

Free Trade Agreement (FTA) Management Web Application

Comprehensive Development Planning Document

Document Version: 1.0

Date: January 15, 2026

Target Audience: Development Team, Product Managers, Business Stakeholders

Table of Contents

1. [Executive Summary](#)
 2. [System Architecture](#)
 3. [Feature Specifications](#)
 4. [UI/UX Design Guidelines](#)
 5. [Database Schema](#)
 6. [API Specifications](#)
 7. [Integration Requirements](#)
 8. [FTA Rules Engine](#)
 9. [Security & Compliance](#)
 10. [Non-Functional Requirements](#)
 11. [User Flows](#)
 12. [Development Roadmap](#)
 13. [Testing Strategy](#)
 14. [Deployment & DevOps](#)
-

1. Executive Summary

1.1 Project Overview

The FTA Management Web Application is a comprehensive, cloud-based platform designed to help companies navigate the complexities of international trade compliance, maximize duty savings through Free Trade Agreements, and streamline cross-border operations. The system automates critical processes including supplier declaration collection, multi-level origin calculations, certificate of origin generation, and compliance verification against global customs regulations.

Primary Problem Statement:

Companies engaged in international trade face significant challenges in managing FTA compliance manually, leading to:

- Missed duty savings opportunities (average 3-7% of import value)
- Compliance risks and potential penalties
- Administrative burden managing hundreds of suppliers
- Lack of visibility into supply chain origin data

- Time-consuming manual certificate creation
- Difficulty tracking regulatory changes across multiple FTAs

Solution:

An intelligent, automated FTA management platform that reduces compliance costs by 60%, increases duty savings realization by 40%, and provides end-to-end visibility and control over trade compliance processes.

1.2 Business Value Proposition

For Importers/Manufacturers:

- **Cost Reduction:** Automated duty savings identification potentially saving \$500K-\$5M annually for mid-to-large enterprises
- **Risk Mitigation:** Reduced compliance violations through automated rule checking and comprehensive audit trails
- **Efficiency Gains:** 80% reduction in time spent on FTA administration and certificate preparation
- **Scalability:** Manage hundreds of suppliers efficiently through automated workflows

For Suppliers:

- **Simplified Compliance:** Easy-to-use portal for submitting origin declarations
- **Reduced Administrative Burden:** Automated reminders and bulk declaration capabilities
- **Better Customer Relationships:** Faster response to customer compliance requirements

Market Opportunity:

- Global trade compliance market valued at \$9.8B (2025), growing at 12% CAGR
- Target market: 50,000+ mid-to-large companies in North America and Europe engaged in international trade
- Average contract value: \$25K-\$250K annually depending on transaction volume and supplier count

1.3 Key Differentiators

Competitive Advantages vs. Thomson Reuters ONESOURCE, MIC, Descartes MSR:

1. User-Centric Design

- Modern, intuitive interface designed for non-technical compliance officers
- Mobile-responsive supplier portal requiring no training
- Contextual help and guided workflows

2. AI-Powered Intelligence

- Machine learning-based origin calculation suggestions
- Predictive analytics for duty savings opportunities
- Automated HS code classification assistance
- Anomaly detection for compliance risk identification

3. Flexible Multi-Tenant Architecture

- Rapid deployment (days vs. months)
- Customizable workflows per company
- Configurable BOM depth and calculation rules
- White-label capabilities for third-party resellers

4. SMB-Friendly Pricing

- Tiered pricing model accessible to smaller companies
- Self-service implementation for SMBs
- Pay-as-you-grow scalability

5. Superior Supplier Collaboration

- Best-in-class supplier portal with document upload, bulk submission
- Automated communication and follow-up
- Supplier performance analytics

6. Advanced Analytics & Reporting

- Real-time duty savings dashboards
- Compliance risk heat maps
- Supplier response time tracking
- ROI calculators

7. API-First Architecture

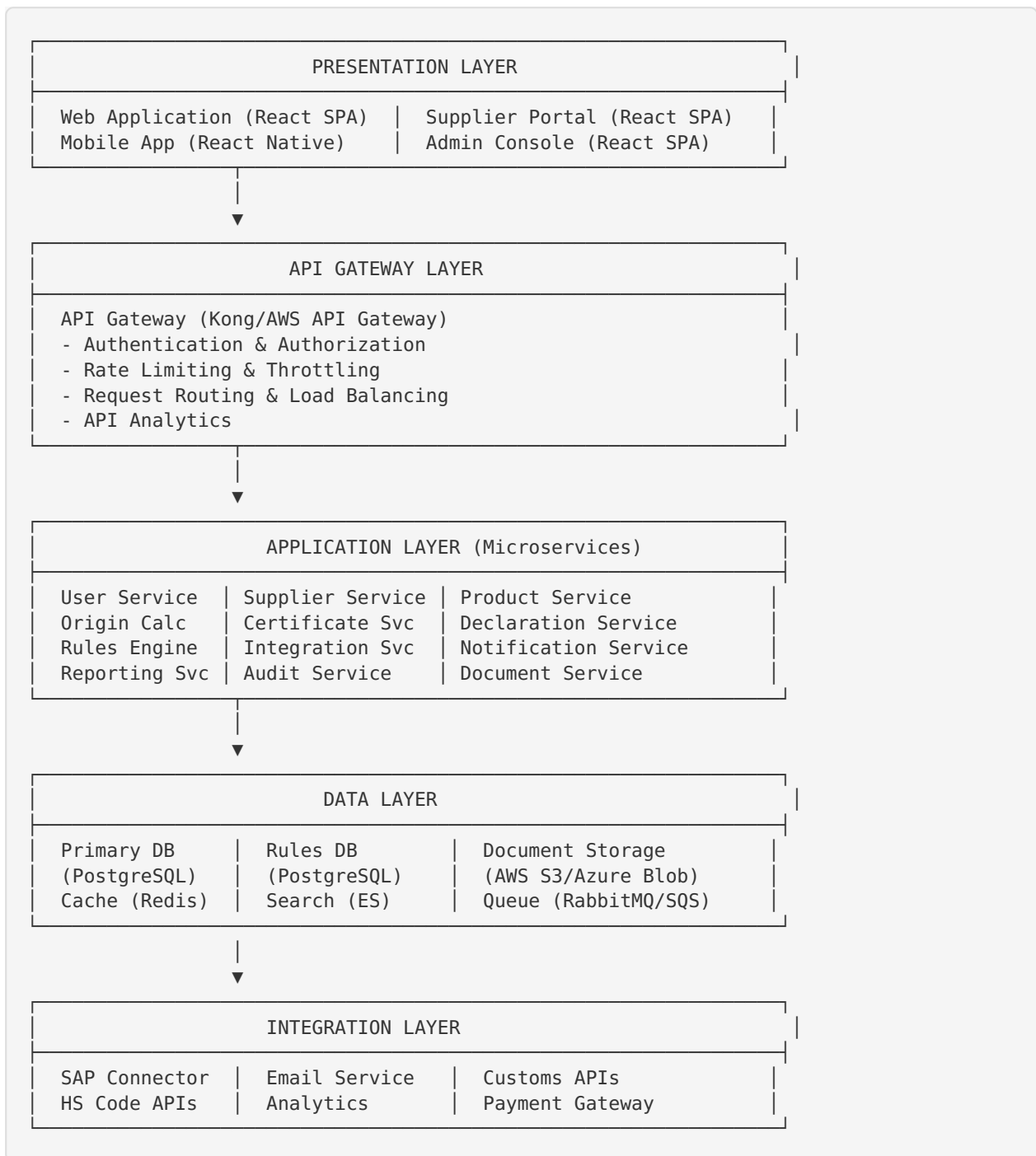
- Seamless ERP integration (SAP, Oracle, Microsoft Dynamics)
- RESTful APIs for all core functions
- Webhook support for real-time notifications

2. System Architecture

2.1 High-Level Architecture Overview

The application follows a modern, cloud-native, microservices-oriented architecture designed for scalability, resilience, and multi-tenancy.

Architecture Diagram Description:



2.2 Multi-Tenant Architecture Approach

Strategy: Row-Level Multi-Tenancy with Schema Isolation for Premium Tiers

Implementation:

- **Data Isolation:** Each database table includes a `tenant_id` column; all queries automatically filtered by tenant
- **Schema per Enterprise Tenant:** Large enterprise customers get dedicated database schemas for enhanced isolation and performance
- **Tenant Context:** Tenant identified via subdomain (company.ftamanager.com) or JWT token claims
- **Resource Quotas:** Per-tenant limits on suppliers, users, API calls, storage

Benefits:

- Cost-effective for SMB customers (shared resources)

- Scalable to thousands of tenants
- Easy data backup and migration
- Premium tier offers enhanced security and performance

Tenant Onboarding Flow:

```
New Company Signup → Tenant Provisioning Service →  
1. Create tenant record in master DB  
2. Initialize tenant data (default roles, FTA templates)  
3. Create subdomain DNS entry (optional)  
4. Send welcome email with admin credentials  
5. Trigger onboarding workflow
```

2.3 Technology Stack Recommendations

Frontend

Component	Technology	Rationale
Web Framework	React 18+ with TypeScript	Industry standard, large ecosystem, excellent performance
UI Component Library	Material-UI (MUI) or Ant Design	Professional components, accessibility built-in, customizable
State Management	Redux Toolkit or Zustand	Predictable state management for complex forms and data flows
Form Management	React Hook Form	Performance-optimized, great validation support
Data Fetching	React Query (TanStack Query)	Caching, auto-refetch, optimistic updates
Charting	Recharts or Apache ECharts	Responsive, customizable charts for analytics dashboards
Mobile	React Native (Phase 2)	Code reuse with web app, native performance

Backend

Component	Technology	Rationale
API Framework	Node.js + Express or Python + FastAPI	Node: JavaScript consistency; Python: better ML/AI libraries
Microservices	NestJS (Node) or FastAPI (Python)	Built-in DI, microservices support, excellent documentation
Authentication	Auth0 or Keycloak	Enterprise-grade, supports SSO, MFA, social logins
API Gateway	Kong or AWS API Gateway	Rate limiting, transformation, analytics
Background Jobs	Bull (Node) or Celery (Python)	Reliable job queuing for origin calculations, notifications
Real-time	Socket.io or Server-Sent Events	Live updates for certificate status, notifications

Database & Storage

Component	Technology	Rationale
Primary Database	PostgreSQL 15+	ACID compliance, JSON support, excellent for multi-tenant
Caching	Redis 7+	Session storage, API caching, rate limiting
Search Engine	Elasticsearch 8 or Meilisearch	Full-text search for products, suppliers, documents
Document Storage	AWS S3 / Azure Blob / MinIO	Scalable object storage for PDFs, certificates
Message Queue	RabbitMQ or AWS SQS	Asynchronous processing, event-driven architecture

Infrastructure & DevOps

Component	Technology	Rationale
Cloud Provider	AWS or Azure or GCP	Recommendation: AWS for maturity and service breadth
Containerization	Docker + Docker Compose (dev)	Consistent environments, easy local development
Orchestration	Kubernetes (EKS/AKS/GKE)	Auto-scaling, self-healing, declarative deployment
CI/CD	GitHub Actions or GitLab CI	Integrated with Git, extensive marketplace
Infrastructure as Code	Terraform	Cloud-agnostic, version-controlled infrastructure
Monitoring	Datadog / New Relic / Grafana + Prometheus	Application performance monitoring, alerting
Logging	ELK Stack or CloudWatch / Azure Monitor	Centralized logging, debugging, compliance

Development Tools

Purpose	Technology
Version Control	Git + GitHub/GitLab
API Documentation	Swagger/OpenAPI 3.0, Postman
Project Management	Jira, Linear, or ClickUp
Design Collaboration	Figma
Code Quality	ESLint, Prettier, SonarQube

2.4 Scalability Considerations

Horizontal Scaling Strategies:

1. Stateless Application Servers

- All services designed to be stateless
- Session data stored in Redis, not server memory
- Load balancers distribute requests across instances

2. Database Scaling

- **Read Replicas:** For reporting and analytics queries
- **Partitioning:** Large tables (declarations, audit_logs) partitioned by tenant_id and date

- **Connection Pooling:** PgBouncer for efficient database connections
- **Caching Strategy:**
 - Redis for FTA rules (rarely change), user sessions
 - Application-level caching for origin calculations (cache key: product_id + BOM hash)

3. **Asynchronous Processing**

- Origin calculations for complex BOMs processed in background workers
- Certificate generation queued for batch processing
- Email notifications sent asynchronously

4. **CDN for Static Assets**

- React SPA, images, CSS served via CloudFront/CloudFlare
- Reduced latency for global users

5. **Auto-Scaling Rules**

- Scale web servers based on CPU >70% or request latency >500ms
- Scale background workers based on queue depth >100 jobs

Performance Targets:

- Web page load: <2 seconds (initial), <500ms (subsequent)
- API response time: <300ms (p95), <1s (p99)
- Origin calculation: <5 seconds for 5-level BOM with 100 components
- Support: 1000+ concurrent users per company, 10,000 suppliers managed

2.5 Security Architecture

Defense in Depth Strategy:

Layer 1: Network Security

- WAF (AWS WAF/Cloudflare)
- DDoS Protection
- VPC with private subnets

↓

Layer 2: Application Security

- API Gateway with rate limiting
- OAuth 2.0 + JWT tokens
- Input validation & sanitization
- SQL injection prevention (parameterized queries)

↓

Layer 3: Data Security

- Encryption at rest (AES-256)
- Encryption in transit (TLS 1.3)
- Database encryption
- Secrets management (AWS Secrets Manager/Vault)

↓

Layer 4: Access Control

- Role-Based Access Control (RBAC)
- Multi-Factor Authentication (MFA) **for** admins
- IP whitelisting (optional **for** enterprises)

↓

Layer 5: Audit & Compliance

- Comprehensive audit logging
- Immutable audit trail
- Regular security scans
- Penetration testing (annually)

Key Security Features:

- **Zero Trust Architecture:** Every request authenticated and authorized
- **Tenant Isolation:** Mandatory tenant_id filtering prevents cross-tenant data access
- **API Key Management:** Rotating API keys for SAP integration
- **Document Sanitization:** Uploaded PDFs scanned for malware
- **Audit Logging:** All data modifications logged with user, timestamp, changes

3. Feature Specifications

3.1 MUST-HAVE Features (MVP - Phase 1)

Feature 1: User Authentication & Authorization

Description: Secure login system with role-based access control supporting four user roles.

User Stories:

- As an **Admin**, I want to create and manage user accounts so that I can control who has access to the system
- As a **Compliance Officer**, I want to log in with my email and password so that I can access FTA

management tools

- As a **Supplier**, I want to access the supplier portal without needing extensive permissions so that I can submit declarations easily
- As an **Auditor**, I want read-only access to all data so that I can review compliance documentation

Acceptance Criteria:

- ☒ Email/password authentication with password strength requirements (min 12 chars, uppercase, lowercase, number, special char)
- ☒ Role-based permissions enforced at API level
- ☒ Password reset via email
- ☒ Session timeout after 30 minutes of inactivity
- ☒ Account lockout after 5 failed login attempts
- ☒ Audit log of all login attempts

Technical Specifications:

- JWT tokens with 1-hour expiry, 7-day refresh tokens
- Passwords hashed with bcrypt (cost factor 12)
- Role permissions stored in database, cached in Redis

Role Permissions Matrix:

Feature	Admin	Compliance Officer	Supplier	Auditor
Manage Users	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Manage Suppliers	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> Read-only
Submit Declarations	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> (own only)	<input checked="" type="checkbox"/>
Calculate Origin	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Generate Certificates	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
View Reports	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> Own data	<input checked="" type="checkbox"/>
Configure FTA Rules	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Access Audit Logs	<input checked="" type="checkbox"/>	<input type="checkbox"/> Limited	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Feature 2: Supplier Management

Description: Centralized database of suppliers with contact information, product assignments, and declaration status tracking.

User Stories:

- As a **Compliance Officer**, I want to add new suppliers with their contact details so that I can request origin declarations from them
- As a **Compliance Officer**, I want to see which suppliers haven't responded to declaration requests so that I can follow up
- As an **Admin**, I want to import suppliers from SAP so that I don't have to enter data manually

Acceptance Criteria:

- ☒ Create, read, update, delete (CRUD) operations for suppliers
- ☒ Required fields: Company name, contact person, email, country
- ☒ Optional fields: Phone, address, tax ID, DUNS number
- ☒ Bulk import via CSV/Excel
- ☒ Supplier status indicators (Active, Inactive, Pending)
- ☒ Link suppliers to specific products/parts
- ☒ Track declaration submission history per supplier
- ☒ Search and filter by name, country, status

Technical Specifications:

- Database table: `suppliers` with `tenant_id` foreign key
- Elasticsearch index for fast search
- CSV import validation: email format, country code (ISO 3166-1)
- Duplicate detection based on company name + country

Database Fields:

```
CREATE TABLE suppliers (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  tenant_id UUID NOT NULL REFERENCES tenants(id),
  company_name VARCHAR(255) NOT NULL,
  contact_person VARCHAR(255),
  email VARCHAR(255) NOT NULL,
  phone VARCHAR(50),
  country_code CHAR(2) NOT NULL,
  address TEXT,
  tax_id VARCHAR(100),
  duns_number VARCHAR(20),
  status VARCHAR(20) DEFAULT 'active',
  last_declaration_date TIMESTAMP,
  created_at TIMESTAMP DEFAULT NOW(),
  updated_at TIMESTAMP DEFAULT NOW(),
  created_by UUID REFERENCES users(id),
  CONSTRAINT unique_supplier_per_tenant UNIQUE(tenant_id, company_name, country_code)
);

CREATE INDEX idx_suppliers_tenant ON suppliers(tenant_id);
CREATE INDEX idx_suppliers_status ON suppliers(tenant_id, status);
```

Feature 3: Product & BOM Management

Description: Product catalog with hierarchical Bill of Materials (BOM) supporting up to 5 levels of nesting (configurable).

User Stories:

- As a **Compliance Officer**, I want to define products with their BOM structures so that I can calculate

origin accurately

- As a **Compliance Officer**, I want to import BOMs from SAP so that I don't have to manually enter component data
- As a **Compliance Officer**, I want to configure BOM depth limits so that calculations align with our manufacturing processes

Acceptance Criteria:

- ☒ CRUD operations for products with HS code, description, unit of measure
- ☒ Define BOM with parent-child relationships (supports up to 5 levels)
- ☒ Quantity and unit cost for each BOM component
- ☒ SAP integration to sync products and BOMs
- ☒ System-wide configuration for BOM depth (default: 5, max: 10)
- ☒ Visual BOM tree display
- ☒ Detect and prevent circular BOM references

Technical Specifications:

- **Adjacency List Model** for BOM hierarchy (simple queries, flexible)
- Alternative: **Nested Sets** or **Closure Table** for complex hierarchy queries
- Recursive queries using PostgreSQL CTEs (Common Table Expressions)

Database Schema:

```
CREATE TABLE products (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  tenant_id UUID NOT NULL REFERENCES tenants(id),
  product_code VARCHAR(100) NOT NULL,
  description TEXT NOT NULL,
  hs_code VARCHAR(10) NOT NULL,
  unit_of_measure VARCHAR(20),
  is_finished_good BOOLEAN DEFAULT FALSE,
  created_at TIMESTAMP DEFAULT NOW(),
  updated_at TIMESTAMP DEFAULT NOW(),
  CONSTRAINT unique_product_code_per_tenant UNIQUE(tenant_id, product_code)
);

CREATE TABLE bom_items (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  tenant_id UUID NOT NULL REFERENCES tenants(id),
  parent_product_id UUID NOT NULL REFERENCES products(id) ON DELETE CASCADE,
  child_product_id UUID NOT NULL REFERENCES products(id),
  quantity DECIMAL(15, 4) NOT NULL,
  unit_cost DECIMAL(15, 2),
  supplier_id UUID REFERENCES suppliers(id),
  level INT DEFAULT 1,
  created_at TIMESTAMP DEFAULT NOW(),
  CONSTRAINT no_self_reference CHECK (parent_product_id != child_product_id)
);

CREATE INDEX idx_bom_parent ON bom_items(tenant_id, parent_product_id);
CREATE INDEX idx_bom_child ON bom_items(tenant_id, child_product_id);
```

BOM Depth Configuration:

```
CREATE TABLE tenant_settings (
  tenant_id UUID PRIMARY KEY REFERENCES tenants(id),
  max_bom_depth INT DEFAULT 5,
  default_fta VARCHAR(20) DEFAULT 'USMCA',
  ...
);
```

Feature 4: Supplier Declaration Portal

Description: Self-service web portal where suppliers submit origin declarations for their products.

User Stories:

- As a **Supplier**, I want to receive an email invitation to the portal so that I know how to submit declarations
- As a **Supplier**, I want to submit origin declarations for multiple parts at once so that I can save time
- As a **Supplier**, I want to upload supporting documents (e.g., supplier certificates) so that I can provide complete documentation
- As a **Compliance Officer**, I want to send automated reminders to suppliers who haven't submitted declarations so that I can ensure timely responses

Acceptance Criteria:

- ☒ Supplier receives email with unique secure link (no login required) or optional login
- ☒ Simple form: Select FTA, declare country of origin, select origin criterion (A, B, C, D for USMCA)
- ☒ Bulk declaration entry for multiple parts
- ☒ Upload supporting documents (PDF, Excel, Word, max 10MB per file)
- ☒ Save draft declarations for later completion
- ☒ Submit declarations with timestamp and electronic signature
- ☒ Automated email confirmations upon submission
- ☒ Automated reminder emails (configurable: e.g., 7 days, 14 days, 30 days after request)


Technical Specifications:

- **Secure Access:** Tokenized links with 30-day expiry or optional supplier account creation
- **Form Validation:** Required fields, date format, origin criterion matching FTA rules
- **Document Storage:** S3 with virus scanning (ClamAV or AWS GuardDuty)
- **Email Service:** SendGrid/AWS SES with templating

Declaration Form Fields:

- **FTA Selection:** [USMCA, CFTA, EU-Canada, ASEAN, CPTPP, Other]
- **Part Number:** [Text input or multi-select **for** bulk]
- **Country of Origin:** [Dropdown - ISO 3166-1]
- **Origin Criterion:** [A, B, C, D] (varies by FTA)
- **Producer:** [Text input]
- **Blanket Period:** [Start date - End date, max 12 months]
- **RVC Percentage (if applicable):** [Numeric input, 0-100%]
- **HS Code:** [Auto-populated or editable]
- **Supporting Documents:** [File upload, multi-file]
- **Comments:** [Text area]
- **Certifier Name:** [Text input]
- **Certifier Title:** [Text input]
- **Electronic Signature:** [Checkbox agreement + typed name]
- **Date:** [Auto-populated]

Workflow:

1. Compliance Officer creates declaration request **for** supplier(s)
2. System sends email to supplier with secure link
3. Supplier clicks link  Opens portal (no login **or** simple PIN)
4. Supplier fills form, uploads documents, submits
5. System validates data, stores declaration, sends confirmation
6. Compliance Officer notified of new submission
7. If no response **in** X days, send automated reminder

Feature 5: Origin Calculation Engine

Description: Automated calculation of product origin based on multi-level BOM analysis and FTA-specific rules.

