# Solution Context

## Overview

The purpose of this solution is to migrate existing scanned documents from an on-premises Oracle database to AWS storage while ensuring security, scalability, and traceability. This migration leverages a batch processing approach, securely transferring data via API Connect and AWS Direct Connect to AWS services including API Gateway, Lambda, S3, and DynamoDB.

## Current State (Today)

• Documents are stored in an Oracle database on-premises.

• There is no centralized cloud storage, leading to challenges in retrieval, scalability, and long-term retention.

• API Connect (API-C) is used for handling API requests but does not directly integrate with AWS storage.

• Limited automation exists for document movement and lifecycle management.

## Future State (Tomorrow)

• Documents will be migrated to Amazon S3, providing scalable, cost-effective storage.

• Amazon DynamoDB will store document metadata (GUID, S3 path, timestamp) for efficient lookup and tracking.

• Migration will happen via on-prem API Connect → Direct Connect → AWS API Gateway → AWS Lambda.

• The system will provide high availability, security, and traceability with AWS services.

## Solution Overview

### 1. Batch Job Execution (Spring Batch - On-Premises)

• A scheduled Spring Batch job extracts document metadata (GUID, file path, timestamp) and content from Oracle.

• The batch job sends an API request to on-prem API Connect (API-C) to initiate the document migration process.

### 2. API Routing & Secure Transfer (On-Prem to AWS)

• API-C validates the request and securely routes it through AWS Direct Connect to AWS.

• The request is processed by AWS API Gateway, which forwards it to the appropriate AWS Lambda function.

### 3. Document Processing & Storage (AWS Lambda, S3, DynamoDB)

• AWS Lambda (Create function):

- Extracts document content and metadata.

- Stores the document securely in Amazon S3.

- Writes metadata (GUID, S3 path, timestamp) to Amazon DynamoDB.

### 4. Updating Source System (Oracle) with GUID

• After successfully storing the document in AWS, Lambda sends the generated GUID back to Oracle via API-C.

• This ensures that the source system retains traceability of the document’s new location.

## Key Benefits

✅ Scalability – S3 & DynamoDB provide auto-scaling capabilities for handling large document volumes.

✅ Performance Optimization – AWS Direct Connect ensures low-latency, high-throughput migration.

✅ Security & Compliance – Data is encrypted at rest (S3, DynamoDB) and in transit (API Gateway, Direct Connect).

✅ Traceability – Oracle maintains the GUID, ensuring easy document retrieval and auditability.

## Considerations

• Error Handling: The system must gracefully handle API failures, connectivity issues, and implement retries.

• Performance Tuning: Batch processing intervals should be optimized to balance the load on Oracle and AWS services.

• Monitoring & Logging: CloudWatch should be configured for real-time monitoring of migration failures and system performance.

## Next Steps

• Review the solution design with stakeholders.

• Implement detailed monitoring and logging strategies.

• Validate the migration process through a proof of concept (PoC).