System Design Document: VSS Insights

Version	Date	Author	Changes	Status
1.0.0	26-05-25	Paras Khosla	Initial creation	Completed
2.0.0	30-05-25	Paras Khosla	Diagrams included	Completed

Table of Contents

Table of Figures

List of Abbreviations

Abbreviation	Meaning
SCOM	System Center Operations Manager
CSV	Comma-Separated Values
IIS	Internet Information Services
KPI	Key Performance Indicator
EXE	Executable File
SDK	Software Development Kit
TAM	Technical Application Manager
App Insights	Application Insights

1. Introduction

The VSS Insights Project aims to enhance the visibility of system health metrics, automate the ingestion of certificate data from CSV files, and monitor SSL/TLS certificate expirations. The project integrates a SCOM dashboard, Azure Blob Trigger Function App, Power BI dashboard, and certificate monitoring to provide a complete overview of system performance and risks.

2. Problem Statement

Before this project, there was no existing system or solution at DAF related to server health monitoring, certificate expiration tracking, or KPI visibility. Everything was done manually or communicated through email, which led to delays, missed issues, and scattered data. This project introduced separate tools for each focus area: SCOM for infrastructure monitoring, Power BI for VSC 2.0 project KPIs, and Application Insights for certificate telemetry. To improve the monitoring process by making it more organized, dependable, and user-friendly.

Prior to this project, DAF lacked a system that can help for tracking server health, monitor certificate expirations, and visualize key performance indicators (KPIs) for the VSC 2.0 project. While some monitoring and reporting existed, it was fragmented across emails and manual checks, which made it difficult sometimes to maintain consistent visibility.

This project introduced structured solutions tailored to each domain: SCOM for infrastructure monitoring, Power BI for KPI dashboards, and Application Insights for telemetry related to certificate status. The goal was to streamline and enhance the monitoring process, making it more reliable and accessible for the VSS team.

3. System Overview

The system is composed of four primary components:

- **SCOM Dashboard:** Monitors CPU, RAM, disk space, and critical alerts from servers.
- **Certificate Monitoring:** Collects IIS certificate details and logs expiry information into Application Insights.
- **Azure Function:** A part of the test automation flow, it processes CSV files uploaded to Azure Blob Storage and sends parsed test result telemetry data to Application Insights for visualization and KPI tracking.
- **Power BI Dashboard:** Visualizes VSC 2.0 KPIs such as A200 errors, ECU programming durations, and certificate-related metrics using telemetry pulled from Application Insights.

3.1 High-Level System Architecture Diagram

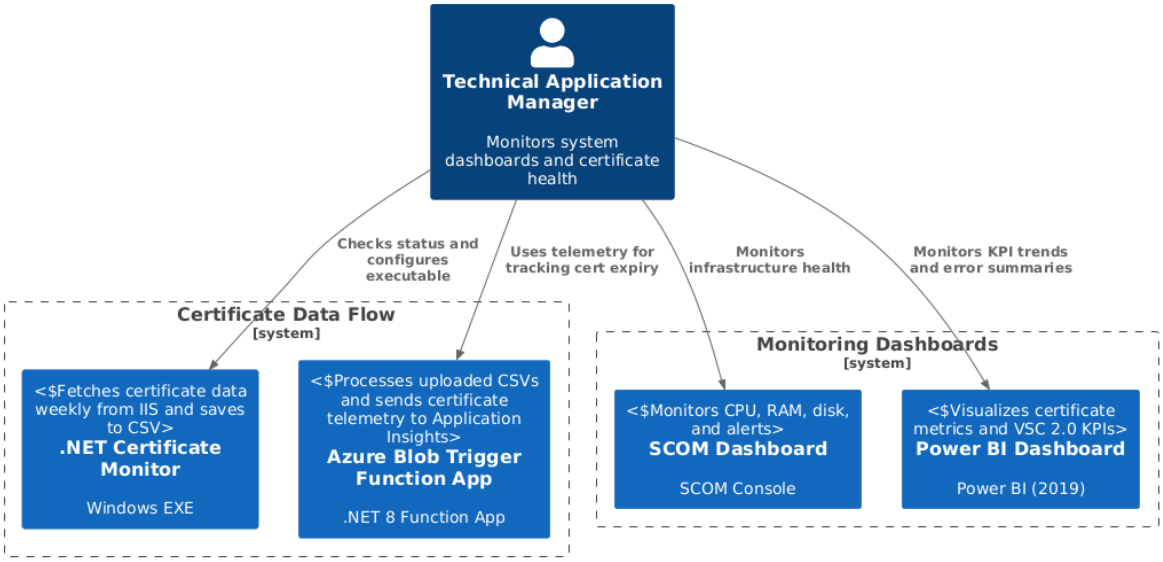

Figure 1: High-level architecture showing how SCOM, Azure Blob Trigger Function App, .NET Certificate Monitor, and Power BI interact.

