# TB141Ic – ICT System Engineering and Rapid Prototyping

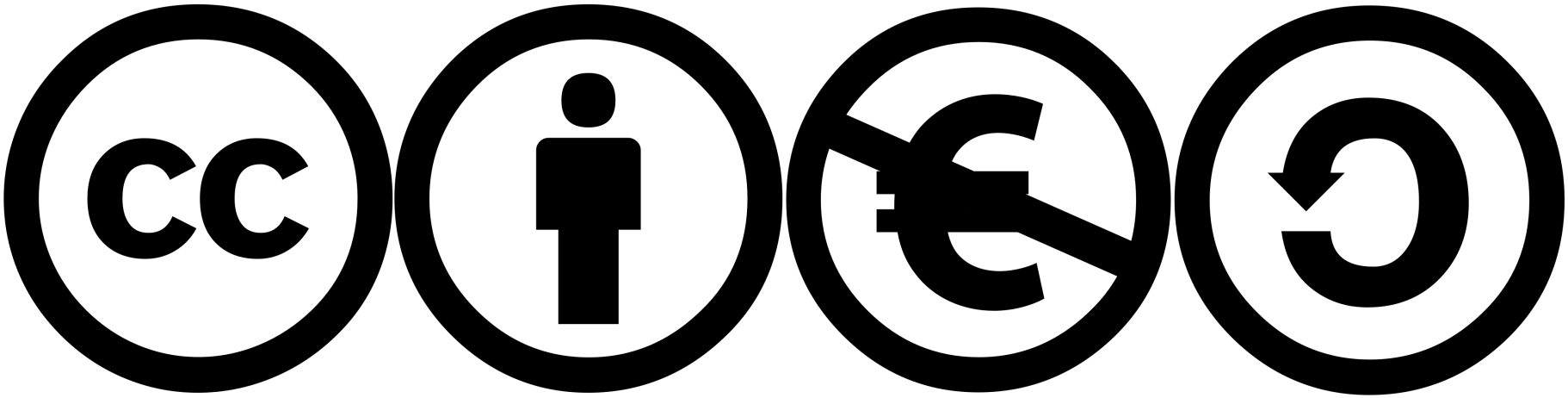
## MVC in Mendix: Controller - Microflows and Event Handling

Jacopo De Stefani, Ph.D

Delft University of Technology, The Netherlands

29/03/2023

# Licensing information



# Learning objectives and related literature

**Learning objectives**

- Create a basic Mendix app
- Identify the components of a Mendix microflow
- Implement event handlers using Microflows
- Implement automatic data computation using Microflows

**Related literature**

- Mendix - Build an App in Mendix Studio Learning Path
- Mendix - Become a Rapid Developer Learning Path
- Mendix - Crash Course