User Stories:

- As a **Compliance Officer**, I want to calculate origin for a finished good so that I can determine if it qualifies for FTA benefits
- As a **Compliance Officer**, I want to see a visual breakdown of origin calculation steps so that I can verify accuracy
- As a **Compliance Officer**, I want the system to flag missing supplier declarations so that I can request them before calculating

Acceptance Criteria:

- ☒ Calculate origin using FTA-specific rules (USMCA, CFTA, EU, ASEAN, CPTPP)
- ☒ Support multiple calculation methods:
 - Transaction Value Method: $RVC = ((TV - VNM) / TV) \times 100$
 - Net Cost Method: $RVC = ((NC - VNM) / NC) \times 100$
- ☒ Recursive calculation through BOM (up to configured depth)
- ☒ Handle tariff shift rules (product-specific rules in Annex 4-B for USMCA)
- ☒ Accumulation of origin across multiple production stages
- ☒ Flag missing declarations for BOM components
- ☒ Display calculation results with detailed breakdown
- ☒ Save calculation results for audit trail
- ☒ Re-calculate automatically when BOM or declarations change



Technical Specifications:

- **Algorithm:** Recursive bottom-up calculation starting from leaf components
- **Caching:** Cache calculation results with cache invalidation on BOM/declaration updates
- **Performance:** Process 5-level BOM with 100 components in <5 seconds
- **Concurrency:** Queue large calculations for background processing

Calculation Flow:

Input: Product ID, FTA, Calculation Method



1. Fetch BOM tree (recursive query up to max depth)
2. For **each** component, fetch supplier declaration
3. If declaration missing  flag component, assume non-originating
4. For **each** level (bottom-up):
 - a. Calculate component value (quantity  unit cost)
 - b. Determine **if** component originates (per declaration)
 - c. Sum originating vs. non-originating value
5. Calculate RVC **for** finished good
6. Apply tariff shift rule (**if** applicable)
7. Apply de minimis rule (**if** applicable)
8. Determine **final** origin status: Qualifying / Non-Qualifying / Indeterminate
9. Save result **with** calculation details



Output: Origin determination result **with** detailed report

Database Schema:

```
CREATE TABLE origin_calculations (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  tenant_id UUID NOT NULL REFERENCES tenants(id),
  product_id UUID NOT NULL REFERENCES products(id),
  fta_code VARCHAR(20) NOT NULL,
  calculation_method VARCHAR(50) NOT NULL,
  total_value DECIMAL(15, 2),
  originating_value DECIMAL(15, 2),
  non_originating_value DECIMAL(15, 2),
  rvc_percentage DECIMAL(5, 2),
  qualifies BOOLEAN,
  missing_declarations JSONB,
  calculation_details JSONB,
  calculated_at TIMESTAMP DEFAULT NOW(),
  calculated_by UUID REFERENCES users(id)
);

CREATE INDEX idx_origin_calcs_product ON origin_calculations(tenant_id, product_id);
```




Feature 6: Certificate of Origin Generation

Description: Automated creation of FTA-compliant certificates of origin based on calculated origin data.

User Stories:

- As a **Compliance Officer**, I want to generate a certificate of origin for qualifying products so that I can provide it to importers
- As a **Compliance Officer**, I want to use a pre-formatted template that complies with FTA requirements so that I ensure acceptance by customs
- As a **Compliance Officer**, I want to generate blanket certificates covering multiple shipments so that I can reduce administrative burden

Acceptance Criteria:

-  Generate certificates for individual products or blanket periods (up to 12 months)
-  FTA-specific templates (USMCA, CFTA, EU, ASEAN, CPTPP)
-  Auto-populate required data elements:

- Certifier information (company, address, contact)
- Producer information
- Importer information (optional)
- Product description and HS code (6-digit minimum)
- Origin criterion (A, B, C, D per USMCA)
- Blanket period (if applicable)
- Authorized signature and date
- ☒ PDF export with company logo and branding
- ☒ Electronic signature support
- ☒ Certificate versioning and revision history
- ☒ Bulk certificate generation for multiple products

Technical Specifications:

- **PDF Generation:** Library like PDFKit (Node.js) or ReportLab (Python)
- **Templates:** HTML/CSS templates rendered to PDF
- **Digital Signatures:** Optional integration with DocuSign/Adobe Sign for legally binding signatures
- **Storage:** S3 with versioning enabled

Certificate Data Elements (USMCA Example):

1. Certifier: Name, title, company, address, email, phone
2. Exporter: Name, address, tax ID
3. Producer: Name, address, tax ID (or "Available upon request")
4. Importer: Name, address, tax ID (or "Various" for blanket)
5. Description of goods and HS classification (6-digit)
6. Origin criterion: A, B, C, or D
7. Blanket period: [Start date - End date] (if applicable)
8. Authorized signature: Electronic/digital signature
9. Date: [Certification date]

Workflow:

1. Compliance Officer selects product(s) with qualifying origin
2. System validates origin calculation **is** current **and** qualifies
3. Officer selects FTA **and** certificate type (single/blanket)
4. System pre-populates form with product **and** company data
5. Officer reviews, adds importer details (**if** known)
6. Officer adds electronic signature (typed name + agreement)
7. System generates PDF certificate
8. Officer downloads/emails certificate to customer
9. System logs certificate issuance **in** audit trail

Feature 7: Dashboard & Reporting

Description: Real-time analytics dashboard showing duty savings, compliance status, supplier performance, and certificate activity.

User Stories:

- As a **Compliance Officer**, I want to see total duty savings realized through FTAs so that I can demonstrate ROI
- As an **Admin**, I want to see supplier response rates so that I can identify non-responsive suppliers

- As a **Compliance Officer**, I want to filter reports by date range, FTA, and product category so that I can analyze specific segments

Acceptance Criteria:

- ☒ Key metrics displayed on homepage dashboard:
 - Total potential duty savings (YTD, MTD)
 - Realized vs. potential savings
 - Number of certificates issued
 - Supplier declaration compliance rate
 - Products with qualifying origin vs. non-qualifying
- ☒ Interactive charts: Line charts for savings trends, pie charts for FTA distribution, bar charts for supplier performance
- ☒ Exportable reports (PDF, Excel, CSV)
- ☒ Customizable date ranges
- ☒ Drill-down capability (e.g., click on FTA to see products using that FTA)

Technical Specifications:

- **Frontend Charting:** Recharts or Apache ECharts
- **Data Aggregation:** Pre-aggregated data in materialized views or scheduled batch jobs
- **Real-time Updates:** WebSocket connection for live dashboard updates (optional)
- **Report Generation:** Background job for large exports

Dashboard Widgets:

1. Duty Savings Calculator

- Formula: $(\text{Non-FTA duty rate} - \text{FTA duty rate}) \times \text{Import value}$
- Display: Total savings YTD, average per shipment
- Visualization: Line chart showing monthly savings trend

1. Compliance Overview

- Total suppliers: X
- Suppliers with current declarations: Y (Z%)
- Products with origin calculated: A (B%)
- Certificates issued this month: C

2. Supplier Performance

- Table: Supplier name, declaration requests sent, responses received, response rate, avg. response time
- Filter: By country, date range
- Actions: Send reminder, view details

3. Origin Status Breakdown

- Pie chart: Qualifying products vs. non-qualifying vs. indeterminate
- Count and percentage for each category

4. Recent Activity Feed

- Latest declarations submitted
 - Certificates generated
 - Origin calculations completed
 - System alerts
-

Feature 8: Audit Trail & Compliance Logging

Description: Comprehensive, immutable audit log of all system activities for compliance verification.

User Stories:

- As an **Auditor**, I want to see a complete history of all changes to a product's BOM so that I can verify accuracy
- As a **Compliance Officer**, I want to see who generated a certificate and when so that I can respond to customs inquiries
- As an **Admin**, I want to ensure audit logs cannot be modified so that they maintain integrity

Acceptance Criteria:

- ☒ Log all data modifications: Create, update, delete operations
- ☒ Record: User ID, timestamp, action type, entity type, entity ID, before/after values (for updates)
- ☒ Immutable logs (append-only, cannot be edited or deleted)
- ☒ Search and filter audit logs by user, date, entity type, action
- ☒ Export audit logs for external review
- ☒ Retention: Minimum 7 years (configurable per compliance requirements)

Technical Specifications:

- **Implementation:** Database triggers or application-level middleware
- **Storage:** Separate audit_logs table with partitioning by date
- **Performance:** Asynchronous logging to avoid blocking user requests

Database Schema:

```
CREATE TABLE audit_logs (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  tenant_id UUID NOT NULL,
  user_id UUID REFERENCES users(id),
  action VARCHAR(50) NOT NULL,
  entity_type VARCHAR(100) NOT NULL,
  entity_id UUID,
  changes JSONB,
  ip_address INET,
  user_agent TEXT,
  created_at TIMESTAMP DEFAULT NOW()
) PARTITION BY RANGE (created_at);

CREATE INDEX idx_audit_logs_tenant ON audit_logs(tenant_id, created_at DESC);
CREATE INDEX idx_audit_logs_user ON audit_logs(user_id, created_at DESC);
CREATE INDEX idx_audit_logs_entity ON audit_logs(entity_type, entity_id);
```

Logged Actions:

- User login/logout
 - Supplier create/update/delete
 - Product create/update/delete
 - BOM create/update/delete
 - Declaration submission
 - Origin calculation
 - Certificate generation
 - Settings changes
 - User role assignments
-

Feature 9: SAP ERP Integration (Basic)

Description: Bidirectional integration with SAP ERP to sync suppliers, products, BOMs, and push certificate data.

User Stories:

- As a **Compliance Officer**, I want to import suppliers from SAP automatically so that I maintain a single source of truth
- As a **Compliance Officer**, I want to push certificate data back to SAP so that purchasing and logistics teams can access it

Acceptance Criteria:


- ☒ Import suppliers from SAP (nightly batch sync or on-demand)
- ☒ Import products and BOMs from SAP (nightly batch sync or on-demand)
- ☒ Export certificate data to SAP (real-time or batch)
- ☒ Error handling and sync status reporting
- ☒ Manual re-sync option for individual records
- ☒ Conflict resolution (e.g., SAP data changed vs. local changes)

Technical Specifications:


- **Integration Method:** SAP OData APIs, RFC calls, or IDoc messages
- **Authentication:** OAuth 2.0 or API key with SAP Cloud Platform
- **Sync Frequency:** Configurable (default: daily at 2 AM)
- **Data Mapping:** Configurable field mappings per tenant

Data Sync Flow:

SAP  FTA App (Import):

1. Scheduled job triggers at configured time
2. Call SAP API to fetch suppliers (filter: new/updated since last sync)
3. **Transform** SAP data format  FTA app format
4. Upsert suppliers **in** FTA app database
5. Log sync status (success/failure, records processed)
6. Send notification to admin **if** errors

FTA App  SAP (Export):

1. Certificate generated **in** FTA app
2. System creates **export** job
3. **Transform** certificate data  SAP format
4. Call SAP API to create/update certificate record
5. Mark certificate **as** "exported to SAP"
6. Log **export** status

API Endpoints (FTA App Side):

- POST /api/integrations/sap/sync-suppliers
- POST /api/integrations/sap/sync-products
- POST /api/integrations/sap/sync-boms
- GET /api/integrations/sap/sync-status

Feature 10: Notification System

Description: Automated email notifications for key events (declaration requests, reminders, certificate generation).

User Stories:

- As a **Supplier**, I want to receive an email when a declaration is requested so that I know I need to take action
- As a **Compliance Officer**, I want to receive an email when a supplier submits a declaration so that I can review it promptly
- As a **Compliance Officer**, I want to configure notification preferences so that I control email volume

Acceptance Criteria:

- ☒ Email notifications for:
 - Declaration request sent to supplier
 - Declaration submitted by supplier
 - Declaration reminder (auto-escalation at 7, 14, 30 days)
 - Certificate generated
 - Origin calculation completed (for long-running calcs)
 - System alerts (e.g., integration failures)
- ☒ Configurable notification preferences per user
- ☒ Email templates with company branding
- ☒ Unsubscribe option for non-critical notifications
- ☒ Notification history log

Technical Specifications:

- **Email Service:** SendGrid, AWS SES, or Mailgun
- **Templates:** HTML email templates with variables
- **Queue:** Asynchronous email sending via RabbitMQ/SQS
- **Retry Logic:** Retry failed sends up to 3 times

Email Templates:**1. Declaration Request**

- To: Supplier
- Subject: "Declaration Request for [Part Number] - [Company Name]"
- Body: Instructions, secure link, deadline

1. Declaration Submitted

- To: Compliance Officer
- Subject: "New Declaration Submitted by [Supplier Name]"
- Body: Supplier name, parts declared, link to review

2. Declaration Reminder

- To: Supplier
- Subject: "Reminder: Declaration Request for [Part Number]"
- Body: Friendly reminder, deadline, link

3. Certificate Generated

- To: Compliance Officer
 - Subject: "Certificate of Origin Generated for [Product]"
 - Body: Product details, certificate number, download link
-

3.2 NICE-TO-HAVE Features (Phase 2+)

Feature 11: Advanced Analytics & Predictive Insights

Description: AI-powered analytics to predict duty savings opportunities, identify compliance risks, and recommend supply chain optimizations.

User Stories:

- As a **Compliance Officer**, I want the system to recommend products that could qualify for FTAs with minor sourcing changes
- As a **Compliance Officer**, I want to see forecasted duty savings for upcoming quarters based on historical data

Capabilities:

- Machine learning models to predict which products are likely to qualify with sourcing adjustments
 - Anomaly detection for unusual declaration patterns (fraud detection)
 - Predictive analytics for duty savings forecasts
 - “What-if” analysis: Change supplier → recalculate origin automatically
 - Supply chain optimization recommendations (e.g., switch supplier from X to Y to gain USMCA eligibility)
-

Feature 12: Mobile Application

Description: Native mobile apps (iOS/Android) for on-the-go access, optimized for suppliers and compliance officers.

User Stories:

- As a **Supplier**, I want to submit declarations from my phone so that I can respond quickly
- As a **Compliance Officer**, I want to review and approve declarations on mobile so that I’m not tied to my desk

Capabilities:

- React Native or Flutter app
 - Push notifications for urgent actions
 - Camera integration for document upload
 - Offline mode with sync when online
 - Simplified UI for small screens
-

Feature 13: Multi-Language Support

Description: Localization for global suppliers (Spanish, French, Chinese, German, etc.).

User Stories:

- As a **Mexican Supplier**, I want to use the portal in Spanish so that I can understand instructions clearly
- As a **French Canadian Supplier**, I want to use the portal in French so that I comply with language preferences

Capabilities:

- i18n (internationalization) framework
- Translated UI and email templates

- Language selection in user profile
 - Auto-detect language from browser settings
-

Feature 14: Advanced HS Code Classification Tool

Description: AI-assisted HS code classification using product descriptions and images.

User Stories:

- As a **Compliance Officer**, I want the system to suggest HS codes based on product descriptions so that I ensure accuracy

Capabilities:

- NLP-based HS code suggestion from product descriptions
 - Integration with customs databases for code validation
 - Image recognition to assist in classification (e.g., upload product photo)
 - Regulatory change alerts for HS code reclassifications
-

Feature 15: Supplier Collaboration Portal (Enhanced)

Description: Full-featured supplier portal with chat, document sharing, and collaborative declaration workflows.

User Stories:

- As a **Supplier**, I want to chat with the compliance team about declaration questions so that I get clarifications quickly
- As a **Compliance Officer**, I want to share documents with suppliers in a secure space so that we collaborate effectively

Capabilities:

- Real-time chat/messaging
 - Shared document repository per supplier
 - Workflow: Draft declaration → Submit for review → Approve/Request changes
 - Supplier training resources and FAQs
-

Feature 16: Duty Drawback Management

Description: Track and manage duty drawback claims for duties paid on imported goods later exported.

User Stories:

- As a **Compliance Officer**, I want to track eligible drawback claims so that I can recover duties paid

Capabilities:

- Identify export transactions eligible for duty drawback
 - Calculate drawback amounts
 - Generate drawback claims documentation
 - Track claim submissions and refunds
-

Feature 17: Customs Ruling Management

Description: Repository of customs rulings and advance rulings for consistent classification and origin determination.

User Stories:

- As a **Compliance Officer**, I want to store and reference advance rulings so that I ensure consistency

Capabilities:

- Upload and categorize customs rulings
 - Link rulings to products
 - Search rulings by HS code, keyword, country
 - Alerts when rulings expire or are superseded
-

Feature 18: API Marketplace & Integrations

Description: Pre-built integrations with other ERP systems (Oracle, Microsoft Dynamics), TMS, WMS, and customs brokers.

User Stories:

- As a **Compliance Officer**, I want to connect to our Oracle ERP so that I can use the system without SAP

Capabilities:

- Pre-built connectors for Oracle, Microsoft Dynamics, NetSuite
 - Integration with customs brokerage systems
 - Webhook support for real-time event notifications
 - Public API for custom integrations
-

Feature 19: Blockchain-Based Certificate Verification

Description: Immutable, blockchain-based certificates for enhanced security and verifiability.

User Stories:

- As an **Importer**, I want to verify a certificate's authenticity via blockchain so that I trust its validity

Capabilities:

- Store certificate hash on blockchain (e.g., Ethereum, Hyperledger)
 - Public verification portal: Enter certificate number → verify authenticity
 - Tamper-proof audit trail
 - Integration with customs authorities' verification systems
-

Feature 20: Advanced Reporting & Business Intelligence

Description: Customizable reports and BI dashboards using tools like Tableau, Power BI integration.

User Stories:

- As an **Executive**, I want to create custom reports in Power BI so that I can present FTA ROI to stakeholders

Capabilities:

- Export data to Power BI, Tableau, Looker
- Pre-built BI templates for common reports
- Scheduled report generation and email delivery
- Custom report builder (drag-and-drop)

4. UI/UX Design Guidelines

4.1 Design Principles

1. Simplicity & Clarity

- Trade compliance is complex; the UI should simplify, not complicate
- Use plain language, avoid jargon where possible
- Progressive disclosure: Show basic options first, advanced options on demand

2. Efficiency

- Minimize clicks to complete common tasks
- Keyboard shortcuts for power users
- Bulk operations for managing multiple items

3. Consistency

- Consistent navigation, button styles, color coding across all screens
- Design system with reusable components (Material-UI or custom)

4. Accessibility

- WCAG 2.1 AA compliance minimum
- High contrast mode support
- Screen reader compatible
- Keyboard navigation for all functions

5. Responsiveness

- Mobile-first design for supplier portal
- Desktop-optimized for compliance officer workflows
- Tablet-friendly dashboards

6. Trust & Transparency

- Clear indication of calculation methods and data sources
- Visible audit trails
- Confirmation dialogs for critical actions (delete, submit)

4.2 Color Scheme & Branding

Primary Colors:

- **Primary Blue:** #1976D2 (trust, professionalism) - for primary actions, headers
- **Secondary Teal:** #00897B (compliance, accuracy) - for accents, success states
- **Dark Gray:** #263238 - for text and navigation
- **Light Gray:** #F5F5F5 - for backgrounds and borders

Status Colors:

- **Success Green:** #4CAF50 - for qualifying origin, completed tasks
- **Warning Orange:** #FF9800 - for pending actions, missing declarations

- **Error Red:** #F44336 - for non-qualifying origin, errors
- **Info Blue:** #2196F3 - for informational messages

Typography:

- **Primary Font:** Inter or Roboto (clean, modern, highly readable)
- **Headings:** Semi-bold, 24px (H1), 20px (H2), 16px (H3)
- **Body Text:** Regular, 14px
- **Small Text:** 12px for labels and metadata

4.3 Detailed Wireframe Descriptions

Wireframe 1: Login Screen

Layout:

[Company Logo]

FTA Manager Login

Email Address
[]

Password
[]

[] Remember me [Forgot password?](#)

[Login Button]

Don't have an account? [Contact your admin](#)

Elements:

- Centered single-column layout
- Logo at top (customizable per tenant)
- Email and password fields with clear labels
- “Remember me” checkbox and “Forgot password?” link
- Prominent “Login” button (primary blue color)
- Footer text for new users
- Error messages appear above form in red box if login fails

Behavior:

- Enter key submits form
- Password field shows/hide toggle (eye icon)
- Loading spinner on button after submit
- Redirect to dashboard on success

Wireframe 2: Supplier Portal - Declaration Submission

Layout:

FTA Manager Supplier Portal [Logout] [Help]

Step 1 of 3: Product Information

Declaration Request from: ABC Manufacturing Inc.
Due Date: January 30, 2026

Part Number(s) *
[PART-001] ▼
☐ Add multiple parts (bulk entry)

Free Trade Agreement *
☐ USMCA ☐ CFTA ☐ EU-Canada ☐ Other: [_____]

Country of Origin *
[United States] ▼

Origin Criterion (USMCA) *
☐ A - Wholly obtained or produced
☐ B - Produced from originating materials
☐ C - Satisfies specific rule of origin
☐ D - Meets regional value content requirement

