

1. Source & Ingestion Layer (Real-Time + Batch)

Source category	Typical feed protocol	Ingestion pattern	Notes
Market data – equities, FX, fixed-income, derivatives, indices	FIX/FAST,ITCH, Kafka topics from vendors (Refinitiv, Bloomberg B-pipe, crypto exchanges)	Streaming via Kafka Connect → schema-validated Avro/Protobuf topics	Millisecond latency target; align timestamps to UTC-microsecond precision.
Trading platforms / OMS / EMS	FIX 4.x order messages, proprietary REST/GRPC	Dual ingest: real-time Kafka + nightly SFTP drops for full trade blotters	Maintain causal ordering (Cl-Ord-ID) across shards.
Transaction & settlement systems	SWIFT MT/MX, ISO 20022 XML	Near-real-time message queue → CDC into lakehouse	Leverage Debezium for RDBMS CDC.
Risk databases & reference data	Oracle/MSSQL, Snowflake, golden-source REST APIs	Batch ELT every 5 min – 1 h (configurable)	Dimension tables for instruments, calendars, curves, legal entities.
Regulatory & external data (Fed filings, ESG, macro)	S3, HTTPS, SecEdgar, ECB SDMX	Scheduled fetch + hash-based upsert	Tag with provenance & retention metadata.

All feeds land in **Kafka** (hot path) and **object storage (e.g., S3 / Azure Data Lake / GCS)** (cold path) with *immutable, partitioned* Parquet files.

2. Storage Layer (Performance-Tiered)

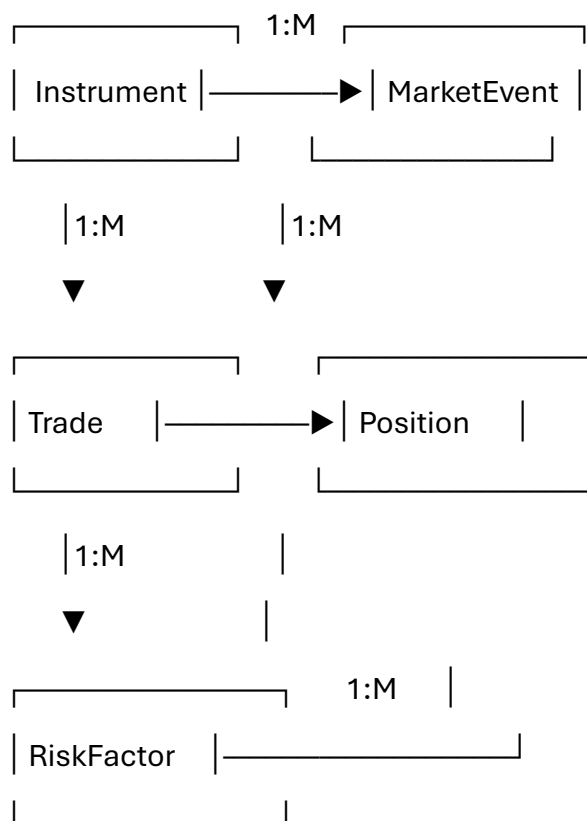
Tier	Technology	Purpose	Key design points
Hot / real-time	Apache Pinot or ClickHouse cluster on NVMe	Sub-second ad-hoc queries on tick data & Greeks	Tuned for time-series; eviction to warm tier after N days.

Tier	Technology	Purpose	Key design points
Warm (analytical lakehouse)	Delta Lake / Apache Iceberg on distributed file system	Petabyte-scale historical storage; ACID for back-tests	Partition by <i>instrument_type / date</i> ; Z-order by symbol.
Risk-calc cache	Redis Cluster or Aerospike (in-memory, replicated)	Serve VaR/ES & stress metrics to dashboards	TTL aligned to risk refresh cycle (e.g., 60 s).
Regulatory vault	WORM-compliant object storage (e.g., S3 Object Lock)	5–7 year retention of trade & quote data for audits	Satisfies SEC Rule 17a-4 (f), MiFID II, etc.

