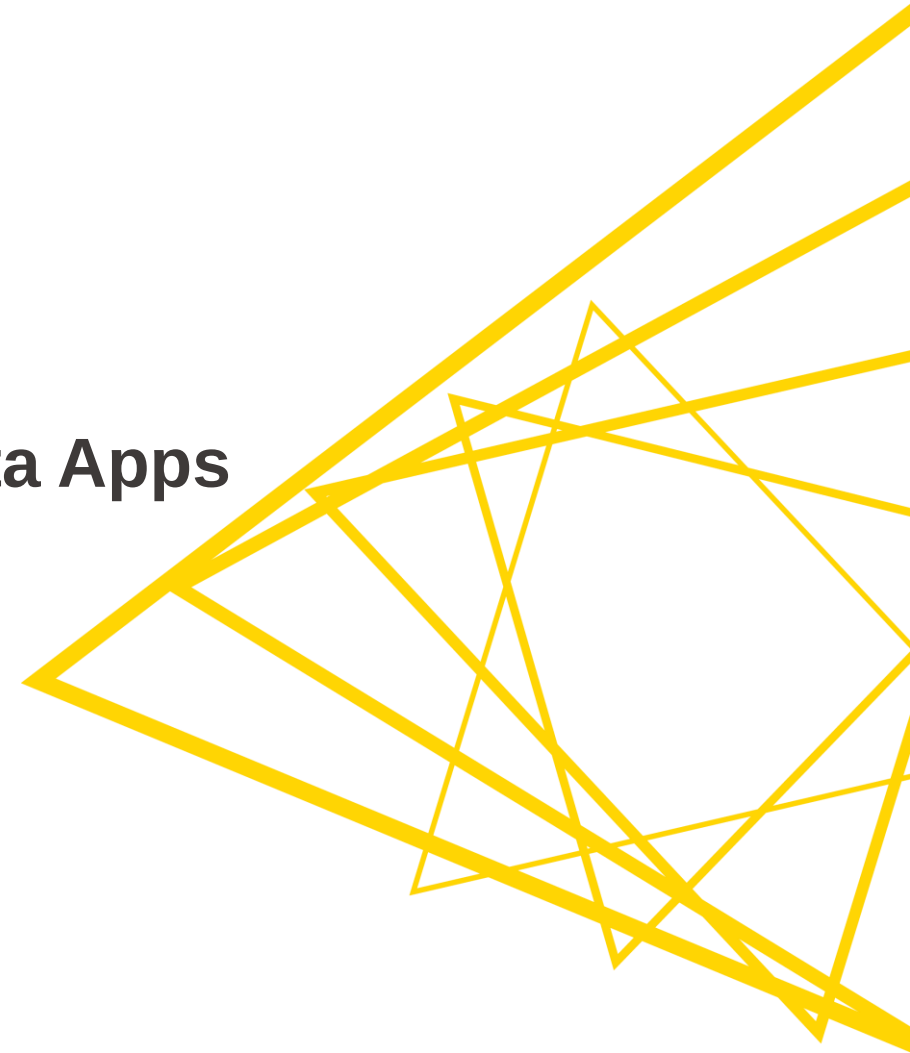
Open for Innovation
KNIME

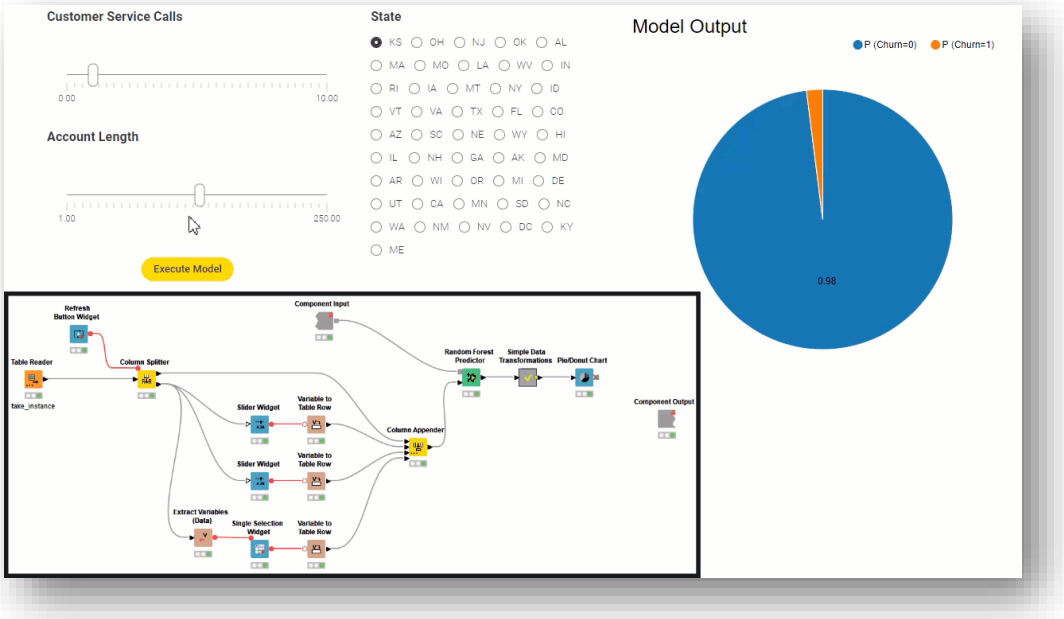
[L3-WP] Productionizing Data Apps

KNIME GmbH



Who is this Course for?

- Anyone who uses (or wants to use) KNIME to:
  - Deploy, share, and monitor data applications that should be made public
  - Build interactive interfaces for data science solutions without using a frontend language
  - Create analytical dashboards that help communicate and share solutions and findings



Structure of the Course

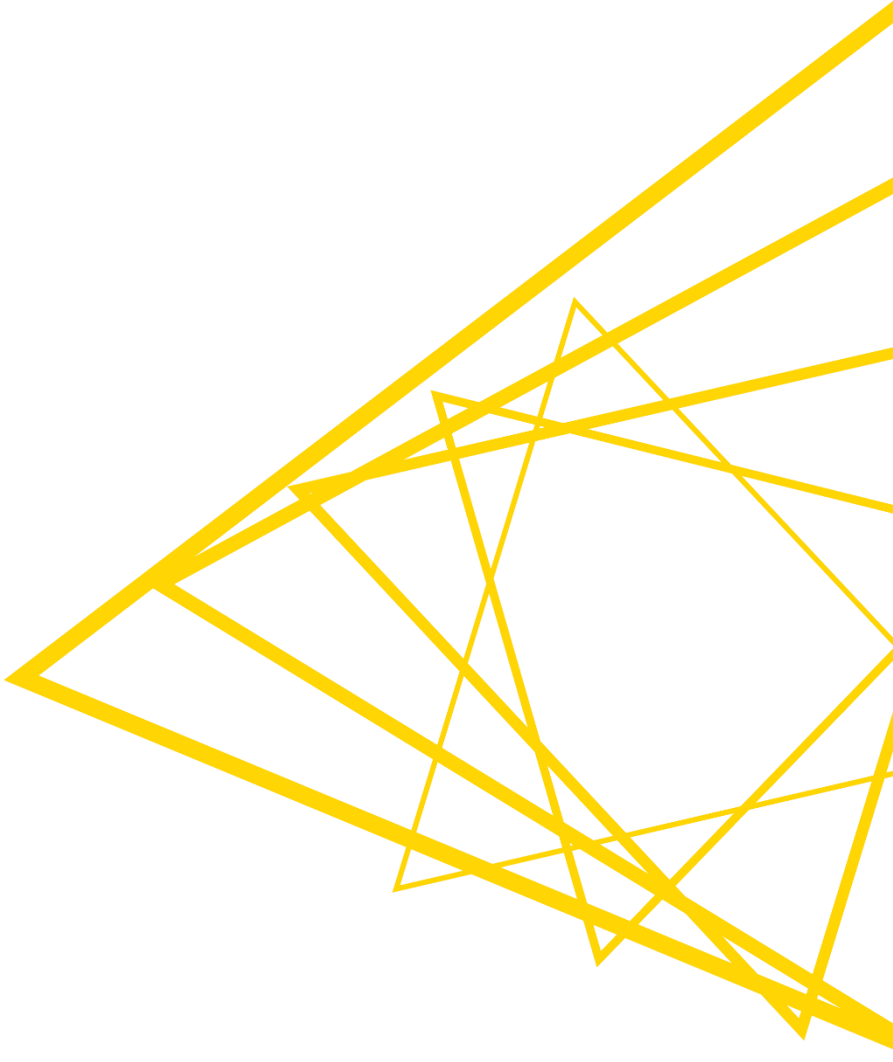
Session	Topic	Duration
Session 1	**What happens after the prototype is ready for deployment?**	75 min
Session 2	Introduction to KNIME Server	75 min
Session 3	Deploying WebPortal Data Apps	75 min
Session 4	Performance optimization, orchestration, error handling, and KNIME Edge	75 min
Session 5	Wrap-up Session	15 min

Structure of each session

- Discussion of past exercises
- Course
- Introduction of next exercises

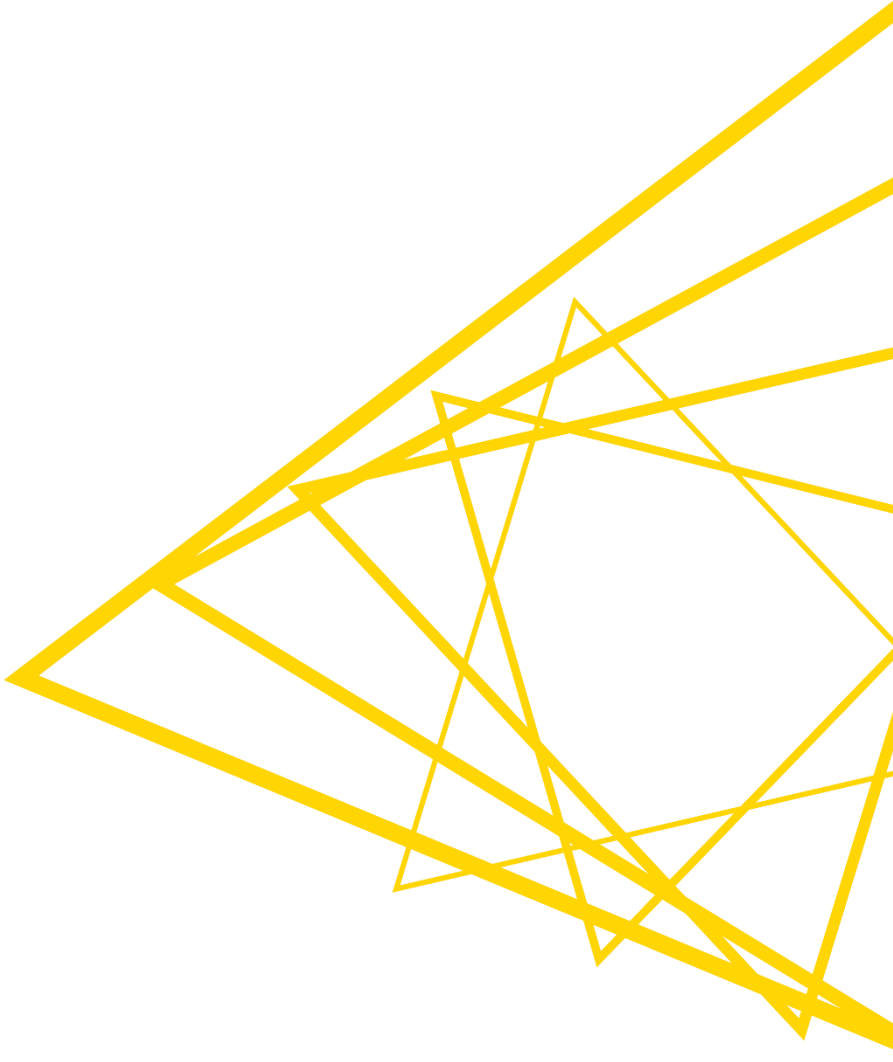
Session 1

What Happens after the Prototype Is Ready for Deployment?

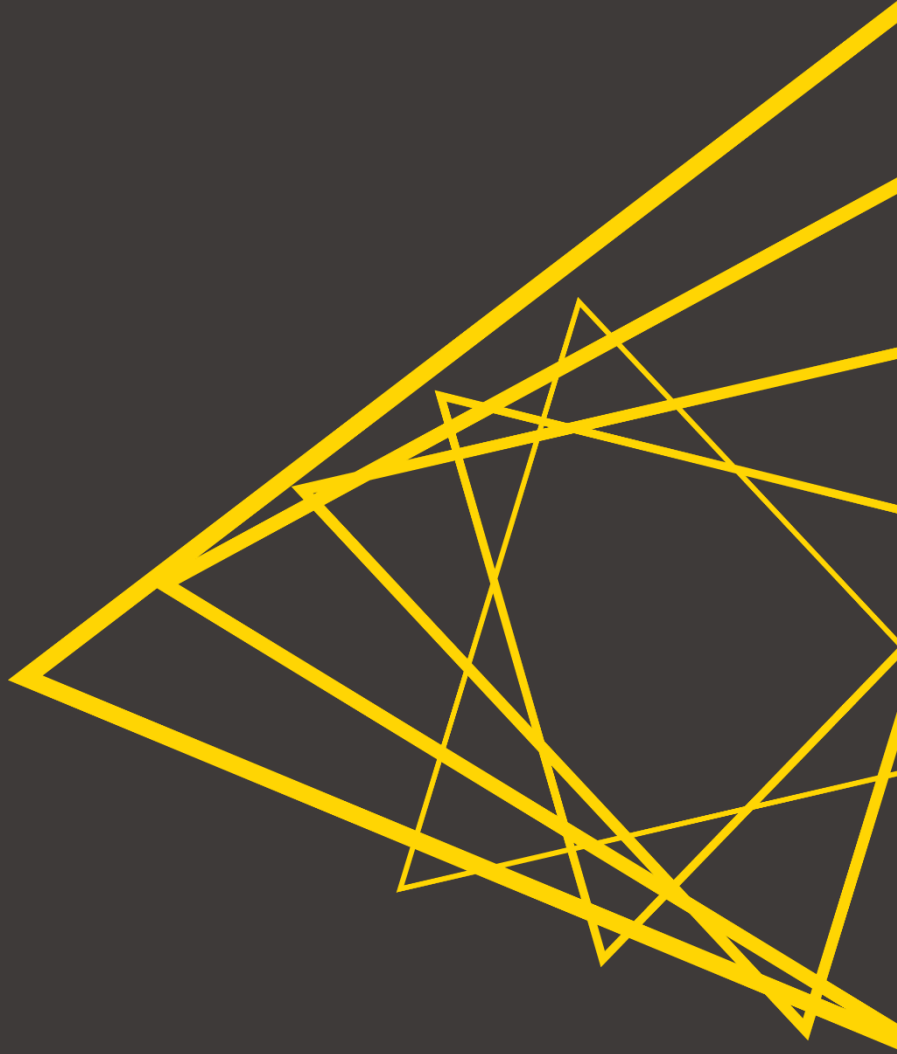


Learning Objectives

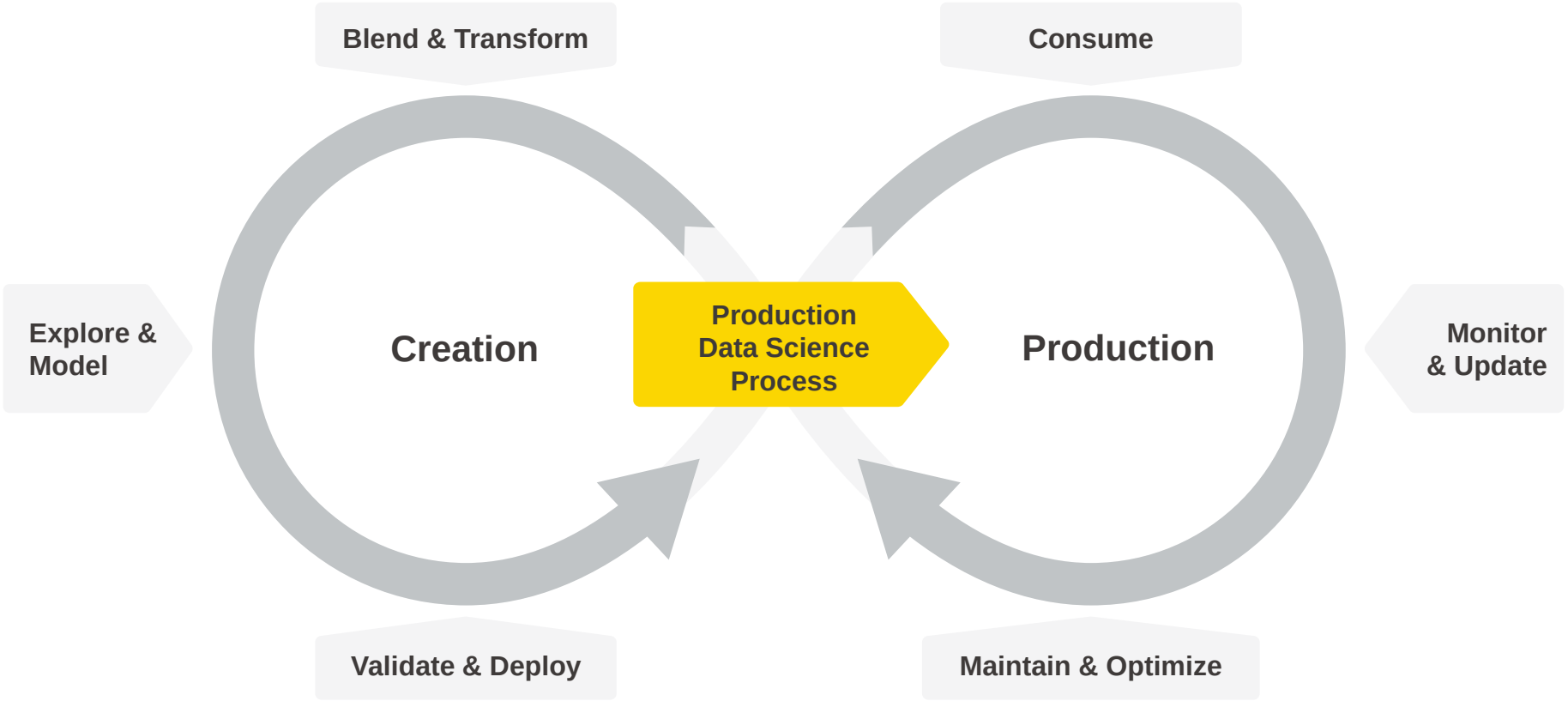
1. Identify the phases of the Data Science Life Cycle
2. Understand the applications of Integrated Deployment and Workflow Services
3. Identify differences and similarities between Integrated Deployment and Workflow Services
4. Use the tools to test workflows, components, and workflow segments
5. Identify the steps to deploy a workflow



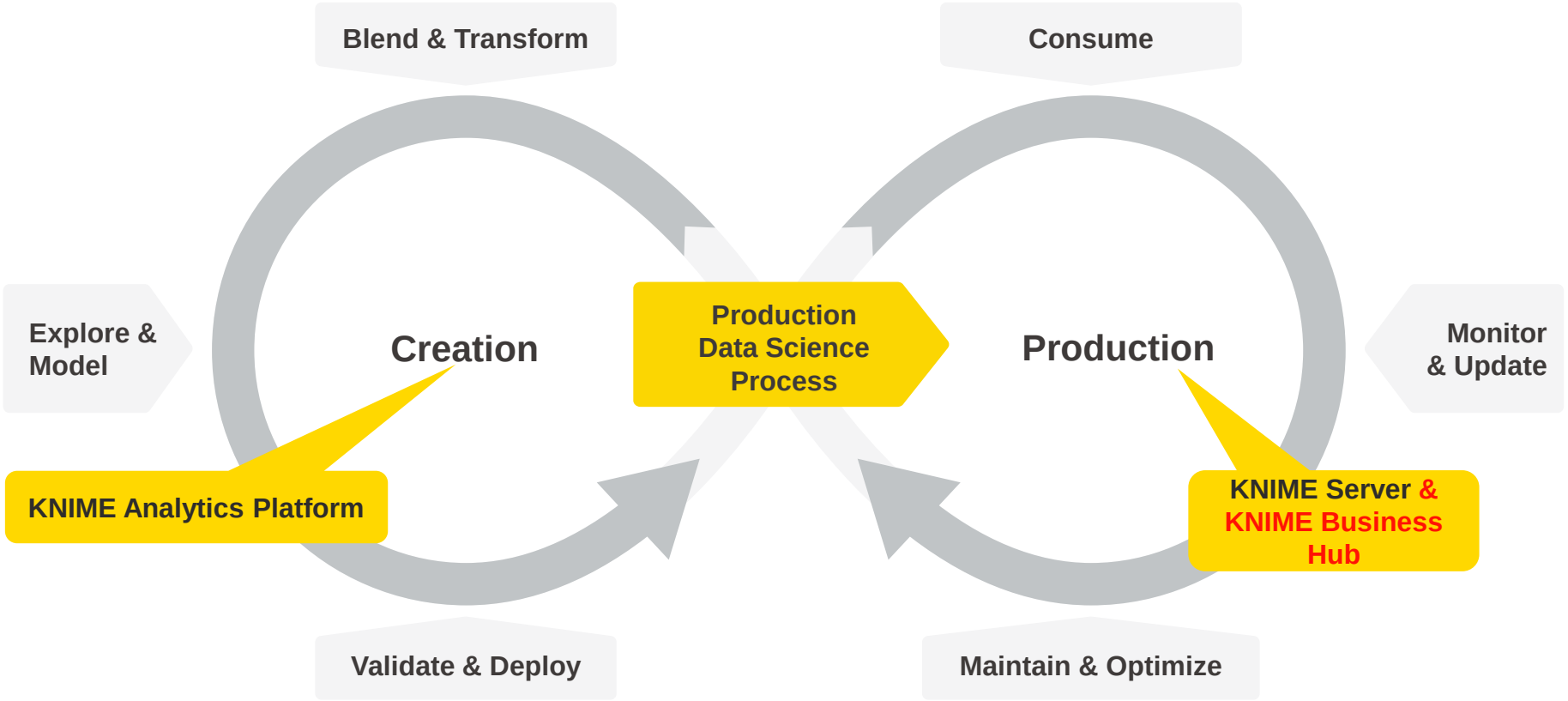
The Data Science Life Cycle



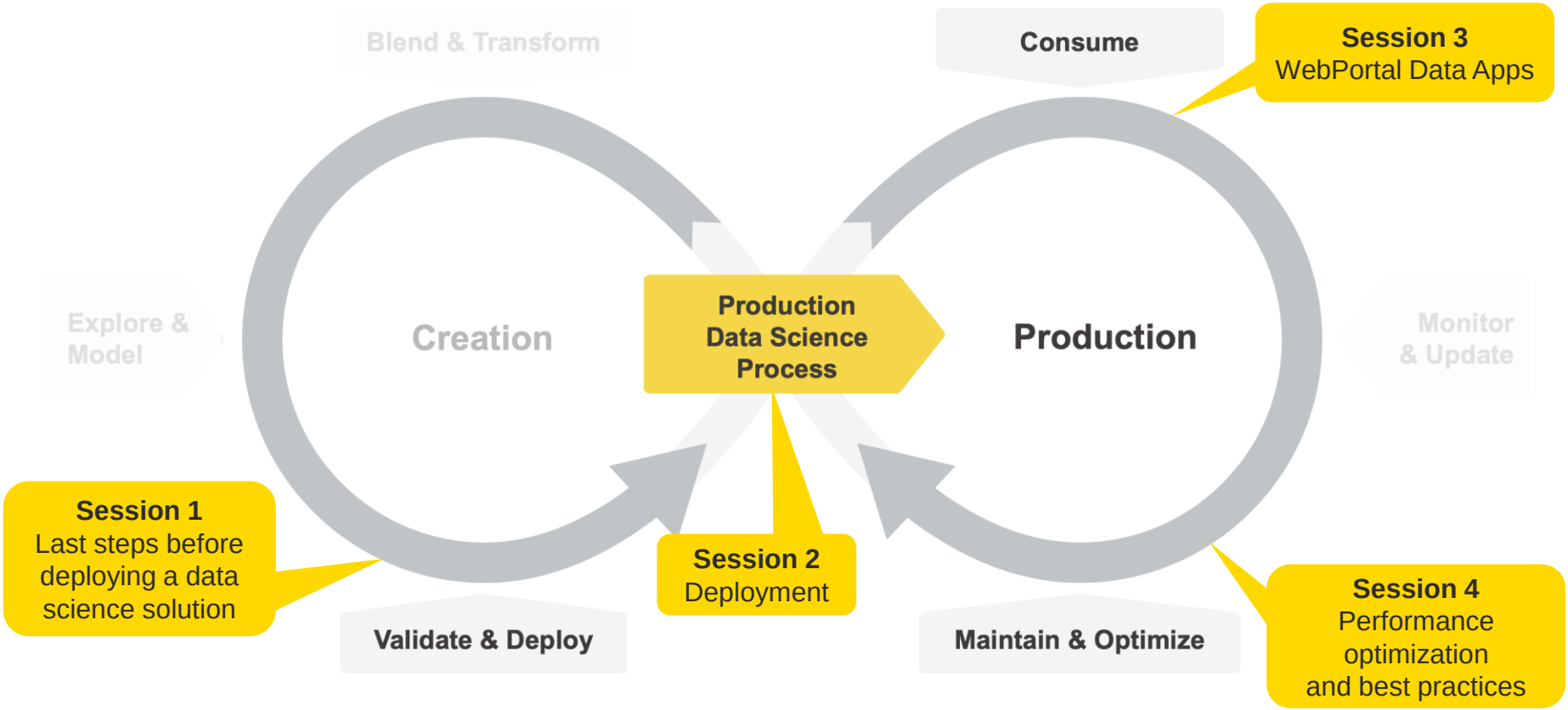
The Data Science Life Cycle



The Data Science Life Cycle

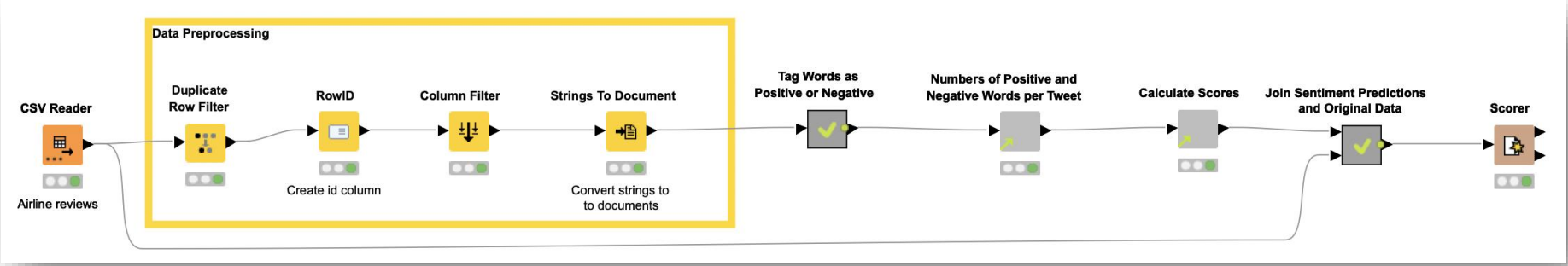


This Course's Sessions



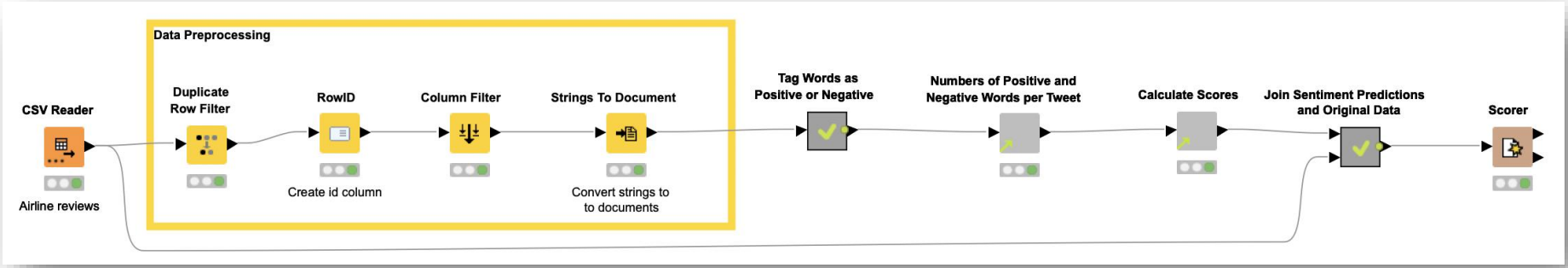
Today's Example

- **Goal:** Productionize a Sentiment Analysis application
  - Lexicon-based predictor: sentiment scores
  - Proof of concept: predictions for a Kaggle dataset with 14K customer tweets on six US airlines
  - Tweets are labeled as positive, negative, or neutral

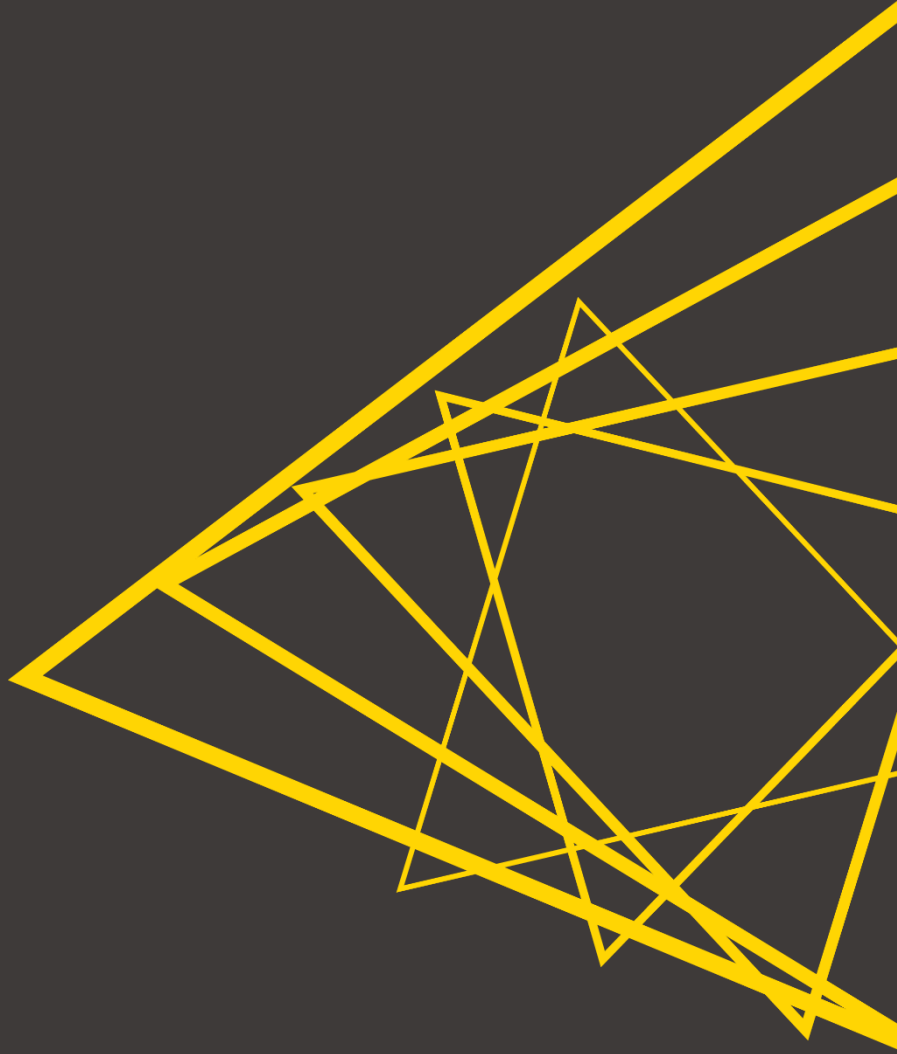


Today's Example

- How can we **test** the different parts of this application for deployment?
- How can we **deploy** it?

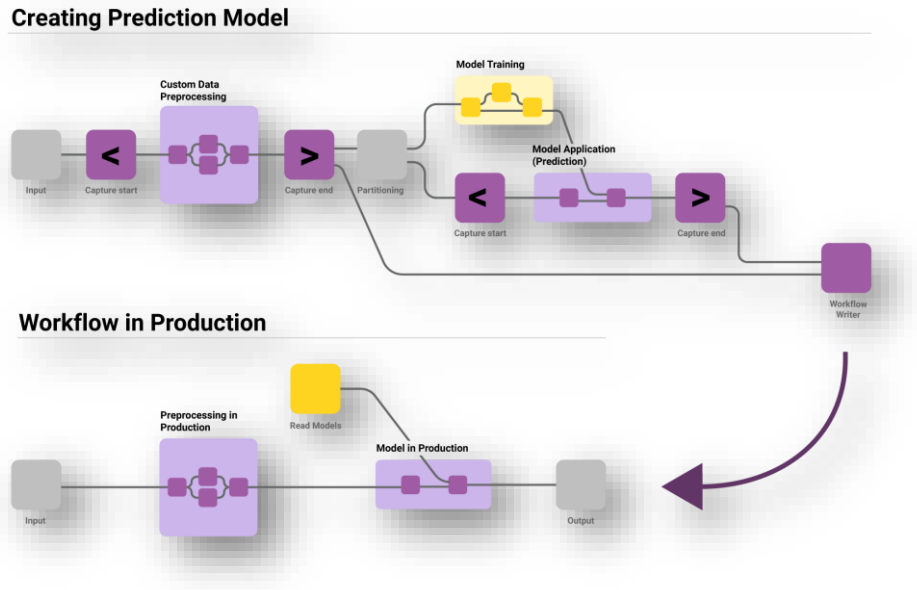


Integrated Deployment



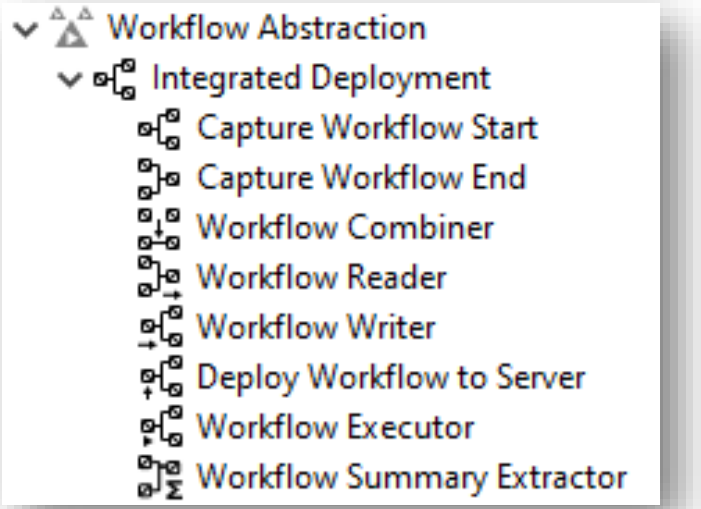
Integrated Deployment

- Integrated Deployment allows you to **capture core segments of your workflow for reuse**
  - Facilitates **independent testing** of different workflow functionalities
  - Facilitates **deployment of relevant parts** of a workflow (e.g., custom data preprocessing, model application for prediction)
- Captured segments are saved automatically as workflows with all the relevant settings and transformations
- Captured segments can be automatically reused with no changes or manual work required



Integrated Deployment

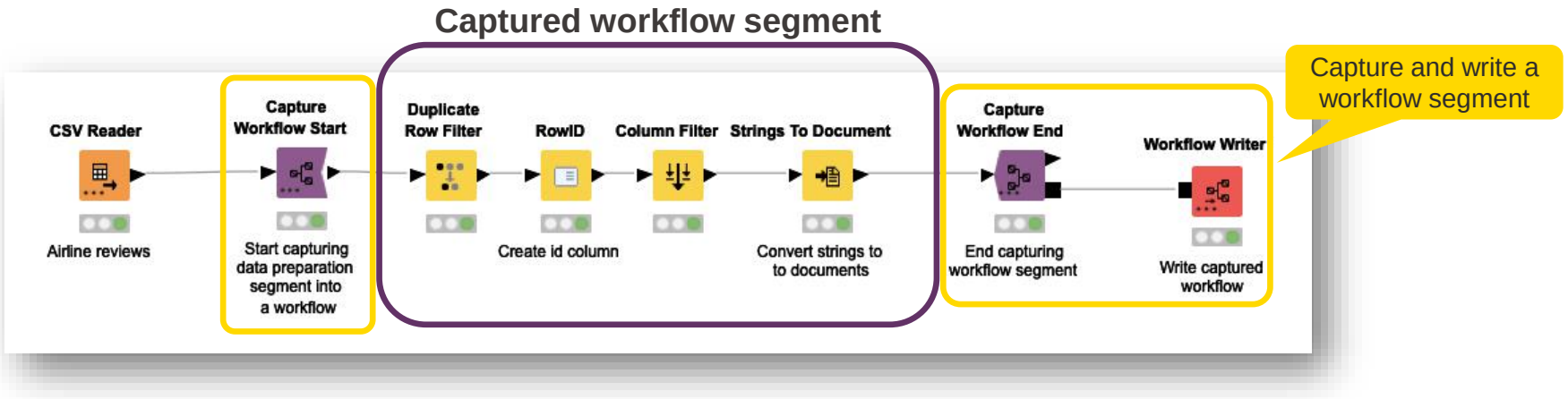
- Benefits
  - Time savings
  - Fewer errors
  - Increased compliance
  - Optimized processes
- Use cases
  - Data pipeline deployment
  - Model deployment
  - Ad-hoc testing
  - Workflow summary extraction



Since we are discussing testing and deployment today, Integrated Deployment is relevant

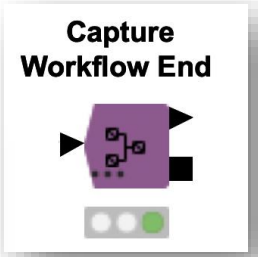
Capturing and Writing Workflow Segments

- Capture a workflow segment and save it using Integrated Deployment nodes



Capture Workflow End Node

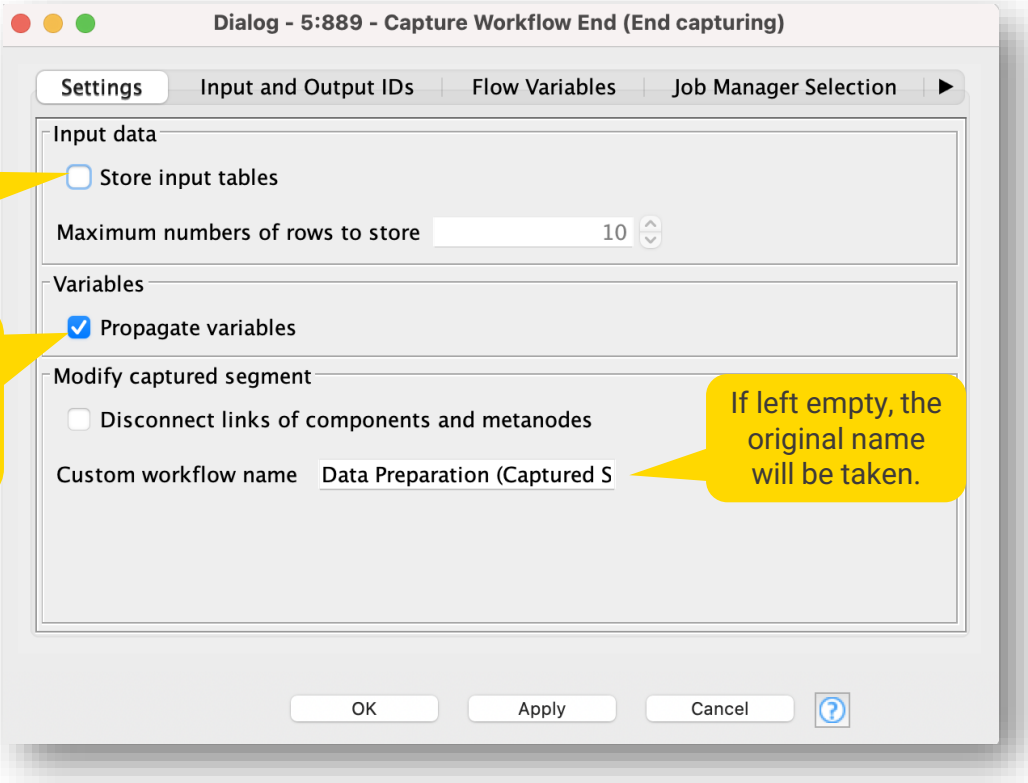
- Marks the end of a *“to be captured”* workflow segment



If checked, input tables at the extracted workflow’s will be stored

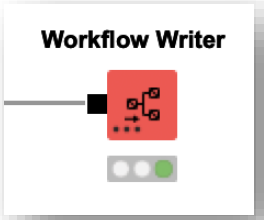
If checked, variables defined (or modified) within the *Capture* block are propagated downstream

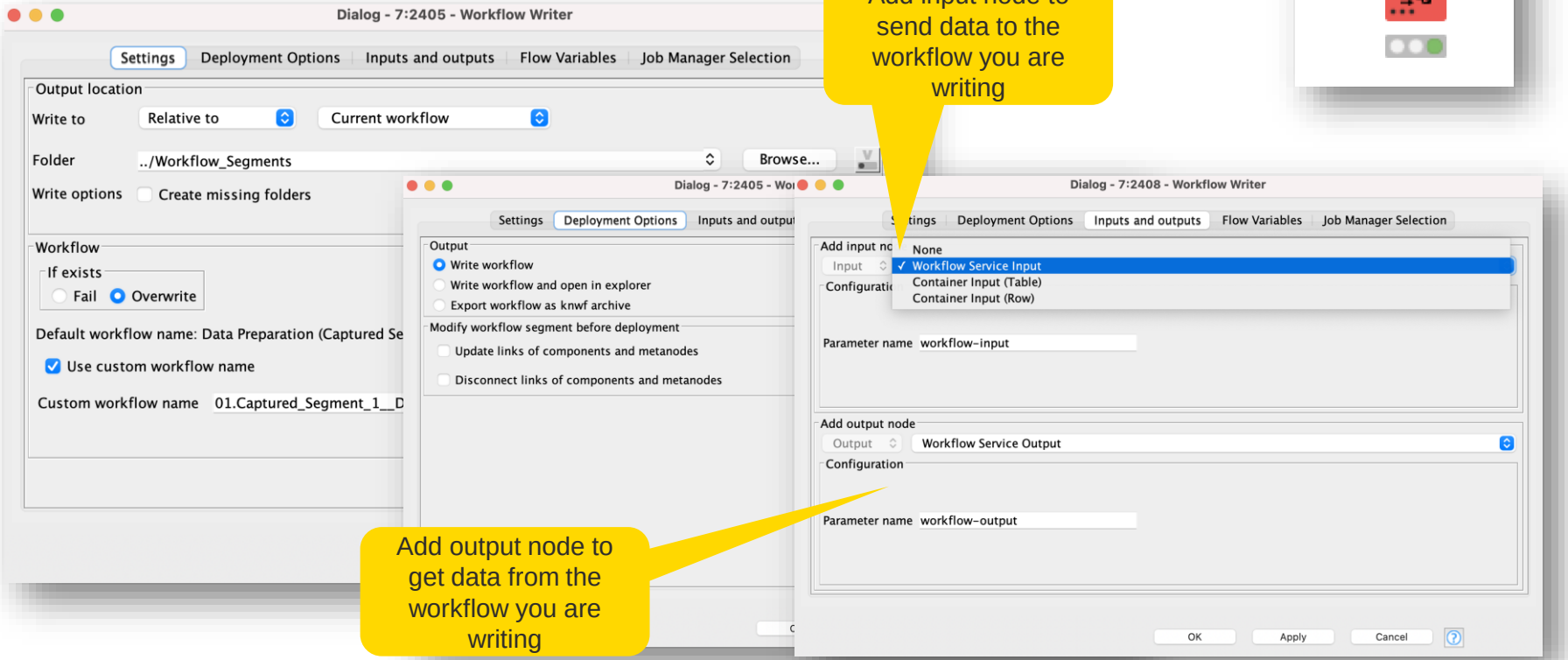
If left empty, the original name will be taken.



