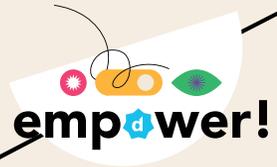


# Welc@me



*ll* Building

a LCA → with DOM



[ YOUR TRAINER THIS SESSION ]

Principal DevOps  
Engineer



Thomas  
Ghysbrecht

thomas.ghysbrecht@skyline.be

# Agenda



Fundamentals track

1. Status system introduction
2. Deploying the model
  1. Quick look at the code
3. Hands-on: Build the LCA from scratch



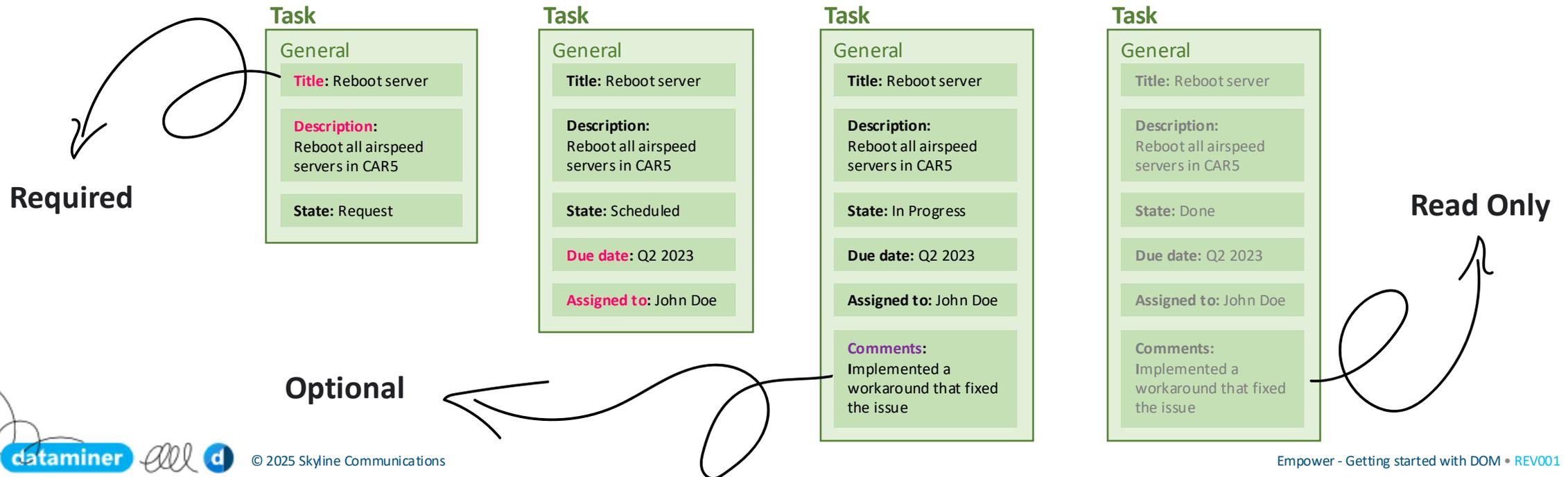
# Status system introduction

Make you DOM model state aware

# Status system

## Statuses

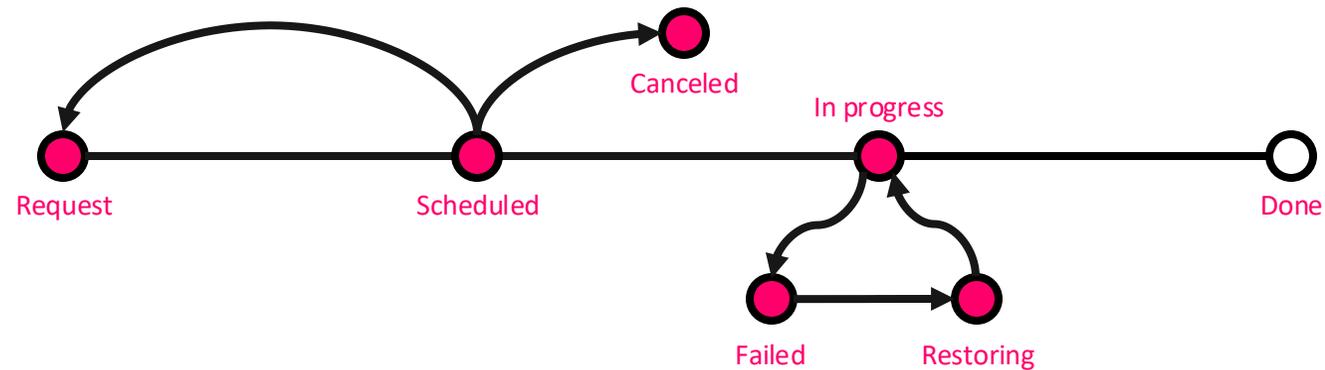
- Allows you to assign a **state** to a DOM instance, **transition** between these states and put **conditions** on whether fields should be **required**, **read-only**, **hidden** etc. for each state.



