Sure! Here's a **technical design white paper** on the PlantUML lakehouse architecture, using your voice and enriched with references to case studies and academic sources.

---

1 1. Introduction

In this white paper, I'm laying out a **Lakehouse architecture** that unifies all data—structured, semi-structured, unstructured—under one roof. You'll see how layers like Bronze, Silver, and Gold mesh with technologies like Delta Lake, Iceberg, or Hudi to enable ACID compliance, governance, scalable compute, and seamless analytics for both BI and ML use cases.

1.1 Architecture Overview

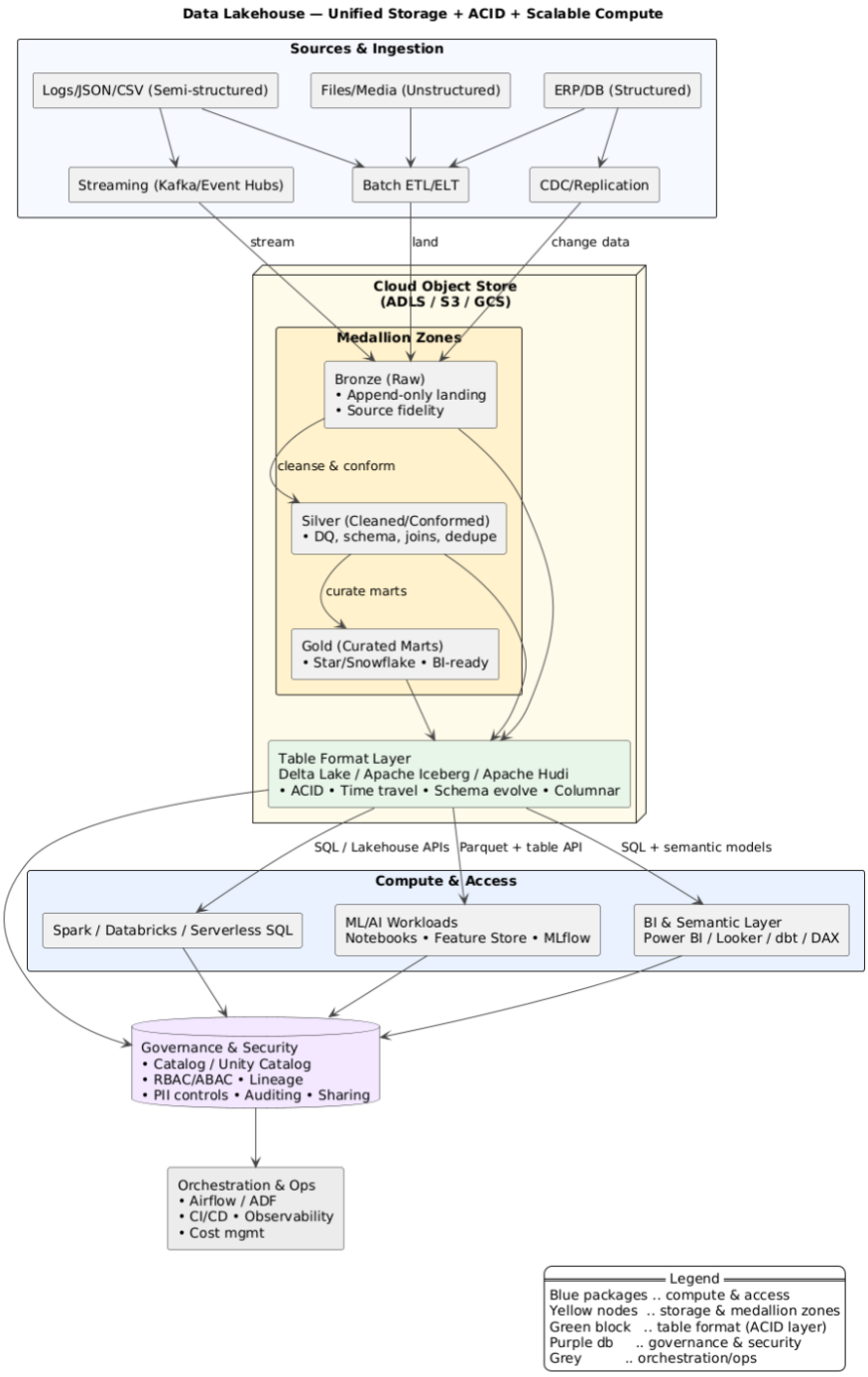

Figure 1: Blueprint Data Lakehouse architecture

Let's walk through each layer, why it matters, and what considerations you should keep front-of-mind.

1.2 Layer-by-Layer Breakdown

1.2.1 Sources & Ingestion

- **Description**: Data enters from structured systems (e.g., ERP/relational DBs), semi-structured logs/files (JSON, CSV), and unstructured sources (media, documents). Ingestion modes include batch ETL/ELT, streaming (Kafka, Event Hubs), and CDC.
- **Considerations**:
  - Latency vs. completeness trade-offs (stream vs batch).
  - Data schema variability—needs robust pipeline design to catch drift.
  - Scalability and fault tolerance of ingestion.
- **Reference**: This layering avoids siloing data in separate warehouses and lakes, streamlining architecture and cost-efficiency (Microsoft Learn).

1.2.2 Cloud Object Storage ("Data Lake")

- **Description**: A centralized blob store (ADLS, S3, GCS) that holds raw and processed data affordably at massive scale.
- **Considerations**:
  - Schema-on-read flexibility vs optimizing for query performance.
  - Data governance, encryption, lifecycle policies.
  - Immutable, append-only storage simplifies auditability.
- **Reference**: Drew from Delta Lake white paper about object storage forming the foundation of the lakehouse (people.eecs.berkeley.edu).

1.2.3 Medallion Zones: Bronze, Silver, Gold

- **Description**:
  - **Bronze**: Append-only raw data ingestion.
