Scaling Databases to Infinity and Beyond!

(Well, almost Infinity!)

NoCOUG – August 2021

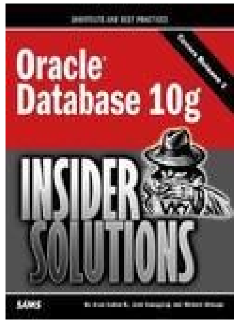
John Kanagaraj, Sr. MTS, Data Architecture, PayPal

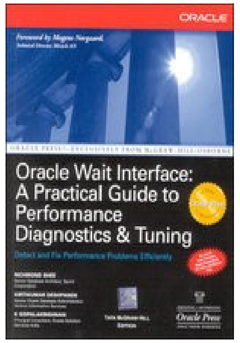
Agenda

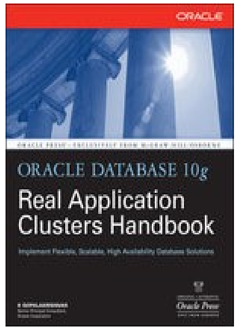
Intro

About the Speaker

- Currently Sr. Database/Data Architect @ PayPal
- Has been working with Oracle Databases and UNIX for 3+ decades
- Working on various NoSQL/Big Data technologies for the past 6 years
- Design and Implement High scale systems – Both Oracle and NoSQL
- Author, Technical editor, Oracle ACE *Alumni*, Frequent speaker
- Loves to mentor new speakers and authors!
- http://www.linkedin.com/in/johnkanagaraj

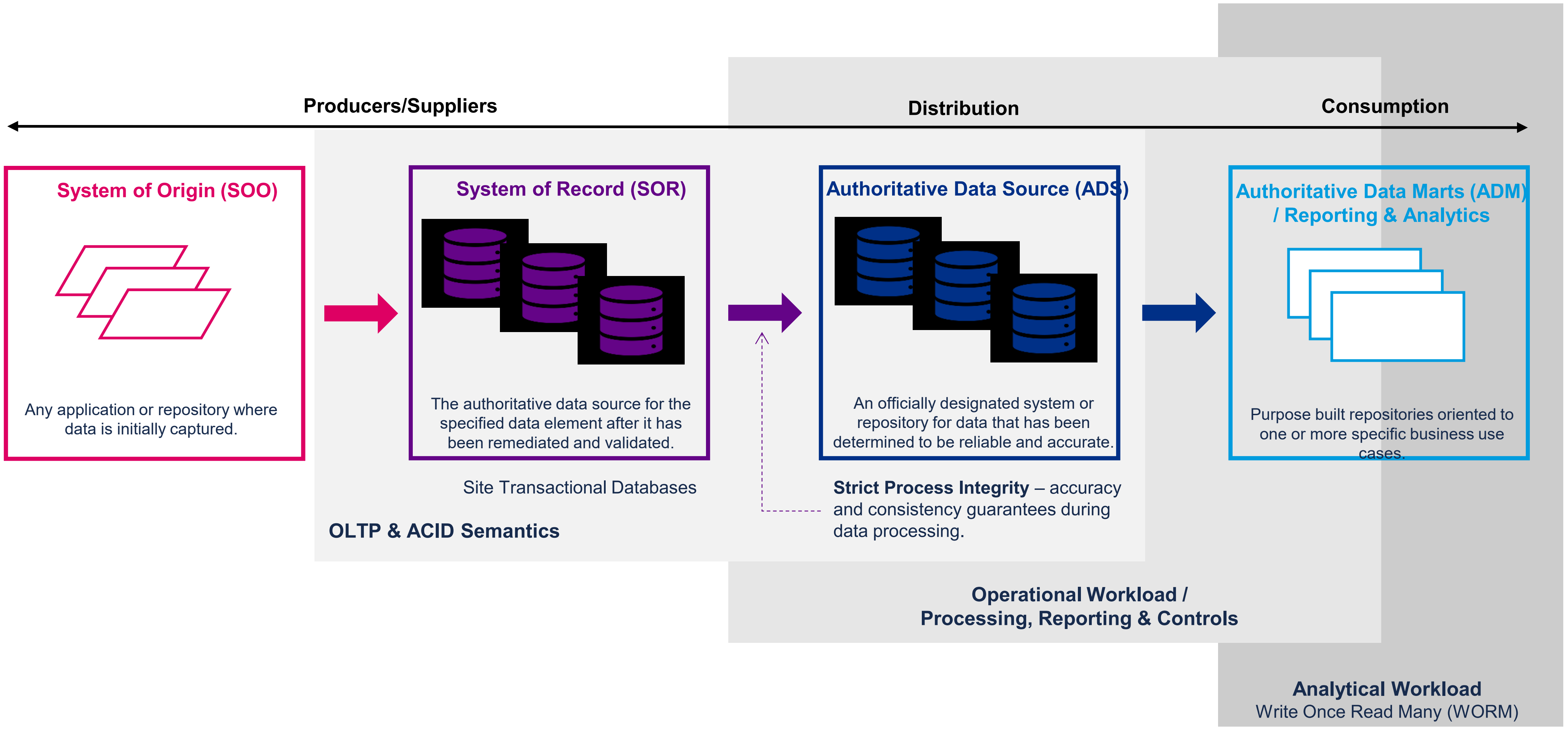

Oracle Database 10g INSIDER SOLUTIONS


Oracle Wait Interface: A Practical Guide to Performance Diagnostics & Tuning


ORACLE DATABASE 10g Real Application Clusters Handbook

ORACLE ACE Alumni

Definitions: Data Platforms



System of Origin (SOO)

Any application or repository where data is initially captured.

System of Record (SOR)

The authoritative data source for the specified data element after it has been remediated and validated.

Site Transactional Databases

OLTP & ACID Semantics

Authoritative Data Source (ADS)

An officially designated system or repository for data that has been determined to be reliable and accurate.

Strict Process Integrity – accuracy and consistency guarantees during data processing.

