

A Tale of Two Stores

DimeStore
re



VS

TakeYourNickelBack
.com



DimeStor

The official unofficial Nickelback Fan-Store



e

NICKELBACK

FEED THE MACHINE TOUR



The Beginning of DimeStore

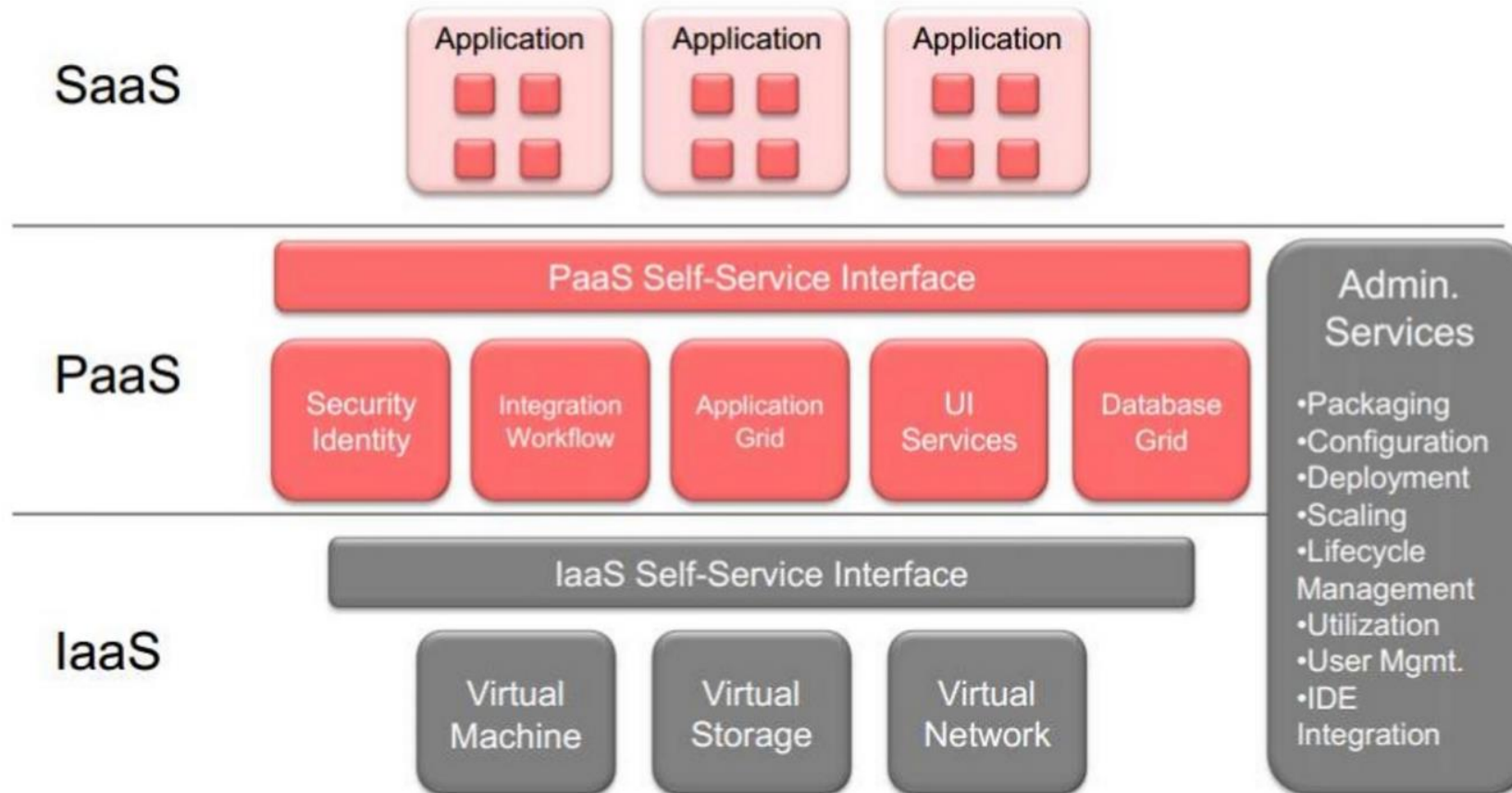




Chad Kroeger



Cloud Computing



Static Web-page with e-mail order form



1. Sign in to the AWS Management Console and open the **Amazon S3** console at <https://console.aws.amazon.com/s3/>.
2. Create two buckets that match your domain name and subdomain (*dimestore.com* and *www.dimestore.com*).

Use the Amazon S3 console to configure the bucket for website hosting

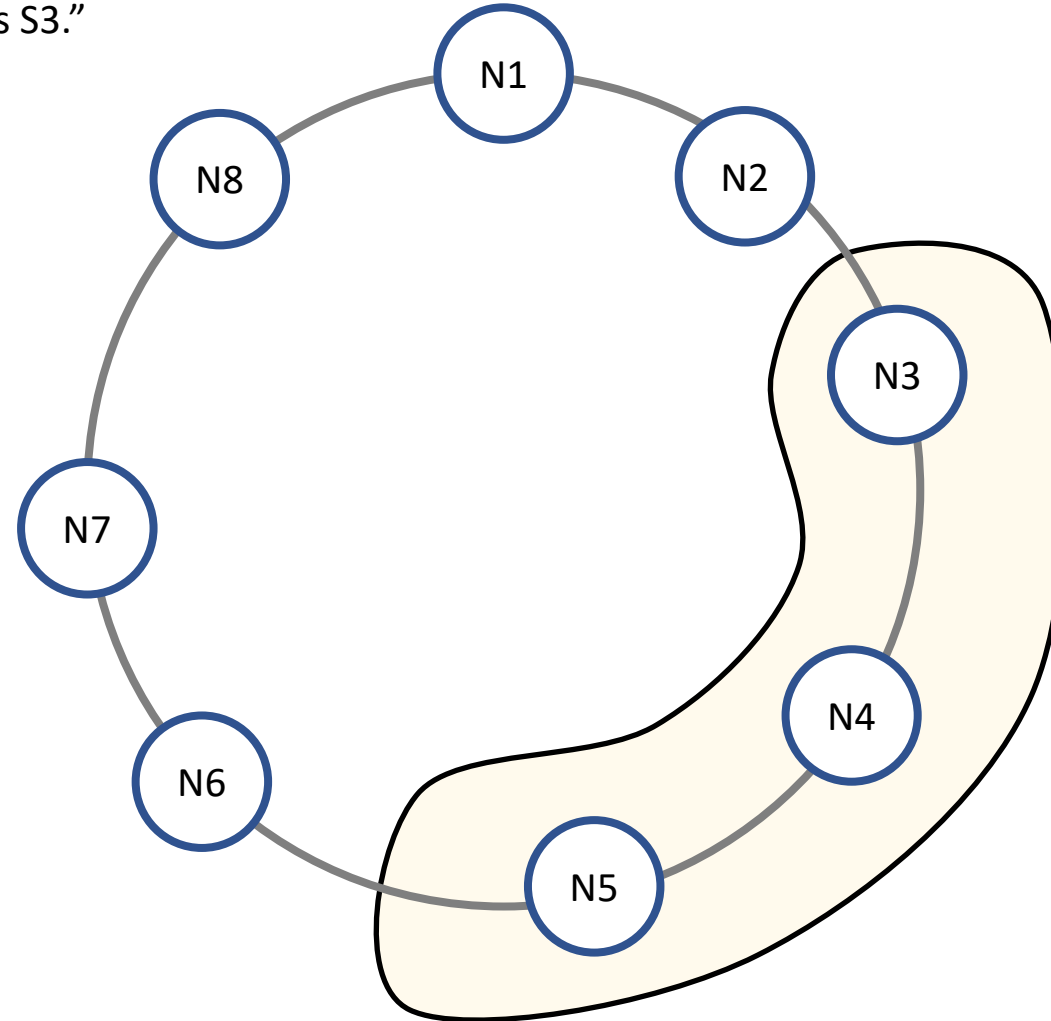
1. In the S3 buckets list, choose the bucket with the same name as your domain. Then Choose Properties -> Static website hosting.-> Use this bucket to host a website & Index Document box, enter the name of your index page (index.html)
2. Redirect requests from *www.dimestore.com* to *dimestore.com*: Choose Properties -> Choose Static website hosting -> Choose Redirect requests. In the Target bucket or domain box, enter your domain (for example, *example.com*).
3. Upload Index and Website Content
4. Edit Block Public Access Settings: choose Edit public access settings. Clear Block all public access, and choose Save.
5. Get Your Endpoints and Test Your Domain Endpoint
6. Add Alias Records for *dimestore.com* and *www.dimestore.com*

Amazon S3

Werner Vogel: “Let me emphasize the internal technology part before it gets misunderstood: Dynamo is not **directly** exposed externally as a web service; however, Dynamo and similar Amazon technologies **are** used to power parts of our Amazon Web Services, such as S3.”



Eventual Consistency



**Take Your Nickel Back
k.com**



TakeYourNickelBack.com

- After one too many arguments about how all the music that everyone else likes is terrible, and Sam finds that he is the only one with impeccable musical taste. Sam decides to start his own Anti-Nickelback website for making fun of these people.
- Sam distrusts all large businesses (except hardware manufacturers for some reason) and decides to host the website on a web server running in his basement