Workflow Writer Node

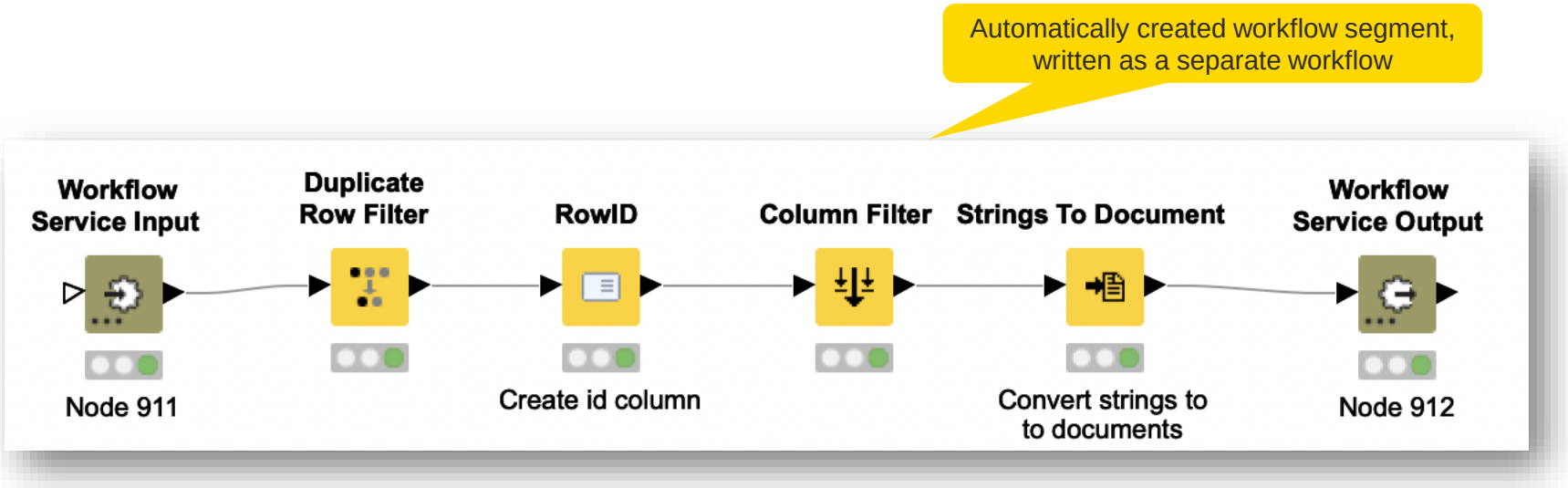
- Write captured workflow to selected output location





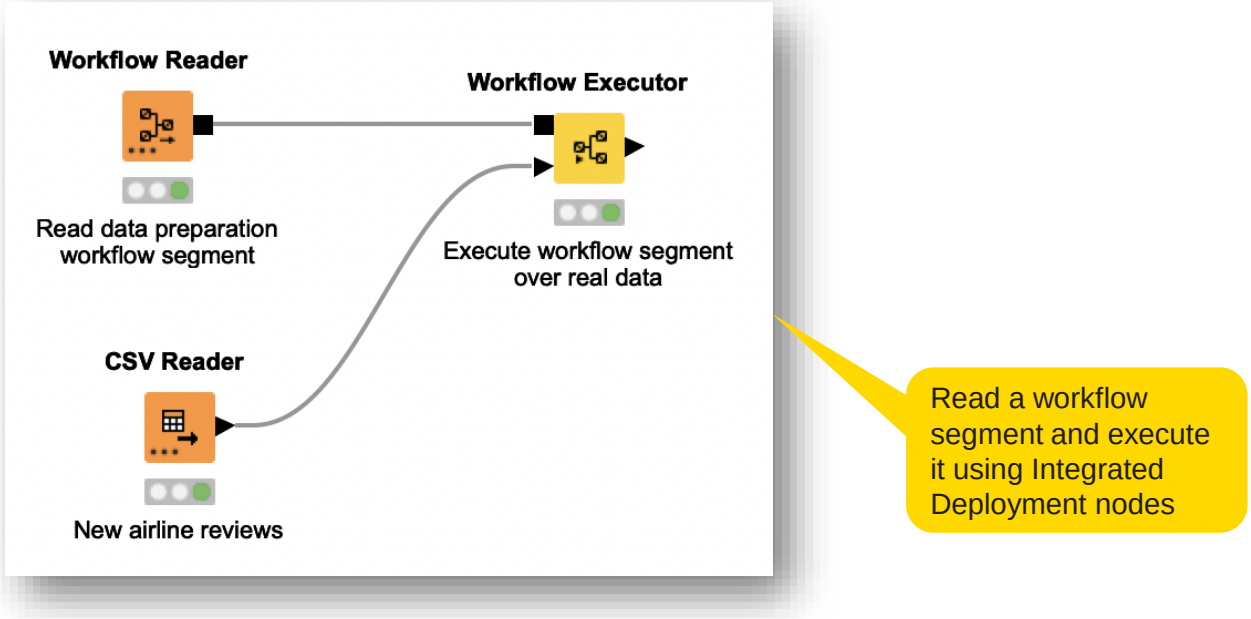
Written Workflow Segment

- Automatically capture a workflow segment and save it using Integrated Deployment nodes



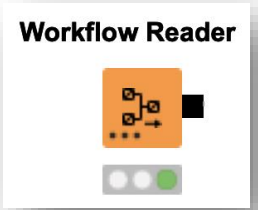
Reading and Executing Workflow Segment

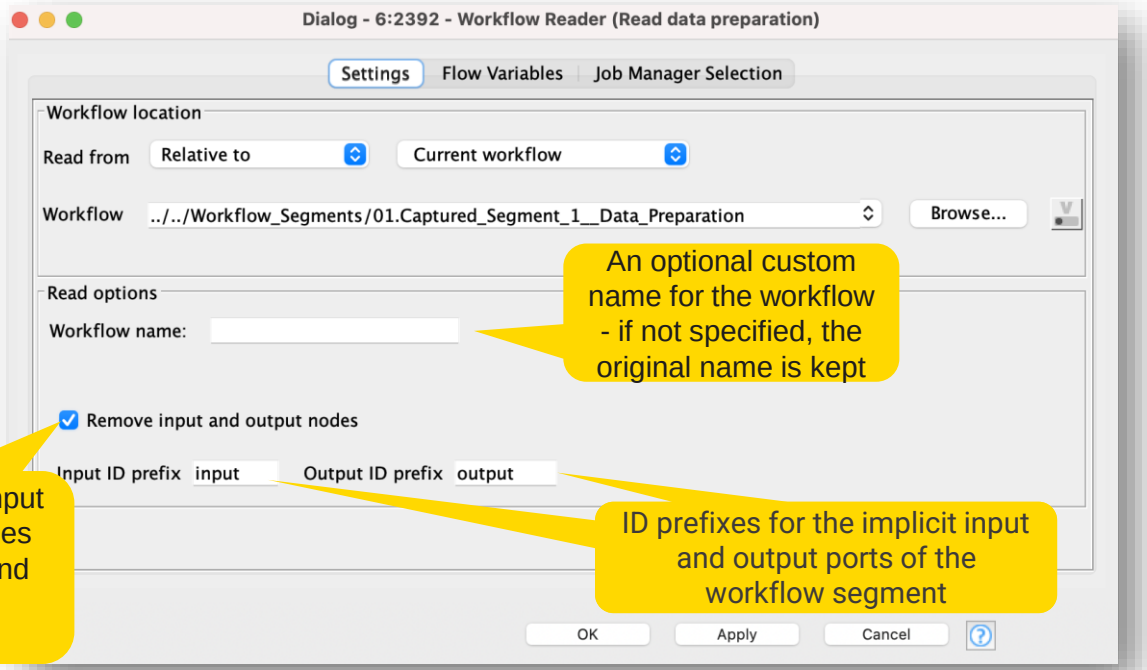
- The Workflow Reader node can be used in conjunction with the Workflow Executor node



Workflow Reader Node

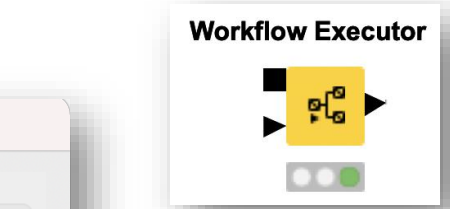
- Reads a single workflow into a workflow port object

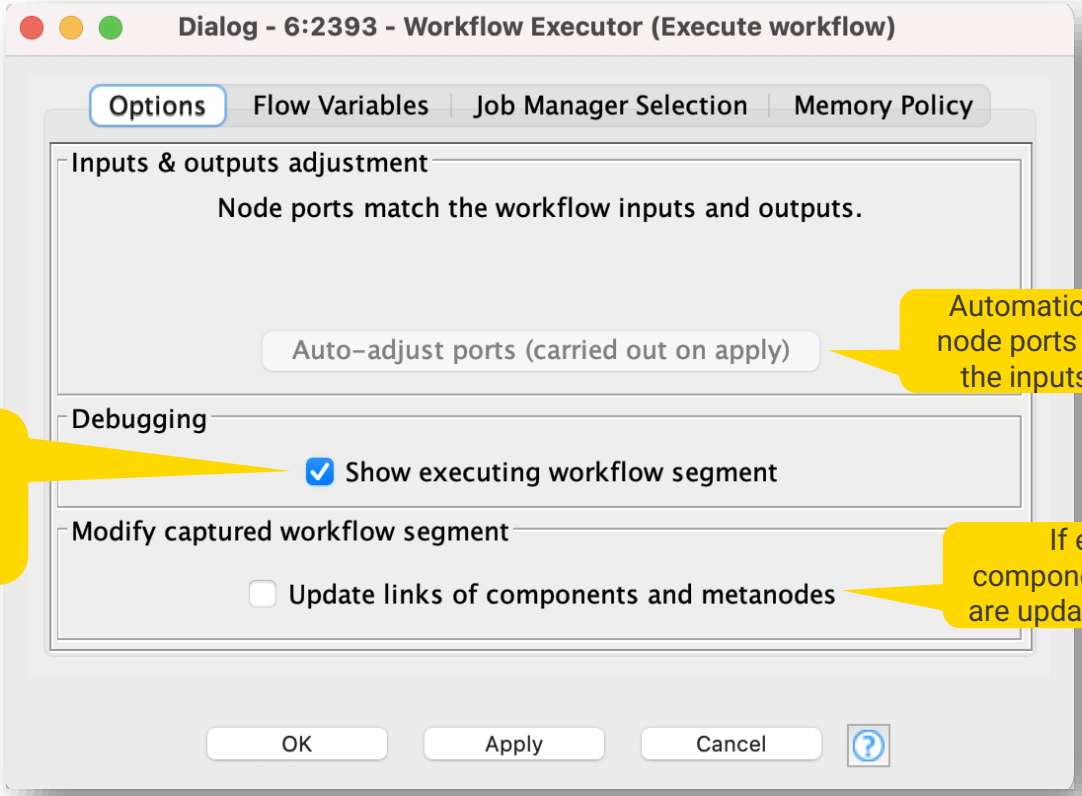




Workflow Executor Node

- Executes a workflow provided at the workflow input port





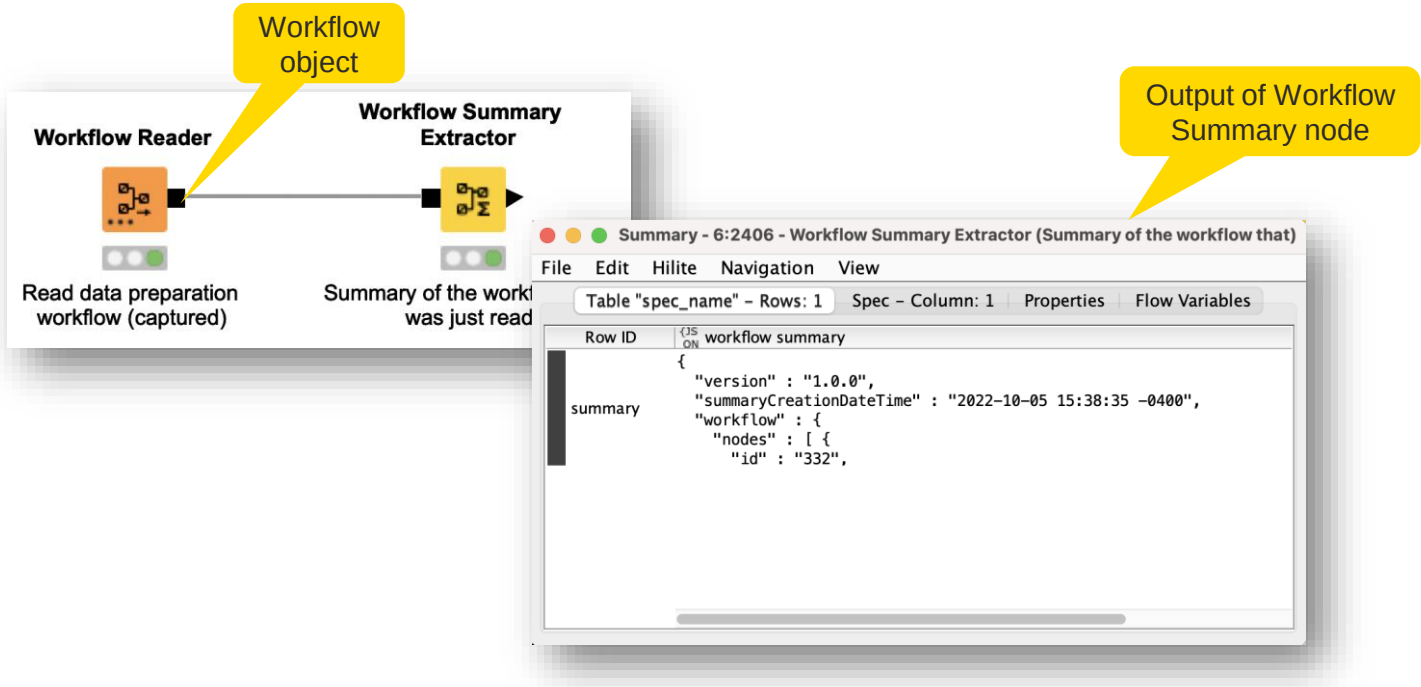
Automatically adjust the node ports with respect to the inputs and outputs

If enabled, the executing workflow is visible as part of a metanode next to this node.

If enabled, linked components and metanodes are updated before execution

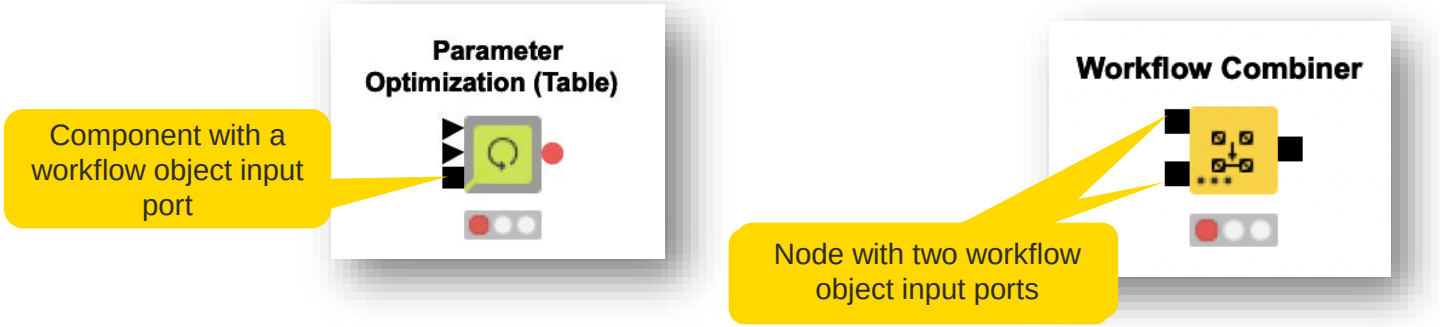
When to Use the Workflow Reader Node?

- If you want to read a workflow object, even if you do not want to execute it right away
  - e.g., if you want to use the Workflow Summary Extractor node to analyze the workflow that was just read



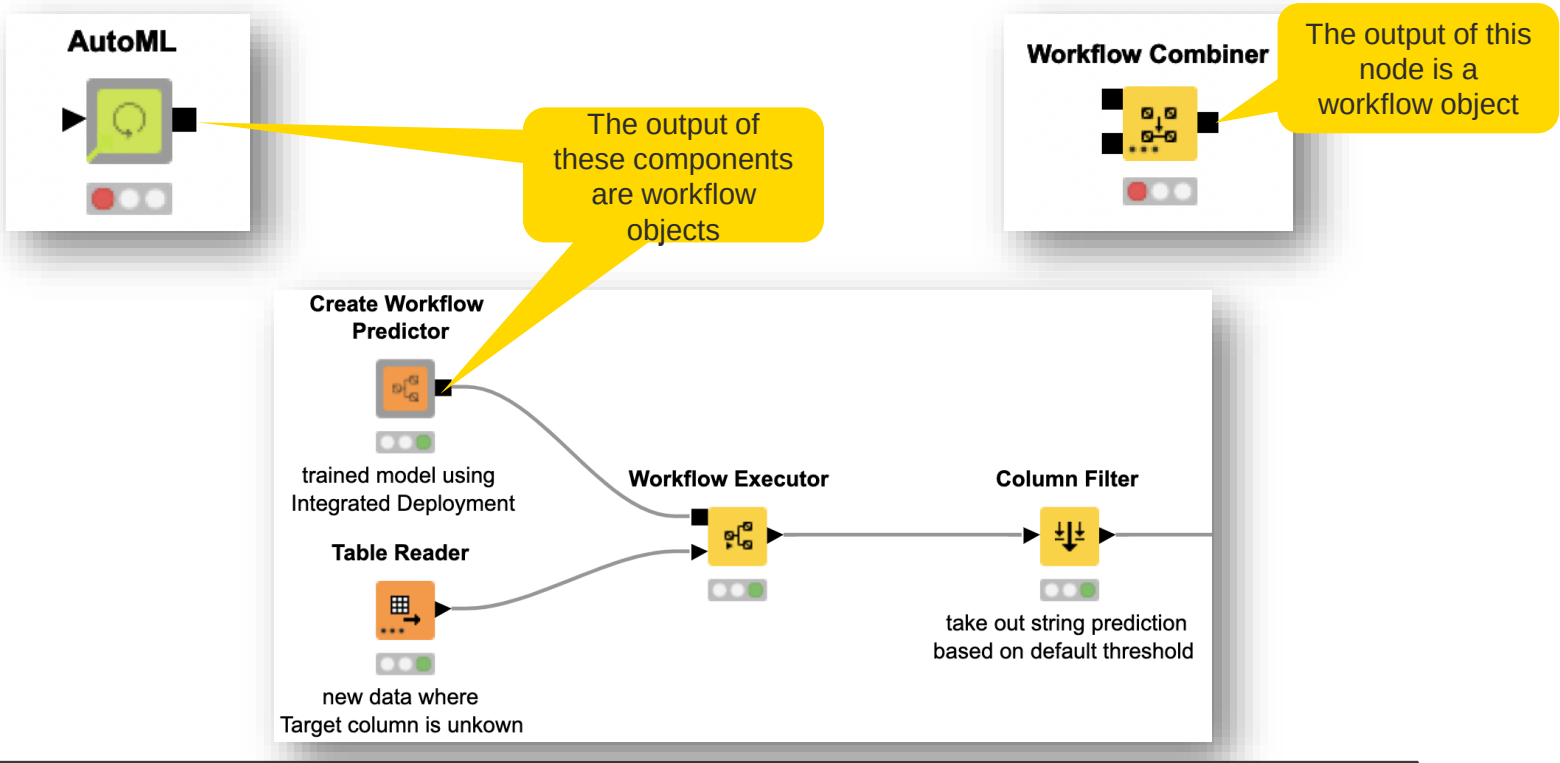
When to Use the Workflow Reader Node?

- If you want to read a workflow object and provide it to a node or component as input
  - The treatment of the workflow object is delegated to the node or component



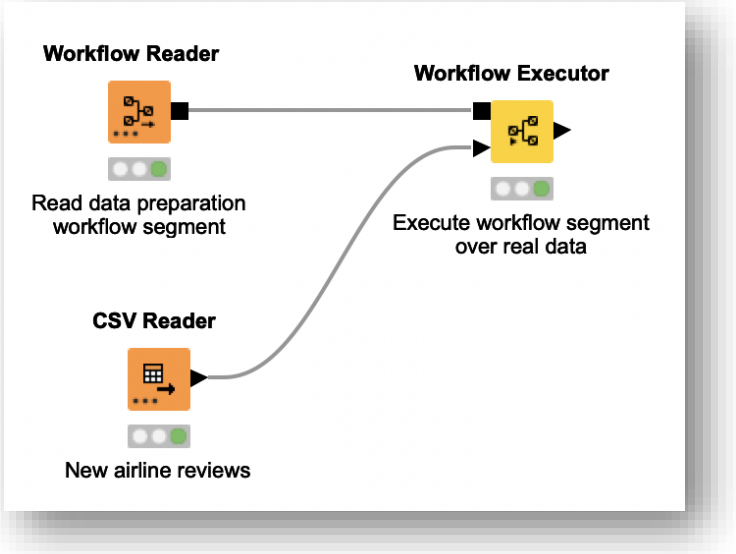
When to Use the Workflow Executor Node?

- When you want to execute a workflow object that is the output of a node or component

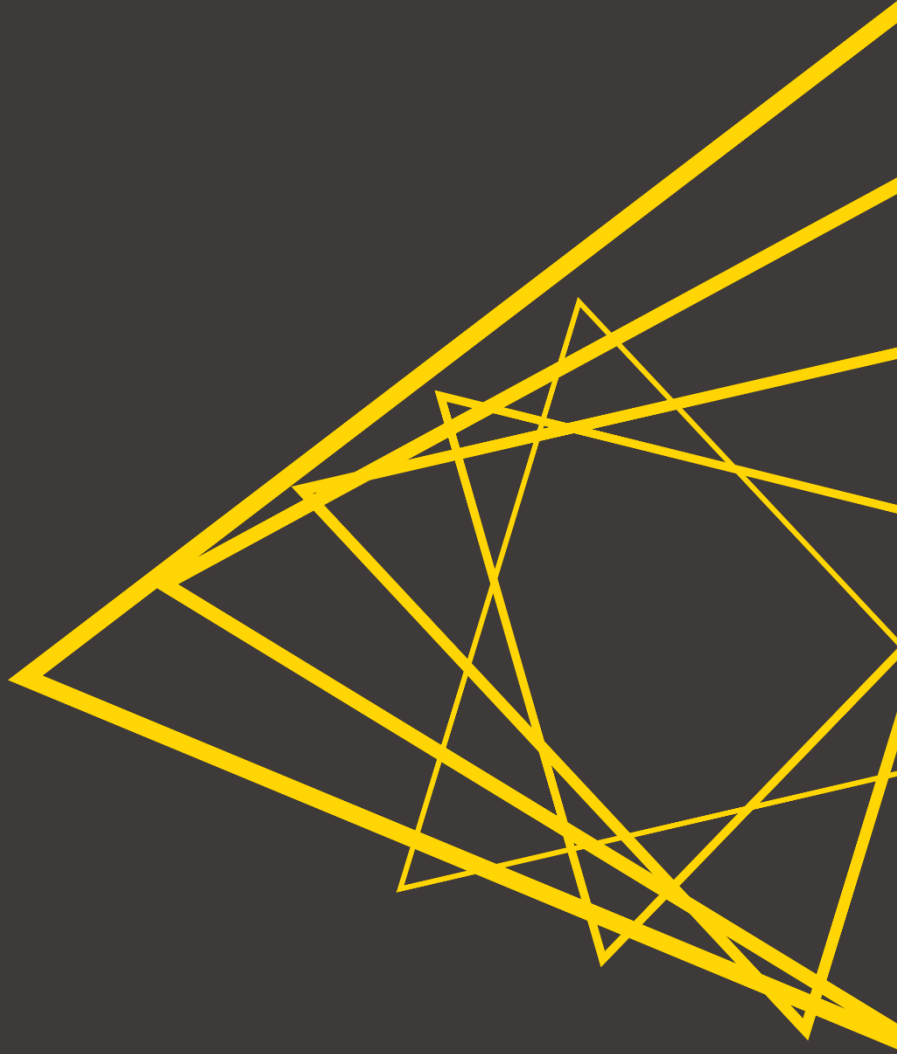


Disadvantages: Workflow Reader and Workflow Executor

- The Workflow Reader node cannot access the data area of the workflow that is read
  - If that data is used during execution, the workflow will crash
- The Workflow Reader node does not read any partially executed workflow - everything gets reset

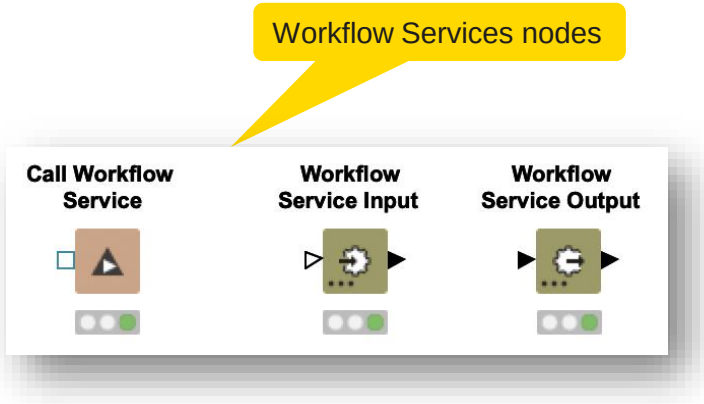


Workflow Services



Workflow Services

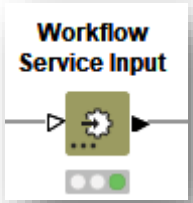
- Workflow Services allow a workflow (caller) to invoke other workflows (callees)
  - Send input to, execute, and obtain output of callee workflows
- Workflow Services:
  - For **KNIME use** only - easier and **faster** to call KNIME workflows from other workflows
  - KNIME native API endpoints – no serialization into/from JSON-objects
  - Share text, models, and many more data types
  - Are commonly used to orchestrate workflows (**Session 4 of this course**)

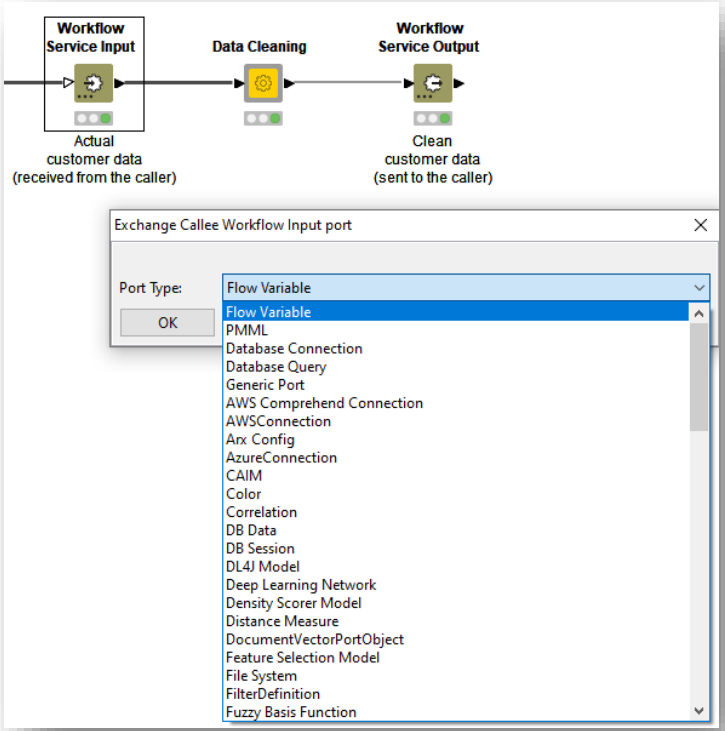


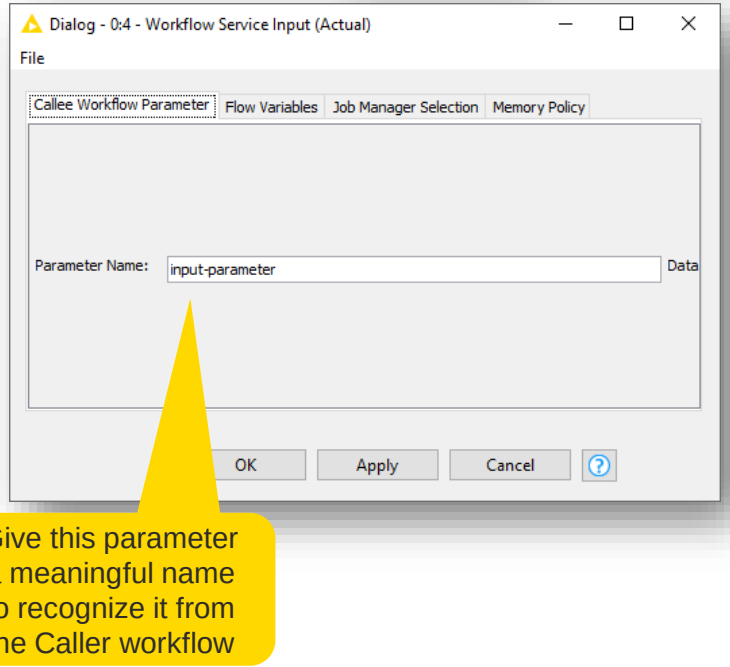
Since we will be discussing deployment today, the efficiency and flexibility of these nodes is relevant

Workflow Service Input Node

- Receives an object from a caller workflow
- Various port types are available

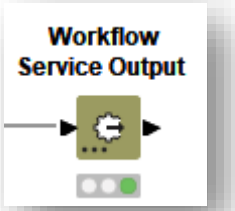


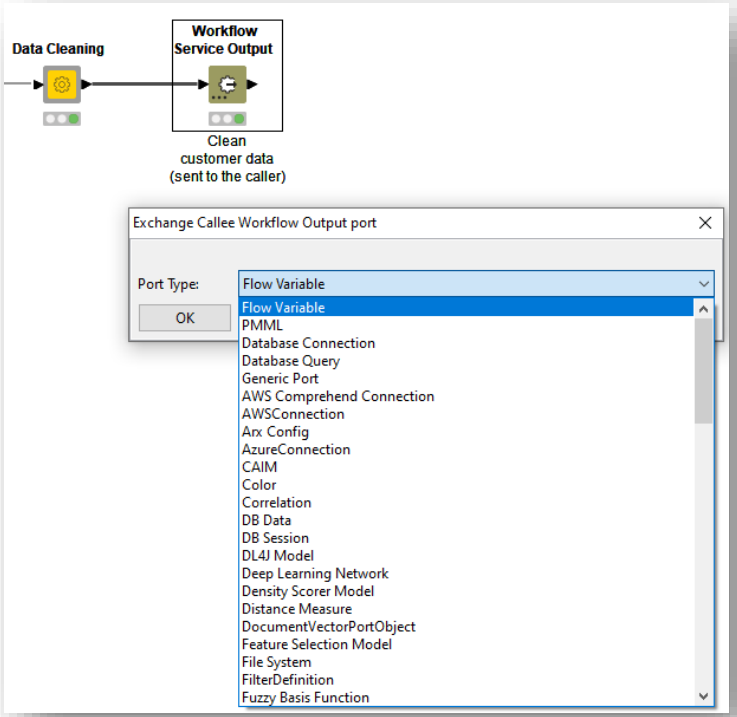


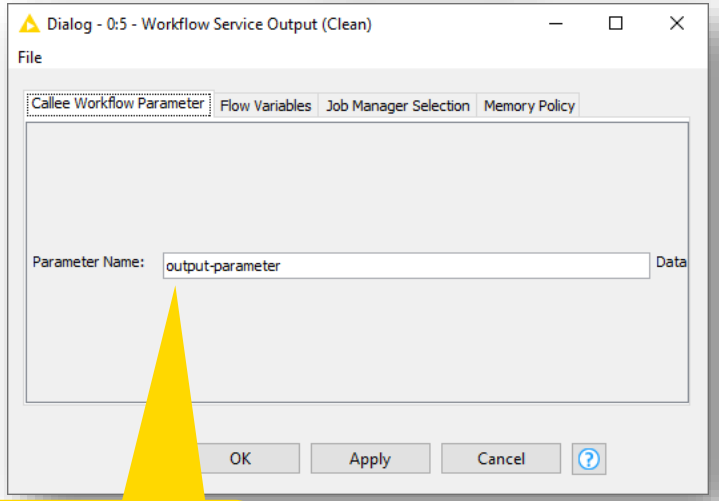


Workflow Service Output Node

- Sends an object to a caller workflow
- Various port types are available



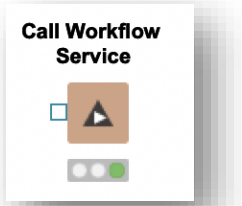


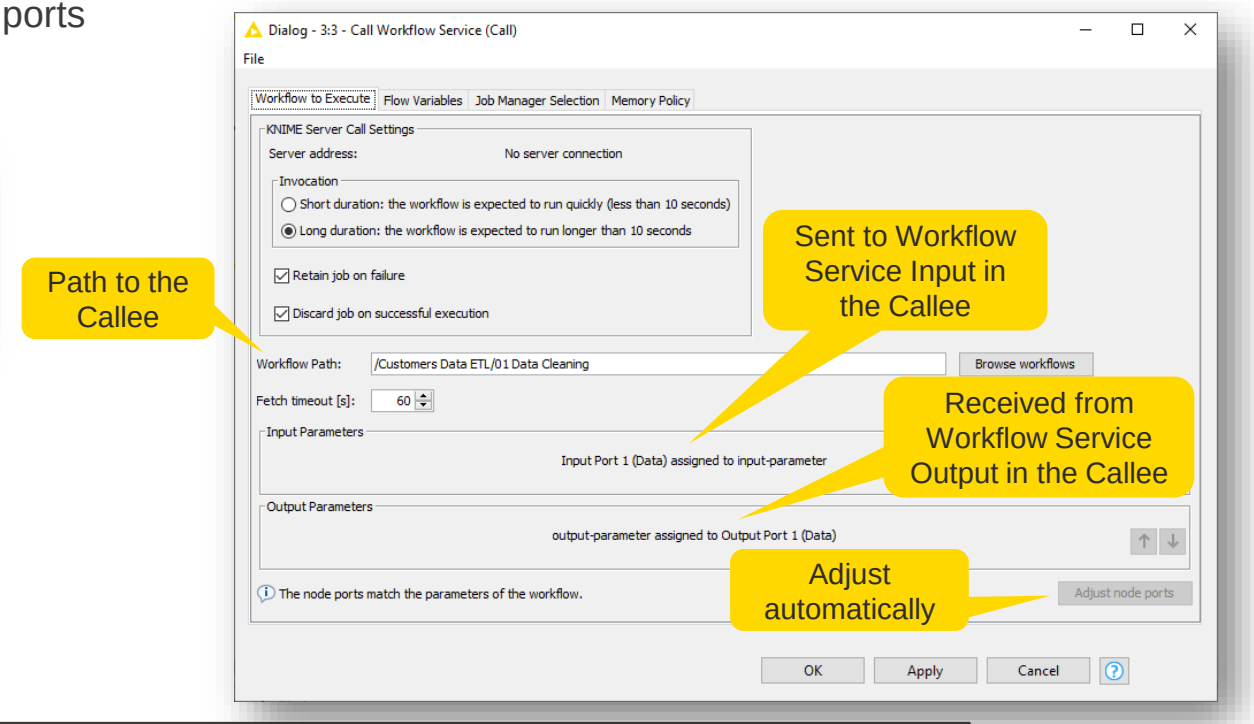


Give this parameter a meaningful name to recognize it from the Caller workflow

Call Workflow Service Node

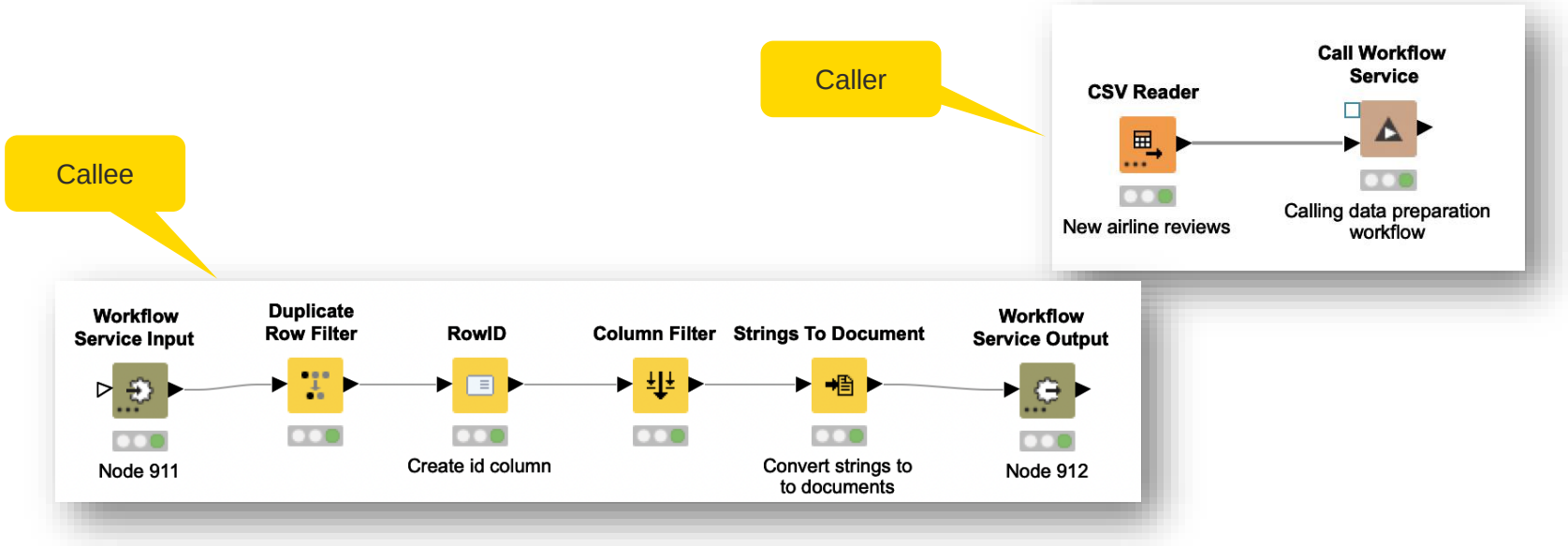
- Calls other local and remote workflows
- Ports are adjusted automatically in accordance with the callee workflow selected in the configuration dialog
  - Various port types, multiple ports





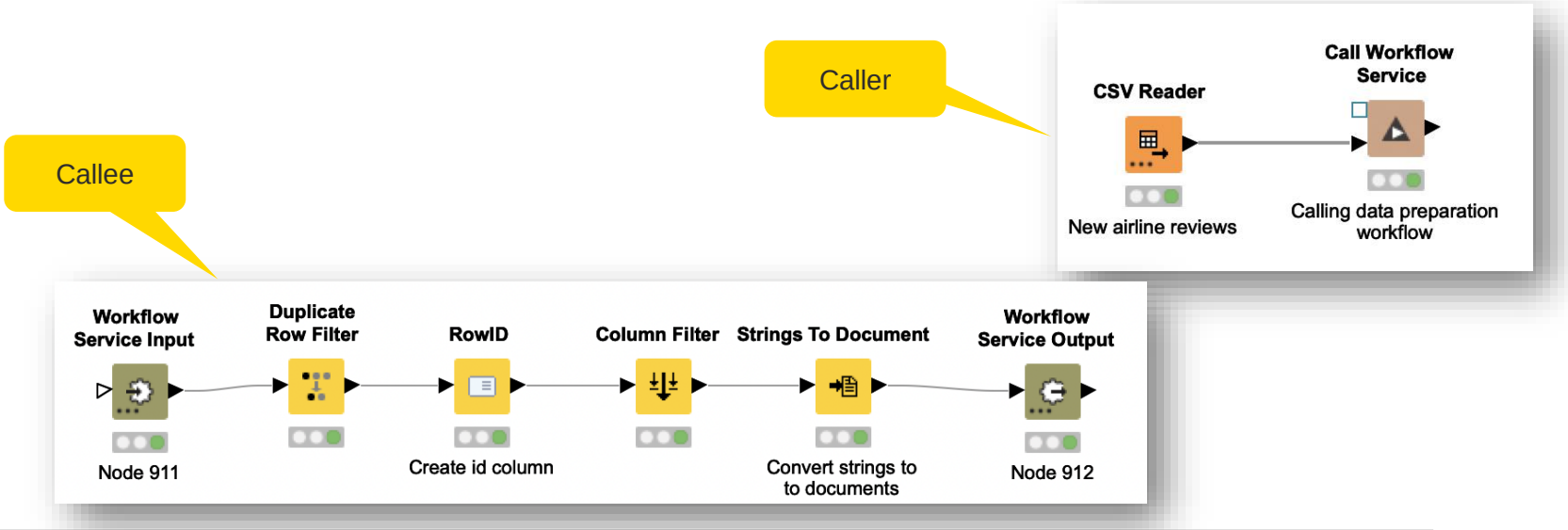
When to Use the Call Workflow Service Node?