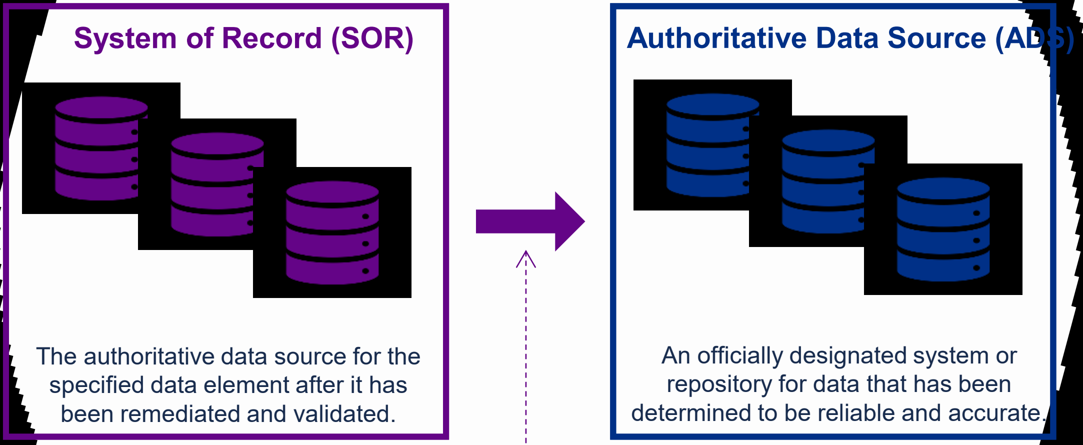
Authoritative Data Marts (ADM) / Reporting & Analytics

Purpose built repositories oriented to one or more specific business use cases.

Operational Workload / Processing, Reporting & Controls

Analytical Workload
Write Once Read Many (WORM)

Site Data Architecture



System of Record (SOR)

The authoritative data source for the specified data element after it has been remediated and validated.

Site Transactional Databases

Authoritative Data Source (ADS)

An officially designated system or repository for data that has been determined to be reliable and accurate.

Strict Process Integrity – accuracy and consistency guarantees during data processing.

Core Tenets

PayPal

Know [Data About] Your Data (KYD)

First Quarter 2021 Summary

Strong performance across key performance metrics

Active Accounts

392M

Includes **31M** active merchant accounts

21% increase y/y

14.5M

Net new active accounts (NNAs)

28% decrease y/y*

*Adjusting for the one-time addition of 10.2 million NNAs from the acquisition of Honey in Q1-20, NNAs grew 45% y/y

Customer Engagement

42.2

Payment transactions per active account (TPA)

7% increase y/y

33% increase y/y in daily active accounts using PayPal core experiences

Total Payment Volume

$285B

>$1T on a trailing 12-month basis

50% spot and **46%** FX-neutral y/y growth

Revenue

$6.03B

31% spot and **29%** FX-neutral y/y growth

Non-GAAP EPS(1)

$1.22

84% increase y/y

Free Cash Flow(1)

$1.5B

27% increase y/y

25% as % of revenue

Site Data Architecture

Core Tenets

SECURITY

Resiliency

99.999% = 5.26 minutes/year

Scalability

Distributed Data Stores

Scale-Out Architecture

Cost of Ownership

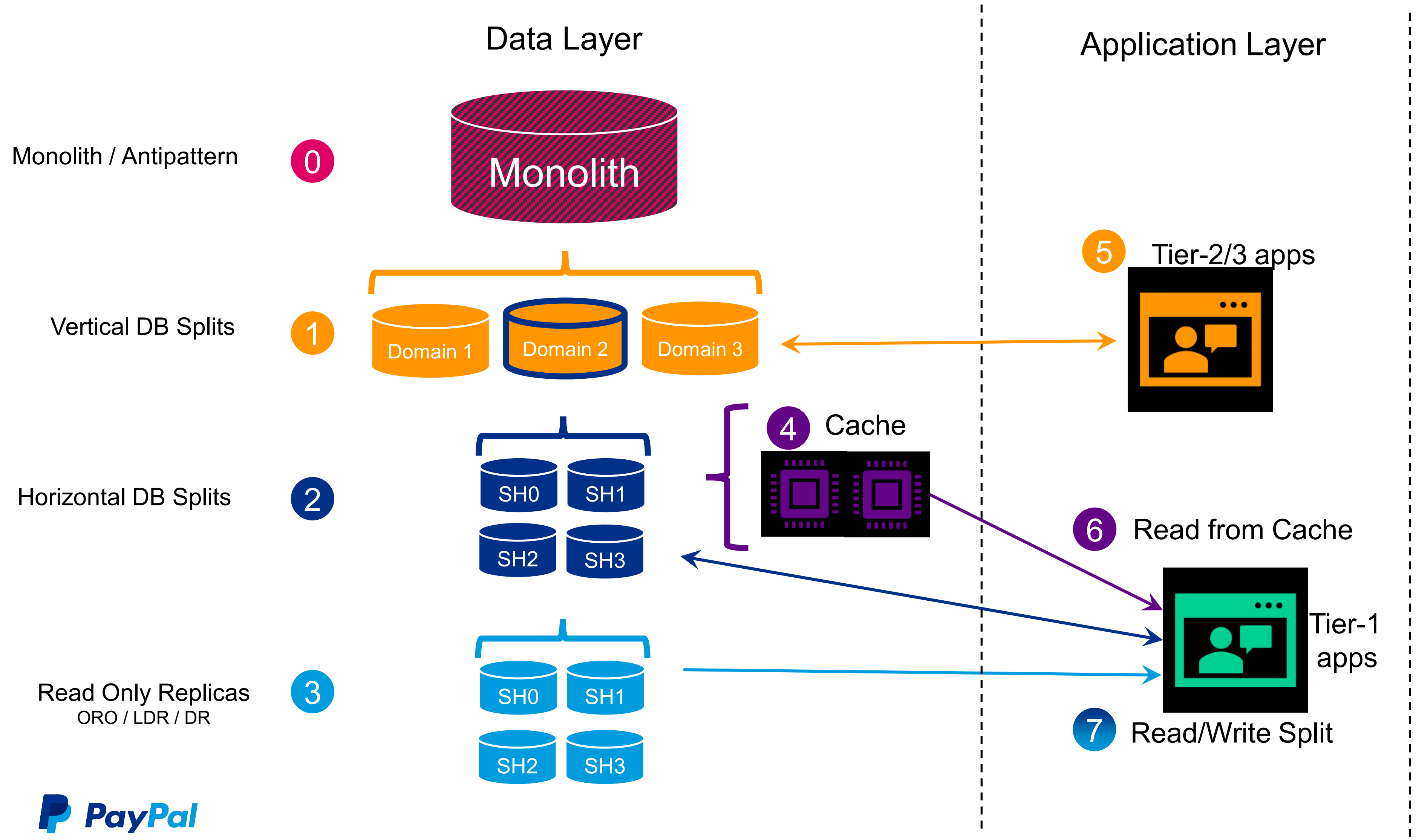
Gotta keep count of the $$!