NGINX

DimeStor

The official unofficial Nickelback Fan-Store



e

NICKELBACK

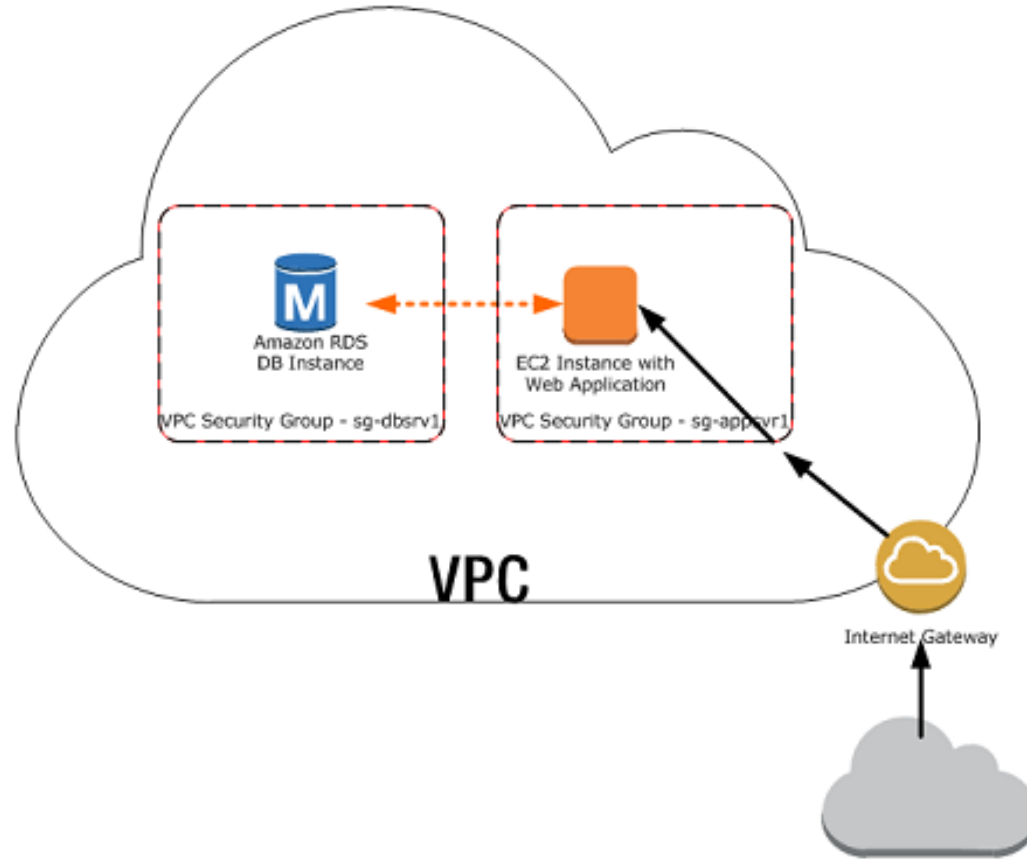
FEED THE MACHINE TOUR



DimeStore is getting traction



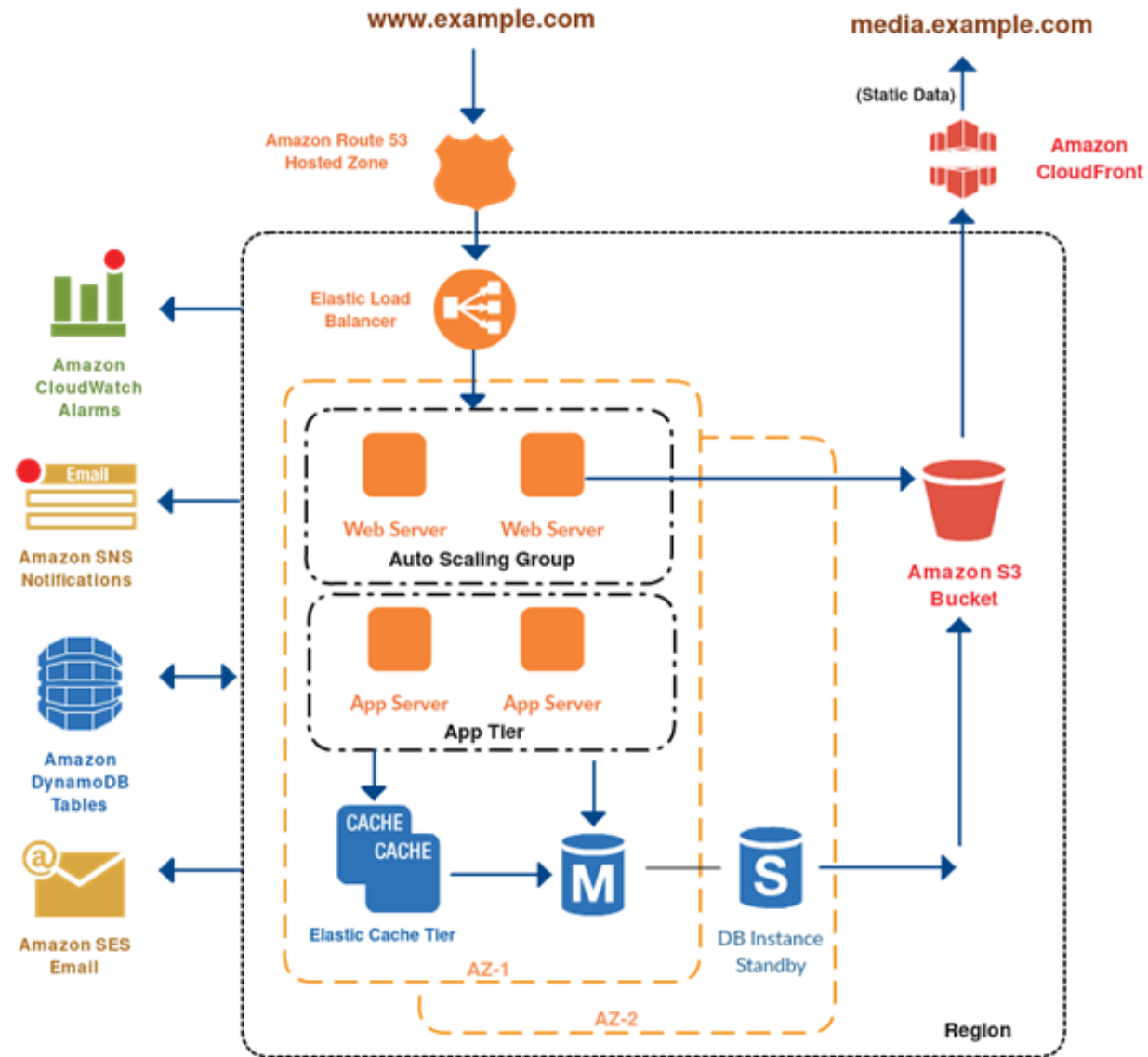
DimeStores first real web-store



Amazon RDS database engines



Typical scalable architecture on Amazon



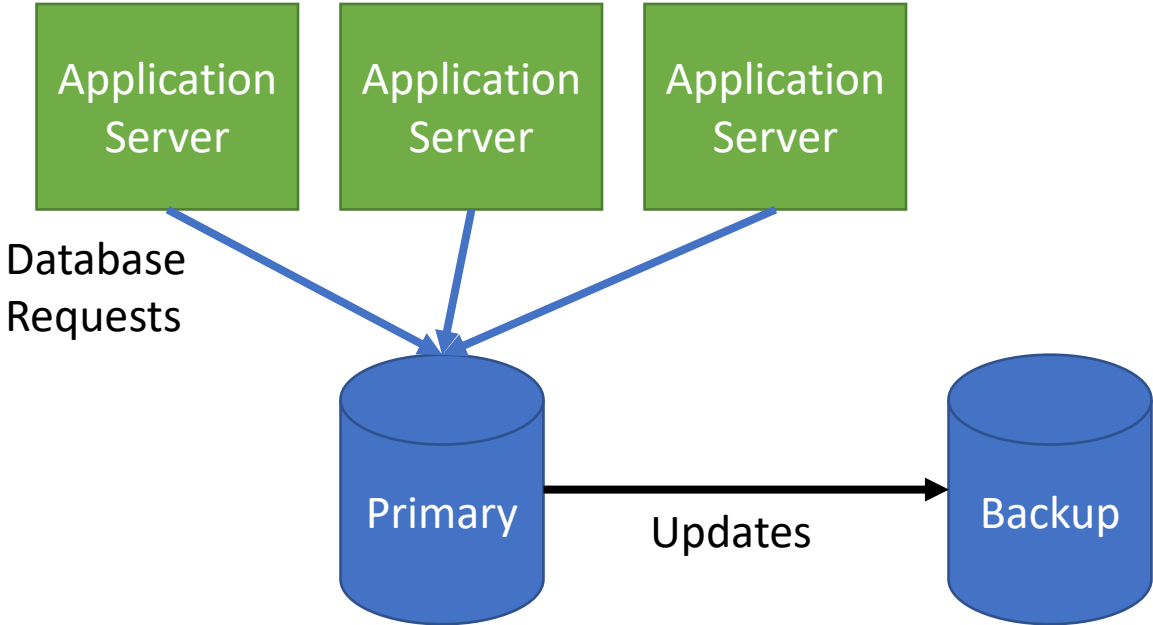
Why do we need a database and transactions?



What do you do when your database fails?



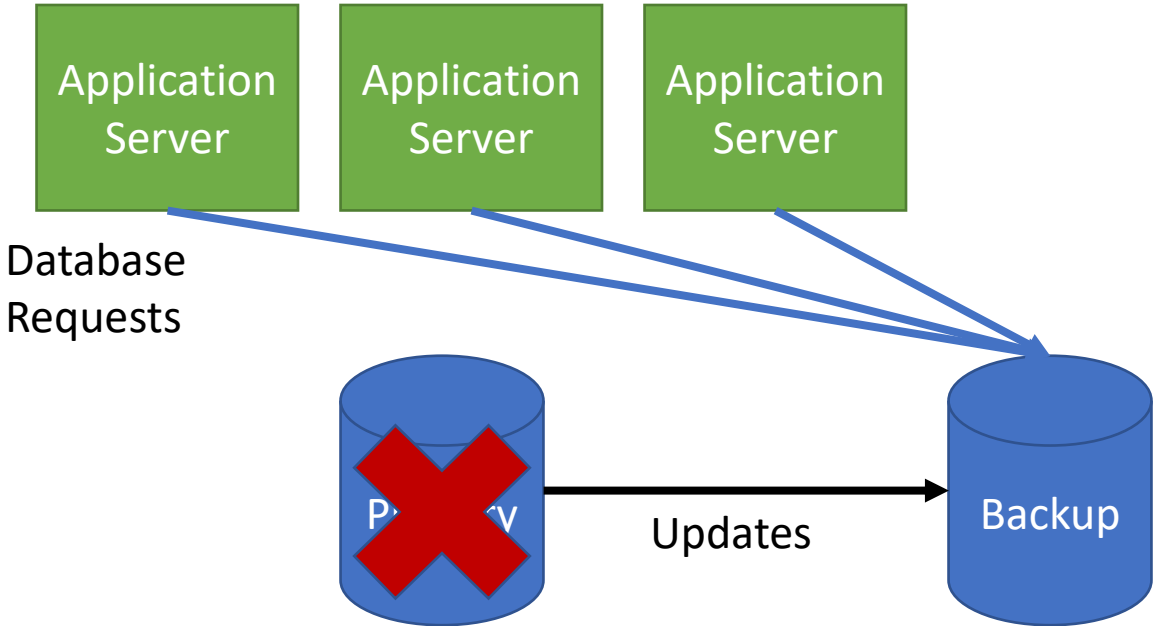
Primary Backup



What do you do when your database fails?



Primary Backup

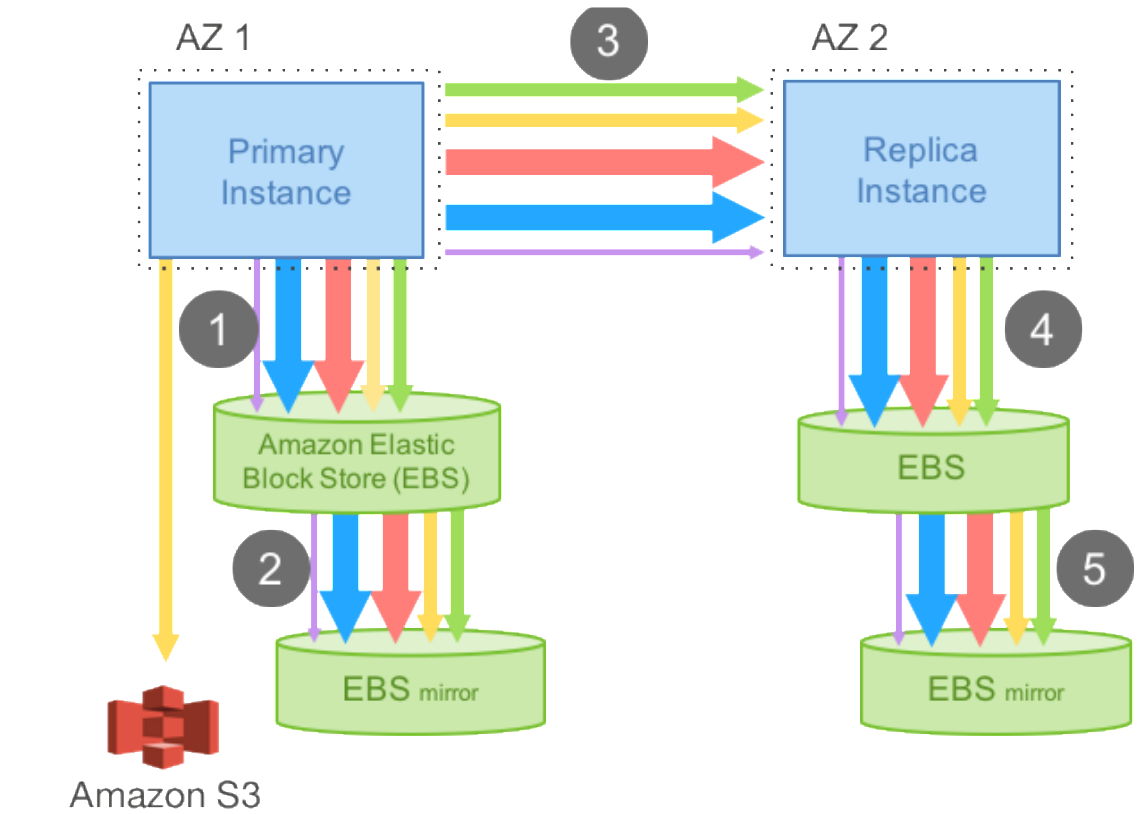




Amazon
RDS



Amazon RDS - Traditional Replication



Steps 1, 3, and 5 are sequential and synchronous.

TYPE OF WRITE



Vertical Scaling

Modify DB Instance: Im1ks7xpixmry6w

Instance Specifications

DB Engine Version

DB Instance Class

Multi-AZ Deployment

Storage Type

Allocated Storage*

Settings

DB Instance Identifier

New Master Password

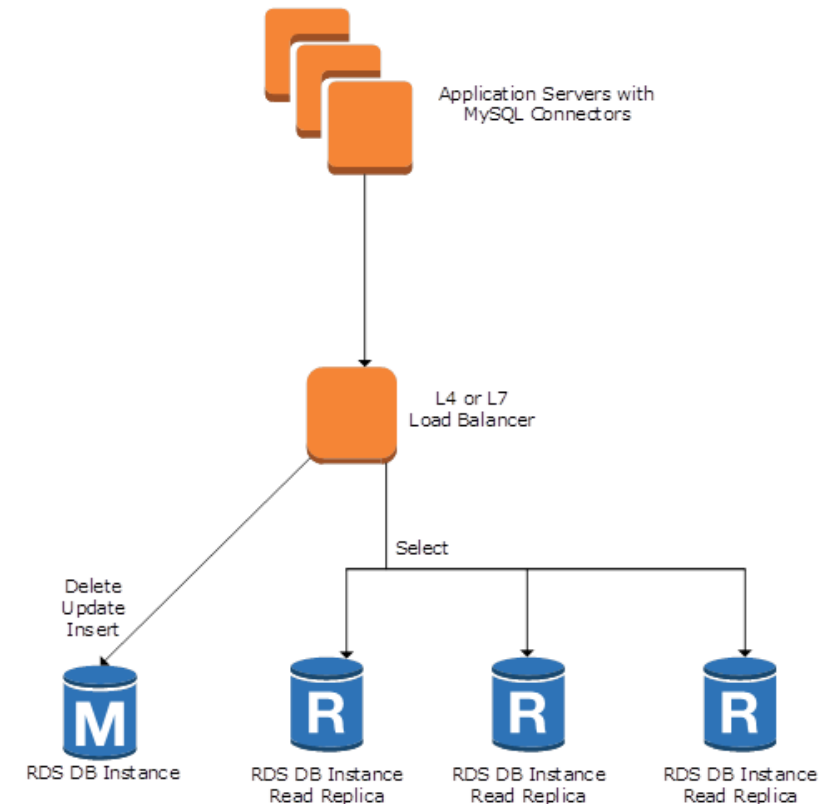
Network & Security

Security Group

- db.t2.micro — 1 vCPU, 1 GiB RAM
- db.t2.small — 1 vCPU, 2 GiB RAM
- db.t2.medium — 2 vCPU, 4 GiB RAM
- db.t2.large — 2 vCPU, 8 GiB RAM
- db.m4.large — 2 vCPU, 8 GiB RAM
- db.m4.xlarge — 4 vCPU, 16 GiB RAM
- db.m4.2xlarge — 8 vCPU, 32 GiB RAM
- db.m4.4xlarge — 16 vCPU, 64 GiB RAM
- db.m4.10xlarge — 40 vCPU, 160 GiB RAM
- db.m3.medium — 1 vCPU, 3.75 GiB RAM
- db.m3.large — 2 vCPU, 7.5 GiB RAM
- db.m3.xlarge — 4 vCPU, 15 GiB RAM
- db.m3.2xlarge — 8 vCPU, 30 GiB RAM**
- db.r3.large — 2 vCPU, 15 GiB RAM
- db.r3.xlarge — 4 vCPU, 30.5 GiB RAM
- db.r3.2xlarge — 8 vCPU, 61 GiB RAM
- db.r3.4xlarge — 16 vCPU, 122 GiB RAM
- db.r3.8xlarge — 32 vCPU, 244 GiB RAM
- db.m2.xlarge — 2 vCPU, 17.1 GiB RAM
- db.m2.2xlarge — 4 vCPU, 34 GiB RAM

