
EDB
Postgres for the AI Generation
Welcome to the
EDB Postgres AI Day Zurich
Renaissance Zurich Tower Hotel
13th November 2024


EDB
Postgres for the AI Generation
EDB Postgres AI Day Zurich | 13 November 2024
NUTANIX
TRADEWARE
AGENDA
13:30 Registration & Reception with Snacks
13:45 Welcome to the EDB Postgres AI Day
14:00 The EDB Postgres Roadmap - Accelerate Your Data Transformation
14:45 Postgres Analytics - Shaping the Future of App Development
15:30 Coffeebreak
16:00 EDB Postgres AI - PG for the 21st Century
16:45 Empowering Application Modernization of Legacy Systems with NDB e2e Database Management
17:15 Closing & Prize Draw
17:20 Drinks & finger food

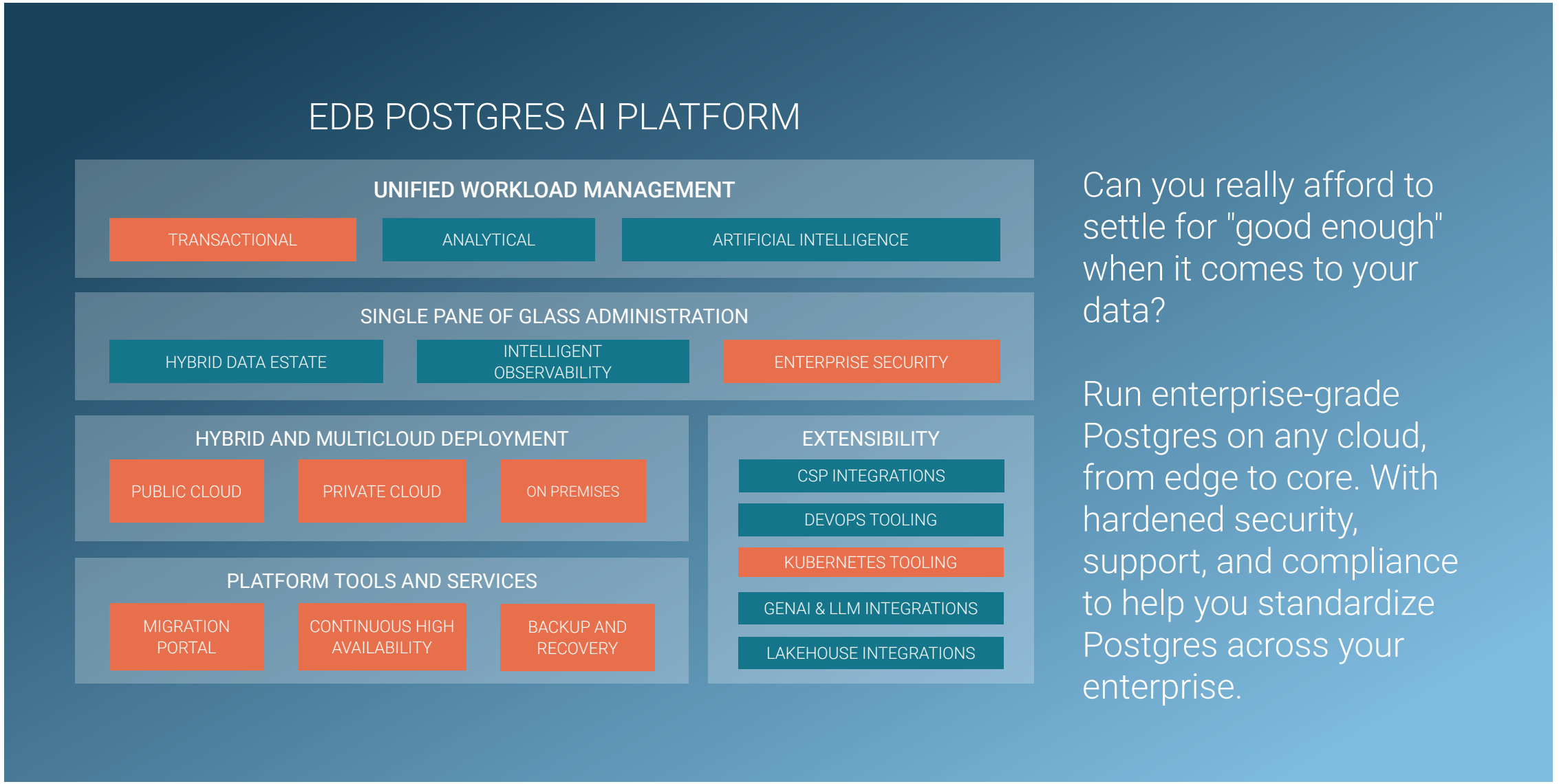

EDB
Postgres for the AI Generation
EDB Postgres AI Day Zurich | 13 November 2024
Swiss PostgreSQL User Group
"AI/ML-Driven Query Optimization Using Balsa and LEON"
Speaker: Stefan Keller
5th December 2024, 19:00 - 21:00 in Zürich
Registration: www.swisspug.org
PUG

The EDB Postgres Roadmap - Accelerate Your Data Transformation

Renaissance Zurich Tower Hotel
13th November 2024

Gianni Ciolli
VP, Practice Lead High Availability, EDB

Enterprise-grade Postgres

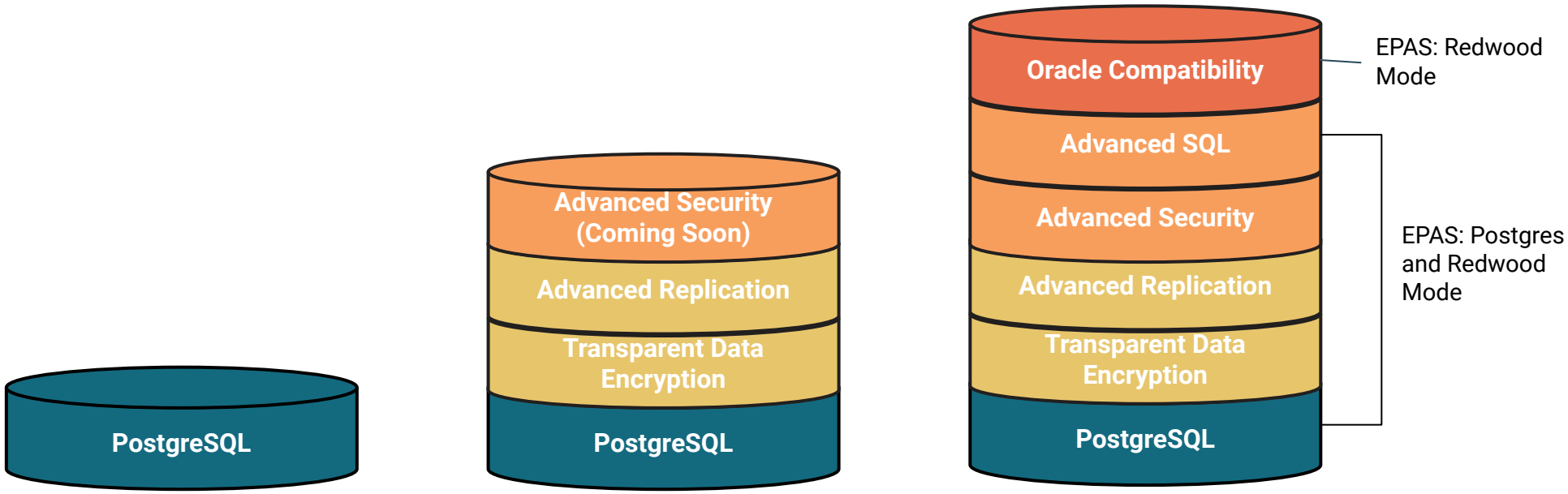


Can you really afford to settle for "good enough" when it comes to your data?

Run enterprise-grade Postgres on any cloud, from edge to core. With hardened security, support, and compliance to help you standardize Postgres across your enterprise.

©EDB 2024 — ALL RIGHTS RESERVED.

Reminder - EDB Postgres AI databases



Community 360

Standard EDB Postgres Extended (PGE) Server

Enterprise EDB Postgres Advanced Server (EPAS)

EDB contributions to PostgreSQL 17 (Released Last Month)

Backup and Recovery

Better Disaster Recovery with Incremental Backup

- Protects your database more efficiently, by only backing up what changes
- Enables more frequent backups of large databases
- Improves security by not requiring direct access to server files
- Protects against disasters and application errors

Developer Productivity/Flexible PG

Finished SQL:2023 SQL/JSON

- Support for latest SQL/JSON Standards

JSON_TABLE

- Easily work with JSON data using a table like interface

Performance Enhancements

Reduced Memory for Part-wise JOINs

- Efficiently join large tables using less memory

NULL Constraint Improvements

- Better execution plans with NULL constraint handling

Business Logic and Replication

Helping EDB customers with Complex Business Logic

- Simplify complex logic with improved subtraction support

Convert Physical Replica to Logical Replica

- Robust way to initialize logical replication for large datasets with pg_create_subscriber

Database Extensions and Tools

Q3 2024 (Tools and Extensions)

Better support for UPDATE heavy workloads by removing bloat

- Support for pg_squeeze

Seamlessly Convert Postgres Databases into a RESTful API

- Support for PostgREST

Q4 2024 (Tools and Extensions)

EDB Query Advisor extended to include extended table statistics

- EDB Query Advisor goes beyond indexes and provide actionable recommendation for extended table statistics for more accurate query plans.

Reducing rewrites when transitioning from Oracle to Postgres

- Support for PGTT (Oracle-style Global Temporary Tables)

EDB Postgres Advanced Server (EPAS) 17

Merge with Upstream PostgreSQL

Community Enhancements

- Incremental backup
- SQL/JSON orthogonality
- JSON_TABLE
- Reduced memory for partition wise JOINs
- ALTER TABLE to mod. exper of gen columns
- Subtransaction SLRU enhancements
- Logical replication improvements
- Partitioning enhancements (MERGE PARTITION, SPLIT PARTITION)
- Identity columns
- ALTER SYSTEM improvements,
- New pg_maintain role

and more…

Simplify Migrations for Legacy Oracle Applications

XMLType Support

- Enable support for Oracle XML packages and additional XML-related features, reducing rewrites for legacy applications

BFILE Support

- Allows for the migration of business logic in the database for binary files, such as graphics, documents, and multimedia, with fewer rewrites, ensuring a smooth transition from Oracle to EPAS.

Oracle Compatibility Enhancements

Oracle Compatibility enhancements

- NLS_UPPER, NLS_LOWER, NLS_INITCAP, SAVE EXCEPTION FOR ALL, FOR ALL MERGE

Customer Requests and Miscellaneous

- JSON Support for EDB Audit Log
- Directory Permissions
- EDB Loader Improvements
- OCI Import Foreign Schema
- Conceal Password

Software Bill of Materials (SBOM)

Q4 2024

What is it?

- EDB will provide SBOM reports directly in its software repositories.
- These comprehensive reports offer detailed inventory of components and dependencies that comprise a software packages.

What are the customer benefits?

- Helps address the challenge of identifying and mitigating security vulnerabilities in Postgres deployments
- Reduces the effort to track dependencies, licensing, and vulnerabilities

Automate compliance standards with Postgres from EDB

Q4 2024

What is it?

- **TPA** (Trusted Postgres Architect) will provide **STIG** and **CIS** options that can be specified when configuring a Postgres cluster.
- Choosing one of these options:
  - Adds relevant Postgres config options to your cluster
  - Enforces other aspects of the relevant standard

What are the customer benefits?

- Industry security standard controls are provided in **large docs** (100+ pages)
  - **Labor-intensive** and **error-prone** process to implement
  - Increased risk of misconfiguration and incomplete implementation
- Choosing one of these TPA options:
  - Reduces misconfiguration **risk**
  - No longer requires manual changes
  - Saves time
  - Enables new features in TPA

2025 and Beyond

Security

Data Masking (EDB Tools)

- Protect sensitive information in lower IT environments (Test & Dev) for internal ops and safe sharing with trusted 3rd parties
- EDB Data Masking will provide an intuitive way to protect Personally Identifiable Information (PII) - such as names, social security numbers, credit card details, mailing addresses — with masking policies on specific columns
- Additional capability to preserve masking policy when cloning the entire schema to another Postgres instance

Scalability

Global Indexes for Partitions (PG+)

- Large partitions counts in Postgres are often slower than regular tables when applications require filtering on the non partition key.
- Often table partitioning is used not only to improve performance, but implement business logic and data retention policies.
- With global indexes, queries that don't filter on the partition key will be better performant and data structures will not need to be changed.

Flexible Postgres

Native Graph capabilities

- SQL standard "Property Graph Queries (SQL/PGQ)" will be in Postgres core, allowing data in Postgres tables to be queried as if it were a graph database
- This eliminates the need to bring in specialty databases and just use Postgres when needing to query complex relationship between data and entities

Q & A

Thank you

EDB Postgres AI: Accelerate Data Innovation with Analytics & Artificial Intelligence

Renaissance Zurich Tower Hotel
13th November 2024

Lucie Zeng - Associate Data Scientist EMEA, EDB
lucie.zeng@enterprisedb.com

Franck Sidi - Senior Director EMEA & WW Field CTO Leader, EDB
franck.sidi@enterprisedb.com

Agenda

Agenda

- Who is EDB?
- EDB for Analytics & AI
  - PG Analytics Accelerator
    - Demos
  - AIDB
    - Demos
- Conclusion
- Q&A

Who is EDB?

1600+ Enterprises and Growing

EDB deeply understands Enterprise Postgres needs.

79 Countries around the World

Global footprint and employee base.

Millions of people using Postgres in the world

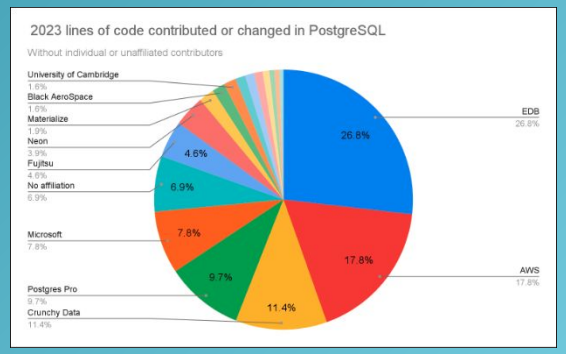
Long-term customers and deep Postgres capabilities.

3 of 7 Postgres Core Team Members, 7 Committers, 40+ Contributors

EDB is the leading Postgres community contributor.

27% of Postgres Code Contributed in 2023

Driving the innovation and foundation of Postgres.



>300 Dedicated Postgres engineers

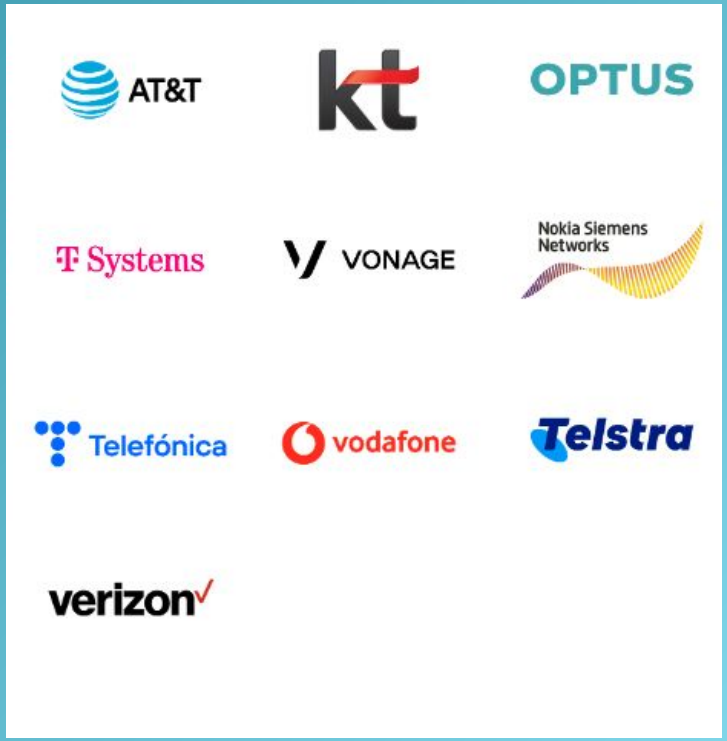
Unparalleled expertise in Postgres.

LEADING ENTERPRISES TRUST EDB

BANKING FINANCIAL

TECHNOLOGY

TELCO



Part I: Postgres Analytics - Shaping the Future of App Development

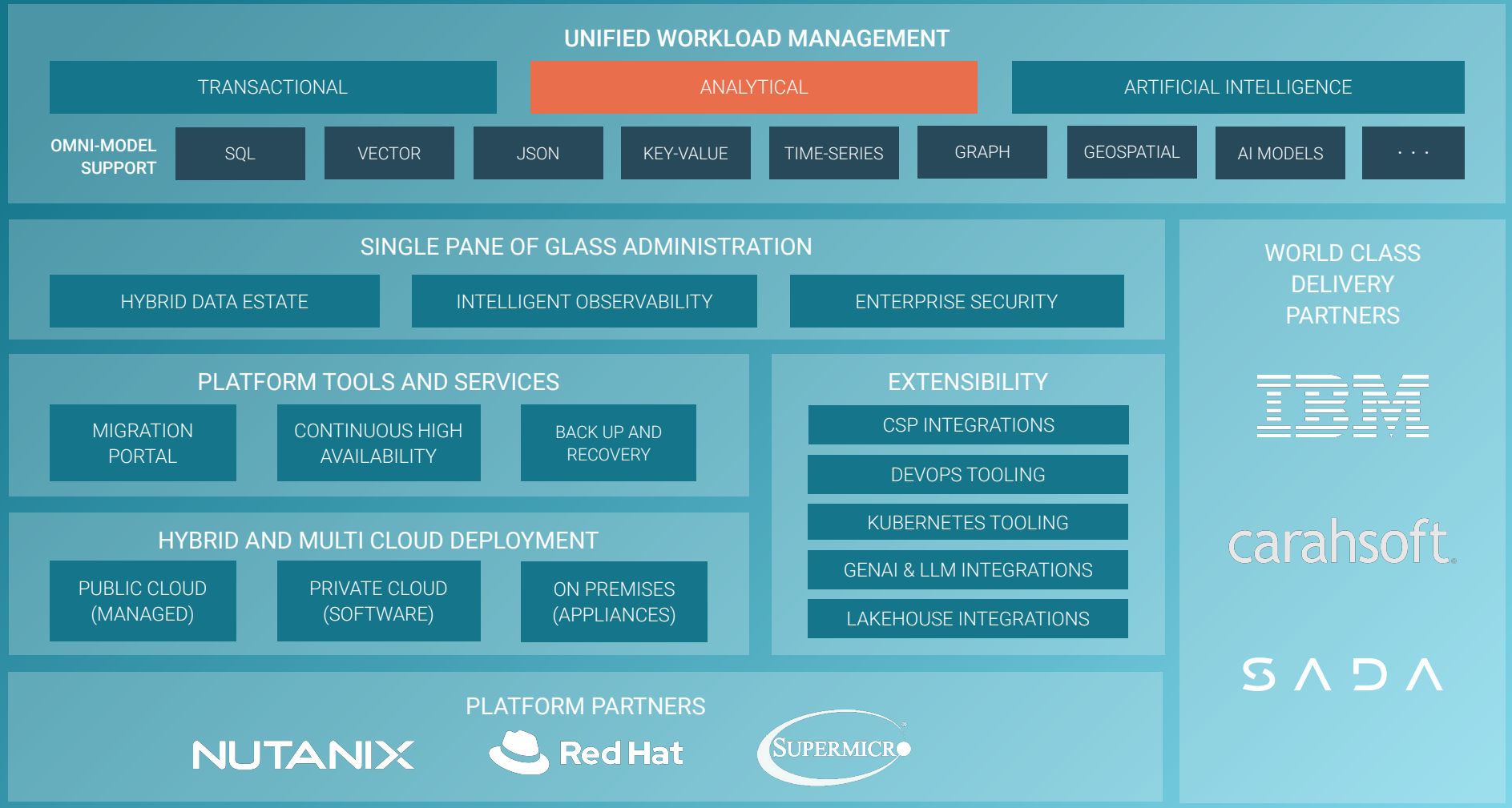
Renaissance Zurich Tower Hotel
13th November 2024

