

VERTICA

What is new in Vertica 9.x

Gianluigi Vigano'
Vertica System Engineer
gianluigi.vigano@microfocus.com

Because is An Open Architecture Integrated with Rich Ecosystem

Data Transformation



Messaging



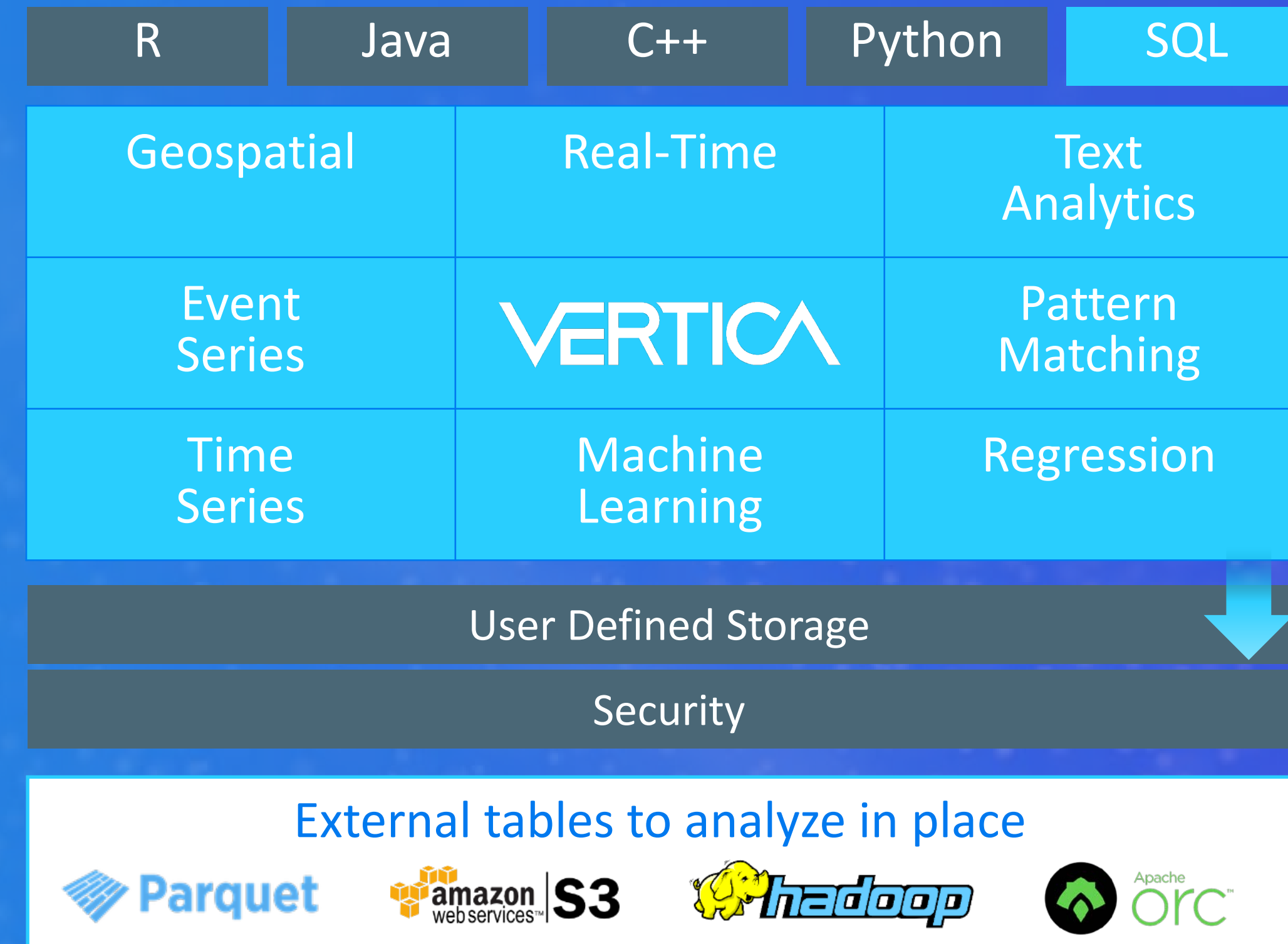
ETL



USER
DEFINED
LOADS



User-Defined Functions



ODBC
JDBC
OLEDB



BI & Visualization

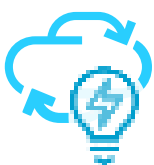


Ongoing Commitment to Innovation

Foundation

- Columnar Store
- Aggressive Data Compression
- MPP Architecture
- HA Architecture
- ANSI SQL Compliant
- Java, Python, R APIs
- ACID Compliance
- No Single Point of Failure
- Management Console
- Database Designer
- Projections and Optimizations

Mission



Multi-Cloud/Deploy Anywhere



Advanced, In-Database Analytics

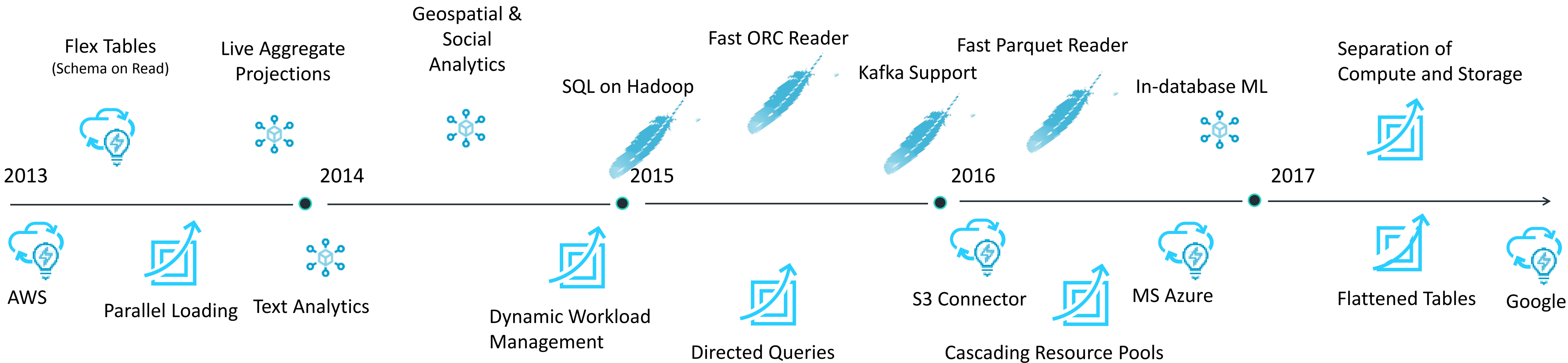


Performance at Exabyte Scale



Open Source Innovation

Innovation Timeline

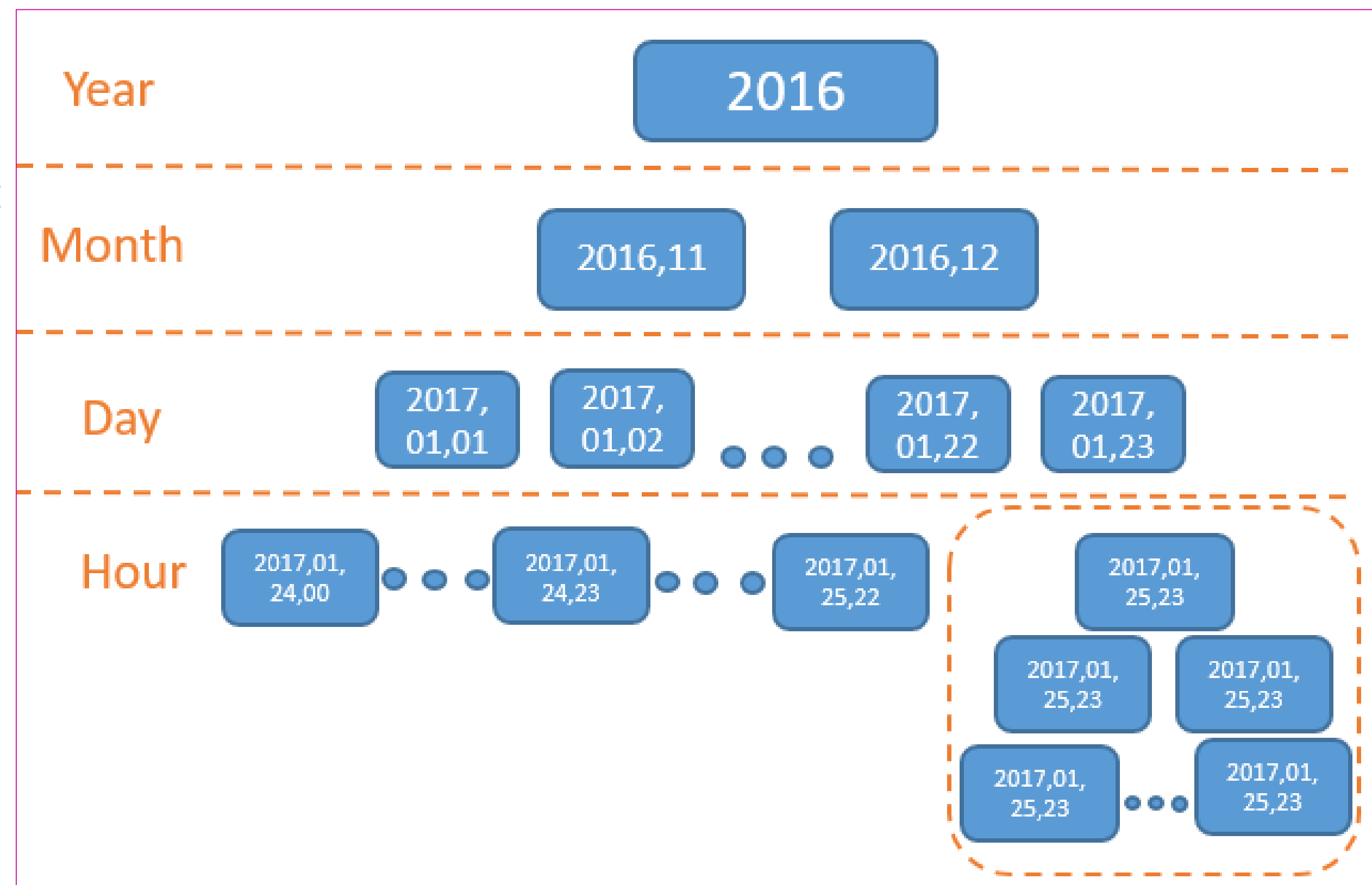


9.X Features Enhancements

- Hierarchical Partition
- Flattened Tables
- Directed Queries
- Python UDX
- vPython

Hierarchical partition management

- **Case:** We want to partition the store_orders table by day and we want to keep six years of historical data
- **Hierarchical Partition:** fine grain partitions for data that is more frequently queried and coarser partitions for less valuable data.
- **Example:**
 - organizes a table's date partitions into a hierarchy of groups:
 - oldest date partitions → grouped by month
 - most recent date partitions → grouped by day.
 - Grouping is dynamic

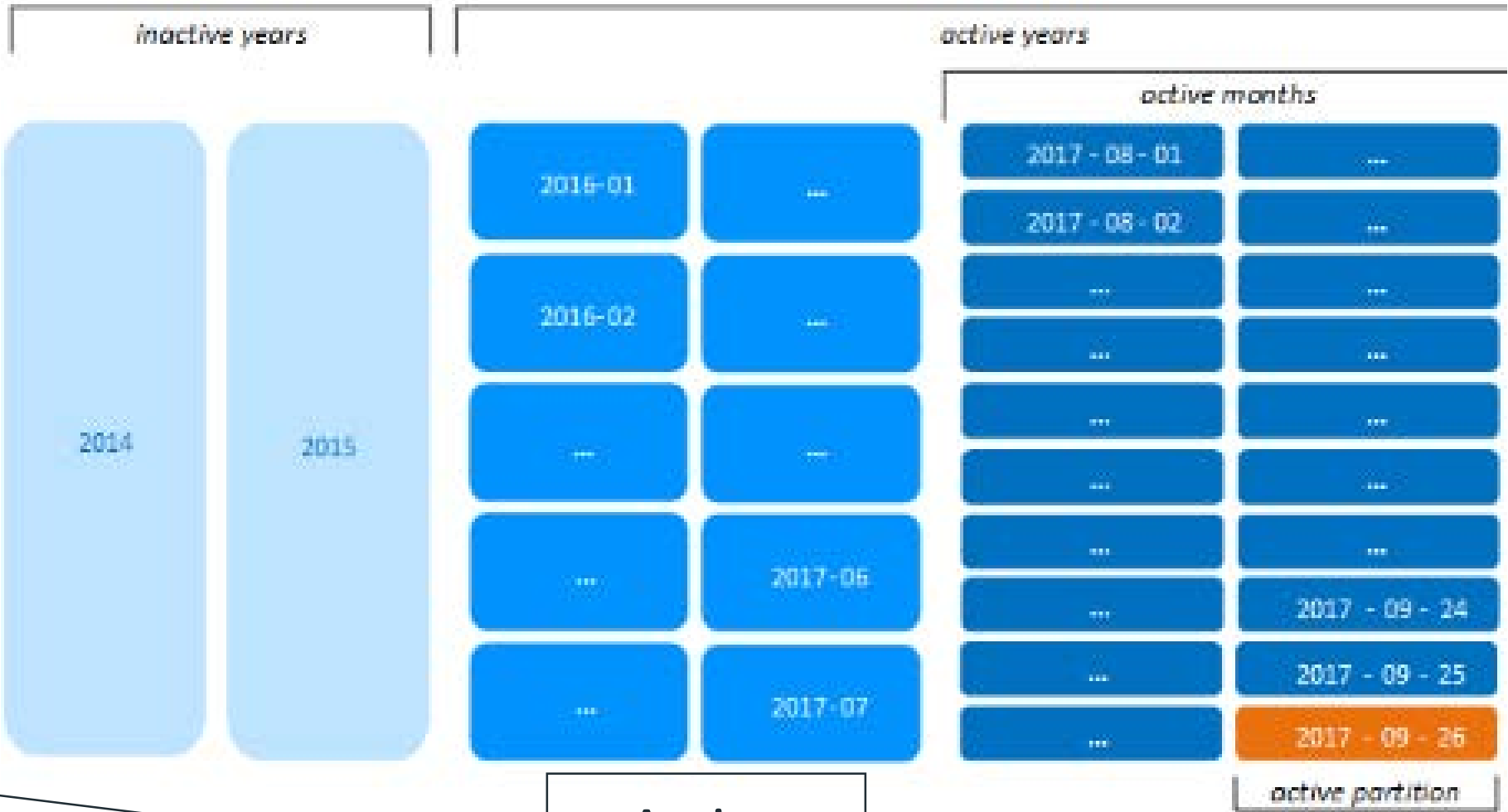


Hierarchical partition management

Is it magic?

```
=> CREATE TABLE public.store_orders
(
  order_no int,
  order_date timestamp NOT NULL,
  shipper varchar(20),
  ship_date date
)
```

```
PARTITION BY order_date::DATE
  GROUP BY CALENDAR_HIERARCHY_DAY(order_date::DATE, 2, 2);
```



Active months

Active years

If the current date is **2017-09-26**

CALENDAR_HIERARCHY_DAY resolves active years and active months to the following date spans:

- Active months: **2017-08-01 to 2017-09-31**. Partitions in active months are grouped into daily ROS containers.
- Active years: **2016-01-01 to 2017-12-31**.

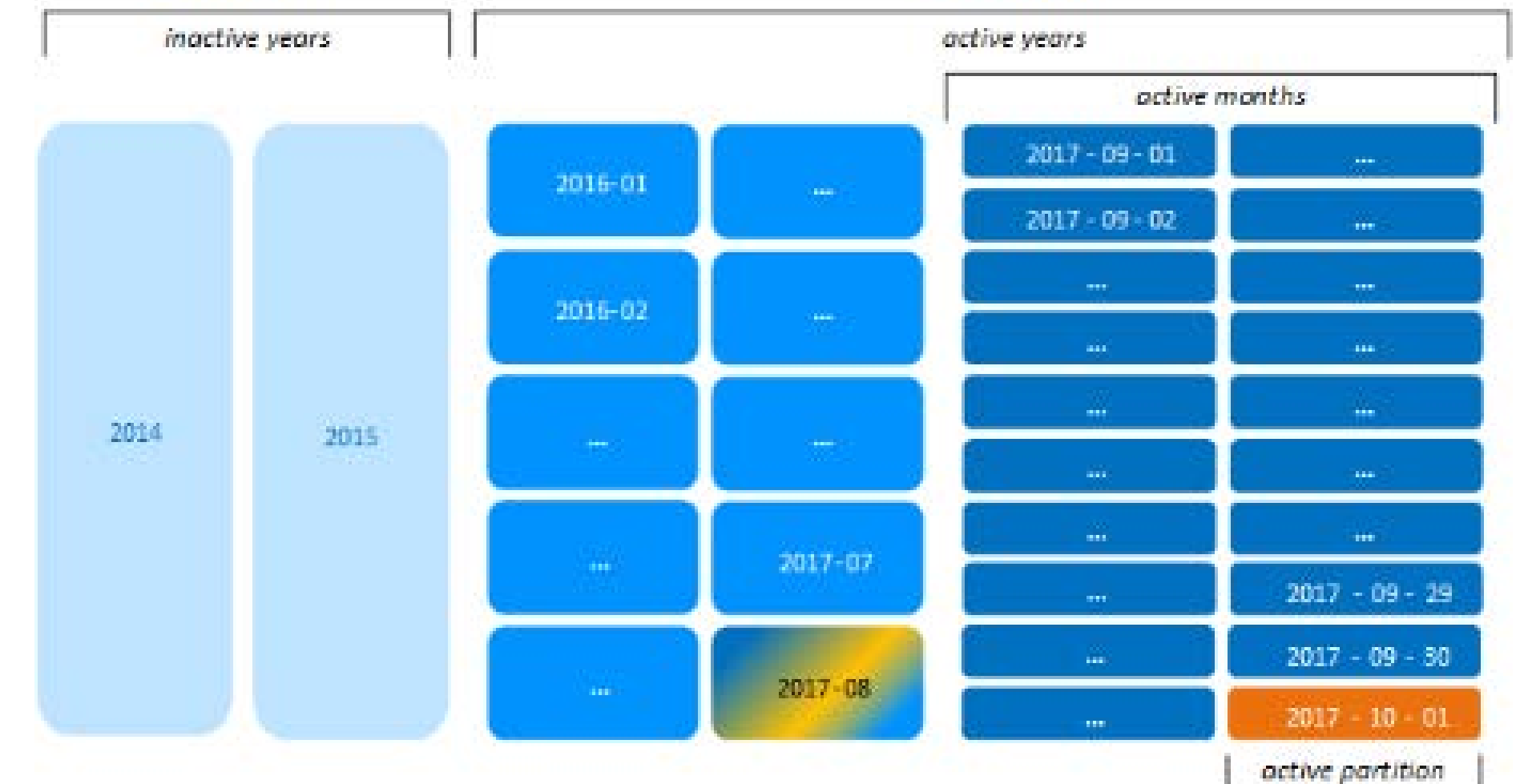
Partitions in active years are grouped into monthly ROS containers or are grouped into daily ROS containers.

