Event Drive Your Apps with Azure Cosmos DB Change Feed

Harpreet Gill

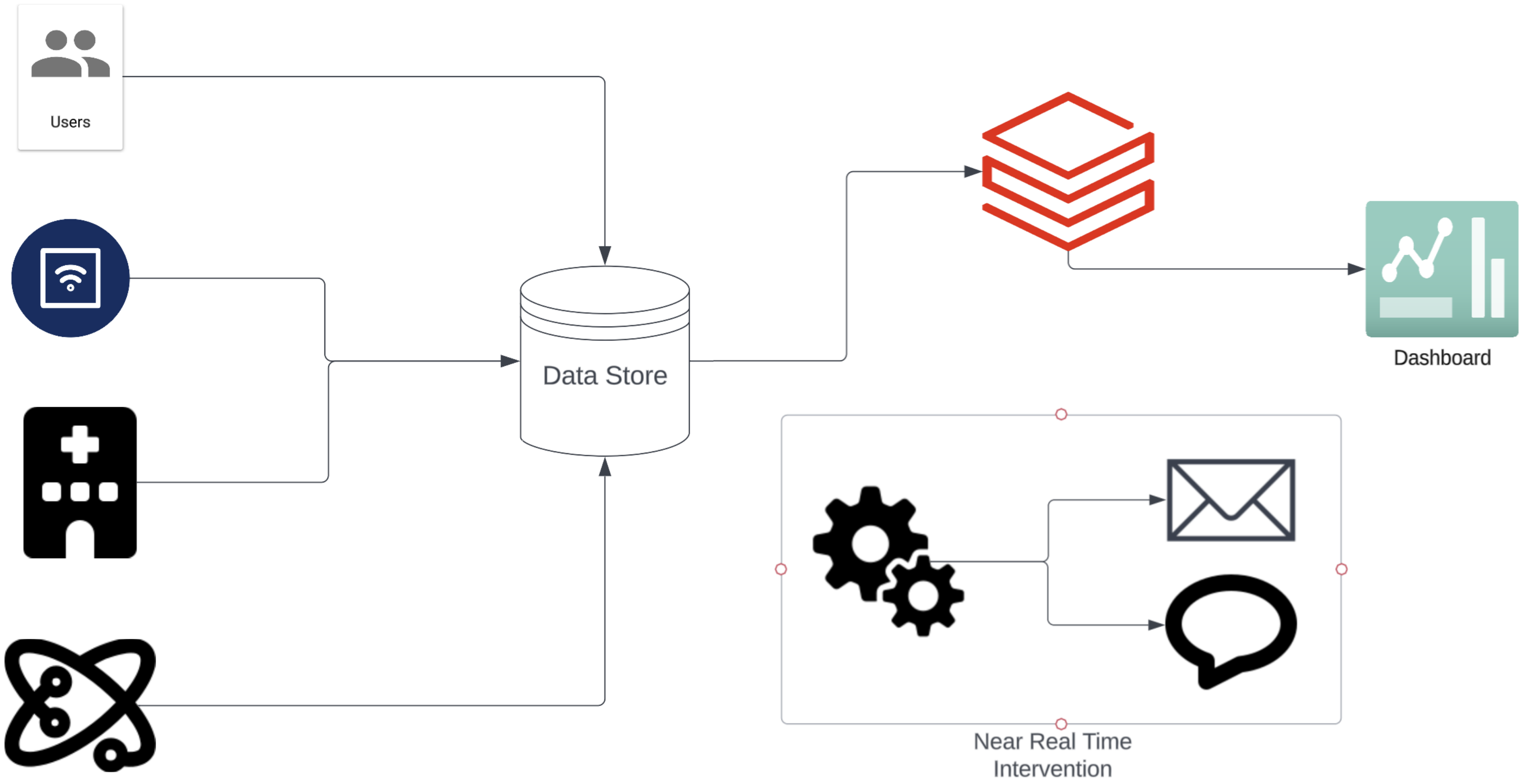
linkedin.com/in/harpreetsgill

🐦 @HarpreetGill03

Why are we here?

- To talk about our Use Case and How Change Feed fits in
- To see how easily Change Feed can bind events
- To see how can we build a two-way SMS system using Comm. Services
- To chat more about Change Feed Implementations

Use Case



What is Azure Cosmos DB?