Franck Sidi - Senior Director EMEA & WW Field CTO Leader
franck.sidi@enterprisedb.com

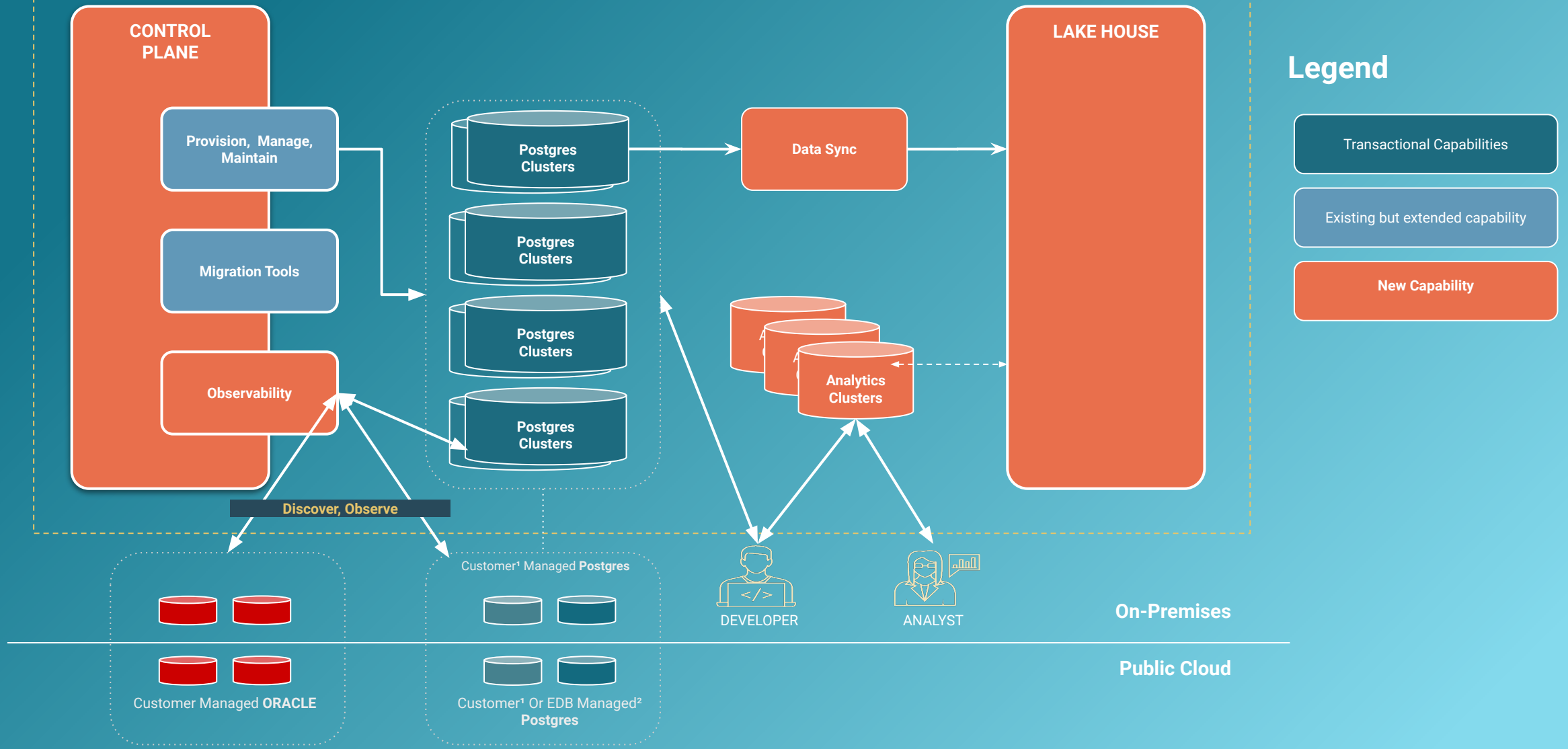
PG Analytics Accelerator

One platform for Transactional,
one platform for Analytical,
and one platform for AI.

The EDB Postgres AI Platform



EDB POSTGRES AI



Legend

- Transactional Capabilities
- Existing but extended capability
- New Capability

Key Business Challenges Addressed by PGAA

High-Performance Analytics: Extremely Fast Queries

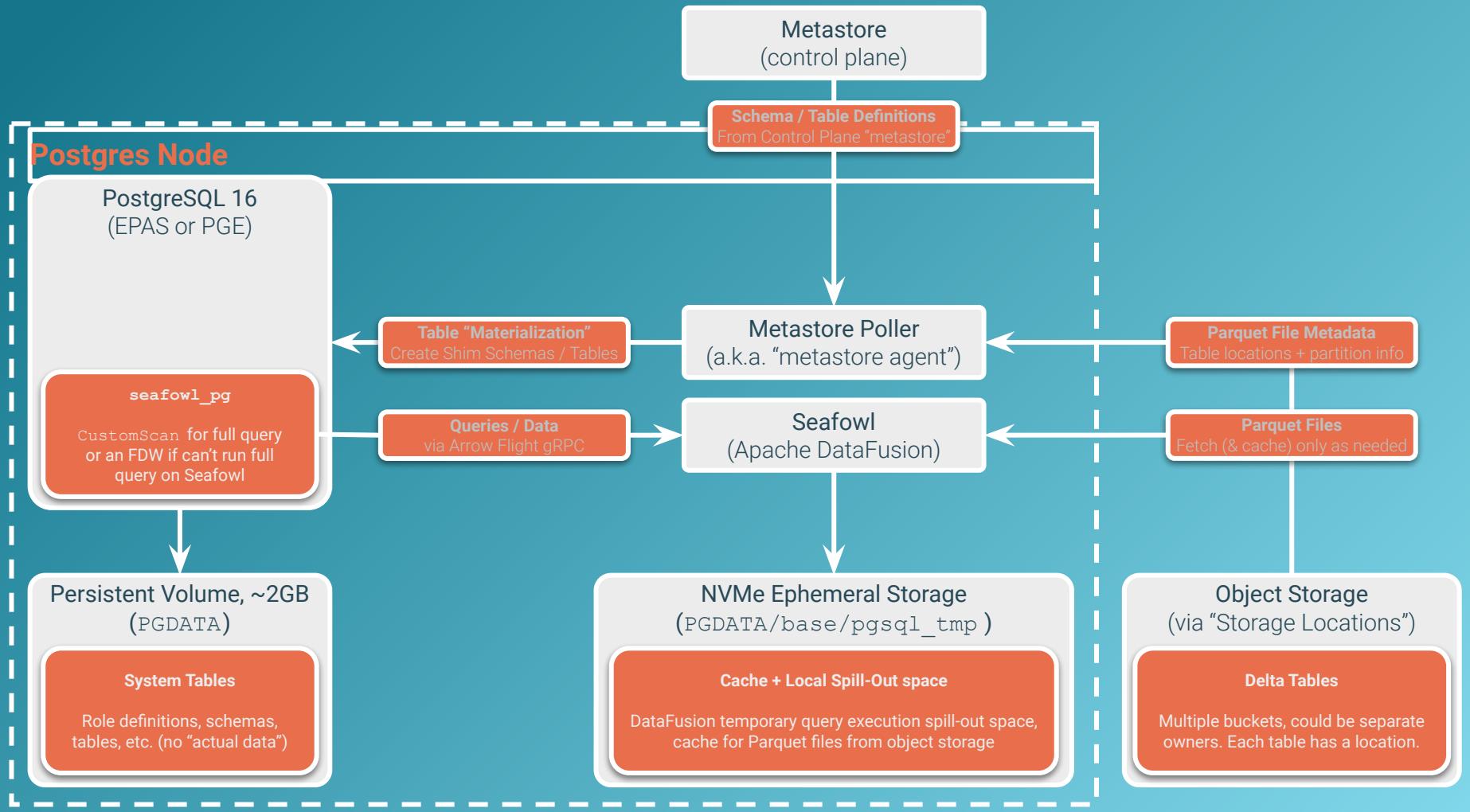
Cost-Effective Solution: Lower Ownerships Costs & Ease of Use

Real-Time Data Insights: Efficient Data Processing & Open Data Formats

Simplified Data Management: Remove complexity of Data Tuning, No Index..

PGAA — Postgres Analytics Accelerator

PGAA is a Postgres extension with a Vectorized Query Engine for querying Lakehouse Tables in external storage locations



PGAA — Postgres Analytics Accelerator

PGAA is a Postgres extension for rapid analytics:

- PGAA **separates storage from compute**, and includes support for S3, MinIO or the local filesystem:

```
SELECT pgaa.create_storage_location('sample-data', 's3://pgaa-sample-data-eu-west-1');
```

- PGAA makes **Lakehouse Tables (Delta Lake)** queryable just like any Postgres table:

```
CREATE TABLE public.customer () USING PGAA
      WITH (pgaa.storage_location = 'sample-data', pgaa.path = 'tpch_sf_1/customer');
```

- PGAA has a **Vectorized Query Engine** for 30x faster queries compared to transactional Postgres:

```
SELECT count(*) AS num_customers from customer;
```

It's a Postgres frontend to the external Lakehouse ecosystem.
But when combined with PGD, it becomes a real-time analytics solution...

PGD + PGAA — Real-time Analytics for Postgres

PGD now ships with an embedded analytics engine, powered by PGAA:

- Replicate transactional tables **directly to Lakehouse Tables**:

```
CREATE TABLE x(a INT PRIMARY KEY) WITH (pgd.replicate_to_analytics = true);
```

- Optionally, **query tables with the analytical engine**:

```
SET LOCAL bdr.prefer_analytics_engine=true;
SELECT * FROM x;
```

- Create **Tiered Tables** that offload storage and processing of “cold” data to analytics:

```
CREATE TABLE test_part_timestamp (a timestamp, b int ) PARTITION BY RANGE (a);
SELECT bdr.autopartition(
        relation := 'test_part_timestamp',
        partition_increment := '1 month',
        partition_initial_lowerbound := CURRENT_TIMESTAMP::text,
        managed_locally := true,
        analytics_offload_period := '3 months');
```

Tiered Tables — Control Storage Costs & Protect Transactions

Transparently offload storage and processing of “cold” data to a separate analytical system & reduce table size on disk

Work with a **Single Version** of all Data

EDB Postgres Distributed (PGD)

Transactional Engine

Insert | Update | Delete | Select

Partitioned Table

Local Partitions

Next Month | Current Month | Last Month | 2 months ago | 3 months ago

Offload Cold Data

Foreign Partitions

4 months ago | 5 months ago | 6 months ago | ...

Replicate Hot Data To Parquet

Query Cold Data

Vectorized Engine

Select

APACHE DATAFUSION

Object Storage

Next Month | Current Month | Last Month | 2 months ago | 3 months ago | 4 months ago | 5 months ago | 6 months ago | ...

```
Partition Increment: 1 month
Offload Period:      3 months
```

Table Objects in PG Analytics Accelerator

		Source of Truth	*Which Data is in Postgres?*	*Which Data is in Storage Location?*	*Query with Analytical Engine?*
Transactional Table	Read/write	Postgres	All	None	No
Just a normal Postgres table	`CREATE TABLE cities ( city_id bigserial NOT NULL PRIMARY KEY, name text NOT NULL, population bigint);`				
External Lakehouse Table	Read-only	External	None	All	Yes

Delta Table outside of Postgres in a Storage Location (S3/FS)	`CREATE TABLE tpch_sf_1000_lfs.lineitem ()`
	`  USING pgaa WITH (pgaa.storage_location = 'local-fs-1000', pgaa.path = 'lineitem');`
	`SELECT COUNT(*) FROM tpch_sf_1000_lfs.lineitem;`				

HTAP Table	Read/write	Local	All	Replicated	Yes, with lag

Postgres table that is replicated to a Delta Table in a Storage Location	`CREATE TABLE cities ( city_id bigserial NOT NULL PRIMARY KEY, name text NOT NULL, population bigint)`
	`  WITH (pgd.replicate_to_analytics = true, pgfs.server = 'my_ai_lakehouse', pgfs.path = 'parquet/table1/');`
	
	`SET LOCAL bdr.prefer_analytics_engine=true;`
	`SELECT COUNT(*) FROM cities;`				

Tiered Table	Read/write	Local	Hot Partitions	Cold Partitions	Yes, via partitions

Partitioned table with "hot" data in local Postgres and "cold" data offloaded to a Storage Location	`CREATE TABLE test_part_timestamp (a timestamp, b int ) PARTITION BY RANGE (a);`
	`SELECT bdr.autopartition(relation := 'test_part_timestamp', managed_locally := true,`
	`  partition_increment := '1 month',`
	`  partition_initial_lowerbound := CURRENT_TIMESTAMP::text,`
	`  analytics_offload_period := '3 months');`				

Volume Table	Read-only	External	None	All	No
Special table for listing the contents of a storage location (S3 or FS)	`CREATE FOREIGN TABLE my_pdf_volume () SERVER my_ai_lakehouse` `  OPTIONS (path 'pdf/', mime_type 'application/pdf');`				

How to Work with PGAA

1. Add EDB Extension
2. Define Storage Location using S3 or Local Storage
3. Migrate Data to Delta Table Format
4. Create Table
5. Run Queries

No Index, No Tuning

```
postgres=# \dx
                    List of installed extensions
  Name   | Version |   Schema   |                Description
---------+---------+------------+---------------------------------------------
 pgaa    | 0.1.0   | pgaa       | pgaa extension: (c) 2023 EnterpriseDB Corporation
 plpgsql | 1.0     | pg_catalog | PL/pgSQL procedural language
(2 rows)
```

```
SELECT pgaa.create_storage_location('local-fs-100', 'file:///mnt/raid/parquet/100', '{}', NULL);
SELECT pgaa.create_storage_location('local-fs-1000', 'file:///mnt/raid/parquet/1000', '{}', NULL);
```

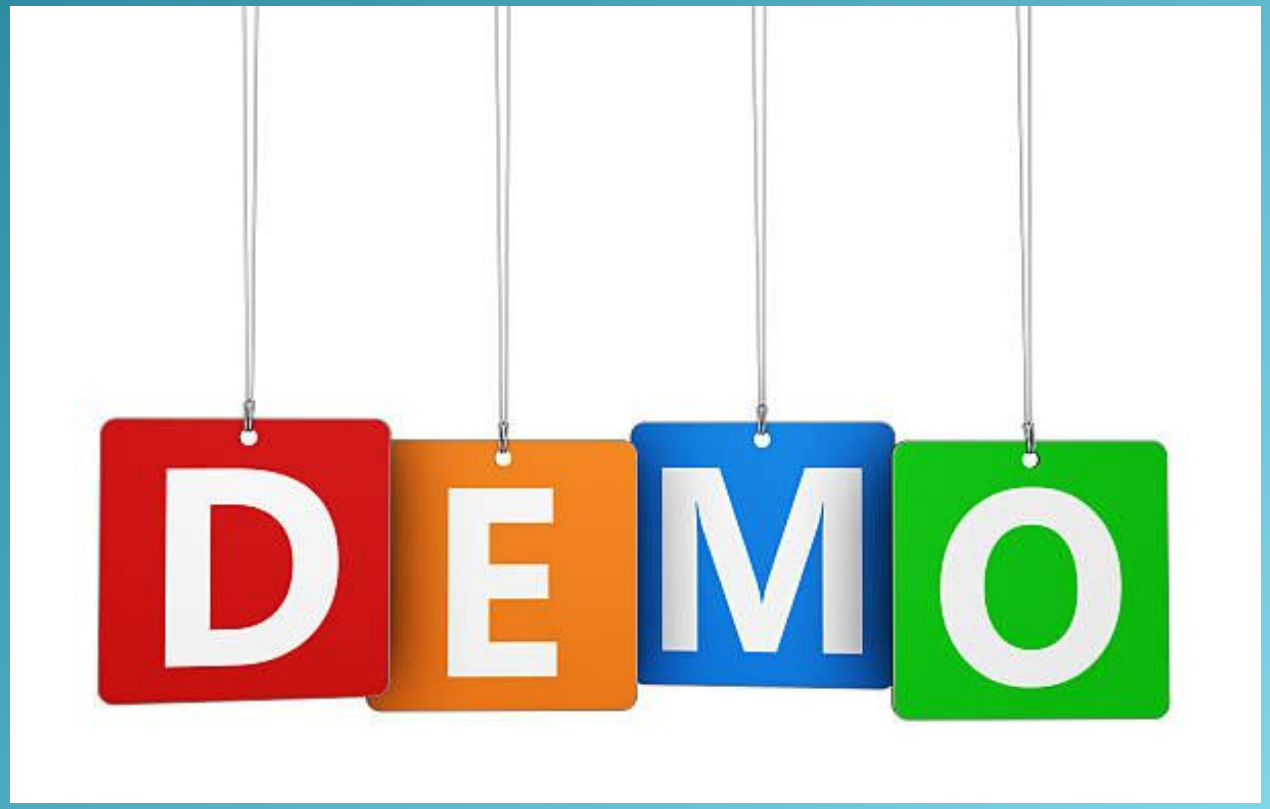
```
./lakehouse-loader pg-to-delta postgres://postgres@localhost:5432/postgres?sslmode=disable -q
"SELECT L_ORDERKEY, L_PARTKEY, L_SUPPKEY, L_LINENUMBER, L_QUANTITY, L_EXTENDEDPRICE,
L_DISCOUNT, L_TAX, L_RETURNFLAG::TEXT AS L_RETURNFLAG, L_LINESTATUS::TEXT AS
L_LINESTATUS, L_SHIPDATE, L_COMMITDATE, L_RECEIPTDATE, L_SHIPINSTRUCT::TEXT AS
L_SHIPINSTRUCT, L_SHIPMODE::TEXT AS L_SHIPMODE, L_COMMENT::TEXT AS L_COMMENT from
tpch_sf_1000.lineitem" file:///mnt/raid/parquet/1000/lineitem
```

```
create table tpch_sf_1000_lfs.lineitem () using PGAA WITH
(pgaa.storage_location = 'local-fs-1000', pgaa.path =
'lineitem');
```

```
postgres=# \timing
Timing is on.
postgres=# select count(*) from tpch_sf_1000_lfs.lineitem;
   count
------------
 5999989709
(1 row)

Time: 121.911 ms
postgres=#
```

Explore a 6 Billion Rows Dataset
in a matter of a few seconds with
PG Analytics Accelerator



Projects Estate

Jane Smith
ACME

Single Pane of Glass

Say farewell to data juggling and welcome clarity. Experience seamless data management through a single pane of glass, where insights and metrics are effortlessly within reach.

