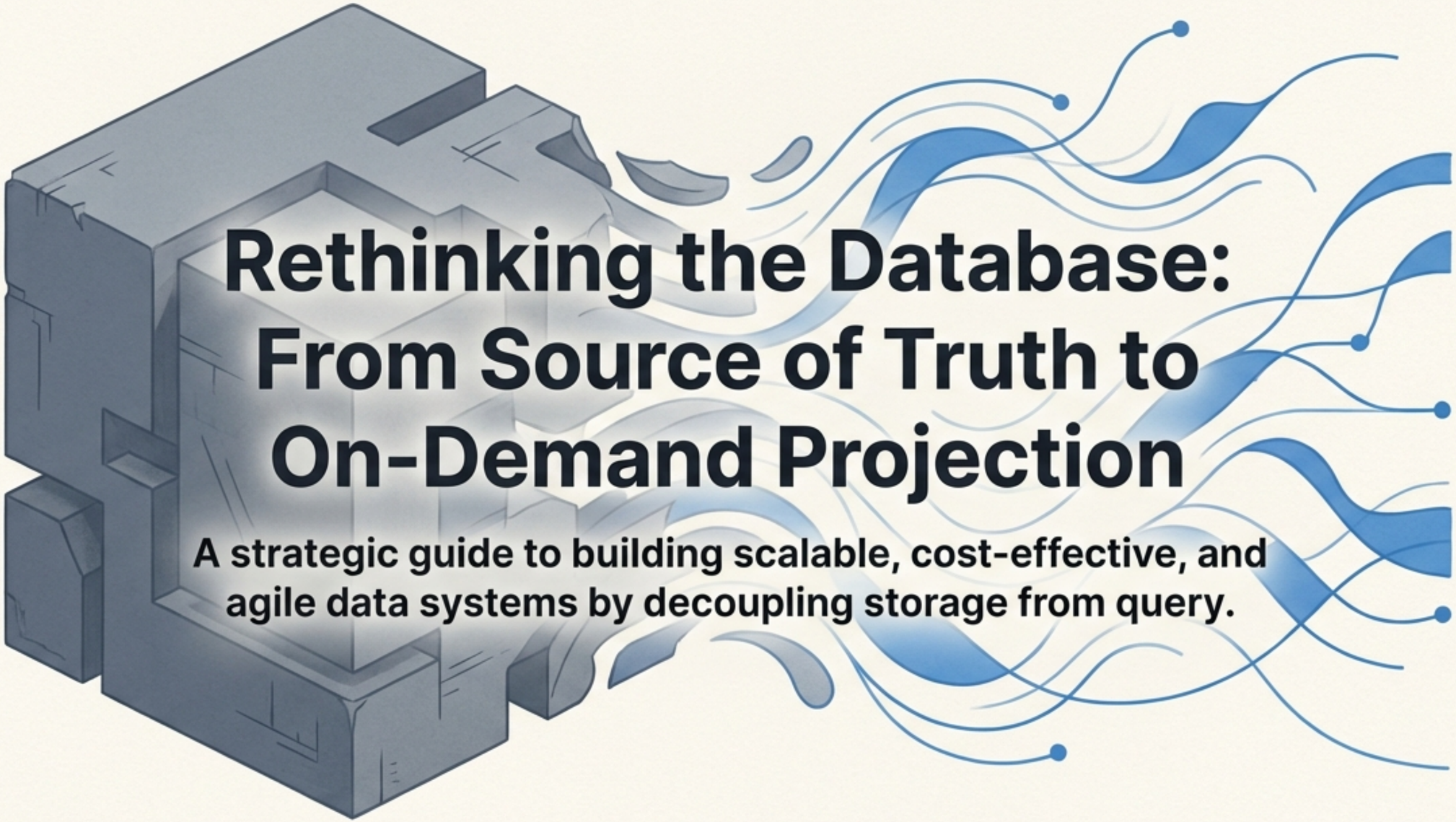
