A black background with a white line

Description automatically generated

A close up of a logo

AI-generated content may be incorrect.

JUL – Angola’s National Logistics Single Window (Phase 1)

High-Level Design

**Document history & quality control**

*This document is copyright and confidential to Ad Ports. No part may be reproduced in any manner what-so-ever without the prior written permission of Abu Dhabi Ports*

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date of Issue | Author | Summary Of Changes |
| 1.0 | December 2, 2025 | Gracia Vijayan | Initial Template - Basic system architecture framework with core microservices design, preliminary technology stack selection, and foundational infrastructure requirements. |
| 1.1 | December 5, 2025 | Linoy Pappachan Malakkaran | Enhanced Integration Layer - Added ASYCUDA integration specifications, SINTECE API definitions, initial security framework with Keycloak implementation, and basic deployment architecture. |
| 1.2 | December 9, 2025 | Linoy Pappachan Malakkaran | Data Architecture Refinement - Implemented WCO Data Model Version 3.10 compliance, added master data management specifications, defined database-per-service pattern, and enhanced data flow diagrams. |
| 1.3 | December 12, 2025 | Linoy Pappachan Malakkaran | Infrastructure Enhancement - Defined Kubernetes cluster architecture for three environments (Dev/Staging/Production), added network segmentation design, implemented monitoring stack with Prometheus/Grafana, and enhanced backup/recovery procedures. |
| 1.4 | December 15, 2025 | Linoy Pappachan Malakkaran | Security & Compliance Update - Added Multi-Factor Authentication requirements, implemented TLS 1.3 encryption standards, defined network security zones (DMZ/Application/Data/Management), enhanced audit logging, and added WTO TFA compliance alignment. |
| 2.0 | December 17, 2025 | Linoy Pappachan Malakkaran | Production Ready Release - Consolidated microservices architecture, upgraded technology stack (Angular 20+, ASP.NET Core 10.0, PostgreSQL 15+, Keycloak 23+), added Camunda 8 workflow engine and Kong API Gateway, implemented complete integration flows with SLA definitions, added 18 architectural diagrams, defined performance targets (99.9% uptime, <3s response, 100 TPS, 10K concurrent users), and comprehensive CI/CD pipeline with GitLab/ArgoCD. |
| 2.1 | December 22, 2025 | Linoy Pappachan Malakkaran | Architectural diagrams added |

**Next Review Date:**

Although this document is classified as FINAL, it is still subject to the Continuous Service Improvement process; therefore, it is not necessary to wait until the annual review date to make improvements.

**Note:** As this document is a controlled document, any updates or improvements must be implemented under strict change control via the process owner. All reviewers must review and approve all updates.

***Document owner:***

|  |  |
| --- | --- |
| Role | Name |
| Document Owner | Ad Ports Technical Team |
| Deputy Document Owner |  |

**Signatures**

***This document has been reviewed by:***

|  |  |  |  |
| --- | --- | --- | --- |
| Role | Name | Signature | Date |
| Manager - Architecture and Business Analysis | Indranil Majumder |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

***This document has been approved by:***

|  |  |  |  |
| --- | --- | --- | --- |
| Role | Name | Signature | Date |
| Manager - Architecture and Business Analysis | Indranil Majumder |  |  |
|  |  |  |  |

Table of Contents

[1 Introduction 7](#_Toc217232390)

[1.1 Purpose and Audience 7](#_Toc217232391)

[1.2 Definitions, Acronyms and Abbreviations 7](#_Toc217232392)

[1.3 References 7](#_Toc217232393)

[1.4 Business Vision 8](#_Toc217232394)

[1.4.1 Business Domain 8](#_Toc217232395)

[1.4.2 Data Domain 8](#_Toc217232396)

[1.4.3 Application Domain 8](#_Toc217232397)

[1.4.4 Technology Domain 8](#_Toc217232398)

[1.4.5 Supply Chain Domain 8](#_Toc217232399)

[2 Enterprise Architecture 8](#_Toc217232400)

[2.1 Project Phasing 8](#_Toc217232401)

[2.2 Technology and Architecture Assumptions and Dependencies 10](#_Toc217232402)

[3 Architecture Objectives 11](#_Toc217232403)

[4 Architecture Constraints 12](#_Toc217232404)

[4.1 Standards 12](#_Toc217232405)

[4.1.1 Technology Standards 12](#_Toc217232406)

[4.1.2 Messaging Standards 12](#_Toc217232407)

[4.1.3 XML Standards 12](#_Toc217232408)

[4.1.4 Portability 12](#_Toc217232409)

[5 Architecture Model 12](#_Toc217232410)

[5.1 Component Descriptions 12](#_Toc217232411)

[5.2 Presentation Layer 13](#_Toc217232412)

[5.2.1 Key Interface Modules 13](#_Toc217232413)

[5.2.2 Business Logic Layer 14](#_Toc217232414)

[5.2.3 Core Business Services 14](#_Toc217232415)

[5.3 Services and Integration Layer 15](#_Toc217232416)

[5.3.1 Integration Services 15](#_Toc217232417)

[5.3.2 Data Flow 17](#_Toc217232418)

[5.4 Data Access Layer 18](#_Toc217232419)

[5.4.1 Data Access Control 18](#_Toc217232420)

[5.4.2 Data Storage 18](#_Toc217232421)

[5.4.3 Connection Pooling 18](#_Toc217232422)

[5.4.4 Concurrent Access and Object Locking 19](#_Toc217232423)

[5.4.5 Transactional Requirements 19](#_Toc217232424)

[5.4.6 Persistence 19](#_Toc217232425)

[5.5 Performance 19](#_Toc217232426)

[5.7 Security 20](#_Toc217232427)

[5.7.1 Authentication 20](#_Toc217232428)

[5.7.2 Authentication 20](#_Toc217232429)

[5.7.3 Authorization 20](#_Toc217232430)

[5.7.4 Access Request Process 20](#_Toc217232431)

[5.7.5 Encryption 21](#_Toc217232432)

[5.8 Scalability 21](#_Toc217232433)

[5.9 Extensibility 21](#_Toc217232434)

[6 Data Conversion and Migration 21](#_Toc217232435)

[6.1 Migration Strategy 21](#_Toc217232436)

[6.2 Data Quality Assurance 23](#_Toc217232437)

[7 Reporting and Information 23](#_Toc217232438)

[7.1 Reporting Strategy 24](#_Toc217232439)

[7.2 Report Delivery Options 26](#_Toc217232440)

[8 Deployment Architecture 27](#_Toc217232441)

[8.1 Deployment Architecture 27](#_Toc217232442)

[8.2 Kubernetes Architecture 28](#_Toc217232443)

[8.3 Network Architecture 29](#_Toc217232444)

[8.4 Environment Specifications 30](#_Toc217232445)

[8.4.1 Kubernetes Configuration 30](#_Toc217232446)

[8.4.2 Network Zones 30](#_Toc217232447)

[9 Infrastructure Architecture 30](#_Toc217232448)

[9.1 Backup, Failover, and Recovery 30](#_Toc217232449)

[9.1.1 Backup 30](#_Toc217232450)

[9.1.2 Fail-over 30](#_Toc217232451)

[9.1.3 Recovery 31](#_Toc217232452)

[9.2 Maintenance Windows 31](#_Toc217232453)

[10 Technology Stack Details 31](#_Toc217232454)

[10.1 Microservices Technology Stack 31](#_Toc217232455)

[10.2 Integration Protocols 31](#_Toc217232456)

[10.3 Monitoring and Observability 31](#_Toc217232457)

[10.4 CI/CD Pipeline 33](#_Toc217232458)

[11 Appendices 34](#_Toc217232459)

[11.1 Appendix A: Diagram References 34](#_Toc217232460)

[11.1.1 System context and stakeholders 34](#_Toc217232461)

[11.1.2 Overall system architecture 35](#_Toc217232462)

[11.1.3 Declaration workflow states 35](#_Toc217232463)

[11.1.4 External system integrations 36](#_Toc217232464)

[11.1.5 Document storage architecture 37](#_Toc217232465)

[11.1.6 Master data synchronization 38](#_Toc217232466)

[11.2 Appendix B: Reference Documents 38](#_Toc217232467)

[11.3 Appendix C: Service Level Agreements 38](#_Toc217232468)

Introduction

Purpose and Audience

This High-Level Design (HLD) document provides a comprehensive architectural blueprint for the JUL (Janela Única Logística) - Angola's National Logistics Single Window System. The document outlines the technical architecture, design decisions, and implementation approach for a modern, scalable trade facilitation platform.

The JUL system serves as a central hub for customs clearance and regulatory compliance, integrating multiple government agencies and private sector stakeholders to streamline trade processes, reduce clearance times, and enhance transparency in logistics operations.

**Target Audience:** This document is intended for technical architects, development teams, infrastructure engineers, project managers, business analysts, and stakeholders involved in the design, implementation, and operation of the JUL system.

**Key Objectives:**

* Provide single-point submission for trade-related documents
* Enable real-time tracking of shipments and cargo
* Facilitate seamless integration with customs and regulatory agencies
* Ensure compliance with World Customs Organization (WCO) standards
* Support concurrent users up to 10,000 with 99.9% system availability
* Process transactions with response time < 3 seconds (95th percentile)

Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| Abbreviation/ Term | Description |
| AGT | Administração Geral Tributária (Angola Customs Authority) |
| ASYCUDA | Automated System for Customs Data (UN customs system) |
| CNCA | Certificado Nacional de Capacidade Aduaneira (National Customs Capacity Certificate) |
| CUSDEC | Customs Declaration Message (UN/EDIFACT) |
| CUSRES | Customs Response Message (UN/EDIFACT) |
| DU | Declaração Única (Single Declaration for customs) |
| HS | Harmonized System (commodity classification) |
| JUL | Janela Única Logística (Angola National Logistics Single Window) |
| OGA | Other Government Agencies |
| SINTECE | Sistema Nacional de Troca Electrónica de Dados (Angola Single Window) |
| SICOEX | Sistema de Comércio Exterior) - Angola's Foreign Trade Computing System |
| WCO | World Customs Organization |
| WTO TFA | World Trade Organization Trade Facilitation Agreement |

References

* JUL-AGT Integration Control Document - Draft - V2 (ASYCUDA integration specifications)
* JUL-SINTECE Integration Control Document - Draft - v2 (SINTECE integration specifications)
* JUL System User Management Architecture (Keycloak implementation details)
* WCO Data Model Version 3.10 (World Customs Organization standards)
* UN/EDIFACT Message Standards (Electronic data interchange specifications)
* UNCTAD Recommendations and Standards for Single Windows
* WTO Trade Facilitation Agreement (TFA) guidelines

Business Vision

Business Domain

* Unified Trade Ecosystem: Deliver a single platform for import, export, and transit procedures.
* Regulatory Coordination: Strengthen collaboration among agencies to ensure compliance and risk-based decision-making.
* Business Imperative: Reduce transaction costs, accelerate clearance times, and enhance Angola’s competitiveness in global trade.

Data Domain

* Data Harmonization: Standardize trade information using the WCO Data Model.
* Interoperability: Enable seamless data exchange across agencies and international partners.
* Business Imperative: Improve transparency, predictability, and trust in trade flows through consistent, reliable information.

Application Domain

* Single Window Portal: Provide a centralized communication and information system aligned with UNCTAD’s framework.
* Workflow Automation: Digitize end-to-end trade processes, minimizing manual intervention.
* Business Imperative: Enhance efficiency, reduce delays, and support Angola’s obligations under the WTO Trade Facilitation Agreement.

Technology Domain

* Scalable Infrastructure: Deploy modular, cloud-native architecture to support growth and resilience.
* Security & Trust: Implement strong cybersecurity and identity management to protect sensitive trade data.
* Business Imperative: Ensure system reliability, resilience, and stakeholder confidence in digital trade operations.

Supply Chain Domain

* Event Logging & Traceability: Monitor land and maritime supply chains with integrated tracking.
* Visibility & Accountability: Provide end-to-end transparency across logistics operations.
* Business Imperative: Strengthen resilience, reduce risks, and build trust among traders, regulators, and international partners.