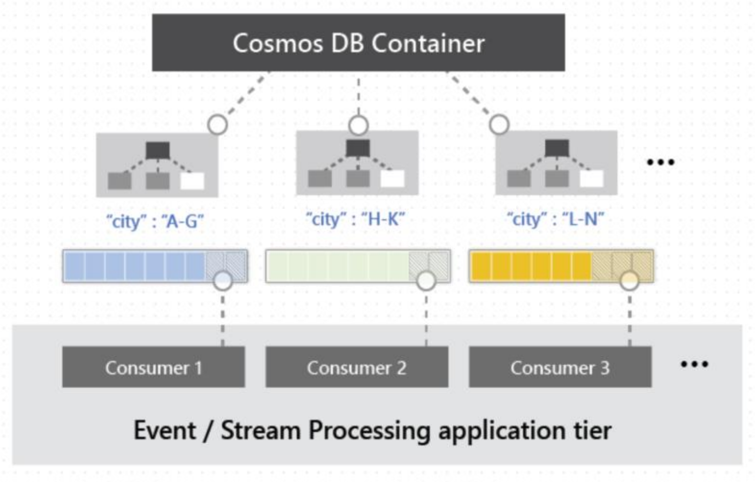
- Fully managed NoSQL and relational database for modern app development
- Provides a cost-effective way to run a serverless and automatically scalable database solution
- Helps simplify application development with easy integrations with other Azure services

Azure Cosmos DB

Source : https://learn.microsoft.com/en-us/azure/cosmos-db/change-feed

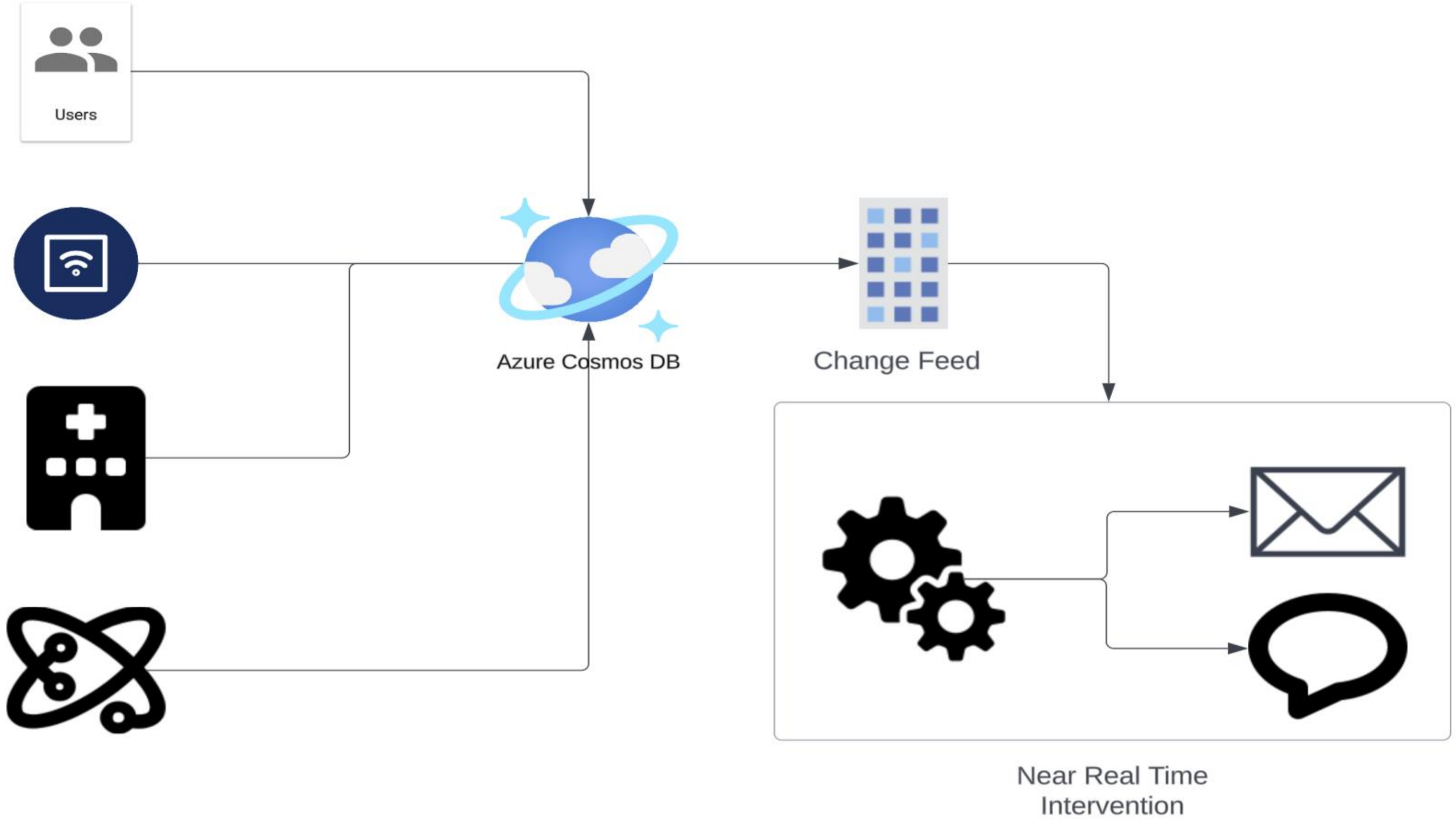
What is Change Feed?

