
aws
SUMMIT
INDIA | MAY 25, 2023

ANA005

# 5 great ways to reimagine your data layer with Redis

Jitender Singh Gahlot
Principal Solutions Architect
Redis

# Agenda

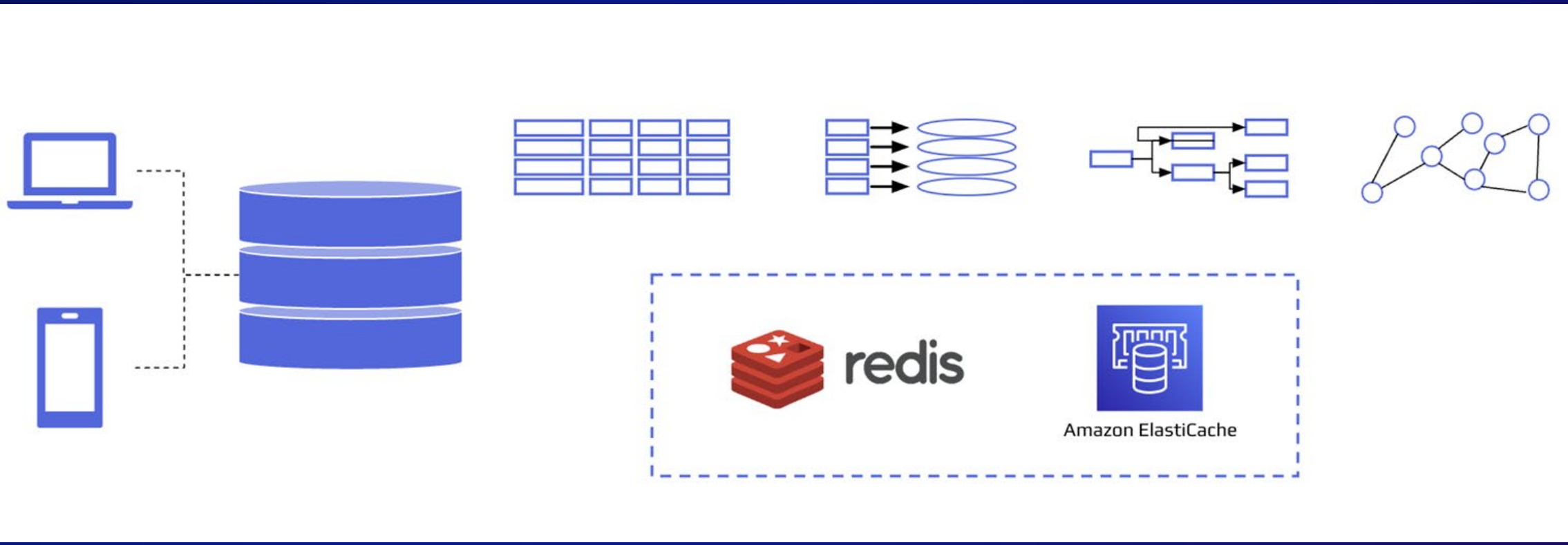
- What is Redis?
- What is the data layer?
- The limitations of today's data layer
- 5 ways Redis can help you reimagine your data layer to shatter these limitations

# What is Redis?

- In-memory datastore
- Flexible data models
- Open-source and enterprise
- Used and loved by millions
- Essential component of data layer

# What is the data layer?

The collection of technologies and services that process, deliver, and store data for applications.