Partitions from earlier years are regarded as inactive and grouped into yearly ROS

Hierarchical partition management

Is it magic?

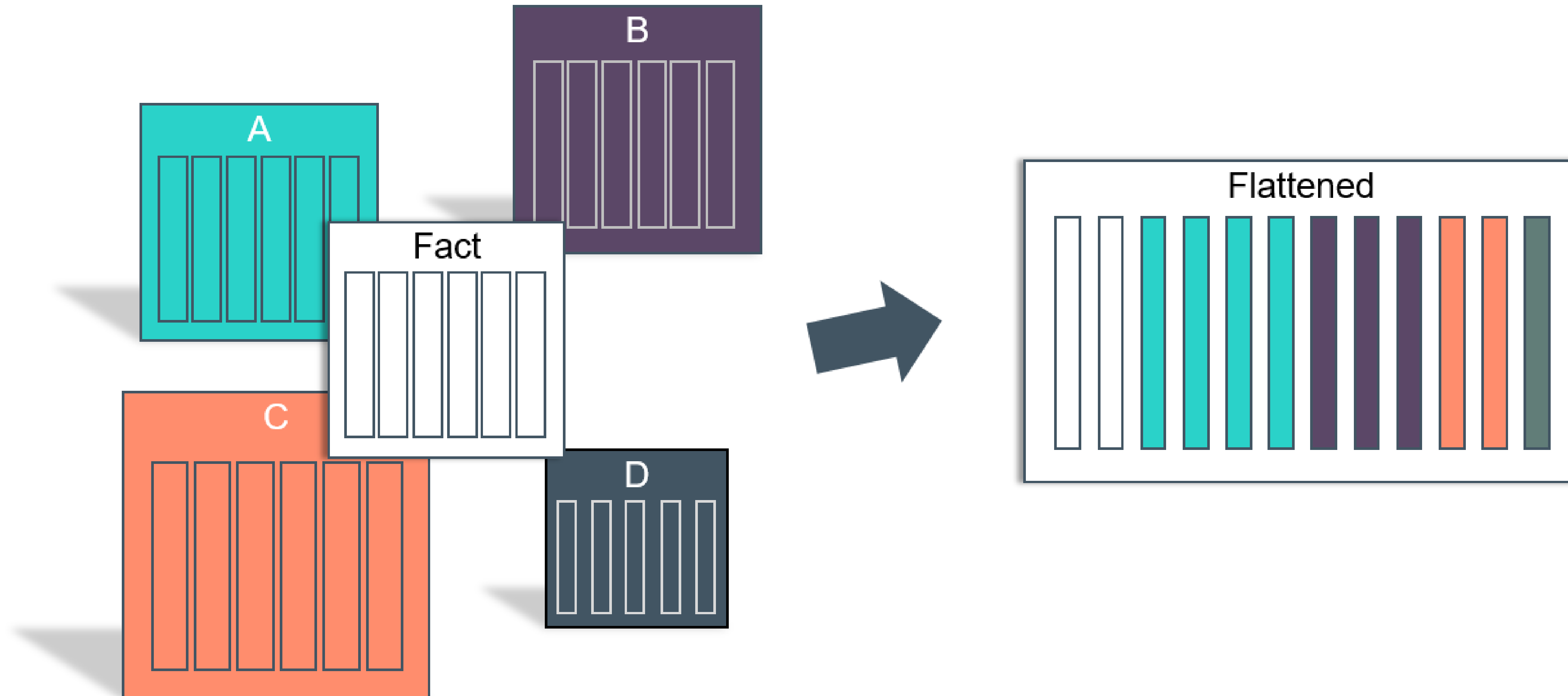
```
=> CREATE TABLE public.store_orders
(
  order_no int,
  order_date timestamp NOT NULL,
  shipper varchar(20),
  ship_date date
)
PARTITION BY order_date::DATE
GROUP BY CALENDAR_HIERARCHY_DAY(order_date::DATE, 2, 2);
```



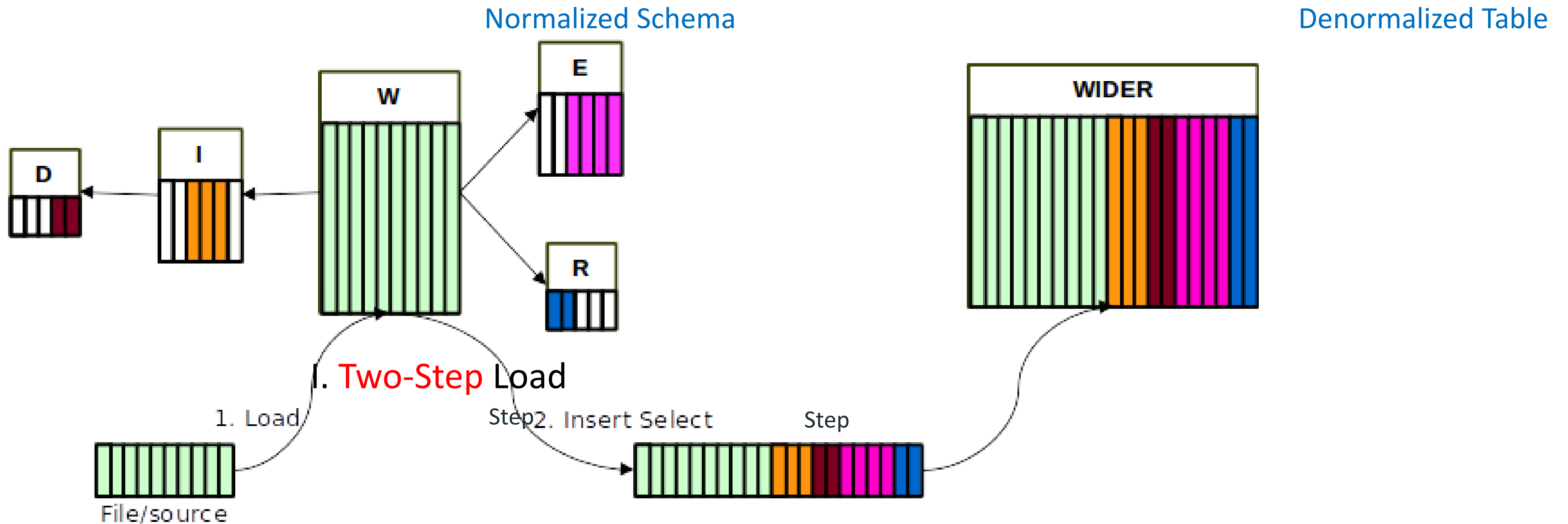
As the calendar advances, the Tuple Mover reevaluates the partition group keys, and moves partitions as needed to different ROS containers.

Thus, given the previous example, **on 2017-10-01** the Tuple Mover creates a monthly ROS container for July partition. All partition keys between **2017-07-01** and **2017-07-31** are grouped in the new ROS container 2018-06:

Flattened Table

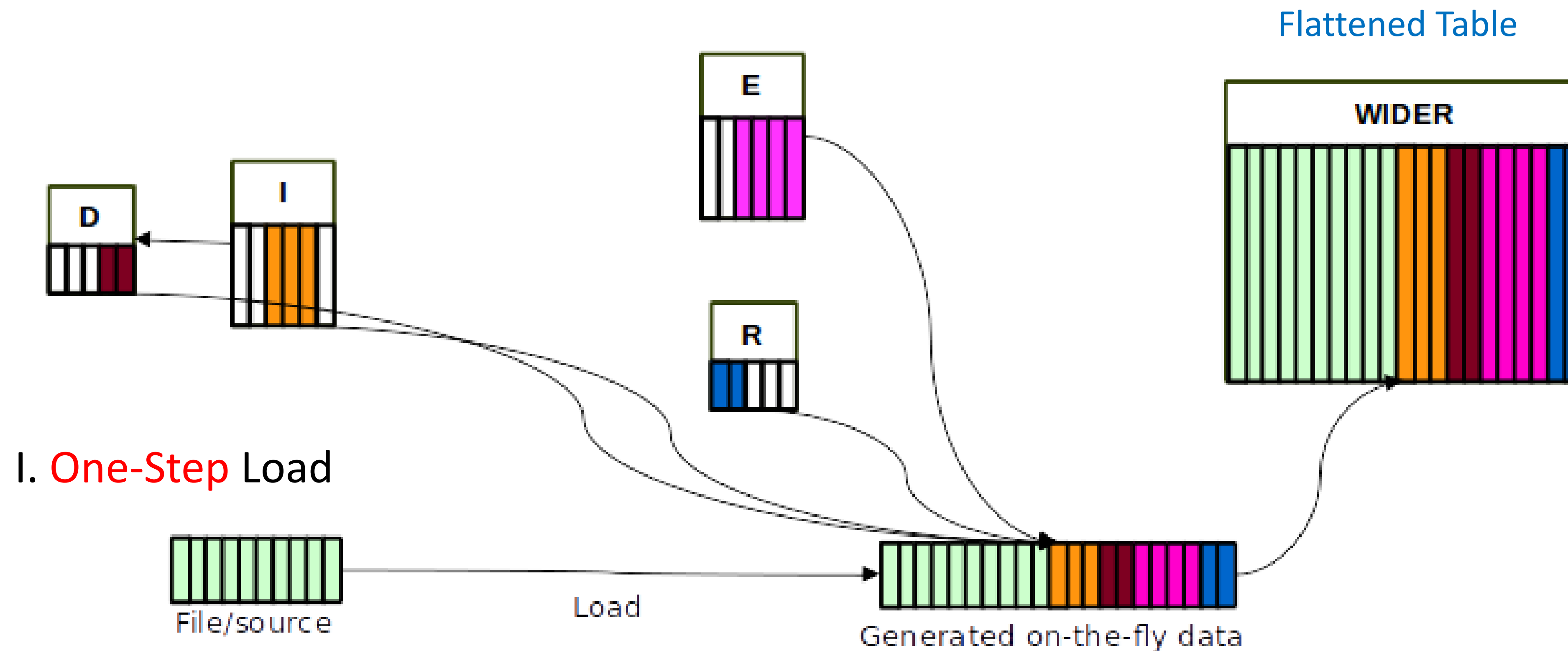


Manual denormalization



- II. **Manually update** table **WIDER** if any of tables **D, E, I, R** gets updated
Many of our customers even choose to **rebuild** table **WIDER** from scratch

Flatten out a table



- II. New syntax **refresh_columns** to update **WIDER** if any of **D**, **E**, **I**, **R** gets updated
- III. Easy and effective **add** and **drop** any columns

So What is a Vertica Flattened Table?

Vertica Goes Beyond That

- **Flattened Table** = **Table** with normalized and denormalized columns
 - A **column** of a Flattened Table is
 - Defined as an **expression** or a query of many tables, Q
 - Added and dropped **as needed**, at any point in time
 - Populated when
 - Data of other columns loaded, or
 - The column is added, or
 - The column is refreshed
- There is **no overhead** on performance of the flattened tables when the values in the source are changing

Create a Flattened Table

Snippet of a CREATE TABLE Statement

- *Denormalized column* is defined using **DEFAULT** and **SET USING** key words

```
CREATE TABLE flattenedTab
(
  col1      .....
  col2      .....
  colX      int DEFAULT ( SELECT val FROM dim WHERE (f_dkey = d_key) )
              SET USING ( SELECT val FROM dim WHERE (f_dkey = d_key) )
  coln      .....
  .....
);
```

A real case: result

Query	execution time (sec)	
	Star Schema	Flat Table
single run	2.13	0.33
10q in parallel (avg)	5.95	0.36
20q in parallel (avg)	8.84	0.62
40q in parallel (avg)	17.29	1.17
50q in parallel (avg)	27.01	1.81
100q in parallel (avg)	41.15	3.08

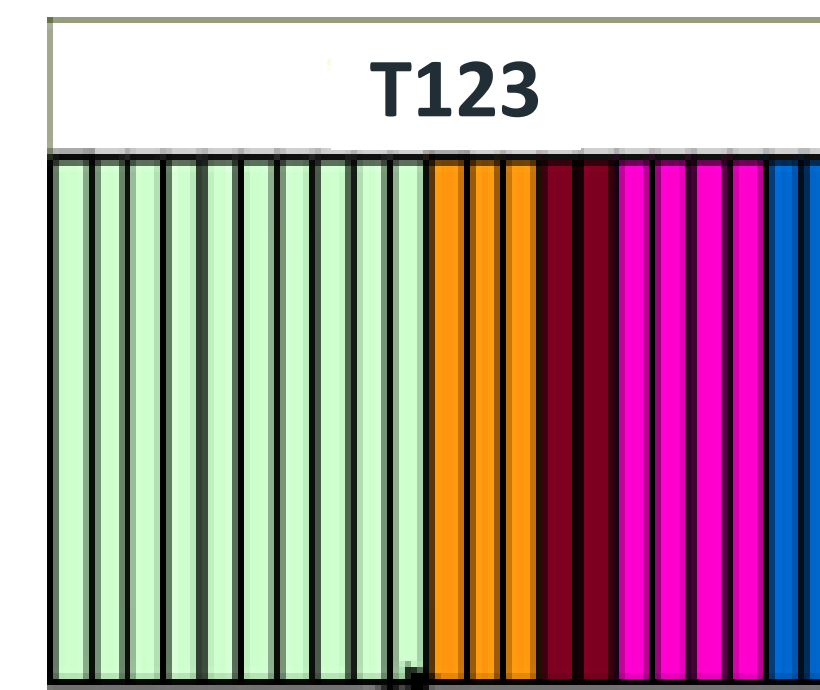
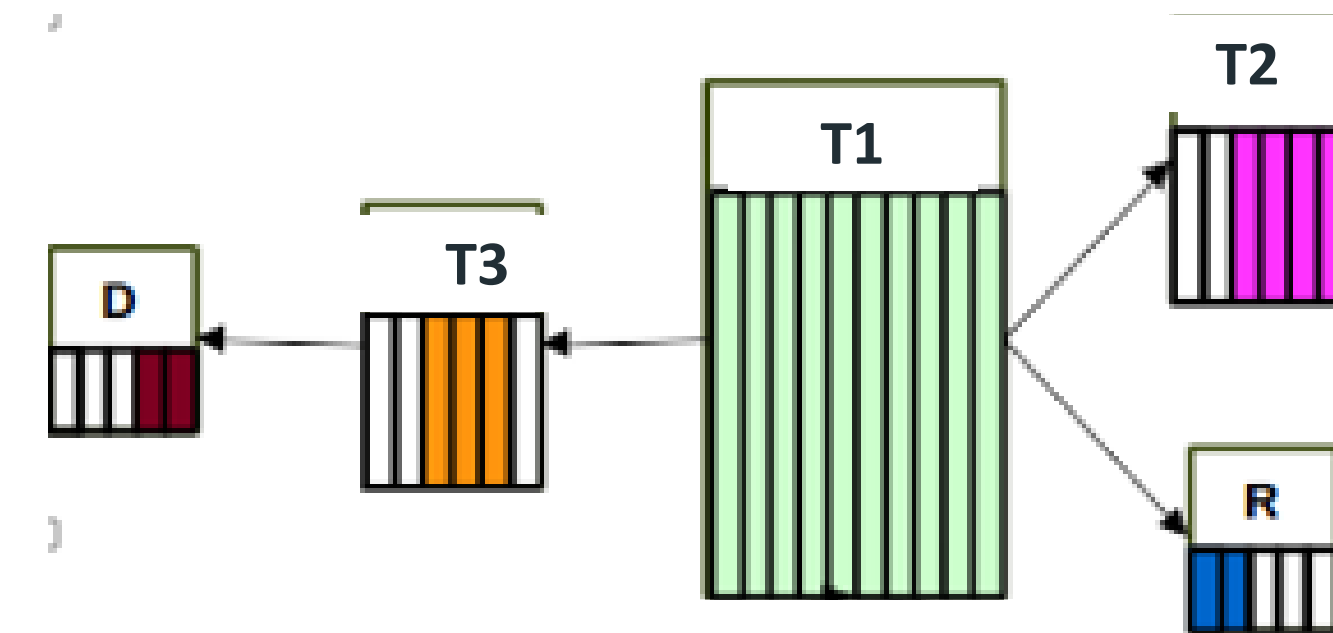
Directed Queries

- Used for:
 - Achieve Stable Query Execution Performance
 - Tell the optimizer you have better plans
 - Flexible Query Execution Plans

Directed Queries

Rewriting join queries

```
SELECT T1.a, T2.b, T3.c  
FROM  
  T1 JOIN T2  
    ON T1.x = T2.x  
  JOIN T3 ON T2.y = T3.y  
  WHERE T3.description ILIKE '%red%';
```



Directed Queries

Rewriting join queries

```
SAVE QUERY SELECT T1.a, T2.b, T3.c
```

```
FROM
```

```
  T1 JOIN T2
```

```
    ON T1.x = T2.x
```

```
  JOIN T3 ON T2.y = T3.y
```

```
    WHERE T3.description ILIKE '%red%';
```

```
CREATE DIRECTED QUERY CUSTOM 'FindReds'
```

```
SELECT a, b, c
```

```
FROM
```

```
  T123
```

```
    WHERE description ILIKE '%red%';
```

```
ACTIVATE DIRECTED QUERY FindReds
```

When the directed query is active, the query optimizer maps all queries that match the original input format to the directed query. Instead of executing the original query, Vertica uses the annotated query, which queries flattened table T123

Directed Queries

Rewriting join queries

```
SELECT T1.a, T2.b, T3.c FROM T1 JOIN T2 ON T1.x = T2.x JOIN T3 ON T2.y = T3.y
WHERE T3.description ILIKE '%green%';
```

```
SAVE QUERY SELECT T1.a, T2.b, T3.c
FROM
    T1 JOIN T2
        ON T1.x = T2.x
    JOIN T3 ON T2.y = T3.y
    WHERE T3.description ILIKE '%any-string%'
/*+IGNORECONST(1)*/;
```

```
CREATE DIRECTED QUERY CUSTOM 'FindAnyColor'
SELECT a, b, c
FROM
    T123
    WHERE description ILIKE '%any-string%';
/*+IGNORECONST(1)*/;
```