- Persistent Record of Changes to a Container in the Order they Occur
- Feed output can be distributed across one or more consumers for parallel processing
  - Change feed items arrive in the order of their modification time
  - Sort order is guaranteed per logical partition key. Outside partition key, order is not guaranteed.
- Feed Includes Inserts and Updates. Deletes needs other strategies(we will chat more about this)
- Feed can be read via Push or Pull model.



Source : https://learn.microsoft.com/en-us/azure/cosmos-db/change-feed

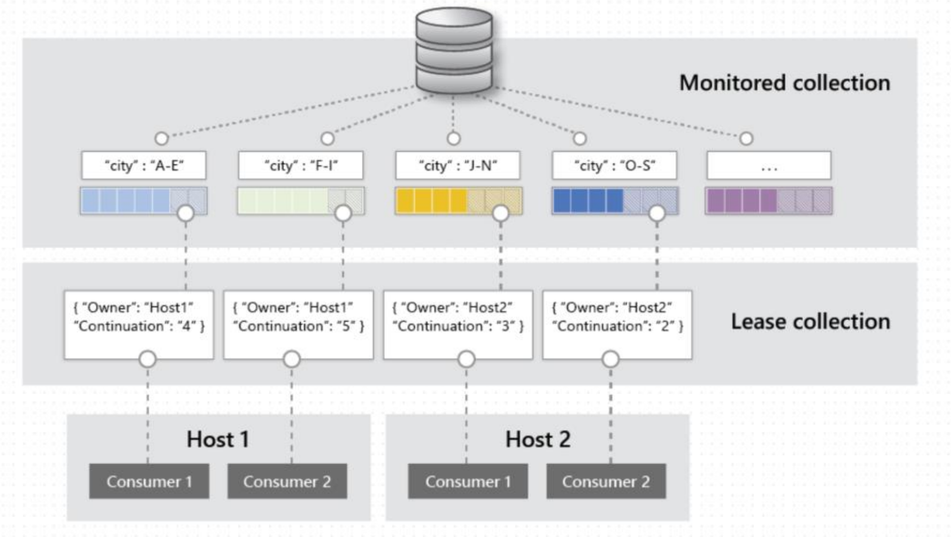
Use Case Revisited with Change Feed



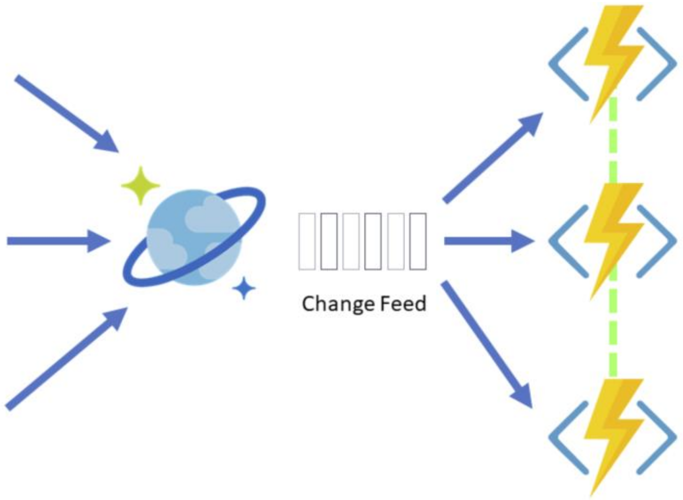
Change Feed Push Model

- Change is Feed is Pushed to a Compute Resource
- No need to Poll Change Feed for Updates.
- Load Balancing and Error Handling

