IQ NUCLEUS

Unified Technical Guidelines & Consolidated Architecture Specifications

PREPARED FOR
The Ministry of Investment Saudi Arabia (MISA)

PREPARED BY
Mostafa Ismail, CTO

DATE
September 2026

1. Multi-Stream Telemetry Ingestion Architecture

To establish a highly resilient intelligence framework feeding the **Capacity Matching Engine (CME)**, IQ Nucleus operates a robust, multi-stream ingestion mechanism. The ingestion engine is designed to consolidate heterogeneous data from various structured local sovereign gateways, global market databases, and real-time financial systems. This architectural setup ensures that sovereign regulatory directives are dynamically checked against real-time global economic shifts.

A. Heterogeneous Ingestion Channels

IQ Nucleus categorizes its data sources into three distinct structures based on volatility, access protocol, and structural authority:

- **Local Sovereign Gateways:** Fast, secure REST API integrations and scheduled batch files connecting with KSA government services. Includes the Ministry of Commerce (MOC) registries, General Authority for Statistics (GASTAT) macroeconomic logs, Saudi Industrial Development Fund (SIDF) capital allocations, Saudi Authority for Industrial Cities and Technology Zones (MODON) utility meters, Etimad procurement contracts, and Qiwa talent queues.
- **Global Databases & Financial Systems:** Financial data streams, commercial intelligence databases, and international corporate registries. Integrates with the Bloomberg Enterprise API (WAPI) for raw materials indexing, global freight rates, and international industrial capital expansions.
- **Unstructured Web Scraping:** Continuous crawling of SEC EDGAR (10-K, 10-Q filings) and international exchange websites (ASX, LSE) to detect capital expenditure allocations, multinational pressrooms, and regional joint-venture interest.

Datasource	Stream Type	Key Telemetry Signals	Ingestion Protocol
MOC (Ministry of Commerce)	Sovereign G2G	Corporate registrations, active commercial licenses, parent-subsidiary trees, UBO (Ultimate Beneficial Owner) structures.	GSB REST API / MTLS
GASTAT (Authority for Stats)	Sovereign G2G	National import/export volumes, sector GDP trackers, HS-Code regional manufacturing volumes, macroeconomic trends.	SFTP Scheduled CSV Batch
MODON (Industrial Cities)	Sovereign G2G	Industrial land allocations, electrical substation live loads, water pipeline capacity metrics, city load curves.	Hourly JSON Pull REST
Qiwa (HRSD MHRSD)	Sovereign G2G	Active recruitment vacancy postings, sector job durations, specialized technical skill shortages, salary ranges.	Daily Anonymized Sync API
Etimad (Procurement)	Sovereign G2G	National project tender announcements, tender budgets, bidder metrics, awarding statuses, national procurement pipelines.	REST Webhook Push / Pull
Bloomberg Enterprise API	Global Financial	Global raw commodities indexing, international container shipping rates, global industrial Capex forecasts, energy costs.	WAPI Client Streaming
SEC EDGAR	Global Public	Corporate 10-K, 10-Q filing filings, international expansions, capex allocations, manufacturing JV interests.	Scrapy / Playwright

B. Concurrent Distributed Worker Pattern

To ensure continuous data fresh-ups without crashing our primary application pipelines, IQ Nucleus offloads all ingestion, crawling, and external syncing to a concurrent, distributed worker pattern built using **Celery** with **Redis** serving as the centralized high-throughput message broker (alternatively orchestrated via **Prefect** workflows). The background pool is split into dedicated task queues:

- sovereign_api_queue: Manages asynchronous REST polling and webhook captures from GSB endpoints, ensuring rate limits are strictly adhered to.
- web_scraping_queue: Executes continuous crawling of SEC EDGAR and exchange portals. To circumvent Cloudflare, WAF, and IP bans, workers execute Scrapy spiders integrated with **Playwright**, routing requests through a **Residential Proxy Rotation** pool (e.g., Bright Data). Spiders employ dynamic User-Agent rotations and randomized delay parameters (1.5s to 4.0s) utilizing exponential Jitter curves to mimic human navigation patterns.
- streaming_data_queue: Continuously listens to Bloomberg Enterprise push channels, streaming high-volatility financial events directly into raw ingestion landing zones.