[Back] [Save Draft] [Next >]

Elements:

- **Header:** Company logo, “Supplier Portal” title, Logout and Help buttons
- **Progress Indicator:** 3-step wizard (Product Info → Details → Review)
- **Request Context:** Shows who requested and due date (highlighted if urgent)
- **Form Fields:**
 - Part number dropdown (pre-populated from request) or multi-select for bulk
 - FTA radio buttons with “Other” text input
 - Country of origin dropdown (all countries, searchable)
 - Origin criterion radio buttons with descriptions
- **Actions:**
 - Back button (disabled on step 1)
 - Save Draft (saves to supplier’s account, can resume later)
 - Next button (validates required fields before proceeding)

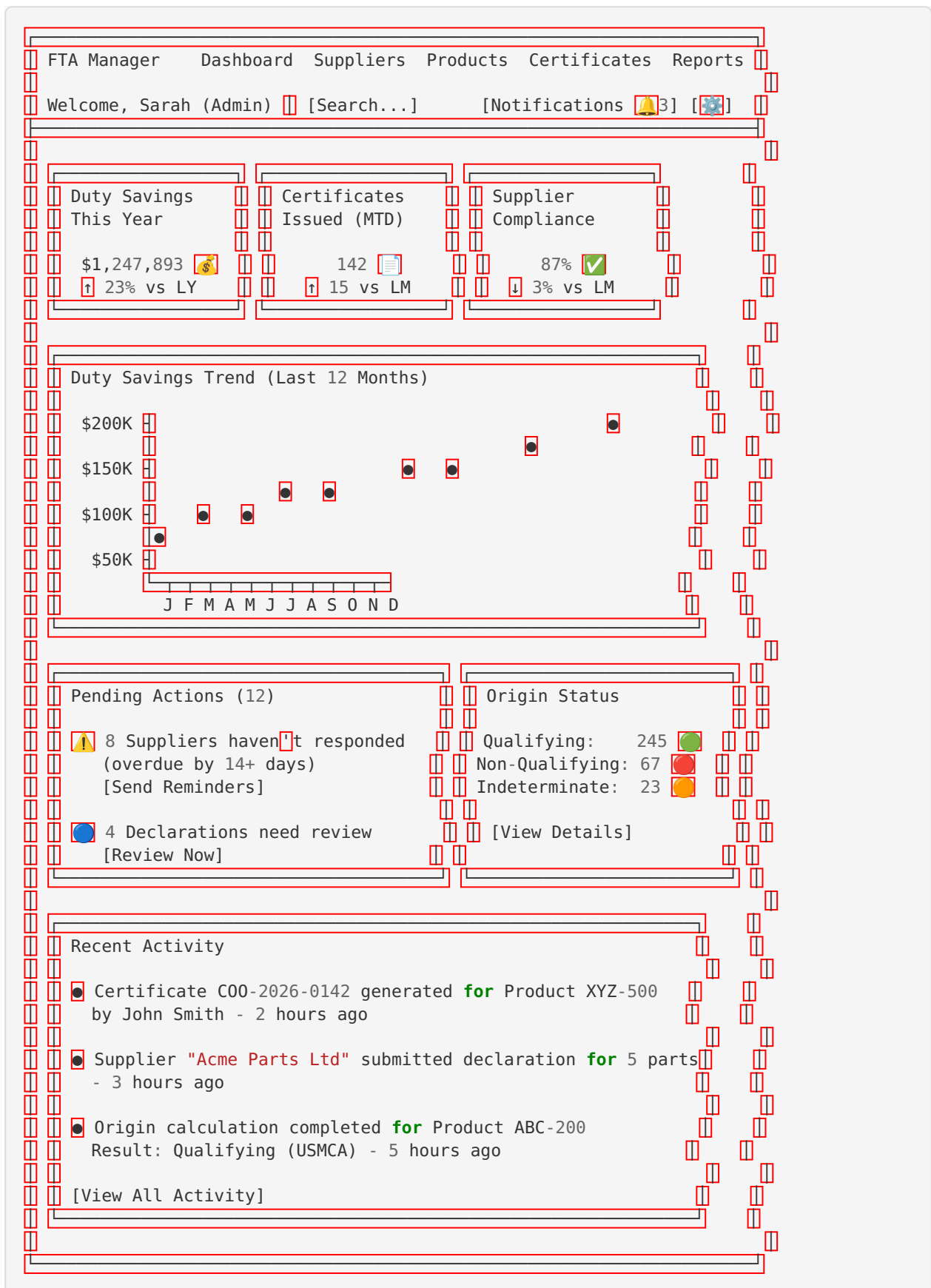
Behavior:

- Required fields marked with asterisk (*)

- Inline validation: Red border + error message if field invalid
 - “Add multiple parts” checkbox expands to multi-select interface
 - Tooltips on hover for origin criterion descriptions
 - Auto-save draft every 60 seconds
-

Wireframe 3: Admin Dashboard

Layout:



Elements:

- **Top Navigation:** Logo, main menu (Dashboard, Suppliers, Products, Certificates, Reports)
- **User Bar:** Welcome message, search box, notifications icon with count, settings gear
- **KPI Cards (3):**

- Duty Savings This Year: Large number with trend indicator and percentage change
- Certificates Issued (MTD): Count with comparison to last month
- Supplier Compliance: Percentage with trend
- **Line Chart:** Duty savings trend over 12 months (interactive, hover for details)
- **Pending Actions Widget:** Alerts with action buttons
- **Origin Status Widget:** Pie chart or simple counts with color coding
- **Activity Feed:** Chronological list of recent system events

Behavior:

- KPI cards clickable to drill down into details
 - Chart interactive: Hover shows exact values, click data point to filter
 - Pending actions: Click buttons to navigate to relevant screens
 - Activity feed: Click items to view full details
 - Dashboard refreshes every 60 seconds for real-time updates
-

Wireframe 4: Compliance Officer Workspace - Origin Calculation**Layout:**

FTA Manager > Products > XYZ-500 > Calculate Origin

Calculate Origin for Product: XYZ-500 Industrial Pump Assembly

Product Details

HS Code: 8413.70.20

Unit Cost: \$2,450.00

Description: Centrifugal pump assembly with motor

Calculation Settings

FTA: [USMCA] Method: [Transaction Value]

BOM Depth: [5] levels [Advanced Settings]

Bill of Materials (5 levels, 47 components)

XYZ-500 Industrial Pump Assembly

Motor Assembly (MA-100)

Rotor (R-200) - Mexico

Stator (S-300) - China

Housing (H-400) - USA

Impeller (IMP-250) - Canada

Casing (CS-180) - USA

Fasteners (FKIT-50) - Various

100%

45%

15%

20%

10%

25%

20%

10%

Legend:

Originating

Non-originating

Missing

Warning: 2 components missing origin declarations. Calculation will assume non-originating. [Request Declarations]

[Cancel]

[Run Calculation]

Elements:

- **Breadcrumb Navigation:** Shows path to current screen
- **Product Summary:** Key details (HS code, cost, description)
- **Calculation Settings:**
 - FTA dropdown (USMCA, CFTA, EU-Canada, etc.)
 - Calculation method dropdown (Transaction Value, Net Cost)
 - BOM depth selector (configurable, max from tenant settings)
 - Advanced settings link (for custom parameters)
- **BOM Tree View:**
 - Expandable/collapsible tree structure
 - Shows component name, origin country, percentage of total cost
 - Color-coded status icons (green, red, orange)
 - Click component to view details
- **Warning Banner:** If missing declarations, shows count and offers action

- Action Buttons:

- Cancel: Returns to product details
- Run Calculation: Starts calculation process

Behavior:

- BOM tree loads on page load (async, shows loading spinner)
 - Click tree nodes to expand/collapse
 - Hover over components to see detailed info tooltip
 - "Request Declarations" button opens modal to select suppliers and send requests
 - "Run Calculation" validates settings, then shows progress modal
 - After calculation, redirects to results page
-

Wireframe 5: BOM Analysis View (Calculation Results)**Layout:**

FTA Manager > Products > XYZ-500 > Origin Calculation Results	
Origin Calculation Results for XYZ-500	
Calculated: January 15, 2026 at 2:45 PM by Sarah Johnson	
RESULT: QUALIFIES <input checked="" type="checkbox"/>	
Under USMCA using Transaction Value Method	
Regional Value Content: 67.5% (Threshold: 60%)	
Calculation Breakdown	
Transaction Value:	\$2,450.00
Originating Materials:	\$1,654.13
Non-Originating Materials:	\$ 795.87
RVC Percentage:	67.5%
Formula: ((TV - VNM) / TV) <input checked="" type="checkbox"/> 100	
((2450 - 795.87) / 2450) <input checked="" type="checkbox"/> 100 = 67.5%	
Component Origin Summary	
Origin Distribution	
<input checked="" type="checkbox"/> Originating: 67.5%	
<input checked="" type="checkbox"/> Non-Originating: 32.5%	
[Pie Chart Visualization]	
Top Originating Components:	
1. Motor Assembly (MA-100) - USA - \$1,102.50	
2. Impeller (IMP-250) - Canada - \$612.50	
3. Casing (CS-180) - USA - \$490.00	
Non-Originating Components:	
1. Stator (S-300) - China - \$490.00 <input checked="" type="checkbox"/> (assumed)	
2. Fasteners (FKIT-50) - Germany - \$245.00	
[Download Report PDF] [Generate Certificate] [Recalculate]	

Elements:

- **Result Banner:** Large, clear indication (green for qualifying, red for non-qualifying)
- **Calculation Details:** Shows FTA, method, RVC percentage vs. threshold
- **Breakdown Section:**
 - Transaction value and component values
 - RVC formula and calculation shown step-by-step

- **Pie Chart:** Visual representation of originating vs. non-originating value
- **Component Lists:**
 - Top originating components (helps identify value drivers)
 - Non-originating components (helps identify opportunities for sourcing changes)
- **Action Buttons:**
 - Download Report PDF: Generates detailed PDF for audit/sharing
 - Generate Certificate: Navigates to certificate creation (only if qualifies)
 - Recalculate: Returns to calculation settings

Behavior:

- Result banner color-coded (green/red) with large font
 - Pie chart interactive: Hover shows exact percentages
 - Component lists expandable (click to see full BOM breakdown)
 - If non-qualifying, "Generate Certificate" button disabled with tooltip explanation
 - "What-if Analysis" link (Phase 2) to simulate sourcing changes
-

Wireframe 6: Certificate Management Interface**Layout:**

FTA Manager > Certificates

Certificates of Origin

[+ Generate Certificate]

Filters

FTA: [All ▼] Status: [All ▼] Date Range: [Last 30 Days ▼]

[Apply Filters] [Clear]

Certificate #	Product	FTA	Type	Date	
C00-2026-0142	XYZ-500	USMCA	Single	Jan 15 '26	
C00-2026-0141	ABC-200	CFTA	Blanket	Jan 14 '26	
C00-2026-0140	DEF-300	USMCA	Single	Jan 13 '26	
C00-2026-0139	GHI-450	EU	Single	Jan 12 '26	
...	

Showing 1-10 of 142 certificates

[< Previous] [Next >]

[Export to Excel]

[Bulk Download PDFs]

Click on row  Opens modal:

Certificate Details [X Close]

Certificate Number: C00-2026-0142

Status: Active

Product: XYZ-500 Industrial Pump Assembly

HS Code: 8413.70.20

FTA: USMCA

Origin Criterion: B - Produced from originating materials

Exporter: ABC Manufacturing Inc.

Importer: Customer Corp (USA)

Certificate Type: Single Shipment

Issue Date: January 15, 2026

Valid Until: January 14, 2027

Certifier: Sarah Johnson, Compliance Officer

Signature: [Electronic signature image]

[Download PDF] [Email to Customer] [Revoke Certificate]

Elements:

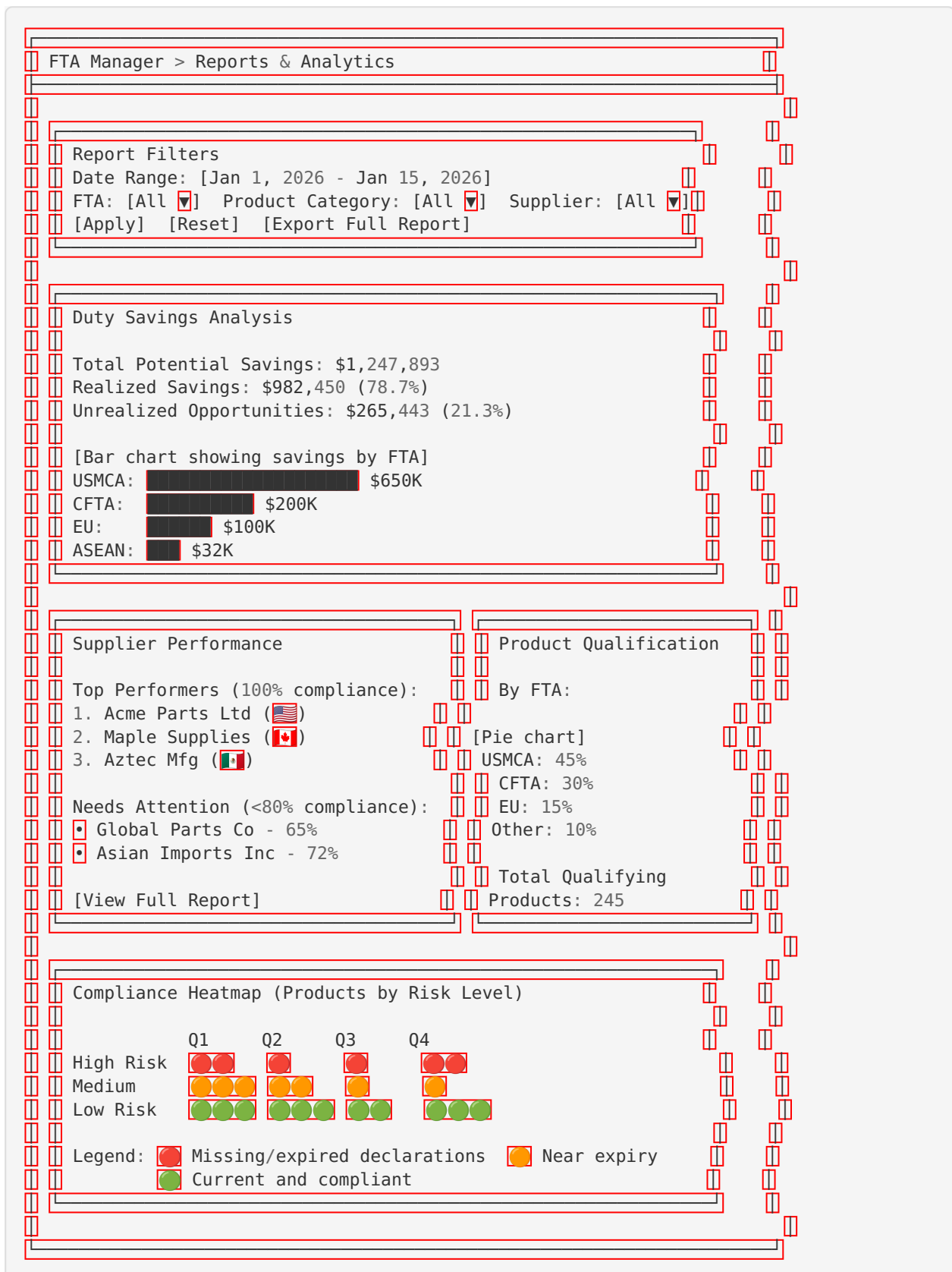
- **Header:** Title and prominent “Generate Certificate” button
- **Filters Section:**
 - FTA dropdown (All, USMCA, CFTA, EU, etc.)

- Status dropdown (All, Active, Revoked, Expired)
- Date range picker
- Apply/Clear buttons
- **Data Table:**
- Columns: Certificate #, Product, FTA, Type, Date, Actions (: menu)
- Sortable columns (click header to sort)
- Row actions: View, Download, Email, Revoke
- **Pagination:** Shows count and navigation buttons
- **Bulk Actions:** Export to Excel, bulk download PDFs
- **Detail Modal:** Opens when clicking row, shows all certificate info

Behavior:

- Table loads with latest certificates first
- Filters update table in real-time
- Click row to open detail modal
- Actions menu (:) for quick actions: Download PDF, Email, View Details, Revoke
- “Generate Certificate” button opens wizard (select product → enter details → review → generate)
- Search box (not shown) for quick find by certificate number or product

Wireframe 7: Reporting/Analytics Dashboard**Layout:**



Elements:

- **Filters:** Date range, FTA, product category, supplier (with apply/reset)
- **Duty Savings Analysis:**
 - Total, realized, unrealized savings with percentages
 - Horizontal bar chart showing savings by FTA
- **Supplier Performance Widget:**

- Top performers list with country flags
- “Needs attention” list with compliance percentages
- Link to full report
- **Product Qualification Widget:**
- Pie chart showing product distribution by FTA
- Total count of qualifying products
- **Compliance Heatmap:**
- Visual grid showing risk levels over time
- Color-coded (red/orange/green) for quick assessment

Behavior:

- All charts interactive: Hover for details, click to drill down
- Filters apply to all widgets simultaneously
- “Export Full Report” generates comprehensive PDF or Excel
- Refresh button to update with latest data
- Customizable widgets (Phase 2): Drag-and-drop to reorder or hide/show

4.4 Navigation Structure

Primary Navigation (Top Menu):

```

FTA Manager
├─ Dashboard (default landing page)
├─ Suppliers
│   ├── All Suppliers (list view)
│   ├── Add Supplier
│   ├── Import from SAP
│   └─ Declaration Requests
├─ Products
│   ├── All Products (list view)
│   ├── Add Product
│   ├── Import from SAP
│   └─ BOM Management
├─ Declarations
│   ├── All Declarations (list view)
│   ├── Pending Review
│   └─ Request Declarations
├─ Origin Calculations
│   ├── All Calculations (list view)
│   └─ New Calculation
├─ Certificates
│   ├── All Certificates (list view)
│   └─ Generate Certificate
├─ Reports & Analytics
│   ├── Duty Savings Report
│   ├── Supplier Performance
│   ├── Compliance Dashboard
│   └─ Custom Reports (Phase 2)
└─ Settings
    ├── Company Profile
    ├── User Management
    ├── FTA Rules Configuration
    ├── Integration Settings (SAP, APIs)
    └─ Notification Preferences
  
```

User Profile Menu (Top Right):

[User Avatar] ▾
☐ My Profile
☐ Change Password
☐ Notification Settings
☐ Help & Documentation
☐ Logout

Breadcrumb Navigation:

Every page shows breadcrumb for easy navigation back:

FTA Manager > Products > XYZ-500 > Calculate Origin

4.5 Responsive Design Considerations

Mobile (320px - 767px):

- **Supplier Portal Optimized:** Simple, single-column forms
- **Hamburger Menu:** Collapsible navigation
- **Stacked Cards:** KPI cards stack vertically
- **Touch-Friendly:** Minimum 44px touch targets
- **Simplified Tables:** Card-based layouts instead of tables

Tablet (768px - 1023px):

- **Two-Column Layouts:** Where appropriate (e.g., dashboard widgets in 2 columns)
- **Collapsible Sidebar:** For navigation
- **Responsive Charts:** Adjust to screen width

Desktop (1024px+):

- **Multi-Column Layouts:** Full dashboard with 3-4 columns
- **Sidebar Navigation:** Always visible
- **Data Tables:** Full-featured with all columns

4.6 Accessibility Requirements

WCAG 2.1 AA Compliance:

- **Color Contrast:** Minimum 4.5:1 for normal text, 3:1 for large text
- **Keyboard Navigation:** All functions accessible via keyboard (Tab, Enter, Arrow keys)
- **Screen Reader Support:** Semantic HTML, ARIA labels, alt text for images
- **Focus Indicators:** Visible focus outlines on interactive elements
- **Form Labels:** Properly associated labels for all inputs
- **Error Identification:** Clear error messages with suggestions for correction
- **Resizable Text:** Support up to 200% zoom without loss of functionality
- **Skip Links:** "Skip to main content" for screen reader users

5. Database Schema

5.1 Entity-Relationship Overview

The database is designed with multi-tenancy at its core, with all major tables including a `tenant_id` foreign key for data isolation.