3. Canonical Data Model

ruby

Copy code



- **Instrument:** ISIN, FIGI, contract specs, calendars
- **MarketEvent:** timestamp, bid/ask, size, venue, Greeks, implied vol
- **Trade:** side, qty, price, trader, desk, book, clearing data
- **Position:** aggregated exposures (Δ , Γ , vega, currency, tenor)
- **RiskFactor:** curve points, scenarios, stress-shocks
- **RegMetrics:** VaR, SVaR, CVA, CCR, liquidity ratios

All entities carry `data_domain_id`, `source_system_id`, `ingested_at`, and *row-level lineage* columns for governance.

4. Processing & Analytics Layer

- **Stream processing** → Flink / Spark Structured Streaming
 - Real-time P&L, intraday VaR, limit-breach alerts.
 - **Batch ELT** → dbt / Spark SQL
 - End-of-day consolidations, scenario runs, back-testing.
 - **Model orchestration** → Airflow / Dagster with MLflow registry
 - Portfolio-optimization models (mean-CVaR, Black-Litterman)
 - Risk models (Historical VaR, Monte Carlo, FRTB IMA).
 - **Serving & visualization** → Grafana + Superset + JupyterHub
 - Trader dashboards, CRO cockpit, regulatory report APIs (XBRL, XML).
-

5. Governance, Quality & Lineage

Control	Mechanism
Data catalog	OpenMetadata / Collibra; automated schema discovery.
Quality rules	Deequ / Great Expectations tests (e.g., $\text{bid} < \text{ask}$, $\text{VaR } \Delta < 10 \%$).
Lineage	OpenLineage tracking at column-level across Kafka → Lake → Warehouse.

Control	Mechanism
Master/Golden records	Surrogate primary keys; SCD-2 for mutable attributes (e.g., counterparty ratings).
Reg-tech compliance	Basel III/IV, Dodd-Frank (CFTC Part 45), EMIR, MAS 610 captured in metadata tags.

6. Security & Privacy (“Defense-in-Depth”)

- **Role-based access control (RBAC)** via LDAP/Okta + fine-grained table ACLs (Apache Ranger).
 - **Column-level & row-level security** (e.g., mask client_ssn, restrict desks to own books).
 - **End-to-end encryption:** TLS 1.3 in flight; AES-256-GCM at rest; HSM-managed keys.
 - **Audit & trace:** Immutable logs in Elastic SIEM; fed to UEBA & anomaly detection.
 - **Segmentation:** Separate VPCs / subnets for dev, UAT, prod; zero-trust service mesh (mTLS).
-

7. Reliability, Performance & Ops

- **High-availability:** multi-AZ clusters; quorum-based replication.
 - **Low-latency:** pushdown filters, vectorized queries, kernel-bypass NICs (DPDK).
 - **Disaster-recovery:** hourly metadata backups, cross-region object replication; RTO < 30 min.
 - **Observability:** Prometheus metrics, OpenTelemetry traces, SLA dashboards.
-

Example Prompt Sequence (if you want to iterate)

1. Initial ask

“Create a detailed data-architecture design for a financial institution’s trading & risk-management system. Address real-time market feeds, high-performance storage, complex risk models, data governance, and regulatory reporting.”

2. Refine models

“Add an entity-relationship description covering instruments, trades, positions, risk factors, and regulatory metrics.”

3. Deep-dive security

“Expand the design’s security section with specific encryption, RBAC, and audit-logging controls that satisfy SEC 17a-4 (f) and MiFID II.”

Iterating like this helps the model keep focus while you progressively layer detail—mirroring the lab’s technique.

Deliverables produced: a *holistic, regulation-ready* data-architecture design that integrates real-time trading, risk analytics, and stringent governance—exactly matching all five considerations in the practice exercise.