# Who is responsible for the data layer?

Developers

DevOps

Architects

# What are their challenges?

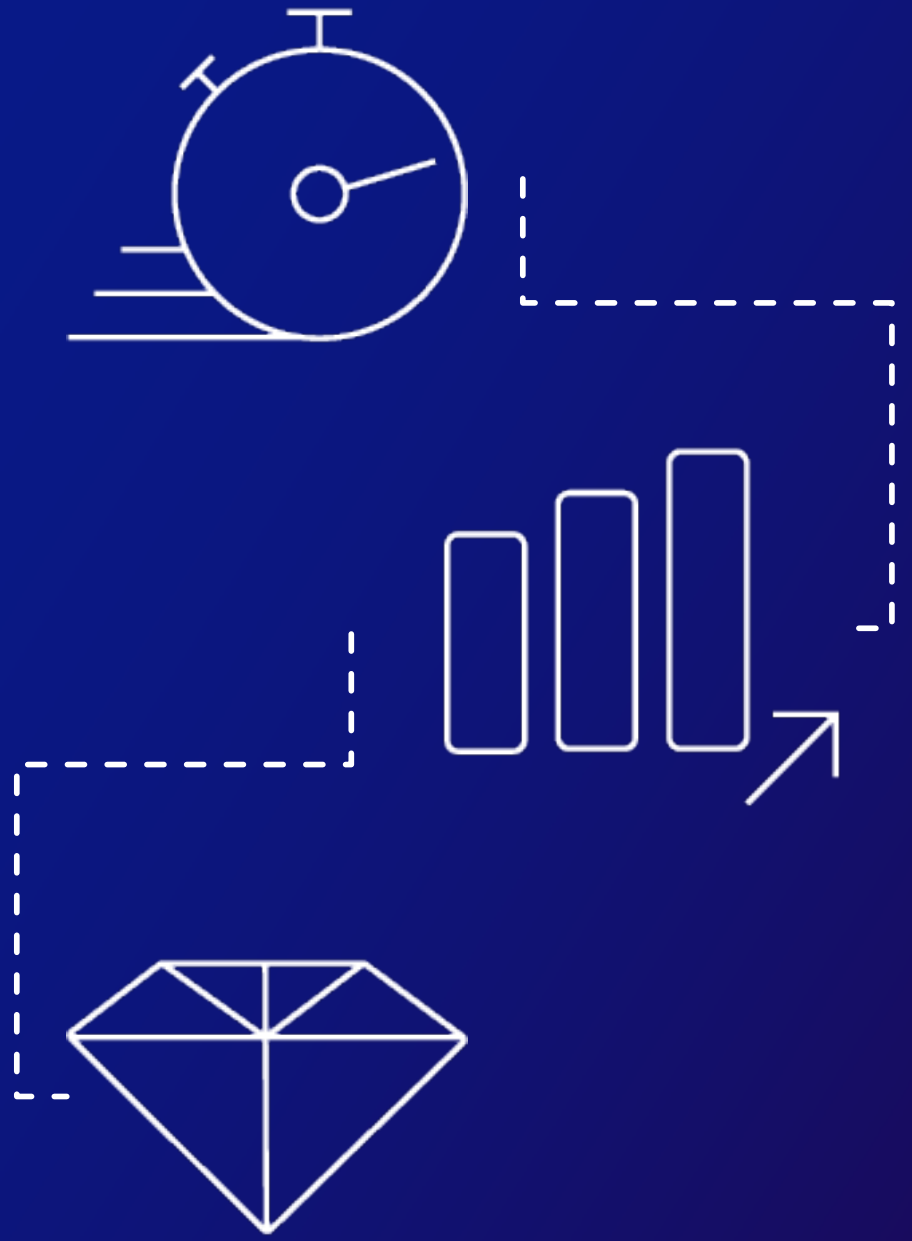
| Developers | DevOps | Architects |
| --- | --- | --- |
| Complexity impacts velocity | Reliability issues | Agility to react to business |
| Focus on coding, not infrastructure | Complexity breeds risk | Ensuring scalability |
| | Lack of expert support when issues arise | Keeping costs under control |

# Limitations of today's data layer

# Performance

Challenge to meet application needs for

- Speed
- Scale
- Resilience



# Complexity

- Services for each data type, use case, and deployment environment
- Added risk
- Difficult to manage

