1. System Idea

We need a system that can take **SDS PDFs**, extract metadata, manage different versions, and make searching super fast and easy.

Goals:

- Fast uploads
- Quick search
- Version management without headaches
- Multi-tenant + role-based access

---

2. Components & How It Works

a. Upload API

- Upload PDFs up to 25MB
- Two ways:
    - Presigned URL → client uploads directly to storage
    - Direct POST → API receives file and stores it
- AuthN: OIDC
- AuthZ: RBAC + tenant scope
- Validates (tenantId, supplier, SKU, docType, version)

b. Object Storage

- Stores raw PDFs
- Versioned by (tenantId, supplier, SKU, docType, version)
- Keeps last N versions, archives older ones

c. Event-Driven Pipeline

- **Queue**: triggered on upload
- **Extractor**: runs OCR/ML, generates metadata JSON
- **Metadata DB**: stores structured metadata
- **Indexer**: pushes data to search engine (Elastic/OpenSearch) for fast search

d. Search API

- Search by SKU, supplier, date ranges
- Supports latest-version only queries
- Reads from index for faster results

e. Observability

- Centralized logs and traces
- Dashboards for upload speed
- Alerts for failures

---

3. Trade-Offs

Requirement	Our Choice	Trade-Off
Fast search	Elastic/OpenSearch	Extra storage & indexing cost
High availability	Queue + multiple extractors	Slightly slower ingestion
Multi-tenant	Tenant-aware metadata	More complex access control
Retention	Archive old versions	Extra storage for backups

---

4. SLAs & Performance

- Upload: 95% ≤ 5s (excluding OCR)
- Search: 95% ≤ 200ms
- Search API availability ≥ 99.9%
- Extractors & indexers can scale horizontally

---

5. Security

- OIDC authN for all APIs
- RBAC + tenant-level authZ
- Secrets management for storage & ML API keys
- PII: redact sensitive fields if needed

---

6. Cost & Operability

- Cloud-managed object storage → low ops overhead
- Elastic/OpenSearch → scale nodes based on load
- Monitoring: dashboards + alerts
- On-call rotation for ingestion/indexer issues

---

7. Sequence Diagram

Sequence Diagram:

Client ->> API: Upload PDF

API ->> Storage: Save PDF

API ->> Queue: Trigger processing

Queue ->> Worker: Process file

Worker ->> DB: Save metadata

Worker ->> SearchEngine: Index metadata

Client ->> API: Search

API ->> SearchEngine: Query

SearchEngine -->> API: Return results

API -->> Client: Show results