SAS Club 2025:

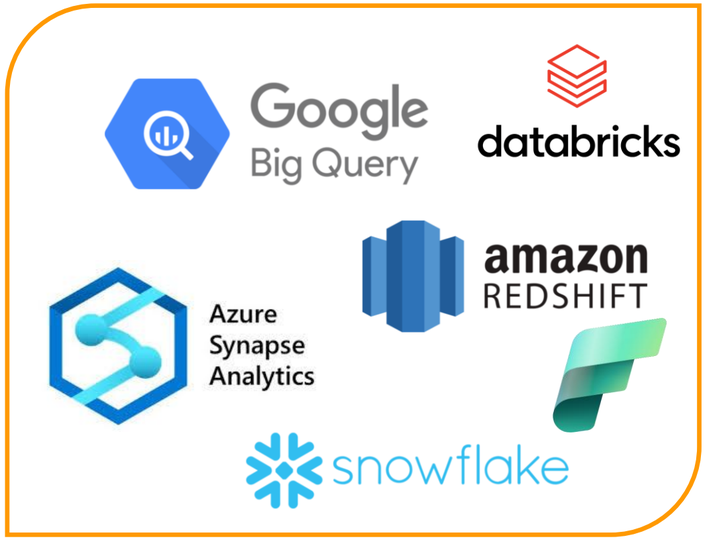
Datenmanagement News
SAS SpeedyStore &
SAS/ACCESS Interface to DuckDB

Wien, 23.10.2025

Data Landscape

2025 and beyond

RDBMS



Cloud Warehouses

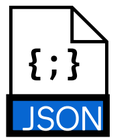
Lakehouse/Delta Lakes

Open File Formats

Continued Evolution

Open **File** Formats

- Columnar storage
- Compression
- Query performance
- Data encoding
- Schema definition