This diagram gives a top-level overview of how each independent component contributes to the VSS Insights Project. While the components (SCOM Dashboard, Azure Blob Trigger Function App, .NET Certificate Monitor, and Power BI Dashboard) operate separately, they are all part of a broader initiative to improve monitoring across infrastructure via Scom using operation console, certificates expiration monitoring, and Vsc2.0 project success metrics. This visual helps map out the major tools involved and shows how they align with the overall system goals.

3.2 Container-Level Diagram

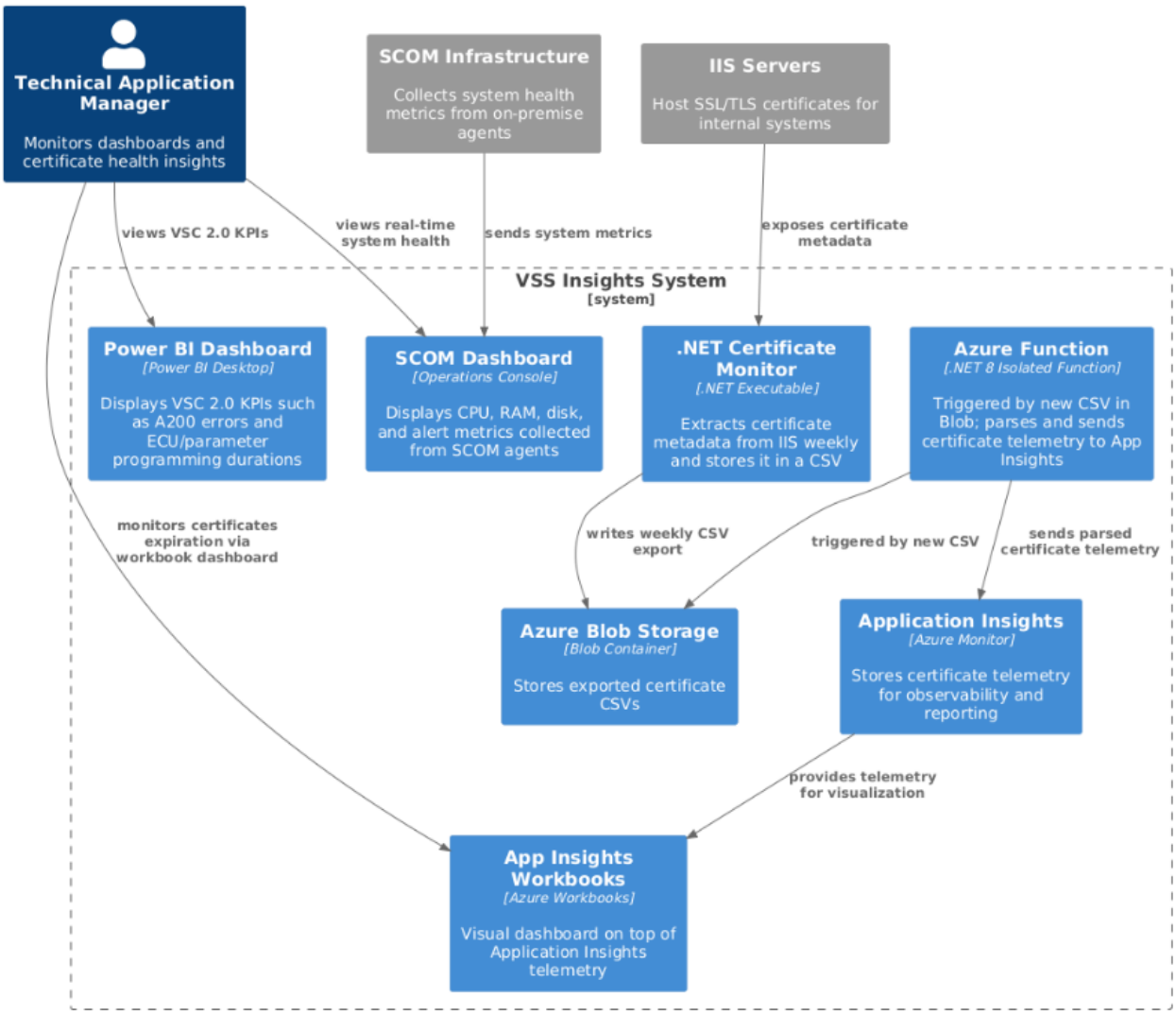

Figure 2: Container-level breakdown of system components using the C4 model.

