Tunnel View Project – Azure Infrastructure and Data Processing Plan

Date: [Insert Date]

Attendees: [List all participants]

Prepared By: [Your Name]

# 1. Project Overview

Tunnel View is an on-premise image processing application currently using Ladybug cameras to capture images of tunnel segments. These images are stored locally and later processed using Python-based scripts. As part of the modernization and scalability strategy, this project aims to migrate the solution to Microsoft Azure, leveraging virtual machines, blob storage, and automation for compute resource management.

# 2. Objective

To analyze infrastructure requirements and define the architectural and operational considerations for deploying the Tunnel View application and processing pipeline on Azure.

# 3. Meeting Agenda

* Requirement gathering for the Tunnel View project.
* Clarifications regarding VM provisioning, storage, caching, monitoring, and job automation.

# 4. Detailed Requirements and Considerations

## Storage Requirements

The application handles large image files generated by Ladybug cameras. Though the exact blob storage requirement wasn't specified, it is estimated to be based on monthly image ingestion of approximately 900GB. Azure Blob Storage will be used for storing images with standard redundancy settings.

## VM Specifications

Processing of images requires highly provisioned VMs. Azure VMs with high vCPU and RAM configurations will be used. VM sizing will depend on parallel job processing load and script requirements. SSD-based storage is recommended for performance.

## Python Script Execution and Automation

Python scripts will be deployed on VMs and support concurrent execution across multiple nodes. To optimize cost, VMs will be spun up only during job execution and terminated afterward. Automation scripts will be implemented for provisioning and deprovisioning.

## High Availability and Tools

High Availability (HA) is optional. The client is expected to share the list of software tools and versions for installation on VMs to ensure compatibility with Azure Marketplace images or custom VM images.

## Data Ingestion and Processing Frequency

Image data amounting to 900GB per month is expected to be uploaded in batch mode. Processing will occur weekly, averaging 30GB per day. The system will retain images for up to one year.

## Database Requirements

There is no immediate need for a relational database such as PostgreSQL. The data pipeline does not rely on DB-backed metadata or lookups at this stage.

## Caching Mechanism

Caching (e.g., Redis or Memcached) is under evaluation. Final strategy will be based on access frequency and performance requirements. This is crucial for scenarios where repeat access of metadata or thumbnails is needed.

## Access Patterns

Users or downstream applications will access processed data via APIs, portals, or direct file access. Azure Blob Storage will be used for storing and distributing the data. Existing access roles from prior projects may be reused.

## Cost Estimation

Cost estimation for compute (VMs), storage, and data processing is essential for budgeting. This includes estimating VM usage hours, blob storage consumption, and ingress/egress data costs.

## Imaging Schedule

Geometric car imaging runs Monday to Thursday. Each day, a specific segment of the tunnel is captured and processed. The image repository maintains a rolling retention of one year.

## Python Script Configuration

Esri will manage Python script configurations and ensure compatibility with the processing infrastructure.

## Access Control

Some team members already have access to existing environments. These access controls will be validated and reused where possible to minimize onboarding efforts.

## Monitoring and Security

Basic monitoring tools (e.g., Azure Monitor, Log Analytics) will be enabled on VMs. The server team will handle security and implement any additional tools such as endpoint protection or SIEM if needed.

## Performance and Processing Time

The estimated processing time for 30GB/day will be determined after benchmarking the Python script on selected VM SKUs. This will help fine-tune VM configurations and automation frequency.

# 5. Action Items

1. Client to share software tools and version requirements for compatibility assessment.
2. Tech team to finalize caching design and choose appropriate services (Redis/Memcached).
3. Infrastructure team to prepare cost estimates based on monthly workloads and retention policy.
4. Esri to confirm retention policy and access model for image data.
5. DevOps to document automation scripts for provisioning and deprovisioning VMs, and finalize specs.

# 6. Conclusion

The above document provides a comprehensive overview of infrastructure, processing logic, and operational considerations for deploying Tunnel View on Azure. The next steps involve finalizing VM specs, automation flow, and caching strategy. Stakeholders are requested to validate their action items and share any updates during the upcoming follow-up meeting.