Learn More

Estate

View Estate

EDB Postgres AI Clusters

13

5 Healthy
2 Need Attention
3 Deleted

5/10

Create New

Self Managed Postgres

9

5 Healthy
2 Need Attention
3 Deleted

5/10

Configure Agent

Cloud Hosted Databases

6

5 Healthy
2 Need Attention
3 Deleted

5/10

Manage Access

Non Postgres Databases

100

5 Healthy
2 Need Attention
3 Deleted

5/10

Configure Agent

Projects

View All Projects (5)

Most Recent

Silly Squirrel — Sales, Approved
49 Clusters • 13 Users

Hopeful Elephant — Sales, Approved
24 Clusters • 8 Users

Gentle Mongoose — Sales, Approved
32 Clusters • 10 Users

Create New Project

Users

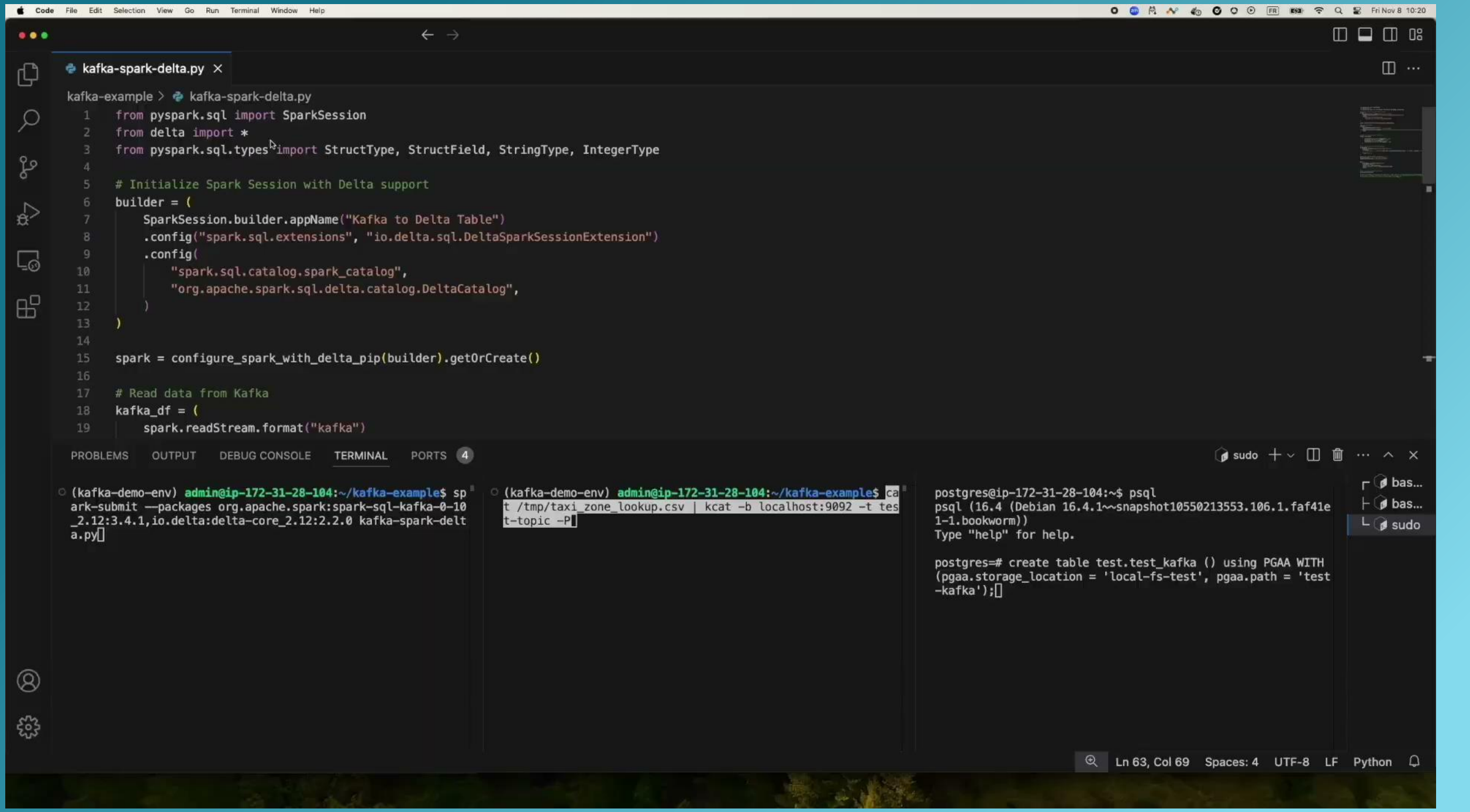
View All Users (25)

Recently Added

John Smith
Owner • john.smith@rocketinsights.com

Jane Richardson
Viewer • jane.richardson@rocketinsights.com

Jay Rodriguez
Editor • jay.rodriguez@rocketinsights.com


kafka-spark-delta.py
kafka-example > kafka-spark-delta.py
from pyspark.sql import SparkSession
from delta import *
from pyspark.sql.types import StructType, StructField, StringType, IntegerType
# Initialize Spark Session with Delta support
builder = (
SparkSession.builder.appName("Kafka to Delta Table")
.config("spark.sql.extensions", "io.delta.sql.DeltaSparkSessionExtension")
.config(
"spark.sql.catalog.spark_catalog",
"org.apache.spark.sql.delta.catalog.DeltaCatalog",
)
)
spark = configure_spark_with_delta_pip(builder).getOrCreate()
# Read data from Kafka
kafka_df = (
spark.readStream.format("kafka")
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS 4
sudo
(kafka-demo-env) admin@ip-172-31-28-104:~/kafka-example$ spark-submit --packages org.apache.spark:spark-sql-kafka-0-10_2.12:3.4.1,io.delta:delta-core_2.12:2.2.0 kafka-spark-delta.py
(kafka-demo-env) admin@ip-172-31-28-104:~/kafka-example$ cat /tmp/taxi_zone_lookup.csv | kcat -b localhost:9092 -t test-topic -P
postgres@ip-172-31-28-104:~$ psql
psql (16.4 (Debian 16.4.1~~snapshot10550213553.106.1.faf41e1-1.bookworm))
Type "help" for help.
postgres=# create table test.test_kafka () using PGAA WITH (pgaa.storage_location = 'local-fs-test', pgaa.path = 'test-kafka');
bas...
bas...
sudo
Ln 63, Col 69
Spaces: 4
UTF-8
LF
Python

Reduced total cost of ownership by decoupling compute and storage.

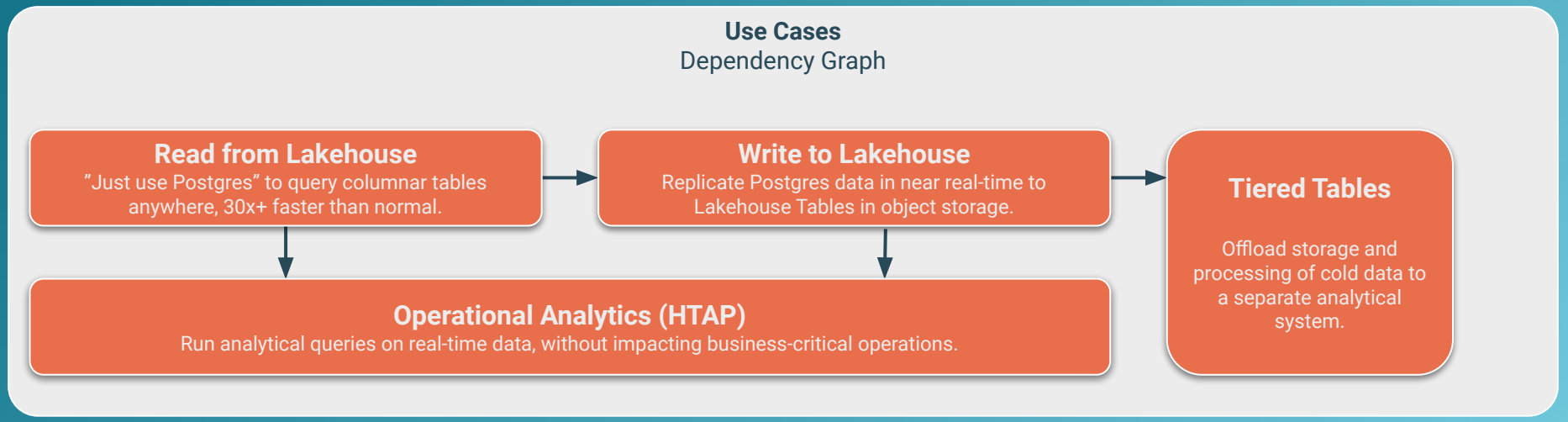
30X
faster
on *average* for analytical queries compared to Postgres

5X
smaller
on disk Lakehouse tables vs. Postgres tables and indexes

18X
cost-effective
Object storage vs. solid state drives (SSDs)

What do we have now, and what's next?

Up Next: Tiered Tables preview release is the next step on "the road to HTAP"



H1: Shipped

PGAA: Read from Lakehouse

"Just use Postgres" to query columnar tables anywhere, 30x+ faster than normal.

Q4: Preview

PGAA+PGD: Tiered Tables

Offload storage and processing of "cold" data to a separate analytical system

H1 2025: Up Next

Continuous Replication & HTAP

Replicate and query Postgres data in near real-time to Lakehouse Tables in object storage.


EDB
Postgres for the AI Generation
EDB Postgres AI Day Zurich | 13 November 2024
Break time

Part II: EDB Postgres AI - PG for the 21st Century

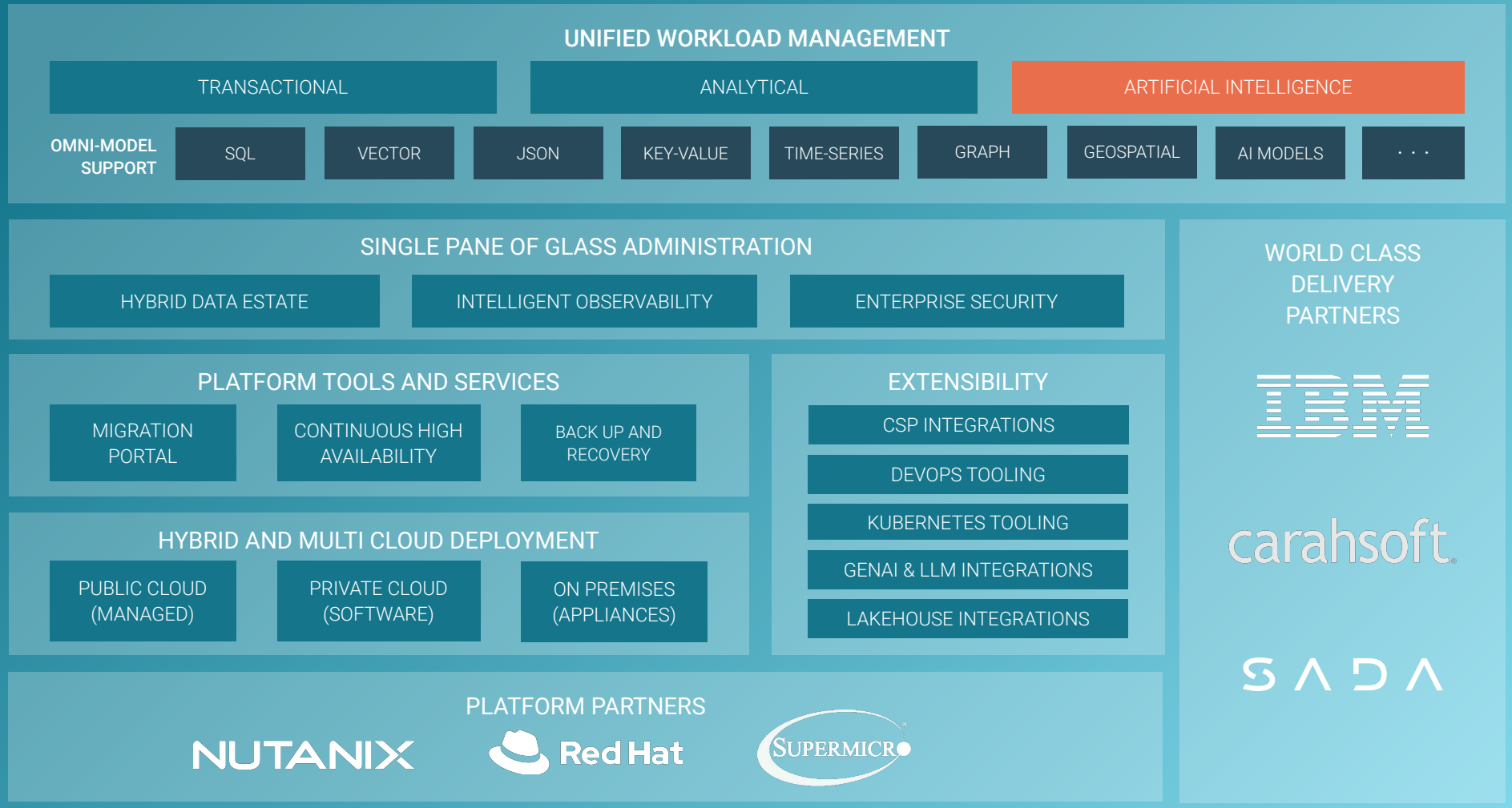
Renaissance Zurich Tower Hotel
13th November 2024

Lucie Zeng - Associate Data Scientist EMEA
lucie.zeng@enterprisedb.com

Franck Sidi - Senior Director EMEA & WW Field CTO Leader
franck.sidi@enterprisedb.com

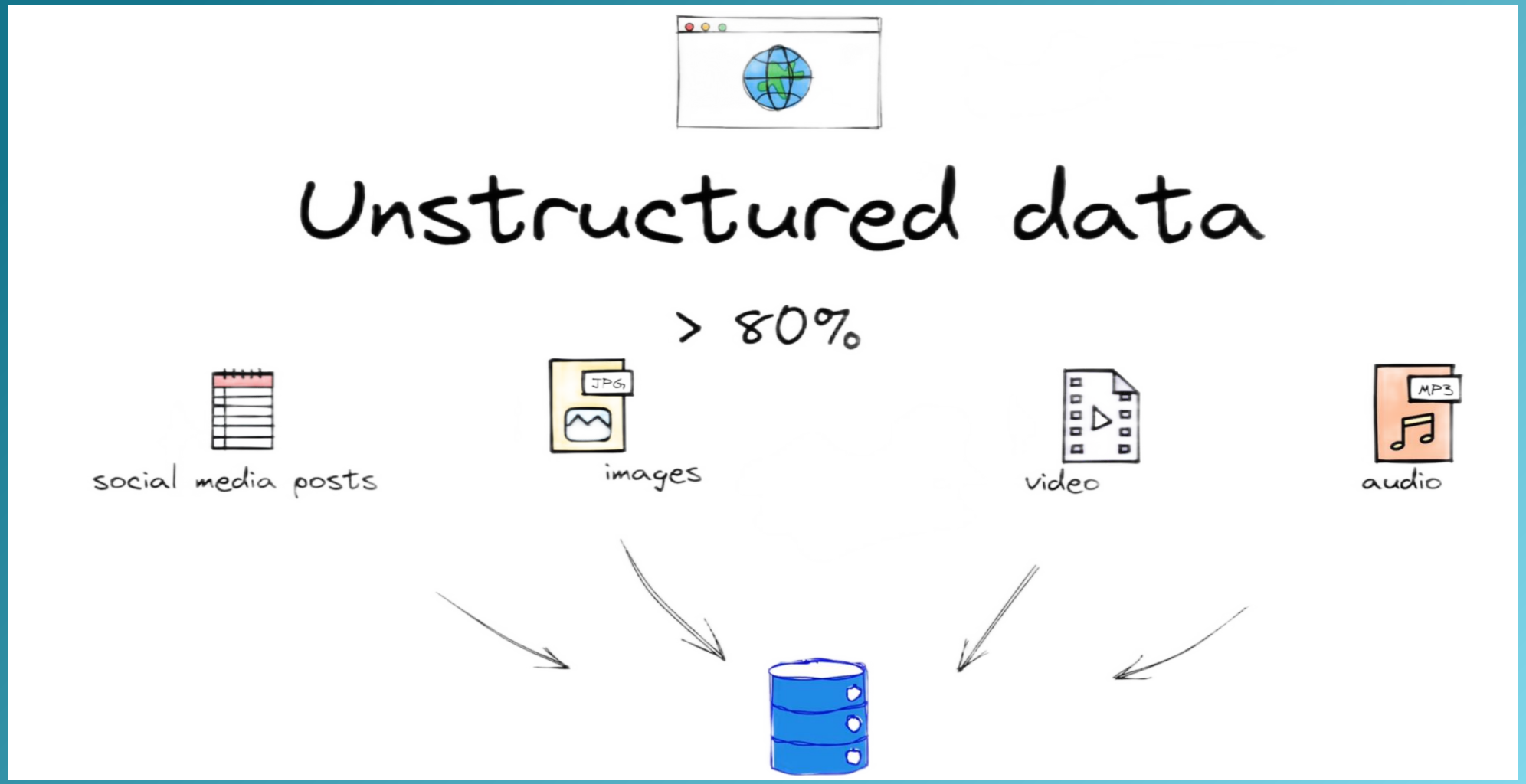
EDB AI

The EDB Postgres AI Platform

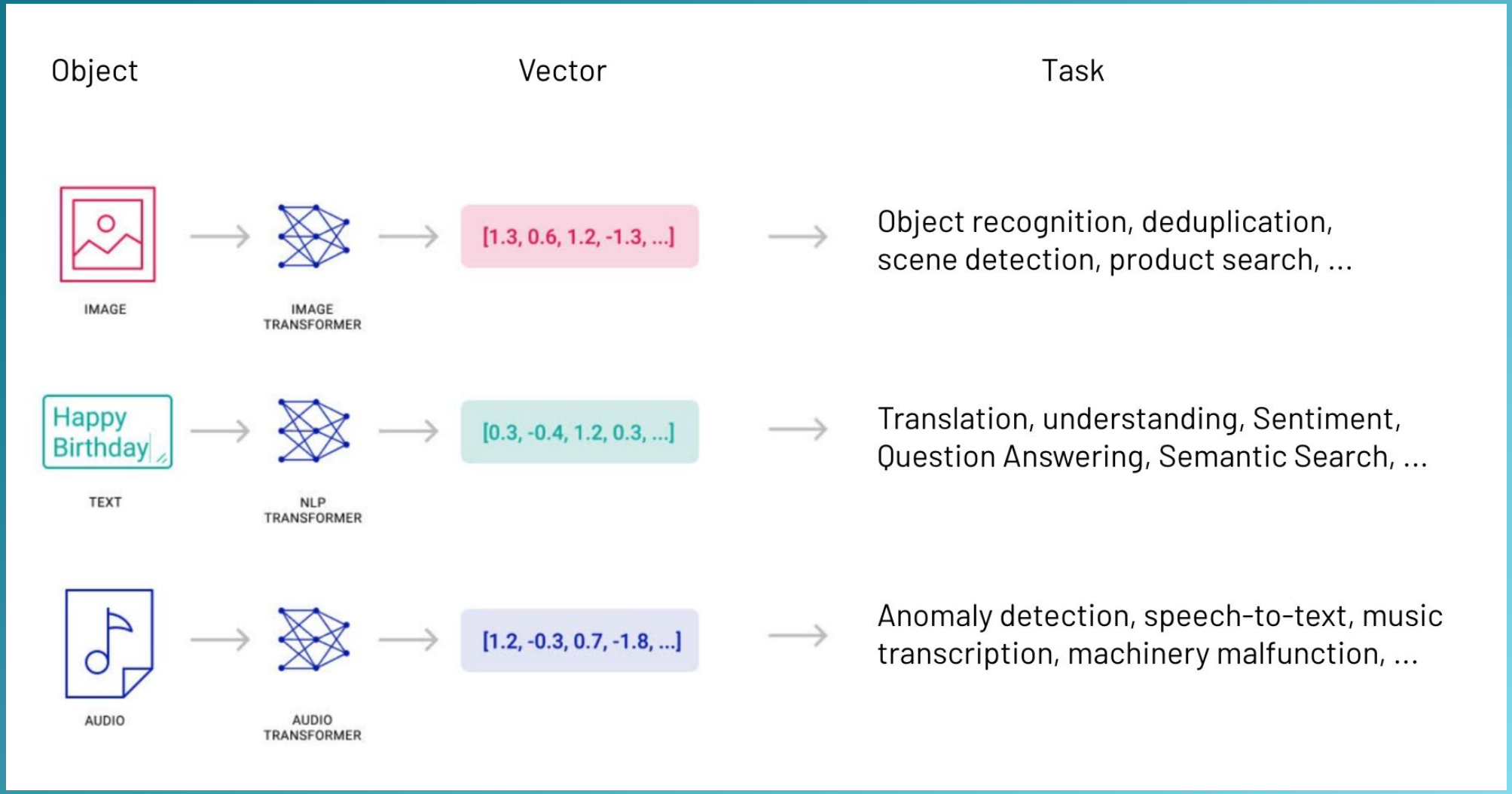


What is PostgreSQL after pg vector extension for a DBA vs AI Scientist?

