IQIS User Interface and Dashboard Architecture

This document outlines the envisioned design, structure, and user interaction model for the Illyad Quant Intelligence System (IQIS) dashboard and control interface. It details how users will interact with IQIS through a unified software environment combining data visualization, agent insight, trade execution, and risk management in a seamless workflow.

# 1. Form Factor

IQIS will operate primarily as a web-based application built with Next.js/React, communicating with the FastAPI backend. This approach ensures accessibility, scalability, and future support for mobile or desktop versions via Electron wrappers. MT5 remains an external execution node communicating via API.

# 2. Dashboard Overview

The dashboard functions as a command and analytics hub for all IQIS operations. It consolidates data streams, agent outputs, risk controls, and performance metrics into an interactive environment designed for both monitoring and control.

## Command Center (Home)

* Displays live P&L (day/week/month), risk status, and drawdown metrics.
* Provides overview of open positions by sector or asset class.
* Shows Agent Consensus Meter with stacked agent scores (Buffett, Munger, Dalio, Ackman, Sentiment).
* Includes Insight Feed with latest impactful news and agent rationales.
* Control panel for trading mode (Halt / Paper / Live) and Precision Mode toggle.

## Portfolio

* Displays current holdings, position sizes, stop losses, and unrealized P&L.
* Visualizes exposures by factor, sector, and currency.
* Includes simple Value-at-Risk and Drawdown monitors.
* Provides Rebalance Preview screen for pending position updates.

## Agents

* Cards for each agent showing top long/short signals and rationale.
* Score breakdowns (Value, Moat, Sentiment, Leverage, etc.).
* Disagreement Radar visual to identify agent divergence.
* Controls to adjust agent weightings, consensus thresholds, and abstain rules.

## News & Sentiment

* Live headline feed with sentiment scores and ticker tagging.
* Filters for watchlist, impact level, and topic category (earnings, legal, M&A).
* Click-to-expand details showing sentiment analysis and original source.
* Integration with MVP live news ingestion (RSS/Polygon/FinBERT).

## Signals & Execution

* Signal queue with statuses (Pending, Working, Filled, Rejected).
* Displays trade latency and slippage metrics.
* Manual or automatic execution mode per strategy.
* One-click trade adjustments (scratch, reduce, close).

## Risk & Limits

* Displays daily drawdown, per-trade risk %, gross/net leverage.
* Interactive 'What-if' sliders to simulate market shocks or volatility spikes.
* Shows exposure summary per instrument, asset class, and region.

## Reports & Learning

* Expectancy reports (win rate, avg R, profit factor) per agent or symbol.
* Performance attribution by regime and asset type.
* System changelog for models, data sources, and configuration updates.

## Settings

* API keys and connection settings (hidden/masked).
* MT5 endpoint configuration and authentication.
* Strategy parameters, news feed management, and user alerts (email/Discord).

# 3. Interaction Methods

Users interact with IQIS through both graphical elements and natural-language commands. A Command Palette (Ctrl/Cmd+K) allows typed queries such as 'show AAPL reasoning' or 'pause live trading'. The system supports contextual chat queries to agents ('Why long NVDA?') and manual approval workflows for high-risk trades.

# 4. Architecture and Integration

The dashboard communicates with backend services via REST and WebSockets for real-time updates. FastAPI serves as the central hub connecting Postgres (data), Redis (event queue), ETL modules (data ingestion), and the MT5 EA for trade execution. This modular approach ensures low latency and transparency across all system layers.

# 5. UI Design Principles

* Dark fintech theme with gradient accents (blue → teal → purple).
* TailwindCSS + shadcn/ui component library for consistent design.
* Responsive layout for desktop and tablet use.
* Minimalist typography, clean grid system, and clear spacing hierarchy.
* Interactive charts via Recharts or Plotly for risk and performance visualization.

# 6. Implementation Order

1. 1. Command Center and Signals/Execution tabs.
2. 2. News & Sentiment live feed integration.
3. 3. Agents visualization and Disagreement Radar.
4. 4. Portfolio and Risk dashboards.
5. 5. Reports and performance analytics.

# 7. Tech Stack Summary

* Frontend: Next.js, Tailwind, shadcn/ui, React Query, WebSockets.
* Backend: FastAPI, SQLAlchemy, Pydantic.
* Database: Postgres/TimescaleDB.
* Cache/Event Bus: Redis.
* Execution: MT5 EA bridge via REST/JSON.
* Auth: Token-based initially; Keycloak later.
* Alerting: Email and Discord webhooks.

# 8. Summary

The IQIS dashboard provides a unified, interactive control surface integrating live market data, multi-agent reasoning, risk management, and execution oversight. Its modular, web-based architecture ensures it can scale from single-user operation to institutional-level multi-strategy control with minimal rework.