**Executive Summary**

**Business Requirement**Marine field technicians and engineers need immediate, on-site access to critical technical drawings, manuals, and troubleshooting guides—without returning to the office for document retrieval and printing.

**Current Pains**

* Time Delays: Manual searches and print-outs consume 10–15 minutes per query, magnified in emergencies.
* Inefficiency: Office trips interrupt operations and reduce equipment uptime.
* Safety Risks: Slow access to safety or troubleshooting documents can compromise crew well-being.

**Successful Solution**  
Deploy a Production-Ready AI-Powered Marine Knowledge System (MK-AI) that delivers:

1. Sub-second semantic search via AI embeddings
2. Intelligent cross-document linkage (e.g., part → drawing → manual)
3. Offline access through a Progressive Web App (PWA)
4. Seamless integration with Hexagon EDMS for real-time accuracy

**Gains and Benefits**

1. Time Savings – Search times drop from minutes to seconds.
2. Improved Accuracy – Always surfaces the latest document versions with built-in version-control metadata.
3. Enhanced Usability – Natural language queries remove exact-keyword dependencies; OCR supports scanned content.
4. Better Decision-Making – Smart linking provides contextual “next steps.”
5. Cross-Document Intelligence – Knowledge graph connects fragmented data for holistic insights.
6. Resilience & Reliability – Offline sync ensures continuity in remote environments.
7. Scalability – Architected for millions of documents and thousands of concurrent queries.

**Potential Approaches**  
• Semantic Search with AI Embeddings: Sentence-BERT → vector database (ChromaDB, Qdrant)  
• OCR for Image-Based Content: Cloud OCR (Google Document AI, AWS Textract) → text extraction  
• Smart Linking / Knowledge Graph: NER/RE pipelines → Neo4j graph database  
• Offline Access / PWA: Service Workers + IndexedDB for selective sync  
• Rule-Based Search: Regex pipelines for highly structured documents  
• Full-Text Search Engine: Elasticsearch/Solr for unstructured text indexing

**High-Level Solution Design**

1. Phase 0: Architecture & Integration Planning  
   – Define HA/scalability/security  
   – Hexagon EDMS API strategy & test plan
2. Phase 1: Core Semantic Indexing Pipeline  
   – PDF/OCR ingestion → embedding generation → vector DB  
   – Backend API + prototype UI
3. Phase 2: Intelligent Linking & Knowledge Graph  
   – Train custom NER/RE models  
   – Populate & expose graph-DB endpoints
4. Phase 3: Production Application  
   – Secure, scalable API & responsive web/mobile UI  
   – Offline sync mechanism  
   – CI/CD, monitoring, and UAT

**Timeline & Budget**Estimated PoC Duration: 3–5 months (Phase 1 prototype)  
Estimated Cost: N/A

**Impact on Business & Technology**

|  |  |  |
| --- | --- | --- |
| **Approach** | **Business Impact** | **Technical Impact** |
| Semantic Search | Dramatic reduction in search times | ML infrastructure, embedding store, query servers |
| OCR | Enables access to legacy/scanned docs | OCR service costs; preprocessing pipeline |
| Knowledge Graph | Deep contextual linking; audit trails | Annotation effort; graph-DB management |
| Offline PWA | Ensures access in no-connectivity zones | Sync/resync complexity; local storage management |
| Rule-Based Search | Fast for well-structured docs | Rigid; high maintenance when docs evolve |
| Full-Text Search Engine | Rapid, scalable unstructured text retrieval | Lacks semantic ranking; requires relevance tuning |

**Risks**

* OCR Limitations: Poor-quality scans or handwriting may fail.
* Data Annotation: NER/RE model training is labor- and time-intensive.
* Model Accuracy: Generic NLP models need fine-tuning for marine jargon.
* Security Complexity: EDMS authentication & authorization integration is non-trivial.
* Offline Engineering: Reliable sync/resync logic adds engineering overhead.
* User Adoption: Requires change management and training.

**Improved Business Processes**

* Semantic Search: Eliminates office trips for quick lookups.
* OCR Indexing: Digitizes paper archives for instant retrieval.
* Knowledge Graph: Visually surfaces related assets for faster troubleshooting.
* Offline PWA: Uninterrupted access in remote or connectivity-challenged locales.
* Rule-Based Queries: Quick wins for standardized document types.

**Constraints, Limitations & Assumptions  
Constraints:**

* Must handle millions of documents with 99.99% uptime.
* Initial PoC limited to sample data; full EDMS integration deferred to later phases.

**Assumptions:**

* Hexagon EDMS API credentials and sample datasets are available.
* Cloud budget is approved for OCR and vector-DB services.
* Field devices (tablets/handhelds) support PWA storage requirements.

**Statement of Work (SOW)**

**Purpose and Scope**Deploy a production-grade AI-powered semantic search and intelligent document-linking system for marine technical documentation, enabling rapid, offline-capable, natural-language access on-site.

**In Scope:**  
• Phases 0–3 (architecture, PoC, knowledge graph, production rollout)  
• Core semantic search, knowledge graph, responsive UI, offline sync

**Out of Scope:**• Real-time collaboration/markup features  
• Integration with IoT or live operational telemetry  
• Predictive analytics beyond retrieval  
• Long-term third-party maintenance beyond rollout

**Requirements**

**Functional**

1. Automated ingestion & indexing (PDF → OCR → embeddings)
2. Natural-language semantic search
3. Intelligent document linking & context navigation
4. Secure, scalable backend API
5. Responsive web/mobile UI
6. Offline document sync & local search

**Technical**• Languages & Frameworks: Python, FastAPI, React/Angular/Vue.js  
• AI/ML: Hugging Face sentence-transformers, custom NER/RE models  
• Data Stores: Vector DB (Weaviate/Qdrant), Graph DB (Neo4j)  
• OCR: Google Document AI or equivalent  
• Cloud & DevOps: AWS/GCP/Azure, Docker, Kubernetes, CI/CD pipelines  
• Workflow: Apache Airflow or Prefect  
• Security: SSO/RBAC, encryption at rest & in transit

**Deliverables**• Phase 0: Architecture blueprint, EDMS integration plan, test strategy  
• Phase 1: Ingestion pipeline, embedding service, prototype API & UI  
• Phase 2: NER/RE models, populated knowledge graph, linking endpoints  
• Phase 3: Production API, responsive UI, offline sync, CI/CD setup, UAT report

**Documentation  
Functional:**• User manuals  
• Quick-start & training guides for field personnel

**Technical:**• System architecture diagrams  
• Component design documents  
• API documentation (OpenAPI/Swagger)  
• Deployment runbooks & operational run-books  
• Code-base README and style guide

Duration: N/A