ChatGPT

Universal Intelligence Layer Technical Blueprint

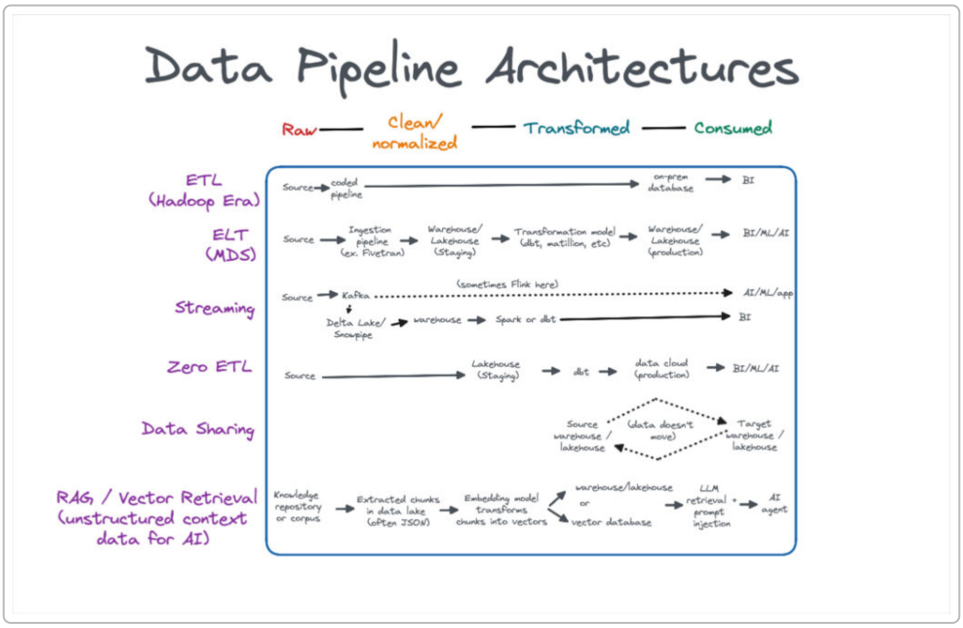
Data Pipeline Architectures
Raw
Clean/ normalized
Transformed
Consumed
ETL (Hadoop Era)
Source
coded pipeline
on-prem database
BI
ELT (MDS)
Source
Ingestion pipeline (ex. Fivetran)
Warehouse/ Lakehouse (Staging)
Transformation model (dbt, matillion, etc)
Warehouse/ Lakehouse (production)
BI/ML/AI
Streaming
Source
Kafka
(sometimes Flink here)
AI/ML/app
Delta Lake/ Snowpipe
warehouse
Spark or dbt
BI
Zero ETL
Source
Lakehouse (Staging)
dbt
data cloud (production)
BI/ML/AI
Data Sharing
Source warehouse / lakehouse
(data doesn't move)
Target warehouse / lakehouse
RAG / Vector Retrieval (unstructured context data for AI)
Knowledge repository or corpus
Extracted chunks in data lake (often JSON)
Embedding model transforms chunks into vectors
warehouse/lakehouse
or
vector database
LLM retrieval + prompt injection
AI agent

Figure: Example data pipeline architectures combining email ingestion, web crawling, and processing. Our ingestion pipeline is multi-stage. **Prioritized Sources:** We subscribe to Substack, Medium, and LinkedIn newsletters via an IMAP/Gmail workflow in n8n. A Gmail node fetches each newsletter email, then a code or HTML node extracts the full HTML content [1]. This raw HTML is parsed to plaintext (stripping images/tracking pixels and ads) before storage. **Web Crawling:** For public news (TechCrunch, AI news sites, HBR), we use Firecrawl (enterprise scraper) as our primary tool [2]. Firecrawl's "Extract" mode can output structured JSON, simplifying data cleaning. As a backup (or for maximum control), a self-hosted LLM-powered crawler like Crawl4AI (open-source Python library) can be used [3] [2]. - **Ingestion Workflow:** Email (IMAP/Gmail) → n8n HTML Extract → clean text; Web (Firecrawl/Crawl4AI) → parse → text. All fetching runs through headless browsers (using Nodriver or Camoufox) and proxies to evade anti-bot measures [4] [5]. For example, Nodriver avoids Selenium/CDP and controls Chrome invisibly [5], and Camoufox (Firefox-based) injects fake fingerprints and blocks ads for stealth [4]. We employ proxy/CAPTCHA services (Bright Data or ScrapingBee) so that requests rotate through real IPs and solve challenges [6] [7]. For CAPTCHAs, we integrate solving (e.g. 2Captcha) when needed [7]. - **Data Storage:** Cleaned article text is saved to PostgreSQL, ready for graph construction (see Database section).

