Parkland Design Blueprint — ACA-Hosted Logic Apps Standard for 275/277/278

Executive Summary

Parkland requires a scalable, secure, and auditable solution to process HIPAA X12 transactions, including 275 attachments, with support for 277 claim status and 278 service review flows. This blueprint delivers the solution using Logic Apps Standard hosted in Azure Container Apps (ACA), with an Integration Account for X12 schema and agreement management.

The architecture leverages Azure Service Bus for decoupling, Azure Blob/Files for attachments and archiving, and Azure SQL for run history. All external access is secured using Managed Identity (MI) and least-privilege RBAC assignments. The design provides both automated ingestion (Blob → Service Bus → Workers) and deterministic replay via HTTP endpoints.

Goals & Design Principles

- Security first: Managed Identity, no secrets, RBAC enforced.
- Auditability: All raw interchanges archived, replay endpoints provided.
- Resilience: Queues decouple ingestion from processing, enabling back-pressure handling.
- Scalability: ACA auto-scales (0→N replicas), KEDA can be enabled on queue depth.
- Extensibility: Same worker pattern applies to 275, 277, 278, reducing complexity.
- Simplicity: Bicep + GitHub Actions for repeatable, consistent deployments.

Architecture Overview

The system consists of:
- Azure Blob Storage: /incoming for uploads, /attachments for extracted files, /archive for raw interchanges.
- Bridge Workflow: Detects new blobs and enqueues references to Service Bus.
- Azure Service Bus: Queues edi-275, edi-277, edi-278.
- ACA Container App: Hosts Logic Apps Standard runtime and workflows (bridge, workers, replayers).
- Integration Account: Partners + agreements for 275, 999, 277, 278.
- Azure SQL: Stores Logic Apps state and run history.
- HTTP Replayers: Allow deterministic reprocessing of known blob payloads.

Figure 1: System-Level Architecture

Workflow Design

- Bridge Workflow: Blob trigger → enqueue blob URL to SB.
- 275 Worker: SB trigger → fetch blob → IA decode (275) → archive raw → extract → encode 999 → output.

- 277 Worker: Same pattern, decode 277.
- 278 Worker: Same pattern, decode 278.
- Replayers (275/277/278): HTTP trigger → enqueue blob URL to SB → re-run same worker logic.

This common pattern simplifies maintenance, with each transaction type isolated for reliability.

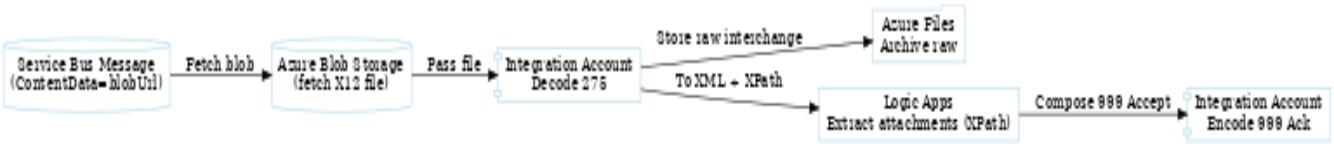

Figure 2: Worker Pattern (275 Example — applies to 277/278)

Implementation Approach

- Bridge Workflow: Ensures ingestion from blob events is consistent and decoupled.
- Workers: One per transaction type; each uses the same decode/archive/acknowledge sequence.
- HTTP Replayers: Accept a { "blobUrl": "" } body, enqueue to SB, enabling audit and testing.
- Integration Account: Holds schemas and agreements; combined agreement includes 275, 999, 277, 278.
- Data Persistence: SQL for run history; Azure Files for archived and extracted artifacts.

Security Model

- User-Assigned Managed Identity (UAMI): attached to ACA container app.
- RBAC:
- Service Bus: Azure Service Bus Data Sender and Data Receiver.
- Storage: Storage Blob Data Contributor.
- Secrets: None stored in code; all connections use MI where supported.
- Integration Account: Requires one-time connection authorization in Azure Portal.

Optional: Private endpoints and VNET-injected ACA environment for production isolation.

Deployment Guide

- All-in-one Bicep: Deploys ACA, UAMI, SQL, Storage, IA, API connections, and RBAC in one template.
- Parameters file: Pre-filled with Parkland values (aurelian-hipaa, dev-integration, stpchphipaa275).
- GitHub Actions Pipeline:
- Builds and pushes the Logic Apps Standard container image to ACR.
- Deploys the Bicep template with containerImage parameter.
- One-time authorization: IA connection authorized in Portal.

Runbook

Day-1 Provisioning:
1. Push Logic Apps container image to ACR.
2. Deploy all-in-one Bicep template.
3. Authorize Integration Account connection.

Day-2 Operations:
- Monitor SB queues and DLQs.

- Monitor run history in SQL.
- Use replay endpoints for testing or reprocessing.
- Archive directory contains immutable record of all raw interchanges.

Extensibility & Next Steps

- Add KEDA-based scaling rules (queue length).
- Add per-partner routing (different agreements per partner).
- Enhance 999 to reflect validation results.
- Optionally front HTTP replay with APIM for access control.

Deliverables & Hand-Off

Cognizant SI team is expected to deliver:
- Deployed ACA environment with Logic Apps container.
- Integration Account with schemas + agreements configured.
- Bicep deployments and parameter files.
- Operational runbook + monitoring alerts.
- Validated 275/277/278 processing flows with end-to-end testing.

Appendices

Interfaces:
- SB queues: edi-275, edi-277, edi-278
- Replay endpoints: POST /api/manual/triggers/manual/invoke
- Blob paths: /incoming, /attachments, /archive

Security Roles:
- SB: Data Sender/Receiver (UAMI)
- Blob: Blob Data Contributor (UAMI)

Operational Metrics:
- Queue length, DLQ count
- Worker run failures
- Decode/encode error rates
- Storage write success/failures
- Replay invocation count