RELTIO

How to Search my Data with Reltio API

Dmitry & Abhradeep

Feb 2023

Agenda

- Intro
    - API-first platform
    - Infrastructure architecture supporting Search
- API and UI scenarios
    - UI search
    - Faceted Search
    - Fuzzy Search
    - Complex filtering
    - Type Ahead Search
    - Saved Search
    - Get by ID / Crosswalk
    - Searching Activities
- Searching Relations and References
    - Relationship Search
    - Searching by the graph - tree, hops and connections
    - APIs coming soon
- Search Performance and Latency

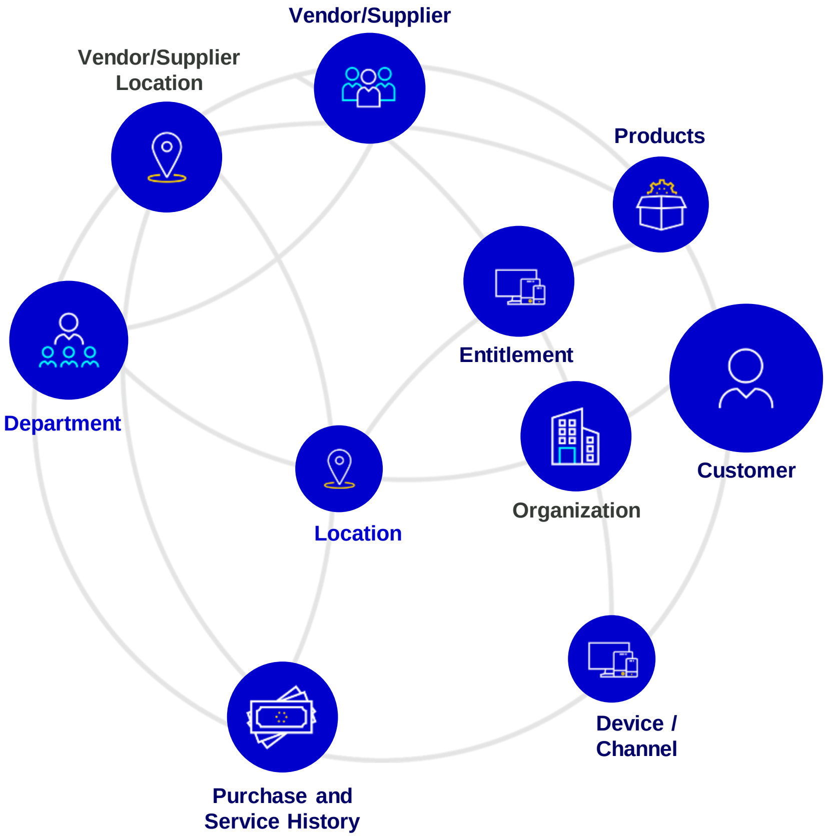
RELTIO

Introduction

Dmitry

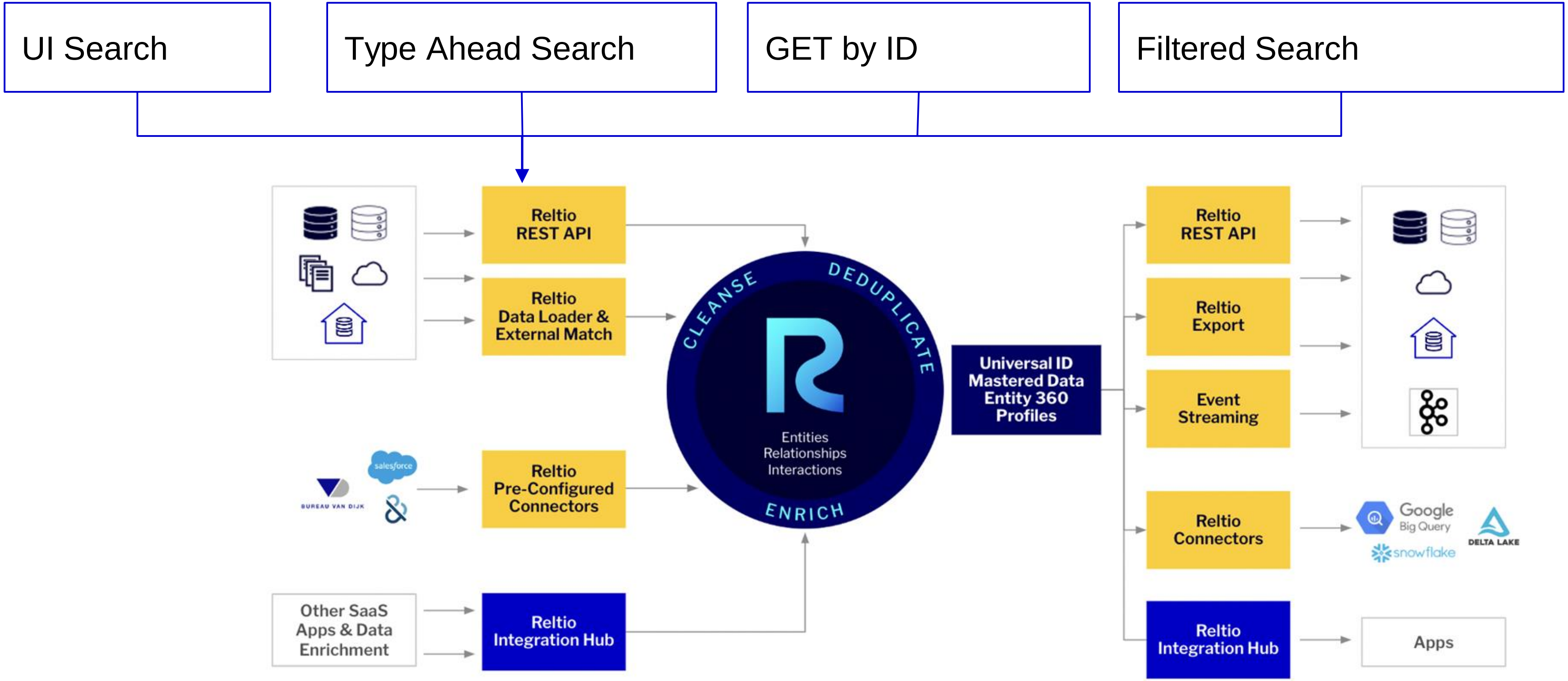
Reltio helps to unlock and accelerate value from your data

- **360° view** by unifying customer, supplier, product, and location data with transactions and interactions
- Instant access to **timely, trusted information** across all systems
- **See relationships** among people, organization, products, locations, suppliers, and more
- **Rich customer, supplier and product insights** by rapidly enriching data from external sources