  - **Silver**: Cleaned, conformed data (data quality, schema enforcement).
  - **Gold**: Curated star/snowflake marts optimized for BI workloads.
- **Considerations**:
  - Data lineage and traceability across zones.
  - Cost of maintaining multiple zones vs query performance gains.
  - Data freshness requirements in each layer.
- **Reference**: This medallion layering is a common best practice in modern lakehouse implementations and aids governance and performance.

1.2.4 Table Format Layer (Delta / Iceberg / Hudi)

- **Description**: Core metadata + transactional layer offering ACID, time travel, schema enforcement, and versioning built atop object storage.
- **Considerations**:

  - ACID compliance enables reliable updates/deletes on immutable storage.
  - Metadata performance at scale—partition pruning, indexing.
  - Choosing between open formats like Delta, Iceberg, or Hudi depending on ecosystem and features.
- **References**:
  - Delta Lake provides ACID, time travel, etc. (GlobalLogic, Academia).
  - Foundational theory on lakehouses hybridizing warehouses and lakes (SpringerLink).

1.2.5 Compute & Access Layer (BI, ML, SQL)

- **Description**: Compute engines like Spark, Databricks, Trino for SQL; BI tools like Looker, Power BI; ML workflows via notebooks, Feature Store, MLflow.
- **Considerations**:
  - Ensuring performance and concurrency at scale.
  - Semantic models vs raw access—who owns the semantic layer?
  - Ensuring model reproducibility and lineage.
- **Reference**: The lakehouse unifies OLAP and ML workloads over one platform (cidrdb.org, SpringerLink, arXiv).

1.2.6 Governance & Security

- **Description**: Catalogs (Unity Catalog), RBAC/ABAC, data lineage, PII masking, auditing, and data sharing (e.g., Delta Sharing).
- **Considerations**:
  - Balancing openness vs compliance.
  - Tracking lineage across transformations.
  - Fine-grained access controls on files and tables.
- **References**: Metadata and governance layers elevate lakehouse above traditional data lakes (SpringerLink).

