ESign Elections Canada - Solution Architecture & Design Document

Part 1: Executive Summary & Architecture Overview

- Version: 1.0
- Last Updated: February 2026
- Document Owner: Platform Engineering Team
- Author: Frederick Pearson

Table of Contents - Complete Document

1. Executive Summary

1.1 Purpose

This document describes the solution architecture for the **ESign Elections Canada Broker Service**, a digital signature platform that provides EC with secure, compliant, and auditable electronic signature capabilities via Nintex AssureSign.

1.2 Scope

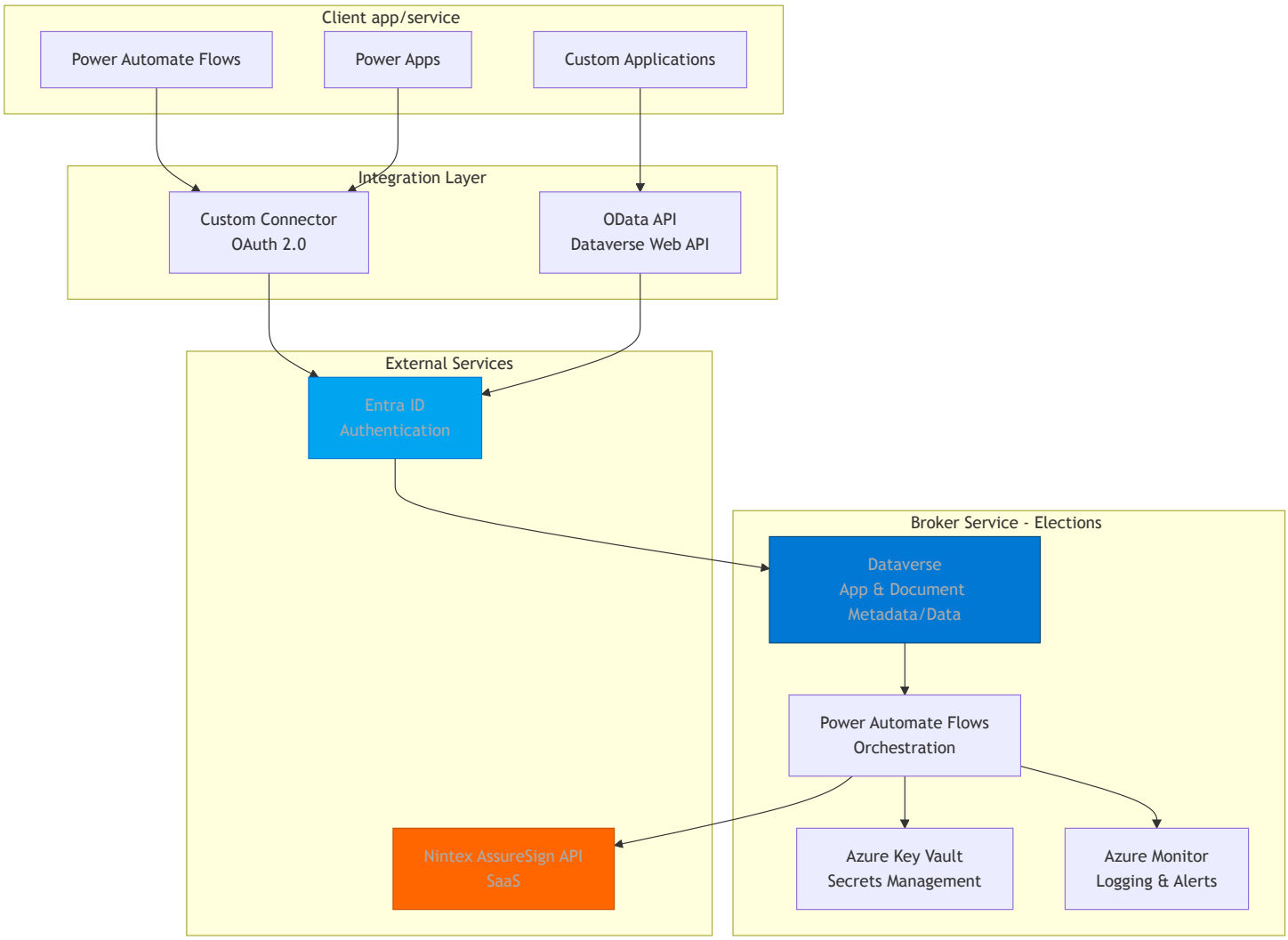
The architecture encompasses:

- **Broker Service Layer**: Microsoft Power Platform (Dataverse) broker (API gateway/middleware)
- **Client Integration Layer**: Custom connectors and OData endpoints for client consumption
- **External Integration**: Nintex AssureSign API integration
- **Security & Compliance**: Protected B controls, ITSG-33 alignment, audit logging
- **Operations**: Monitoring, logging, backup, disaster recovery

1.3 Key Architectural Principles

Principle	Description	Implementation
Support for mulitple business units	Logical isolation between client app/services	Row-level security (RLS) in Dataverse & business units/security roles for API access governance
API-First	OData/REST endpoints as primary integration method	Dataverse Web API + Custom Connectors
Zero Trust	Never trust, always verify	OAuth 2.0, service principals, conditional access
Defense in Depth	Layered security controls	Network, identity, application, data encryption
Audit Everything	Complete audit trail	Dataverse audit logs + Azure Monitor
Least Privilege	Minimal necessary permissions	Custom security roles, Entra ID RBAC

1.4 High-Level Architecture



1.5 Technology Stack

Layer	Technology	Version	Purpose
Platform	Microsoft Power Platform	Latest	Broker service host
Database	Dataverse	9.2+	Data store & API
Orchestration	Power Automate	N/A	Workflow automation
Identity	Microsoft Entra ID	Latest	Authentication & authorization
Secrets	Azure Key Vault	Latest	Credential management
Monitoring	Azure Monitor	Latest	Logging & alerting
API Gateway	Dataverse Web API	9.2+	OData endpoints
External SaaS	Nintex AssureSign	3.7+	Digital signature platform

2. Architecture Overview

2.1 Architectural Patterns

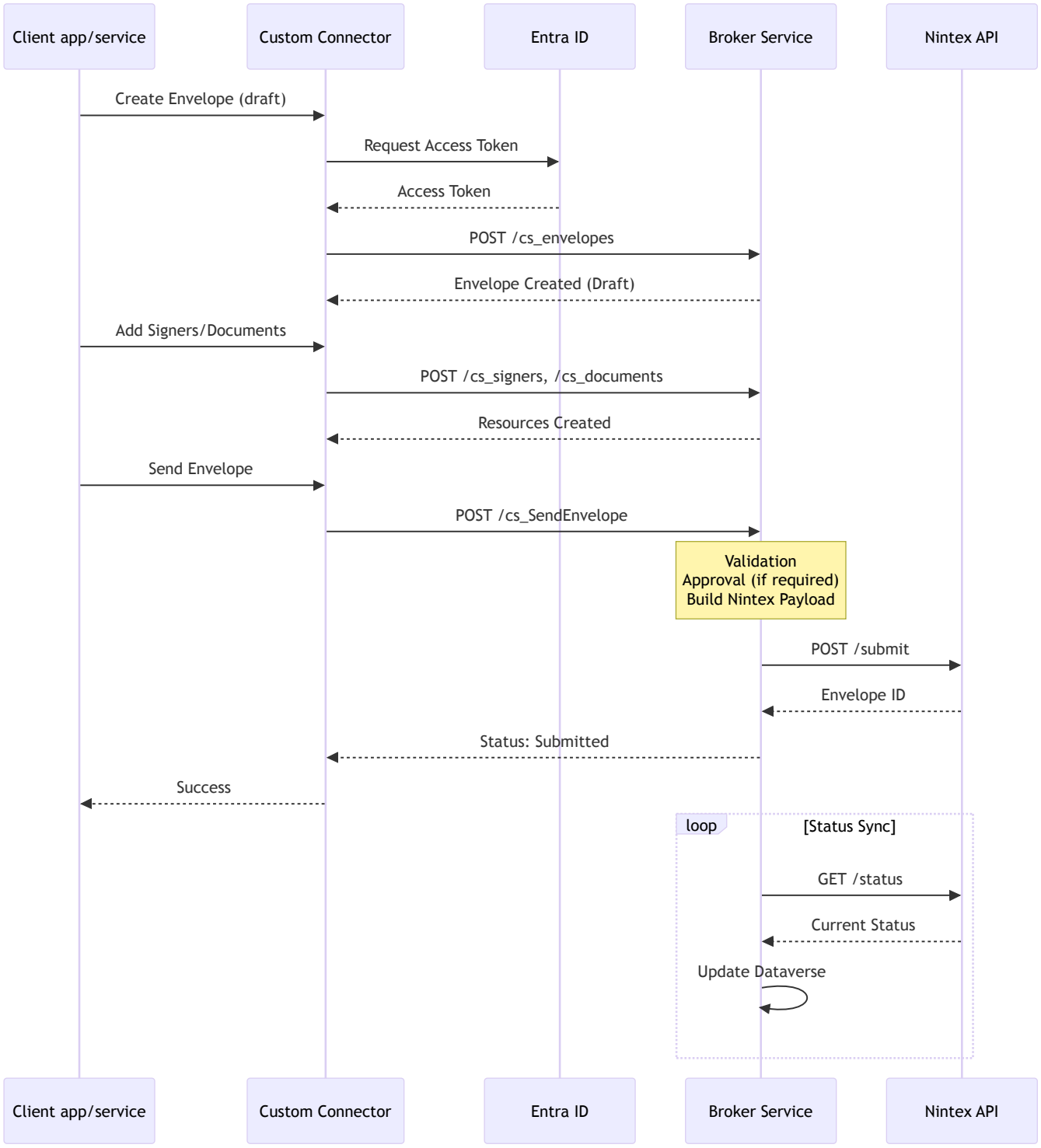
2.1.1 Broker Pattern

The solution implements a **broker architectural pattern** where:

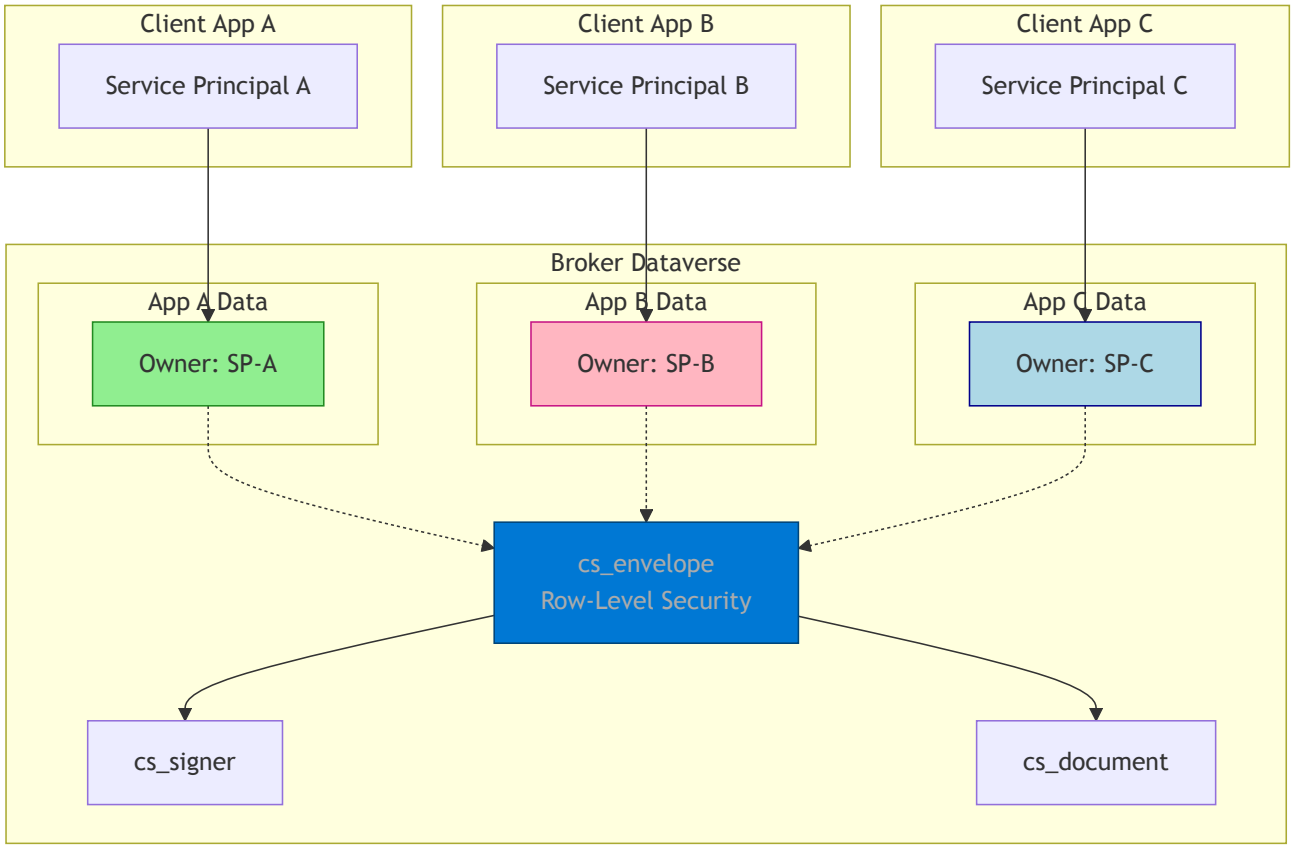
1. **Client service** interact with the broker via standard ODATA API (Dataverse API)
2. **Broker service** manages complexity, approvals, and integration layer applications (client services) and Office & PDF files (api middleware)
3. **Nintex AssureSign** provides the underlying signature capabilities

Purpose:

- Centralized Nintex license management
- Consistent security controls
- Simplified client integration
- Centralized audit and compliance


Client app/service
Custom Connector
Entra ID
Broker Service
Nintex API
Create Envelope (draft)
Request Access Token
Access Token
POST /cs_envelopes
Envelope Created (Draft)
Add Signers/Documents
POST /cs_signers, /cs_documents
Resources Created
Send Envelope
POST /cs_SendEnvelope
Validation
Approval (if required)
Build Nintex Payload
POST /submit
Envelope ID
Status: Submitted
Success
loop
[Status Sync]
GET /status
Current Status
Update Dataverse
Client app/service
Custom Connector
Entra ID
Broker Service
Nintex API

2.1.2 Client Service -> Broker Service Architecture



Isolation Mechanisms:

Mechanism	Implementation	Security Level
Identity	Unique service principal per client	Entra ID tenant-level
Data	Owner-based RLS in Dataverse	Row-level
Network	Shared (PaaS), IP restrictions optional	Tenant-level
Logging	Separate Log Analytics workspaces per client	Subscription-level

2.2 Dataverse as Integration Platform

Why Dataverse is paramount:

1. **OData Protocol**: Industry-standard REST API
   - Queryable via $filter$,select, $expand
   - Standardized authentication (OAuth 2.0)
   - Native pagination, batch operations
2. **Web API Capabilities**:
   - RESTful CRUD operations
   - Custom Actions (e.g., cs_SendEnvelope)
   - Webhooks for real-time integration
   - SDK support (C#, JavaScript, Python)
3. **Built-in Security**:
   - Row-level security (RLS)

- Column-level security (CLS)
- Audit logging
- Data encryption at rest and in transit

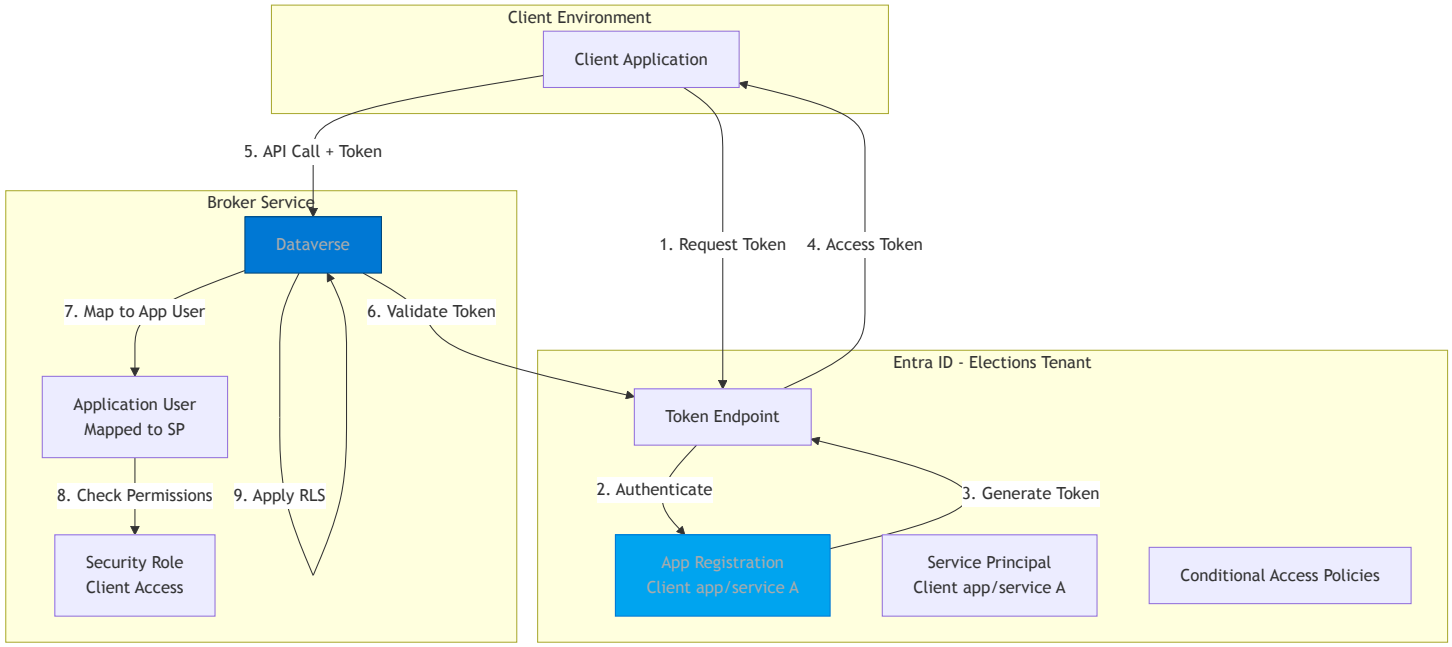
4. **Power Platform Integration**:
   - Native Power Automate triggers
   - Power Apps data source
   - Custom connectors
   - AI Builder capabilities