- The Call Workflow Service node can be used any time we want a workflow (caller) to invoke another workflow (callee)
  - Send input to, execute, & receive results from callee workflow



Advantages: Call Workflow Service

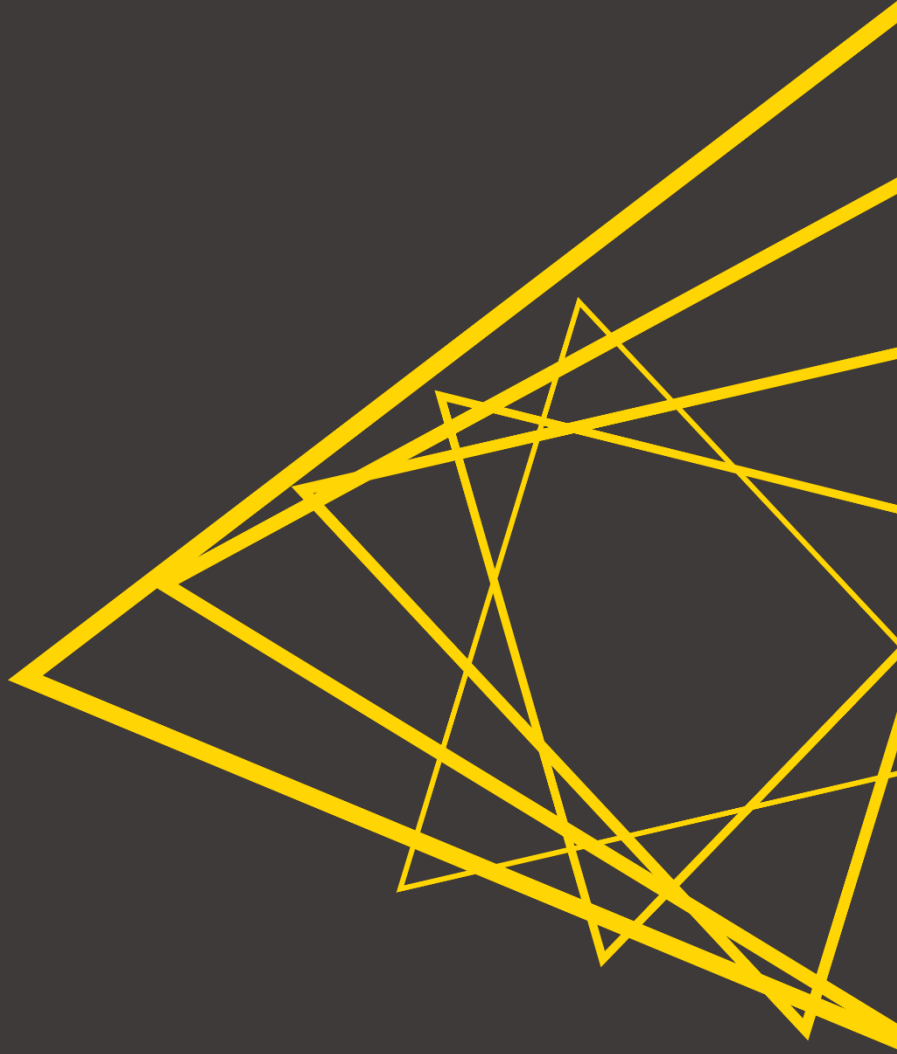
- Advantages w.r.t. Workflow Reader and Workflow Executor nodes:
  - It handles the data areas of invoked workflows adequately
  - It leverages the infrastructure of KNIME Server and KNIME Business Hub, leading to better scalability
  - It is more efficient than using Workflow Reader and Workflow Executor nodes



KNIME Knowledge Check

- How many Workflow Services nodes are required to invoke a callee workflow?
  - And what if you were using Integrated Deployment?

Testing



Testing is a Form of Validation

- Why test?
  - Guarantee that a KNIME workflow, component, or workflow segment works as expected
  - Detect issues as early as possible

- When to test?
  - Regularly
  - After changes in the environment
    - For example, new KNIME Analytics Platform release or updates in the workflow
  - **Test before deployment** and create a testing framework for future redeployments

- How to test?
  - Tests should be automated
    - Separate special workflows (testflows) comparing expected outcomes with the actual outcomes of components, workflows, or workflow segments

Different Types of Test

- Unit test
  - Checks if a small part of the workflow that can be logically isolated (e.g., a component) is behaving correctly
- Application test
  - Checks if an entire application meets functionality, usability, or consistency requirements
- Model test
  - Checks if a model reaches the expected performance, and/or behaves as expected for given inputs
- Integration test
  - Checks if the integration across different modules in an application works as intended (e.g., the modules communicate without issues)
- Stress test
  - Checks if an application is stable and reliable under “stressful” conditions (e.g., too many requests or too much data)

Different Types of Test

- Unit test — **Focus today**
  - Checks if a small part of the workflow that can be logically isolated (e.g., a component) is behaving correctly
- Application test
  - Checks if an entire application meets functionality, usability, or consistency requirements
- Model test
  - Checks if a model reaches the expected performance, and/or behaves as expected for given inputs
- Integration test
  - Checks if the integration across different modules in an application works as intended (e.g., the modules communicate without issues)
- Stress test
  - Checks if an application is stable and reliable under “stressful” conditions (e.g., too many requests or too much data)

Testflows for Unit and Application Tests

- Testflows – KNIME workflows that
  - Provide test input (“golden” or baseline tables or files) to a workflow
  - Check that the outputs align with the test input, otherwise raise alerts
  - Check that things work OR that things break in an expected way

Possible issues	Possible reasons
Failing for unexpected reasons, wrong output, unexpected error messages	Invalid inputs, configuration, changes during an update, changes in external services

“Golden” Tables, Data, Files, etc.

- Golden table is a small sample of input or output data used in the test (same for golden data, files, etc.)
  - Input golden table – processed input to be supplied to the object of testing
  - Output golden table – expected result of the object of testing

- Do not process the golden data to avoid new errors
- Do not change the golden data
  - Keep in the separate folder for testing
  - Restrict permissions if possible

Testing in KNIME Analytics Platform

- KNIME Testing Framework UI extension
- Additionally, use other nodes to set up your custom tests

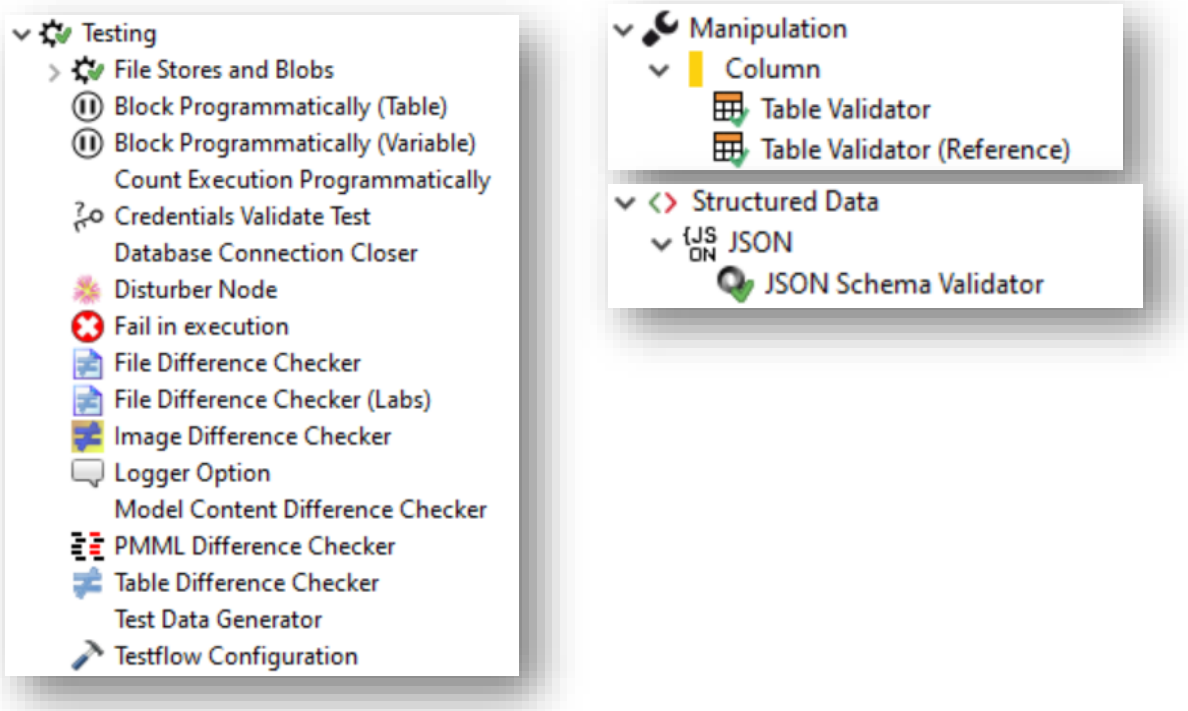


Table Difference Checker & Table Difference Finder Nodes

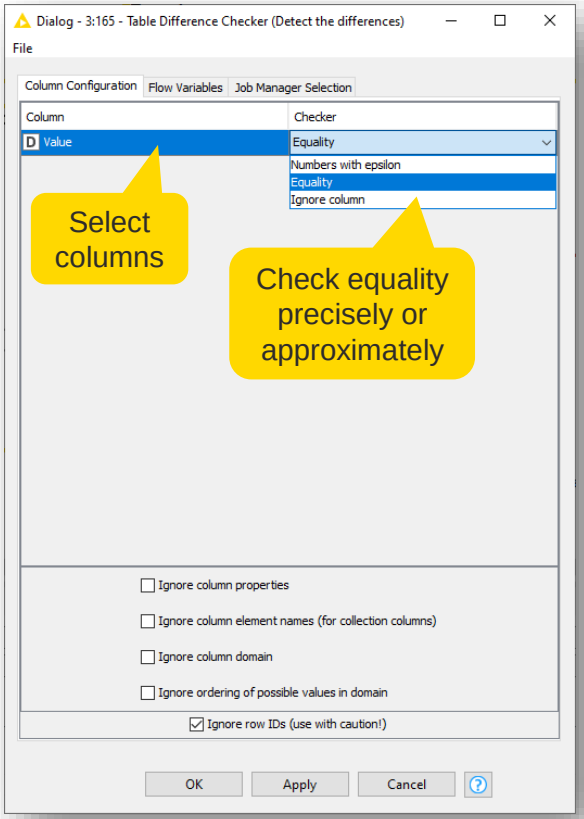
- Determine if there is any difference between two tables

Fails if there are differences

Table Difference Checker

Actual output

Output golden table



Select columns

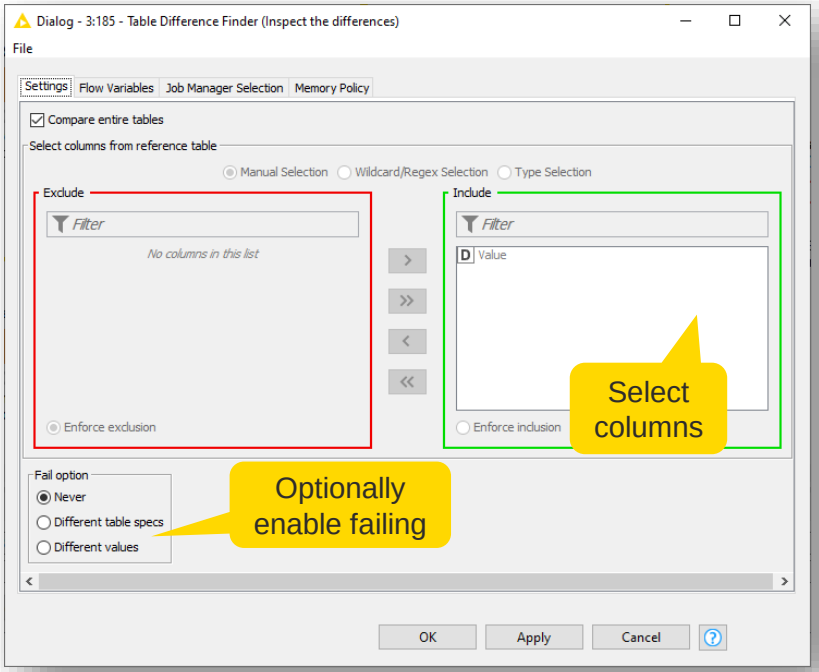
Check equality precisely or approximately

Returns the differences

Table Difference Finder

Value differences

Domain differences

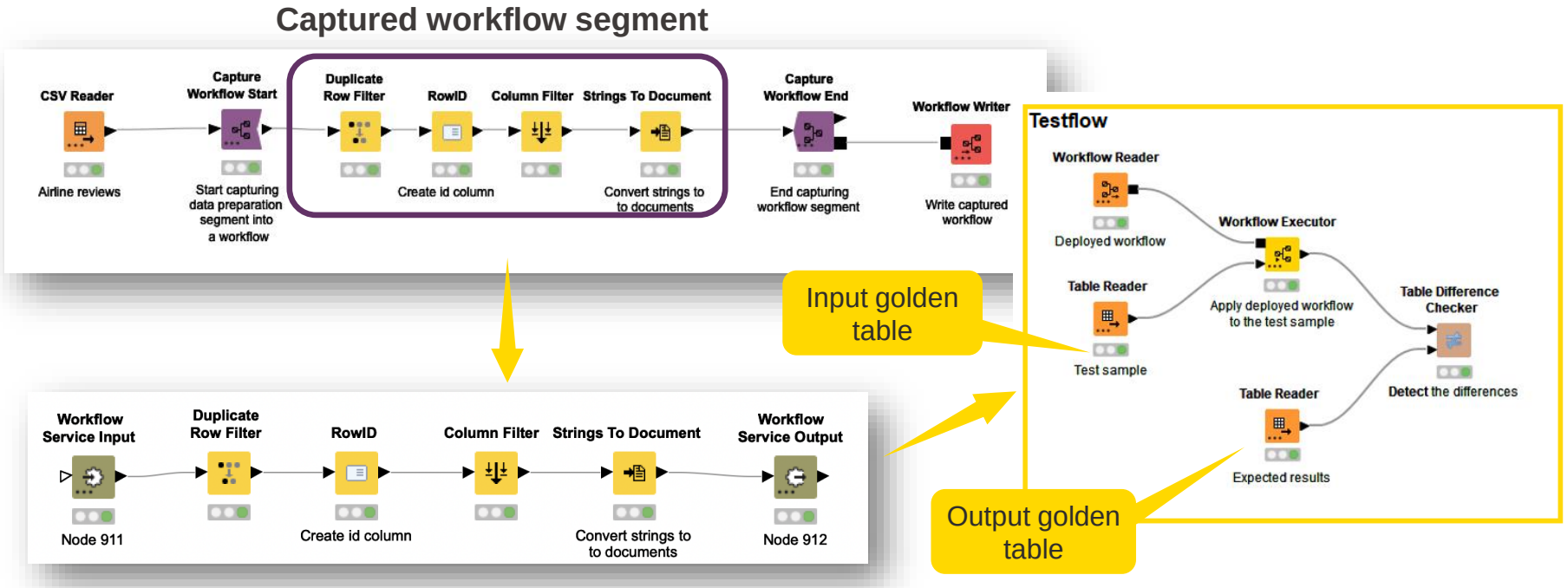


Select columns

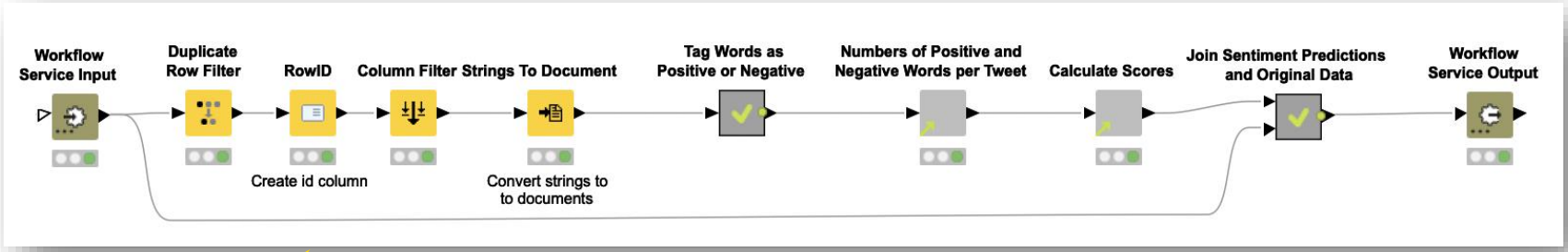
Optionally enable failing

Testing a Workflow Segment: Unit Test

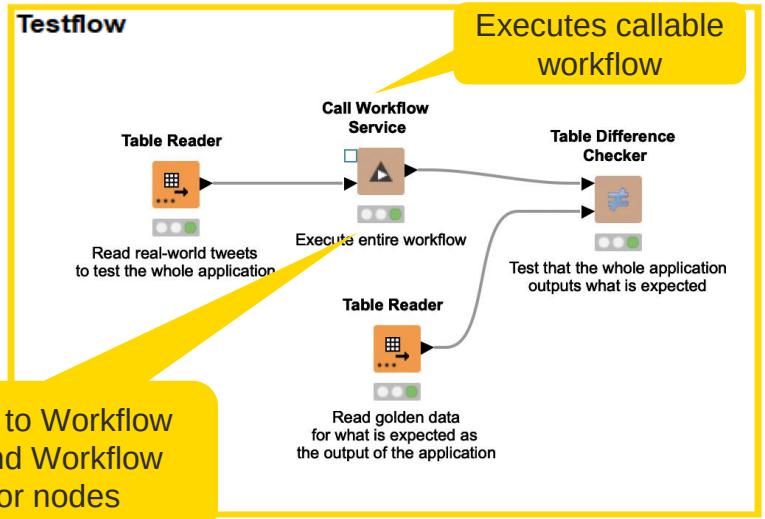
- A simple and elegant way to test workflow segments
  - Capture a workflow segment and save it using Integrated Deployment nodes
  - Provide the Workflow Executor node with the captured workflow and execute it on the test data
  - Perform tests on the output



Testing an Entire Workflow: Application Test



Lexicon-based Sentiment Analysis Predictor

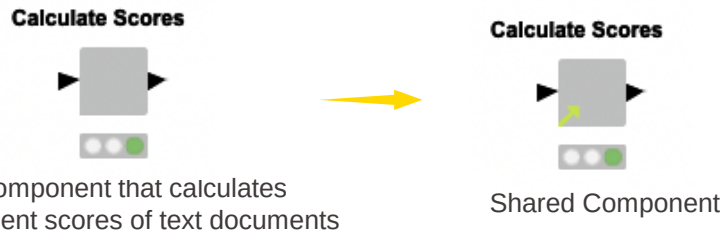


Executes callable workflow

Alternative to Workflow Reader and Workflow Executor nodes

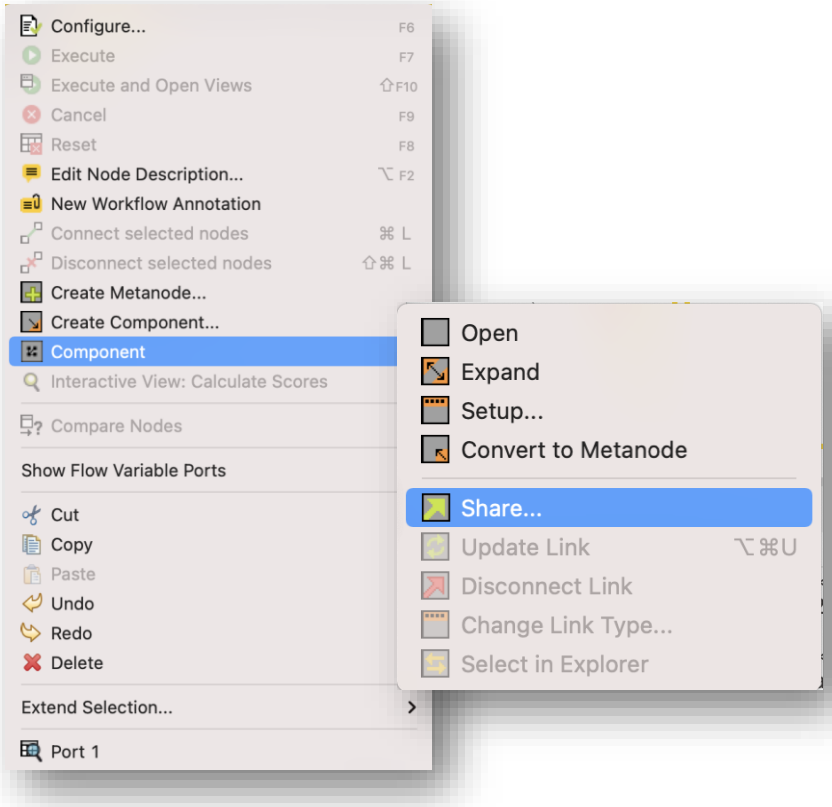
Testing a Component: First We Must Share It

- To test a component independent from what workflows use it, we must share it first
- What is a Shared Component?
  - Components can be saved in your KNIME workspace for later reuse
  - To do this, right-click any Component and select “Share...”
  - Shared Components are read-only instances of a Component
  - Public Shared Components are available on the EXAMPLES Server and on KNIME Hub

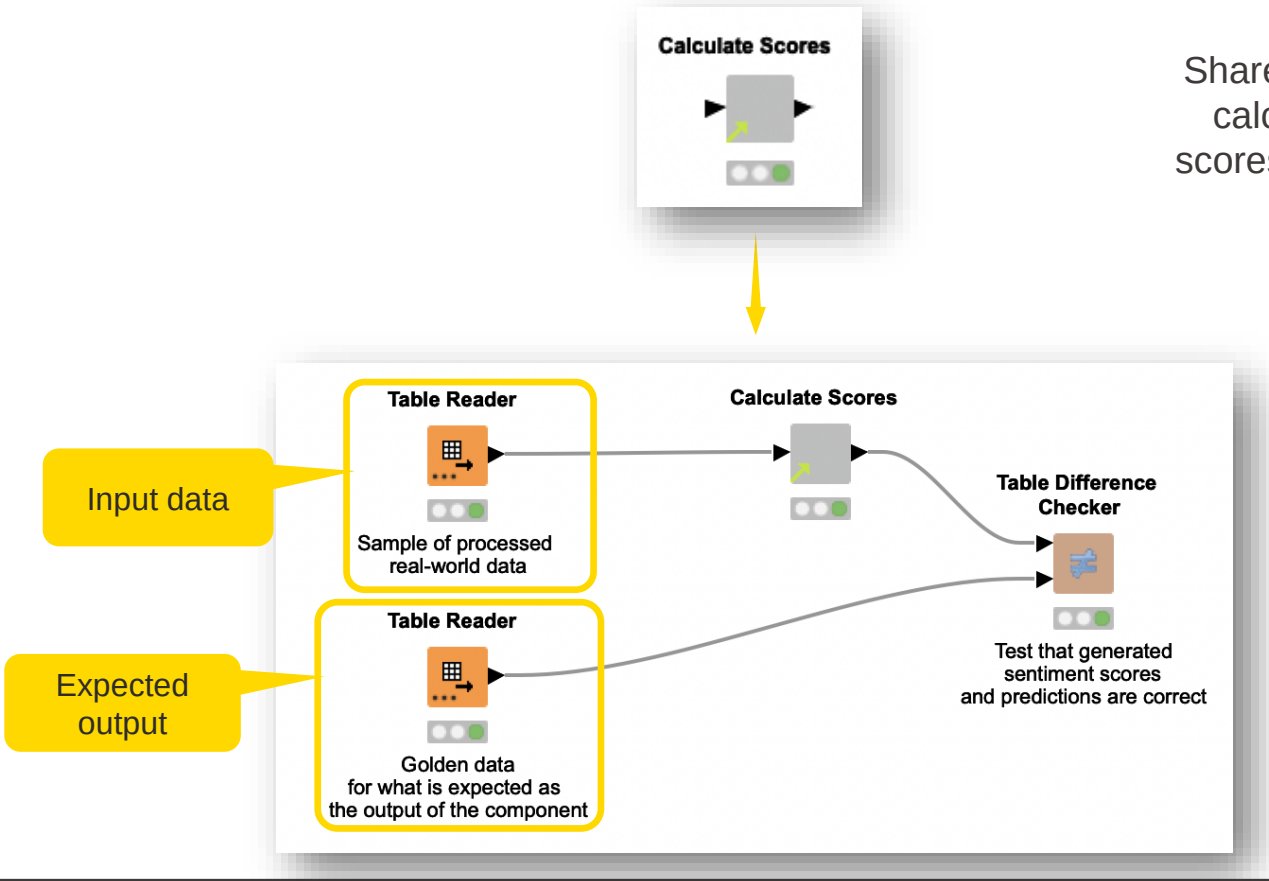


Component that calculates sentiment scores of text documents

Shared Component



Testing a Component

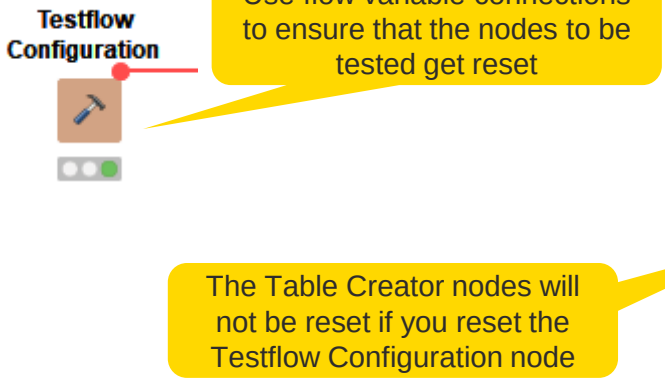


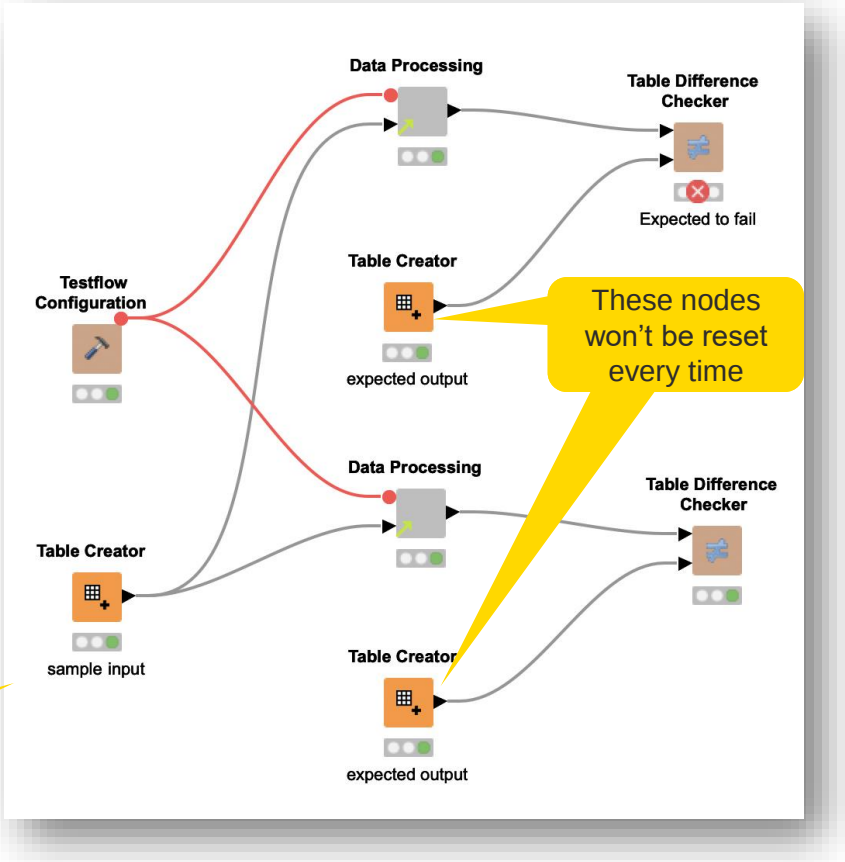
Shared component that calculates sentiment scores of text documents

Testflow that checks If component calculates sentiment scores correctly

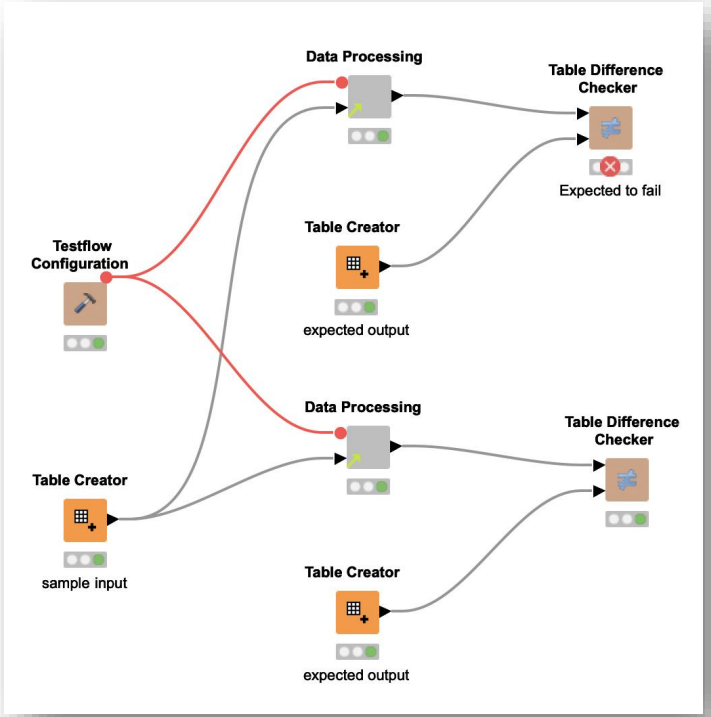
Executing Multiple Tests in a Single Testflow

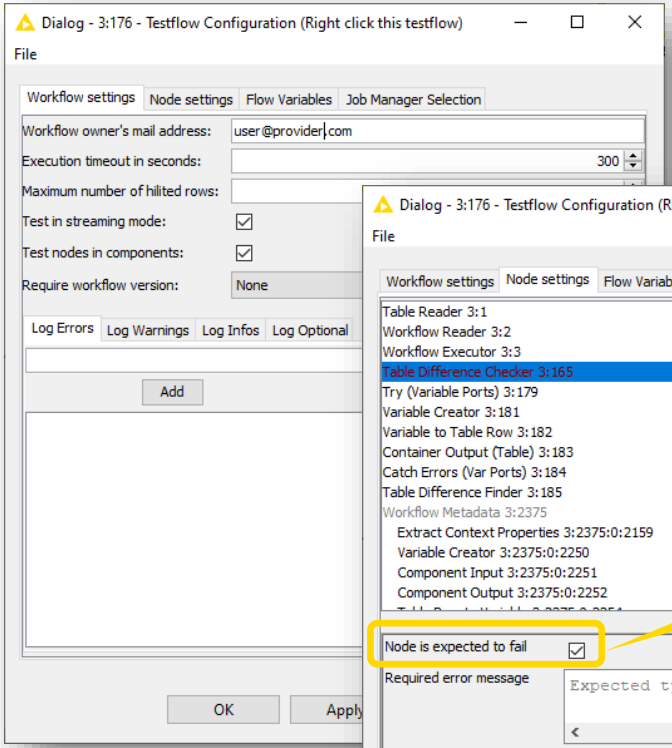
- The Testflow Configuration node can be especially useful here
  - It defines which nodes are expected to fail or succeed