Core Entities:

- **Tenants:** Companies using the platform
- **Users:** Admins, compliance officers, suppliers, auditors
- **Suppliers:** External parties providing origin declarations
- **Products:** Finished goods and components
- **BOM Items:** Bill of materials relationships
- **Declarations:** Supplier origin declarations
- **Origin Calculations:** Results of origin determination
- **Certificates:** Certificates of origin generated
- **FTA Rules:** Free trade agreement rules and thresholds
- **Audit Logs:** Immutable record of all changes

Relationships:

- **Tenant ↔ Users:** 1:N (one tenant has many users)
- **Tenant ↔ Suppliers:** 1:N (one tenant has many suppliers)
- **Tenant ↔ Products:** 1:N (one tenant has many products)
- **Product ↔ BOM Items:** 1:N (one product has many components, hierarchical)
- **Supplier ↔ Declarations:** 1:N (one supplier provides many declarations)
- **Product ↔ Declarations:** 1:N (one product can have declarations from multiple suppliers)
- **Product ↔ Origin Calculations:** 1:N (one product can have multiple calculations over time)
- **Product ↔ Certificates:** 1:N (one product can have multiple certificates)

5.2 Detailed Table Definitions

Table: tenants

Multi-tenant master table for companies using the platform.

```
CREATE TABLE tenants (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  company_name VARCHAR(255) NOT NULL,
  subdomain VARCHAR(100) UNIQUE,
  country_code CHAR(2) NOT NULL,
  address TEXT,
  tax_id VARCHAR(100),
  logo_url TEXT,
  subscription_tier VARCHAR(50) DEFAULT 'basic', -- basic, professional, enterprise
  status VARCHAR(20) DEFAULT 'active', -- active, suspended, trial
  max_users INT DEFAULT 10,
  max_suppliers INT DEFAULT 500,
  trial_ends_at TIMESTAMP,
  created_at TIMESTAMP DEFAULT NOW(),
  updated_at TIMESTAMP DEFAULT NOW()
);

CREATE INDEX idx_tenants_subdomain ON tenants(subdomain);
CREATE INDEX idx_tenants_status ON tenants(status);
```

Table: users

All system users (admins, compliance officers, suppliers, auditors).

```

CREATE TABLE users (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  tenant_id UUID NOT NULL REFERENCES tenants(id) ON DELETE CASCADE,
  email VARCHAR(255) NOT NULL,
  password_hash VARCHAR(255) NOT NULL,
  first_name VARCHAR(100),
  last_name VARCHAR(100),
  role VARCHAR(50) NOT NULL, -- admin, compliance_officer, supplier, auditor
  supplier_id UUID REFERENCES suppliers(id) ON DELETE SET NULL, -- if role=supplier
  is_active BOOLEAN DEFAULT TRUE,
  last_login_at TIMESTAMP,
  failed_login_attempts INT DEFAULT 0,
  locked_until TIMESTAMP,
  email_verified BOOLEAN DEFAULT FALSE,
  mfa_enabled BOOLEAN DEFAULT FALSE,
  mfa_secret VARCHAR(100),
  created_at TIMESTAMP DEFAULT NOW(),
  updated_at TIMESTAMP DEFAULT NOW(),
  CONSTRAINT unique_email_per_tenant UNIQUE(tenant_id, email)
);

CREATE INDEX idx_users_tenant ON users(tenant_id);
CREATE INDEX idx_users_email ON users(email);
CREATE INDEX idx_users_role ON users(tenant_id, role);
CREATE INDEX idx_users_supplier ON users(supplier_id);

```

Table: suppliers

Supplier companies providing origin declarations.

```

CREATE TABLE suppliers (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  tenant_id UUID NOT NULL REFERENCES tenants(id) ON DELETE CASCADE,
  company_name VARCHAR(255) NOT NULL,
  contact_person VARCHAR(255),
  email VARCHAR(255) NOT NULL,
  phone VARCHAR(50),
  country_code CHAR(2) NOT NULL,
  address TEXT,
  city VARCHAR(100),
  state_province VARCHAR(100),
  postal_code VARCHAR(20),
  tax_id VARCHAR(100),
  duns_number VARCHAR(20),
  status VARCHAR(20) DEFAULT 'active', -- active, inactive, pending
  last_declaration_date TIMESTAMP,
  total_declarations_submitted INT DEFAULT 0,
  compliance_rate DECIMAL(5, 2) DEFAULT 0.00, -- percentage
  notes TEXT,
  created_at TIMESTAMP DEFAULT NOW(),
  updated_at TIMESTAMP DEFAULT NOW(),
  created_by UUID REFERENCES users(id),
  CONSTRAINT unique_supplier_per_tenant UNIQUE(tenant_id, company_name, country_code)
);

CREATE INDEX idx_suppliers_tenant ON suppliers(tenant_id);
CREATE INDEX idx_suppliers_status ON suppliers(tenant_id, status);
CREATE INDEX idx_suppliers_country ON suppliers(tenant_id, country_code);

```

Table: products

Finished goods and components.

```
CREATE TABLE products (  
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),  
  tenant_id UUID NOT NULL REFERENCES tenants(id) ON DELETE CASCADE,  
  product_code VARCHAR(100) NOT NULL,  
  description TEXT NOT NULL,  
  hs_code VARCHAR(10) NOT NULL,  
  category VARCHAR(100),  
  unit_of_measure VARCHAR(20),  
  unit_cost DECIMAL(15, 2),  
  is_finished_good BOOLEAN DEFAULT FALSE,  
  is_active BOOLEAN DEFAULT TRUE,  
  image_url TEXT,  
  notes TEXT,  
  sap_material_number VARCHAR(100), -- for SAP integration  
  created_at TIMESTAMP DEFAULT NOW(),  
  updated_at TIMESTAMP DEFAULT NOW(),  
  created_by UUID REFERENCES users(id),  
  CONSTRAINT unique_product_code_per_tenant UNIQUE(tenant_id, product_code)  
);  
  
CREATE INDEX idx_products_tenant ON products(tenant_id);  
CREATE INDEX idx_products_hs_code ON products(tenant_id, hs_code);  
CREATE INDEX idx_products_category ON products(tenant_id, category);  
CREATE INDEX idx_products_finished_good ON products(tenant_id, is_finished_good);  
CREATE INDEX idx_products_sap ON products(tenant_id, sap_material_number);
```

Table: bom_items

Bill of materials hierarchical relationships.

```

CREATE TABLE bom_items (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  tenant_id UUID NOT NULL REFERENCES tenants(id) ON DELETE CASCADE,
  parent_product_id UUID NOT NULL REFERENCES products(id) ON DELETE CASCADE,
  child_product_id UUID NOT NULL REFERENCES products(id),
  quantity DECIMAL(15, 4) NOT NULL,
  unit_cost DECIMAL(15, 2),
  supplier_id UUID REFERENCES suppliers(id) ON DELETE SET NULL,
  level INT DEFAULT 1, -- depth in BOM tree (1 = direct child)
  percentage_of_total DECIMAL(5, 2), -- percentage of total finished good cost
  notes TEXT,
  created_at TIMESTAMP DEFAULT NOW(),
  updated_at TIMESTAMP DEFAULT NOW(),
  CONSTRAINT no_self_reference CHECK (parent_product_id != child_product_id),
  CONSTRAINT unique_bom_item UNIQUE(tenant_id, parent_product_id, child_product_id)
);

CREATE INDEX idx_bom_parent ON bom_items(tenant_id, parent_product_id);
CREATE INDEX idx_bom_child ON bom_items(tenant_id, child_product_id);
CREATE INDEX idx_bom_supplier ON bom_items(supplier_id);

-- Function to prevent circular BOM references (trigger-based check)
-- Implementation: Recursive CTE to detect cycles before insert/update

```

Table: declaration_requests

Requests sent to suppliers for origin declarations.

```

CREATE TABLE declaration_requests (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  tenant_id UUID NOT NULL REFERENCES tenants(id) ON DELETE CASCADE,
  supplier_id UUID NOT NULL REFERENCES suppliers(id) ON DELETE CASCADE,
  product_id UUID NOT NULL REFERENCES products(id) ON DELETE CASCADE,
  fta_code VARCHAR(20) NOT NULL, -- USMCA, CFTA, EU, ASEAN, CPTPP
  requested_by UUID REFERENCES users(id),
  requested_at TIMESTAMP DEFAULT NOW(),
  due_date DATE,
  status VARCHAR(20) DEFAULT 'pending', -- pending, submitted, overdue, cancelled
  reminder_sent_at TIMESTAMP,
  reminder_count INT DEFAULT 0,
  notes TEXT,
  secure_token VARCHAR(255) UNIQUE, -- for supplier portal access
  token_expires_at TIMESTAMP,
  created_at TIMESTAMP DEFAULT NOW(),
  updated_at TIMESTAMP DEFAULT NOW()
);

CREATE INDEX idx_decl_requests_tenant ON declaration_requests(tenant_id);
CREATE INDEX idx_decl_requests_supplier ON declaration_requests(supplier_id);
CREATE INDEX idx_decl_requests_status ON declaration_requests(tenant_id, status);
CREATE INDEX idx_decl_requests_token ON declaration_requests(secure_token);

```

Table: declarations

Supplier origin declarations.

```

CREATE TABLE declarations (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  tenant_id UUID NOT NULL REFERENCES tenants(id) ON DELETE CASCADE,
  declaration_request_id UUID REFERENCES declaration_requests(id) ON DELETE SET NULL,
  supplier_id UUID NOT NULL REFERENCES suppliers(id) ON DELETE CASCADE,
  product_id UUID NOT NULL REFERENCES products(id) ON DELETE CASCADE,
  fta_code VARCHAR(20) NOT NULL,
  country_of_origin CHAR(2) NOT NULL,
  origin_criterion VARCHAR(10), -- A, B, C, D for USMCA; varies by FTA
  producer_name VARCHAR(255),
  blanket_period_start DATE,
  blanket_period_end DATE,
  rvc_percentage DECIMAL(5, 2),
  hs_code VARCHAR(10),
  certifier_name VARCHAR(255) NOT NULL,
  certifier_title VARCHAR(100),
  electronic_signature TEXT, -- typed name or digital signature
  declaration_date DATE NOT NULL,
  comments TEXT,
  status VARCHAR(20) DEFAULT 'submitted', -- submitted, under_review, approved, rejected
  reviewed_by UUID REFERENCES users(id),
  reviewed_at TIMESTAMP,
  submitted_at TIMESTAMP DEFAULT NOW(),
  created_at TIMESTAMP DEFAULT NOW(),
  updated_at TIMESTAMP DEFAULT NOW()
);

CREATE INDEX idx_declarations_tenant ON declarations(tenant_id);
CREATE INDEX idx_declarations_supplier ON declarations(supplier_id);
CREATE INDEX idx_declarations_product ON declarations(product_id);
CREATE INDEX idx_declarations_fta ON declarations(tenant_id, fta_code);
CREATE INDEX idx_declarations_status ON declarations(tenant_id, status);
CREATE INDEX idx_declarations_blanket_period ON declarations(blanket_period_start, blanket_period_end);

```

Table: declaration_documents

Supporting documents uploaded with declarations (PDFs, Excel, etc.).

```

CREATE TABLE declaration_documents (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  declaration_id UUID NOT NULL REFERENCES declarations(id) ON DELETE CASCADE,
  file_name VARCHAR(255) NOT NULL,
  file_size_bytes BIGINT,
  file_type VARCHAR(100),
  s3_bucket VARCHAR(255),
  s3_key VARCHAR(500) NOT NULL,
  virus_scan_status VARCHAR(20) DEFAULT 'pending', -- pending, clean, infected
  uploaded_at TIMESTAMP DEFAULT NOW()
);

CREATE INDEX idx_decl_docs_declaration ON declaration_documents(declaration_id);

```

Table: origin_calculations

Origin calculation results.

```
CREATE TABLE origin_calculations (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  tenant_id UUID NOT NULL REFERENCES tenants(id) ON DELETE CASCADE,
  product_id UUID NOT NULL REFERENCES products(id) ON DELETE CASCADE,
  fta_code VARCHAR(20) NOT NULL,
  calculation_method VARCHAR(50) NOT NULL, -- transaction_value, net_cost
  total_value DECIMAL(15, 2),
  originating_value DECIMAL(15, 2),
  non_originating_value DECIMAL(15, 2),
  rvc_percentage DECIMAL(5, 2),
  required_rvc_threshold DECIMAL(5, 2),
  qualifies BOOLEAN,
  qualification_reason TEXT, -- why it qualifies or doesn't
  missing_declarations JSONB, -- array of component IDs with missing declarations
  calculation_details JSONB, -- full breakdown (component values, origins, formulas)
  tariff_shift_applied BOOLEAN DEFAULT FALSE,
  de_minimis_applied BOOLEAN DEFAULT FALSE,
  calculated_at TIMESTAMP DEFAULT NOW(),
  calculated_by UUID REFERENCES users(id),
  is_current BOOLEAN DEFAULT TRUE, -- flag for latest calculation
  notes TEXT,
  created_at TIMESTAMP DEFAULT NOW()
);

CREATE INDEX idx_origin_calcs_tenant ON origin_calculations(tenant_id);
CREATE INDEX idx_origin_calcs_product ON origin_calculations(product_id);
CREATE INDEX idx_origin_calcs_fta ON origin_calculations(tenant_id, fta_code);
CREATE INDEX idx_origin_calcs_current ON origin_calculations(product_id, is_current);
```

Table: certificates

Certificates of origin generated.

```

CREATE TABLE certificates (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  tenant_id UUID NOT NULL REFERENCES tenants(id) ON DELETE CASCADE,
  certificate_number VARCHAR(100) NOT NULL,
  product_id UUID NOT NULL REFERENCES products(id) ON DELETE CASCADE,
  origin_calculation_id UUID REFERENCES origin_calculations(id) ON DELETE SET NULL,
  fta_code VARCHAR(20) NOT NULL,
  certificate_type VARCHAR(50) NOT NULL, -- single_shipment, blanket
  exporter_name VARCHAR(255) NOT NULL,
  exporter_address TEXT,
  exporter_tax_id VARCHAR(100),
  producer_name VARCHAR(255),
  producer_address TEXT,
  producer_tax_id VARCHAR(100),
  importer_name VARCHAR(255),
  importer_address TEXT,
  importer_tax_id VARCHAR(100),
  product_description TEXT NOT NULL,
  hs_code VARCHAR(10) NOT NULL,
  origin_criterion VARCHAR(10) NOT NULL,
  blanket_period_start DATE,
  blanket_period_end DATE,
  certifier_name VARCHAR(255) NOT NULL,
  certifier_title VARCHAR(100),
  electronic_signature TEXT,
  issue_date DATE NOT NULL,
  valid_until DATE,
  status VARCHAR(20) DEFAULT 'active', -- active, revoked, expired
  revoked_at TIMESTAMP,
  revoked_by UUID REFERENCES users(id),
  revocation_reason TEXT,
  pdf_s3_bucket VARCHAR(255),
  pdf_s3_key VARCHAR(500),
  created_at TIMESTAMP DEFAULT NOW(),
  created_by UUID REFERENCES users(id),
  CONSTRAINT unique_certificate_number_per_tenant UNIQUE(tenant_id, certificate_number)
);

CREATE INDEX idx_certificates_tenant ON certificates(tenant_id);
CREATE INDEX idx_certificates_product ON certificates(product_id);
CREATE INDEX idx_certificates_fta ON certificates(tenant_id, fta_code);
CREATE INDEX idx_certificates_status ON certificates(tenant_id, status);
CREATE INDEX idx_certificates_number ON certificates(certificate_number);

```

Table: fta_rules

FTA-specific rules and thresholds.

```

CREATE TABLE fta_rules (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  fta_code VARCHAR(20) NOT NULL UNIQUE, -- USMCA, CFTA, EU_CANADA, ASEAN, CPTPP
  fta_name VARCHAR(255) NOT NULL,
  member_countries JSONB, -- array of country codes
  default_rvc_threshold DECIMAL(5, 2), -- percentage
  calculation_methods JSONB, -- array: ["transaction_value", "net_cost"]
  origin_criteria JSONB, -- array of valid origin criteria (A, B, C, D, etc.)
  tariff_shift_rules JSONB, -- complex rules per HS chapter/heading
  de_minimis_percentage DECIMAL(5, 2), -- e.g., 10% for USMCA
  max_blanket_period_days INT DEFAULT 365,
  effective_date DATE,
  notes TEXT,
  created_at TIMESTAMP DEFAULT NOW(),
  updated_at TIMESTAMP DEFAULT NOW()
);

-- Pre-populate with major FTAs
INSERT INTO fta_rules (fta_code, fta_name, member_countries, default_rvc_threshold, calculation_methods, origin_criteria, de_minimis_percentage) VALUES
('USMCA', 'United States-Mexico-Canada Agreement', '["US", "CA", "MX"]', 60.00, ['transaction_value', 'net_cost'], ['A', 'B', 'C', 'D'], 10.00),
('CFTA', 'Canada Free Trade Agreement', '["CA"]', 50.00, ['transaction_value'], ['A', 'B', 'C'], 10.00),
('EU_CANADA', 'EU-Canada CETA', '["CA", "EU"]', 50.00, ['transaction_value'], ['A', 'B', 'C'], 10.00),
('ASEAN', 'ASEAN Trade in Goods Agreement', '["BN", "KH", "ID", "LA", "MY", "MM", "PH", "SG", "TH", "VN"]', 40.00, ['transaction_value'], ['A', 'B'], 10.00),
('CPTPP', 'Comprehensive and Progressive Trans-Pacific Partnership', '["AU", "BN", "CA", "CL", "JP", "MY", "MX", "NZ", "PE", "SG", "VN"]', 45.00, ['transaction_value', 'net_cost'], ['A', 'B', 'C', 'D'], 10.00);

```

Table: product_specific_rules

Product-specific rules of origin (e.g., USMCA Annex 4-B).

```

CREATE TABLE product_specific_rules (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  fta_code VARCHAR(20) NOT NULL REFERENCES fta_rules(fta_code),
  hs_code_from VARCHAR(10) NOT NULL, -- range start
  hs_code_to VARCHAR(10), -- range end (if applicable)
  rule_text TEXT NOT NULL, -- tariff shift rule, e.g., "A change to heading 8413 from any other chapter"
  rvc_threshold DECIMAL(5, 2), -- if different from default
  additional_requirements TEXT,
  created_at TIMESTAMP DEFAULT NOW(),
  updated_at TIMESTAMP DEFAULT NOW()
);

CREATE INDEX idx_product_rules_fta ON product_specific_rules(fta_code);
CREATE INDEX idx_product_rules_hs ON product_specific_rules(hs_code_from, hs_code_to);

```

Table: tenant_settings

Tenant-specific configuration (BOM depth, default FTA, etc.).

```
CREATE TABLE tenant_settings (
  tenant_id UUID PRIMARY KEY REFERENCES tenants(id) ON DELETE CASCADE,
  max_bom_depth INT DEFAULT 5,
  default_fta VARCHAR(20) DEFAULT 'USMCA',
  default_calculation_method VARCHAR(50) DEFAULT 'transaction_value',
  auto_reminder_days JSONB DEFAULT '[7, 14, 30]', -- days after request to send re-
minders
  notification_preferences JSONB,
  branding JSONB, -- logo, colors, etc.
  sap_integration_enabled BOOLEAN DEFAULT FALSE,
  sap_config JSONB, -- connection details (encrypted)
  created_at TIMESTAMP DEFAULT NOW(),
  updated_at TIMESTAMP DEFAULT NOW()
);
```

Table: audit_logs

Immutable audit trail of all actions.

```
CREATE TABLE audit_logs (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  tenant_id UUID NOT NULL,
  user_id UUID REFERENCES users(id) ON DELETE SET NULL,
  action VARCHAR(50) NOT NULL, -- login, create, update, delete, calculate, gener-
ate_certificate
  entity_type VARCHAR(100) NOT NULL, -- user, supplier, product, bom_item,
declaration, etc.
  entity_id UUID,
  changes JSONB, -- before/after values for updates
  ip_address INET,
  user_agent TEXT,
  created_at TIMESTAMP DEFAULT NOW()
) PARTITION BY RANGE (created_at);

-- Create partitions for each year (automated via cron job)
CREATE TABLE audit_logs_2026 PARTITION OF audit_logs
  FOR VALUES FROM ('2026-01-01') TO ('2027-01-01');

CREATE INDEX idx_audit_logs_tenant ON audit_logs(tenant_id, created_at DESC);
CREATE INDEX idx_audit_logs_user ON audit_logs(user_id, created_at DESC);
CREATE INDEX idx_audit_logs_entity ON audit_logs(entity_type, entity_id);
```

Table: integrations

Integration configurations (SAP, APIs, etc.).

```

CREATE TABLE integrations (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  tenant_id UUID NOT NULL REFERENCES tenants(id) ON DELETE CASCADE,
  integration_type VARCHAR(50) NOT NULL, -- sap, oracle, api, webhook
  name VARCHAR(255),
  config JSONB NOT NULL, -- connection details (encrypted sensitive data)
  is_active BOOLEAN DEFAULT TRUE,
  last_sync_at TIMESTAMP,
  last_sync_status VARCHAR(20), -- success, error
  last_sync_message TEXT,
  created_at TIMESTAMP DEFAULT NOW(),
  updated_at TIMESTAMP DEFAULT NOW()
);

CREATE INDEX idx_integrations_tenant ON integrations(tenant_id);

```

Table: notifications

Notification queue and history.

```

CREATE TABLE notifications (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  tenant_id UUID NOT NULL REFERENCES tenants(id) ON DELETE CASCADE,
  user_id UUID REFERENCES users(id) ON DELETE CASCADE,
  notification_type VARCHAR(50) NOT NULL, -- email, in_app, sms
  subject VARCHAR(255),
  message TEXT NOT NULL,
  status VARCHAR(20) DEFAULT 'pending', -- pending, sent, failed
  sent_at TIMESTAMP,
  error_message TEXT,
  retry_count INT DEFAULT 0,
  metadata JSONB, -- additional data (e.g., declaration_id, certificate_id)
  created_at TIMESTAMP DEFAULT NOW()
);

CREATE INDEX idx_notifications_tenant ON notifications(tenant_id);
CREATE INDEX idx_notifications_user ON notifications(user_id);
CREATE INDEX idx_notifications_status ON notifications(status, created_at);

```

5.3 Multi-Tenant Data Isolation Strategy

Row-Level Multi-Tenancy:

- Every tenant-specific table includes `tenant_id` column
- PostgreSQL Row-Level Security (RLS) policies enforce tenant isolation:


```
-- Example: Enable RLS on suppliers table
ALTER TABLE suppliers ENABLE ROW LEVEL SECURITY;

-- Policy: Users can only see suppliers from their tenant
CREATE POLICY tenant_isolation_policy ON suppliers
USING (tenant_id = current_setting('app.current_tenant_id')::UUID);

-- Application sets tenant context per request:
SET app.current_tenant_id = '<tenant_uuid>';
```

Schema Isolation for Enterprise Tenants (Optional):

- Large enterprise customers get dedicated PostgreSQL schemas
- Schema name: `tenant_<tenant_id>`
- All tables duplicated in dedicated schema
- Connection pooling configured per tenant schema
- Provides: Better performance isolation, dedicated backups, custom indexes

Benefits:

- **Security:** Prevents cross-tenant data leakage
- **Compliance:** Easier to provide tenant-specific data exports/deletions (GDPR)
- **Flexibility:** Can migrate enterprise tenants to dedicated databases if needed

6. API Specifications

6.1 API Architecture

RESTful API Design Principles:

- Resource-based URLs: `/api/v1/suppliers`, `/api/v1/products`
- HTTP methods: GET (retrieve), POST (create), PUT (update), DELETE (delete), PATCH (partial update)
- Stateless: Each request contains all necessary information (JWT token)
- JSON request/response bodies
- Standard HTTP status codes