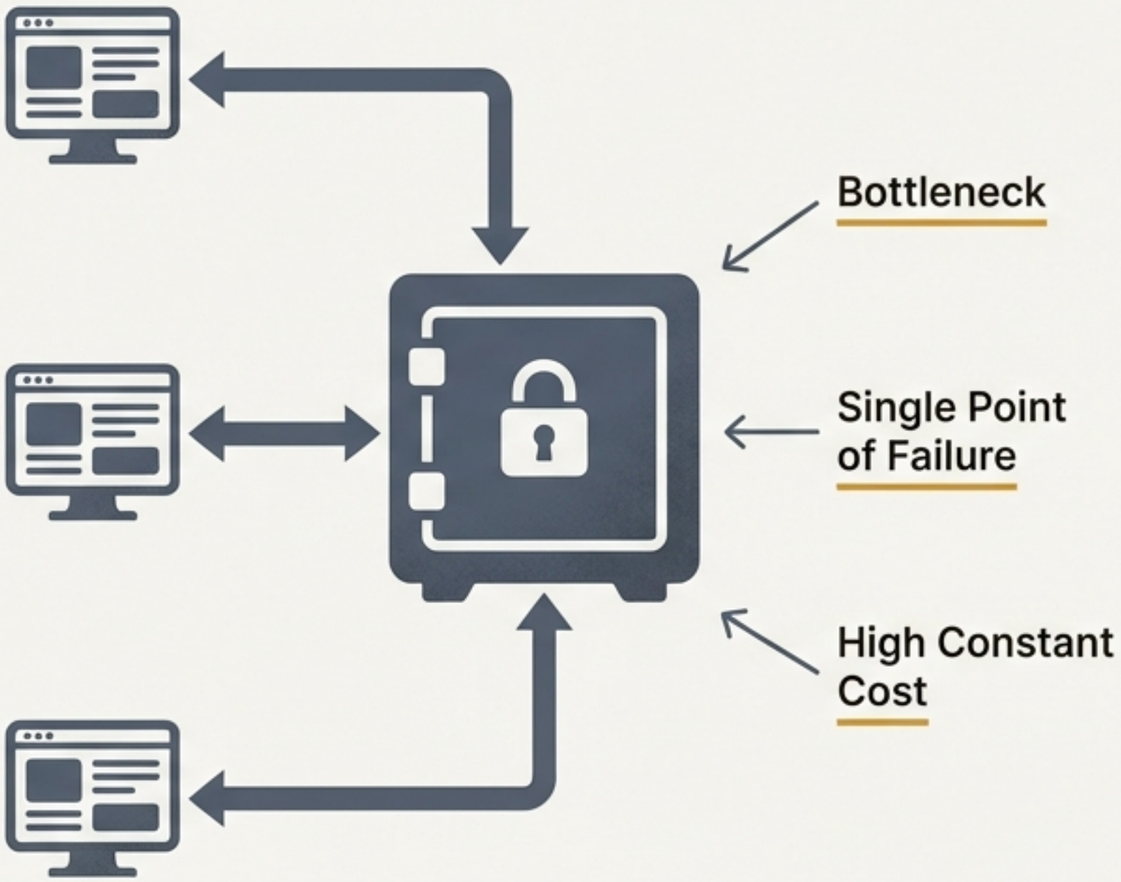