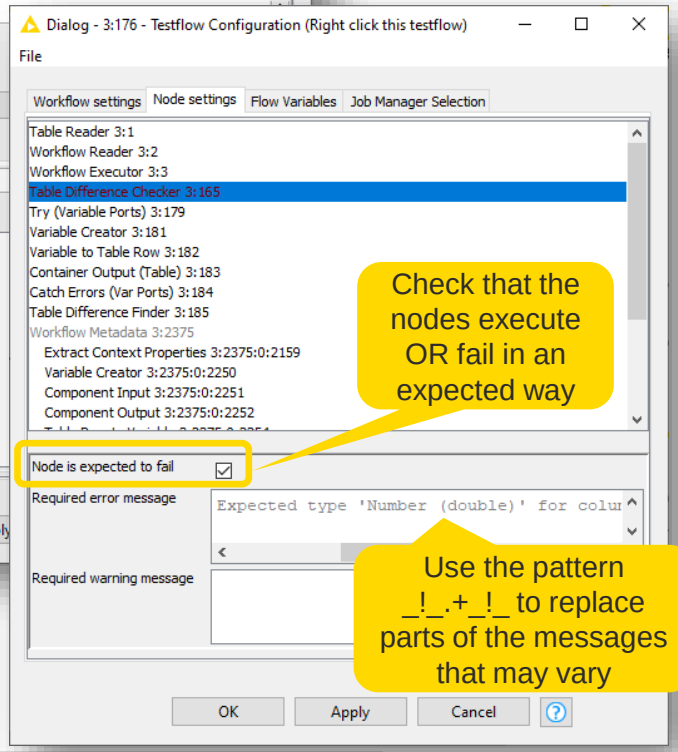




Testflow Configuration Node

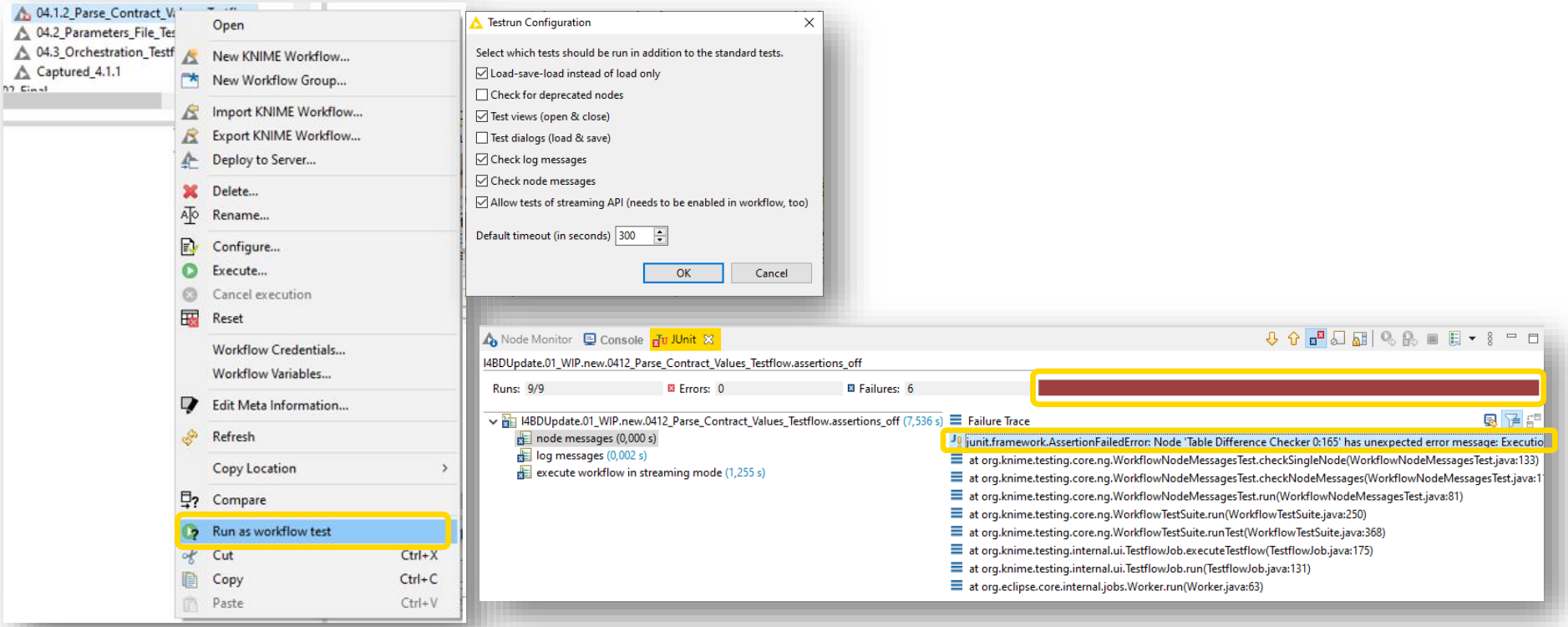






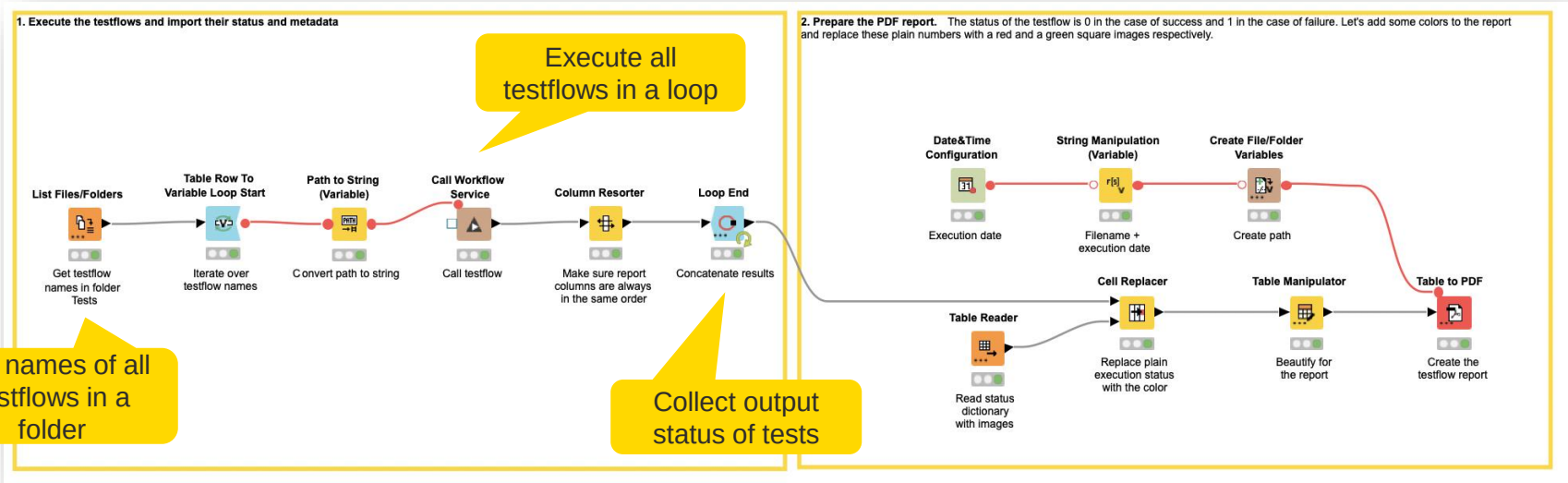
KNIME Testing Framework – JUnit

- Browse the results manually
- Audit failing nodes or unexpected behavior as well as the failure trace

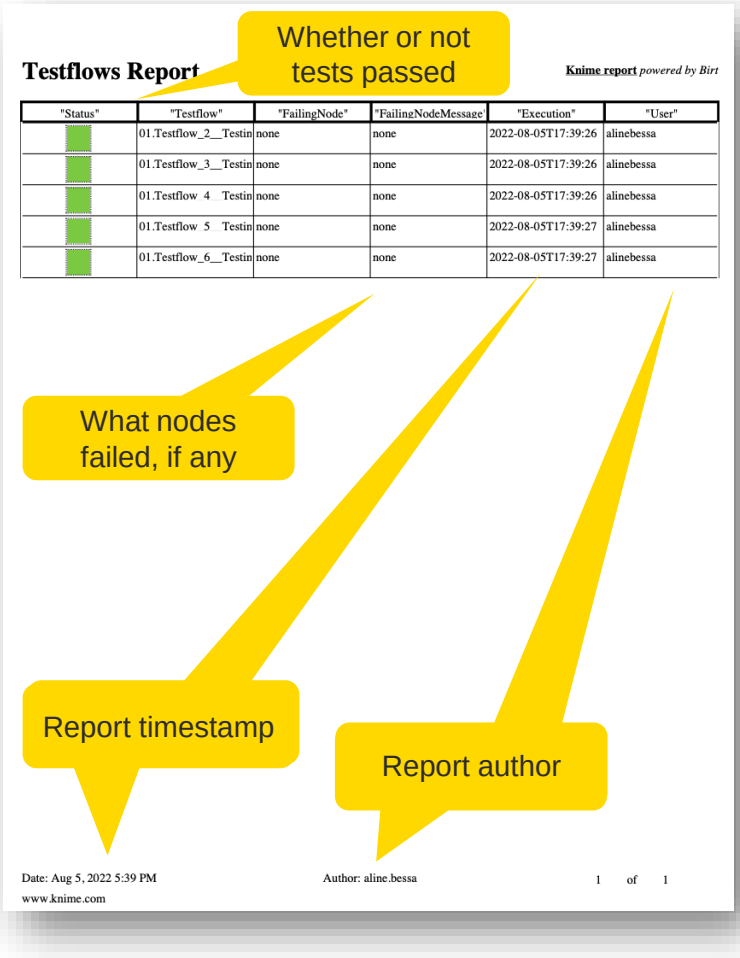


Executing Multiple Testflows Automatically

- Call all the testflows and collect their results
  - Make sure failing nodes in the testflows are captured in Try&Catch nodes
- Merge the testflows' metadata into one report, e.g., pdf

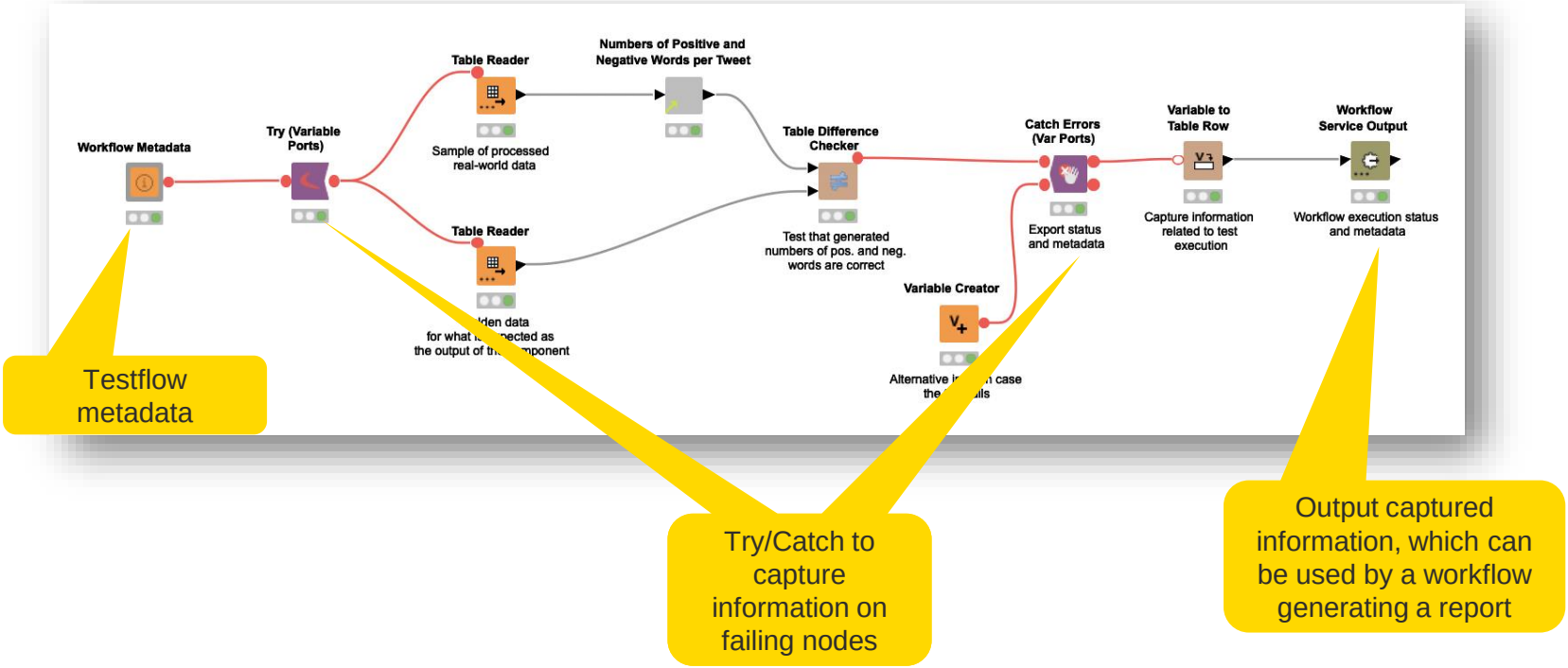


Executing Multiple Testflows Automatically: Report

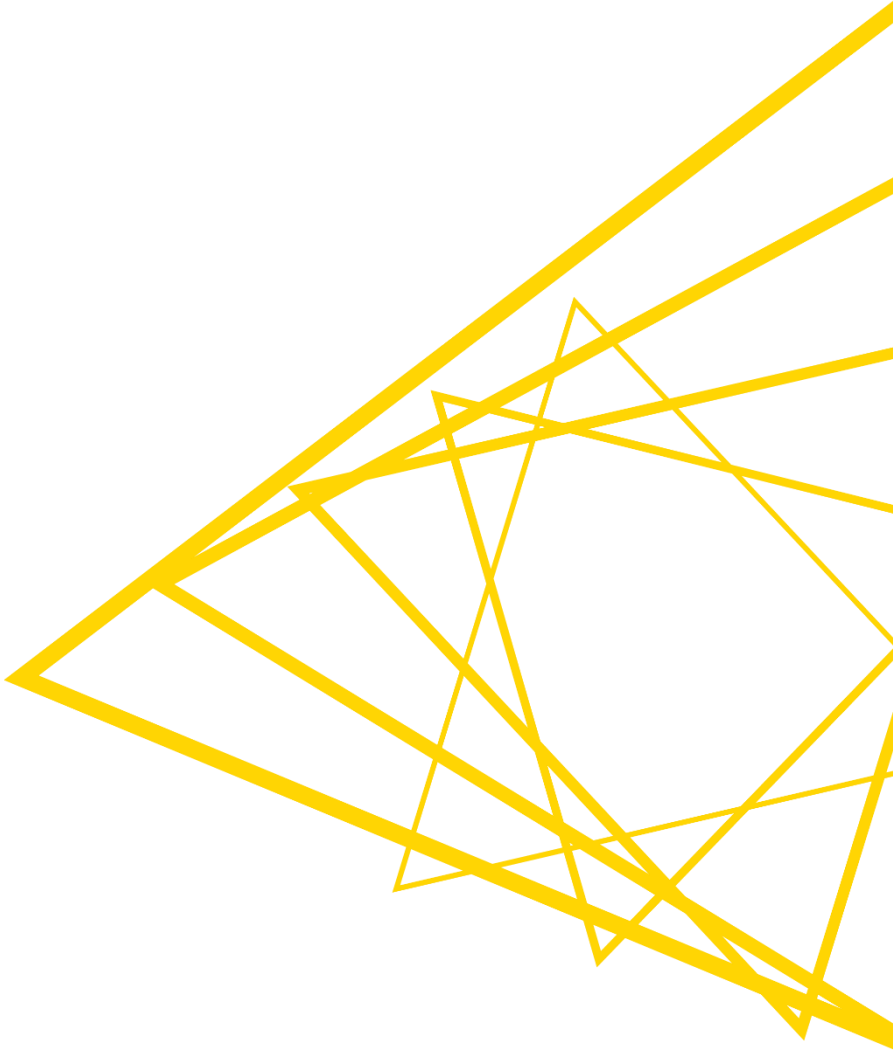


Example of Testflow for a Report

- Unit test with additional structure for a report



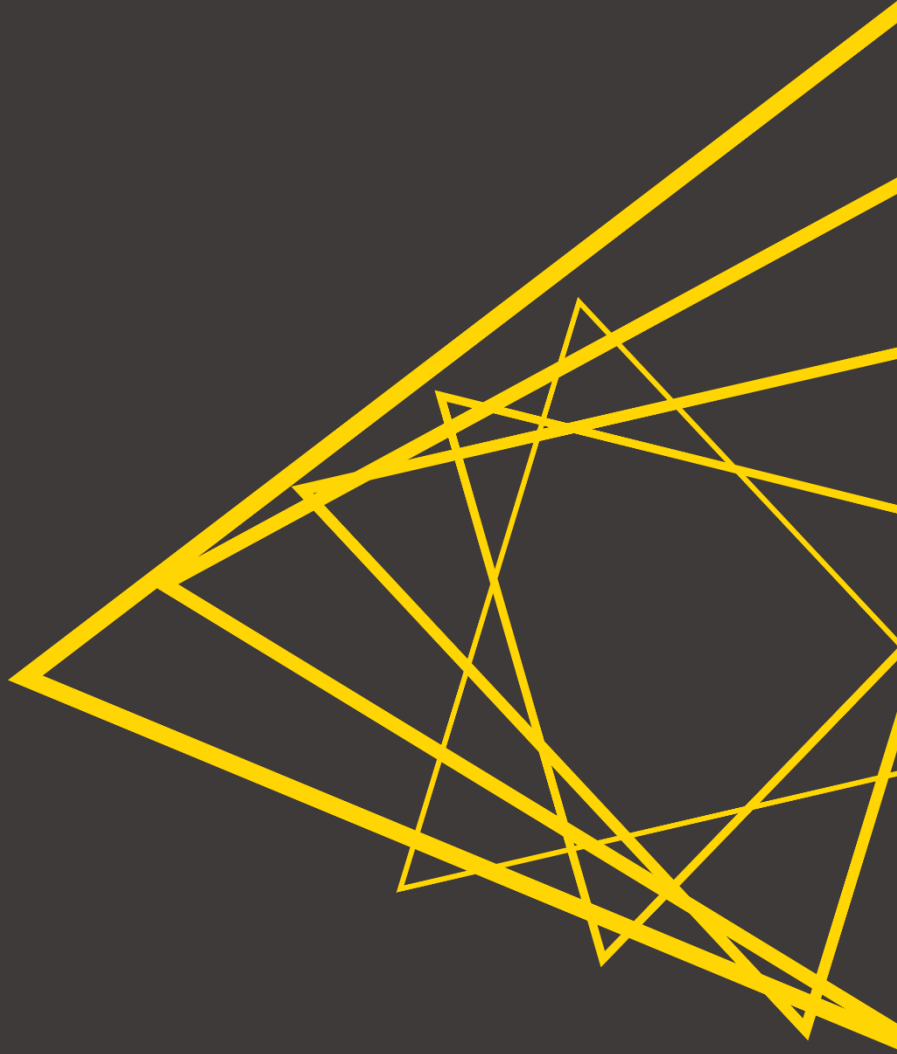
Demo



Best Practices for Testing

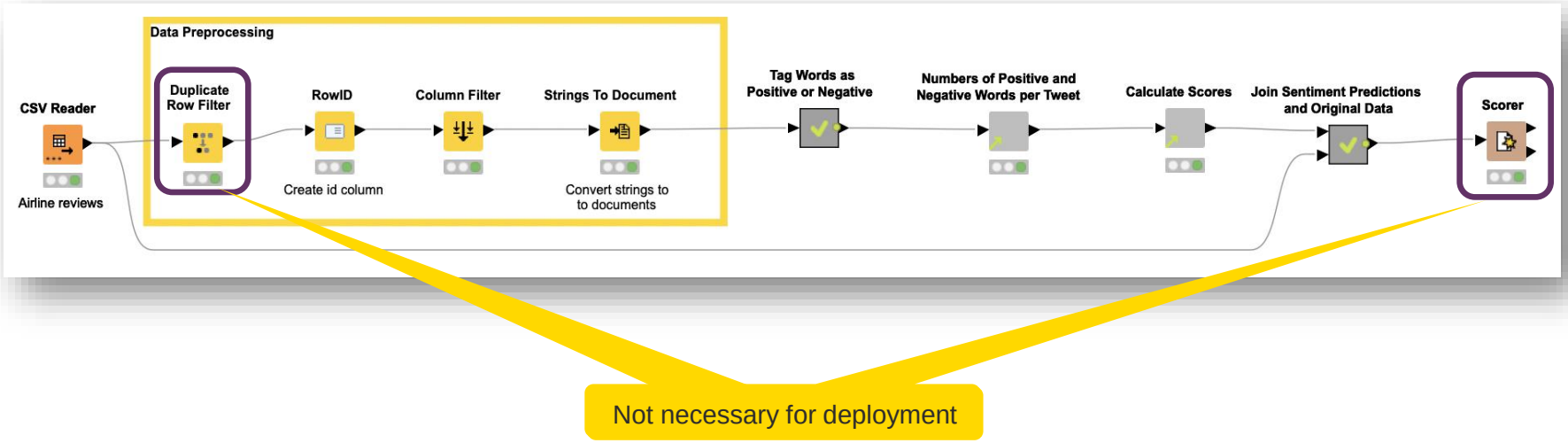
- Keep testflows under a single folder, facilitating their batch execution
  - You can have more than one folder for different types of tests
- A testflow should test one single entity (e.g., a component, a workflow segment)
  - It can have, however, many unit tests for the same single entity
- Create testflows that output useful information
  - Fail with custom error messages when input validation fails or in other expected scenarios
- Test regularly
- Components should be testable independently on the workflows that are using them

Deploying a Workflow



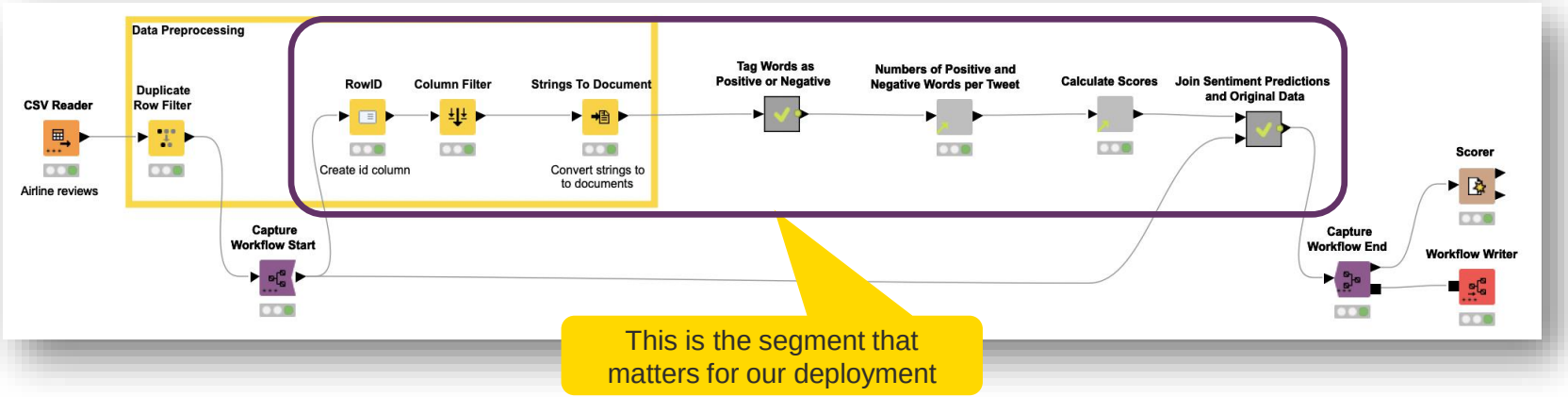
Automatically Creating a Workflow for Deployment

- Lexicon-based sentiment analysis prediction
- Assumptions
  - Once deployed, only one text comes at a time (no need for deduplication)
  - No ground-truth sentiment is given, so performance scoring is not viable



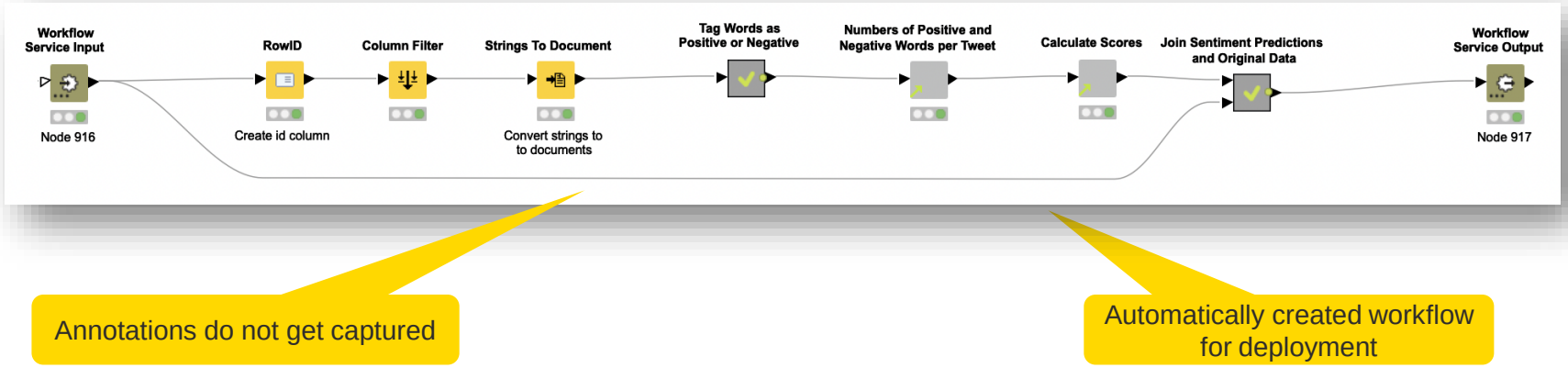
Automatically Creating a Workflow for Deployment

- Let’s now use Integrated Deployment to create a workflow for deployment



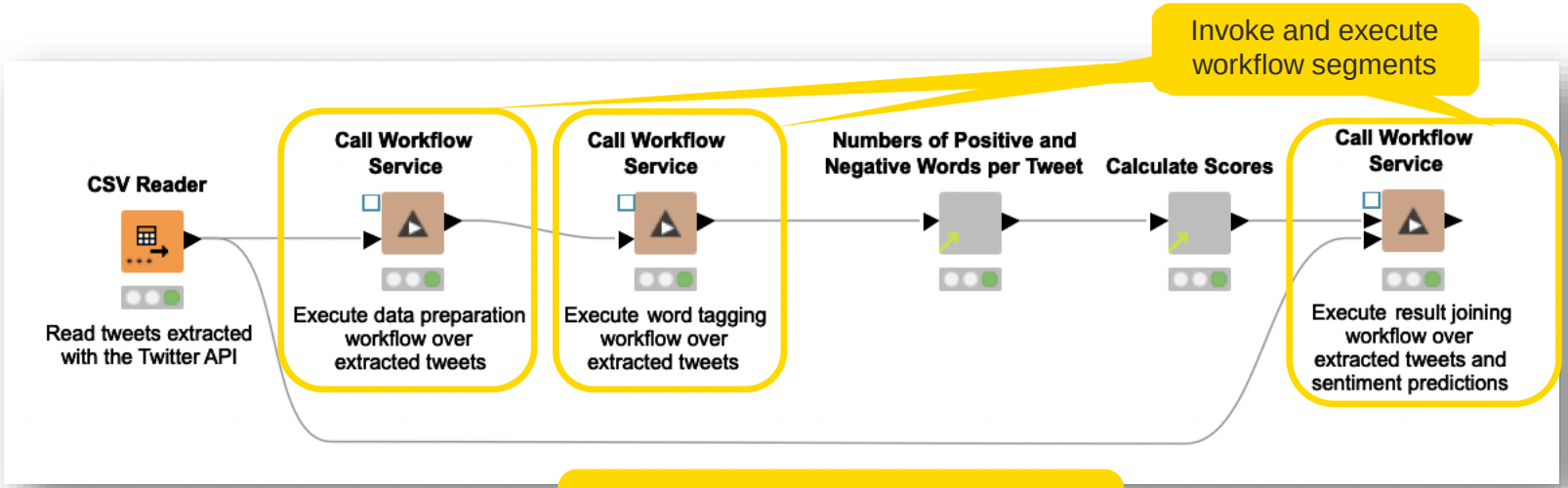
Automatically Creating a Workflow for Deployment

- Let’s now use Integrated Deployment to create a workflow for deployment

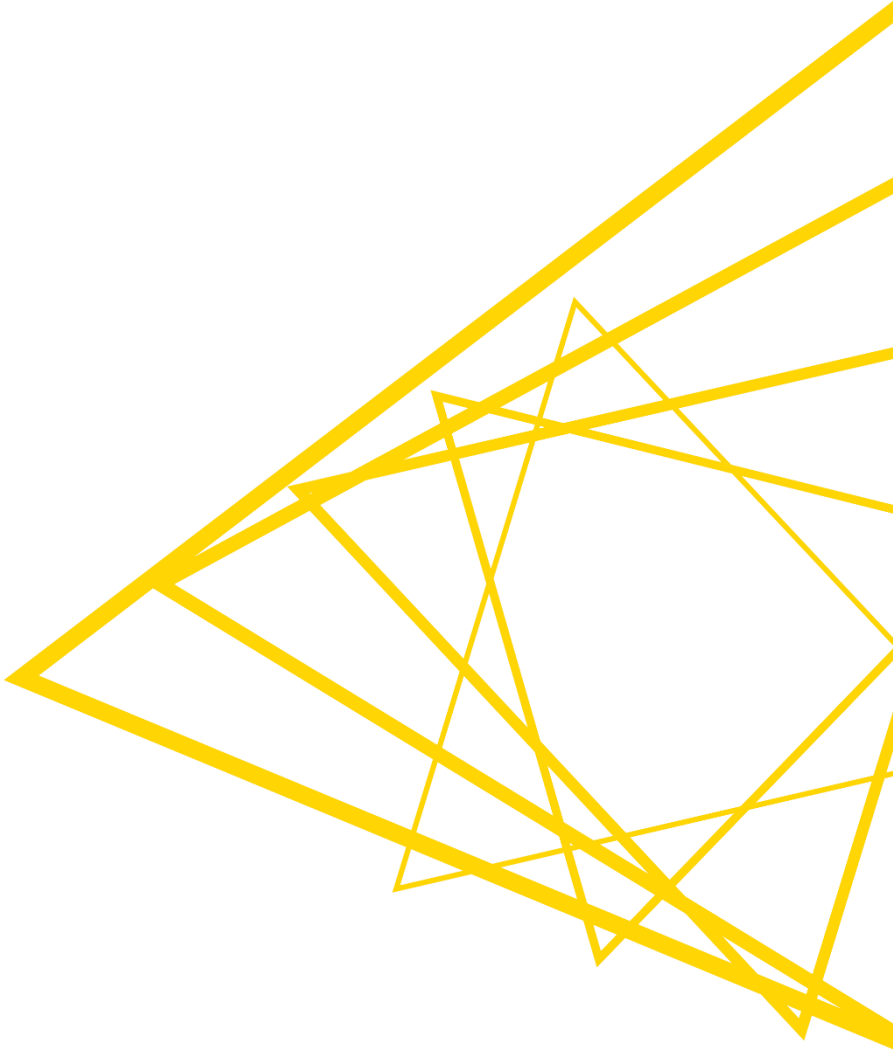


Manually Creating a Workflow for Production

- If you captured workflow segments for testing, e.g.:
  - Segment for data preprocessing
  - Metanode for word tagging
  - Metanode for joining original data with sentiment predictions
- ...You can leverage them and manually create a workflow for deployment

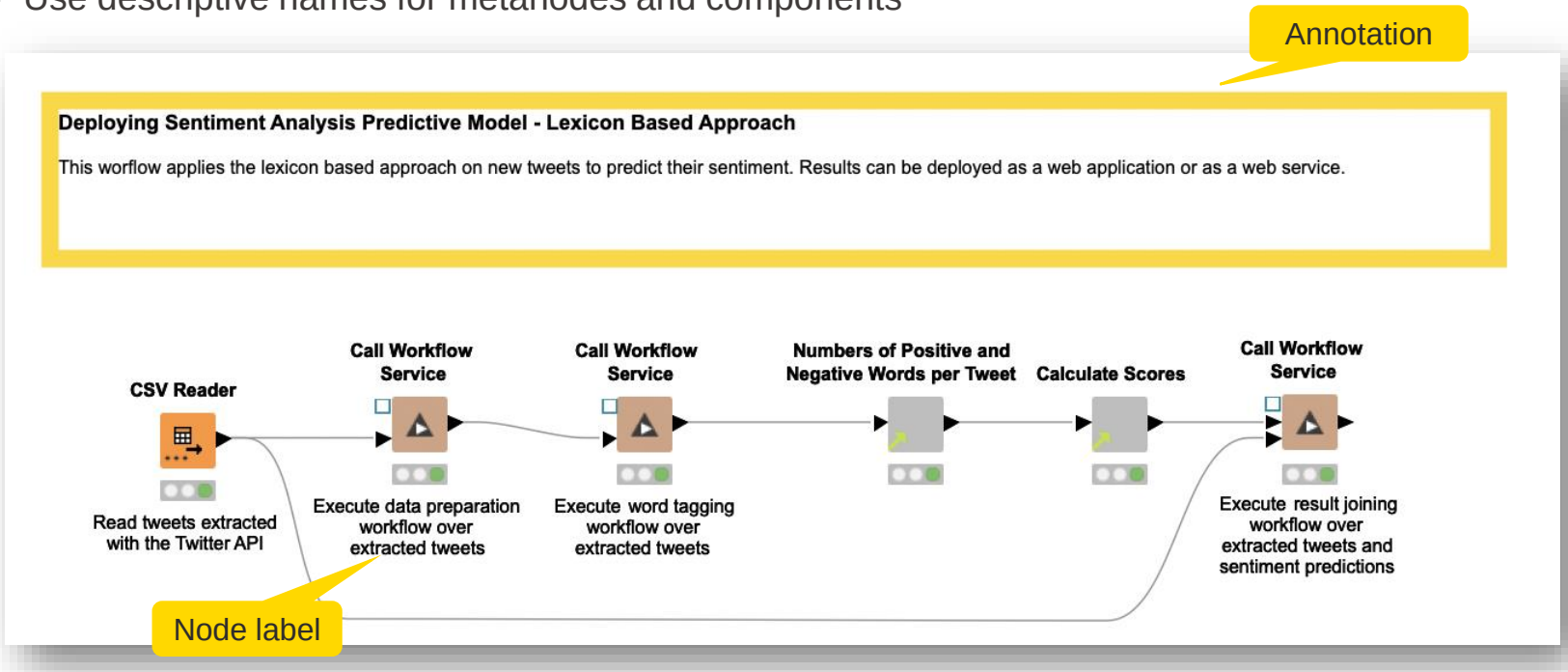


Demo



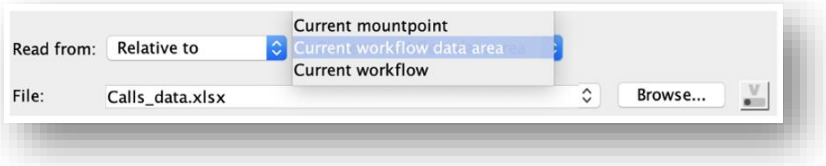
Best Practices for Deploying Workflows

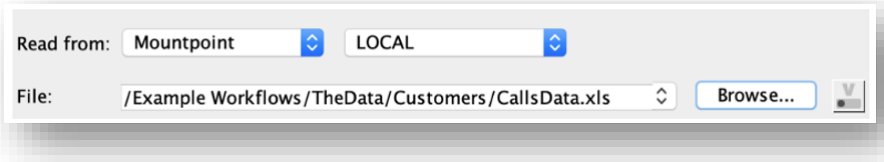
- Document your workflows properly
  - Add annotations to give an overview of what the workflow does
  - Label nodes to highlight what they are used for
  - Use descriptive names for metanodes and components

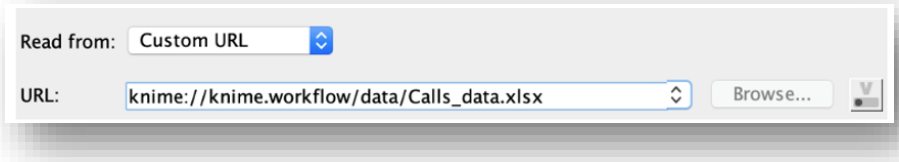


Best Practices for Deploying Workflows

- Do not use local file path types
  - Explicit references to files in your local machine break when the workflow is deployed to KNIME Server
  - Use relative, mountpoint, custom_url, or connected file path types instead

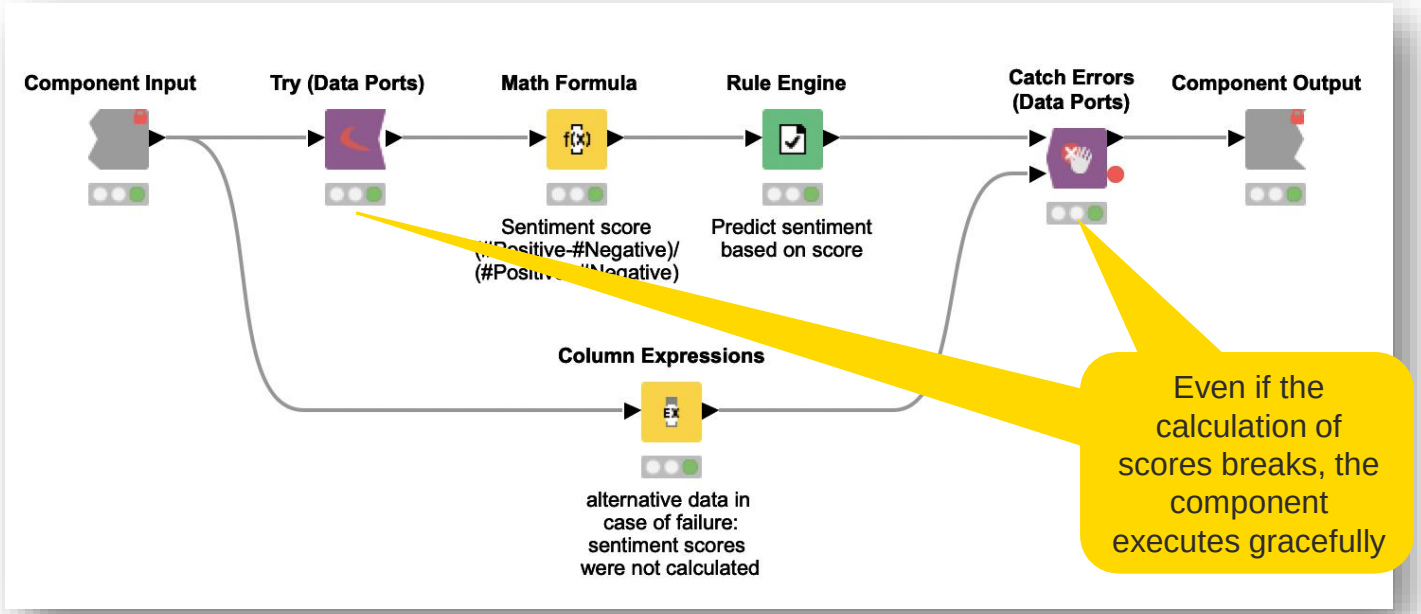






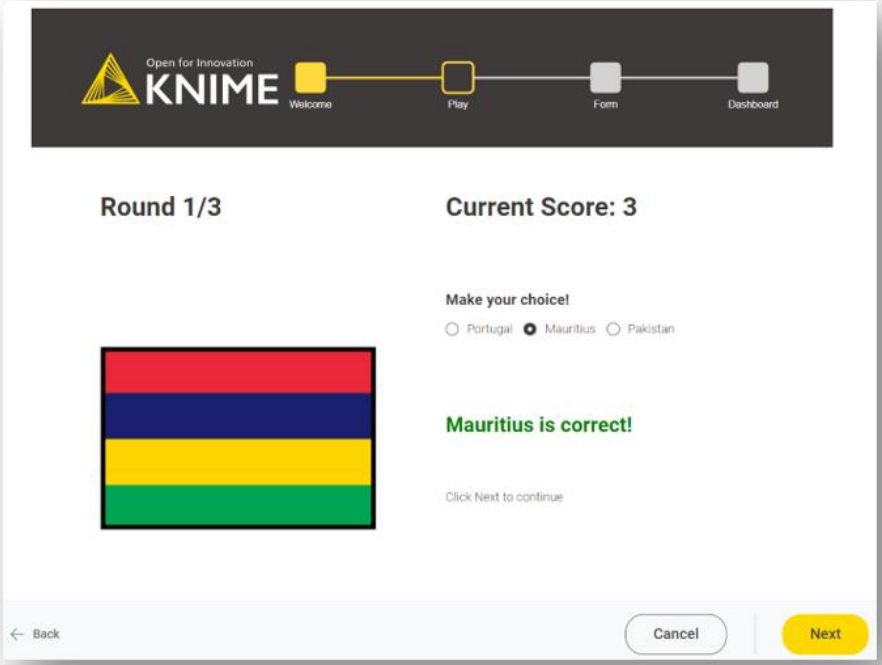
Best Practices for Deploying Workflows

- Make sure that the key functionalities of your workflow work as expected or fail gracefully
  - Test a functionality before deployment to avoid having a broken workflow in production
  - Add Try/Catch nodes to prevent execution problems in production



Deploying a Workflow as a Web Application

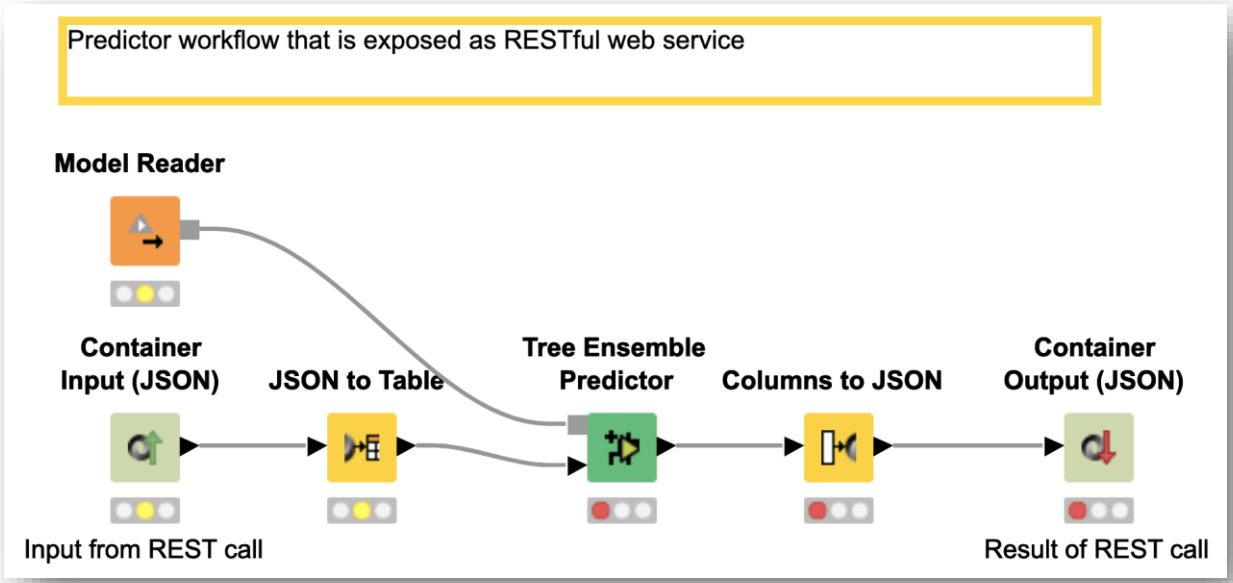
- The workflow is deployed as an application with a UI that can be accessed through a web browser running on the client's machine
- **Data apps in KNIME WebPortal** - focus of this course!