# Mendix

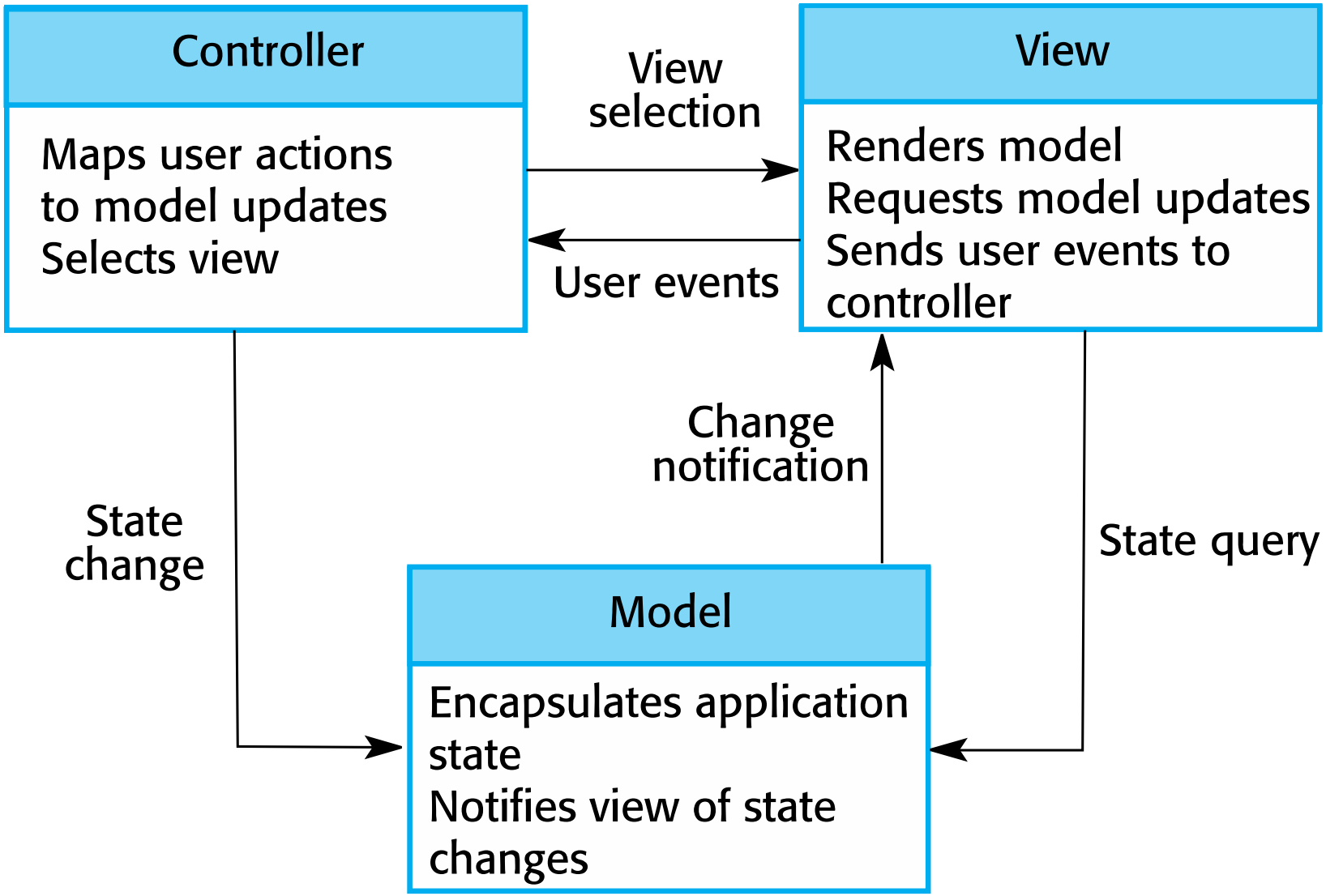
# Mendix


- Low-code development platform
- UML-like Diagrams → Code/Data structures
- Visual Sketches → User Interface

