

Building Future-Proof Software Systems

Maintainability, AI Integration, and
Interoperability in the Digital Age of
Shipping

Achilleas Mantzios — Analyst,
Dynacom Tankers Mgmt

The Three Pillars of Future-Proofing

- Easily Maintainable: Code that adapts, not corrodes
- AI-Integratable: Ready to leverage intelligence, not just data
- Inherently Interoperable: Designed to connect, not to be an island

Pillar 1: Easily Maintainable Systems

- The Problem: Business logic scattered everywhere. Simple changes become complex, risky endeavors
- The Goal: A Single Source of Truth
- The Solution: Database-First Development

How It Works: Database as the Contract

- Step 1: Model your business entities rigorously in the database (e.g., shipments, containers, voyages)
- Step 2: Use tools to generate type-safe code directly from your schema
- Step 3: Build your application using these generated classes as the foundation

Pillar 2: Seamless AI Integration - The New Landscape

- AI is not magic. It's a new layer of intelligence for your systems
- It needs structured access to data, context, and tools
- Beyond the Hype: Key Concepts to Master

The AI Toolbox: Key Terminology

Foundational Models (LLMs)

- LLMs (Large Language Models): The "brains" - GPT-4, Claude, Llama. Understand and generate language
- Multimodal Models: Models that can process text, images (e.g., container damage photos), and audio (Google Gemini and GPT-4o, etc)

AI Integration Patterns

- RAG (Retrieval-Augmented Generation): Grounding AI in your private data. Prevents "hallucinations"
- Fine-Tuning: Customizing a base model on your specific data (e.g., charter parties) to make it a domain expert

The AI Toolbox: Key Terminology

AI Agents

- AI Agents: Systems that use LLMs to reason, plan, and take action by calling APIs/tools
- Orchestration (e.g., LangChain, LlamaIndex): Frameworks to build and manage these complex AI workflows

Connectivity & Safety

- MCP (Model Context Protocol): A standard "plug" for safely connecting AI to tools and data
- Guardrails: Techniques to ensure AI outputs are safe, on-topic, and use only permitted tools

How to Build for AI Today

- Design API-First: Every core function (calculateETD, getVesselPosition) should be a well-documented API. Agents consume APIs
- Centralize Your Knowledge: Use vector databases (e.g., pgvector) to store and search your documents for RAG
- Think in Events: AI Agents can be triggered by events (e.g., VesselDeviatedFromRoute)
- Start with a Pilot: A RAG-powered Q&A system for your internal procedures is a great first step

Pillar 3: Inherent Interoperability

- No system is an island, especially in global shipping
- You must connect with: Port Authorities, Customs, Logistics Partners, IoT Sensors, Customer Portals
- The Challenge: Diverse formats, protocols, and legacy systems

Achieving Interoperability: Events & APIs

Embrace an Event-Driven Architecture:

- Example Event: ContainerDischarged, VoyageUpdated, CustomsClearanceReceived

Technology Stack:

- Apache Kafka / Pulsar: The central nervous system for events
- Debezium: Captures database changes and streams them as events. (Perfect fit with our DB-first approach!)
- REST & GraphQL: For synchronous, request-response communication

Bringing It All Together: A Practical Example

Scenario: "Anomalous refrigeration unit detected on a shipment"

- 1. DB & Schema: The reefer_temperature alert is logged in the DB (PostgreSQL)
- 2. Debezium & Kafka: Captures the change, emits ReeferAlert event
- 3. Interoperability: Event notifies the partner's logistics system automatically
- 4. AI Integration: An AI Agent is triggered
- It uses RAG with a Vector Database (pgvector) to search historical data
- It reasons and uses MCP to call an action: "Execute notifyTechnician() and suggestTemperatureAdjustment()"

Key Takeaways & Your Action Plan

- Maintainability: Start with the Database. Use Database-First development for a robust core
- AI Integration: Design APIs for AI. Understand RAG for knowledge and MCP for action. Leverage pgvector
- Interoperability: Think in Events. Adopt Kafka and CDC (Debezium) to make your data flow naturally

Thank you!

Questions ?

mantzios.achill@gmail.com

<https://www.linkedin.com/in/mantzios/>