"Guess the Flag" game deployed as a web application on KNIME WebPortal

Deploying a Workflow as a Web Service

- Other workflows or software, which can be in other machines, can interact with your workflow
- Deployment in **KNIME Server or KNIME Business Hub**

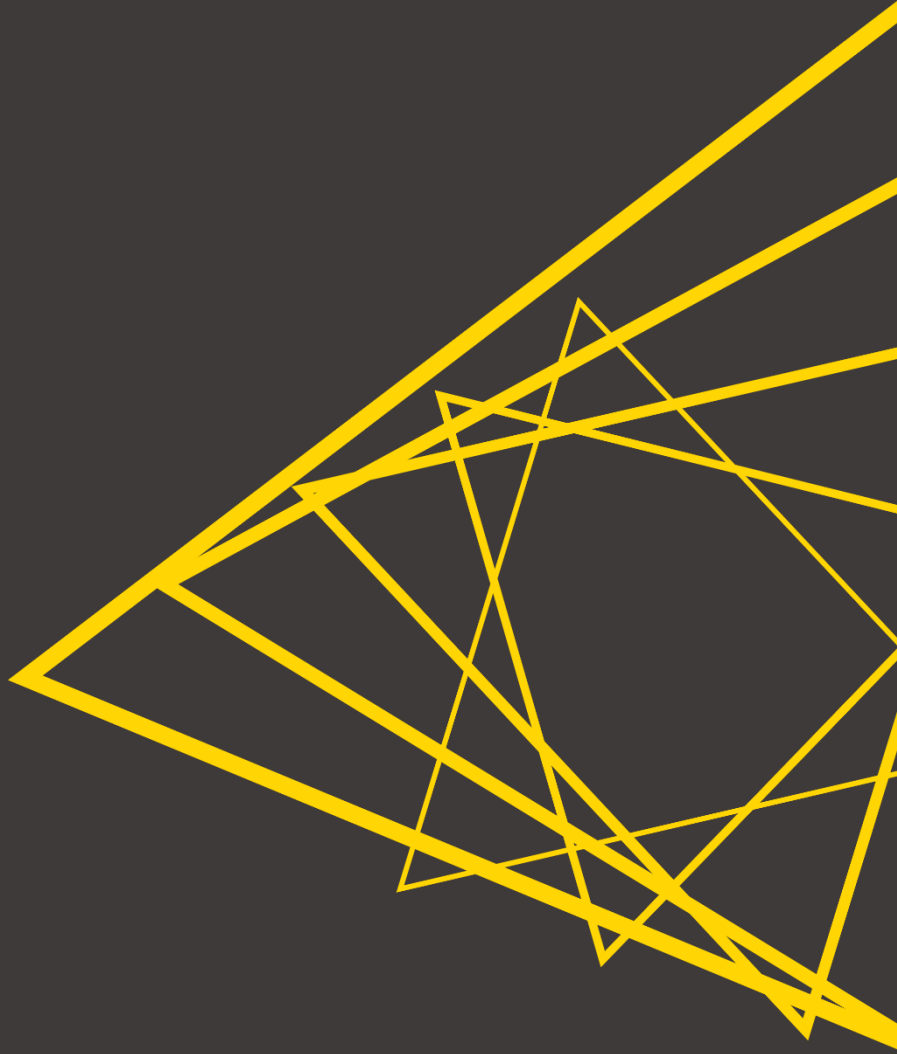


Session 1: Summary

After this session, you should be able to:

- Identify the different phases of the Data Science Life Cycle
- Understand the applications of Integrated Deployment and Workflow Services
- Identify differences and similarities between Integrated Deployment and Workflow Services
- List and use the tools to test Workflows, Components, and Workflow Segments
- Identify the necessary steps to deploy a workflow

Exercises



Exercises - Session 1

- **Exercise 01 -** Use integrated deployment techniques to capture segments of a workflow that builds a lexicon-based sentiment analysis predictor
- **Exercise 02 -** Create a unit test for a component that calculates sentiment scores
- **Exercise 03 -** Automate the execution of different testflows and concatenate their results into a test report
- **Exercise 04 -** Use workflow services to invoke workflow segments and create a workflow for deployment

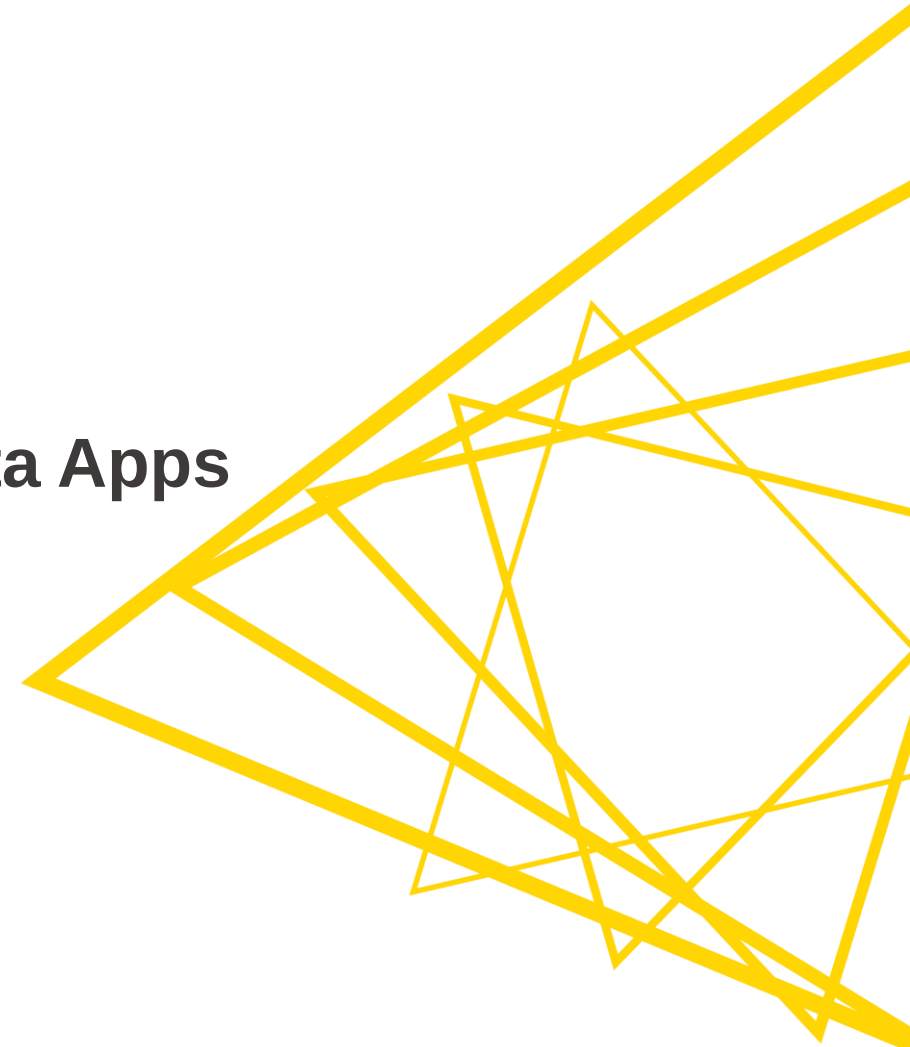
Open for Innovation
KNIME

Thank You!

Open for Innovation
KNIME

[L3-WP] Productionizing Data Apps

KNIME GmbH



Structure of the Course

Session	Topic	Duration
Session 1	What happens after the prototype is ready for deployment?	75 min
Session 2	**Introduction to KNIME Server**	75 min
Session 3	Deploying WebPortal Data Apps	75 min
Session 4	Performance optimization, orchestration, error handling, and KNIME Edge	75 min
Session 5	Wrap-up Session	15 min

Structure of each session

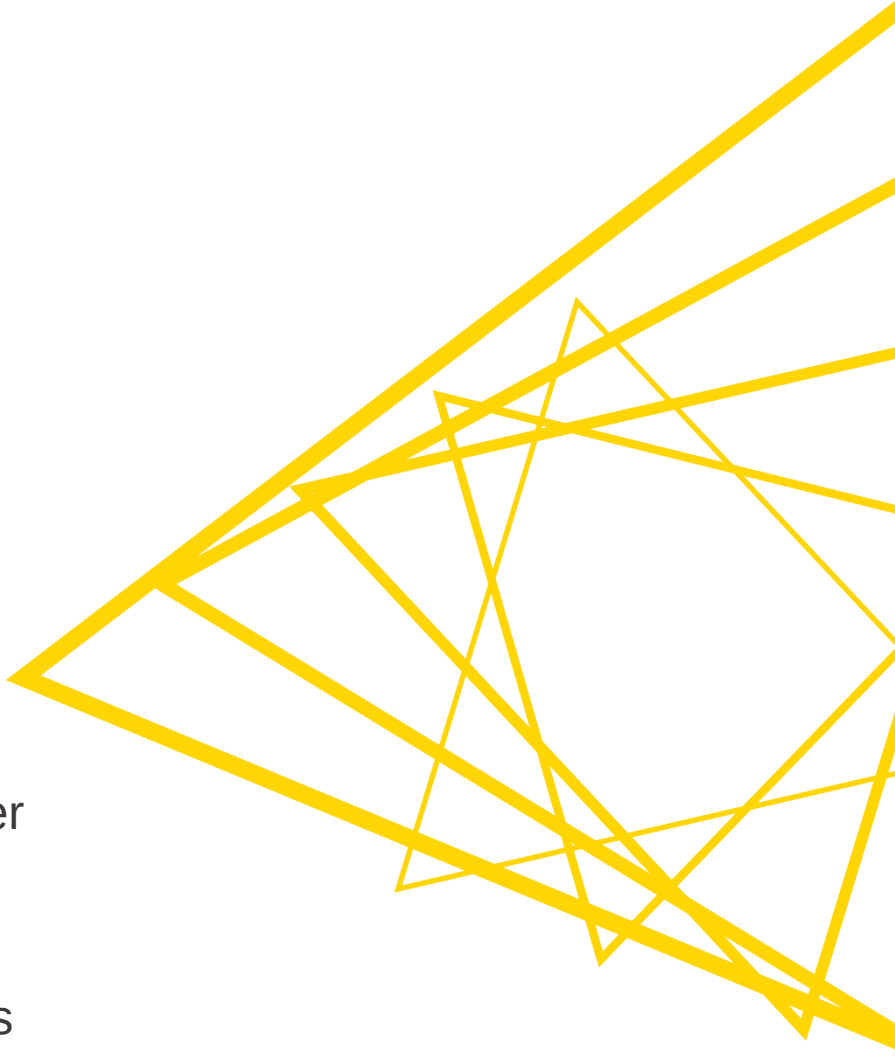
- Discussion of past exercises
- Course
- Introduction of next exercises

Exercises - Session 1

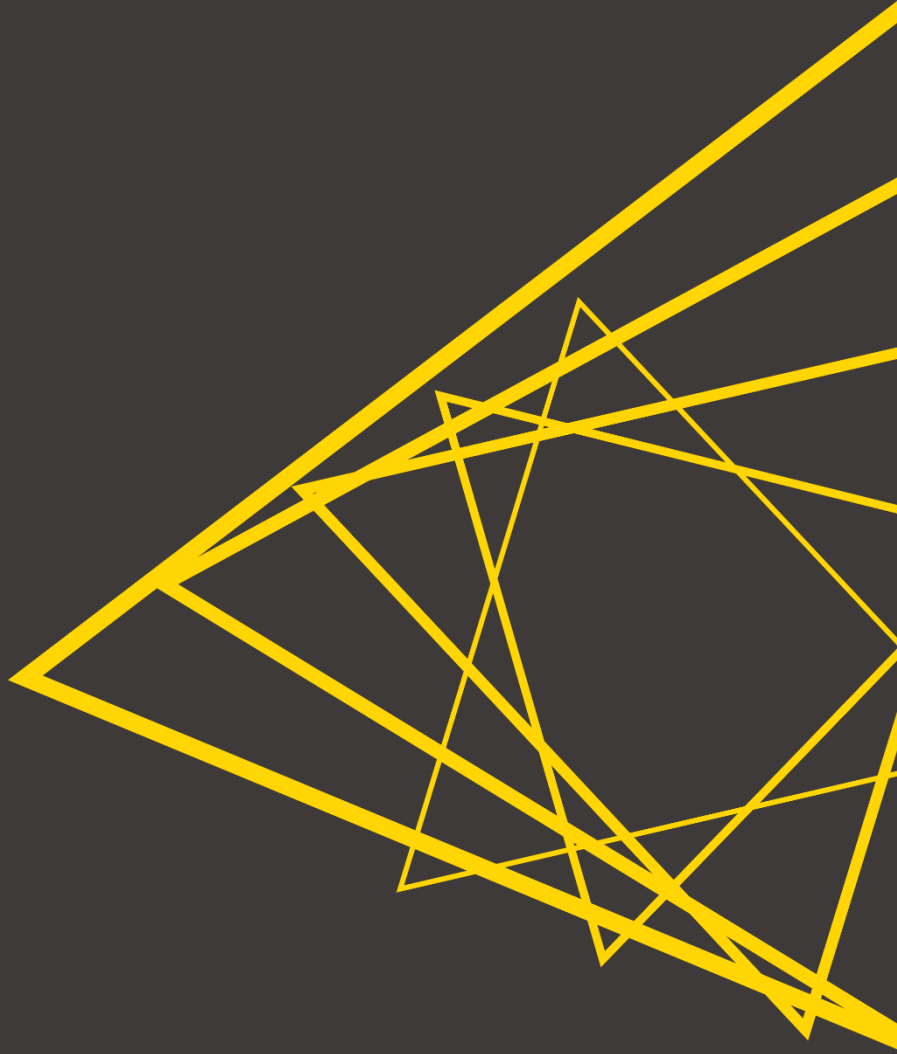
- **Exercise 01 -** Use integrated deployment techniques to capture segments of a workflow that builds a lexicon-based sentiment analysis predictor
- **Exercise 02 -** Create a unit test for a component that calculates sentiment scores
- **Exercise 03 -** Automate the execution of different testflows and concatenate their results into a test report
- **Exercise 04 -** Use workflow services to invoke workflow segments and create a workflow for deployment

Learning Objectives

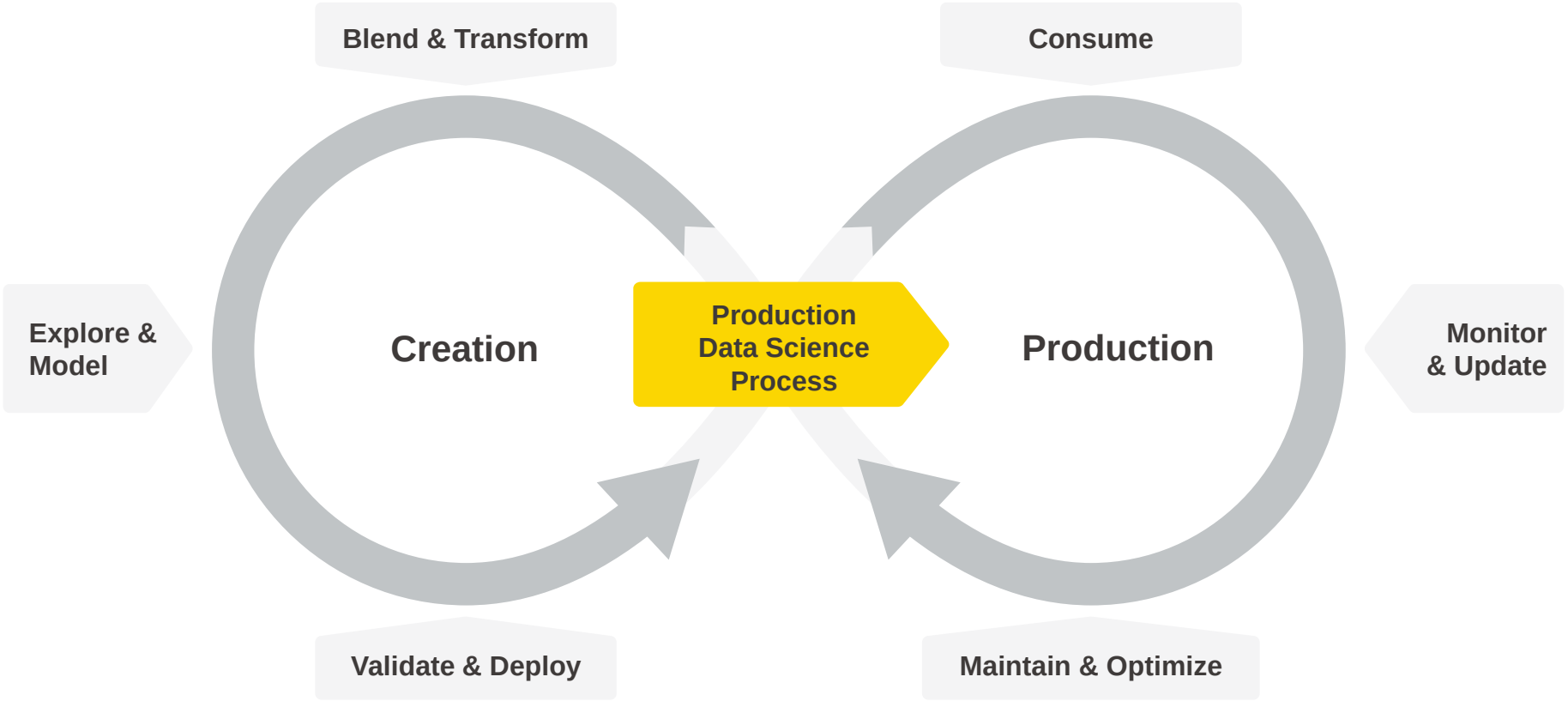
1. Identify the different KNIME Server use cases and stakeholders
2. Understand the steps to connect and deploy to KNIME Server
3. Understand how to perform workflow execution and scheduling in KNIME Server
4. Edit workflows with the Remote Workflow Editor
5. Modify workflow and directory permissions and version them



KNIME Server in a Nutshell



Data Science for the Business: Creation & Production



KNIME Software - One Ecosystem

Create

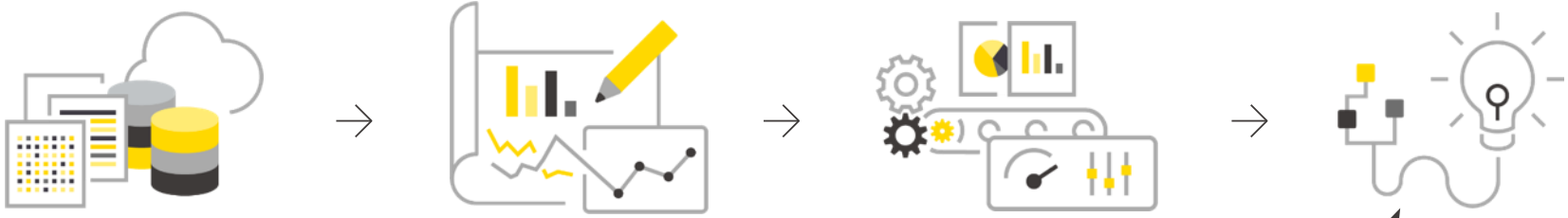
Productionize

Gather & Wrangle

Model & Visualize

Deploy & Manage

Consume & Optimize



Open Source
KNIME Analytics Platform

- KNIME Extensions
- KNIME Integrations
- Community Extensions
- Partner Extensions

Commercial
KNIME Server

To become KNIME Business Hub

- KNIME WebPortal
- Data Science as a Service

Collaboration Among Multiple Stakeholders

Data Engineers
Data Science “coders” (Python, etc.)
Data Science Specialists
Data Science Visual workflow / generalists

Smart Business Users (more than Excel)
Application Users – Interaction required
Application Users – Made to spec
Report Consumers

Model / ML Operations
Feeding production systems

Operations Consumption
(Applications, Systems, Edge, etc.)

IT Operations
- Centralized resources / strategies
- Standards and preferred platforms used, infrastructure options
- Exit strategies

IT Security
- Data, applications

Financial / Risk Oversite
- Costs allocation

Compliance officer
Data/model Governance, traceability, GDPR

Data Science Practice: Teams!

Data Engineers
Data Science “coders” (Python, etc.)
Data Science Specialists
Data Science Visual workflow / generalists

Smart Business Users (more than Excel)
Application Users – Interaction required
Application Users – Made to spec
Report Consumers

Model / ML Operations
Feeding production systems

Operations Consumption
(Applications, Systems, Edge, etc.)

IT Operations
- Centralized resources / strategies
- Standards and preferred platforms used, infrastructure options
- Exit strategies

IT Security
- Data, applications

Financial / Risk Oversite
- Costs allocation

Compliance officer
Data/model Governance, traceability, GDPR

Collaboration across Team Members

Sharable / Reusable / Instantiated Workflows, Components and Collaborative Development

Database Specialists

Data Engineers

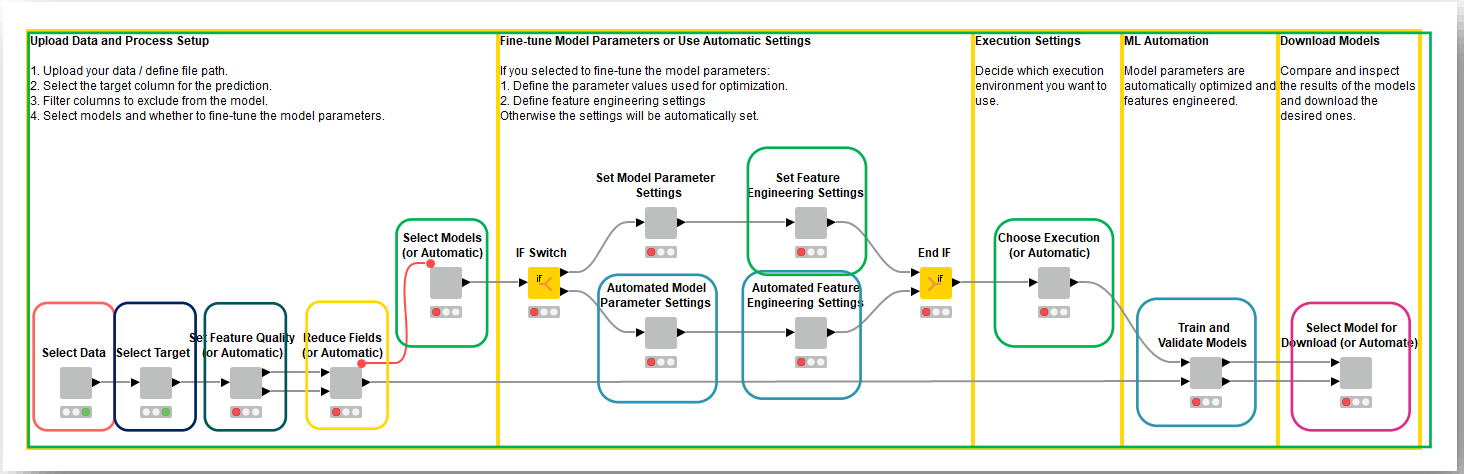
Python Specialists

Data Science Specialists

Visualization Specialists

Data Governance

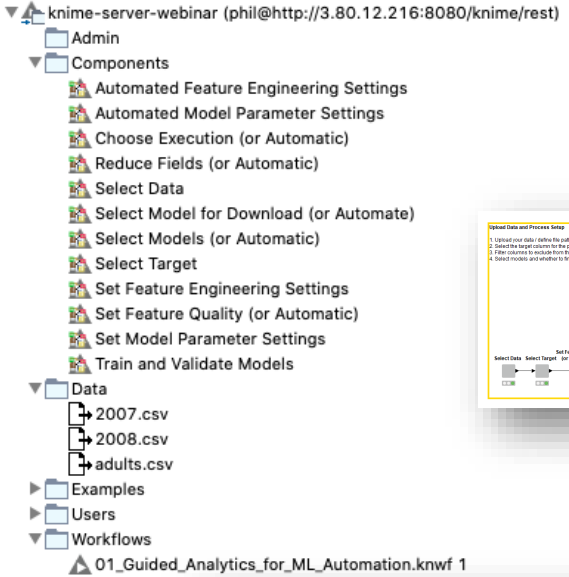
Data Science Generalists

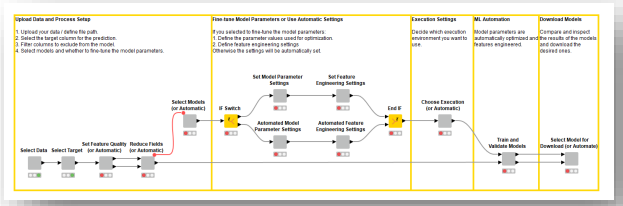


Open for Innovation KNIME

Recoverable Workflows, Backward Compatibility

Workflows, Components and Collaboration





Snapshots for /Workflows/01_Guided_Analytics_for_ML_Automation.knwf 1

Element/Creation date	Creator	Comment
Mon 2020-06-08, 09:36:42h	phil	Reworded label text of a few of the user interaction nodes
Mon 2020-06-08, 09:34:40h	phil	Tweeked parameter optimization settings.
Mon 2020-06-08, 09:33:37h	phil	Updated documentation

https://www.knime.com/blog/knime-meets-knime-will-they-blend

Features:

- Instantiated / Updated
- Versioned and snapshots
  - Version comparison
- Locked / Encrypted
- Reproducible
  - Guaranteed backward compatible

Data Science Practice: Multiple Stakeholders' Needs

Data Engineers
Data Science "coders" (Python, etc.)
Data Science Specialists
Data Science Visual workflow / generalists

Smart Business Users (more than Excel)
Application Users – Interaction required
Application Users – Made to spec
Report Consumers

Model / ML Operations
Feeding production systems

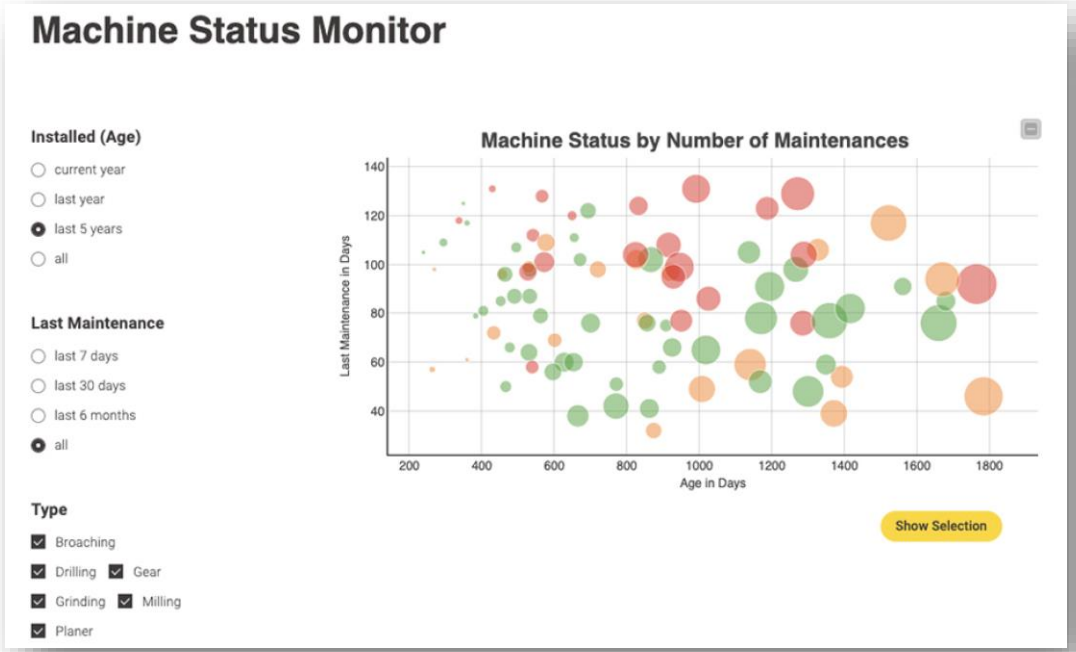
Operations Consumption
(Applications, Systems, Edge, etc.)

IT Operations
Centralized resources / strategies
Standards and preferred platforms used, infrastructure options
Exit strategies
IT Security
Data, applications

Financial / Risk Oversite
Costs allocation
Compliance officer
Data/model Governance, traceability, GDPR

Empower Business Users Appropriately

Delivering reports and output to business users appropriately



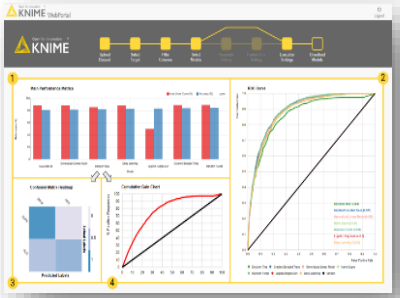
https://www.knime.com/data-apps

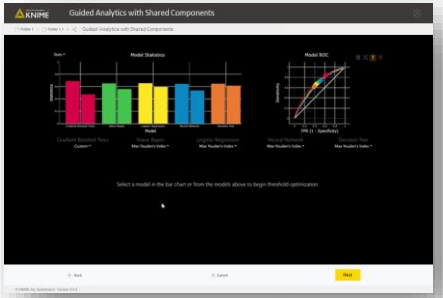
Features:

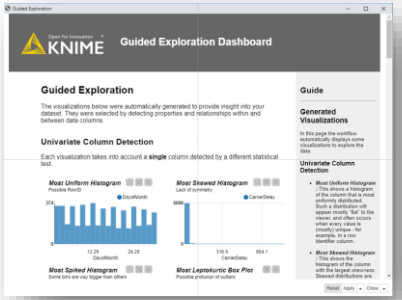
- Dashboards with KNIME Data Apps
- Reports creation with BIRT
- Integration with
  - Excel
    - Functionality exploitation, not just CSVs
  - PowerBI
  - Tableau
  - Qlik
  - Spotfire
  - …

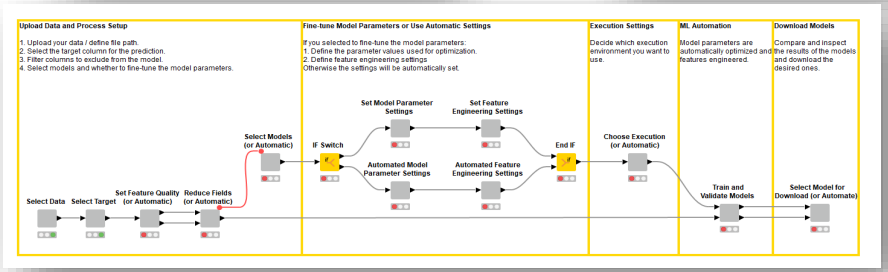
Empower Business Users Appropriately

- Guided Analytics for Building Applications
- Appropriate levels of Automation & Human Interaction
  - Depending on **task** and **audience**









Features:

- Workflows and WebPortal nodes build interactive applications & dashboards
- KNIME WebPortal manages access

Data Science Practice: Multiple Stakeholders' Needs

Data Engineers
Data Science "coders" (Python, etc.)
Data Science Specialists
Data Science Visual workflow / generalists

Smart Business Users (more than Excel)
Application Users – Interaction required
Application Users – Made to spec
Report Consumers

Model / ML Operations
Feeding production systems

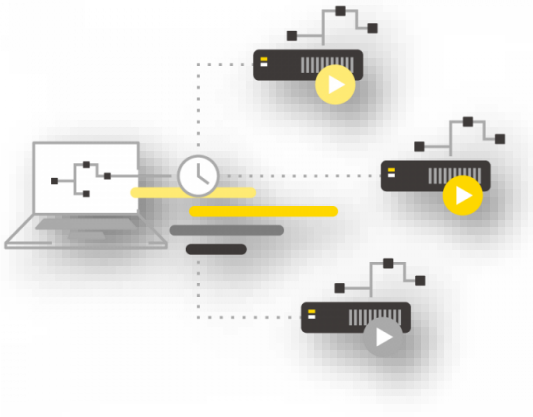
Operations Consumption
(Applications, Systems, Edge, etc.)

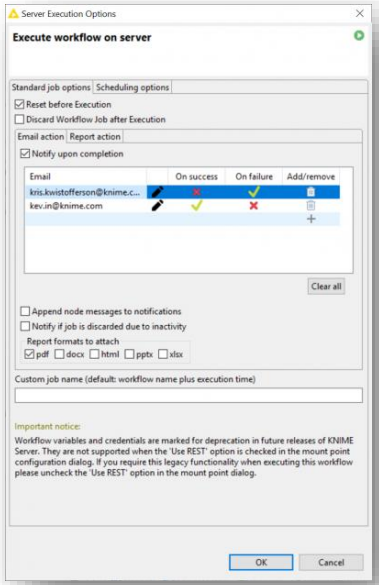
IT Operations
Centralized resources / strategies
Standards and preferred platforms used, infrastructure options
Exit strategies
IT Security
Data, applications

Financial / Risk Oversite
Costs allocation
Compliance officer
Data/model Governance, traceability, GDPR

Flexible Delivery Options: Automate

Automated workflow execution





Features:

- Scheduled
- Triggered
- Called (REST / SAAS)
- Call actions based on status
- View, edit, execute workflows remotely

https://docs.knime.com

Data Science Practice: Multiple Stakeholders' Needs

Data Engineers
Data Science "coders" (Python, etc.)
Data Science Specialists
Data Science Visual workflow / generalists

Smart Business Users (more than Excel)
Application Users – Interaction required
Application Users – Made to spec
Report Consumers

Model / ML Operations
Feeding production systems

Operations Consumption
(Applications, Systems, Edge, etc.)

IT Operations
- Centralized resources / strategies
- Standards and preferred platforms used, infrastructure options
- Exit strategies

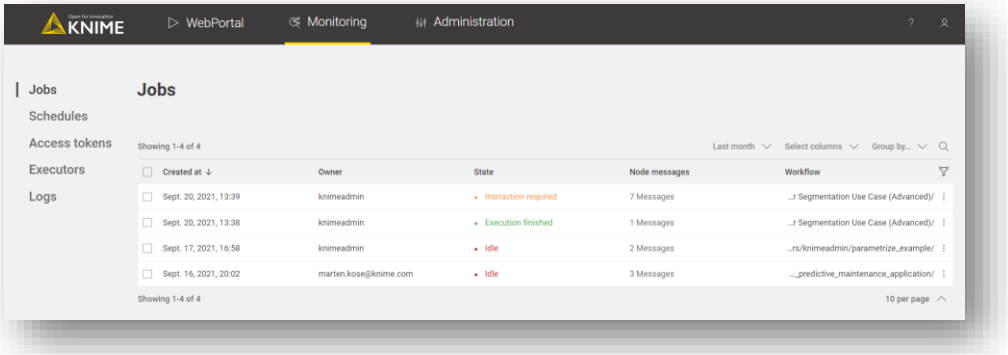
IT Security
- Data, applications

Financial / Risk Oversite
- Costs allocation

Compliance officer
Data/model Governance, traceability, GDPR

Manage Infrastructure & Users

Central Management and Monitoring capabilities

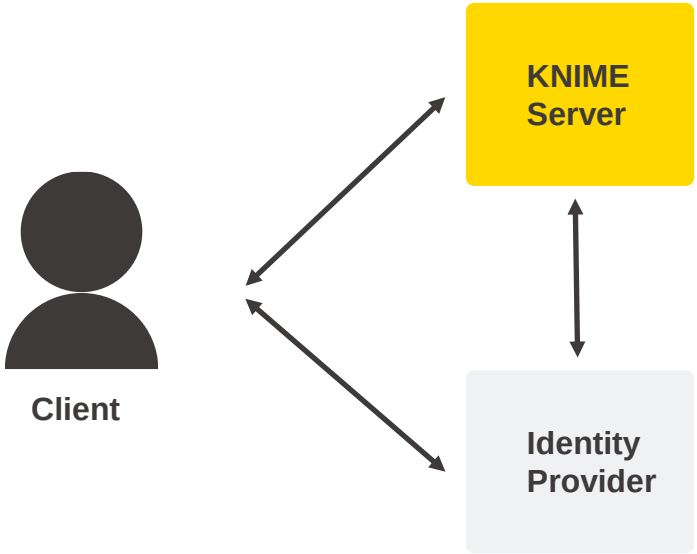


Features:

- Client Customizations
  - Custom update sites
  - Manage preferences via profiles
  - Node repository & libraries
- Monitor server activity
  - Running and scheduled jobs
  - Adjust permissions
  - Manage ongoing services

Single Sign-on, Integrate with multiple Security protocols

OAUTH, LDAP, AD Integration



Features:

- Single sign-on (SSO) to KNIME Server
- Integrate with multiple identity providers
- Flexible configuration capabilities to map users and groups
- Manage all aspects of KNIME usage

https://docs.knime.com

Data Science Practice: Multiple Stakeholders' Needs

Data Engineers
Data Science "coders" (Python, etc.)
Data Science Specialists
Data Science Visual workflow / generalists

Smart Business Users (more than Excel)
Application Users – Interaction required
Application Users – Made to spec
Report Consumers

Model / ML Operations
Feeding production systems

Operations Consumption
(Applications, Systems, Edge, etc.)

IT Operations
- Centralized resources / strategies
- Standards and preferred platforms used, infrastructure options
- Exit strategies

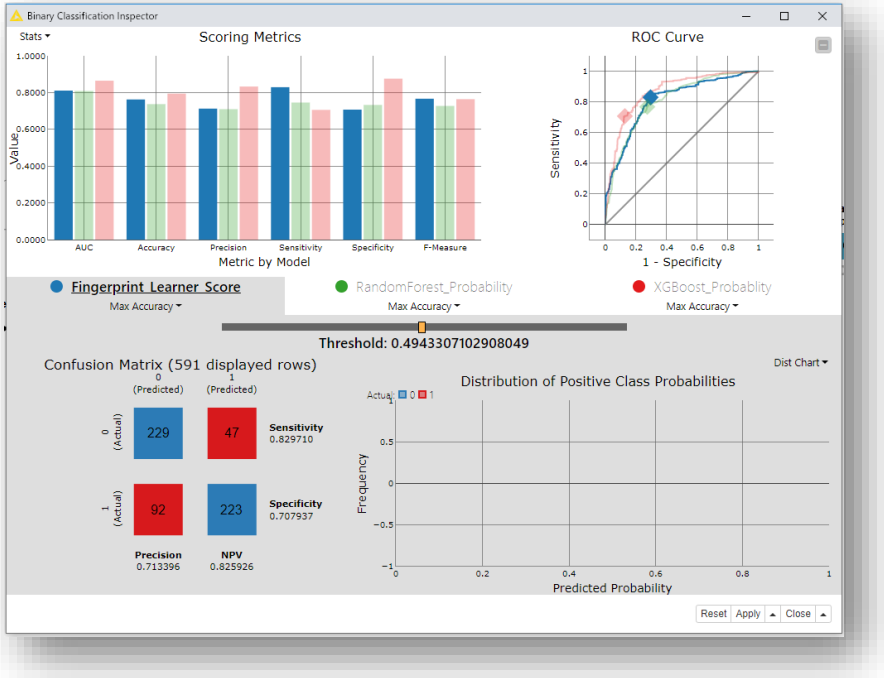
IT Security
- Data, applications

Financial / Risk Oversite
- Costs allocation

Compliance officer
Data/model Governance, traceability, GDPR

Governance & Compliance

Explainability / Interpretability of models



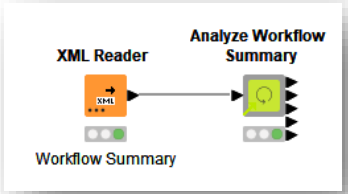
Features:

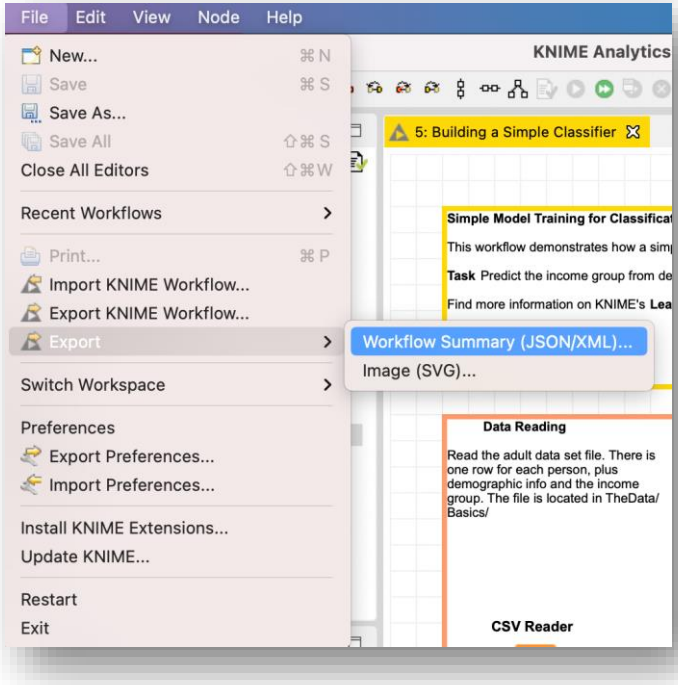
- Many Techniques available
  - LIME
  - SHAP
  - Shapley
  - Partial Dependence / ICE
  - Binary Classification Inspector
- [Model Interpretability Components]

https://hub.knime.com/knime/extensions/org.knime.features.mli/latest

Governance & Compliance

- Data or model lineage
  - The workflow summary is a detailed **structured** description of a workflow
    - Workflow structure, node & workflow annotations
    - Node configuration settings, ports, connections, execution info
    - Workflow metadata and its execution environment
    - Plugins Installed at the time of workflow execution
  - Exported in XML or JSON
    - Parse manually
    - Or use Analyze Workflow Summary component
  - After parsing
    - Data lineage
    - Custom log files / reports
    - Workflow audit
    - etc.