OData Endpoint Example:

```
GET https://{lce-broker-subdomain}.crm3.dynamics.com/api/data/v9.2/cs_envelopes
    ?$filter=cs_status eq 'Draft' and _ownerid_value eq '{client-sp-id}'
    &$select=cs_name,cs_subject,cs_status
    &$expand=cs_envelope_signer($select=cs_email,cs_fullname)
    &$top=50

Authorization: Bearer {access_token}
```

3. Access Control & Identity Management

3.1 Authentication Architecture



3.2 Authentication Methods

3.2.1 Service Principal (Recommended for Automation)

Use Case: Power Automate, custom applications, scheduled jobs

Setup:

```
# Create App Registration
az ad app create \
  --display-name "ESign-Clientapp/service-Prod" \
  --sign-in-audience AzureADMyOrg

# Create Client Secret
az ad app credential reset \
  --id {app-id} \
  --append \
  --years 2

# Grant API Permissions
az ad app permission add \
  --id {app-id} \
  --api 00000007-0000-0000-c000-000000000000 \
  --api-permissions 78ce3f0f-a1ce-49c2-8cde-64b5c0896db4=Role

# Admin consent
az ad app permission admin-consent --id {app-id}
```

Token Acquisition:

```
POST https://login.microsoftonline.com/{tenant-id}/oauth2/v2.0/token
Content-Type: application/x-www-form-urlencoded

client_id={client-id}
&client_secret={client-secret}
&scope=https://{lce-broker-subdomain}.crm3.dynamics.com/.default
&grant_type=client_credentials
```

3.2.2 Managed Identity (For Azure-Hosted Applications)

Use Case: Azure Functions, Logic Apps, VMs

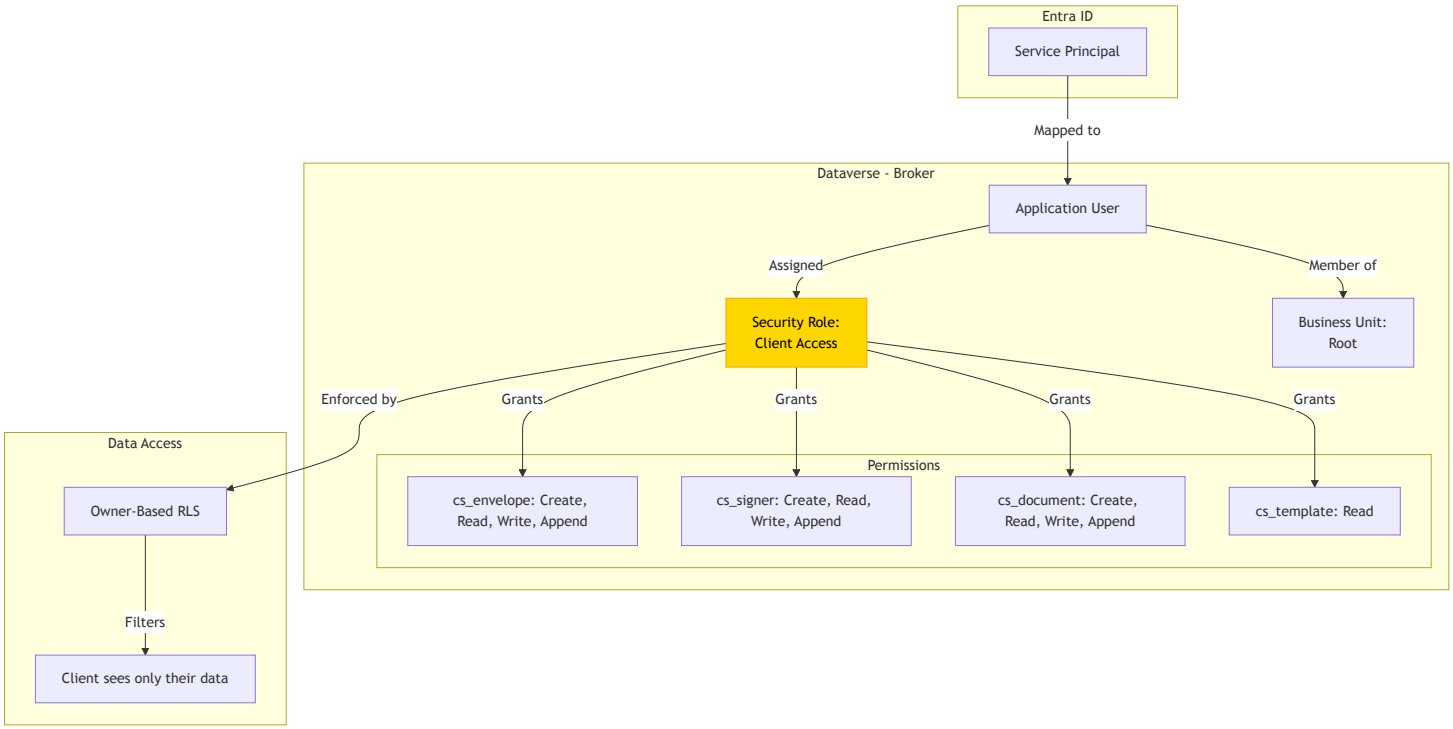
Configuration:

```
# Enable system-assigned managed identity
az webapp identity assign --name myapp --resource-group mygroup

# Grant Dataverse permissions via Entra ID
az role assignment create \
  --assignee {managed-identity-principal-id} \
  --role "Dynamics 365 User" \
  --scope /subscriptions/{sub-id}/resourceGroups/{rg}
```

Token Acquisition:

```
GET http://169.254.169.254/metadata/identity/oauth2/token
    ?api-version=2018-02-01
    &resource=https://{lce-broker-subdomain}.crm3.dynamics.com

Metadata: true
```

3.3 Authorization Model



3.4 Security Roles Configuration

Client Access Role

Privilege	Create	Read	Write	Delete	Append	Append To	Assign
cs_envelope	Organization	Organization	Organization	Organization	Organization	Organization	None
cs_signer	Organization	Organization	Organization	Organization	Organization	Organization	None
cs_document	Organization	Organization	Organization	Organization	Organization	Organization	None
cs_field	Organization	Organization	Organization	Organization	Organization	Organization	None
cs_template	None	Organization	None	None	None	None	None
cs_apirequest	Organization	Organization	None	None	None	None	None

Notes:

- Organization-level privileges required due to Dataverse API behavior
- RLS enforced via `ownerid` field filtering
- Client cannot query or modify records owned by other clients

Broker Administrator Role

Privilege	Create	Read	Write	Delete	Append	Append To	Assign
cs_envelope	Organization	Organization	Organization	Organization	Organization	Organization	Organization
cs_signer	Organization	Organization	Organization	Organization	Organization	Organization	Organization
cs_document	Organization	Organization	Organization	Organization	Organization	Organization	Organization

Privilege	Create	Read	Write	Delete	Append	Append To	Assign
cs_apirequest	Organization	Organization	Organization	Organization	Organization	Organization	None
cs_template	Organization	Organization	Organization	Organization	Organization	Organization	None
cs_webhook	Organization	Organization	Organization	Organization	None	None	None

3.5 Conditional Access Policies

Recommended Policies for Broker Service:

Policy Name	Conditions	Controls
Require MFA for Admins	User role = Admin	Require MFA
Block Legacy Auth	Client apps = Legacy	Block access
Require Compliant Device	Device state = Not compliant	Block access
IP Restriction (Optional)	IP address not in GC range	Block access
Session Controls	All users	Sign-in frequency = 8 hours

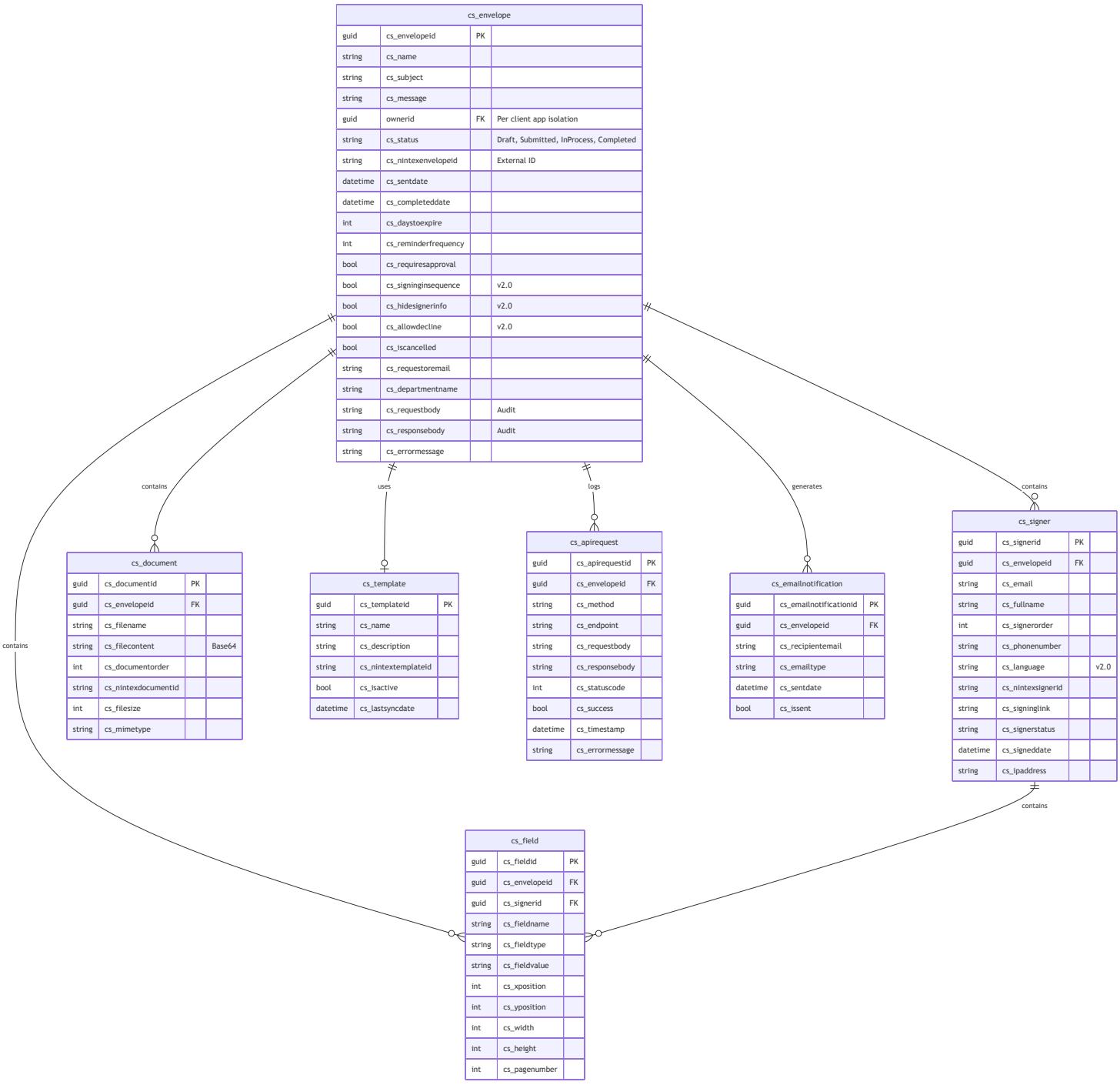
Configuration Table:

Environment	Policy Enabled	Allowed IP Ranges	MFA Required
Production	☑	[FILL: GC IP ranges]	☑
QA	☑	[FILL: GC IP ranges]	☑
Development	☐	Any	☐

Part 2: Data & Integration Architecture

4. Data Architecture

4.1 Entity Relationship Diagram



4.2 Data Model Details

4.2.1 Core Tables

cs_envelope

Purpose: Primary envelope entity representing a signature request

Key Fields:

Field	Type	Purpose	Security
`ownerid`	Lookup	Per App isolation	RLS filter
`cs_status`	Choice	Workflow state	Indexed
`cs_nintexenvelopeid`	String	External reference	Unique
`cs_signinginsequence`	Boolean	Sequential vs parallel	v2.0 feature
`cs_hidesignerinfo`	Boolean	Privacy mode	v2.0 feature
`cs_requestbody`	Memo	Audit trail	Protected B
`cs_responsebody`	Memo	Audit trail	Protected B

Status Values:

Status	Description	Transitions To
Draft	Being built by client	Pending Approval, Submitted
Pending Approval	Awaiting internal approval	Approved, Rejected
Approved	Approved, queued for Nintex	Submitted
Submitted	Sent to Nintex API	InProcess, Failed
InProcess	Signers notified	Completed, Declined, Cancelled
Completed	All signatures obtained	None (terminal)
Declined	Signer declined	None (terminal)
Cancelled	User cancelled	None (terminal)
Rejected	Approval rejected	None (terminal)
Failed	Technical error	Submitted (retry)

cs_signer

Purpose: Individuals who must sign the envelope

Key Fields:

Field	Type	Purpose	PII Classification
`cs_email`	String	Contact	PII
`cs_fullname`	String	Display name	PII

Field	Type	Purpose	PII Classification
`cs_phonenumber`	String	Optional contact	PII
`cs_signinglink`	String	Unique URL	Sensitive
`cs_ipaddress`	String	Audit trail	PII

cs_document

Purpose: Files attached to envelope

Key Fields:

Field	Type	Purpose	Storage Consideration
`cs_filecontent`	Memo	Base64 encoded	Max 10MB, Dataverse storage consumed
`cs_filename`	String	Display name	User-controlled
`cs_filesize`	Integer	Validation	Pre-calculate before insert

Storage Calculation:

- Base64 encoding increases size by ~33%
- 10MB PDF = ~13.3MB base64
- Dataverse capacity: 1GB per environment base + usage-based

cs_apirequest

Purpose: Complete audit log of all Nintex API interactions

Key Fields:

Field	Type	Purpose	Retention
`cs_requestbody`	Memo	Full request payload	7 years
`cs_responsebody`	Memo	Full response	7 years
`cs_statuscode`	Integer	HTTP status	7 years
`cs_timestamp`	DateTime	Chronological order	7 years

4.3 Row-Level Security Implementation

Mechanism:

1. Each client has a unique Service Principal in Entra ID
2. Service Principal mapped to Application User in Dataverse
3. Application User assigned to custom security role
4. Security role grants Organization-level privileges
5. **RLS enforced via `ownerid` field filtering in Dataverse Web API**

Filter Logic:

```
// Automatic filter applied by Dataverse
GET /api/data/v9.2/cs_envelopes
// Becomes:
GET /api/data/v9.2/cs_envelopes?$filter=_ownerid_value eq '{calling-app-user-id}'
```

Validation:

Test Scenario	Expected Result
Client A queries all envelopes	Returns only Client A's envelopes
Client A queries Client B envelope by ID	404 Not Found
Client A attempts to update Client B envelope	403 Forbidden
Admin queries all envelopes	Returns all envelopes (no filter)

4.4 Column-Level Security

Sensitive Fields with CLS:

Field	Table	Access Level	Reason
`cs_requestbody`	cs_envelope	Read: Admin only	Contains PII in payload
`cs_responsebody`	cs_envelope	Read: Admin only	Contains Nintex tokens
`cs_signinglink`	cs_signer	Read: Owner + Admin	Unique access URL
`cs_ipaddress`	cs_signer	Read: Admin only	PII - tracking data