This diagram provides a breakdown of the VSS Insights system into high-level containers using the C4 model. Each container represents a deployable or executable unit that serves a dedicated responsibility. It helps illustrate the individual roles of systems like the SCOM dashboard, certificate monitoring app, Azure Blob Trigger Function App, and Power BI. Each container functions independently and supports specific monitoring goals such as infrastructure visibility, telemetry collection, or KPI reporting.

4. Component Design

4.1 SCOM Dashboard

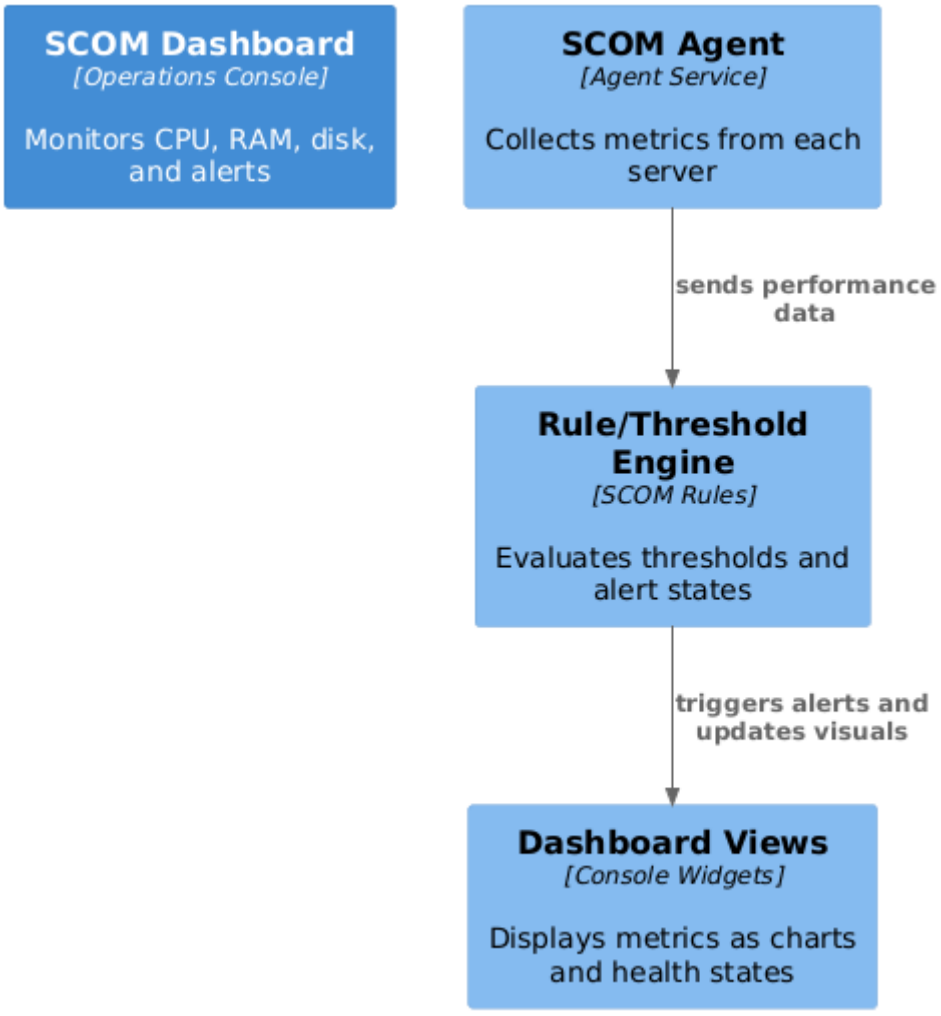

Figure 3: SCOM Dashboard component diagram showing how agents collect and display system health metrics.

