ArcGIS Online: Best Practices with Hosted Feature Services

Khaled Hassen (khassen@esri.com)

Online Feature Service Development Lead

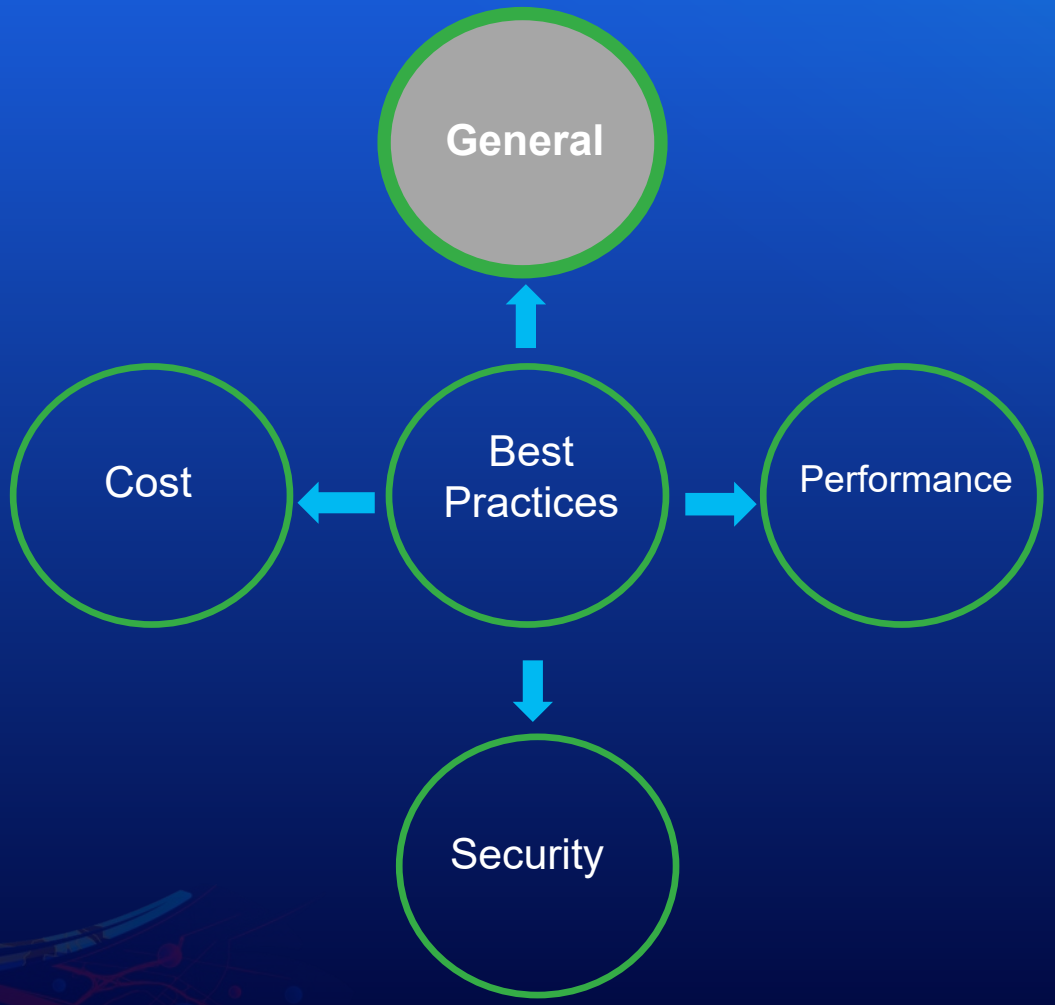
2021 ESRI
DEVELOPER SUMMIT

Best Practices

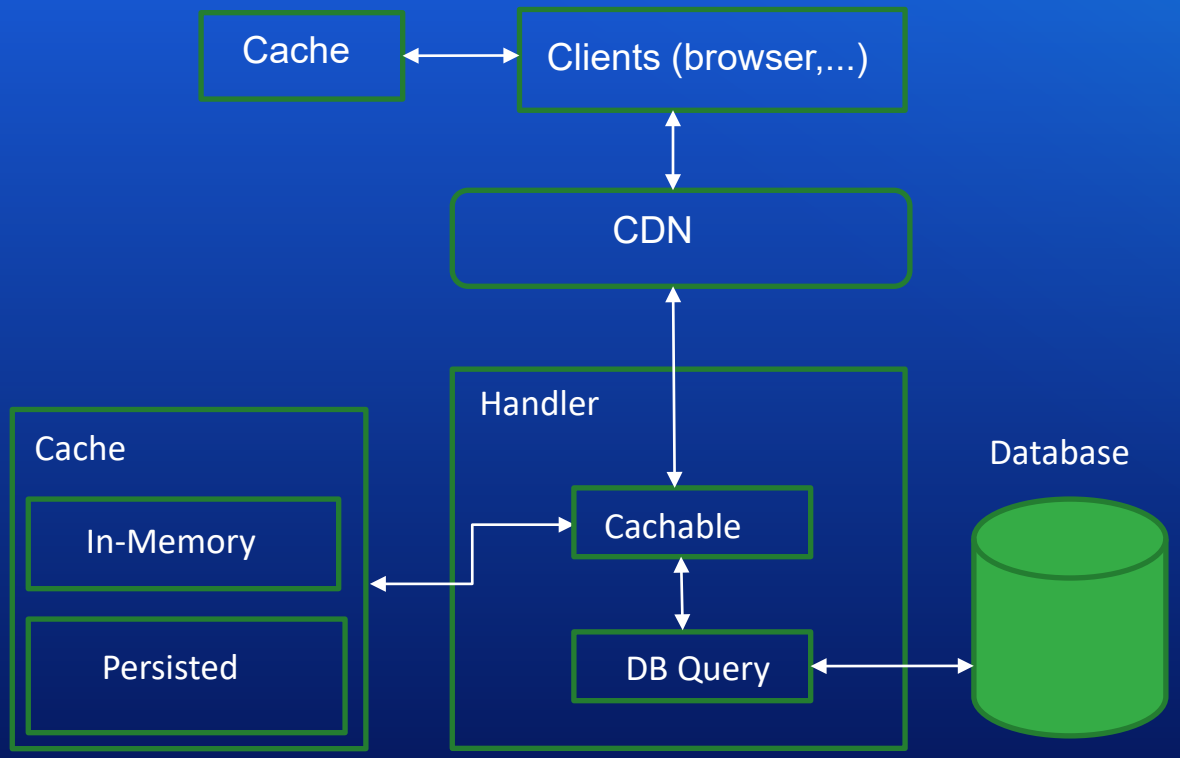
Online Hosted Feature Service

- Online feature service is cloud native SAAS
- Migration is easy with many publishing options
- Full support of WebGIS infomation model and open standards
  - FC / Table / RC / Attachments / Domains / SubTypes, ContigentValues, etc.
  - Geojson, XML metadata specification, GeoPackage, OGC features
- Supports offline use and portal collaboration
- Supports viral load with best practice
- Exposes admin API
- Exposes high performance append API
- Scale up with optional upgrade to premium data stores

Best Practices : General



System Architecture



Keys to Efficient Sharing of Maps and Apps at Scale

- Queries that are identically shared between different concurrent users
  - Tile requests (raster, vector, 3D tiles)
  - Dynamic feature tiles (resultType=tile / cacheHint, auto updated)
  - Queries designed to be shared between different users
    - cacheHint=true
- Response caching
  - Feature server caching
  - CDN edge caching
  - Caching controlled by maxAge
- Client side queries and analytics on retrieved data
  - Helps applications leverage the power of cores on the user's device
- Backed by:
  - Autoscaling of server side compute as needed
  - Scaling up databases to handle massive numbers of unique query loads if present

Best Practices : Maximum Request Rates

Query Request Units (RU or rUnits)

Query Type	Request Units (RU)
Aggregation / Stats	3 Units
Geometry + Attribute	2 Units
All other queries	1 Unit