Enterprise Architecture

The JUL system implementation follows a phased approach to managing complexity and ensuring business value delivery at each stage:

Project Phasing

* Core platform setup, including IAM, company management, and basic declaration processing with ASYCUDA integration
* SICOEX and SINTECE integration, license management, agent nomination, and document management capabilities.

| Tool Category | License Type | Tool Category | License Type |
| --- | --- | --- | --- |
| UML Modeling Enterprise Architect / PlantUML | Open Source / Commercial | Configuration Git / GitLab | Open Source |
| IDE Visual Studio 2022 / VS Code | Commercial / Open Source | Syntax Check SonarQube | Commercial |
| HTML/CSS/JS ESLint / Prettier / StyleLint | Open Source | Data/SQL Check pgAdmin / DBeaver | Open Source |
| Integration Architect Postman / Swagger | Commercial / Open Source | Web Check Lighthouse / Webpage Test | Open Source |
| Unit Test xUnit / Jest / Jasmine | Open Source | Deployment Kubernetes / Helm / Argo CD | Open Source |
| Performance Test JMeter / k6 | Open Source | Load Test Apache JMeter / Gatling | Open Source |
| Security Test Checkmarx / OWASP ZAP | Commercial / Open Source | API Test Postman / Rest Assured | Commercial / Open Source |
| Container Docker / Harbor | Open Source | Monitoring Prometheus / Grafana | Open Source |
| Log Management ELK Stack (Elasticsearch) | Open Source / Commercial | Code Quality SonarQube / SonarLint | Commercial / Open Source |
| Version Control GitLab / GitHub | Commercial / Open Source | CI/CD GitLab CI / Jenkins | Open Source |
| Database PostgreSQL / pgAdmin | Open Source | Trace/Debug Jaeger / OpenTelemetry | Open Source |
| Package Manager npm / NuGet | Open Source | Documentation Swagger UI / ReDoc | Open Source |

Technology and Architecture Assumptions and Dependencies

| Assumption & Dependencies | Architecture Decision |
| --- | --- |
| Infrastructure Availability: On-premises Kubernetes cluster with adequate compute, storage, and network resources is provisioned and operational. | Deploy on Kubernetes with 3 environments (Dev: 3 nodes, Staging: 5 nodes, Production: 10+ nodes). |
| External System Connectivity: Network connectivity to ASYCUDA (AGT), SINTECE, and SICOEX systems is established with appropriate firewall rules and VPN access. | Implement DMZ zone for external integration with the API Gateway. Use certificate-based mutual authentication for ASYCUDA SOAP/XML. REST/JSON for SINTECE and OGAs with OAuth 2.0 token-based authentication. |
| Database Platform: PostgreSQL database platform is available with high availability configuration and backup infrastructure. | Adopt a database-per-microservice pattern using PostgreSQL 15+. Production: Primary + 2 read replicas per service. Implement streaming replication with automatic failover using Patroni. |
| Identity Management: An Enterprise identity provider is available for user authentication and authorization integration. | Deploy Keycloak 23+ as a centralized IAM solution. Implement OAuth 2.0/OpenID Connect for API authentication, SAML 2.0 for enterprise SSO. Mandatory MFA for customs officers using TOTP. |
| Master Data Sources: Nightly batch feeds available from AGT for HS codes, tariff schedules, port codes, and regulatory reference data. | Implement Master Data Management Service with scheduled batch synchronization at 02:00 AM WAT. Use ETL pipeline with data validation, transformation, and distribution to all microservices. Implement rollback capability for failed synchronizations. |
| Object Storage: A Scalable object storage solution is available for document management with an S3-compatible API. | Deploy MinIO in distributed mode (6+ nodes in production) for S3-compatible object storage. Implement erasure coding for data protection. Integrate with Keycloak for authentication and RBAC. |
| Message Queue Platform: A Message broker infrastructure is available for asynchronous communication between microservices. | Deploy RabbitMQ 3.12+ in cluster mode (3 nodes) with mirrored queues. Implement dead letter queues for failed messages. Use AMQP protocol with persistent messages for critical events. |
| Monitoring Infrastructure: Monitoring and logging infrastructure is available for production observability. | Implement Prometheus + Grafana for metrics and visualization. Deploy ELK Stack for centralized logging. Integrate Jaeger for distributed tracing. Configure alerting via email/SMS for critical events. |
| Azure CI/CD Platform: Source control and CI/CD pipeline infrastructure is available for automated build and deployment. | Use Azure for source control with Azure CI for automated pipelines. Implement ArgoCD for GitOps-based Kubernetes deployments. Integrate SonarQube for code quality gates and Checkmarx for security scanning. |
| Backup Infrastructure: Backup storage and disaster recovery infrastructure are available with off-site replication. | Implement automated backup strategy: Transaction logs every 15 minutes, database backups every 6 hours, full system backup daily. Maintain 30-day retention with a 7-year archive. Cross-site replication to the DR site. Above can be updated based on client requirements |
| WCO Data Model Documentation: Access to WCO Data Model Version 3.10 specifications and UN/EDIFACT message standards. | Implement a data model fully compliant with WCO Data Model 3.10. Use JSON schemas for API validation. Map database entities to WCO classes (Declaration, Goods Item, Party, etc.). Document deviations from standard. |
| Data Retention Policy: Legal requirements for data retention (7 years minimum) are documented and approved. | Implement an automated archival process for data older than 1 year to cold storage. Maintain audit trail for 7 years minimum. Implement data lifecycle management with automated deletion after retention period (with legal hold capability). |
| Business Continuity Plan: Disaster recovery site and procedures are defined with RTO < 4 hours and RPO < 1 hour. | Implement hot standby at the DR site with real-time replication. Maintain a complete environment replica at the DR location. Conduct quarterly DR testing. Document failover and failback procedures. Monitor replication lag continuously. |
| User Acceptance Testing Environment: A Staging environment with production-like data is available for UAT activities. | Provision a staging environment with 5 Kubernetes nodes (16 vCPU, 64 GB RAM each). Implement data masking for production data copied to staging. Provide dedicated testing slots for business users. Reset the environment weekly. |

Architecture Objectives

The architecture is designed to achieve the following core objectives:

* Scalability: Support up to 10,000 concurrent users and 100 transactions per second with horizontal scaling capabilities.
* Availability: Achieve 99.9% uptime through redundancy, failover mechanisms, and comprehensive monitoring.
* Performance: Deliver response times under 3 seconds for 95% of API requests through optimized architecture and caching.
* Security: Implement comprehensive security measures, including encryption, authentication, authorization, and audit logging.
* Maintainability: Enable independent deployment and updates of microservices with minimal system disruption.
* Interoperability: Ensure seamless integration with external systems using industry-standard protocols and data formats.
* Compliance: Adhere to WCO Data Model, UN/EDIFACT standards, and Angola regulatory requirements.
* Resilience: Provide disaster recovery capabilities with RTO < 4 hours and RPO < 1 hour.

Architecture Constraints

Standards

Technology Standards

* Frontend: Angular 20+ with TypeScript and Module Federation
* Backend: ASP.NET Core 10.0 with C# for all microservices
* •Database: PostgreSQL 15+ for relational data storage
* •Object Storage: MinIO for S3-compatible document storage
* Authentication: Keycloak 23+ with OAuth 2.0/OpenID Connect/SAML 2.0
* Containerization: Docker with OCI container specifications
* Orchestration: Kubernetes for deployment and scaling

Messaging Standards

* Internal Messaging: RabbitMQ 3.12+ with AMQP protocol
* REST APIs: OpenAPI 3.0 specifications with JSON payloads
* REST APIs Integration: OpenAPI 3.0 specifications with JSON payloads
* Event Messaging: JSON schema validation for all domain events
* API Gateway: Kong for unified API management and security

XML Standards

* UN/EDIFACT: CUSDEC and CUSRES message formats for customs declarations
* WCO Data Model: XML schemas based on WCO Data Model Version 3.10
* SOAP/XML: Standard SOAP envelope structure for ASYCUDA integration
* Schema Validation: XSD validation for all XML message exchanges

Portability

The architecture is designed with portability in mind using containerization and standard APIs:

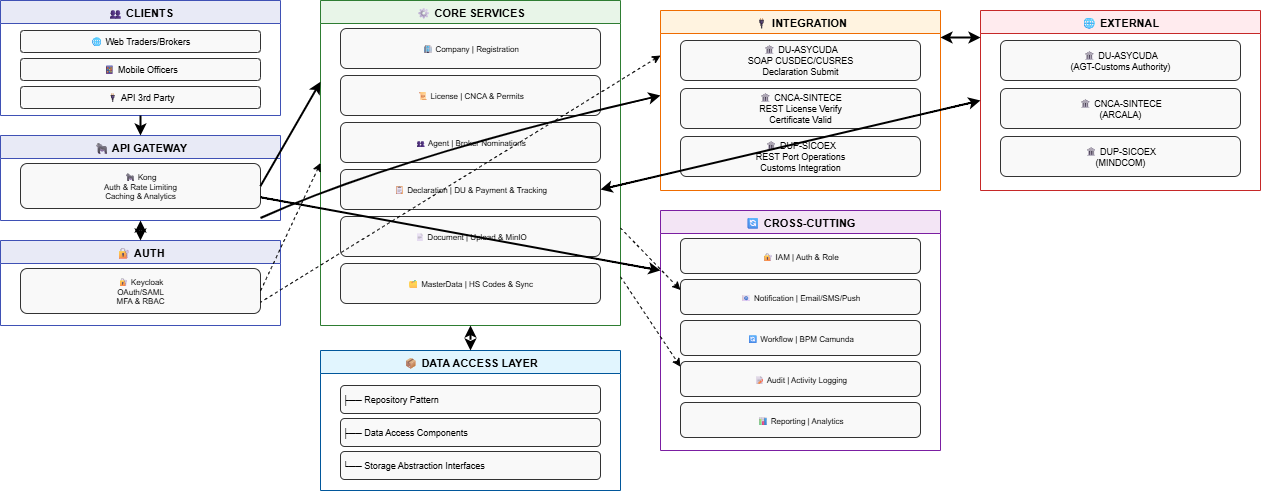
* All services are containerized using Docker with OCI standards
* Kubernetes deployment enables cloud-agnostic infrastructure
* Database abstraction layer allows migration between PostgreSQL-compatible systems
* MinIO S3-compatible API enables migration to cloud object storage
* Standard protocols (REST, SOAP) ensure external system portability

Architecture Model

Component Descriptions

The JUL system employs a microservices architecture consisting of 14+ specialized services grouped into three categories:

* Company Management, License & Permits, Agent Nomination, Declaration Management, Document Management, Master Data Management Core Business Services (6 services):
* ASYCUDA Integration, SINTECE Integration, SICOEX Integration Services (3 services):
* IAM (Keycloak), Notification, Workflow (Camunda), Audit & Logging, Reporting Cross-Cutting Services (5 services):



Presentation Layer

The presentation layer provides user interfaces for different stakeholder groups using Angular 20+ with Module Federation for micro-frontend architecture:

* Web Portal: Responsive design supporting desktop and mobile web browsers
* User Roles: Traders, Customs Brokers, Freight Forwarders, Customs Officers, System Administrators
* Localization: Full Portuguese and English language support with dynamic switching
* Accessibility: WCAG 2.1 AA compliant with screen reader support
* Progressive Web App: Offline capabilities for essential functions
* Material Design: Angular Material UI components for a consistent user experience

Key Interface Modules

* Personalized dashboards with role-based information, status updates, and quick actions Dashboard Module
* Registration wizard, profile management, user invitation, document library, Company Management Module
* Step-by-step declaration creation, document attachment, status tracking, payment integration Declaration Module
* Application forms, OGA coordination, renewal processing, compliance tracking, License & Permits Module
* Agent search, power of attorney management, authorization tracking, Agent Nomination Module
* Secure upload, document viewer, collaboration features, advanced search, Document Management Module
* Pre-built templates, custom report builder, data visualization, scheduled delivery, Reporting Module

Business Logic Layer

The business logic layer implements domain-driven design principles with clear bounded contexts:

* Independent services with database-per-service pattern Microservices Architecture
* Event-driven communication for loose coupling between services: Domain Events
* Configurable validation and processing rules, Business Rules Engine
* Camunda 8 for complex business process management, Workflow Orchestration
* Command Query Responsibility Segregation for complex read scenarios, CQRS Pattern

Core Business Services

Company Management Service (Port 8081)

* Company registration and onboarding with verification workflows
* Profile management and updates with document validation
* User-company relationship management
* Integration with Keycloak for user provisioning
* Database: company\_db (PostgreSQL)

License & Permits Service (Port 8082)

* CNCA certificate management and processing
* Multi-agency permit coordination
* License renewal and compliance tracking
* Integration with SINTECE for verification
* Database: license\_db (PostgreSQL)

Agent Nomination Service (Port 8083)

* Customs broker/agent nomination processing
* Power of attorney management
* Authorization validity monitoring
* Agent performance tracking
* Database: agent\_db (PostgreSQL)

**Declaration Management Service (Port 8084)**

* DU (Declaração Única) creation and validation
* State machine for declaration lifecycle
* Integrated payment processing
* ASYCUDA integration for customs processing
* Workflow management via Camunda
* Database: declaration\_db (PostgreSQL)

**Document Management Service (Port 8085)**

* Secure document upload and storage
* MinIO integration for object storage
* Document validation and format verification
* Metadata management and search
* Database: document\_metadata\_db + MinIO storage

**Master Data Management Service (Port 8086)**

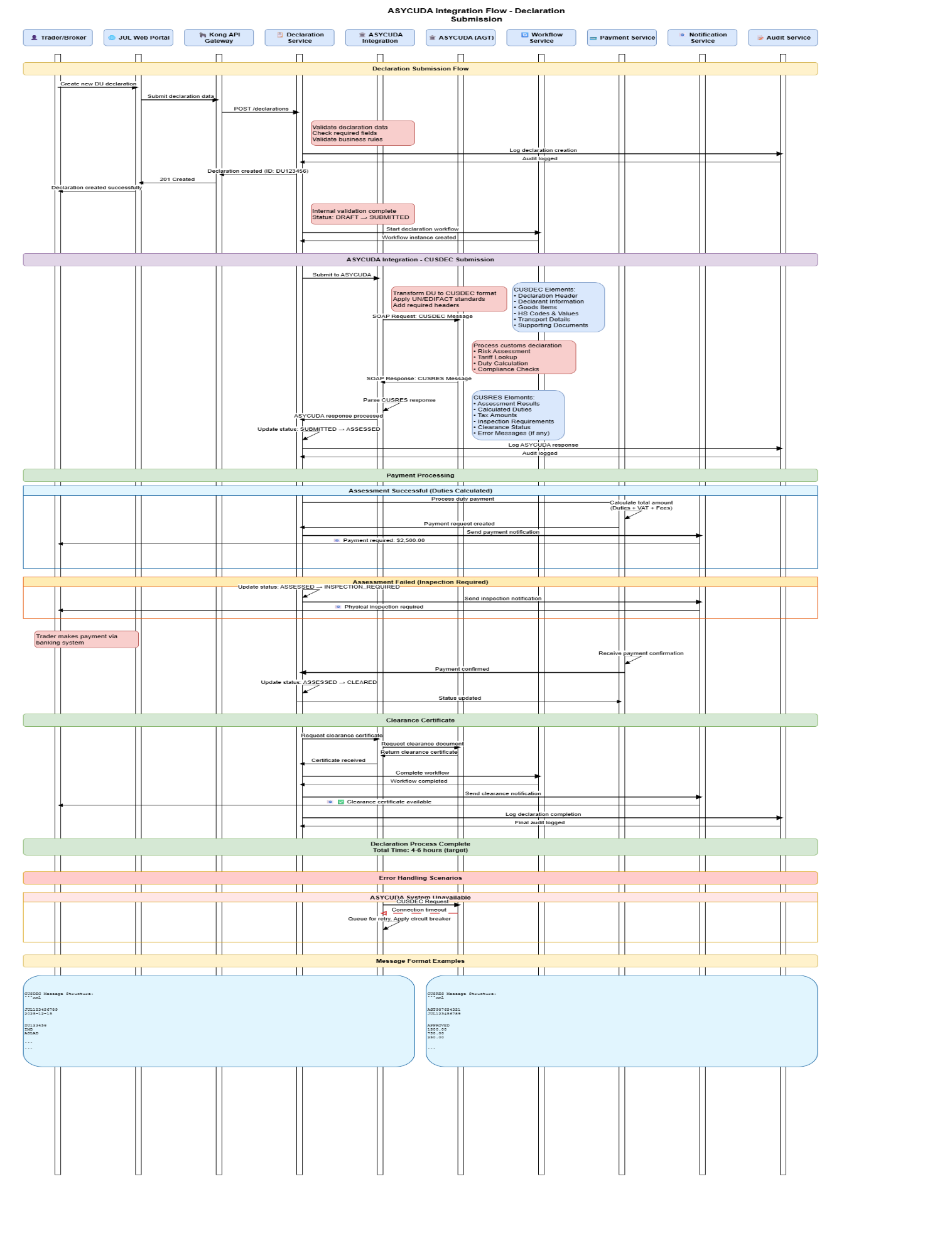
* Nightly batch synchronization from AGT systems
* HS Code management (WCO HS2017/2022)
* Port codes, country codes, currency codes
* Distribution to all microservices
* Database: masterdata\_db (PostgreSQL)

Services and Integration Layer

Integration Services

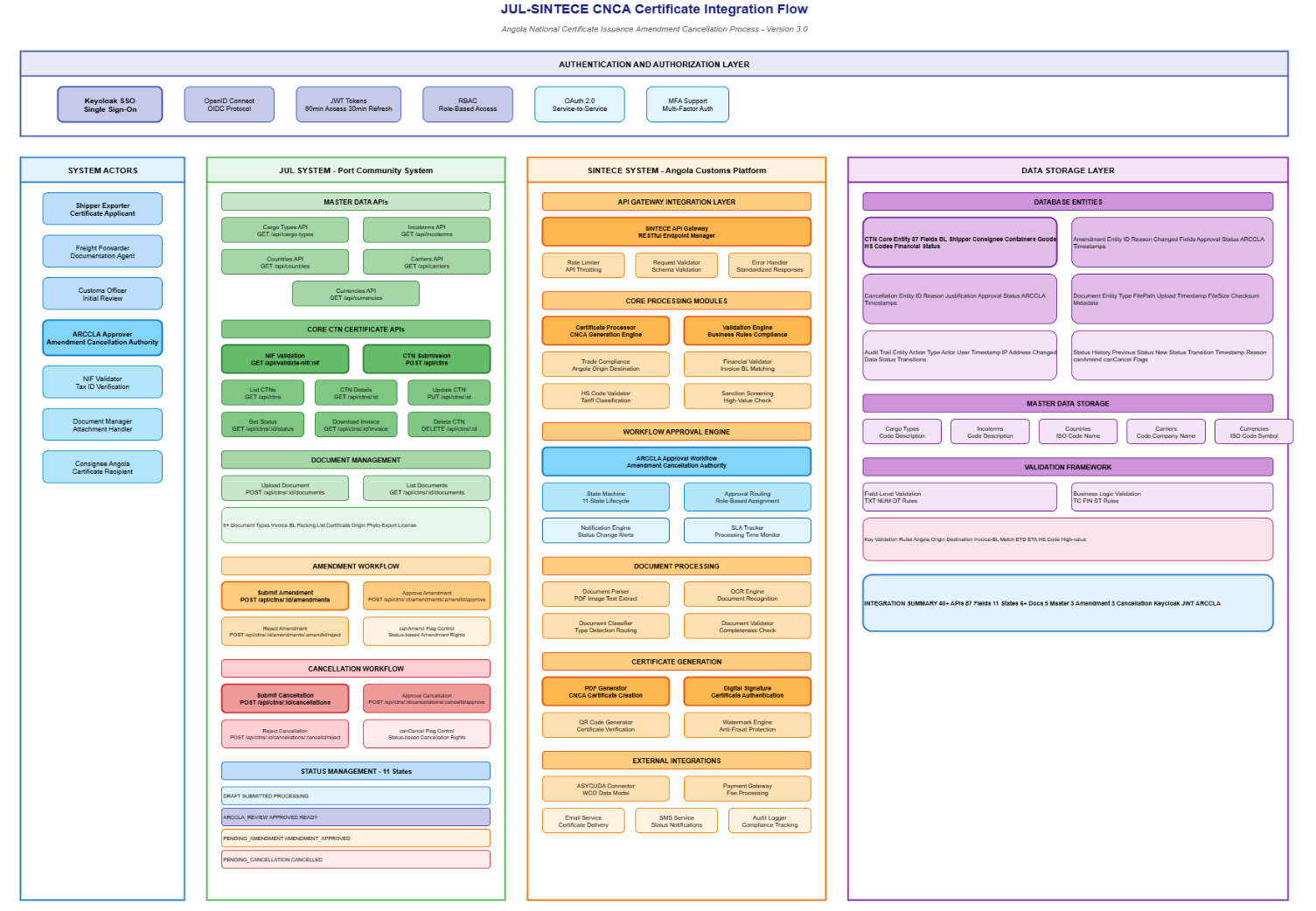
**ASYCUDA Integration Service (Port 9081)**

* SOAP/XML message handling for customs declarations
* CUSDEC submission and CUSRES response processing
* Duty calculation and assessment results
* Clearance certificate retrieval



**SINTECE Integration Service (Port 9082)**

* REST/JSON API integration
* License and permit verification
* Certificate validation and authenticity checking
* Single window data synchronization



**OGA Integration Service (Port 9083)**

* Multi-agency REST/JSON integration
* LPCO requirements coordination
* Regulatory certificate management
* Asynchronous processing via message queue

Data Flow

Data flows through the system following these patterns

* User submits declaration → Declaration Service validates → ASYCUDA Integration sends CUSDEC → Response processed → User notified Synchronous Flow
* License application → Queue message → OGA Integration → External processing → Status update event → User notification Asynchronous Flow
* Declaration state change → Event published → Multiple services react → Audit logged → Reports updated Event-Driven Flow



Data Access Layer

Data Access Control

Data access is controlled through multiple security layers:

* Row-Level Security: PostgreSQL RLS policies enforce tenant isolation
* Column-Level Encryption: Sensitive fields encrypted at the application layer
* API-Level Authorization: OAuth 2.0 scopes and role-based permissions
* Audit Logging: Complete tracking of data access and modifications

Data Storage

|  |  |  |
| --- | --- | --- |
| Storage Type | Technology | Use Case |
| Relational Database | PostgreSQL 15+ | Transactional data, master data, user data |
| Object Storage | MinIO | Documents, attachments, large files |
| Cache | Redis 7+ | Session data, frequently accessed data |
| Message Queue | RabbitMQ 3.12+ | Asynchronous messaging, event processing |

Connection Pooling

Connection pooling optimizes database performance:

* Connection Pool Size: Min 10, Max 100 connections per service
* Connection Lifetime: Maximum 30 minutes with automatic renewal
* Connection Validation: Health checks before query execution
* Connection Timeout: 30 seconds for acquiring a connection from the pool

Concurrent Access and Object Locking

Concurrency is managed through multiple mechanisms:

* Optimistic Locking: Version numbers on critical entities
* Pessimistic Locking: Database row locks for financial transactions
* Distributed Locks: Redis-based locks for cross-service operations
* Idempotency Keys: Prevent duplicate transaction processing

Transactional Requirements

ACID properties are maintained through:

* Local Transactions: PostgreSQL ACID guarantees within service boundaries
* Distributed Transactions: Saga pattern for cross-service operations
* Compensation Logic: Automated rollback for failed distributed transactions
* Transaction Isolation: Read Committed isolation level for most operations

Persistence

Data persistence strategy ensures durability:

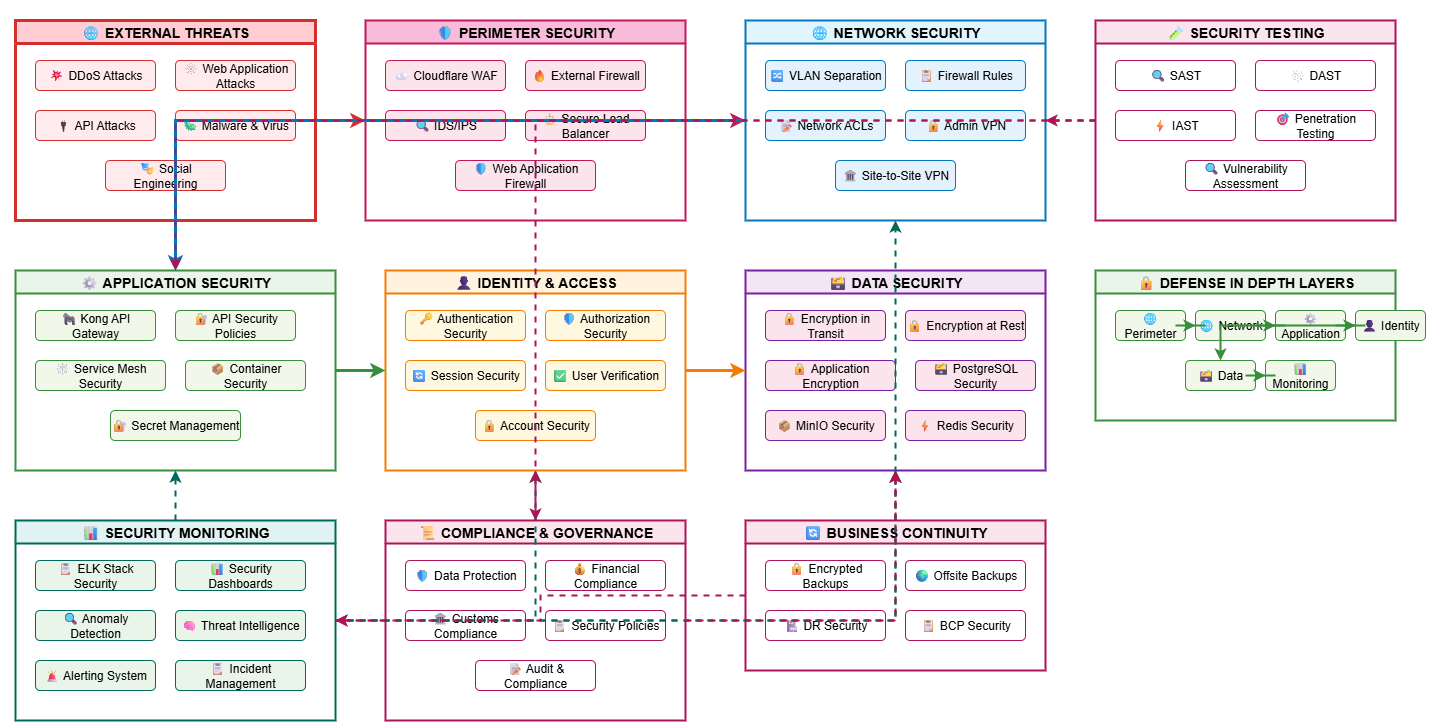
* Write-Ahead Logging: PostgreSQL WAL for crash recovery
* Replication: Streaming replication to read replicas
* Point-in-Time Recovery: Continuous archiving of WAL files
* Backup Strategy: Full daily backups plus continuous WAL archiving

Performance

Performance targets and optimization strategies:

|  |  |  |
| --- | --- | --- |
| Metric | Target | Strategy |
| API Response Time | < 3 seconds (95%ile) | Caching, database indexing, CDN |
| Throughput | 100 TPS | Horizontal scaling, load balancing |
| Concurrent Users | 10,000 | Stateless services, session clustering |
| Database Query | < 500ms (95%ile) | Query optimization, materialized views |
| Page Load Time | < 2 seconds | Code splitting, lazy loading, and compression |

## Security



Authentication

Authentication

* Identity Provider: Keycloak 23+ with OAuth 2.0/OpenID Connect
* Multi-Factor Authentication: TOTP/SMS required for customs officers
* Password Policy: Minimum 12 characters, complexity requirements, 90-day rotation
* Session Management: JWT tokens with 30-minute expiry, refresh tokens for 7 days
* Single Sign-On: SAML 2.0 for enterprise integration

Authorization

* Role-Based Access Control: Fine-grained permissions per role
* Resource-Level Permissions: User can only access their own company data
* API Scopes: OAuth 2.0 scopes limit API access
* Dynamic Authorization: Permissions evaluated at runtime based on context

Access Request Process

User access is granted through a formal workflow:

* Step 1: User submits access request through the portal
* Step 2: The Company administrator reviews and approves
* Step 3: System administrator provisions the user in Keycloak
* Step 4: User receives email with activation link
* Step 5: User sets password and enables MFA if required
* Step 6: Access logged-in audit trail

Encryption

* Data in Transit: TLS 1.3 with modern cipher suites
* Data at Rest: AES-256 encryption for databases and object storage
* Field-Level Encryption: Additional encryption for PII and financial data
* Key Management: Hardware Security Module or software KMS

Scalability

System scales horizontally across multiple dimensions:

* Stateless Services: All application services are stateless for easy horizontal scaling
* Database Read Replicas: PostgreSQL replicas for read-heavy workloads
* Kubernetes HPA: Automatic pod scaling based on CPU/memory metrics
* Message Queue Clustering: RabbitMQ cluster scales with message volume
* Object Storage: MinIO distributed deployment scales storage capacity
* CDN Integration: Static content delivery scales geographically

Extensibility

Architecture supports future extensions:

* Plugin Architecture: New services can be added without modifying existing services
* API Versioning: Multiple API versions supported simultaneously
* Event-Driven: New services can subscribe to existing events
* Feature Flags: New features can be gradually rolled out
* Microservices: New capabilities added as independent services

Data Conversion and Migration

Migration Strategy

Initial data migration for Phase 1 deployment follows a phased approach with comprehensive validation and rollback capabilities. Master data is synchronized from authoritative sources (AGT, SINTECE) with automated quality checks. Transactional data remains in source systems initially with on-demand access.

| Module | Business Object | Owner | Source Table | Constraints | Target Table | Interface | Rule Sequence |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Master Data | HS Codes | AGT Customs | AGT\_HS\_CODES | Unique HS code (6-10 digits), Valid tariff rate, Active status | hs\_codes | Batch ETL | 1. Validate format 2. Check WCO compliance 3. Map tariff rates 4. Insert/Update |
| Master Data | Country Codes | AGT Customs | AGT\_COUNTRIES | ISO 3166-1 alpha-2, Valid country name | countries | Batch ETL | 1. Validate ISO code 2. Map names 3. Set active flag |
| Master Data | Currency Codes | AGT Customs | AGT\_CURRENCIES | ISO 4217 code, Valid exchange rate | currencies | Batch ETL | 1. Validate ISO code 2. Validate rate 3. Set effective date |
| Master Data | Port Codes | AGT Customs | AGT\_PORTS | UN/LOCODE format, Valid port name | ports | Batch ETL | 1. Validate LOCODE 2. Verify coordinates 3. Map customs office |
| Master Data | Customs Offices | AGT Customs | AGT\_OFFICES | Unique office code, Valid address | customs offices | Batch ETL | 1. Validate code 2. Map region 3. Set operating hours |
| Company Mgmt | Company Registry | SINTECE | SINTECE\_COMPANIES | Valid NIF, Unique registration number, Active status | companies | API Sync | 1. Validate NIF format 2. Check duplicates 3. Verify documents 4. Set status |
| Company Mgmt | Company Documents | SINTECE | SINTECE\_DOCS | Valid document type, Not expired | company documents | API Sync | 1. Validate type 2. Check expiry 3. Upload to MinIO 4. Link to company |
| IAM | User Accounts | Manual Import | USER\_IMPORT.CSV | Valid email, Unique username, Valid role | keycloak\_users | Bulk Import | 1. Validate email 2. Check duplicates 3. Generate temp password 4. Send activation |
| IAM | User Roles | Manual Config | ROLES\_CONFIG.CSV | Valid role name, Valid permissions | keycloak\_roles | Manual Config | 1. Create role 2. Assign permissions 3. Map to groups |
| License Mgmt | License Types | SINTECE | SINTECE\_LICENSE\_TYPES | Valid type code, Active status | license types | API Sync | 1. Validate code 2. Map requirements 3. Set validity period |
| License Mgmt | Existing Licenses | SINTECE | SINTECE\_LICENSES | Valid license number, Not expired, Valid holder | licenses | API Sync | 1. Validate number 2. Check expiry 3. Verify holder 4. Import documents |
| Master Data | Document Types | AGT Customs | AGT\_DOC\_TYPES | Unique type code, Valid description | document types | Batch ETL | 1. Validate code 2. Map category 3. Set requirements |
| Master Data | Unit Measures | AGT Customs | AGT\_UNITS | UN/CEFACT code, Valid conversion factor | unit measures | Batch ETL | 1. Validate UN code 2. Set conversion 3. Map aliases |
| Master Data | Tariff Schedules | AGT Customs | AGT\_TARIFFS | Valid HS code FK, Valid duty rate, Effective dates | tariff schedules | Batch ETL | 1. Validate HS code 2. Check dates 3. Validate rates 4. Set preferences |
| Declaration | Historical Declarations | ASYCUDA | ASYCUDA\_DECLARATIONS | Complete declaration, Valid status | Not Migrated | On-Demand API | Access via ASYCUDA API as needed - no bulk migration |

Data Quality Assurance

* Validation Rules: Business rules applied during import to ensure data integrity, format compliance, and referential integrity
* Duplicate Detection: Automated identification of duplicate records based on natural keys (NIF, HS codes, license numbers)
* Data Cleansing: Standardization of formats (dates, phone numbers, addresses), removal of special characters, trimming whitespace
* Reconciliation Reports: Detailed comparison of source vs target data with record counts, field-level differences, and error logs
* Rollback Capability: Point-in-time database snapshots before migration with the ability to revert failed migrations completely
* Error Handling: Failed records logged to the error table with reason codes, reprocessing capability, and notification to data owners
* Data Profiling: Pre-migration analysis of source data quality, completeness, and patterns to identify potential issues
* Staging Environment: All migrations are tested in the staging environment before production with user acceptance testing

Reporting and Information

This section describes the strategy for creating reports for each use case that involves reporting capabilities. Reports support operational monitoring, regulatory compliance, and business intelligence requirements.

