Architecture: Databases

Relational

Bare Metal

Lift and shift Oracle workloads to Google Cloud

Cloud SQL

Managed MySQL, PostgreSQL, and SQL Server

Cloud Spanner

Cloud-native with unlimited scale, consistency, and 99.999% availability

Non- relational / NoSQL

(Key Value)

Cloud BigTable

Cloud-native NoSQL wide-column store for large scale, low-latency workloads

(Document)

Firestore

Cloud-native Serverless scalable document store

Firestore Realtime Database

Store and sync data in real time

(In-memory)

Memorystore

Fully managed Redis and Memcached for sub-millisecond data access

(Additional NoSQL)

mongoDB. Atlas

Global cloud database service for modern applications

Partners

Managed offerings from open-source partner network, including MongoDB, Datastax, Redis Labs, and Neo4j.

redis
mongoDB
elastic
influxdb
DATASTAX
neo4j
confluent
MariaDB

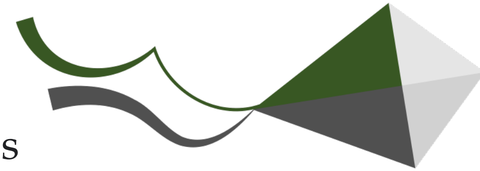
Source: https://cloud.google.com/docs, https://cloud.google.com/products

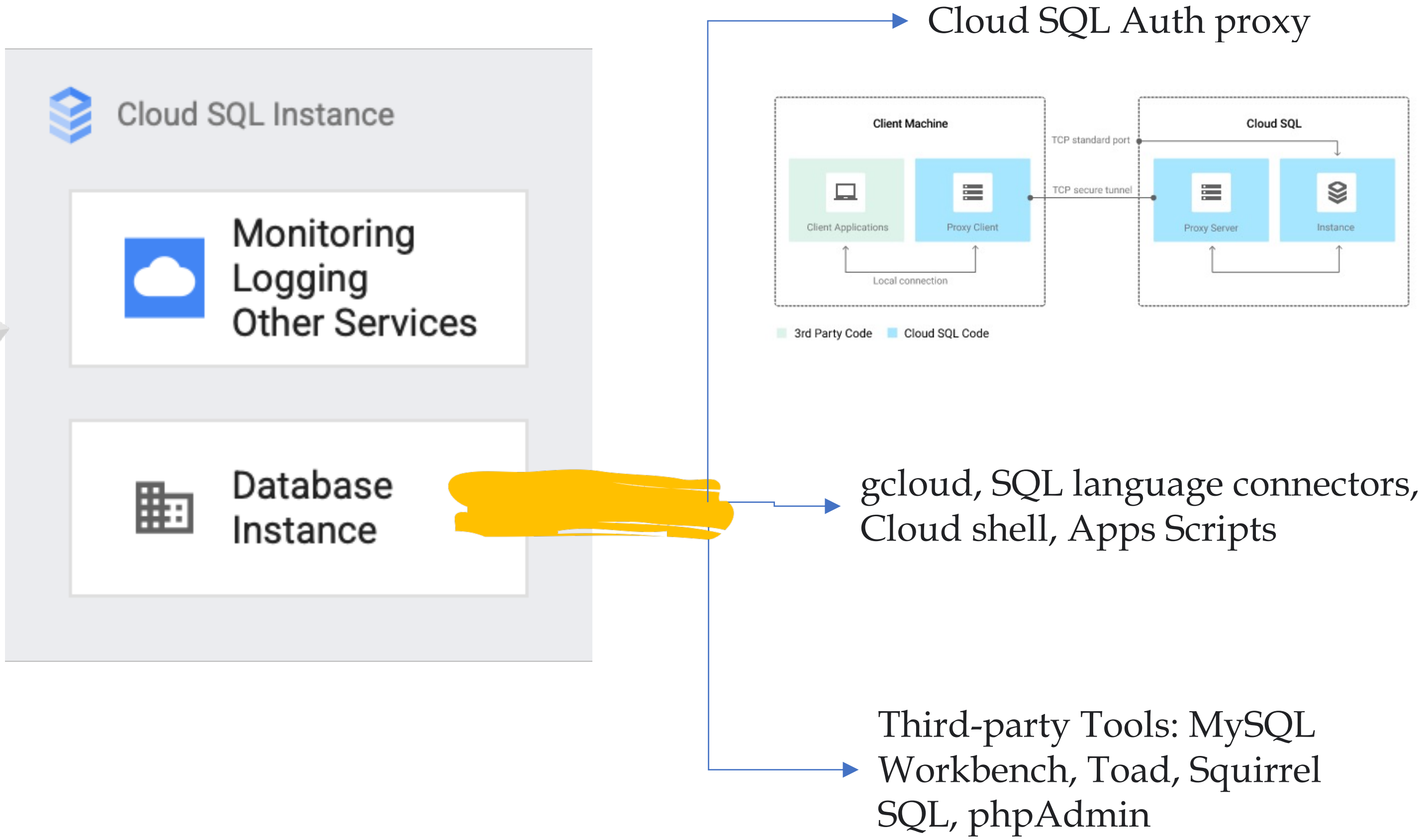
Databases: Cloud SQL

Fully managed relational database service for MySQL, PostgreSQL, and SQL Server.

Administrative overhead of SQL Databases:

- Backups
- High availability and failover
- Network connectivity
- Export and import
- Maintenance and updates
- Monitoring
- Logging