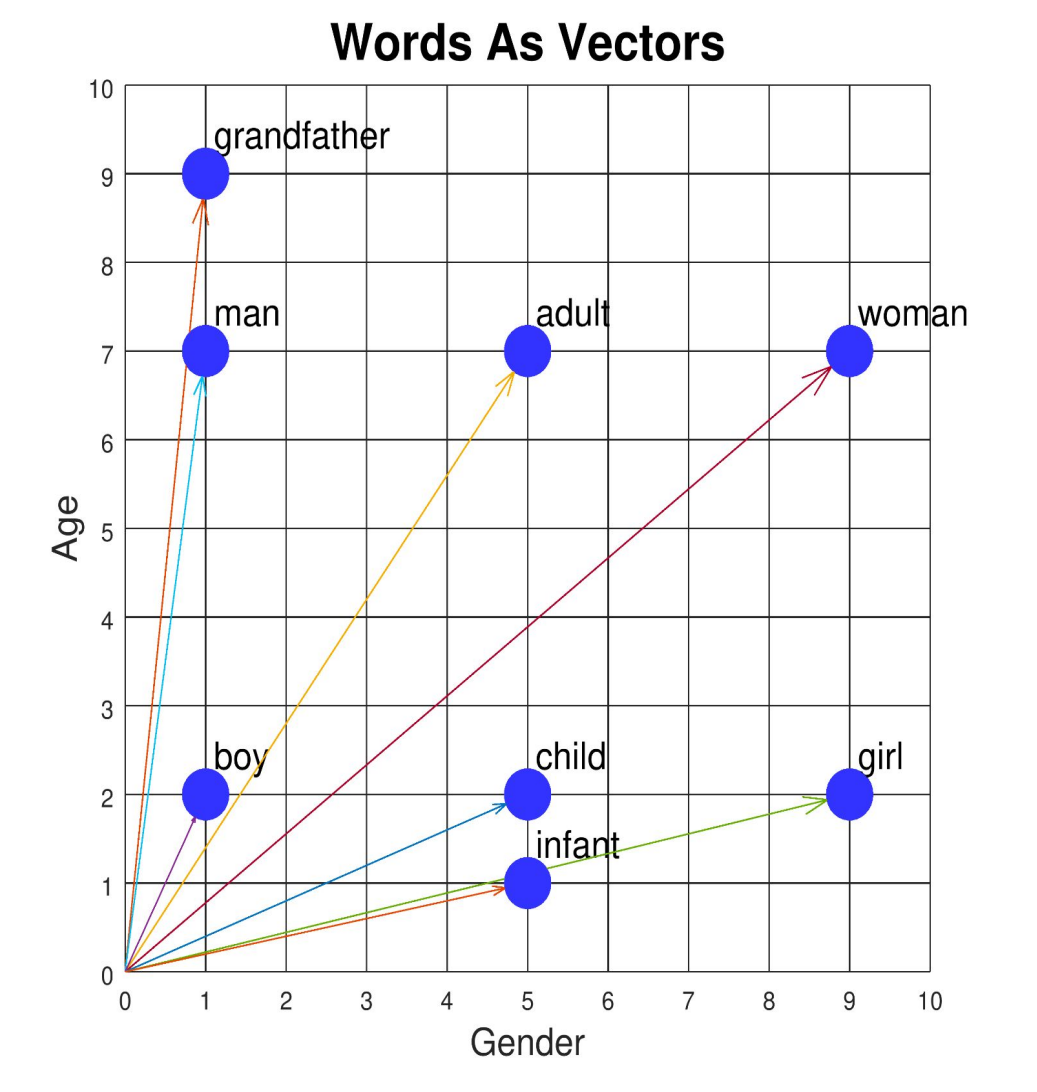
How to analyse Unstructured Data?

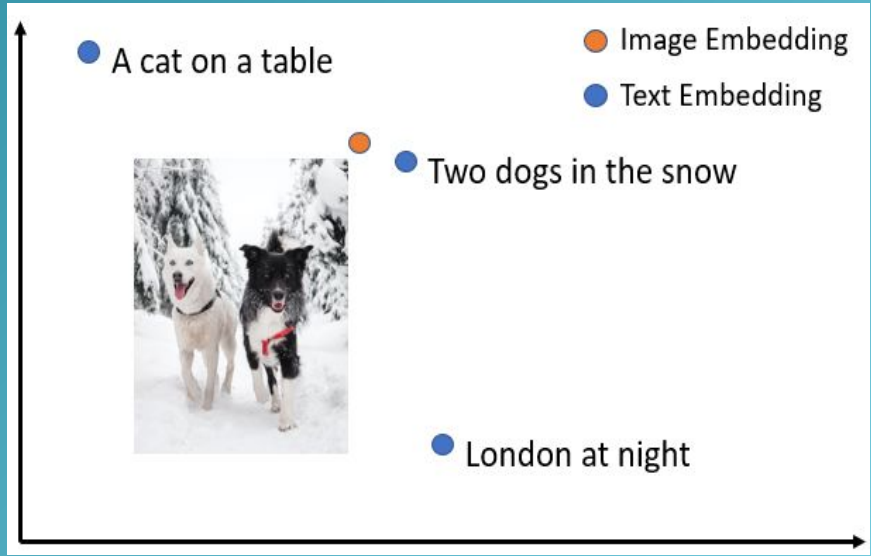


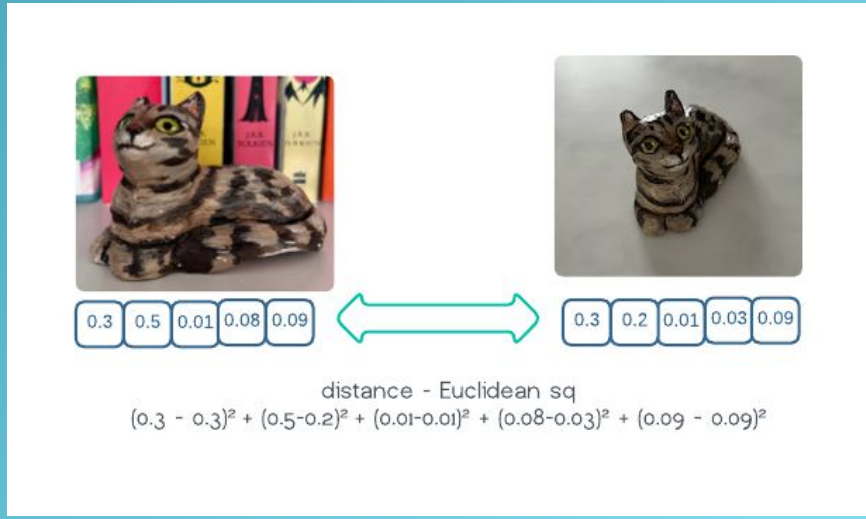
How are data & vectors connected to each other?



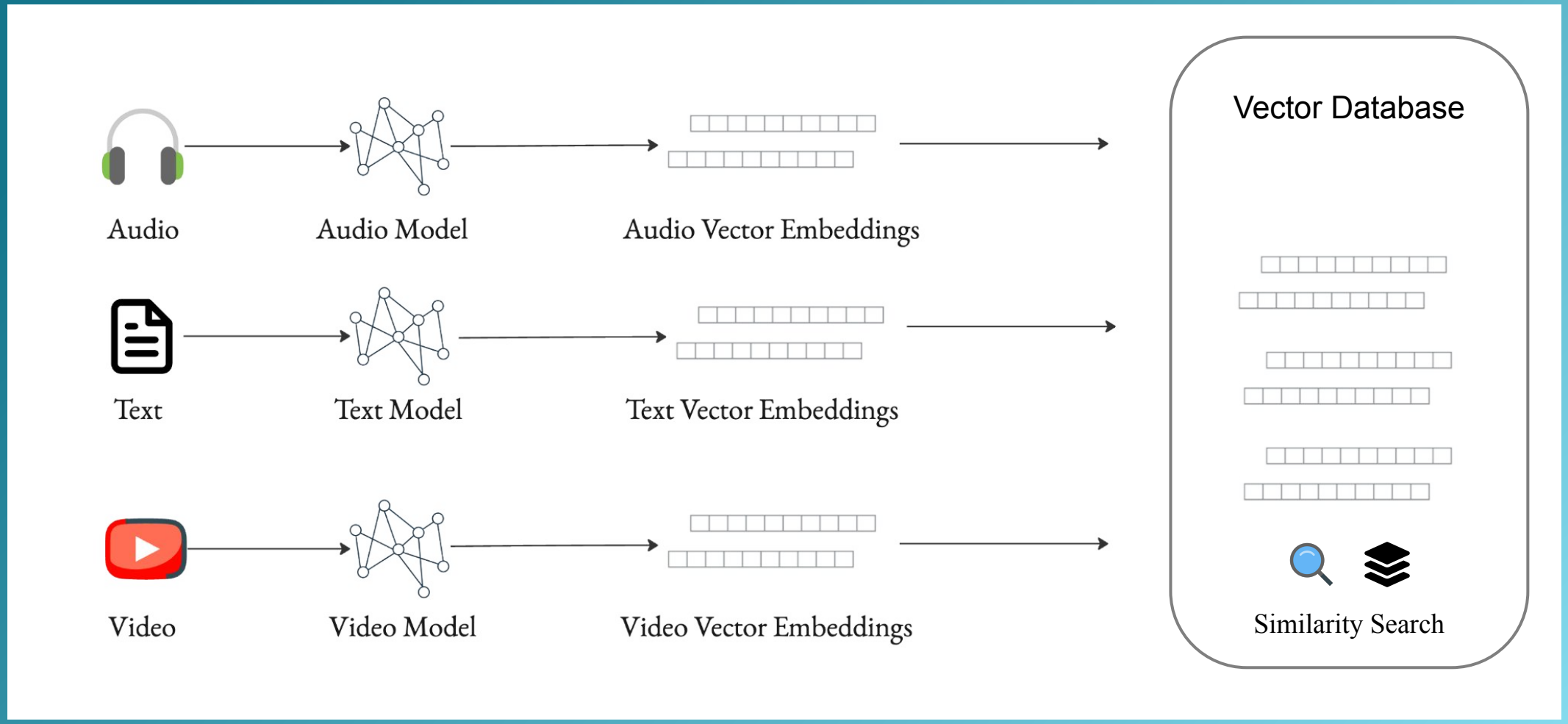
Similarity Search



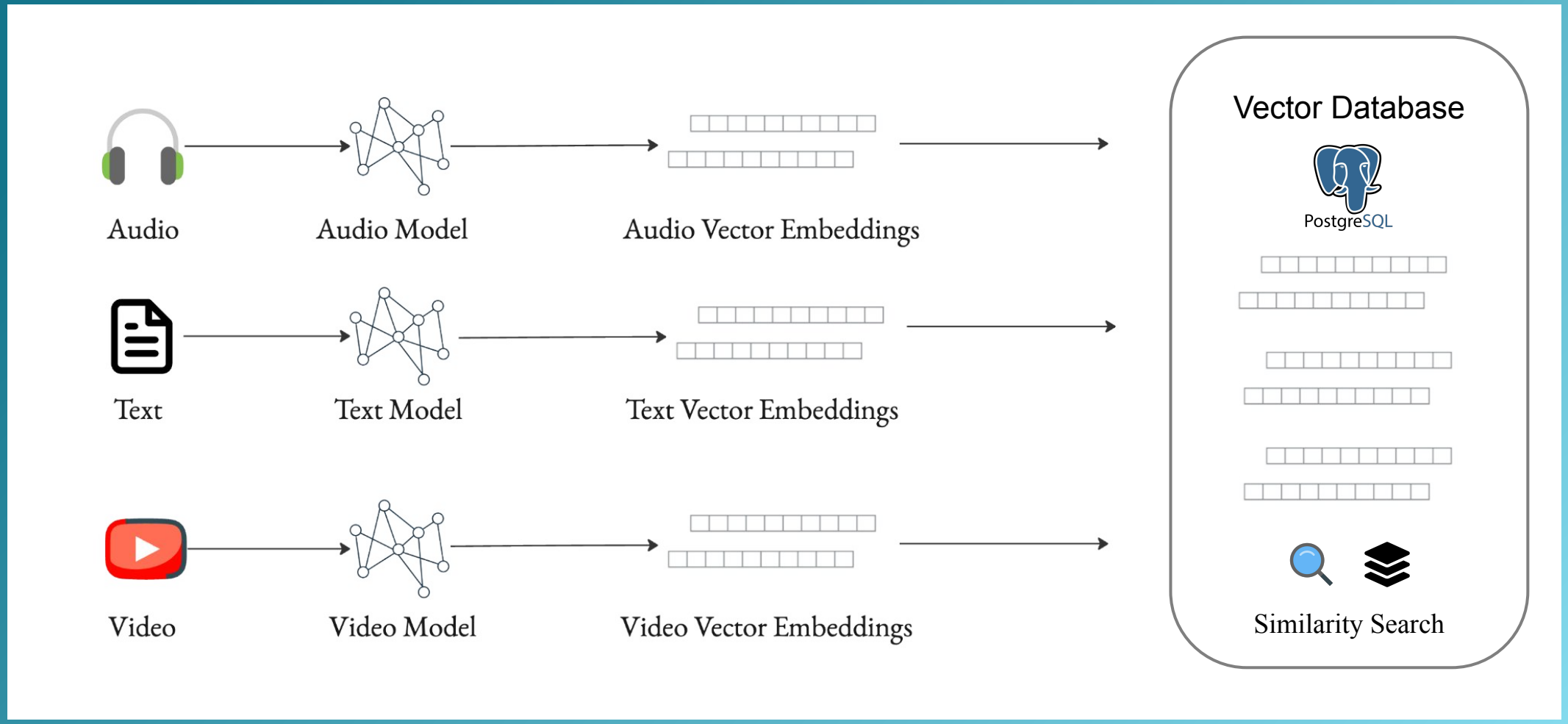




How are data & vectors connected to each other?



How are data & vectors connected to each other?



pgVector Use Cases Overview

Use Cases

- Conversational Assistance
- Chatbot
- Semantic search
- Visual and Semantic search for images
- Speech recognition and audio search (e.g. call center audio logs)
- Real-time and personalized search experience
- …

Content examples

- Documents (contracts, user guides, product/technical documentation, policies, CVs…)
- Emails
- Intranet documents (e.g. sharepoint, …)
- ….

Models and Vector DB benefits

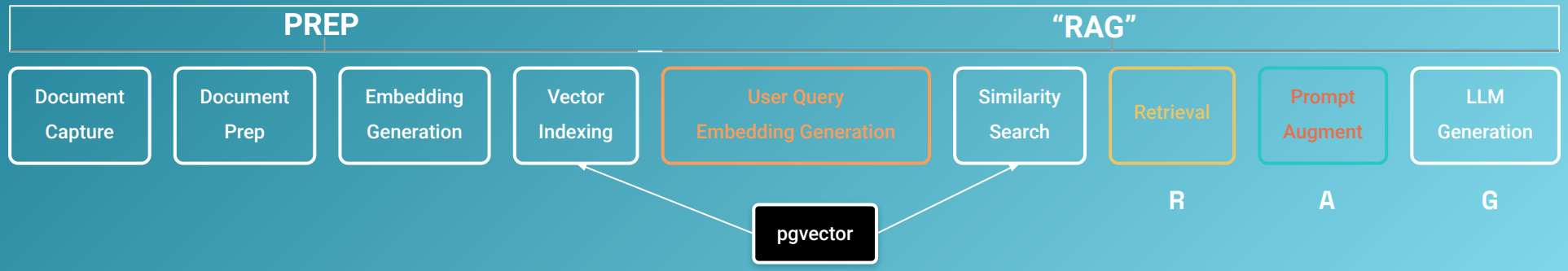
- Core for vector similarity search that provides fast and scalable (with improved accuracy) experience

Generative AI Use Cases

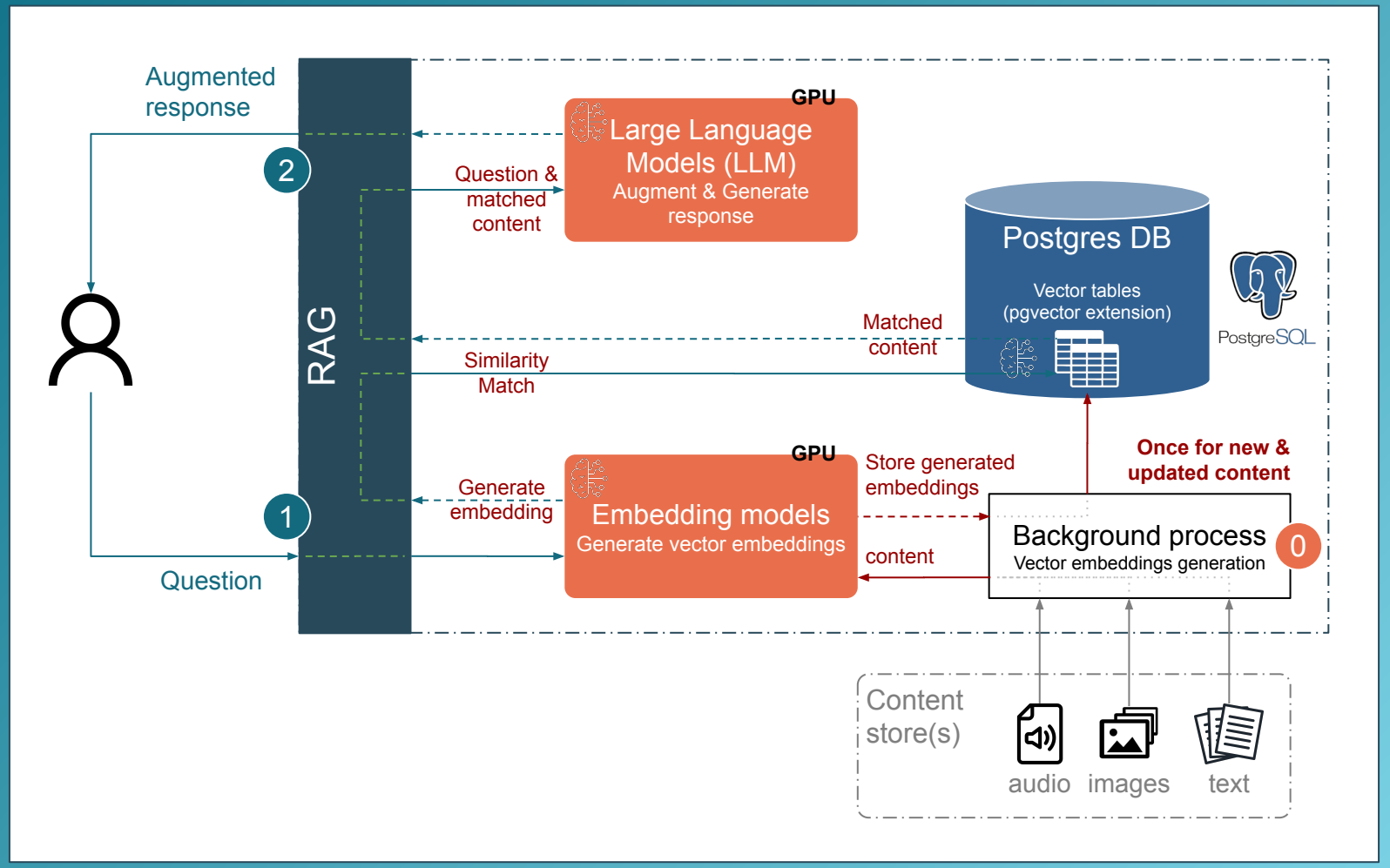
💸 **GenAI Applications** are the dominant focus of investment across the IT Industry today.

😵‍💫 **GenAI Applications** are *data centric*, *complex*, and the solutions are *nascent and largely piecemeal*.

