Welcome

empower!

[YOUR TRAINER THIS SESSION]

Data Exploration

Robin Debel

robin.debel@skyline.be

Mastering Flows and Variables in LCA

[WHAT YOU WILL SEE THIS SESSION]

Agenda

Advanced track

1. Variables
    1. What are they?
    2. Why do we them?
    3. How do we use them?
    4. Change their content
2. Flows
    1. What are they?
    2. Why do we them?
    3. How do we use them?

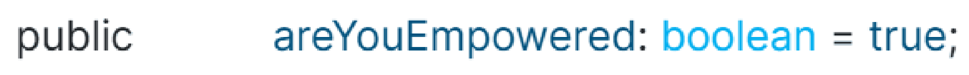
Variables

Ever-Changing, Always Yours!

What are Variables?

- **Typed**, **independent** and **dynamically updateable** data objects.

- Like variables in a programming language:



```
public        areYouEmpowered: boolean = true;
```

- Stored **inside** the app or dashboard
  - **Volatile**
  - **Instantly** accessible
  - **No security**
  - **Scoped** availability

10.4.12

10.3.0 CU21 | 10.4.0 CU9

Why do we need Variables?

- **Improve Performance**

  Reduce server calls by storing static data in variables — GQI isn't always the best solution.

- **Better UX**

  Define your own state management to create custom features without relying on visual elements.

- **Reusability & Portability**

  Use variables across multiple locations. When migrating to another system, update the value in one place.

How do we use Variables?

1. Create a new Variable. (Data → Variables → +)

2. It all starts with a **Name.**

3. Provide context through its **Type.**
   - **General**: Text, number, boolean***** and **table**
   - **DataMiner objects**: Element, Service, View and Object manager instance