Cloud SQL Instance

Monitoring
Logging
Other Services

Database Instance

Cloud SQL Auth proxy

gcloud, SQL language connectors, Cloud shell, Apps Scripts

Third-party Tools: MySQL Workbench, Toad, Squirrel SQL, phpAdmin

Source: https://cloud.google.com/docs, https://cloud.google.com/products

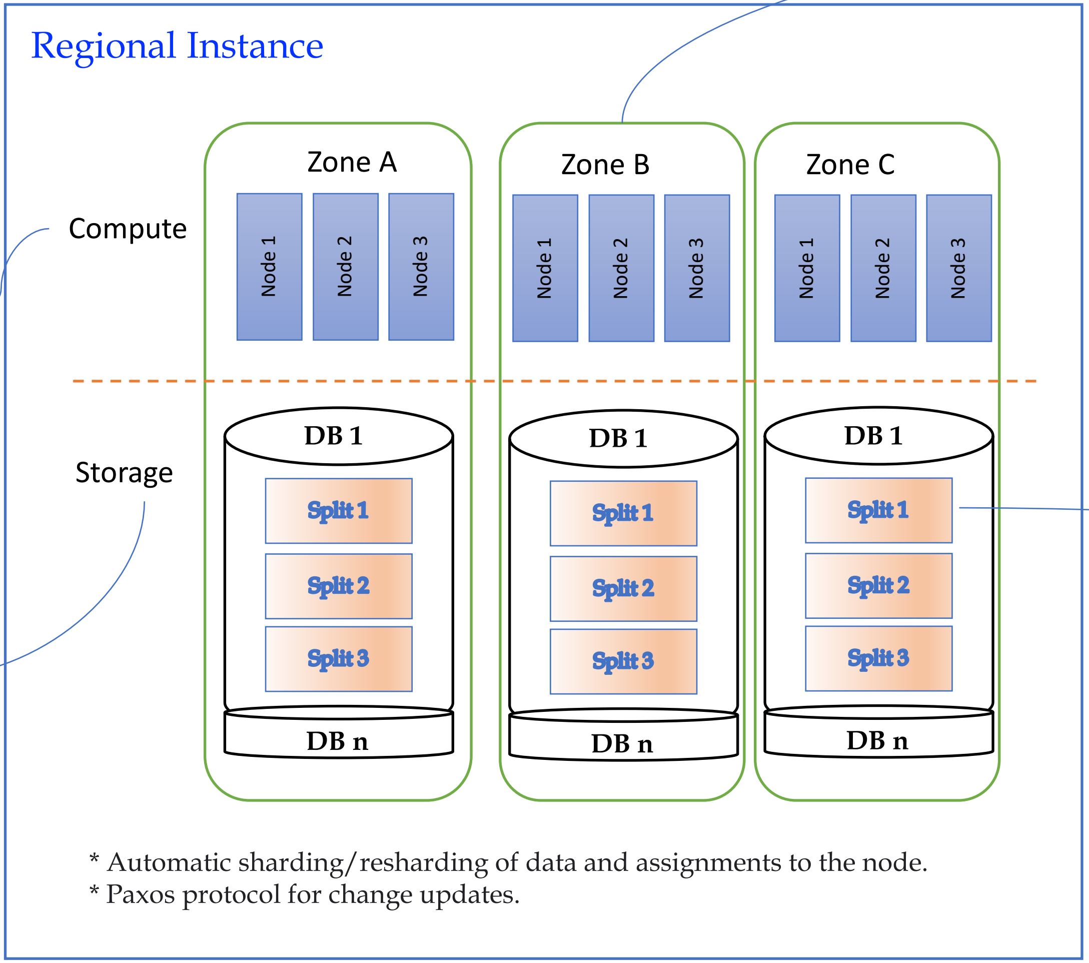
Databases: Cloud Spanner

Fully managed relational database with unlimited scale (horizontal scale across regions), strong consistency, and up to 99.999% availability.
Relational Semantics (Schemas, ACID transactions, SQL) + Horizontal Scale (99.999% SLA, fully managed)

Architecture overview:

Node servers serve the read and write/commit transaction requests, but they don't store the data.

The data is stored in Google's underlying Colossus distributed replicated file system.



* Automatic sharding/resharding of data and assignments to the node.
* Paxos protocol for change updates.

Everything is replicated across the three zones – the same goes for split management. Using Paxos consensus protocols, one of the zones is determined to be a leader. The leader is responsible for managing write transactions for that split, while the other replicas can be used for reads.

Each table in the database is stored sorted by primary key. Tables are divided by ranges of the primary key and these divisions are known as splits.

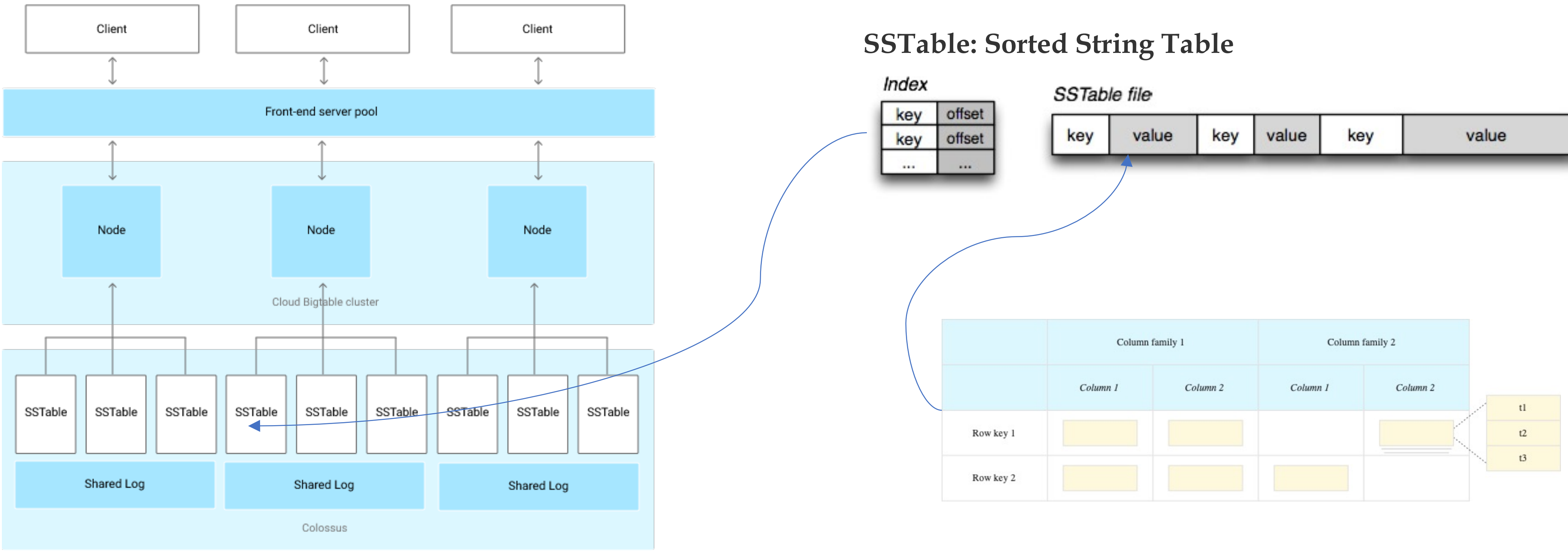
Source: https://cloud.google.com/docs, https://cloud.google.com/products

Databases: Cloud Bigtable

A fully managed, scalable NoSQL database service for large analytical and operational workloads with up to 99.999% availability.