Feature Overview

- In this course you are going to learn about the following features:
  - How to create a KNIME Server **mountpoint** on KNIME Analytics Platform
  - How to **connect** to KNIME Server and **deploy** workflows
  - How to **execute** workflows in KNIME Server, including **scheduled executions**
  - How to **monitor** workflow executions on KNIME Server
  - How to **recover previous runs** from executed jobs on KNIME Server
  - How to **version** your workflows on KNIME Server
  - How to assign **permissions** to workflows and workflow groups on KNIME Server
  - How to deploy workflows as **data apps** to KNIME Server's WebPortal

KNIME Business Hub Teaser: Same Audience and Features

Data Engineers
Data Science "coders" (Python, etc.)
Data Science specialists
Data Science Visual workflow / generalists

Collaboration

Smart Business Users (more than Excel)
Application Users – Interaction required
Application Users – Made to spec
Report Consumers

Dashboards and reports

Model / ML Operations
Feeding production systems

Automated execution

Operations Consumption
(Applications, Systems, Edge, etc.)

IT Operations
- Centralized resource strategies
- Standards and preferred platforms used, infrastructure options
- Exit strategies

IT Security
- Data, applications

Management and monitoring

Financial / Risk Oversite
- Costs allocation

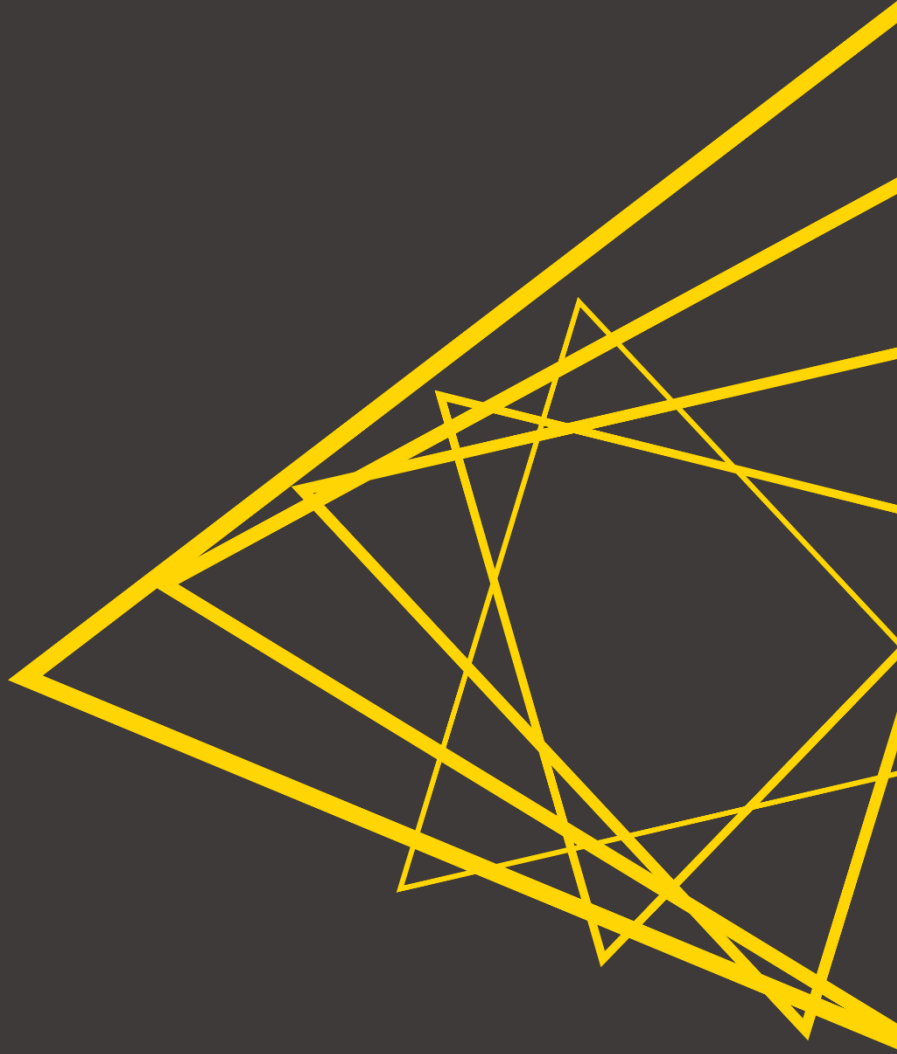
Compliance office
Data/model Governance, traceability, GDPR

Explainability and lineage

KNIME Business Hub: Feature Teasers

- In this course you are also going to see teasers for some KNIME Business Hub features:
  - How to create a KNIME Business Hub **mountpoint** on KNIME Analytics Platform
  - How to **connect** to KNIME Business Hub and **deploy** workflows
  - How to **execute** workflows in KNIME Business Hub
  - How to deploy workflows as **data apps**

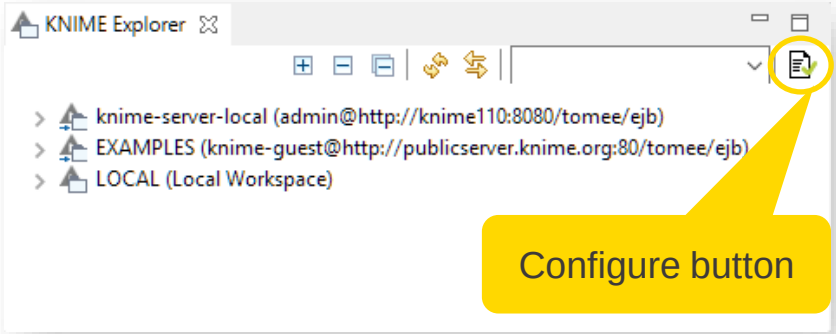
Connecting and Deploying to KNIME Server

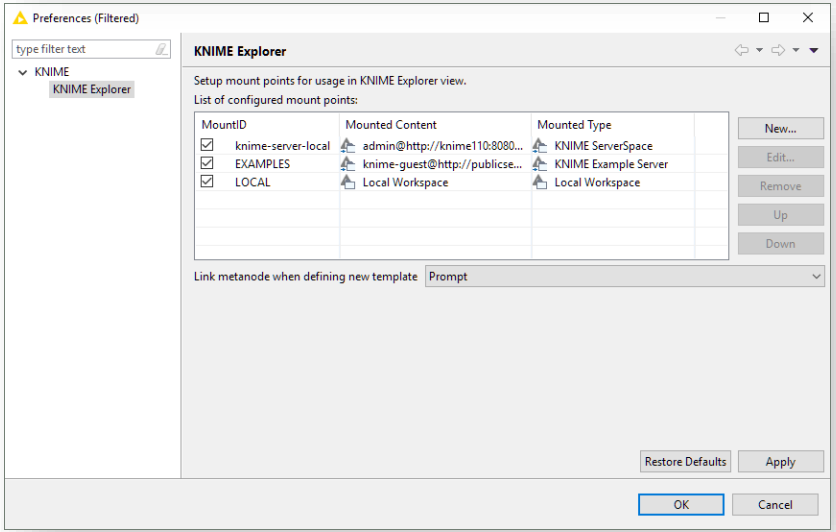


Set Up a New Mount Point

Server connections are shown as “mount points” in the KNIME Explorer. To add a new mount point simply:

1. Click the Configure button in the KNIME Explorer.
2. Click New…
3. Configure a mount point with your details

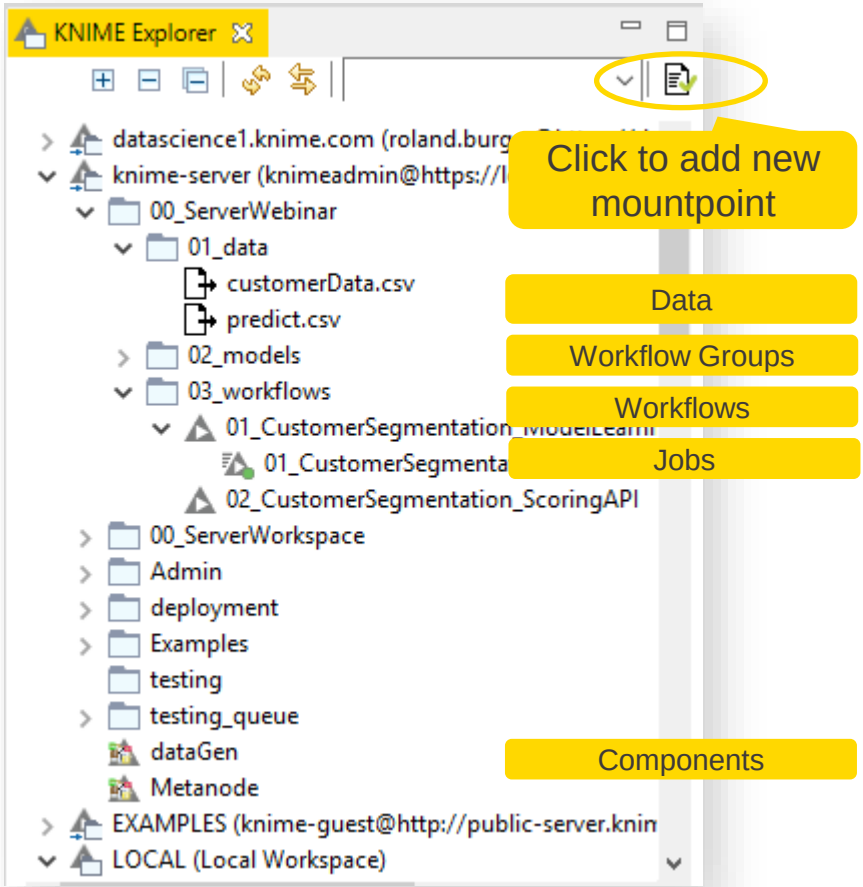




Server Mount Point as a Shared Repository (1/2)

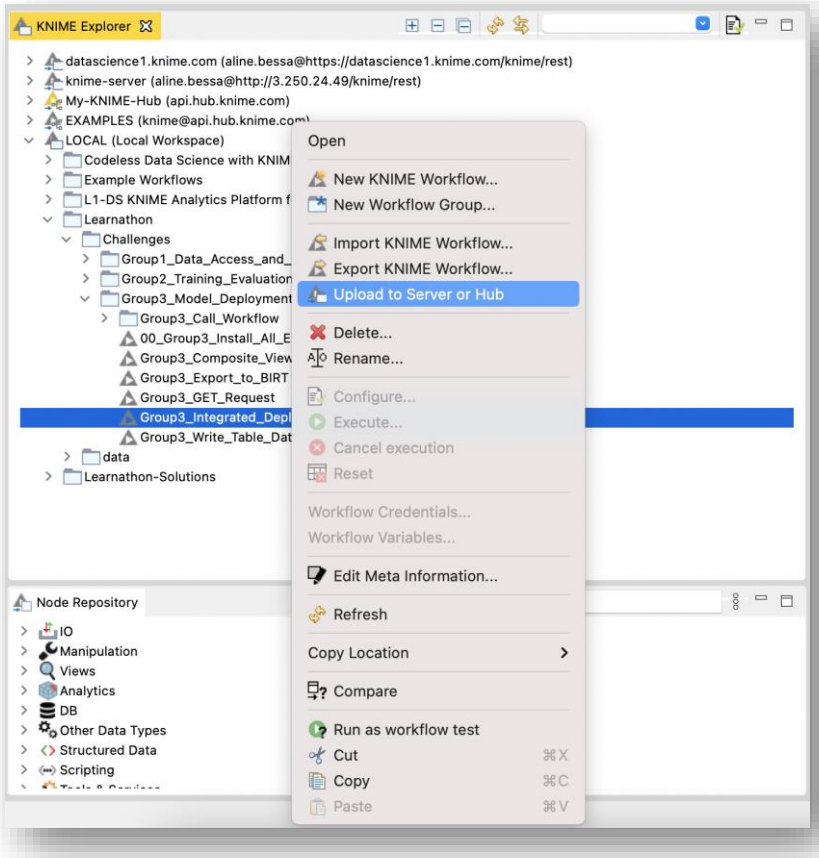
The Server provides an area in the Explorer for sharing work with your colleagues. Use workflow groups to organize your workflows, components, and data files.

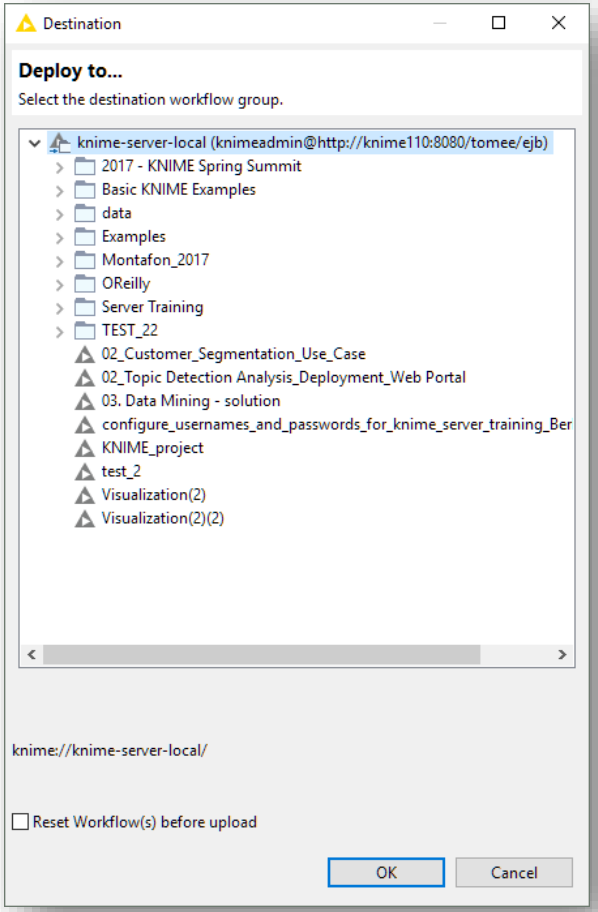
To move resources simply drag and drop or copy and paste



Server Mount Point as a Shared Repository (2/2)

Another way to deploy resources on KNIME Server...



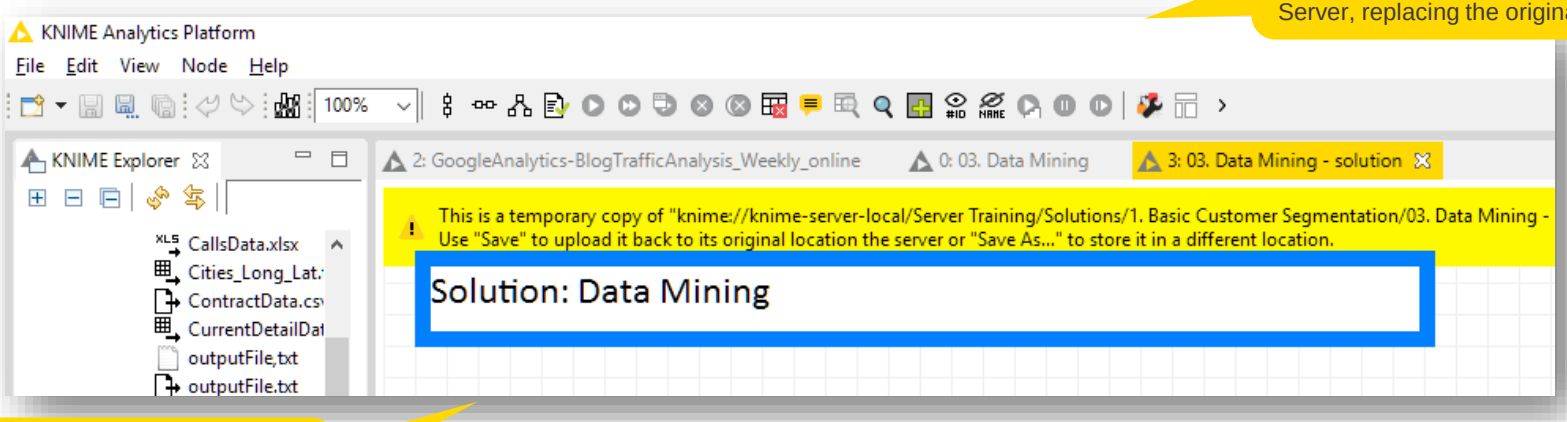


Open for Innovation KNIME

Inspecting a Workflow from KNIME Server

- By double-clicking a workflow on KNIME Server, the client downloads it (to a temporary location) and subsequently opens it automatically
- The yellow bar at the top of the editor indicates that this is a temporarily downloaded server workflow

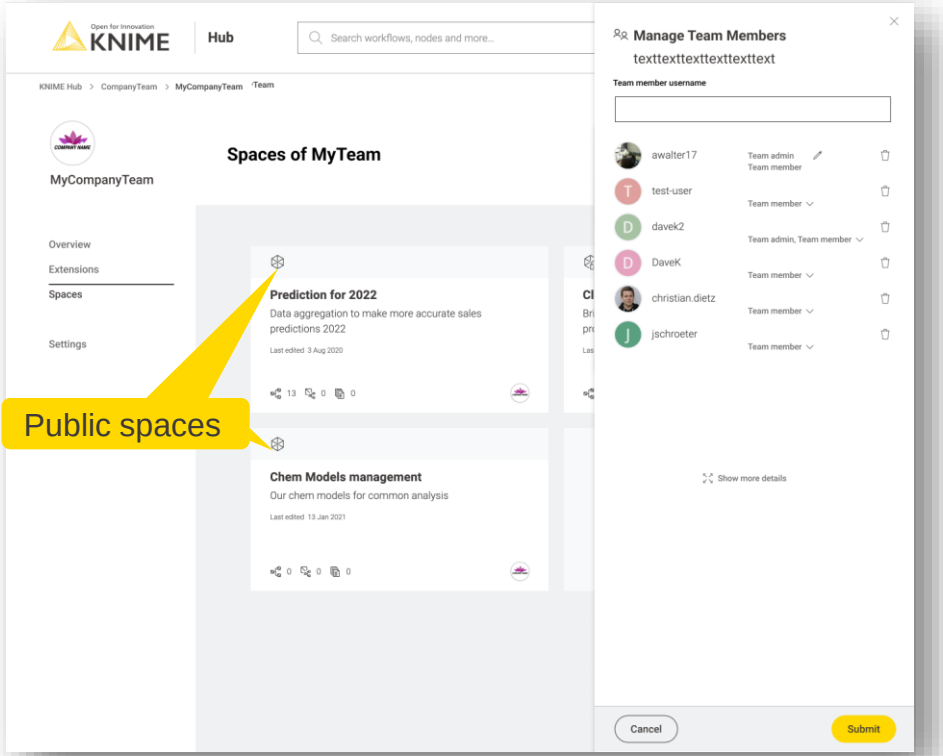
If you save the temporary copy after editing it (ctrl+s), you get the option of deploying it to KNIME Server, replacing the original copy



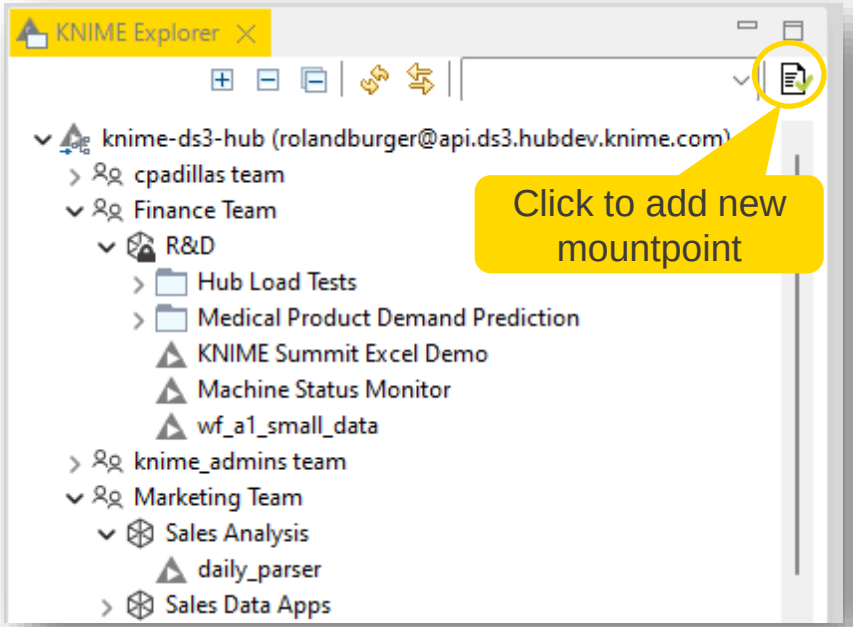
Some functionalities do not work on temporary copies (e.g., you cannot open BIRT reports)

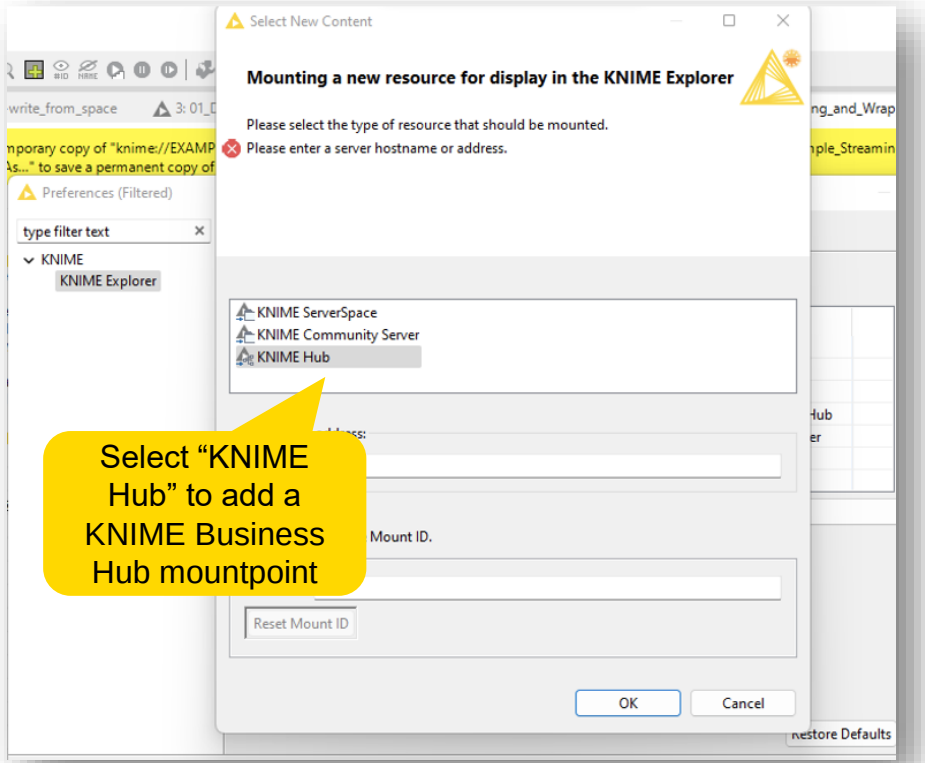
KNIME Business Hub: Teaser!

- UI similar to KNIME Hub
- Users collaborate as **teams** in **spaces** (**public** or **private**)
- **Spaces** store a team's files, components & workflows
  - Accessible via URL
- Each team determines:
  - Read, write or execute **permissions** for its spaces
  - **Execution contexts** (e.g., number of cores, what GPU or CPU) for its spaces' deployments



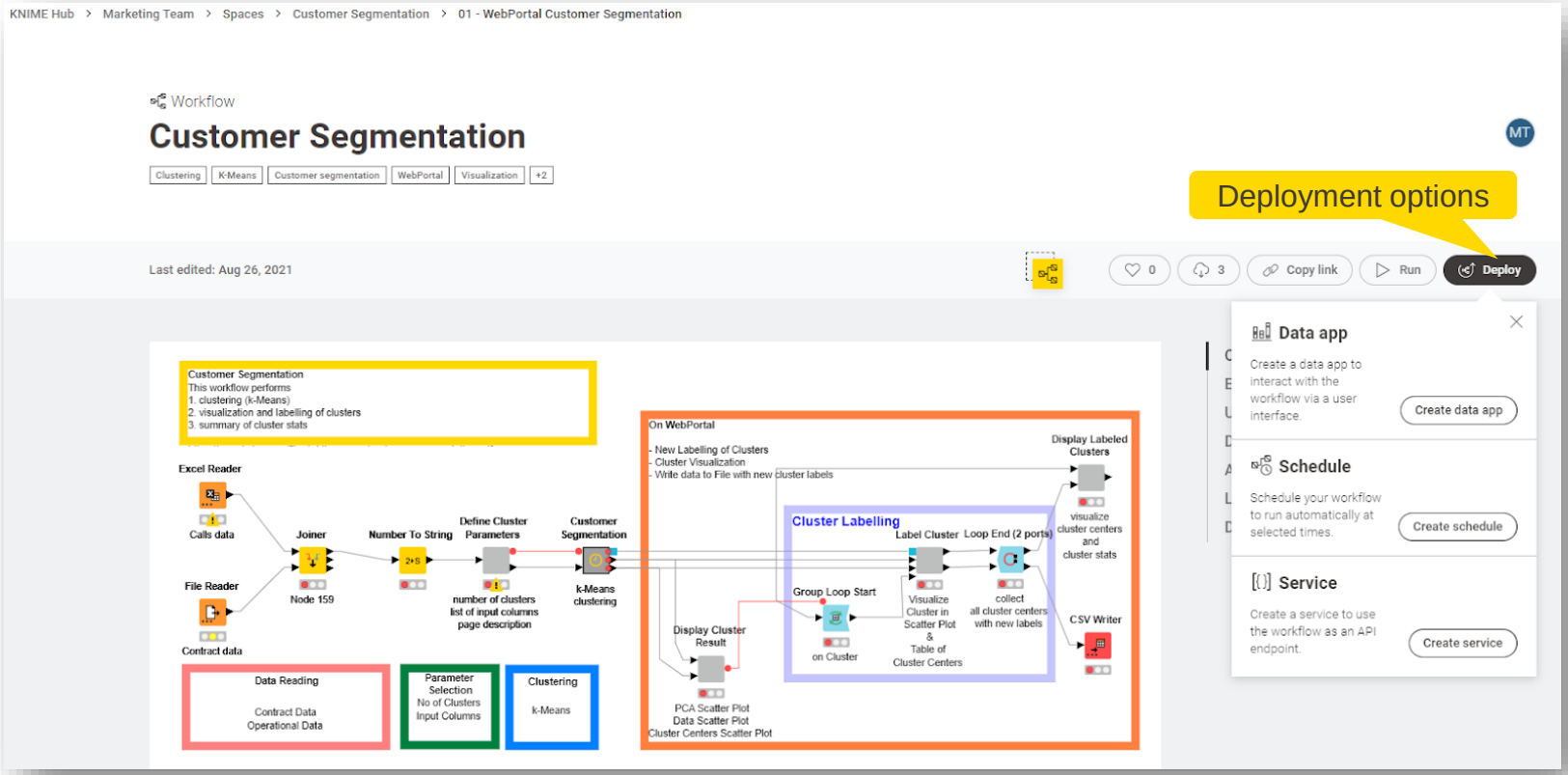
Preview: Connecting to KNIME Business Hub





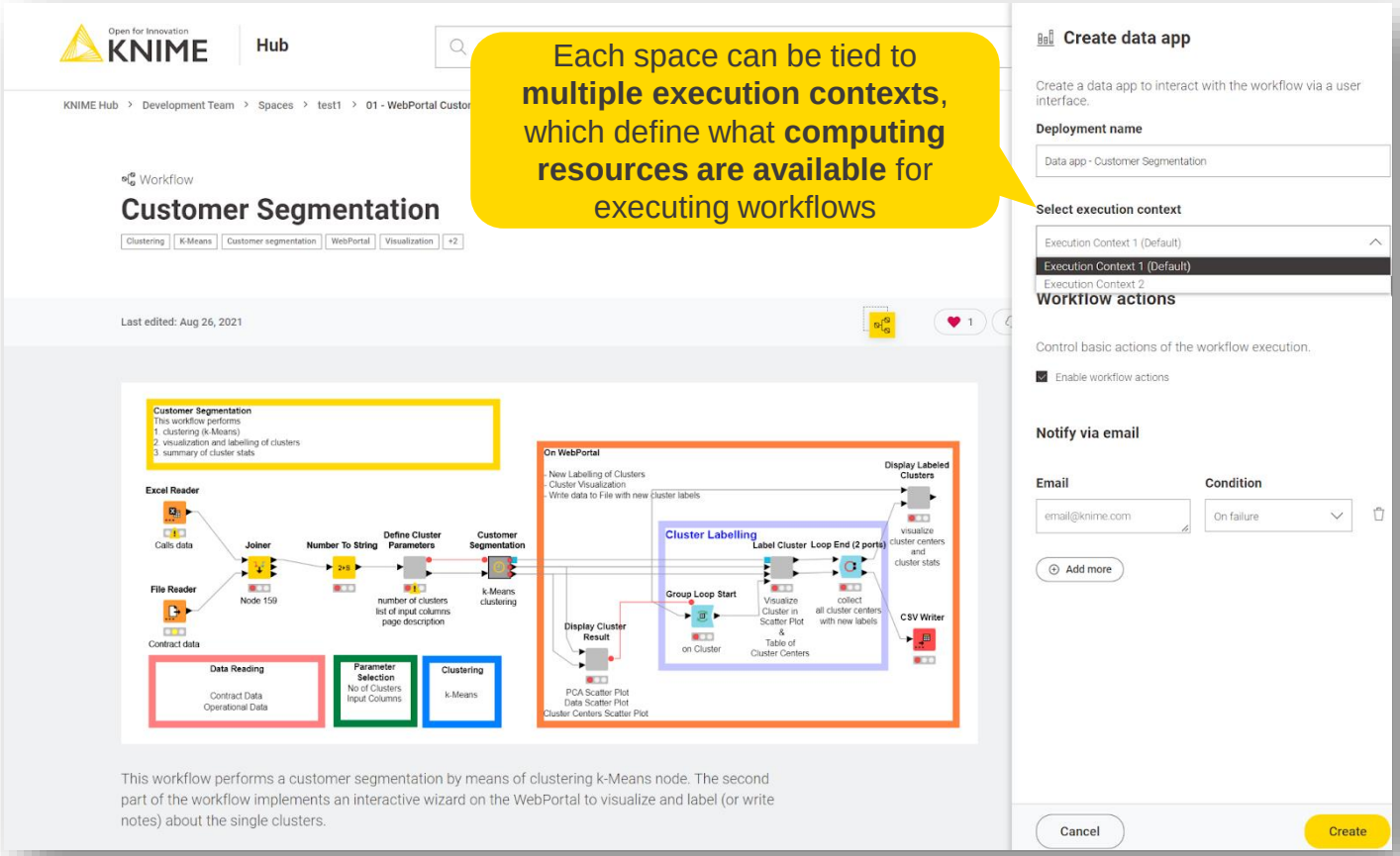
Preview: Deploying to KNIME Business Hub

- Moving a workflow to KNIME Business Hub doesn't mean that it's deployed yet.

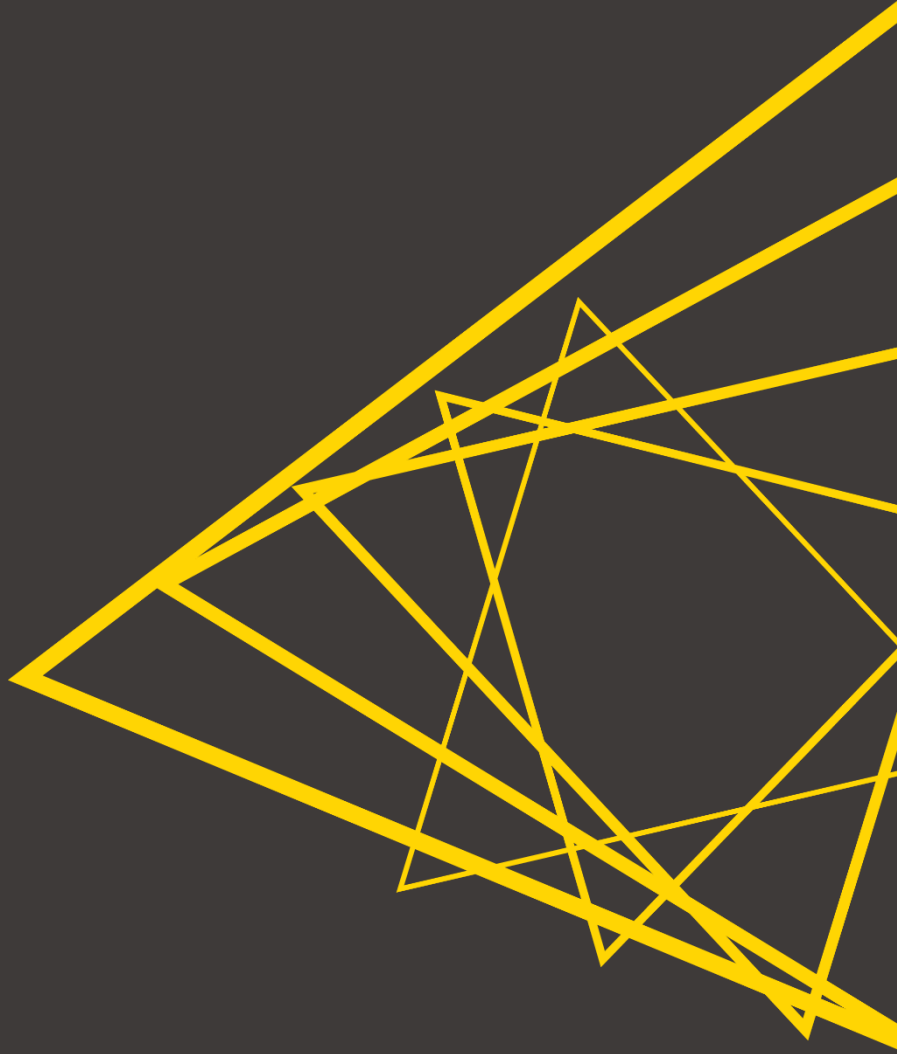


Preview: Deploying to KNIME Business Hub

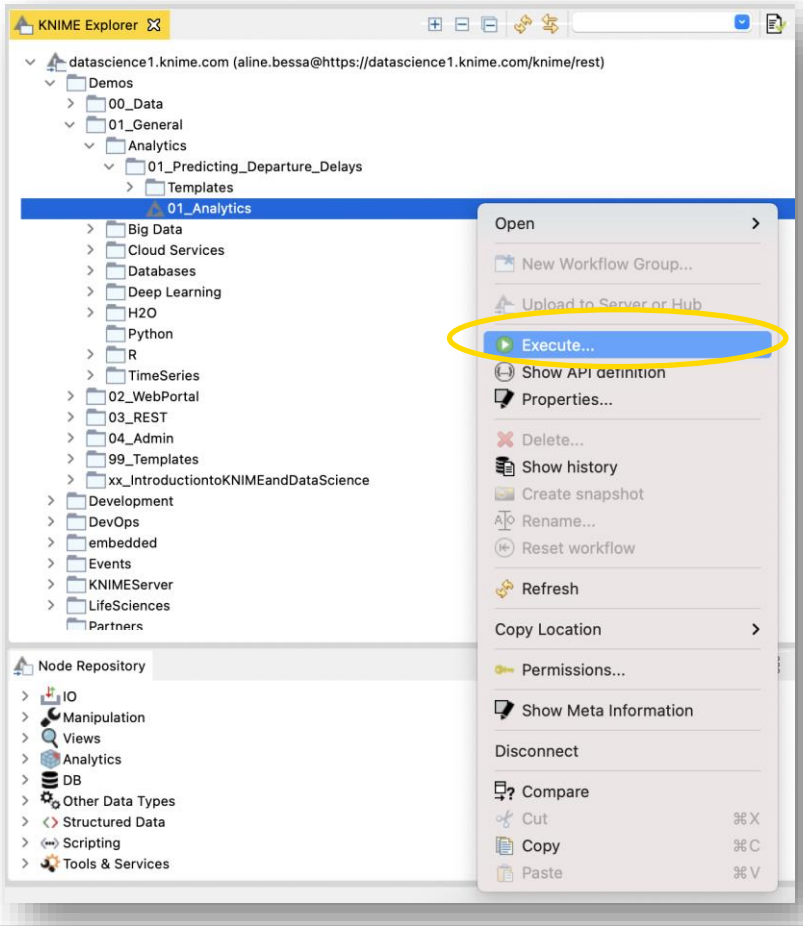
- When deploying a workflow, you can pick its **execution context**

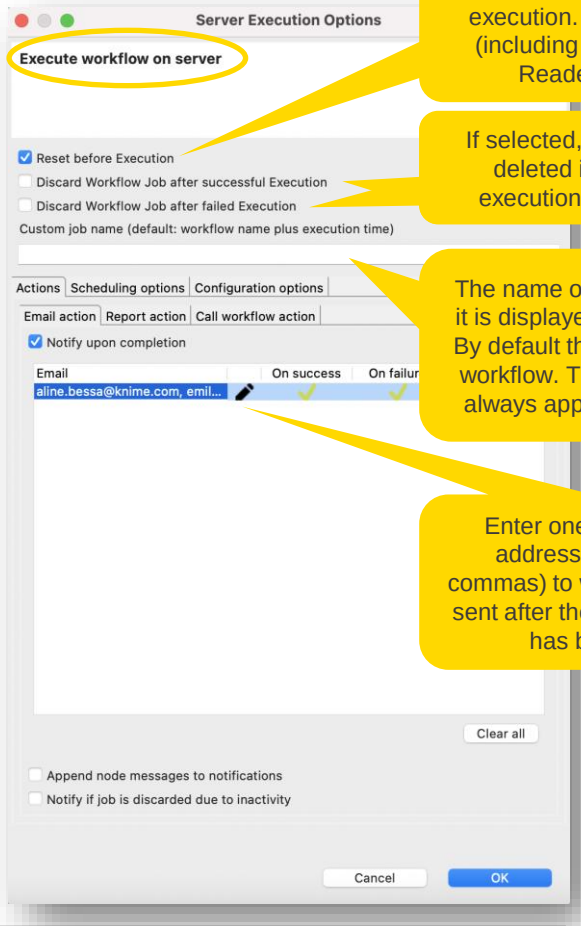


Workflow Execution and Scheduling



Executing a Workflow on the Server – Remote Execution





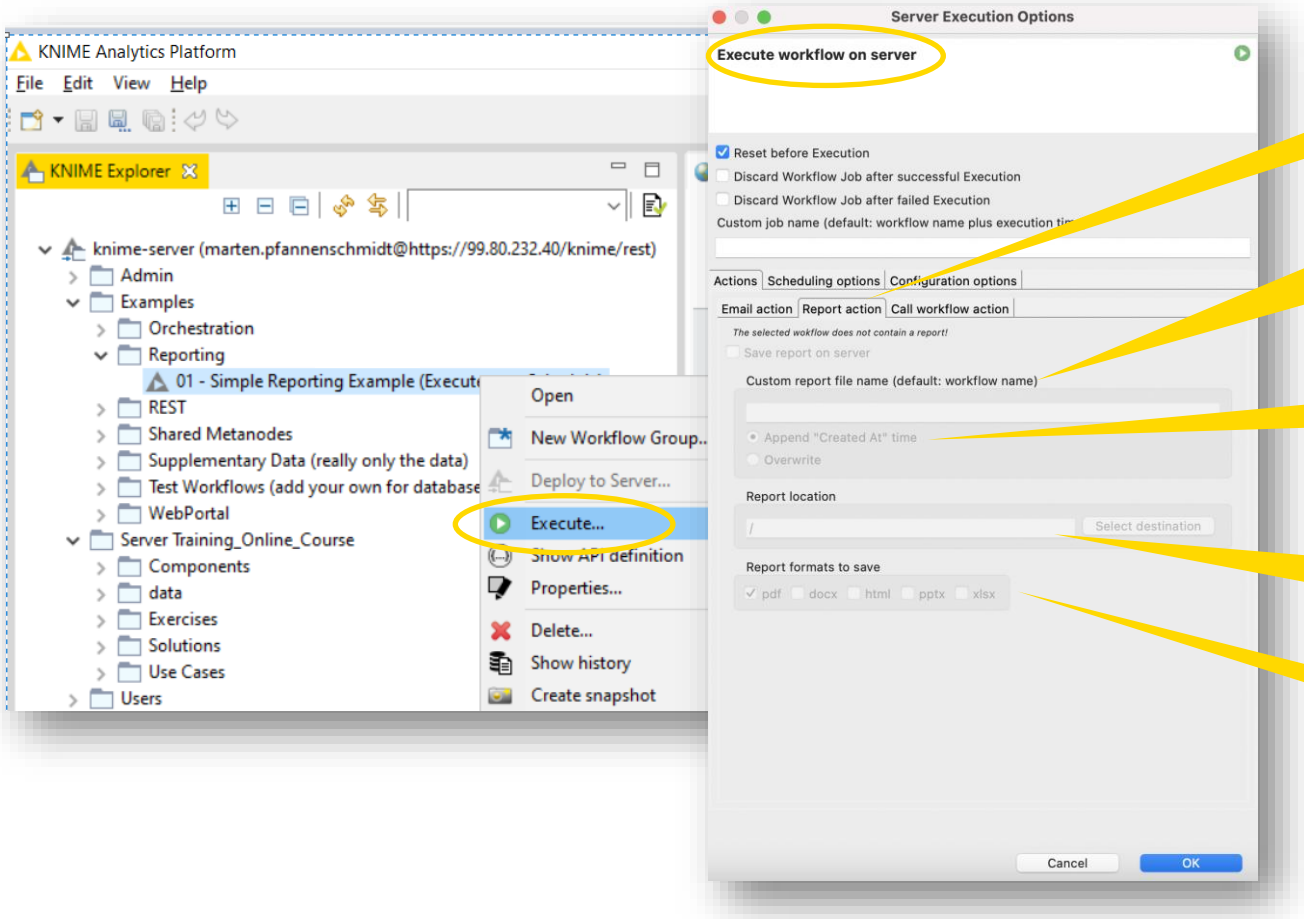
Check to reset workflow before execution. All nodes are reset (including File and Database Reader nodes, etc.).