# Status system

## Transitions

- Allows you to assign a **state** to a DOM instance, **transition** between these states and put **conditions** on whether fields should be **required, read-only, hidden etc.** for each state.



# Status system

## Buttons and actions

- **Buttons** can be configured on the Behavior Definition, so they show up on the DOM form.
- Every button is linked to one **DOM action**, currently always a **DataMiner automation script**.
- Often used to facilitate the transitions.

VIH\_2-AMQ-112\_8

License Plate ⓘ  
**2-AMQ-112**

Location ⓘ  
**Belgium**

Capabilities ⓘ  
**HD**

Size ⓘ  
**Medium**

Number Of Cameras ⓘ  
**8**

To In Use To Maintenance

```
public class Script
{
    [AutomationEntryPoint(AutomationEntryPointType.Types.OnDomAction)]
    0 references | 0 changes | 0 authors, 0 changes
    public void OnDomActionMethod(IEngine engine, ExecuteScriptDomActionContext context)
    {
        var domInstanceId = context.ContextId as DomInstanceId;
        if (domInstanceId == null)
        {
            engine.ExitFail("The action was not triggered in the context of a DOM instance.");
            return;
        }

        var helper = new DomHelper(engine.SendSLNetMessages, domInstanceId.ModuleId);
        helper.DomInstances.DoStatusTransition(domInstanceId, "a_to_b");
    }
}
```

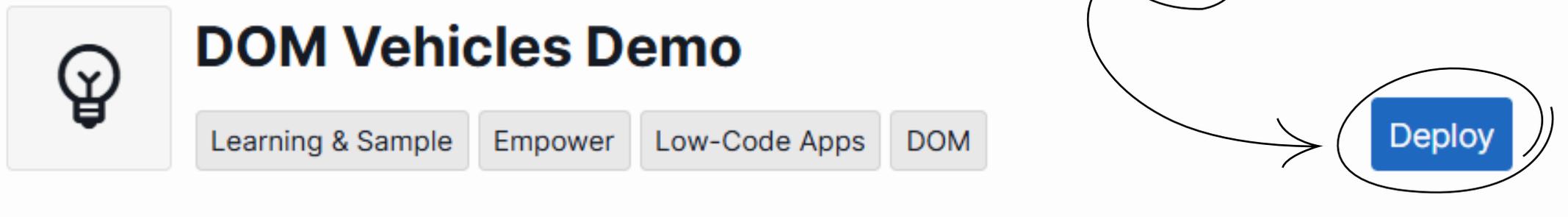
# Deploying the model

## Make you DOM model state aware

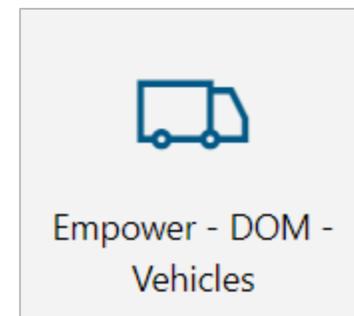
# DOM Module Setup

Deploy the catalog package

1. Go to **catalog.dataminer.services**
2. Search for “**DOM Vehicles Demo**”
3. Deploy this package to your DaaS



4. Check if the example Low Code App is deployed



# DOM Module Setup

## Reference app

EMPOWER - DOM - VEHICLES

ADD VEHICLE REFRESH

Acti...

**License Plate**

Filter

**Location**

no lo... and no up...