C. Validation, Anonymization, and Normalization Layer

Before data points are integrated into the main database or routed to the CME, they undergo three phases of real-time preprocessing in the ingestion sandboxes:

1. **Validation Layer (Pydantic v2):** Evaluates incoming payloads against mathematical schemas. Any payload that does not match strict formats (e.g. non-numeric MODON loads, invalid HS-Codes, missing UUIDs) is blocked from downstream databases and automatically routed to the ingest_quarantine schema, prompting a high-priority system audit notification.
2. **Anonymization & De-PII:** To maintain complete compliance with national data privacy standards, a middleware anonymization layer sanitizes sensitive personal data at the boundaries. Qiwa recruitment profiles have individual employee names, National IDs, and phone numbers stripped, transforming individual transactions into aggregate integers (e.g., job codes, vacancy counts, durations). MOC feeds anonymize shareholder names, mapping corporate ownership trees using isolated corporate UUIDs.
3. **Local Model Normalization:** Corporate pressrooms, filings, and patent registrations contain unstructured natural language. The ingestion system routes these texts to local model agents (run locally on Ollama/vLLM or Hugging Face Transformers). The **local model** parses text paragraphs, identifies entities, and normalizes the raw content into structured JSON representations that conform to our standard schemas, bypassing any dependency on proprietary commercial APIs.

ARCHITECTURAL DESIGN HIGHLIGHT

To guarantee sovereign compliance and national data residency, all LLM-based parsing and entity normalizations are handled exclusively by 'local model' deployments. No corporate data or unstructured web scrapes are ever transmitted to third-party, commercial, or cloud-based AI providers. The entire data processing lifecycle is fully sandboxed on-premises.

2. Standardized Data Schemas & Normalization

Every raw unstructured feed crawled from the web and every structured sovereign record is forced into strict mathematical normalization at the entry boundaries. Standardized JSON schemas are represented as **Pydantic v2** models on the ingestion workers. The following Pydantic classes represent the mathematical contracts used by the 'local model' to structure unstructured texts and validate heterogeneous gateway feeds.

A. Normalized Ingestion Schemas (Pydantic Models)

```
import uuid
from typing import Literal, Optional
from pydantic import BaseModel, Field, field_validator

class NormalizedOpportunitySignal(BaseModel):
    """
    System boundary schema to ingest and validate heterogeneous telemetry.
    Forces strict typing on sovereign metrics and global signals.
    """
    signal_id: uuid.UUID = Field(default_factory=uuid.uuid4)
    sector: Literal['AI_INFRA', 'BIOPHARMA', 'GREEN_ENERGY', 'CLEAN_MOBILITY', 'FOOD_SECURITY'
]
    hs_code: str = Field(..., pattern=r'^[0-9]{4,10}$')
    import_value_usd_yoy_growth: float = Field(..., ge=-1.0, le=100.0)
    qiwa_vacancy_count: int = Field(..., ge=0)
    qiwa_average_vacancy_duration_days: float = Field(..., ge=0.0)
    modon_grid_request_mw: float = Field(..., ge=0.0)
    etimad_projected_tender_value_sar: float = Field(..., ge=0.0)
    source_reliability_score: float = Field(..., ge=0.0, le=1.0)

    @field_validator('hs_code')
    @classmethod
    def validate_hs_code(cls, v: str) -> str:
        # Custom logic to verify against world HS-Code standards
        return v
```

B. Unstructured Scraped Payload Schema

For web-scraped corpora, the local model acts as an extraction agent, mapping text paragraphs into a structured JSON schema:

```
class ScrapedTextPayload(BaseModel):
    """
    Schema for local model agent extraction from unstructured text.
    """
    extracted_company_name: str
    primary_capability_detected: str = Field(..., description="E.g., Direct-to-chip CDU assemb
ly")
    geographic_expansion_intent: Literal['KSA', 'GCC', 'GLOBAL_ONLY']
    announced_capex_usd: Optional[float] = None
    patent_identifiers: Optional[list[str]] = None
    local_partner_intent: bool = Field(default=False)
```

