flags on every ticket record. Every flow that sends a notification checks these flags before acting. If a flow retriggers due to a Dataverse platform event or a scheduled watcher overlap, the flags prevent duplicate cards or escalation messages from being sent.

Error Capture

All watcher flows use Try/Catch patterns. Notification and escalation errors are written to dedicated Dataverse fields (NotifyError, EscalationError) rather than disappearing into flow run history. A support engineer can identify and investigate any failed notification from a Dataverse record view - without needing flow admin access.

10 AUTHENTICATION: V1 PUBLIC vs V2 ENTRA ID

Both versions of the solution are fully functional. The choice of authentication mode is a deployment decision, not a capability gap.

Mode	Behaviour
V1 - Public / Demo	No sign-in required. The agent collects the user's email address as a conversation input. Lower friction for demos and portfolio reviews. Identity in Dataverse is whatever the user typed.
V2 - Entra ID Authenticated	Requires Entra ID sign-in before the intake topic begins. AadObjectId and UPN captured from authenticated context. Identity in Dataverse is a verified organisational identity. Suitable for production enterprise deployment.

The V2 Entra ID variant also adds an authorisation check inside the Helper-CreateTicket flow - verifying that the submitting identity has the End User security role before creating a Dataverse record. This prevents anonymous or unauthorised submissions from writing to the system of record even if the agent is accessed directly.

PORTFOLIO DECISION

Building both authentication modes demonstrates the ability to architect for different deployment contexts. V1 allows recruiters to test the full experience without an organisational sign-in. V2 proves the architecture can be hardened to enterprise identity standards - including verified identity capture, FLS enforcement, and authorisation checks - without redesigning the conversation flow.

11 CHALLENGES SOLVED

1. Eliminating duplicate manager notifications

The most common failure mode in trigger-based notification architectures is duplicate messages when a flow retriggers. Solved with an explicit NotificationSent flag and sent timestamp. The watcher checks the flag before sending - even if Dataverse triggers the flow multiple times for the same record.

2. Preventing duplicate director escalations

The SLA watcher runs every 15 minutes. Without an idempotency control, a single overdue ticket would trigger a director escalation on every watcher run until resolved. The AlreadyEscalated flag is set at first escalation - subsequent runs skip the ticket entirely.

3. Making AI summarisation safe for production

Embedding AI calls directly in the create flow makes the entire ticket submission fragile if the model throttles or is unavailable. Solved by building summarisation as a child flow with an explicit Try/Catch and a fallback return value. The parent flow always completes - AI is an enhancement, not a dependency.

4. Capturing verified identity without disrupting the experience

Collecting identity from typed input introduces data quality risk and opens the door to impersonation. V2 solves this by capturing AadObjectId and UPN from the Entra sign-in context before the conversation starts - the user never types an email address.

5. Protecting decision data from unauthorised writes

Table-level RBAC does not prevent an End User from attempting to write to ManagerDecision or DecisionNotes via direct API calls. Field-Level Security profiles applied at the column level close this gap - only the Automation service identity and the Manager role can write to those fields.

12 TESTING APPROACH

The solution was tested against explicit scenarios, validating both the conversational behaviour and the resulting Dataverse state after each action.

- Ticket creation - V1 public variant, all categories and priorities
- Ticket creation - V2 Entra ID variant, identity fields verified in Dataverse post-creation
- AI summary generation - successful call, correct AiSummary write to Dataverse
- AI summary fallback - model unavailable, fallback string returned, ticket creation not blocked
- Manager notification - Teams Adaptive Card posted on first watcher run
- Idempotency - watcher re-run after notification sent, no duplicate card posted
- Manager write-back - acknowledge, resolve, and escalate actions tested, Dataverse fields verified
- SLA breach detection - ticket aged past threshold, SlaBreached flag set on watcher run
- Director escalation - escalation card posted, AlreadyEscalated set to true
- Escalation idempotency - watcher re-run after escalation, no duplicate director message
- Status check - GetStatus topic returns correct current state for each lifecycle stage
- V2 authorisation gate - unauthenticated submission attempt blocked by flow authorisation check

13 WHAT THIS DEMONSTRATES

This case study is structured to evidence the skills a Microsoft Partner expects from a developer working on enterprise Power Platform engagements.

Skill Area	How It Is Demonstrated
Decoupled architecture	Designed a two-layer system separating public intake from private operations - each layer independently deployable and testable.
Dataverse as system of record	Authoritative lifecycle state, idempotency flags, FLS profiles, and structured error capture - not just a data store.
Power Automate patterns	Four focused flows with single responsibilities, idempotency controls, Try/Catch error handling, and child flow reuse.
Copilot Studio	Structured intake, status queries, and two distinct authentication modes in a single agent.
Teams Adaptive Cards	Manager notifications with write-back - manager decisions captured as Dataverse data, not as chat messages.
AI - enterprise pattern	BYOM Azure OpenAI via child flow with environment variable config, Key Vault secret, and explicit fail-safe fallback.
Autonomous operations	Scheduled SLA watcher - detects breaches, escalates, and maintains idempotency state without human intervention.
Field-Level Security	FLS profiles on sensitive decision fields - governance below RBAC level, applied at column level.
Entra ID authentication	Verified identity capture in V2 - AadObjectId and UPN from sign-in context, plus authorisation check in create flow.
ALM discipline	Solution packaging (ITHelpdeskPortfolio), publisher prefix (lai_), environment variables, connection references.

14 WHAT I WOULD BUILD NEXT

Scope was drawn deliberately to produce a complete, demonstrable system. These are the natural next iterations for a production deployment:

- Power Apps model-driven app - operational dashboard for the support team with filtered ticket views, SLA breach indicators, and workload assignment
- Priority-based SLA thresholds - different SLA timers for Critical, High, Medium, and Low tickets configured via environment variables, not hardcoded
- Ticket routing logic - assign tickets to specific managers or teams based on category and affected system, driven by a Dataverse routing table
- Power BI reporting layer - ticket volume, SLA compliance, category breakdown, and mean-time-to-acknowledge dashboards on Dataverse data
- Integration with IT service management tooling - write-back to ServiceNow, Jira, or Freshdesk on ticket creation and resolution via connector or HTTP action
- Full Managed Environment deployment with DLP policies, CoE dashboard integration, and audit log review process

ALM ALM & DEPLOYMENT

The solution is packaged to ship - not just to demonstrate. Every environment-specific value is externalised so the same managed artefact tested in UAT is the artefact promoted to Production.

Deployment Practices

Practice	Implementation
Solution naming	ITHelpdeskPortfolio. Publisher prefix lai_ applied consistently across all tables, columns, flows, and environment variables - no Default publisher usage.
Managed vs Unmanaged	Unmanaged solutions for development only. Only Managed solutions are exported and deployed. Managed solutions are immutable post-import - flows are not edited in UAT or Production.
Environment Variables	Azure OpenAI endpoint, deployment name, API version, system prompt, SLA threshold (in hours), and routing configuration are all environment variables. Nothing environment-specific is embedded in a flow action.
Connection References	Every connector - Dataverse, Teams, HTTP - is abstracted behind a Connection Reference. Service identity bindings are set by the importing admin at import time.
Versioning	MAJOR.MINOR.PATCH.BUILD format. PATCH increments for bug fixes, MINOR for new features (e.g. V2 authentication), MAJOR for breaking changes requiring an Upgrade operation.
Rollback	The prior managed solution zip is retained after every promotion. Production is never left without a validated prior state to return to.