Challenges at Scale

- Pushing the limits
  - Connections
  - Memory
  - Interconnect
  - CPU
  - DDL on busy tables
  - RAC reconfiguration
  - Redo rate
  - I/O latencies
  - SAN Storage limits
  - Replication latencies
  - HA requirements

- Solutions
  - Custom Connection pooling and multiplexing (OCC)
  - Read Scale out (replication)
  - Microservice oriented architecture (logical separation)
  - Custom HA caching (Juno)
  - Custom Sharding
  - Active-Active operation using Oracle RO‘s and GoldenGate
  - Storage Tiering and Archiving
  - .....
  - Moving to Cloud! ☺

Scaling and Blast Radius Resiliency Patterns



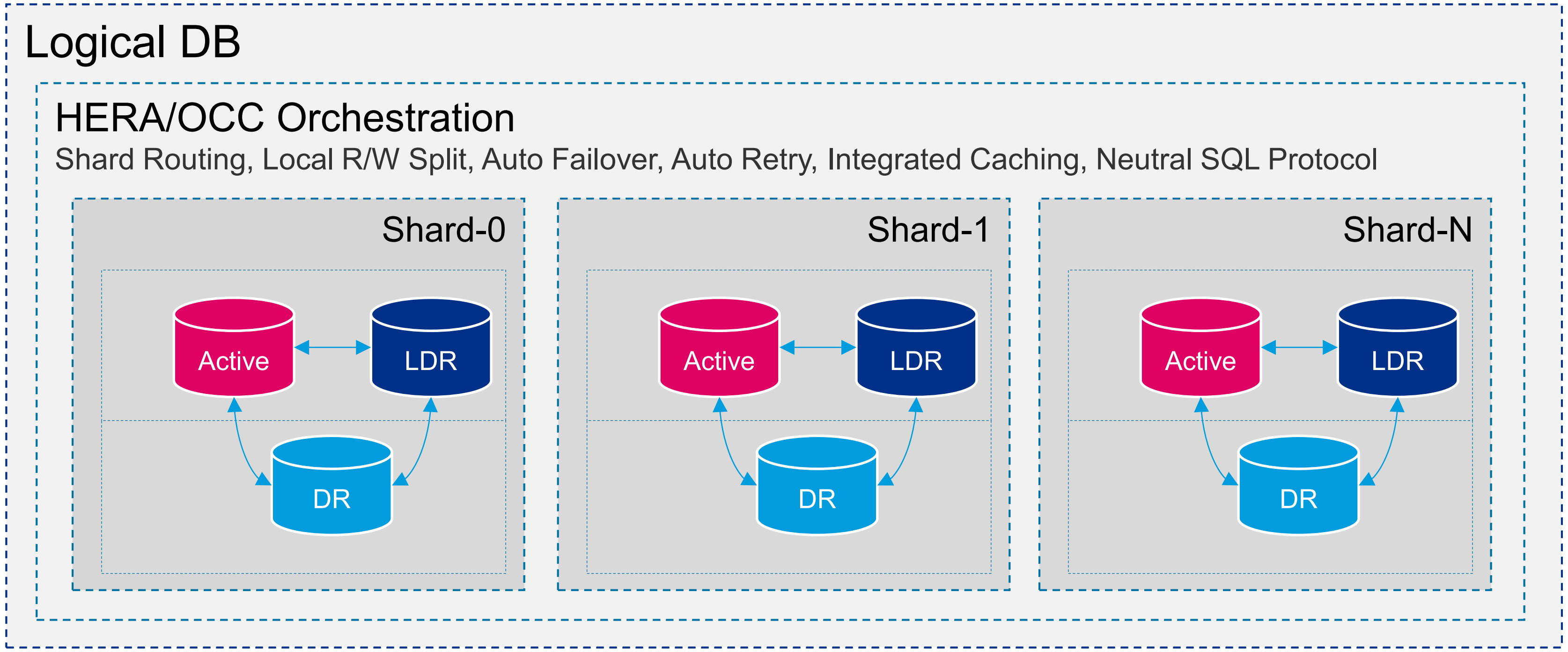
Observations

- Cutovers and Failovers were the only way to know the blast radius.
- When Tier-1 service starts to depend on non-Tier-1 db.
- Why different shards have different blast radius (>2x).
- Fragmented R/W traffic split for Tier-1 services.

PayPal

Resiliency Vision

- Commodity Shards
- Light DB Units
- DB Engine Agnostic
- HERA/OCC Fronted
- DB Neutral Wire Protocol
- Push Button Automation



Logical DB
HERA/OCC Orchestration
Shard Routing, Local R/W Split, Auto Failover, Auto Retry, Integrated Caching, Neutral SQL Protocol
Shard-0
Active
LDR
DR
Shard-1
Active
LDR
DR
Shard-N
Active
LDR
DR

- Design for Survival
- RTO in Seconds
- RPO is Zero
- Autonomous Disaster Detection
- Autonomous DR Triggering
- Automated DR Procedures

PayPal

Keys: Timed UUID

Timed UUID / PayPal Variant

UUID Version
1- timed; 4 - random

128-bit number in hexadecimal format: 6E8BC430-9C3A-11D9-9669-0800200C9A66

[time-low]-[time-mid]-[version-and-time-high]-[clock-misc.]-[node]

count of 100 nanosecond intervals
Since 10/15/1582 00:00:00.00 UTC

Decimal equivalent:	[SS.FFFFFF]-[HH24:MI]-[YYYY-MM-DD]	[50.123456]-[11:20]-[2021-05-27]
Decimal equivalent **unshuffled:**	[YYYY-MM-DD]-[HH24:MI]-[SS.FFFFFF]	[2021-05-27]-[11:20]-[50.123456]

Hexadecimal **unshuffled:** 11D9-9C3A-6E8BC430-9669-0800200C9A66

4 bits 60 time-based bits 64 random bits

■ Globally Unique

■ Decentralized

■ Sortable

■ Primary Key, Idempotency Key, Time-Based Partition Key

PayPal

Timed UUID / PayPal Variant / Collision Rate

Hexadecimal **unshuffled**

$\underbrace{\texttt{1}}_{\text{4 bits}}\underbrace{\texttt{1D9-9C3A-6E8BC430}}_{\text{60 time-based bits}}\texttt{-}\underbrace{\texttt{9669-080F210C9A66}}_{\text{64 random bits}}$

■ Globally Unique
■ Decentralized
■ Sortable

To allow for 50% probability of one collision,

we need to generate 9.1×10^9 UUIDs within 100 nanoseconds (ns).