3. Matching & Convergence Algorithms (CME)

Once telemetry data points are structured, validated, and normalized inside our PostgreSQL landing zone, they are processed by the **Capacity Matching Engine (CME)**. The CME operates on a three-stage mathematical alignment and traversal system designed to locate the optimal joint venture (JV) or direct investment match between local industrial capacity and global tech leaders.

A. Convergence Engine (Signal Correlation & Alerting)

Anomalies across signal families do not trigger actions independently. To prevent high rates of false positives, the Convergence Engine calculates a **Rolling Weighted Z-Score Indicator ($Z_{Combined}$)**. An investment warning or deal synthesis is triggered only when cross-family signal convergence is established mathematically.

Mathematical Formula:

$$Z_{Combined} = \sum (w_i \times Z_i) / \sqrt{(\sum w_i^2)}$$

Where:

- Z_i represents the rolling z-score of signal family *i*, computed against a 90-day moving window.
- w_i is the priority weight assigned to each signal family.

Signal Family	Sovereign / Global Source	Anomaly Threshold Condition	Weight (w_i)
Thermal CDU Imports	GASTAT / ZATCA Import Logs	Import value USD YoY growth exceeds 35%	0.25 (High)
Grid Load Headroom	MODON Electrical Substation Live Meters	Available grid capacity margin drops below 18%	0.30 (Critical)
Talent Vacancy Durations	Qiwa (MHRSD Recruitment Database)	Average vacancy duration for technical codes > 45 days	0.20 (Medium)
Procurement Tenders	Etimad National Procurement Portal	Projected tender value for infrastructure exceeds 50M SAR	0.25 (High)

B. Stage 1: Entity Resolution (MOC to Global SEC)

When web crawlers identify corporate expansion mentions in SEC filings or pressrooms, they must be resolved against valid, registered Saudi business entities in the MOC database. The system executes a high-performance **TF-IDF Character N-Gram Vectorization + Cosine Similarity** match, paired with **Jaro-Winkler string distance** for English-to-Arabic transliterations. If the combined string score exceeds 0.85, the records are merged into a single, global Entity ID.

Mathematical Formula (Cosine Similarity):

$$\text{Cosine Similarity}(A, B) = (A \bullet B) / (||A|| \times ||B||) = [\sum A_i B_i] / [\sqrt{(\sum A_i^2)} \times \sqrt{(\sum B_i^2)}]$$

C. Stage 2: Capability Vector Extraction

To discover what global investors can manufacture and map them to local requirements, the system parses corporate descriptions and patent registrations using a fine-tuned **Named Entity Recognition (NER)** model in SpaCy alongside RoBERTa, extracting CAPABILITY and PRODUCT tags. These capability tags are converted into dense, 1536-dimensional vector embeddings using a **local model** embeddings engine.

Example: A scraped text like *'Manufactures custom cooling distribution units (CDUs) and manifolds for high-density servers'* is embedded. When queried against an opportunity requirement vector for *'Liquid Cooling Assembly'*, a cosine similarity of **> 0.78** establishes a technical fit.

D. Stage 3: Graph Traversal & Physical Constraints

Resolved entities, capabilities, and geographic boundaries are written to a **Neo4j Graph Database**. The matching engine traverses the graph to locate paths where underutilized local manufacturers (utilization rate < 0.6) are linked to global players who have the necessary technical capabilities but do not yet possess a physical footprint in Saudi Arabia.

```
// CME TRAVERSAL AND CONSTRAINT QUERY
MATCH (o:Opportunity {sector: "AI_INFRA"})-[:REQUIRES_CAPABILITY]->(cap:Capability)
MATCH (l:LocalPartner)-[:HAS_INFRASTRUCTURE]->(infra:Infrastructure)
MATCH (l)-[:HAS_CAPACITY]->(s:SheetMetalCapacity) WHERE s.utilization_rate < 0.6
MATCH (g:GlobalInvestor)-[:HAS_CAPABILITY]->(cap)
WHERE NOT (g)-[:HAS_LOCAL_FOOTPRINT]->(:LocalFootprint {country: "KSA"})
RETURN l.name AS LocalPartner, g.name AS GlobalPartner, cap.name AS MatchCapability, s.utiliza
tion_rate AS CapacityMargin
```

