AWS SUMMIT INDIA

Redis data integration: Cost-effective performance at scale

Shekhar Suman,
Redis Multi-modeling Data Architect, Redis

Abhishek Srivastava,
Redis Data Integration Guru, Redis

The race for digital transformation and being relevant

Declining top-line growth

Financial firms are experiencing declining top-line growth due to low interest rates and new regulations

Changing expectations

Consumers and clients are demanding responsive, always-on, omni-channel experiences

Industry disruptors

The pandemic led to between a 21% and 26% increase in the relative rate of daily downloads of Fintech apps.

Availability drives revenue

E-commerce companies seeing revenue loss for each second their applications are down

Problems with legacy systems

Slow

Legacy systems are slow and unable to keep up with real-time needs.

Fragmented

Separate point solutions have resulted in fragmented data silos.

Not resilient

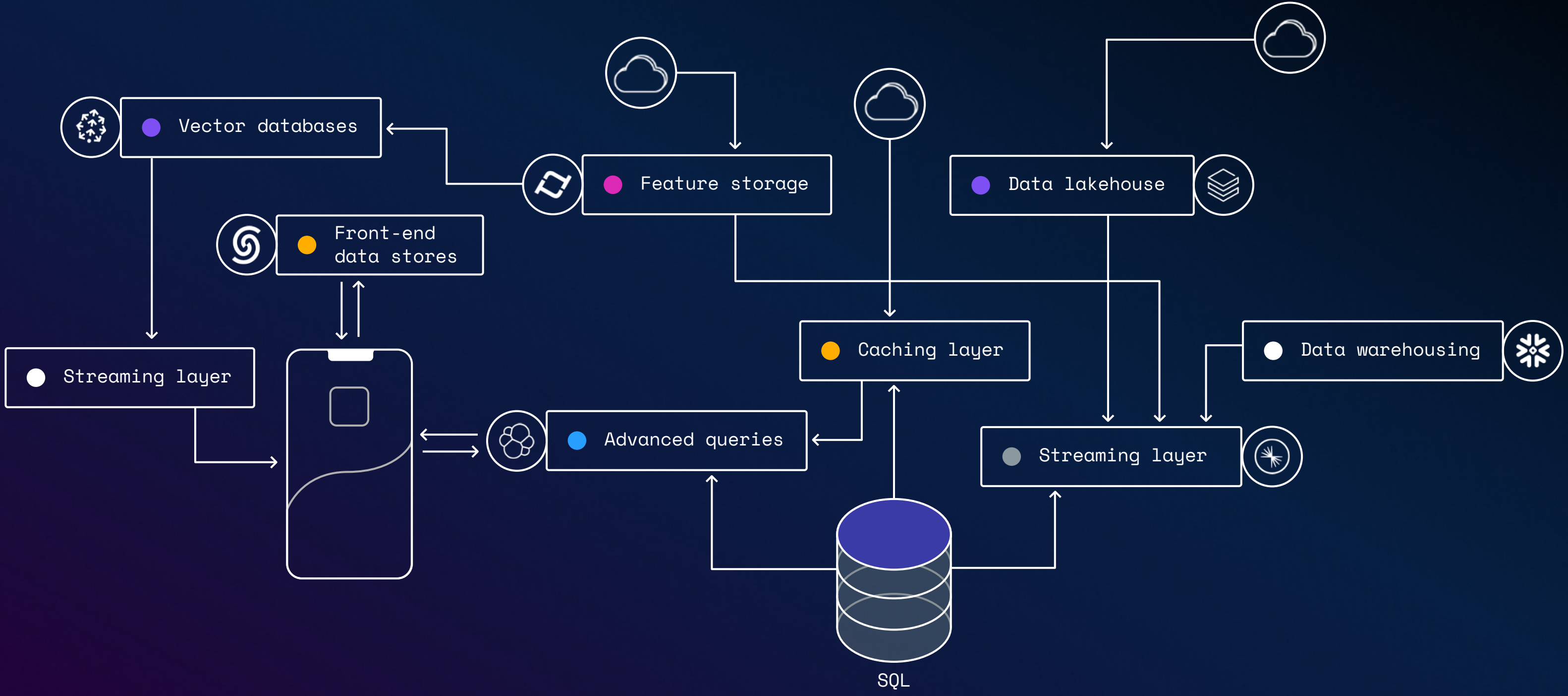
Business disruption and growing digital demands are resulting in systems failures.

