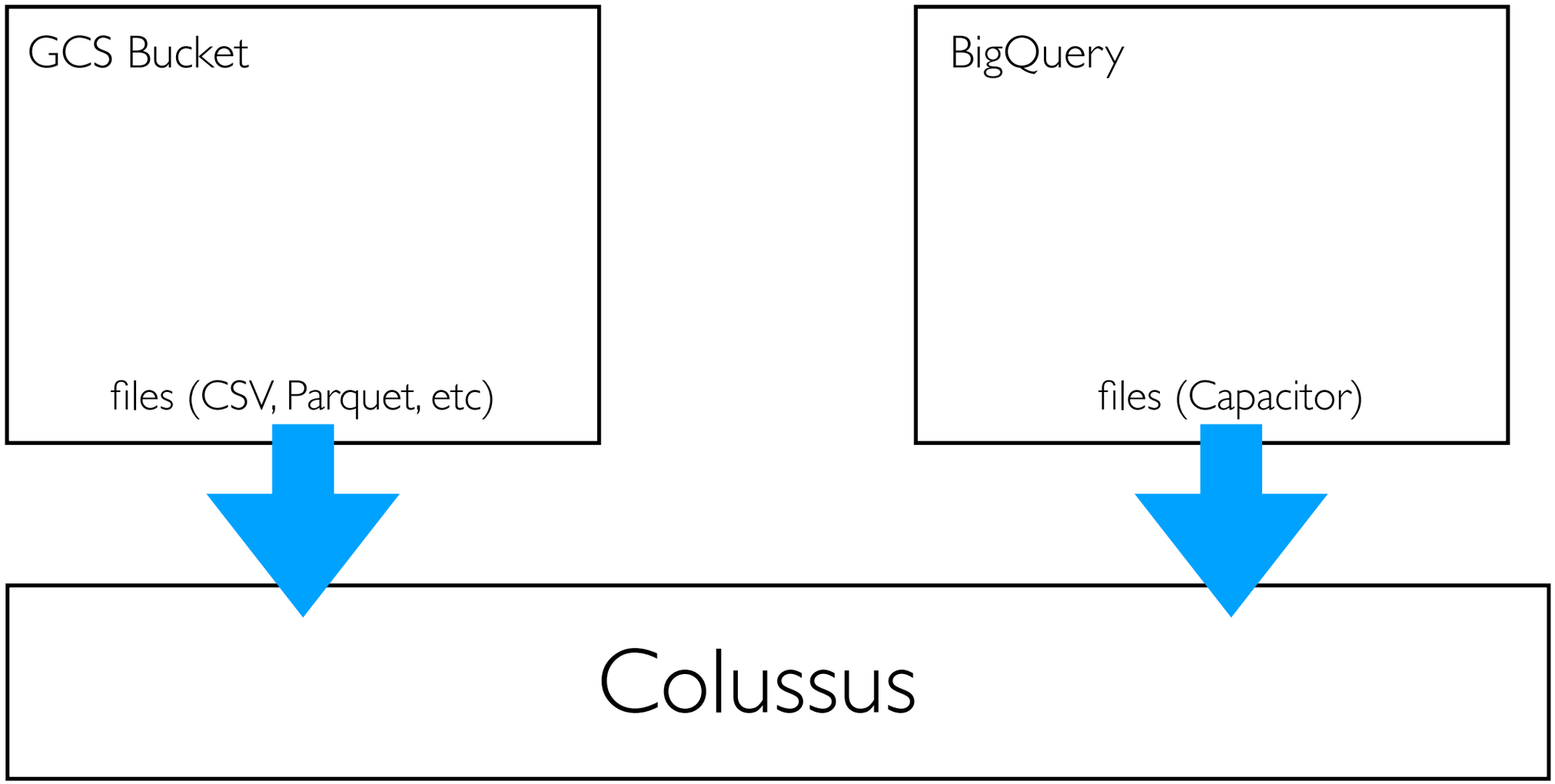
[544] BigQuery: Data Sources and Geographic Data