If selected, the executed job is deleted immediately after execution, and is not saved.

The name of the workflow job as it is displayed in the server view. By default this is the name of the workflow. The execution date is always appended to the name.

Enter one or multiple email addresses (separated by commas) to which a notification is sent after the workflow execution has been finished.

Executing a Workflow on the Server – Remote Execution



If the workflow contains a report, you can select to save it on KNIME Server.

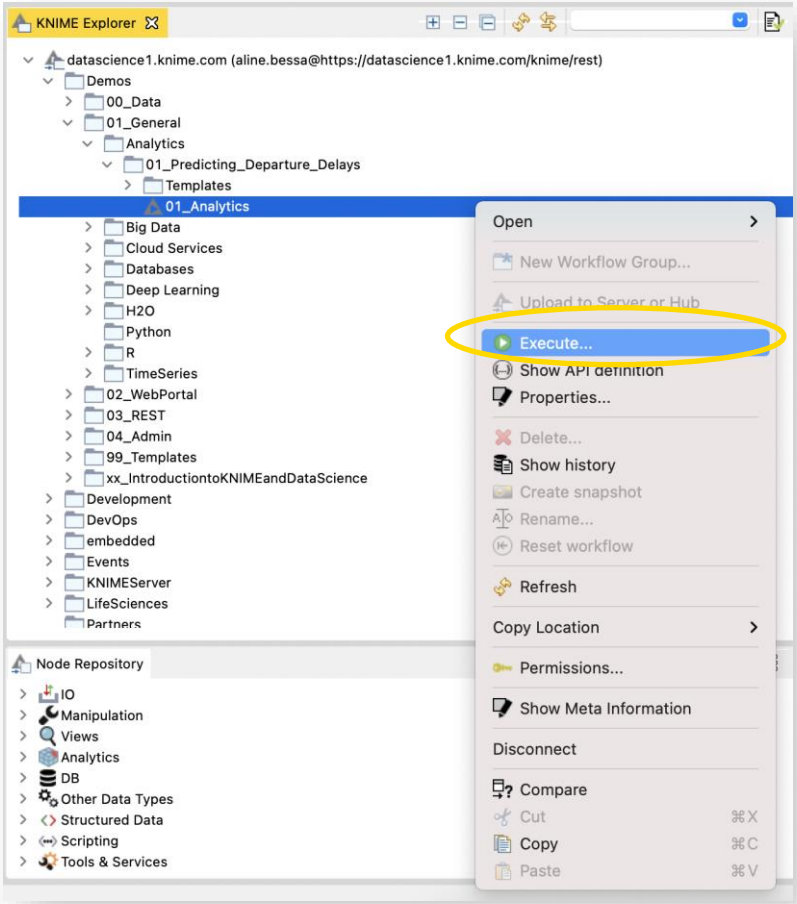
By default the name of the report is the name of the workflow. You can define a custom report file name, if you wish so.

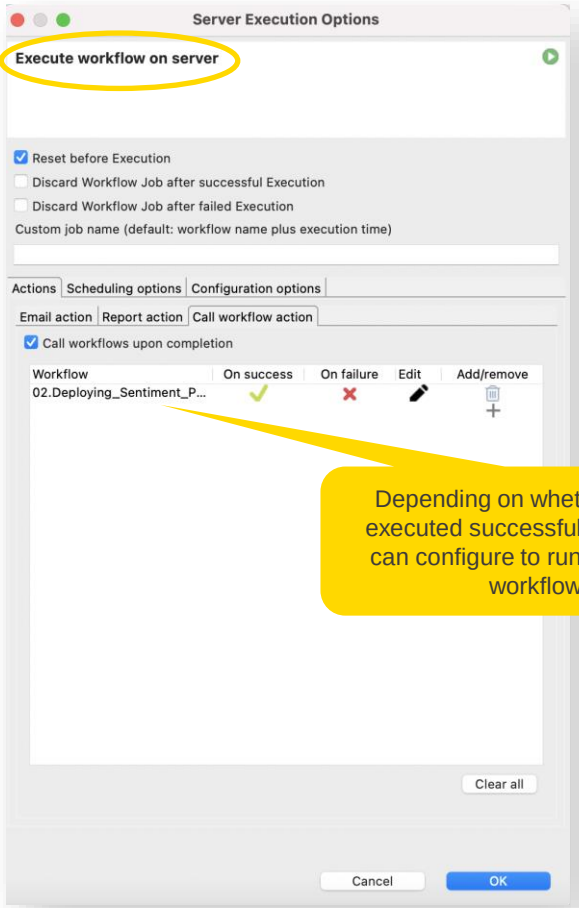
You can overwrite the report with every execution or append a timestamp.

Here you can define the location, where you want to store the report on KNIME Server.

You can select in which formats the report should be saved.

Executing a Workflow on the Server – Remote Execution

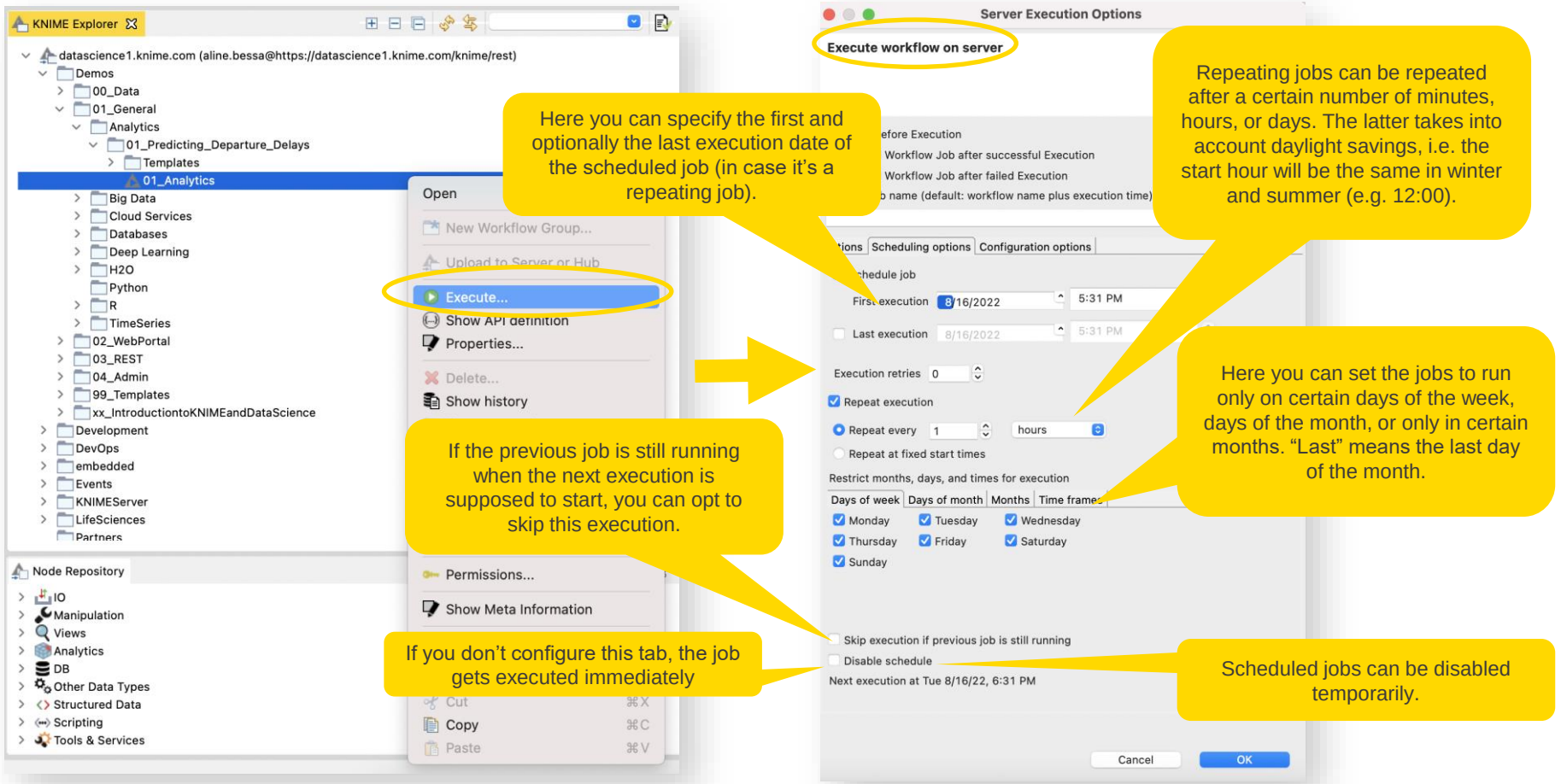




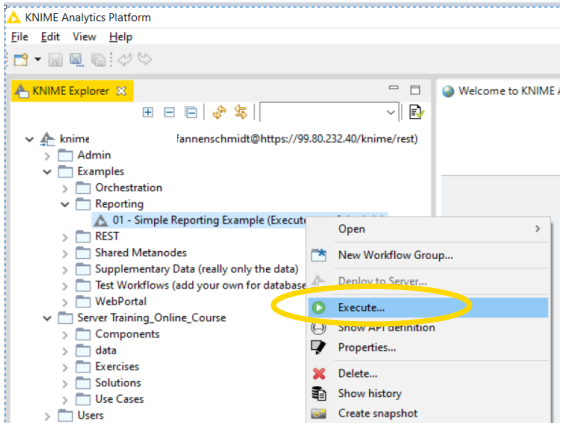
Depending on whether your job executed successfully or not, you can configure to run subsequent workflows

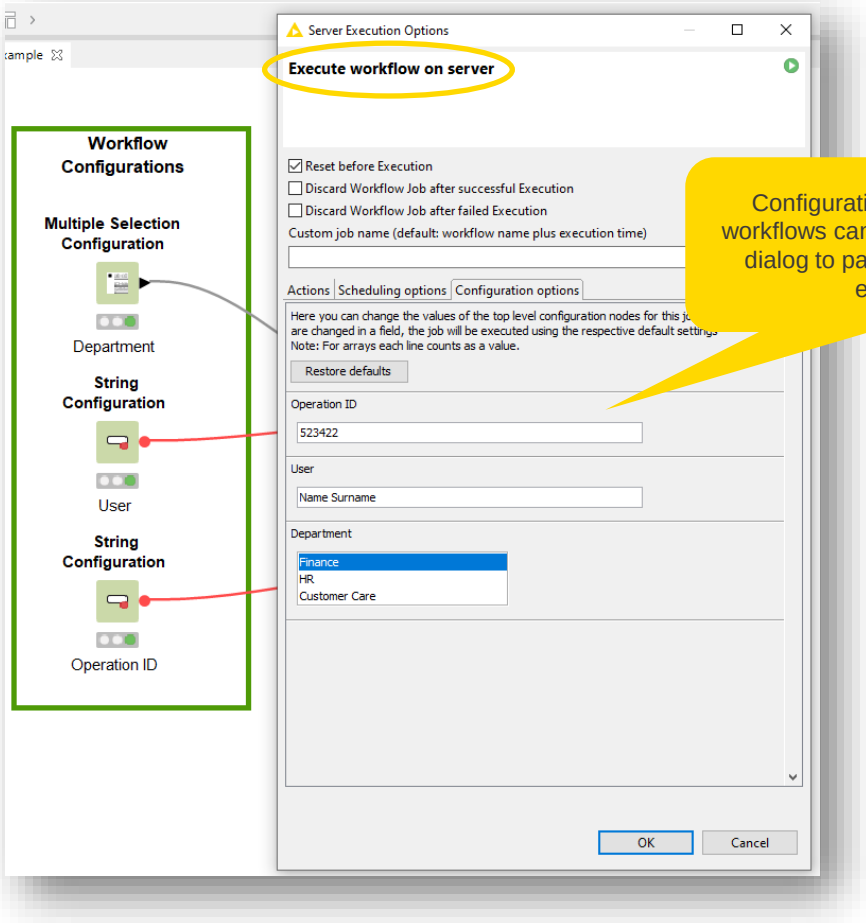
Open for Innovation KNIME

Executing a Workflow on the Server – Remote Execution



Executing a Workflow on the Server – Remote Execution

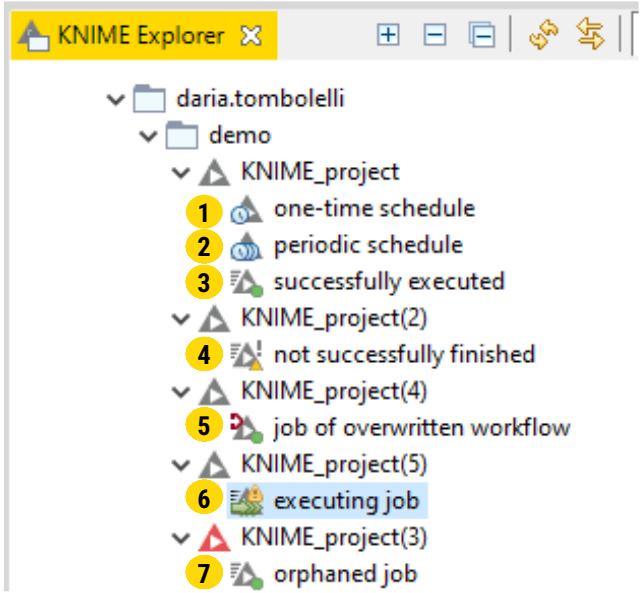




Workflow Jobs

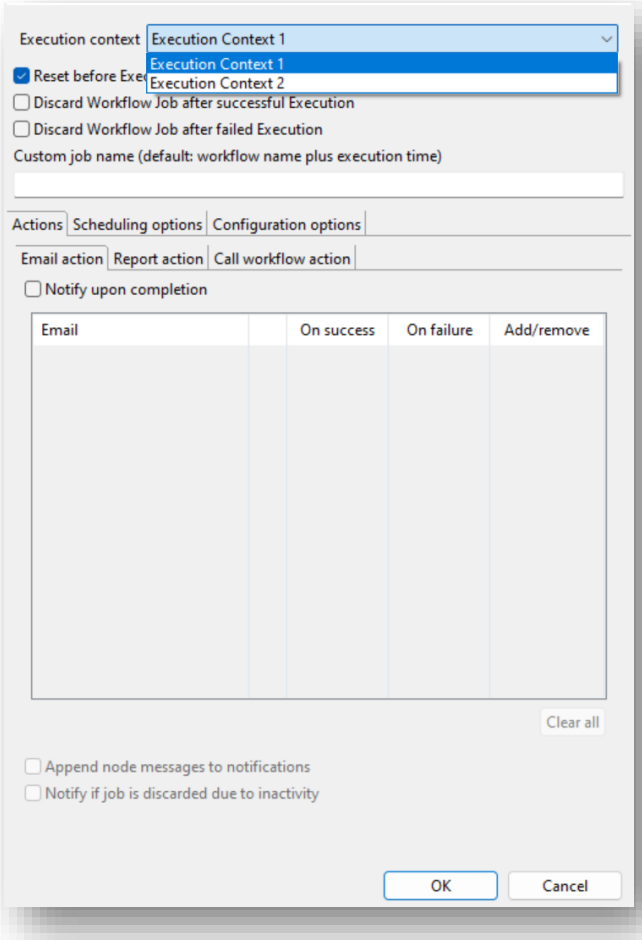
Remotely executed workflows are run as ***Jobs***:

- A workflow job is a copy of the workflow with specific settings and data
- Jobs are tied to the version from when the job was created
- Orphaned jobs are colored red
- Jobs have messages (e.g. successful or failure)
- Jobs can be saved as a workflow for data provenance and debugging
  (right-click → Save as)

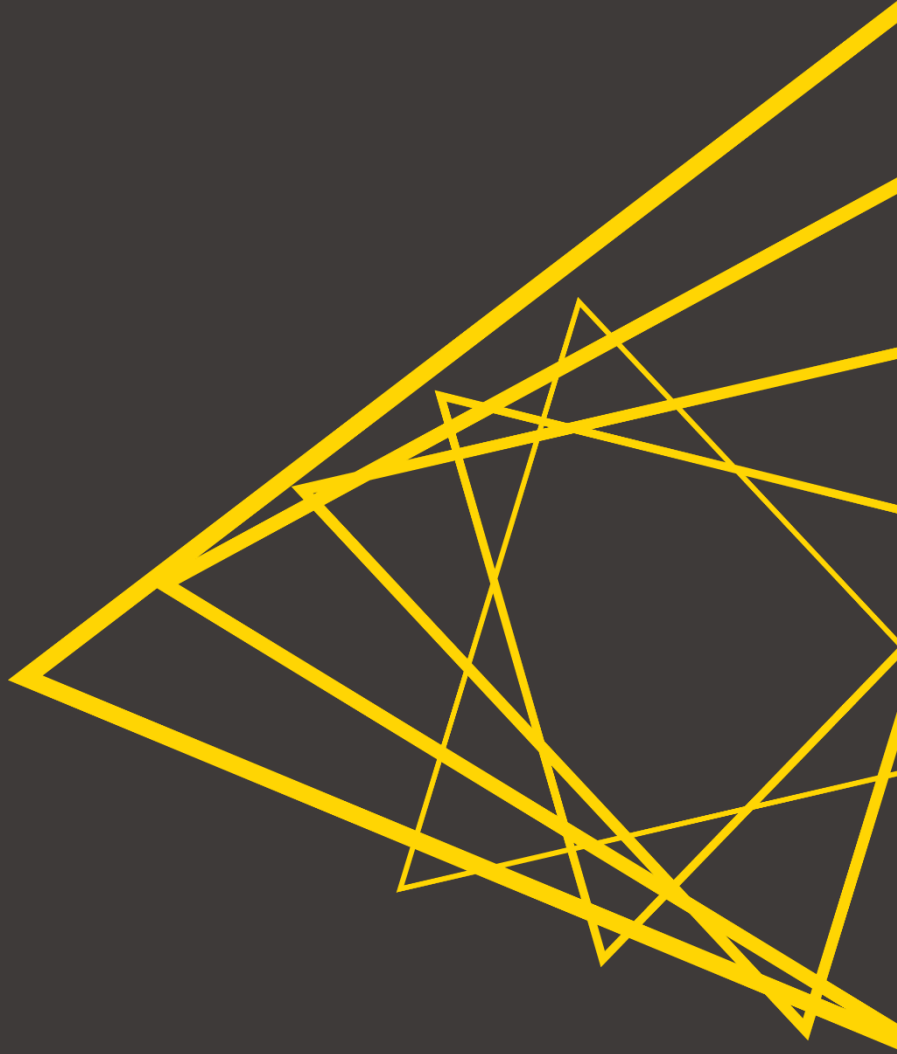


Preview: Executing Workflows in KNIME Business Hub

- It is possible to execute workflows in KNIME Business Hub from KNIME Analytics Platform
- Very similar to the KNIME Server process
  - **New:** You can choose the execution context for your workflow

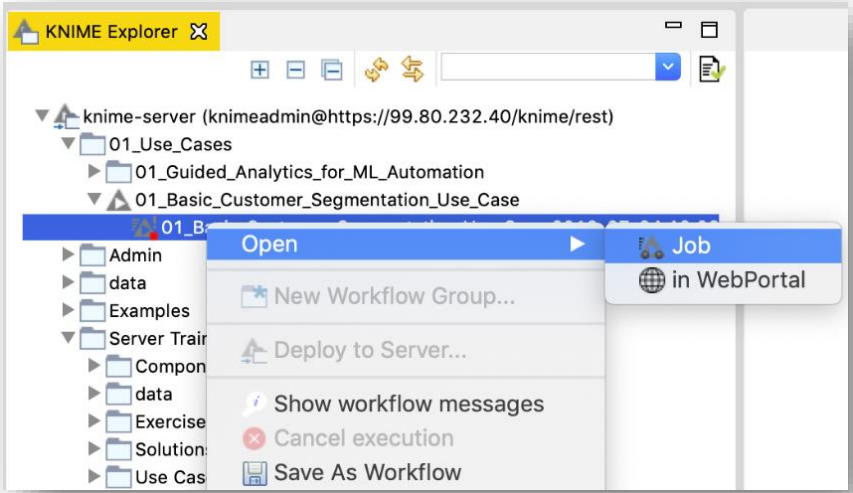


Remote Workflow Editor

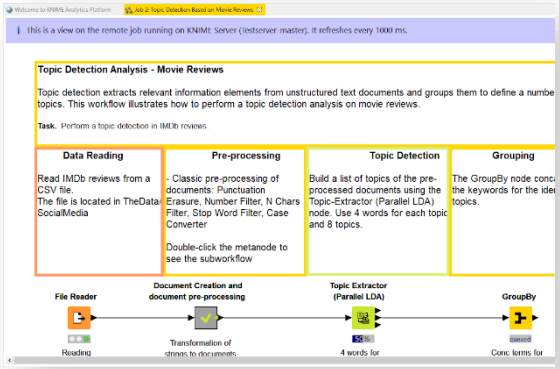


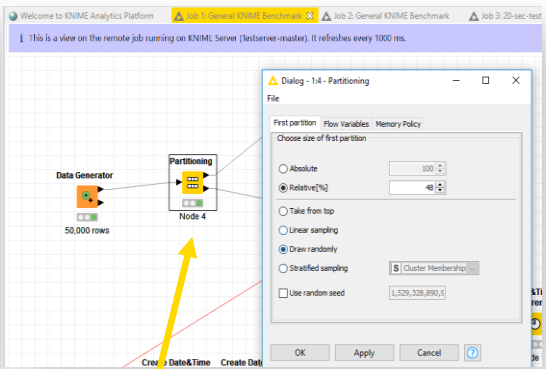
Remote Workflow Editor

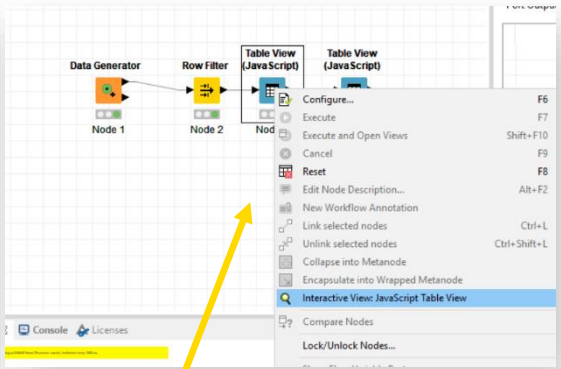
- Remote Control of Job running on KNIME Server
- Capabilities:
  - Live update of workflow job execution (executing node and progress)
  - Execute and cancel execution supported
  - Add/delete nodes
  - Change node settings
  - Inspect data tables / flow variables
  - JavaScript nodes can show data/views
- KNIME Remote Workflow Editor extension



Remote Workflow Editor







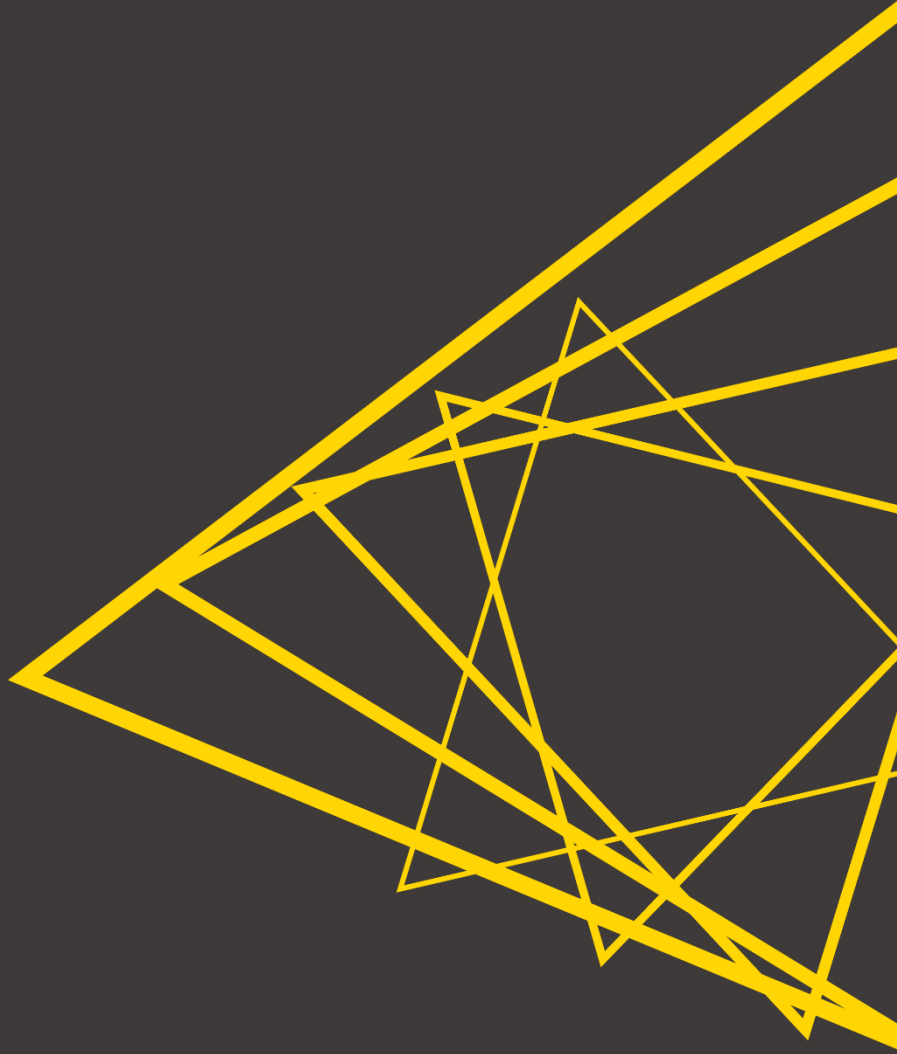
What's my workflow doing now?

Change node configurations in the Job

JavaScript View support

Open for Innovation KNIME

Permissions and Versioning



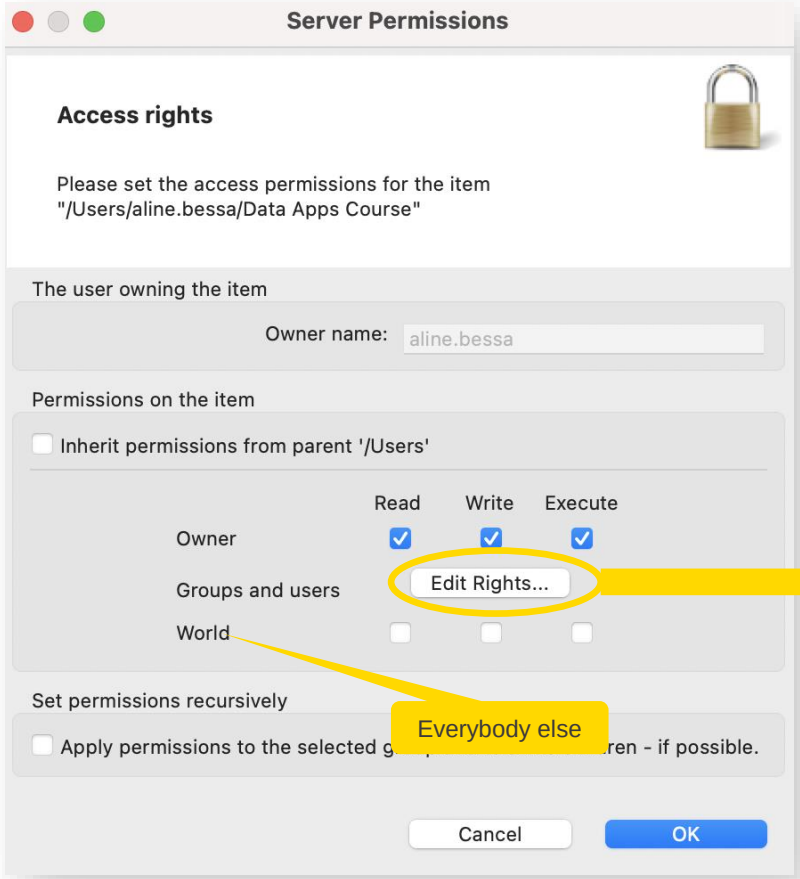
Permissions

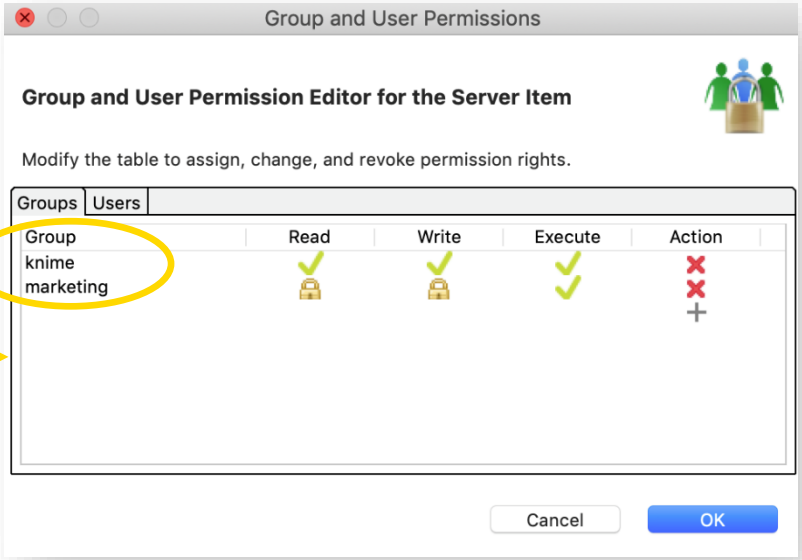
- Permissions can be set for all types of items: workflows, workflow groups, components, and data files
- Permissions are assigned to either individual people or user groups
- The user who uploads an item, automatically becomes its owner
- Users with admin rights have no restrictions on permissions
- The owner, plus everyone with admin rights, can assign and change permissions for an item
- It is also possible to set permissions on schedules, such that a schedule can be maintained/changed by a team member while the owner is, e.g., on vacation

Permission Types

Type	Workflow	Workflow Groups	Files/Components
Read	Download a workflow job - including data	See the content of a workflow group	▪ File: download data and execute workflows that use the data ▪ Component: use and download
Write	Overwrite, create snapshots, and delete workflows	Create and upload new items in a workflow group	Overwrite a file or component
Execute	Execute a workflow by creating a workflow job		

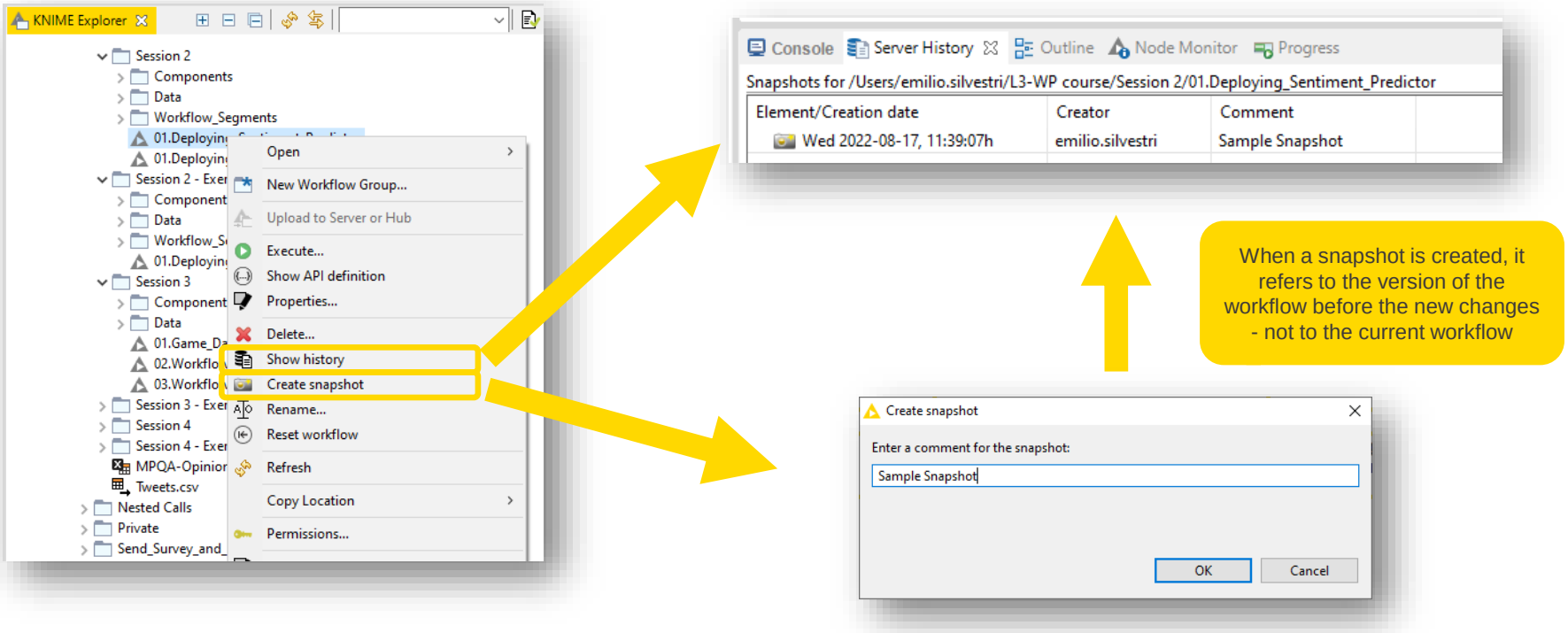
Setting Permissions





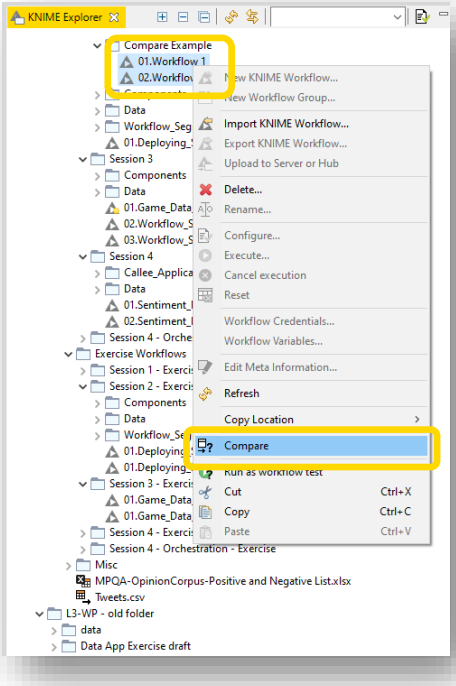
Versioning

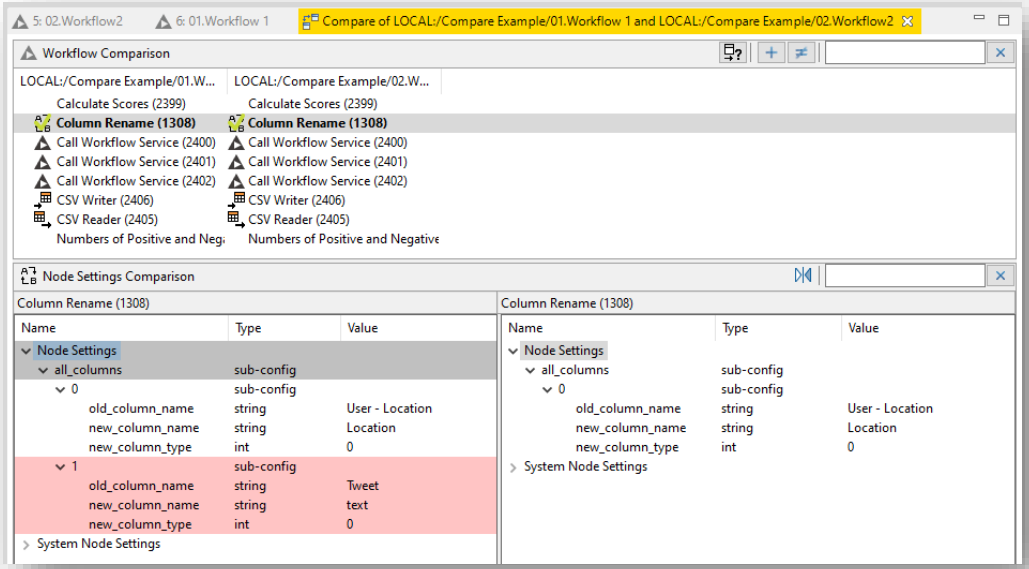
- Possibility of creating a history of items on Server
- Create snapshots of workflows, data files, and components. These are stored with a timestamp and a comment