Reporting Strategy

The reporting architecture provides three tiers of reporting: (1) Operational reports for day-to-day monitoring, (2) Compliance reports for regulatory requirements, and (3) Business intelligence dashboards for strategic decision-making. All reports support multiple export formats (PDF, Excel, CSV) and can be scheduled for automated delivery

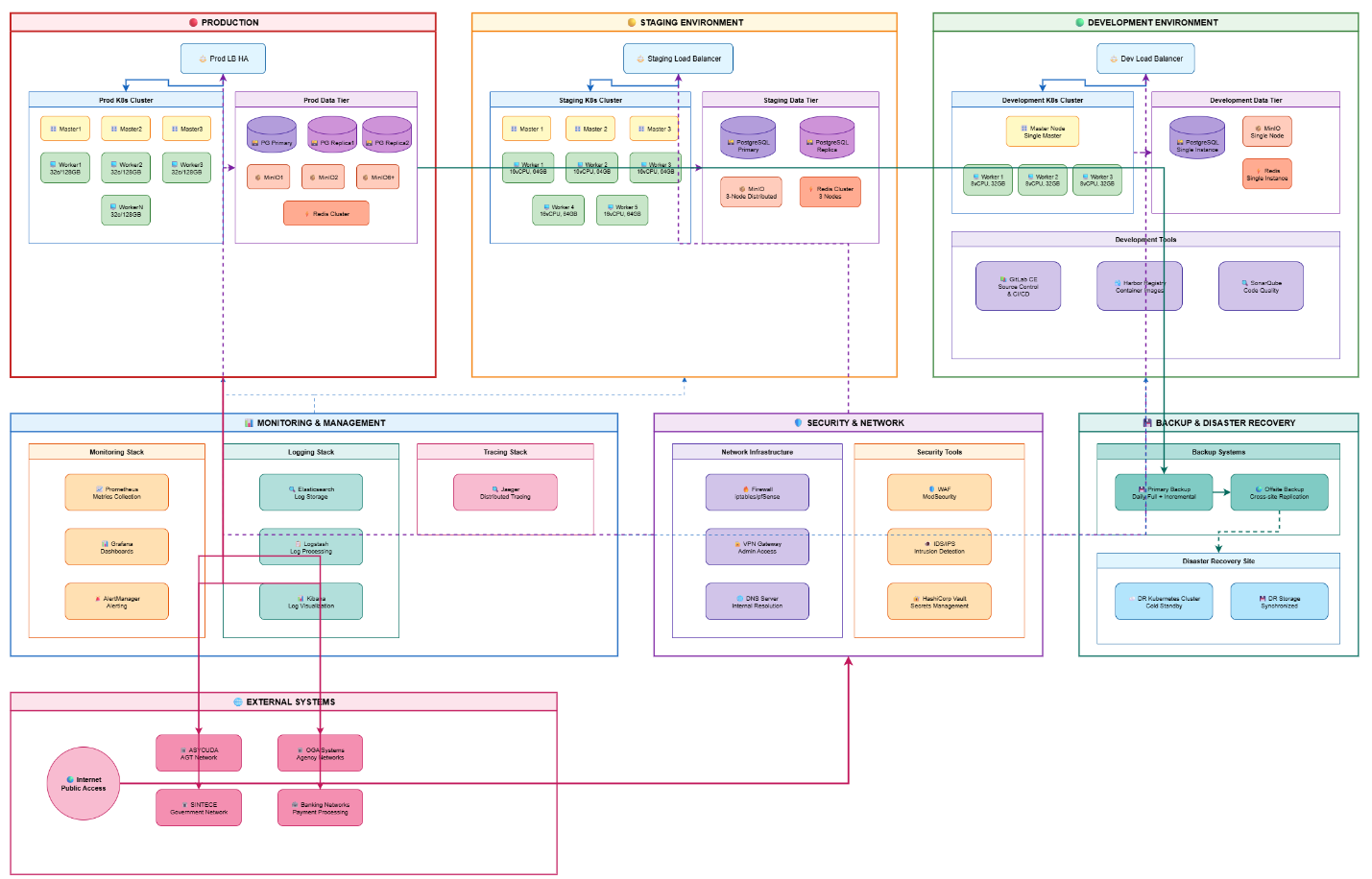
| Use Case | Subject Domain | Business Object | Type Report | Description | Comment |
| --- | --- | --- | --- | --- | --- |
| UC-001: Monitor Declaration Processing | Operations | Declaration | Operational | Declaration statistics, including volume, processing times, clearance rates by status, customs office, and time period | Real-time dashboard with drill-down capability. Exportable to Excel/PDF |
| UC-002: Track Payment Transactions | Finance | Payment | Operational | Payment transaction volumes, collection amounts, payment methods, and reconciliation status by date range | Daily reconciliation report with bank statement matching |
| UC-003: Monitor User Activity | Security | User | Operational | Login statistics, feature usage, session duration, concurrent users, failed login attempts by user role, and time | Security monitoring with alerting on suspicious patterns |
| UC-004: System Performance Metrics | Operations | System | Operational | API response times, system availability, error rates, throughput by service and endpoint, with percentile breakdown | Technical monitoring dashboard integrated with Prometheus/Grafana |
| UC-005: Integration Health Status | Operations | Integration | Operational | External system connectivity status, transaction success rates, average response times for ASYCUDA, SINTECE, and OGAs | Real-time status board with alerting on failures |
| UC-006: Audit Trail Report | Compliance | Audit Log | Compliance | Complete user activity and data modification history with user, timestamp, action, old/new values for the audit period | Required for regulatory audits. 7-year retention. Immutable logs |
| UC-007: WTO TFA Compliance | Compliance | Declaration | Compliance | Trade Facilitation Agreement compliance indicators: average clearance time, document requirements, and advance rulings | Quarterly submission to WTO. Benchmarking against international standards |
| UC-008: Security Incident Report | Security | Security Event | Compliance | Access logs, authentication failures, authorization violations, and security incidents by severity and user | Security operations center (SOC) report. Incident response tracking |
| UC-009: Data Retention Compliance | Compliance | All Objects | Compliance | Evidence of compliance with the 7-year data retention requirement showing archived records by object type and date | Annual audit report. Demonstrates legal compliance |
| UC-010: Trade Statistics Dashboard | Business Intelligence | Declaration | Dashboard | Real-time visualization of import/export volumes, top commodities, trading partners, revenue by HS code, and country | Executive dashboard with charts and graphs. Auto-refresh every 5 minutes |
| UC-011: Trend Analysis Report | Business Intelligence | Declaration | Analytical | Historical comparison and forecasting of trade volumes, seasonal patterns, and growth trends by commodity and region | Monthly strategic planning report. Predictive analytics |
| UC-012: Customs Revenue Report | Finance | Payment | Operational | Duty collected, VAT collected, other fees by declaration type, HS chapter, customs office, and time period | Daily, weekly, and monthly aggregations. Budget vs actual comparison |
| UC-013: License Application Status | Operations | License | Operational | License application pipeline: submitted, under review, approved, rejected by license type, and applicant | Operational tracking for license officers. SLA monitoring |
| UC-014: Agent Performance Report | Operations | Agent Nomination | Operational | Customs broker/agent activity: declarations filed, success rate, average processing time, client satisfaction | Agent ranking and performance monitoring. Quality assurance |
| UC-015: Document Compliance Report | Compliance | Document | Compliance | Required documents submission rate, missing documents, expired certificates by declaration, and commodity | Quality control report. Identifies non-compliance patterns |
| UC-016: HS Code Usage Analysis | Business Intelligence | HS Code | Analytical | Most frequently used HS codes, tariff revenue by HS chapter, classification disputes, and reclassification trends | Strategic report for tariff policy decisions |
| UC-017: User Access Report | Security | User Role | Compliance | User access rights, role assignments, permission changes, and access certification status by user and company | Quarterly access review. Compliance with access control policies |
| UC-018: System Availability Report | Operations | System | Operational | System uptime/downtime, planned maintenance windows, incident response times, and SLA compliance by month | Monthly operations report. SLA tracking and reporting |
| UC-019: OGA Coordination Report | Operations | OGA Integration | Operational | OGA permit requests, approval rates, average processing time, bottlenecks by agency, and permit type | Multi-agency coordination report. Process improvement |
| UC-020: Custom Ad-Hoc Reports | All Domains | Any Object | Ad-Hoc | User-defined custom reports using drag-and-drop report builder with filters, grouping, and calculations | Self-service reporting. Export to PDF, Excel, CSV. Save and schedule |

Report Delivery Options

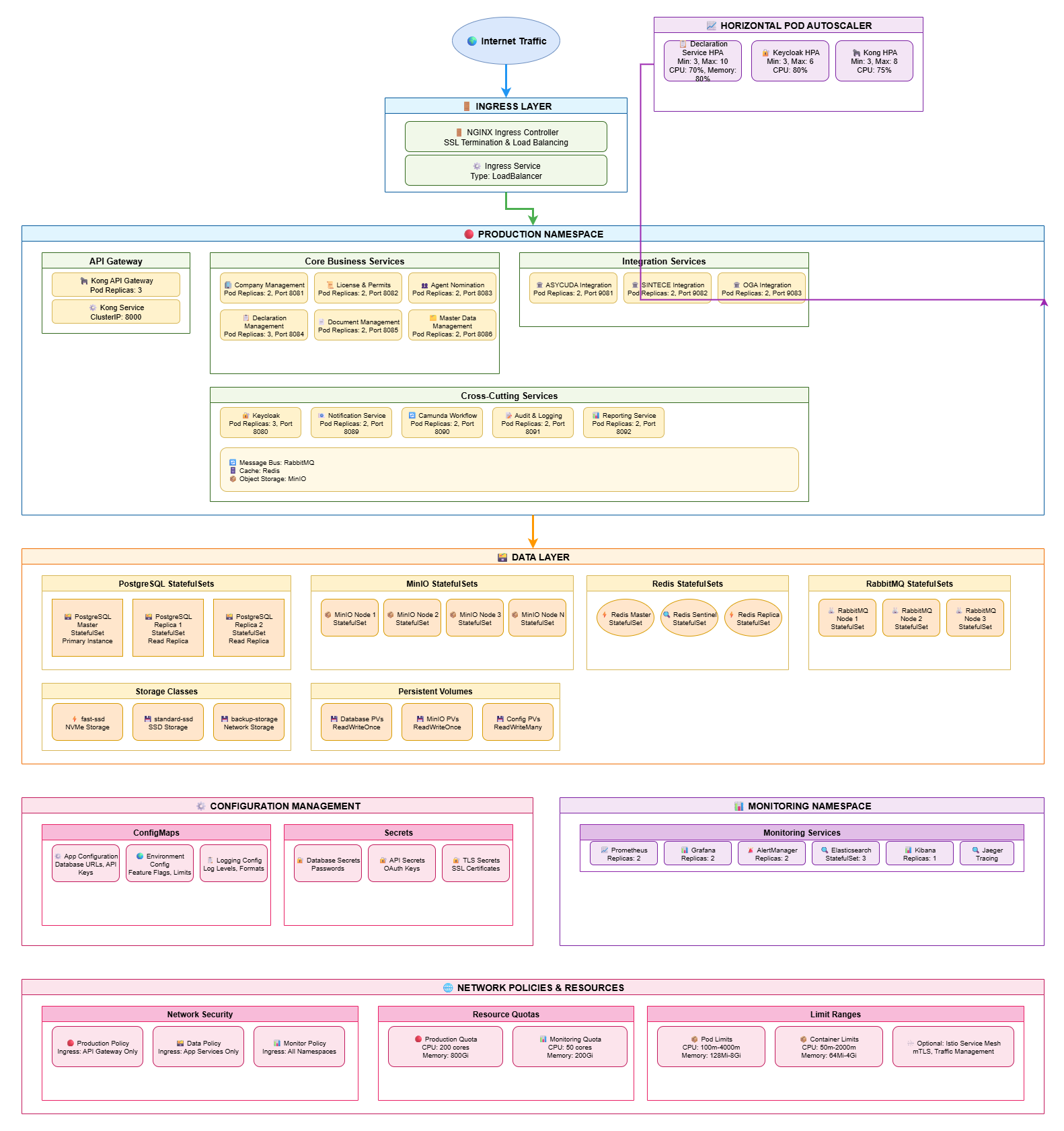
* Export Formats: PDF (formatted for printing), Excel (with formulas and pivot tables), CSV (for data analysis)
* Scheduled Reports: Daily, weekly, monthly, quarterly schedules with automated email delivery to distribution lists
* On-Demand Generation: Real-time report generation with parameter selection and immediate download
* Dashboard Access: Web-based interactive dashboards with real-time data refresh and drill-down navigation
* Mobile Access: Responsive reports optimized for mobile devices with touch-friendly navigation
* API Access: RESTful API endpoints for programmatic report generation and data extraction
* Data Visualization: Charts, graphs, heat maps, and geographical maps for visual data representation
* Report Archival: Automated archival of generated reports for historical reference and compliance