RELTIO

Reltio is an API-led platform



UI Search

Type Ahead Search

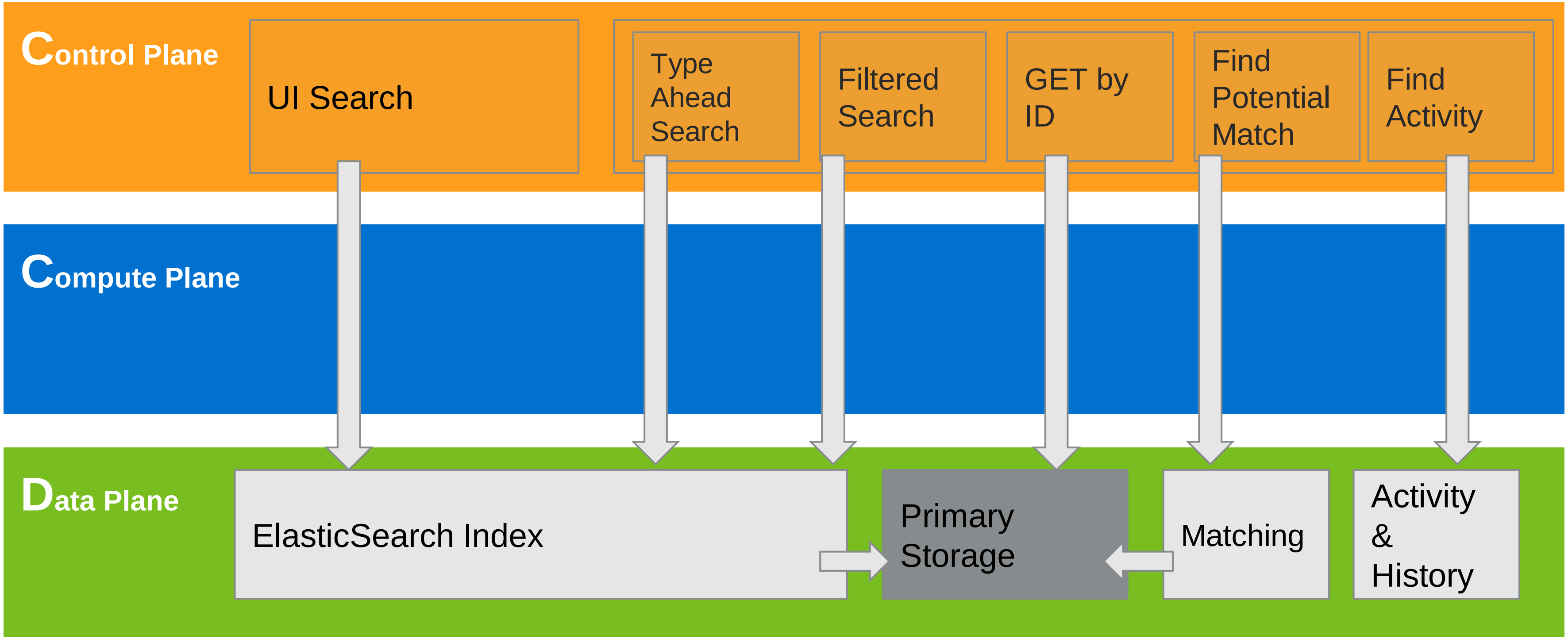
GET by ID

Filtered Search

RELTIO

3-plane Architecture and Search

PLANES SEPARATE CONTROL, COMPUTE AND DATA CONCERNS



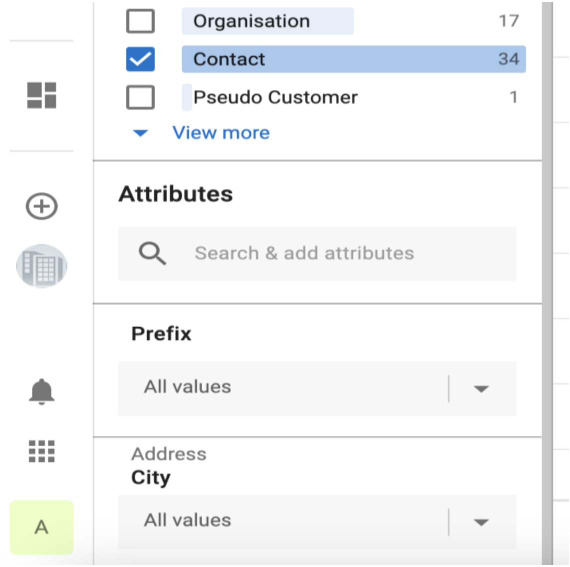
RELTIO

Reltio API & UI
Entity Search Scenarios

Abhradeep

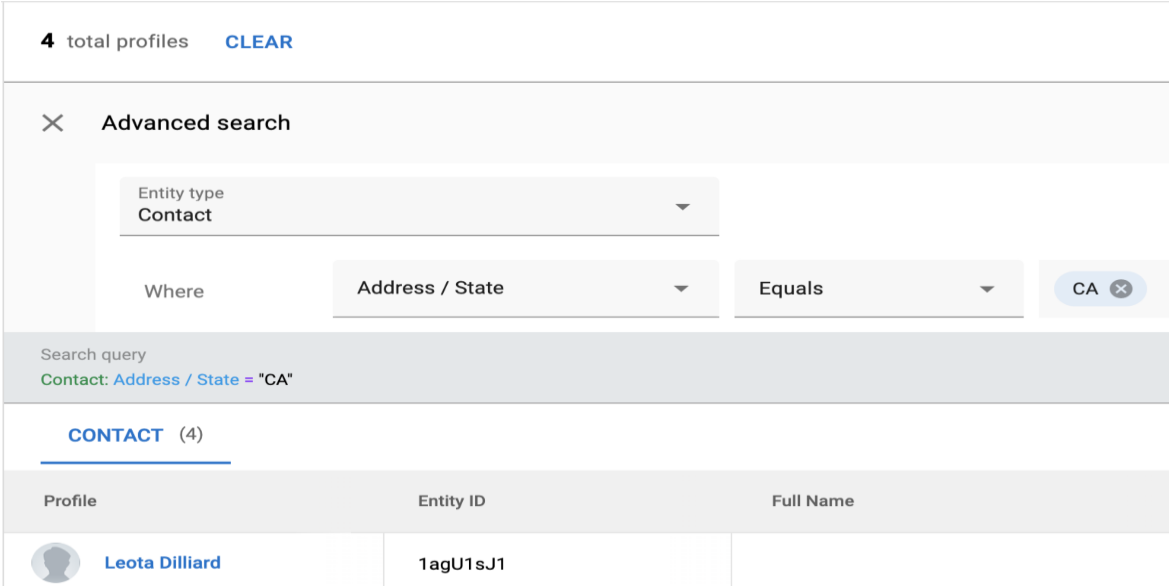
Different Searches in Reltio UI

Faceted Search



☐ Organisation 17
☑ Contact 34
☐ Pseudo Customer 1

This is Advanced Search



4 total profiles

Entity type: Contact

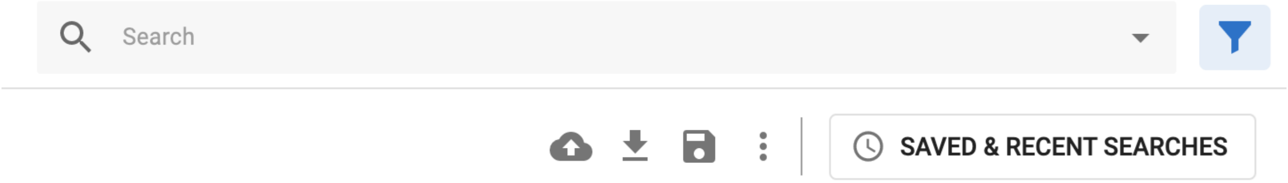
Where: Address / State Equals CA

Search query
Contact: Address / State = "CA"

CONTACT (4)

Profile	Entity ID	Full Name
Leota Dilliard	1agU1sJ1	

Global Search
& Saved Search



RELTIO

Faceted Search

Definition: The entities search considers all entities, but the entities faceted search considers entities with the facet term only

Use Case : Use Faceted Search API to get a distinct list of values with numbers

Examples:

GET {{tenantURL}}/entities/_facets?facet=roles,type&filter=(equals(type,'configuration/entityTypes/Organization')).

Faceted crosswalk count:

GET {{tenantURL}}/entities/_facets?facet=crosswalks_count&filter=(equals(type,'configuration/entityTypes/Contact'))

For multiple Facets - we can use POST or GET API

GET {{tenantURL}}/entities/_facets?facet=attributes.Prefix,type&filter=(equals(type,'configuration/entityTypes/Contact'))

Fuzzy Search

What is Fuzzy Search? - The fuzzy match operator is useful to match terms even when there are minor differences between the query and indexed data. This is not available in UI.