- Australia
- USA
- UK
- Belgium
- Finland

LICENSE PLATE	LOCATION	STATE	SIZE	NUMBER OF CAMERAS
YN-82EV	Australia	In Use	Large	30
IK22-BQ	Australia	Available	Extra Large	10
BJD-6333	USA	Maintenance	Extra Large	32
KMD-6558	USA	Available	Small	4
HD03-MGX	UK	In Use	Medium	16
2-TMA-654	Belgium	In Use	Extra Large	40
2-AMQ-112	Belgium	Available	Medium	8
CGI-964	Finland	Maintenance	Medium	6

# DOM Module Setup

## Deploy package code

40 references | 0 changes | 0 authors, 0 changes

```
public class VehicleDomModelConstants
```

```
{
```

3 references | 0 changes | 0 authors, 0 changes

```
public static string ModuleId { get; } = "empower_vehicle_demo";
```

2 references | 0 changes | 0 authors, 0 changes

```
public static DomDefinitionId DomDefinitionId { get; } = new DomDefinitionId(Guid.Parse("eb7c180f-bb6b-4816-a450-dbced2eed144"));
```

11 references | 0 changes | 0 authors, 0 changes

```
public static SectionDefinitionID SectionDefinitionId { get; } = new SectionDefinitionID(Guid.Parse("a451c946-2bf4-46c0-b6f0-9bef5f064bd5"));
```

```
/// <summary>
```

```
/// Describes the license plate of the vehicle.
```

```
/// Type: String value
```

```
/// </summary>
```

7 references | 0 changes | 0 authors, 0 changes

```
public static FieldDescriptorID LicensePlateFieldDescriptorId { get; } = new FieldDescriptorID(Guid.Parse("23e7a506-d63a-4851-af2d-7e42d326f322"));
```

```
/// <summary>
```

```
/// Describes the location of the vehicle.
```

```
/// Type: Single value int enum
```

```
/// </summary>
```

6 references | 0 changes | 0 authors, 0 changes

```
public static FieldDescriptorID LocationDescriptorId { get; } = new FieldDescriptorID(Guid.Parse("f4e34c3a-3ce2-4caf-9fa7-d4e3af8fae70"));
```

2 references | 0 changes | 0 authors, 0 changes

```
public static string[] Locations { get; } = { "Australia", "USA", "UK", "Belgium", "Finland" };
```



# Hands-on

## Build the LCA from scratch

# Hands-on

## Main page basics

- Create a **new LCA** (name, icon,...)
- Add a **table component**
- Add a **query filter component**
- Create a new **GQI query** to retrieve our **vehicle DOM instances**
- **Feed** the GQI query to the query filter and the query filter output to the table
  
- Add a 'ADD VEHICLE' **button** to the header (leave empty for now)
- Add a 'REFRESH' **button** to the header
  - Let it **trigger** the 'Fetch the data' **component action** on the table

# Hands-on

## 'Create DOM instance' panel

- Create a **new panel** 'Add vehicle'
- Add the **form component** to the panel
  - Feed the 'Vehicle' DOM definition
- Add a 'SAVE' header bar **button** and link actions:
  - Execute component action > Form, Save current changes
  - Close a panel > This panel
  - Execute component action > Table, Fetch the data
- Add a 'DISCARD' header bar **button** and link action:
  - Close a panel > This panel
- On the main page, add the 'Open panel' **action** to the 'ADD VEHICLE' button

# Hands-on

## ‘View/edit/delete DOM instance’ panel

- Create a **new panel** ‘Edit vehicle’
- Add the **form component** to the panel
  - Feed the ‘Selected rows > Object manager instances’ from the table component
- Add a ‘SAVE’ header bar **button** and link actions:
  - Execute component action > Form, Save current changes
  - Close a panel > This panel
  - Execute component action > Table, Fetch the data
- Add a ‘DELETE’ header bar **button** and link action:
  - Execute component action > Form, Delete instance
  - Close a panel > This panel
  - Execute component action > Table, Fetch the data
- Add an empty header bar **button** with ‘X’ icon and ‘Close a panel’ action

# Hands-on

## 'View/edit DOM instance' panel

- Edit the table settings > actions
  - Configure the 'On double click' **action** with the 'Open a panel' option

# Hands-on

## Spice things up

- Add **conditional coloring** on the 'State' column
- Add **conditional coloring** on the 'Number Of Cameras' column
- Add **stepper component** to the view/edit panel
- Create a **table template** with an **icon** to edit the vehicle
- Extend the **table template** to display the **license plate** like a real one
- Add **counters** that display the amount of vehicles

State

- New
- Available
- Maintenance
- In Use

Number Of Cameras

- > 0 and ≤ 10
- ≥ 11 and ≤ 20
- ≥ 21 and ≤ 30
- ≥ 31 and ≤ 40
- ≥ 40 and **no upper bound**

New Available In Use

2-TMA-654



2-TMA-654

2-AMQ-112

Vehicles in use

3

Vehicles available

3

[ Your opinion matters ]

# Feedback



Don't forget to grab your present on the way out!



[ ASK AWAY ]



# Questions?