4. Define its usage by marking it **Read-only** or updateable.

5. Optionally, assign a **Default value**. (required when read-only)

6. Use your Variable in **components, queries**, **actions**, **flows**, ...

* 10.5.3

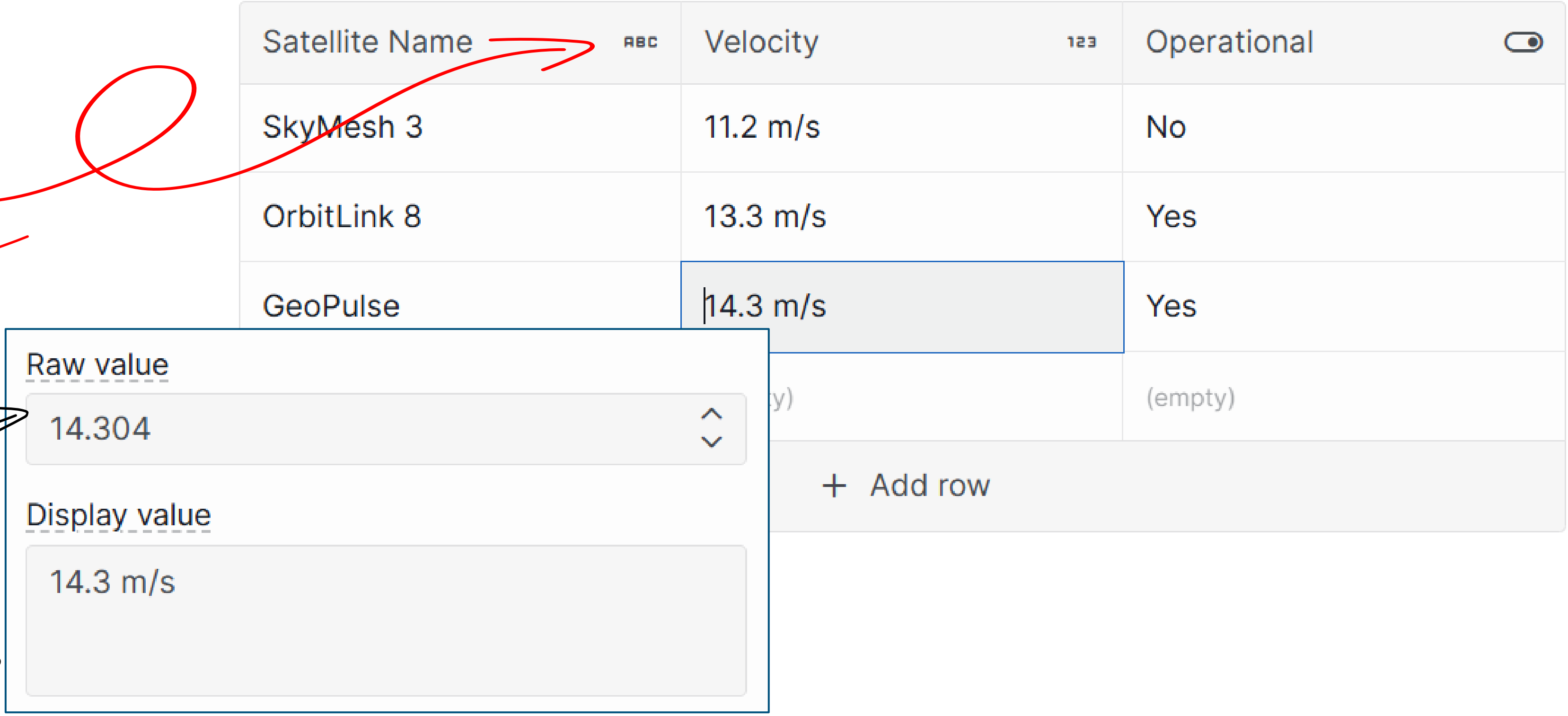
10.4.0 CU12 | 10.5.0 CU0

Table variables

The fan favorite

- Limited to **100 rows** and **20 columns**
  - **Do**: static filters
  - **Don't**: replace databases

- Columns are also typed
  - Text, number, boolean
  - Display and raw value



Satellite Name ABC	Velocity 123	Operational
SkyMesh 3	11.2 m/s	No
OrbitLink 8	13.3 m/s	Yes
GeoPulse	14.3 m/s	Yes
	y)	(empty)