"There is not a defacto enterprise grade standard"



RAG (**R**etrieval **A**ugmented **G**eneration) Overview



RAG Interactions - detailed flow

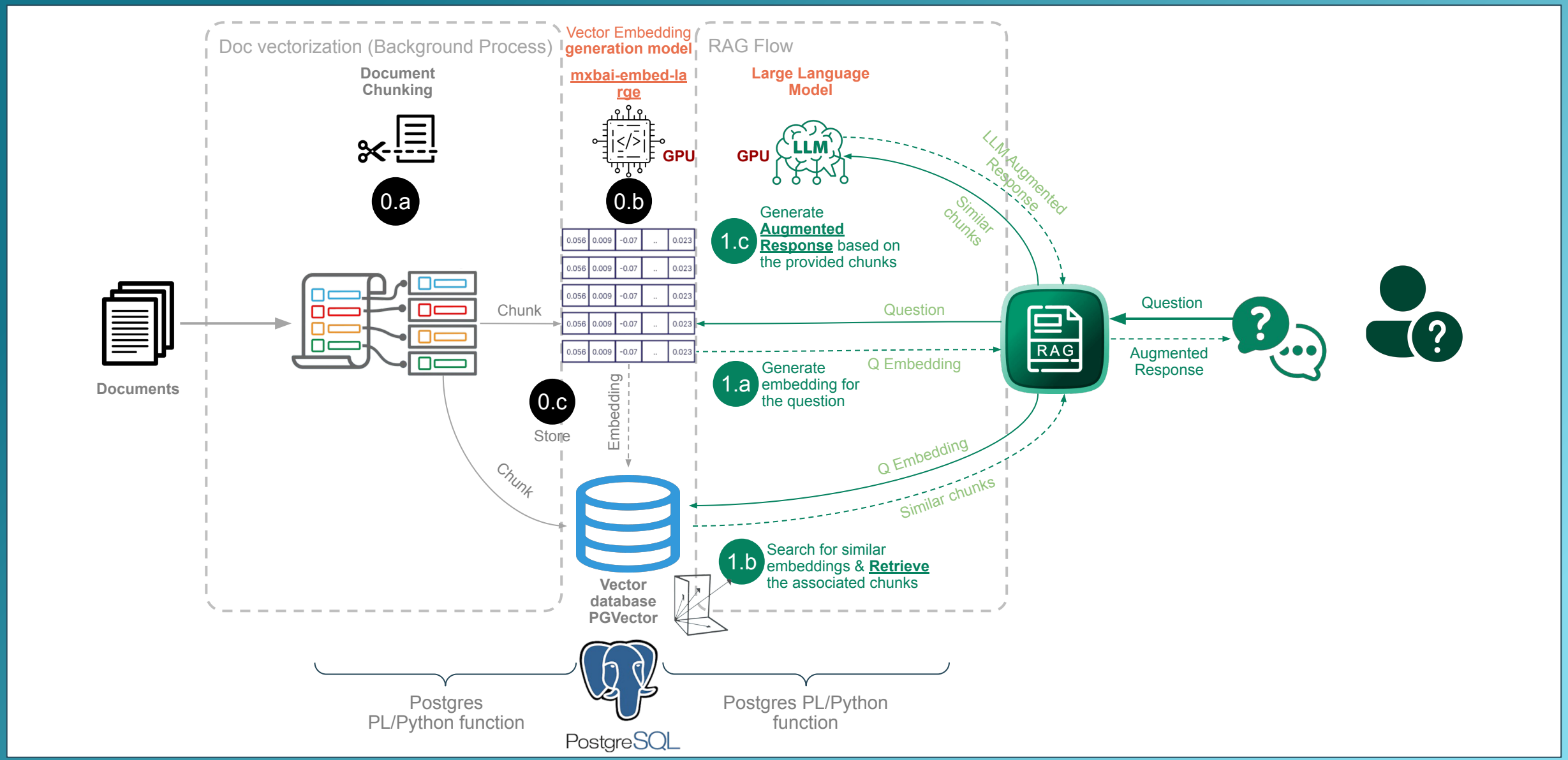
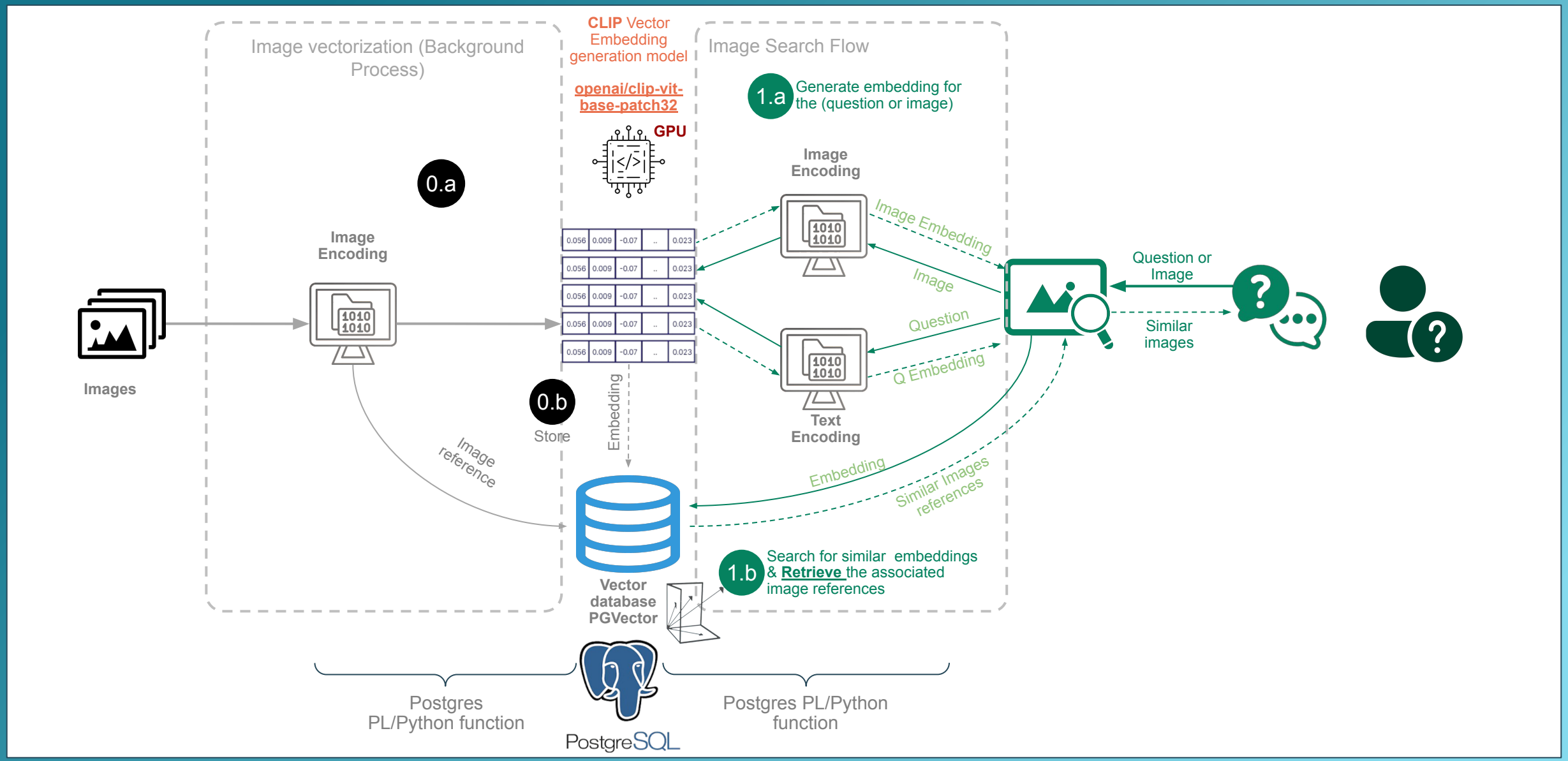
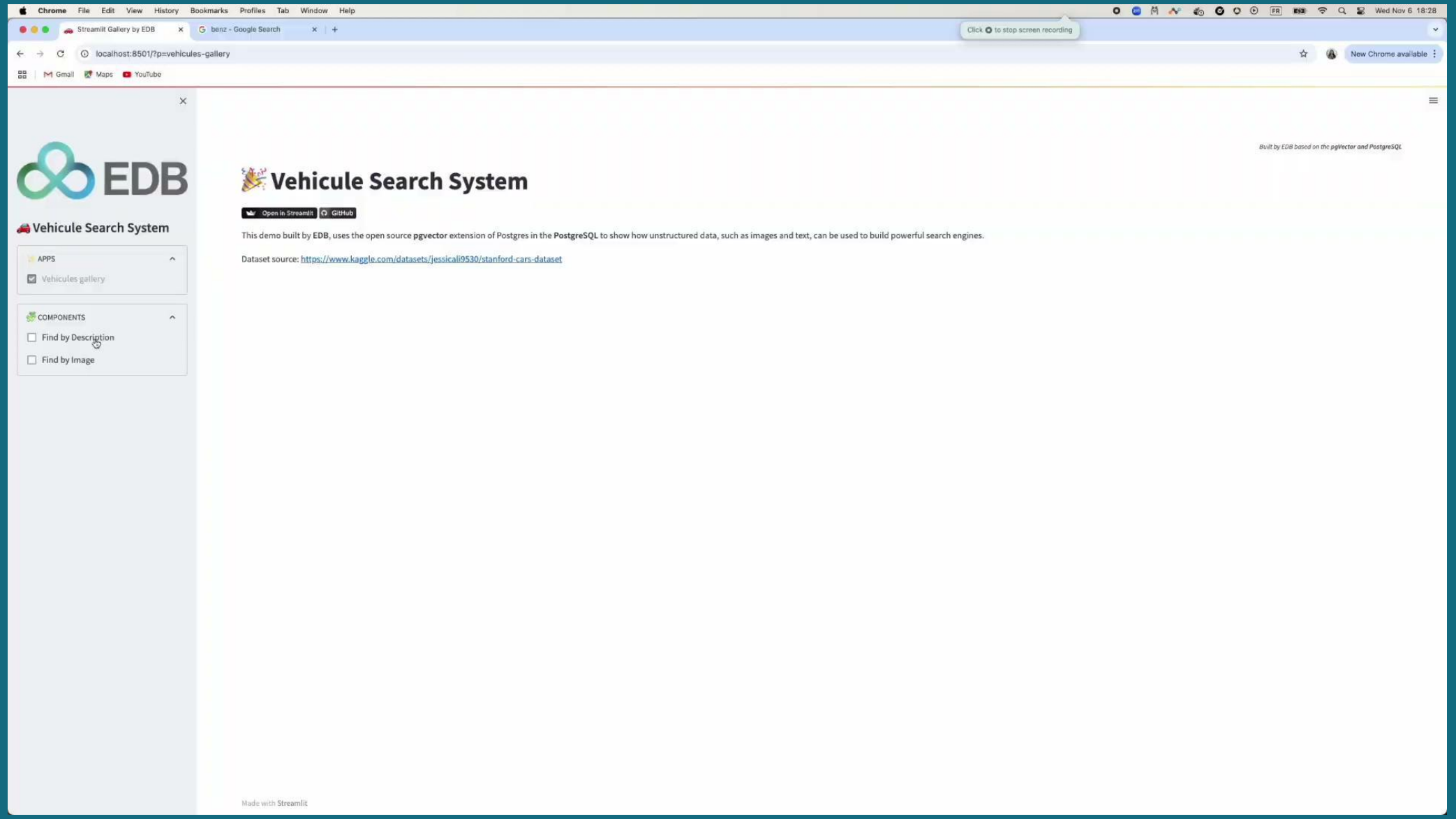


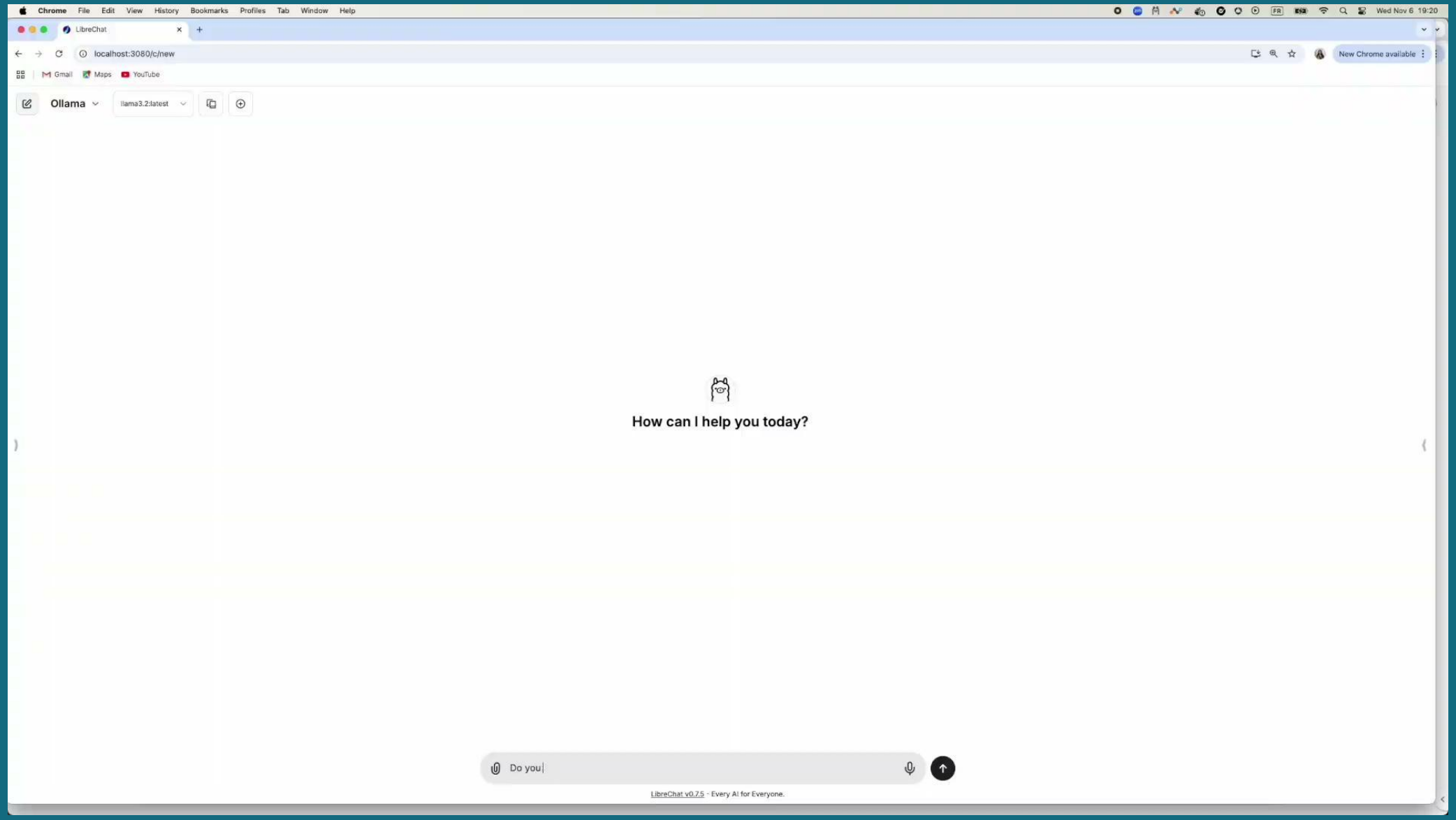
Image Search Interactions - detailed flow



Demo Similarity Search



Demo RAG


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Wed Nov 6 19:20
LibreChat
localhost:3080/c/new
New Chrome available
Gmail Maps YouTube
Ollama
llama3.2:latest
How can I help you today?
Do you
LibreChat v0.7.5 - Every AI for Everyone.

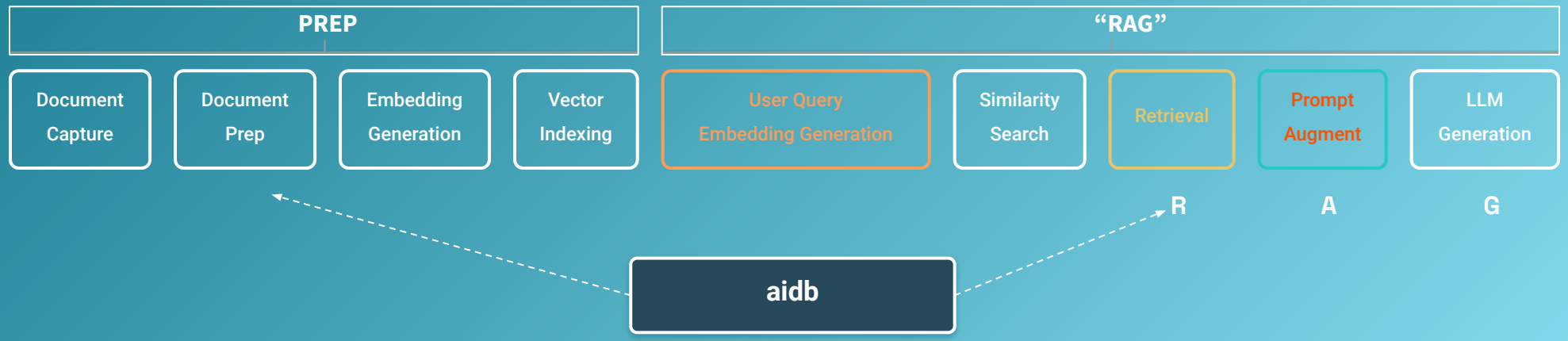
What about aidb?

aidb Extension

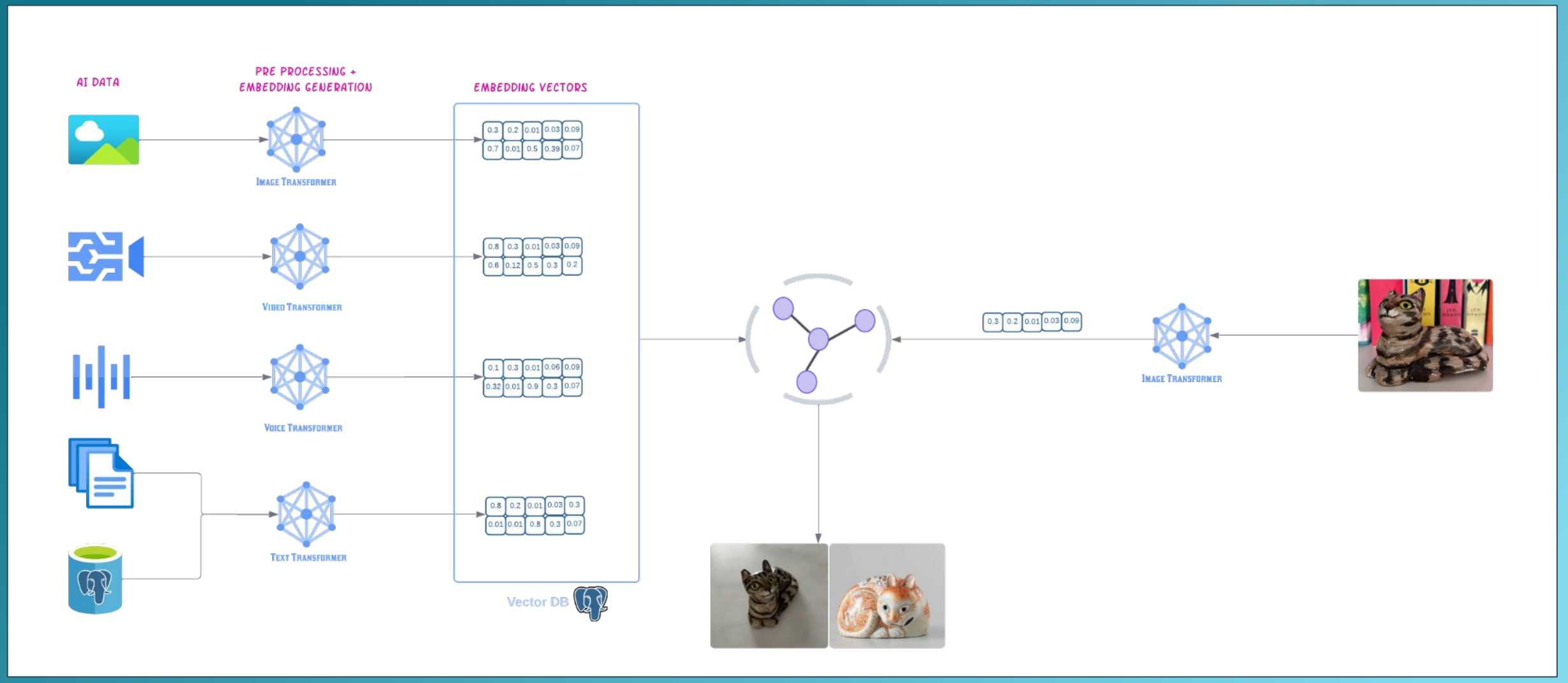
Our opportunity is necessitated by two dominant factors:

🥇 For businesses to run enterprise-grade, mission-critical GenAI apps, they need an *enterprise-grade data management platform*.