Inefficient

Legacy systems hinder cost efficiency due to expensive technologies and talent shortage.

"Organisations need an efficient, cost-effective and proven technology thereby boosting their team's productivity and addressing most of their technical problems"

Applications are becoming more complex



The Problem Statement

- Have fleet of different traditional databases. Hence performance and scale are limited
- Business won't scale due to limitations of underlying databases
- Application is critical - application is real time - Need faster time to production
- Improve overall user experience with consistent performance
- Standards are high and market is tough - want to stay ahead in competition
- Need a technology that provides availability SLA guarantees & linearly scalable
- TCO and maintainability are important

How do we solve this problem?

“We solve this problem by bringing in a technology that not only can serve as a high-throughput database but can also simplify application architectures and enhance operational efficiency.

This technology should fit harmoniously with traditional databases to facilitate a seamless transition“

Let’s understand by building a true real-time app

A sample securities trading and account portfolio application

Building Blocks

Redis Data Integration*
Data Pipeline between Legacy systems and Redis

Ingest
Works as tool to ingest data from various sources to Redis

Write Behind
Works as a tool to write data from Redis down to various databases

Configuration Driven
No Code required, works completely in configurations provided in yaml files

Multi database Support
Most of the databases are supported out of the box and new features being added continuously

Core Components
Part of Redis enterprise

Streams
High throughput streaming system which supports persistence and consumer groups

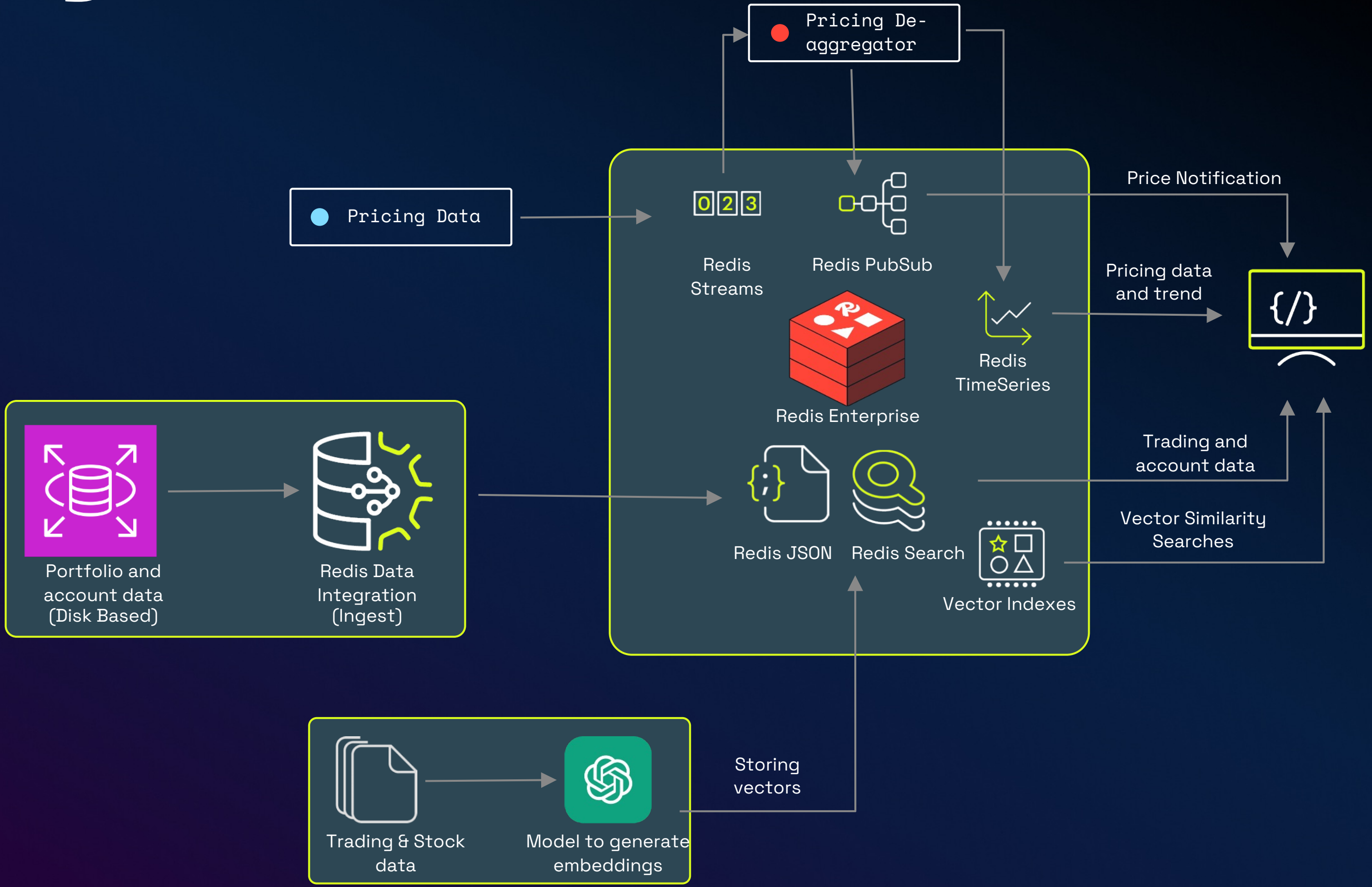
Search
Complex Queries on JSON data along with filters aggregations and Full text Search

