Title

design, As built etc

Client

3/1/2024

BDO Digital

Fill in Author

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Author | Date | Revision Notes |
| 1.0 | Fill in Author | 3/1/2024 | Initial Design |
| 1.1 |  |  |  |
| 1.2 |  |  |  |
| 1.3 |  |  |  |

Table of Contents

[Revision History 2](#_Toc172640220)

[1 Executive Summary 4](#_Toc172640221)

[1.1 Background 4](#_Toc172640222)

[1.2 Approach 4](#_Toc172640223)

[1.3 Laptop Purchase 4](#_Toc172640224)

[1.3.1 Purchase Stratagy 4](#_Toc172640225)

[1.4 High Level Design 5](#_Toc172640226)

[1.4.1 Entra ID 5](#_Toc172640227)

[1.4.2 Autopilot 5](#_Toc172640228)

[1.4.3 Intune 5](#_Toc172640229)

[1.5 Rollout Strategy 6](#_Toc172640230)

# Overview

This document summarizes the Logic App design, flow, and external integrations based on the provided ARM template.

# Purpose

The purpose of this Logic App is to support automated workflows for mailbox delegation and lifecycle events. Key functional steps include:

* Trigger workflow via HTTP from Lifecycle.
* Execute Azure Automation runbook.
* Evaluate job status and parse results.
* Notify via Office 365 email on failure.
* Post callback to lifecycle endpoint.

# Architecture

The Logic App architecture includes connections to multiple Azure services and Microsoft 365 components. Relevant configuration and metadata include:

* Logic App: SampleLogicApp
* Region: East US
* Includes: Azure Automation, Office 365, Microsoft Graph

# Execution

The following outlines the step-by-step execution path from the initiation to conclusion of the workflow:

* Start: Trigger from external HTTP request.
* Call: Azure Automation to delegate mailbox.
* Parse: Check JSON results for success/failure.
* Notify: On failure, send email.
* Callback: Send status to lifecycle API.

# Security

Security practices applied within the Logic App include identity management, encrypted connections, and restricted API access:

* Secure connections via HTTPS.
* Managed Identity for Automation access.
* OAuth tokens for Microsoft Graph API.

# Error Handling

In the event of failed automation runs or parsing errors, the Logic App takes the following steps to gracefully manage failures:

* Branching logic to detect runbook failures.
* Compose HTML email with failure summary.
* Send failure alert via Office 365 connector.
* Trigger failure callback endpoint.

# Logic App Flow Diagram

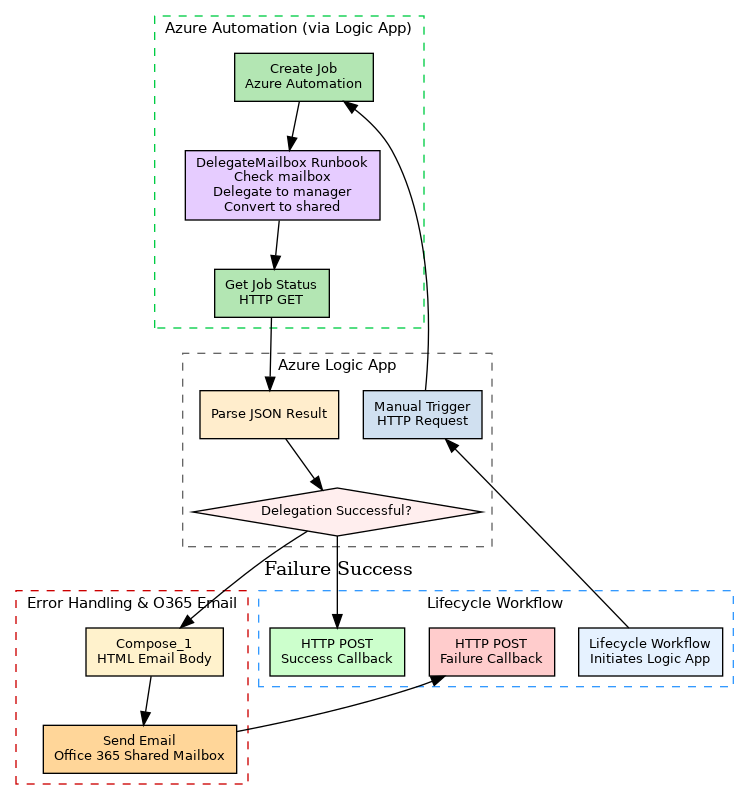


Figure 1: Workflow Execution Flow

# Data Flow Diagram

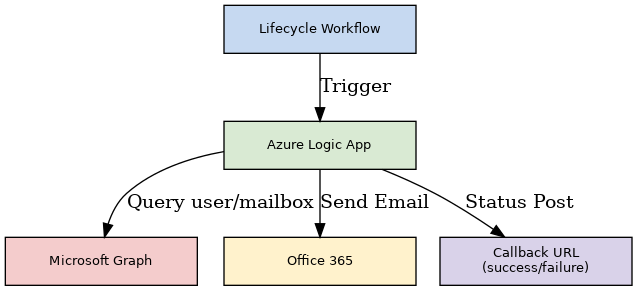


Figure 2: Data Flow Integration

# Hybrid Integration Diagram

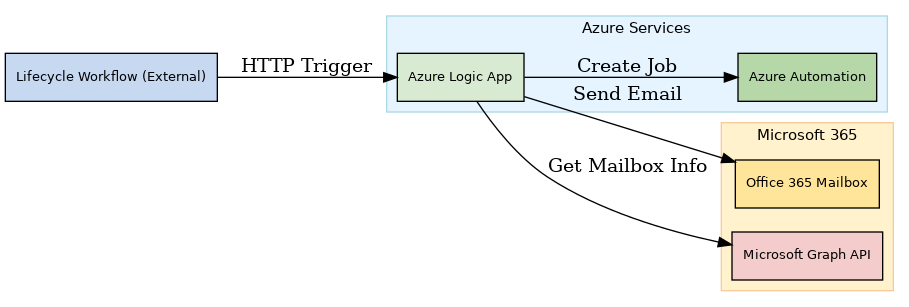


Figure 3: Cloud & M365 Integration Map

# Appendix

Helpful documentation and reference links related to Azure Logic Apps and integrated services:

* https://learn.microsoft.com/en-us/azure/logic-apps/
* https://learn.microsoft.com/en-us/graph/
* https://learn.microsoft.com/en-us/connectors/office365/