okay

can you pelase do a full fledged research on below
Tech/Non-Functional
Detailed SLAs, high-availability, automated testing, extensibility SDK

Here is a deep-dive, best-practices research and requirements analysis for **Tech/Non-Functional aspects** of a next-gen compliance SaaS platform, including SLAs, high-availability, automated testing, and extensibility SDKs.

Tech/Non-Functional Requirements – Full-Fledged Research

1. Service Level Agreements (SLAs) & Performance

- **Availability**: 99.99% uptime SLA, measured monthly, with financial credits for missed SLAs. Requires multi-AZ cloud deployments, load balancers, and database failover.
- **Response Time**: Core APIs (e.g., screen order, license check) should respond within 400–800ms for 95% of requests under normal load.
- **Throughput & Scale**: System must handle >10,000 concurrent users and >1,000,000 transactions/day per tenant without degradation.
- **Capacity Planning**: Auto-scaling microservices and elastic DBs ensure scaling for end-of-quarter or regulatory surge events.

2. Reliability, Disaster Recovery, Monitoring

- **High-Availability Architecture**: Geo-redundant, containerized (Kubernetes) deployments, managed failover, and stateless services.
- **Disaster Recovery (DR)**: RPO ≤1 hour (max data loss acceptable), RTO ≤4 hours (max downtime post-disaster); regular failover/fire drill testing.
- **Daily Backups**: Automated encrypted backups, multi-region replication, self-service restore.
- **Health Monitoring**: Prometheus/Grafana, ELK dashboards for latency, error rate, throughput, failed integrations, system load. Alert escalation to on-call teams.

3. Automated Testing, CI/CD, and Quality

- **CI/CD Pipelines**: Automated pipelines (e.g., GitHub Actions, Jenkins) for build, unit/integration/test/deploy stages, with rollback support.
- **Test Coverage**: 90%+ unit test coverage on rule and critical logic. End-to-end tests for licensing, classification, document flows.
- **Integration & Load Testing**: JMeter/Gatling used regularly for simulating peak loads. Mock environments for connected ERP/OMS platforms.
- **Zero-Downtime Releases**: Blue-green/canary deployments. Release managers can safely roll out or roll back changes with no user impact.

4. Extensibility & SDKs

- **Plugin Architecture**: SDK available (Python, JS/Node, Java options) for adding custom rules, UI widgets, report templates, or connectors.
- **Marketplace/Plugin Registry**: Central store for sharing, vetting, and auto-updating vetted extensions (e.g., custom sanctions, third-party data).
- **Event Bus/Webhooks**: Allow triggers for workflow, alerts, and automation steps to enable low-code integration with external SaaS and on-prem tools.
- **OpenAPI/Swagger Docs**: All public APIs documented, example integrations provided, versioned for forward/backward compatibility.

5. Security, Compliance, and Privacy

- **End-to-End Encryption**: TLS 1.3 for all flows, AES-256 at rest, KMS or BYOK options for major tenants.
- **Penetration Testing & Vulnerability Scanning**: Quarterly, plus on each major release. OWASP Top 10 and regulatory-specific threat mitigations.
- **Change/Config Management**: All config changes (rules, quotas, integrations) logged, reviewable, and governed via approval workflow.

6. Usability/Operational Readiness

- **Operational Dashboards**: Real-time health, release, and SLA performance shown to both vendor and customer admins.
- **Self-Service, No-Vendor Required**: Admins can restart services, trigger DR, update plugins, or spin off test sandboxes without waiting for vendor involvement.

7. Cost/Resource Efficiency

- **Multi-Tenant Support**: Logical/cost isolation, with optimized resource sharing, so SMEs get “enterprise grade” performance at low TCO.
- **Resource Optimization**: Idle resource scaling and predictive burst handling reduce cloud/storage/bandwidth costs.

Acceptance Criteria/Benchmarks

- All core user journeys (screening, license, doc search, reporting, data entry) 99.9% within published SLAs.
- 95% of new critical code published via CI/CD is fully covered by automated tests.
- 100% of customer-impacting config changes logged and auditable.
- No-customer-impact zero-downtime upgrades proven in production.
- Marketplace/SDK/Plugin APIs tested and documented with at least 3 open extensions on launch.