Use Case : This is one of the most commonly used search APIs . This allows you to find appropriate matches even when you make a typographical error or minor spelling errors or not sure about the exact search term

Examples:

GET {{tenantURL}}/entities?filter=((equals(type,'configuration/entityTypes/Contact') and fuzzy(attributes.LastName,'Smit'))) & max=200

This will return contacts with last name = ‘Smith’ due to fuzzy nature of the request

Complex Filtering Based advanced Search

Definition: This supports combination of multiple search operators to form complex query based API

Supported Operators: Contains , Starts With , Less Than(equal to) , Greater Than (equal to) , regexp , exists , missing , not , in , range

Use Case : This is one of the most commonly used search APIs where users can apply complex filtering logic to get back list of specific entity types

Examples:

Filetring (Complex)

```
GET {{tenantURL}}/entities?filter=((equals(type,'configuration/entityTypes/Contact') and fuzzy(attributes.LastName,'Smit') and startsWith(attributes.Email.Email , 'robin'))) & max=200
```

```
GET {{tenantURL}}/entities?filter=((equals(type,'configuration/entityTypes/Contact') and contains(attributes.Email.Email , '*reltio*'))) & max=200
```

Search entities by crosswalk count

```
GET {{tenantURL}}/entities?filter=(equals(crosswalks_count,3))&max=20
```

Global / Type Ahead Search

Definition: JSON array of entity objects for each type from tenant matching filter request in format, order that is defined by query parameters.