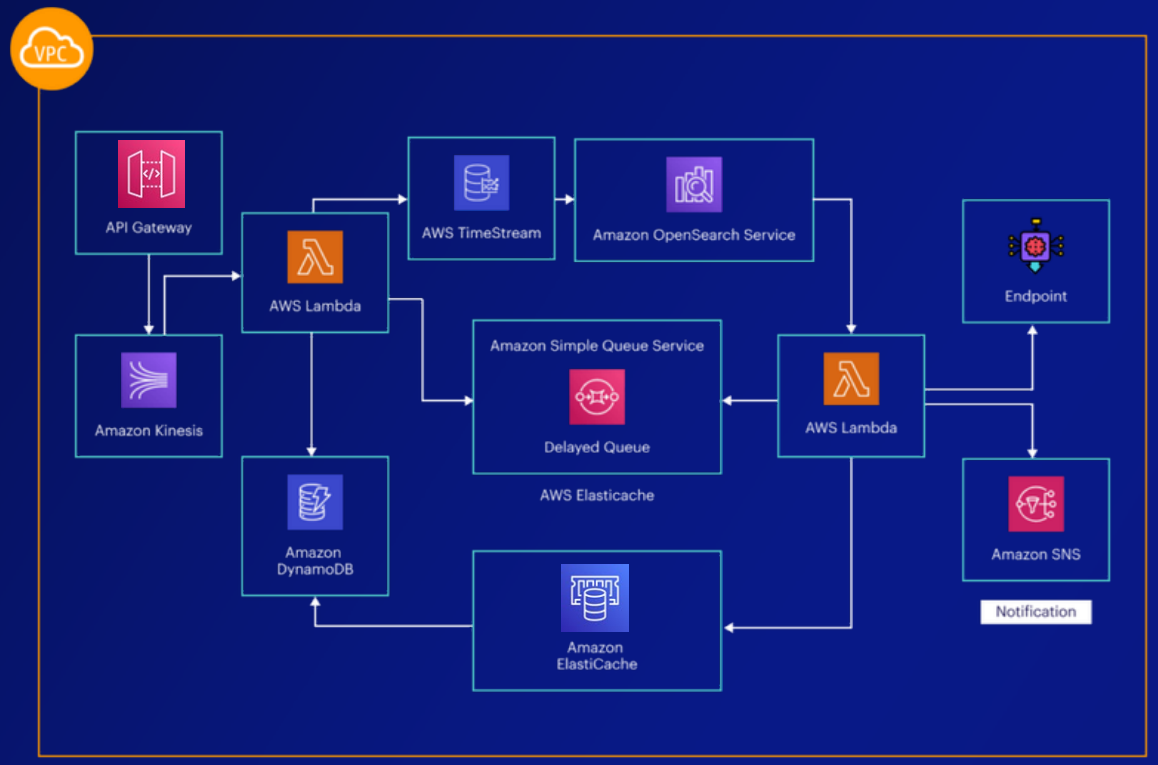
# Costs

- Optimized performance =/= optimized costs
- Data storage
- Spending on a multitude of services
- Can become complex and expensive to manage