Tyler Caraza-Harter

Learning Objectives

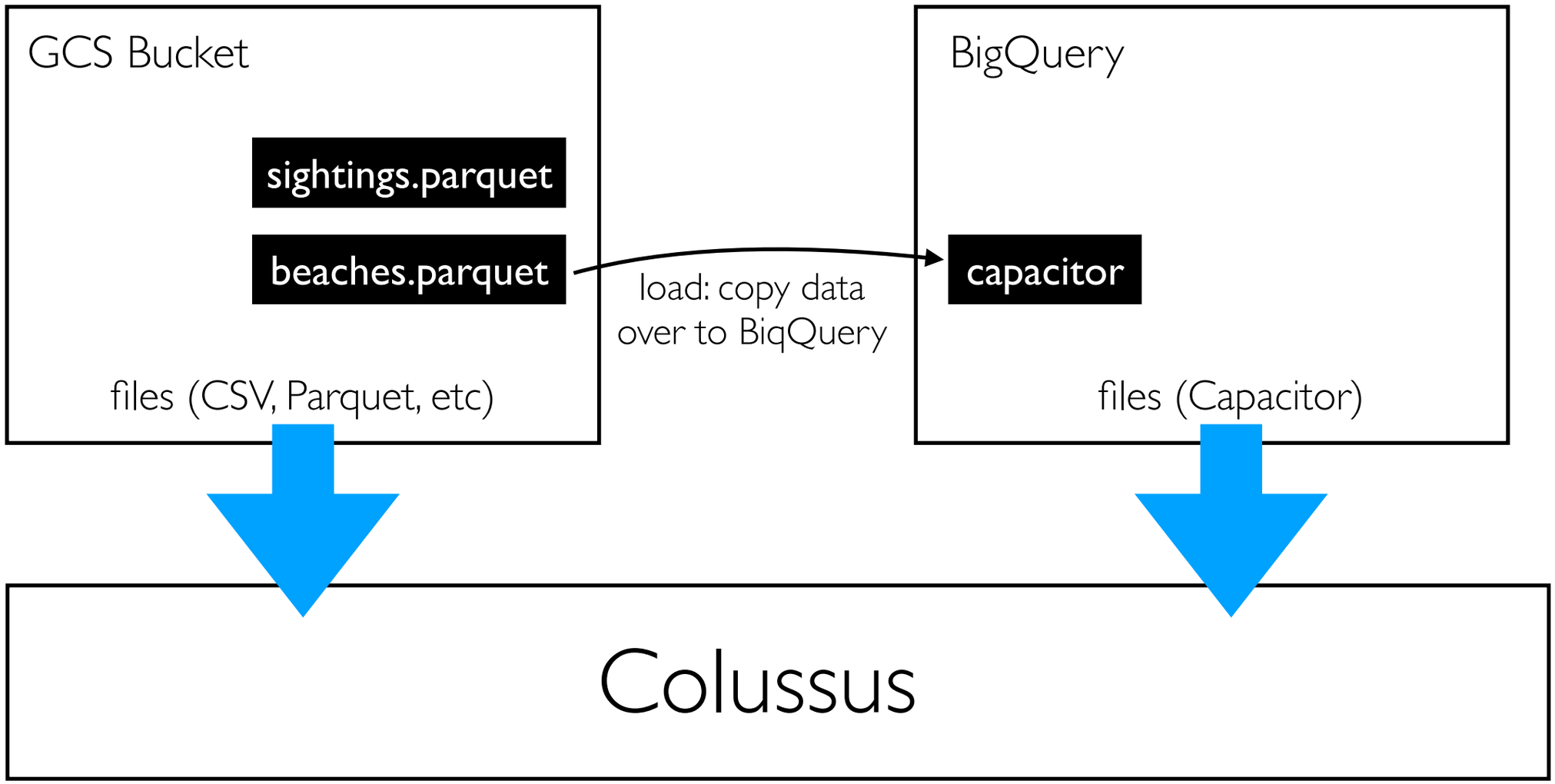
- interact with a GCS bucket
- write load queries to bring data from various sources into BigQuery
- create Dataform pipelines to populate BigQuery datasets:
- perform spatial operations on geographic data