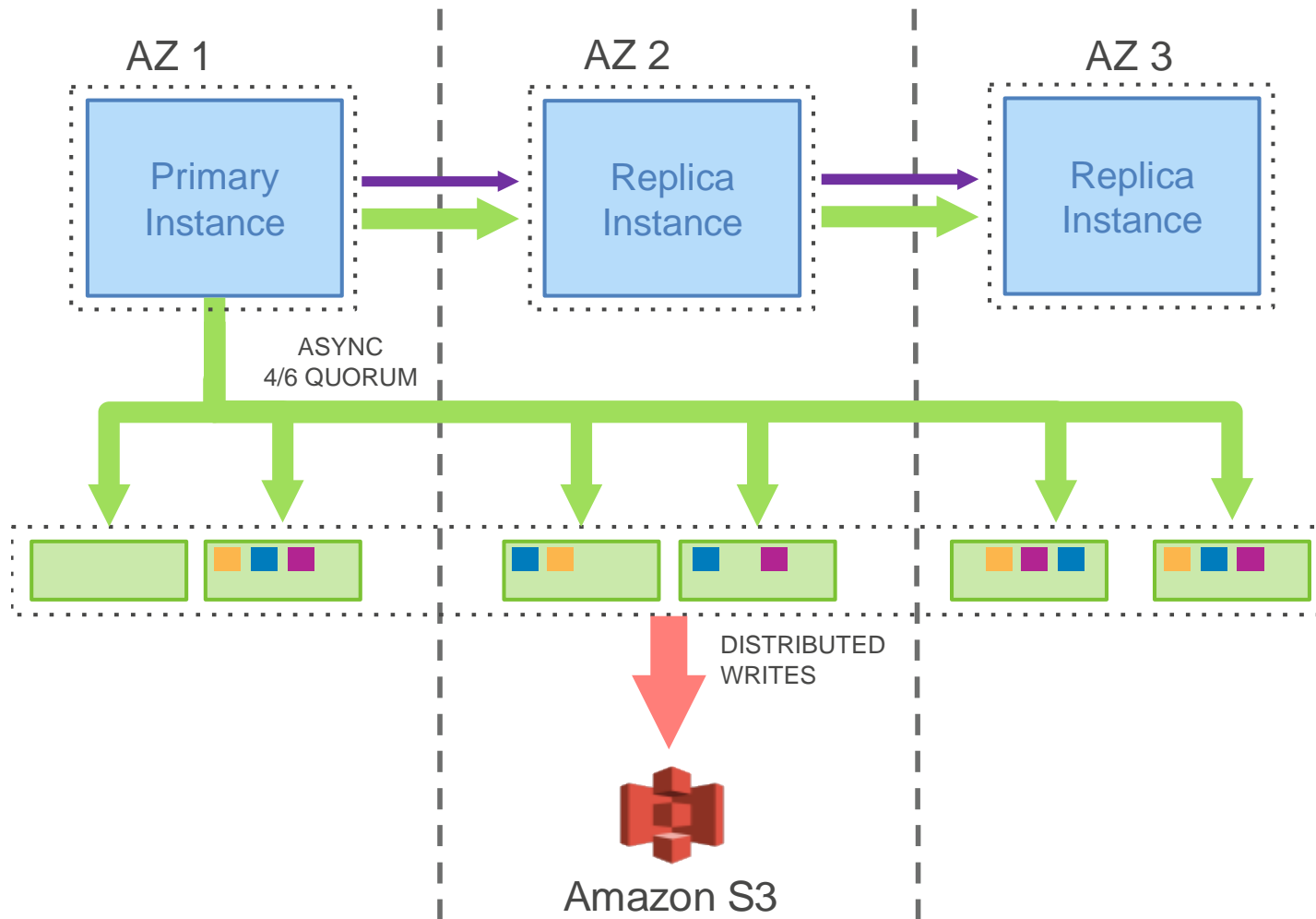
Creates potential downtime

Horizontal Scaling



Consistent reads

Amazon Aurora



Question:
Why does Amazon do 4/6 replication?

The log stream generated by the writer and sent to the storage nodes is also sent to all read replicas. Each replica typically lags behind the writer by a short interval (20 ms or less)

Based on MySQL. Changed InnoDB version

**Take Your Nickel Back
k.com**



Everyone thinks they have impeccable musical taste and emails wanting to buy anti-nickelback merch.



TakeYourNickelBack.com



- Has a forum for people complaining about Nickelback
- Has a store for great Anti-Nickelback merch
- Allows users to design their own merch and the top voted designs will be sold on the website.
- Has a blog where I talk about how much better my music tastes are than my friends. Allows users to like and comment.
- Maybe more in the future.

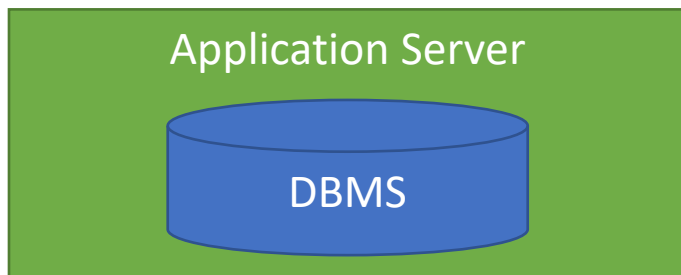
- Maybe I could build on top of an existing embedded database key value store like RocksDB or SQLite

Choice of DBMS



Embedded DBMS

- If you application server fails, you lose data
- Do not easily scale to application workload
- Usually runs in the same process



SQLite



- An embedded SQL database that supports transactions. A row-store optimized for transactions.
- Disadvantages: Single user and not built for not built for large applications.

SQLite



- An embedded SQL database that supports transactions. A row-store optimized for transactions.
- Disadvantages: ?

RocksDB



- Log Structure Merge Tree (previous lecture by Sam)
- You get a key-value database (like a hash table) that stores data on disk, supports transactions and is optimized for write-heavy workloads, and can support range scans.
- Is used as a storage engine for other systems (MySQL)
- Disadvantages: ?

RocksDB



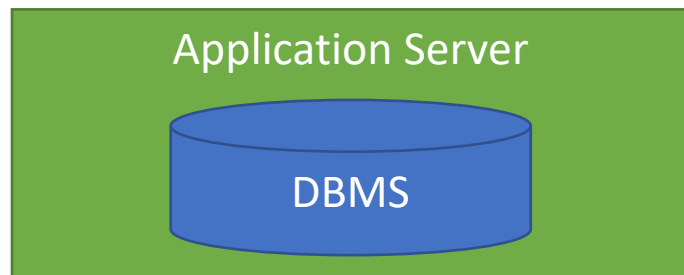
- Log Structure Merge Tree (previous lecture by Sam)
- You get a key-value database (like a hash table) that stores data on disk, supports transactions and is optimized for write-heavy workloads, and can support range scans.
- Is used as a storage engine for other systems (MySQL)
- Disadvantages:
 1. Lacks a full query language, and many common database features. If you need foreign keys, columns, indexes etc you have to build it yourself.
 2. Single User
 3. Doesn't scale beyond a single process (what do we do when our website grows to more users?)

Choice of DBMS



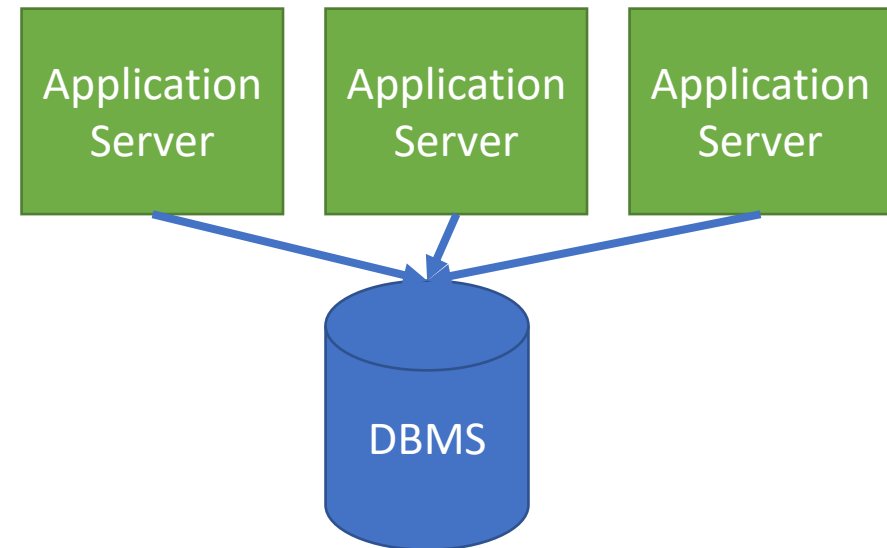
Embedded DBMS

- If you application server fails, you lose data
- Do not easily scale to application workload
- Usually runs in the same process



Independent DBMS

- Scale Application and DB independently
- Designed for many concurrent users



What are my options for data management?



- A full featured transaction processing SQL system like Postgres, MySQL (MariaDB), Oracle or Microsoft SQL Server (but not open source). Optimized for many concurrent users and offers failover features.
- Used by the largest companies for their most important operational data. If tuned well, you can trust them to handle your data properly.
- Since we are running these ourselves, we need to hire a database administrator to ensure things are running smoothly and safely.



What does it mean to host your own database on-premise (or on EC2)



- Increasing reliability requires buying new machines
- Scale-out/Scale-up requires buying new machines
- They have to be configured correctly
 - Independent drives
 - Backups
 - ...
- Disaster recovery needs to be automated (how to recover from power-outage)
- Load-balancing etc. needs to be implemented
- Hire administrators
-

What about NoSQL?



mongoDB



- Stores JSON Documents in key-value pairs
- JSON only API
- Distributed across multiple nodes
- Has indexes for fast access
- Tend to store data in a denormalized way (instead of customer, orders table, each customer document stores all their orders)
- Started with no multi-document transactions, no joins. Have since added this functionality
- Have to give up SQL functionality, query optimization and other common DB features.
- ~~Is Web Scale~~

MongoDB for building a BitCoin exchange?



How Do You Transfer BitCoins with MongoDB?

“Flexcoin was a Bitcoin exchange that shut down on March 3rd, 2014, when someone allegedly hacked in and made off with 896 BTC in the hot wallet. Because the **half-million dollar** heist from the hot wallet was too large for the company to bear, it folded.”

“The attacker successfully exploited a flaw in the code which allows transfers between flexcoin users. **By sending thousands of simultaneous requests, the attacker was able to "move" coins from one user account to another until the sending account was overdrawn, before balances were updated.** “

```
mybalance = database.read("account-number")  
newbalance = mybalance - amount  
database.write("account-number", newbalance)  
dispense_cash(amount) // or send bitcoins to customer
```

See more at: <http://hackingdistributed.com/2014/04/06/another-one-bites-the-dust-flexcoin/#sthash.rWoYKi78.dpuf>

MongoDB Started To Support Secondary Indexes



No Recovery for Indexes???



Not **S**
Only **Q****Q****L**

The rapid growth of Nickelback fans causes a backlash, and my store is growing!

- I bought a ton of new machines and put them in my basement!
- My ISP is starting to wonder what I'm doing and my hardware costs are through the roof.