Time Series
Fast timeseries storage with supports downsampling out of the box

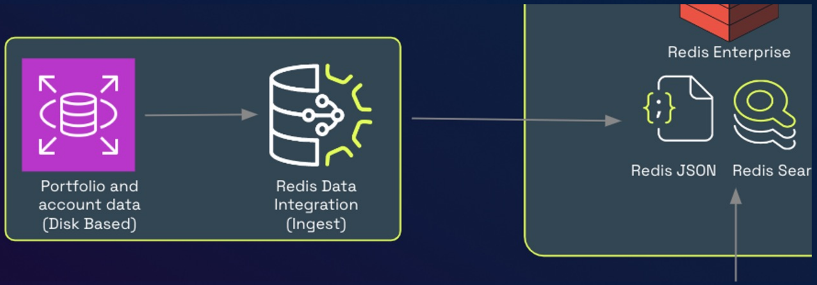
JSON
High performance NoSQL Document storage which can be scaled to very high throughputs

* RDI is currently in preview for on-premises installations and not available in the cloud.

Redesigned Architecture

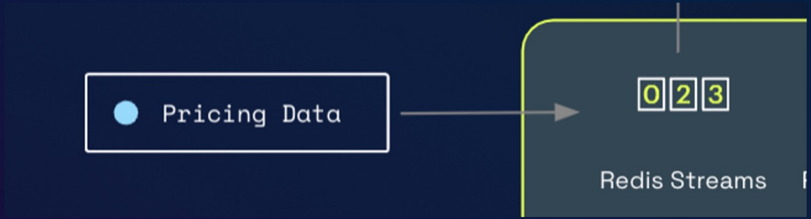


Ingest Account and security data


Portfolio and account data (Disk Based)
Redis Data Integration (Ingest)
Redis Enterprise
Redis JSON
Redis Sear