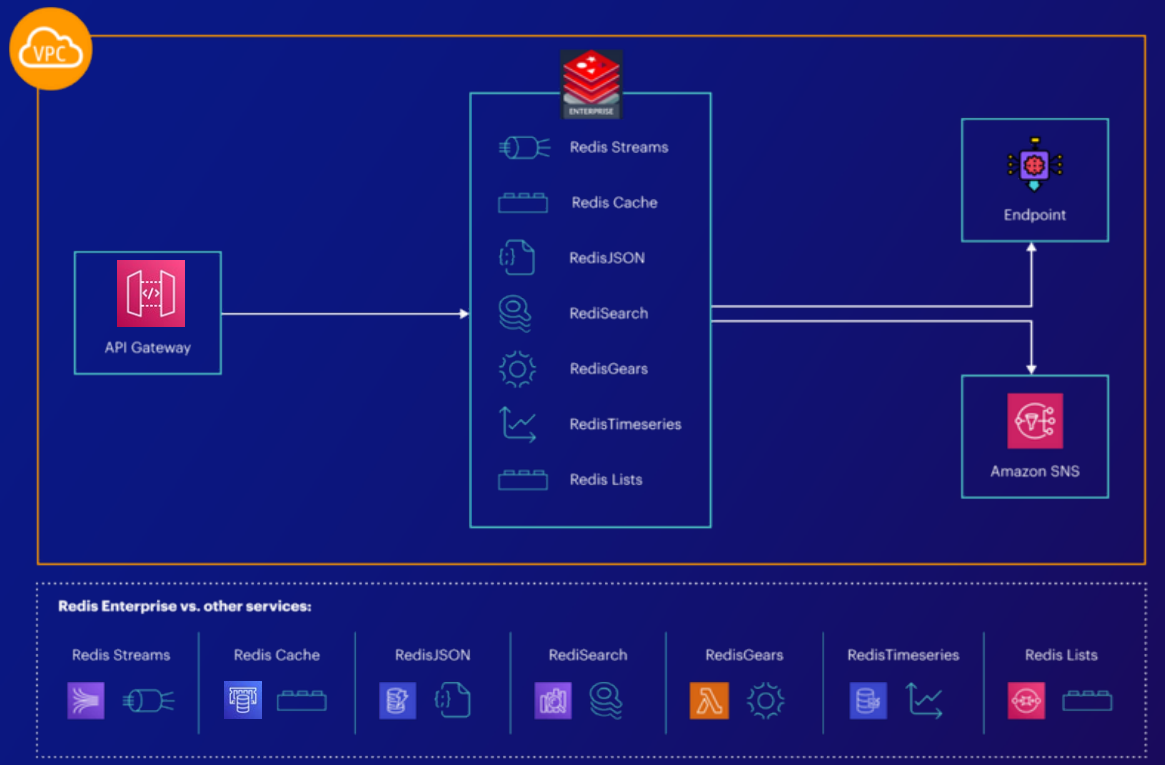
# 5 ways to overcome these limitations and reimagine your data layer with Redis