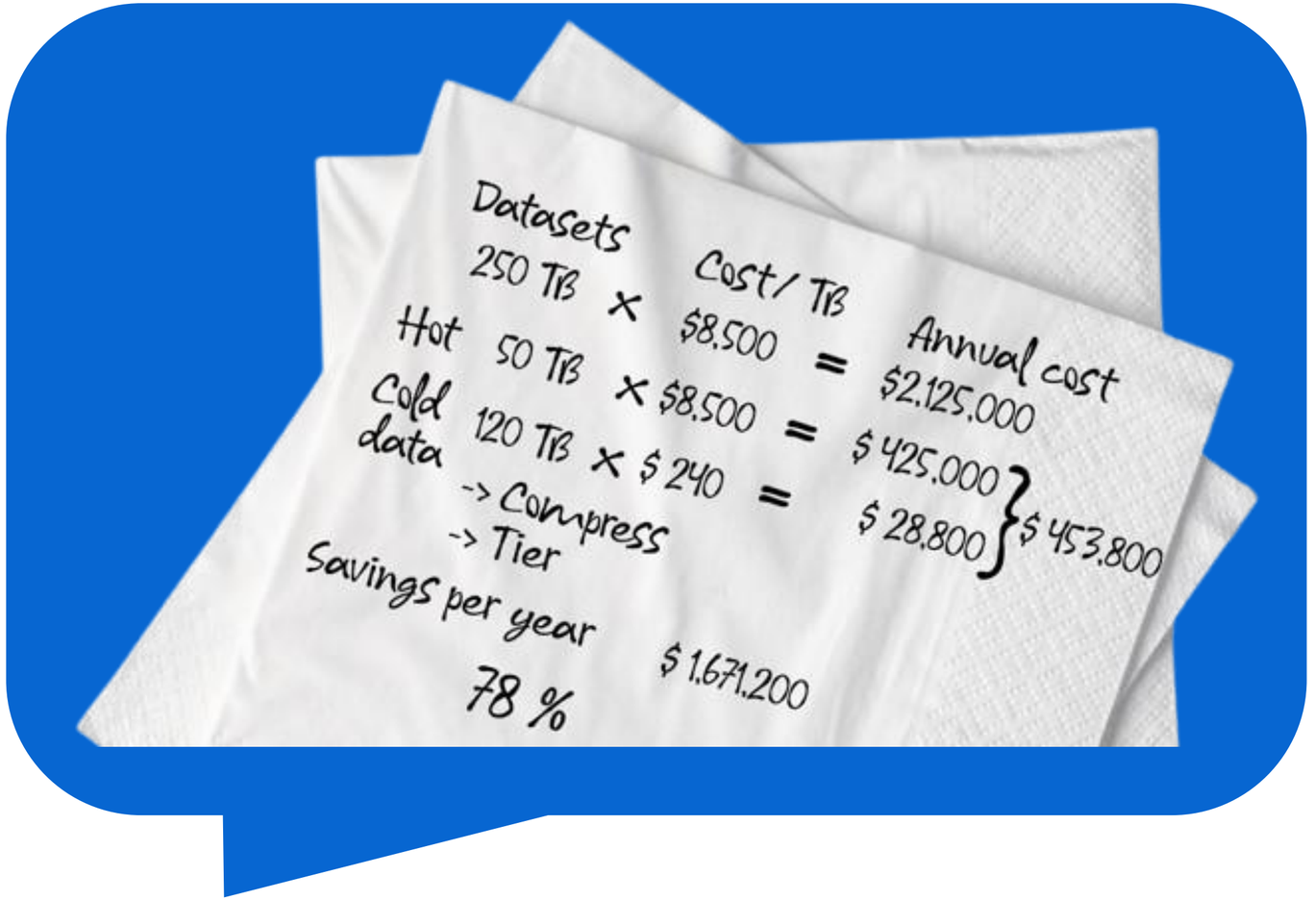
managed by

Open **Table** Formats

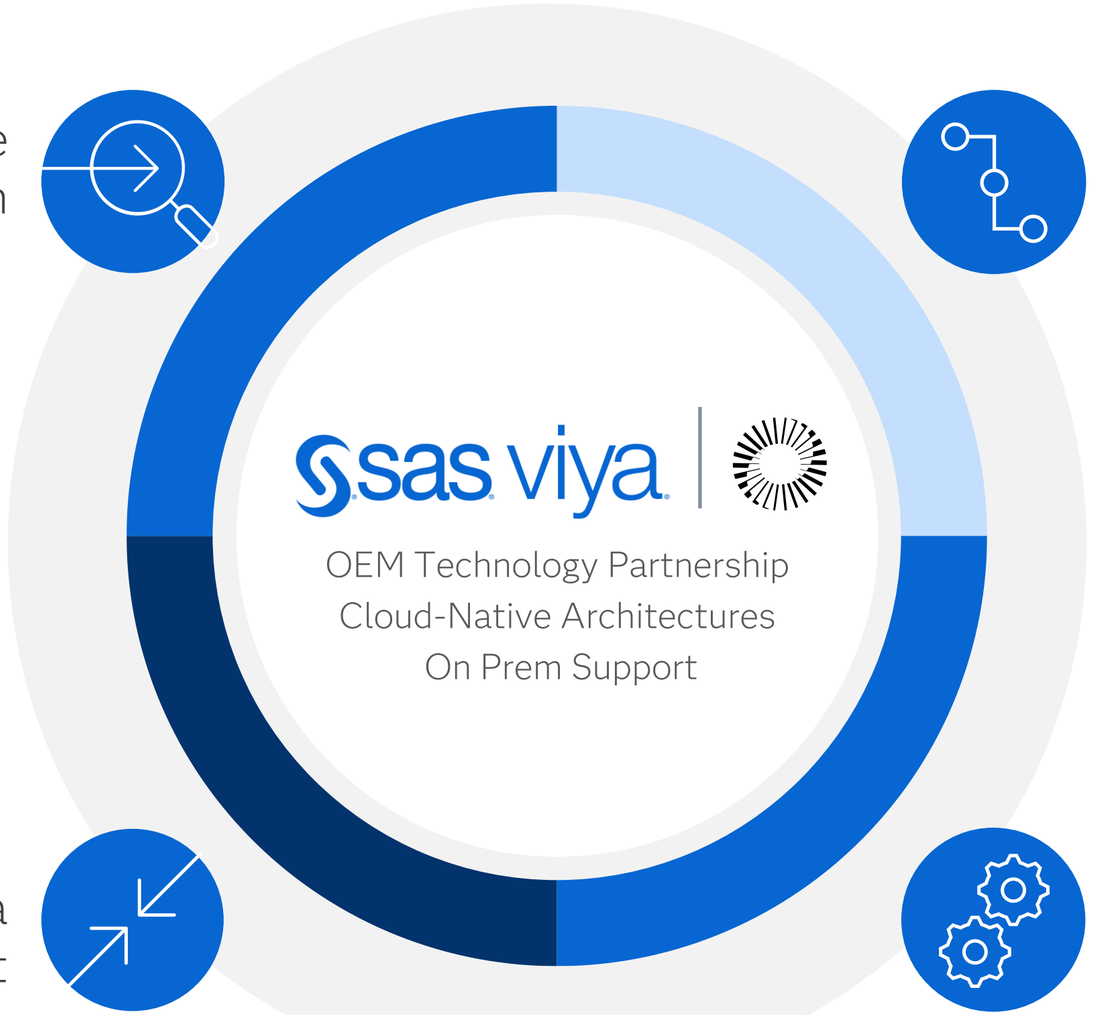
- Metadata management
- Time travel
- Acid transactions
- Abstraction layer
- Schema management

