DRW AI Infrastructure Proposal

Naor Aspir - AI Platform Architecture

Executive Summary

The Challenge: DRW needs rapid access to cutting-edge AI capabilities while enabling non-technical users to create sophisticated AI workflows. Current solutions require either deep technical expertise or sacrifice advanced capabilities for simplicity.

The Solution: A unified, GCP-native AI platform that transforms DRW from "using AI tools" to "having an AI platform" - where every capability enhances the others through shared infrastructure and knowledge.

Core Architecture: A 4-layer unified platform consisting of User Layer (chat interface + visual workflow builder + admin console), API Gateway (single entry point routing), AI Services (model runtime + RAG engine + workflow engine as microservices), and Infrastructure (Kubernetes + managed GCP AI services + monitoring).

Key Benefits:

- 24-hour model deployment with automated pipeline from detection to chat and workflow availability
- RAG provides immediate domain knowledge while fine-tuning runs in background for improved accuracy
- Specialized models for financial, quantitative, and research workflows with intelligent routing
- Visual workflow builder creates autonomous agents with memory, planning, and tool integration

The Bottom Line: This platform delivers compound value where each capability improves the others - new models instantly benefit from existing knowledge, fine-tuned models improve RAG accuracy, and agentic workflows become more capable as knowledge grows. DRW gains a competitive advantage through an AI platform that evolves with their needs and scales with their ambitions.

System Architecture

Platform Philosophy

This solution leverages **Google Cloud Platform (GCP)** to maximize AI/ML capabilities through managed services including Vertex AI, AutoML, Vector Search, and BigQuery ML. The architecture treats all AI capabilities as interconnected services within a single, coherent system that becomes more valuable as each component enhances the others.

4-Layer Architecture

User Experience Layer

- Chat Interface: Direct model interaction with conversation history and context
- Workflow Builder: Drag-and-drop visual interface for creating agentic workflows
- Admin Console: Model management, monitoring, and system configuration

API Gateway & Orchestration

- Unified API Gateway: Single entry point with integration pipeline
- Authentication, rate limiting, load balancing, and intelligent request routing
- Handles both real-time chat interactions and automated workflow execution

Core AI Services Layer

- **Model Runtime**: LLM serving with auto-scaling, A/B testing, and health monitoring. Handles both general-purpose and specialized models (financial, quantitative, research) with dynamic resource allocation
- **RAG Engine**: Vector storage, retrieval, embeddings, and knowledge-powered generation
- **Workflow Engine**: Agentic chains with state management, memory persistence, and error handling

Infrastructure Layer

- Compute Cluster: Kubernetes clusters with GPU/CPU pools and auto-scaling
- Data Platform: Data lake, feature store, and model registry with version control
- Observability: Comprehensive monitoring, logging, and alerting across all services