Org Query Maximum Request Rate

Feature Data Store	RequestUnits/Minute (ru/m)
Standard	10K (166 ru/sec)
Premium (M1)	12K (200 ru/sec)
Premium (M2)	24K (400 ru/sec)
Premium (M3)	48K (800 ru/sec)

Shared Query

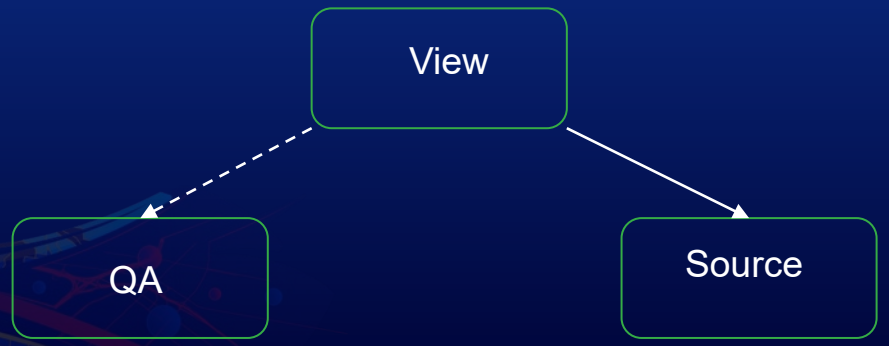
Khaled Hassen

Best Practice : General

- Always use async=true if available
- Add attribute indexes as needed
  - Auto-indexing is now ON for all data
- Use pagination
- Use Webhooks for change notification
- Use data extraction / append to copy data
- Delete features in batches
  - Use rollbackOnFailure=true
  - Use layer truncate API
- Push analytic queries to the database server
  - Windows functions is supported in queryAnalytic API

Best Practices : Data Updates

Pattern	Pros	Cons
Publish / Overwrite	Delete / recreate Preserve itemId	Data is not accessible during update
Live Update - Append	Support upsert Data accessible during append	Does not support deletes, attachments (coming) or Schema changes
View Update	Long publishing job Preserve itemId	During update, data size is doubled.
Sync and Collaboration	Data update is relevant to changes	Requires portal-portal collaboration



Best Practices : Working with GeoTransformations

- Feature service stores GeoTransformations when publishing SD file
- Admin API can be used to change list of supported transformations
- Using geo transformation is transparent during query, editing, and sync
  - Query API supports specifying custom transformation

```
{ "datumTransformations" : [
    {
      "transformDirection" : "forward",
      "transformJson" : "…",
      "fromSR" : {
        "wkid" : 4269,
      },
      "toSR" : {
        "wkid" : 4326,
      }
    }]
… }
```

Best Practices : Performance

Best Practices : Multiscale Geometry

- Supported on all polygon / polyline layers / all projections
- Geometries are duplicated in the data store at different scales
- Transparent to query, editing, and sync operations
- Query execution will be optimized
  - Smaller geometry to quantize
  - Fast database query

```
{
    "multiScaleGeometryInfo" : {
        "levels" : [1,3,5,7, 9]
  },
  …
}
```

Best Practice : Attachments

- Online feature service is designed to handle large attachments
- Use multi-parts when uploading
- Use uploadId when adding or updating
- Return attachmentByUrl instead of base64 (&returnAttachmentsDataByUrl=true)

Attachment: File: Choose File No file chosen

OR

UploadId:

Keywords:

Format: HTML

Add Attachment

Best Practice : Attachments Resizing / Keywords / EXIF

- Attachment image resizing is available in online
  - Supports w=<100, 200, 400, 800, 1600, 2000, 2400>
  - Aspect ratio is preserved
  - Resizing images are cached
- Attachment can have keywords metadata that are searchable
- Attachment EXIF metadata is returned with &returnMetadata=true
  - Used with append API to create features from image collection