DimeStor

The official unofficial Nickelback Fan-Store

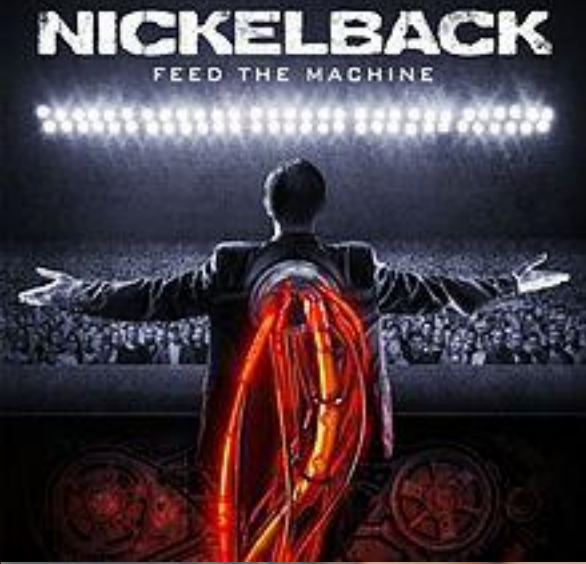


e

NICKELBACK

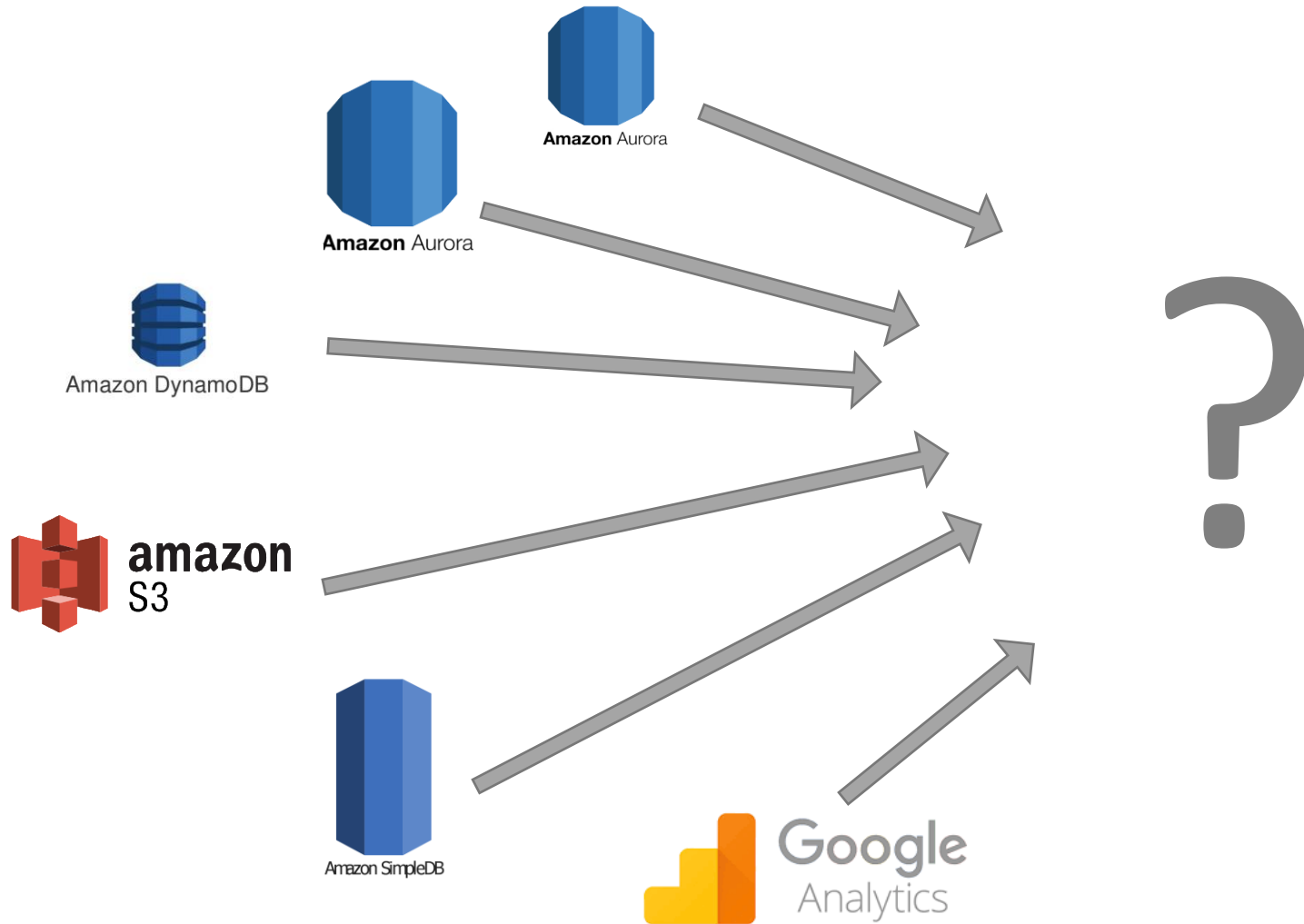
FEED THE MACHINE TOUR



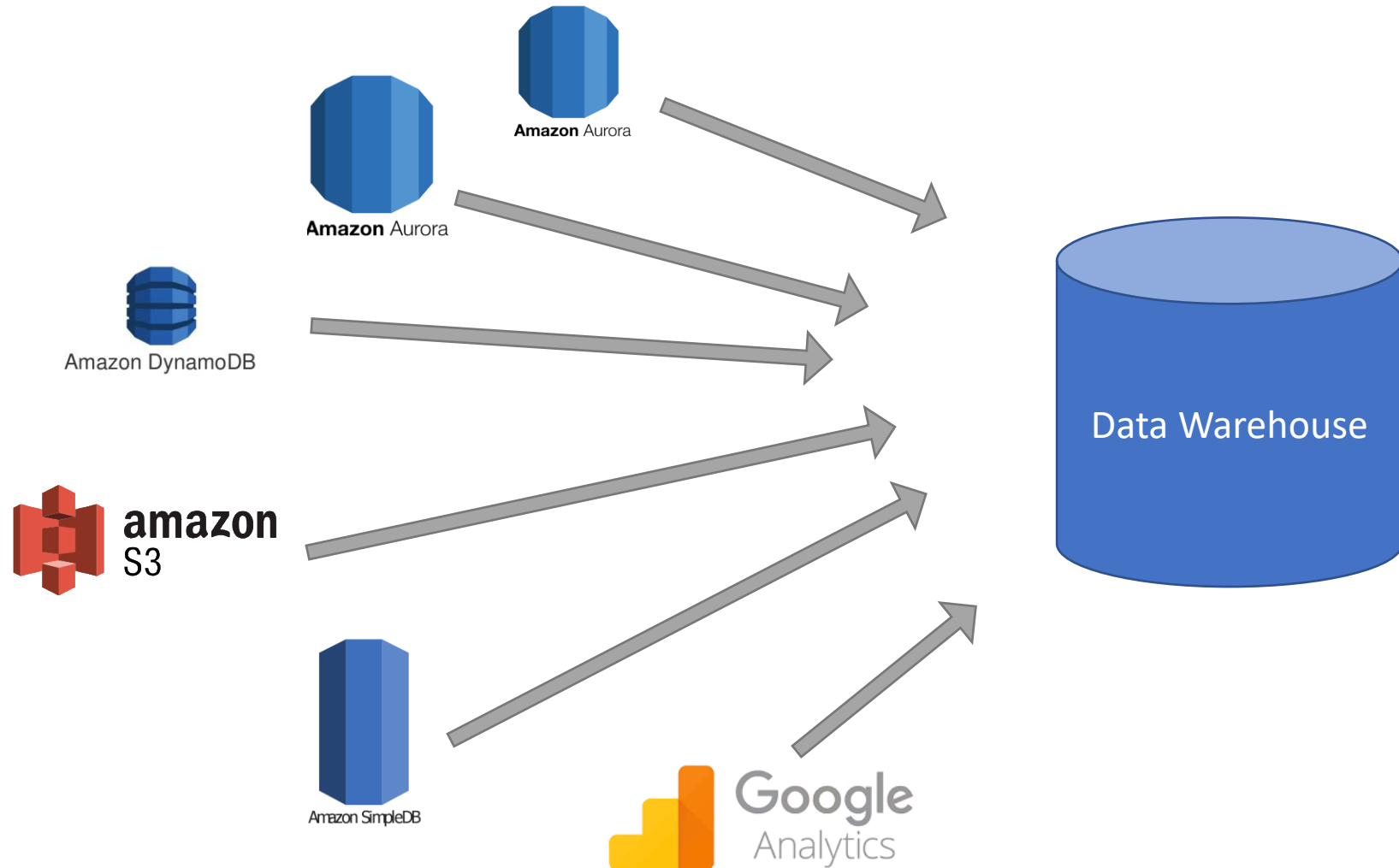


**Sales are dropping and we need to find new revenue.
Data Science to the rescue**

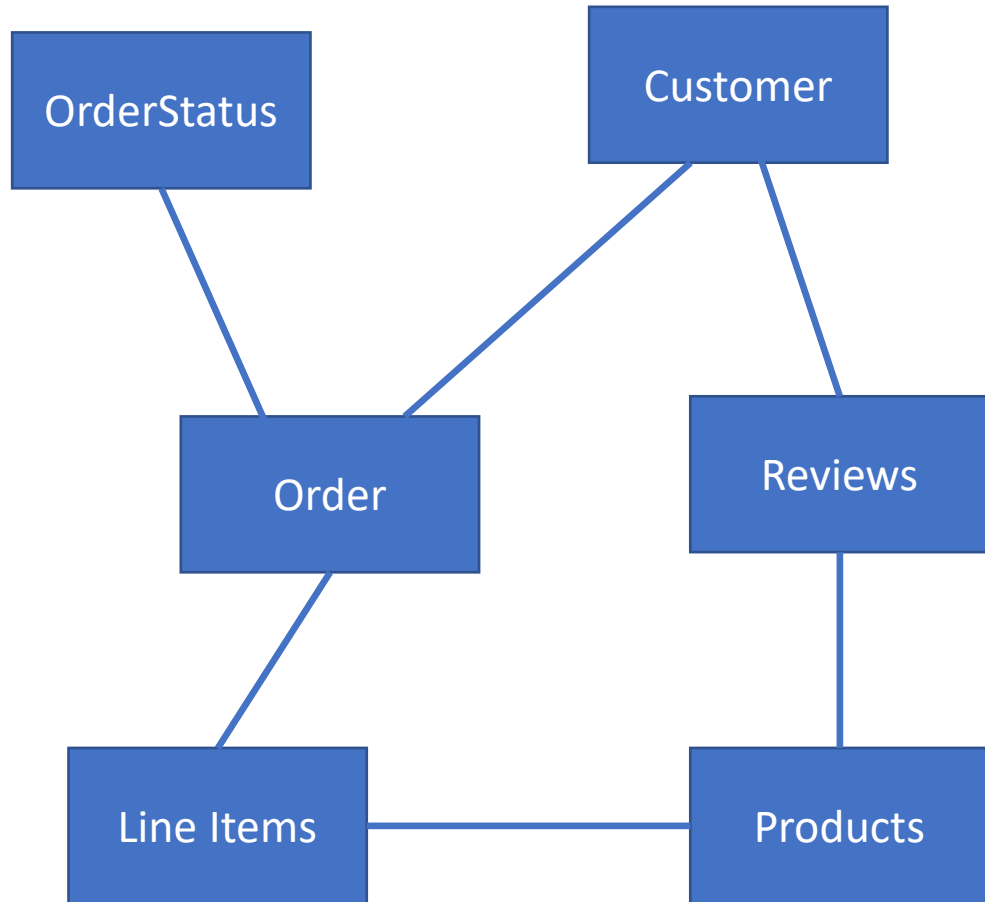
Data Science to the rescue



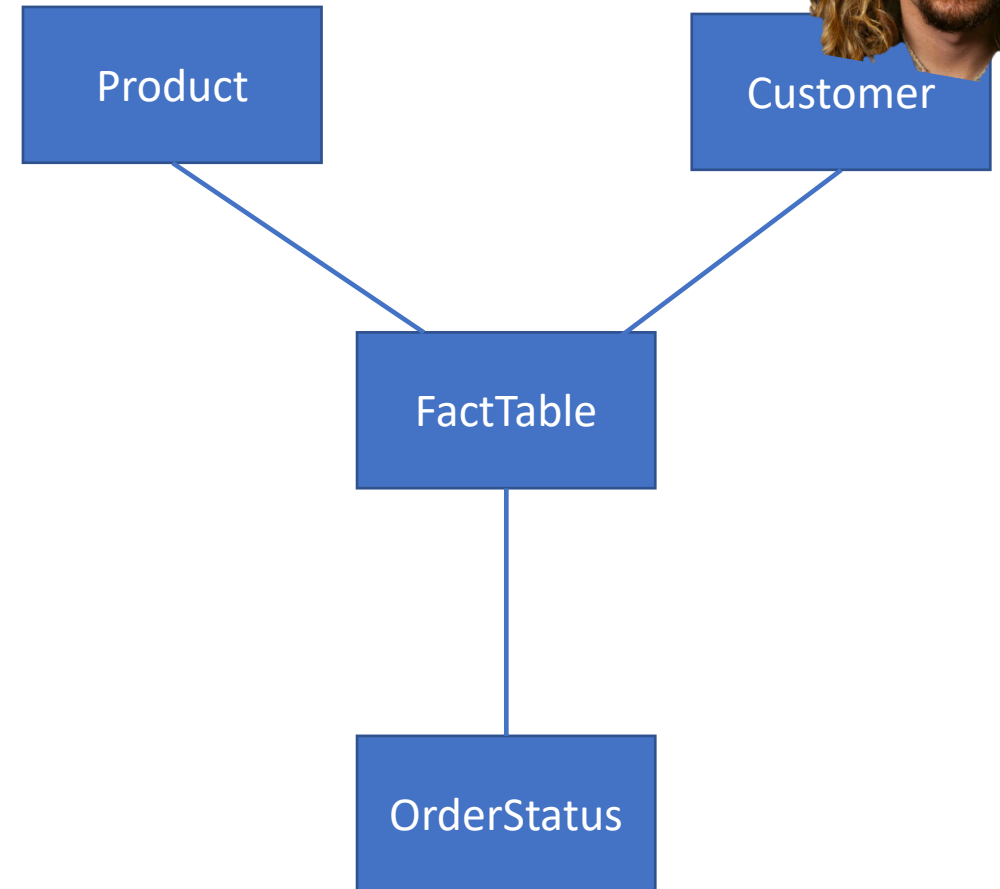
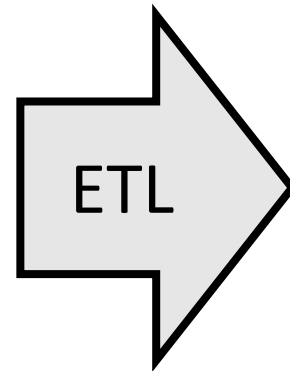
Data Science to the rescue



OLTP Schema



OLAP Schema



AWS Glue

Star Schema

Shop
- <u>Shop ID</u>
- Business_Type
- City
- City_Population
- State
- ...

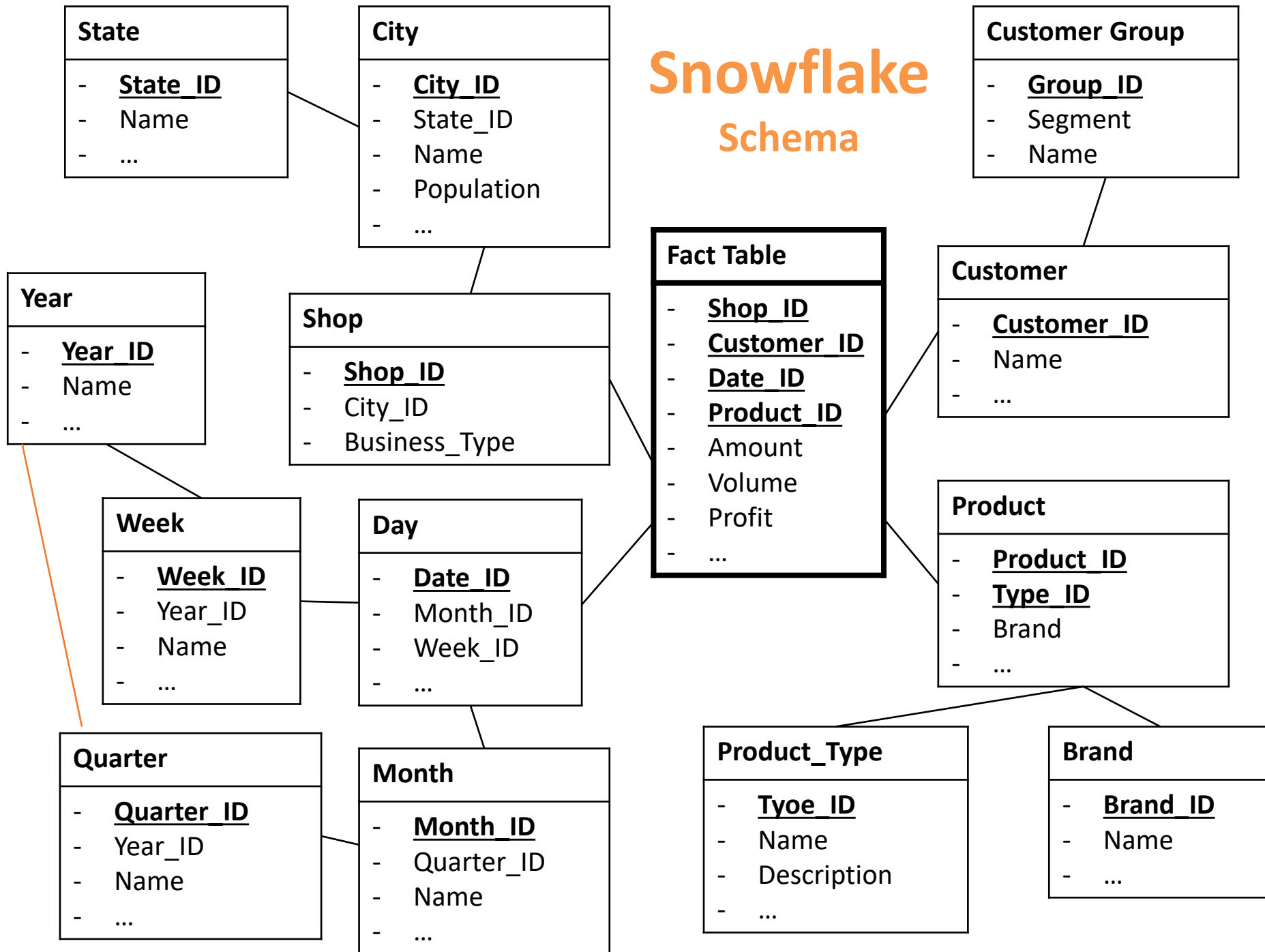
Customer
- <u>Customer ID</u>
- Name
- Segment
- Group_Name
- ...