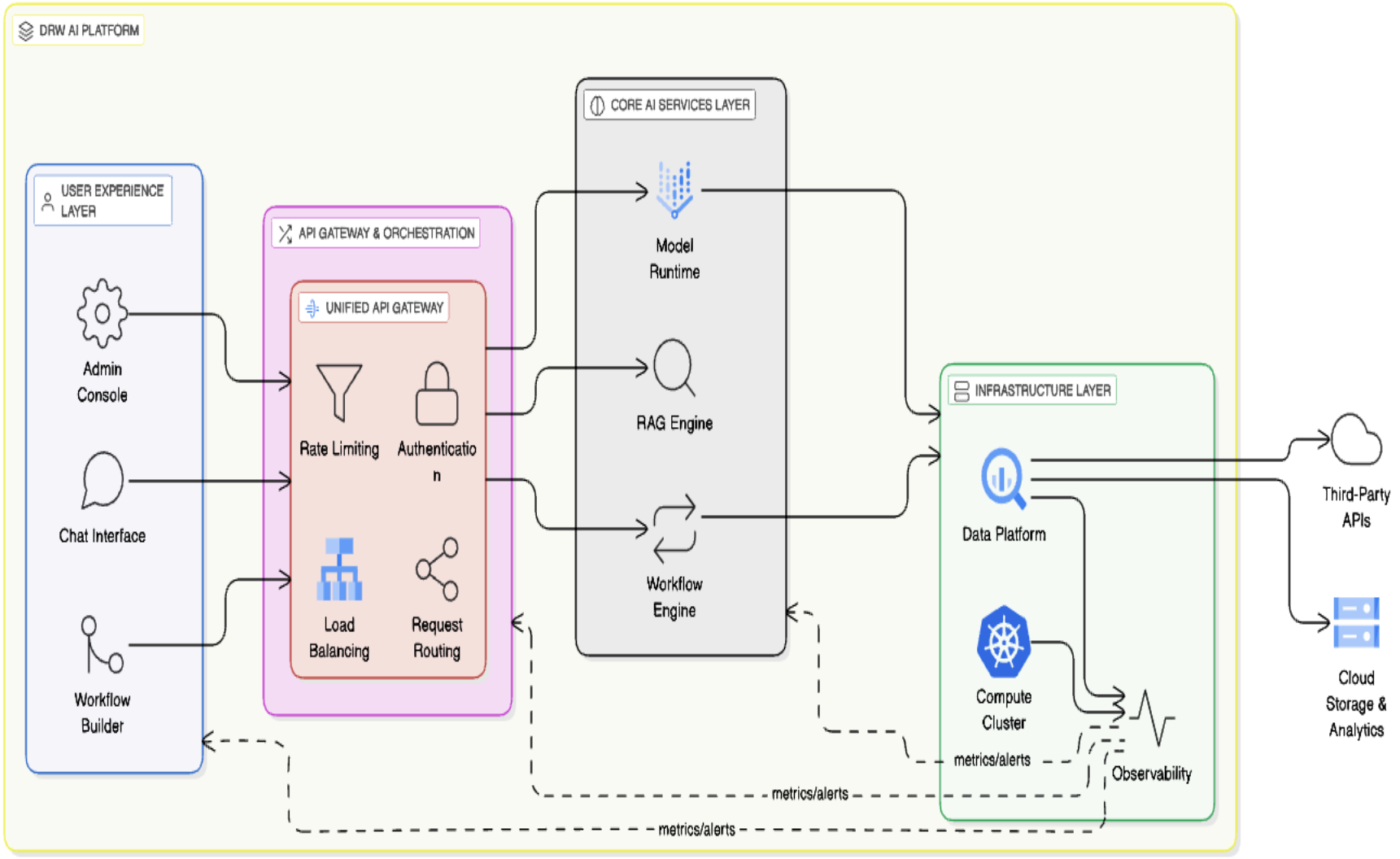
Service Integration & Data Flow

Cross-Service Communication: Event-driven architecture using GCP Pub/Sub for message queues and event streams. Services maintain loose coupling while sharing state through Redis-based caching for session and workflow persistence.

Model Deployment Pipeline: Automated detection of new models → containerized deployment → quality validation → production rollout with blue-green deployment patterns. New models become immediately available through both chat interface and workflow API within 24 hours.

Knowledge Integration: RAG Engine continuously ingests data from DRW sources, processes through chunking and embedding pipelines, and provides context-aware responses. Fine-tuning processes run in parallel, using RAG-retrieved context to improve specialized model accuracy over time.

Specialized Model Management: Model Registry catalogs all models with capability tags and domain classifications. API Gateway intelligently routes requests to optimal specialized models based on request type and business context, while dynamic resource allocation ensures appropriate GPU/CPU provisioning.


DRW AI PLATFORM
USER EXPERIENCE LAYER
Admin Console
Chat Interface
Workflow Builder
API GATEWAY & ORCHESTRATION
UNIFIED API GATEWAY
Rate Limiting
Authentication
Load Balancing
Request Routing
CORE AI SERVICES LAYER
Model Runtime
RAG Engine
Workflow Engine
INFRASTRUCTURE LAYER
Data Platform
Compute Cluster
Observability
Third-Party APIs
Cloud Storage & Analytics
metrics/alerts
metrics/alerts
metrics/alerts

RAG Architecture Deep Dive

RAG as the Knowledge Backbone

The RAG Engine serves as the central knowledge backbone that transforms static documents into queryable intelligence, enabling knowledge-augmented AI across all system capabilities. Rather than treating retrieval as a separate component, RAG is architected as the foundational layer that makes every LLM interaction contextually aware and domain-specific.

Core RAG Architecture

Vector Storage & Retrieval: Built on GCP Vector Search for managed, scalable vector operations with standardized embedding pipelines that work with any model. Implements hybrid search combining semantic similarity with keyword matching and business rules for comprehensive retrieval with metadata preservation for citation and filtering.

Document Processing Pipeline: Intelligent chunking preserves document context while creating searchable segments, with multi-format support handling various document types through unified processing. The pipeline continuously ingests from DRW data sources through automated workflows.