- Consistent sub-10ms latency—handle millions of requests per second
- Ideal for use cases such as personalization, ad tech, fintech, digital media, and IoT

Bigtable architecture:



	Column family 1		Column family 2	
	Column 1	Column 2	Column 1	Column 2
Row key 1				
Row key 2				

Source: https://cloud.google.com/docs, https://cloud.google.com/products

Databases: Cloud Bigtable

Design Principle:

- ✓ RISC (Reduced Instruction Set Computing) - Simplify the operations.
- ✓ Cloud Bigtable is a learning system. - "hot spots"
- ✓ Bigtable is ideal for applications that need very high throughput and scalability for key/value (< 10 MB) data.
- ✓ Bigtable is not a relational database. It does not support SQL queries, joins, or multi-row transactions.

OLTP: **OLAP:** **Document:** **In-Memory:** **Real-time Sync:**

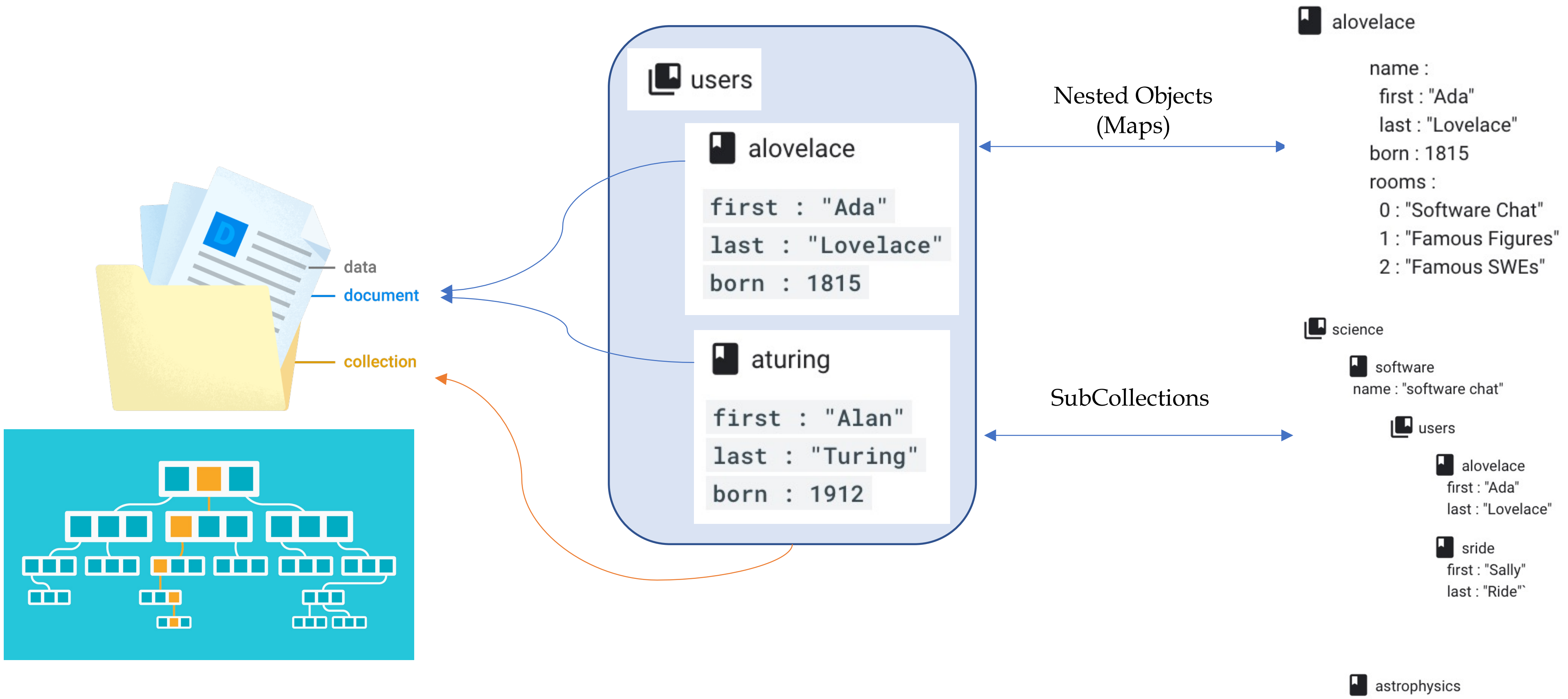
Bigtable use cases:

- **Time-series data**, such as CPU and memory usage over time for multiple servers.
- **Marketing data**, such as purchase histories and customer preferences.
- **Financial data**, such as transaction histories, stock prices, and currency exchange rates.
- **Internet of Things data**, such as usage reports from energy meters and home appliances.
- **Graph data**, such as information about how users are connected to one another.

Source: https://cloud.google.com/docs, https://cloud.google.com/products

Databases: Firestore

Firestore is a NoSQL, document-oriented database. Unlike a SQL database, there are no tables or rows. Instead, data is stored in documents, which are organized into collections.



```
users

  alovelace
    first : "Ada"
    last : "Lovelace"
    born : 1815

  aturing
    first : "Alan"
    last : "Turing"
    born : 1912
```

```
alovelace
    name :
      first : "Ada"
      last : "Lovelace"
    born : 1815
    rooms :
      0 : "Software Chat"
      1 : "Famous Figures"
      2 : "Famous SWEs"
```

```
science
    software
    name : "software chat"
        users
            alovelace
            first : "Ada"
            last : "Lovelace"

            sride
            first : "Sally"
            last : "Ride"`

    astrophysics
        ...
```