Rethinking the Database: From Source of Truth to On-Demand Projection

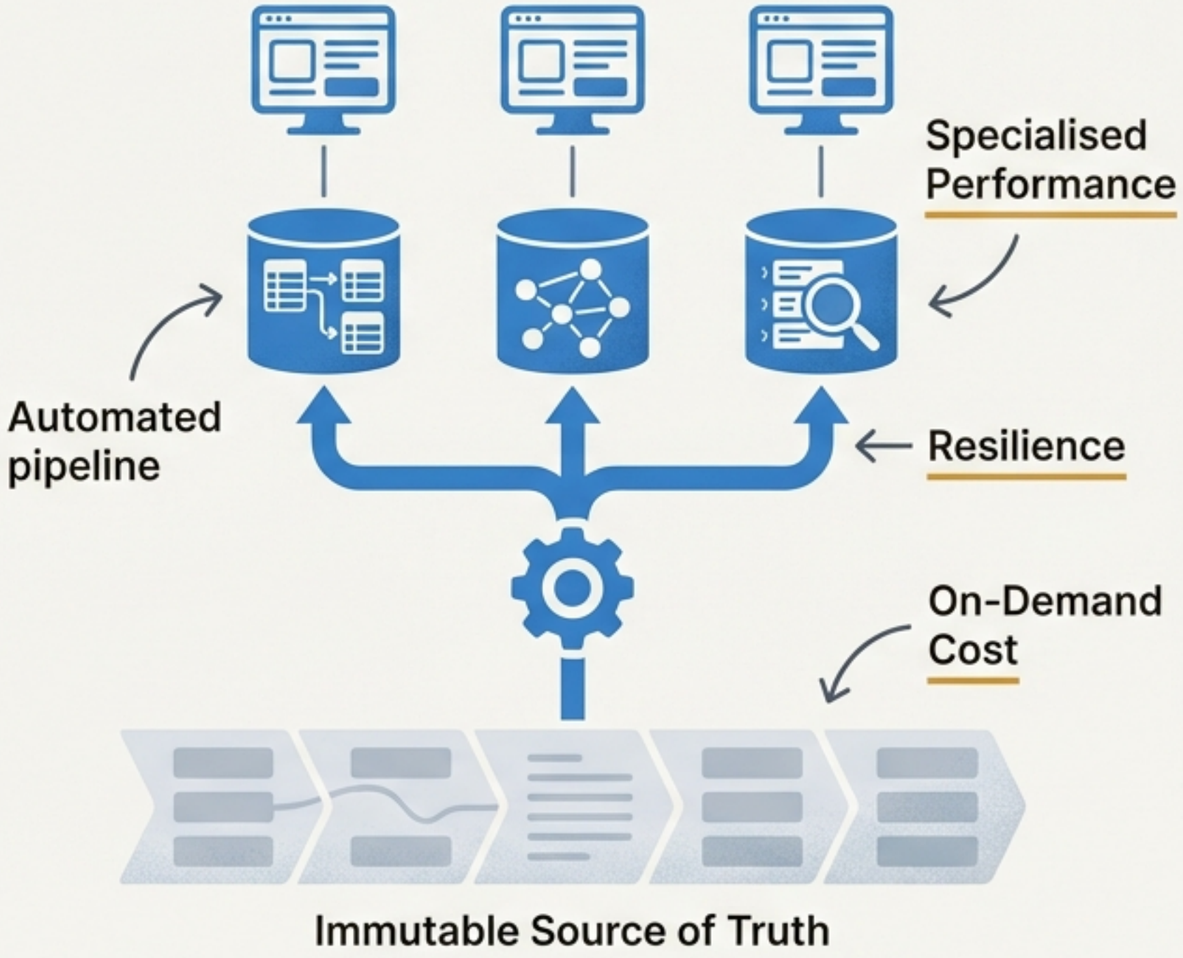
A strategic guide to building scalable, cost-effective, and agile data systems by decoupling storage from query.