Configuration:

```
<attribute LogicalName="cs_requestbody">
  <IsSecured>true</IsSecured>
  <Descriptions>
    <Description>Full API request payload (Admin only)</Description>
  </Descriptions>
</attribute>
```

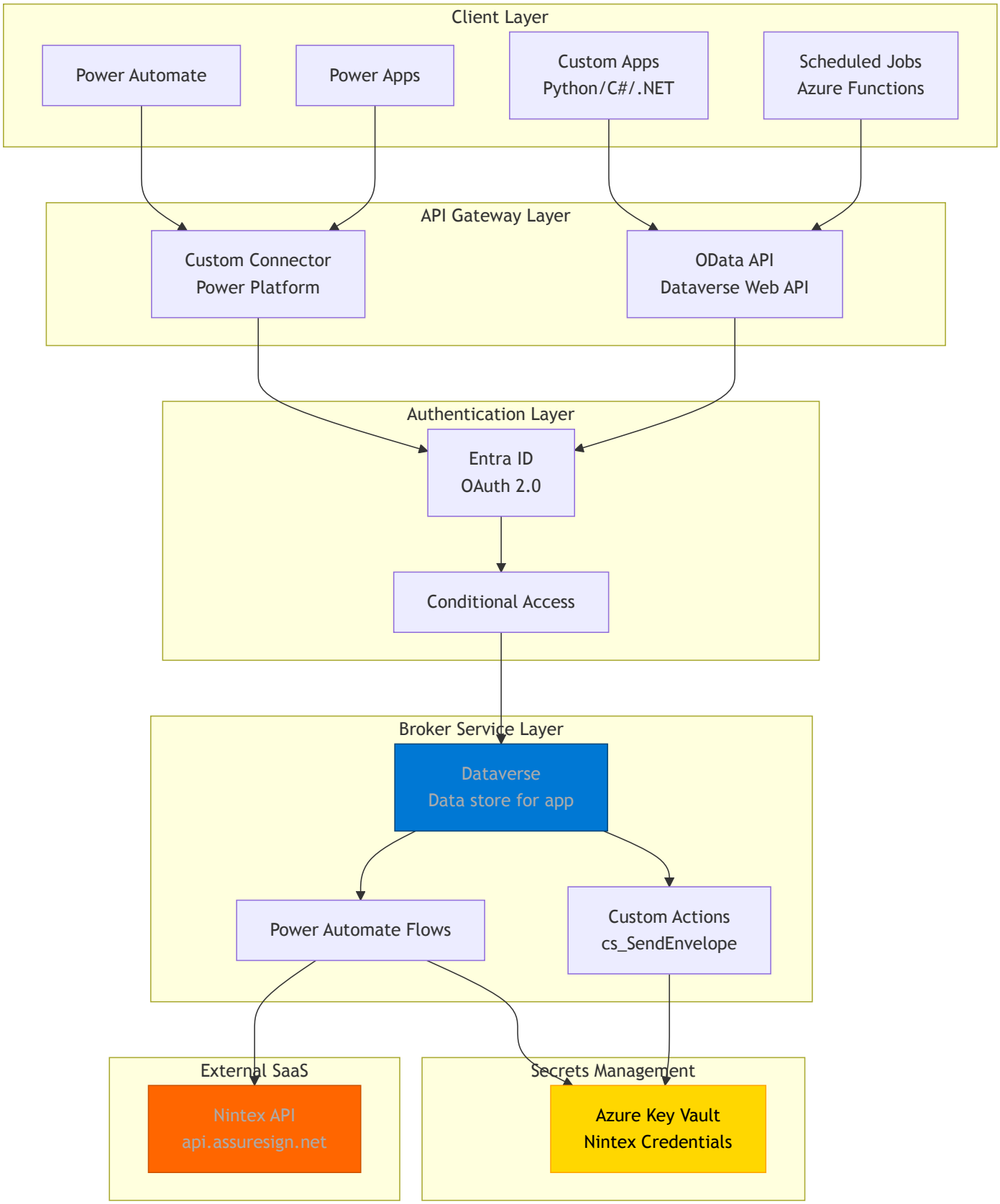
4.5 Data Encryption

Layer	Encryption Method	Key Management
At Rest	AES-256 (Dataverse TDE)	Microsoft-managed
In Transit	TLS 1.2+	Certificate pinning
Application	Base64 encoding (not encryption)	N/A
Backup	AES-256	Microsoft-managed

Note: Base64 encoding of documents is for transport, not security. Documents remain readable in Dataverse.

5. Integration Architecture

5.1 Integration Layers



5.2 Custom Connector Architecture

Connector Type: OpenAPI 2.0 Specification

Authentication: OAuth 2.0 with Authorization Code flow

Connector File Structure:

```
{
  "swagger": "2.0",
  "info": {
    "title": "ESign Elections Canada",
    "version": "2.0.0"
  },
  "host": "{environment}.crm3.dynamics.com",
  "basePath": "/api/data/v9.2",
  "securityDefinitions": {
    "oauth2": {
      "type": "oauth2",
      "flow": "accessCode",
      "authorizationUrl": "https://login.microsoftonline.com/{tenant}/oauth2/v2.0/authorize",
      "tokenUrl": "https://login.microsoftonline.com/{tenant}/oauth2/v2.0/token",
      "scopes": {
        "https://{environment}.crm3.dynamics.com/.default": "Access broker"
      }
    }
  }
}
```

Operations Exposed:

Operation	HTTP Method	Dataverse Endpoint	Description
CreateEnvelope	POST	/cs_envelopes	Create draft envelope
SendEnvelope	POST	/cs_envelopes({id})/Microsoft.Dynamics.CRM.cs_SendEnvelope	Send envelope to Nintex
AddSigner	POST	/cs_signers	Add signer to envelope
UpdateSigner	PATCH	/cs_signers({id})	Modify signer
RemoveSigner	DELETE	/cs_signers({id})	Delete signer
AddDocument	POST	/cs_documents	Attach document
RemoveDocument	DELETE	/cs_documents({id})	Remove document
GetEnvelope	GET	/cs_envelopes({id})	Retrieve envelope details
ListEnvelopes	GET	/cs_envelopes	Query envelopes
GetSigners	GET	/cs_envelopes({id})/cs_envelope_signer	Get envelope signers
ListTemplates	GET	/cs_templates	Available templates

Operation	HTTP Method	Dataverse Endpoint	Description
UpdateEnvelope	PATCH	/cs_envelopes({id})	Modify or cancel
DeleteEnvelope	DELETE	/cs_envelopes({id})	Delete envelope

5.3 OData Endpoint Integration

Base URL Format:

```
https://{environment}.{region}.dynamics.com/api/data/v{version}/
```

Example Queries:

```
# Get all draft envelopes
GET /api/data/v9.2/cs_envelopes?$filter=cs_status eq 'Draft'

# Get envelope with signers expanded
GET /api/data/v9.2/cs_envelopes(guid)?$expand=cs_envelope_signer

# Get envelopes created in last 7 days
GET /api/data/v9.2/cs_envelopes
    ?$filter=createdon ge 2026-02-11T00:00:00Z

# Batch operation
POST /api/data/v9.2/$batch
Content-Type: multipart/mixed;boundary=batch_requests

--batch_requests
Content-Type: application/http

POST /api/data/v9.2/cs_signers HTTP/1.1
Content-Type: application/json

{
  "cs_email": "signer1@example.com",
  "cs_fullname": "John Doe",
  "cs_envelopeid@odata.bind": "/cs_envelopes(guid)"
}
--batch_requests--
```

Query Options Supported:

Option	Purpose	Example
`$filter`	Filter results	`cs_status eq 'Completed'`
`$select`	Specify columns	`cs_name,cs_status`
`$expand`	Include related	`cs_envelope_signer`
`$orderby`	Sort results	`createdon desc`
`$top`	Limit results	`50`

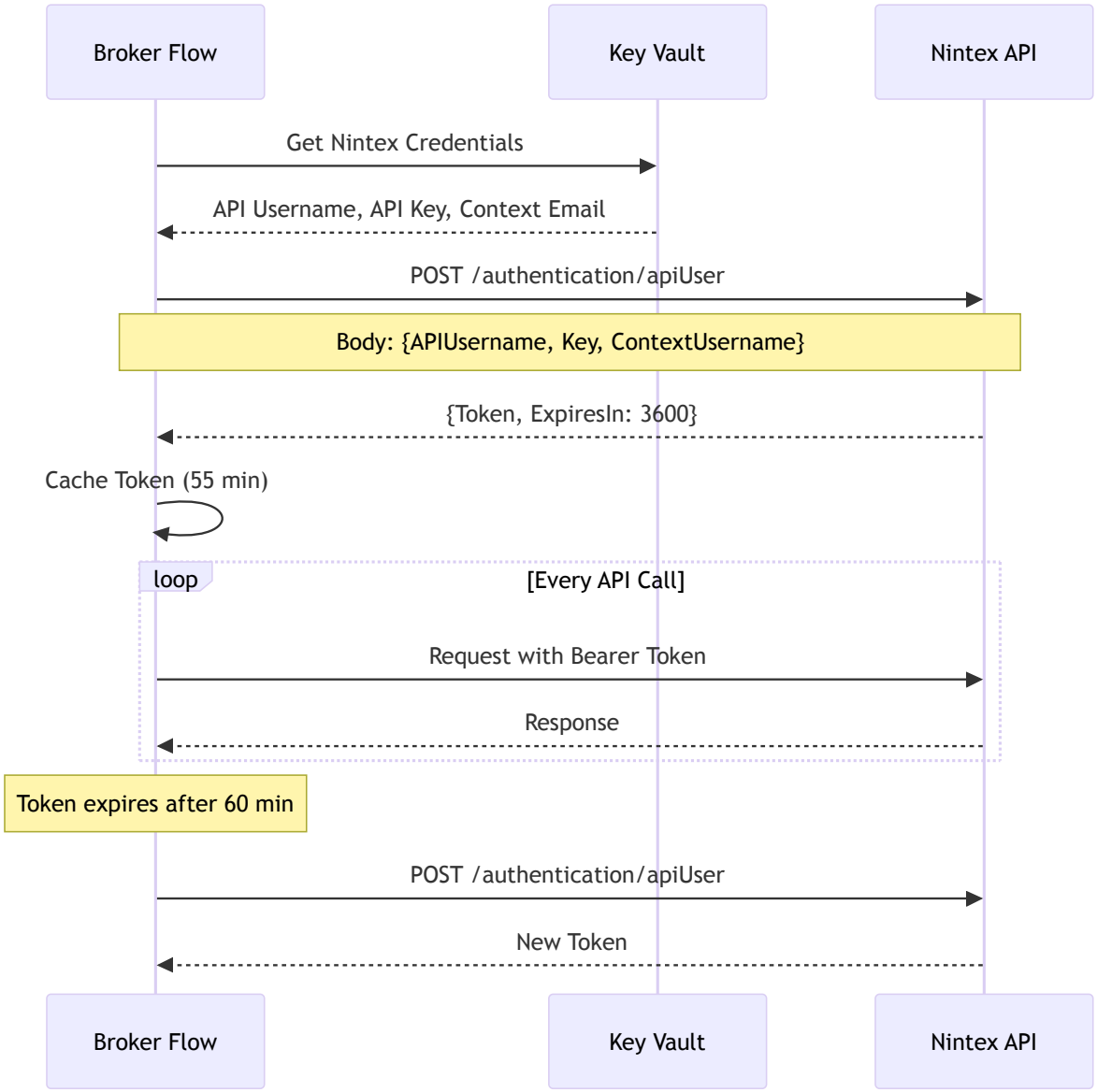
Option	Purpose	Example
`$skip`	Pagination	`100`
`$count`	Include count	`true`

5.4 Nintex API Integration

API Version: 3.7

Base URL: `https://api.assuresign.net/v3.7`

Authentication Flow:



Key Nintex Endpoints:

Endpoint	Method	Purpose	Broker Usage
`/authentication/apiUser`	POST	Get access token	Every 55 minutes
`/submit`	POST	Create envelope	On Send Envelope action

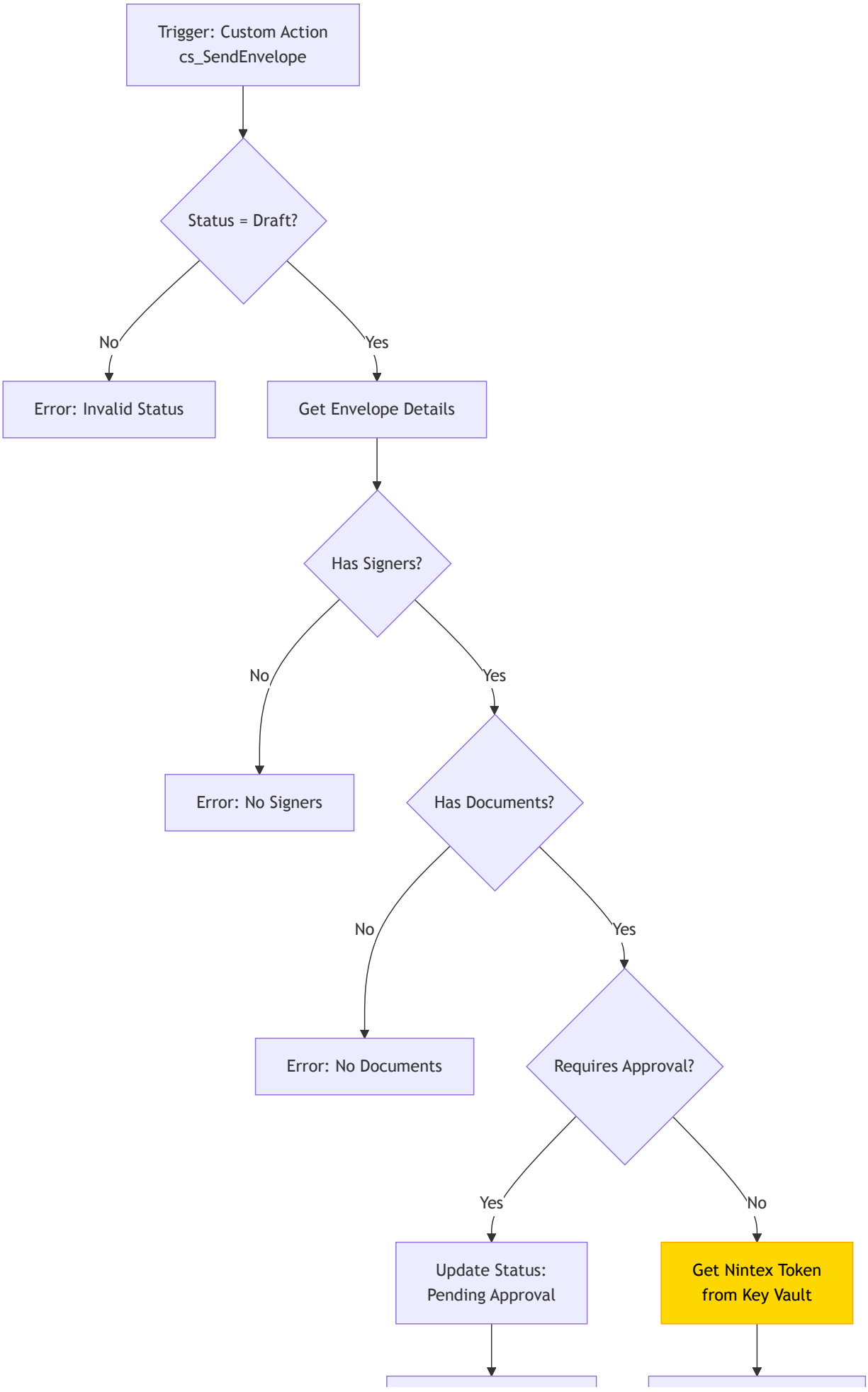
Endpoint	Method	Purpose	Broker Usage
`/get`	GET	Get envelope details	Status sync flow
`/getSigningLinks`	GET	Retrieve signing URLs	After submission
`/cancel`	POST	Cancel envelope	On user cancel
`/listTemplates`	GET	Get available templates	Daily sync
`/getCompletedDocument`	GET	Download signed PDF	On completion