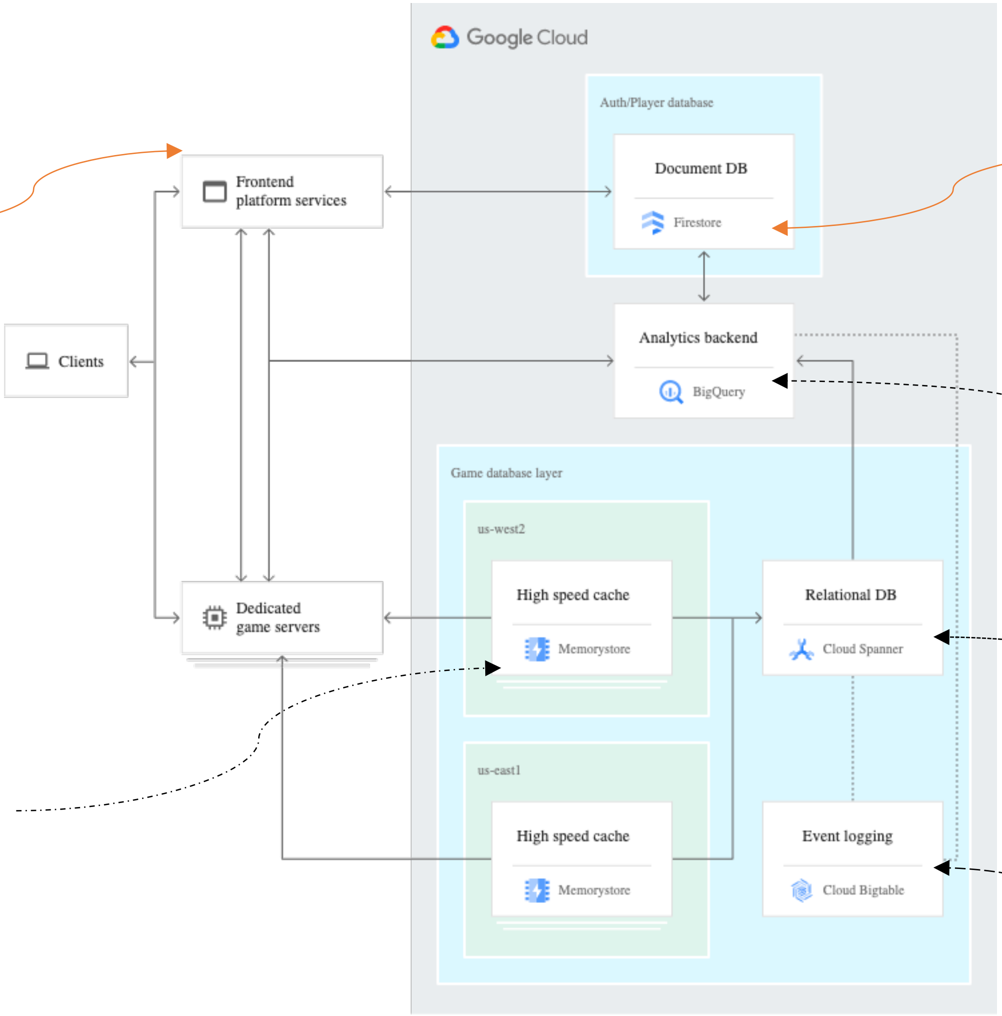
*Use Cases: **User Profiles, Real-time inventories, User session management, Distributed counters***

Source: https://cloud.google.com/docs, https://cloud.google.com/products

Databases: Firestore

Use Case: A Global multi-player gaming platform

The game's frontend services use Firestore to store billions of documents with hierarchical world state data.



Firestore holds user data like user configuration, party memberships, guilds, friends lists, and presence data.

Data from frontend and backend databases is regularly imported to BigQuery to run data analytics pipelines.

Spanner keeps inventory or match history for massive player populations anywhere in the world.

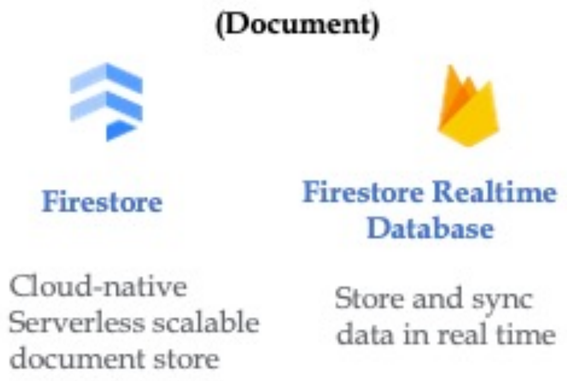
A regional in-memory cache is deployed on Memorystore for Redis to speed access to frequently used data.

Events are logged to Bigtable, where developers or support staff can access them for troubleshooting.

Source: https://cloud.google.com/docs, https://cloud.google.com/products

Databases: Firestore Realtime Database

The Firebase Realtime Database is a cloud-hosted NoSQL database.



(Document)

Firestore

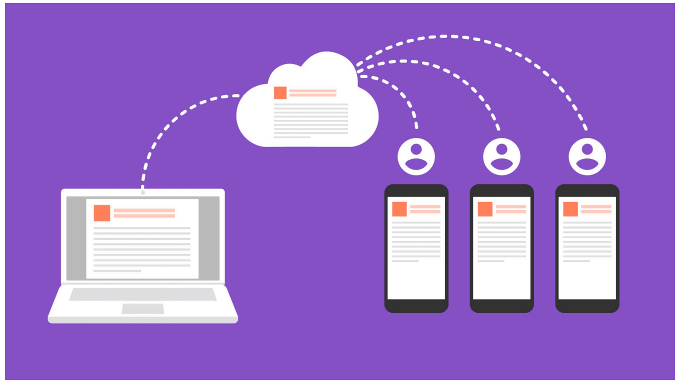
Cloud-native
Serverless scalable
document store

Firestore Realtime Database

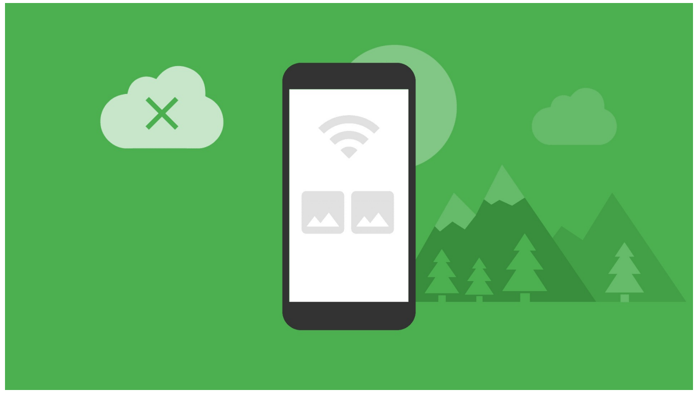
Store and sync
data in real time

Data is synchronized in real-time to every connected client. When you build cross-platform apps with iOS, Android, and JavaScript SDKs, all clients share one Realtime Database instance and automatically receive updates with the newest data.