Change Feed Processor


Monitored collection
"city" : "A-E"
"city" : "F-I"
"city" : "J-N"
"city" : "O-S"
...
{ "Owner": "Host1" "Continuation": "4" }
{ "Owner": "Host1" "Continuation": "5" }
{ "Owner": "Host2" "Continuation": "3" }
{ "Owner": "Host2" "Continuation": "2" }
Lease collection
Host 1
Consumer 1
Consumer 2
Host 2
Consumer 1
Consumer 2

Azure Functions



Source - https://learn.microsoft.com/en-us/azure/cosmos-db/nosql/read-change-feed

Change Feed Push Model

Change Feed Processor

- Fault Tolerant, “at-least-once” Delivery
  - Failure on first ever execution are not retired.
- Gives more control over change feed.
- Uses Observer Pattern to Bind Change Feed
- Part of Azure Cosmos DB SDK
- Life Cycle Event hooks
- Can be hosted on Multiple Platforms
  - Long running process friendly

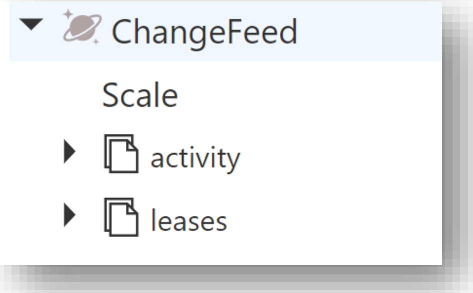
Azure Functions

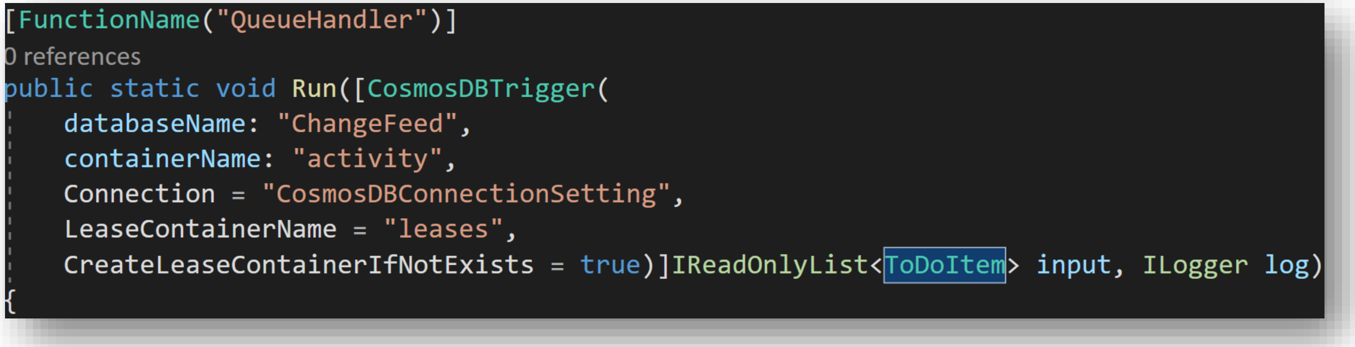
- Uses Azure Functions Trigger to Bind
- Quick and Easy to set up
- Uses Change Feed Processor library but abstracts the details from end users
- Simplifies Deployment of Change Feed Compute

Source - https://learn.microsoft.com/en-us/azure/cosmos-db/nosql/read-change-feed