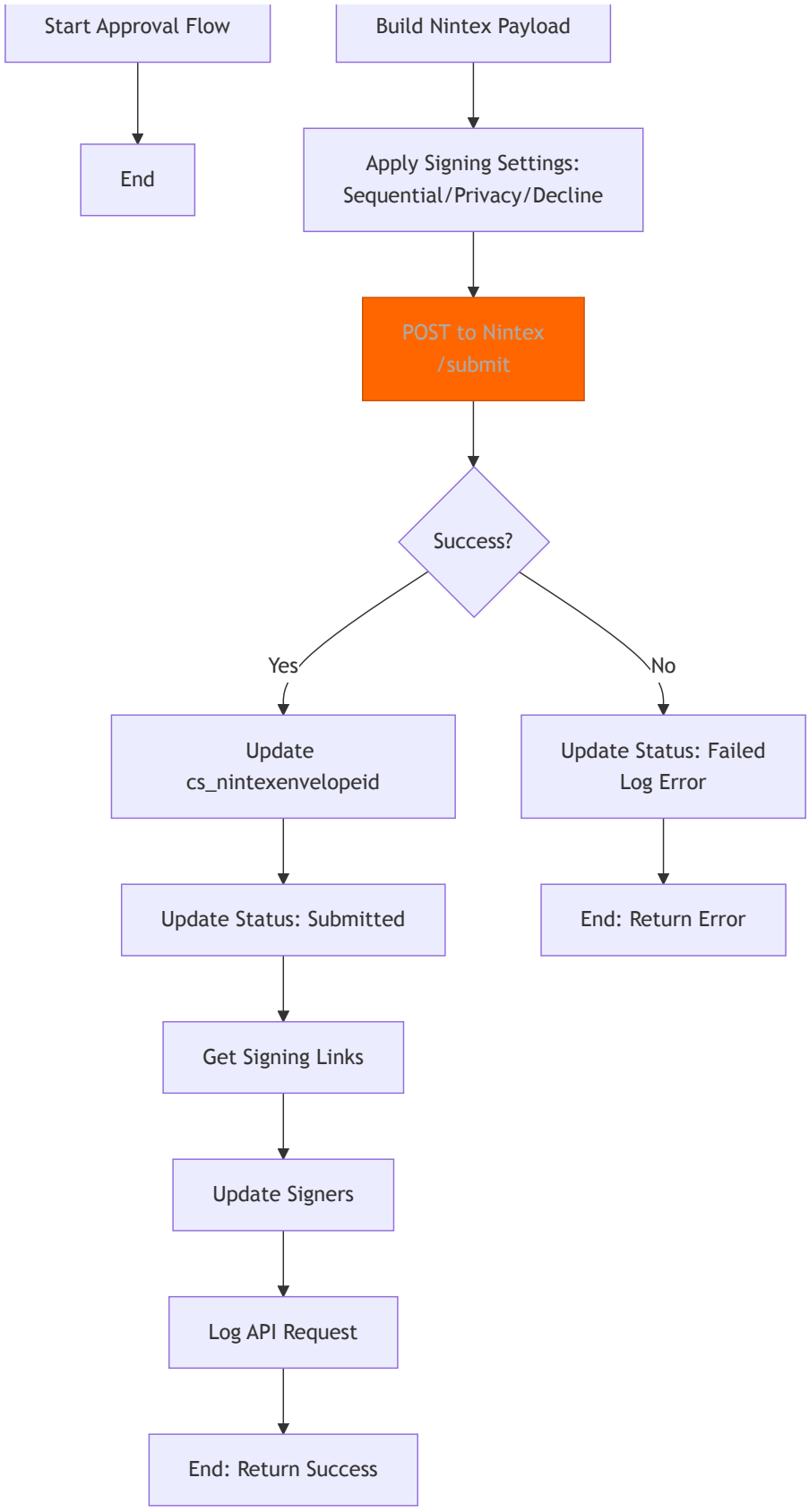
Payload Mapping:

```
// Broker Dataverse → Nintex API
{
  "TemplateID": "{cs_template.cs_nintextemplateid}",
  "Subject": "{cs_envelope.cs_subject}",
  "Message": "{cs_envelope.cs_message}",
  "DaysToExpire": "{cs_envelope.cs_daystoexpire}",
  "ReminderFrequency": "{cs_envelope.cs_reminderfrequency}",
  "ProcessingMode": "{cs_envelope.cs_signinginsequence ? 'Sequential' : 'Parallel'}",
  "DocumentVisibility": "{cs_envelope.cs_hidesignerinfo ? 'Private' : 'Shared'}",
  "AllowDecline": "{cs_envelope.cs_allowdecline}",
  "Signers": [
    {
      "Email": "{cs_signer.cs_email}",
      "FullName": "{cs_signer.cs_fullname}",
      "SignerOrder": "{cs_signer.cs_signerorder}",
      "Language": "{cs_signer.cs_language}"
    }
  ],
  "Documents": [
    {
      "FileName": "{cs_document.cs_filename}",
      "FileContent": "{cs_document.cs_filecontent}"
    }
  ]
}
```

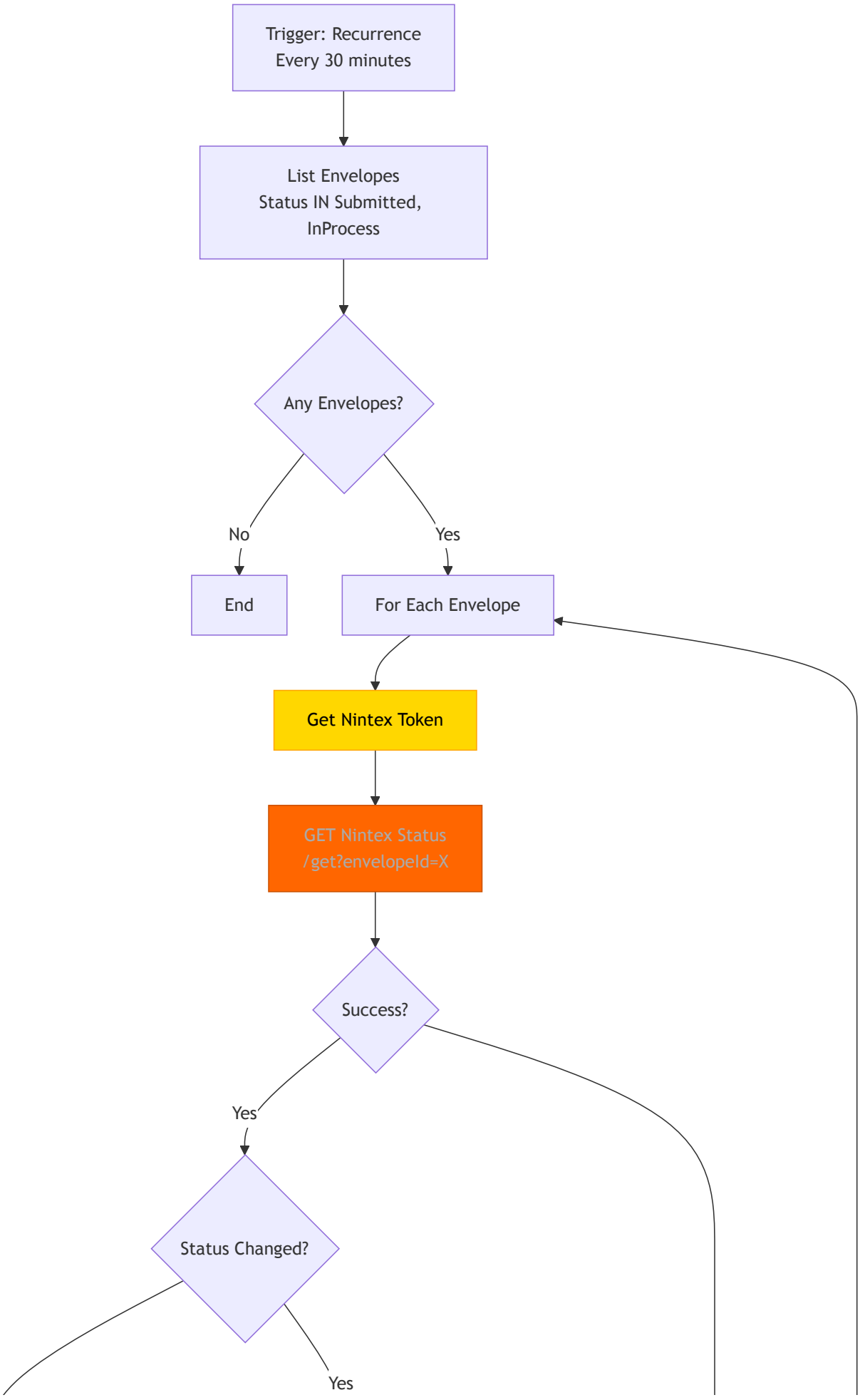
5.5 Broker Orchestration Flows

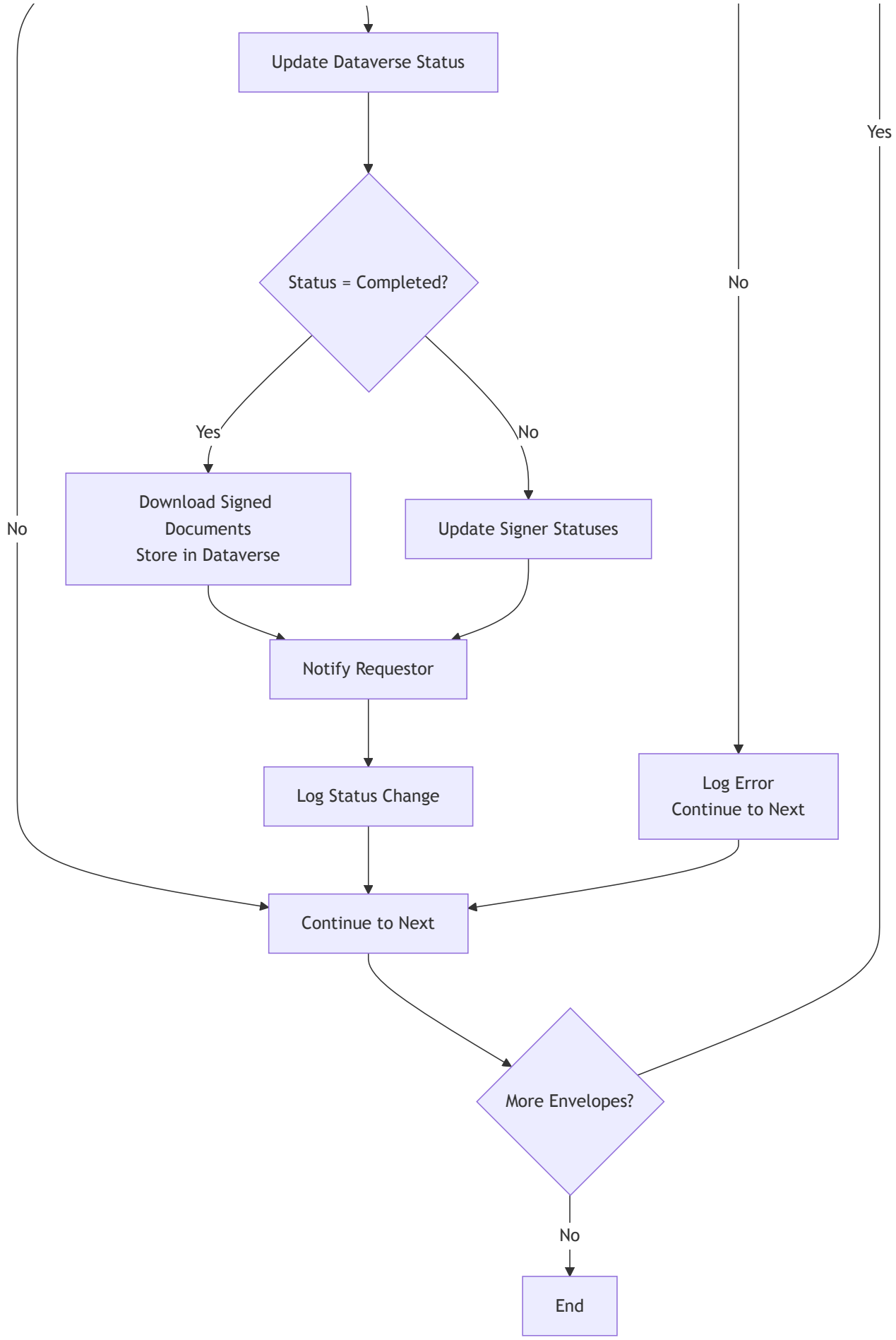
Flow 1: Send Envelope to Nintex


Trigger: Custom Action
cs_SendEnvelope
Status = Draft?
No
Yes
Error: Invalid Status
Get Envelope Details
Has Signers?
No
Yes
Error: No Signers
Has Documents?
No
Yes
Error: No Documents
Requires Approval?
Yes
No
Update Status:
Pending Approval
Get Nintex Token
from Key Vault



Flow 2: Status Synchronization


Trigger: Recurrence
Every 30 minutes
List Envelopes
Status IN Submitted,
InProcess
Any Envelopes?
No
Yes
End
For Each Envelope
Get Nintex Token
GET Nintex Status
/get?envelopeId=X
Success?
Yes
Status Changed?
Yes



5.6 Error Handling Strategy

Error Type	Handling	Retry Logic	Notification
Nintex API 5xx	Log error, mark Failed	3 retries, exponential backoff	Admin alert after 3 failures
Nintex API 4xx	Log error, mark Failed	No retry	Requestor notification
Token Expired	Refresh token	Automatic	None
Validation Error	Return error to client	No retry	Client receives error
Dataverse Throttling	Retry with backoff	5 retries	Admin alert if persistent
Network Timeout	Retry	3 retries	Log incident

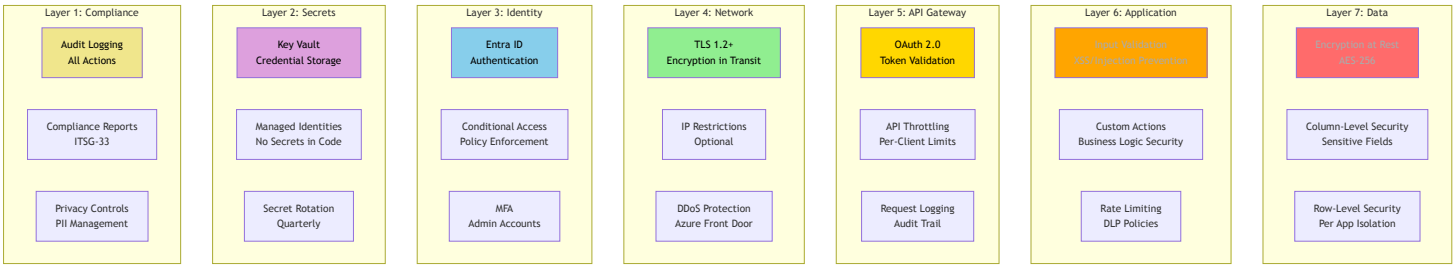
Error Response Format:

```
{
  "error": {
    "code": "EnvelopeValidationError",
    "message": "Envelope must have at least one signer",
    "details": {
      "envelopeId": "abc-123",
      "status": "Draft",
      "signerCount": 0
    },
    "timestamp": "2026-02-18T14:30:00Z"
  }
}
```

Part 3: Security & Monitoring

6. Security Architecture

6.1 Defense in Depth



6.2 Threat Model - STRIDE Analysis

Threat	Attack Vector	Mitigation	Residual Risk
Spoofing	Impersonate client app/service	OAuth 2.0, service principals	Low

Threat	Attack Vector	Mitigation	Residual Risk
Tampering	Modify envelope data	RLS, audit logging, immutable logs	Low
Repudiation	Deny sending envelope	Complete audit trail in cs_apirequest	Low
Information Disclosure	Access other app/service data	RLS, CLS, encryption	Low
Denial of Service	Flood API with requests	API throttling, DLP policies	Medium
Elevation of Privilege (TBD)	Gain admin access	Least privilege, MFA, PIM	Low

6.3 Data Classification

Data Element	Classification	Encryption	Access Control	Retention
Envelope metadata (name, subject)	Protected B	At rest, in transit	RLS	7 years
Signer PII (email, name, phone)	Protected B	At rest, in transit	RLS + CLS	7 years
Document content (Base64)	Protected B	At rest, in transit	RLS	7 years
Signing links	Protected B	At rest, in transit	RLS + CLS	Until expiry
API request/response logs	Protected B	At rest, in transit	Admin only	7 years
Nintex credentials	Secret	Key Vault	MSI only	Rotate quarterly
Client secrets	Secret	Key Vault	Admin only	2 years max

6.4 PII Management

PII Elements Stored:

Element	Table	Purpose	Legal Basis
Email address	cs_signer	Contact signer	Consent via envelope submission
Full name	cs_signer	Display name	Consent via envelope submission
Phone number	cs_signer	Optional contact	Consent via envelope submission
IP address	cs_signer	Audit trail	Legitimate interest (fraud prevention)