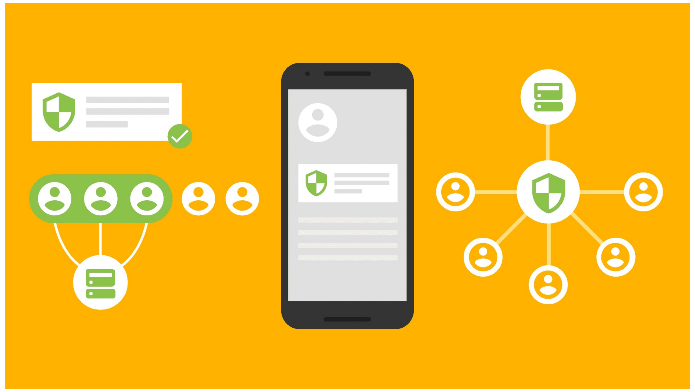
Collaboration across devices



offline use



user-based security



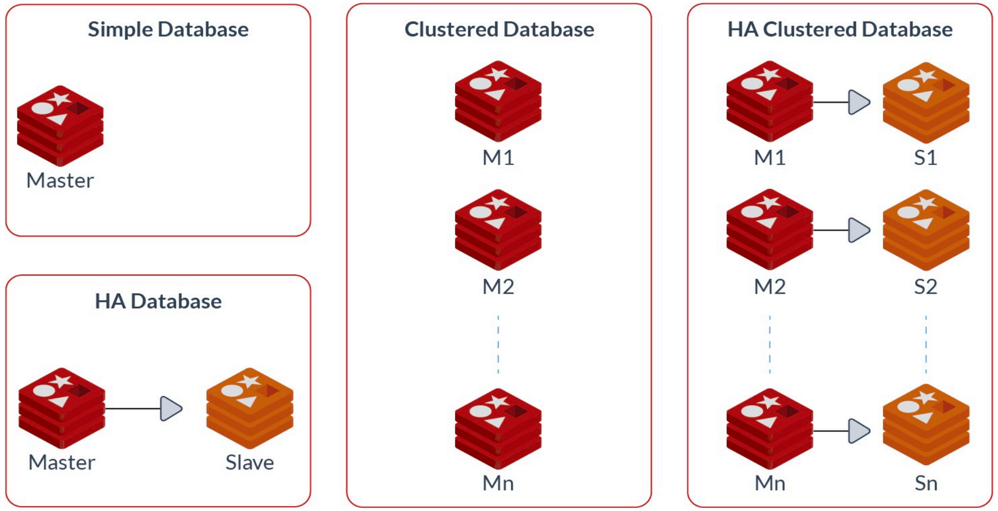
Source: https://cloud.google.com/docs, https://cloud.google.com/products

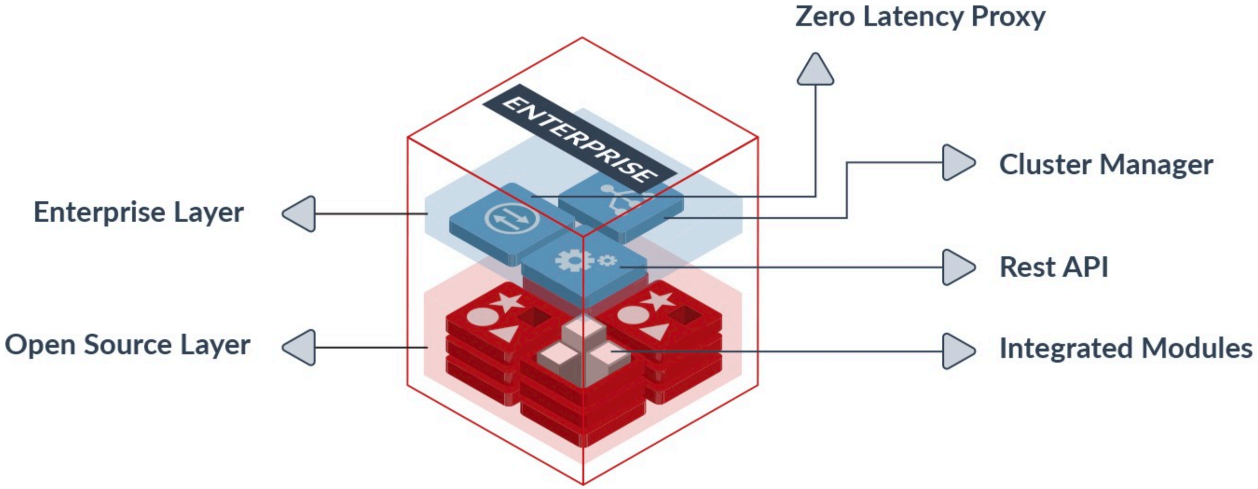
Databases: Memorystore

Reduce latency with scalable, secure, and highly available in-memory service for Redis and Memcached.

- Build application caches that provide sub-millisecond data access
- 100% compatible with open source Redis and Memcached
- Migrate your caching layer to cloud with zero code change

Redis Enterprise Cluster Architecture (Shared-nothing, linearly scalable, multi-tenant, symmetric architecture)