~7.9B people / 100 ns = 1 sec / 10M

PayPal

KYD: Table Categories

Table Categories

	Immutable	Mutable
Master Data	Reference or configuration data. Country Lookup	Standard business objects like merchant, customer, customer account, customer address, etc.. Customer, Customer Balance, Merchant

Table Categories

	Immutable	Mutable
Master Data	Reference or configuration data. Country Lookup	Standard business objects like merchant, customer, customer account, customer address, etc.. Customer · Customer Balance · Merchant
Transactional Data	These are time-based recordings of events. Customer Activity Log · Journal	Time-based events that are modifiable until they 'close', such as payments or customer cases. Payment · Customer Case

Hybrid Tables

[Master + Transactional Data]

[Do Not Mix]

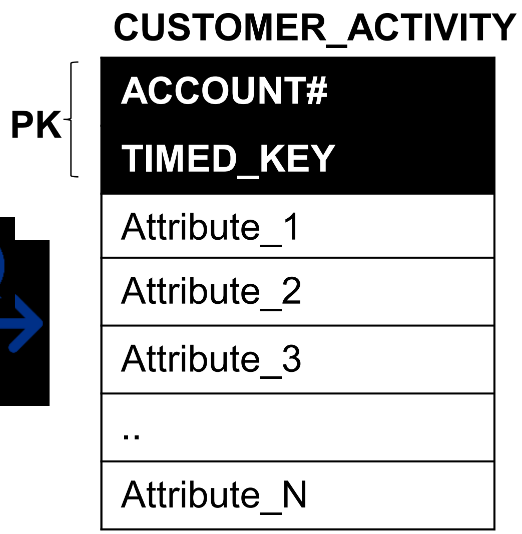
KYD: Access Path

PayPal

Problem Statement

1. Customer Activities

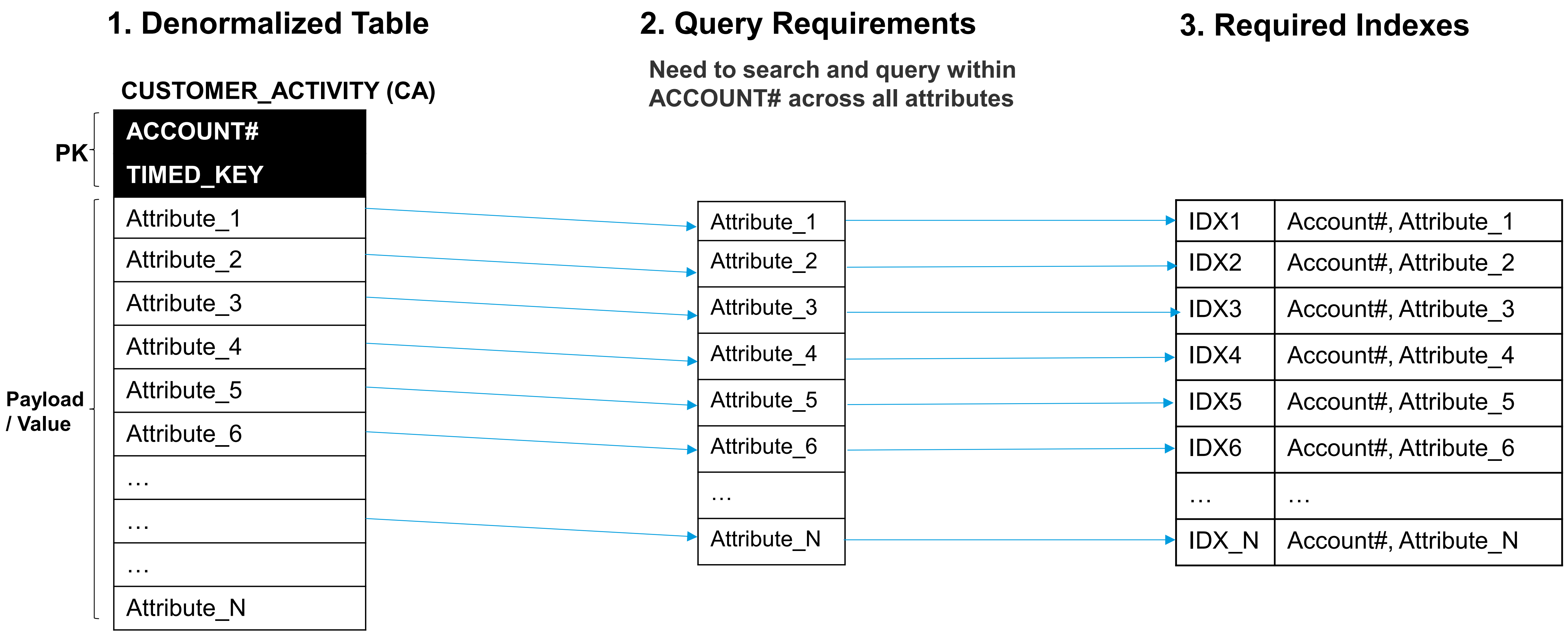
Acct #	Time	Attribute 1	Attribute 2	Attribute 3	Attribute N
100	t1	A	X	6	!@
400	t1	A	Y	7	#$
300	t2	B	X	8	%^
100	t3	C	Z	0	^&
300	t4	D	Z	9	*(
..	..	..	..	..	..



2. Query Requirements

Customers need to query and search their data.

Problem Statement cont.



1. Denormalized Table

CUSTOMER_ACTIVITY (CA)

PK	**ACCOUNT#**
	TIMED_KEY
Payload / Value	Attribute_1
	Attribute_2
	Attribute_3
	Attribute_4
	Attribute_5
	Attribute_6
	…
	…
	…
	Attribute_N

2. Query Requirements

Need to search and query within ACCOUNT# across all attributes

Attribute_1
Attribute_2
Attribute_3
Attribute_4
Attribute_5
Attribute_6
…
Attribute_N

3. Required Indexes

IDX1	Account#, Attribute_1
IDX2	Account#, Attribute_2
IDX3	Account#, Attribute_3
IDX4	Account#, Attribute_4
IDX5	Account#, Attribute_5
IDX6	Account#, Attribute_6
…	…
IDX_N	Account#, Attribute_N