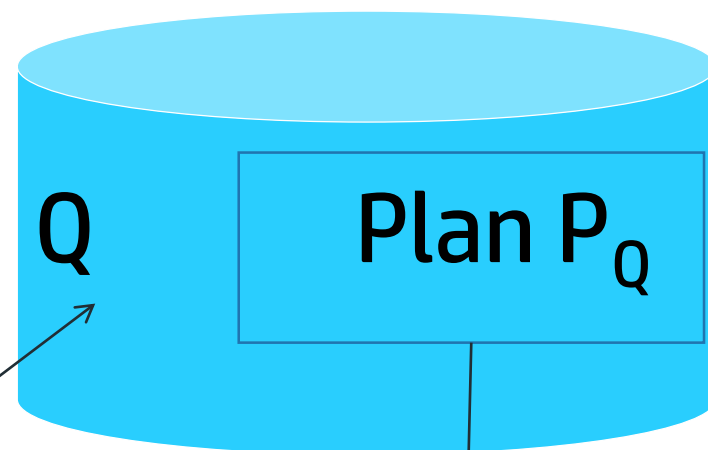
```
ACTIVATE DIRECTED QUERY FindAnyColor
```


Directed Queries

Summary

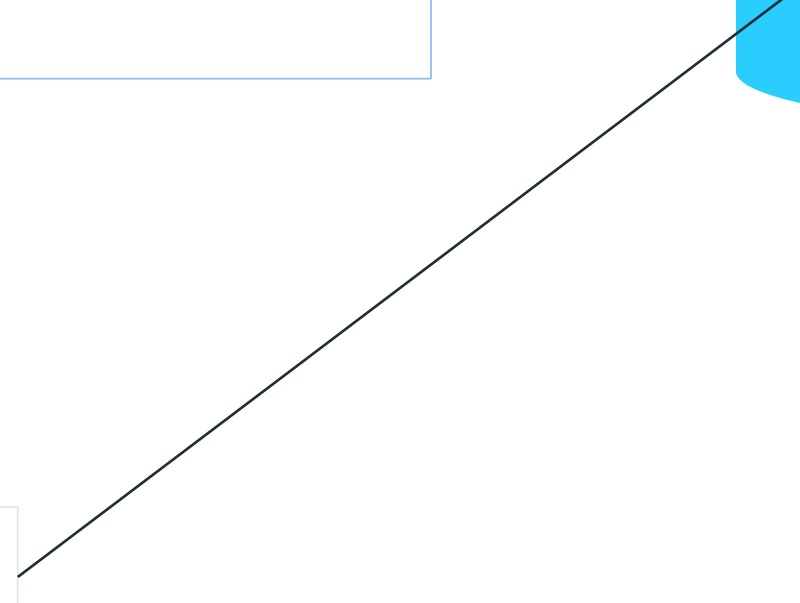
Save a plan for query Q

OPTIMIZER
CUSTOM



Execute P_Q

Run query Q



Plan P_Q



```
SELECT columnX FROM fact WHERE columnX = 'value' ..  
/*+IGNORECONST(1)*/;
```

Annotated Query
= SQL + HINTS

SDK

- C++ and Java Udx's support cancellation
 - All UDX types now support cancellation callbacks.
 - CANCEL() function will perform any UDX-specific cleanup
- Python Udx supports user-defined transform functions
 - UDTFs already supported in C++, Java and R

Transform Functions

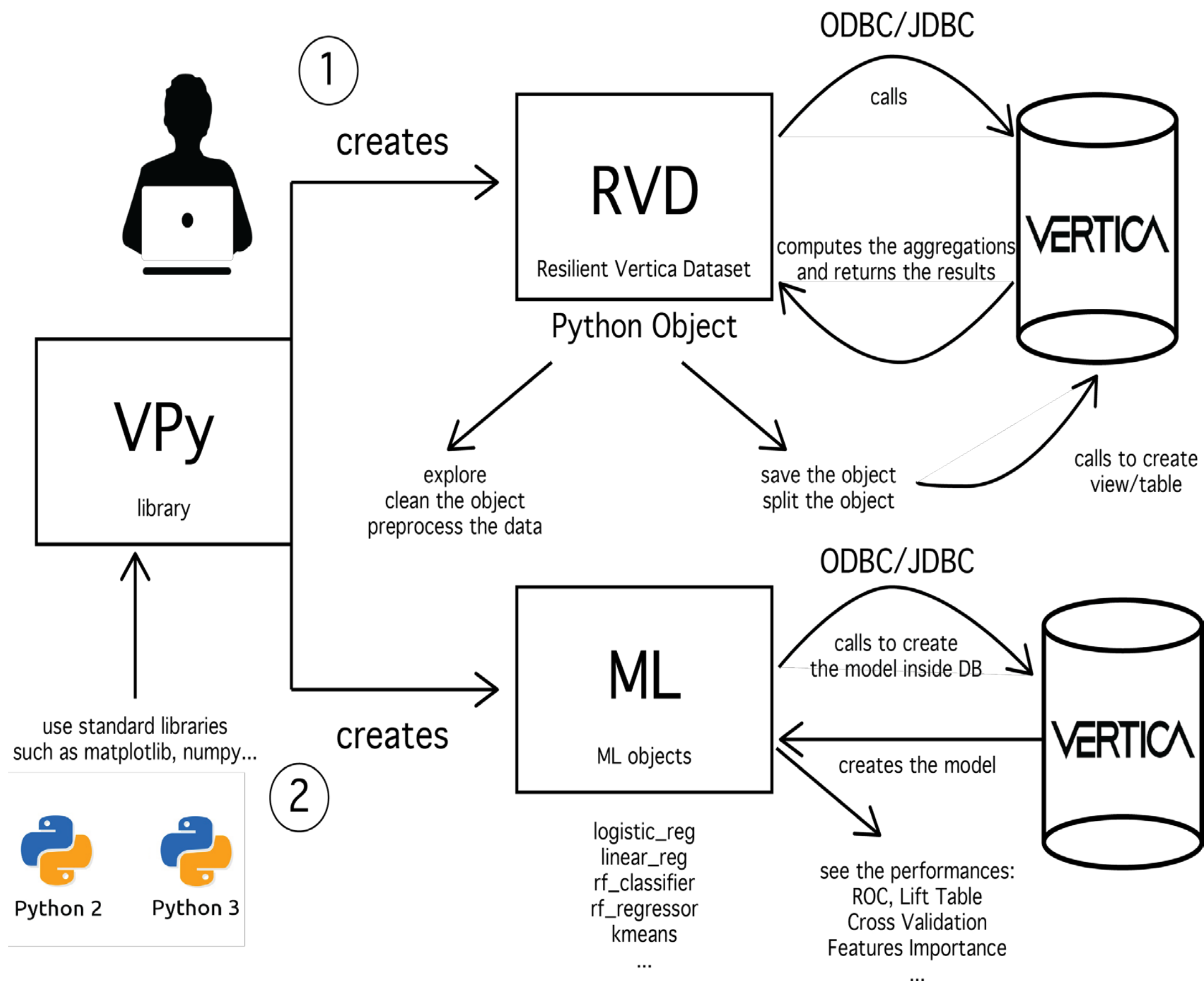
- Function which maps N rows to M rows
 - This is the most generic type of function
- Supported by our C++, Java, and R SDKs
 - 9.1 adds support in Python
- Install the same as other Python UDXes:

```
CREATE LIBRARY pylib AS 'path-to-library' LANGUAGE 'Python';  
  
CREATE TRANSFORM FUNCTION TopKPerPartition  
  
    AS NAME 'TopKPerPartitionFactory' LIBRARY pylib;
```

- Usually run over a partition of data

```
SELECT TopKPerPartition OVER (PARTITION BY a, b ORDER BY a, b, c) FROM foo;
```

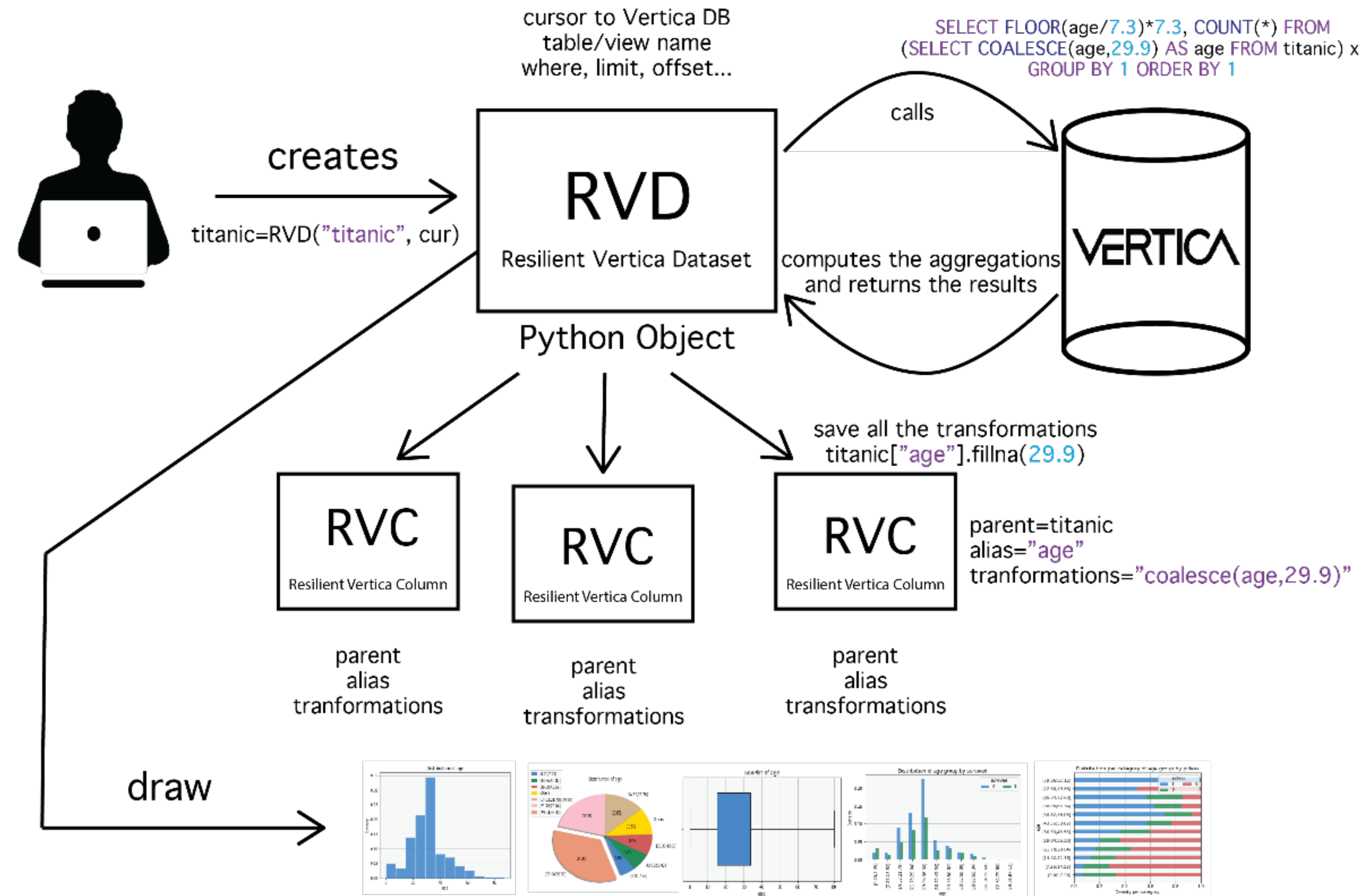

Vertica ML Python



- Python library that exposes sci-kit like functionality
- Data stored in Vertica, thus taking advantage Vertica's and push all heavy computation to VERTICA
- Can be used for data preparation, data exploration i ML
- uses a 'pipeline' mechanism to sequentialize data transformation...

<https://github.com/vertica/Vertica-ML-Python>

Vertica ML Python



<https://github.com/vertica/Vertica-ML-Python>

Vertica Management Console

- External Table visualization

Table Visualisation in Management Console

In 9.1, the Table Utilization page is updated to display information for:

- Vertica external tables (external tables created using Vertica)
- HCatalog tables
- Accessing table information is easier by using MC's **Infrastructure** page - **Storage View** tab

Infrastructure | Storage view:

Show Vertica External Tables for a database

Vertica Management Console

uidbadmin Log out 0 (1) ?

Infrastructure

Auto Refresh Last update: 22 Mar 2018 17:17:37

Database and Cluster ViewStorage View

Database Name: IP

[KM_TEST: 10.20.40.173](#)

[Natalia44_DB: 10.20.80.44](#)

External Tables Details for KM_TEST

Auto-Resize all columns

Schema Name ▲ ₀	Table Name ▲ ₁	Source Format ▼	Total File Count ▼	Total File Size ▼	Source Statement
			greater than less than	greater than less than	
et_details_test	bad_source		0	0 Bytes	copy source Hdfs1(url='http://eng126:50070/webhdfs/v1/us
et_details_test	bzip	DELIMITED	0	0 Bytes	copy from 'hdfs:///user/release/vt_et_details/perm_test/" BZ
et_details_test	csv_hdfs_lib	DELIMITED	1	2.22 KB	copy from 'hdfs:///user/release/vt_et_details/csv_hdfs/"
et_details_test	csv_hdfs_web	DELIMITED	1	2.22 KB	copy from 'webhdfs://eng126:50070/user/release/vt_et_det
et_details_test	csv_local	DELIMITED	1	2.22 KB	copy from '/qa-scratch01/vt_et_details/csv_local/"
et_details_test	dir_not_exists_hdfs_lib	DELIMITED	0	0 Bytes	copy from 'hdfs:///user/release/vt_et_details/dir_not_exists_h
et_details_test	dir_not_exists_hdfs_web	DELIMITED	0	0 Bytes	copy from 'webhdfs://eng126:50070/user/release/vt_et_det
et_details_test	dir_not_exists_local	DELIMITED	0	0 Bytes	copy from '/qa-scratch01/vt_et_details/dir_not_exists_local/"
et_details_test	empty_dir_hdfs_lib	DELIMITED	0	0 Bytes	copy from 'hdfs:///user/release/vt_et_details/empty_dir_hdfs
et_details_test	empty_dir_hdfs_web	DELIMITED	0	0 Bytes	copy from 'webhdfs://eng126:50070/user/release/vt_et_det

1 5 25 items per page 1 - 25 of 111 items

Close

[Vertica Tables Storage](#) | [External Tables](#) | [HCatalog Details](#)

[Vertica Tables Storage](#) | [External Tables](#)

Filter and auto-resize options available

Infrastructure | Storage view:

Show HCatalog Details for a Database

Vertica Management Console

uidbadmin Log out

Infrastructure

Auto Refresh Last update: 22 Mar 2018 17:17:37

Database and Cluster View Storage View

Database Name: IP Database Size: Database Mode Storage Type View

[KM_TEST: 10.20.40.173](#) Load Size Enterprise Linux | External Tables | Hive Tables [Vertica Tables Storage](#) | [External Tables](#) | [HCatalog Details](#)

HCatalog Details for KM_TEST

Auto-Resize all columns

HCatalog ...	Owner	Creator	Time Creat...	Hostname	Port	Hiveserver...	Webhdfs ...	HCatalog ...	SSL Config	HCatalog ...	
default_sc...	release				-1	eng126	eng126.ve...	default		release	View Tables
etd	release				-1	eng126	eng126.ve...	etd		release	View Tables
hive110_g...	release				-1	eng126	eng126.ve...	hive110_g...		release	View Tables
hive110_s...	release				-1	eng126	eng126.ve...	hive110_s...		release	View Tables

Close

Infrastructure | Storage view:

Show Details of HCatalog Tables

Vertica Management Console

Home > Infrastructure

Database and Cluster View

Database Name: IP

[KM_TEST: 10.20.40.173](#)

[Natalia44_DB: 10.20.80.44](#)

HCatalog Details for KM_TEST

Auto-Resize all columns

HCatalog ...	Owner	Creator	Time Creat...	Hostname	Port	Hiveserver...	Webhdfs ...	HCatalog ...	SSL Config	HCatalog ...	
default_sc...	release				-1	eng126	eng126.ve...	default		release	View Tables
etd	release				-1	eng126	eng126.ve...	etd		release	View Tables
hive110_g...	release				-1	eng126	eng126.ve...	hive110_g...		release	View Tables
hive110_s...	release				-1	eng126	eng126.ve...	hive110_s...		release	View Tables

HCatalog tables for schema default_schema, Hive Server eng126

Auto-Resize all columns

Table Schema	Creator	Table Name	Input Format	Partitioned	Partition Expre...	Total Number o...	Total File Size	Location
						<div>greater than</div> <div>less than</div>	<div>greater than</div> <div>less than</div>	
default_schema		array_test	org.apache.had...			0	0 Bytes	hdfs://eng126....
default_schema		array_test1	org.apache.had...			0	0 Bytes	hdfs://eng126....
default_schema		bar	org.apache.had...			0	0 Bytes	hdfs://eng126....
default_schema		customer	org.apache.had...			0	0 Bytes	hdfs://eng126....
default_schema		customer_table	org.apache.had...			0	0 Bytes	hdfs://eng126....
default_schema		customers	org.apache.had...			0	0 Bytes	hdfs://eng126....
default_schema		foo	org.apache.had...			2	614 Bytes	hdfs://eng126....
default_schema		hive_test	org.apache.had...			0	0 Bytes	hdfs://eng126....
default_schema		kevin	org.apache.had...			0	0 Bytes	hdfs://eng126....
default_schema		mytable_external	org.apache.had...		(ss_sold_date_s...	0	0 Bytes	hdfs://eng126....

1 / 1 25 items per page 1 - 17 of 17 items

Close

uidbadmin Log out 0 (0) ?

Last update: 22 Mar 2018 17:17:37

[Home](#) | [External Tables](#) | [HCatalog Details](#)

[Home](#) | [External Tables](#)

27

VERTICA

MICRO FOCUS

Table Utilization Activity Chart: View HCatalog Tables by selecting HCatalog Schema

Vertica Management Console

uidbadmin Log out 3 (0) ?

