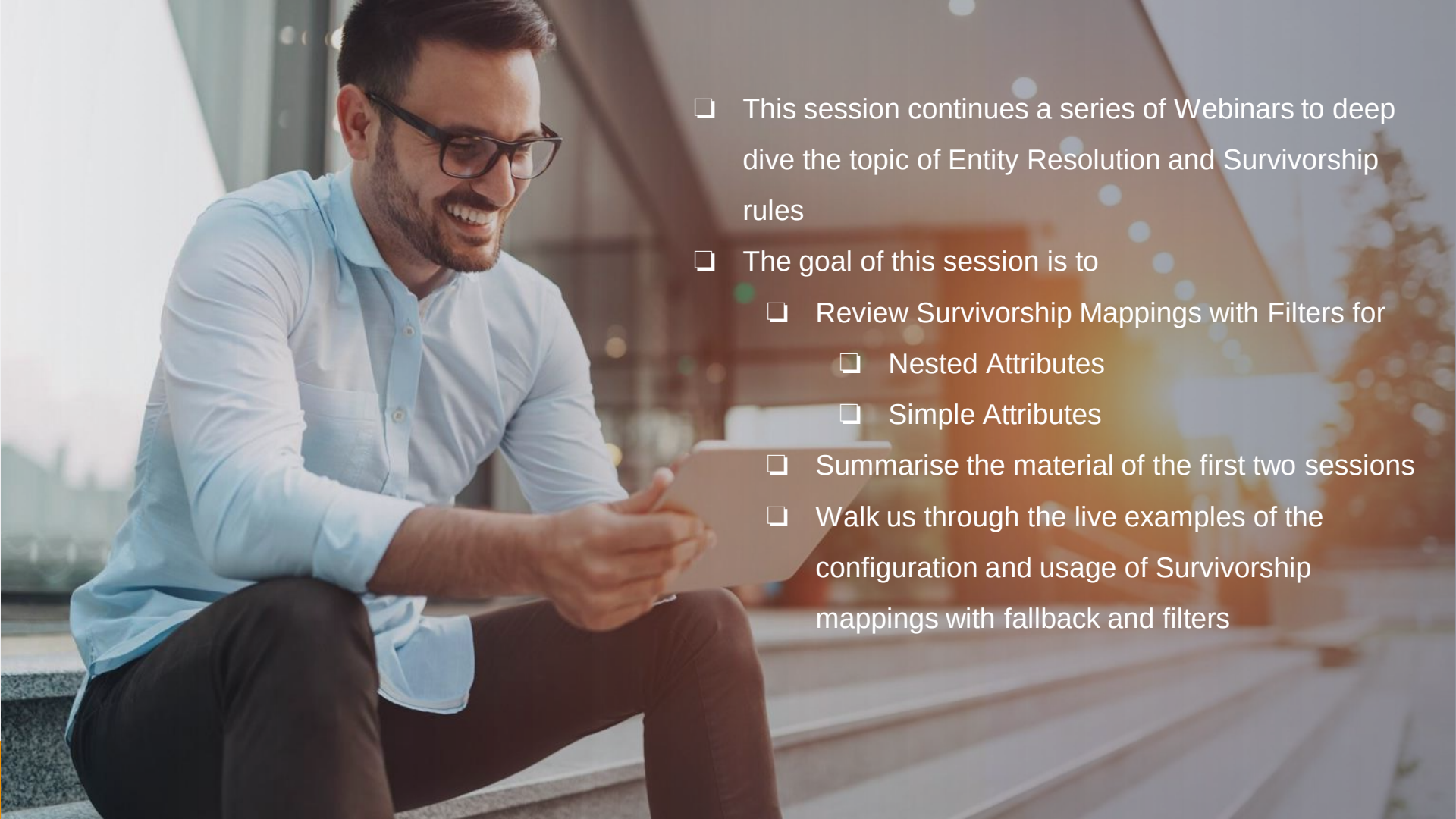
Deep Dive into Advanced Survivorship Strategies
Fallbacks and Filters

February 2022

Tatyana Selezneva - QA Engineer
Yakov Goyhman - Engineering Manager
Dmitry Blinov - Sr. Product Manager