Deployment Architecture

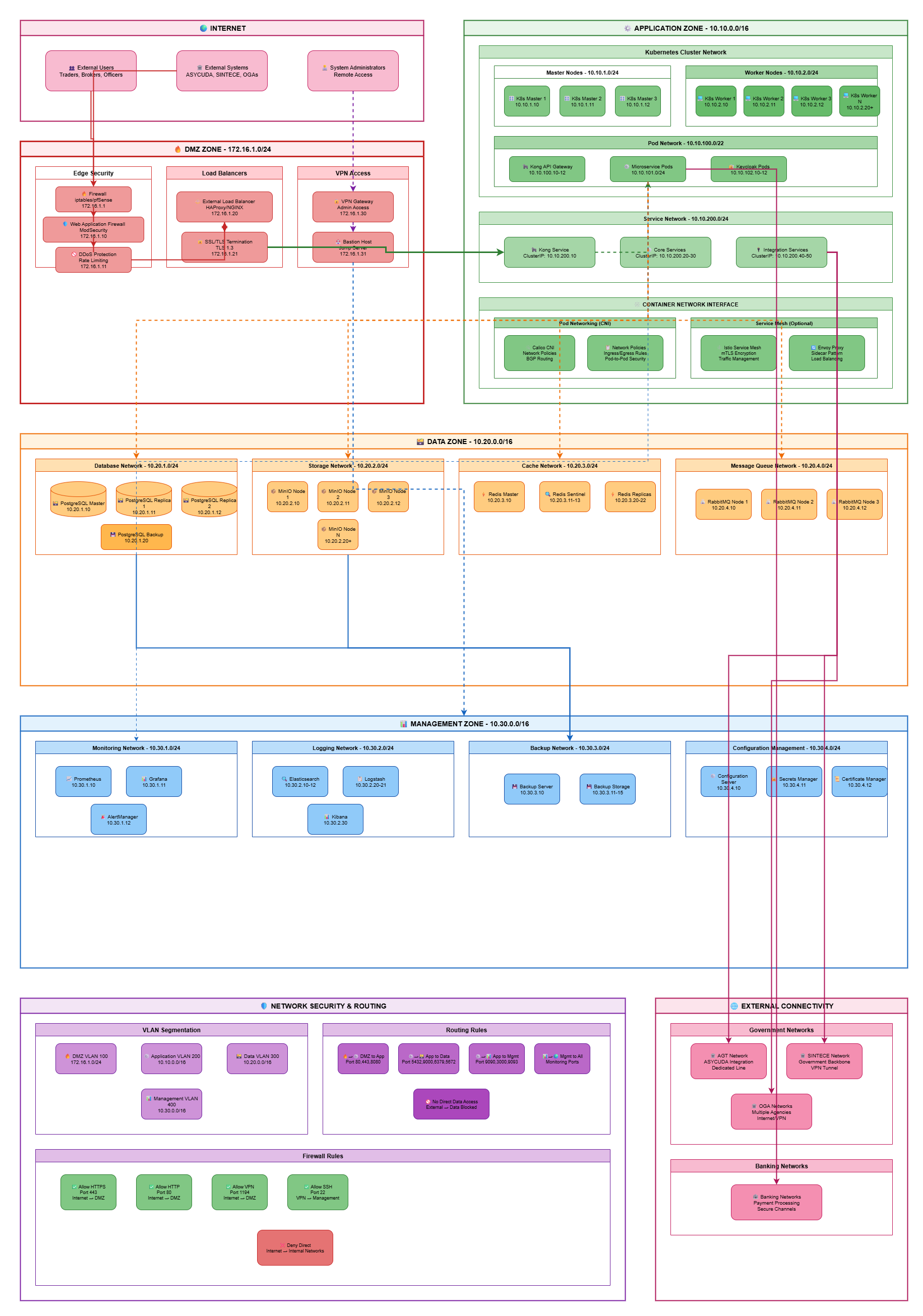
Deployment Architecture



Kubernetes Architecture



Network Architecture



Environment Specifications

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Environment | Nodes | vCPU per Node | RAM per Node | Storage |
| Production | 10+ | 32 | 128 GB | 20+ TB |
| Staging | 5 | 16 | 64 GB | 5 TB |
| Development | 3 | 8 | 32 GB | 2 TB |

Kubernetes Configuration

* Control Plane: 3 master nodes for high availability
* Worker Nodes: Distributed across availability zones
* Ingress Controller: NGINX with SSL termination
* Service Mesh: Optional Istio for advanced traffic management
* Storage Classes: Dynamic provisioning for persistent volumes
* Namespaces: Logical separation by environment and function

Network Zones

* DMZ Zone (172.16.1.0/24): Load balancer, WAF, firewall
* Application Zone (10.10.0.0/16): Kubernetes cluster, all services
* Data Zone (10.20.0.0/16): Databases, MinIO, Redis, RabbitMQ
* Management Zone (10.30.0.0/16): Bastion host, monitoring, backups

Infrastructure Architecture

Backup, Failover, and Recovery

Backup

* Transaction Logs: Every 15 minutes to prevent data loss
* Database Backups: Every 6 hours with compression
* Full System Backups: Daily complete backup of all data
* Backup Retention: 30 days active, 7 years archive
* Offsite Storage: Replicated to the disaster recovery site
* Backup Validation: Monthly restore testing

Fail-over

* Database Failover: Automatic promotion of read replica to primary
* Service Failover: Kubernetes restarts failed pods automatically
* Load Balancer Failover: Health checks remove unhealthy backends
* Redis Failover: Sentinel monitors and promotes a new master
* RabbitMQ Failover: Cluster continues with remaining nodes
* Failover Testing: Quarterly chaos engineering exercises

Recovery

* Recovery Time Objective (RTO): < 4 hours for the complete system
* Recovery Point Objective (RPO): < 1 hour maximum data loss
* Database Recovery: Point-in-time recovery from WAL files
* Service Recovery: Kubernetes self-healing restarts services
* Disaster Recovery Site: Full environment at alternate location
* Recovery Procedures: Documented runbooks for all scenarios

Maintenance Windows

Planned maintenance is scheduled to minimize business impact:

* Regular Maintenance: Sunday 02:00-06:00 WAT (weekly)
* Emergency Maintenance: As needed with advance notification
* Rolling Updates: Zero-downtime deployments for services
* Database Maintenance: During regular window with minimal impact
* User Notification: Email and portal announcements 48 hours in advance
* Maintenance Tracking: All activities logged in the change management system

Technology Stack Details

Microservices Technology Stack

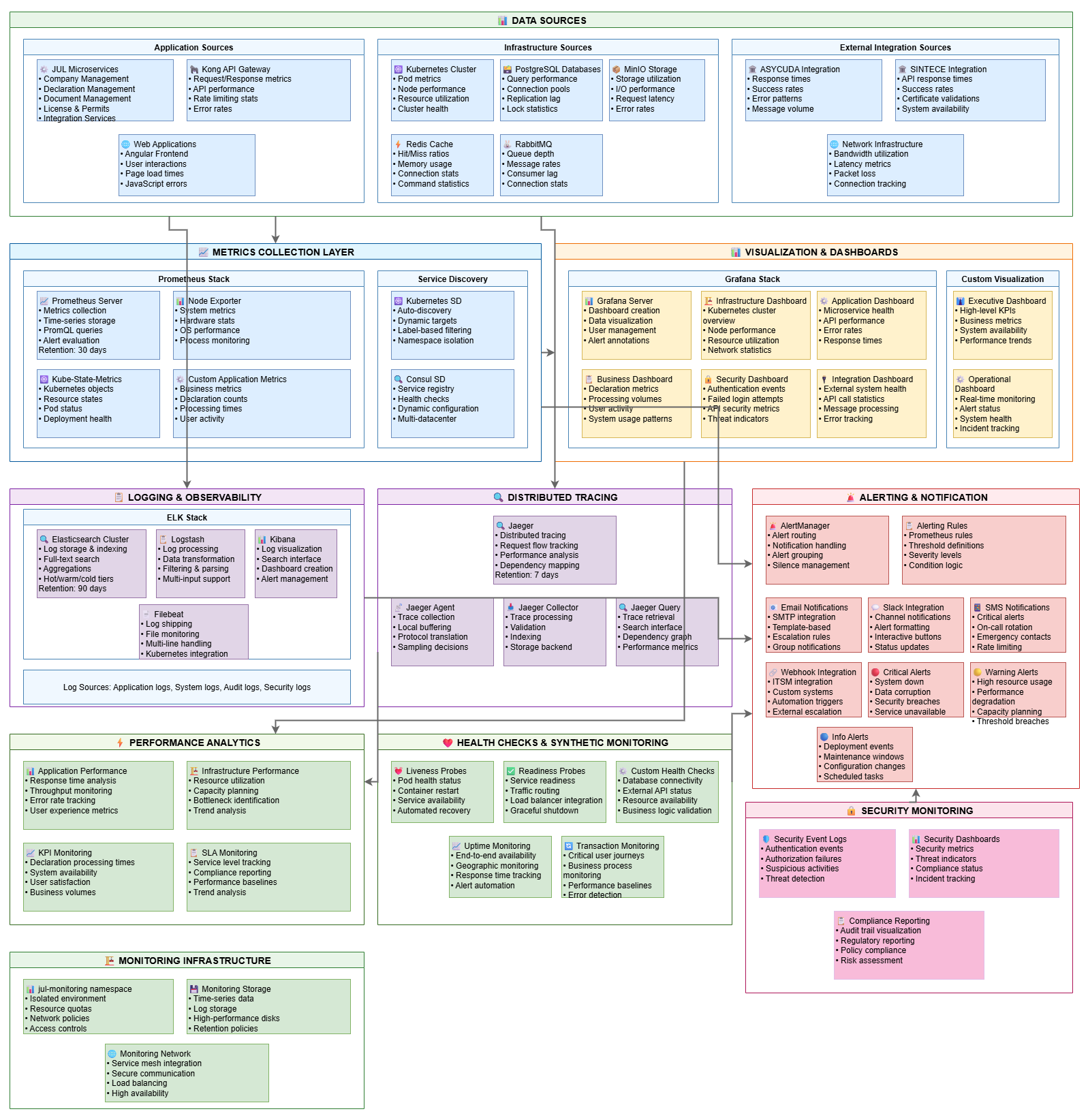
|  |  |  |
| --- | --- | --- |
| Component | Technology | Version |
| Frontend Framework | Angular | 17+ |
| Backend Framework | ASP.NET Core | 8.0 |
| Database | PostgreSQL | 15+ |
| Object Storage | MinIO | Latest |
| Message Queue | RabbitMQ | 3.12+ |
| Cache | Redis | 7+ |
| IAM | Keycloak | 23+ |
| Workflow Engine | Camunda | 8 |

Integration Protocols

|  |  |  |  |
| --- | --- | --- | --- |
| External System | Protocol | Format | Pattern |
| ASYCUDA (AGT) | SOAP/XML | UN/EDIFACT | Synchronous |
| SINTECE | REST/JSON | JSON | Synchronous |
| OGAs | REST/JSON | JSON | Asynchronous |
| Internal Services | REST/JSON | JSON | Both |

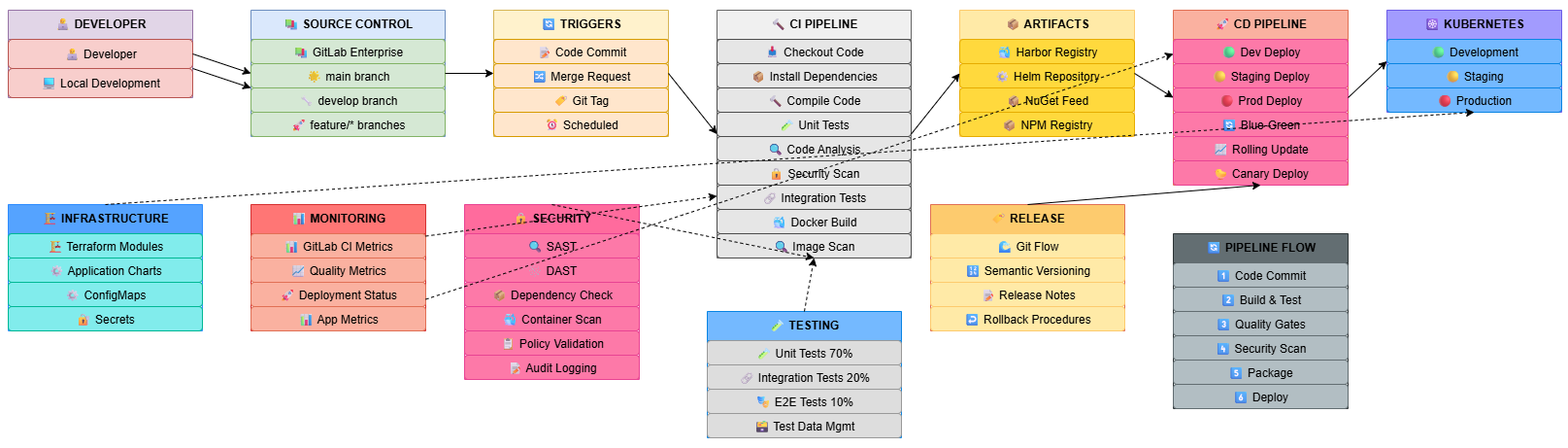
Monitoring and Observability

* Metrics Collection: Prometheus scrapes metrics from all services
* Visualization: Grafana dashboards for system and business metrics
* Log Aggregation: ELK Stack (Elasticsearch, Logstash, Kibana) for centralized logging
* Distributed Tracing: Jaeger for request flow tracking across services
* Alerting: Prometheus Alert Manager with email/SMS notifications
* APM: Application Performance Monitoring for user experience tracking
* Health Checks: Kubernetes’ liveness and readiness probes