1.2.7 Orchestration & Ops

- **Description**: Pipelines scheduled via Airflow or ADF, CI/CD for deployments, monitoring, observability, and cost tracking.
- **Considerations**:
  - Data pipeline reliability and failure recovery.
  - Automation of schema changes and deployments.
  - Cost visibility and optimization at compute and storage layers.

---

2 Supporting Evidence & Case Studies

- **"Lakehouse: A New Generation…" (Armbrust et al., 2021)**: Introduced the lakehouse concept with open formats and ML support (cidrdb.org).
- **"The Data Lakehouse: Data Warehousing and More" (Mazumdar et al., 2023)**: Detailed how lakehouses unify system strengths (arXiv).
- **Schneider et al. (2024)**: Surveyed lakehouse concepts and technologies, showing maturity and use cases (SpringerLink).
- **Sawadogo & Darmont (2021)**: Highlighted data lake metadata challenges, underscoring need for governance in lakehouse (arXiv).
- **Harby et al. (2025)**: Surveyed ingestion, transformation, and analytics in lakehouse environments (ScienceDirect).
- **GlobalLogic Delta Lake white paper (2024)**: Practical overview of Delta Lake implementation and benefits (GlobalLogic).
- **Dell Technologies lakehouse white paper (2023)**: Real-world validated lakehouse with Delta/Iceberg on OpenShift (Dell).

---

3 Conclusion

Putting it all together: This architecture gives you the **flexibility of a data lake** with the **transactional reliability and performance of a warehouse**, wrapped in governance, orchestration, and modern compute access layers. It's a clean, scalable design that serves both analytic BI and emergent ML workloads—everything you need and nothing you don't.

4 Data Lakehouse Glossary

Plain-English, Technical, and Scenario views

Generated: 11 August 2025 • Author: Iain Toolin

4.1 Sources & Ingestion

Term	Non-technical	Technical	Illustrative scenario
Batch ingestion	Load data in chunks on a schedule.	Periodic bulk loads via ETL/ELT into storage; no event-time guarantees.	Nightly ERP extracts land at 01:00 into Bronze.
Streaming ingestion	Data drips in constantly.	Low-latency, record-at-a-time ingestion (Kafka/Event Hubs) with exactly/at-least-once semantics.	Clickstream and IoT readings feed the lake in near-real time.
CDC (Change Data Capture)	"Only send what changed."	Log-based capture of inserts/updates/deletes from sources; ordered change stream.	S/4HANA sales order updates replicate to Silver within minutes.
ETL / ELT	Prep the ingredients / cook in the pan.	Transform before load (ETL) vs. load then transform in-lake (ELT).	Raw CSVs land first; Spark jobs standardise later (ELT).
Kafka / Event Hubs	The conveyor belts.	Distributed logs for durable, scalable streaming pipelines.	Orders topic partitions by region to parallelise consumers.

4.2 Storage & Table Formats

Term	Non-technical	Technical	Illustrative scenario
Object storage (ADLS/S3/GCS)	Cheap, bottomless filing cabinet.	Durable blob store; eventual consistency; lifecycle & tiering.	Raw and curated data live in ADLS with lifecycle to cool/archive.
Data lake	Big, flexible data dump.	Schema-on-read repository for diverse file types.	PDFs, JSON logs, and parquet sit side-by-side for later use.
Data lakehouse	Lake + warehouse in one.	Lake on object storage plus table format giving ACID/SQL semantics.	BI dashboards and ML models run on the same curated tables.
Table format (Delta/Iceberg/Hudi)	The index + rules for files.	Transaction log & metadata layer enabling ACID, time travel, schema ops.	Analysts query a Delta table as if it were a database table.
Apache Iceberg	Big tables without the pain.	Spec-driven table format with hidden	Historic queries read only referenced