Databases and Clusters > KM_TEST > Activity Table Utilization

Auto Refresh Last update: [Refresh]

Schemas: default_schema Show Only: External Tables Show as: Table | Map Legend: [Color scale] Darker shows higher table % utilization

Show 100 entries Previous Next Note: tables in this schema are HCatalog

Table Name	Table Type	Row Count	Usage in Queries	Table Type
Filter tables	Int or Ext	From to	From to	
kevin	External	-	-	HCatalog
array_test	External	-	-	HCatalog
bar	External	-	-	HCatalog
customer	External	-	-	HCatalog
customer_table	External	-	-	HCatalog
customers	External	-	-	HCatalog
foo	External	-	-	HCatalog
hive_test	External	-	-	HCatalog
array_test1	External	-	-	HCatalog
mytable_external	External	-	-	HCatalog
myview	External	-	-	HCatalog
nation	External	-	-	HCatalog
orc_hive13_uncompressed	External	-	-	HCatalog
raw	External	-	-	HCatalog
raw_rcfile	External	-	-	HCatalog
raw_sequence	External	-	-	HCatalog
xxx	External	-	-	HCatalog

Showing 1 to 17 of 17 entries

Overview Activity Manage Design Load Query Execution Query Plan License Settings

Projections Summary

Schema: default_schema (Click numbers for details.)

(including base)

0 / 0 (total/distinct) Segmented Projections

0 / 0 (total/distinct) Unsegmented Projections

0

Projections Showing Distribution Skew >= 0%

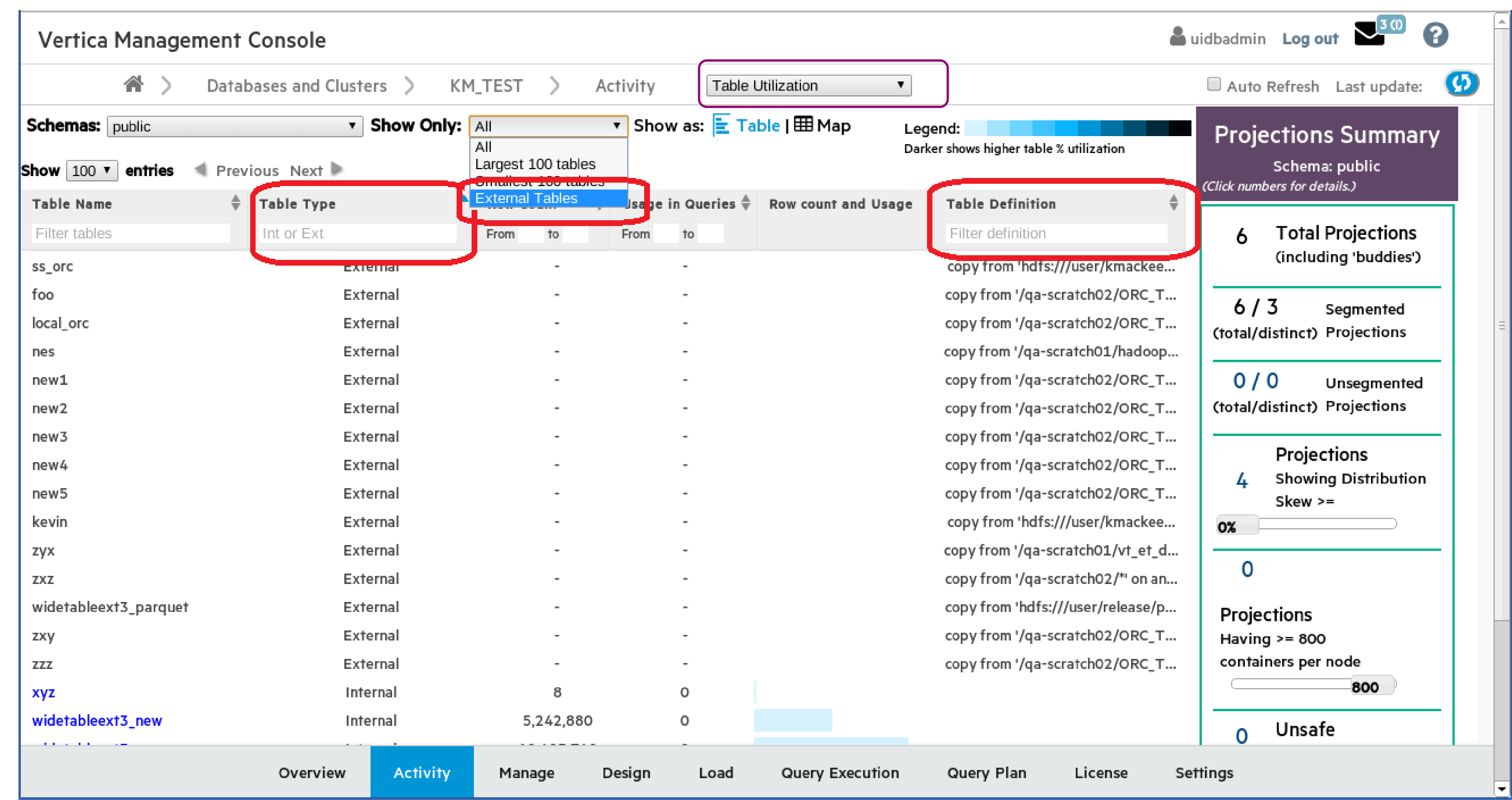
0

Projections

The MC now allows the user to select HCatalog schemas as well as regular schemas that may exist.

Table Utilization Activity Chart:

Additional information for External Tables



Web browser lightweight SQL query editor for Vertica

Get started with Vertica in the cloud without having to configure client drivers & SSH connections

Vertica Management Console

uidbadmin Log out 59 () ?

Databases and Clusters > MCDemoDB > Query Runner

Query History

Clear all

Filter previous queries

SELECT order_number, date_ordered
FROM store.store_orders_fact orders
WHERE orders.store_key IN

</> ⬆ ⬇ ⬆ ⬇ ?

```
1 SELECT order_number, date_ordered
2 FROM store.store_orders_fact orders
3 WHERE orders.store_key IN (
4     SELECT store_key
5     FROM store.store_dimension
6     WHERE store_state = 'MA')
7 AND orders.vendor_key NOT IN (
8     SELECT vendor_key
9     FROM public.vendor_dimension
10    WHERE vendor_state = 'MA')
11 AND date_ordered < '2003-03-01';
```

Execute Query

SELECT order_nu

Query Results Query Plan Query Profile Export Data Expand all columns Search query results

order_number	date_ordered
10499337	2003-02-09
17548215	2003-02-02
763882	2003-01-04
29839276	2003-02-03
20620021	2003-02-10

9353 rows | Execution time: 26.742s

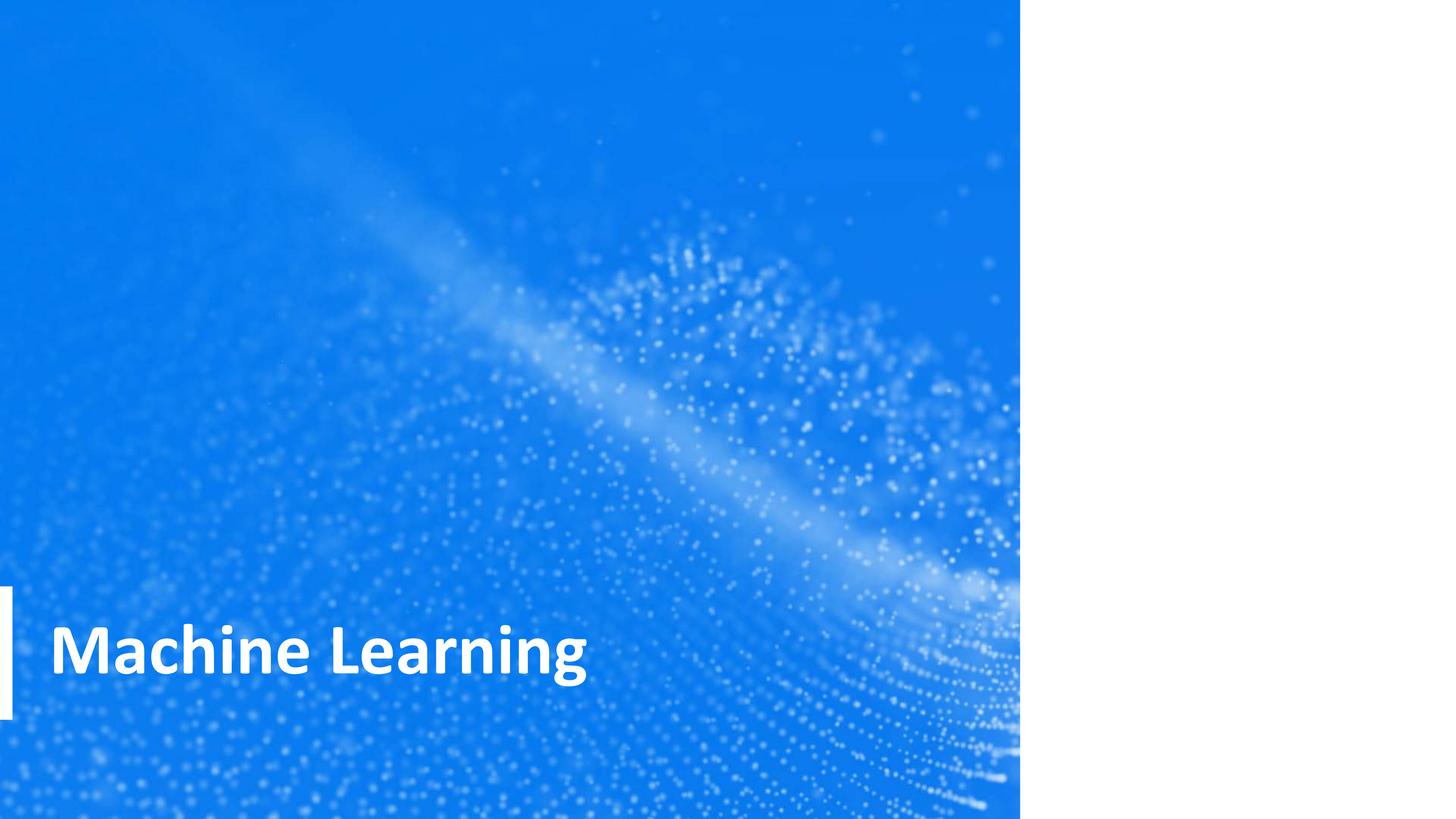
Overview Activity Manage Design Load Query Plan License Settings Run Queries

Import and format SQL

Save data entry by re-running a previously executed query

Result data in spreadsheet-like view. Can export or search

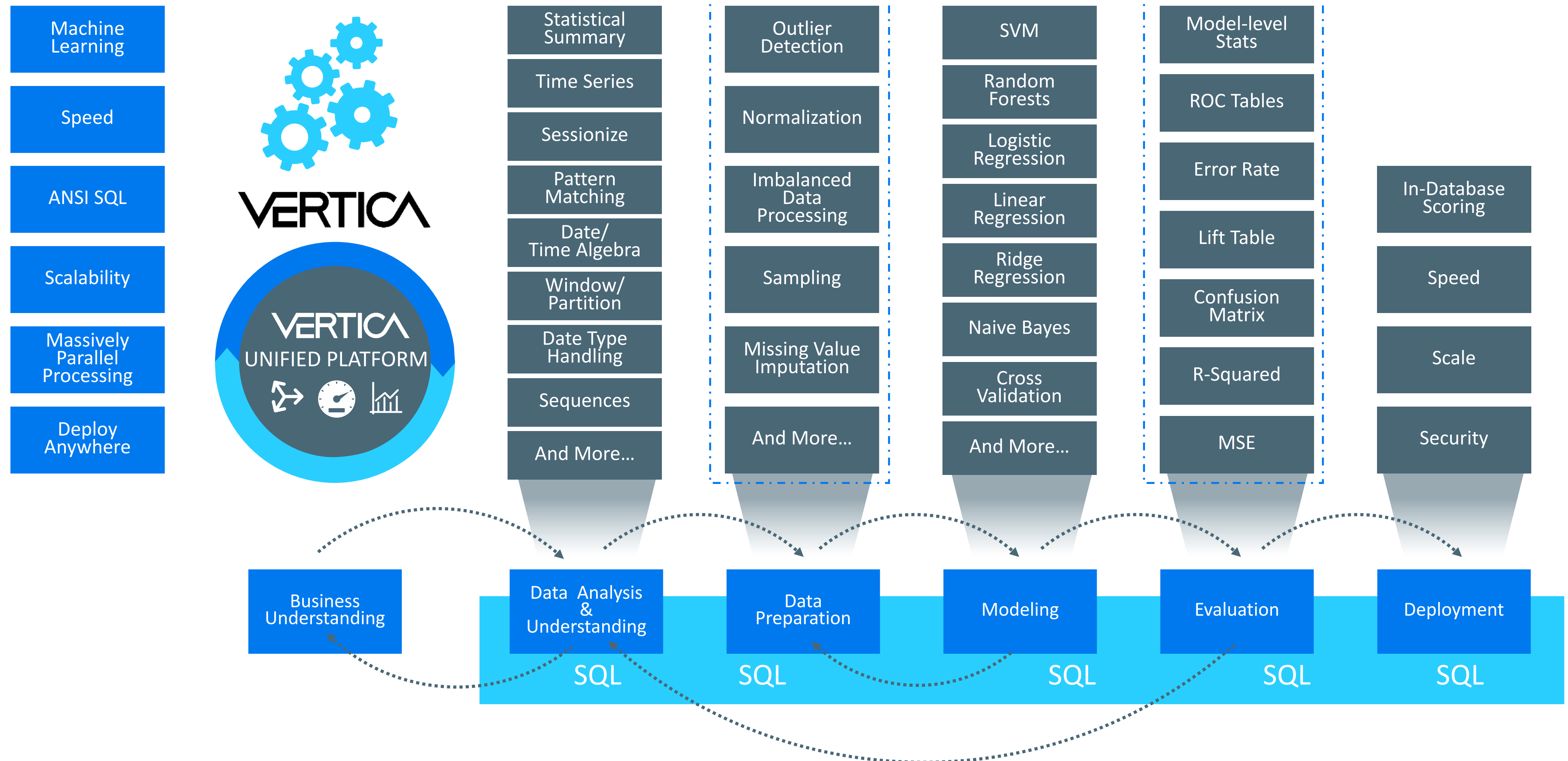
View query plans and profile information in the same area



Machine Learning

Supporting the Entire Machine Learning Process

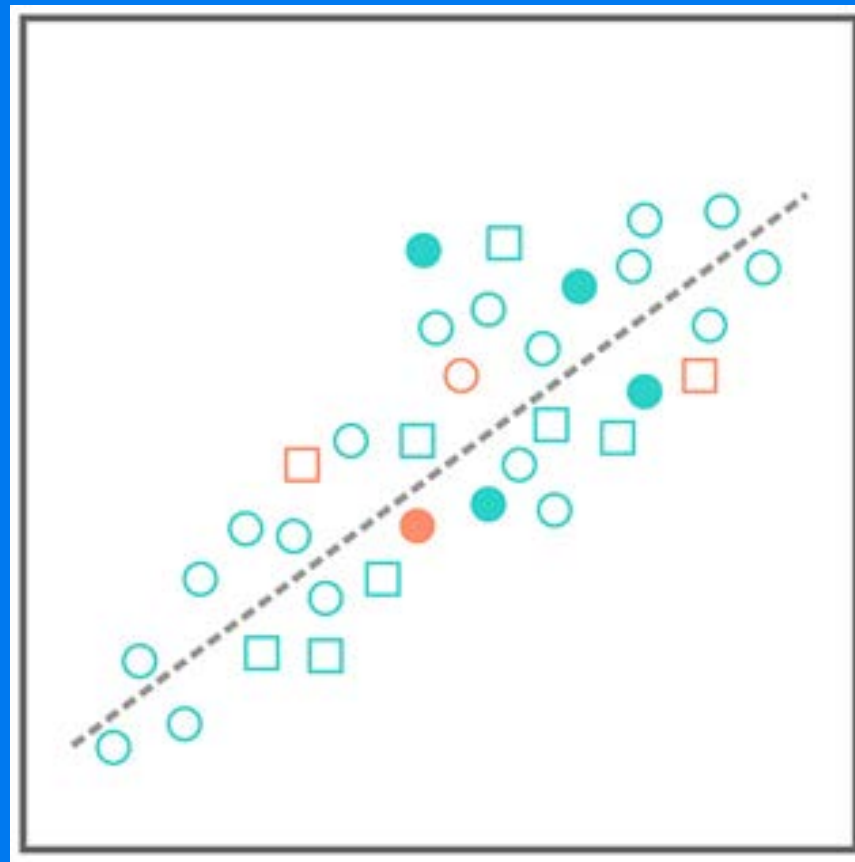
V9.1
Enhancements



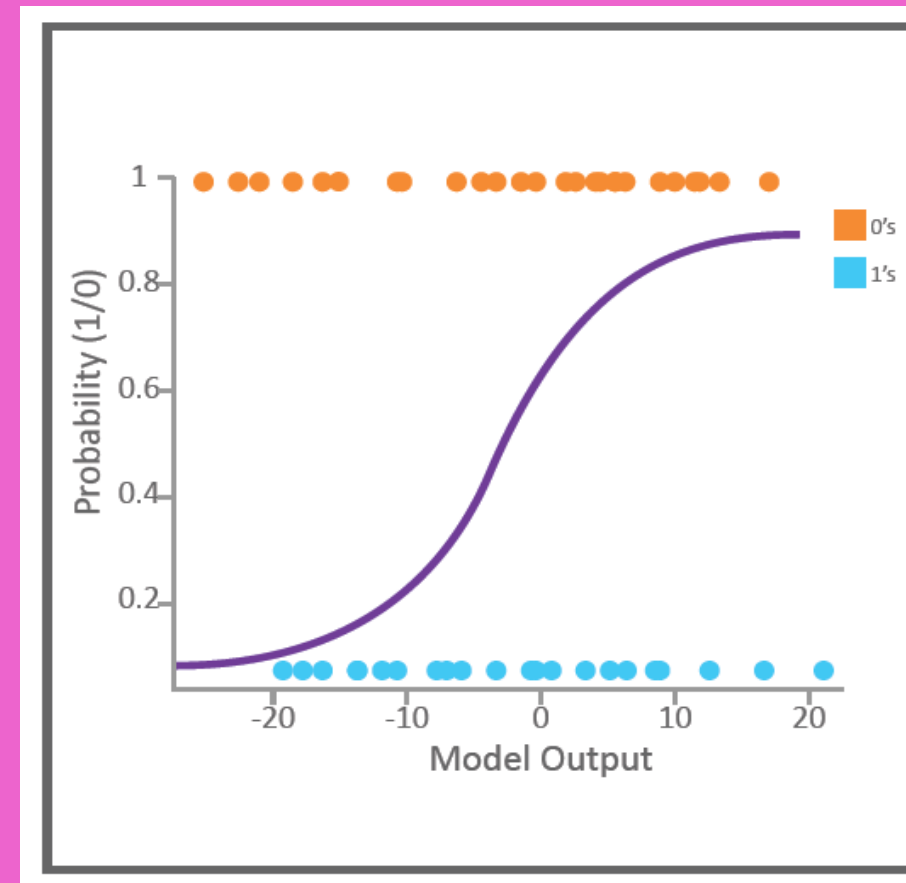
Vertica ML Algorithms and Sample Use Cases