CI/CD Pipeline

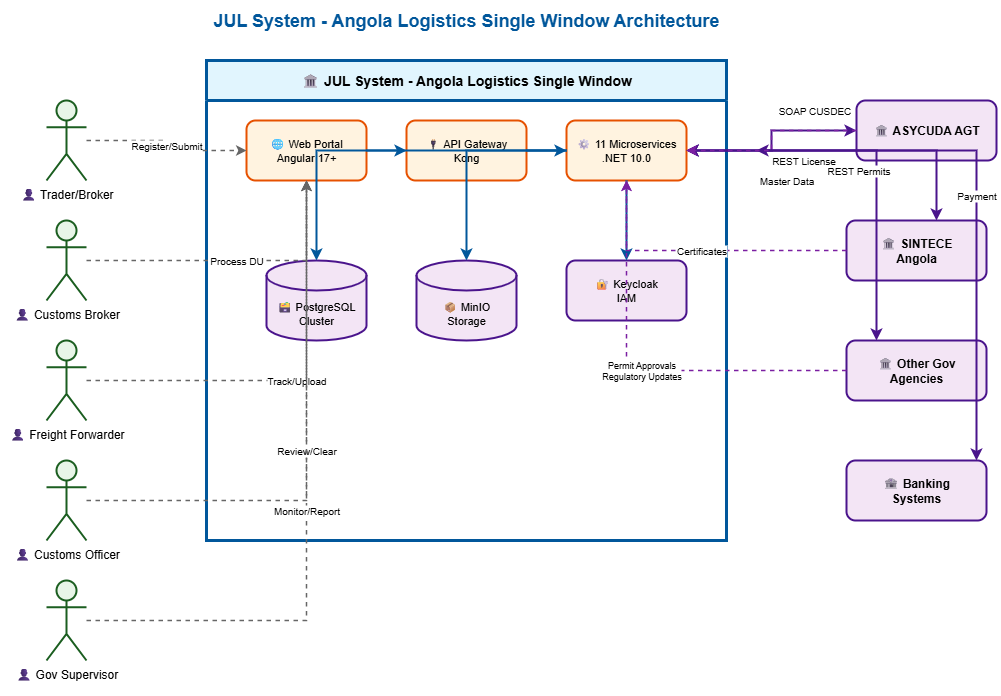
* Source Control: GitLab for version control and collaboration
* CI Pipeline: Automated build, test, and quality gates
* Code Quality: SonarQube for static code analysis
* Security Scanning: Checkmarx for vulnerability detection
* Container Registry: Harbor for private image storage
* CD Pipeline: GitOps-based deployment with ArgoCD
* Environment Promotion: Dev → Staging → Production with approvals