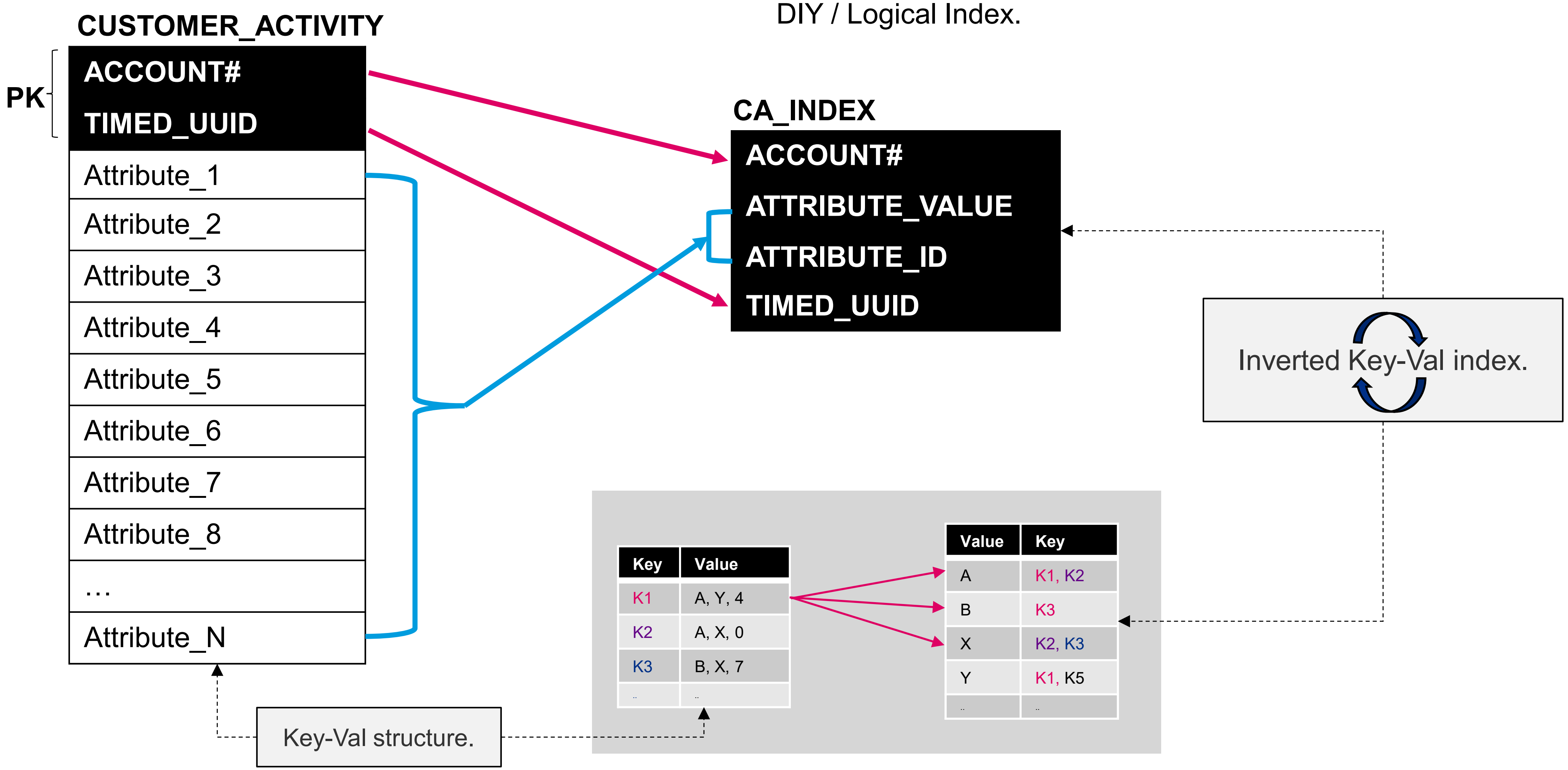
Denormalized Index

(Denormalized)^2

1. Denormalized Table

2. “Denormalized” Index

DIY / Logical Index.



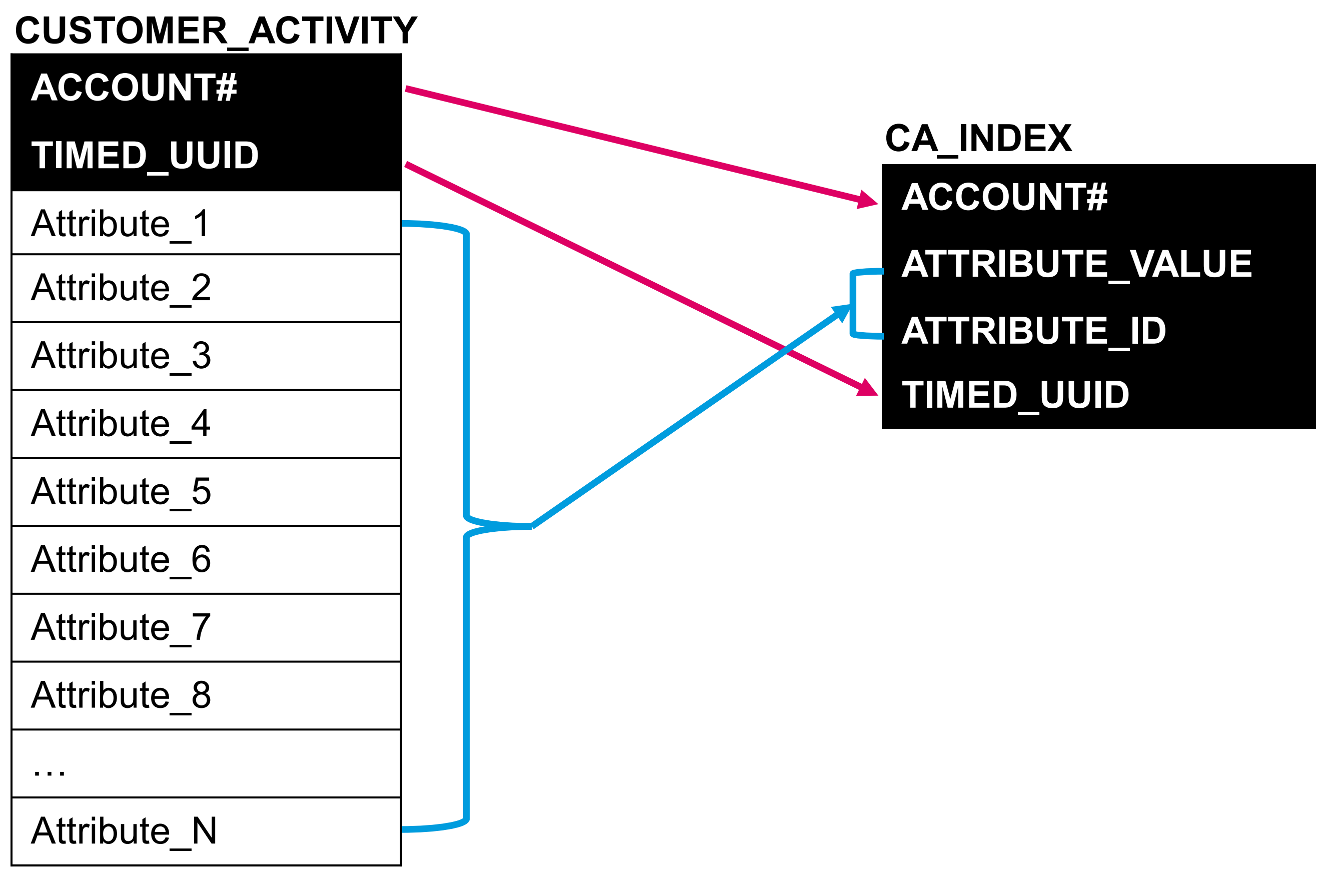
Key	Value
K1	A, Y, 4
K2	A, X, 0
K3	B, X, 7
..	..