Storage Cost Optimization

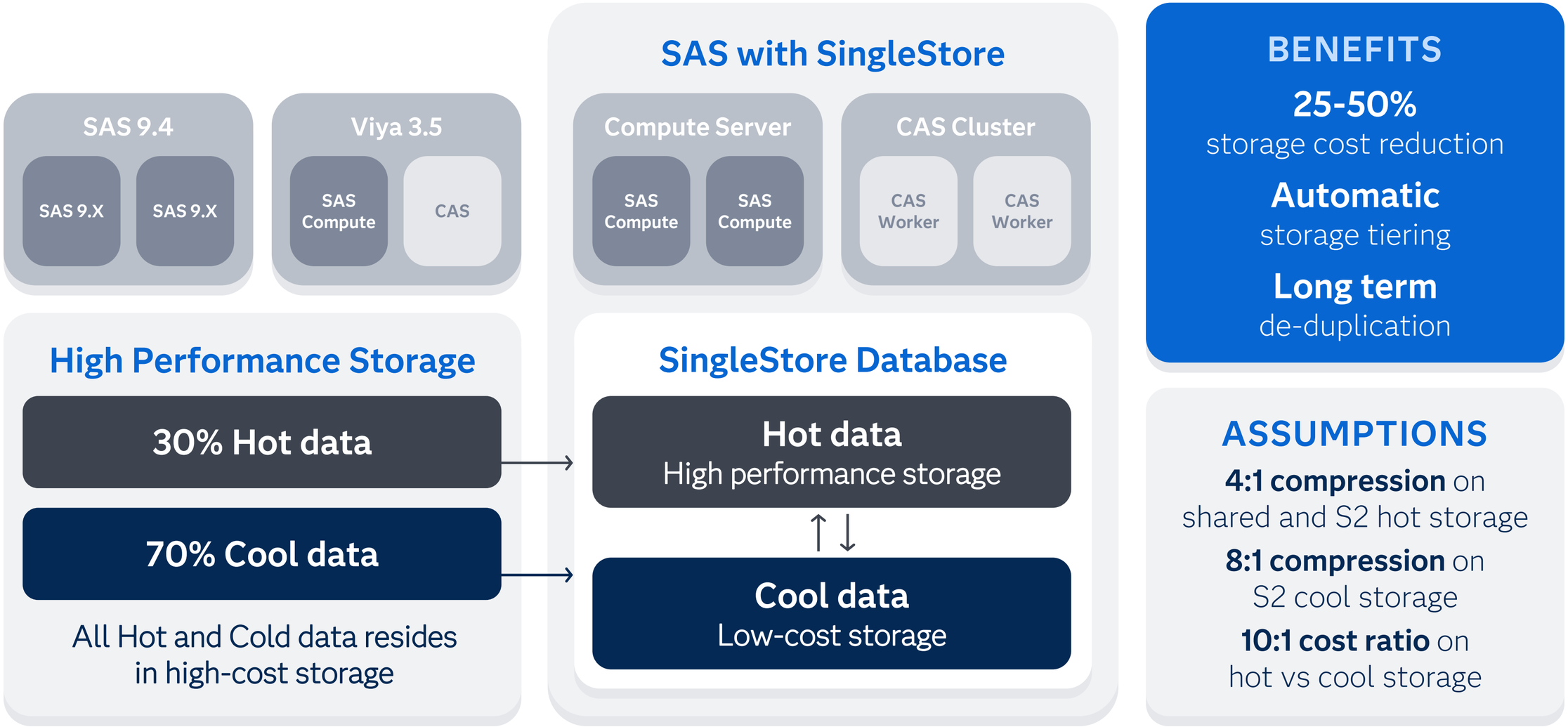
Estimated cost savings with Data Tiering



