Technical Requirements - Invoice Processing Automation

System Architecture

Our current invoice processing system is built on a **monolithic architecture** running on-premises. The application is written in Java (Spring Boot) and uses Oracle Database for data persistence. We have approximately 15 different systems that need to integrate with the invoice processing solution.

Core Technologies

- **Application Server**: WebLogic 12c
- **Database**: Oracle 19c Enterprise Edition
- **Frontend**: Angular 12
- **Messaging**: IBM MQ for asynchronous processing
- **File Storage**: Network-attached storage (NAS)

Integration Landscape

The invoice processing system must integrate with: - SAP ERP for purchase orders and vendor master data - Workday for employee approvals - DocuSign for electronic signatures - Email systems for invoice receipt - Banking systems for payment processing

We currently use **REST APIs** for most integrations, with some legacy systems still using SOAP. Average invoice processing volume is 50,000 invoices per month, with peaks during month-end reaching 5,000 invoices per day.

Data Infrastructure

Invoice data is stored in Oracle database with the following characteristics: - Structured data: PO numbers, amounts, dates, vendor IDs - Unstructured data: PDF invoices, email attachments - Data retention: 7 years for compliance

We handle **PII data** including vendor contact information and bank account details. All data must remain within Australia due to data residency requirements.

Security Posture

Current security measures include: - Active Directory for authentication - Role-based access control (RBAC) for application access - TLS 1.2 for data in transit - Database encryption at rest using Oracle TDE - Annual penetration testing

Performance Requirements

- Invoice upload response time: < 3 seconds

- Three-way matching completion: < 30 seconds per invoice
- System availability: 99.5% during business hours (7am-7pm AEST)
- Disaster recovery: 24-hour RTO, 4-hour RPO

Development & Deployment

- Version control: Git (Bitbucket)
- CI/CD: Jenkins pipelines for automated builds
- Environments: Dev, UAT, Production
- Deployment frequency: Monthly releases
- Testing: Manual UAT, limited automated testing

Note: This document covers current state technical architecture. Missing information includes cloud migration plans, AI/ML experience, observability tooling details, and auto-scaling capabilities.