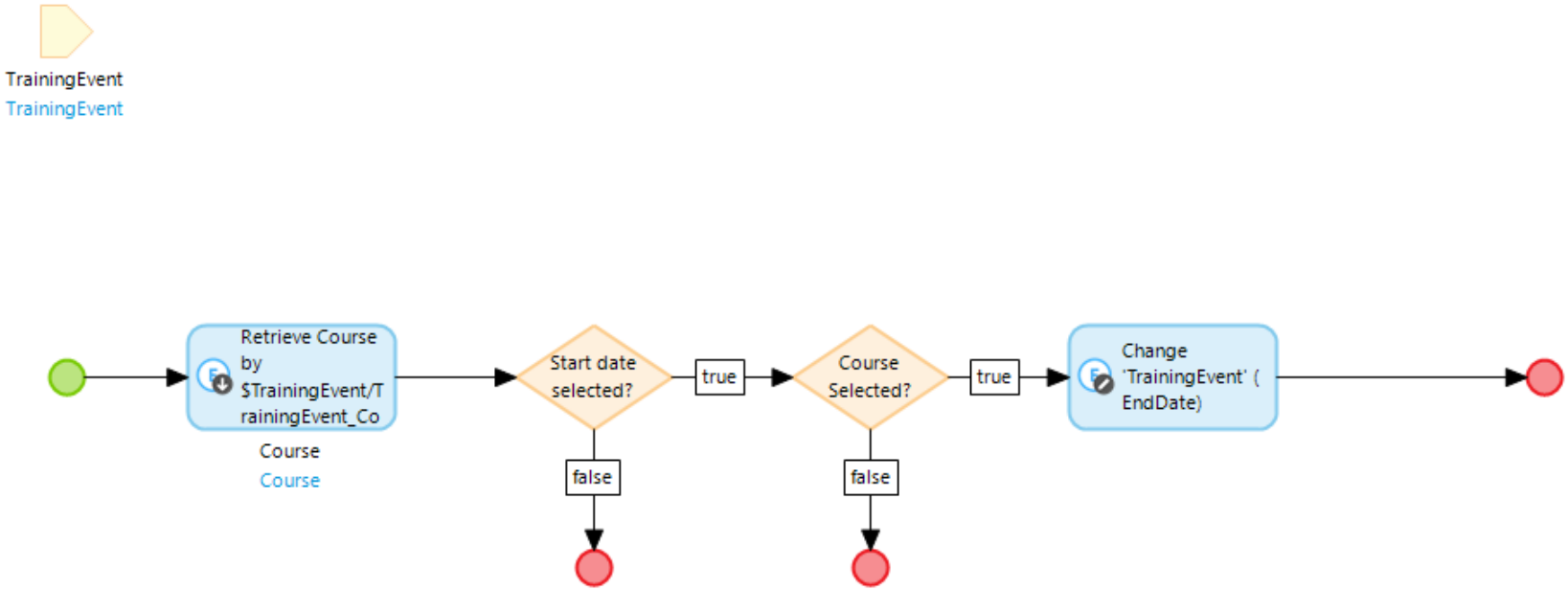
# Mendix


- **Software Development Methodology:** Agile (with SCRUM project management)
- **SW Architecture:** MVC
- **HW Architecture:** Client-Server
- **Programming Language:** Graphical
  - Domain model ≡ Data structures → UML Class-like diagrams
  - Microflows ≡ Code → UML Activity-like diagrams
  - Pages ≡ User Interface → Graphical sketches