This monitoring solution provides real-time visibility into system resource usage and critical alerts across our server infrastructure. The implementation utilizes System Center Operations Manager (SCOM) via Operations Console as the primary monitoring platform.

Data collection is handled through SCOM agents deployed on each monitored server, which continuously gather performance metrics and system health information. The dashboard presents key performance indicators including CPU utilization, RAM consumption, and available disk space as percentage values, along with a comprehensive view of critical alerts showing both alert counts and detailed information.

The system incorporates color-coded health states to quickly identify issues requiring attention. Additionally, the solution offers an aggregated view that allows administrators to monitor multiple servers simultaneously, streamlining the oversight process and enabling faster response times to potential problems.

4.2 Certificate Export for monitoring

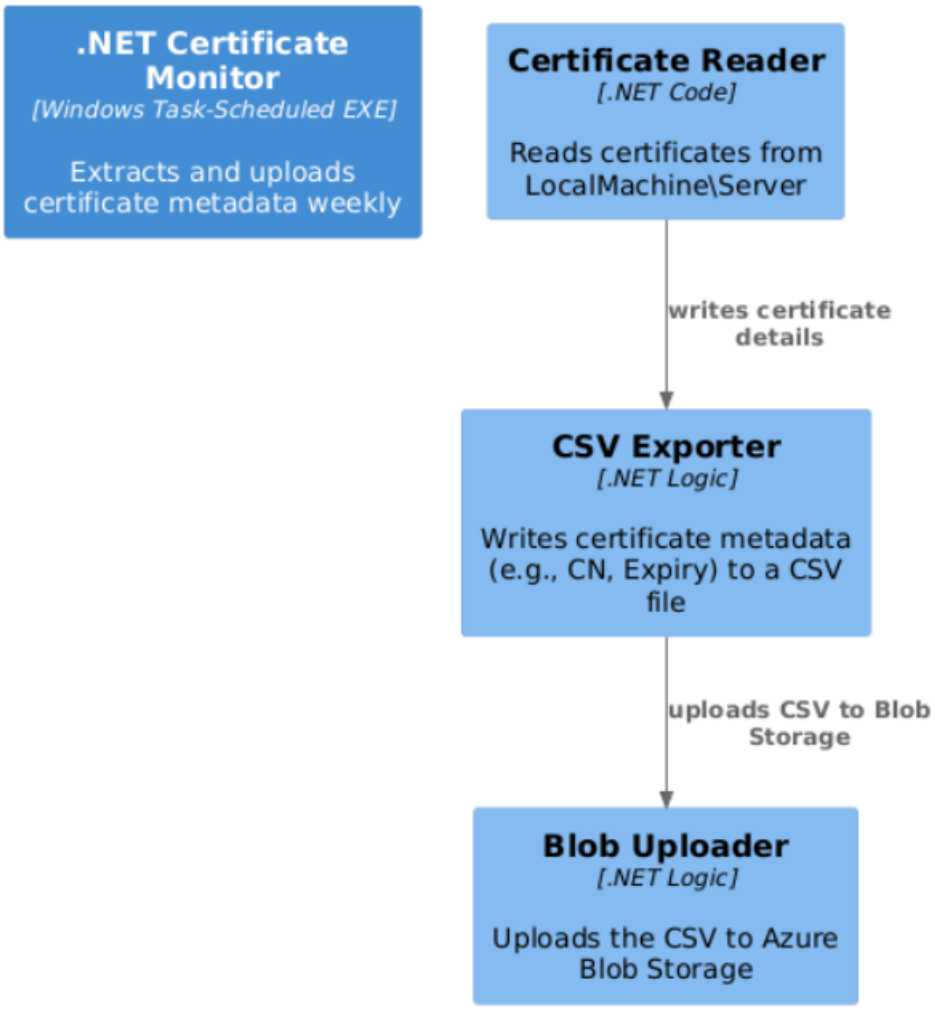

Figure 4: Certificate Monitoring component diagram outlining .NET executable workflow and CSV export to Blob Storage.

This solution provides automated monitoring of SSL/TLS certificate expiration dates across our IIS infrastructure. The system exports certificate data to CSV format for monitoring purposes.