Knowledge Integration: Context-aware generation uses retrieved knowledge with reranking algorithms to ensure relevance. The system maintains continuous improvement through feedback loops, learning from user interactions and outcomes to enhance retrieval accuracy over time.

Integration with Platform Capabilities

24-Hour Model Deployment: New LLMs immediately access the existing knowledge base through standardized retrieval APIs. Consistent prompt templates work across different model architectures, ensuring knowledge remains available regardless of underlying model changes. This integration enables new models to provide domain-specific value from day one.

Fine-Tuning Integration: RAG provides training context for domain-specific model fine-tuning, with retrieved documents serving as few-shot examples for improved model performance. The knowledge base grows with fine-tuned model outputs, creating a compound learning effect where better models improve knowledge quality.

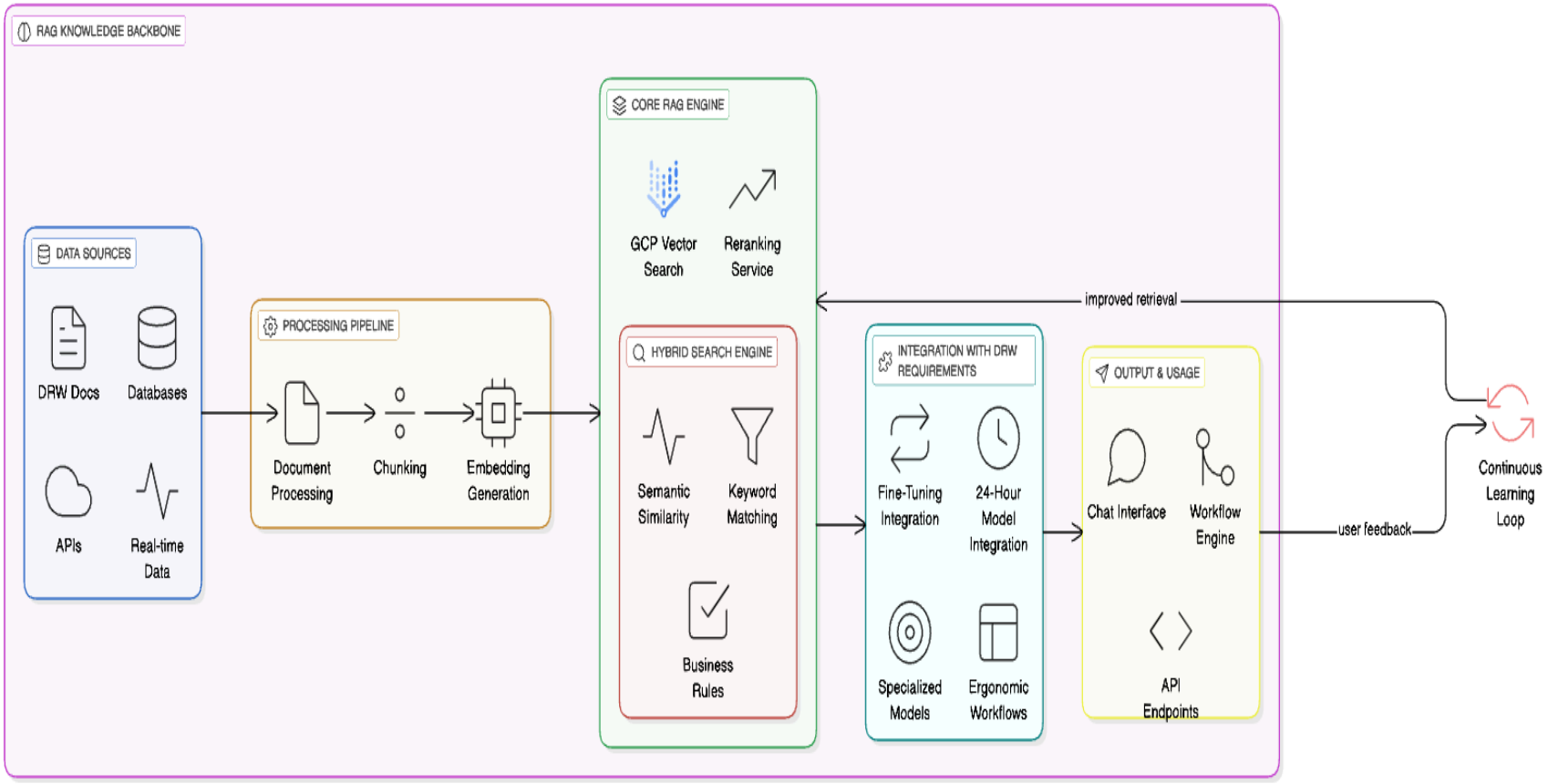
Specialized Model Support: Different embedding models can be deployed for specific document types, with specialized retrieval strategies for different business domains. Domain-specific ranking and filtering capabilities ensure that financial models access financial knowledge, quantitative models access analytical data, and research models access research documents.