Use Case : This is one of the most commonly used search APIs where users can apply complex filtering logic to get back list of specific entity types

Options: Both POST (Preferred) and GET methods are possible

Examples:

POST {{tenantURL}}/entities/_typeAheadSearch

```
{
 "filter": " (containsWordStartingWith(attributes, 'Copy')",
 "select": "uri,label,attributes._lookupCodes,attributes._lookupValues",
 "max": 10,
 "options": "sortByOV,ovOnly",
 "activeness": "active",
 "scoreEnabled": true
}
```

Comma-separated list of different options. Available options:

- **sendHidden**- disabled by default, entity's JSON will contain hidden attributes if this option is enabled.
- **searchByOv**- disabled by default, to search by all attributes with ov only.
- **ovOnly**- return only attribute values that have ov=true flag.
- **nonOvOnly**- return only attribute values that have ov=false flag. If you have a nested or reference attribute value where ov=true, but sub-attributes where ov=false, then these sub-attributes will not appear in the response.

Working with Saved Searches - GET & CRUD

Definition: User can save search queries for convenience and keep it as personal or share with others.

Use Case : When user wants to manage saved searches using API

Examples:

GET {{tenantURL}}/personal/savedSearches OR {{tenantURL}}/personal/allSavedSearches

For deleting any saved searches

DEL {{tenantURL}}/personal/savedSearches/<Saved Search ID>

Update Saved Search

PUT {{tenantURL}}/personal/savedSearches/<id> - In the body you need to provide the updated JSON (from response of GET)

Create Saved Search

POST <tenant URI>/personal/savedSearches - In the body you need to provide the search details

Get By Entity ID / Crosswalk (Primary Storage Based)

Entity ID Definition: This operation returns an entity object (full or partial) by URI from the tenant.

Examples:

GET {{tenantURL}}/entities/1BUjHPHI - all attributes

GET {{tenantURL}}/entities/1BUjHPHI?select=uri,type,tags - with select clause for few attributes

- sendHidden: disabled by default,contains hidden attributes
- ovOnly: return only attribute values that have the ov=true flag.
- nonOvOnly: return only attribute values that have the ov=false flag.

Crosswalk Definition: This operation returns an entity object (full or partial) by crosswalk type and ID value from tenant

Examples:

GET {{tenantURL}}/entities/_byCrosswalk/129207-1?type=BvD

- sendHidden:,contains hidden attributes
- ovOnly: return only attribute values with ov=true.
- nonOvOnly: return only attribute values that have the ov=false flag

RELTIO

Search Activity Log

Definition: This operation will return activities happened in the tenant for a certain time period based on query parameters

Note: You can search Activity Log Records up to a maximum number of 10,000 items. You may use Search Activities with Cursor API to avoid this limitation. Also, you may use Activities Export API for large amounts of data

Use Case : Availability of audit log for Reltio UI and external applications

Examples:

GET {{tenantURL}}/activities - This will return list of all activities for the tenant

The maximum number of Items being indexed (thus, the amount of information that is used for searching) is controlled by the maxChangedObjects parameter in the Tenant Storage Configuration

Activity objects property that must be used for sorting. Can be used in the combination with the "order" parameter to have reverse order. Default sorting is by timestamp in descending order.

Filtered Query:

GET {{tenantURL}}/activities?filter=(equals(user,'jyoti.bardiya@reltio.com'))&max=20

Search with Cursor

Definition: It returns a character sequence to the API user as a part of the response. All subsequent requests are returned with the cursor and then the next portion of data in the defined order. As soon as all data is returned, there will be only a cursor (without data) in the response.

Use Case : Search Entity with Cursor returns the records that are matched with specific criteria in a strictly defined order.

Note :

Query is required in first request in sequence (scan request without filter would cause an error)

Request Body contains the cursor definition. It is needed for all requests except first one.

Examples:

Initial request - **POST {TenantURL}/entities/_scan?filter=equals(attributes.Address.StateProvince, 'CA')&max=100**

Subsequent requests:

POST {TenantURL}/entities/_scan

```
{
 "cursor": {
  "value": "cXVlcnlBbmRGZXRjaDsxOzE0NDI3OmpzdTdBNGNnUWU2YlBqc1JQbTlNbnc7MDs="  }
}
```