Best Practice : QueryAnalytic API

- Compute aggregates and ranking based on SQL windows partitioning.
- Very efficient and avoid the need to perform self join, or post processing
- Can be used to add a new analytic value to the query results (sync or async)
- Supported analytics:
  - Aggregates (MIN, MAX, SUM)
  - Ranking (RANK, NTILE, ROW_NUMBER)
  - Analytics (CUM_DIST / PERCENT_RANK, FIRST_VALUE / LAST_VALUE, LAG / LEAD)

```
{  [{
      "analyticType": "PERCENT_RANK",
      "onAnalyticField": "POP1990",
      "analyticParameters": {
         "orderBy": "POP1990“,
        "partitionBy“ :  "state_name“
      },
      "outAnalyticFieldName": "Rank_Value"
 }]
```

Best Practice : QueryAnalytic Example (LAG / LEAD)

- **Return a previous/next value in a group ordered by an expression or field**

Dept	Year	Sales	Prev_Sales
DC	2015	35000.0	0.0
DC	2016	41000.0	35000.0
DC	2017	81000	41000.0
DC	2018	79000.0	81000.0
DC	2019	91000.0	79000.00

```
{  [{
      "analyticType": “LAG",
      "onAnalyticField": "Sales",
      "analyticParameters": {
         "orderBy": "Year",
         “offset“ :  1 // Default is 1
     }
     "outAnalyticFieldName": "Prev_Sales"
 }]
```

Query Analytics Demo

Khaled Hassen

Best Practice : Sync / Collaborations

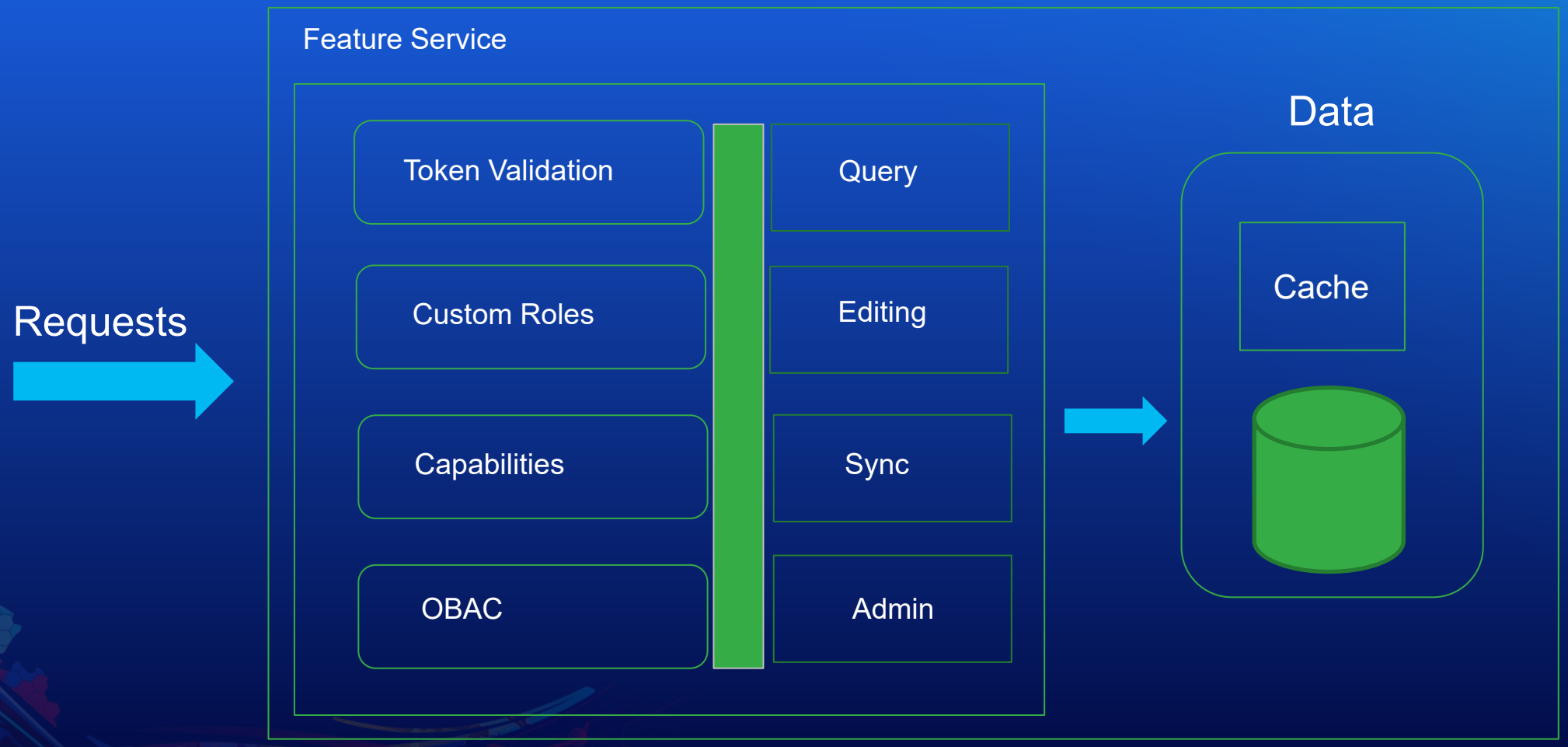
- Create offline maps are expensive
  - Create once and reuse with registration
  - Use pre-planed workflows
- Scheme changes are not currently supported
  - Requires recreating offline maps
- Can expose sync on views
  - Join-View does not support sync
- Use portal collaborations to sync data
  - Online-OnPremise, vise versa, one-way or two-ways sync