The Core Architectural Shift

Traditional Model: The Database as Master

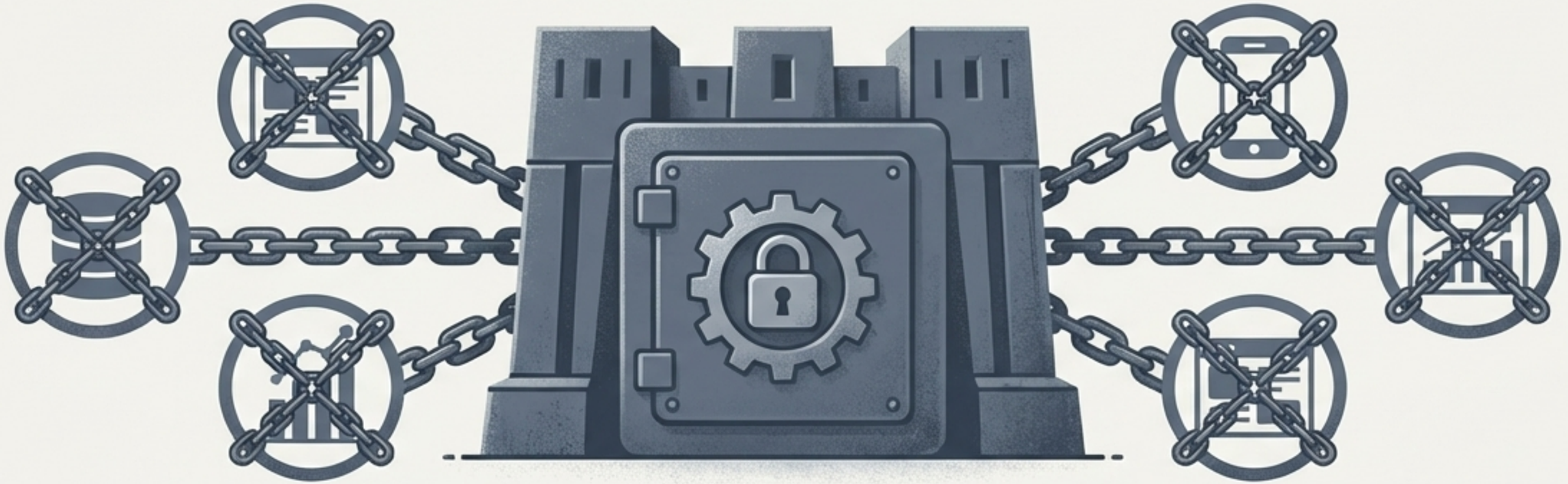


Projection Model: The Database as Servant



The Traditional World: When the Database is the Single Source of Truth

For decades, the authoritative state of our systems has lived inside a single DBMS. While familiar, this “one-size-fits-all” approach is becoming a liability as data volume and variety grow.



Single Point of Failure & Complexity: The entire system is at risk. Scaling or migrating a database containing a superset of all data needs is a complex and high-risk operation.

High Constant Costs: Significant compute, memory, and licensing costs are incurred 24/7, even when the database is idle.

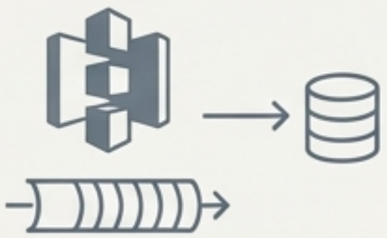
Unnecessary Data Bloat: Databases store far more data than any single use-case requires, consuming expensive resources for infrequently used or historical data.

Lack of Ephemeral Environments: Stateful, long-lived databases hinder modern serverless and on-demand computing. They cannot be easily spun up or torn down as needed.

The New Paradigm: Treating Databases as Data Projections

Decouple the data's storage from its query engine.

In this model, the database is no longer the permanent owner of data. It becomes a flexible, disposable, and purpose-built query layer. The database is a generated view, a reproducible cache.

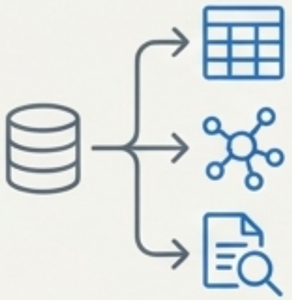


Single Source of Truth Elsewhere

The authoritative copy resides in scalable storage (e.g., an object store like S3, or an immutable event log like Kafka).