Physical Constraint Filtering: After compiling potential match paths, the CME checks physical sovereign headroom constraints: $\textbf{Headroom}_{\textbf{Available}}$ **- Demand**$_{\textbf{Opportunity}}$ ≥ **15% Substation Margin**. If a specific substation in Sudair is overloaded, the system automatically shifts the matched deal's land allocation to Jubail or Dammam and regenerates the Gate Evidence Packet.

4. High-Level Implementation & Delivery Phases

To deploy the consolidated data ingestion mechanism and the Capacity Matching Engine within a highly disciplined 12-week timeframe, MISA and Meshkah will execute the following four phases of engineering, validation, and sovereign rollout:

Phase I: Foundation & Ingestion Core (Weeks 1–4)

- **Database Setup:** Initialize secure PostgreSQL instances on port 54329, dividing schemas into raw landing zones, normalized tables, and quarantined areas with Row Level Security (RLS) enabled.
- **Sovereign API Mocking & Adapters:** Build REST adapters for MOC, MODON, and Qiwa. Create mock gateway JSON endpoints simulating sovereign API responses for unit and integration testing.
- **Scraping Clusters:** Deploy Celery workers with Redis. Configure Scrapy-Playwright crawlers targeting SEC EDGAR RSS streams, complete with rotating proxy structures and delay curves.
- **Local Normalizer:** Install and configure a **local model** instance via Ollama/vLLM. Seed the extraction prompt schemas to translate scraped filings into valid JSON payloads.

Phase II: CME Core & Algorithmic Integration (Weeks 5–8)

- **Entity Resolver:** Implement the TF-IDF and Jaro-Winkler string processing pipeline, loading MOC tables into memory for rapid entity matching.
- **Capability Vectorizer:** Deploy local sentence embeddings models to extract and represent capability tags. Load vector indices in pgvector.
- **Graph Traverser:** Setup and configure the Neo4j Graph DB cluster. Write the production Cypher matching queries and test constraints against electrical load margins in PostgreSQL.

Phase III: Active PoC & The Steel Thread (Weeks 9–10)

- **Stress Injection:** Force-inject artificial data spikes: surge import volumes for thermal CDUs, increase Qiwa hiring vacancy wait times, and register high electricity requests in MODON.
- **Steel Thread Compilation:** Track the automatic end-to-end flow of signals. Verify that the z-score threshold engine fires an alert, pre-populates a Gate Evidence Packet, and suggests an investor match.
- **Dashboard Cockpit:** Complete the Next.js (TypeScript) dashboard. Wire up the alert list cards, drill-down panels, and SVG data graphs.

Phase IV: Sovereign Rollout & Production Audit (Weeks 11–12)

- **PDF Generation Compiler:** Connect the Python ReportLab compiler to the Next.js cockpit. Enable analysts to generate the official *Capacity Match Pack* with a single click.
- **Security Hardening & Penetration Testing:** Perform deep security audits on MTLS connections. Hard-code de-PII constraints across all sovereign channels.
- **NCA Compliance & MISA Review:** Verify system compliance against National Cybersecurity Authority (NCA) standards. Present the working PoC and technical guidelines to the MISA Review Panel for production go-live authorization.

CONCLUDING ARCHITECTURAL DIRECTIVE

By enforcing strict physical constraints, real-time data accuracy scores, and sandboxed 'local model' normalizations, IQ Nucleus provides MISA with a highly reliable, sovereign decision platform. This consolidated blueprint outlines a path toward high-velocity, low-risk transaction enablement.