# Mendix - MVC

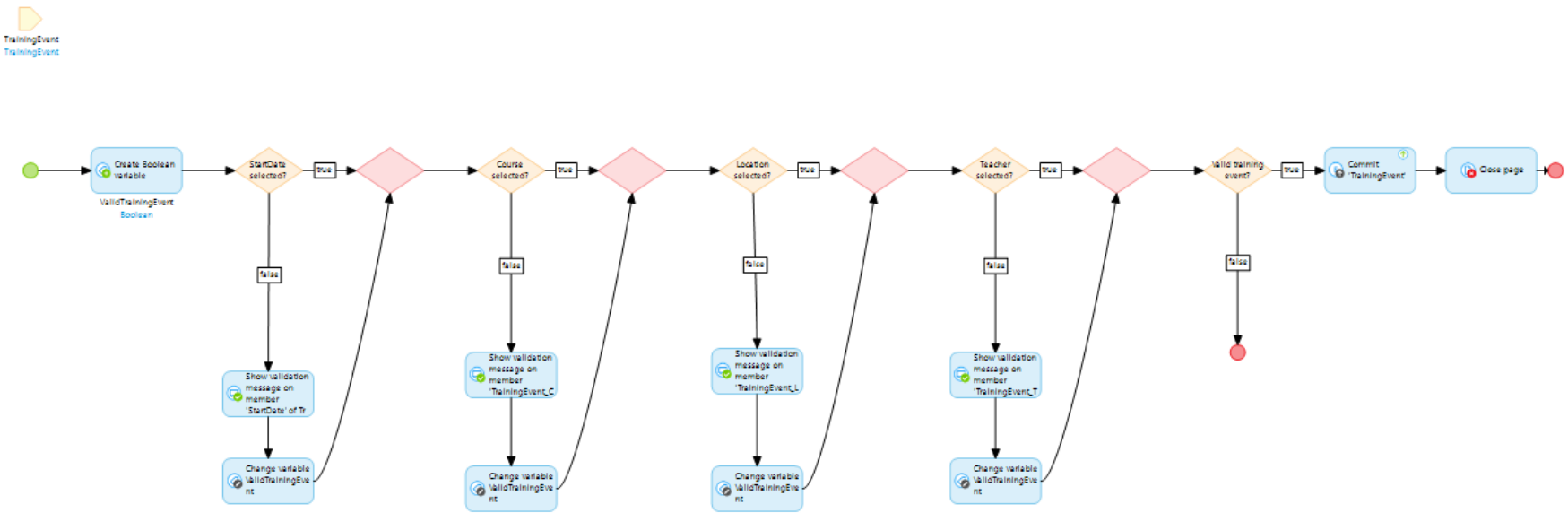

Controller
Maps user actions to model updates
Selects view
View selection
User events
View
Renders model
Requests model updates
Sends user events to controller
State change
Change notification
State query
Model
Encapsulates application state
Notifies view of state changes

# Mendix - Simple Microflow




Source: Mendix Academy - Rapid Developer Learning Path

# Mendix - More complex microflow




Source: Mendix Academy - Rapid Developer Learning Path

# Mendix - Microflows

- Graphical representation of the logic of the applications
  - Close resemblance to UML Activity Diagrams
  - Easily understandable by technical and non-technical people
- Serve different purposes in the application
  - Validate data
  - Execute conditional operations based on input data
  - Create new entities
  - Include more advanced built-in actions (integrations with external platforms)

# Mendix - Microflows - Elements

## Events

Events represent the start and end points of a microflow.

| Graphic | Name | Description |
|---|---|---|
|  | Start Event | A start event is the starting point of the microflow. A microflow can only have one start event. |
|  | End Event | An end event represents the end of the microflow. Depending on the return type of the microflow, in some cases a value must be specified. There can be more than one end event. This depends on the number possible outcomes of your microflow. |

## Flows

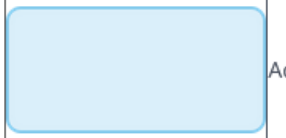
Flows form the connection between elements.

| Graphic | Name | Description |
|---|---|---|
|  | Sequence Flow | A sequence flow is an arrow on which all elements that form the logic of the microflow will be placed. The elements will be executed in sequence (in the direction of the arrow), so the order in which you place the elements is important! |
|  | Annotation flow | An annotation flow is a connection that can be used to connect an annotation to another element, to visually show that this annotation says something about this element. |

# Mendix - Microflows - Elements

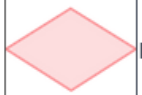
## Activities

Activities are the actions that are executed in a microflow.

| Graphic | Name | Description |
|---|---|---|
|  | Activity | An activity does something. There are all kinds of activities. Create or delete an object, open a page, show a message, etc, etc. |

## Decisions

Decisions deal with making choices and merging different paths again.

| Graphic | Name | Description |
|---|---|---|
|  | Decision | Decision defines a choice based on a condition (a check). This decision will result in several outgoing flows, one for every possible outcome. The microflow will follow only one of the outgoing flows, based on the outcome of the check. Decisions are defined using microflow expressions. These will be explained in more detail later on. |
|  | Merge | A merge can be used to combine multiple sequences flows back into one. If a choice is made in a microflow and afterward some common work needs to be done, you can combine the two (or more) paths using a merge. |