Best Practice : Webhooks For Change Notification

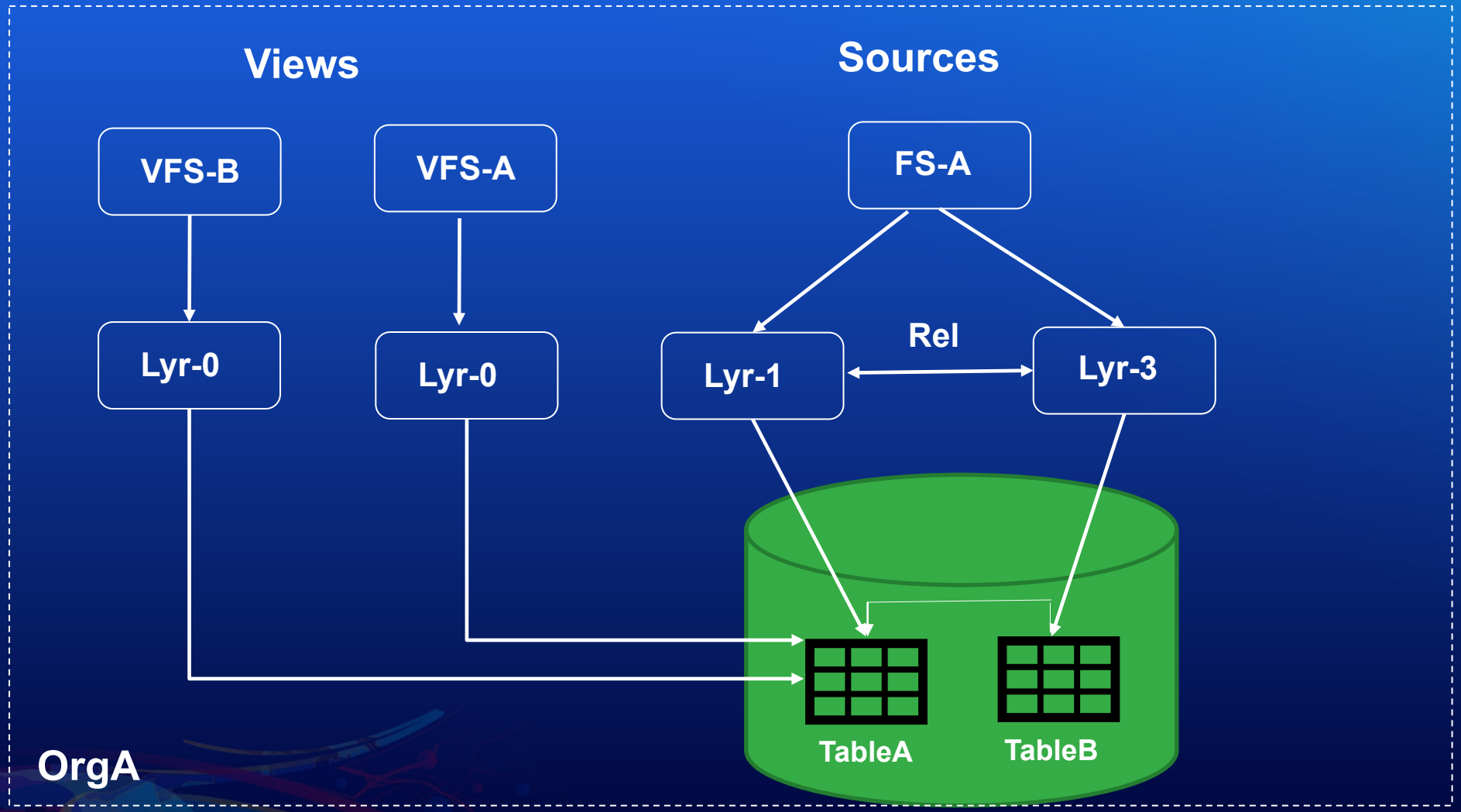
- Cannot avoid throttling with client side polling
- Secured scalable support for near real time change notification
  - Server-server communication
  - Requires webhook url
  - Support all platforms including Integromat and Microsoft Flow
- Change notification is sent as scheduled
  - Min is 20 secs
- You can opt-in to all current and future changes
  - Subscribe only to change you are planning on handling
- Change notification has a similar JSON schema and unique payloads