2. Cross-Domain Synthesis Prompt

- **Model Stack:** We leverage DeepSeek-R1 (via DeepInfra or Together AI) for the heavy "thinking" tasks of cross-domain NER and insight extraction. DeepSeek is a chain-of-thought-optimized model, which "generating thinking trajectories" helps NER by capturing deeper context and disambiguating entities [8]. For initial steps (summarization, ad-stripping, keyword extraction) we use cheaper public models (e.g. Google Gemini 2.5 Flash or GPT-4o-mini) to keep costs low.
- **Prompt Design:** We employ a structured system/user prompt schema [9]. The **System** message frames the AI as a behavioral-science-to-business expert. The **User** message provides two texts: an academic excerpt and a related news/business article. For example:

System: “You are an AI assistant linking behavioral science research to business strategy. When given an academic text and a news article, identify the core psychological principle in the first and find its practical implementation in the second.”
User: “Academic paper excerpt: ‘…variable reward schedules increase engagement…’; Business article: ‘…a startup using gamified rewards for customers…’.”
The assistant should output something like `{"principle": "Variable Reward", "business_case": "..."}.` This aligns with best practices: use the System prompt for role/tasks and the User prompt to supply data [9] .

- **Connective Extraction:** The agent’s task is to map between domains. For example, if the academic text describes “social conformity,” the agent should recognize **Social Proof** in the news context. We then verify its output against known concepts (see Ontology). The system prompts instruct it to produce both the **Behavioral Principle** and a short **Business Implication**. Using DeepSeek-R1’s reasoning capabilities will help catch subtle links that a one-pass model might miss [8] .

3. Graph-RAG & Database Infrastructure

- **Graph-RAG vs Vector-RAG:** We adopt a graph-augmented RAG approach. Instead of pure vector search, we store semantic facts as a graph (nodes and edges) for richer retrieval. For example, the Cognee n8n integration (“cognify” node) converts each ingested document into embeddings **and** a knowledge graph (nodes + edges) [3] . This means we can query by concept **and** by vector similarity. Similarly, a Neo4j + LangGraph setup would extract entities/relations into Neo4j (with embeddings) and combine graph queries with vector search [10] [11] .
- **Postgres + Apache AGE:** We do **not** introduce a separate graph database. Instead, we use PostgreSQL with Apache AGE extension to simulate the graph. AGE stores each label’s vertices/edges in SQL tables [12] . Concretely, each edge is a row with `source_id` and `target_id` referencing vertex rows [12] . We can index edge relationships and even attach a pgvector column to edges or nodes for semantic similarity. In practice, Apache AGE lets us use Cypher-like queries on these tables while staying in Postgres [13] [12] . For smaller sets, a custom table (e.g. `edges(id, source, target, type)`) with pgvector columns is an alternative; however note that “Postgres isn’t good for graph work unless the graph is very shallow” and that pgvector may become a bottleneck at scale [14] .
- **Efficiency:** Using AGE unifies storage: we avoid syncing data between DBs. n8n workflows simply write new nodes/edges into Postgres (via AGE calls). This retains ACID guarantees and simplifies joins. We can still perform hybrid queries (Cypher via AGE + SQL) for our RAG pipeline, effectively achieving Graph-RAG on a single DB [10] [12] .

4. Neuro-Business Ontology Scope

- **Core Concepts:** Our “anchor nodes” cover broad behavioral science and neuroscience ideas. For **consumer psychology**, include classic influence principles: *Scarcity* (limited offers), *Social Proof* (herd behavior) [15] [16] , *Loss Aversion*, *Framing*, *Reciprocity*, *Anchoring*, *Cognitive Biases*, *Variable Reward* (gamification).
- **Organizational Behavior:** Include *Pareto Principle* (80/20 rule in productivity and sales) [17] , *Dunbar’s Number* (optimal team size) [18] , *Groupthink*, *Social Loafing*, *Nudge Theory*, and basic learning principles (e.g. *Operant Conditioning*, *Feedback Loops*).
- **Leadership/Neuroscience:** Include *Cognitive Load Theory* (minimize overload in design/training) [19] , *Attention/Memory Limits*, as well as neurochemicals relevant to leadership: *Dopamine*

(reward/motivation), *Serotonin* (mood), *Cortisol* (stress) vs *Oxytocin* (trust/bonding) [20] (high cortisol impairs decision-making, oxytocin fosters team trust). *Mirror neurons* (empathy/mimicry in teams) and *Neuroplasticity* (learning) are included. Each of these should be a graph node category ("Psychology Concept", "Behavioral Bias", "Neuroscience Fact", etc.) so the BAE can tag findings to these anchors and ensure advice aligns with them.

n8n Environment Variables (1,000+ sources/day)

We run n8n in **queue mode** for scale. Key settings include:

- `EXECUTIONS_MODE=queue` (ensures distributed queue processing) and configuring a Redis broker (via `BULLMQ_REDIS_HOST`, `BULLMQ_REDIS_PORT/PASSWORD`) since "Redis acts as the message broker and the database persists data" [21] .
- `N8N_CONCURRENCY_PRODUCTION_LIMIT` set to a high value (e.g. 20+) to allow many parallel executions [22] . In queue mode, this caps concurrent jobs per worker. Each worker is launched with `--concurrency=N` (n8n recommends ≥5) to allow parallel workflow runs [23] .
- `WEBHOOK_URL` configured to our domain [24] . We also set `N8N_DISABLE_PRODUCTION_MAIN_PROCESS=true` so that the main n8n process only handles the editor/UI and webhook dispatch, while all workflow runs go to workers.
- Optional: health-check endpoints (`QUEUE_HEALTH_CHECK_ACTIVE`), and execution pruning vars (e.g. `EXECUTIONS_DATA_MAX_AGE`, etc.) to manage DB size. Ensure `N8N_ENDPOINT_WEBHOOK` matches any custom webhook paths.
- **Scaling:** We run multiple worker containers (each with the above vars) behind a load balancer. The main process is separate from workers. Monitoring shows queued executions moving through as capacity frees. By tuning `N8N_CONCURRENCY_PRODUCTION_LIMIT` and worker count, we can sustain 1,000+ daily triggers.

Strategic Note

By using **Apache AGE** to add graph capabilities to Postgres, we keep a single database for both structured data and our knowledge graph [13] [12] . This avoids the overhead of a separate graph system. n8n workflows can write to different Postgres schemas or views (one for "business plan" tables, one for "KG" nodes/edges) and still query across them. Once the ontology schema is defined, Claude (our collaborator) can use it to generate precise SQL/AGE (Cypher) queries that join new news articles to decade-old neuroscience principles, ensuring every insight is grounded in both data and behavioral theory [13] [3] .

---

[1] Create daily newsletter digests from Gmail using GPT-4.1-mini | n8n workflow template
https://n8n.io/workflows/7255-create-daily-newsletter-digests-from-gmail-using-gpt-41-mini/

[2] Crawl4AI vs. Firecrawl: Features, Use Cases & Top Alternatives
https://brightdata.com/blog/ai/crawl4ai-vs-firecrawl

[3] Cognee - n8n × cognee: Persistent Workflow Context with AI Memory
https://www.cognee.ai/blog/integrations/n8n-cognee-integration-build-workflows-with-memory

[4] [5] [6] [7] Web Scraping without getting blocked (2026 Solutions)
https://www.scrapingbee.com/blog/web-scraping-without-getting-blocked/

[8] Deepseek R1 and Open-Source Reasoning Models: Their Performance on NLP Tasks and the New AI Landscape
https://www.austinai.io/blog/deepseek-r1-and-open-source-reasoning-models

[9] Prompt Engineering NER
https://dswithmac.com/posts/prompt-eng-ner/

[10] [11] How to Build a Knowledge Graph RAG Agent Locally with Neo4j, LangGraph, and watsonx.ai – Thomas Suedbroecker's Blog
https://suedbroecker.net/2025/09/26/how-to-build-a-knowledge-graph-rag-agent-locally-with-neo4j-langgraph-and-watsonx-ai/

[12] apache age - AGE Graph is actually stored as a postgreSQL Table, Right ? How to retrieve that Table (not Graph)? - Stack Overflow
https://stackoverflow.com/questions/75554331/age-graph-is-actually-stored-as-a-postgresql-table-right-how-to-retrieve-that

[13] Apache AGE
https://age.apache.org/

[14] I implemented "Sleep Cycles" (async graph consolidation) on top of pgvector to fix RAG context loss : r/AIMemory
https://www.reddit.com/r/AIMemory/comments/1pou4rg/i_implemented_sleep_cycles_async_graph/

[15] On the role of scarcity in marketing: Identifying research opportunities across the 5Ps - PMC
https://pmc.ncbi.nlm.nih.gov/articles/PMC10242224/

[16] Social Proof - The Decision Lab
https://thedecisionlab.com/reference-guide/psychology/social-proof

[17] Pareto principle - Wikipedia
https://en.wikipedia.org/wiki/Pareto_principle

[18] Dunbar's number - Wikipedia
https://en.wikipedia.org/wiki/Dunbar%27s_number

[19] Smart Design Hacks: Using Cognitive Load Theory for a Better User Experience | Bootcamp
https://medium.com/design-bootcamp/smart-design-hacks-using-cognitive-load-theory-for-a-better-user-experience-part1-6b7761ebbcd6

[20] Exploring the Neuroscience of Leadership in Decision-Making
https://www.leadershipscienceinstitute.com/exploring-the-neuroscience-of-leadership-in-decision-making/

[21] [23] [24] Configuring queue mode | n8n Docs
https://docs.n8n.io/hosting/scaling/queue-mode/

[22] Concurrency control | n8n Docs
https://docs.n8n.io/hosting/scaling/concurrency-control/