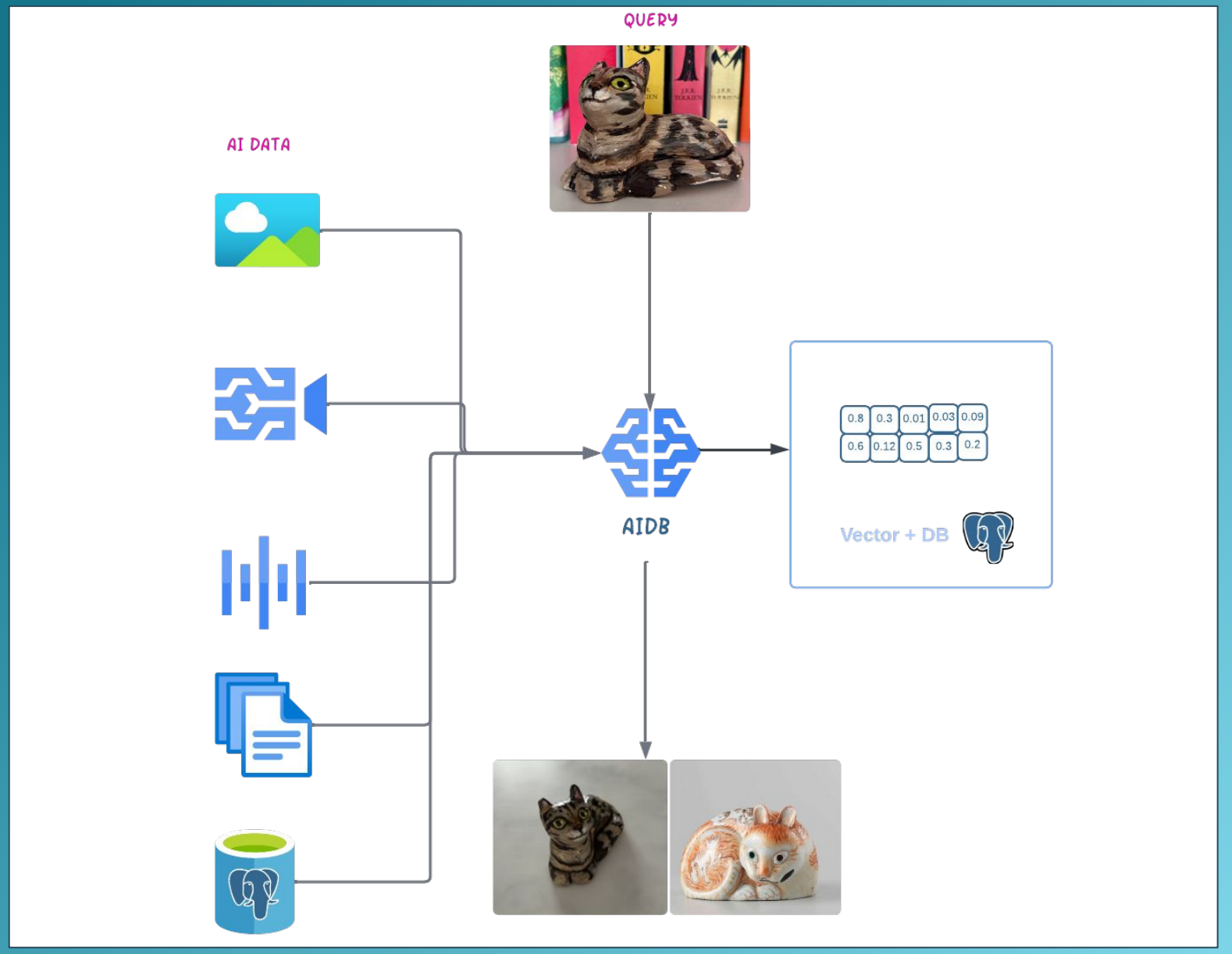
🥈 Our *partner strategy* is anchored around enabling the development of GenAI applications on Postgres *within those partners' ecosystems*.



A recommendation engine with pgvector



A recommendation engine with AIDB



AIDB - Encoders

```
postgres=# select * from aidb.encoders;
 id |                 name                 |  provider   | max_tokens | default_distance_metric | dimensions
----+--------------------------------------+-------------+------------+-------------------------+------------
  1 | text-embedding-ada-002               | openai      |       8191 | cosine                  |       1536
  2 | text-embedding-3-small               | openai      |       8191 | cosine                  |       1536
  3 | text-embedding-3-large               | openai      |       8191 | cosine                  |       2000
  4 | clip-vit-base-patch32                | openai      |        512 | cosine                  |        512
  5 | gtr-t5-xxl                           | huggingface |        512 | dot                     |        768
  6 | gtr-t5-xl                            | huggingface |        512 | dot                     |        768
  7 | sentence-t5-xxl                      | huggingface |        256 | dot                     |        768
  8 | gtr-t5-large                         | huggingface |        512 | dot                     |        768
  9 | all-mpnet-base-v1                    | huggingface |        512 | dot                     |        768
 10 | multi-qa-mpnet-base-cos-v1           | huggingface |        512 | dot                     |        768
 11 | all-roberta-large-v1                 | huggingface |        256 | dot                     |       1024
 12 | sentence-t5-xl                       | huggingface |        256 | dot                     |        768
 13 | all-MiniLM-L12-v1                    | huggingface |        256 | dot                     |        384
 14 | gtr-t5-base                          | huggingface |        512 | dot                     |        768
 15 | sentence-t5-large                    | huggingface |        256 | dot                     |        768
 16 | all-MiniLM-L6-v1                     | huggingface |        256 | dot                     |        384
 17 | msmarco-bert-base-dot-v5             | huggingface |        512 | dot                     |        768
 18 | multi-qa-MiniLM-L6-dot-v1            | huggingface |        512 | dot                     |        384
 19 | sentence-t5-base                     | huggingface |        256 | dot                     |        768
 20 | msmarco-distilbert-base-tas-b        | huggingface |        512 | dot                     |        768
 21 | msmarco-distilbert-dot-v5            | huggingface |        512 | dot                     |        768
 22 | multi-qa-mpnet-base-dot-v1           | huggingface |        512 | dot                     |        384
 23 | multi-qa-distilbert-dot-v1           | huggingface |        512 | dot                     |        768
 24 | paraphrase-MiniLM-L6-v2              | huggingface |        128 | cosine                  |        384
 25 | paraphrase-TinyBERT-L6-v2            | huggingface |        128 | cosine                  |        768
 26 | paraphrase-MiniLM-L12-v2             | huggingface |        256 | cosine                  |        384
 27 | paraphrase-distilroberta-base-v2     | huggingface |        256 | cosine                  |        768
 28 | paraphrase-mpnet-base-v2             | huggingface |        512 | cosine                  |        768
 29 | all-mpnet-base-v2                    | huggingface |        384 | cosine                  |        768
 30 | all-distilroberta-v1                 | huggingface |        512 | cosine                  |        768
 31 | all-MiniLM-L12-v2                    | huggingface |        256 | cosine                  |        384
 32 | multi-qa-distilbert-cos-v1           | huggingface |        512 | cosine                  |        768
 33 | all-MiniLM-L6-v2                     | huggingface |        256 | cosine                  |        384
 34 | multi-qa-MiniLM-L6-cos-v1            | huggingface |        512 | cosine                  |        384
 35 | paraphrase-multilingual-mpnet-base-v2 | huggingface |        128 | cosine                  |        768
 36 | paraphrase-albert-small-v2           | huggingface |        256 | cosine                  |        768
 37 | paraphrase-multilingual-MiniLM-L12-v2 | huggingface |        128 | cosine                  |        384
 38 | paraphrase-MiniLM-L3-v2              | huggingface |        128 | cosine                  |        384
 39 | distiluse-base-multilingual-cased-v1 | huggingface |        128 | cosine                  |        512
 40 | distiluse-base-multilingual-cased-v2 | huggingface |        128 | cosine                  |        512
(40 rows)
```

```
SELECT provider, count(*) encoder_model_count FROM aidb.encoders gro
```

OUTPUT

```
  provider   | encoder_model_count
-------------+---------------------
 huggingface |                  36
 openai      |                   4
(2 rows)
```

AIDB - Create Retriever - Postgres as a Source

```
SELECT aidb.create_pg_retriever(
    'product_embeddings_auto', -- Retriever name
    'public', -- Schema
    'product_id', -- Primary key
    'all-MiniLM-L6-v2', -- embedding model
    'text', -- data type
    'products', -- Source table
    ARRAY['product_name', 'description'], -- Columns to vectorize
    TRUE -- auto embeddings TRUE to set trigger
);
```

AIDB - Create Retriever - S3 as a Source

```
SELECT aidb.create_s3_retriever(
    'image_embeddings', -- Name of the similarity retrieval setup
    'public', -- Schema of the source table
    'clip-vit-base-patch32', -- Embeddings encoder model for similar
    'img', -- data type, could be either img or text
    'torsten', -- S3 bucket name
    '', -- prefix
    'https://s3.us-south.cloud-object-storage.appdomain.cloud' -- s3
);
```

```
SELECT aidb.refresh_retriever('image_embeddings');
```

AIDB - Retrieve Data from Retriever

```
SELECT data FROM aidb.retrieve(
'I like it', -- The query text to retrieve the top similar data
 5, -- top K
'product_embeddings_auto' -- retriever's name
);
```

OUTPUT

```
data
------------------------------------
 {'data': 'Hamburger - Tasty'}
 {'data': 'Cheesburger - Very tasty'}
 {'data': 'Pizza - Mkay'}
 {'data': 'Sandwich - So what'}
 {'data': 'Kebab - Maybe'}
(5 rows)
```

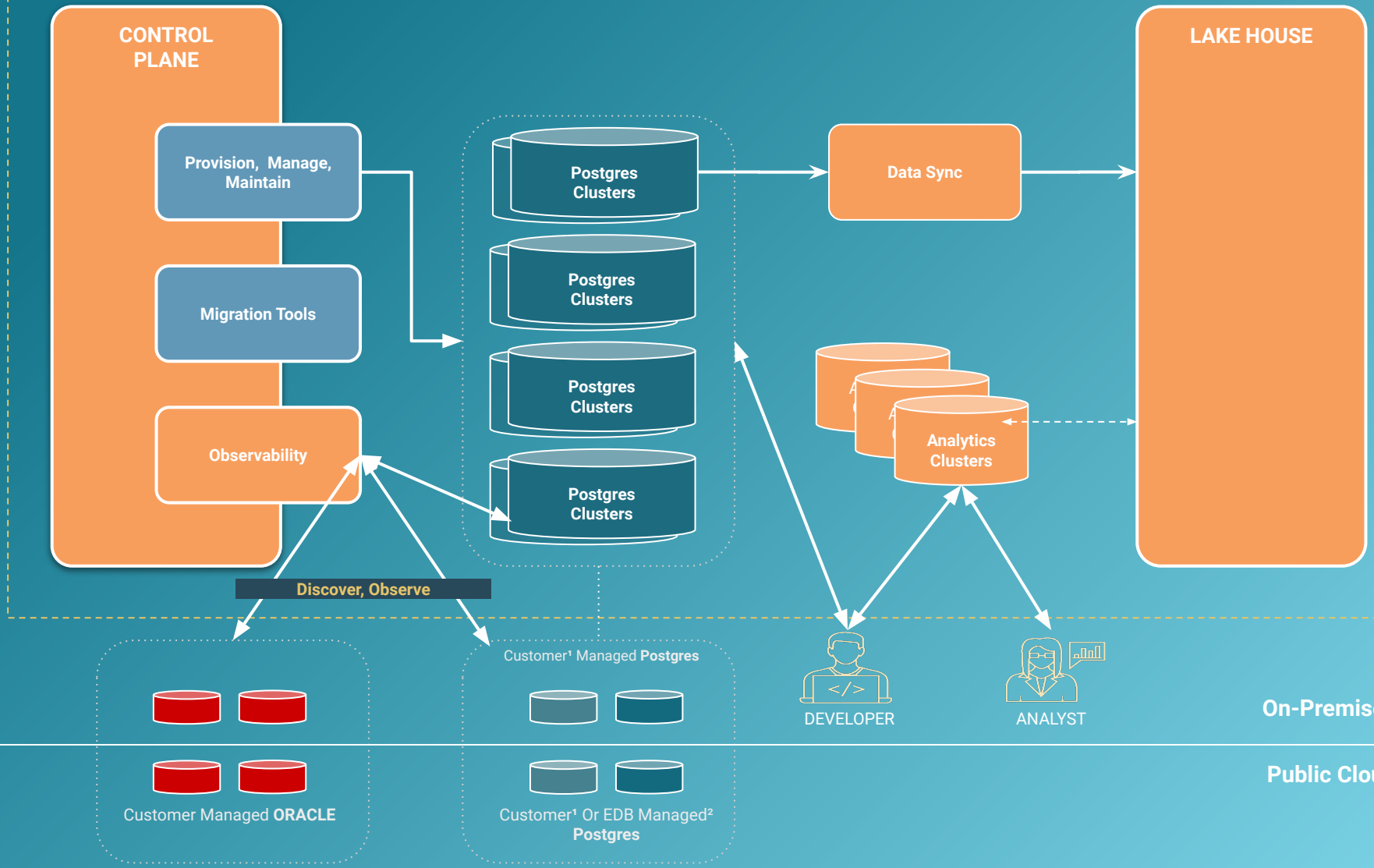
```
SELECT data from aidb.retrieve_via_s3(
    'image_embeddings', -- retriever's name
     1, -- top K
    'torsten', -- S3 bucket name
    'foto.jpg', -- object name
    'https://s3.us-south.cloud-object-storage.appdomain.cloud'
);
```

OUTPUT

```
data
--------------------
 {'img_id': 'foto'}
 (1 row)
```

Conclusion

EDB POSTGRES AI



Legend

- Transactional Capabilities
- Existing but extended capability
- New Capability

EDB AI: Innovative Platform 100% Based on Postgres

New Apps and Workloads: Analytics, HTAP, RAG, Recommender

Deploy Anywhere and leverage All Infrastructure: True Hybrid Cloud Solution. Leverage GPUs, Object Storage

Support Any Data Format: Efficient Data Processing & Open Data Formats including Columnar & Compression


EDB
Postgres for the AI Generation
EDB Postgres AI Day Zurich | 13 November 2024
THANK YOU!
Are you interested in a
PERSONALISED DEMO?
Let us know today or contact us at
emeamarketing-team@enterprisedb.com

Empowering Application Modernization of Legacy Systems with NDB e2e Database Management

Renaissance Zurich Tower Hotel
13th November 2024

Alberto Belotti,
NDB Solution Architect, Nutanix

NUTANIX

Businesses are Driving Strategic Initiatives with Numerous Constraints

Initiatives

Transform
Operate in new ways, engage new customers, and deliver exceptional experiences.

Grow
Automate to protect existing revenue and innovate to develop new revenue streams.

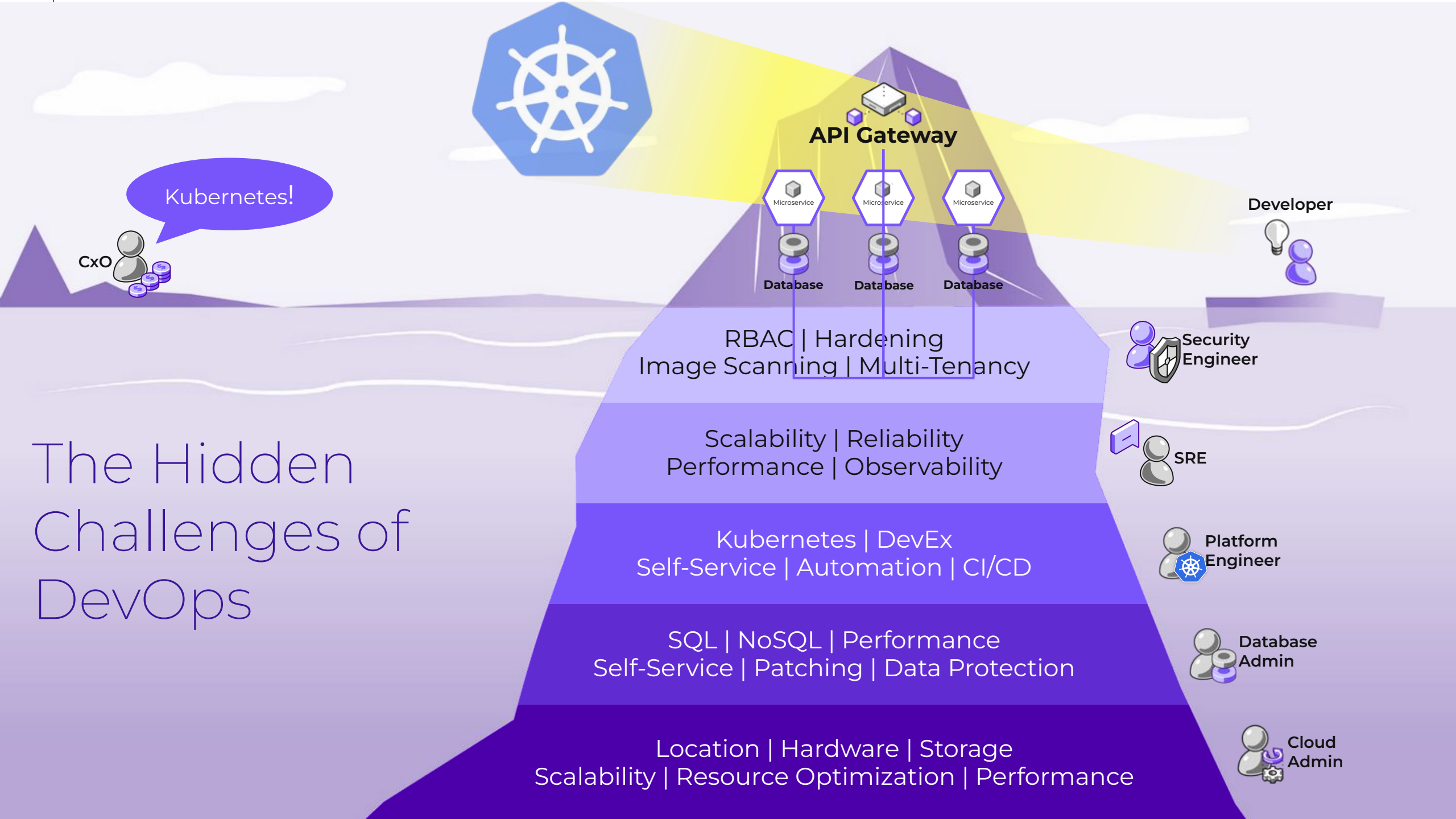
Manage Risk
Secure everything and maintain compliance across diverse locations, systems, networks, and apps.

Constraints

Budget
Do more with less, focus on cost control, squeeze as much out of existing systems as possible.

Staffing
Address skills gaps and lack of specialized training that are barriers to transformation and growth.

Time
Accelerate development under economic and competitive pressure.



The Hidden Challenges of DevOps

Siloed Operating Models and Tools Just Can't Keep Up

Why?