Table Types: Standard Tables, External Tables, and Views



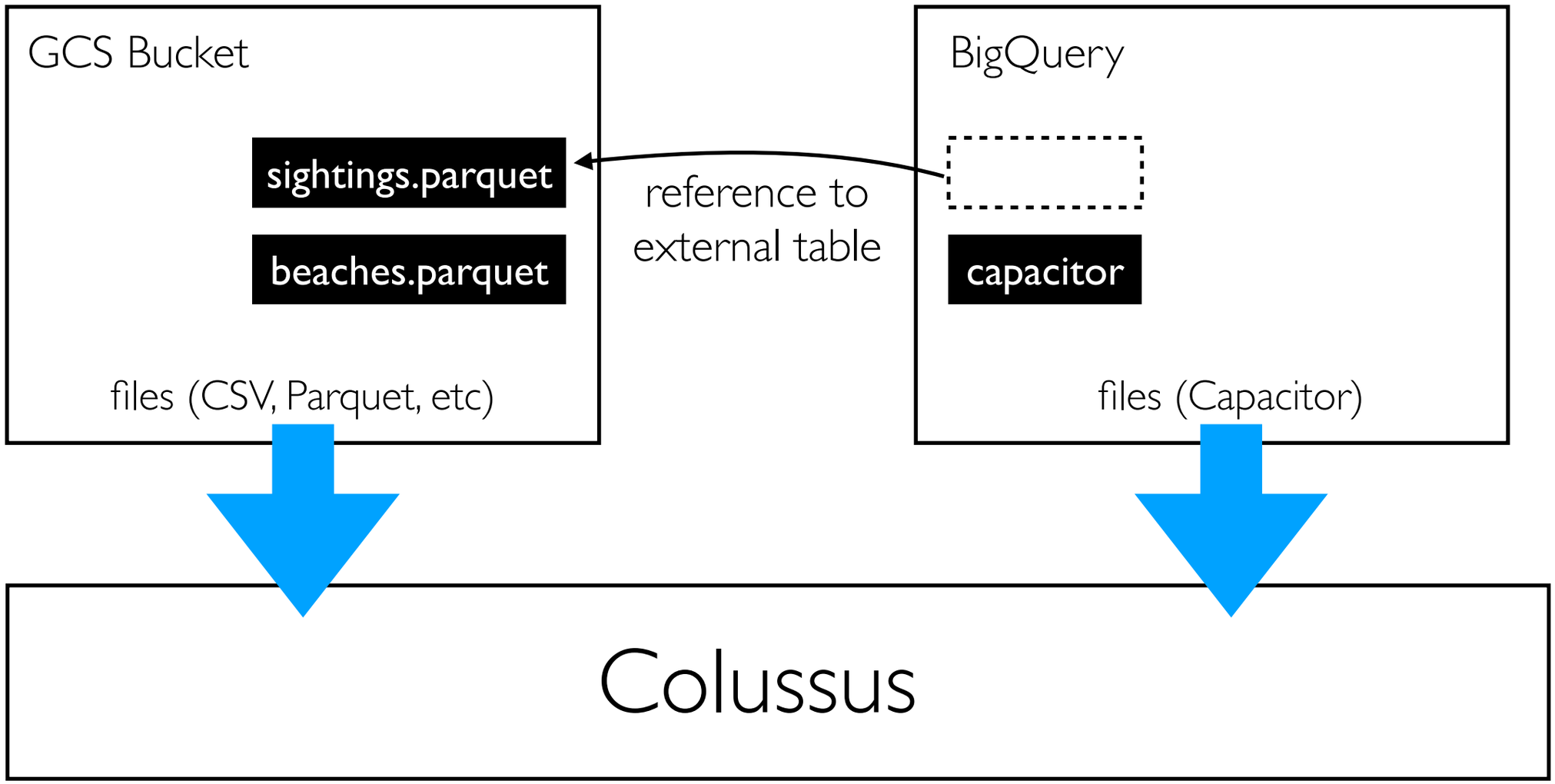
https://cloud.google.com/bigquery/docs/tables-intro
https://cloud.google.com/blog/products/bigquery/inside-capacitor-bigquerys-next-generation-columnar-storage-format