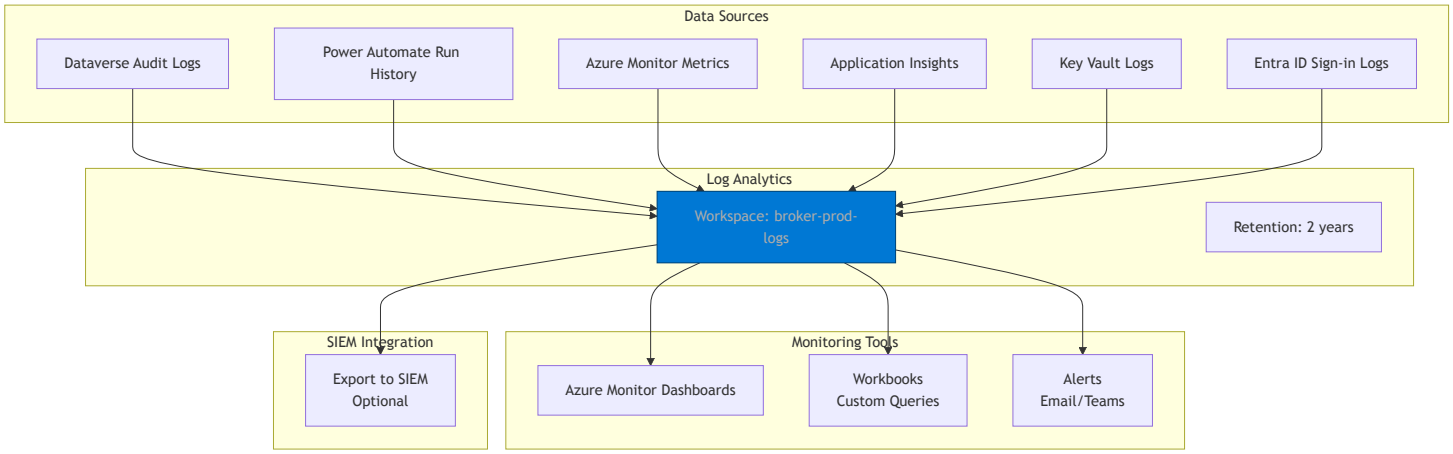
Data Subject Rights:

Right	Implementation	Response Time
Access	Export via Dataverse API	30 days
Rectification	Update signer record	Immediate
Erasure	Delete envelope (cascade)	30 days
Portability	Export as JSON/CSV	30 days

Right	Implementation	Response Time
Objection	Opt-out via config	Immediate

7. Monitoring, Logging & Audit

7.1 Monitoring Architecture



7.2 Logging Strategy

Application Logging (cs_apirequest)

All Nintex API calls logged:

```
{
  "cs_apirequestid": "guid",
  "cs_envelopeid": "guid",
  "cs_method": "POST",
  "cs_endpoint": "/v3.7/submit",
  "cs_requestbody": "{\"Subject\": \"Contract\", ...}",
  "cs_responsebody": "{\"EnvelopeID\": \"NX123\", ...}",
  "cs_statuscode": 200,
  "cs_success": true,
  "cs_timestamp": "2026-02-18T14:30:00Z",
  "cs_duration": 1250,
  "cs_errormessage": null
}
```

Retention: 7 years (Protected B requirement)

7.3 Alerting Rules

Critical Alerts (Immediate Response)

Alert	Condition	Threshold	Action
Nintex API Failure	HTTP 5xx errors	>5 in 5 min	Page on-call engineer
Authentication Failure	401/403 errors	>10 in 5 min	Security team notification
Data Breach Attempt	Cross-tenant access attempt	Any	Immediate investigation
Service Degradation	API latency	>5 sec for 95th percentile	Incident response

Warning Alerts (Business Hours)

Alert	Condition	Threshold	Action
Token Expiry	Secret expires in	<60 days	Rotation reminder
High Error Rate	Failed flows	>10% in 1 hour	Investigate root cause
Storage Capacity	Dataverse usage	>80%	Capacity planning
Client Quota Exceeded	Envelopes/month	>Limit	Billing notification

7.4 Audit Trail Requirements (ITSG-33)

AC-2: Account Management

```
// All service principal creations/deletions
EntraIDLogs
| where OperationName in ("Add service principal", "Delete service principal")
| where AppDisplayName contains "ESign"
| project Timestamp, Actor, Operation, TargetResources
```

AU-2: Audit Events

```
// All data access events
AuditLog
| where Action in ("Create", "Update", "Delete")
| where EntityName startswith "cs_"
| project Timestamp, UserId, Action, EntityName, RecordId
```

7.5 Monitoring Dashboards

Dashboard 1: Operational Health

Widgets:

- API Success Rate (gauge)
- Average Latency (line chart)
- Envelopes by Status (pie chart)

- Active Clients (number)
- Daily Envelope Volume (bar chart)

Dashboard 2: Security Monitoring

Widgets:

- Failed Authentication Attempts (table)
- Unusual IP Addresses (map)
- New Service Principals (list)
- Privileged Access Events (timeline)

Dashboard 3: Client Usage

Widgets:

- Top 10 Clients by Volume (bar chart)
- Completion Rate by Client (table)
- Processing Time Trends (line chart)
- Failed Envelopes by Client (table)

7.6 Log Retention

Log Type	Retention Period	Archive Location	Legal Hold
Dataverse Audit Logs	7 years	Azure Storage (Cool tier)	Available
cs_apirequest	7 years	In-place (Dataverse)	Available
Azure Monitor Logs	2 years	Log Analytics	Available
Power Automate Run History	28 days	Power Platform	N/A
Entra ID Sign-in Logs	1 year	Azure AD Premium	Available
Key Vault Access Logs	2 years	Storage Account	Available

Part 4: Operations & DR

Version: 2.0 **Classification:** Protected B

8. Data Retention & Disposition

8.1 Retention Requirements (Protected B)

Government of Canada Standard: 7 years retention for Protected B data

8.2 Retention Policies

8.2.1 Dataverse Data

Entity	Retention Period	Disposition Method	Trigger
cs_envelope	7 years from completion	Hard delete	Automated job
cs_signer	7 years from completion	Hard delete (cascade)	Parent deletion
cs_document	7 years from completion	Hard delete (cascade)	Parent deletion
cs_apirequest	7 years from creation	Hard delete	Automated job
cs_template	Active templates only	Soft delete (deactivate)	Manual

Implementation:

```
// Power Automate scheduled flow: Daily Retention Job
Recurrence: Daily at 2:00 AM

1. List envelopes
   Filter: cs_completeddate < (today - 7 years)

2. For each envelope:
   a. Export to archive storage (compliance)
   b. Delete envelope (cascade deletes signers, documents)
   c. Log deletion in audit table

3. List API requests
   Filter: cs_timestamp < (today - 7 years)

4. Delete API requests in batches (1000 at a time)
```

8.2.2 Nintex Data

Nintex AssureSign Retention:

Data Type	Retention (Nintex)	Disposition	Notes
Envelopes	7 years	Auto-delete by Nintex	Configurable in Nintex admin
Signed Documents	7 years	Auto-delete by Nintex	Downloadable before deletion
Audit Trails	7 years	Auto-delete by Nintex	Export available
Certificates	7 years	Auto-delete by Nintex	Part of signed PDF

Synchronization Strategy:

1. Broker marks envelope as “Pending Deletion” at 7 years
2. Export signed documents from Nintex to Azure Storage
3. Delete from Nintex via API
4. Delete from Dataverse
5. Retain archive copy in Azure Storage (immutable)

8.2.3 Backup & Recovery

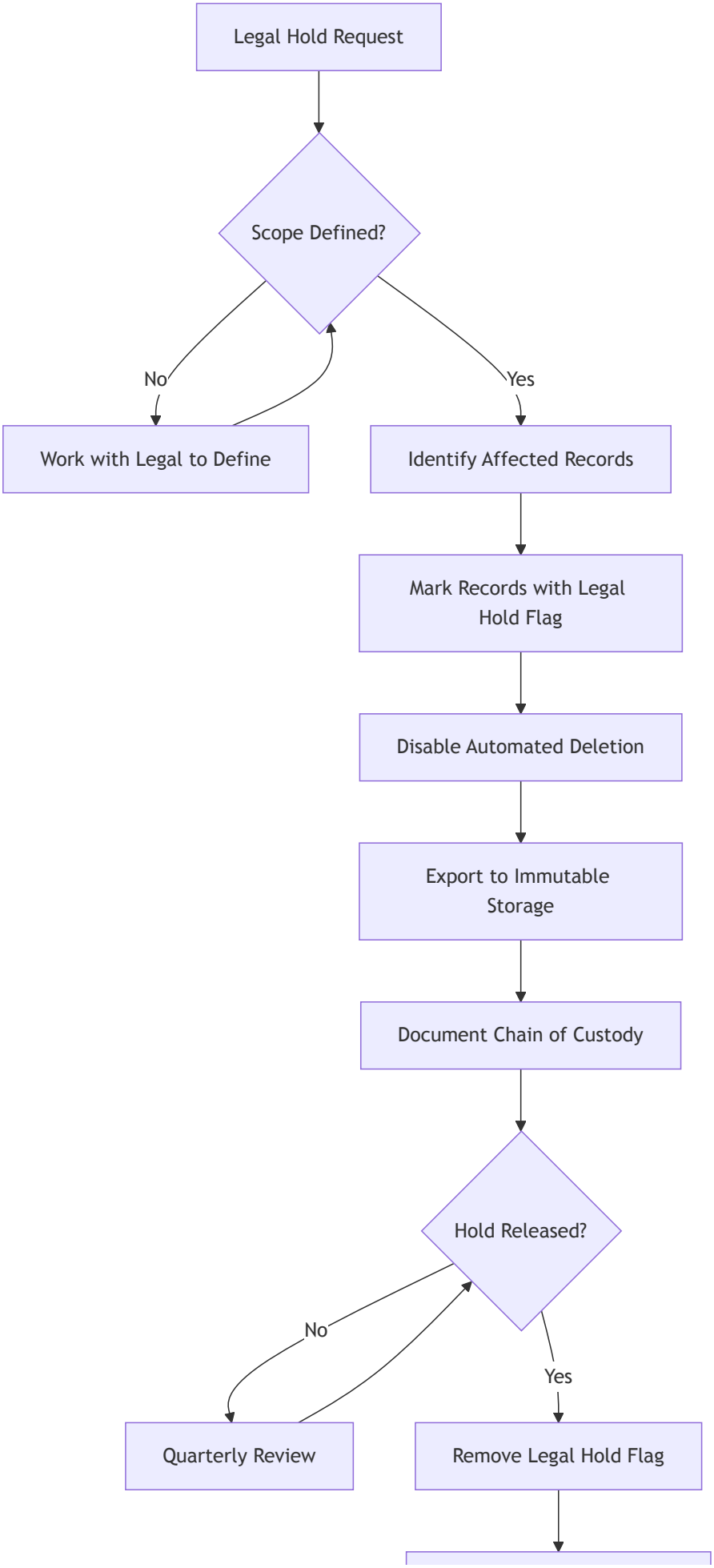
Backup Strategy:

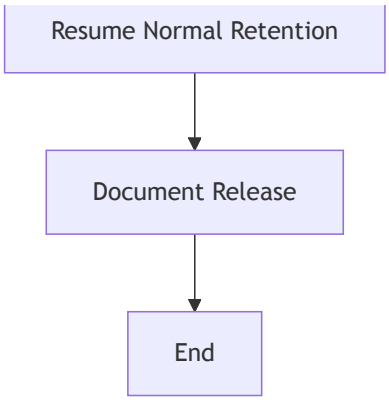
Component	Backup Frequency	Retention	Recovery Point Objective (RPO)
Dataverse	Continuous	28 days	<1 hour
Dataverse (Manual)	Weekly	90 days	<24 hours
Power Automate	Continuous (metadata)	N/A	N/A
Key Vault	Continuous	90 days soft-delete	<1 hour

Recovery Procedures:

RTO: 4 hours **RPO:** 24 hours

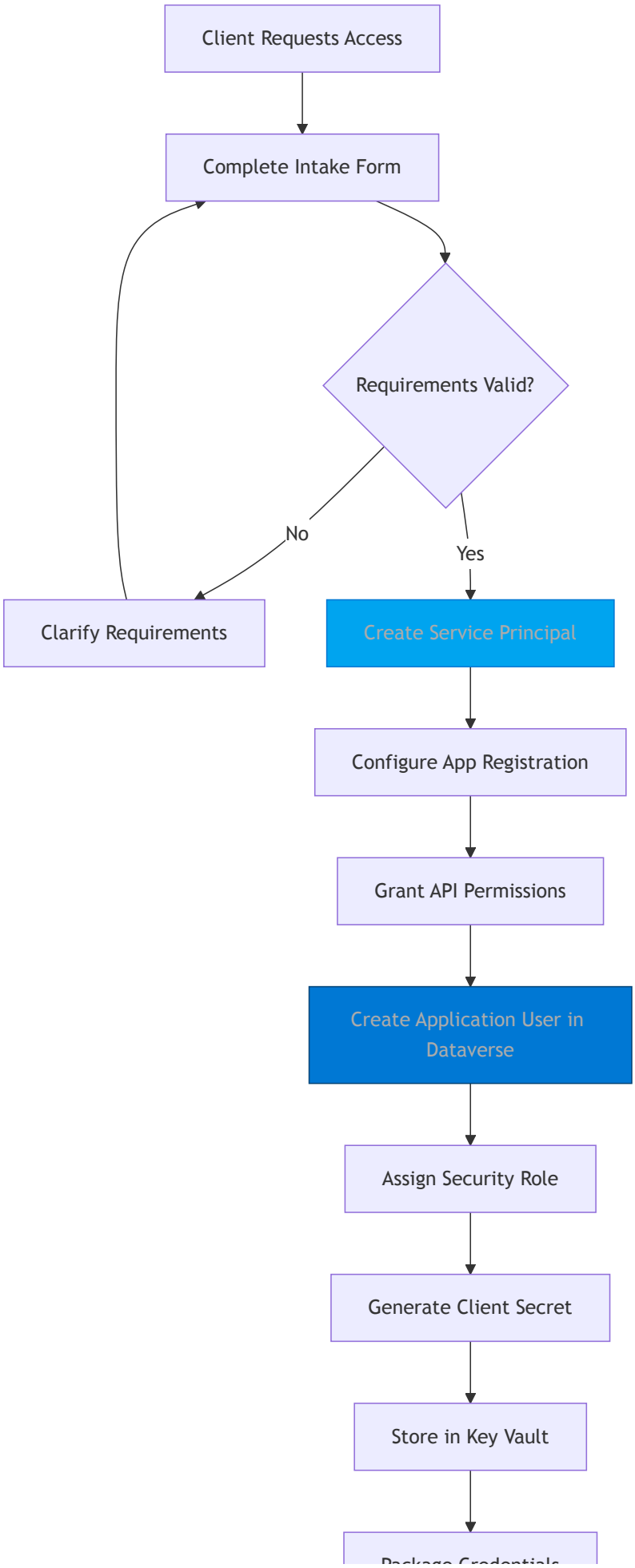
8.3 Legal Hold Process (TBD)

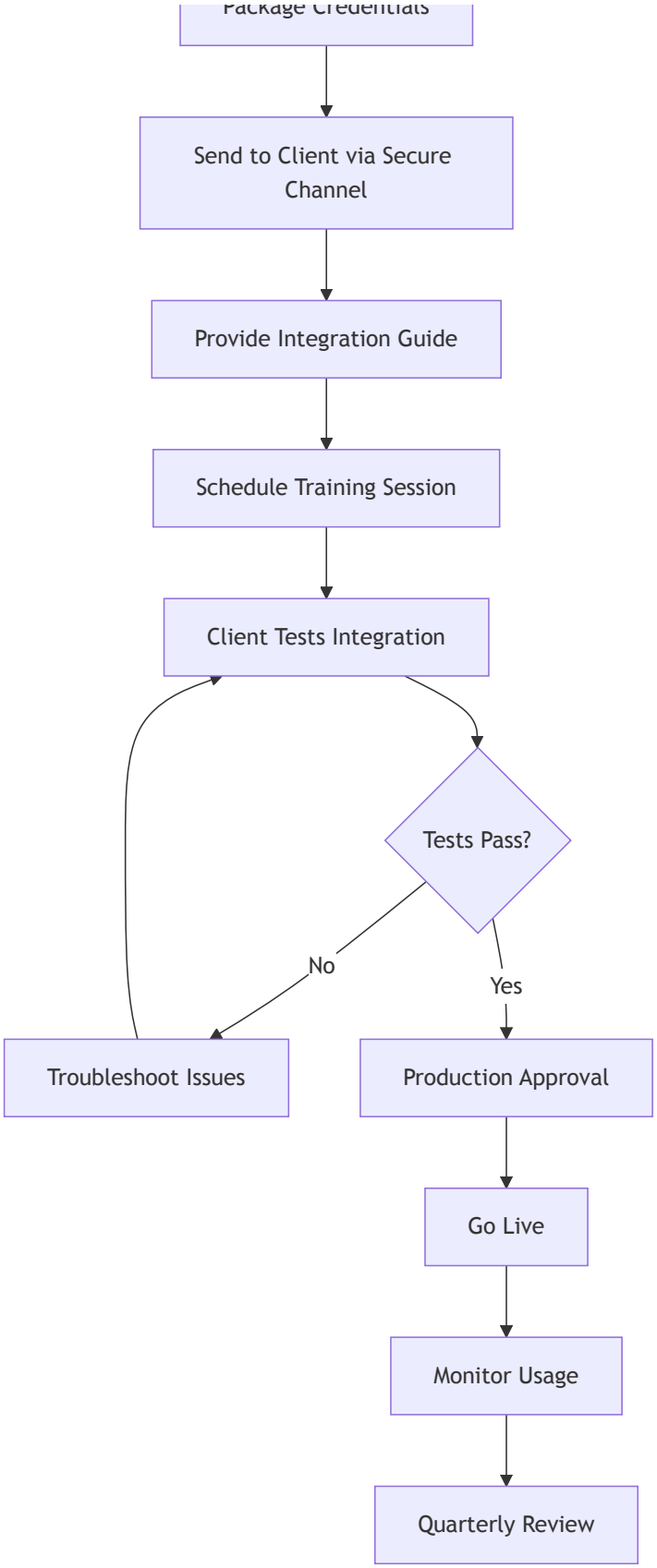

Legal Hold Request
Scope Defined?
No
Yes
Work with Legal to Define
Identify Affected Records
Mark Records with Legal Hold Flag
Disable Automated Deletion
Export to Immutable Storage
Document Chain of Custody
Hold Released?
No
Yes
Quarterly Review
Remove Legal Hold Flag