SAS SpeedyStore



SAS SpeedyStore Data Storage Savings



BENEFITS

25-50%
storage cost reduction

Automatic
storage tiering

Long term
de-duplication

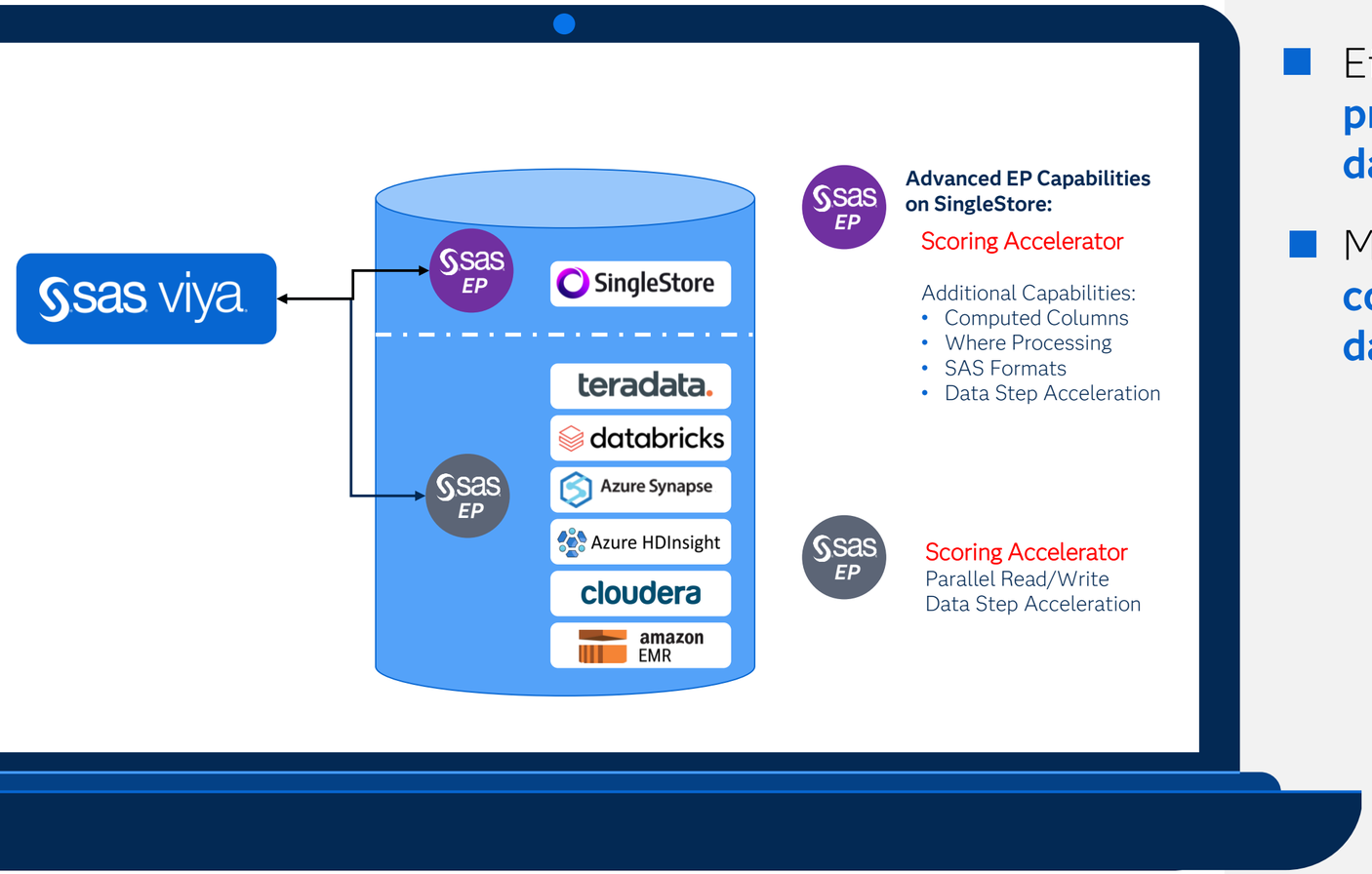
ASSUMPTIONS

4:1 compression on shared and S2 hot storage

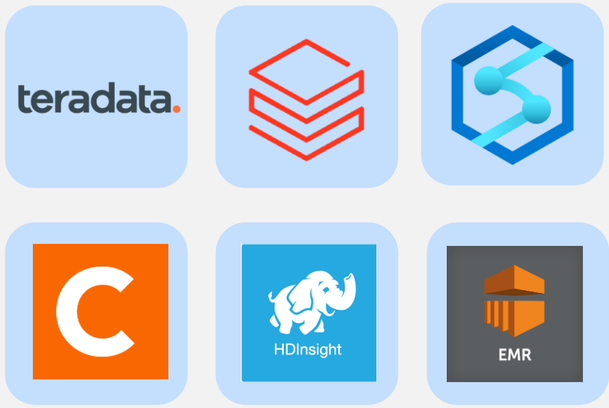
8:1 compression on S2 cool storage

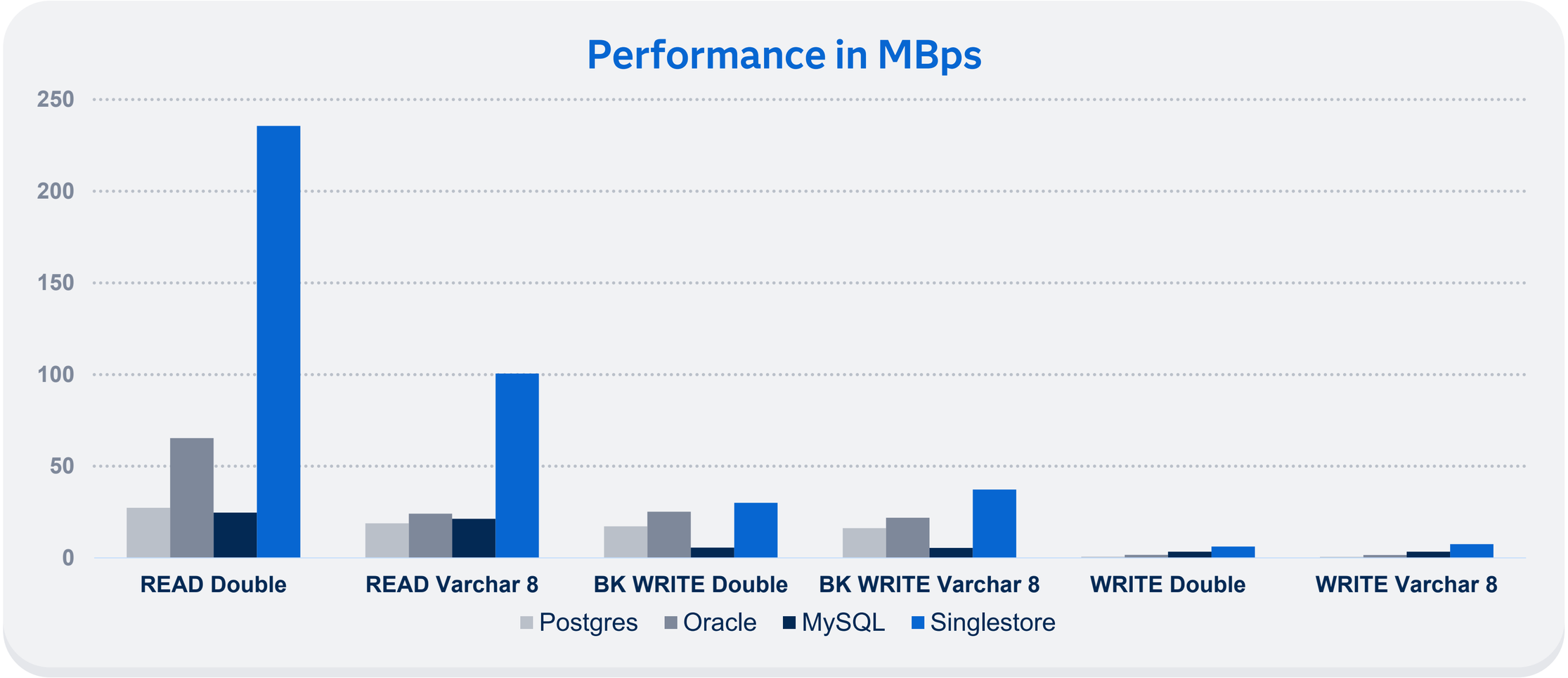
10:1 cost ratio on hot vs cool storage

Deploy & Score Models In-Database with Scoring Accelerator