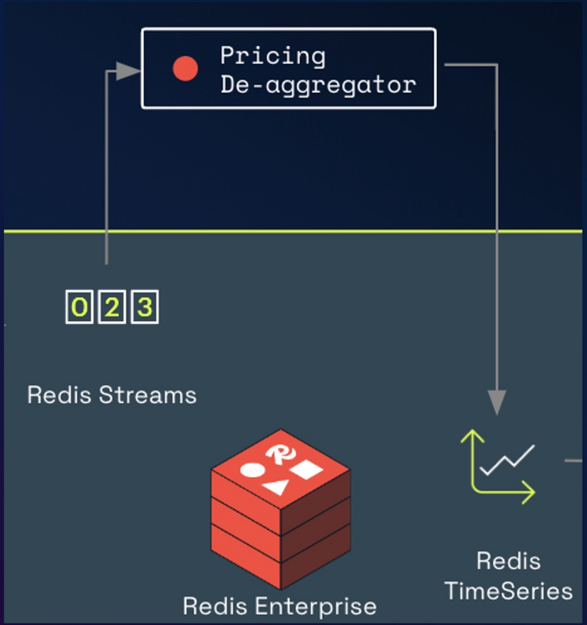
- MySQL (RDS) contains account and trading data
- Use RDI to ingest the data from MySQL to Redis Enterprise
- Data in Redis is denormalized as a JSON documents which will be indexed using Redis Search capability
- The user can query this data for analytical and reporting purposes

Generate pricing data



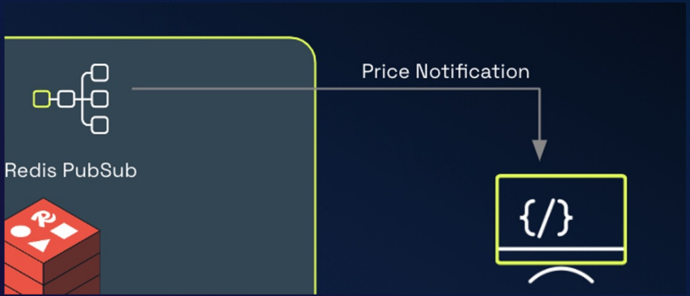
- High speed stock ticker data is ingested into Redis Streams topic in Redis Enterprise
- The consumer will process these ticker data asynchronously

Consume & deaggregate pricing data



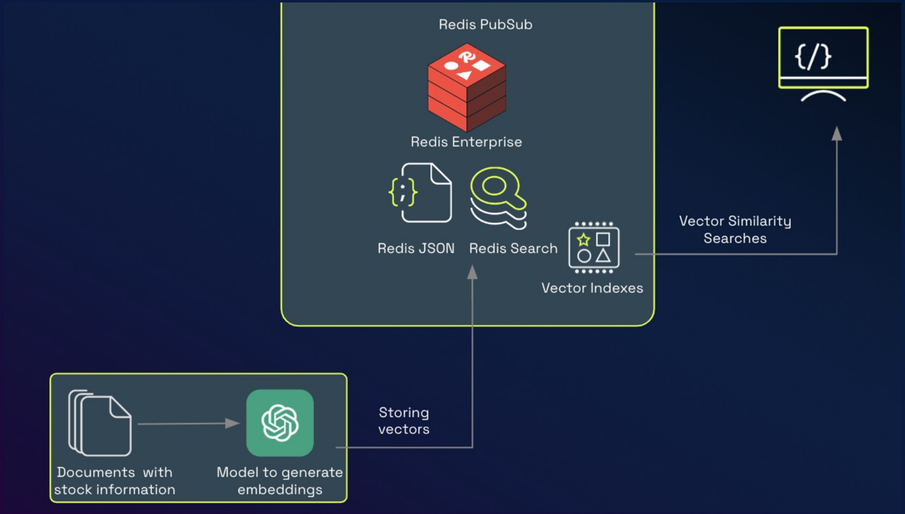
- Pricing De-aggregator acts as a consumer to the Redis Streams topic and processes these pricing data
- The pricing data is then converted into Timeseries data and can be down-sampled

Send pricing notifications



- The De-aggregated data is also sent to redis pubsub which can trigger a price notification to the user