API Versioning:

- Version in URL path: `/api/v1/`, `/api/v2/`
- Maintain backward compatibility for at least 12 months when introducing breaking changes

API Gateway:

- Kong or AWS API Gateway
- Handles: Authentication, rate limiting, request transformation, logging, analytics

6.2 Authentication & Authorization

Authentication:

- OAuth 2.0 + JWT Tokens

- Login flow:

1. Client sends POST `/api/v1/auth/login` with email/password
2. Server validates credentials, generates JWT access token (1-hour expiry) and refresh token (7-day expiry)
3. Client includes access token in Authorization header: `Authorization: Bearer <token>`
4. When access token expires, client sends refresh token to `/api/v1/auth/refresh` to get new access token

JWT Token Structure:

```
{
  "sub": "user_uuid",
  "tenant_id": "tenant_uuid",
  "role": "compliance_officer",
  "email": "user@company.com",
  "iat": 1704067200,
  "exp": 1704070800
}
```

Authorization:

- Role-based access control (RBAC) enforced at API endpoint level
- Middleware checks JWT role claim against endpoint permissions
- Example: Only `admin` and `compliance_officer` can POST `/api/v1/suppliers`

6.3 Key API Endpoints**6.3.1 Authentication Endpoints****POST** `/api/v1/auth/login`

- **Description:** User login
- **Request Body:**

```
{
  "email": "user@company.com",
  "password": "SecurePassword123!"
}
```

• **Response (200 OK):**

```
{
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "refresh_token": "dGhpcyBpcyBhIHJlZnJlc2ggdG9rZW4...",
  "token_type": "Bearer",
  "expires_in": 3600,
  "user": {
    "id": "uuid",
    "email": "user@company.com",
    "role": "compliance_officer",
    "first_name": "John",
    "last_name": "Smith"
  }
}
```

• **Errors:**

- 401 Unauthorized: Invalid credentials
- 403 Forbidden: Account locked

POST `/api/v1/auth/refresh`

- **Description:** Refresh access token
- **Request Body:**

```
{
  "refresh_token": "dGhpcyBpcyBhIHJlZnJlc2ggdG9rZW4..."
}
```

• **Response (200 OK):**

```
{
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "expires_in": 3600
}
```

POST /api/v1/auth/logout

- **Description:** Logout (invalidate refresh token)
- **Request Headers:** Authorization: Bearer <access_token>
- **Response (204 No Content)**

6.3.2 User Management Endpoints

GET /api/v1/users

- **Description:** List all users (admin only)
- **Query Parameters:**
 - `role` (optional): Filter by role
 - `page` (default: 1)
 - `limit` (default: 20, max: 100)
- **Response (200 OK):**

```
{
  "data": [
    {
      "id": "uuid",
      "email": "user@company.com",
      "first_name": "John",
      "last_name": "Smith",
      "role": "compliance_officer",
      "is_active": true,
      "last_login_at": "2026-01-15T10:30:00Z",
      "created_at": "2025-06-01T12:00:00Z"
    }
  ],
  "pagination": {
    "page": 1,
    "limit": 20,
    "total": 45
  }
}
```

POST /api/v1/users

- **Description:** Create new user (admin only)
- **Request Body:**

```
{
  "email": "newuser@company.com",
  "first_name": "Jane",
  "last_name": "Doe",
  "role": "compliance_officer",
  "password": "TempPassword123!"
}
```

- **Response (201 Created):**

```
{
  "id": "uuid",
  "email": "newuser@company.com",
  "role": "compliance_officer",
  "created_at": "2026-01-15T14:00:00Z"
}
```

6.3.3 Supplier Management Endpoints**GET** /api/v1/suppliers

- **Description:** List suppliers
- **Query Parameters:**
 - status (optional): active, inactive, pending
 - country (optional): ISO 3166-1 alpha-2 code
 - search (optional): Search by company name
 - page, limit
- **Response (200 OK):**

```
{
  "data": [
    {
      "id": "uuid",
      "company_name": "Acme Parts Ltd",
      "contact_person": "Bob Johnson",
      "email": "bob@acmeparts.com",
      "phone": "+1-555-0100",
      "country_code": "US",
      "status": "active",
      "last_declaration_date": "2026-01-10T15:00:00Z",
      "compliance_rate": 95.5
    }
  ],
  "pagination": {
    "page": 1,
    "limit": 20,
    "total": 150
  }
}
```

POST /api/v1/suppliers

- **Description:** Create new supplier
- **Request Body:**

```
{
  "company_name": "Maple Supplies Inc",
  "contact_person": "Alice Green",
  "email": "alice@maplesupplies.ca",
  "phone": "+1-416-555-0200",
  "country_code": "CA",
  "address": "123 Main St, Toronto, ON M5V 3A8",
  "tax_id": "123456789RT0001"
}
```

- **Response (201 Created):**

```
{
  "id": "uuid",
  "company_name": "Maple Supplies Inc",
  "status": "active",
  "created_at": "2026-01-15T14:30:00Z"
}
```

PUT /api/v1/suppliers/{id}

- **Description:** Update supplier
- **Request Body:** (partial or full supplier object)
- **Response (200 OK):** Updated supplier object

DELETE /api/v1/suppliers/{id}

- **Description:** Delete supplier (soft delete, sets status to inactive)
- **Response (204 No Content)**

POST /api/v1/suppliers/import

- **Description:** Bulk import suppliers from CSV
- **Request:** Multipart form-data with CSV file
- **Response (202 Accepted):**

```
{
  "job_id": "uuid",
  "status": "processing",
  "message": "Import job started. You will be notified when complete."
}
```

6.3.4 Declaration Submission Endpoints

POST /api/v1/declaration-requests

- **Description:** Create declaration request for supplier
- **Request Body:**

```
{
  "supplier_id": "uuid",
  "product_ids": ["uuid1", "uuid2"],
  "fta_code": "USMCA",
  "due_date": "2026-02-01",
  "notes": "Urgent request for quarterly audit"
}
```

- **Response (201 Created):**

```
{
  "id": "uuid",
  "secure_token": "abc123xyz789",
  "portal_url": "https://company.ftamanager.com/supplier-portal?token=abc123xyz789",
  "status": "pending"
}
```

GET /api/v1/declarations

- **Description:** List declarations
 - **Query Parameters:**
 - supplier_id, product_id, fta_code, status
 - page, limit
 - **Response (200 OK):** Array of declaration objects
-

POST /api/v1/declarations (Supplier Portal)

- **Description:** Submit declaration (by supplier)
- **Request Headers:** X-Portal-Token: <secure_token> (for suppliers without login)
- **Request Body:**

```
{
  "declaration_request_id": "uuid",
  "product_id": "uuid",
  "fta_code": "USMCA",
  "country_of_origin": "US",
  "origin_criterion": "B",
  "producer_name": "Acme Manufacturing",
  "blanket_period_start": "2026-01-15",
  "blanket_period_end": "2027-01-14",
  "rvc_percentage": 75.0,
  "certifier_name": "Bob Johnson",
  "certifier_title": "Quality Manager",
  "electronic_signature": "Bob Johnson",
  "declaration_date": "2026-01-15",
  "comments": "All materials sourced from USMCA region"
}
```

• **Response (201 Created):**

```
{
  "id": "uuid",
  "status": "submitted",
  "submitted_at": "2026-01-15T16:00:00Z",
  "message": "Declaration submitted successfully. Thank you!"
}
```

POST /api/v1/declarations/{id}/documents

- **Description:** Upload supporting documents
- **Request:** Multipart form-data with file(s)
- **Response (201 Created):**

```
{
  "document_id": "uuid",
  "file_name": "certificate.pdf",
  "file_size_bytes": 245678,
  "uploaded_at": "2026-01-15T16:05:00Z"
}
```

6.3.5 Origin Calculation Endpoints

POST /api/v1/origin-calculations

- **Description:** Calculate origin for a product
- **Request Body:**

```
{
  "product_id": "uuid",
  "fta_code": "USMCA",
  "calculation_method": "transaction_value",
  "custom_bom_depth": 5
}
```

- **Response (202 Accepted):** (for async processing)

```
{
  "job_id": "uuid",
  "status": "processing",
  "estimated_completion": "2026-01-15T16:10:00Z"
}
```

- **Alternative Response (200 OK):** (for immediate calculations)

```
{
  "id": "uuid",
  "product_id": "uuid",
  "fta_code": "USMCA",
  "qualifies": true,
  "rvp_percentage": 67.5,
  "required_threshold": 60.0,
  "calculation_details": {
    "total_value": 2450.00,
    "originating_value": 1654.13,
    "non_originating_value": 795.87,
    "missing_declarations": []
  },
  "calculated_at": "2026-01-15T16:05:00Z"
}
```

GET /api/v1/origin-calculations/{id}

- **Description:** Get calculation result
 - **Response (200 OK):** Full calculation object with details
-

GET /api/v1/origin-calculations/by-product/{product_id}

- **Description:** Get latest calculation for a product
 - **Query Parameters:** fta_code (optional)
 - **Response (200 OK):** Calculation object or 404 if none found
-

6.3.6 Certificate Generation Endpoints

POST /api/v1/certificates

- **Description:** Generate certificate of origin
- **Request Body:**


```
{
  "product_id": "uuid",
  "origin_calculation_id": "uuid",
  "fta_code": "USMCA",
  "certificate_type": "blanket",
  "exporter_name": "ABC Manufacturing Inc",
  "exporter_address": "123 Export St, Detroit, MI 48201",
  "exporter_tax_id": "12-3456789",
  "producer_name": "ABC Manufacturing Inc",
  "importer_name": "Customer Corp",
  "importer_address": "456 Import Ave, Toronto, ON M5V 1A1",
  "blanket_period_start": "2026-01-15",
  "blanket_period_end": "2027-01-14",
  "certifier_name": "Sarah Johnson",
  "certifier_title": "Compliance Officer",
  "electronic_signature": "Sarah Johnson"
}
```

• **Response (201 Created):**

```
{
  "id": "uuid",
  "certificate_number": "C00-2026-0145",
  "pdf_url": "https://storage.ftamanager.com/certificates/C00-2026-0145.pdf",
  "status": "active",
  "issue_date": "2026-01-15",
  "valid_until": "2027-01-14"
}
```

GET /api/v1/certificates

- **Description:** List certificates
- **Query Parameters:** fta_code, product_id, status, page, limit
- **Response (200 OK):** Array of certificate objects

GET /api/v1/certificates/{id}/pdf

- **Description:** Download certificate PDF
- **Response (200 OK):** PDF file (application/pdf)

POST /api/v1/certificates/{id}/revoke

- **Description:** Revoke a certificate
- **Request Body:**

```
{
  "reason": "Product BOM changed, origin no longer qualifies"
}
```

• **Response (200 OK):**

```
{
  "id": "uuid",
  "status": "revoked",
  "revoked_at": "2026-01-15T17:00:00Z"
}
```

6.3.7 SAP Integration Endpoints

POST /api/v1/integrations/sap/sync-suppliers

- **Description:** Trigger supplier sync from SAP

- **Request Body:**

```
{
  "full_sync": false,
  "filter": {
    "country_codes": ["US", "CA", "MX"]
  }
}
```

• **Response (202 Accepted):**

```
{
  "job_id": "uuid",
  "status": "started",
  "message": "SAP supplier sync initiated"
}
```

POST /api/v1/integrations/sap/sync-products

- **Description:** Trigger product/BOM sync from SAP

- **Response (202 Accepted):** Similar to supplier sync

GET /api/v1/integrations/sap/sync-status/{job_id}

- **Description:** Check sync job status

- **Response (200 OK):**

```
{
  "job_id": "uuid",
  "status": "completed",
  "records_processed": 250,
  "records_created": 10,
  "records_updated": 240,
  "errors": [],
  "completed_at": "2026-01-15T17:30:00Z"
}
```

POST /api/v1/integrations/sap/export-certificate

- **Description:** Export certificate data to SAP

- **Request Body:**

```
{
  "certificate_id": "uuid"
}
```

• **Response (200 OK):**

```
{
  "status": "exported",
  "sap_document_number": "4500012345",
  "exported_at": "2026-01-15T17:45:00Z"
}
```

6.4 Error Handling

Standard Error Response Format:

```
{
  "error": {
    "code": "VALIDATION_ERROR",
    "message": "Invalid input data",
    "details": [
      {
        "field": "email",
        "message": "Email format is invalid"
      }
    ],
    "request_id": "uuid"
  }
}
```

HTTP Status Codes:

- **200 OK:** Successful GET, PUT, PATCH
- **201 Created:** Successful POST
- **202 Accepted:** Request accepted for async processing
- **204 No Content:** Successful DELETE
- **400 Bad Request:** Invalid input data
- **401 Unauthorized:** Missing or invalid authentication token
- **403 Forbidden:** Authenticated but insufficient permissions
- **404 Not Found:** Resource doesn't exist
- **409 Conflict:** Resource conflict (e.g., duplicate)
- **422 Unprocessable Entity:** Valid format but semantic errors
- **429 Too Many Requests:** Rate limit exceeded
- **500 Internal Server Error:** Server-side error
- **503 Service Unavailable:** Temporary server overload

6.5 Rate Limiting

Limits:

- **Authenticated Users:** 1000 requests/hour
- **Supplier Portal (Token-based):** 100 requests/hour
- **Admin Users:** 5000 requests/hour

Response Headers:

```
X-RateLimit-Limit: 1000
X-RateLimit-Remaining: 995
X-RateLimit-Reset: 1704070800
```

429 Response:

```
{
  "error": {
    "code": "RATE_LIMIT_EXCEEDED",
    "message": "Too many requests. Please try again later.",
    "retry_after": 3600
  }
}
```

6.6 Pagination

Query Parameters:

- `page` (default: 1)
- `limit` (default: 20, max: 100)

Response Format:

```
{
  "data": [...],
  "pagination": {
    "page": 1,
    "limit": 20,
    "total": 150,
    "total_pages": 8
  },
  "links": {
    "first": "/api/v1/suppliers?page=1&limit=20",
    "prev": null,
    "next": "/api/v1/suppliers?page=2&limit=20",
    "last": "/api/v1/suppliers?page=8&limit=20"
  }
}
```

7. Integration Requirements

7.1 SAP Integration Approach

Objectives:

- Bidirectional sync of suppliers, products, BOMs

- Export certificate data to SAP for use in purchasing and logistics
- Real-time or scheduled batch sync

Integration Methods:

Option 1: SAP OData APIs (Recommended)

- Modern RESTful APIs exposed by SAP S/4HANA and SAP Business Technology Platform
- Authentication: OAuth 2.0 with SAP Cloud Platform
- Endpoints:
- Business Partners (Suppliers): `/sap/opu/odata/sap/API_BUSINESS_PARTNER/A_BusinessPartner`
- Materials (Products): `/sap/opu/odata/sap/API_MATERIAL_STOCK_SRV/A_Material`
- BOMs: `/sap/opu/odata/sap/API_BILL_OF_MATERIAL_SRV/A_BillofMaterial`

Option 2: SAP RFC (Remote Function Calls)

- Traditional method for on-premise SAP ECC
- Requires SAP .NET Connector (NCo) or Java Connector (JCo)
- More complex but supports older SAP versions

Option 3: SAP IDoc (Intermediate Document)

- Asynchronous message exchange
- Good for large batch transfers
- Requires SAP PI/PO (Process Integration) middleware

Recommended Approach:

- **Primary:** SAP OData APIs for modern SAP S/4HANA environments
- **Fallback:** Custom RFC calls for legacy SAP ECC systems
- **Implementation:** Dedicated integration service (microservice) handling SAP communication

7.2 Data Mapping: SAP ↔ FTA App

Supplier Mapping:

SAP Field	FTA App Field	Notes
BusinessPartner	supplier.company_name	
FirstName + LastName	supplier.contact_person	Concatenate
EmailAddress	supplier.email	
PhoneNumber	supplier.phone	
Country	supplier.country_code	Convert to ISO 3166-1 alpha-2
StreetName + CityName + PostalCode	supplier.address	Concatenate
TaxNumber	supplier.tax_id	

Product Mapping:

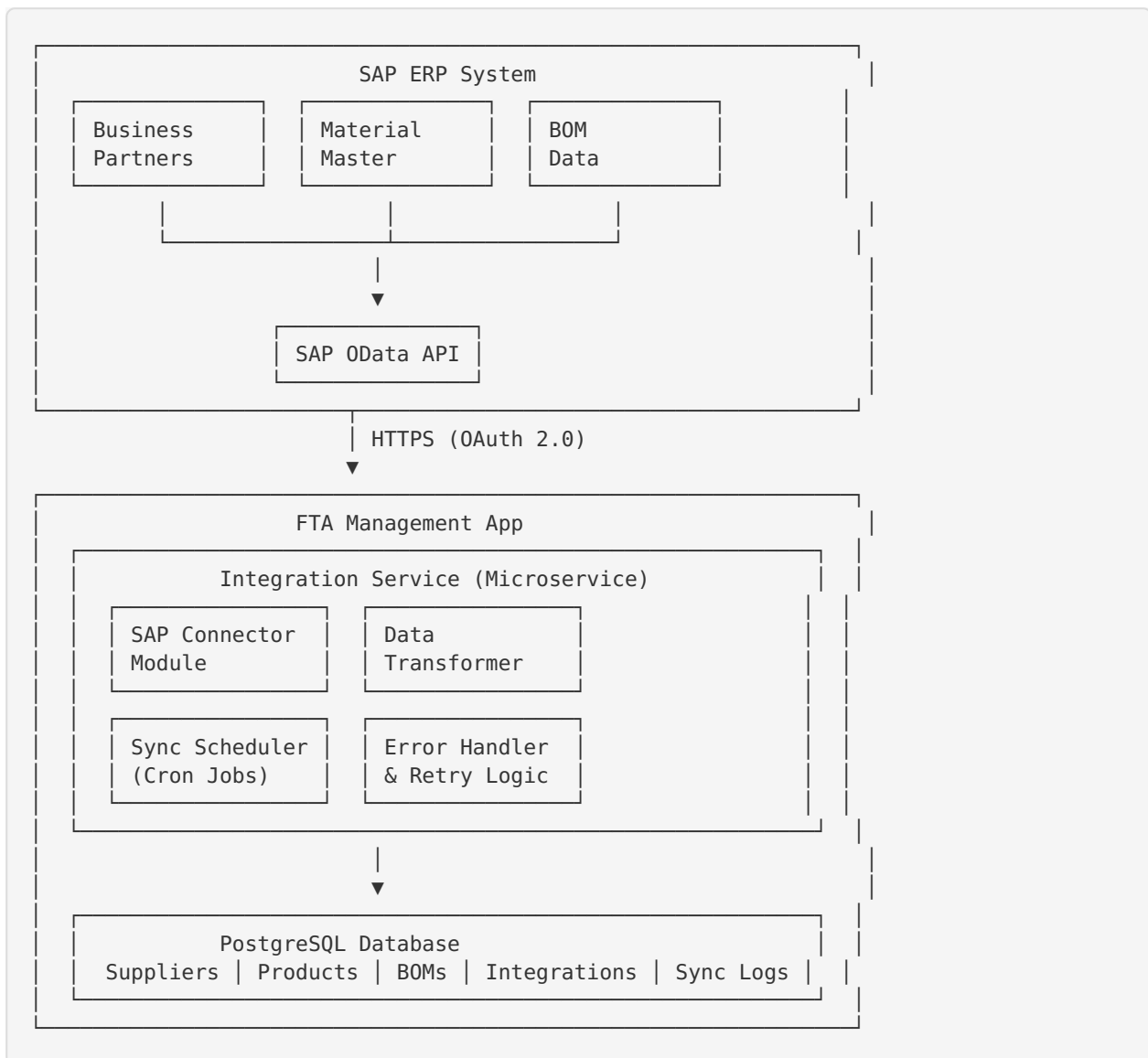
SAP Field	FTA App Field	Notes
Material	product.product_code	Unique identifier
MaterialDescription	product.description	
MaterialGroup	product.category	
BaseUnitOfMeasure	product.unit_of_measure	
StandardPrice	product.unit_cost	Convert currency if needed
CountryOfOrigin	(stored in declarations)	May not be in SAP product master

BOM Mapping:

SAP Field	FTA App Field	Notes
BillOfMaterial	parent_product_id	Header material
BillOfMaterialComponent	child_product_id	Component material
BillOfMaterialItemQuantity	bom_items.quantity	
BillOfMaterialComponentUnit	unit_of_measure	

7.3 Integration Architecture

Architecture Diagram:



Components:

1. SAP Connector Module:

- Handles authentication (OAuth token management)
- Makes API calls to SAP OData endpoints
- Fetches suppliers, products, BOMs in batches

2. Data Transformer:

- Maps SAP data format to FTA app format
- Handles currency conversion, unit conversion
- Validates data before inserting into database

3. Sync Scheduler:

- Cron jobs for scheduled syncs (e.g., daily at 2 AM)
- Configurable per tenant (frequency, scope)
- Tracks last sync timestamp for incremental updates

4. Error Handler & Retry Logic:

- Logs errors (network failures, data validation errors)
- Implements exponential backoff retry for transient failures
- Sends notifications to admins on persistent failures

7.4 Sync Workflows

Supplier Sync (SAP → FTA App):

1. Scheduler triggers sync job at configured time
2. SAP Connector fetches Business Partners from SAP (filter: **new**/updated **since** last sync)
3. **Data** Transformer maps SAP fields **to** FTA app supplier schema
4. **For** each supplier:
 - a. Check **if** exists in FTA app (by SAP BP number)
 - b. **If** exists: Update existing record
 - c. **If new**: Create **new** supplier record
5. **Log** sync results (records processed, created, updated, errors)
6. Send **notification** **to** admin **if** errors occurred

Product & BOM Sync (SAP → FTA App):

1. Fetch Materials from SAP (filter: **new**/updated)
2. Map **to** product schema
3. Upsert products in FTA app
4. Fetch BOMs from SAP **for** each product
5. Delete existing BOM items in FTA app **for** products being synced
6. Insert **new** BOM items from SAP
7. Recalculate BOM levels (depth) **for** updated products
8. Mark **origin** calculations as stale (require recalculation)

Certificate Export (FTA App → SAP):

1. User generates certificate in FTA app
2. **System** creates **export** job
3. Transform certificate **data** **to** SAP **format** (e.g., create custom Z-table entry **or** attach **to** material master)
4. Call SAP API **to** create/update record
5. Store SAP document number in FTA app certificate record
6. Mark certificate as **"exported to SAP"**

7.5 Data Sync Configuration (Per Tenant)

UI Settings Screen:

- **Enable SAP Integration:** On/Off toggle
- **SAP Connection Details:**
 - SAP System URL
 - Client ID / Client Secret (OAuth)
 - SAP Client Number (for ECC systems)
- **Sync Schedule:**
 - Frequency: Manual only, Hourly, Daily, Weekly
 - Time: Specific time for daily/weekly syncs
- **Sync Scope:**
 - Sync Suppliers: Yes/No
 - Sync Products: Yes/No
 - Sync BOMs: Yes/No
 - Export Certificates: Yes/No
- **Conflict Resolution:**
 - On data conflict: Prefer SAP data / Prefer FTA app data / Manual review

Stored in Database:

```

UPDATE tenant_settings
SET sap_integration_enabled = TRUE,
    sap_config = '{
      "url": "https://sap.company.com",
      "client_id": "encrypted_client_id",
      "client_secret": "encrypted_secret",
      "sync_schedule": "daily",
      "sync_time": "02:00",
      "sync_suppliers": true,
      "sync_products": true,
      "sync_boms": true,
      "export_certificates": true,
      "conflict_resolution": "sap_wins"
    }'::jsonb
WHERE tenant_id = '<tenant_uuid>';

```

7.6 Error Handling & Monitoring

Error Categories:

- **Authentication Errors:** OAuth token expired, invalid credentials
- **Network Errors:** Timeout, connection refused
- **Data Validation Errors:** Invalid field values, missing required fields
- **Business Logic Errors:** Duplicate records, circular BOM references

Monitoring:

- **Sync Status Dashboard:** Shows last sync time, records processed, errors for each tenant
- **Alerts:** Email notifications to admins on sync failures
- **Logs:** Detailed logs in `integrations` and dedicated `integration_logs` table

Integration Logs Table:

```

CREATE TABLE integration_logs (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  tenant_id UUID NOT NULL,
  integration_id UUID REFERENCES integrations(id),
  sync_type VARCHAR(50), -- supplier_import, product_import, certificate_export
  status VARCHAR(20), -- success, partial_success, error
  records_processed INT DEFAULT 0,
  records_created INT DEFAULT 0,
  records_updated INT DEFAULT 0,
  records_failed INT DEFAULT 0,
  error_details JSONB,
  started_at TIMESTAMP DEFAULT NOW(),
  completed_at TIMESTAMP
);

```

8. FTA Rules Engine

8.1 Modeling FTA Rules

Challenge:

FTA rules are complex, vary by agreement, and change over time. The system must be flexible to accommodate diverse rules without requiring code changes.