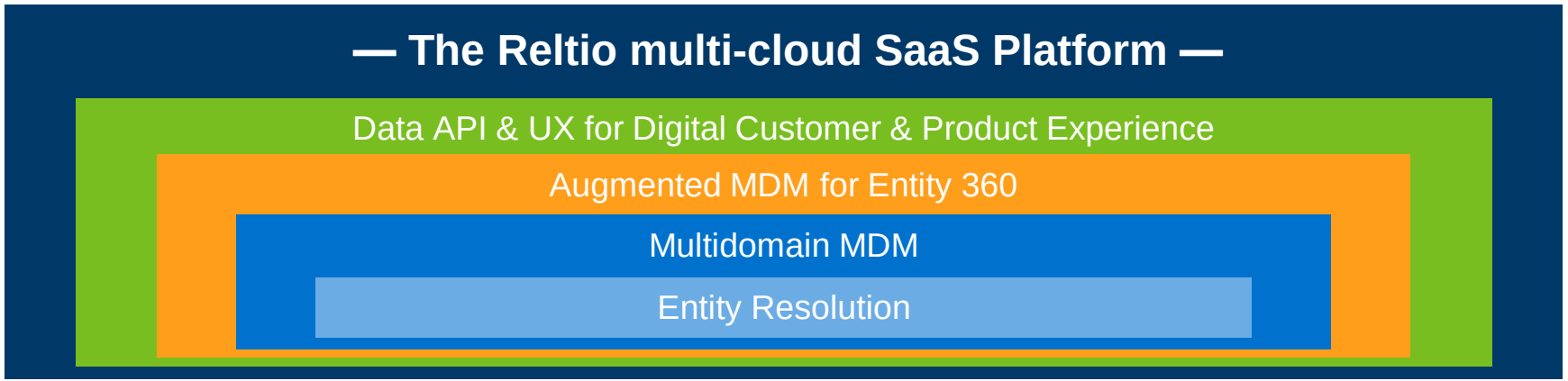


- This session continues a series of Webinars to deep dive the topic of Entity Resolution and Survivorship rules
- The goal of this session is to
  - Review Survivorship Mappings with Filters for
    - Nested Attributes
    - Simple Attributes
  - Summarise the material of the first two sessions
  - Walk us through the live examples of the configuration and usage of Survivorship mappings with fallback and filters

Entity Resolution is in the core of the Multidomain MDM

CUSTOMER BENEFITS

- Single source of truth at scale
- Real-time, analytics, data science and UX for all workloads
- Secure & governed access via UI, APIs, streaming and connectors
- SaaS with portability across multiple clouds
- Zero downtime maintenance and disruption-free upgrades



Entity Resolution

Entity Resolution happens on the attribute level where each attribute value loses or wins (survive). Survived value is called operational value OV, it is calculated by OV calculator based on survivorship rules set for each attribute type.

Crosswalk(data source) A	
Attribute	**Value**
First Name	Michael
Last Name	Frasca

Crosswalk(data source) B	
Attribute	**Value**
First name	Mike
Last Name	F.

Resolved Object		
Attribute	**Operational Value**	**Non Operational Value**
First Name	Mike	Michael
Last Name	Frasca	F.

Reltio

Re-fresh on survivorship strategies

- **Data Source (Crosswalk) based strategies**
  - **LUD** - make all values with the most recent update date to be winners for this attribute
  - **SRC_SYS** - make values, belonging to a source with the highest priority to be winners
  - **Oldest Value** - make values that came first to be winners
  - **Other Attribute Winner Crosswalk** - make attributes from the source that wins in the other attribute to be winners for this attribute
  - **WinnerEntityCrosswalk** - make all values of the attribute from the winner entity to be winners
- **Data (Attribute Value) based strategies**
  - **Frequency -** make values that came from the most number of different sources to be winners
  - **Aggregation** - make all values to be winners
  - **Min Value** - make min values to be winners
  - **MaxValue** - make max values to be winners

Reltio

Filters in survivorship mappings

Filter in survivorship mapping splits all values from an attribute by filter to different sets of values.
Those sets of values can intersect with each other if a value meets the condition for different filters.
For each set, winners are calculated separately.
Each filtering condition specified in separate attribute mapping.
If some value contained in many subsets and the value becomes a winner in just one of them - calculation stops and this value becomes an OV.

If a value doesn't match any filter from mappings, it goes to the set which will be calculated by default strategy and also separately from other sets.

This strategy should be specified separate mapping without filtering property otherwise LUD will be applied

```
{
     "attribute": "configuration/entityTypes/HCP/attributes/FirstName",
     "filter": {
       "equals": [
         {
           "uri": "configuration/entityTypes/HCP/attributes/Indicator",
           "value": "true"
         }
       ]
     },
     "fallbackStrategies": [
       {
         "attribute":
"configuration/entityTypes/HCP/attributes/FirstName",
. . .
"survivorshipStrategy": "SRC_SYS"
```

Filter use cases

Common rules

- All attribute operation participate in the filtering - OV and non-OV
- Only “equals” condition is supported
- Only attribute value can be used as a parameter for the equals condition

Nested attributes

- **comparisonAttributeUri** should be defined
- this is an entity-internal level use case
- this use case provides an ability to apply more complex logic to merge the nested attributes and define a final set of sub-attributes operational values

Simple attributes

- An entity-external use case
- Use an attribute with a pre-defined or controlled value of attribute A to control the survivorship strategy applied to attribute B

Reltio

Advanced behavior

1. **LUD**
    a. if some values have the same most recent crosswalk, but one (or many) of them has **more** recent singleAttributeUpdateDate - this value only becomes winner, but winner crosswalk will still be same.
    b. if some values have many crosswalks with the same most recent date - all those crosswalks become winners and all values with those crosswalks become winners too.
2. **OldestValue**
    a. if some values have many crosswalks with the same oldest create date - all those crosswalks become winners and all values with those crosswalks become winners too.
3. **SRC_SYS**
    a. Notes regarding this strategy:
        i. If the strategy is applied to one value only, this value and its crosswalk become winner, irrespective of whether this crosswalk has source from priority list or not.
        ii. If there are no values with crosswalks from priority list, the winner value will be the value which has most recent Cassandra update time. All crosswalks of this value become winner crosswalks.
    b. The above notes (i and ii) are not relevant when advanced behavior is switched on: there are not any "internal" fallbacks triggering.
    c. So, only below are applicable to a new survivorship:
        i. If an attribute doesn't have values from sources of the priority list, priority of sources are taken from the source types configuration of the business model.
        ii. If business model does not have priority of sources wither - nothing is survived (all values have ov=false)

Reach out to Reltio Support to enable the advanced behavior in your tenant if you do not have it yet.

Reltio

Demo