Source: Mendix Academy - Rapid Developer Learning Path

# Mendix - Microflows - Elements

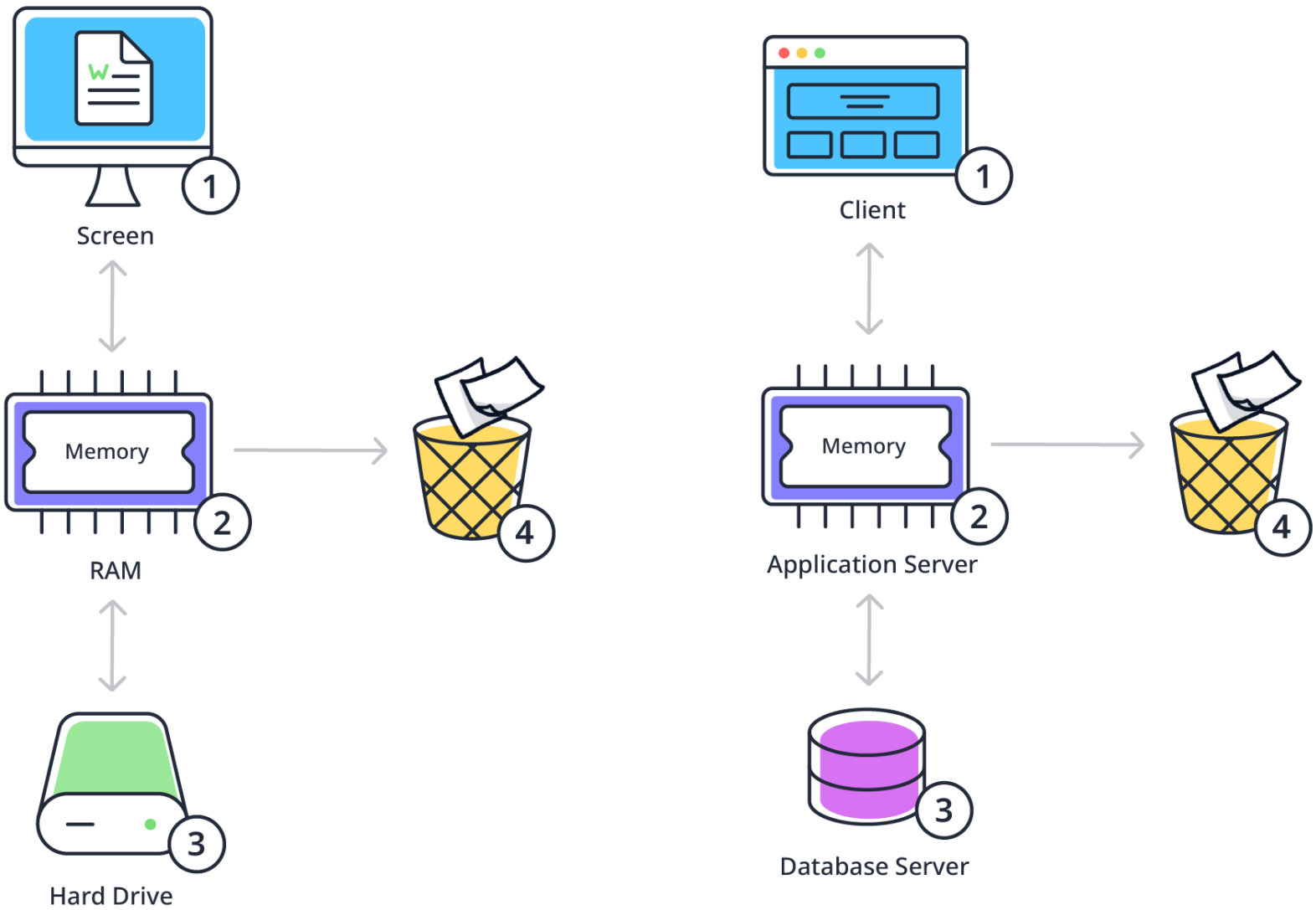
## Artifacts

Artifacts provide the microflow with input and allow comments to be made.