Fact Table
- <u>Shop ID</u>
- <u>Customer ID</u>
- <u>Date ID</u>
- <u>Product ID</u>
- Amount
- Volume
- Profit
- ...

Time
- <u>Date ID</u>
- Month
- Quarter
- Year
- ...

Product
- <u>Product ID</u>
- <u>Type</u>
- Brand
- Description
- ...

Snowflake Schema



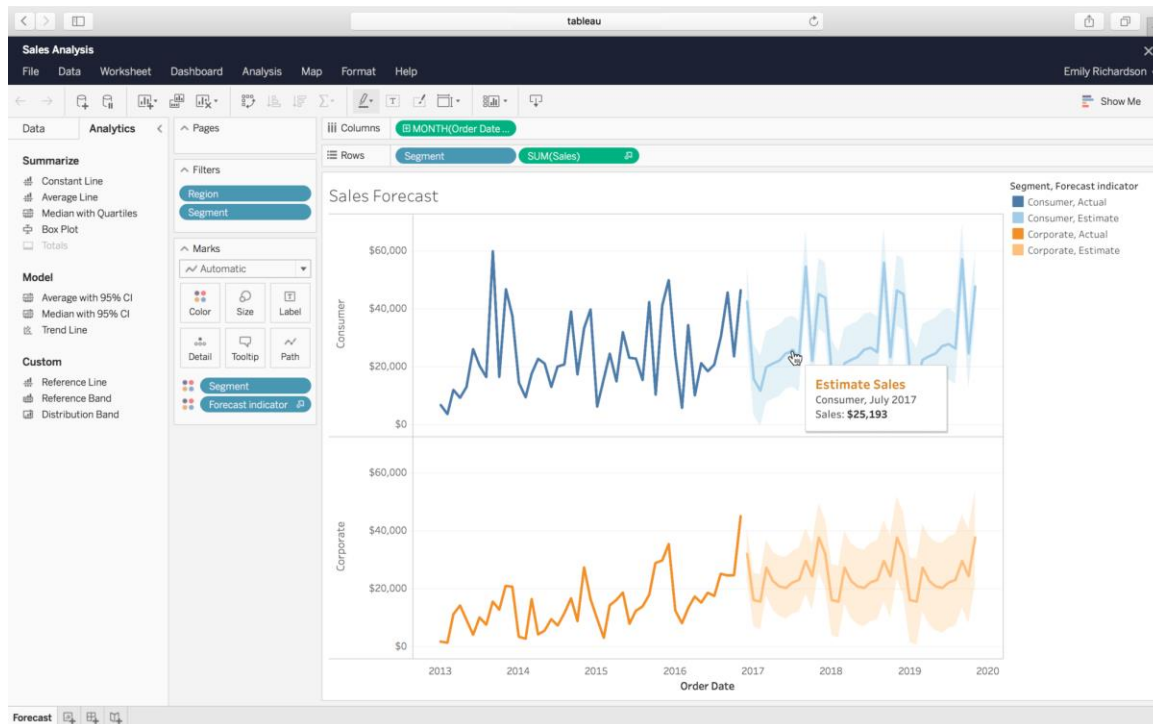
Star vs. Snowflake Schema

	Snowflake	Star
Space		
Query Performance		
Ease of Use		
When to use		
Normalization/ De-Normalization		

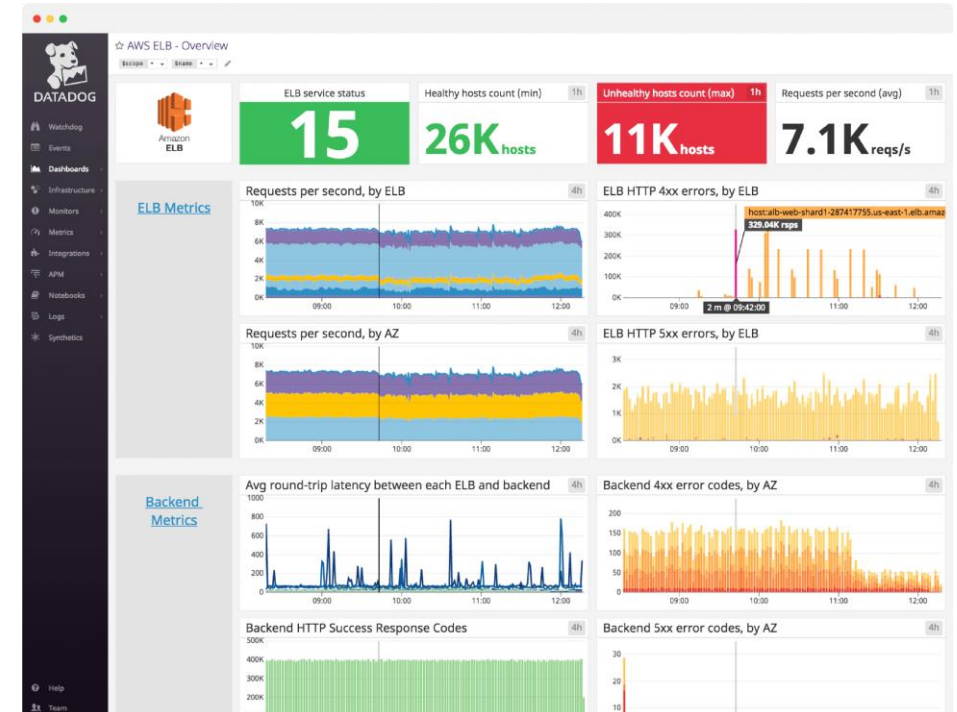
Star vs. Snowflake Schema

	Snowflake	Star
Space	Smaller	Bigger (Redundancy)
Query Performance	More Joins → slower	Fewer Joins → faster
Ease of Use	Complex Queries	Pretty Simply Queries
When to use	When dimension table is relatively big in size, snowflaking is better as it reduces space.	When dimension table contains less number of rows, we can go for Star schema.
Normalization/De-Normalization	Dimension Tables are in Normalized form but Fact Table is still in De-Normalized form	Both Dimension and Fact Tables are in De-Normalized form

What are the most common use cases for a datawarehouse?



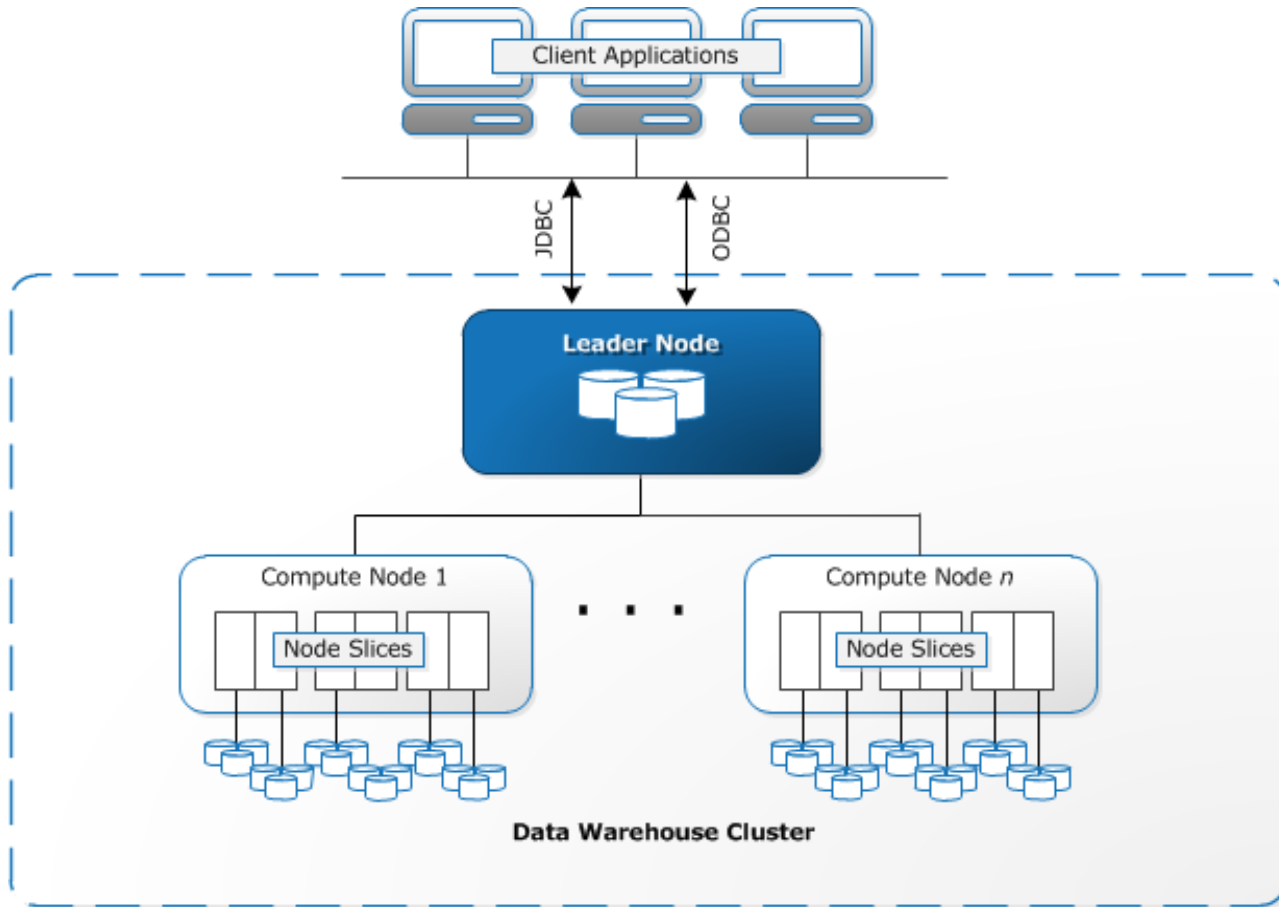
Data Exploration



Dashboards

(AWS QuickSight)

Amazon Redshift



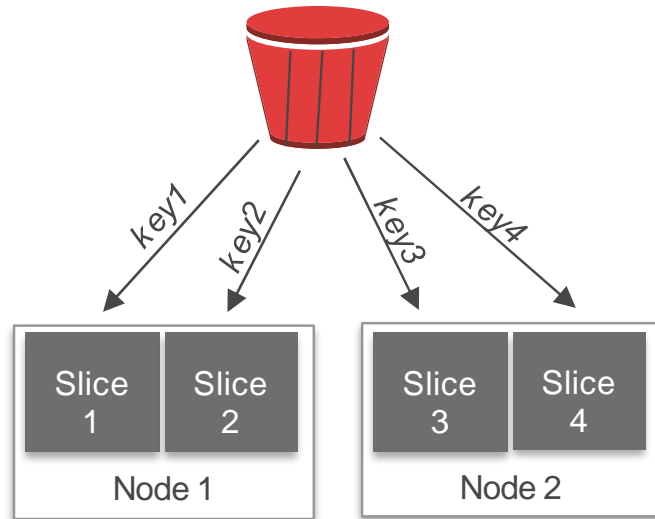
The leader node accepts connections from client programs, parses requests, generates & compiles query plans for execution on the compute nodes, performs final aggregation of results when required

Amazon Redshift



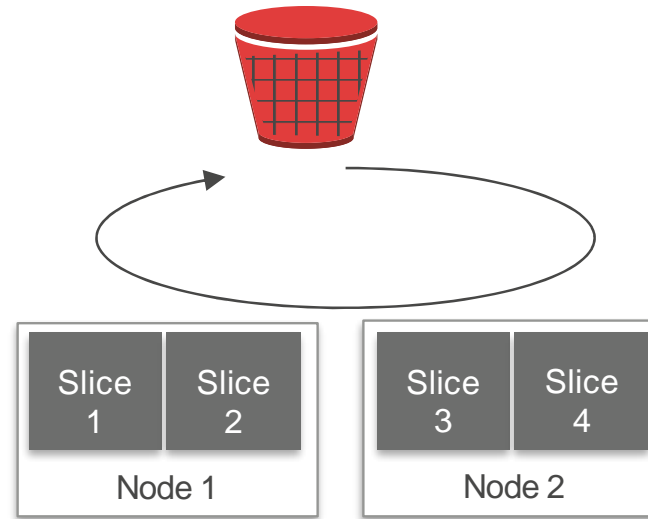
Key

Same key to same location



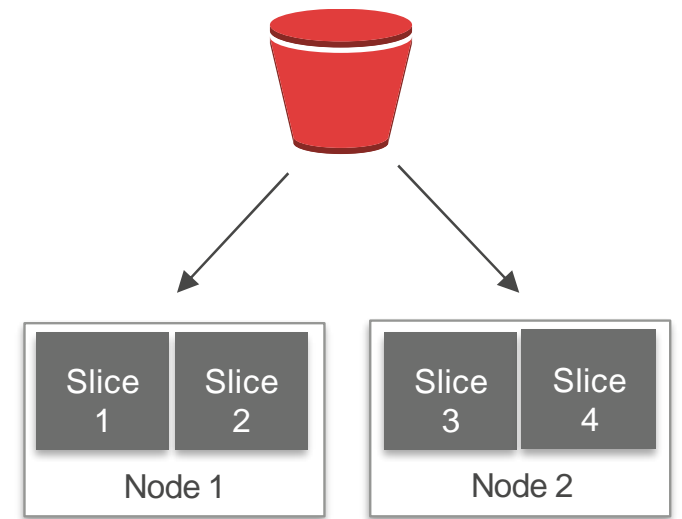
Even

Round robin distribution



All

All data on every node



The user can specify whether data is distributed in a round robin fashion, hashed according to a distribution key, or duplicated on all slices