# 1. Pioneering a full real-time data platform

## Complex platform




## Simple platform



# 2. Unifying the cache and primary database

## Cache only

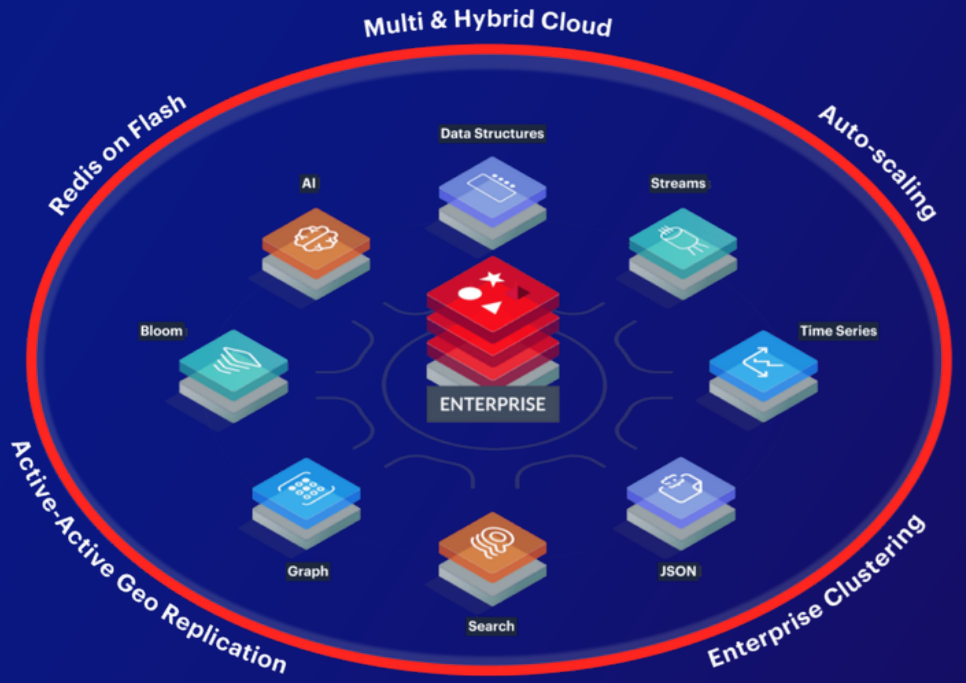
Amazon ElastiCache

Cloud X data store

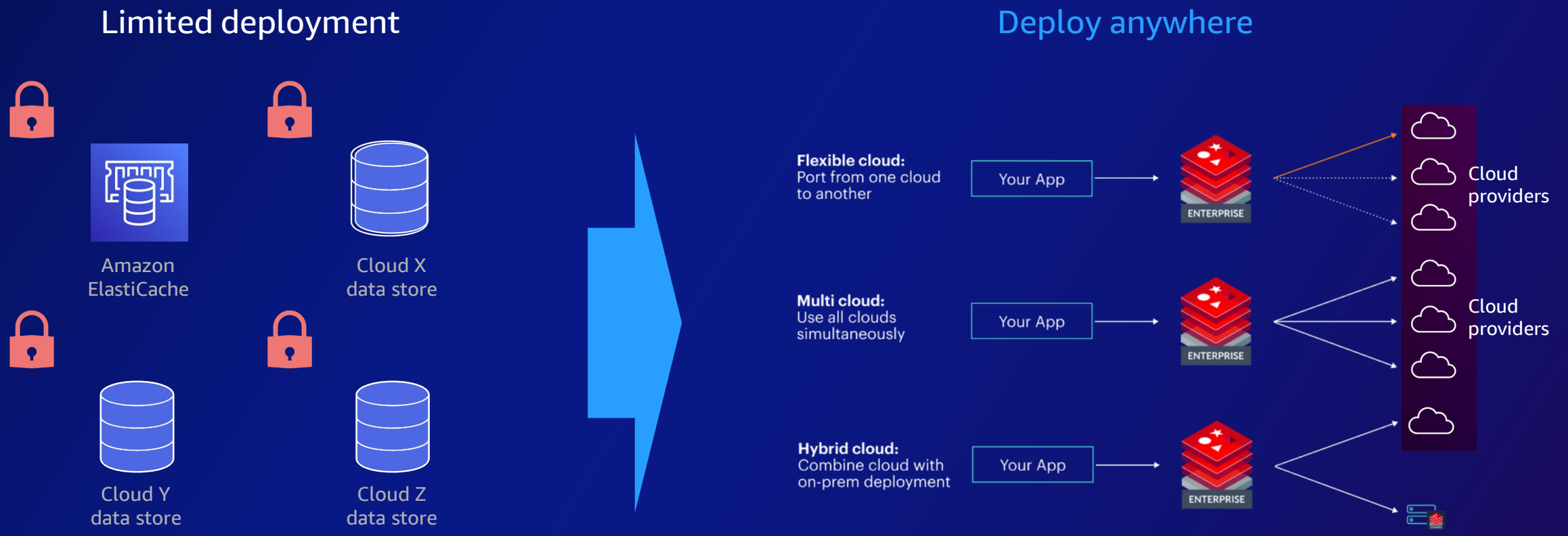
Cloud Y data store

Cloud Z data store

## Cache and primary database



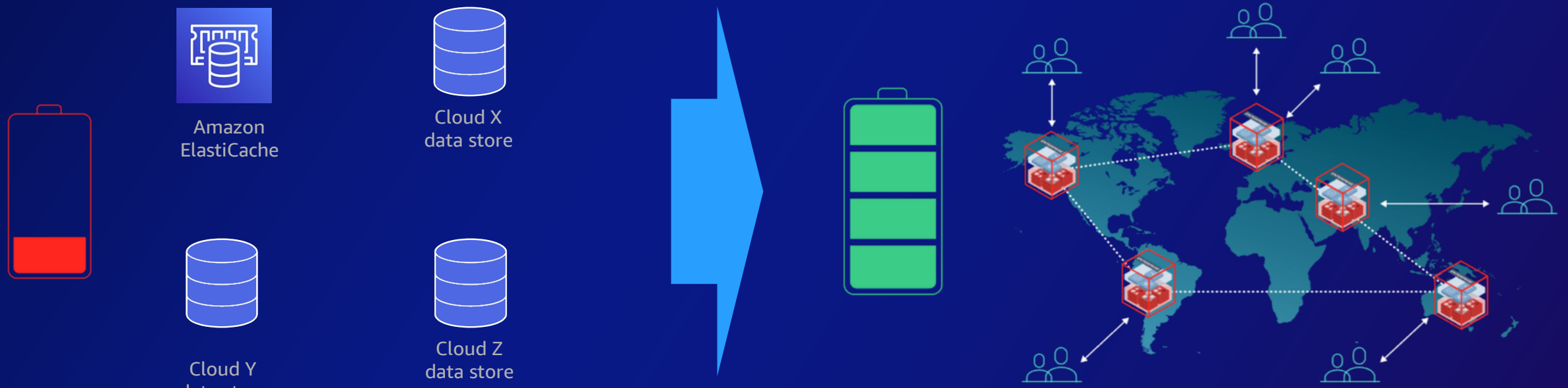