Databases as Derived Views

Projections are populated programmatically from the source. If a database is lost, it can be recreated.



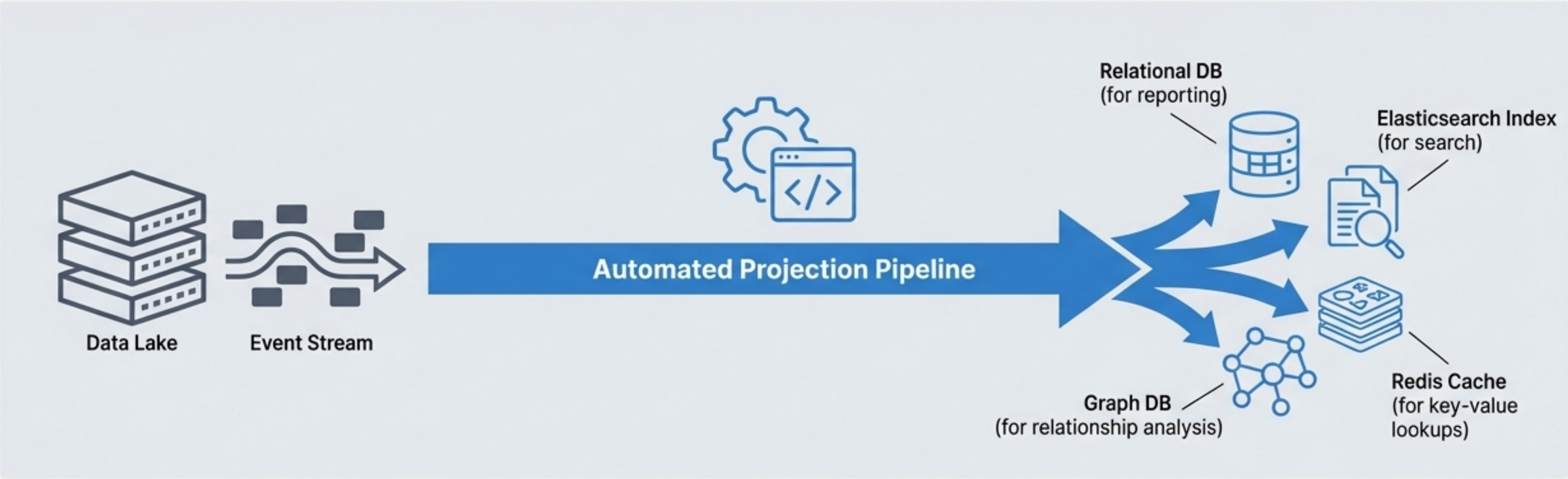
Multiple, Specialised Databases

Use the right tool for the job: a relational DB for transactions, a graph DB for analytics, a search index for search—all from the same source.

Ephemeral & On-Demand Usage

Spin up database instances when needed and tear them down after use, eliminating the need for an always-on server.

Anatomy of the Projection Model



The durable, authoritative, primary copy of all data. Stored in raw or semi-structured, query-unfriendly formats (e.g., JSON, Parquet, Avro).

Repeatable, testable code (batch ETL, stream consumers) that transforms source data into query-optimised formats. This is 'Infrastructure-as-Code' for data.

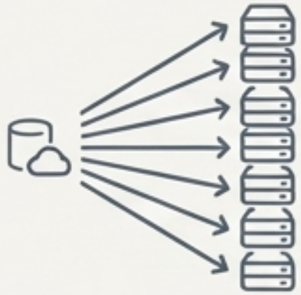
Multiple, disposable databases, each tailored to a specific query pattern. They are caches that can be rebuilt on demand.