Amazon Redshift Spectrum



Life of a query

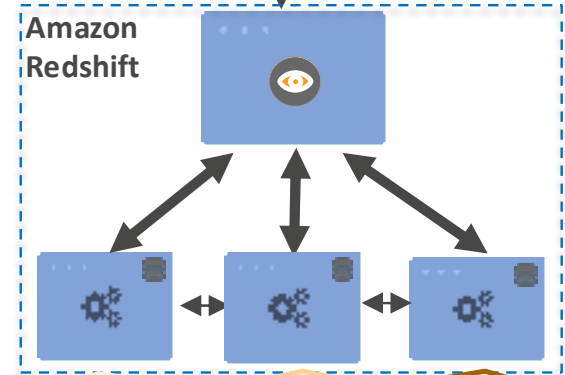


1

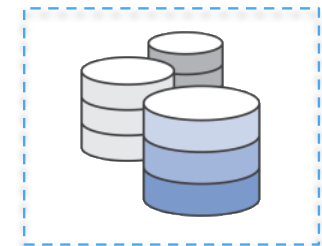
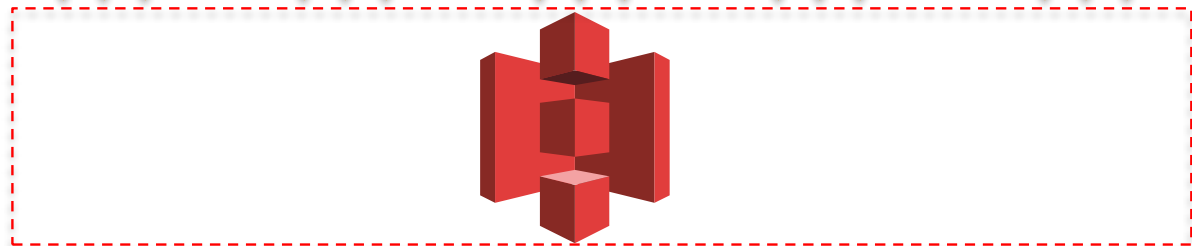
Query
`SELECT COUNT(*)
FROM S3.EXT_TABLE
GROUP BY...`



JDBC/ODBC

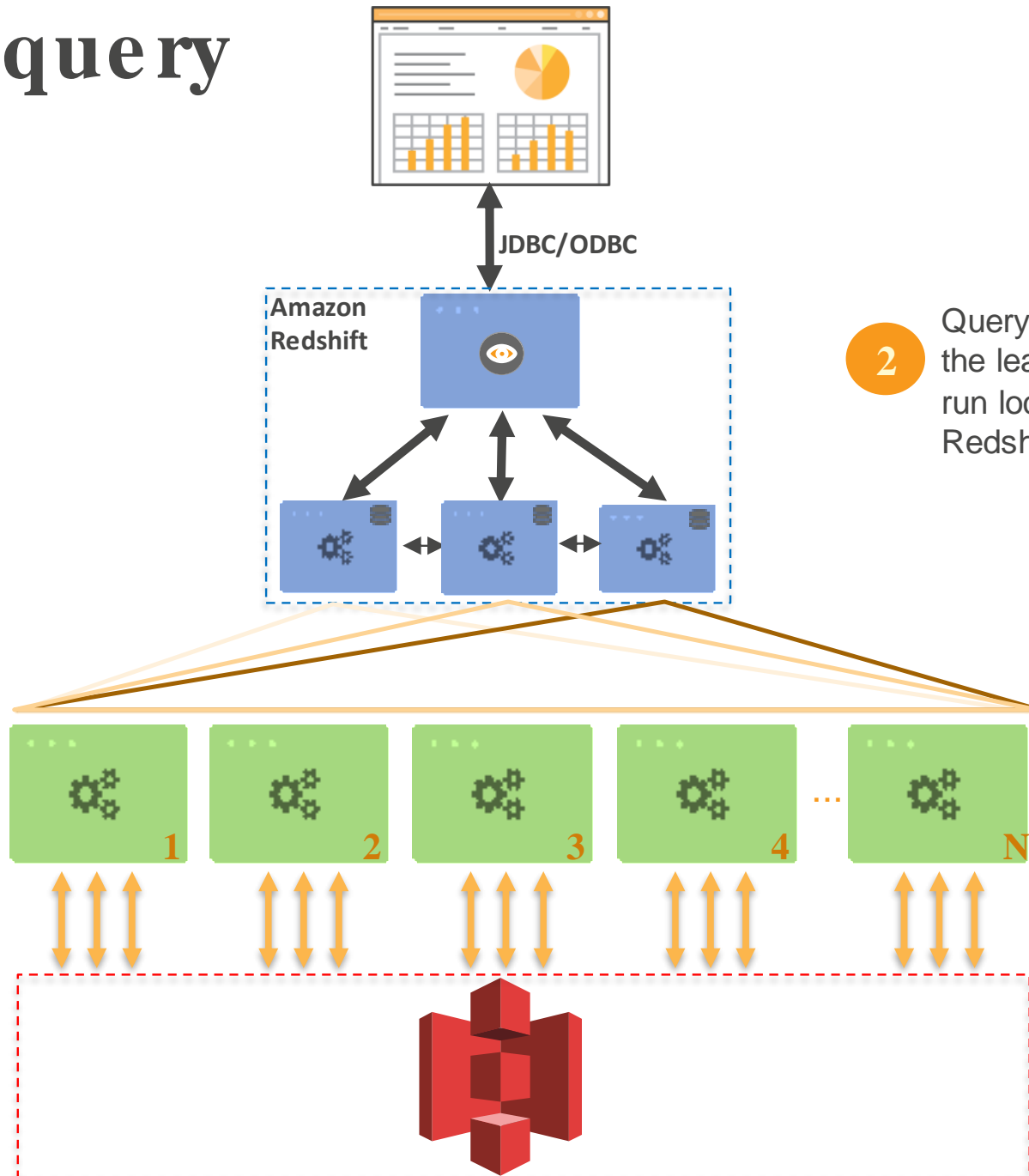


Amazon S3
Exabyte-scale object storage



Data Catalog
Apache Hive Metastore

Life of a query



2

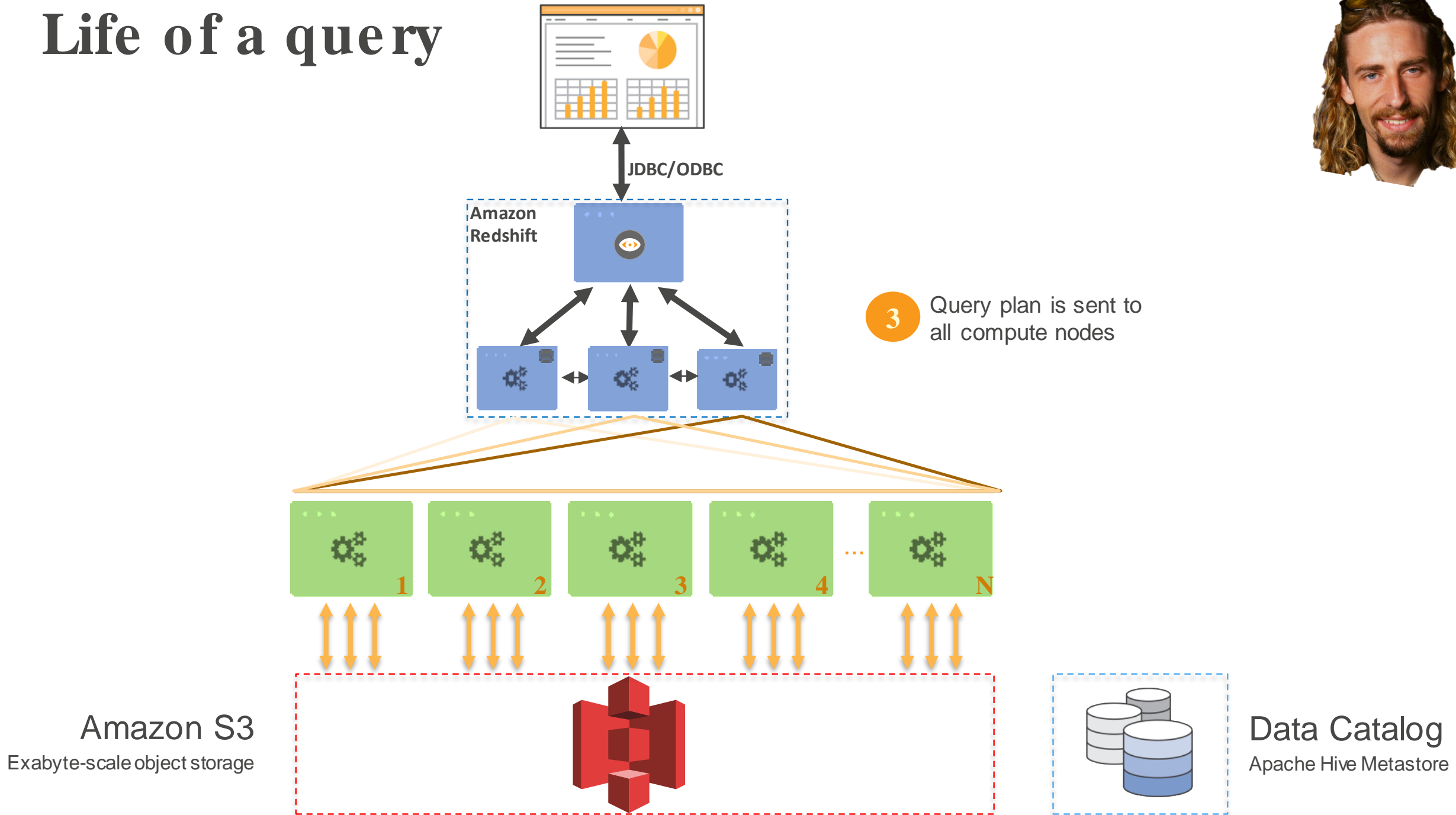
Query is optimized and compiled at the leader node. Determine what gets run locally and what goes to Amazon Redshift Spectrum