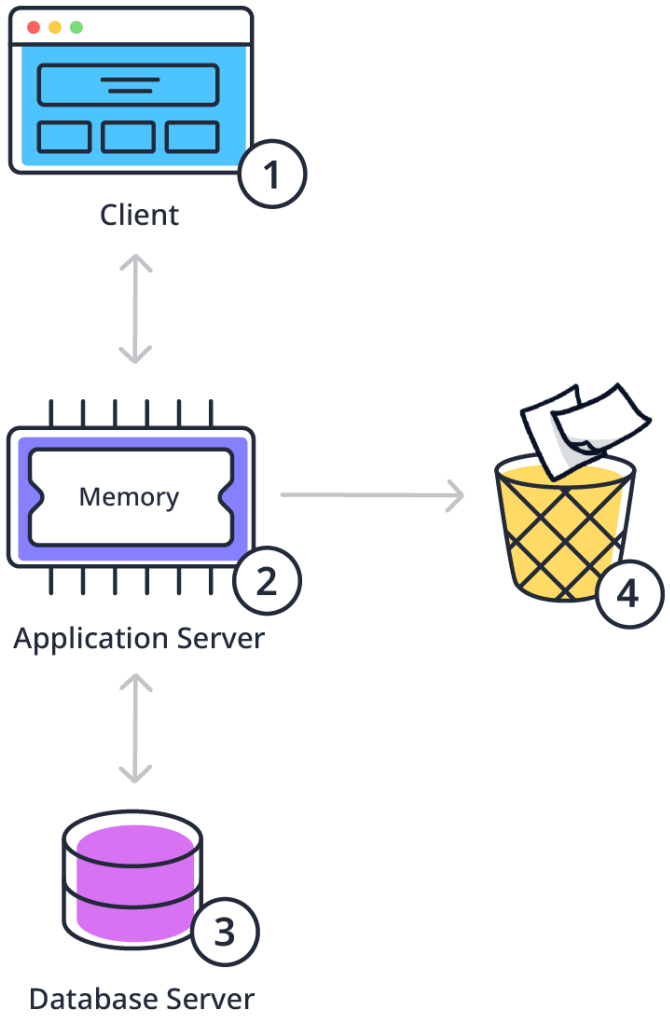
| Graphic | Name | Description |
|---|---|---|
|   | Parameter | A parameter is data that serves as input for the microflow. Parameters are filled at the location from where the microflow is triggered.<br>For example: You click on a specific course in the course overview and then click on a button that triggers a microflow. Then the course you selected in the overview page will be the input parameter for that microflow. |
|  | Annotation | An annotation is an element that can be used to put comments in a microflow. These annotations don't really do anything, apart from serve as a reference for developers. |

Source: Mendix Academy - Rapid Developer Learning Path

# Mendix - Retrieve data



# Mendix - Retrieve data




- **By Association Retrieve**
  - Looks for objects in memory (2) and the database (3)
  - Can retrieve transient (not yet committed) objects
- **From Database Retrieve**
  - Looks in the database only (3)
  - Cannot retrieve transient (not yet committed) objects, only gives back last stored version.

# Mendix - Events

- Specific events on **Pages** and the **Domain Model** can trigger **Microflows**.

Source: Mendix Academy - Rapid Developer Learning Path

# Mendix - Events

- Specific events on **Pages** and the **Domain Model** can trigger **Microflows**.
- **On the Pages:**
  - Associated to specific input widgets
  - **On enter:** Triggered while clicking on input widget
  - **On change:** Triggered while modifying the content of the input widget
  - **On leave:** Triggered while exiting the content of the input widget
  - Used to perform specific actions in response to user interaction.

Source: Mendix Academy - Rapid Developer Learning Path

# Mendix - Events

- Specific events on **Pages** and the **Domain Model** can trigger **Microflows**.
- **On the Domain Model**
  - Associated to a specific entity - Object Event Handler
  - Allow **a before/after** modifier
  - **On create:** Triggered before/after an object creation
  - **On commit:** Triggered before/after the commit of an object to the database
  - **On delete:** Triggered before/after the deletion of an object to the database
  - **On rollback:** Triggered before/after cancelling before saving.
  - Used to perform specific actions on data manipulations.

Source: Mendix Academy - Rapid Developer Learning Path

# Mendix - Pages Trigger - Example

# Mendix - Domain Model Trigger - Example

# Mendix - Development

Now it is your turn to develop...

# Mendix - Development - Exercise 13

In the app defined in **Exercise 13**, add two attributes: `SaleDuration` (in days) and `EndDate` to the `ProductStock` object.
Then, create a microflow to perform an automatic computation of the `EndDate` based on `StartDate` and `SaleDuration`, every time that either the `StartDate` or `SaleDuration` changes.

# Mendix - Development - Exercise 14

In the app defined in **Exercise 13**, add an attribute: `Subtotal` (in Euro) to the `ProductStock` object.
Then, create a microflow to perform an automatic computation of the `SubTotal` based on `Quantity` and `Price`.
A retrieve operation might be required to solve the exercise.

# Mendix - Development - Exercise 15

In the app defined in **Exercise 14**, add an attribute: `ExpectedSales` (in days) to the `Business` object.
Then, create a microflow to perform an automatic update of the `ExpectedSales` every time a new `ProductStock` object is created.