Table Types: Standard Tables, External Tables, and Views



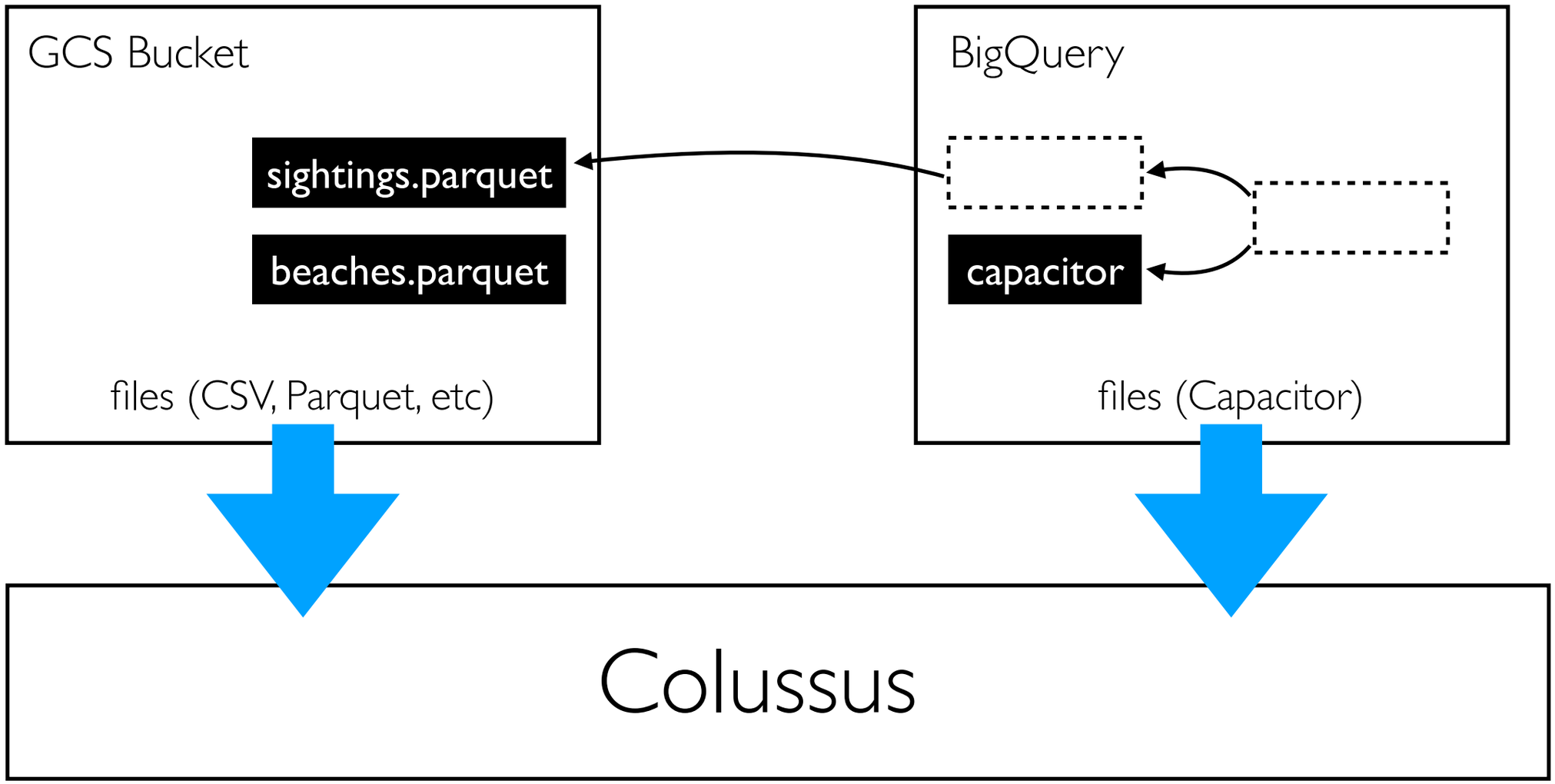
https://cloud.google.com/bigquery/docs/tables-intro
https://cloud.google.com/blog/products/bigquery/inside-capacitor-bigquerys-next-generation-columnar-storage-format