Approach:

- **Database-Driven Rules:** Store rules in `fta_rules` and `product_specific_rules` tables
- **Rule Versioning:** Track effective dates for rules; apply correct version based on declaration/certificate date
- **Modular Rule Engine:** Separate calculation logic from rule definitions

8.2 FTA Rule Components

General FTA Rules (Table: `fta_rules`):

- **FTA Code:** Unique identifier (e.g., USMCA, CFTA)
- **Member Countries:** List of countries in the agreement
- **Default RVC Threshold:** Minimum regional value content (e.g., 60% for USMCA)
- **Calculation Methods:** Supported methods (transaction value, net cost)
- **Origin Criteria:** Valid origin criteria (A, B, C, D)
- **De Minimis Percentage:** Allowable non-originating content (e.g., 10%)
- **Max Blanket Period:** Maximum validity for blanket certificates (typically 365 days)

Product-Specific Rules (Table: `product_specific_rules`):

- **HS Code Range:** Applies to specific HS codes or ranges
- **Tariff Shift Rule:** E.g., "A change to heading 8413 from any other chapter"
- **RVC Threshold Override:** If different from default
- **Additional Requirements:** Special conditions (e.g., automotive labor value content for USMCA)

Example: USMCA Automotive Rules

```
INSERT INTO product_specific_rules (fta_code, hs_code_from, hs_code_to, rule_text,
rvc_threshold, additional_requirements) VALUES
('USMCA', '8701', '8708', 'A change to heading 8701 through 8708 from any other
chapter', 75.00, 'Labor Value Content: 40-45% from facilities paying min $16/hour;
Steel/Aluminum: 70% North American');
```

8.3 Origin Calculation Algorithm

High-Level Algorithm:

```

FUNCTION CalculateOrigin(product, fta_code, calculation_method):
1. Fetch FTA rules for fta_code
2. Fetch product-specific rules for product.hs_code and fta_code
3. Build BOM tree for product (up to tenant's max_bom_depth)
4. For each component in BOM:
  a. Fetch latest declaration for component + fta_code
  b. If no declaration: Mark as non-originating, flag as missing
  c. If declaration exists: Determine origin status from origin_criterion
5. Calculate values (bottom-up):
  a. For each level (leaf to root):
    i. Component value = quantity × unit_cost
    ii. If component originates: Add to originating_value
    iii. Else: Add to non_originating_value
6. Calculate RVC:
  IF calculation_method = 'transaction_value':
    RVC = ((total_value - non_originating_value) / total_value) × 100
  ELSE IF calculation_method = 'net_cost':
    RVC = ((net_cost - non_originating_value) / net_cost) × 100
7. Apply tariff shift rule (if applicable):
  a. Check if product HS code differs from all non-originating components per rule
  b. If yes: Consider tariff shift satisfied
8. Apply de minimis rule:
  a. If non_originating_value / total_value ≤ de_minimis_percentage: Ignore for
  certain products
9. Determine qualification:
  IF RVC ≥ required_rvc_threshold AND tariff_shift_satisfied (if required):
    qualifies = TRUE
  ELSE:
    qualifies = FALSE
10. Save calculation result with detailed breakdown
11. Return result

```

Example Calculation (USMCA, Transaction Value Method):

Product: XYZ-500 Industrial Pump Assembly

- **Transaction Value:** \$2,450.00

- **BOM:**

- Motor Assembly (MA-100) - USA - \$1,102.50 → Originating
- Impeller (IMP-250) - Canada - \$612.50 → Originating
- Casing (CS-180) - USA - \$490.00 → Originating
- Stator (S-300) - China - \$490.00 → Non-originating (missing declaration assumed)
- Fasteners (FKIT-50) - Germany - \$245.00 → Non-originating

Calculation:

- Originating Value = \$1,102.50 + \$612.50 + \$490.00 = \$2,205.00
- Non-Originating Value = \$490.00 + \$245.00 = \$735.00 (correction from example: should total \$2,450)

Wait, let me recalculate:

- Total Value = \$2,450.00
- Originating Value = \$1,102.50 + \$612.50 + \$490.00 = \$2,205.00
- Non-Originating Value = \$490.00 + \$245.00 = \$735.00
- Check: \$2,205.00 + \$735.00 = \$2,940.00 (exceeds total, likely overlapping components)

Corrected Assumption: Motor Assembly already includes Stator, so:

- Originating: Motor Assembly - Stator + Impeller + Casing = \$1,102.50 - \$490.00 + \$612.50 + \$490.00 = \$1,715.00

- Non-Originating: Stator + Fasteners = \$490.00 + \$245.00 = \$735.00
- Total: \$1,715.00 + \$735.00 = \$2,450.00 ✓

RVC Calculation:

- $RVC = ((2,450 - 735) / 2,450) \times 100 = (1,715 / 2,450) \times 100 = \mathbf{70.0\%}$
- USMCA Threshold: 60%
- **Result: QUALIFIES** ✓

8.4 Handling Complex Scenarios

Scenario 1: Missing Declarations

- **Issue:** Component lacks origin declaration
- **Handling:** Assume non-originating; flag in calculation result; notify user to request declaration
- **Impact:** May cause product to not qualify when it otherwise could

Scenario 2: Expired Declarations

- **Issue:** Blanket declaration expired
- **Handling:** Treat as non-originating if current date > blanket_period_end; send alert to request renewal

Scenario 3: Circular BOM References

- **Issue:** Product A contains Product B, which contains Product A
- **Handling:** Prevent on insert/update via database trigger; detect via recursive CTE check

Scenario 4: Multi-Country Components

- **Issue:** Component sourced from multiple countries
- **Handling:** User specifies primary supplier per BOM item; use that supplier's declaration

Scenario 5: Accumulation (USMCA)

- **Issue:** Materials originating in one USMCA country used to produce goods in another USMCA country can be considered originating
- **Handling:** Track origin country in declarations; if origin_country IN ('US', 'CA', 'MX'), consider originating for USMCA calculations

8.5 Rule Versioning & Updates

Need:

FTA rules change (e.g., new trade agreements, threshold adjustments). System must support rule updates without invalidating historical calculations.

Approach:

- **Effective Date Field:** `fta_rules.effective_date` , `product_specific_rules.effective_date`
- **Immutable Rules:** Never update existing rules; insert new versions with new effective dates
- **Calculation Uses Historical Rules:** When calculating origin for a past date (e.g., validating old certificate), use rules effective at that time

Example:

```
-- Original USMCA rule
INSERT INTO fta_rules (fta_code, default_rvc_threshold, effective_date) VALUES ('US-
MCA', 60.00, '2020-07-01');

-- Updated USMCA rule (hypothetical increase)
INSERT INTO fta_rules (fta_code, default_rvc_threshold, effective_date) VALUES ('US-
MCA', 65.00, '2027-01-01');

-- Query for applicable rule:
SELECT * FROM fta_rules
WHERE fta_code = 'USMCA' AND effective_date <= '2026-01-15'
ORDER BY effective_date DESC
LIMIT 1;
-- Returns: 60.00 threshold
```

8.6 User-Configurable Rules (Advanced Phase)

Phase 2 Feature:

Allow admins to define custom rules per tenant (e.g., internal company policies stricter than FTA requirements).

Implementation:

- New table: `custom_rules` with `tenant_id` foreign key
- Rule engine checks custom rules first, then standard FTA rules
- UI for creating/editing custom rules (rule builder interface)

9. Security & Compliance

9.1 Data Encryption

Encryption at Rest:

- **Database:** PostgreSQL with Transparent Data Encryption (TDE) or disk-level encryption (AWS EBS encryption, Azure Disk Encryption)
- **File Storage (S3/Blob):** Server-side encryption (SSE) with AES-256
- AWS S3: Use SSE-S3 or SSE-KMS (AWS Key Management Service)
- Azure Blob: Use Azure Storage Service Encryption
- **Backups:** Encrypted backups using provider-managed or customer-managed keys

Encryption in Transit:

- **TLS 1.3:** All API communication over HTTPS
- **Certificate Management:** Let's Encrypt or commercial SSL/TLS certificates
- **HSTS (HTTP Strict Transport Security):** Force HTTPS connections

Sensitive Data Encryption:

- **Passwords:** Hashed with bcrypt (cost factor 12), never stored in plain text
- **API Keys/Secrets:** Encrypted in database using AES-256; decrypted only in memory when needed
- **SAP Credentials:** Stored encrypted in `tenant_settings.sap_config` JSONB field

9.2 Multi-Tenant Data Isolation

Database-Level Isolation:

- **Row-Level Security (RLS):** PostgreSQL policies enforce `tenant_id` filtering

- **Query Validation:** All queries include `WHERE tenant_id = :current_tenant_id`
- **Application-Level Context:** Set tenant context at request start; all DB queries use it

Example RLS Policy:

```
ALTER TABLE suppliers ENABLE ROW LEVEL SECURITY;

CREATE POLICY tenant_isolation ON suppliers
USING (tenant_id = current_setting('app.current_tenant_id')::UUID);
```

Testing:

- Unit tests verify queries include tenant_id filter
- Integration tests attempt cross-tenant access (should fail)
- Penetration testing to validate isolation

9.3 Audit Logging

What to Log:

- All data modifications (create, update, delete)
- User authentication events (login, logout, failed attempts)
- Certificate generation and revocation
- Origin calculations
- SAP sync events
- Settings changes

Log Structure (audit_logs table):

- User ID, tenant ID, action type, entity type, entity ID
- Before/after values (JSONB)
- IP address, user agent
- Timestamp

Immutability:

- Append-only table (no updates or deletes allowed)
- Partition by date for performance
- Archive old logs to cold storage after 2 years (retain for 7 years total)

Access:

- Auditors and admins can view logs via UI
- Exportable for external review (compliance audits)

9.4 GDPR & Data Privacy Compliance

Personal Data:

- User email, name, phone
- Supplier contact person details

GDPR Requirements:

1. Right to Access:

- Users can download their personal data via account settings
- API endpoint: `GET /api/v1/users/me/data-export` (returns JSON with all user data)

2. Right to Erasure (“Right to be Forgotten”):

- Users can request account deletion

- Anonymize personal data instead of deleting (to preserve audit trail integrity)
- Process: Mark user as deleted, replace email with `deleted_user_<id>@example.com`, clear name/phone
- API endpoint: `DELETE /api/v1/users/me` (with confirmation)

3. Data Portability:

- Export data in machine-readable format (JSON, CSV)

4. Consent Management:

- Clear terms of service and privacy policy
- Checkbox consent during signup
- Log consent timestamp

5. Data Minimization:

- Only collect necessary data
- Regularly purge unused supplier contacts (opt-in to keep)

6. Data Processing Agreement:

- Provide DPA to enterprise customers (required for GDPR)
- Document data flows and subprocessors

Data Retention Policy:

- Active user data: Retained indefinitely while account active
- Deleted accounts: Anonymized data retained for audit purposes (7 years)
- Audit logs: 7 years (compliance requirement)
- Certificates: 10 years (customs audit requirement)

9.5 Role-Based Access Control (RBAC)

Roles:

- **Admin:** Full access to all features, user management, settings
- **Compliance Officer:** Manage suppliers, products, calculations, certificates; no user management or settings
- **Supplier:** Access supplier portal only; submit declarations for own company
- **Auditor:** Read-only access to all data; view audit logs; no modifications

Permission Matrix (Detailed):

Feature	Admin	Compliance Officer	Supplier	Auditor
Users				
View all users	✓	✗	✗	✓
Create user	✓	✗	✗	✗
Edit user	✓	✗	✗	✗
Delete user	✓	✗	✗	✗
Suppliers				
View suppliers	✓	✓	🔍 Own only	✓
Create supplier	✓	✓	✗	✗
Edit supplier	✓	✓	✗	✗
Delete supplier	✓	✓	✗	✗
Products				
View products	✓	✓	✗	✓
Create product	✓	✓	✗	✗
Edit product/ BOM	✓	✓	✗	✗
Delete product	✓	✓	✗	✗
Declarations				
View declara- tions	✓	✓	🔍 Own only	✓
Submit declara- tion	✓	✓	✓	✗
Approve declar- ation	✓	✓	✗	✗
Delete declara- tion	✓	✓	✗	✗
Origin Calcula- tions				

Feature	Admin	Compliance Officer	Supplier	Auditor
View calculations	✓	✓	✗	✓
Run calculation	✓	✓	✗	✗
Certificates				
View certificates	✓	✓	✗	✓
Generate certificate	✓	✓	✗	✗
Revoke certificate	✓	✓	✗	✗
Settings				
View settings	✓	✗	✗	✓
Edit settings	✓	✗	✗	✗
Configure integrations	✓	✗	✗	✗
Audit Logs				
View audit logs	✓	🔍 Own actions	✗	✓
Export audit logs	✓	✗	✗	✓

Implementation:

- Middleware checks JWT role claim against required permissions for each endpoint
- Database RLS policies provide additional layer (defense in depth)

9.6 Additional Security Measures**Multi-Factor Authentication (MFA):**

- Optional for all users, mandatory for admins
- TOTP-based (Google Authenticator, Authy)
- Backup codes for account recovery

IP Whitelisting (Enterprise Tier):

- Restrict API access to specific IP ranges
- Configured per tenant in settings

API Key Management:

- For programmatic access (e.g., SAP integration)

- Rotating API keys (expire after 90 days, auto-rotate option)
- Separate keys for read vs. write operations

Input Validation & Sanitization:

- Validate all user inputs against expected formats
- Sanitize inputs to prevent SQL injection, XSS attacks
- Use parameterized queries (ORM or prepared statements)

Dependency Scanning:

- Automated scanning of npm/Python dependencies for known vulnerabilities (Snyk, Dependabot)
- Regular updates of libraries

Penetration Testing:

- Annual penetration tests by third-party security firm
- Bug bounty program (Phase 2)

Compliance Certifications (Future):

- **SOC 2 Type II:** Demonstrates security, availability, confidentiality controls
- **ISO 27001:** Information security management
- **GDPR Compliance:** Self-assessment + DPA for customers

10. Non-Functional Requirements

10.1 Performance Benchmarks

Response Time Targets:

- **Web Page Load:**
 - Initial load (first-time visit): <2 seconds
 - Subsequent loads (cached): <500ms
- **API Response Time:**
 - Simple queries (GET single resource): <100ms (p50), <300ms (p95)
 - List queries (GET with pagination): <200ms (p50), <500ms (p95)
 - Complex operations (origin calculation): <5 seconds for 5-level BOM with 100 components
 - Background jobs (SAP sync, report generation): No real-time requirement; must complete within configured window (e.g., 1 hour for nightly sync)

Throughput:

- **Concurrent Users:** Support 1,000+ concurrent users per tenant (enterprise tier)
- **API Requests:** Handle 10,000 requests/minute (aggregate across all tenants)
- **Origin Calculations:** Process 100+ calculations/minute in background queue

Database Performance:

- **Query Latency:** <50ms for indexed queries, <200ms for complex joins
- **Write Throughput:** 1,000+ inserts/second
- **Connection Pooling:** PgBouncer with 100-200 max connections per database

10.2 Availability & Uptime Targets

SLA Commitments (by Tier):

- **Basic Tier:** 99.5% uptime (max 3.65 hours downtime/month)
- **Professional Tier:** 99.9% uptime (max 43 minutes downtime/month)
- **Enterprise Tier:** 99.95% uptime (max 21 minutes downtime/month)

High Availability Architecture:

- **Multi-AZ Deployment:** Application servers and databases across multiple availability zones
- **Load Balancers:** Distribute traffic, health checks, auto-failover
- **Database Replication:** Primary-replica setup with automatic failover (AWS RDS Multi-AZ, Azure SQL Database)
- **Redundant Services:** At least 2 instances of each microservice

Planned Maintenance:

- Scheduled during low-traffic windows (e.g., Sundays 2-4 AM UTC)
- Advance notification to customers (7 days for major updates, 24 hours for minor patches)

10.3 Data Backup & Disaster Recovery

Backup Strategy:

- **Database Backups:**
 - Automated daily backups (AWS RDS automated backups, Azure SQL Database)
 - Retention: 30 days for automated backups
 - Weekly manual snapshots retained for 1 year
- **File Storage Backups:**
 - S3/Blob versioning enabled (retain 90 days)
 - Cross-region replication for critical documents (certificates, declarations)
- **Backup Testing:** Quarterly restore tests to validate backup integrity

Disaster Recovery Plan:

- **RTO (Recovery Time Objective):** 4 hours (time to restore service after disaster)
- **RPO (Recovery Point Objective):** 1 hour (max data loss acceptable)
- **DR Procedure:**
 1. Detect outage (automated monitoring alerts)
 2. Assess severity (regional outage vs. temporary glitch)
 3. If disaster declared: Failover to secondary region
 4. Restore database from latest backup to secondary region
 5. Update DNS to point to secondary region
 6. Notify customers of temporary service disruption
 7. Monitor recovery, validate data integrity
 8. Post-mortem analysis

Geographic Redundancy:

- Primary region: US East (AWS us-east-1 or Azure East US)
- Secondary region: US West (AWS us-west-2 or Azure West US) for disaster recovery
- EU customers: Primary in EU West (AWS eu-west-1 or Azure West Europe) for data residency compliance

10.4 Scalability

Horizontal Scaling:

- **Application Servers:** Auto-scaling groups (AWS Auto Scaling, Azure VMSS)
 - Scale up: CPU >70% for 5 minutes
 - Scale down: CPU <30% for 10 minutes
 - Min instances: 2, Max instances: 20 (adjustable per demand)
- **Background Workers:** Separate worker pool for async jobs, scales based on queue depth
 - Scale up: Queue depth >100 jobs
 - Scale down: Queue depth <10 jobs

Database Scaling:

- **Vertical Scaling:** Increase instance size (CPU, RAM) for primary database during initial growth
- **Horizontal Scaling (Read Replicas):** Add read replicas for reporting queries, analytics
- **Sharding (Future):** For extreme scale (10,000+ tenants), shard by tenant_id

Caching Strategy:

- **Redis Caching:**
 - User sessions (30-minute TTL)
 - FTA rules (24-hour TTL, invalidate on rule updates)
 - Origin calculations (cache key: product_id + fta_code + bom_hash; 7-day TTL)
- **CDN Caching:** Static assets (JavaScript, CSS, images) cached at edge locations (CloudFront, Azure CDN)

Load Testing:

- Regular load tests simulating peak traffic (e.g., end-of-quarter compliance rush)
- Tools: Apache JMeter, Gatling, or K6
- Target: Support 5x normal traffic without degradation

10.5 Compliance Certifications (Roadmap)

Immediate:

- **GDPR Compliance:** Self-assessment, privacy policy, DPA for customers
- **PCI DSS:** Not required (no credit card processing in MVP; payment gateway handles)

Phase 2 (12-18 months):

- **SOC 2 Type II:** Audit of security, availability, confidentiality controls
- Demonstrates trust to enterprise customers
- Required by many large companies for vendor onboarding

Phase 3 (24+ months):