- Efficient way to **process widespread data**
- Moves the **computation to the data**
- Eliminates massive data movement & **reduce processing times**
- Improves data security as **data never leaves the cluster**





SAS Viya Performance

READ 10 million rows

BULKLOAD WRITE 2.5 million rows

WRITE 100k rows

Support JSON, key-value, time series, vectors

BENEFITS

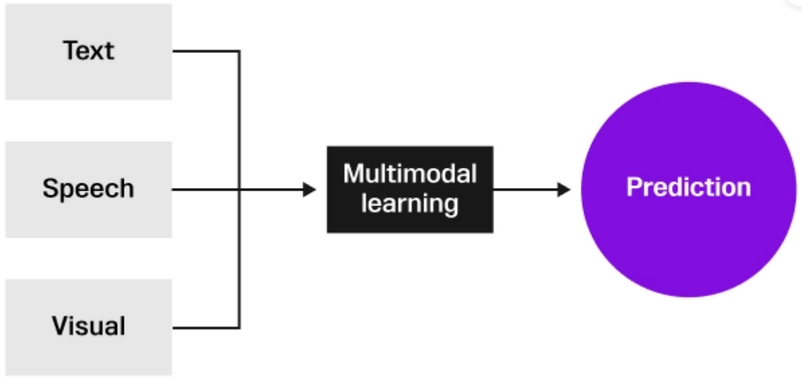
Consolidate your database landscape, only one database to manage