The Payoff: Unlocking Cost Efficiency, Scalability, and Flexibility

Cost Efficiency

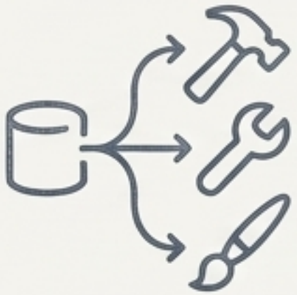
Store raw data in economical object stores, not expensive databases. Pay for high-performance compute only when running queries.

> Pay for processing only when it runs. No costs incurred when the database is idle (since it isn't running at all).
> — Dinis Cruz, on on-demand graph analytics.



Scalability & Elasticity

The storage layer scales independently and almost infinitely. Handle bursty workloads by spinning up dozens of database database instances in parallel—a horizontal scaling model not possible with a monolithic DB.



Polyglot Persistence (The Right Tool for the Job)

Use the optimal data store for each access pattern without fragmenting the source of truth. Leads to faster queries and more flexible feature development.

> Model once, represent everywhere. — Netflix's Unified Data Architecture philosophy.

Strategic Wins: Improving Resilience, Quality, and Team Autonomy

Resilience & Reproducibility

Since projections are built from source via automated pipelines, disaster recovery becomes a simple matter of regeneration.

> **If a database instance crashes or its data becomes suspect, you can confidently rebuild it from scratch. No complex backup restores needed.**

Improved Data Quality & Testing

Treat data transformations as repeatable, testable code. Enforce schemas and run data quality checks on every build, catching anomalies early.

> **Data tests can be treated like unit tests for code, automatically verifying the integrity of each projection upon creation.**

Domain Autonomy & Faster Changes

Decouple schemas to avoid the "one big database" bottleneck. Teams can evolve their own domain-specific projections without impacting others, accelerating development.

This Isn’t Theory: It’s Already Happening in Practice

Event Sourcing & CQRS

The event log is the immutable source of truth. Queryable state is derived by projecting events into one or more disposable read databases.

Source: AWS Prescriptive Guidance

Data Lakehouse Architectures

Durable storage (S3, HDFS) in open formats is the base. Warehouse compute clusters are an effective projection layer.

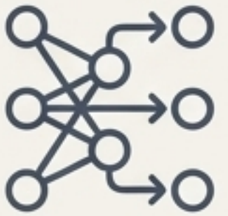
Source: Industry Analysis

Serverless Analytics

Services like Amazon Athena and Google BigQuery query data directly from object storage without a persistent DB server, proving the ephemeral model at scale.

Domain-Oriented Data Mesh

Advocates for domain teams owning their data pipelines and creating their own optimised databases (projections) from common raw data sources.



A Strategic Roadmap for Implementation

1

Establish an Immutable, Central Source of Truth

Identify a scalable storage solution (cloud object store, event streaming platform) and standardise on open formats (Parquet, Avro). Applications may need to shift from writing to a DB to writing an event or a file.

2

Automate Projection Pipelines

Create reproducible processes (batch ETL, streaming) to build or refresh databases. Use containerisation (Docker, Kubernetes) and IaC to treat databases as build artifacts, just like software.

3

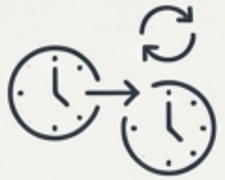
Leverage Virtual Query Technologies

For on-the-fly needs, use engines like Presto/Trino or Spark SQL that can query raw data in place, acting as a transient projection without the need for materialisation.

4

Design for Eventual Consistency

Embrace the slight delay between data arriving at the source and appearing in projections. Design idempotent, retryable update processes to ensure robustness.



Realities and Considerations: Acknowledging the Challenges

The Challenge	The Mitigation Strategy
Initial Complexity. Requires building and maintaining data pipelines and new infrastructure.	Start with a hybrid approach. Use modern data orchestration platforms and automation to manage pipelines effectively.
Update Latency. There will be a delay between source updates and projection availability.	A worthwhile trade-off for most read-heavy scenarios. Design for asynchronous updates and use fast streaming where needed.
Consistency & Reconciliation. Projections could become out of sync with the source due to bugs or failures.	Implement automated monitoring and reconciliation (e.g., comparing counts/checksums). The ability to trigger a full rebuild from scratch is the ultimate safety net.