Model training, prediction and evaluation supported for each algorithm

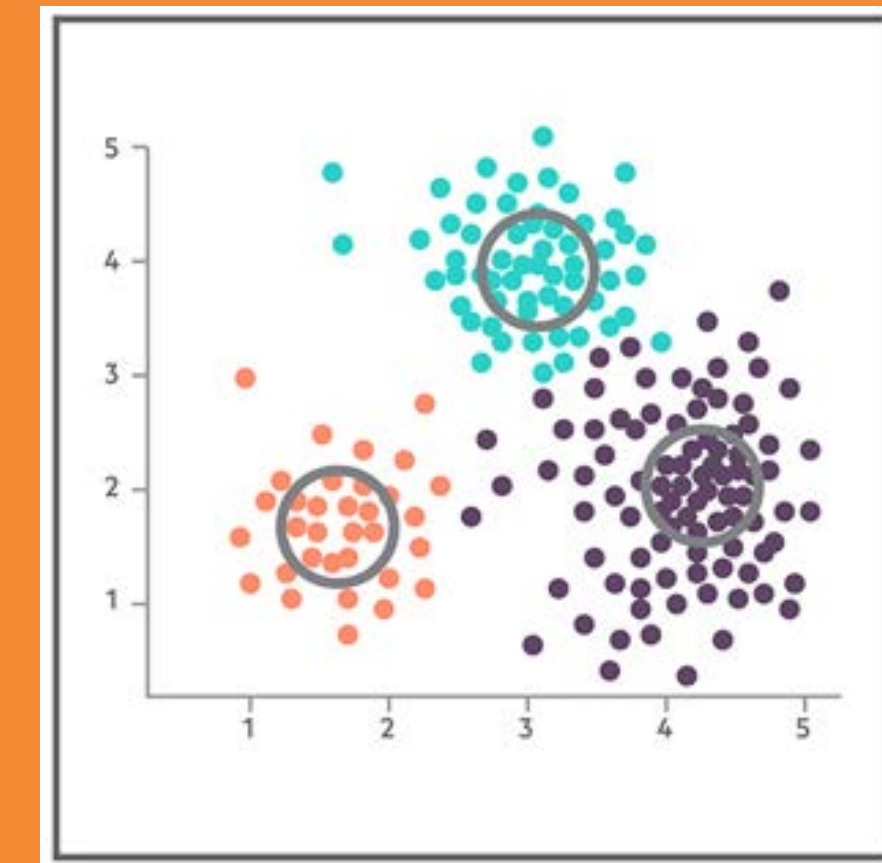
Linear Regression



Logistic Regression



K-means



What it does

Numerical prediction

Binary classification

Clustering

Sample use case

Forecasting sales revenues

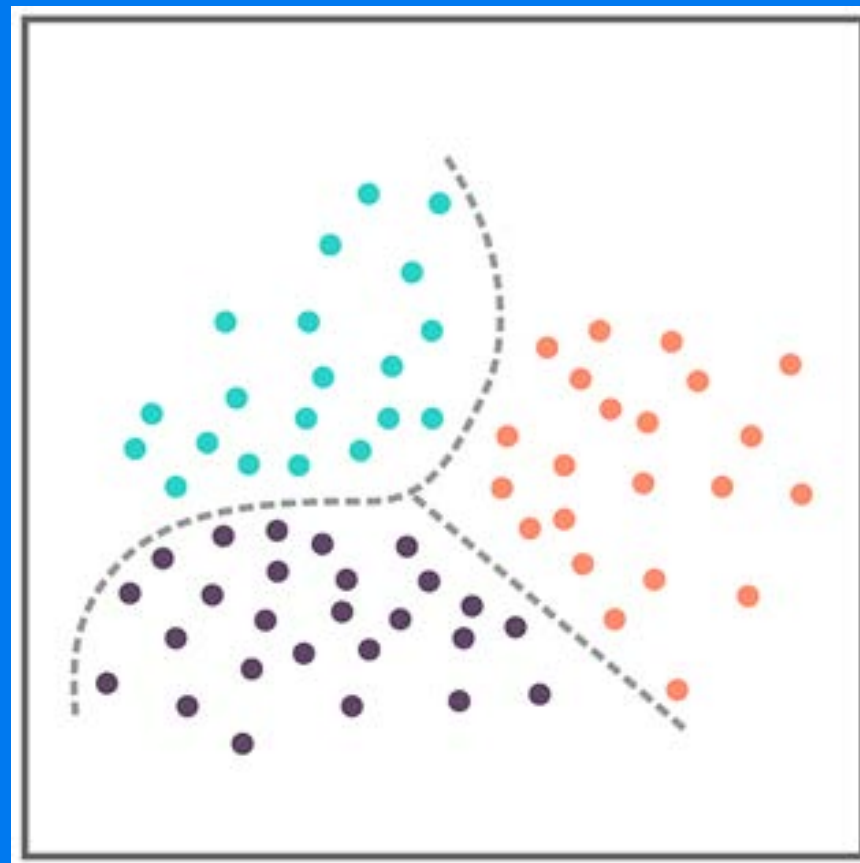
Customer response modeling

Customer segmentation

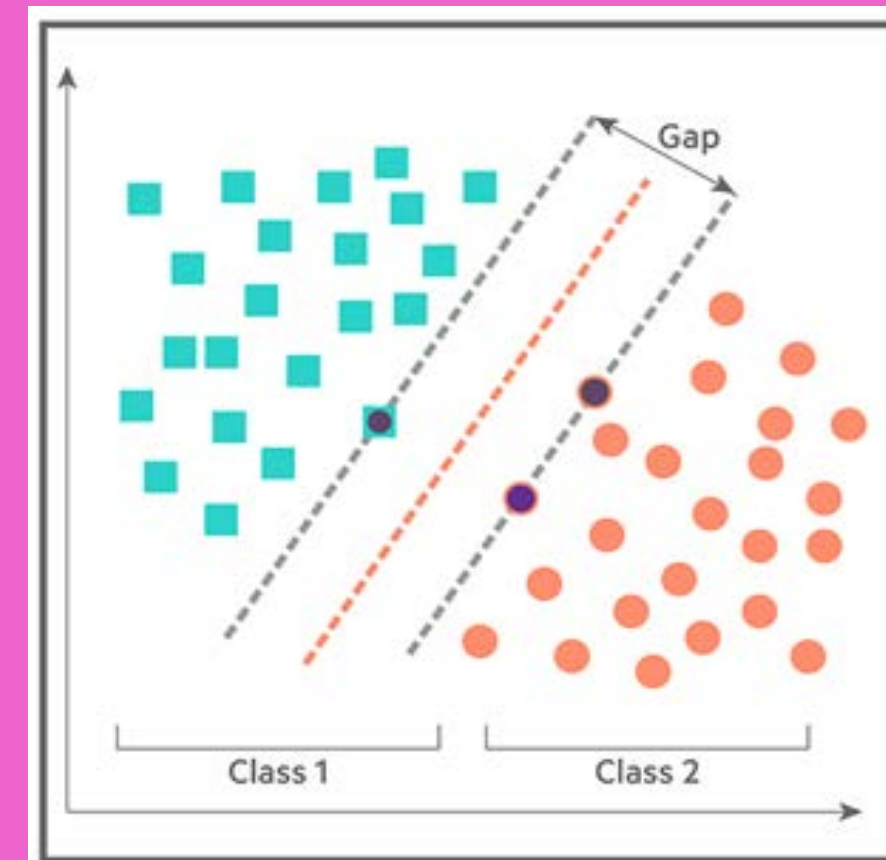
Vertica ML Algorithms and Sample Use Cases - cont.

Model training, prediction and evaluation supported for each algorithm

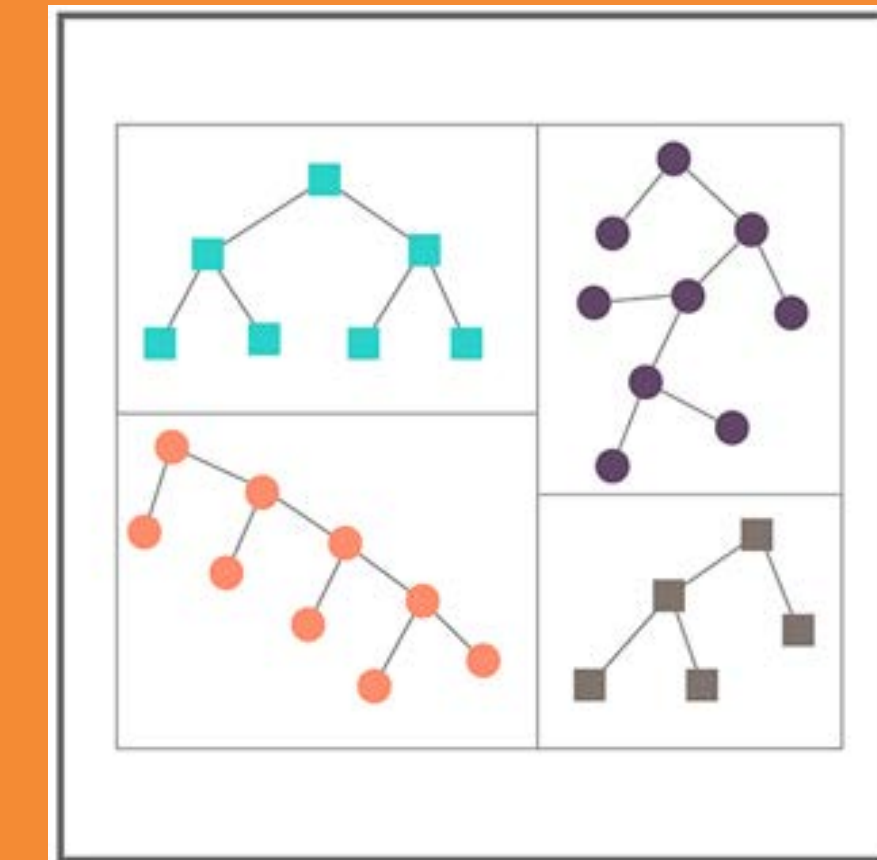
Naïve Bayes



Support Vector Machines



Random Forests



What it does

Multi-class Classification

Sample use case

Classify Gene Expression
Data for Drug Discovery

Binary classification, Regression
(with Regularization)

Customer Response Modeling

Multi-class Classification

Refine Keywords to Improve
Click Through Rate (CTR)

Vertica ML Algorithms and Sample Use Cases

Feature reduction

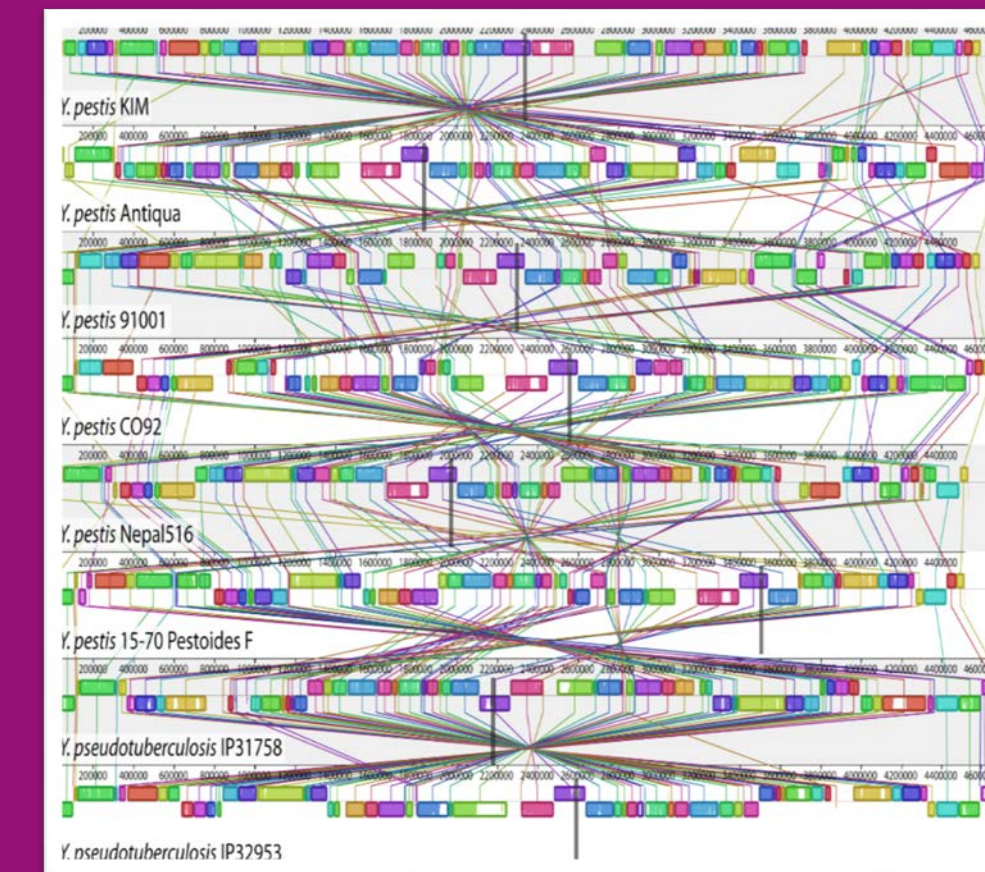
SVD and PCA

- Drastically reduces the number of features for machine learning modeling
- Reduces time to train a machine learning model
- Makes it easier to visualize the data

Choice of Machine Learning Evaluation Functions

- F-1 score offers a single metric for the precision-recall curve to compare machine learning models
- Area under the curve (AUC) evaluation metric
 - Evaluating performance of binary classifiers
 - Useful for imbalanced classes

PCA, SVD



What it does

Feature dimension
reduction

Sample use case

Reduce time of model
calculation

Optional automatic conversion of categorical data to numerical data

One-hot encoding for categorical columns

- **Challenge:** Customers frequently work with categorical data such as state = CA. Some algorithms required users to *manually* convert categorical data to numerical data.
- **Solution:** Create a function to automatically create what are known as one-hot encoding columns, or dummy variables, from a given column in Vertica.
- **Benefit:** Less manual data preparation work. For example, user would not have to script fifty if-then-else clauses to convert state names to numerical codes.
- **Example:** Customer data table contains information including gender values from a post-purchase survey. In order to feed this data to a linear regression algorithm, they need to perform one-hot encoding on the gender column.

ID	Gender		ID	Female	Male	Not specified
1	Female	One-hot encoding →	1	1	0	0
2	Male		2	0	1	0
3	Not specified		3	0	0	1
4	Female		4	1	0	0
5	Not specified		5	0	0	1

Cross Validation

Evaluate, tune, and select ML models

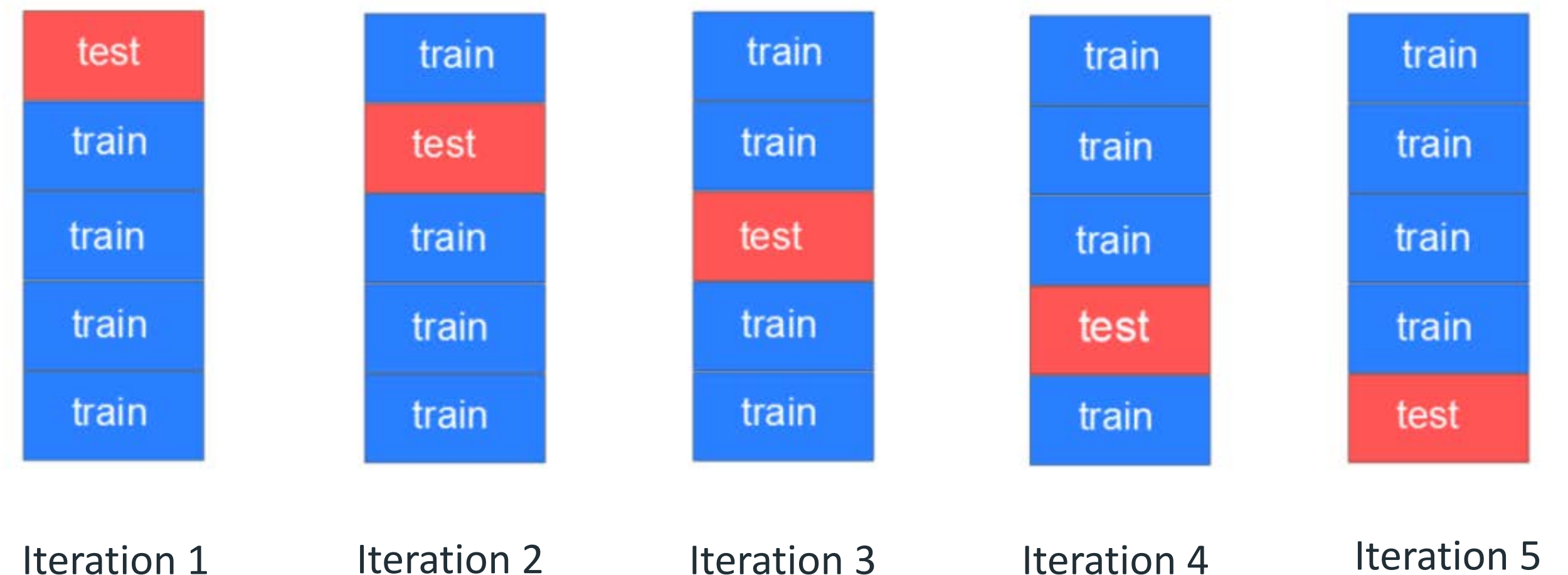
What is it?

- A new way to compare different algorithms to evaluate which one is most reliable
- Vertica has added a cross-validation function which helps data scientists in model selection

Why is it important?

- While trying to predict churn from a given data set, is a logistic regression better than an SVM?