Get access to all data in one database

Reduction and simplification of data pipelines

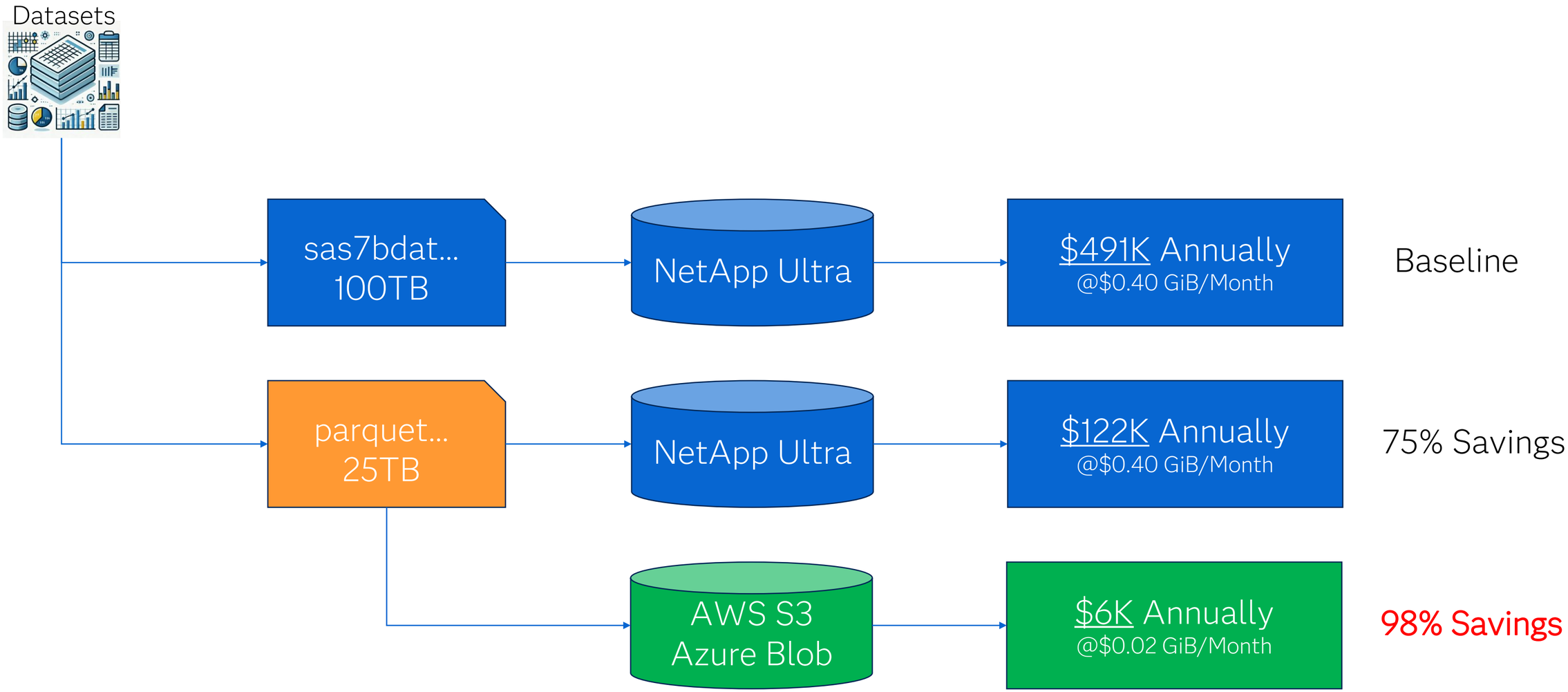
Speeds up the development process

Easy to combine different data types to build multi modal models and applications



Storage Cost Optimization

Example



SAS/Access Interface to DuckDB

- DuckDB is an open source, in-process analytics engine
- Vectorised, optimised query execution on columnar storage
- Requires no server installation and no external dependencies
- Included as part of SAS Viya from 2025.07 release
- Extensive support for open file formats and **open table formats**

SAS/Access Interface to DuckDB

Performance (27GB dataset)

```
PROC SQL;
  SELECT
    passenger_count,
    payment_type,
    count(*)           AS num_trips,
    avg(trip_distance) AS avg_distance,
    avg(fare_amount)   AS avg_fare,
    avg(tip_amount)    AS avg_tip
  FROM
    saslib.yellow_tripdata_2011
  WHERE
    passenger_count is not NULL AND
    passenger_count > 0 AND
    passenger_count < 5 AND
    trip_distance < 100 AND
    trip_distance > 0
  GROUP BY
    passenger_count, payment_type
  ORDER BY
    payment_type, passenger_count;
QUIT;
```

```
NOTE: PROCEDURE SQL used (Total process time):
      real time           1:56.27
      cpu time            1:22.98
```

```
PROC SQL;
  SELECT
    passenger_count,
    payment_type,
    count(*)           AS num_trips,
    avg(trip_distance) AS avg_distance,
    avg(fare_amount)   AS avg_fare,
    avg(tip_amount)    AS avg_tip
  FROM
    duklib.'2011/*.parquet'n
  WHERE
    passenger_count is not NULL AND
    passenger_count > 0 AND
    passenger_count < 5 AND
    trip_distance < 100 AND
    trip_distance > 0
  GROUP BY
    passenger_count, payment_type
  ORDER BY
    payment_type, passenger_count;
QUIT;
```

```
NOTE: PROCEDURE SQL used (Total process time):
      real time           2.54 seconds
      cpu time            26.24 seconds
```