Value	Key
A	K1, K2
B	K3
X	K2, K3
Y	K1, K5
..	..

Query

1. Denormalized Table

2. “Denormalized” Index



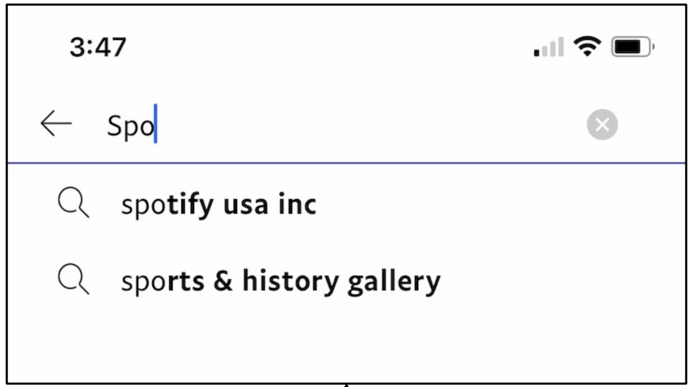
3. How does it work?

Phase 1: Query Index Table
Phase 2: Fetch from CA

```
--pseudo code
SELECT *
FROM customer_activity
JOIN
        (SELECT account#, timed_uuid   FROM
        ca_index
         WHERE account#=:a
           AND attribute_value = ‘Arch’
           AND attribute_id = 42
        ) i
 ON ca.account# = i.account#
AND ca.timed_uuid =  i.timed_uuid

--dual PK access
```

Search

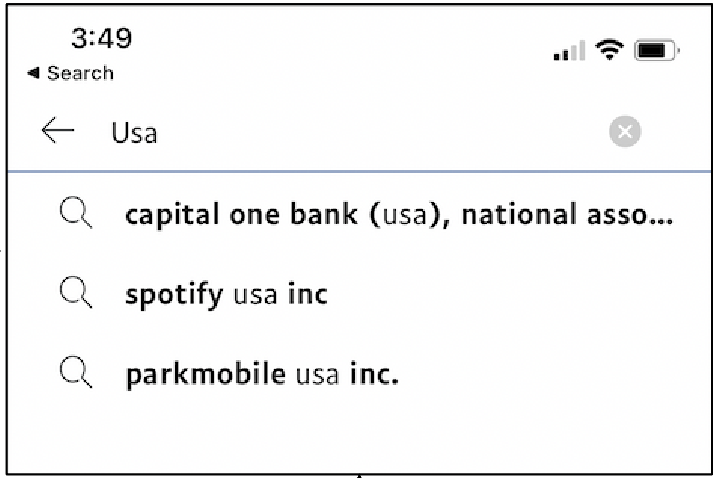


3:47

Spo

spotify usa inc

sports & history gallery



3:49

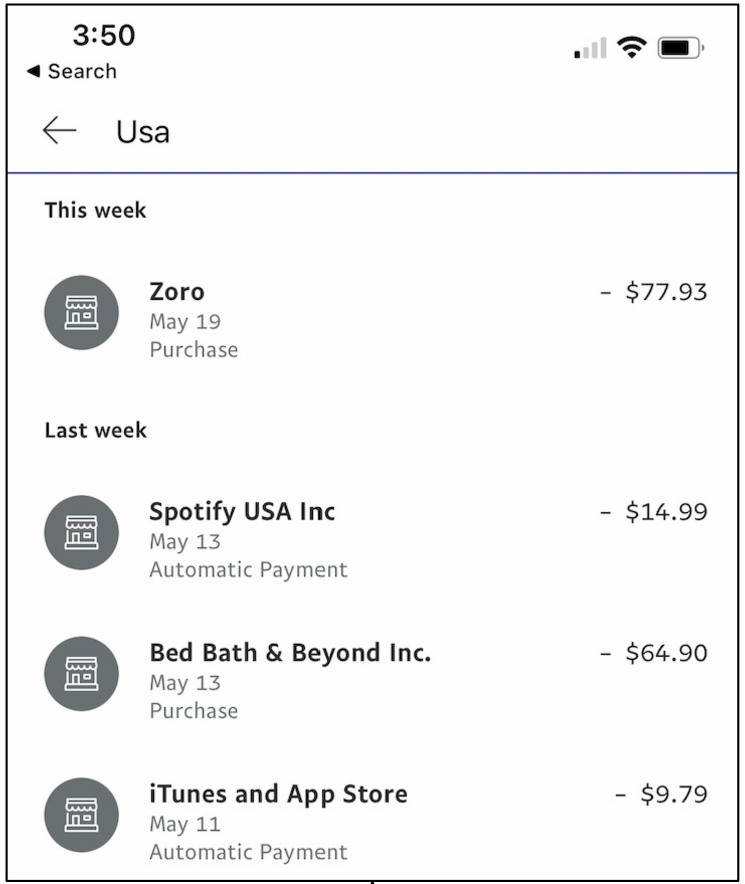
◂ Search

Usa

capital one bank (usa), national asso...

spotify usa inc

parkmobile usa inc.