Ergonomic Workflow Abstractions: Pre-built RAG components hide retrieval complexity from users through template-based knowledge queries and visual workflow integration. Business users can create workflows using high-level operations like "Find similar documents",

"Summarize context", or "Extract key insights" without understanding the underlying vector search mechanics.

Platform Enablement Through RAG

RAG transforms the platform from basic "AI model access" to "intelligent knowledge-powered workflows" by providing model-agnostic interfaces that work with any LLM, ensuring knowledge continuity across model updates and specialized deployments. The scalable foundation built on GCP managed services grows with organizational knowledge while maintaining user accessibility through abstracted operations that hide complexity behind business-friendly interfaces.


RAG KNOWLEDGE BACKBONE
DATA SOURCES
DRW Docs
Databases
APIs
Real-time Data
PROCESSING PIPELINE
Document Processing
Chunking
Embedding Generation
CORE RAG ENGINE
GCP Vector Search
Reranking Service
HYBRID SEARCH ENGINE
Semantic Similarity
Keyword Matching
Business Rules
INTEGRATION WITH DRW REQUIREMENTS
Fine-Tuning Integration
24-Hour Model Integration
Specialized Models
Ergonomic Workflows
OUTPUT & USAGE
Chat Interface
Workflow Engine
API Endpoints
improved retrieval
user feedback
Continuous Learning Loop

Requirements & Solutions

DRW Requirement	Solution Approach	Key Integration Points
24-Hour Model Deployment	Automated pipeline with model detection, containerized deployment, quality validation, and blue-green rollout. New models immediately available through chat interface and workflow API.	API Gateway routes to new models; Model Runtime handles serving; RAG Engine provides instant domain knowledge to new models
Fine-Tuning + RAG Infrastructure	RAG delivers immediate domain knowledge from documents while fine-tuning processes run in background (2-8 hours). Custom embeddings enhance retrieval precision through batch optimization.	Training Pipeline integrates with RAG workflows; Model Registry tracks base/fine-tuned relationships; Vector Store receives custom embeddings
Specialized Model Deployment	Model Registry catalogs domain-specific models (financial, quantitative, research) with capability tags. API Gateway intelligently routes requests to optimal specialized models with dynamic resource allocation.	Model Runtime serves specialized models; Smart routing based on request context; Performance tracking vs general models
Ergonomic Agentic Workflows	Visual workflow builder with pre-built RAG components, template-based knowledge queries, and autonomous agents with memory, planning, and tool integration. No-code interface abstracts LLM complexity.	Workflow Engine manages agent state; Agent Runtime provides memory/planning; Tool Registry enables API access; RAG components power knowledge operations

Cross-Requirement Synergies

Compound Value Creation: Each capability enhances the others through shared infrastructure. New models instantly benefit from existing RAG knowledge, fine-tuned models improve RAG accuracy, specialized models enhance all user interactions, and agentic workflows become more capable as the knowledge base grows.

Unified Data Flow: All requirements share the same underlying data platform, vector storage, and model serving infrastructure, ensuring consistent performance and enabling seamless integration between chat interactions and automated workflows.

Enterprise Scalability: The microservices architecture allows independent scaling of each capability based on demand, while shared services like the API Gateway and RAG Engine provide consistent interfaces and knowledge access across all requirements.

Technology Stack & Implementation

Technology Stack

Component	Technology	Rationale
Model Serving	vLLM + TensorRT	High-performance LLM inference with dynamic batching (vLLM) and GPU acceleration for specialized models (TensorRT). Optimized for production workloads.
ML Framework	PyTorch + Hugging Face	PyTorch for fine-tuning with full control, Hugging Face for simplified pipelines. Both support fine-tuning and specialized model creation.
Vector Storage	GCP Vector Search	Managed, scalable production vector database powering RAG workflows with reliable retrieval. Native GCP integration.
MLOps Platform	Vertex AI + MLflow	Vertex AI for managed training/deployment/monitoring, MLflow for experiment tracking and model versioning. Enables complete model lifecycle management.
Agent Runtime	LangGraph + Redis	LangGraph for agent state management and planning, Redis for session persistence and workflow state. Powers ergonomic agentic workflows with memory and reasoning.
Orchestration	Kubernetes (GKE)	Container orchestration with GPU scheduling, auto-scaling, and microservices deployment. AI/ML optimized clusters.
Monitoring	GCP Operations Suite	Comprehensive logging, monitoring, and alerting across all services with AI/ML specific metrics and dashboards.