		partitioning & metadata trees.	snapshots—fast & cheap.
Apache Hudi	Jackets for changing data.	Upserts, deletes, incremental pulls on lake tables.	GDPR delete requests propagate as Hudi deletes to Parquet files.
Delta Lake	ACID for your lake.	Optimistic concurrency, commit log, time travel, schema enforcement.	Restore last week's version after a dodgy transform—no panic.
ACID transactions	No half-baked writes.	Atomicity, Consistency, Isolation, Durability on lake tables.	Concurrent jobs write safely without corrupting the table.
Time travel	Roll back the clock.	Snapshot-based reads by version or timestamp.	Investigate a KPI drift by querying "as of" last month.
Schema enforcement/evolution	Keep shapes consistent (but adaptable).	Enforce column types; allow controlled add/rename with history.	New "marketing_region" column added without breaking readers.
Partitioning / pruning	Slice the cake; only eat your slice.	Data layout by keys; engines skip non-matching partitions.	Query "2025-08" only touches that month's folders.

4.3 Transformation & Zones

Term	Non-technical	Technical	Illustrative scenario
Medallion architecture	Bronze → Silver → Gold conveyor.	Raw → cleansed/conformed → curated marts for analytics.	Logs land in Bronze, standardised in Silver, joined to facts in Gold.
Bronze	"Just land it."	Append-only raw, minimally processed.	Dump supplier files untouched for traceability.
Silver	"Make it usable."	Cleaned, typed, deduped, conformed to shared dimensions.	Customer IDs reconciled across CRM and ERP.
Gold	"Make it consumable."	Star/snowflake models, performance-tuned aggregates.	Finance dashboard reads a Gold "sales_summary" table.
Star schema	Simple hub-and-spokes.	Central fact table with denormalised dimensions.	FactSales joins to DimCustomer/DimProduct for fast BI.
Snowflake schema	Star with tidy cupboards.	Normalised dimensions to reduce redundancy.	DimCustomer splits into Customer ↔ Geography ↔ Industry.

4.4 Compute & Access

Term	Non-technical	Technical	Illustrative scenario
⚡ Apache Spark	Big data workhorse.	Distributed compute for SQL, batch, streaming, ML.	A Spark job standardises 2 TB of logs in 6 minutes.
💎 Databricks	Spark with guardrails.	Managed platform: notebooks, clusters, Delta, MLflow, governance hooks.	One workspace runs jobs, notebooks, and dashboards over Unity Catalog.
🧭 Trino	Any-lake SQL.	MPP SQL engine with connectors across lakes/warehouses.	Ad-hoc joins across Delta tables and a Postgres ref set.
🧠 Semantic model	Agreed business definitions.	Logical layer of measures/dimensions over physical tables.	"Gross margin" calculated once, reused everywhere.
📊 Power BI / Looker	The shop window.	BI tools for modelling, viz, and scheduled reports.	Exec scorecards pull from Gold models hourly.
🧰 Feature Store	Reusable ML ingredients.	Versioned, governed feature definitions with offline/online sync.	"30-day order frequency" feature powers churn and upsell models.
🧪 MLflow	ML lab notebook.	Experiment tracking, model registry, reproducible runs.	Register best XGBoost model and promote to "Production" stage.

4.5 Governance & Security

Term	Non-technical	Technical	Illustrative scenario
📦 Unity Catalog	Central permissions & lineage.	Cross-workspace metastore, RBAC, data lineage, audits.	Data stewards grant "SELECT on catalog.sales to Finance_Analysts".
🔐 RBAC / ABAC	Who can see what (and why).	Role- or attribute-based policy evaluation on objects/rows/cols.	Only UK analysts can read UK rows via policy tags.
🧬 Data lineage	Follow the breadcrumbs.	Column/table-level provenance across jobs and versions.	Trace a KPI to the upstream CDC topic and source table.
🙀 PII masking	Blur sensitive bits.	Dynamic data masking/tokenisation at query time.	Names appear masked unless the user has "PII_View" role.
🔗 Delta Sharing	Share without copies.	Open protocol to share tables securely across platforms.	Supplier reads your "on-time_delivery" table live from your lake.
🔒 Encryption (at rest/in transit)	Lock the doors.	KMS-managed encryption for storage and TLS for data in flight.	Keys rotated quarterly; TLS enforced end-to-end.