Amazon S3
Exabyte-scale object storage

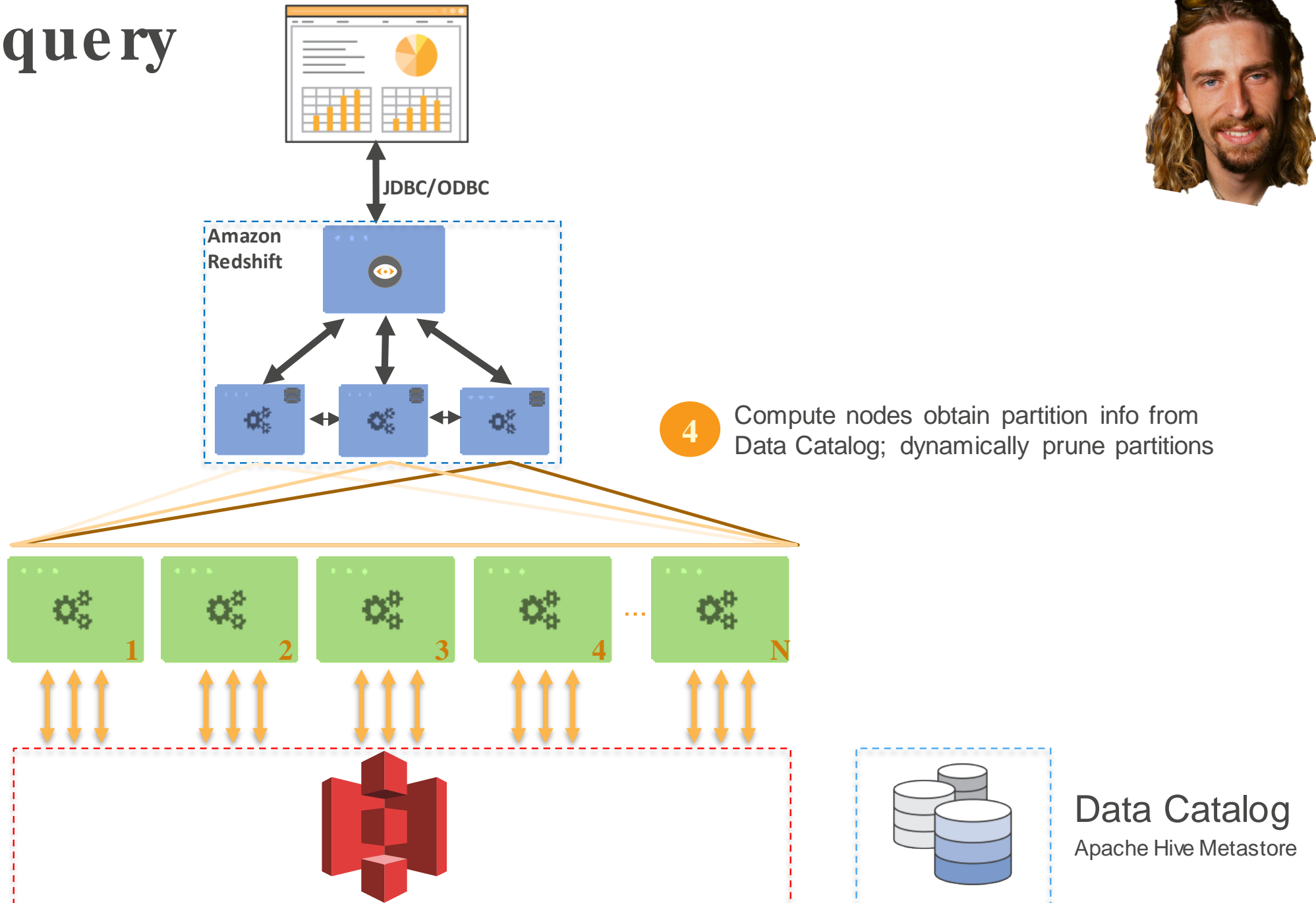


Data Catalog
Apache Hive Metastore

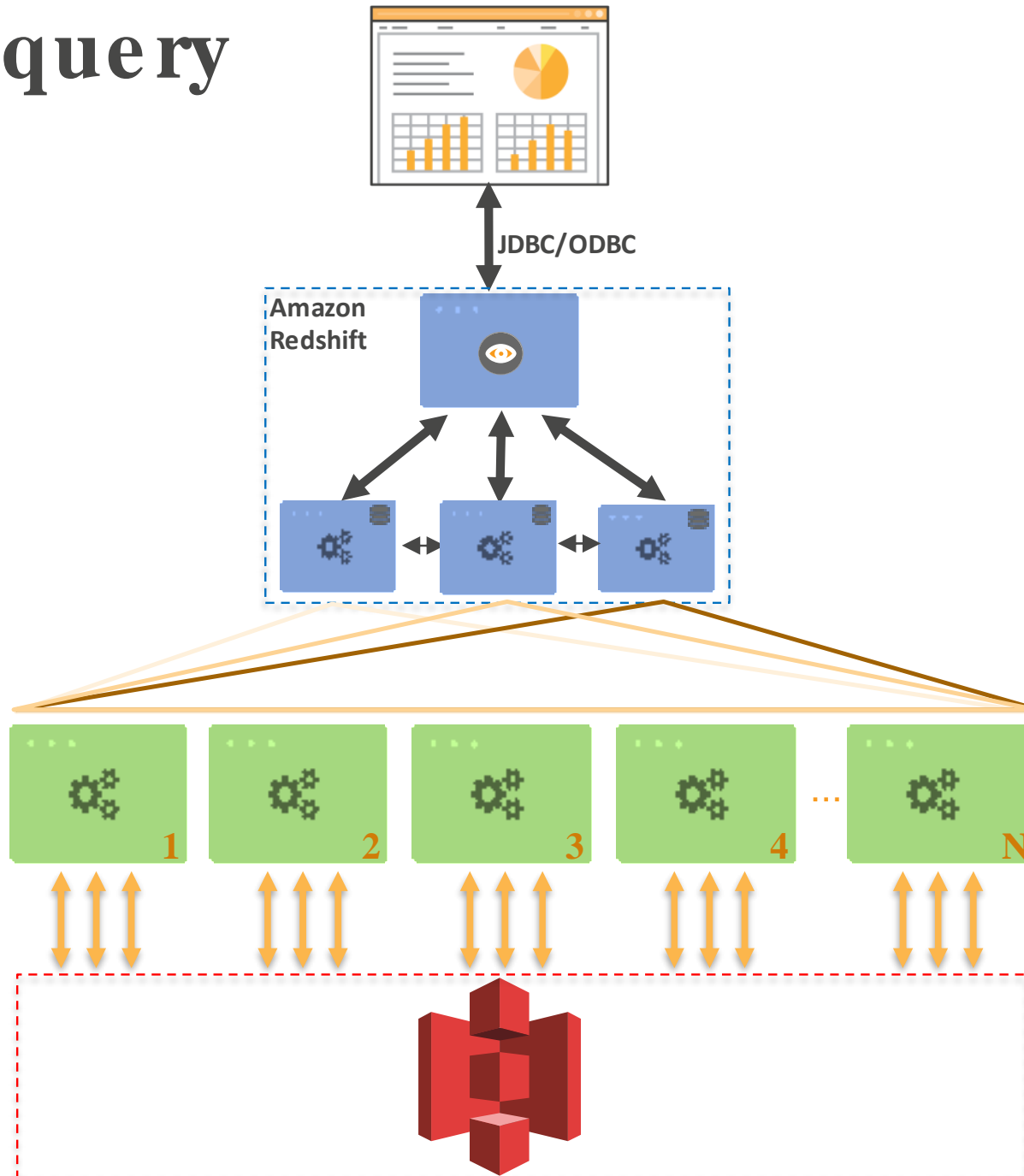
Life of a query



Life of a query



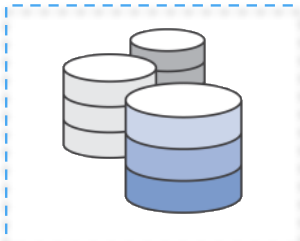
Life of a query



5

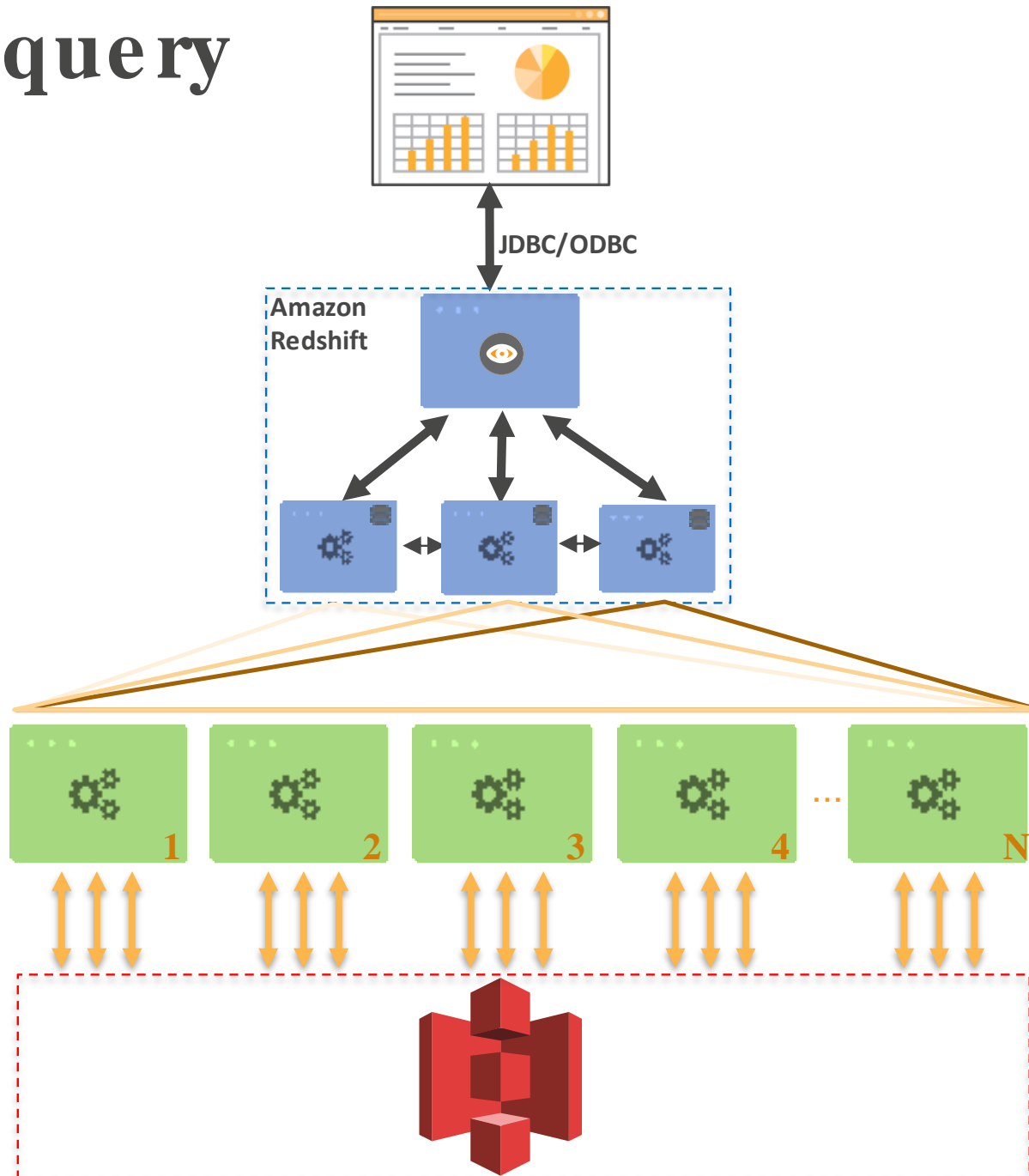
Each compute node issues multiple requests to the Amazon Redshift Spectrum layer

Amazon S3
Exabyte-scale object storage



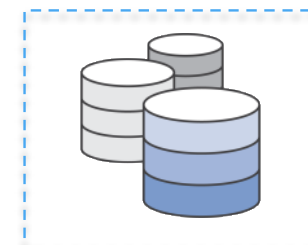
Data Catalog
Apache Hive Metastore

Life of a query



6 Amazon Redshift Spectrum nodes scan your S3 data

Amazon S3
Exabyte-scale object storage

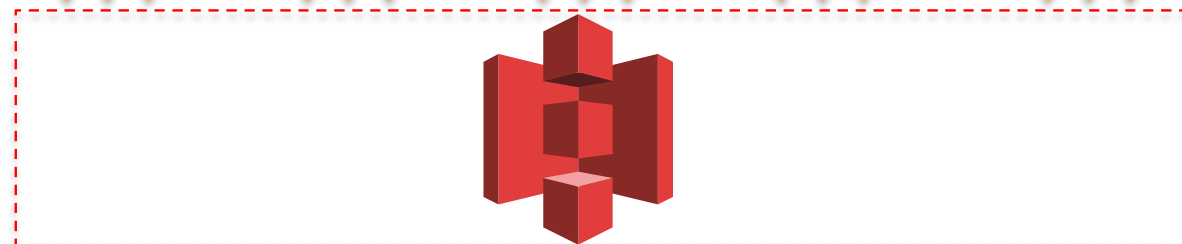
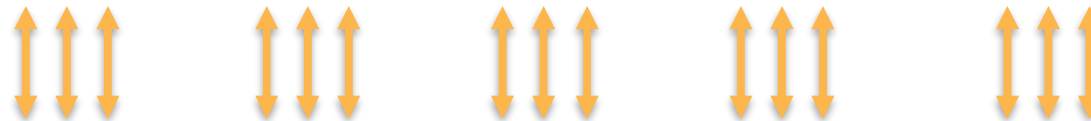
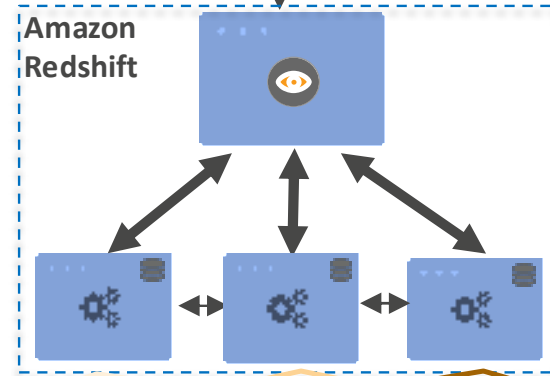


Data Catalog
Apache Hive Metastore

Life of a query



JDBC/ODBC

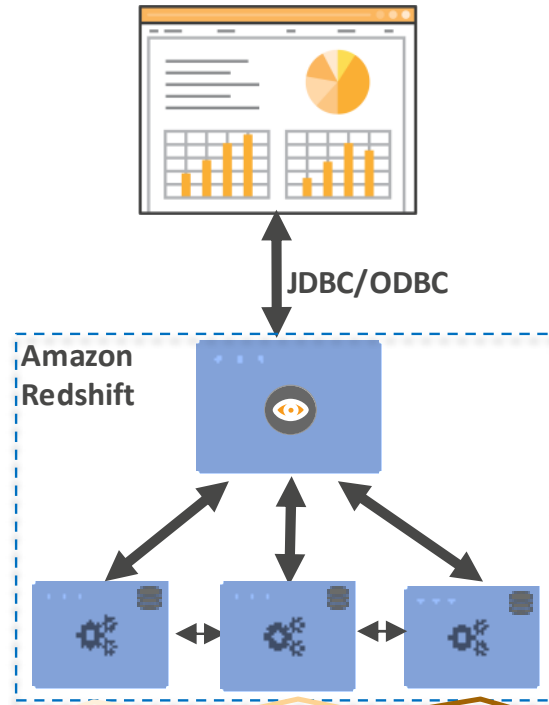


7 Amazon Redshift Spectrum projects, filters, joins and aggregates

Amazon S3
Exabyte-scale object storage

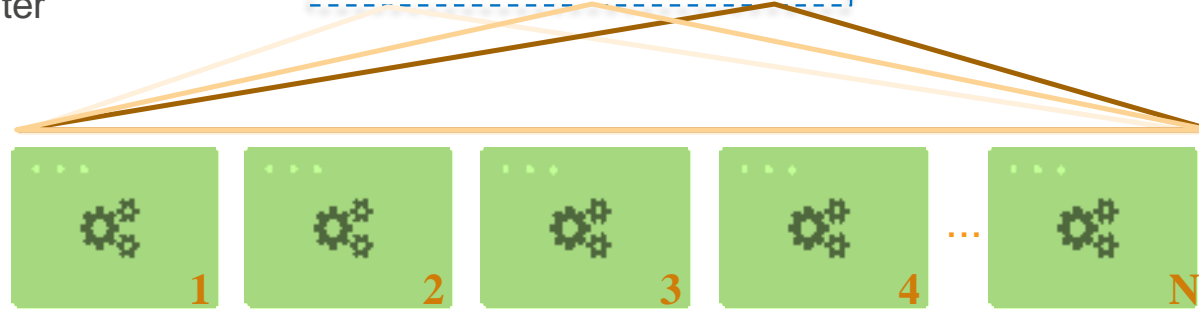
Data Catalog
Apache Hive Metastore

Life of a query

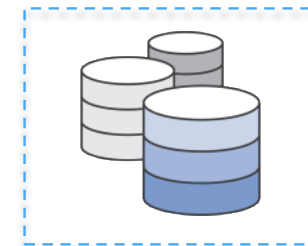
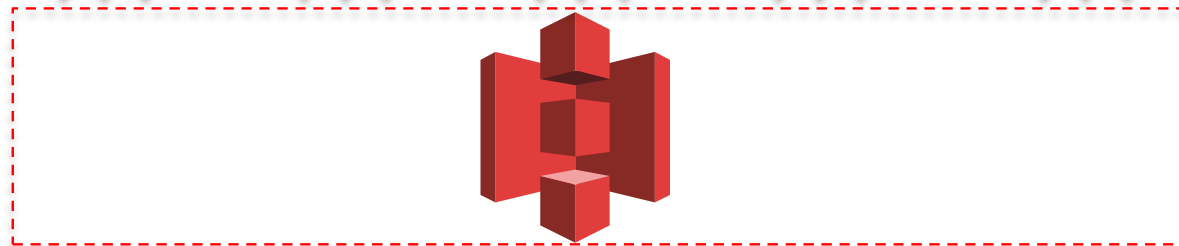