The Most Important Shift Is in Your Mindset

Old Mindset: The Database is the Asset

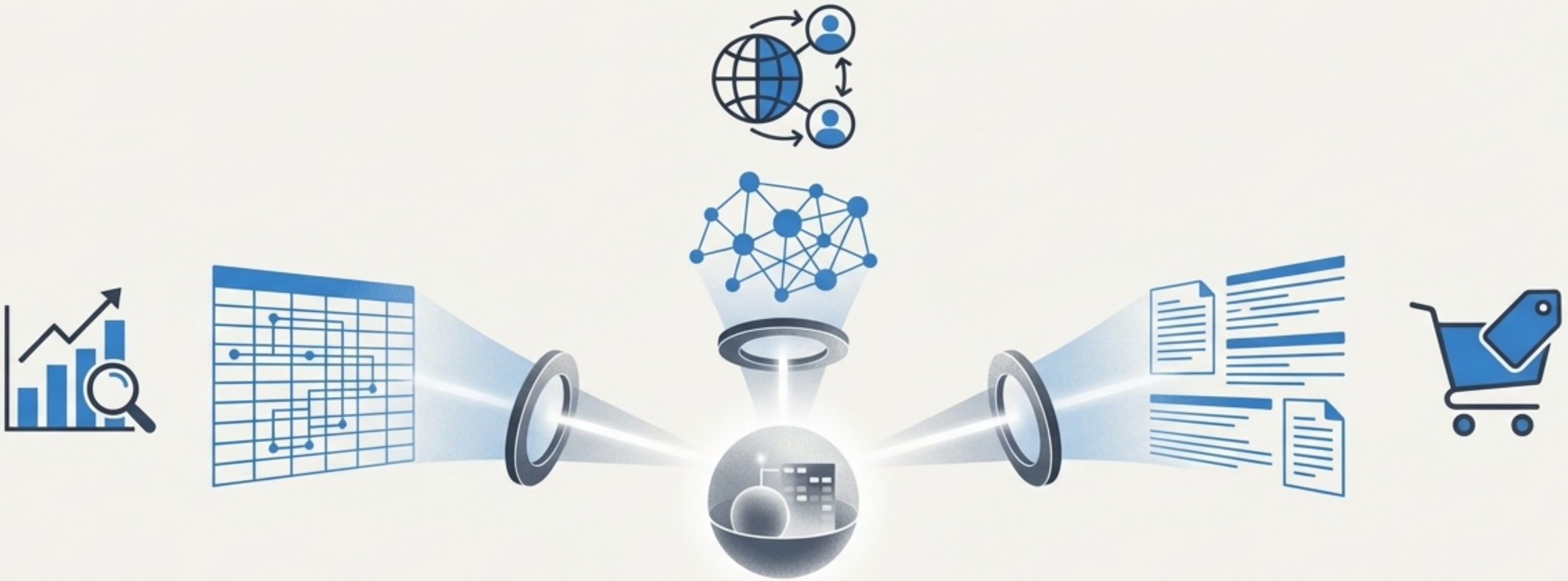
Protect the database server.

New Mindset: The Data is the Asset

Automate the projection pipeline.