# 3. Providing flexibility to deploy anywhere



# 4. Delivering enterprise-grade performance

Complex platform

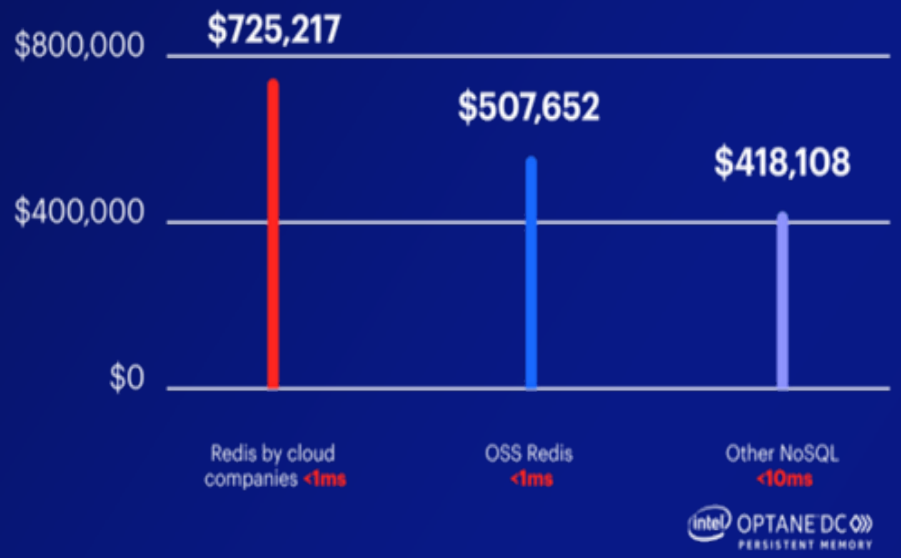
Simple platform



# 5. Eliminating performance/cost tradeoffs

## Expensive

- Tiering




## Cost-effective

- Improved CPU utilization
- Multi-tenancy
- Tiering

# Thank you!

Please complete the session survey

Jitender Singh Gahlot

Principal Solutions Architect
Redis