Table Types: Standard Tables, External Tables, and Views



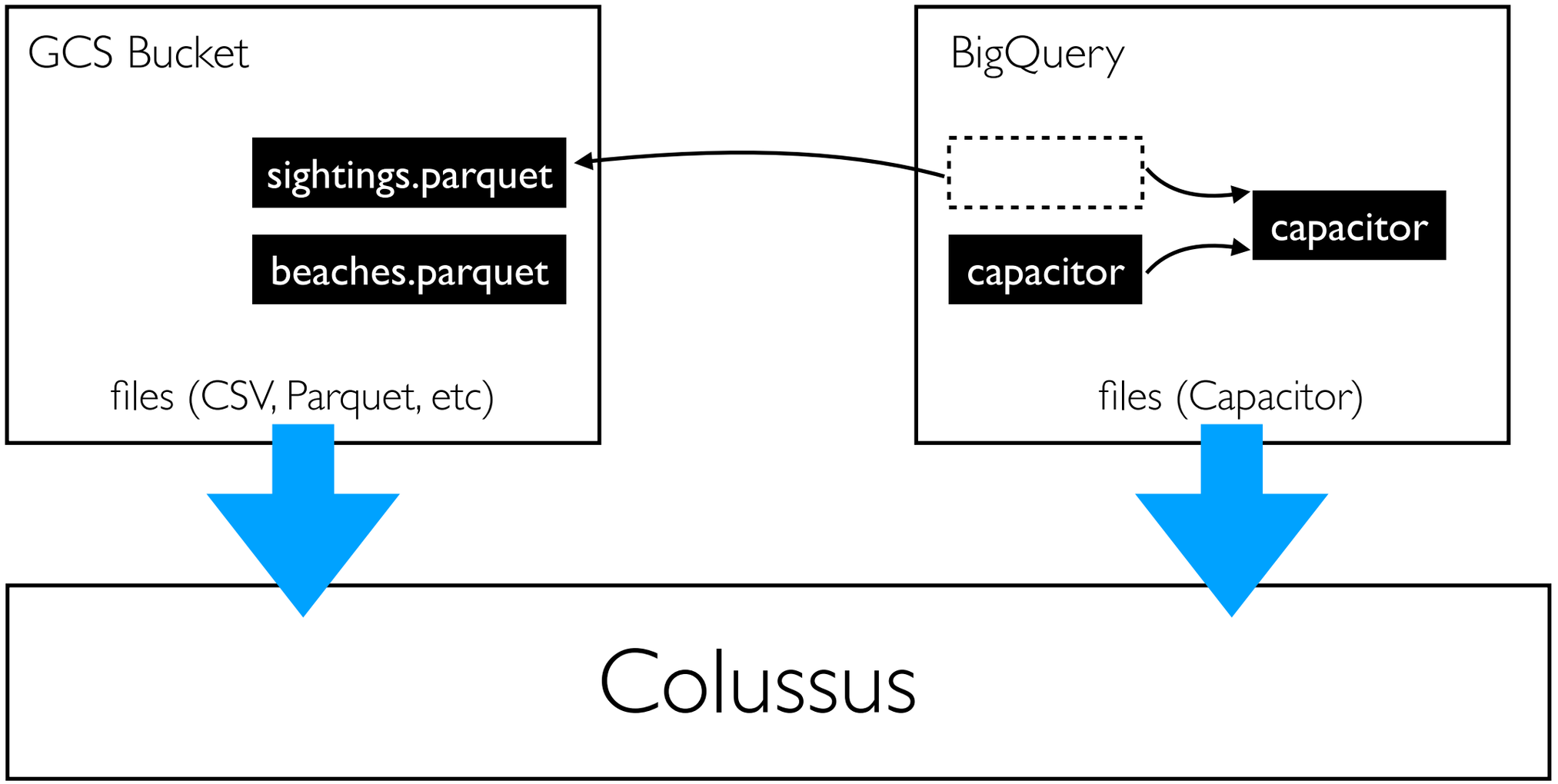
https://cloud.google.com/bigquery/docs/tables-intro
https://cloud.google.com/blog/products/bigquery/inside-capacitor-bigquerys-next-generation-columnar-storage-format

