Expert Review of the Comprehensive Requirements Document

System Architect

September 16, 2025

Introduction

As a seasoned system architect with over 40 years of experience in designing global trade compliance platforms, including contributions to early iterations of systems like SAP GTS and Oracle GTM, I commend the thoroughness of this document. It effectively outlines functional requirements for a scalable, AI-enhanced export/import license determination and compliance platform targeted at mid-market and SME users. The focus on automation, regulatory alignment (e.g., EAR, OFAC, EU sanctions), and cost reduction (aiming for 40% savings) is well-articulated, drawing apt comparisons to established solutions like Descartes and Thomson Reuters ONESOURCE. Strengths include detailed data models, process workflows, and innovation suggestions, which provide a solid foundation for development.

However, as a best-in-class platform, the document lacks depth in several critical non-functional, technical, and operational areas. These omissions could lead to challenges in scalability, maintainability, security, and real-world deployment. Below, I identify key gaps and provide guidance on enhancements, organized by category. My recommendations are based on lessons from deploying similar systems across 200+ jurisdictions, where overlooking these aspects often results in rework, compliance risks, or poor adoption.

1 Non-Functional Requirements

The document touches on performance indirectly (e.g., rapid alert resolution <2 min in UX) but lacks comprehensive non-functional specs, which are essential for ensuring the platform handles SME growth (e.g., from 100 to 10,000 transactions/day) without degradation.

1.1 Gaps Identified

- No explicit performance metrics (e.g., response times for rule evaluations or API calls).
- Scalability not addressed for concurrent users or data volume (e.g., handling 1M+ master data records).

- Reliability aspects like uptime SLAs, fault tolerance, and load balancing are missing.
- Usability metrics beyond basic UX (e.g., error rates in AI classifications).

1.2 Guidance

Add a dedicated section (e.g., "12 Non-Functional Requirements") with quantifiable targets:

- **Performance:** Rule engine must process license determinations in <500ms for 95% of queries; OCR extraction in <5s per document.
- **Scalability:** Horizontal scaling via microservices; support auto-scaling for peak loads (e.g., end-of-quarter customs filings).
- **Reliability & Availability:** 99.99% uptime SLA; implement redundant databases and failover mechanisms.
- **Backup & Disaster Recovery:** Automated daily backups with RPO <1 hour and RTO <4 hours; geo-redundant storage for multi-jurisdiction compliance.
- **Monitoring:** Integrate metrics for system health (e.g., CPU/memory usage, error rates) using tools like Prometheus.

This ensures the platform meets enterprise-grade standards while remaining SME-affordable.

2 Technology Stack Preferences

The document mentions AI/ML (e.g., for classifications) and integrations (e.g., APIs) but provides no guidance on underlying technologies, leading to potential mismatches in development or vendor lock-in.

2.1 Gaps Identified

- No preferences for programming languages, frameworks, databases, or cloud providers.
- AI/ML tools unspecified beyond "AI-assisted" features.
- Deployment options (cloud vs. on-prem) not clarified, despite SME needs for flexibility.
- No consideration for open-source vs. proprietary components to control costs.

2.2 Guidance

Incorporate a "Technology Stack" subsection under "7 Integration and Automation" or as a new section:

- **Backend:** Python or Java for rule engines (e.g., Drools for logic); Node.js for real-time integrations.
- **Databases:** PostgreSQL for relational data (master data, audits); MongoDB for document storage; Redis for caching quotas/sanctions lists.
- **AI/ML:** Use TensorFlow or Hugging Face for NLP in end-user statements; scikit-learn for predictive risk scoring.
- **Cloud/On-Prem:** Hybrid model with AWS/Azure for cloud (SMEs prefer pay-as-you-go); Kubernetes for containerization to enable on-prem deployments.
- **Frontend:** React.js for dashboards; ensure mobile responsiveness for SME field users.

Prioritize technologies that support low-code customization (e.g., Camunda for workflows) to align with SME needs.

3 API Specifications

While integrations are mentioned (e.g., ERP sync via JSON/XML), details are superficial, risking integration failures or security vulnerabilities.

3.1 Gaps Identified

- No endpoint definitions, request/response schemas, or authentication methods.
- Error handling, rate limiting, and versioning not specified.
- Limited data formats (only JSON/XML/CSV; no mention of EDI standards like X12 for customs).

3.2 Guidance

Expand "7.2 Data Model" to include an "API Design" subsection:

- **Endpoints:** Define RESTful APIs (e.g., POST /v1/license/determine with params: productID, destinationCountry; response: {licenseRequired: boolean, reason: string}).
- **Authentication:** OAuth 2.0 with JWT; role-based access control (RBAC) integrated with user roles.
- **Error Handling:** Standardized codes (e.g., 429 for rate limits at 100 req/min per API key); detailed messages for debugging.
- **Rate Limiting & Versioning:** Throttle based on subscription tiers; use semantic versioning (e.g., /v1, /v2) for backward compatibility.
- **Data Formats:** Support EDI 810/856 as noted, plus protobuf for high-volume data.