Training and testing data sets shown for k value of 5



Export of models from one Vertica cluster to another

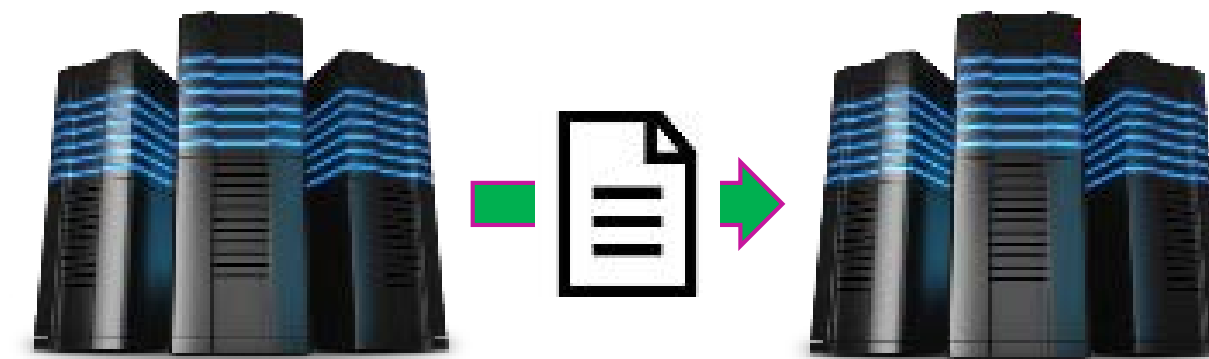
What is it?

- An easy way to train models in one cluster and copy the result to another

```
SELECT EXPORT_MODELS('/home/user1', 'myschema.*');
```

Why is it important?

- Customers often need separate clusters for defining new models vs deploying the models
- Users can train machine learning models on a Vertica cluster and then deploy it in another Vertica cluster



Easier data exploration

Statistical summary function for numeric columns

- **Challenge:** Data scientists need to explore their data as part of the machine learning process. They need to apply many functions to see the key statistics of their numeric continuous data.
- **Solution:** Vertica has added a statistical summary function to enable users to get the important statistical information on the numeric columns.
- **Benefit:** Users don't have to make several SQL calls every time to find the mean/mode, min/max, standard deviation, etc. for their data. They can use one function to get all that information on a particular numeric column or multiple numeric columns.

- **Example:**

```
SELECT SUMMARIZE_NUMCOL(age) OVER( ) FROM customer_data_table;
```



Vertica in Eon Mode

- Why Eon?
- When to use
- How to use

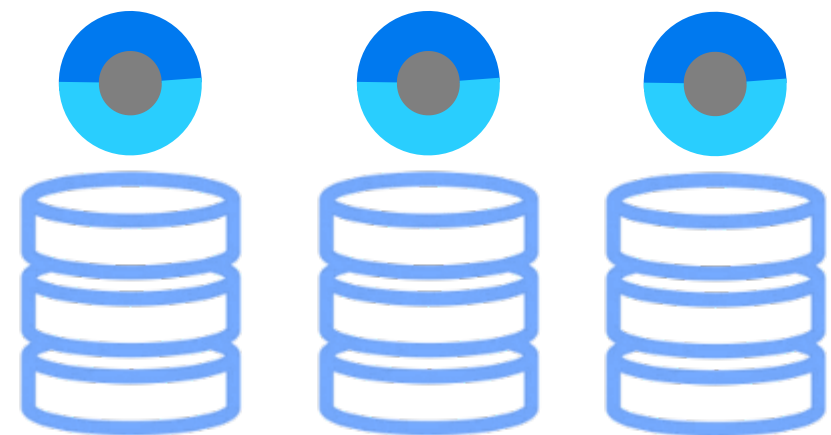
Eon for fast and economical cloud data warehousing



First Generation

Using the cloud as a data center (IaaS)

Amazon
Microsoft Azure
Google Cloud



Use Case

- Savings on data center costs
- Faster deployment

COST SAVINGS

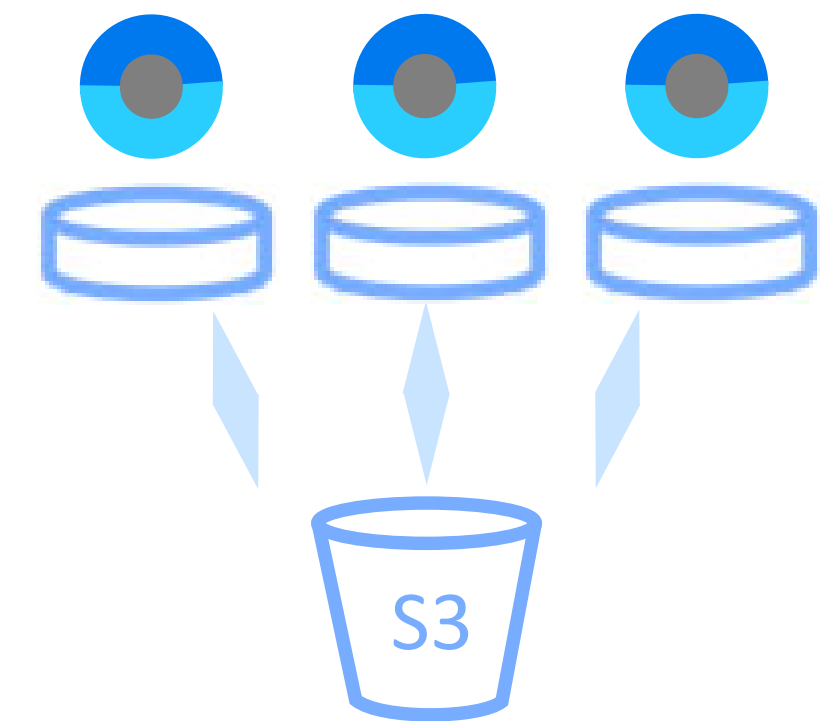
- | | |
|-------------------------------------|---------------------------|
| • Hardware depreciation (lease) | • Data center floor space |
| • Hardware maintenance | • Provisioning time |
| • Software purchase or depreciation | • Backup infrastructure |
| • Software maintenance | • Backup media |
| • Power consumption | • Cost of procurement |



Second Generation

Separation of compute and storage

Amazon



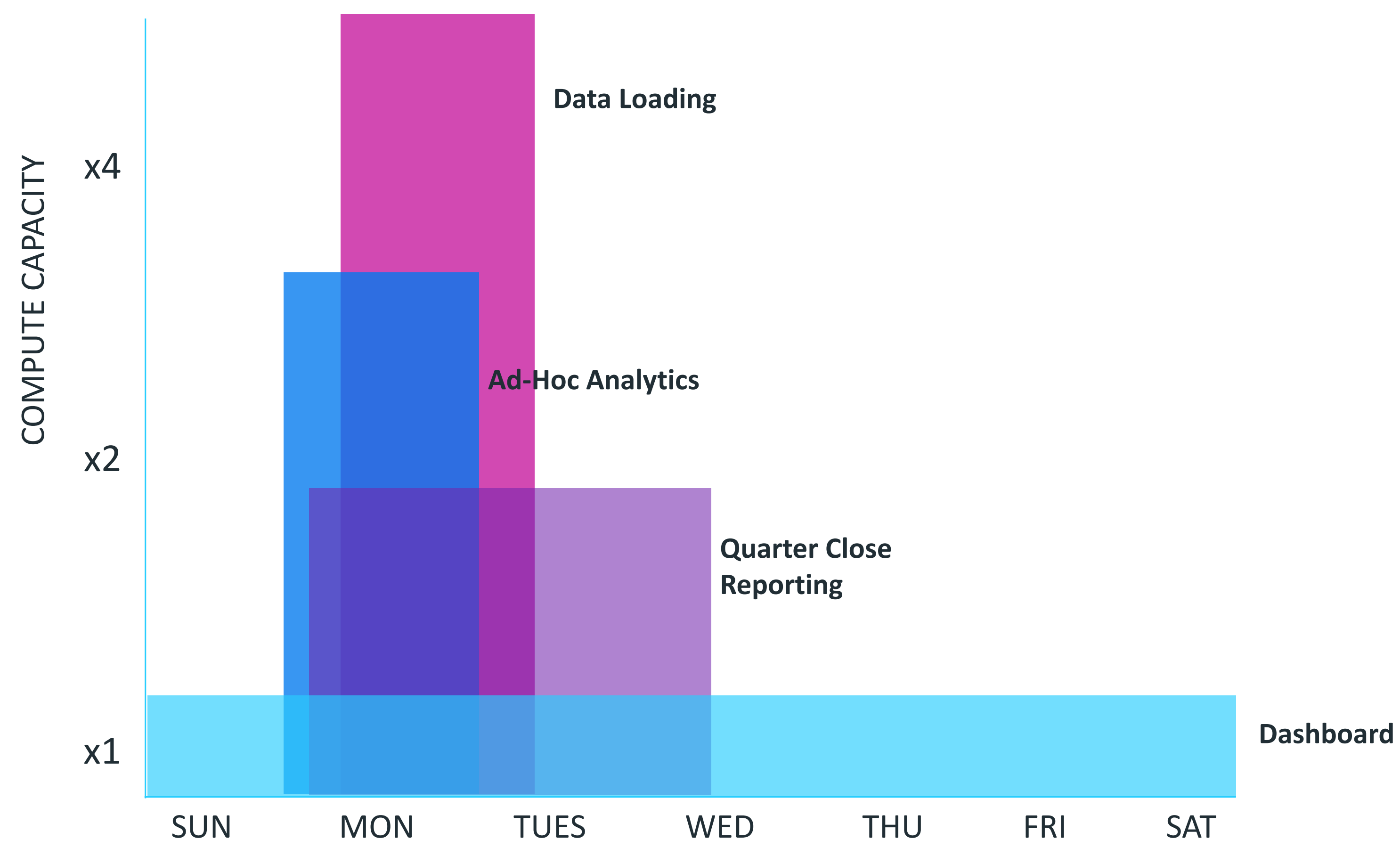
Use Case

- Exploit cloud economics
- Variable workloads

COST SAVINGS

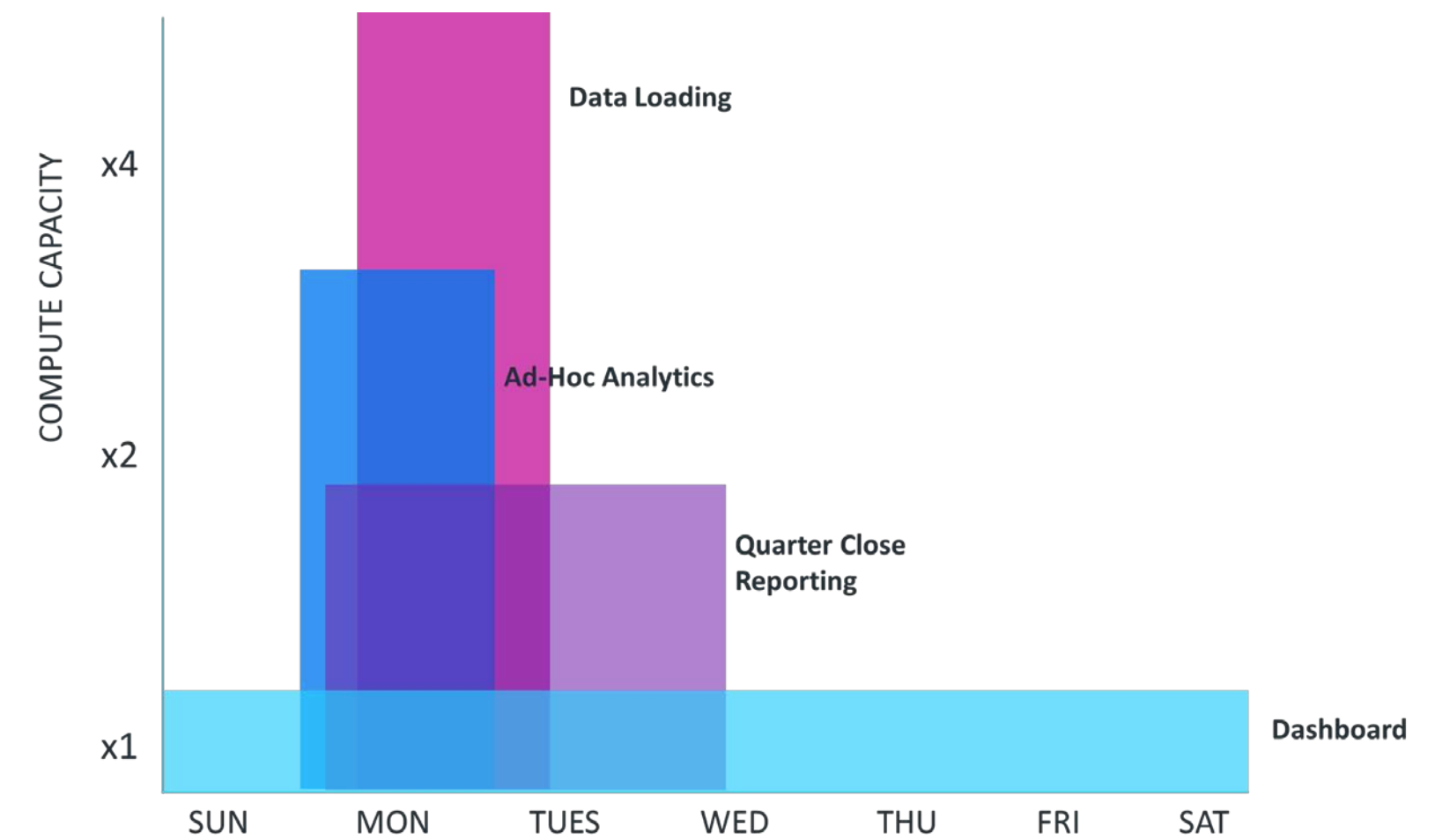
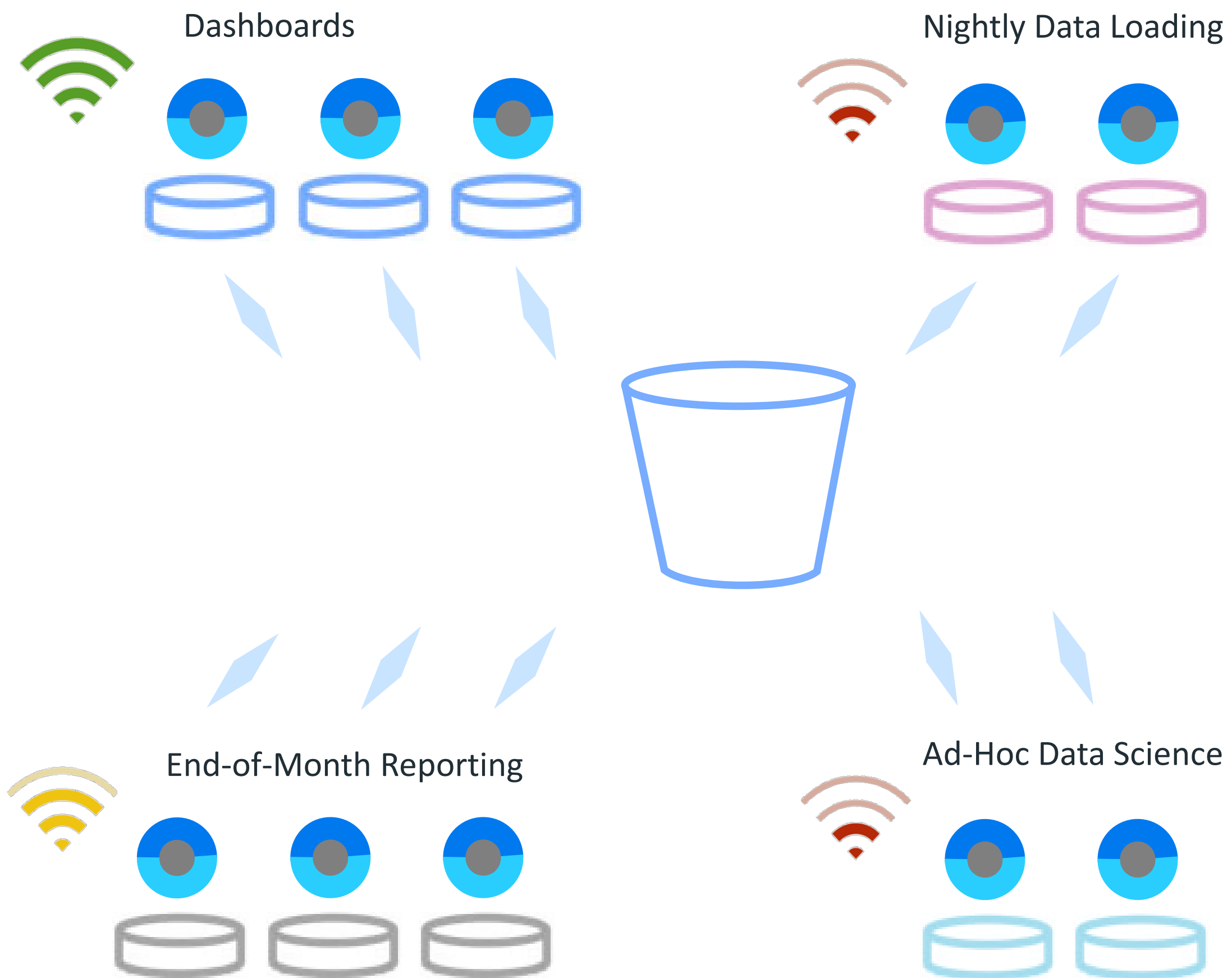
- Costs of provisioning for peak workloads
 - Then paying for idle capacity during slack workloads
- Cost of paying for compute when not in use
 - Rather than Hibernate
- Moving data to handle new workloads
 - Rather than spin up a new sub-cluster