Table Types: Standard Tables, External Tables, and Views



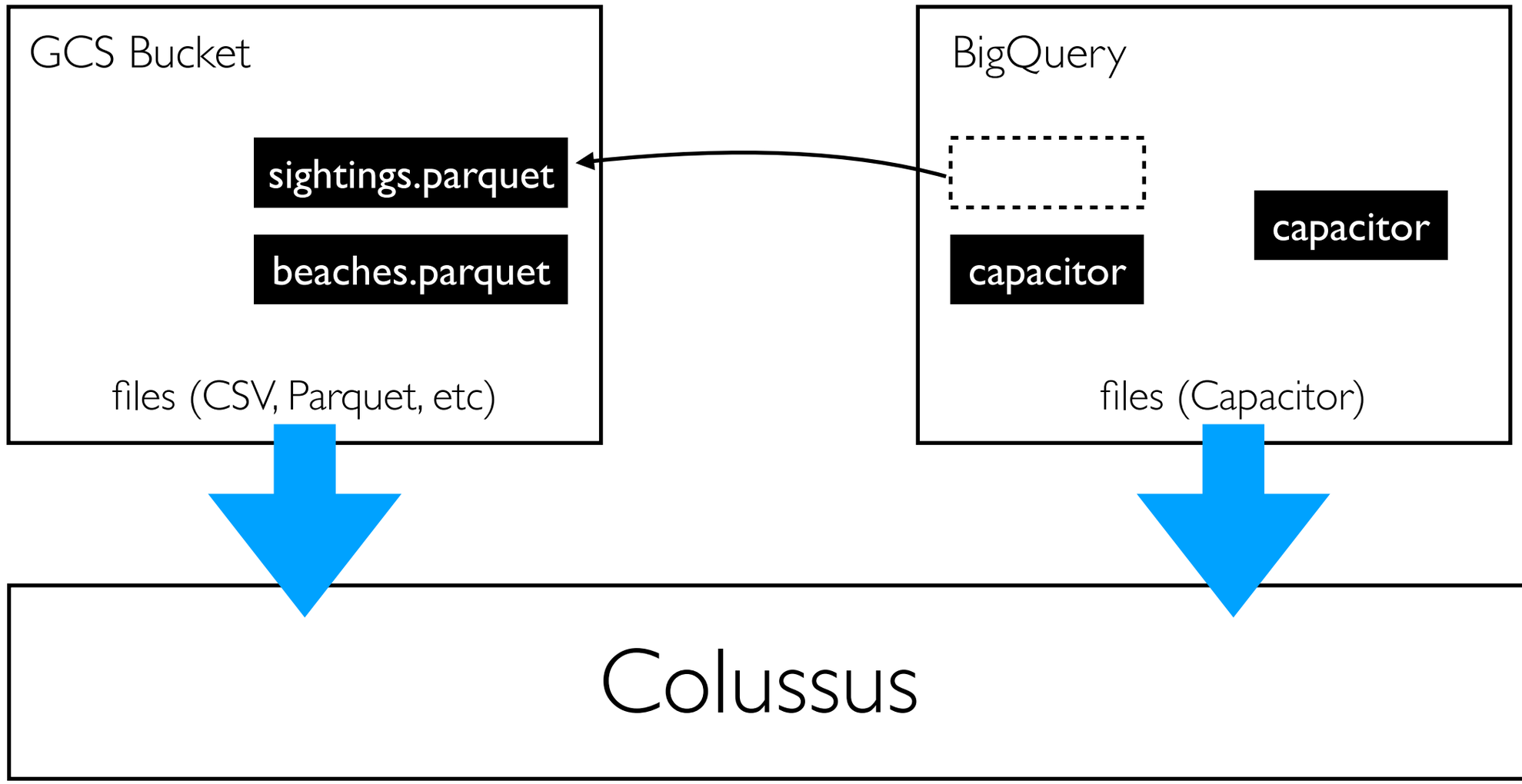

```
CREATE VIEW beach_animals
AS
SELECT s.*
FROM sightings s
JOIN beaches b ON s.beach_id = b.beach_id
ON b.name = "Bernies Beach";
```