RELTIO

Searching Relations and References

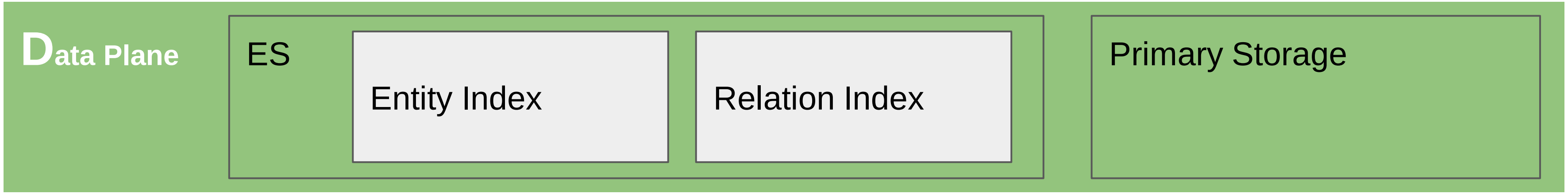
Dmitry

Relation Search - Need Relation Index Enabled

- Search capability for Relations requires **separate index**
- Index for Relations can be **enabled on demand**
- You can search Relations Records up to a maximum number of **10,000 items**
- You can use the same parameters as with entities search: **max, select, filter, order, options, sort** etc.
- Relation Index enables the API for Search Relations, for example:

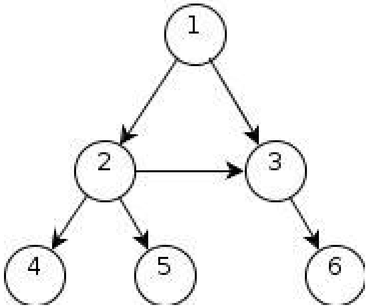
GET {tenantURL}}/**relations**?filter=(equals(startObject,'entities/1BUjHPHI'))&max=2

- Result: JSON array of relation objects from the tenant matching filter request



Hops/ Connection Search Relation APIs

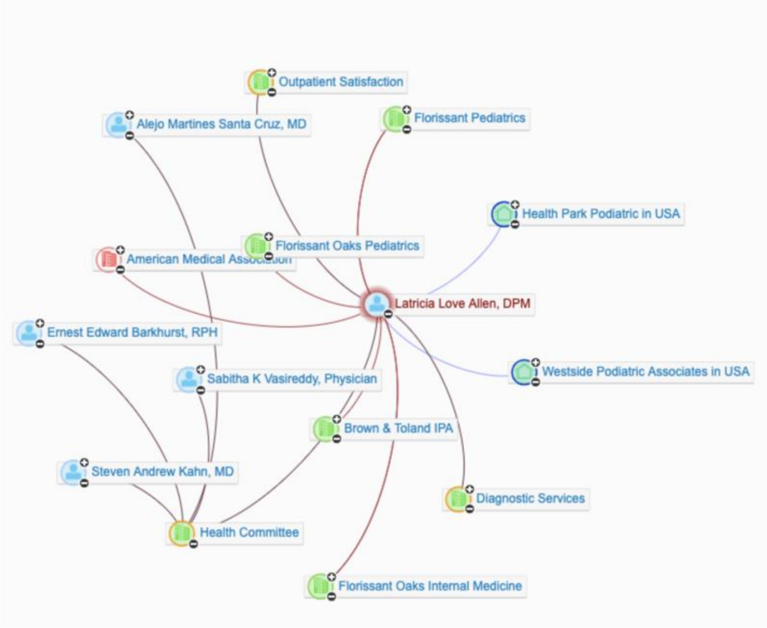
- Relationships are the links between entities, which together with the entities form the Reltio Entity Graph
- Reltio’s relationship definition includes
    - Name & Description
    - Direction - undirected, directed, bi-directional
    - Start and End entities for the relationship
    - Directional context
    - Base attributes



- APIs to search through the Graph:
    - **Find Tree** (search for the object and return the entire tree of relations and objects attached to it) : GET {TenantURL}/{entity object URI}/_tree
    - **Find Connections** (return full graph for the object with selected depth): POST {tenantURL}}/entities/QpaL088/_connections
    - **Find Hops** (return full set of traversed objects and their relations with the selected depth): GET {{tenantURL}}/entities/QpaL088/_hops

Coming soon - new API to search connected objects

- Find Connected Parties
- Find Connections
- Find Connections by crosswalk
- Find Connections by URI
- Search before Save


Outpatient Satisfaction
Alejo Martines Santa Cruz, MD
Florissant Pediatrics
Health Park Podiatric in USA
Florissant Oaks Pediatrics
American Medical Association
Latricia Love Allen, DPM
Ernest Edward Barkhurst, RPH
Sabitha K Vasireddy, Physician
Westside Podiatric Associates in USA
Brown & Toland IPA
Steven Andrew Kahn, MD
Diagnostic Services
Health Committee
Florissant Oaks Internal Medicine