Eon can adapt to your workload every day



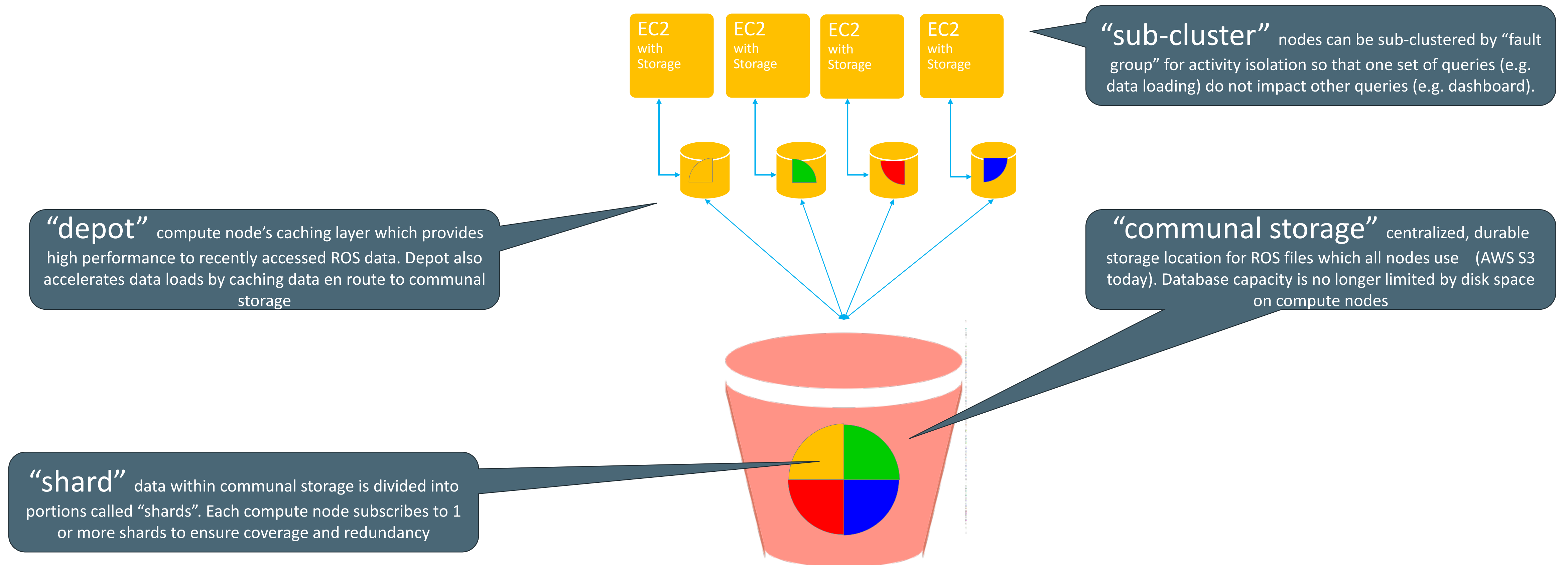
A new degree of workload isolation and flexibility

Sub-cluster Feature, Same data used for different, variable workloads

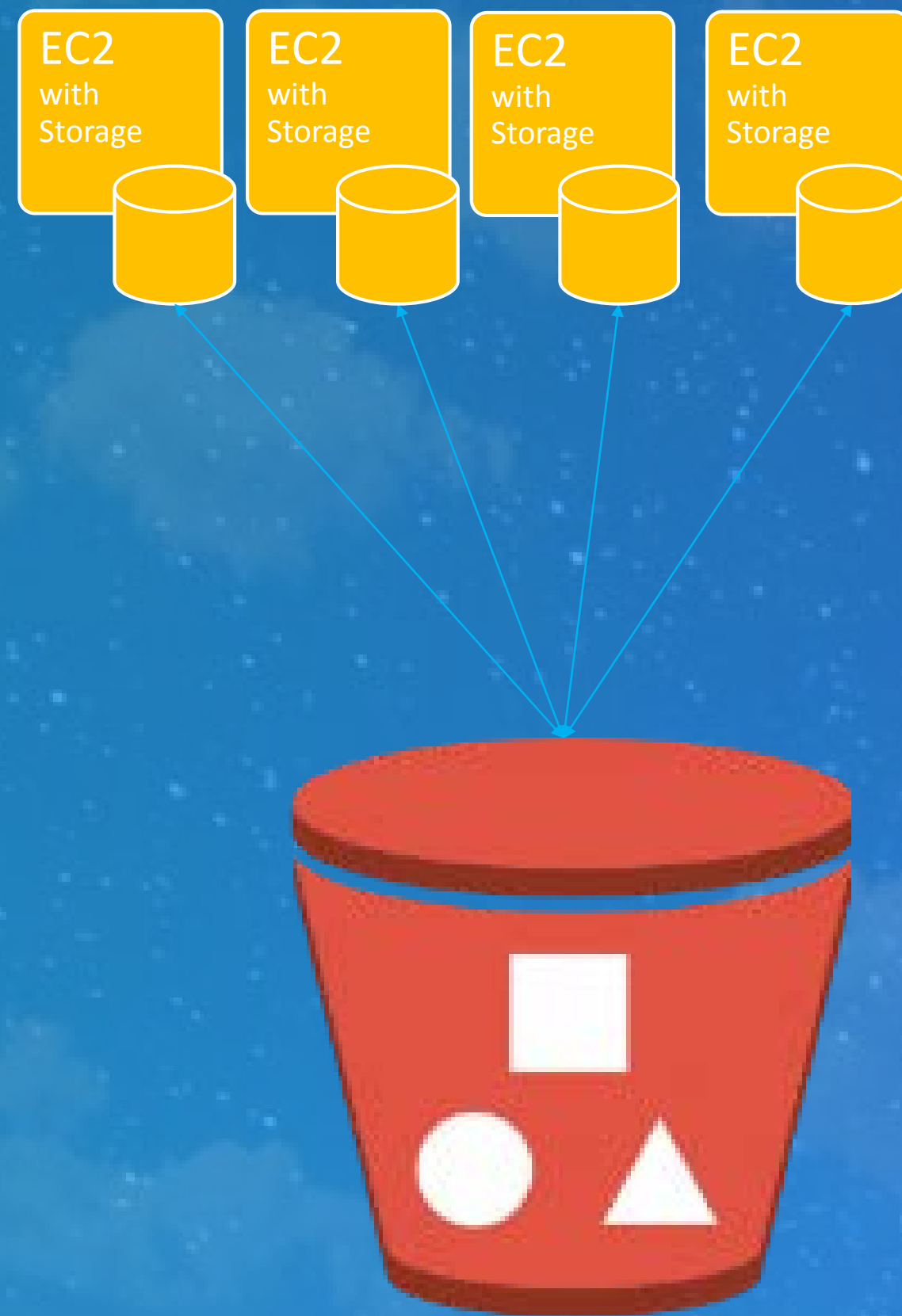


Each different activity gets its own sub-cluster, provisioned when needed with the best instance type for the job.

Eon Mode concept

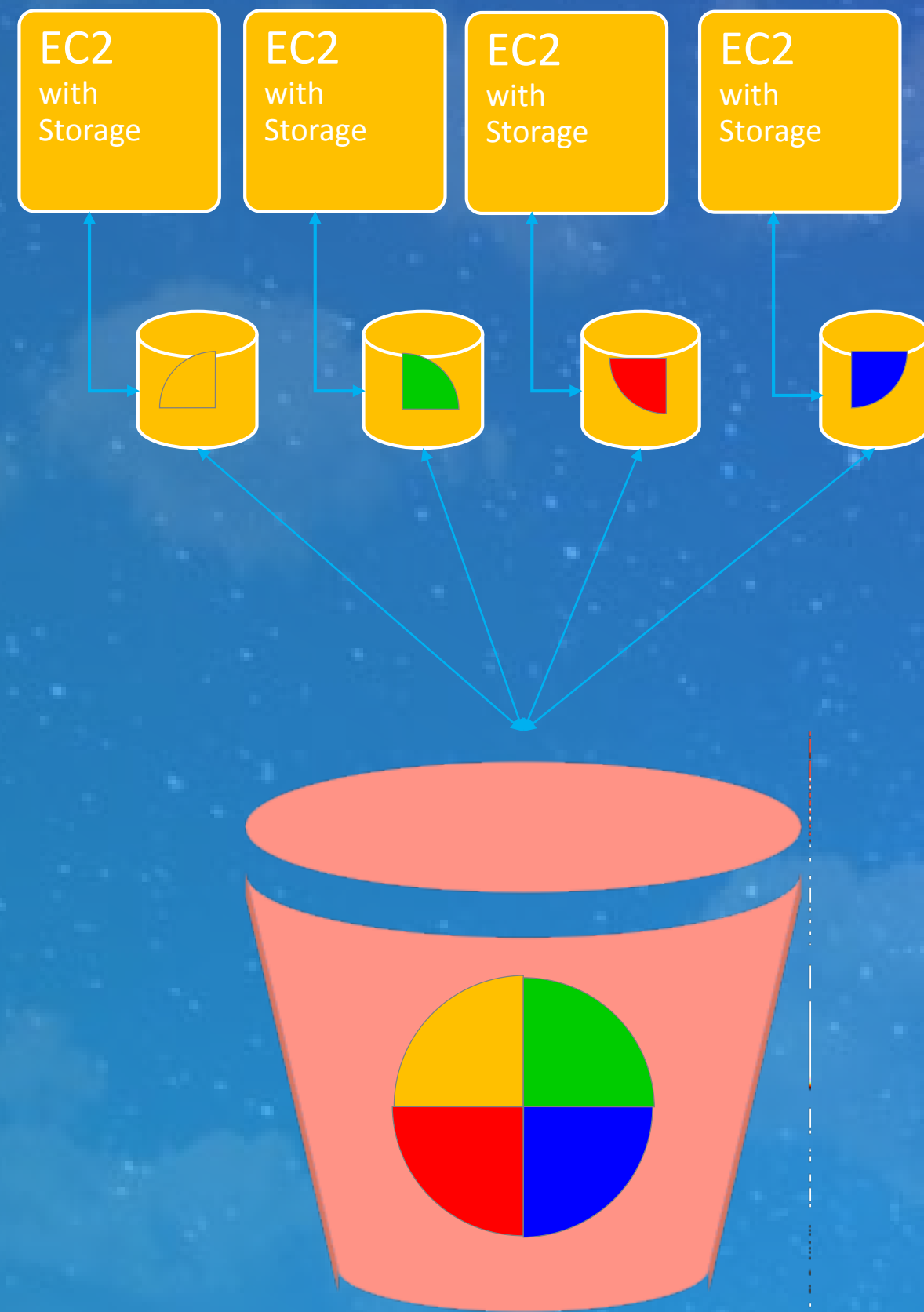


Simpler, Faster Provisioning



All you need are EC2 instances with storage and an S3 Bucket.

How does EON Mode work?



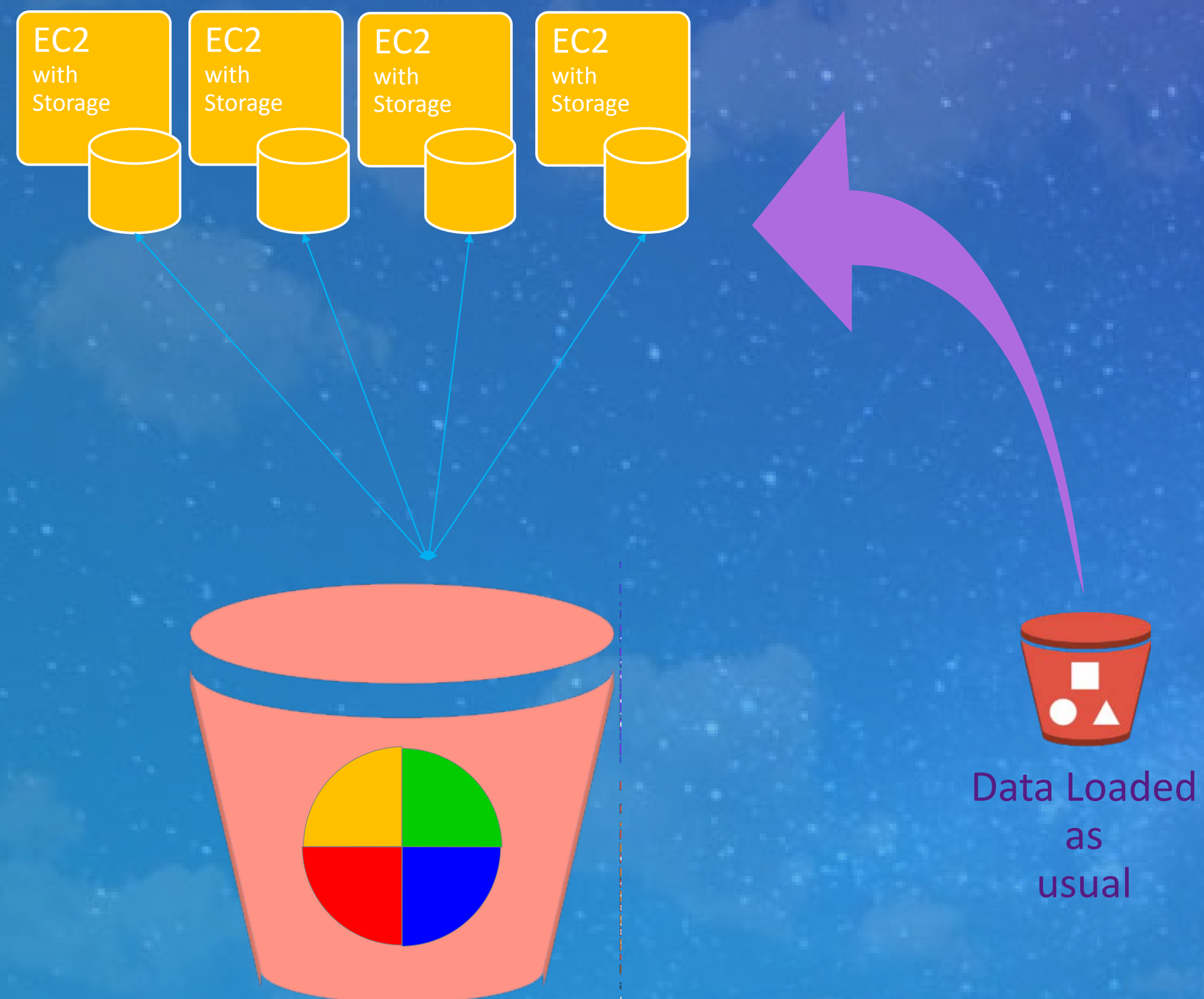
Nodes with instance storage are provisioned.

Each node caches a shard of the database from S3 into instance storage from which it can service queries with the blazing performance you expect from Vertica.

Collectively, this ephemeral instance storage layer is called the “depot” and remains consistent with the full database maintained in S3.

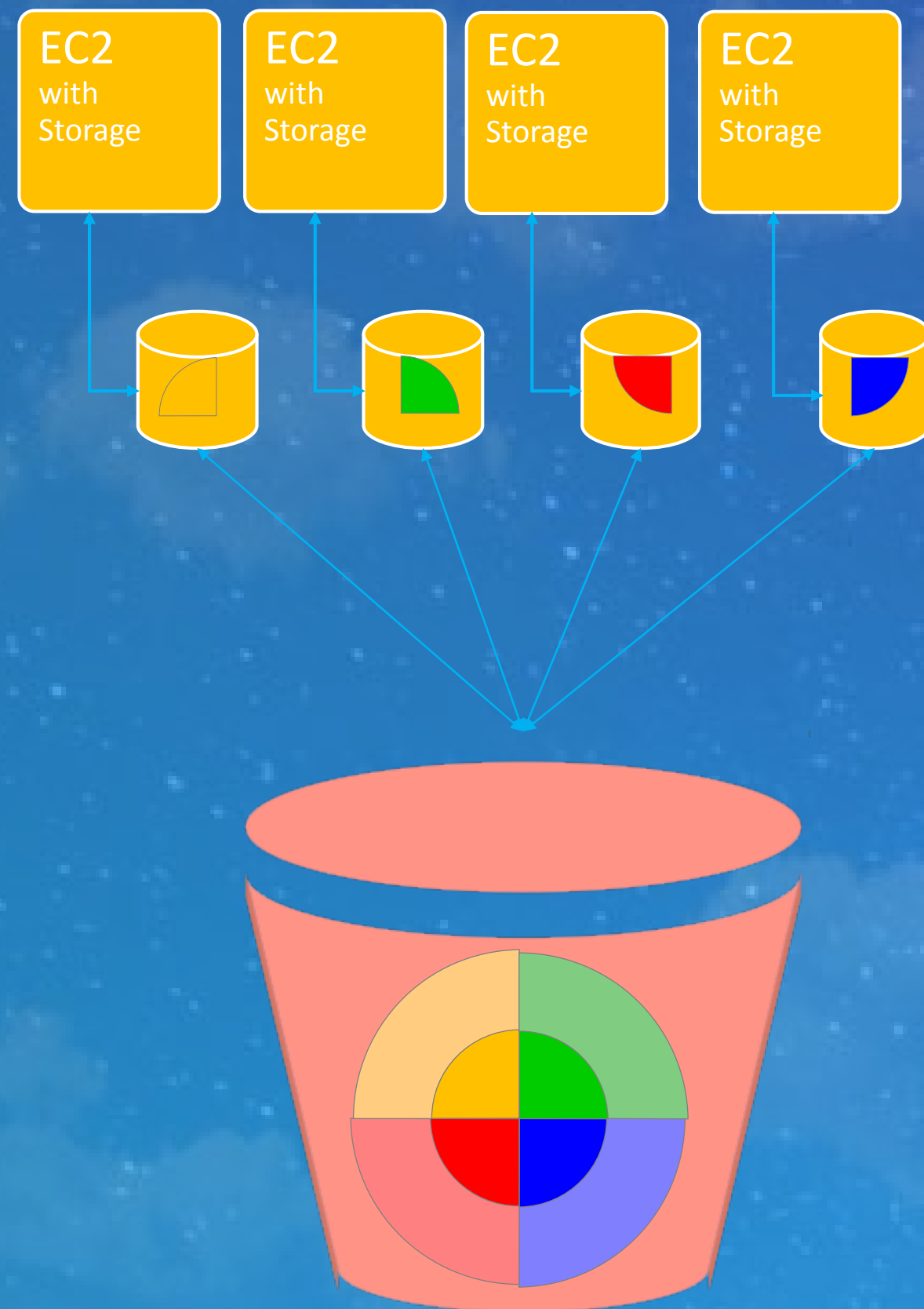
Your database is stored in S3 with 11 9’s of reliability.

Load Data the Same Way, but Easier



Data is loaded into Vertica Eon mode the same way it is loaded into Vertica Enterprise mode.