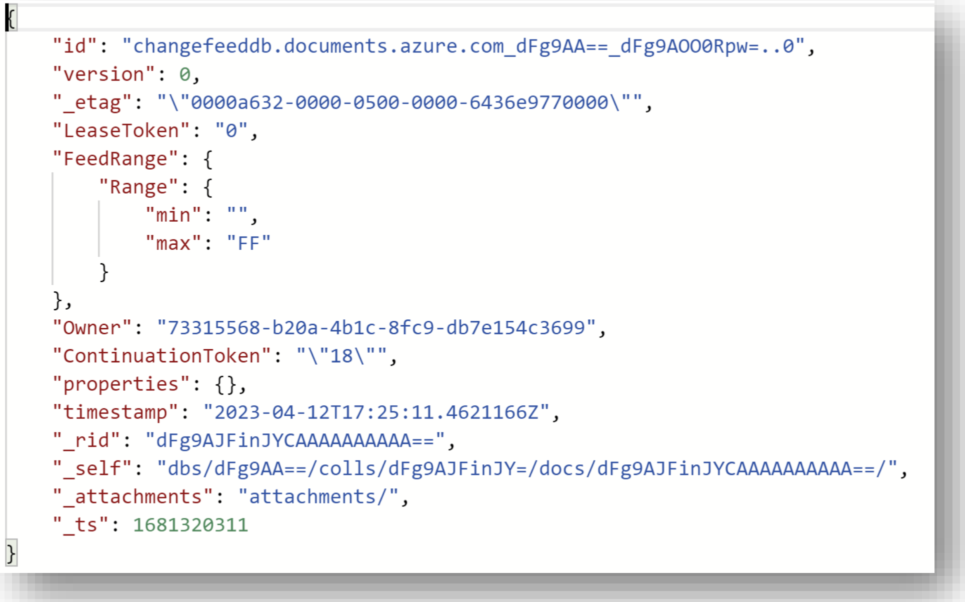
Change Feed Binding – Azure Functions

- **Monitored Container** – Container Being Monitored by Change Feed
- **Lease Container** – Container to maintain state of Azure Functions Feed instances.
  - Multiple Consumers will have different entries.





```
[FunctionName("QueueHandler")]
0 references
public static void Run([CosmosDBTrigger(
    databaseName: "ChangeFeed",
    containerName: "activity",
    Connection = "CosmosDBConnectionSetting",
    LeaseContainerName = "leases",
    CreateLeaseContainerIfNotExists = true)]IReadOnlyList<ToDoItem> input, ILogger log)
{
```



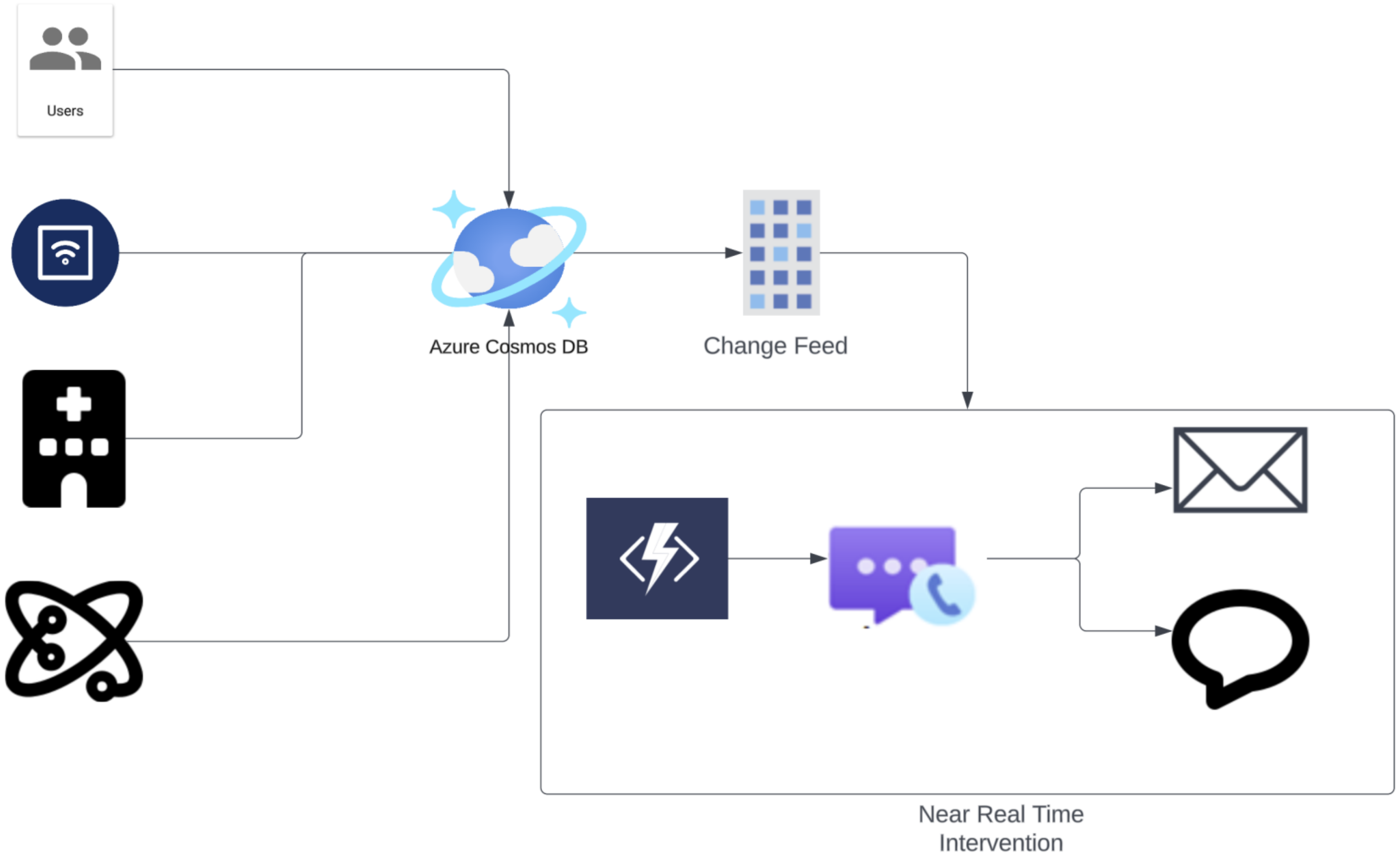
```
{
    "id": "changefeeddb.documents.azure.com_dFg9AA==_dFg9AOO0Rpw=..0",
    "version": 0,
    "_etag": "\"0000a632-0000-0500-0000-6436e9770000\"",
    "LeaseToken": "0",
    "FeedRange": {
        "Range": {
            "min": "",
            "max": "FF"
        }
    },
    "Owner": "73315568-b20a-4b1c-8fc9-db7e154c3699",
    "ContinuationToken": "\"18\"",
    "properties": {},
    "timestamp": "2023-04-12T17:25:11.4621166Z",
    "_rid": "dFg9AJFinJYCAAAAAAAAAA==",
    "_self": "dbs/dFg9AA==/colls/dFg9AJFinJY=/docs/dFg9AJFinJYCAAAAAAAAAA==/",
    "_attachments": "attachments/",
    "_ts": 1681320311
}
```