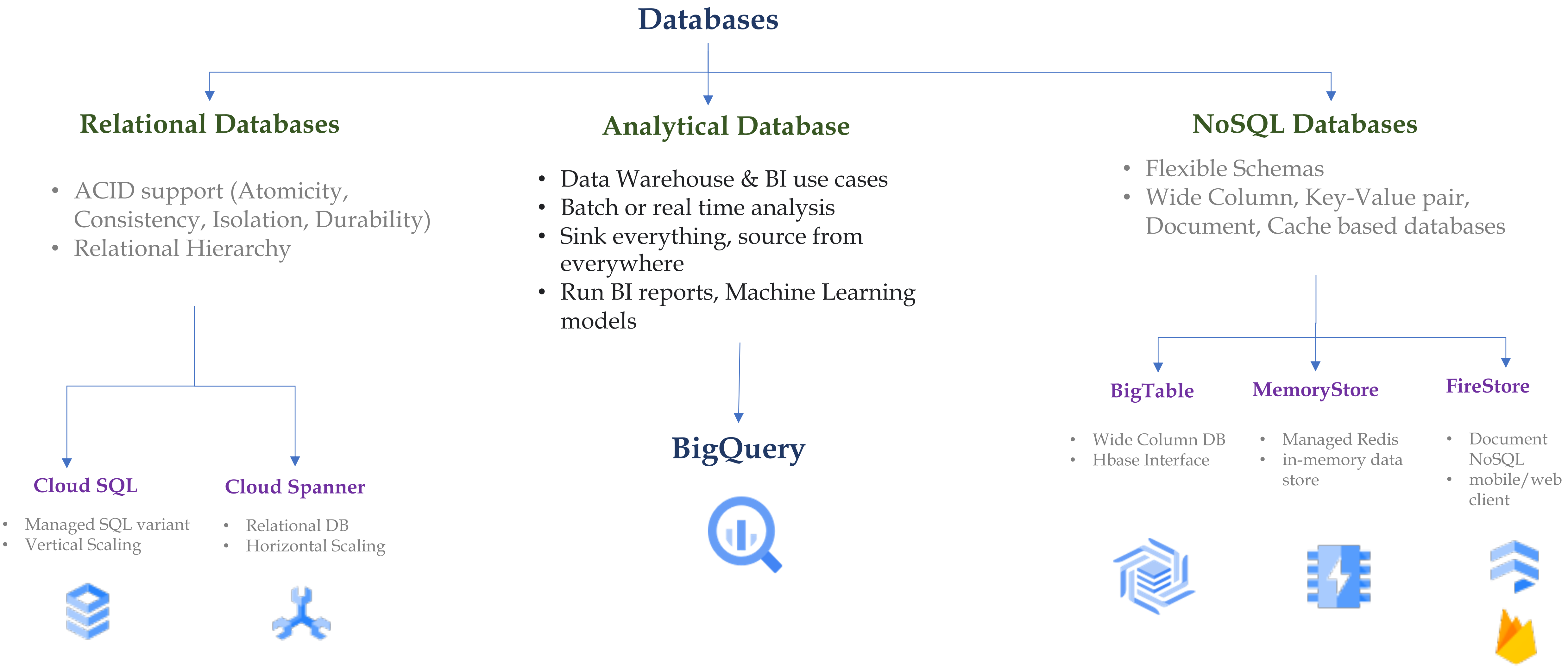




Memorystore for Redis supports Basic and Standard Tiers.

Source: https://cloud.google.com/docs, https://cloud.google.com/products

Databases: Analytical Database



Databases

Relational Databases

- ACID support (Atomicity, Consistency, Isolation, Durability)
- Relational Hierarchy

Cloud SQL

- Managed SQL variant
- Vertical Scaling

Cloud Spanner

- Relational DB
- Horizontal Scaling

Analytical Database

- Data Warehouse & BI use cases
- Batch or real time analysis
- Sink everything, source from everywhere
- Run BI reports, Machine Learning models

BigQuery

NoSQL Databases

- Flexible Schemas
- Wide Column, Key-Value pair, Document, Cache based databases

BigTable

- Wide Column DB
- Hbase Interface

MemoryStore

- Managed Redis
- in-memory data store

FireStore

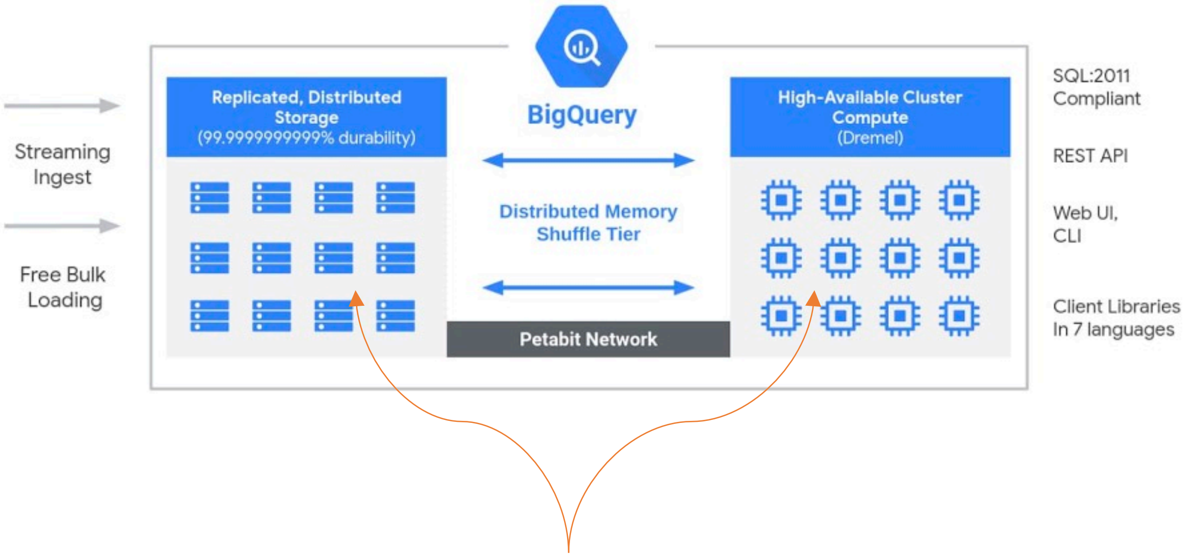
- Document NoSQL
- mobile/web client

Source: https://cloud.google.com/docs, https://cloud.google.com/products

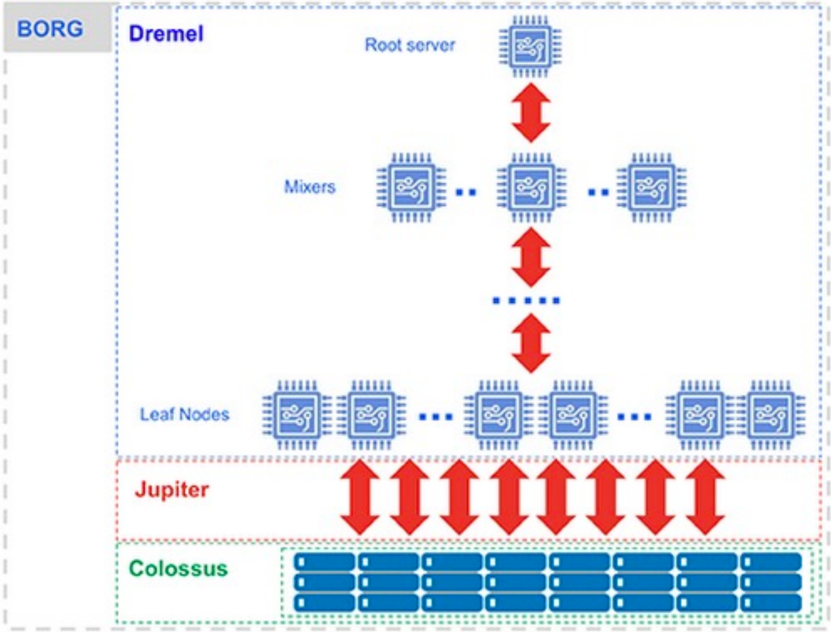
BigQuery: Architecture

BigQuery is Google Cloud's fully managed, petabyte-scale, and cost-effective analytics data warehouse that helps you manage and analyze data with built-in features like machine learning, geospatial analysis, and business intelligence.