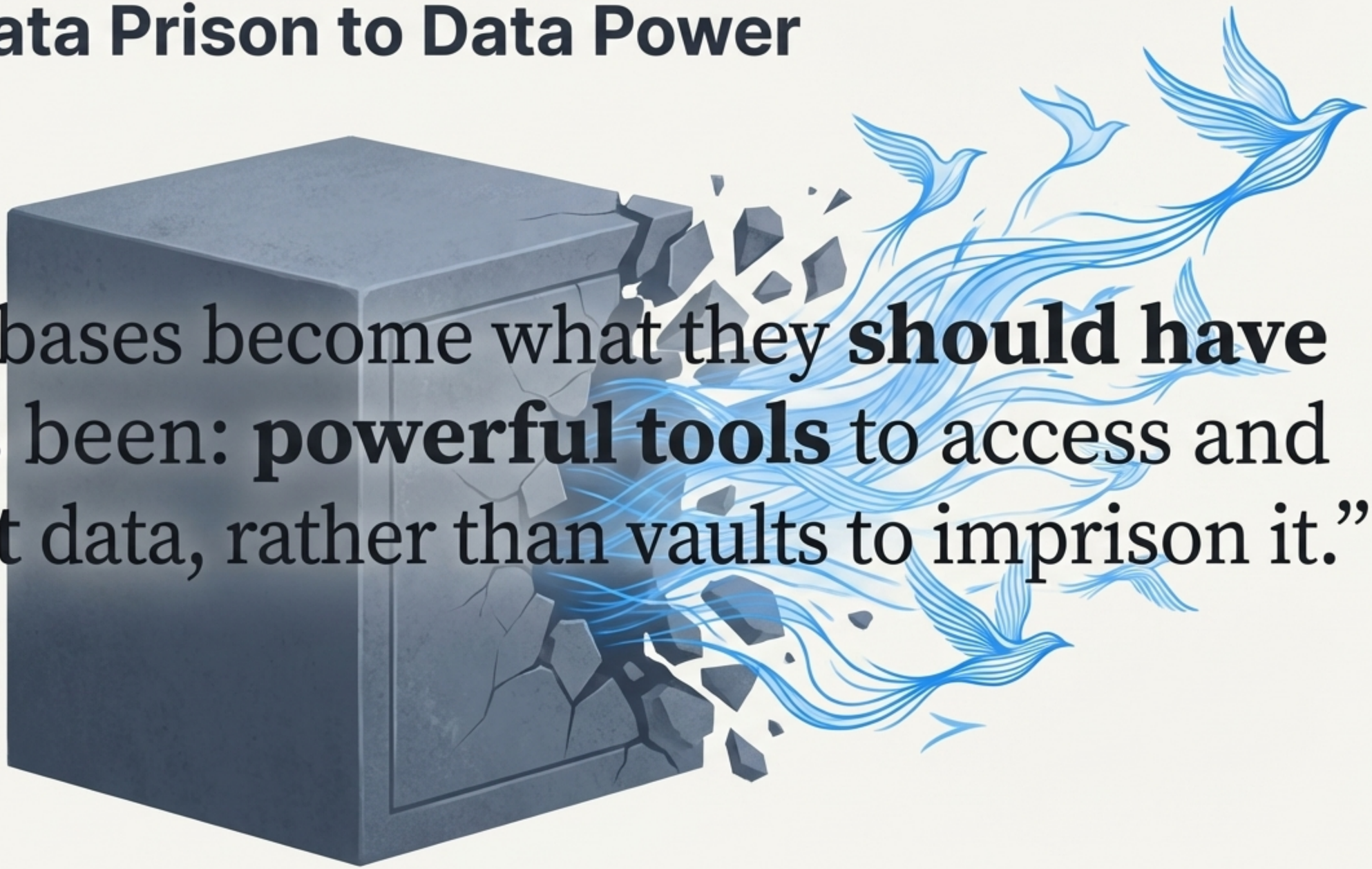
Treat your data as a **first-class asset**, independent of any particular database technology. Databases are powerful but replaceable tools.

The Vision: Model Data Once, Use It Everywhere



By separating the canonical data model from its many representations, you can eliminate data silos and bottlenecks. This allows your organisation to build a data infrastructure that is modular, resilient, and aligned with the fast-changing needs of modern applications.

From Data Prison to Data Power



"...databases become what they **should have** always been: **powerful tools** to access and **project** data, rather than vaults to imprison it."