8

Final aggregations and joins with local Amazon Redshift tables done in-cluster



Amazon S3
Exabyte-scale object storage

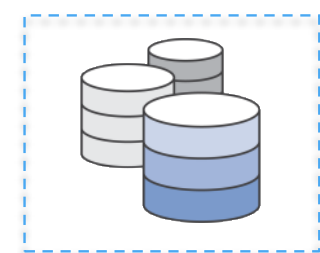
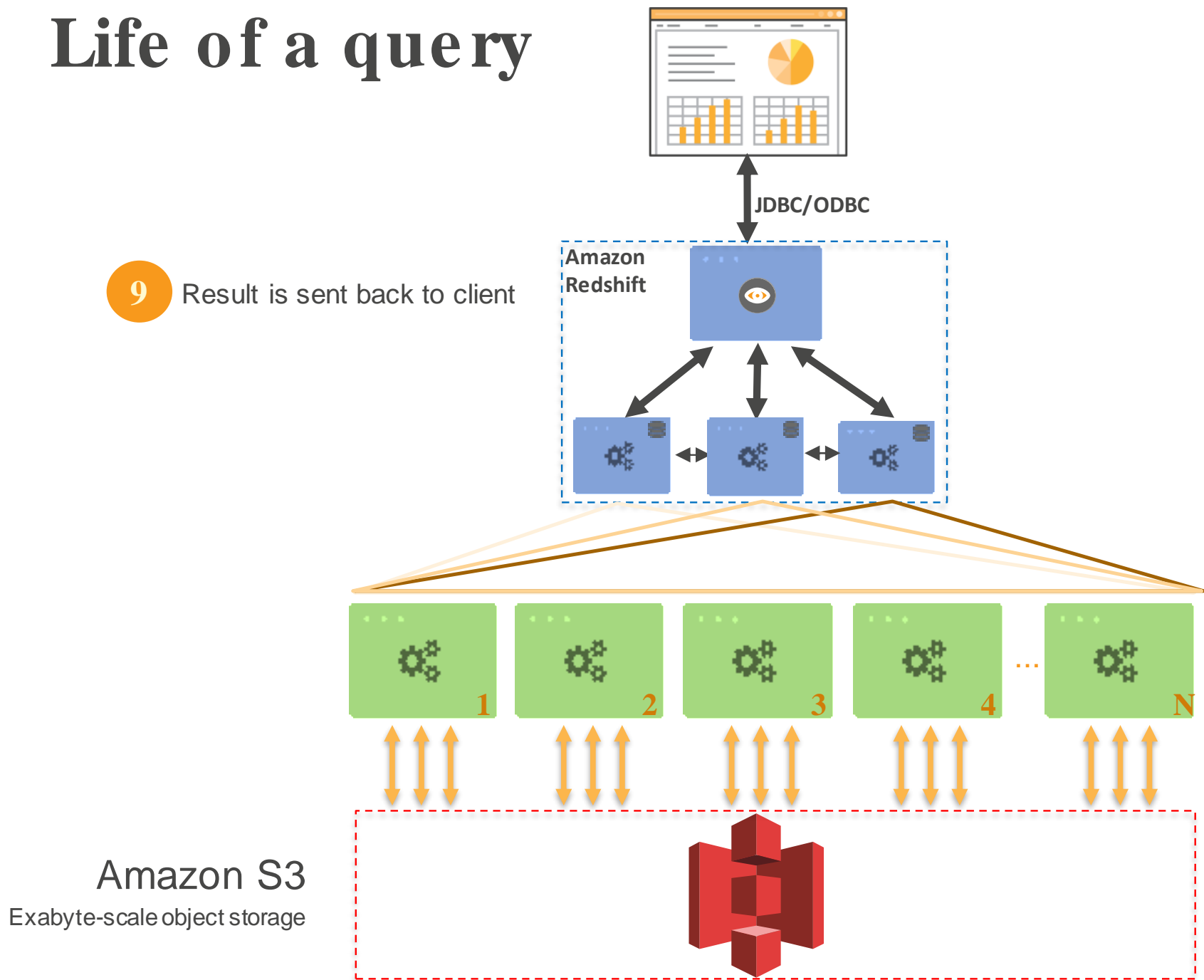


Data Catalog
Apache Hive Metastore

Life of a query

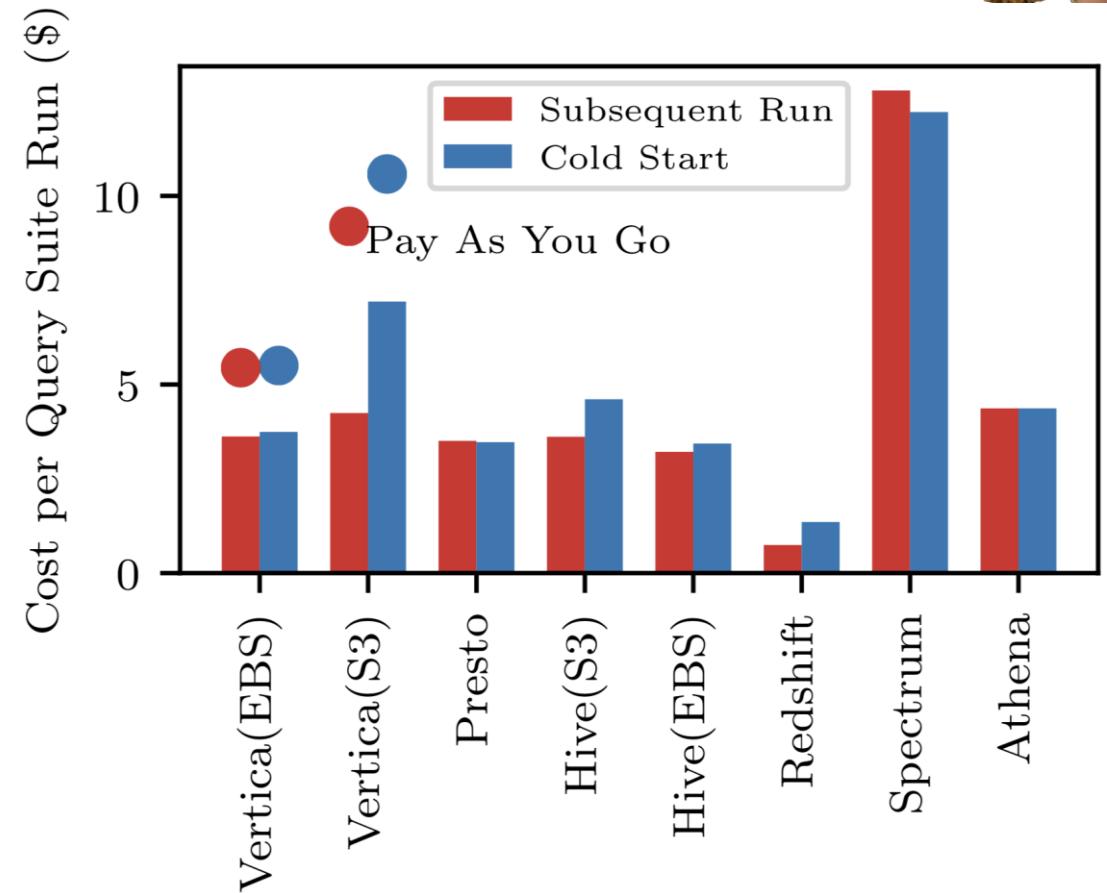
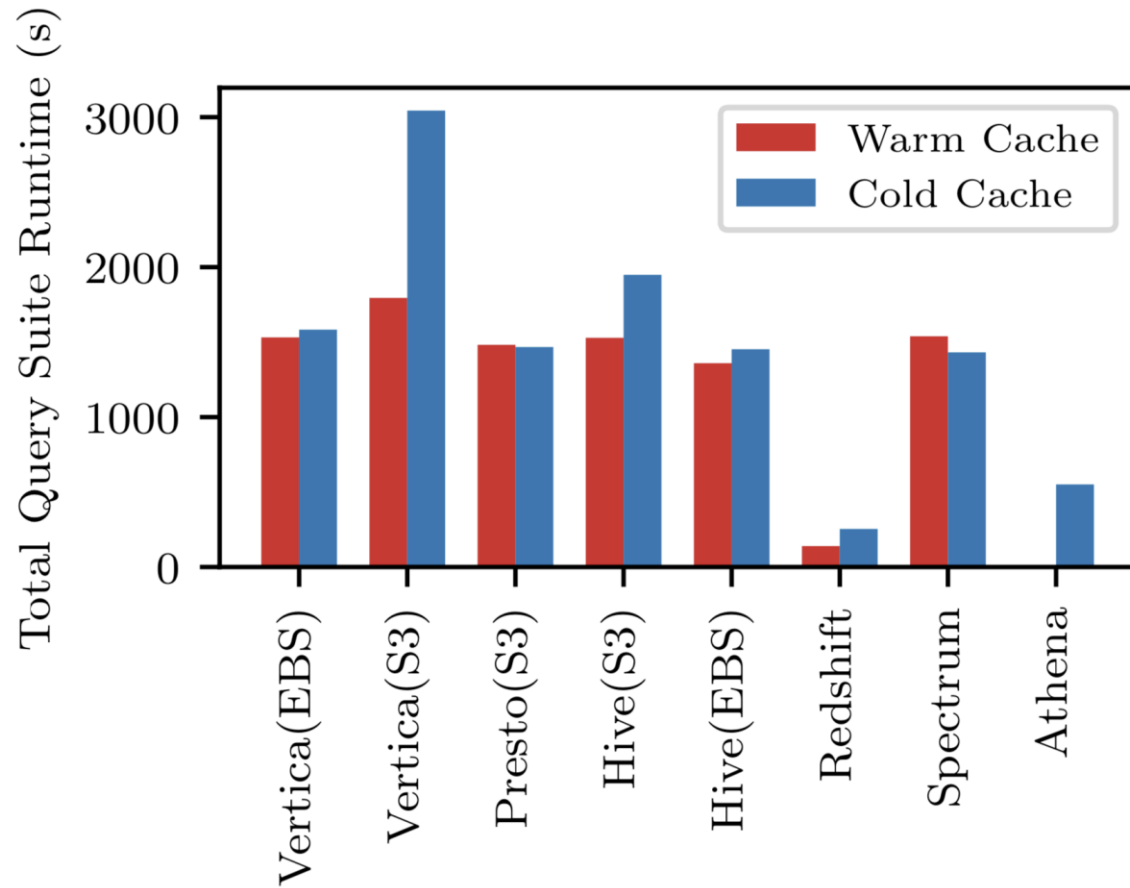


9 Result is sent back to client



Data Catalog
Apache Hive Metastore

Comparison of DW



[Junjay Tan](#), [Thanaa Ghanem](#), [Matthew Perron](#), [Xiangyao Yu](#), Michael Stonebraker, [David J. DeWitt](#), [Marco Serafini](#), [Ashraf Aboulnaga](#), [Tim Kraska](#): **Choosing A Cloud DBMS: Architectures and Tradeoffs.** [PVLDB 12\(12\)](#): 2170-2182 (2019)

AWS Lambda



<https://aws.amazon.com/blogs/architecture/ten-things-serverless-architects-should-know/>

**Take Your Nickel Back
k.com**



TakeYourNickelBack.com



- As we get more cash hungry, we decide to do analytics on our companies data. We purchase data from Cambridge Analytica to find more potential customers (shh)
- Now we need an analytical system that can store tens of terabytes of data and query it quickly (for our new Marketing team)



Data Warehousing systems

- Not as many options for Open source systems
- Mostly have columnar storage and are optimized for analytical queries on TBs of data
- Run SQL but are not optimized for small transactions
- Usually ingesting new data from operational database at a granularity of minutes to days





Extract Transform Load

- Need a way to get our data from operational system to the analytical systems.
- Extract data from the operational system (where we are making sales)
- Transform it as necessary for our analytical system
- Load it into the analytical system
- Could use Spark or Hadoop to write these jobs (and many others)



DimeStor

The official unofficial Nickelback Fan-Store



e

NICKELBACK

FEED THE MACHINE TOUR



Unfortunately it only got worse





Last attempt – Pay for positive reviews



Make Money by working on HITs

HITs - *Human Intelligence Tasks* - are individual tasks that you work on. [Find HITs now.](#)

As a Mechanical Turk Worker you:

- Can work from home
- Choose your own work hours
- Get paid for doing good work



Get Results from Mechanical Turk Workers

Ask workers to complete HITs - *Human Intelligence Tasks* - and get results using Mechanical Turk. [Get started.](#)

As a Mechanical Turk Requester you:

- Have access to a global, on-demand, 24 x 7 workforce
- Get thousands of HITs completed in minutes
- Pay only when you're satisfied with the results



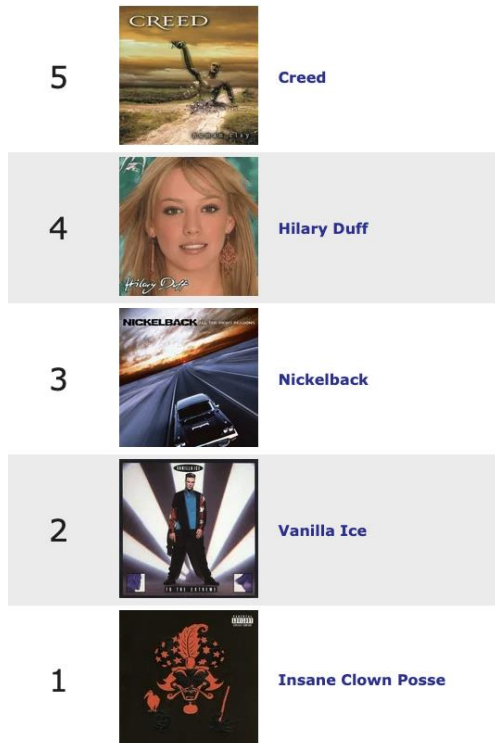


Microtasking – Virtualized Humans

- Current leader: Amazon Mechanical Turk
- Requestors place Human Intelligence Tasks (HITs)
 - Minimum price: \$0.01
 - Other parameters: #of replicas (assignments), expiration, **User Interface**,...
 - API-based: “createHit()”, “getAssignments()”, “approveAssignments()”, “forceExpire()”
 - Requestors approve jobs and payment
- Workers (a.k.a. “turkers”) choose jobs, do them, get paid

Aftermath

Nickelback ranked one of the worst bands ever



According to rollingstone.com



Readers' Poll: The Ten Worst Bands of the Nineties

1. Creed. It's no surprise that Creed won this poll.
2. Nickelback. It's hard not to feel a little bad for Nickelb
3. Limp Bizkit. If you think that Limp Bizkit fans are a bu
bullies, you aren't alone. ...
4. Hanson. ...
5. Nirvana. ...
6. Hootie and the Blowfish. ...
7. Bush. ...
8. Spin Doctors. ...



DimeStore was banned on AWS
because of UA violations

BANNED

<https://www.rollingstone.com/music/music-lists/readers-poll-the-ten-worst-bands-of-the-nineties-13654/5-nirvana-241095/>