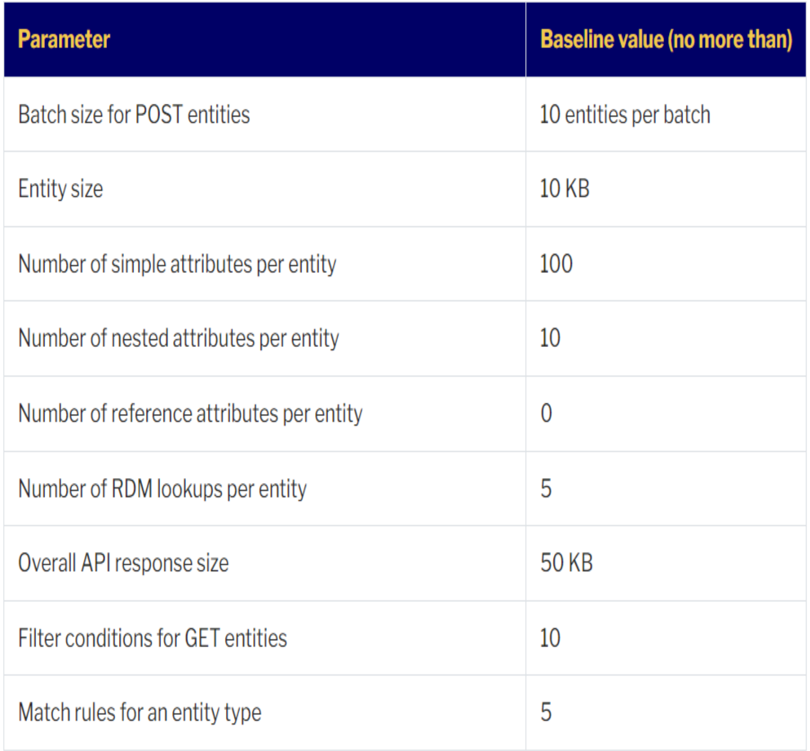
RELTIO

Search API Performance and Latency

Dmitry

Data Factors impacting Search API Latency

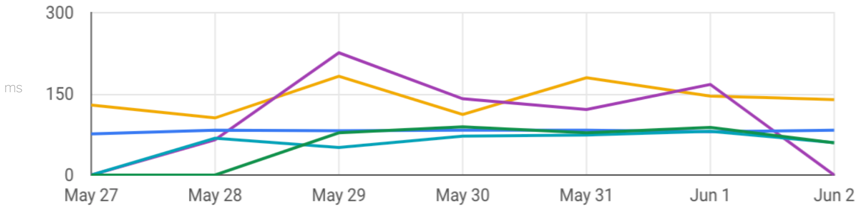
- Every 10 filter conditions increase latency by 1% on average
- Each 10 additional crosswalks may increase latency by 40% on average
- Each 5 extra results in the search request (5 objects found) may increase latency by 40%
- 5 additional lookups may increase latency by 40%
- Every additional reference attribute increase latency by 80%

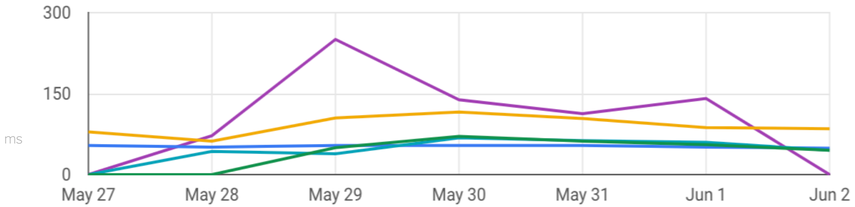


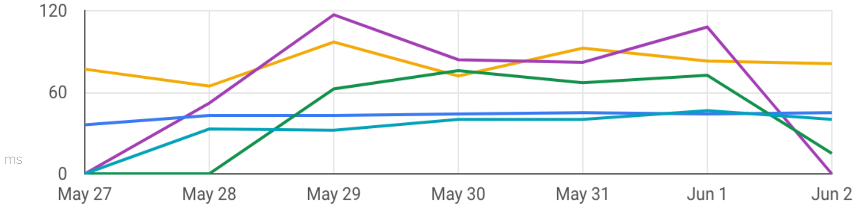
Parameter	Baseline value (no more than)
Batch size for POST entities	10 entities per batch
Entity size	10 KB
Number of simple attributes per entity	100
Number of nested attributes per entity	10
Number of reference attributes per entity	0
Number of RDM lookups per entity	5
Overall API response size	50 KB
Filter conditions for GET entities	10
Match rules for an entity type	5