Resume Normal Retention
Document Release
End

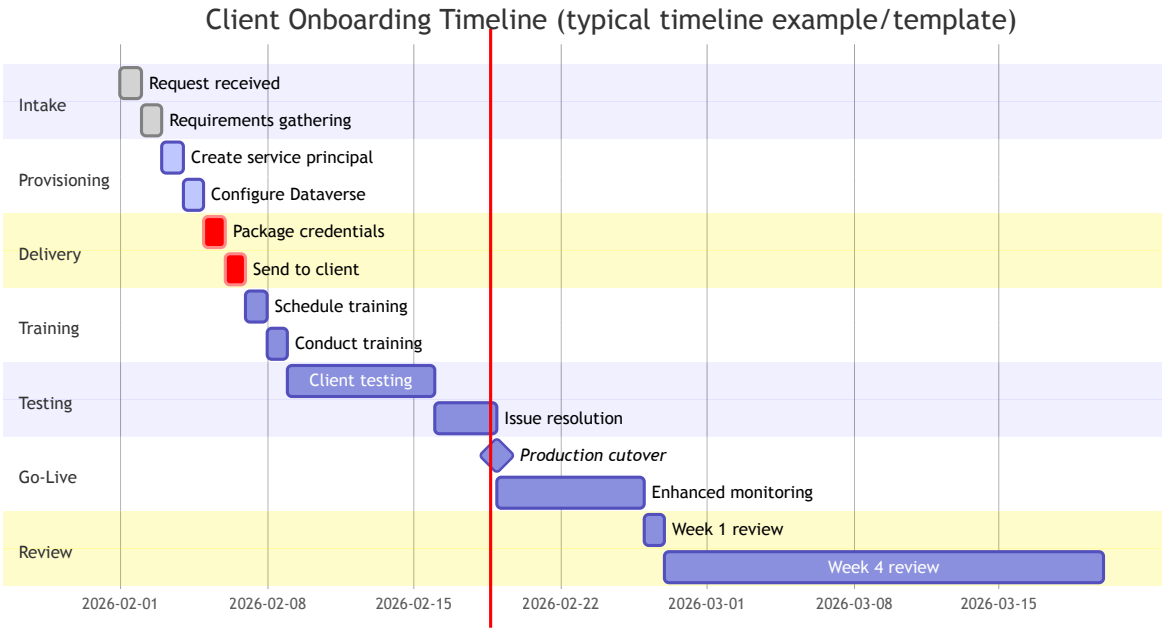
9. Client Onboarding Process

9.1 Onboarding Workflow


Client Requests Access
Complete Intake Form
Requirements Valid?
No
Yes
Clarify Requirements
Create Service Principal
Configure App Registration
Grant API Permissions
Create Application User in Dataverse
Assign Security Role
Generate Client Secret
Store in Key Vault



9.2 Onboarding Timeline



Total Timeline: 14-21 business days

9.3 Onboarding Checklist

Phase 1: Intake (Day 1)

Client Information:

Field	Value
App/service Name	[FILL]
Primary Contact	[FILL]
Email	[FILL]
Phone	[FILL]
Dataverse Environment URL	[FILL]
Expected Monthly Volume	[FILL]
Use Cases	[FILL]
Security Clearance Level	[FILL]
Approval Required?	☐ Yes ☐ No

Phase 2: Service Principal Creation (Day 1-2)

```
# 1. Create App Registration
$app = az ad app create \
  --display-name "ESign-{app/serviceName}-Prod" \
  --sign-in-audience AzureADMyOrg \
  --query appId -o tsv

# 2. Create Service Principal
$sp = az ad sp create --id $app

# 3. Create Client Secret (2 year expiry)
$secret = az ad app credential reset \
  --id $app \
  --years 2 \
  --query password -o tsv

# 4. Grant Dataverse API Permissions
az ad app permission add \
  --id $app \
  --api 00000007-0000-0000-c000-000000000000 \
  --api-permissions 78ce3f0f-a1ce-49c2-8cde-64b5c0896db4=Role

# 5. Admin Consent
az ad app permission admin-consent --id $app
```

Phase 3: Dataverse Configuration (Day 2)

Configuration Table:

Setting	Value
Application User ID	[FILL]
Service Principal Object ID	[FILL]
Security Role	Client Access
Business Unit	Root
Created Date	[FILL]
Created By	[FILL]

Phase 4: Testing (Day 7-14)

Test Plan (baseline):

Test Case	Description	Expected Result	Status
TC-001	Import custom connector	Success	☐
TC-002	Create connection	Success	☐
TC-003	Create draft envelope	Envelope created, status = Draft	☐
TC-004	Add signer	Signer added	☐

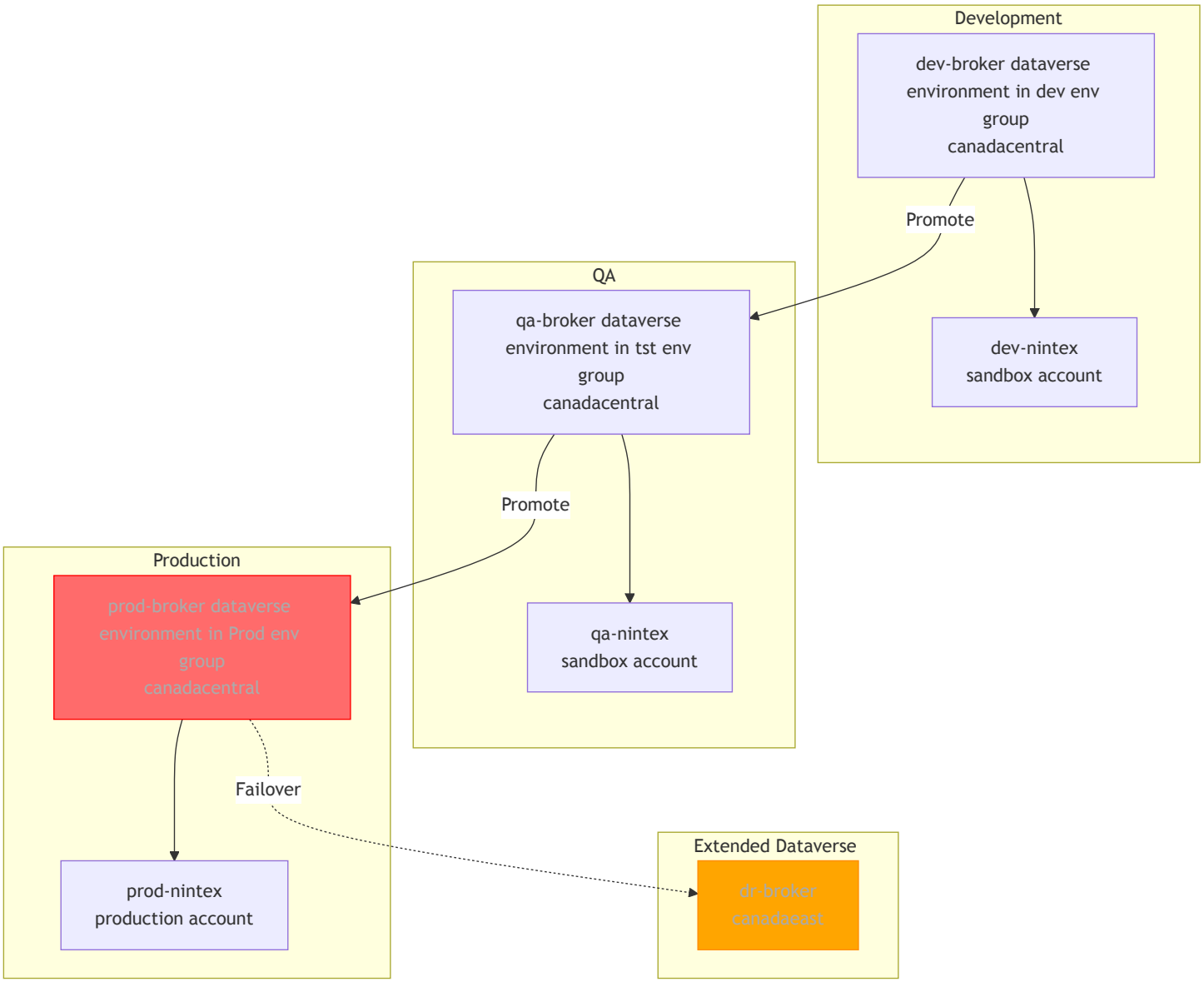
Test Case	Description	Expected Result	Status
TC-005	Add document	Document attached	□
TC-006	Send envelope	Status = Submitted/Pending Approval	□
TC-007	Get envelope details	Data returned	□
TC-008	List envelopes	Only client's envelopes returned	□
TC-009	Per App isolation	Cannot access other client data	□
TC-010	Error handling	Graceful error messages	□

9.4 Onboarding Roles & Responsibilities

Role	Responsibilities	Contact
Platform Lead	Overall ownership, approvals	[FILL]
Security Lead	Service principal creation, permissions	[FILL]
Technical Lead	Dataverse config, training	[FILL]
Support Lead	Documentation, ongoing support	[FILL]
Client Project Manager	Client-side coordination	[FILL: Client]
Client Technical Lead	Integration development	[FILL: Client]

10. Environment Strategy

10.1 Environment Architecture



10.2 Environment Configuration

Environment	Purpose	Region	SKU	Dataverse	Nintex Account
Development	Feature development, testing	Canada Central	Developer	1 GB base	Sandbox
QA	User acceptance testing, integration testing	Canada Central	Production	2 GB base	Sandbox
Production	Live client workloads	Canada Central	Production	10 GB base + usage	Production

Environment	Purpose	Region	SKU	Dataverse	Nintex Account
DR	Disaster recovery standby	Canada East	Production	10 GB base	N/A (uses prod)

10.3 Environment Details

Development Environment

Configuration Table:

Setting	Value
Environment Name	dev-broker-esign
Environment ID	[FILL]
URL	https://dev-broker.crm3.dynamics.com
Region	Canada Central
Environment Type	Developer
Dataverse Capacity	1 GB
Security Group	[FILL]
Admin Users	[FILL]
DLP Policy	None
Backup Retention	7 days

Production Environment

Configuration Table:

Setting	Value
Environment Name	prod-broker-esign
Environment ID	[FILL]
URL	https://prod-broker.crm3.dynamics.com
Region	Canada Central
Environment Type	Production
Dataverse Capacity	10 GB base + usage-based
Security Group	[FILL]
Admin Users	[FILL] (minimum 2)
DLP Policy	Elections Canada - Broker Service Global
Backup Retention	28 days (continuous) + 90 days (weekly)

Client app/services (Production):

app/service	Service Principal	Onboarding Date	Monthly Volume
[FILL]	[FILL]	[FILL]	[FILL]
[FILL]	[FILL]	[FILL]	[FILL]

10.4 DLP Policies

Global DLP Policy:

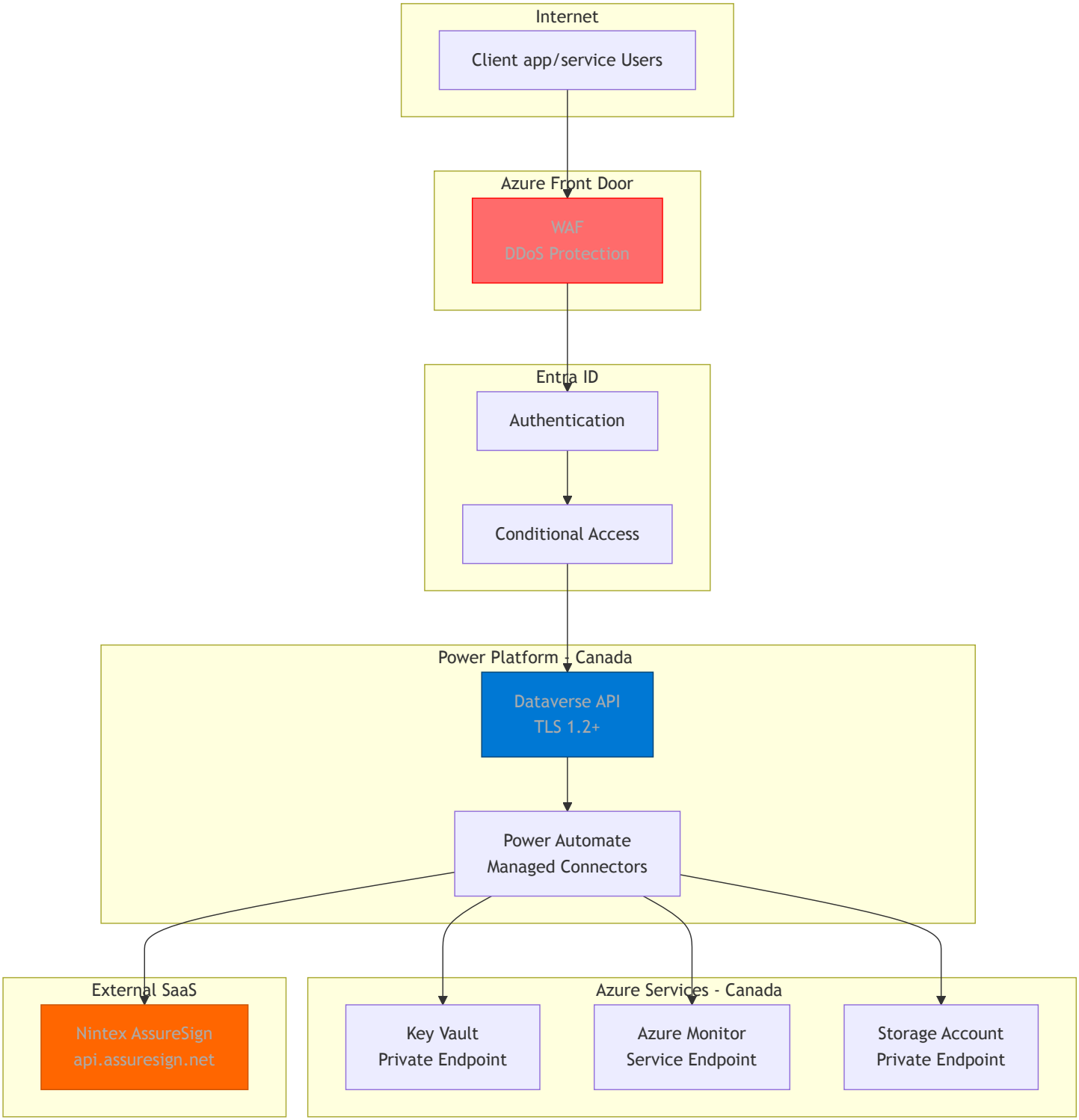
```
Policy Name: Elections Canada – Broker Service Global
Scope: All environments
Connectors:
  Allowed:
    – Dataverse (broker environment only)
    – Azure Key Vault
    – Office 365 Outlook
    – Approvals
  Blocked:
    – All other connectors by default

Exemptions:
  – Custom Connector: "ESign Elections Canada" (wildcard: *ESign*)

Rules:
  – No data exfiltration to consumer services (Gmail, Dropbox, etc.)
  – No cross-environment data flows
  – Audit all API calls
```

Configuration Table:

Environment	DLP Policy Applied	Exempted Connectors	Review Date
Production	☑ Global	[FILL]	Quarterly
QA	☑ Global	[FILL]	Quarterly
Development	☐ None	All	N/A