The implementation consists of a .NET executable that connects to designated IIS servers to retrieve certificate information. Once collected, the application forwards key certificate metadata including certificate names, expiration dates, and last update timestamps to Application Insights for monitoring.

The exported data captures three critical data points for each certificate: the certificate common name (CN), the expiration date, and the timestamp of the last update. This

information enables proactive certificate management and helps prevent service disruptions due to expired certificates.

4.3 Azure Blob Trigger Function App

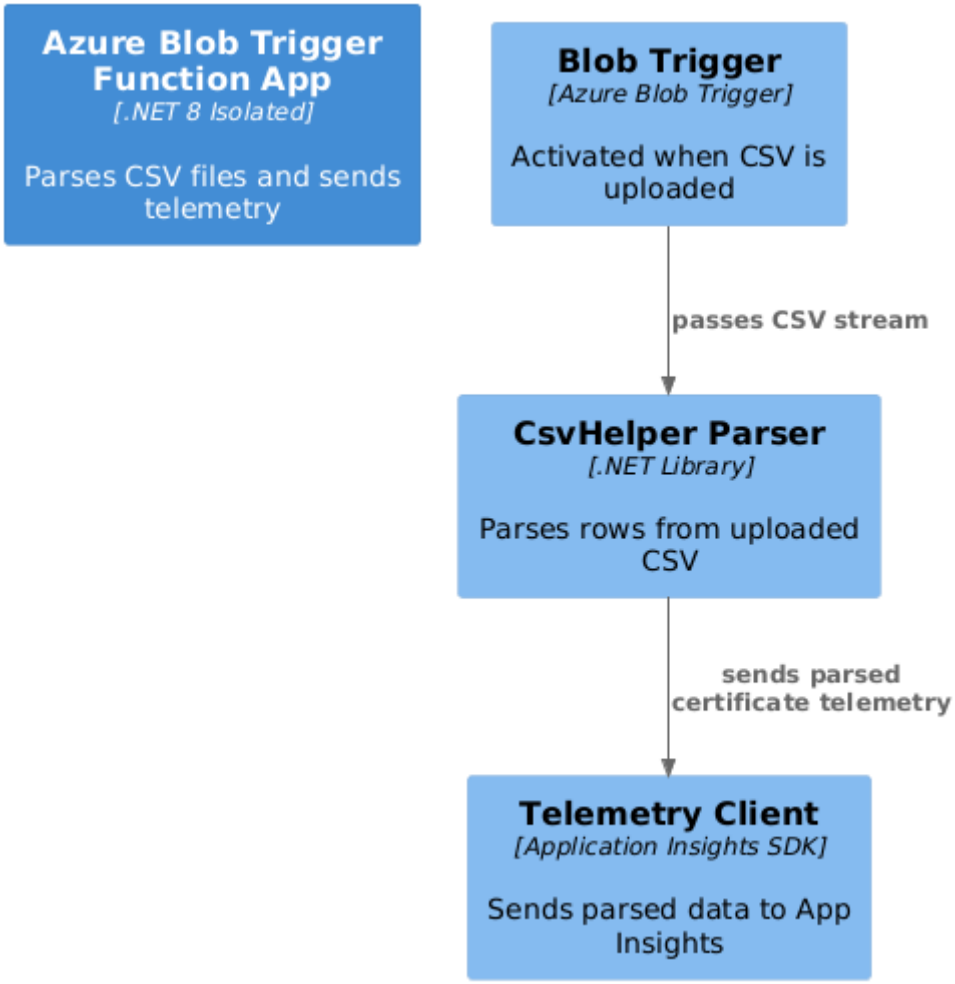

Figure 5: Azure Blob Trigger Function App component diagram illustrating CSV trigger, parsing, and telemetry pipeline.

This is a .NET 8 Azure Function that handles certificate processing when CSV files get uploaded. It uses a blob trigger to automatically kick off whenever a new CSV hits the storage container.

The basic flow is pretty straightforward - when a CSV gets uploaded, the function parses through each row using CsvHelper, then creates telemetry events for every certificate entry

it finds. All of that telemetry data gets pushed to Application Insights, so we can track the rows using a workbook.

I've built in some error handling for when the CSV parsing goes sideways, plus retry logic to deal with temporary failures. The function also writes out processing status info to help with debugging and tracking how things are running.