- **ISO 27001:** Information security management system certification
- **ITAR Compliance (if targeting defense contractors):** U.S. defense trade controls

11. User Flows

11.1 Supplier Declaration Submission Flow

Actors: Compliance Officer (requestor), Supplier (submitter)

Preconditions: Supplier exists in system with valid email

Flow:

1. Compliance Officer logs **in** to FTA Manager
2. Navigates to Declarations > Request Declarations
3. Selects supplier from dropdown (or creates new supplier)
4. Selects product(s) needing declarations (can select multiple)
5. Selects FTA (e.g., USMCA)
6. Sets due date (**default**: 30 days from today)
7. Adds optional notes/instructions
8. Clicks "Send Request"
- ↓
9. System creates declaration_request record(s)
10. System generates unique secure_token **for** each request
11. System sends email to supplier:
 - Subject: "Origin Declaration Request from [Company Name]"
 - Body: Includes link to supplier portal with token
 - Example: <https://company.ftamanager.com/supplier-portal?token=abc123xyz789>
12. Compliance Officer sees confirmation: "Request sent to [Supplier Name]"
- ↓
13. Supplier receives email, clicks link
14. Supplier portal opens (no login required, token-based access)
15. Portal shows:
 - Requested by: [Company Name]
 - Due date: [Date]
 - Product(s): [List]
 - FTA: [USMCA]
16. Supplier fills form:
 - Selects country of origin from dropdown
 - Selects origin criterion (A, B, C, or D with descriptions)
 - Enters producer name (optional)
 - Selects blanket period (optional, up to 12 months)
 - Enters RVC percentage (**if** applicable)
 - Uploads supporting documents (e.g., supplier's supplier certificates)
 - Enters certifier name, title
 - Checks electronic signature agreement ("I certify this information is accurate")
 - Types name as signature
17. Clicks "Save Draft" (can **return** later) OR "Submit Declaration"
- ↓
18. System validates form:
 - All required fields filled?
 - Date ranges valid?
 - File uploads <10MB and allowed types?
19. If validation fails: Show error messages, allow correction
20. If validation passes:
 - Save declaration record
 - Upload documents to S3 (with virus scan)
 - Send confirmation email to supplier
 - Send notification email to compliance officer
 - Mark declaration_request status as "submitted"
21. Supplier sees: "Thank you! Your declaration has been submitted."
- ↓
22. Compliance Officer receives notification: "New declaration from [Supplier]"
23. Officer navigates to Declarations > Pending Review
24. Reviews declaration details, downloads supporting documents
25. Officer clicks "Approve" or "Request Changes"
26. If approved:
 - Declaration status = "approved"
 - Notification sent to supplier (optional)
 - Declaration available **for** origin calculations
27. If changes requested:
 - Declaration status = "under_review"
 - Email sent to supplier with feedback
 - Supplier can edit and resubmit

END

Alternate Flow: Reminder Escalation

```
If supplier hasn't submitted by due date:  
1. System's scheduled job runs daily at 9 AM  
2. Checks for declaration_requests with status="pending" and due_date < today  
3. Marks request as "overdue"  
4. Sends reminder email to supplier (if reminder_count < 3)  
5. Increments reminder_count  
6. Sends alert to compliance officer: "Overdue: [Supplier] hasn't responded to  
  declaration request"  
7. Officer can manually follow up or send additional reminder
```

11.2 Origin Calculation Workflow

Actor: Compliance Officer

Preconditions: Product exists with BOM defined; declarations available for components

Flow:

1. Compliance Officer logs in
2. Navigates to Products > [Select Product XYZ-500]
3. Clicks "Calculate Origin" button
↓
4. System opens origin calculation screen:
 - Displays product details (HS code, description, unit cost)
 - Shows BOM tree (expandable, color-coded by declaration status)
 - Provides calculation settings:
 - * FTA dropdown (default: tenant's default FTA)
 - * Calculation method dropdown (Transaction Value or Net Cost)
 - * BOM depth selector (default: tenant's max_bom_depth)
5. Officer reviews BOM tree:
 - Green icon (✓) = Component has valid declaration
 - Red icon (✗) = Component non-originating
 - Orange icon (⚠) = Missing declaration
6. If missing declarations:
 - System shows warning:
 - "2 components missing declarations. Calculation will assume non-originating."
 - Officer can click "Request Declarations" to send requests immediately
 - OR proceed with calculation anyway
7. Officer selects FTA = USMCA, Method = Transaction Value
8. Clicks "Run Calculation"
↓
9. System validates:
 - BOM tree exists and has no circular references?
 - FTA rules available for selected FTA?
10. If simple BOM (< 50 components, ≤ 3 levels):
 - Calculation runs synchronously (real-time)
 - Shows progress indicator for 2-5 seconds
11. If complex BOM (> 50 components or > 3 levels):
 - Calculation queued for background processing
 - User sees: "Calculation started. You'll be notified when complete (est. 1-2 min)."
 - User can navigate away; notification sent when done
- ↓
12. Calculation engine runs (see Algorithm in Section 8.3):
 - Fetches all declarations for components
 - Calculates originating vs. non-originating values recursively
 - Computes RVC percentage
 - Applies tariff shift and de minimis rules
 - Determines qualification
13. Saves result to origin_calculations table
14. Marks is_current = TRUE, previous calculations = FALSE
↓
15. User redirected to Calculation Results page (or notified if async):
 - Large banner: "QUALIFIES ✓" or "DOES NOT QUALIFY ✗"
 - Shows RVC percentage vs. threshold (e.g., 67.5% > 60%)
 - Displays calculation breakdown:
 - * Total value
 - * Originating value
 - * Non-originating value
 - * Formula applied
 - Shows pie chart (originating vs. non-originating)
 - Lists top originating and non-originating components
 - If missing declarations: Lists components assumed non-originating
16. Officer reviews results
17. Actions available:
 - Download Report PDF: Generates detailed PDF for audit/sharing
 - Generate Certificate: (if qualifies) Navigates to certificate creation
 - Recalculate: Returns to settings to adjust parameters
 - What-If Analysis (Phase 2): Change supplier for component, recalculate instantly

END

Alternate Flow: Non-Qualifying Result

If result = DOES NOT QUALIFY:

1. System shows red banner: "DOES NOT QUALIFY ✗"
2. Displays RVC percentage below threshold (e.g., 55% < 60%)
3. Shows which components contributed most to non-originating value
4. Suggestions (Phase 2):
 - "If you source Stator (S-300) from a USMCA country, RVC would increase to 70%"
 - "Request declaration from [Supplier X] to verify origin"
5. Officer can:
 - Request missing declarations
 - Review BOM to identify sourcing changes
 - Mark product as "non-qualifying" for this FTA, try another FTA

11.3 Certificate Generation & Distribution Flow

Actor: Compliance Officer

Preconditions: Product has qualifying origin calculation

Flow:

1. Compliance Officer on Product Details or Calculation Results page
2. Sees "Generate Certificate" button (enabled only **if** latest calculation qualifies)
3. Clicks "Generate Certificate"



4. System opens Certificate Generation Wizard (3 steps):

STEP 1: Certificate Type & Basic Info

5. Officer selects:
 - Certificate Type: Single Shipment or Blanket (up to 12 months)
 - FTA: [Pre-filled from calculation]
 - Product: [Pre-filled]
6. If Blanket: Selects start and end dates (max 365 days apart)
7. Clicks "Next"



STEP 2: Party Information

8. System pre-fills exporter info from tenant profile:
 - Exporter Name: [Company Name]
 - Address: [Company Address]
 - Tax ID: [Company Tax ID]
9. Officer can edit **if** needed (e.g., different export entity)
10. Officer enters Producer info:
 - Producer Name: [Same as exporter or different]
 - Address: [If different]
 - Tax ID: [Optional or "Available upon request"]
11. Officer enters Importer info (**if** known):
 - Importer Name: [Customer name or "Various" **for** blanket]
 - Address: [Customer address]
 - Tax ID: [Customer tax ID]
 - NOTE: Can leave blank **if** importer unknown (common **for** blanket certificates)
12. Clicks "Next"



STEP 3: Review & Certify

13. System displays certificate preview:
 - All data elements populated
 - Product description and HS code
 - Origin criterion (from calculation)
 - Blanket period (**if** applicable)
 - Certifier: [Officer's name from user profile]
 14. Officer reviews all details
 15. Officer enters:
 - Certifier Title: [e.g., "Compliance Officer"]
 - Electronic Signature: [Types name or draws signature]
 - Checkbox: "I certify that the information is true and accurate"
 16. Clicks "Generate Certificate"
-
17. System validates:
 - All required fields completed?
 - Origin calculation still current and qualifying?
 18. System generates certificate:
 - Creates certificate record **in** database
 - Assigns unique certificate_number (e.g., C00-2026-0145)
 - Generates PDF using template (pre-formatted per FTA requirements)
 - Uploads PDF to S3
 - Logs certificate issuance **in** audit trail
 19. System shows success message: "Certificate C00-2026-0145 generated successfully!"
 20. Officer presented with options:
 - Download PDF: Immediately download certificate
 - Email to Customer: Enter customer email, send certificate via email
 - View Certificate: See certificate details page

- Generate Another: Return to wizard **for** another product

↓

21. Officer selects "Email to Customer"
22. Enters customer email address(es) (comma-separated **for** multiple)
23. Adds optional message
24. Clicks "Send Email"
25. System sends email:
 - Subject: "Certificate of Origin - [Product Name]"
 - Body: Professional template with company logo
 - Attachment: Certificate PDF
26. Email sent, confirmation shown

↓

27. Officer navigates to Certificates > All Certificates
28. Sees newly generated certificate **in** list
29. Can view, download, re-send, or revoke as needed

END

Alternate Flow: Certificate Revocation

If product BOM changes or origin recalculated and no longer qualifies:

1. Compliance Officer navigates to Certificates > [Select Certificate]
2. Clicks "Revoke Certificate" button
3. System prompts: "Are you sure? This action cannot be undone."
4. Officer enters revocation reason: [Text input]
5. Clicks "Confirm Revoke"
6. System:
 - Updates certificate status = "revoked"
 - Records revoked_at timestamp and revoked_by user
 - Logs **in** audit trail
7. Certificate marked as revoked **in** list (grayed **out** or with "Revoked" badge)
8. If certificate was exported to SAP: Send notification to update SAP
9. Officer can notify customer of revocation (manual email or system-generated)

11.4 Audit Trail Review Flow

Actor: Auditor or Admin

Preconditions: User has auditor role or admin role

Flow:

1. Auditor logs **in** to FTA Manager
 2. Navigates to Settings > Audit Logs (or dedicated Audit menu item)
 - ↓
 3. Audit Logs page displays:
 - Filters:
 - * Date range picker (**default**: last 30 days)
 - * User dropdown (All users or specific user)
 - * Action type dropdown (All, Create, Update, Delete, Login, Certificate, etc.)
 - * Entity type dropdown (All, Supplier, Product, BOM, Declaration, Certificate, etc.)
 - c.)
 - Search box: Search by entity ID or keywords
 4. Auditor applies filters:
 - Date range: January 1-15, 2026
 - Action type: Update
 - Entity type: BOM
 5. Clicks "Apply Filters"
 - ↓
 6. System queries audit_logs table:
 - Filters by tenant_id, date range, action, entity type
 - Paginates results (20 per page)
 7. Displays audit log entries **in** table:
 - Columns: Timestamp, User, Action, Entity Type, Entity ID, Changes, IP Address
 - Example row:
 - * 2026-01-10 14:23:45 | Sarah Johnson | Update | BOM Item | uuid-123 | [View Changes] | 192.168.1.50
 8. Auditor clicks "View Changes" **for** a specific entry
 - ↓
 9. System opens modal showing before/after values:
 - Before:
 - * child_product_id: uuid-abc (Stator S-300)
 - * quantity: 1.0
 - * unit_cost: \$450.00
 - After:
 - * child_product_id: uuid-abc (Stator S-300)
 - * quantity: 1.0
 - * unit_cost: \$490.00 (CHANGED)
 10. Auditor reviews change, notes unit cost increase
 11. Can click entity ID to navigate to entity details (**if** permissions allow)
 - ↓
 12. Auditor continues reviewing logs:
 - Identifies user who made change
 - Checks **if** change triggered recalculation (looks **for** subsequent calculation log entry)
 - Validates change aligns with business process
 13. Auditor can export logs:
 - Clicks "Export to CSV"
 - System generates CSV with all filtered logs
 - Downloads to local machine
 14. Auditor uses export **for** compliance report or external audit
- END

Use Cases for Audit Logs:

- **Internal Audit:** Verify compliance officer actions are documented
- **Customs Audit:** Provide authorities with complete history of BOM changes, calculations, certificates
- **Security Incident:** Investigate unauthorized access or data modification
- **Dispute Resolution:** Trace who made a specific change and when

12. Development Roadmap

12.1 Phase 1: MVP (Months 1-6)

Objective: Launch core FTA management functionality for early adopters (SMB customers, pilot enterprise customers)

Sprint Breakdown (2-week sprints):

Sprints 1-2 (Weeks 1-4): Project Setup & Foundation

- Set up development environment (Git repo, CI/CD pipeline, staging/production environments)
- Define database schema, create initial migration scripts
- Set up cloud infrastructure (AWS/Azure, Kubernetes, databases)
- Implement authentication service (JWT, login/logout, password reset)
- Build basic frontend shell (React app, navigation, layout)

Sprints 3-4 (Weeks 5-8): User & Supplier Management

- Implement user management APIs (CRUD, role-based permissions)
- Build admin UI for user management
- Implement supplier management APIs (CRUD, CSV import)
- Build supplier management UI (list, create, edit, delete, import)
- Add search and filtering for suppliers

Sprints 5-6 (Weeks 9-12): Product & BOM Management

- Implement product management APIs (CRUD)
- Build product management UI (list, create, edit, delete)
- Implement BOM management APIs (hierarchical relationships, circular reference detection)
- Build BOM tree UI (expandable tree view, add/remove components)
- SAP integration (Phase 1): Basic import of suppliers, products, BOMs (manual trigger)

Sprints 7-8 (Weeks 13-16): Declaration System

- Implement declaration request APIs (create, send to supplier)
- Build declaration request UI (select supplier, products, send request)
- Implement supplier portal (token-based access, no login required)
- Build supplier declaration form (multi-step wizard, file upload)
- Implement email notification system (request sent, declaration submitted)
- Build declaration review UI for compliance officers

Sprints 9-10 (Weeks 17-20): Origin Calculation Engine

- Implement FTA rules database (pre-populate USMCA, CFTA, EU, ASEAN, CPTPP)
- Build origin calculation algorithm (transaction value and net cost methods)
- Implement calculation APIs (run calculation, get result)
- Build calculation UI (settings, trigger calculation, view results)
- Add BOM tree visualization with declaration status indicators
- Implement background job queue for complex calculations

Sprints 11-12 (Weeks 21-24): Certificate Generation

- Implement certificate generation APIs (create certificate, generate PDF)
- Build certificate templates for major FTAs (USMCA, CFTA, EU)
- Build certificate generation wizard UI (3-step: type, parties, review)
- Implement PDF generation and storage (S3)
- Build certificate management UI (list, view, download, revoke)
- Add email certificate to customer functionality

Sprint 13 (Weeks 25-26): Dashboard & Reporting

- Build dashboard with KPI cards (duty savings, certificates issued, compliance rate)
- Implement basic reporting (supplier performance, origin status breakdown)
- Add charts (line chart for savings trend, pie chart for FTA distribution)
- Build recent activity feed

Sprint 14 (Weeks 27-28): Audit Logging & Security Hardening

- Implement comprehensive audit logging (all CRUD operations, logins, calculations, certificates)
- Build audit log UI (filter, search, view changes, export)
- Security review and penetration testing (internal)
- Implement rate limiting, input validation improvements
- GDPR compliance: Data export, anonymization for deleted users

Sprint 15 (Weeks 29-30): Testing, Bug Fixes, Documentation

- Comprehensive testing (unit, integration, end-to-end)
- User acceptance testing (UAT) with pilot customers
- Bug fixes and performance optimization
- Write user documentation (help articles, video tutorials)
- API documentation (Swagger/OpenAPI)

Sprint 16 (Weeks 31-32): Launch Preparation & Deployment

- Final QA and regression testing
- Set up production monitoring and alerting
- Deploy to production environment
- Customer onboarding materials (welcome emails, onboarding guides)
- **LAUNCH MVP!** 🚀

Deliverables:

- Core features: User management, supplier management, product/BOM management, declaration portal, origin calculation, certificate generation
- Multi-tenant architecture operational
- SAP integration (basic import)
- Web app (desktop and mobile-responsive)
- Admin, compliance officer, supplier, and auditor roles functional
- Documentation and onboarding materials

Success Metrics:

- 10+ pilot customers onboarded
- 1,000+ declarations submitted via supplier portal
- 500+ certificates generated
- 95%+ uptime
- Average customer satisfaction score: 4+/5

12.2 Phase 2: Enhanced Features (Months 7-12)

Objective: Add advanced features, improve UX, expand market reach

Key Features:**Month 7:**

- **Advanced Analytics & Predictive Insights:**

- Duty savings forecasting (ML model predicting future savings based on historical data)
- Compliance risk scoring (identify products/suppliers at risk)
- “What-if” analysis tool (change supplier, see impact on origin calculation)
- **Multi-Language Support:**
- Spanish, French, German, Chinese
- Translated UI and email templates
- Language selection in user profile

Month 8:

- **Mobile Application (React Native):**
- iOS and Android apps for suppliers and compliance officers
- Push notifications for declaration requests, approvals
- Camera integration for document upload
- Offline mode with sync
- **Enhanced Supplier Portal:**
- Supplier login accounts (optional, for suppliers managing many parts)
- Bulk declaration upload via Excel template
- Supplier dashboard showing declaration history, pending requests

Month 9:

- **Advanced HS Code Classification Tool:**
- AI-assisted HS code suggestion based on product description (NLP)
- Integration with customs databases for code validation
- Image recognition for classification (upload product photo)
- **Customs Ruling Management:**
- Repository for advance rulings, binding tariff information
- Link rulings to products for consistent classification
- Alerts when rulings expire

Month 10:

- **Duty Drawback Management:**
- Track export transactions eligible for duty drawback
- Calculate drawback amounts
- Generate drawback claims documentation
- **API Marketplace & Integrations:**
- Pre-built connectors for Oracle ERP, Microsoft Dynamics
- Integration with customs brokerage systems (e.g., Descartes, C.H. Robinson)
- Webhook support for real-time event notifications
- Public API for custom integrations (developer documentation)

Month 11:

- **Enhanced Reporting & BI:**
- Custom report builder (drag-and-drop)
- Pre-built Power BI / Tableau templates
- Scheduled report generation and email delivery
- Export to Excel with pivot tables
- **User-Configurable Rules (Advanced):**
- Allow admins to define custom origin rules (stricter than FTA requirements)
- Rule builder UI for creating custom rules
- Validation against both standard and custom rules

Month 12:**- Supplier Collaboration Portal:**

- Real-time chat/messaging between compliance team and suppliers
- Shared document repository per supplier
- Workflow: Draft declaration → Submit for review → Approve/Request changes
- Supplier training resources and FAQs

- Performance Optimization:

- Database query optimization, indexing improvements
- Caching enhancements (Redis for more data types)
- Load testing and auto-scaling improvements

Deliverables:

- Mobile apps (iOS, Android)
- Multi-language support (5 languages)
- AI-powered HS code classification
- Advanced analytics and forecasting
- Duty drawback module
- API integrations (Oracle, Dynamics, customs brokers)
- Enhanced reporting and BI capabilities

Success Metrics:

- 50+ paying customers
- 20% of customers using mobile app
- 30% of declarations in non-English languages
- 50+ API integrations active
- Customer satisfaction score: 4.5+/5

12.3 Phase 3: Advanced Features & Scale (Months 13-24)

Objective: Differentiate from competitors, target enterprise market, achieve scale

Key Features:**Months 13-15:****- Blockchain-Based Certificate Verification:**

- Store certificate hashes on blockchain (Ethereum or Hyperledger)
- Public verification portal (enter certificate number, verify authenticity)
- Tamper-proof audit trail
- Integration with customs authorities' verification systems

- Advanced SAP Integration:

- Real-time bidirectional sync (webhooks, event-driven)
- Support for SAP SuccessFactors (employee data for labor value content)
- SAP Ariba integration (supplier network)

- AI-Powered Compliance Assistant:

- Chatbot for answering FTA compliance questions
- Intelligent suggestions for sourcing changes to improve origin
- Automated BOM anomaly detection (e.g., component costs changed significantly)

Months 16-18:**- Global Trade Content Service:**

- Automated FTA rule updates from official sources

- Daily regulatory change alerts
- Coverage for 50+ FTAs globally
- Integration with WCO (World Customs Organization) databases
- **Advanced Duty Optimization:**
- Support for Foreign Trade Zones (US), IMMEX (Mexico), Customs Warehousing (EU)
- Identify optimal duty suspension programs
- Simulate different trade scenarios for cost optimization
- **Enterprise-Grade Security & Compliance:**
- SOC 2 Type II certification
- ISO 27001 certification
- Single Sign-On (SSO) via SAML 2.0, Active Directory integration
- Advanced audit features (immutable logs, blockchain-backed)

Months 19-21:

- **White-Label & Reseller Platform:**
- Allow third parties to rebrand and resell FTA Manager
- Multi-level tenant hierarchy (resellers, sub-tenants)
- Reseller portal for managing customers
- Revenue sharing model
- **Industry-Specific Solutions:**
- Automotive module (USMCA labor value content tracking, steel/aluminum tracing)
- Textile & Apparel module (yarn-forward rules, preferential cut)
- Electronics module (semiconductor tracing, China processing trade)
- **Marketplace for Trade Services:**
- Connect customers with trade consultants, customs brokers, lawyers
- On-demand FTA compliance audits
- Certified origin specialist reviews

Months 22-24:

- **AI-Driven Sourcing Recommendations:**
- Analyze global supplier market, recommend suppliers in FTA regions
- Integration with supplier databases (Thomasnet, Alibaba)
- Cost-benefit analysis: Sourcing change cost vs. duty savings
- **Advanced Workflow Automation:**
- Custom workflow designer (drag-and-drop)
- Automate approval chains, escalations, notifications
- Integration with enterprise tools (Jira, ServiceNow for compliance tickets)
- **Platform Expansion:**
- Support for additional trade programs (GSP, preference programs)
- Tariff engineering tools (classification optimization)
- Trade war scenario modeling (e.g., tariff changes, FTA suspensions)

Deliverables:

- Blockchain certificate verification
- SOC 2 Type II and ISO 27001 certified
- AI-powered compliance assistant
- Global trade content service (50+ FTAs)
- White-label platform for resellers
- Industry-specific modules (automotive, textiles, electronics)
- Marketplace for trade services

Success Metrics:

- 200+ enterprise customers
 - 5+ white-label reseller partners
 - 99.95% uptime
 - 100,000+ certificates generated annually
 - Market leader in North American FTA management
-

13. Testing Strategy

13.1 Unit Testing

Scope: Test individual functions, methods, and components in isolation

Frameworks:

- **Frontend (React):** Jest + React Testing Library
- **Backend (Node.js/Python):** Jest (Node), Pytest (Python)

Coverage Target: 80%+ code coverage

Key Areas:

- **Utilities:** Date formatting, currency conversion, validation functions
- **Business Logic:** Origin calculation algorithm, RVC formula, tariff shift checks
- **API Endpoints:** Input validation, authorization checks
- **React Components:** Rendering, user interactions, state management

Example Test (Origin Calculation):

```
describe('Origin Calculation - Transaction Value Method', () => {
  it('should return qualifying status when RVC exceeds threshold', () => {
    const product = { id: 'uuid', total_value: 2450.00 };
    const bomComponents = [
      { value: 1102.50, originates: true },
      { value: 612.50, originates: true },
      { value: 490.00, originates: false },
      { value: 245.00, originates: false }
    ];
    const fta = { code: 'USMCA', rvc_threshold: 60.00 };

    const result = calculateOrigin(product, bomComponents, fta, 'transaction_value');

    expect(result.rvc_percentage).toBeCloseTo(70.0, 1);
    expect(result.qualifies).toBe(true);
  });
});
```

13.2 Integration Testing

Scope: Test interactions between components, API endpoints with database, service integrations

Frameworks:

- **API Testing:** Supertest (Node.js), Pytest + requests (Python)
- **Database:** Test database (PostgreSQL) with fixtures

Key Areas:

- **API Workflows:** Create supplier → Create product → Add BOM → Submit declaration → Calculate origin → Generate certificate
- **Database Transactions:** Ensure data integrity, rollback on errors
- **SAP Integration:** Mock SAP API responses, test data mapping
- **Email Notifications:** Mock email service, verify emails sent

Example Test (Declaration Submission API):

```
describe('POST /api/v1/declarations', () => {
  it('should create declaration and send notification', async () => {
    const supplierToken = 'valid_token_123';
    const declarationData = {
      declaration_request_id: 'uuid-req',
      product_id: 'uuid-prod',
      fta_code: 'USMCA',
      country_of_origin: 'US',
      origin_criterion: 'B',
      certifier_name: 'Bob Johnson',
      electronic_signature: 'Bob Johnson'
    };

    const response = await request(app)
      .post('/api/v1/declarations')
      .set('X-Portal-Token', supplierToken)
      .send(declarationData)
      .expect(201);

    expect(response.body.status).toBe('submitted');

    // Verify database record created
    const declaration = await db.declarations.findById(response.body.id);
    expect(declaration.supplier_id).toBe('uuid-supplier');

    // Verify email sent (mock check)
    expect(emailService.sendEmail).toHaveBeenCalledWith({
      to: 'compliance@company.com',
      subject: expect.stringContaining('New Declaration'),
      body: expect.any(String)
    });
  });
});
```

13.3 User Acceptance Testing (UAT)

Scope: Validate features meet business requirements from end-user perspective

Participants: Pilot customers, internal compliance team (acting as users)

Process:

1. **Test Plan:** Document user stories and acceptance criteria per feature
2. **Test Cases:** Create step-by-step test scenarios (e.g., "As a compliance officer, submit a declaration request")
3. **Execution:** Pilot users perform tasks in staging environment
4. **Feedback:** Collect feedback via surveys, interviews, bug reports
5. **Iteration:** Fix bugs, refine UX based on feedback
6. **Sign-Off:** Pilot customers approve feature as production-ready

Example UAT Test Case:

Feature: Supplier Declaration Submission

User Story: As a supplier, I want to submit an origin declaration via the portal

Test Case:

1. Receive email with portal link
2. Click link, verify portal opens without login
3. Review requested product **and** FTA
4. Fill form: Select country (Canada), origin criterion (B), enter producer name
5. Upload supporting PDF document (< 10MB)
6. Enter certifier name **and** type signature
7. Click "**Submit Declaration**"
8. Verify confirmation message displayed
9. Verify email confirmation received
10. Verify compliance officer receives notification email

Expected Results:

- All steps complete without errors
- Form validation prevents submission **if** required fields missing
- File upload succeeds, shows file name after upload
- Confirmation message clear **and** reassuring
- Emails sent within 1 minute

Pass/Fail: [Tester marks after execution]

13.4 Compliance Validation Testing

Scope: Ensure system correctly applies FTA rules and produces accurate origin determinations

Process:

1. **Test Data:** Create realistic test products with known origin status
2. **Manual Calculation:** Trade compliance expert manually calculates origin using FTA rules
3. **System Calculation:** Run same calculation in FTA Manager
4. **Comparison:** Verify system result matches manual calculation
5. **Edge Cases:** Test boundary conditions (exactly at RVC threshold, de minimis edge, tariff shift edge)

Example Compliance Test:

Product: Automotive Part (HS 8708.99)

FTA: USMCA

BOM:

- US-sourced steel: \$500 (originating)
- Canadian-sourced plastic: \$300 (originating)
- Chinese-sourced electronics: \$200 (non-originating)

Total Value: \$1,000

Manual Calculation:

RVC = $((1000 - 200) / 1000) \times 100 = 80\%$


USMCA Automotive Threshold: 75%

Result: QUALIFIES

System Calculation:

Input: Product ID, FTA=USMCA, Method=Transaction Value

Output: RVC=80%, Qualifies=True

Comparison:  PASS

Compliance Test Suite:

- Test all major FTAs (USMCA, CFTA, EU, ASEAN, CPTPP)
- Test both calculation methods (transaction value, net cost)
- Test all origin criteria (A, B, C, D)
- Test blanket declarations (validity period checks)
- Test missing declarations (should flag and assume non-originating)
- Test tariff shift rules (product-specific rules from Annex 4-B)

13.5 Performance Testing

Scope: Validate system meets performance benchmarks under load

Types:

- **Load Testing:** Simulate normal traffic (e.g., 1,000 concurrent users)
- **Stress Testing:** Simulate peak traffic (e.g., 5,000 concurrent users, 5x normal load)
- **Endurance Testing:** Sustained load over extended period (e.g., 24 hours) to detect memory leaks
- **Spike Testing:** Sudden traffic spikes (e.g., 0 to 5,000 users in 1 minute)

Tools:

- Apache JMeter, Gatling, K6, Locust

Test Scenarios:**1. API Load Test:**

- 1,000 virtual users making API calls (mix of GET, POST)
- Target: <300ms response time (p95), <1% error rate

2. Origin Calculation Load:

- 100 concurrent origin calculations (5-level BOM, 100 components each)
- Target: All complete within 10 seconds

3. Supplier Portal Load:

- 500 suppliers simultaneously submitting declarations
- Target: All submissions succeed, <5 seconds per submission

Performance Benchmarks (from Section 10.1):

- Web page load: <2s initial, <500ms subsequent
- API response: <300ms (p95)
- Origin calculation: <5s for 5-level BOM with 100 components

13.6 Security Testing

Scope: Identify vulnerabilities and ensure data protection

Types:

- **Penetration Testing:** Simulate attacks by ethical hackers
- **Vulnerability Scanning:** Automated scans for known vulnerabilities (OWASP Top 10)
- **Authentication Testing:** Attempt to bypass login, access unauthorized resources
- **Injection Testing:** SQL injection, XSS, command injection attempts
- **Data Security Testing:** Verify encryption at rest and in transit, test tenant isolation

Tools:

- **Automated:** OWASP ZAP, Burp Suite, Nessus
- **Manual:** Third-party penetration testing firm (annually)

Security Test Cases:

- **SQL Injection:** Attempt to inject SQL in form fields (should be blocked by parameterized queries)

- **Cross-Tenant Access:** User from Tenant A tries to access Tenant B's data (should fail with 403/404)
- **Weak Password:** Attempt to create account with weak password (should be rejected)
- **Session Hijacking:** Steal JWT token, use from different IP (implement IP binding or anomaly detection)
- **CSRF Protection:** Attempt cross-site request forgery (ensure CSRF tokens used)

Remediation:

- Fix identified vulnerabilities within 30 days (high severity), 90 days (medium)
- Re-test after fixes
- Document fixes in security log

14. Deployment & DevOps

14.1 CI/CD Pipeline

Continuous Integration (CI):

Every code commit triggers automated build and test process

CI Pipeline Stages:

1. **Code Commit:** Developer pushes code to Git (GitHub/GitLab)
2. **Trigger Build:** GitHub Actions / GitLab CI detects commit
3. **Install Dependencies:** `npm install` or `pip install -r requirements.txt`
4. **Linting:** ESLint (frontend), Pylint (backend) - enforce code standards
5. **Unit Tests:** Run all unit tests, generate coverage report
6. **Build Application:** Compile React app (`npm run build`), package backend
7. **Code Quality:** SonarQube scan for code smells, bugs, security issues
8. **Artifacts:** Upload build artifacts (Docker images) to registry

Continuous Deployment (CD):

Automated deployment to staging and (optionally) production

CD Pipeline Stages:

1. **Deploy to Staging:** Automatically deploy every successful build to staging environment
2. **Integration Tests:** Run integration and end-to-end tests against staging
3. **Smoke Tests:** Quick sanity checks (login works, critical APIs respond)
4. **Manual Approval (for Production):** Product manager or tech lead approves deployment
5. **Deploy to Production:** Kubernetes rolling update (zero-downtime deployment)
6. **Health Checks:** Monitor application health, rollback if errors detected
7. **Notifications:** Slack/email notification on deployment success/failure

CI/CD Tool: GitHub Actions (recommended) or GitLab CI

Example GitHub Actions Workflow (.github/workflows/ci-cd.yml):

```

name: CI/CD Pipeline

on:
  push:
    branches: [main, develop]
  pull_request:
    branches: [main]

jobs:
  build-and-test:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - name: Set up Node.js
        uses: actions/setup-node@v3
        with:
          node-version: '18'
      - name: Install dependencies
        run: npm install
      - name: Lint
        run: npm run lint
      - name: Run unit tests
        run: npm test -- --coverage
      - name: Build
        run: npm run build
      - name: Build Docker image
        run: docker build -t ftamanager/webapp:${{ github.sha }} .
      - name: Push to Docker Hub
        run: docker push ftamanager/webapp:${{ github.sha }}

  deploy-staging:
    needs: build-and-test
    if: github.ref == 'refs/heads/develop'
    runs-on: ubuntu-latest
    steps:
      - name: Deploy to staging
        run: kubectl set image deployment/webapp webapp=ftamanager/webapp:${{ github.sha }} -n staging
      - name: Run integration tests
        run: npm run test:integration -- --env=staging

  deploy-production:
    needs: build-and-test
    if: github.ref == 'refs/heads/main'
    runs-on: ubuntu-latest
    environment: production
    steps:
      - name: Deploy to production
        run: kubectl set image deployment/webapp webapp=ftamanager/webapp:${{ github.sha }} -n production
      - name: Verify deployment
        run: kubectl rollout status deployment/webapp -n production

```

14.2 Environment Strategy

Environments:

1. Development (Local):

- Developer's local machine
- Docker Compose for local services (database, Redis, etc.)

- Mock external services (SAP, email)
- Hot-reloading for fast iteration

2. Staging:

- Cloud-hosted environment mirroring production
- Kubernetes cluster (smaller instance types)
- Separate database (same schema as production)
- Real integrations (SAP sandbox, test email accounts)
- Purpose: Integration testing, UAT, demo to customers
- Deployed automatically on every commit to `develop` branch

3. Production:

- Cloud-hosted production environment
- Kubernetes cluster (auto-scaling, multi-AZ)
- Production database with backups, replication
- Real integrations (customer SAP systems, production email)
- Deployed manually (or automatically with approval) on commits to `main` branch
- Blue-Green or Canary deployment for zero downtime

Environment Configuration:

- **Environment Variables:** Stored in Kubernetes Secrets or AWS Secrets Manager
- **Config Files:** Separate configs per environment (`config.dev.js` , `config.staging.js` , `config.prod.js`)
- **Feature Flags:** LaunchDarkly or custom feature flag service to toggle features per environment

Environment Parity:

- Staging should closely match production (same OS, database version, Kubernetes version)
- Use Infrastructure as Code (Terraform) to ensure consistency

14.3 Monitoring & Logging

Application Performance Monitoring (APM):

- **Tool:** Datadog, New Relic, or Grafana + Prometheus
- **Metrics:**
 - Request rate, error rate, latency (RED metrics)
 - CPU, memory, disk usage per service
 - Database query performance (slow queries, connection pool usage)
 - Background job queue depth and processing time
 - Custom metrics: Origin calculations/minute, certificates generated/day

Logging:

- **Centralized Logging:** ELK Stack (Elasticsearch, Logstash, Kibana) or CloudWatch Logs
- **Log Levels:** DEBUG, INFO, WARN, ERROR
- **Structured Logs:** JSON format for easy parsing

```
json
{
  "timestamp": "2026-01-15T14:30:00Z",
  "level": "INFO",
  "service": "origin-calculation",
  "message": "Calculation completed",
  "tenant_id": "uuid",
  "product_id": "uuid",
```

```
"duration_ms": 2340,
"result": "qualifying"
}
```

- **Log Retention:** 30 days in hot storage, 1 year in cold storage (S3)

Alerting:

- **Tool:** PagerDuty, Opsgenie, or built-in cloud monitoring alerts
- **Alert Rules:**
 - Error rate >1% for 5 minutes → Page on-call engineer
 - API latency p95 >1s for 5 minutes → Slack alert
 - Database CPU >80% for 10 minutes → Email alert
 - Disk usage >90% → Critical alert
 - SAP sync job failed → Email to admin
- **On-Call Rotation:** Designate on-call engineer for 24/7 support (enterprise tier)

Dashboards:

- **Operations Dashboard:** Real-time system health (request rate, errors, latency, resource usage)
- **Business Metrics Dashboard:** Certificates generated, declarations submitted, active users, duty savings calculated
- **Custom Tenant Dashboards:** Per-tenant metrics for large enterprise customers

14.4 Infrastructure Recommendations

Cloud Provider: AWS (recommended for maturity and service breadth); Azure or GCP acceptable

Compute:

- **Kubernetes (EKS/AKS/GKE):** Container orchestration for microservices
- Namespace per environment (staging, production)
- Auto-scaling: Horizontal Pod Autoscaler based on CPU/memory
- Node pools: Separate pools for web servers vs. background workers
- **Alternative:** AWS ECS (Elastic Container Service) if Kubernetes overhead not justified initially

Database:

- **Primary Database:** AWS RDS for PostgreSQL (Multi-AZ for high availability)
- Instance size: Start with db.t3.medium, scale to db.r5.large+ for production
- Automated backups (daily), 30-day retention
- Read replicas for reporting queries
- **Caching:** AWS ElastiCache for Redis
- Instance size: cache.t3.medium, scale as needed
- **Search:** Amazon OpenSearch (managed Elasticsearch) or self-hosted Meilisearch

Storage:

- **Object Storage:** AWS S3 for documents (certificates, declarations)
- Lifecycle policies: Move to S3 Glacier after 1 year
- Versioning enabled for critical buckets
- Server-side encryption (SSE-S3 or SSE-KMS)

Networking:

- **Load Balancer:** AWS ALB (Application Load Balancer) or NGINX Ingress Controller (Kubernetes)
- **CDN:** CloudFront for static assets
- **VPC:** Isolated network with public and private subnets
- Public: Load balancers, NAT gateways

- Private: Application servers, databases
- **Security Groups:** Firewall rules restricting access (e.g., database only accessible from app servers)

DNS & SSL:

- **DNS:** Route 53 (AWS) or CloudFlare
- **SSL Certificates:** Let's Encrypt (free, auto-renewing) or AWS Certificate Manager

Disaster Recovery:

- **Multi-Region:** Primary in us-east-1, failover in us-west-2
- **Database:** Cross-region replication (RDS read replica in secondary region)
- **S3:** Cross-region replication for critical buckets

Infrastructure as Code (IaC):

- **Terraform:** Define all infrastructure (VPC, subnets, RDS, EKS, S3, etc.) in code
- **Version Control:** Store Terraform configs in Git
- **State Management:** Terraform state stored in S3 with locking (DynamoDB)

Example Terraform (simplified):

```

provider "aws" {
  region = "us-east-1"
}

resource "aws_vpc" "main" {
  cidr_block = "10.0.0.0/16"
}

resource "aws_db_instance" "postgres" {
  identifier      = "ftamanager-db"
  engine          = "postgres"
  engine_version  = "15.3"
  instance_class  = "db.t3.medium"
  allocated_storage = 100
  storage_encrypted = true
  multi_az        = true
  username         = var.db_username
  password         = var.db_password
  backup_retention_period = 30
}

resource "aws_eks_cluster" "main" {
  name     = "ftamanager-cluster"
  role_arn = aws_iam_role.eks_cluster.arn

  vpc_config {
    subnet_ids = aws_subnet.private[*].id
  }
}

resource "aws_s3_bucket" "documents" {
  bucket = "ftamanager-documents"

  versioning {
    enabled = true
  }

  server_side_encryption_configuration {
    rule {
      apply_server_side_encryption_by_default {
        sse_algorithm = "AES256"
      }
    }
  }
}

```

14.5 Deployment Process (Production)

Steps:

1. **Code Review:** All code changes reviewed via pull request (GitHub/GitLab)
2. **Merge to Main:** Approved PRs merged to `main` branch
3. **CI Pipeline:** Automated build and tests run
4. **Tag Release:** Create Git tag (e.g., `v1.2.0`) for versioning
5. **Build Docker Images:** Tagged images built and pushed to registry
6. **Deploy to Staging:** Kubernetes deployment updated with new image
7. **Smoke Tests:** Automated smoke tests run against staging
8. **Manual QA (Optional):** QA team validates critical flows in staging
9. **Approval:** Product manager approves deployment to production
10. **Deploy to Production:** Kubernetes rolling update

- Update deployment manifest with new image tag
 - `kubectl apply -f deployment.yaml`
 - Kubernetes gradually replaces old pods with new ones (zero downtime)
11. **Monitor:** Watch metrics and logs for errors
12. **Rollback (if needed):** `kubectl rollout undo deployment/webapp` to revert

Deployment Strategies:

Rolling Update (Default):

- Gradually replace old pods with new ones
- Ensures zero downtime
- Easy rollback

Blue-Green Deployment:

- Run two identical production environments (Blue=current, Green=new)
- Deploy new version to Green environment
- Switch traffic from Blue to Green (instant switchover)
- Keep Blue environment for quick rollback
- More complex, requires double resources

Canary Deployment:

- Deploy new version to small subset of servers (e.g., 10%)
- Monitor for errors
- Gradually increase traffic to new version (20%, 50%, 100%)
- Rollback if errors detected
- Reduces risk of widespread issues

Recommendation for FTA Manager:

- **Phase 1:** Rolling updates (simple, effective)
- **Phase 2+:** Canary deployments for critical updates (reduces risk)

Conclusion

This comprehensive development planning document provides a detailed roadmap for building a world-class FTA Management Web Application. It covers all critical aspects from system architecture and feature specifications to security, testing, and deployment strategies.

Key Takeaways:

1. **Multi-Tenant Architecture:** Enables efficient scaling to serve hundreds of companies on a shared platform while maintaining data isolation and security
2. **User-Centric Design:** Prioritizes ease of use for both compliance officers managing complex trade rules and suppliers submitting declarations with minimal friction
3. **Automated Intelligence:** Origin calculation engine automates complex multi-level BOM analysis, saving hours of manual work and reducing compliance errors
4. **Scalability & Performance:** Cloud-native architecture with Kubernetes, auto-scaling, and caching ensures the system can grow from 10 to 10,000+ companies

5. **Compliance-First:** Comprehensive audit logging, GDPR compliance, and accurate FTA rule implementation ensure the system meets regulatory requirements
6. **Phased Approach:** 6-month MVP focuses on core features, with 12-24 month roadmap for advanced capabilities and market differentiation

Next Steps:

1. **Finalize Technology Stack:** Choose between Node.js vs. Python, AWS vs. Azure based on team expertise and budget
2. **Assemble Development Team:** Hire or assign full-stack developers, DevOps engineer, QA engineer, trade compliance expert
3. **Set Up Infrastructure:** Provision cloud environments, databases, CI/CD pipeline
4. **Begin Sprint 1:** User authentication and basic frontend shell (Week 1-2)
5. **Engage Pilot Customers:** Recruit 5-10 early adopters for UAT and feedback
6. **Iterate and Launch:** Build, test, refine, and launch MVP in 6 months

Success Factors:

- **Executive Support:** Ensure leadership commitment and resources
- **Customer Feedback:** Continuously engage customers throughout development
- **Agile Methodology:** Embrace iterative development, adapt to changing requirements
- **Quality Focus:** Invest in testing and code quality from day one
- **Trade Expertise:** Partner with trade compliance professionals to ensure accuracy

With this plan as a foundation, your development team is well-equipped to build a competitive, scalable, and compliance-focused FTA management solution that delivers real value to importers, exporters, and suppliers worldwide.

Document Version: 1.0

Last Updated: January 15, 2026

Prepared By: FTA Manager Development Team

Contact: dev@ftamanager.com