Using a Standard Table as a Materialized "View"




```
CREATE TABLE beach_animals
AS
SELECT s.*
FROM sightings s
JOIN beaches b ON s.beach_id = b.beach_id
ON b.name = "Bernies Beach";
```

Performance

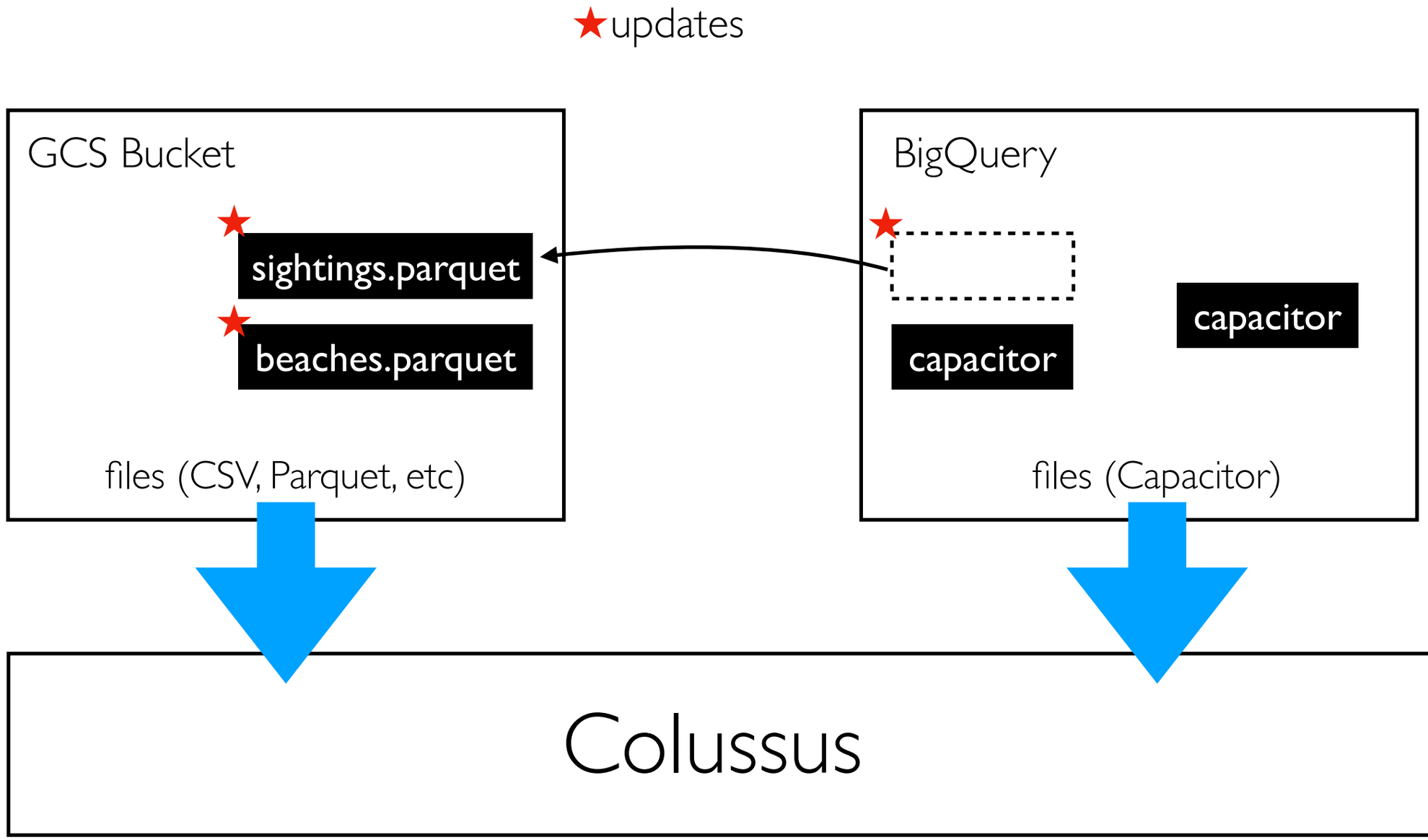


which approach will give us better performance and lower cost when querying beach_animals?

```
CREATE VIEW beach_animals
AS
...
```

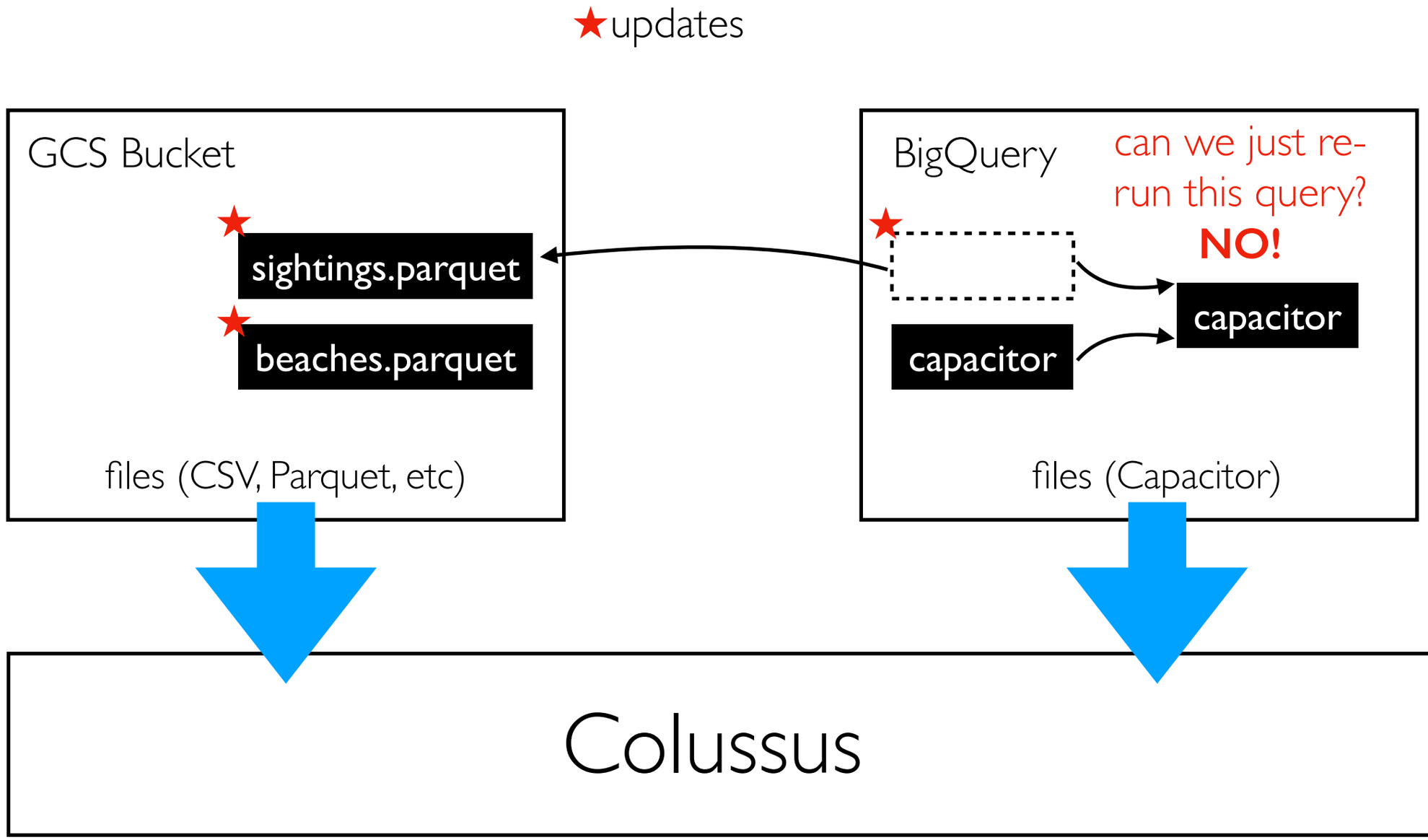
```
CREATE TABLE beach_animals
AS
...
```