3:50

◂ Search

Usa

This week

Zoro May 19 Purchase		– $77.93

Last week

Spotify USA Inc May 13 Automatic Payment		– $14.99
Bed Bath & Beyond Inc. May 13 Purchase		– $64.90
iTunes and App Store May 11 Automatic Payment		– $9.79

```
--Type Ahead
SELECT attribute_value
  FROM ca_index
 WHERE account#=:a
   AND attribute_value like '%USA%'
AND rownum < 10;
```

```
SELECT * FROM customer_activity
JOIN
              (SELECT account#, timed_uuid
                 FROM ca_index
                WHERE account#=:a
                  AND attribute_value = '%USA%'
                  AND attribute_id = 42
              ) i
 ON ca.account# = i.account#
AND ca.timed_uuid = i.timed_uuid
```

Scaling it Out

CUSTOMER_ACTIVITY

ACCOUNT#
TIMED_UUID
Attribute_1
Attribute_2
Attribute_3
…
Attribute_N

CA_INDEX

ACCOUNT#
ATTRIBUTE_VALUE
ATTRIBUTE_ID
TIMED_UUID

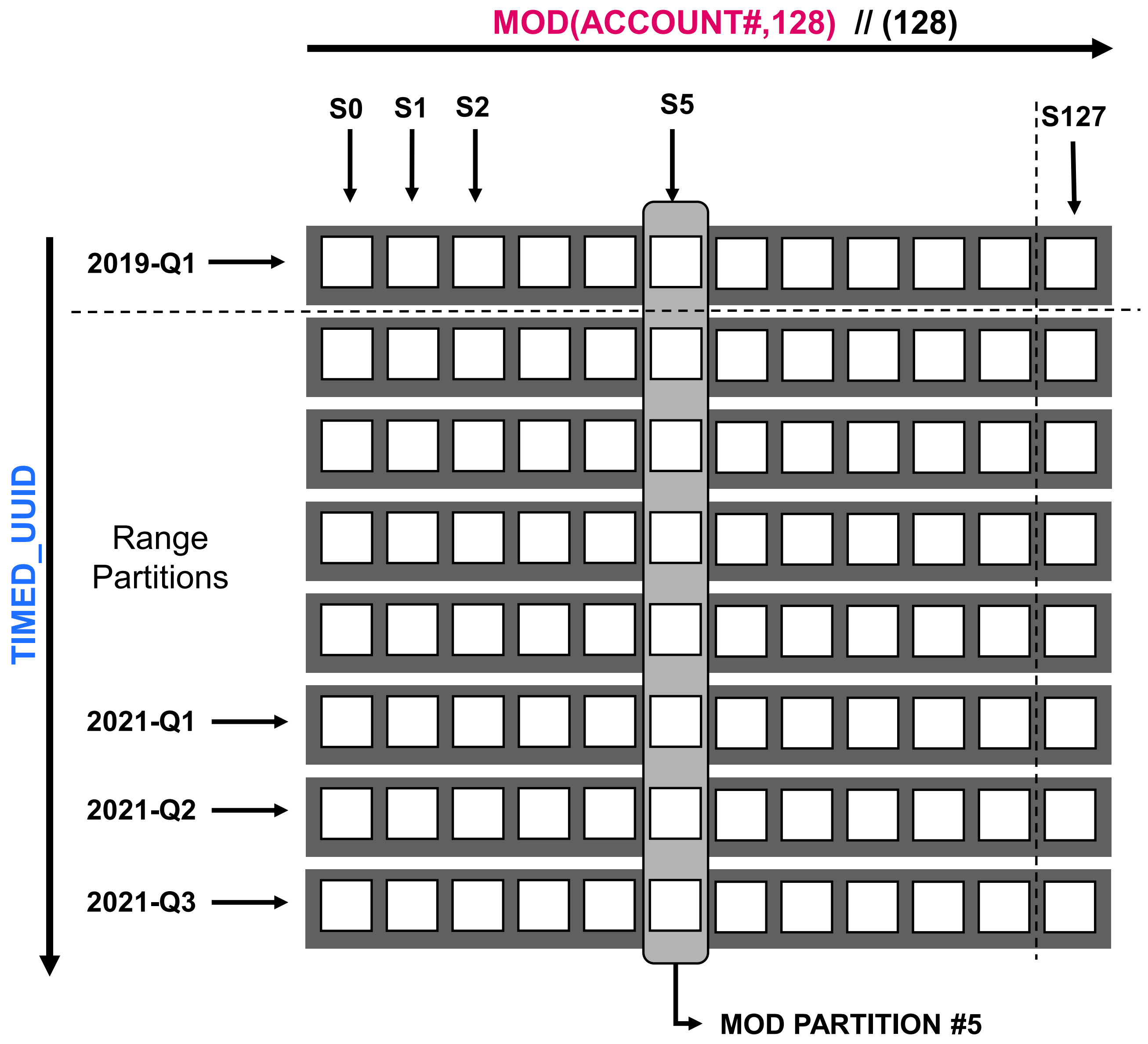
ACCOUNT# = Shard Key
TIMED_UUID = Range Key

PK on (ACCOUNT#, TIMED_UUID)

Current State (7+ y/o CAM)

- 8 Physical Shards
- ~100TB/Shard
- >billions of reads/writes/day/table

Sharding / Scale-Out



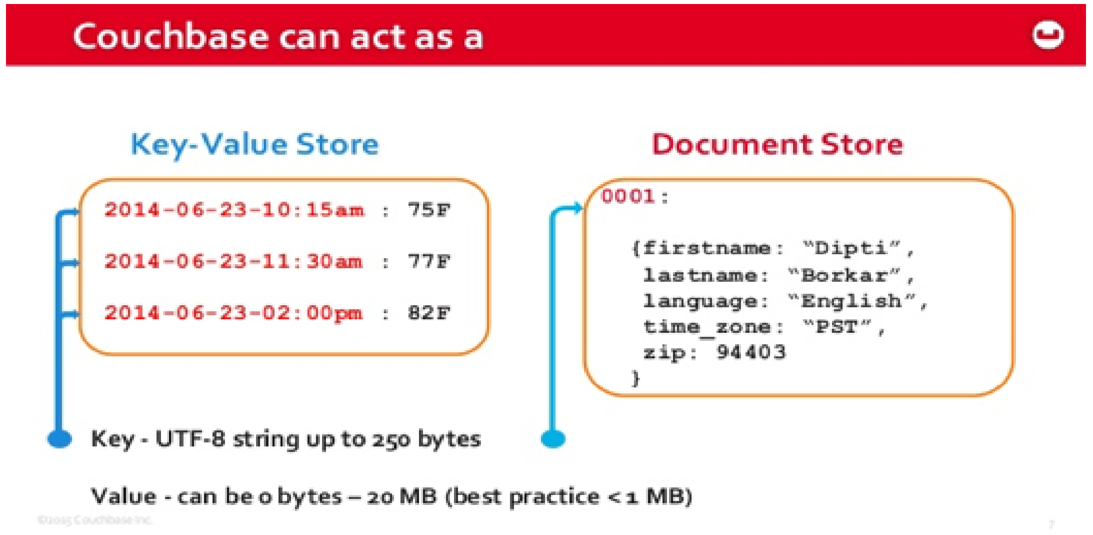
DB Engine Agnostic

Why use NoSQL?

Flexibility
- Flexible schemas that enable faster and iterative development.
- Ideal for semi-structured and unstructured datasets.