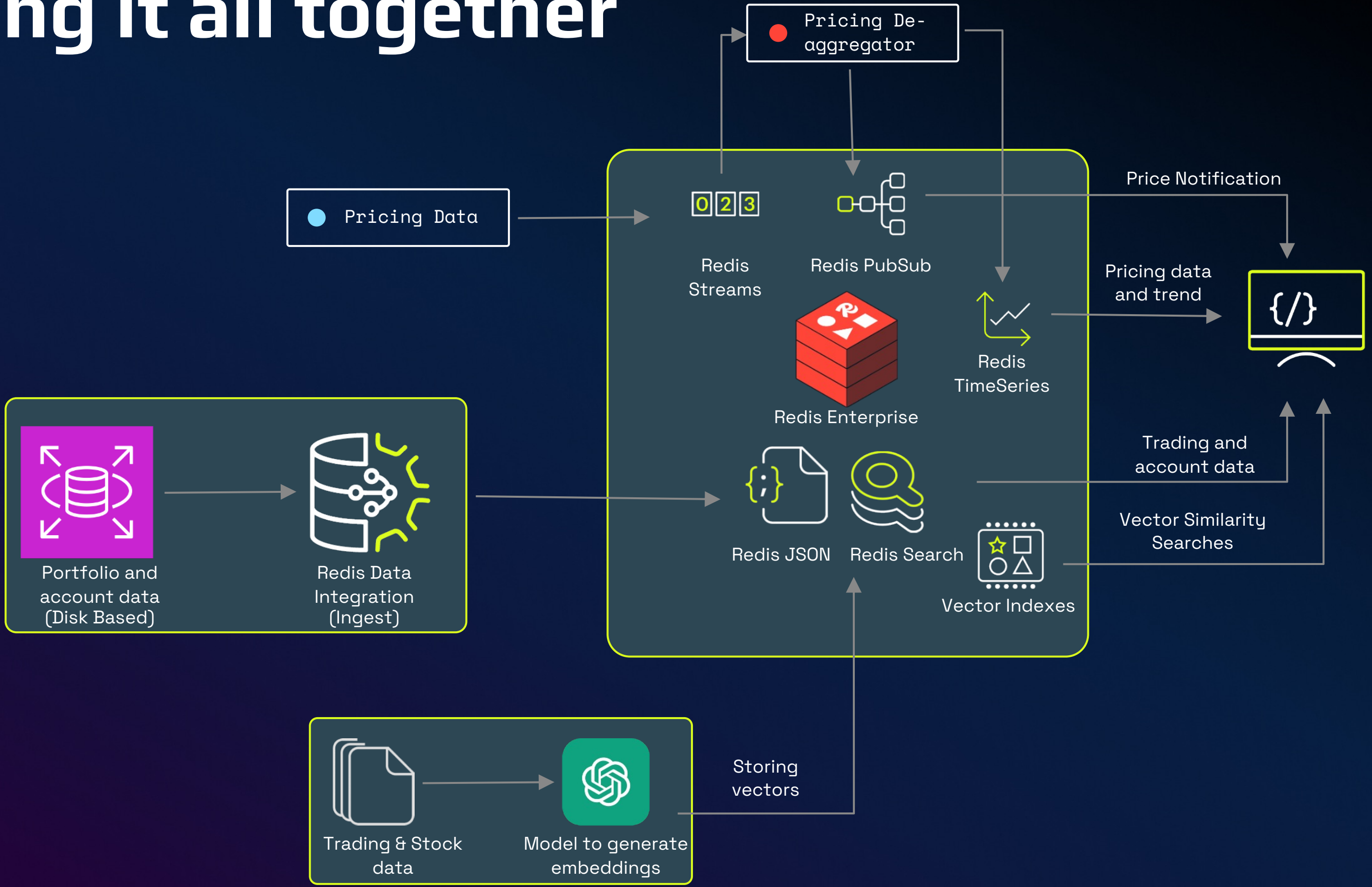
Perform semantic searches on related docs



Redis PubSub
Redis Enterprise
Redis JSON
Redis Search
Vector Indexes
Vector Similarity Searches
{/}
Storing vectors
Documents with stock information
Model to generate embeddings

- Documents are passed through a model to generate embeddings
- These embeddings are then stored in Redis
- Appropriate indexes are created on the vectors along with other data
- Vector Similarity searches are performed to do semantic and hybrid searches on the indexes