BigQuery Architecture:


Streaming Ingest
Free Bulk Loading
Replicated, Distributed Storage (99.9999999999% durability)
BigQuery
Distributed Memory Shuffle Tier
Petabit Network
High-Available Cluster Compute (Dremel)
SQL:2011 Compliant
REST API
Web UI, CLI
Client Libraries In 7 languages

BigQuery's serverless architecture decouples storage and compute and allows them to scale independently on demand.



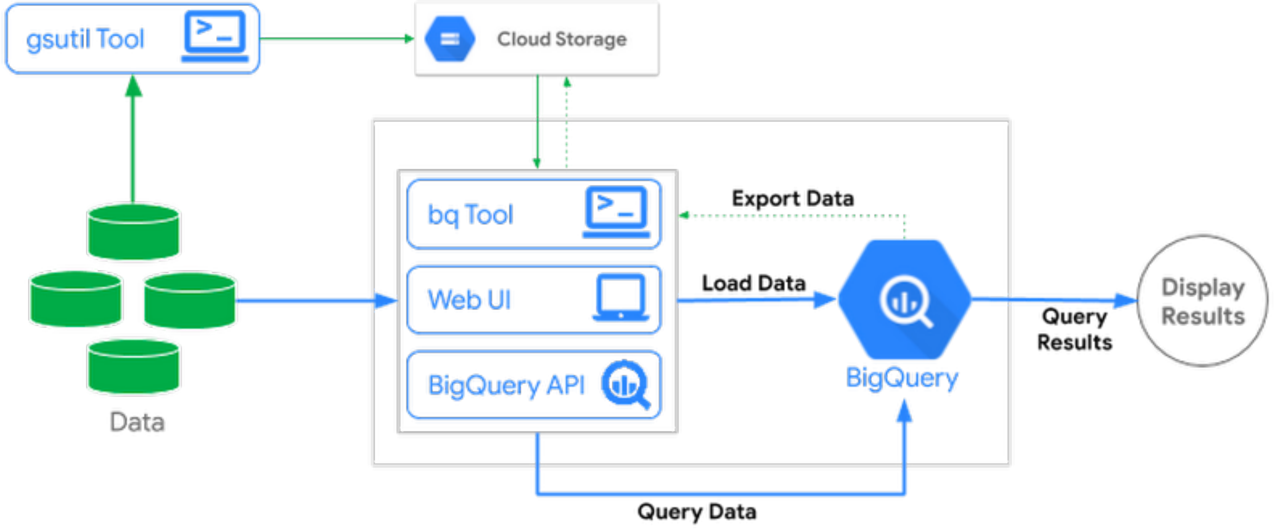
Under the hood, BigQuery employs a vast set of multi-tenant services driven by low-level Google infrastructure technologies like Dremel, Colossus, Jupiter and Borg.

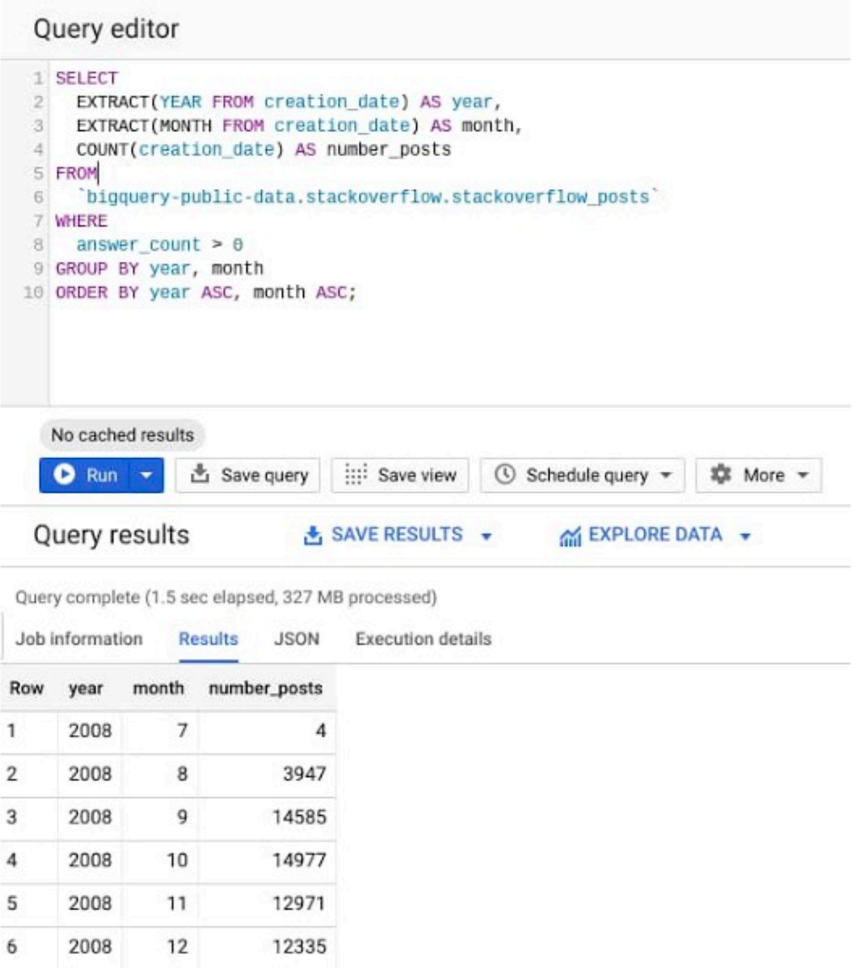
Source: https://cloud.google.com/docs, https://cloud.google.com/products

BigQuery: using BigQuery

BigQuery can be accessed in multiple ways:

- Using the GCP console
- Using the command line tool bq
- Making calls to the BigQuery REST API
- Using the variety of client libraries such as Java, .NET or Python





Query editor

```
SELECT
  EXTRACT(YEAR FROM creation_date) AS year,
  EXTRACT(MONTH FROM creation_date) AS month,
  COUNT(creation_date) AS number_posts
FROM
  `bigquery-public-data.stackoverflow.stackoverflow_posts`
WHERE
  answer_count > 0
GROUP BY year, month
ORDER BY year ASC, month ASC;
```

No cached results

Query results

Query complete (1.5 sec elapsed, 327 MB processed)