- Too Slow
- Too Complicated
- Too Expensive
- Not Enough Skilled Talent

What's Needed?

- Adopt a DevOps culture
- Implement a Cloud Platform
- Focus on Automation
- Establish Best Practices

Research Says that Organizations Want One Place to Manage Everything: A Consistent Cloud Operating Model

74%

of IT teams are expected to leverage more than one IT infrastructure in the next 1-3 years, including a mix of private and public clouds, multiple public clouds, or an on-premises datacenter along with a hosted datacenter

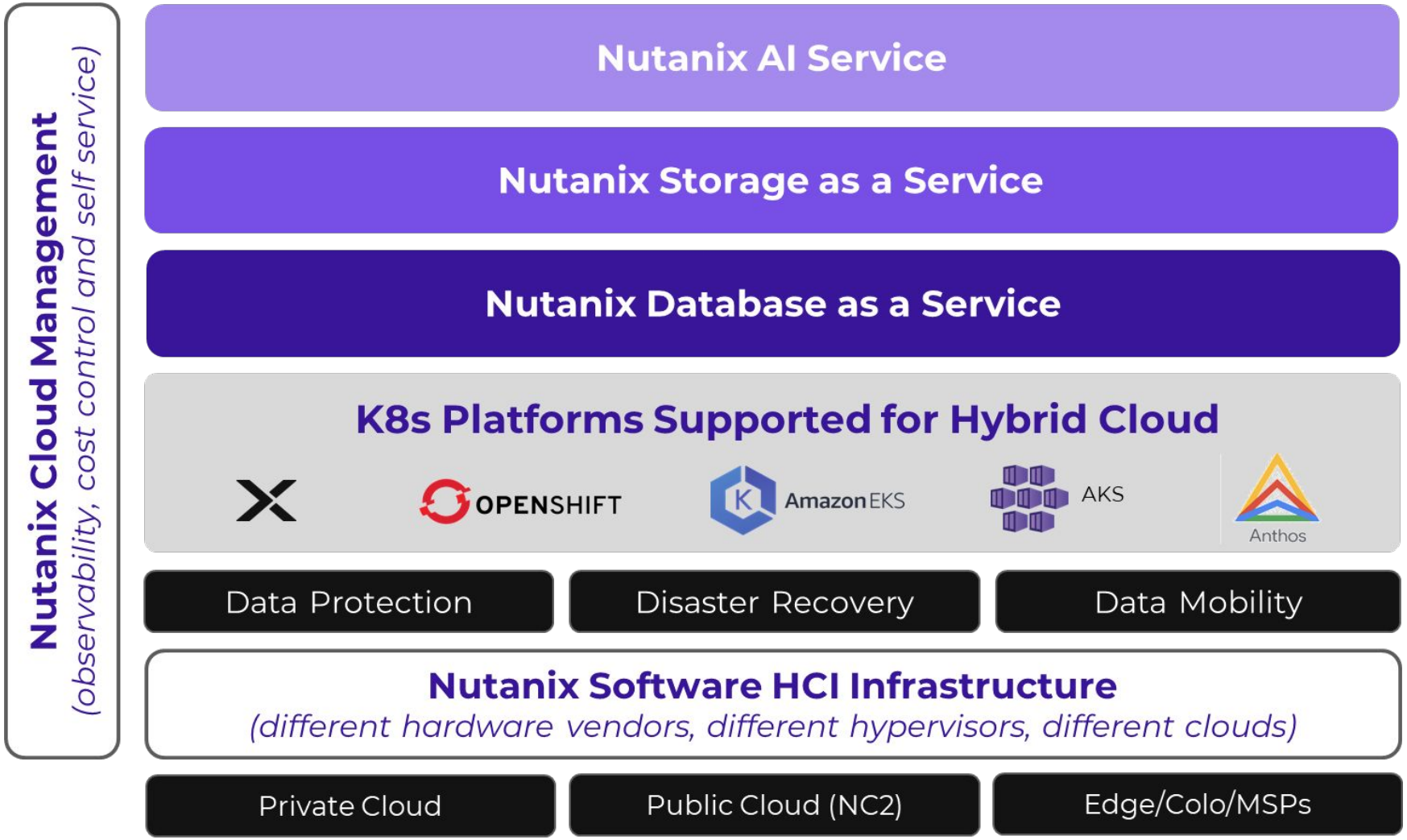
94%

say they'd benefit from having a single, unified place to manage applications and data across diverse environments

5th Annual Nutanix Enterprise Cloud Index Report, April 2023

The Nutanix Cloud Platform

for legacy applications, modern applications and AI-driven applications



Automation is the key to simplify DevOps adoption

DEVELOPMENT

Developer teams need self-service capabilities to accelerate the transition from coding to production.

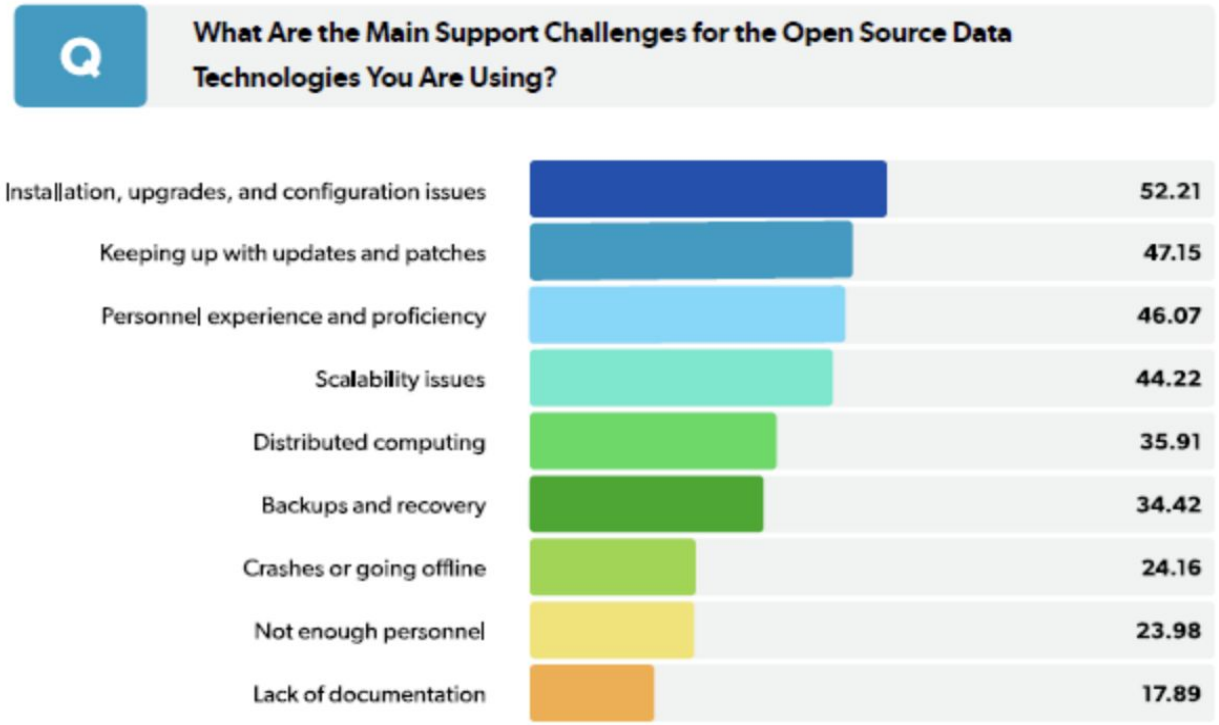
- Innovation
- Time to Market
- Reliability
by enhancing collaboration, reducing bottlenecks and not increasing costs

OPERATIONS

Operations teams should leverage existing skills and knowledges to effectively manage emerging technologies and navigate increasing complexity.

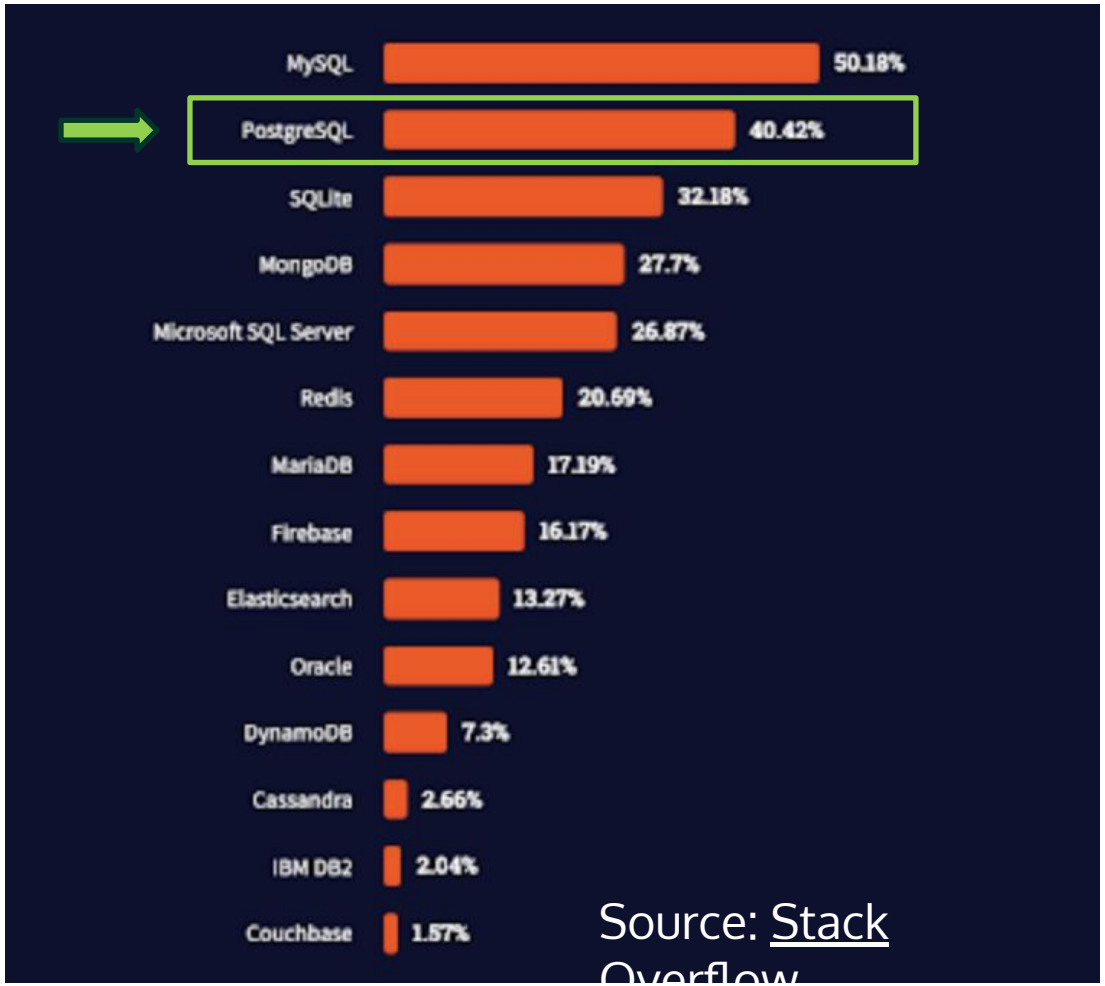
The Rise of Opensource Relational DB and NoSQL DB

Modern applications are driving the adoption of specialized, primarily open-source database technologies. Business-critical applications demand complex architectures that require advanced skills, which are often hard to find in the market.



Source: https://www.openlogic.com/resources/2022-open-source-report

Top-rated databases in 2022



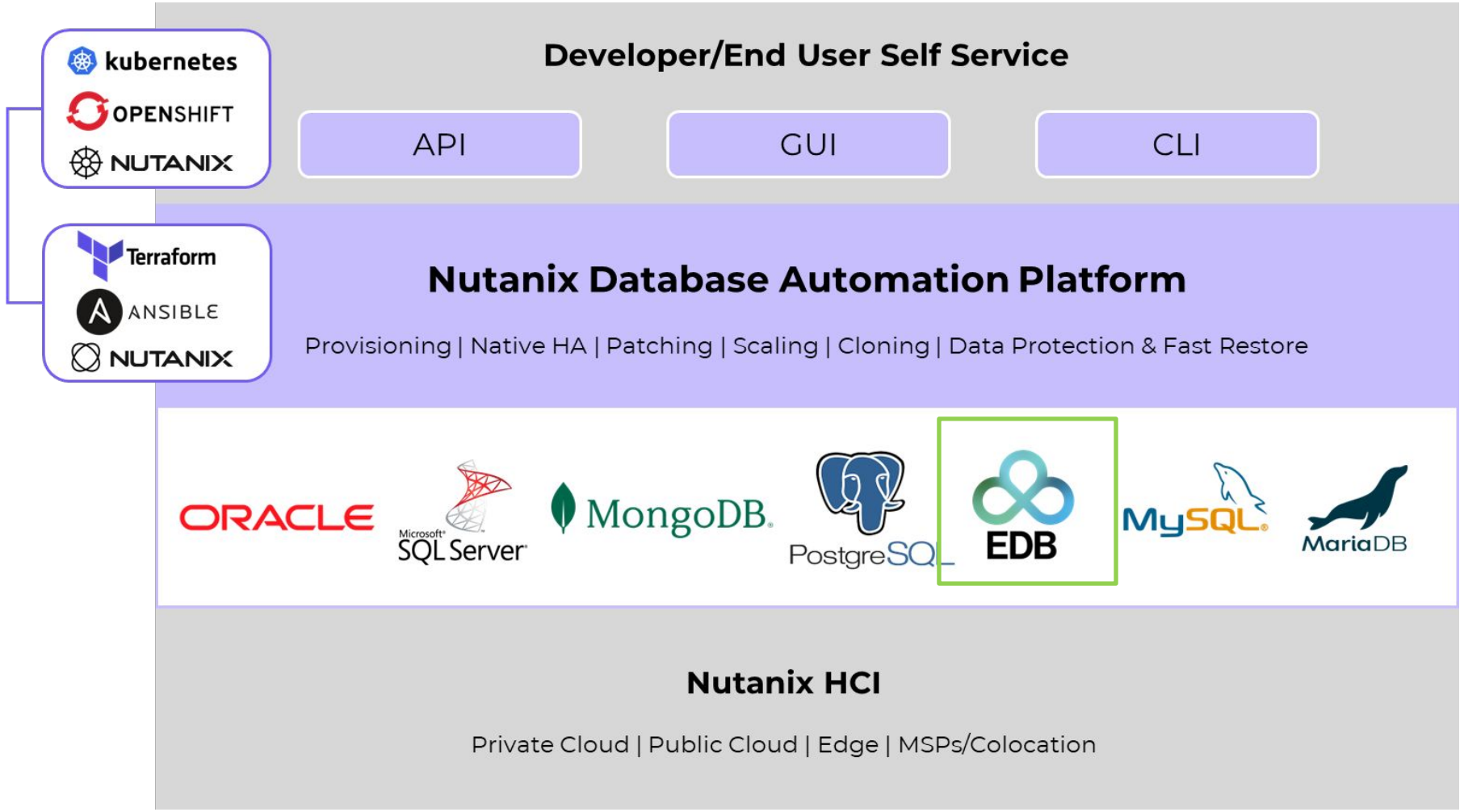
Source: Stack Overflow

Overcoming Modern Database Challenges

Customer pain points, and challenges

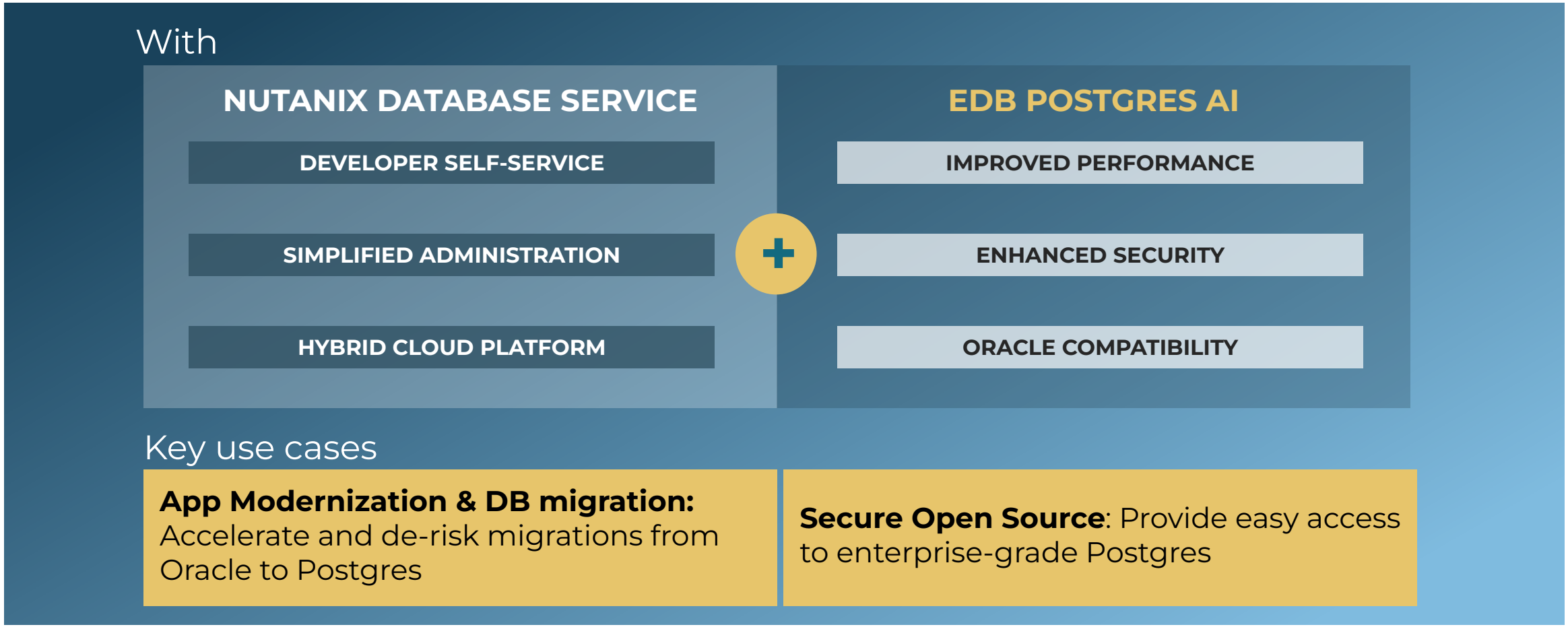
- Complex Infrastructures: Streamlines management across disparate systems with minimal manual intervention.
- Real-Time Data: Facilitates faster and more reliable data integration and analytics.
- Cost Efficiency: Reduces operational costs through automation and simplified management.
- **Migration from Legacy systems:** Mitigate the challenges of migrating from outdated technologies

