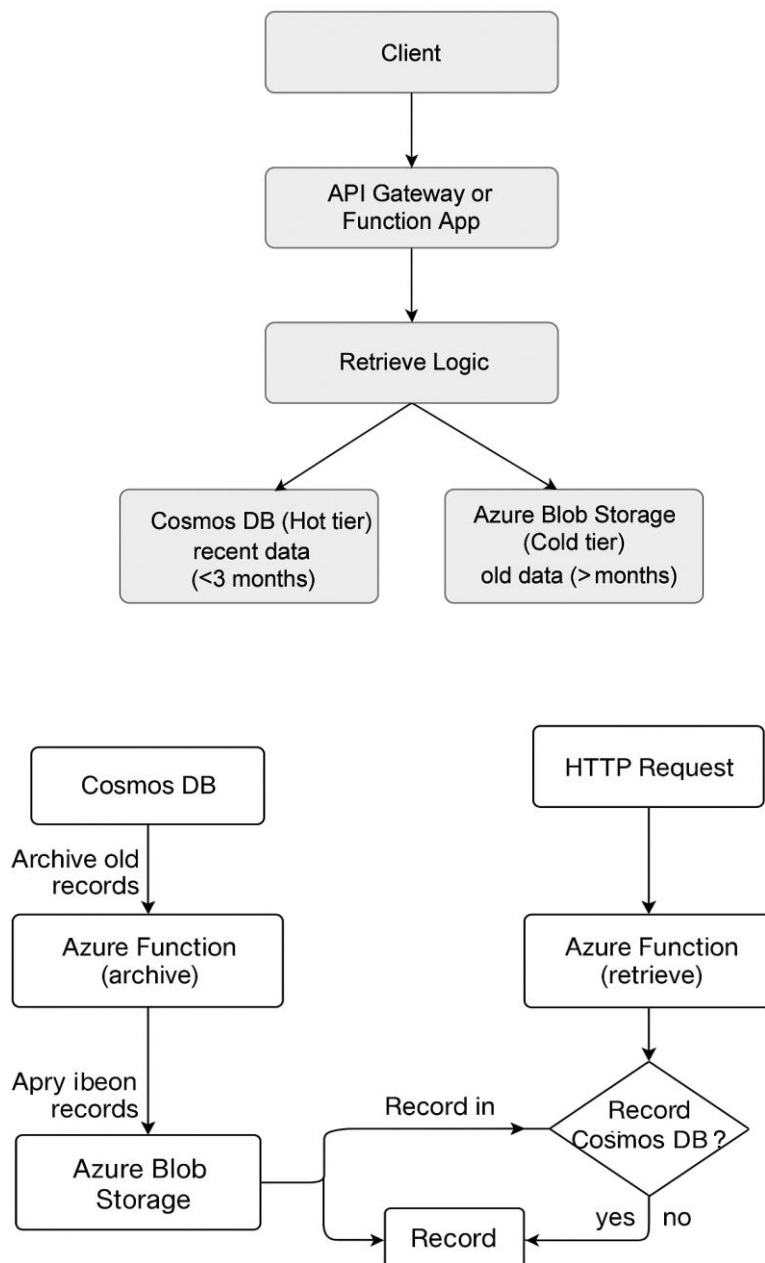


# REPORT

**Azure Cost Optimizer** is a practical solution for managing data growth and optimizing cloud expenditure. This project introduces a streamlined archival pipeline for Azure Cosmos DB that automatically migrates stale records (e.g., older than 90 days) to Azure Blob Storage. The retrieval layer ensures seamless access to both active and archived records, offering a cost-effective and performant data architecture.

## Architecture Diagram



## **archive\_pipeline/archive\_logic.py**

### **Azure Cost Optimizer – Archival pipeline**

#### **\*Algorithm (pseudocode)\*\***

1. Connect to Cosmos + Blob
2. Ensure blob container exists (defensive, idempotent)
3. Compute `threshold\_date = now – N days`
4. Query Cosmos for documents with timestamp < threshold
5. For each document:
  - 5a. Build partitioned blob path <type>/<YYYY-MM>/<id>.json
  - 5b. Upload JSON to blob (overwrite=False)
  - 5c. Delete item from Cosmos DB using partition key (userId)
  - 5d. If userId missing → log + skip deletion
6. Return the number of deleted docs (== archived docs minus skips)

## **retrieval\_layer/retrieve.py**

### **HTTP-triggered retrieval function.**

Fallback logic:

1. Try Cosmos DB first (hot path)
2. If not found, scan blob archive by deterministic path
3. Return JSON record if located, else 404 payload

❑ Try Cosmos DB

try:

```
    item = COSMOS_CONTAINER.read_item(item=record_id,  
partition_key=user_id or record_id)  
  
    return item
```

except cosmos\_ex.CosmosResourceNotFoundError:

```
pass # Fall through to blob lookup
```

#### ❏ Blob fallback

```
# The blob path is deterministic: <type>/<YYYY-MM>/<id>.json, but we
```

```
# may not know `type` or month – iterate or build an index in prod.
```

```
for blob in BLOB_CONTAINER.list_blobs():
```

```
    if record_id in blob.name: # naïve substring match for demo
```

```
        data = BLOB_CONTAINER.download_blob(blob.name).readall()
```

```
        return json.loads(data)
```

#### ❏ Not found

```
return {"error": "record not found"}
```

Thank you