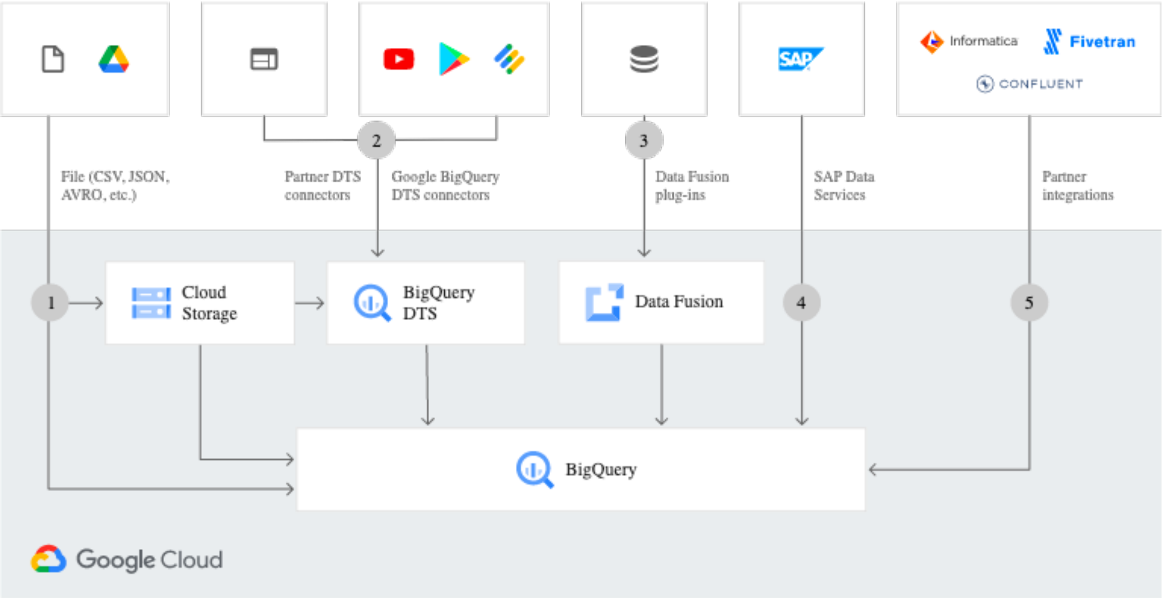
Row	year	month	number_posts
1	2008	7	4
2	2008	8	3947
3	2008	9	14585
4	2008	10	14977
5	2008	11	12971
6	2008	12	12335

Source: https://cloud.google.com/docs, https://cloud.google.com/products

BigQuery: Features

Data Ingestion flexibility:

Upload data files from local sources, Google Drive or Cloud Storage buckets, take advantage of BigQuery Data Transfer Service (DTS), Data Fusion plug-ins, or leverage Google's industry-leading data integration partnerships.

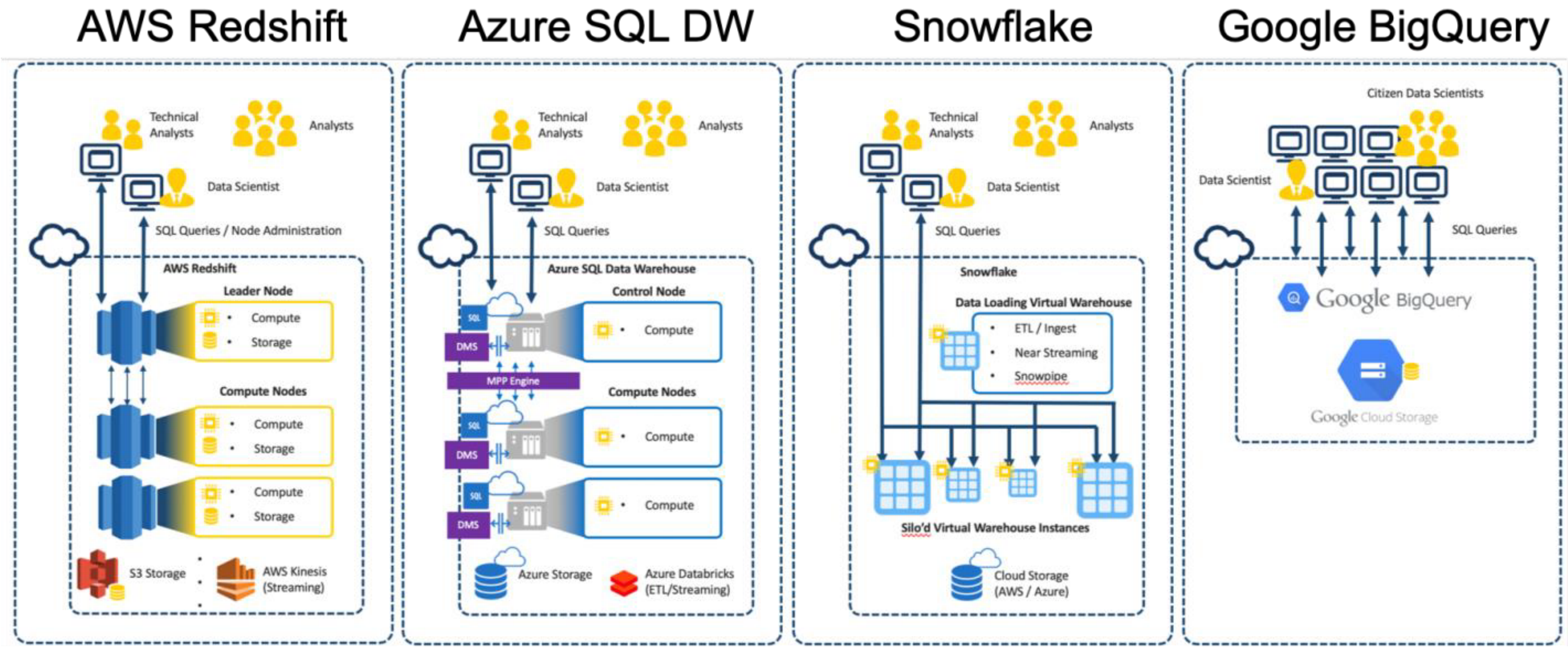


Other Features:

- Multicloud capabilities – BigQuery Omni(Preview)
- Built-in ML and AI integrations
- BI Engine
- Connected Sheets
- Geospatial data types and functions – BigQuery GIS
- Federated query and logical data warehousing
- Public Datasets

BigQuery: Cloud EDW’s

Functional Comparison of Cloud-based EDW Solutions:



- **Elimination of upfront investment and planning**
- **Reduction in operational expenses**

Source: Enterprise Strategy Group

Source: https://cloud.google.com/docs, https://cloud.google.com/products