Scalability
- Designed to scale out by using distributed clusters of hardware.
- Some cloud providers manage it behind-the-scenes

Performance
- Optimized for specific data models (e.g., document, key-value)
- Optimized for access patterns.



Couchbase can act as a

Key-Value Store

```
2014-06-23-10:15am  :  75F

2014-06-23-11:30am  :  77F

2014-06-23-02:00pm  :  82F
```

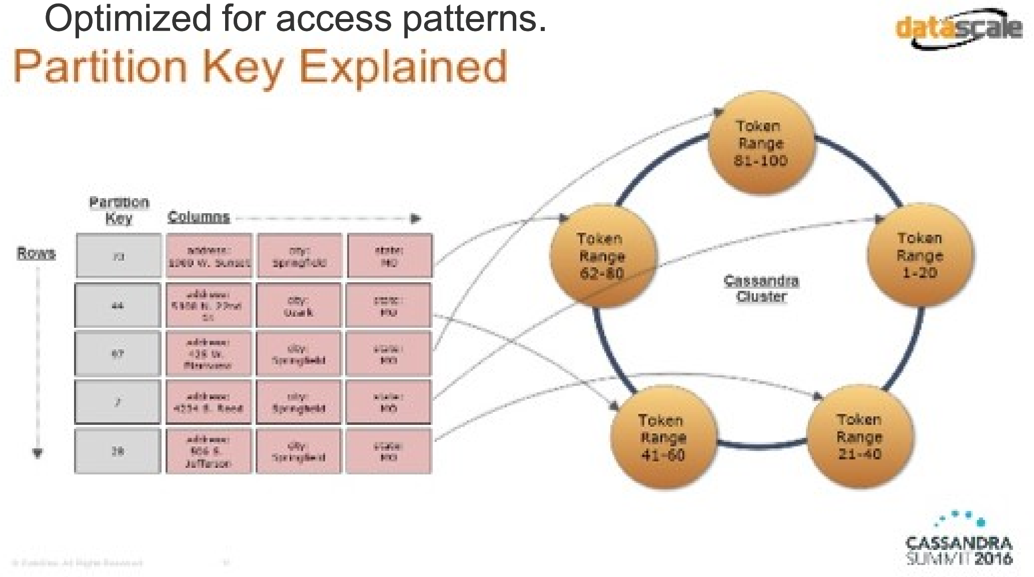
Document Store

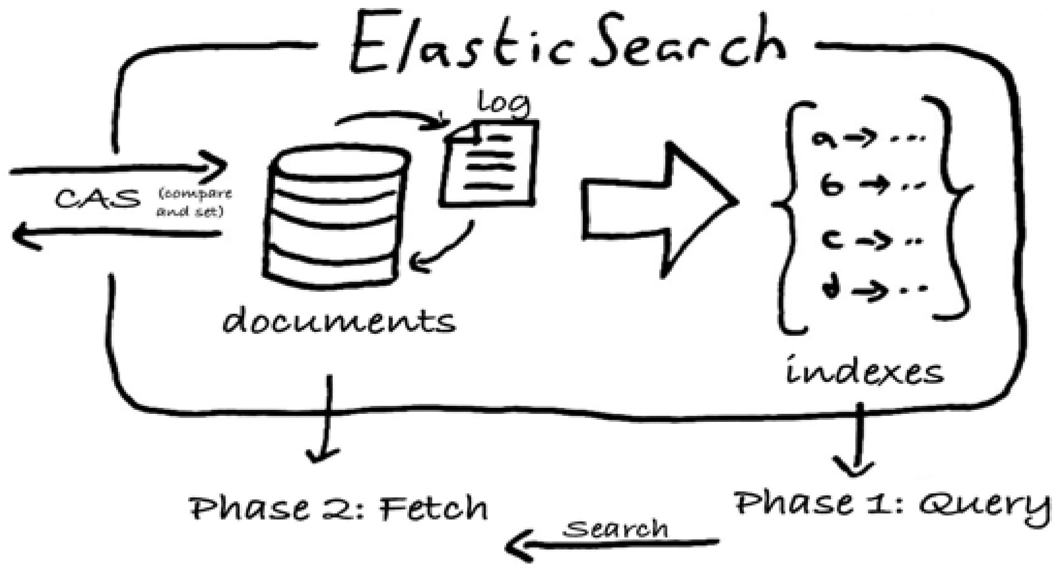
```
0001:

  {firstname: "Dipti",
   lastname: "Borkar",
   language: "English",
   time_zone: "PST",
   zip: 94403
  }
```

Key - UTF-8 string up to 250 bytes

Value - can be 0 bytes – 20 MB (best practice < 1 MB)





Appendix

PayPal

Query vs Search

	Query Engine	Search Engine (ES)
1.Input	Know exactly what you are looking for.	Exact value is not required for searching. Supports fuzzy, partial, proximity, etc., match.
2. Output	Returns only results that match.	Top-N ranked matches based on relevance scoring using tf-idf.
	Completeness and accuracy is guaranteed.	Do not need to retrieve all results. First few pages, OK.
3. Data Access	Predominately index-based access. Primary or secondary index based.	Query each shard in the index (Phase 1). Populate local priority queue (top-n results). Combine into global queue (global top-n). Fetch top-n documents (Phase 2).
4. Performance	Optimal performance / access path.	Search has much more work to do.
5. Service Time	Supports millions of executions per second. From sub-msec.	Takes longer than regular query. 10-20 msec - considered good. 100-200 msec - under heavy load. "Depending on the search complexity (term vs phrase vs proximity), it can be 10 to 20 times longer than simple term search."
6. Core Strengths	Predictable / Systematic Lookups.	Interactive (Human!) Search / Complex Investigation.

PayPal

Key-Val and Inverted Indexes

Data Table

Key	Value
K1	A1, B3, C4
K2	A1, C2, X0
K3	C2, X0, Y4
..	..

Index Table

Value	Key
A1	K1, K2
B2	K1
C2	K2, K3
C4	K1, K5
X0	K2, K3
Y4	K3
..	..

Inverted Key-Val index.

PayPal

Composite Key-Val Structures

Data Table

Acc#	Timed Key	Value
ABC	TK1	A1, B3, C4
ABC	TK2	A1, C2, X0
ABC	TK3	C2, X0, Y4
..	..	..

Index Table

Acc#	Value	Timed Key
ABC	A1	TK1, TK2
ABC	B2	TK1
ABC	C2	TK2, TK3
ABC	C4	TK1, TK5
ABC	X0	TK2, TK3
ABC	Y4	TK3
	..	..

Inverted Key-Val index with a leading edge.