Include Swagger/OpenAPI specs for developer documentation to facilitate third-party integrations.

4 Deployment and DevOps

No mention of how the platform will be built, tested, or maintained, which is crucial for agile development and zero-downtime updates in a regulatory environment.

4.1 Gaps Identified

- Absence of CI/CD, testing, or monitoring strategies.
- No containerization or orchestration details.
- Operational aspects like logging aggregation missing beyond basic audit trails.

4.2 Guidance

Add a new section "13 Deployment and Operations":

- **CI/CD Pipelines:** Use GitHub Actions or Jenkins for automated builds/tests/deployments; include compliance checks in pipelines.
- **Testing Frameworks:** Unit tests for rule logic (e.g., JUnit); integration tests for ERP sync; load testing with JMeter for scalability.
- **Containerization:** Docker for services; Kubernetes/Helm for orchestration.
- **Monitoring & Logging:** ELK Stack (Elasticsearch, Logstash, Kibana) for centralized logs; integrate with audit trails for real-time alerts on anomalies (e.g., sanction hits).
- **Zero-Downtime Upgrades:** Blue-green deployments for rule updates; canary releases for new features.

This minimizes risks during regulatory changes, ensuring seamless updates.

5 User Roles and Security

User roles are briefly noted (e.g., Officer, Approver), but security is limited to GDPR/CCPA compliance in storage.

5.1 Gaps Identified

- Granular permissions not defined (e.g., view-only vs. edit for sensitive data).
- No authentication standards, session management, or threat modeling.
- Missing encryption details for data in transit/rest.

5.2 Guidance

Enhance “5.2 Data Model” and add to “8 Audit, Reporting, and Compliance”:

- **Granular Permissions:** Extend RBAC (e.g., Compliance Officer: edit rules; Auditor: read-only logs); use attribute-based access control (ABAC) for jurisdiction-specific access.
- **Authentication/Authorization:** Multi-factor authentication (MFA); SAML/OIDC for SSO; session timeouts at 15 min.
- **Security Measures:** End-to-end encryption (TLS 1.3 for transit, AES-256 for rest); regular penetration testing; vulnerability scanning with OWASP ZAP.
- **Threat Modeling:** Include STRIDE analysis for risks like data breaches in sanction screening.

6 Change Management

Regulatory updates are mentioned (e.g., quarterly reviews), but no process for system-wide changes.

6.1 Gaps Identified

- No protocols for updating rules/master data without disrupting operations.
- System upgrades and rollback strategies absent.

6.2 Guidance

In “3.3 Process Workflow” and “6.3 Process Workflow,” add:

- **Rule/Master Data Updates:** Automated versioning with approval workflows; shadow testing for new rules.
- **System Upgrades:** Scheduled maintenance windows; automated rollbacks if issues detected.
- **Impact Assessment:** Require change impact reports for regulatory adaptations.

7 Data Privacy and Compliance Beyond GDPR/CCPA

Limited to basic mentions; ignores region-specific laws.

7.1 Gaps Identified

- No data residency requirements (e.g., for EU data in EU servers).
- Missing handling for laws like China’s PIPL or Brazil’s LGPD.

7.2 Guidance

Expand “6.2 Data Model”:

- **Privacy Enhancements:** Data minimization principles; pseudonymization for customer data.
- **Residency:** Configurable data localization (e.g., AWS regions per jurisdiction).
- **Compliance Mapping:** Table of laws (e.g., PIPL: consent management; LGPD: data subject rights workflows).

8 Localization and Internationalization

Multi-language UI is noted, but incomplete.

8.1 Gaps Identified

- No support for right-to-left languages or localized reports/documents.

8.2 Guidance

In “9.1 Key Features”:

- **Full I18n:** Unicode support; locale-specific date/currency formats; translate reports/documents via AI (e.g., Google Translate API integration).

9 Extensibility and Custom Plugins

Customization is covered, but not third-party extensions.

9.1 Gaps Identified

- No plugin architecture for custom logic.

9.2 Guidance

In “5.1 Key Features”:

- **Plugins:** SDK for custom modules (e.g., JavaScript plugins for reports); marketplace for pre-built extensions.

10 Training and Documentation

Briefly mentioned in innovations (gamified training), but lacks plans.

10.1 Gaps Identified

- No end-user/admin/developer docs.

10.2 Guidance

Add to “9 User Experience and Support”:

- **Training:** On-demand videos, interactive simulations; certification paths.
- **Documentation:** API docs via Postman; admin guides in PDF/wiki; in-app contextual help.

Conclusion

Incorporating these enhancements will elevate the platform to truly best-in-class status, ensuring robustness for SMEs while mitigating risks. I recommend prioritizing non-functional and security aspects in the next revision, as they underpin all functional features. If needed, I can provide sample diagrams or prototypes for these additions.