Nutanix Database Service: Database-as-Code



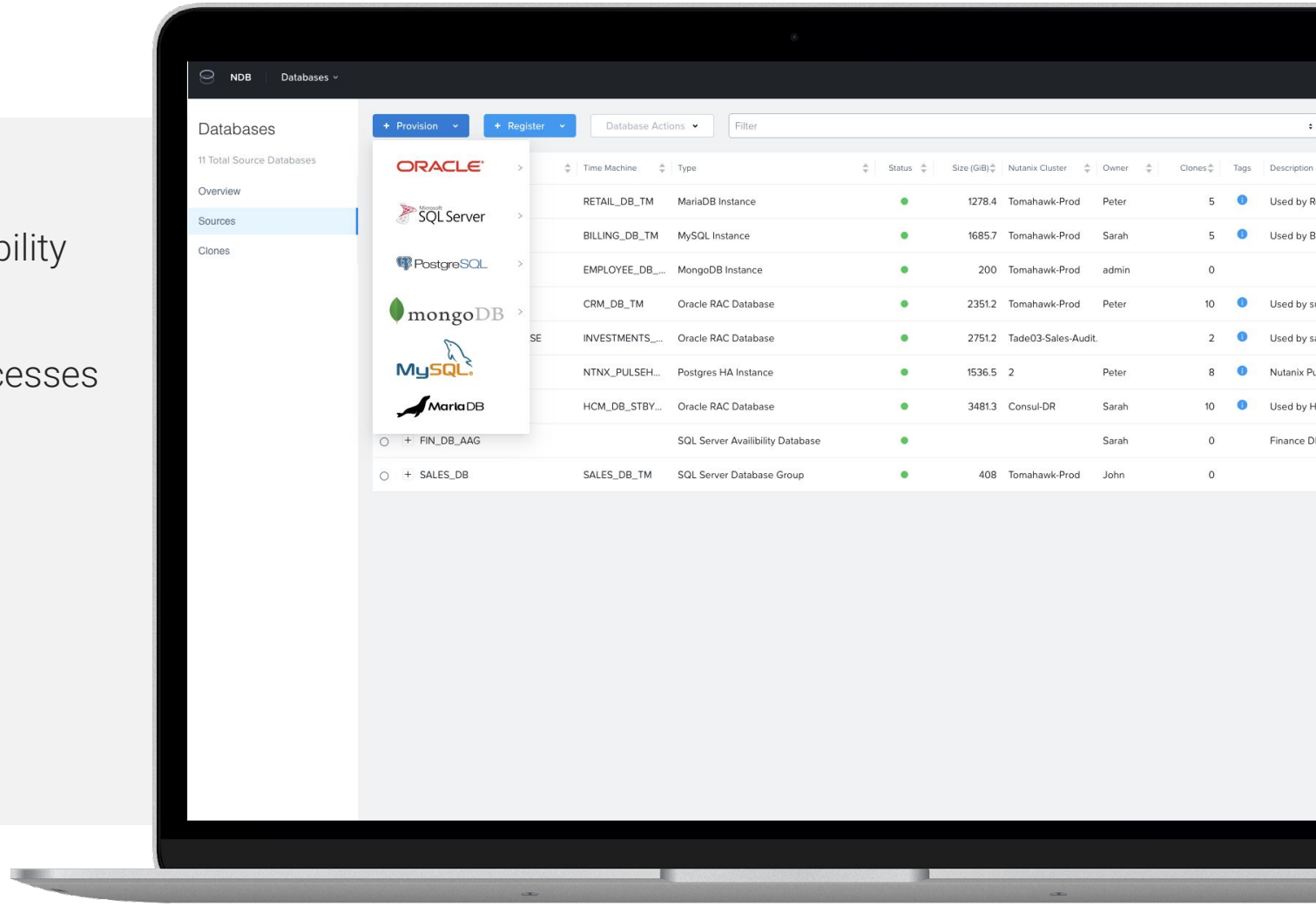
Nutanix Database Service + EDB Postgres AI

The leading hybrid multi-cloud DBaaS, now supporting the leading enterprise-grade Postgres database



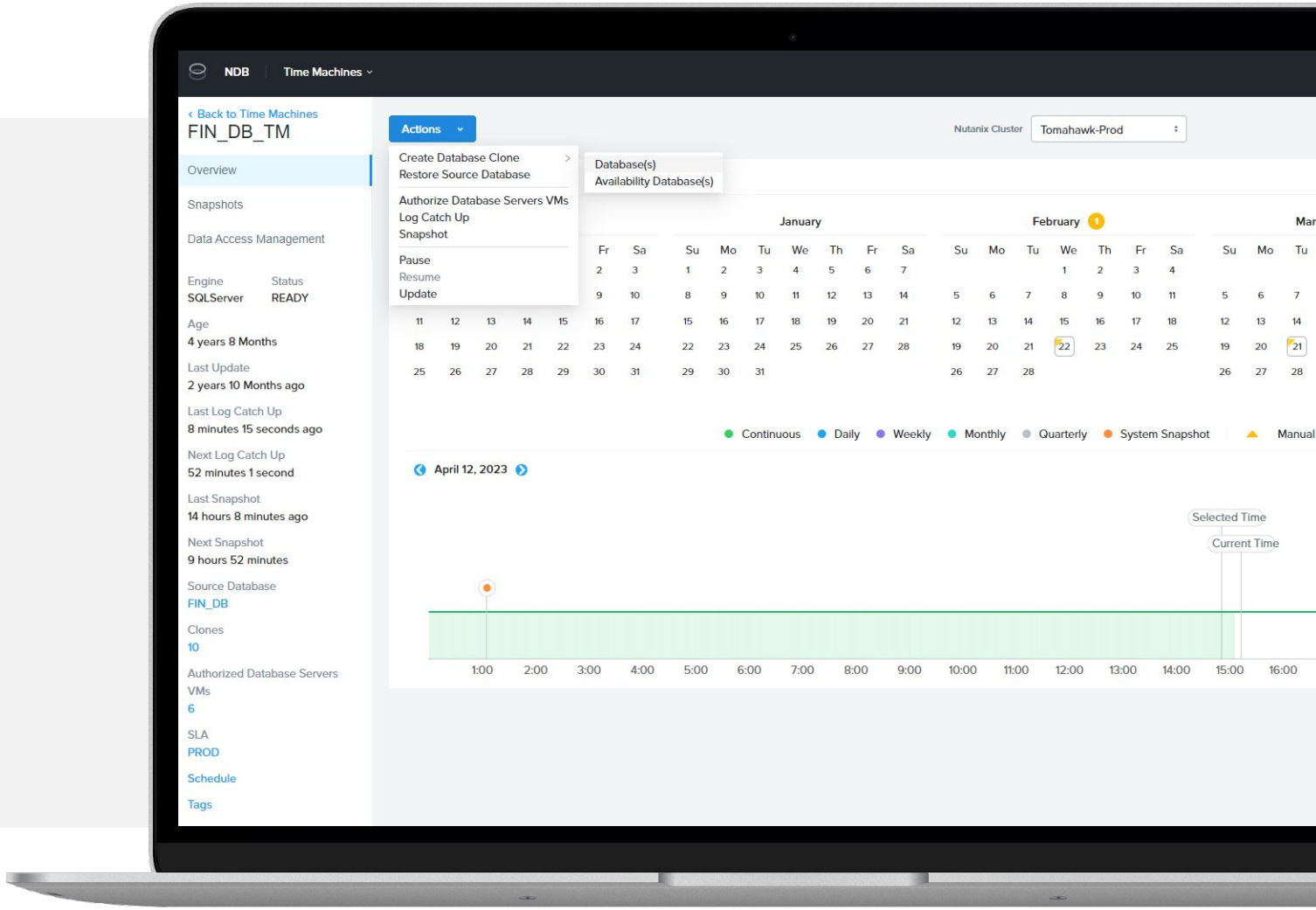
Automate Database Provisioning

- Provision both Single instance and High Availability database clusters with best practices
- Easy integration with existing provisioning processes through REST APIs and script call outs
- Maintain control with ability to customize your database images



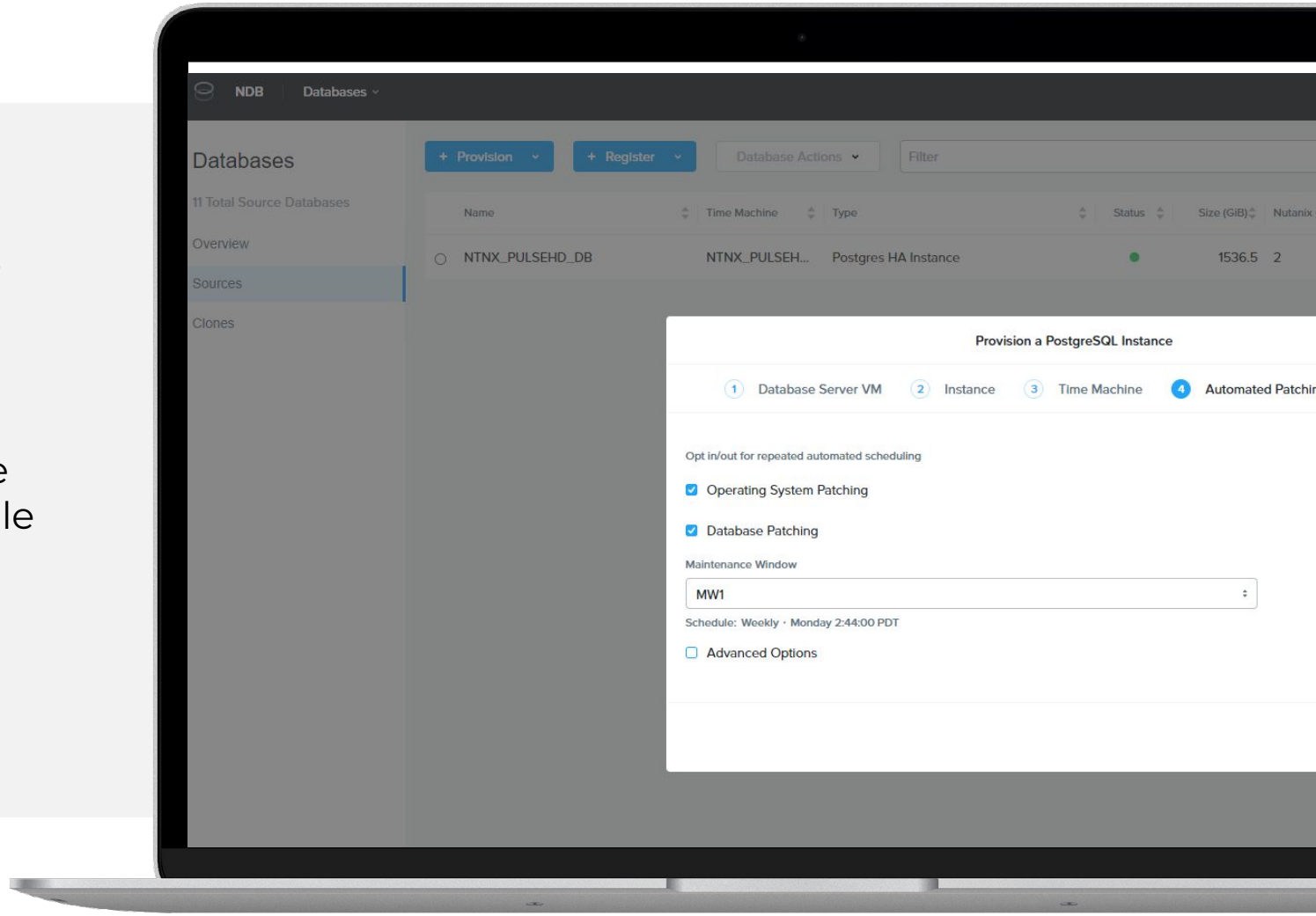
Speed up Database Protection and Cloning

- Automated snapshot protection out-of-the-box with point-in-time-recovery (PITR)
- Create database clones quickly (few minutes)
- Create clones with minimal storage overhead across one or more Nutanix clusters
- Refresh clones to any point in time with an automated schedule



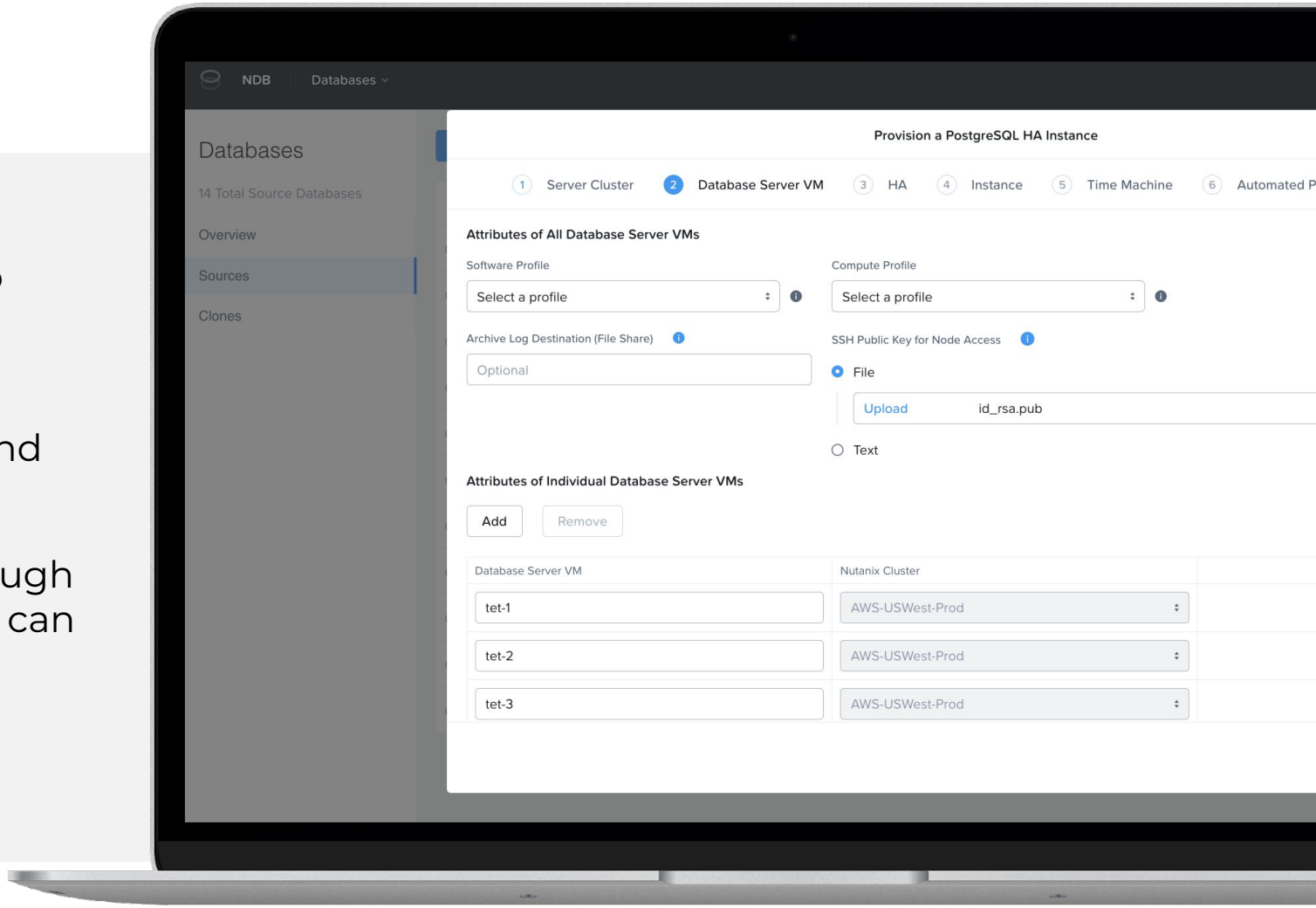
Simplify Patching with Maintenance Windows

- Enable scheduled Operating System and Database patching maintenance windows
- Optionally opt into Operating system or Database patching
- Define a maintenance window and choose which DB servers are a part of that schedule



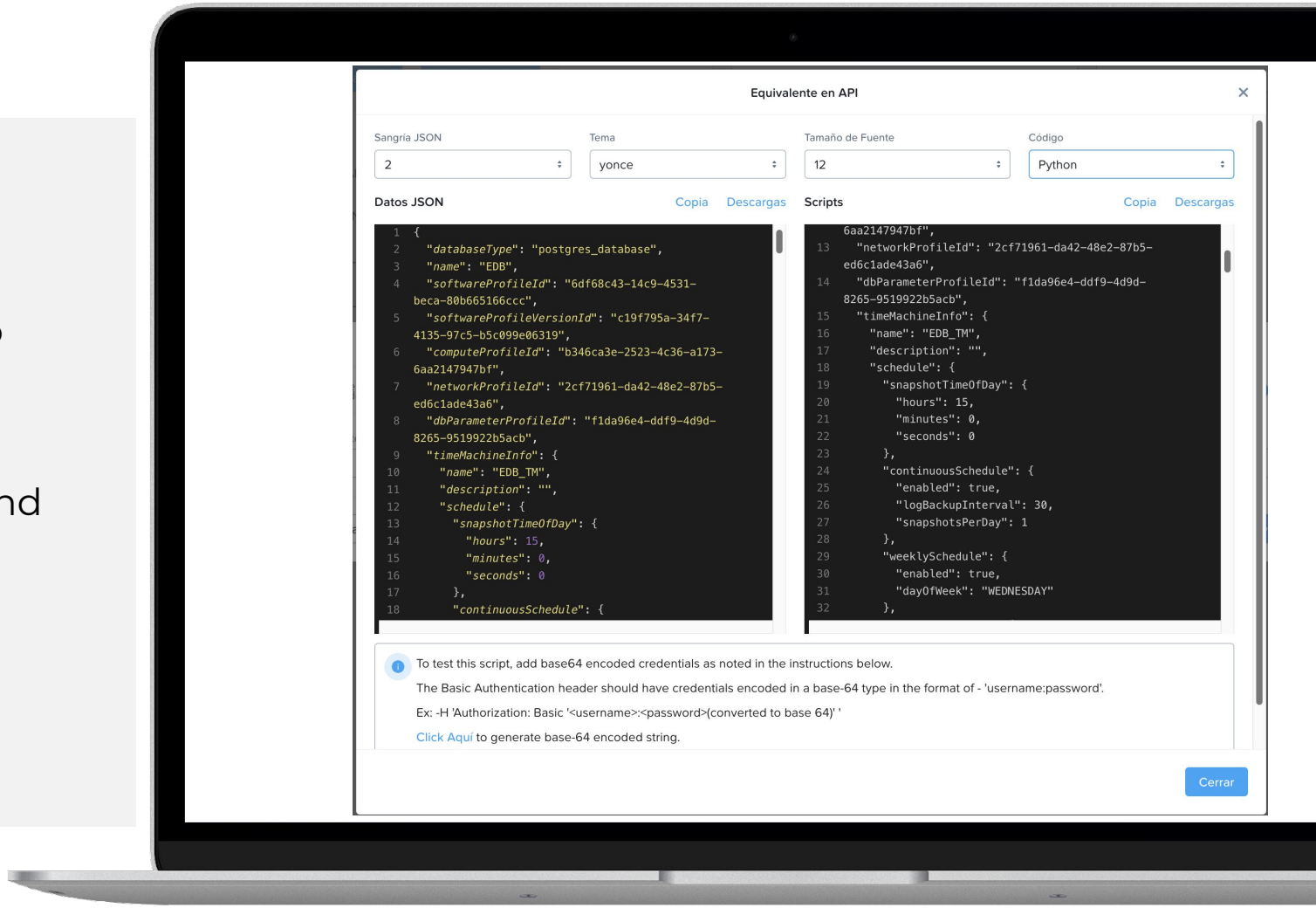
Power Modern Apps with EDB Postgres DBaaS

- Developer self-service for EDB Postgres to accelerate application development
- Simplified EDB Postgres management, including automated patching, cloning, and backups.
- End-to-end support – from hardware through EDB Postgres software - from experts you can trust



Database-as-Code

- Developer self-service for EDB Postgres to accelerate application development
- Simplified EDB Postgres management, including automated patching, cloning, and backups.



Takeaways: With EPAS on Nutanix Database Service, 1+1=3

EDB Postgres Advanced Server

- Industry-leading solution for Oracle-to-Postgres migration, simplifies database transition for developers
- Provides consistency across on-premises and public clouds
- Extends Postgres security with TDE, Data Redaction, EDB Audit, User Profiles, and SQL/Protect

+

Nutanix Database Service

- Industry-leading Database-as-a-Service
- Database provisioning, management, and orchestration solution supports hundreds to thousands of databases—new and existing
- Streamlines database lifecycle management, automation and control, self-service deployment, and database protection

→

EPAS on Nutanix Database Service

- Reduces enterprise Postgres migration efforts using EPAS, coupled with streamlined database provisioning and management from Nutanix Database Service
- Accelerates Postgres adoption for net-new applications and reduces time to market
- Enhances developer experience
- Improves consistency, repeatability, and efficiency of Oracle-compatible Postgres deployment


EDB
Postgres for the AI Generation
EDB Postgres AI Day Zurich | 13 November 2024
Thank you for joining the
EDB Postgres AI Day Zurich