Best Practices : Security

Best Practices : Security



Best Practice : Feature Service Views



Best Practice : Security / Views

- Customize capabilities as needed
  - Control read access, editing, and sync
- Hide sensitive information
- Unlimited number of views
- Can be a join view

Best Practices : Ownership-based Access-Control (OBAC)

- OBAC allows restricting users from accessing features they do not own
- Can be ON/OFF at any time
- Capabilities and OBAC can be specified per layer

```
{
        "editorTrackingInfo" : {
                "enableEditorTracking" : true,
                "enableOwnershipAccessControl" : true,
                "allowOthersToQuery" : true,
                "allowOthersToUpdate" : false,
                "allowOthersToDelete" : false,
                "allowAnonymousToQuery" : true,
                "allowAnonymousToUpdate" : true,
                "allowAnonymousToDelete" : true
        }
        …
}
```

Best Practices : Layer Overrides

- Layer overrides allow individual layer to customize feature service capabilities, OBAC, allowGeometryUpdates
  - Example: A layer in an editable feature service can have editing OFF
- Layer override is only available in editable feature service
- Extract, Append, Sync still feature service based

```
{
        "layerOverrides" : [
                "allowGeometryUpdates",
                "ownershipBasedAccessControl",
                "allowGeometryUpdates"
        ]
        …
}
```

Layer Overrides

Webhooks and Layer Overrides Demo

Khaled Hassen

XSS Prevention / SQL Injection

- XSS prevention is always enabled
- It is based on fixed white list of tags
  - Extensible only by the server
- Unsafe content will be encoded or rejected
- All SQL are validated for SQL injection
  - Standardized SQL is always enabled
  - No need to use native data store SQL
  - Can use standard SQL to define join views

Best Practices : Cost / Upgrade

- Standard data store subscription is a shared resource model
- Premium data stores
  - Heavy queries, editing, analytics, more storage, more concurrent users
  - Dedicated hardware resources
  - Many tiers available (M1-M4)
  - Fixed cost per month
- When upgrade is needed?
  - Heavy offline sync (thousands of field workers) or analytics queries
  - Predictable high performance level
  - More storage
  - When cost per storage is more than the upgrade

Feature Service Road Map (2021)

- Support OGC features
- Attribute rules
  - Arcade expression
  - Validation rules
- Supports curves and 3D objects
- Support big / temporal data
  - Support of high efficient solution for temporal data
  - Archiving support
  - Handle very large data as a feature service
    - Add new data stores
    - Sharding spatial data

OGC Features Demo

Khaled Hassen

Please provide your feedback for this session by clicking on the session survey link directly below the video.

esri®

THE SCIENCE OF WHERE®

Copyright © 2021 Esri. All rights reserved.