Why This Level is Critical

- Missing these areas in early-stage design leads to project risks, rework, scaling bottlenecks, integration failure, and compliance breaches.
- Modern buyers (even SME) expect visibility into uptime and rollback, not just features. TCO, speed to market, audit evidence, and “freedom to DIY” are now differentiators.

Would you like any of these non-functional themes converted into more formal user stories or acceptance criteria for your platform documentation?Here is a detailed, best-practices-driven research summary for Tech/Non-Functional requirements for compliance SaaS—focused on SLAs, high-availability, automated testing, and extensibility SDKs. These principles and story elements are based on RegTech system architect reviews, multiple market studies, and deployment models for scalable compliance platforms.

Tech/Non-Functional Requirements – Detailed Research

SLAs and Performance

- Publish clear SLAs to users: 99.99% uptime (multi-region failover, geo-redundant backups), and response times for core APIs/rule checks under 800ms for 95% of queries.
- High throughput: system must support 10,000+ concurrent users and 1M+ transactions per day per tenant with no degradation.
- Metrics and automated alerting: dashboards for latency, retried jobs, error rates, slowest endpoints, resource exhaustion, etc.

High Availability, Scalability, and Recovery

- Use auto-scaling microservices/containerization (Kubernetes/Docker) for web/app layers, cloud-managed DB clusters (e.g., Aurora/Cloud SQL/NoSQL), and stateless API gateways for dynamic resource scaling.
- Multi-AZ or multi-cloud failover; RPO ≤1hr, RTO ≤4hr; documented regular DR fire drills.
- User-facing health dashboards and proactive uptime SLAs are public.

Automated Testing and CI/CD

- 90% unit test coverage for all business-critical logic and APIs.
- Automated integration, regression, and load test coverage with CI/CD pipeline (GitHub Actions/Jenkins/Gitlab).
- Blue-green/canary deployments for safe, zero-downtime releases, plus change rollback mapped to config/version control, not just code.
- Security and vulnerability scanning as part of every build (OWASP ZAP, Snyk, etc).

Extensibility, APIs and SDKs

- Plugin/extension SDK (Python or JavaScript first, expand to Java/.NET) for user-built rule modules, UI widgets, connectors, or reporting add-ons.
- Publicly documented, versioned OpenAPI/Swagger for REST APIs, with sample clients/QuickStart guides for popular languages.
- Event/webhook framework for automation/bots/apps/integration in ERP/OMS or customer-embedded UIs.
- Authentication built around OAuth2/SAML, API key management, and permission scopes per client/app.

Cost/Resource Optimization and Accessibility

- Multi-tenant design and cost budgeting for SMEs as well as enterprise: horizontal scale-out with quota and abuse controls.
- Self-service sandbox/test environments; admin control for restarts, plugin updates, resource scaling, backup/restore, etc.—no vendor ticketing for basic ops.
- Resource monitoring and policy-based auto-scale/auto-suspend for cost efficiency and regulatory fit.

Acceptance/Benchmark Examples

- 100% of transaction journals, config updates, and deploys immutable and audit-logged.
- All critical APIs, workflows, and plugin events covered by unit/integration and load tests in CI/CD.
- Average real-user workflow (screen, classify, license, audit) within 1s on 99% of activity for 98% of customers.
- 3+ open-source SDK/plugin samples live at launch; customer extensions published to a curated marketplace.

Rationale:
These requirements prevent scale-out failures, reduce post-launch ops costs, and ensure the platform attracts integrators and ISVs, not just end-users. Projects omitting these non-functionals face growth and reliability bottlenecks—not just in the enterprise but even in SME cloud compliance.

If you want, I can break these down into explicit user stories or acceptance criteria for each non-functional area—let me know your preference!

⁂

1. Expert-Review-of-the-Comprehensive-on-ths-doc.pdf
2. ComprehensiveRequirementsfortheGlobalTradeLicenseHub-Deep-seek.pdf
3. licenseiq_architecture_and_flow.pdf
4. licenseiq_architecture.pdf
5. ComprehensiveRequirementsforaBest-in-ClassExportand-Groke.pdf
6. Consolidated_RegTech_Market_Analysis_Detailed.pdf
7. Export-License-Determination-and-Sanctions-Screening-Software.pdf