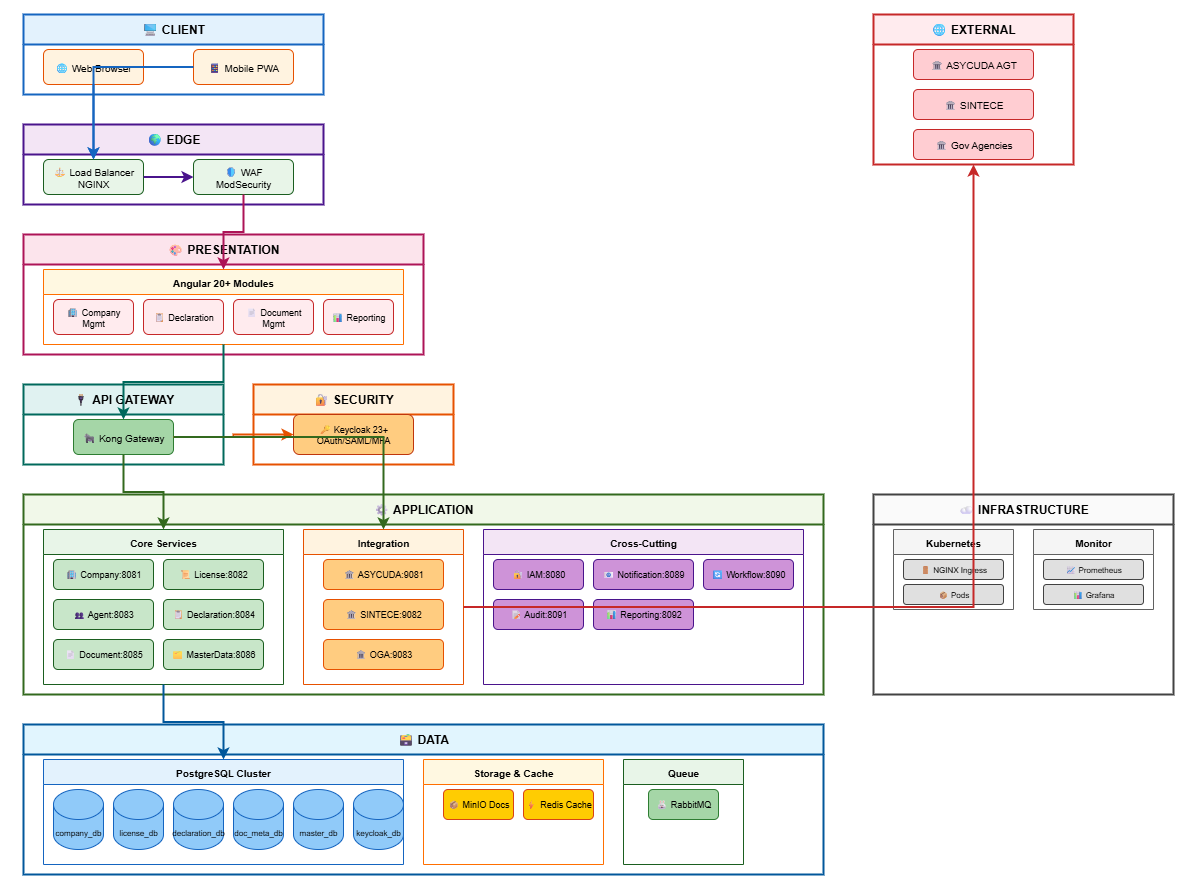
Appendices

Appendix A: Diagram References

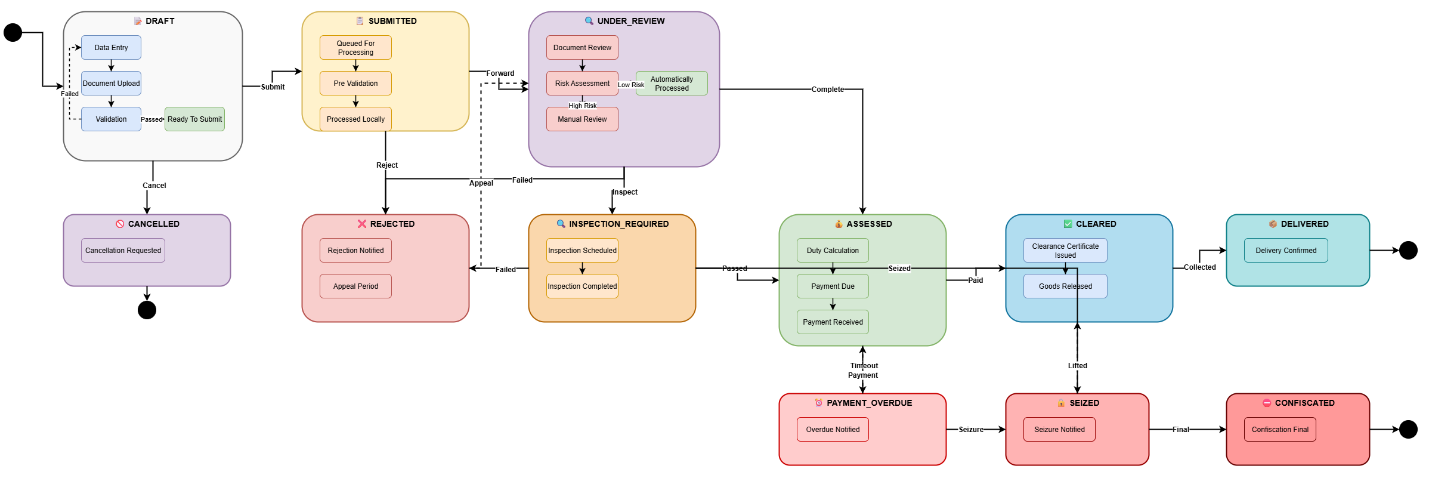
System context and stakeholders



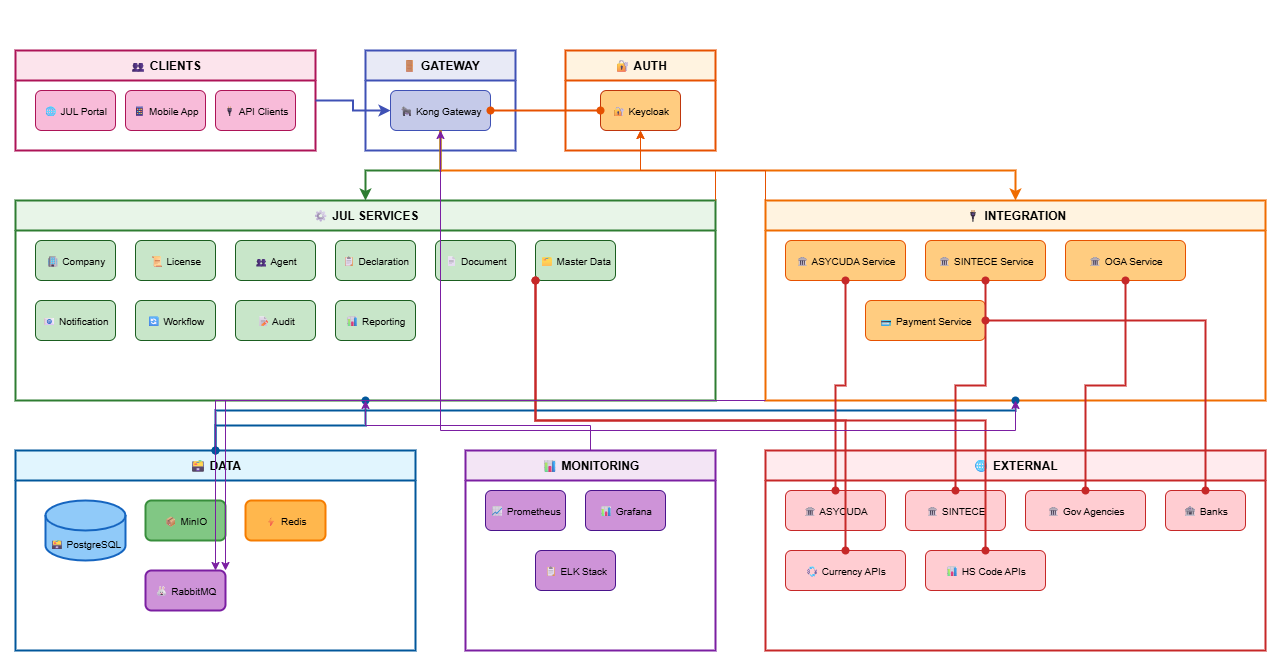
Overall system architecture



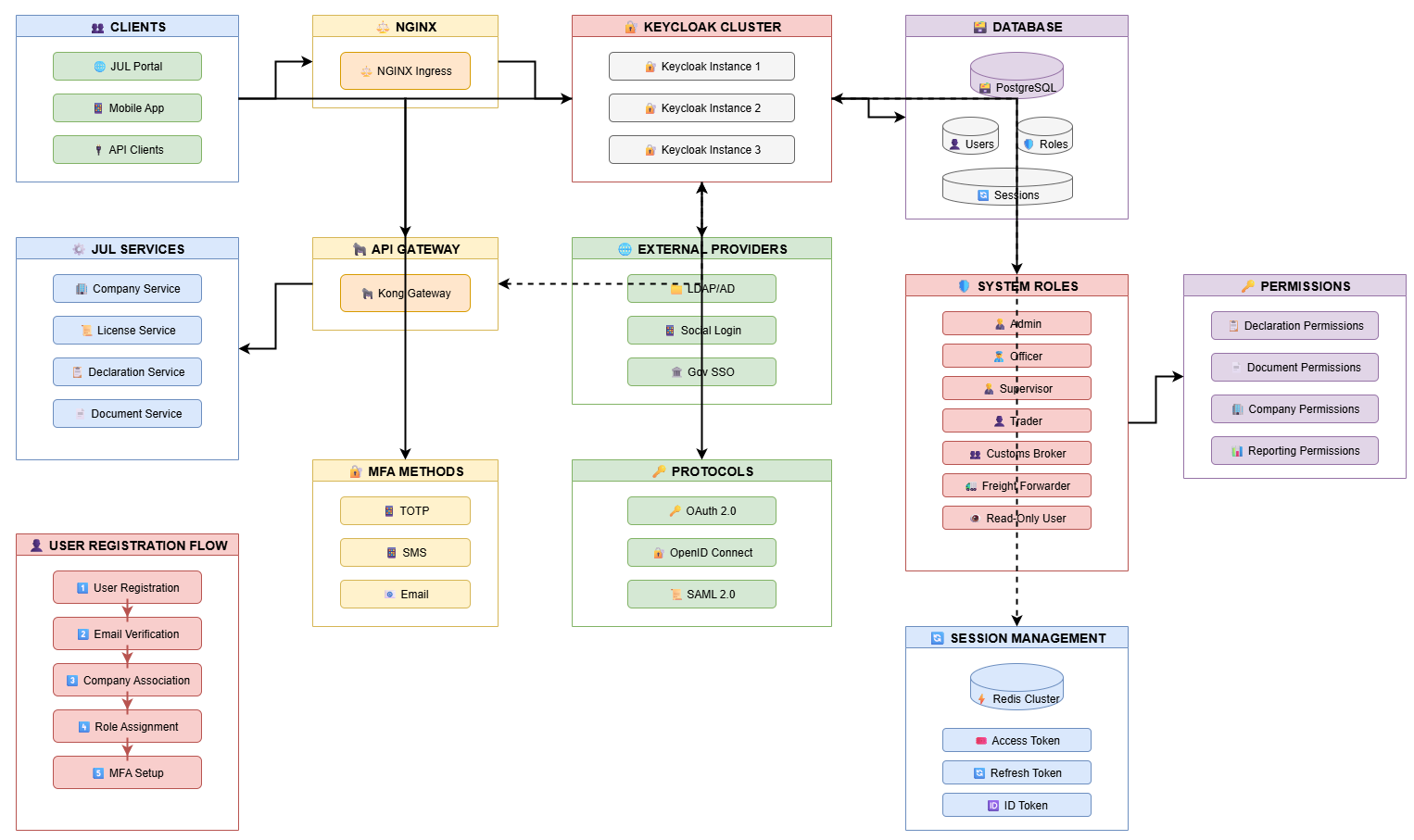
Declaration workflow states



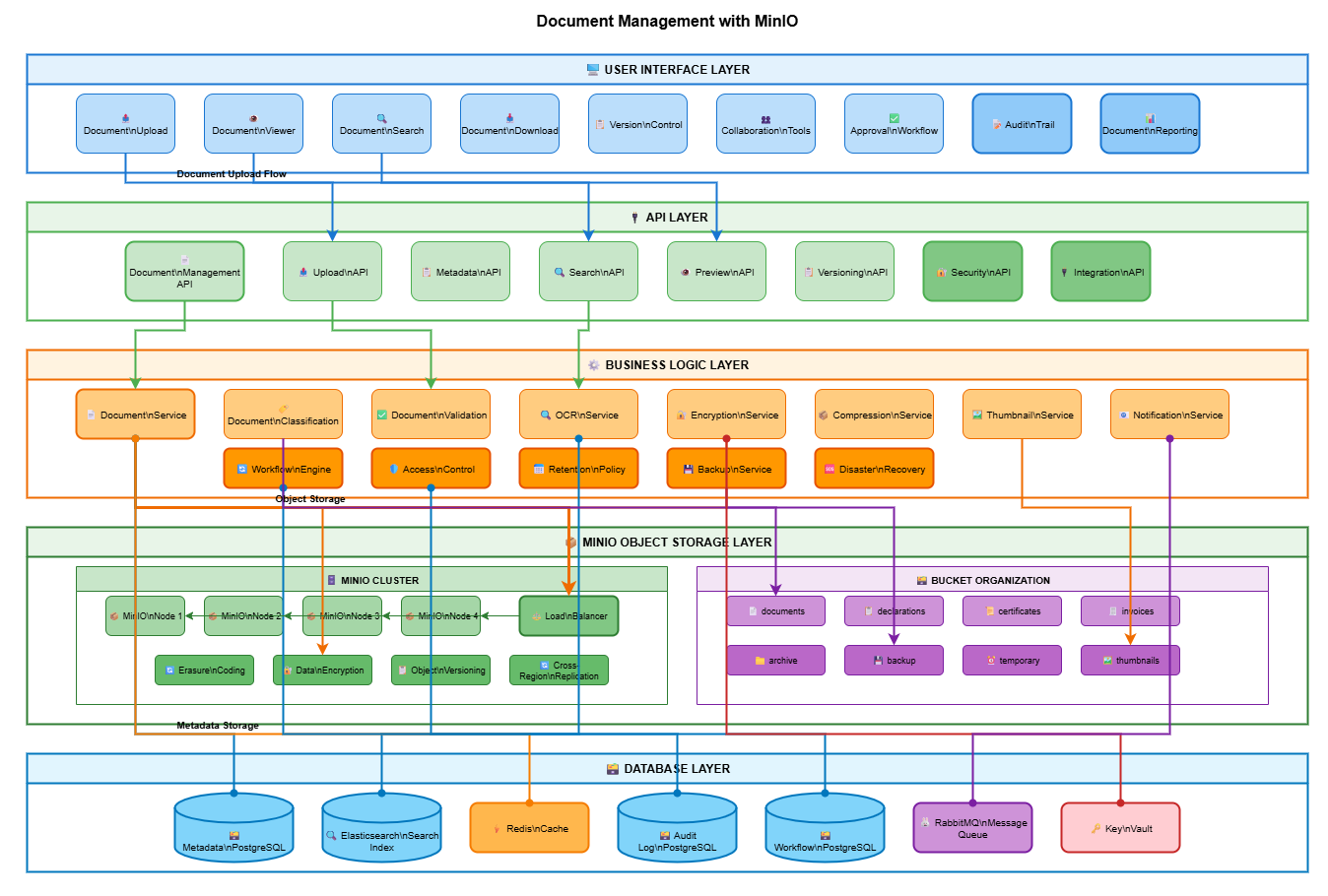
External system integrations



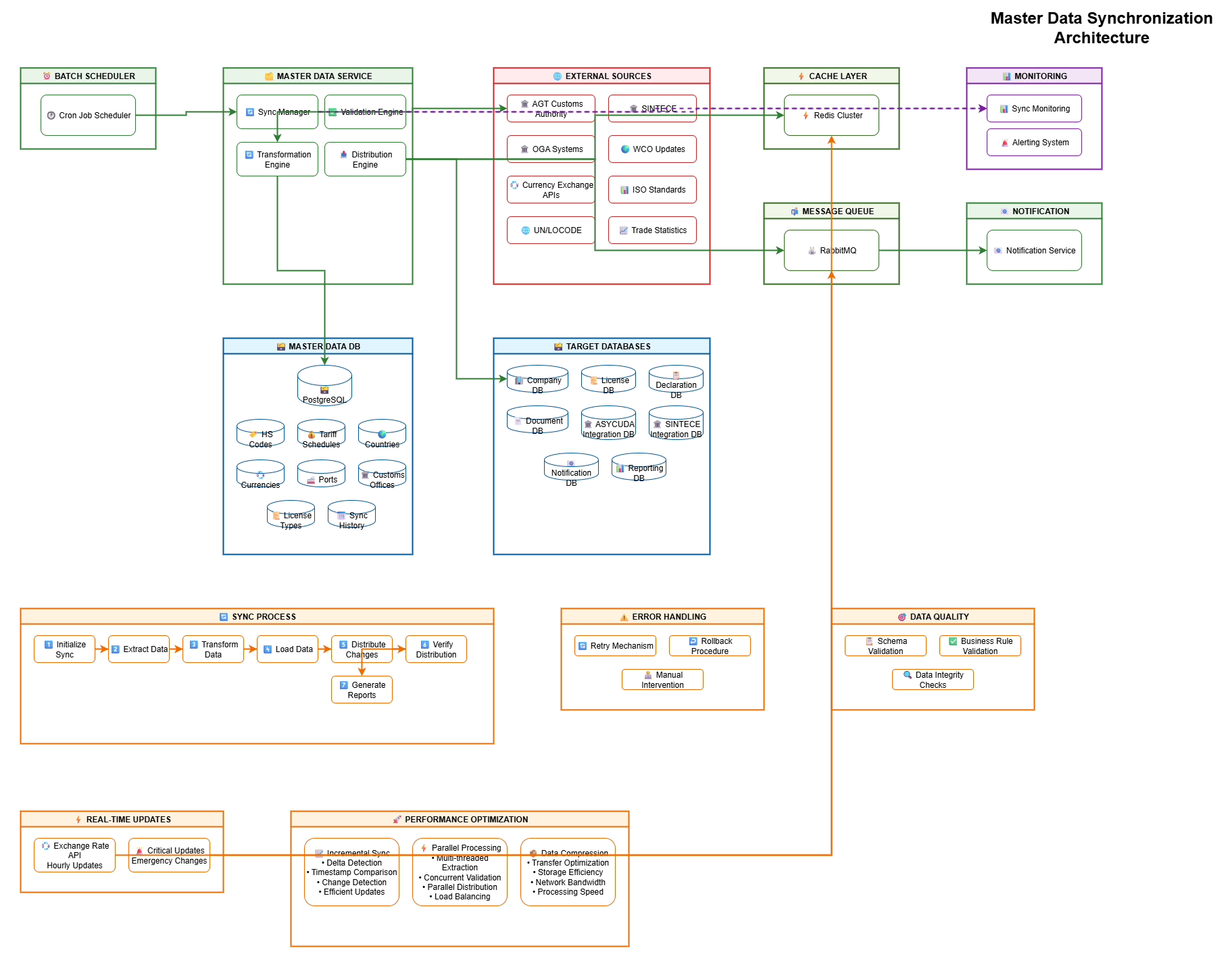
* Identity management architecture



Document storage architecture



Master data synchronization



Appendix B: Reference Documents

* JUL-AGT Integration Control Document - Draft - V2
* JUL-SINTECE Integration Control Document - Draft - v2
* JUL System User Management Architecture
* WCO Data Model Version 3.10
* UN/EDIFACT Message Standards
* UNCTAD Single Window Recommendations
* WTO Trade Facilitation Agreement Guidelines

Appendix C: Service Level Agreements

|  |  |  |
| --- | --- | --- |
| Metric | Target | Measurement |
| System Availability | 99.9% | 8.76 hours downtime/year maximum |
| API Response Time | < 3 seconds | 95th percentile of all requests |
| Throughput | 100 TPS | Sustained transaction processing rate |
| Recovery Time | < 4 hours | RTO for disaster recovery scenarios |
| Data Loss | < 1 hour | RPO for backup and recovery |