Update Visibility



even though tables usually give the best performance and lowest cost, up-stream updates aren't immediately reflected (like they are with external tables and views)

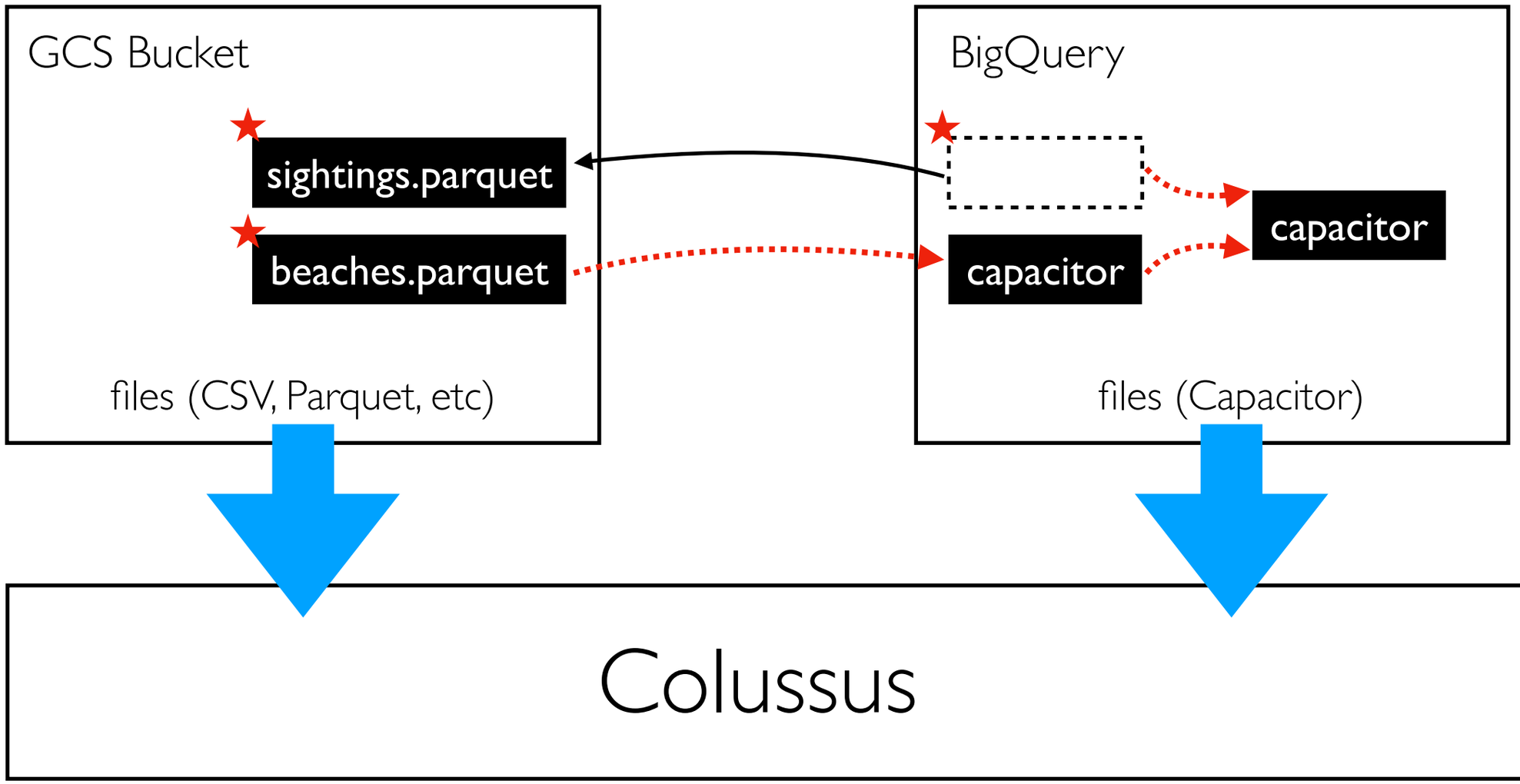
Refreshing Materialized Data




```
CREATE TABLE beach_animals
AS
SELECT s.*
FROM sightings s
JOIN beaches b ON s.beach_id = b.beach_id
ON b.name = "Bernies Beach";
```

Dependency DAGs (Directed Acyclic Graphs)

★updates



we want to keep track of the graph of data dependencies so we can re-run the necessary operations in order to update a specific data resource.

very important for big DAGs (think 100s of nodes!)

Dataform

Key features

- version control on pipeline/DAG code (integrates with Git)
- adds syntax to SQL for specifying references to data objects
- infers DAG structure, enabling runs with dependencies
- scheduler integration

our focus

Cost

- free, because you end up paying for BigQuery when you use Dataform

Demos...