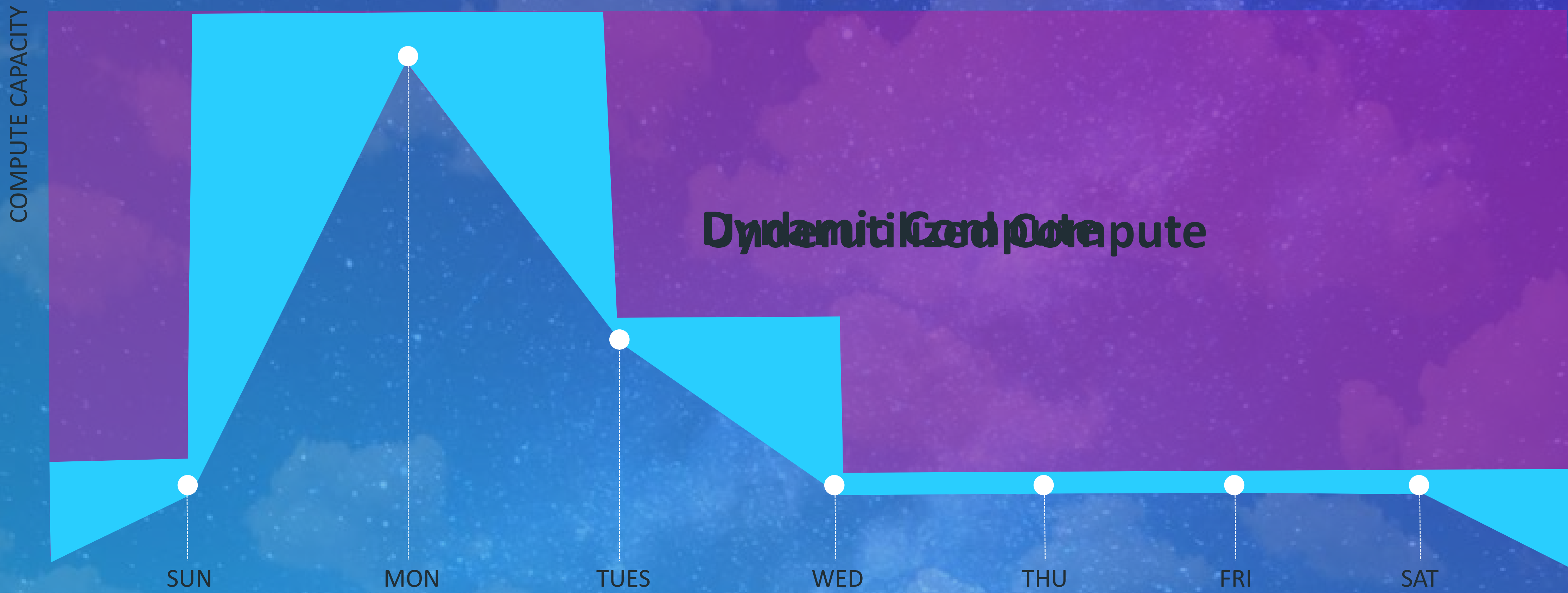
How does EON Mode work?



A node may not store the entire shard in the depot.

When a new query can't be satisfied by the depot, the node will run the query directly against S3 while updating the depot.

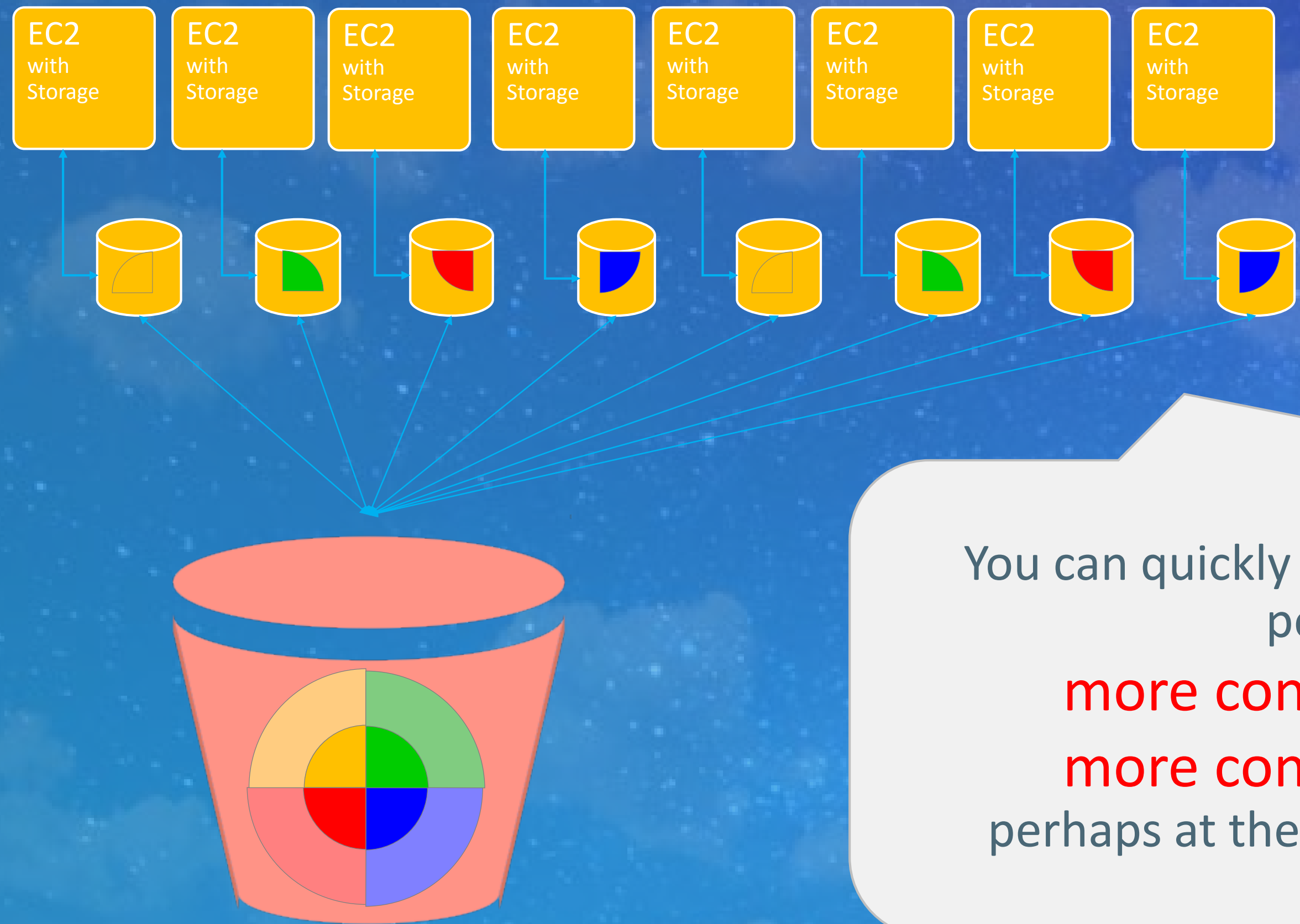
Scaling to Available Workload



Dynamically Scaled Compute

Idle Compute

Rapid Scaling



You can quickly spin-up additional nodes to maintain blazing fast performance when you need to run:

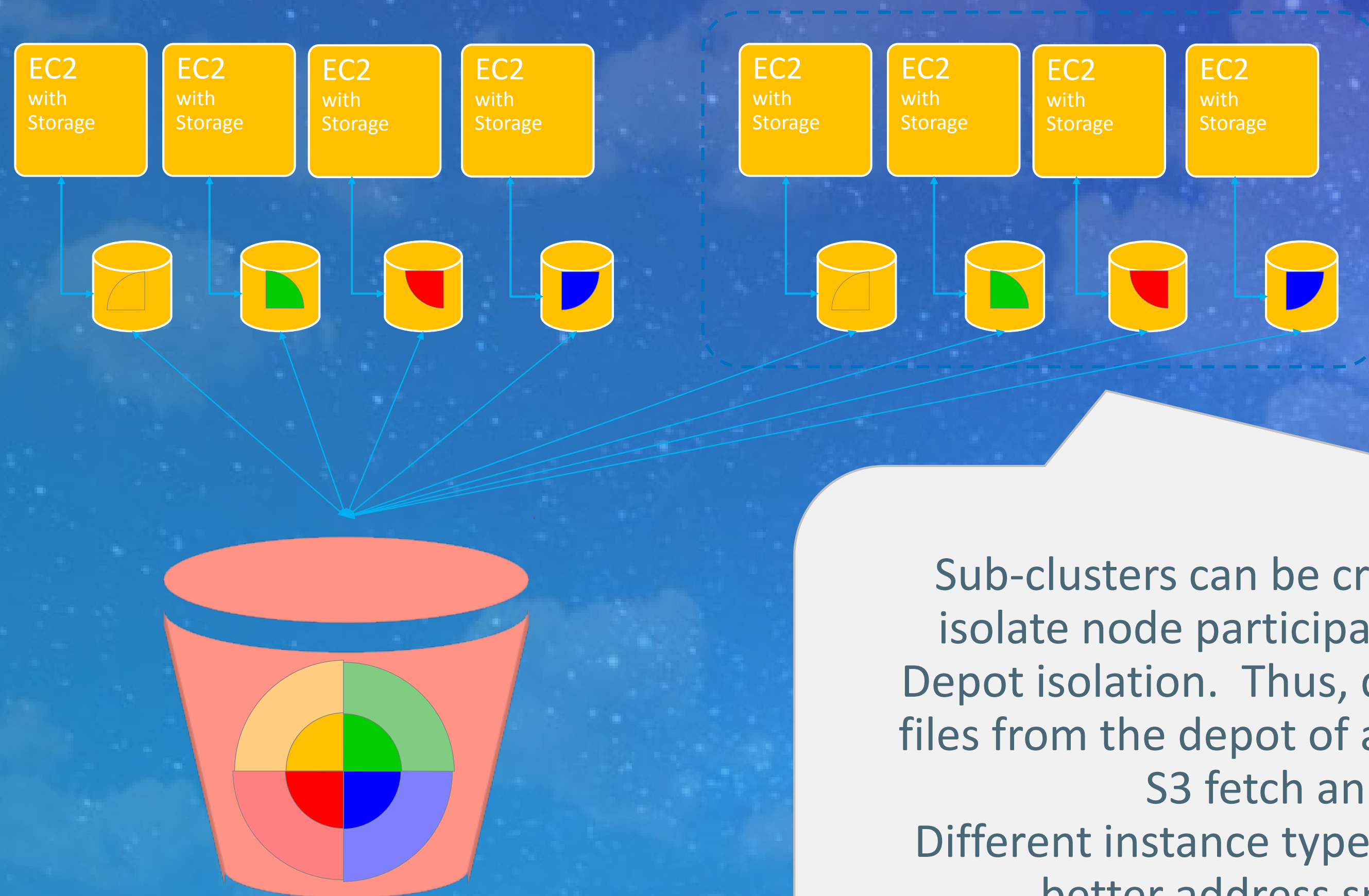
more concurrent queries or

more concurrent loads

perhaps at the end of a month or simply at the end of the day.

* Scaling for query speed remains a roadmap item

Isolate Workloads using Sub-clusters



Sub-clusters can be created using Fault Groups. Fault Groups isolate node participation in queries which, in turn, results in Depot isolation. Thus, queries in one sub-cluster will never evict files from the depot of another sub-cluster which would force an S3 fetch and reduce overall performance. Different instance types can be used in different sub-clusters to better address specific activities of the sub-cluster.

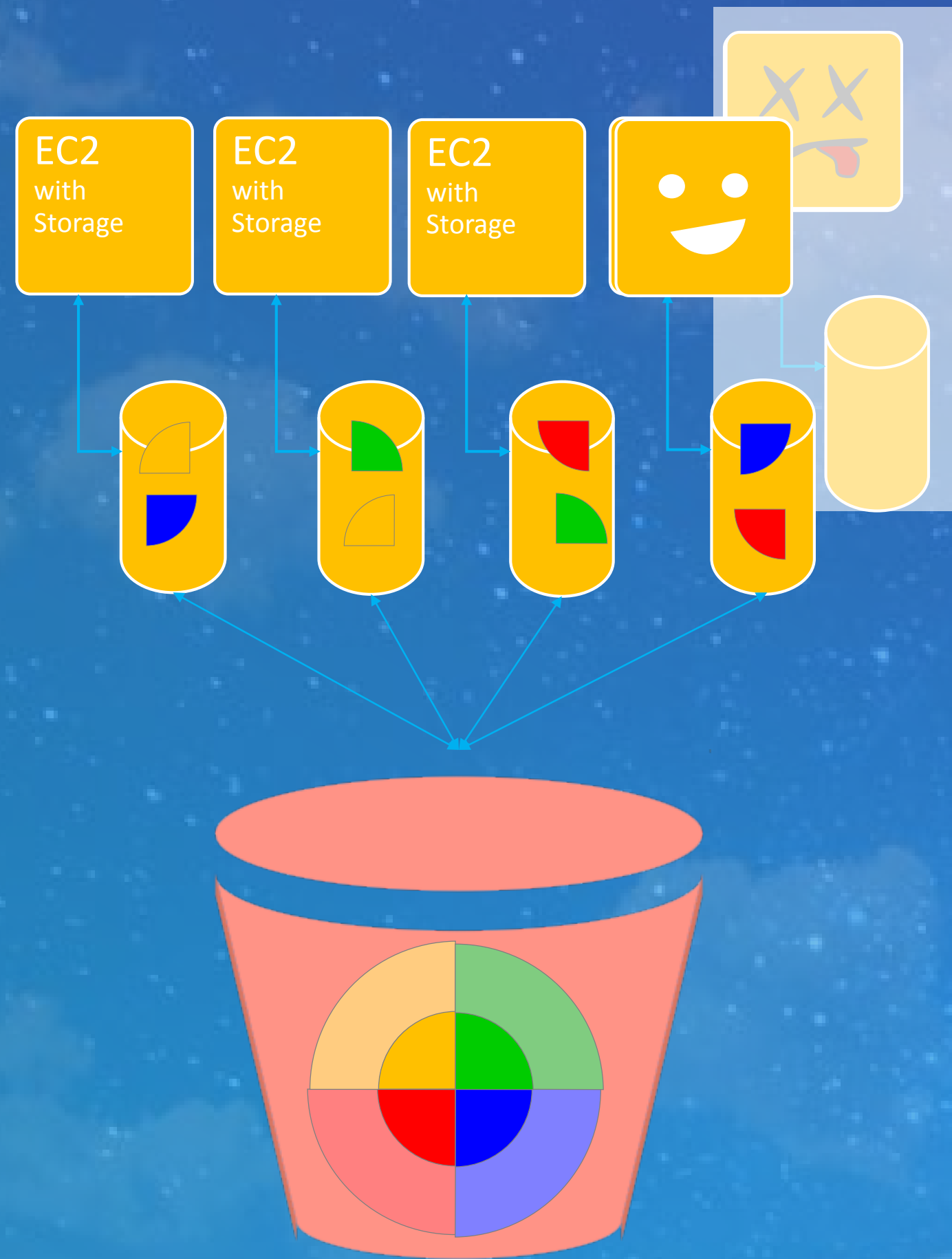
High Availability through Multiple Shards



Each node is responsible for a multiple shards so that when a node goes down, queries continue to be satisfied by alternate nodes responsible for the shard.

This is similar to K-Safety.

Rapid Node Recovery



A failed node can be restored or quickly replaced with a new node.

The new node starts up quickly by filling its cache from peer nodes or directly from S3. Performance is maintained because table locks are not required.

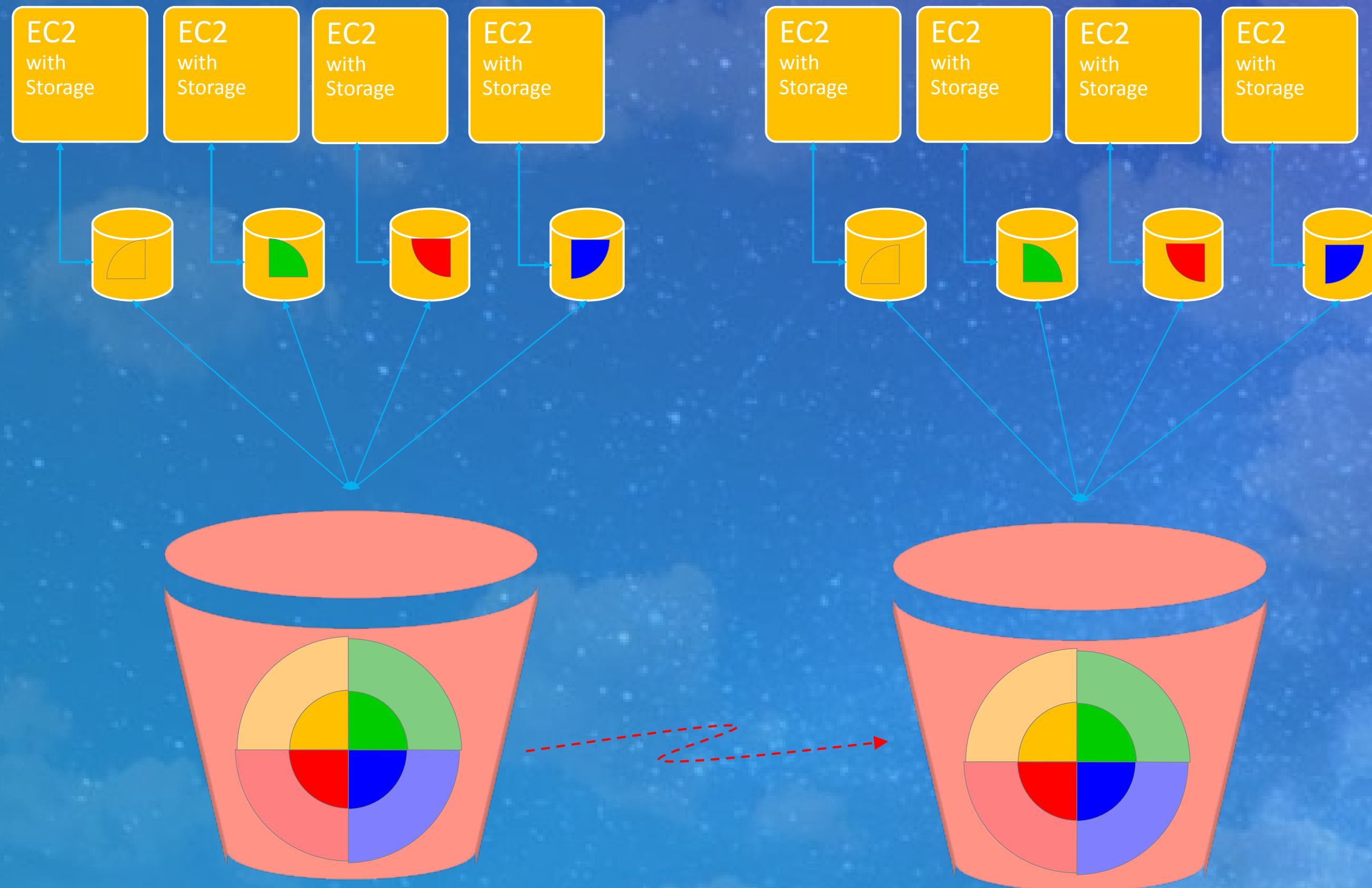
Hibernating



Spin down your entire cluster to save money if your system or project goes dormant or you want to use your Vertica license on other data.

Simply spin it back up when you need it again.

Replication through S3 Snapshots



Snapshot your database for replicas on-demand.

For example, dedicate a cluster to service another geographical region or cost center.

“Revive” a replica by simply adding nodes.



Thank you