Raw value

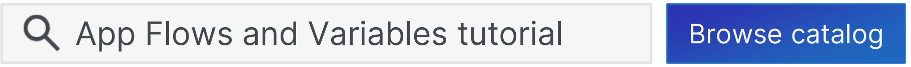
14.304

Display value

14.3 m/s

EXERCISE

Time to get our hands dirty



App Flows and Variables tutorial

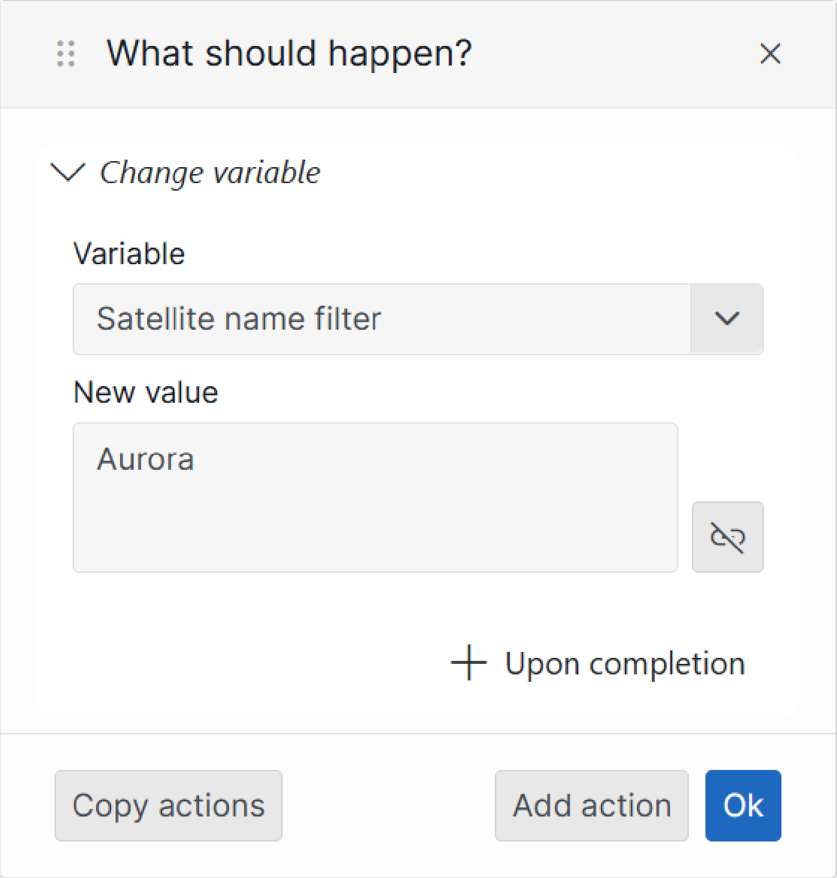
Change its content

We reached the "variable" part

- A new action **Change variable**
  - Set value
  - Add row (table only)
  - Clear table (table only)

- Read-only variables cannot change at runtime.

- Variable changes are **immediately** applied.



What should happen?

Change variable

Variable: Satellite name filter

New value: Aurora

10.5.1

10.4.0 CU10

EXERCISE

Time to get our hands even more dirty

[ASK AWAY]

Questions?

Flows

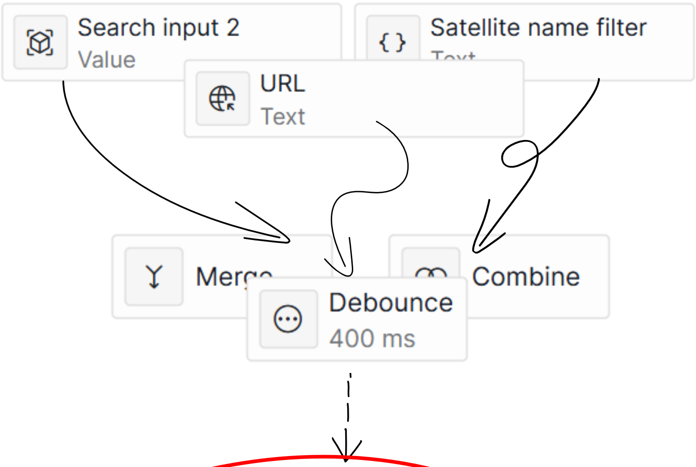
Why Work Harder When You Can Flow Smarter?

What are Flows?

GQI in the browser???

“A **Flow** is a structured **stream of data** originating from **one or more sources**, undergoing one or multiple processing steps, before ultimately resulting in a **single typed value**.”