Change Feed Binding Demo

Azure Communication Services

- Cloud Based Communication Services
  - Voice
  - Text
  - Email
- Office 365 and Azure AD Integration
- Server and Client SDKs to build a complete communication solution

Comm. Services Architecture



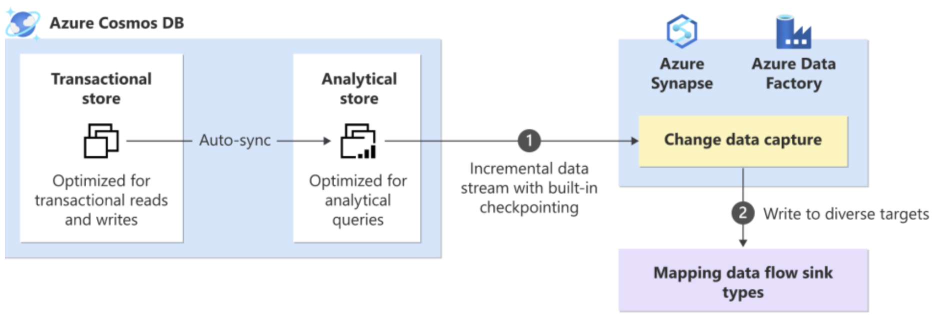
Comm. Services Binding Demo

Change Feed Continued

- **RU Consumption**
  - Change feed read operations on the monitored container consume request units
  - Throttling on monitored container can cause delays to Change Feed

- **Deletes**
  - Soft Marker Strategy – Use a Soft Delete strategy to mark a record delete.
  - Time to Live – Use TTL as a solution to delete items.