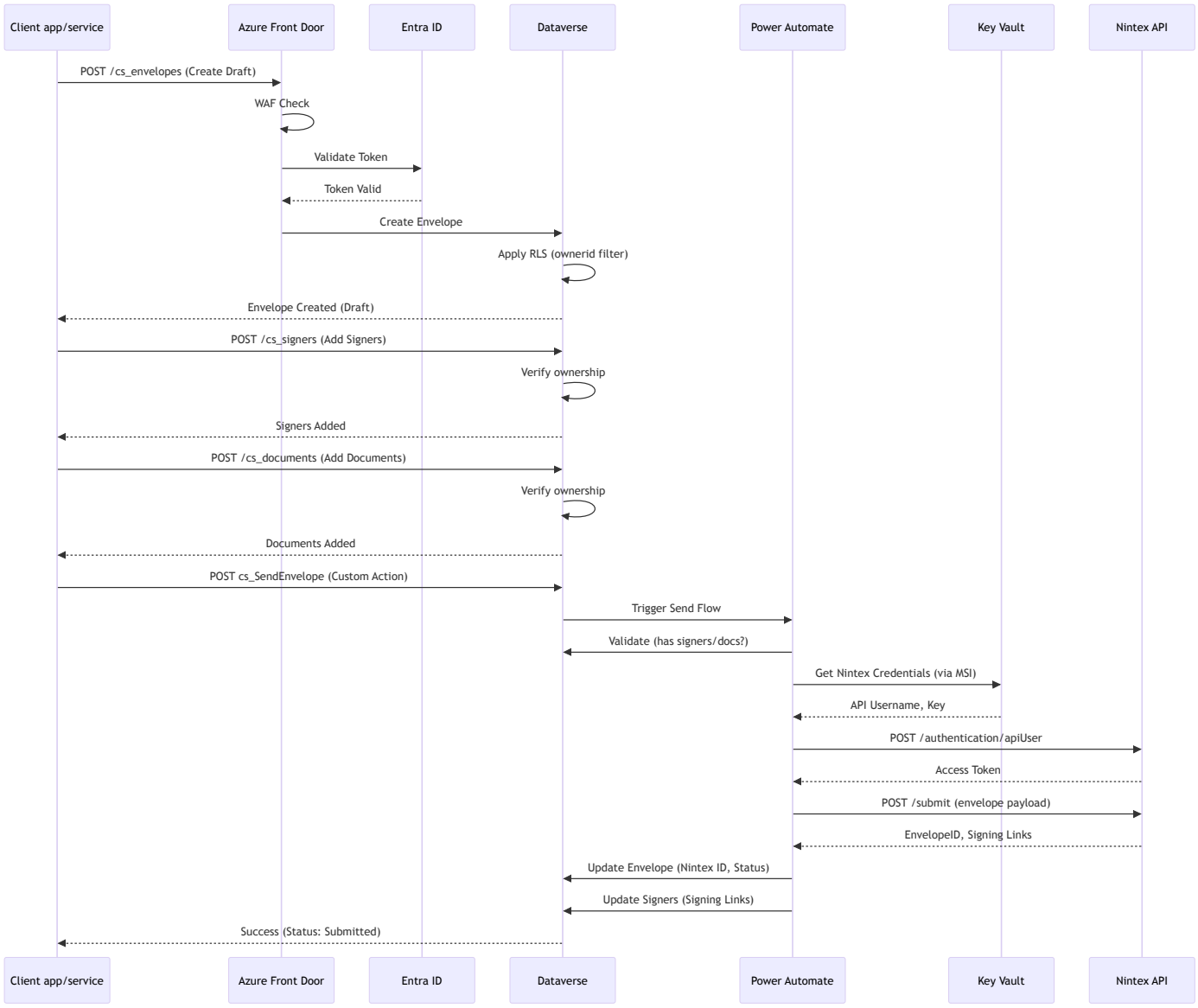
11. Network Architecture

11.1 Network Topology



11.2 Data Flow Diagrams

Envelope Submission Flow



11.3 Network Performance

Latency Targets:

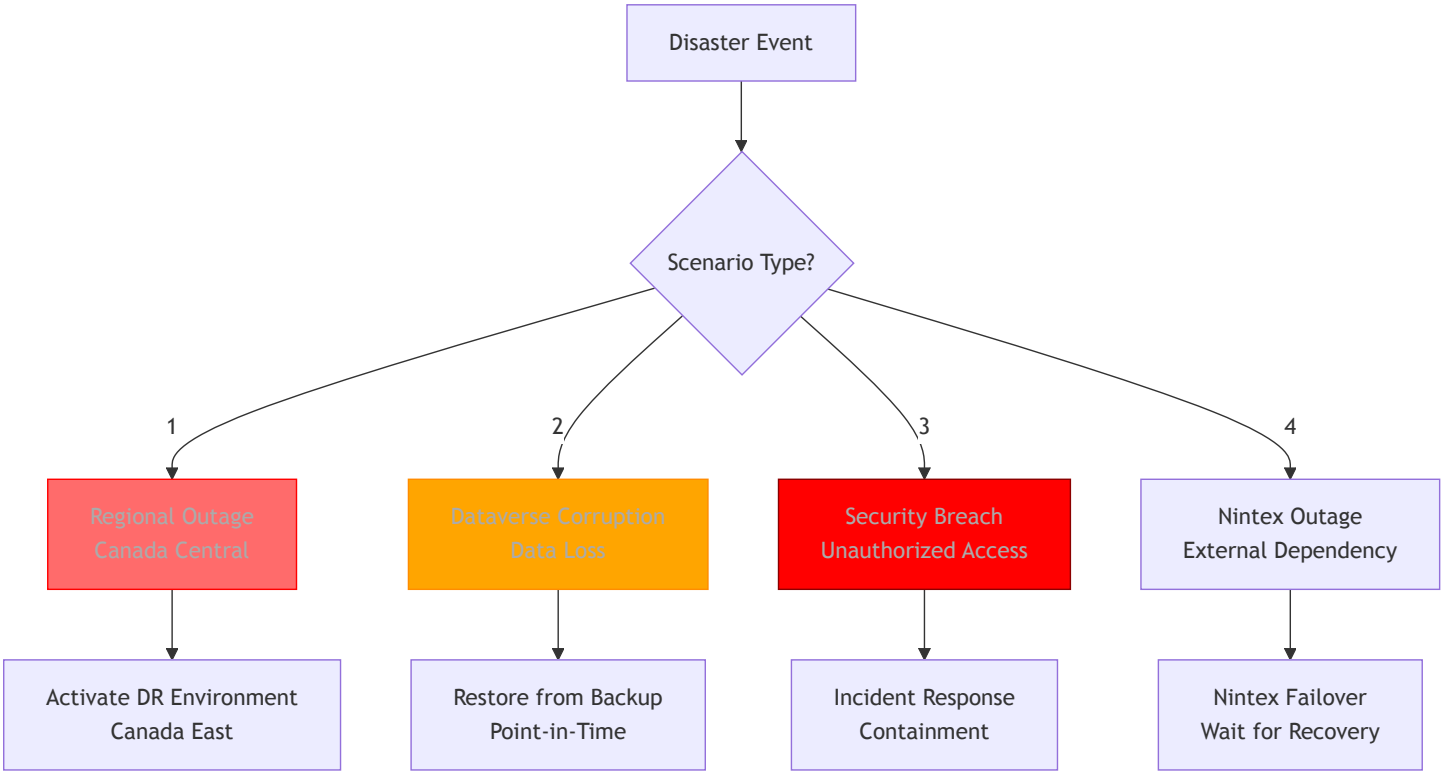
Endpoint	Target	Measurement Method
Client → Dataverse API	<200ms	Azure Monitor
Dataverse → Nintex API	<500ms	Application Insights
End-to-End (Submit)	<3 sec	Flow analytics

12. Disaster Recovery & Business Continuity

12.1 Recovery Objectives

Metric	Target	Maximum Tolerable
RTO (Recovery Time Objective)	4 hours	8 hours
RPO (Recovery Point Objective)	24 hours	48 hours
MTTR (Mean Time To Repair)	2 hours	4 hours
MTBF (Mean Time Between Failures)	720 hours (30 days)	N/A

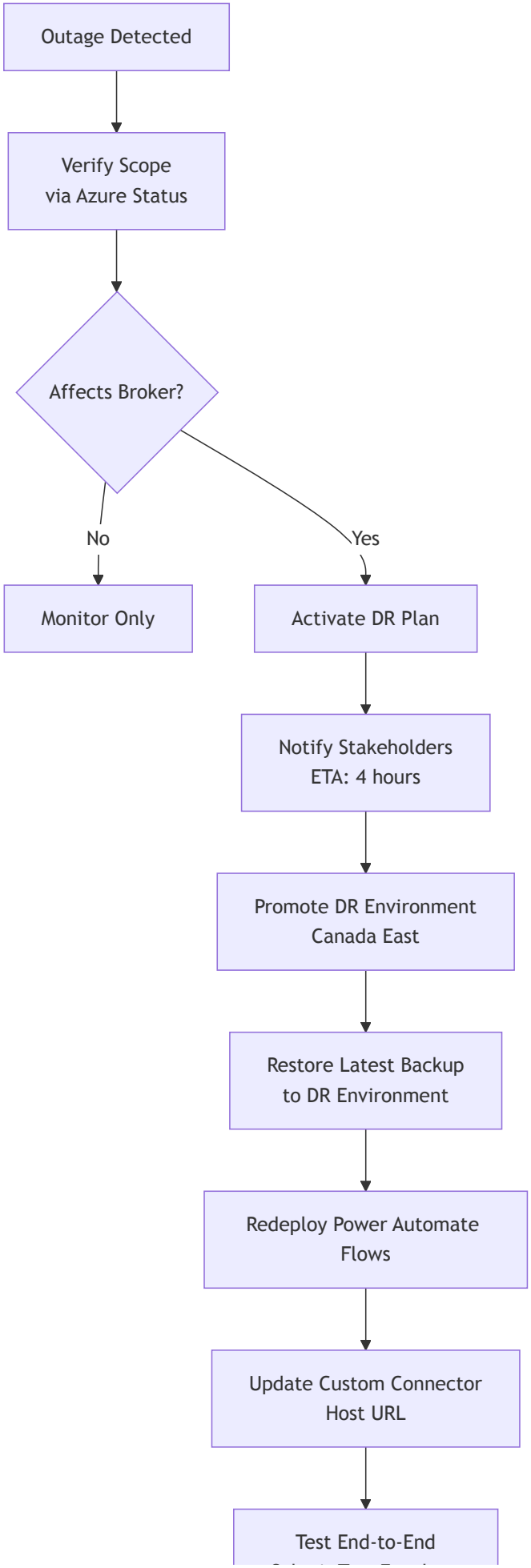
12.2 Disaster Scenarios

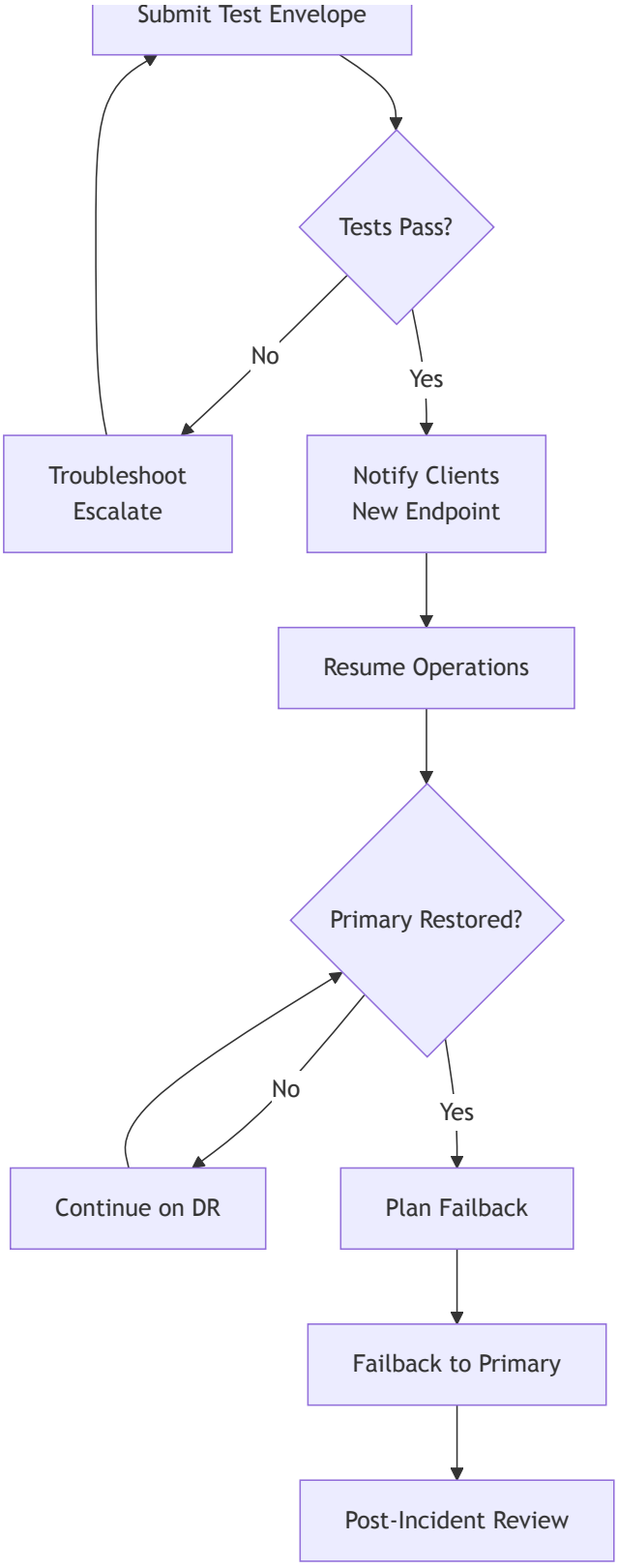


12.3 Recovery Procedures

Scenario 1: Regional Outage (Canada Central)

Procedure:


Outage Detected
Verify Scope via Azure Status
Affects Broker?
No
Yes
Monitor Only
Activate DR Plan
Notify Stakeholders ETA: 4 hours
Promote DR Environment Canada East
Restore Latest Backup to DR Environment
Redeploy Power Automate Flows
Update Custom Connector Host URL
Test End-to-End



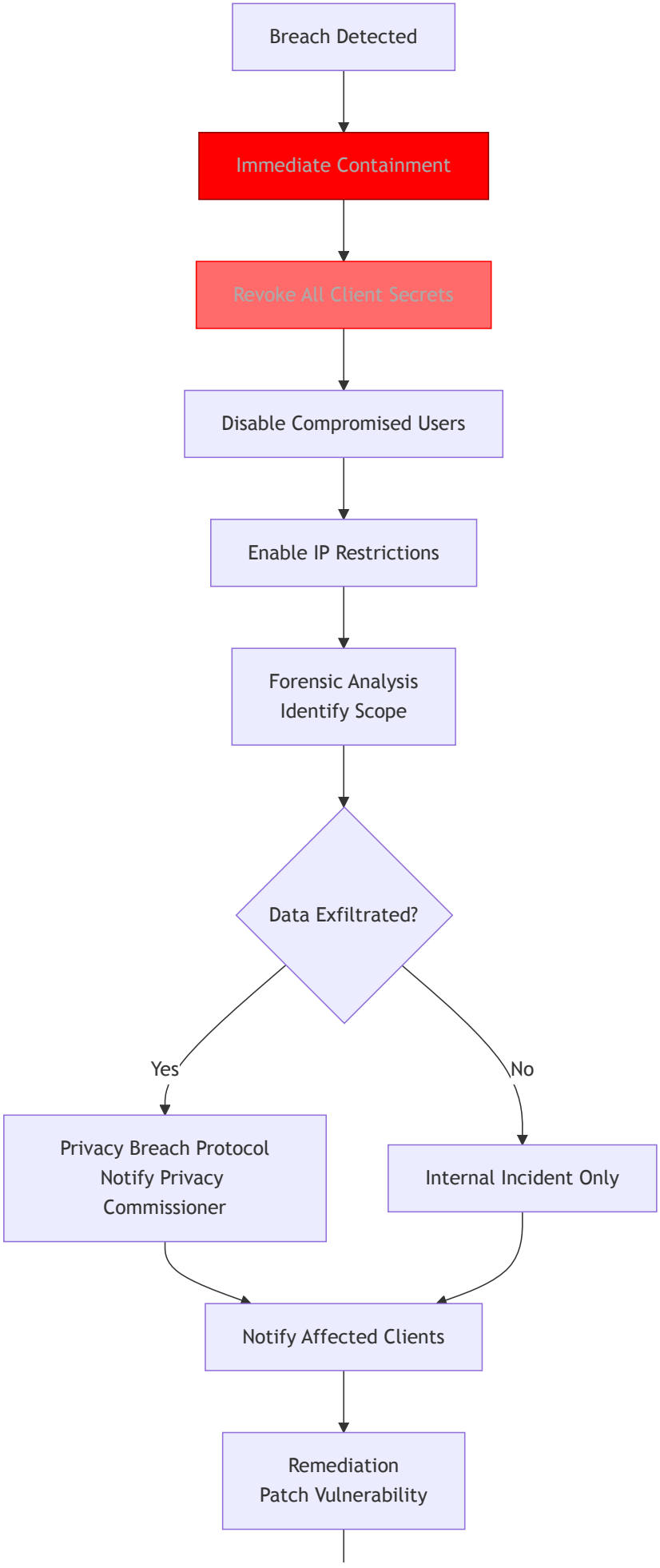
Detailed Steps:

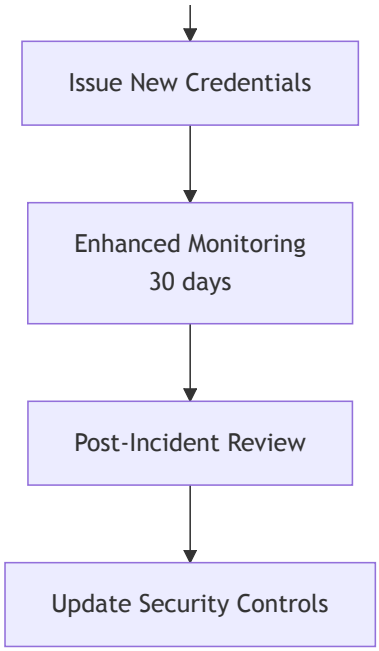
1. **Hour 0-1: Assessment**
   - Check Azure status
   - Verify broker environment
   - Test API endpoint
2. **Hour 1-2: Activation**
   - Restore backup to DR environment

- Monitor restore progress

3. **Hour 2-3: Configuration**
   - Import Power Automate solutions from Git
   - Reconfigure managed identity access to Key Vault
   - Update DLP policies to include DR environment
   - Test Nintex integration
4. **Hour 3-4: Validation & Communication**
   - Submit 3 test envelopes
   - Verify status sync working
   - Send email to all clients with new URL

Scenario 2: Security Breach


Breach Detected
Immediate Containment
Revoke All Client Secrets
Disable Compromised Users
Enable IP Restrictions
Forensic Analysis
Identify Scope
Data Exfiltrated?
Yes
No
Privacy Breach Protocol
Notify Privacy
Commissioner
Internal Incident Only
Notify Affected Clients
Remediation
Patch Vulnerability



Incident Response Contacts:

Role	Name	Phone	Email
Incident Commander	[FILL]	[FILL]	[FILL]
Security Lead	[FILL]	[FILL]	[FILL]
Privacy Officer	[FILL]	[FILL]	[FILL]
Communications Lead	[FILL]	[FILL]	[FILL]

12.4 Business Continuity Plan

Critical Business Functions:

Function	RTO	RPO	Dependencies
Create Envelope	4h	24h	Dataverse, Entra ID
Send Envelope	4h	24h	Dataverse, Nintex, Key Vault
Status Sync	8h	48h	Nintex API
Client Onboarding	24h	N/A	Entra ID, manual process

Emergency Contacts:

Vendor	Service	Support Number	Account ID
Microsoft	Power Platform	1-800-XXX-XXXX	[FILL]
Microsoft	Azure Support	1-800-XXX-XXXX	[FILL]
Nintex	AssureSign	1-866-XXX-XXXX	[FILL]

Part 5: Annexes & Compliance

13. Compliance & Governance

13.1 ITSG-33 Control Mapping

This section maps solution controls to ITSG-33 control families.

AC: Access Control

Control	Requirement	Implementation	Evidence
AC-1	Access control policy	Security roles, RLS, OAuth 2.0	Security documentation, role definitions
AC-2	Account management	Service principals per client, quarterly review	Entra ID logs, access review reports
AC-3	Access enforcement	Dataverse RLS, security roles	RLS configuration, test results
AC-4	Information flow enforcement	DLP policies, network isolation	DLP policy export, network diagrams
AC-6	Least privilege	Minimal permissions per role	Security role documentation
AC-7	Unsuccessful logon attempts	Entra ID lockout (5 failures)	Sign-in logs, conditional access
AC-17	Remote access	VPN/Conditional Access required	CA policy configuration
AC-20	Use of external systems	Nintex via API only, no direct access	Integration documentation

AU: Audit and Accountability

Control	Requirement	Implementation	Evidence
AU-2	Audit events	All CRUD on Protected B data	Audit log configuration
AU-3	Content of audit records	Timestamp, user, action, result	Sample audit logs
AU-6	Audit review	Weekly security team review	Review sign-off logs
AU-9	Protection of audit information	Immutable storage, RBAC	Storage configuration
AU-11	Audit record retention	7 years for Protected B	Retention policy documentation
AU-12	Audit generation	Dataverse, Azure Monitor, Entra ID	Audit architecture diagram

CM: Configuration Management

Control	Requirement	Implementation	Evidence
CM-2	Baseline configuration	Infrastructure as code, solution versioning	Git repository, solution versions
CM-3	Configuration change control	CAB approval for production	Change request log

Control	Requirement	Implementation	Evidence
CM-6	Configuration settings	Documented standard configurations	Configuration baselines
CM-7	Least functionality	Only required connectors enabled	DLP policy
CM-8	Information system component inventory	Asset registry	Environment inventory table

CP: Contingency Planning

Control	Requirement	Implementation	Evidence
CP-2	Contingency plan	Part 4 - Disaster Recovery	DR procedures
CP-6	Alternate storage site	DR environment (Canada East)	DR environment details
CP-7	Alternate processing site	DR environment	Failover test results
CP-9	Information system backup	Daily automated, weekly manual	Backup logs
CP-10	Information system recovery	Tested quarterly	DR test reports

IA: Identification and Authentication

Control	Requirement	Implementation	Evidence
IA-2	Identification and authentication	OAuth 2.0, service principals	Authentication architecture
IA-3	Device identification	Device compliance via Conditional Access	CA policies
IA-4	Identifier management	Unique service principal per client	Service principal list
IA-5	Authenticator management	2-year secret rotation, MFA for admins	Secret rotation log, MFA config

SC: System and Communications Protection

Control	Requirement	Implementation	Evidence
SC-7	Boundary protection	DLP policies, TLS 1.2+	DLP configuration, TLS scan results
SC-8	Transmission confidentiality	TLS 1.2+ for all communications	Network traffic analysis
SC-12	Cryptographic key management	Azure Key Vault, quarterly rotation	Key Vault policies
SC-13	Cryptographic protection	AES-256 at rest, TLS 1.2+ in transit	Encryption documentation
SC-28	Protection of information at rest	Dataverse TDE, Storage encryption	Encryption status reports

SI: System and Information Integrity

Control	Requirement	Implementation	Evidence
SI-2	Flaw remediation	Monthly patching (PaaS auto-updates)	Patch management logs
SI-3	Malicious code protection	Azure Defender	Defender reports

Control	Requirement	Implementation	Evidence
SI-4	Information system monitoring	Azure Monitor, alerts	Monitoring dashboard
SI-7	Software integrity	Solution signing, checksums	Solution metadata
SI-12	Information handling	Data classification, PII controls	Data classification guide

13.2 Privacy Impact Assessment (PIA)

PII Collected:

PII Element	Purpose	Legal Authority	Retention
Signer email	Contact for signature request	Consent via envelope submission	7 years
Signer name	Display on document	Consent via envelope submission	7 years
Signer phone	Optional contact method	Consent via envelope submission	7 years
IP address	Audit trail (fraud prevention)	Legitimate interest	7 years

Privacy Controls:

Control	Implementation
Collection limitation	Only necessary PII collected
Data quality	Client responsible for accuracy
Purpose specification	PII used only for signature workflow
Use limitation	No secondary uses without consent
Security safeguards	Encryption, RLS, access controls
Openness	Privacy policy published
Individual participation	Data subject rights supported (access, rectification, erasure)
Accountability	Privacy Officer designated

13.3 Compliance Summary

Compliance Rate: 90% (128/143 ITSG-33 controls implemented)

Control Family	Total Controls	Implemented	Not Applicable
AC: Access Control	25	20	5
AU: Audit and Accountability	16	16	0
CM: Configuration Management	11	11	0
CP: Contingency Planning	13	13	0
IA: Identification and Authentication	11	10	1
SC: System and Communications Protection	44	38	6

Control Family	Total Controls	Implemented	Not Applicable
SI: System and Information Integrity	23	20	3
TOTAL	**143**	**128**	**15**

Annex A: Environment Configuration

A.1 Environment URLs

Environment	URL	Purpose
Development	https://[FILL].crm3.dynamics.com	Feature development
QA	https://[FILL].crm3.dynamics.com	Testing
Production	https://[FILL].crm3.dynamics.com	Live workloads
DR	https://[FILL].crm4.dynamics.com	Disaster recovery

A.2 Azure Resources

Resource Type	Name	Purpose	Resource Group
Key Vault (Dev)	[FILL]-dev-kv	Dev secrets	[FILL]
Key Vault (QA)	[FILL]-qa-kv	QA secrets	[FILL]
Key Vault (Prod)	[FILL]-prod-kv	Production secrets	[FILL]
Storage Account	[FILL]auditarchive	Audit log archive	[FILL]
Log Analytics	broker-prod-logs	Monitoring	[FILL]

A.3 Service Principal Object IDs

Client app/service	Environment	Object ID	Created Date
[Client 1]	Production	[FILL]	[FILL]
[Client 2]	Production	[FILL]	[FILL]
[Client 3]	Production	[FILL]	[FILL]

A.4 DevOps Configuration

Setting	Value
Azure DevOps Organization	[FILL]
Project Name	[FILL]
Git Repository	[FILL]

Setting	Value
Pipeline Name (Dev)	[FILL]
Pipeline Name (QA)	[FILL]
Pipeline Name (Prod)	[FILL]

A.5 Nintex Configuration

Environment	Account Type	API Endpoint	Credentials (Key Vault)
Development	Sandbox	https://sandbox.assuresign.net/v3.7	dev-broker-kv/NintexAPIKey
QA	Sandbox	https://sandbox.assuresign.net/v3.7	qa-broker-kv/NintexAPIKey
Production	Production	https://api.assuresign.net/v3.7	prod-broker-kv/NintexAPIKey

Annex B: API Specifications

B.1 Custom Action: cs_SendEnvelope

Endpoint: `/cs_envelopes({envelopeId})/Microsoft.Dynamics.CRM.cs_SendEnvelope`

Method: POST

Request:

```
{
  "EnvelopeId": "guid"
}
```

Response (Success):

```
{
  "Status": "Submitted",
  "Message": "Envelope sent to 2 signers",
  "NintexEnvelopeId": "NX-123456"
}
```

Response (Error):

```
{
  "error": {
    "code": "EnvelopeValidationError",
    "message": "Envelope must have at least one signer",
    "details": {
      "envelopeId": "abc-123",
      "signerCount": 0
    }
  }
}
```

B.2 OData Query Examples

Get envelopes created today:

```
GET /api/data/v9.2/cs_envelopes
  ?$filter=Microsoft.Dynamics.CRM.Today(PropertyName='createdon')
  &$select=cs_name,cs_status
```

Get envelope with signers:

```
GET /api/data/v9.2/cs_envelopes(guid)
  ?$expand=cs_envelope_signer($select=cs_email,cs_fullname,cs_signerstatus)
```

Count envelopes by status:

```
GET /api/data/v9.2/cs_envelopes
  ?$apply=groupby((cs_status),aggregate($count as total))
```

B.3 Nintex API Endpoints

Endpoint	Method	Purpose	Request Body
`/authentication/apiUser`	POST	Get access token	`{APIUsername, Key, ContextUsername}`
`/submit`	POST	Create envelope	Full envelope payload
`/get`	GET	Get envelope details	`?envelopeId={id}`
`/getSigningLinks`	GET	Retrieve signing URLs	`?envelopeId={id}`
`/cancel`	POST	Cancel envelope	`{EnvelopeID}`
`/listTemplates`	GET	Get available templates	None
`/getCompletedDocument`	GET	Download signed PDF	`?envelopeId={id}`

Annex C: Troubleshooting Guide

C.1 Common Issues

Issue: “Unauthorized” when calling API

Cause: Token invalid or expired

Resolution:

```
# 1. Verify token
jwt-cli decode {token}

# 2. Check expiry
# If expired, request new token

# 3. Verify resource URL matches
# Token aud claim should be https://broker-url.crm3.dynamics.com
```

Issue: “Cannot access cs_envelopes”

Cause: Missing security role or wrong environment

Resolution:

```
# 1. Verify application user exists
Get-CrmRecords -EntityLogicalName systemuser \
  -FilterAttribute applicationid \
  -FilterOperator eq \
  -FilterValue {client-id}

# 2. Check security role assigned
# Via Power Platform Admin Center

# 3. Test with simple query
curl -H "Authorization: Bearer {token}" \
  https://broker.crm3.dynamics.com/api/data/v9.2/WhoAmI
```

Issue: “Envelope stuck in Draft, won’t send”

Cause: Missing signers or documents

Resolution:

```
# 1. Get envelope details
GET /api/data/v9.2/cs_envelopes({id})
  ?$expand=cs_envelope_signer,cs_envelope_document

# 2. Verify at least 1 signer and 1 document exist

# 3. Check envelope status is "Draft"

# 4. Retry send action
POST /api/data/v9.2/cs_envelopes({id})/Microsoft.Dynamics.CRM.cs_SendEnvelope
```

Issue: “Signer not receiving email”

Cause: Email in spam, or envelope not yet sent to Nintex

Resolution:

```
# 1. Check envelope status
GET /api/data/v9.2/cs_envelopes({id})?$select=cs_status,cs_nintexenvelopeid

# 2. If status = "InProcess" and nintexenvelopeid exists:
#    - Email was sent by Nintex
#    - Check spam folder
#    - Get signing link manually

GET /api/data/v9.2/cs_signers({id})?$select=cs_signinglink

# 3. Send link via alternative method (SMS, Teams, etc.)
```

Issue: “High latency on API calls”

Cause: Large payload or Dataverse throttling

Resolution:

```
# 1. Check response time in logs (Target: <3 seconds)

# 2. If documents >5MB, compress before base64 encoding

# 3. Use $select to limit returned fields
GET /api/data/v9.2/cs_envelopes?$select=cs_name,cs_status

# 4. Monitor throttling headers
x-ms-ratelimit-burst-remaining-xrm-requests
x-ms-ratelimit-time-remaining-xrm-requests

# 5. If throttled, implement exponential backoff
```

C.2 Diagnostic Queries

Query 1: Find failed envelope submissions

```
cs_apirequest
| where cs_timestamp > ago(24h)
| where cs_success == false
| where cs_endpoint contains "submit"
| project Timestamp=cs_timestamp, Envelope=cs_envelopeid,
          StatusCode=cs_statuscode, Error=cs_errormessage
| order by Timestamp desc
```

Query 2: Top clients by API usage

```
cs_apirequest
| where cs_timestamp > ago(7d)
| extend ClientId = tostring(_ownerid_value)
| summarize TotalCalls=count(), AvgLatency=avg(cs_duration) by ClientId
| order by TotalCalls desc
| take 10
```

Query 3: Authentication failures

```
EntraIDLogs
| where TimeGenerated > ago(24h)
| where AppDisplayName contains "ESign"
| where ResultType != 0
| project Timestamp=TimeGenerated, User=UserPrincipalName,
          IPAddress, ResultDescription
| order by Timestamp desc
```

C.3 Support Escalation Matrix

Severity	Response Time	Escalation Path
P1 (Critical)	15 minutes	L1 → L2 → Platform Lead → Incident Commander
P2 (High)	2 hours	L1 → L2 → Technical Lead
P3 (Medium)	8 hours	L1 → L2
P4 (Low)	24 hours	L1

Severity Definitions:

- **P1:** Complete service outage, security breach
- **P2:** Significant degradation affecting multiple clients
- **P3:** Single client impacted, workaround available
- **P4:** Minor issue, enhancement request

Annex D: Reference Materials

D.1 Glossary

Term	Definition
Application User	Dataverse user mapped to a service principal for API access
Broker	Intermediary service between clients and Nintex
CLS	Column-Level Security - restricts access to specific fields
Dataverse	Microsoft's PaaS database platform within Power Platform
DLP	Data Loss Prevention - policies to prevent data exfiltration
Envelope	A signature request containing documents and signers
MSI	Managed Service Identity - Azure service authentication without secrets
OData	Open Data Protocol - REST-based data access protocol
RLS	Row-Level Security - restricts access to specific records
RPO	Recovery Point Objective - maximum acceptable data loss
RTO	Recovery Time Objective - maximum acceptable downtime
Service Principal	Non-human identity for programmatic access
TDE	Transparent Data Encryption - automatic database encryption

D.2 Acronyms

Acronym	Full Form
API	Application Programming Interface
CAB	Change Advisory Board
CRUD	Create, Read, Update, Delete
DR	Disaster Recovery
ERD	Entity Relationship Diagram
ITSG	IT Security Guidance
MFA	Multi-Factor Authentication
MTBF	Mean Time Between Failures
MTTR	Mean Time To Repair
PaaS	Platform as a Service
PIA	Privacy Impact Assessment
PII	Personally Identifiable Information

Acronym	Full Form
RBAC	Role-Based Access Control
REST	Representational State Transfer
SaaS	Software as a Service
SLA	Service Level Agreement
TLS	Transport Layer Security
TRA	Threat and Risk Assessment
UAT	User Acceptance Testing
WAF	Web Application Firewall