Implementation Timeline

Phase	Duration	Key Deliverables	Success Criteria
Phase 1: MVP Foundation	Months 1-3	API Gateway + Model Runtime, RAG Engine with Vector Search, Basic chat interface, Initial workflow builder	Basic 24-hour model deployment working, Simple RAG workflows functional, All 4 capabilities demonstrable
Phase 2: Production Ready	Months 4-8	Fine-tuning pipeline with Vertex AI, Advanced RAG with hybrid search, Specialized model registry, Agentic workflows with LangGraph	Production pilot with limited users, 85% RAG retrieval accuracy, Specialized models deployed and routed
Phase 3: Enterprise Scale	Months 9-12	Advanced agent capabilities with tool integration, Performance optimization (vLLM/TensorRT), Comprehensive monitoring, Security and compliance framework	70% monthly active users, 99.5% system uptime, Full enterprise deployment ready

Risk Buffers: +2 weeks (Phase 1), +3 weeks (Phase 2), +4 weeks (Phase 3) for integration complexities and agentic workflow challenges.

Design Tradeoffs

Decision	Chosen Approach	Alternative	Rationale
Architecture	Microservices	Monolith	Independent scaling of GPU-heavy model serving vs lightweight RAG retrieval. Fault isolation prevents cascading failures.
Cloud Strategy	GCP-Native	Multi-cloud/On-premise	Managed AI services (Vertex AI, Vector Search) reduce operational overhead. Elastic GPU scaling for variable demands.
Build vs Buy	Hybrid: Build orchestration, Buy infrastructure	Full build or Full buy	Focus engineering on differentiating capabilities while leveraging proven managed services. Faster time to market.
Model Serving	Unified runtime for all models	Separate specialized infrastructure	Shared infrastructure reduces complexity. Dynamic resource allocation handles different model requirements efficiently.

Risk Management

High Risk - Agentic Workflow Complexity: Memory management and state persistence across complex workflows.
Mitigation: Progressive implementation starting with simple agents, extensive testing with mock workflows, LangGraph's proven state management patterns.

Medium Risk - GPU Resource Management: Balancing cost vs performance across variable workloads.
Mitigation: Predictive scaling based on usage patterns, reserved capacity for critical workloads, automated resource optimization.

Medium Risk - Multi-Model Orchestration: Ensuring consistent behavior across different model architectures.
Mitigation: Standardized prompt templates, comprehensive integration testing, gradual rollout with A/B testing.

Low Risk - RAG Accuracy vs Latency: Comprehensive retrieval may increase response times. **Mitigation**: Parallel retrieval with configurable accuracy/speed profiles, caching for frequent queries, early termination for time-sensitive requests.

Conclusion

This unified AI platform represents more than infrastructure - it's a strategic foundation that positions DRW to leverage AI as a competitive advantage. By architecting all capabilities as interconnected services within a single platform, DRW gains compound value where each component enhances the others.

The Technical Foundation: The GCP-native, microservices architecture ensures enterprise-grade reliability while enabling rapid innovation. The RAG knowledge backbone transforms every AI interaction from generic responses to domain-specific intelligence, while the visual workflow builder democratizes advanced AI capabilities across the organization.

The Business Impact: Within 12 months, DRW will have transformed from "using AI tools" to "having an AI platform" that evolves with their needs. New models become immediately valuable through existing knowledge, specialized models enhance decision-making across trading and research workflows, and non-technical users create sophisticated AI agents without coding.

The Competitive Edge: This platform doesn't just solve today's requirements - it creates a learning system that becomes more valuable over time. As DRW's knowledge base grows and models improve, the platform compounds its value, creating a sustainable competitive advantage that competitors cannot easily replicate.

The foundation is enterprise-ready, the timeline is realistic, and the architecture is designed for scale. DRW gains not just AI capabilities, but an AI platform that grows smarter with every interaction.