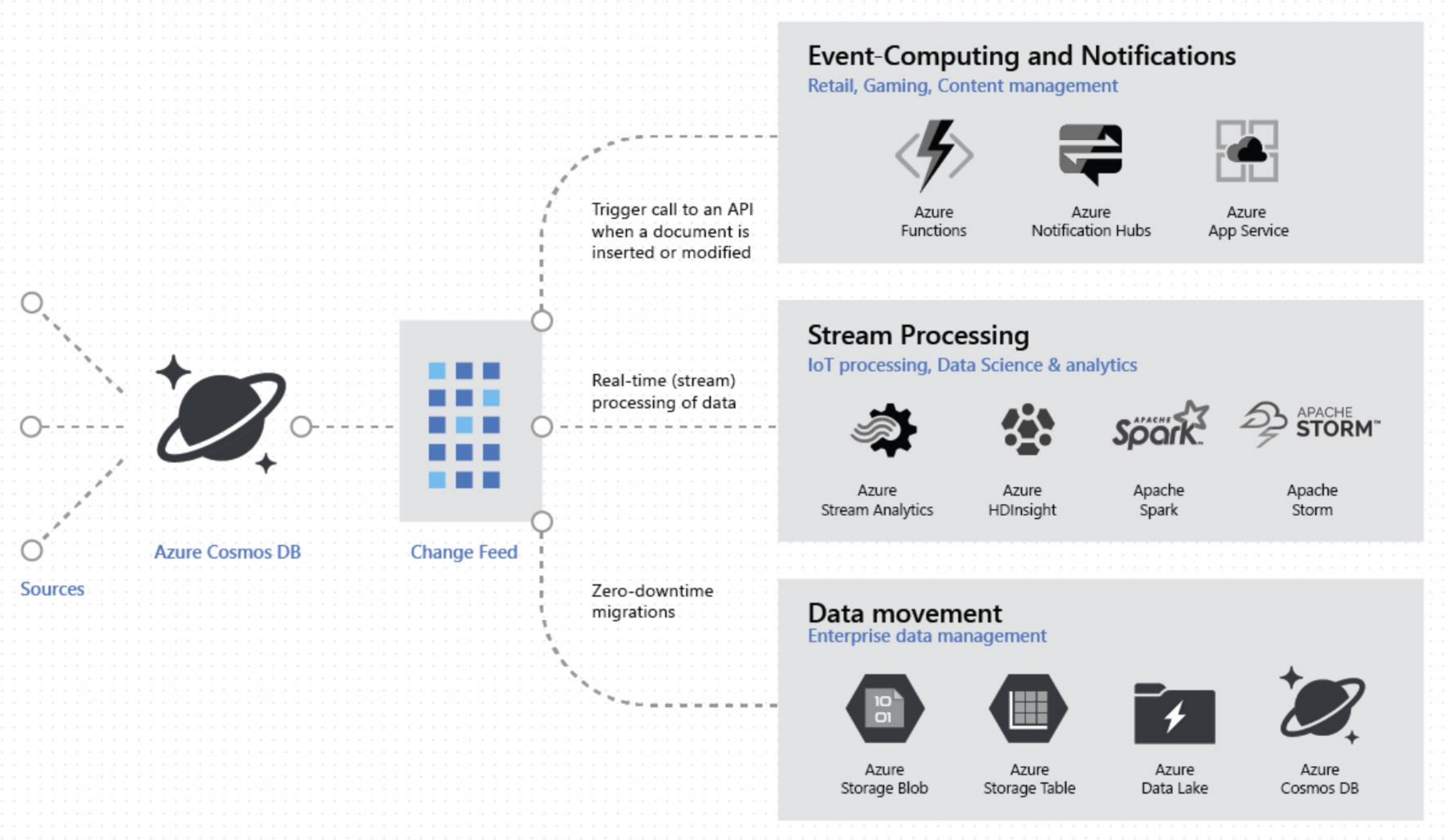
- **Change data capture (CDC)**
  - Consume a continuous and incremental feed of changed data from analytical store (Azure Synapse and Azure Data Factory)
  - Does not consume RUs
  - Captures Deletes


Azure Cosmos DB
Transactional store
Optimized for transactional reads and writes
Auto-sync
Analytical store
Optimized for analytical queries
1
Incremental data stream with built-in checkpointing
Azure Synapse
Azure Data Factory
Change data capture
2
Write to diverse targets
Mapping data flow sink types

Source - https://learn.microsoft.com/en-us/azure/cosmos-db/analytical-store-change-data-capture

Change Feed Continued

- **Design Patterns**



Source - https://learn.microsoft.com/en-us/azure/cosmos-db/nosql/change-feed-design-patterns

Final Thoughts

- Change Feed can event drive your application!
- Each situation is unique, consider how much value you will get from implementing such a system.
- Change Feed is not a full operation log.
- Consider Design Patterns which might work well for your solution.

Children’s Mercy
KANSAS CITY

Research Institute