— ChatGPT



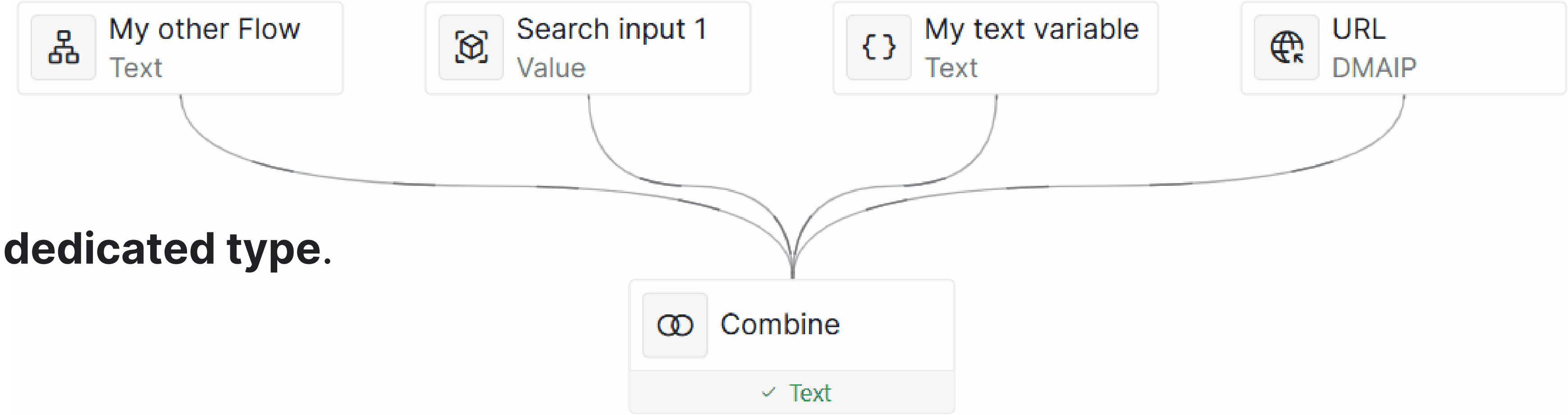
one single value

10.4.12

10.3.0 CU21 | 10.4.0 CU9

Data Sources

- Anything that holds data within the app
  - Other **Flows**
  - **Component** data
  - **Variables**
  - **URL**



- Each data source has **one dedicated type**.

- You **cannot mix different types*** within a single flow.

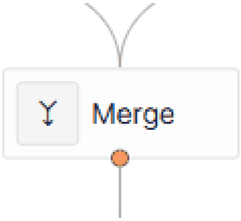
Operators

Only three, for now ;)

- **Merge**

  Combines multiple independent sources into a single stream.

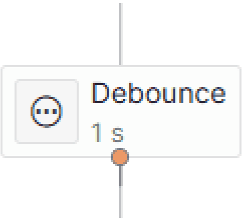
  Just like merging lanes on a highway.



- **Debounce**

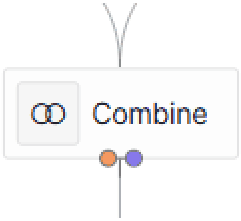
  Waits for a pause in rapid events, then processes only the latest one.

  A classic real-world application is in form inputs, particularly in search bars.



- **Combine**

  Pairs values from multiple sources, ensuring they work together for calculations or decisions.



Why do we need Flows?

- **Reusability**

  Duplicating **panels**, each linked to a component → One **reusable panel** linked to a **Flow**

  Duplicating **queries**, each linked to a component → One **reusable query** linked to a **Flow**

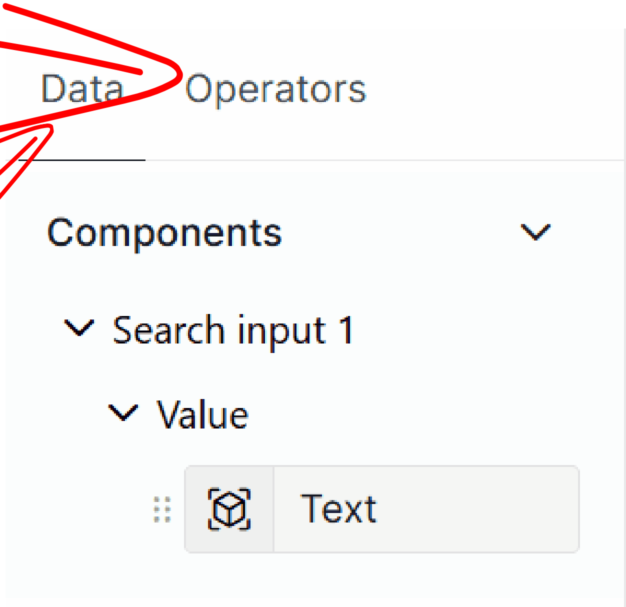
  You get the drill…

- **Optimize server load**

  Fetch the query on filter change → Only fetch when the filter hasn't changed for e.g., 250ms

How do we use Flows?

1. Open the **Flow editor.** (Data → Flows → +)
2. Again, a meaningful **Name** is key here.
3. Drag & drop at least one **Source**.
4. Chain an **Operator** to every source.
5. Connect all the dots and hit **Create**.
6. Use your Flow in **components, queries**, **actions**, **flows**, …



FINAL EXERCISES

So let’s make it count, by doing two at once!

[ASK AWAY]

Questions?

empower!