4.4 VSC 2.0 Power BI Dashboard

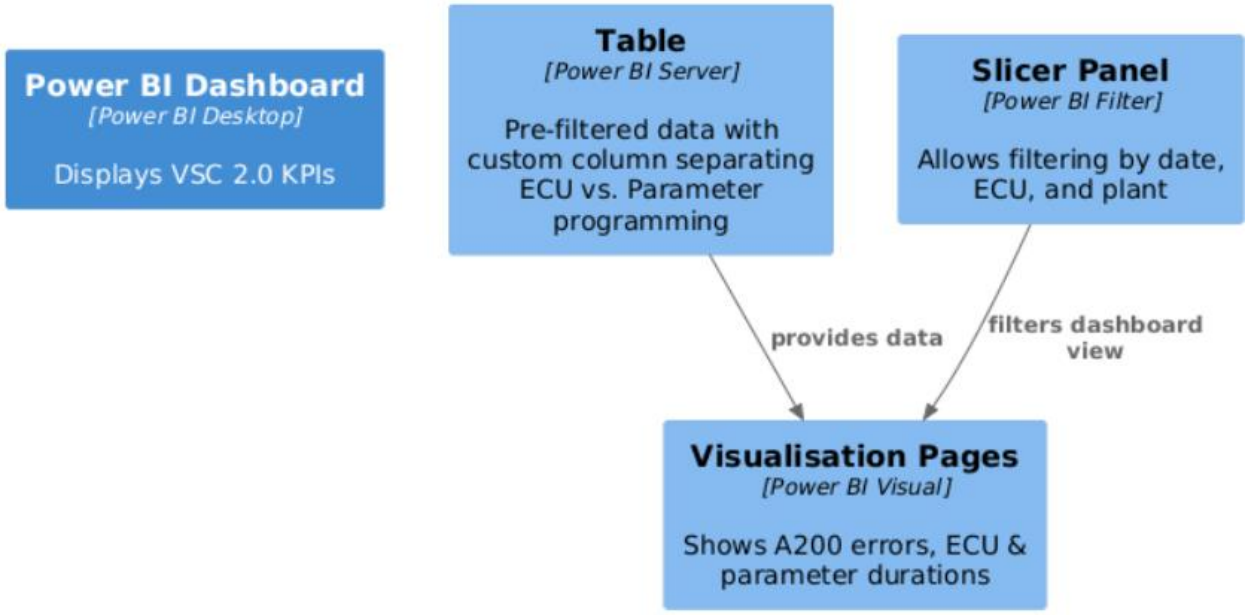

Figure 6: Power BI component diagram highlighting data source, slicers, and visualized KPIs.

This Power BI dashboard tracks key performance indicators for the VSC 2.0 project, providing insights into project success metrics and system reliability. The solution uses Microsoft Power BI's September 2019 version, connecting directly to our database through the PowerBI server for data access.

The dashboard presents several important views of system performance, including A200 error tracking broken down by day, ECU, and plant location. Certificate error analysis is displayed from the DAVIE perspective, while programming metrics show both ECU and parameter programming times with statistical breakdowns including averages, medians, and ranges across different plants. Monthly A200 error comparisons help identify trends and patterns over time.

Interactive filtering capabilities allow users to drill down into specific timeframes, ECU types, and plant locations for targeted analysis. The dashboard also includes exportable visualizations and KPI summary views, making it easy to share findings with stakeholders and track progress against project objectives.

5. Security Considerations

The solution implements several security measures to protect sensitive data and restrict access appropriately. Blob storage access is controlled through SAS tokens combined with role-based access controls to ensure only authorized users can interact with stored data.

For the Azure Function component, we've configured Managed Identity authentication to securely connect with Application Insights, eliminating the need for stored credentials. Certificate monitoring focuses exclusively on metadata collection—no private key information is ever accessed or stored by the system.

SCOM access remains limited to administrative users only, maintaining the existing security model for operations management while providing the necessary monitoring capabilities.

6. Conclusion

The VSS Insights Project successfully integrates multiple monitoring and reporting tools into a unified system. By automating certificate data collection through a .NET executable, using Azure Functions to handle telemetry ingestion, and using Power BI for data visualization, the system enhances the visibility, reliability, and efficiency of monitoring activities at DAF. The solution replaces scattered manual processes with centralized, automated insights. Empower the Technical Application Manager (TAM) to stay well informed. This architecture lays a scalable foundation for future improvements in observability and operational awareness.