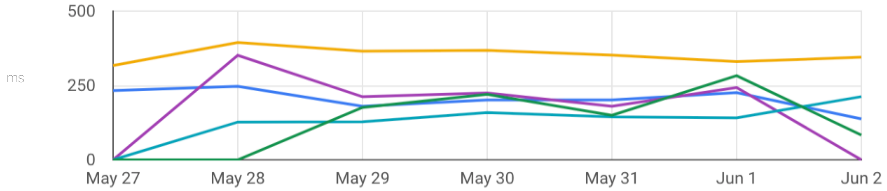
RELTIO

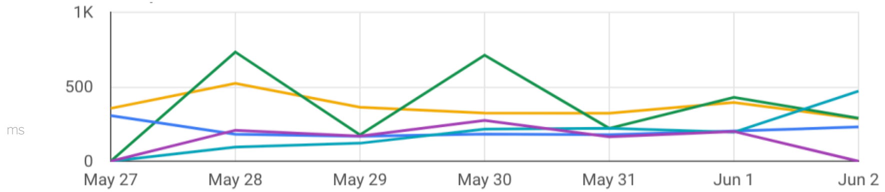
API OPERATIONS, MEDIAN PERFORMANCE


Get Potential Matches, milliseconds


Get Relationships Facets, milliseconds


Get Entity Details, milliseconds


Type-Ahead Search, ms

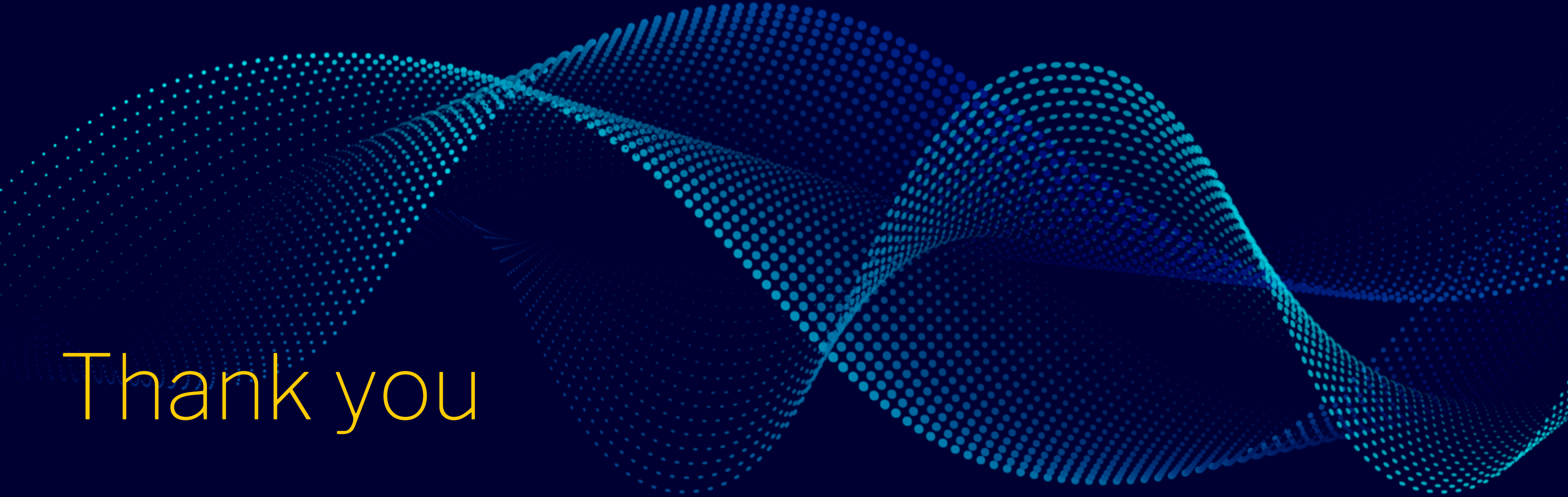

Search Results, milliseconds

Tenant Sizes:

🟦 - 800 M profiles
🟧 - 3.4 M profiles
🟩 - 11 M profiles
🟪 - 0.3 M profiles
▬ - 9 M profiles

RELTIO

Data Volume is not a factor for individual search APIs, Data Model is a factor!



Thank you

Performance & Latency , Limits