4.6 Orchestration & Ops

Term	Non-technical	Technical	Illustrative scenario
Airflow	The scheduler.	DAG-based orchestration with retries, sensors, SLAs.	Bronze→Silver→Gold runs as a chained DAG with alerts.
Azure Data Factory (ADF)	Pipelines without fuss.	Managed pipelines, mappings, triggers, integration runtimes.	Copy activity pulls SAP data; ADF triggers a Spark notebook.
CI/CD (data/infra)	Promote with discipline.	Git-based versioning, tests, approvals, and automated deploys.	PR adds a new Silver table; pipeline runs unit + data quality tests.
Observability	Know when it's wobbling.	Metrics, logs, lineage, data quality SLAs, cost telemetry.	Alert fires when late-arriving files breach a 15-min SLO.
FinOps / cost governance	Keep the tab sane.	Showback/chargeback, auto-stop, right-sizing, storage tiering.	Idle clusters auto-terminate; cold data moves to archive.
Data quality rules	Trust, but verify.	Constraint checks (nulls, ranges, freshness) with fail/alert actions.	Job quarantines records with negative quantities.

Notes: Icons are for 'scan ability'.

Icon	Label	Meaning in this glossary	Used for
	Batch ingestion	Scheduled, chunked loads.	Nightly dumps/ETL landings.
	Streaming ingestion	Always-on feeds, low latency.	Kafka/Event Hubs streams.
	Change/Flow	Movement of data or pipelines.	CDC; ADF pipelines (context decides).
	Transform/Test/Experiment	Data cooking or ML experiments & checks.	ETL/ELT; MLflow; Data quality rules.
	Message bus	Event conveyor belt.	Kafka / Event Hubs topics.
	Object storage	Blob stores.	ADLS/S3/GCS buckets/containers.
	Data lake	Raw, flexible repository.	Schema-on-read zone.
	Lakehouse	Lake + warehouse model.	ACID tables in object storage.
	Table format	Metadata/transaction layer.	Delta/Iceberg/Hudi tables.
	Apache Iceberg	Table format flavour.	Hidden partitioning, snapshots.
	Apache Hudi	Upsert-friendly tables.	Incrementals, deletes.
	Delta Lake	ACID on the lake.	Time travel, schema control.
	ACID	Safe, consistent writes.	Transactions.
	Time travel	Point-in-time reads.	Debug/rollback.
	Schema control	Enforce/evolve shapes.	Types/columns management.

Icon	Label	Meaning in this glossary	Used for
	Partitioning	Prune what you don’t need.	Faster scans.
	Medallion	Bronze → Silver → Gold.	Zone progression.
	Bronze/Silver/Gold	Raw → cleansed → curated.	Zone tiers.
	Star schema	Simple facts + dims.	BI models.
	Snowflake schema	Normalised dims.	Larger models.
	Apache Spark	Distributed compute.	Batch/stream/ML jobs.
	Databricks	Managed Spark + governance.	Notebooks, clusters, Unity.
	Trino	SQL over many sources.	Ad-hoc queries.
	Semantic model	Shared definitions.	Metrics/dimensions.
	BI tools	Visuals & reports.	Power BI / Looker.
	Feature Store	Reusable ML features.	Online/offline features.
	Unity Catalog	Centralised governance.	RBAC, lineage.
	RBAC/ABAC	Who can see what.	Access policies.
	Lineage	Data provenance.	Traceability.
	PII masking	Hide sensitive bits.	Dynamic masking/tokenisation.
	Delta Sharing	Share without copies.	External sharing.
	Encryption	At rest & in transit.	KMS/TLS.
	Airflow	Orchestration.	DAG scheduling.
	CI/CD	Promote safely.	Versioned deploys.
	Observability	Watch health/cost/SLAs.	Alerts, metrics.
	FinOps	Cost discipline.	Showback/rightsizing.