Putting it all together



Go beyond caching

Enterprise caching

Primary database

Session management

Messaging

Fraud detection

Real-time inventory

Leaderboards

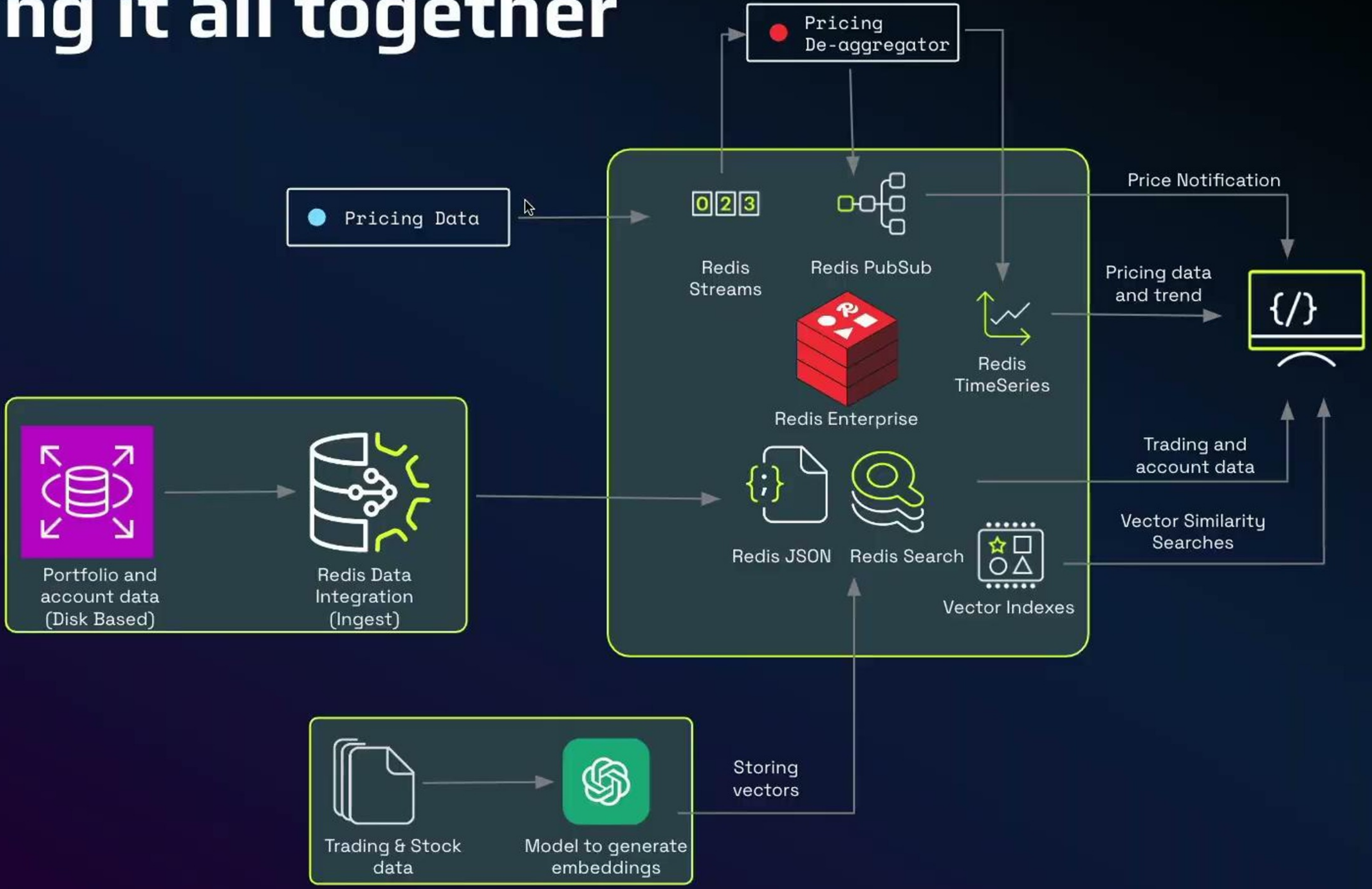
Vector database

Online feature store

Showcase

A sample securities trading and account portfolio application

Putting it all together



Thank you!

Please complete the session survey in the mobile app