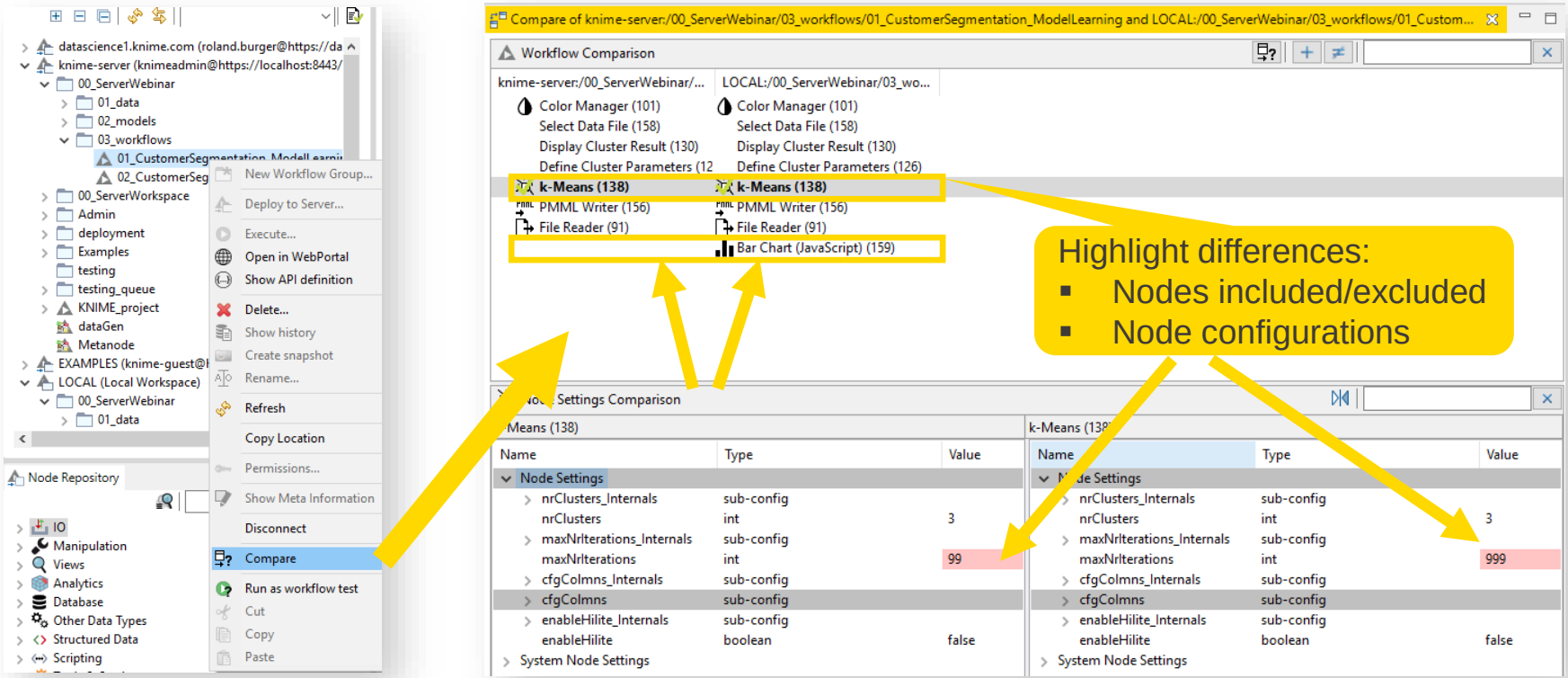
KNIME Workflow Difference (1/2)

- Automates identification and comparison of nodes in a workflow, metanodes, and two different workflows
- Identifies insertions, deletions, substitutions, and parameter change





KNIME Workflow Difference (2/2)

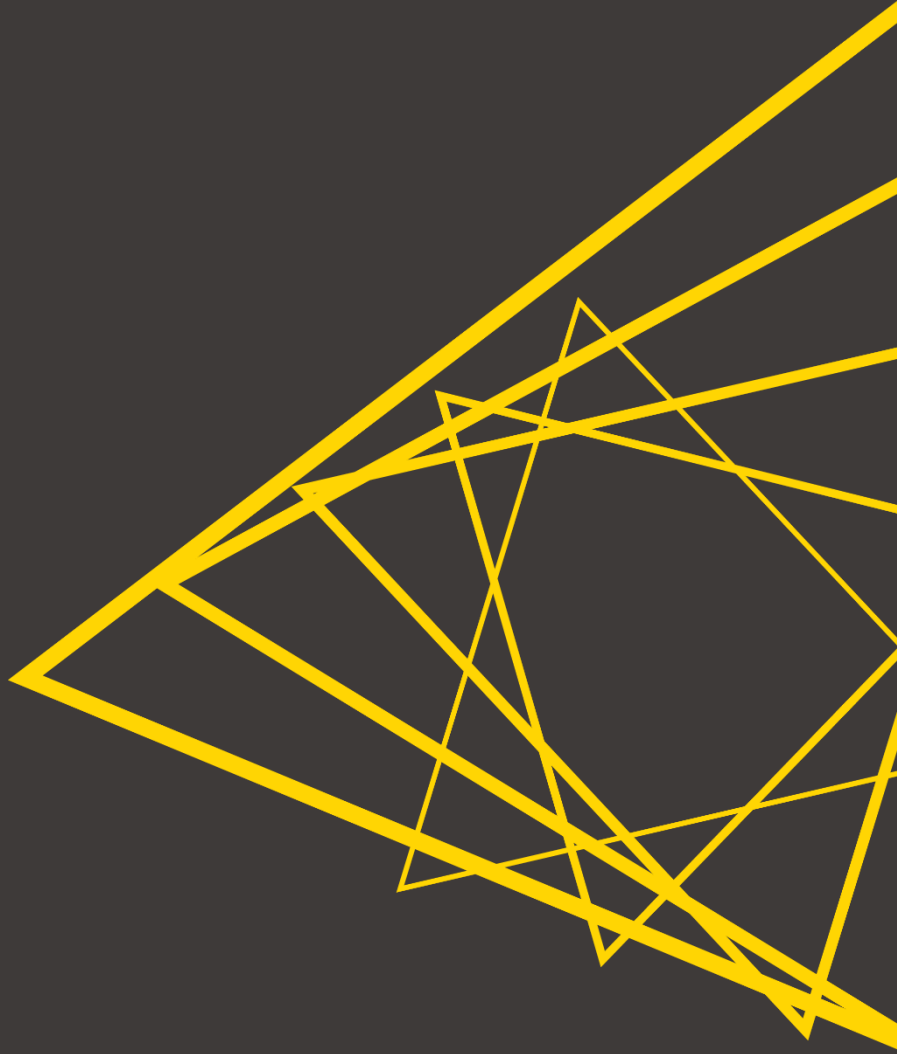


Session 2: Summary

After this session, you should be able to:

- Identify the different KNIME Server use cases and stakeholders
- Understand the steps to connect and deploy to KNIME Server
- Understand how to perform workflow execution and scheduling in KNIME Server
- Edit workflows with the Remote Workflow Editor
- Modify workflow and directory permissions and version them

Exercises



Config Details - Access KNIME Server

- KNIME Server address: http://34.207.219.51:8080/knime/

Login credential: *firstname.lastname*

Password: *knime*

Exercises

- **Exercise 01 -** Deploy the workflow group Session 2 - Exercises to the server, inside the workflow group *yourname.yoursurname*
- **Exercise 02 -** Schedule an execution of this workflow (2 minutes later) and send yourself a notification of successful/unsuccessful execution -- spoiler, it will fail!
- **Exercise 03 -** Open the workflow job on the server, recognize the error, fix it, and re-deploy the workflow. Create a snapshot when deploying the fixed workflow.
- **Exercise 04 -** Execute the workflow again on the server and check the produced results.

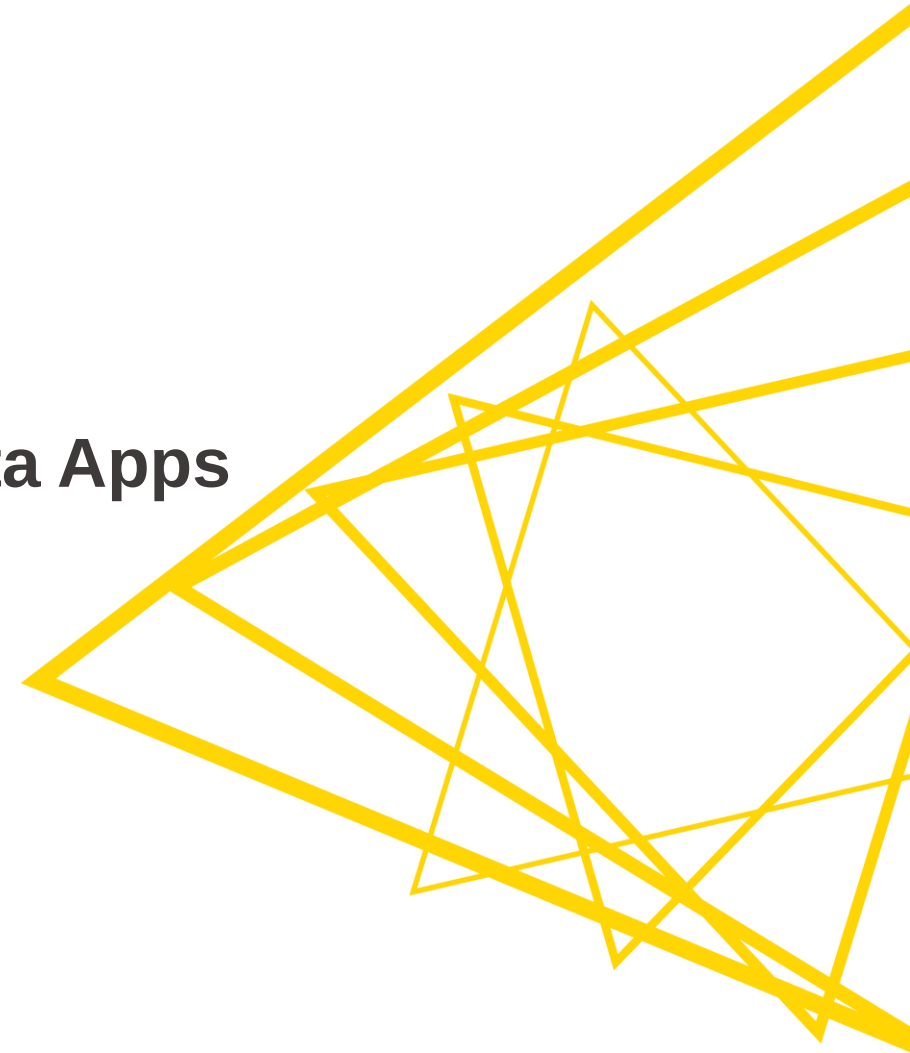
Open for Innovation
KNIME

Thank You!

Open for Innovation
KNIME

[L3-WP] Productionizing Data Apps

KNIME GmbH



Structure of the Course

Session	Topic	Duration
Session 1	What happens after the prototype is ready for deployment?	75 min
Session 2	Introduction to KNIME Server	75 min
Session 3	**Deploying WebPortal Data Apps**	75 min
Session 4	Runtime optimization, KNIME Edge, orchestration, and error handling	75 min
Session 5	Wrap-up Session	15 min

Structure of each session

- Discussion of past exercises
- Course
- Introduction of next exercises

Exercises Session 2

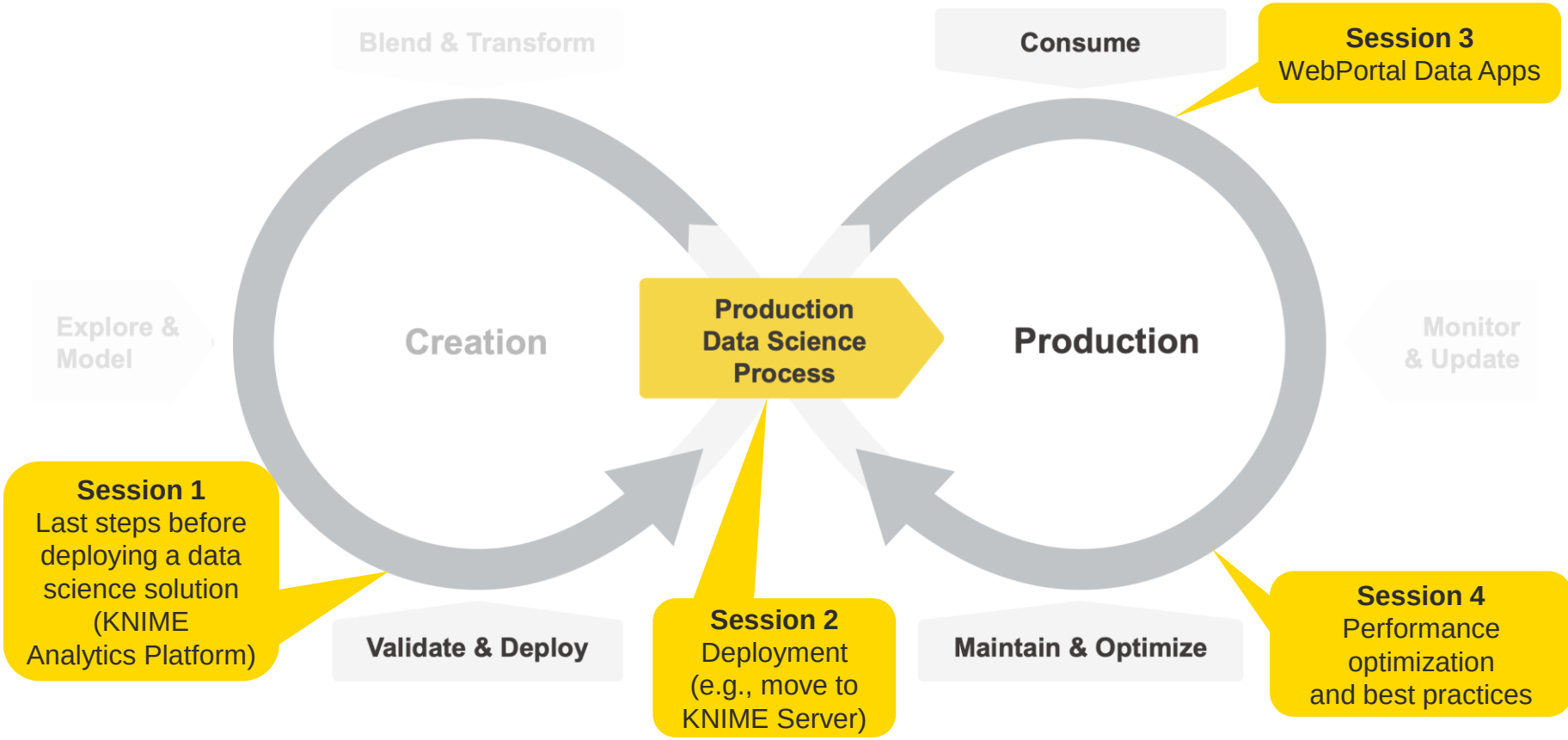
- **Exercise 01 -** Deploy the workflow group Session 2 - Exercises to the server, inside the workflow group *yourname.yoursurname*
- **Exercise 02 -** Schedule an execution of this workflow (2 minutes later) and send yourself a notification of successful/unsuccessful execution -- spoiler, it will fail!
- **Exercise 03 -** Open the workflow job on the server, recognize the error, fix it, and re-deploy the workflow. Create a snapshot when deploying the fixed workflow.
- **Exercise 04 -** Execute the workflow again on the server and check the produced results.

Open for Innovation
KNIME

Session 3

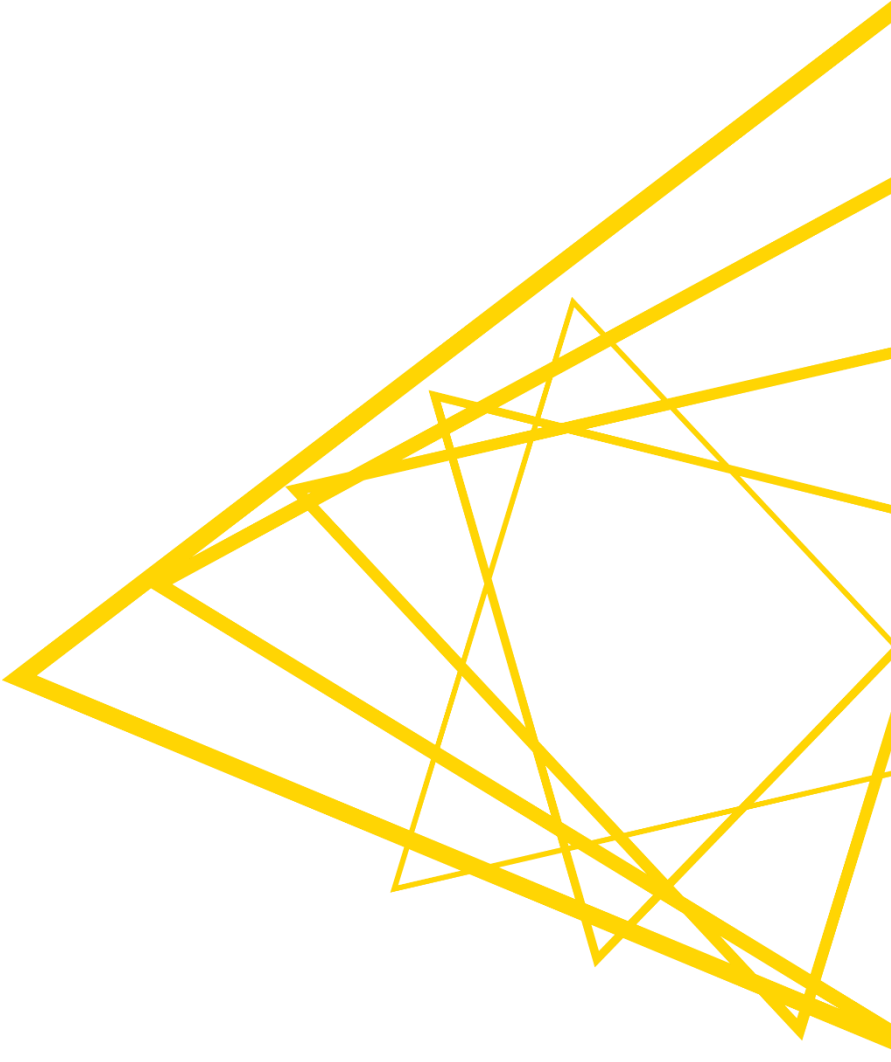
Deploying WebPortal Data Apps

This Course's Sessions

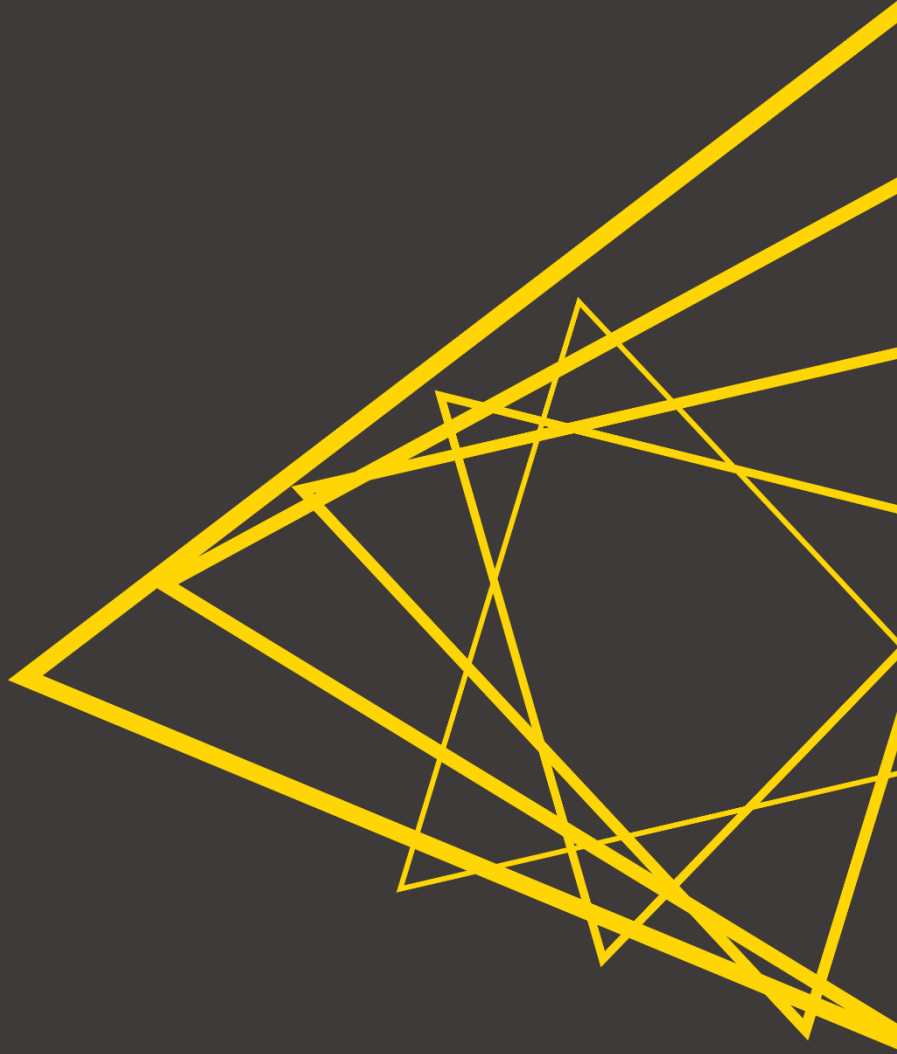


Learning Objectives

1. Define the concepts of WebPortal and Data App
2. Identify and apply the main WebPortal functionalities
3. List and use the tools to customize a Data App
4. Outline the steps to deploy and share a Data App



Introduction to the KNIME WebPortal

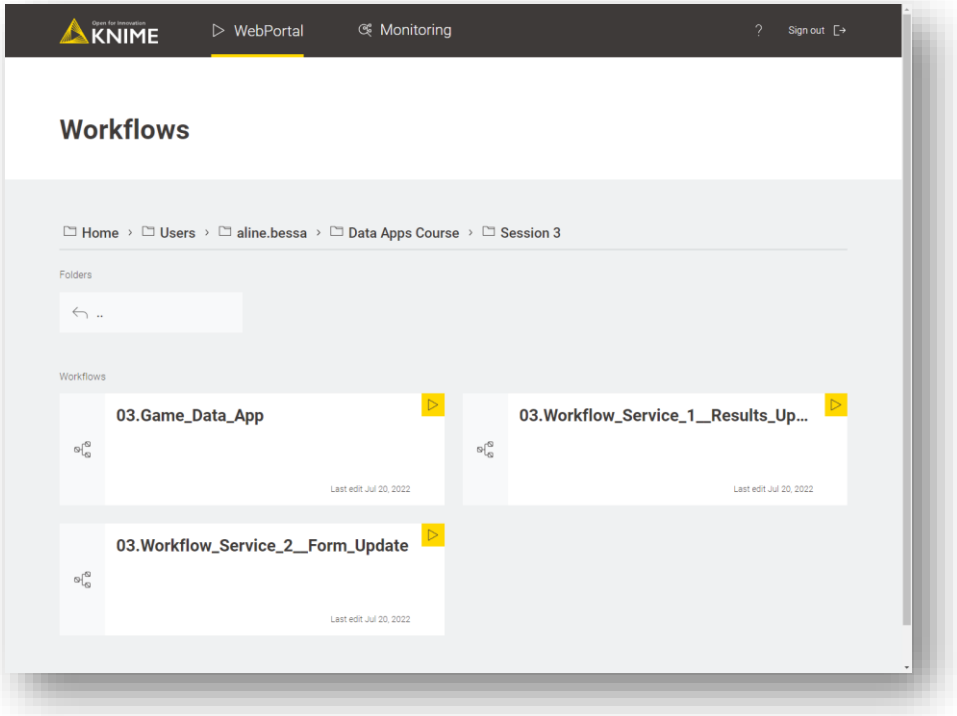


KNIME WebPortal: Features

- KNIME WebPortal is an extension to KNIME Server

Main purpose
Give data science access to non-experts

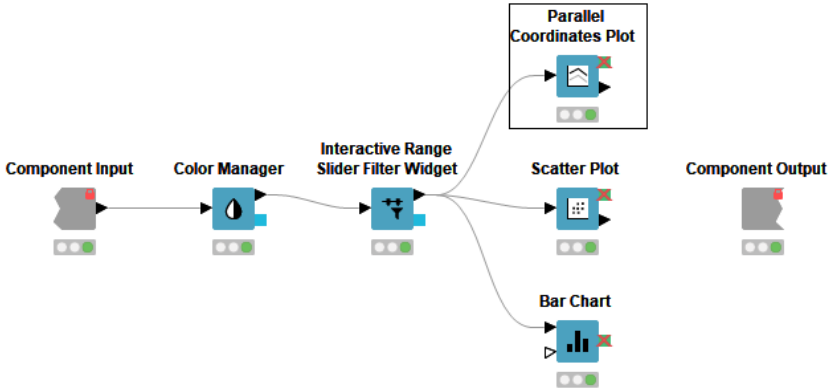
Main functionality
Interact with a workflow "without seeing it"

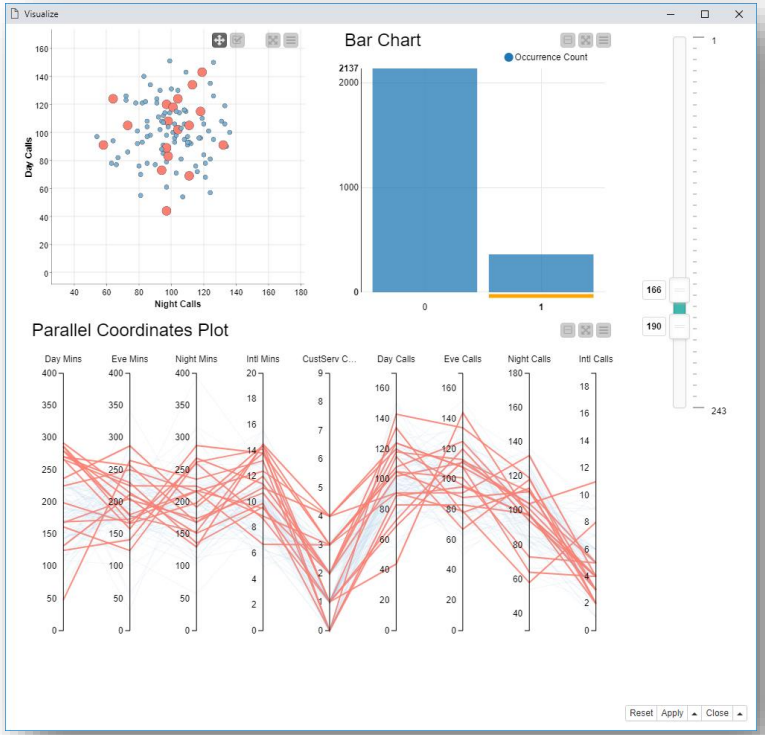


- Provides a Web Interface to
  - List accessible workflows
  - Execute workflows
  - Visualize workflow's Composite Views as web pages

Refresh: Component Composite View

- Multiple JavaScript View nodes can be combined in Components
- Selections are transmitted to all other views
- Widgets enable interactivity





Composite Views: KNIME AP vs KNIME WebPortal

KNIME Analytics Platform (AP)

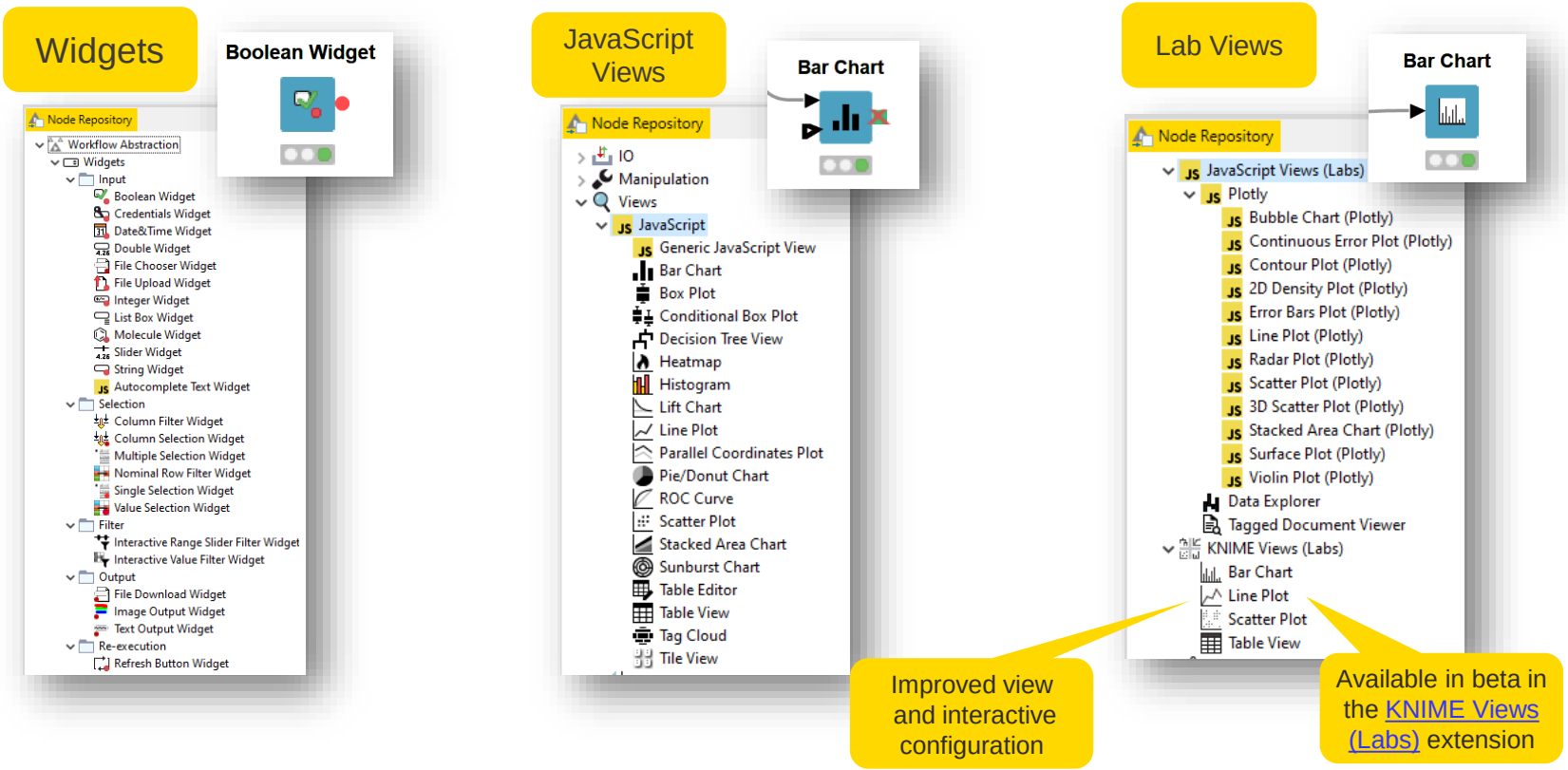
Executed locally
Visualized in KNIME AP

KNIME WebPortal

Executed on KNIME Server
Visualized in Browser as **Data App**

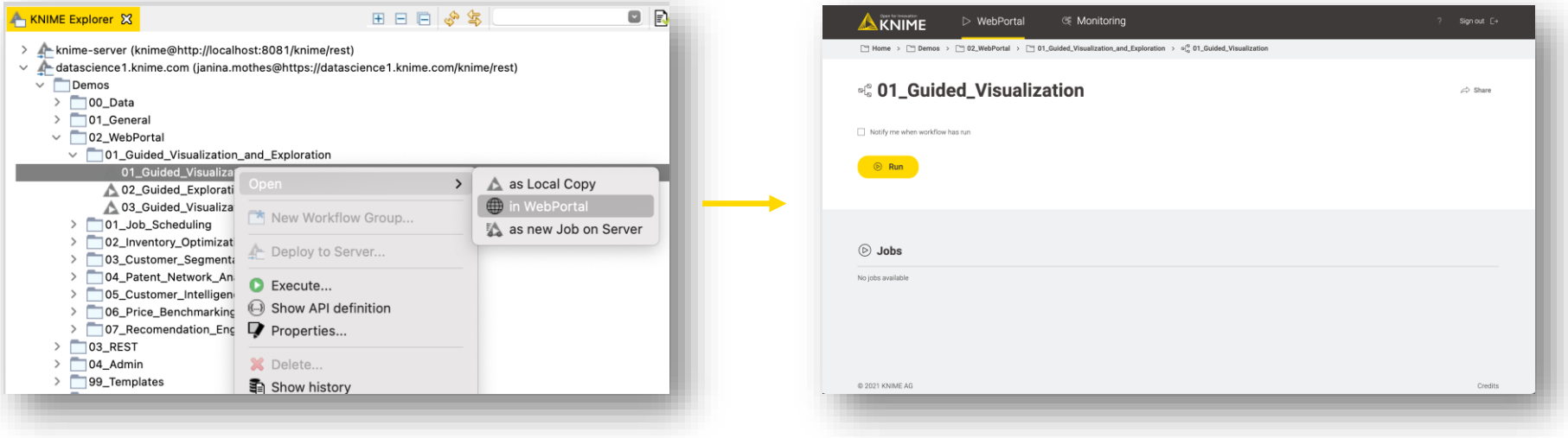
Refresh: Nodes that appear in the composite view

- If it's blue, you get a view!



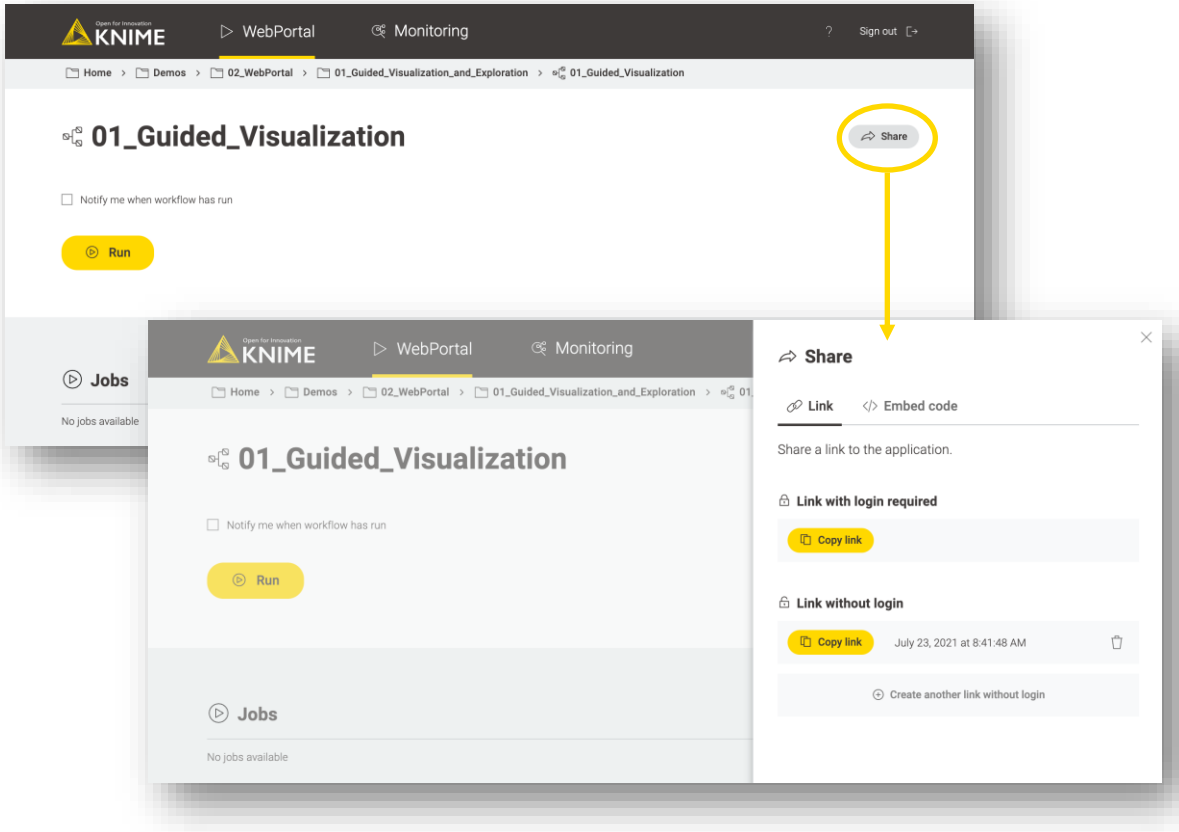
Access to KNIME WebPortal from the Client

- You can start the execution of a workflow on KNIME WebPortal directly from KNIME Analytics Platform
  - Right-click on the workflow available in the KNIME Server Mountpoint and click *Open in WebPortal*



Access to KNIME WebPortal from the Browser

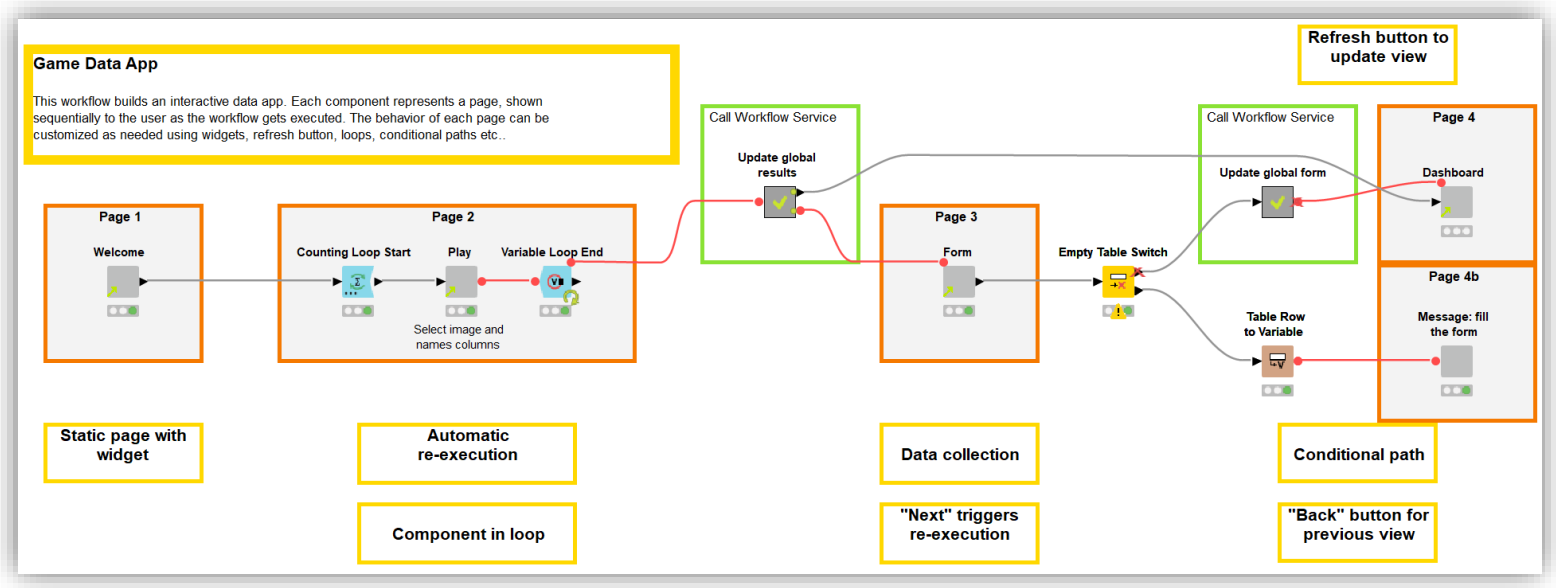
- KNIME WebPortal can also be accessed directly in the Browser with a specific URL
- Simply copy the link via the Share option



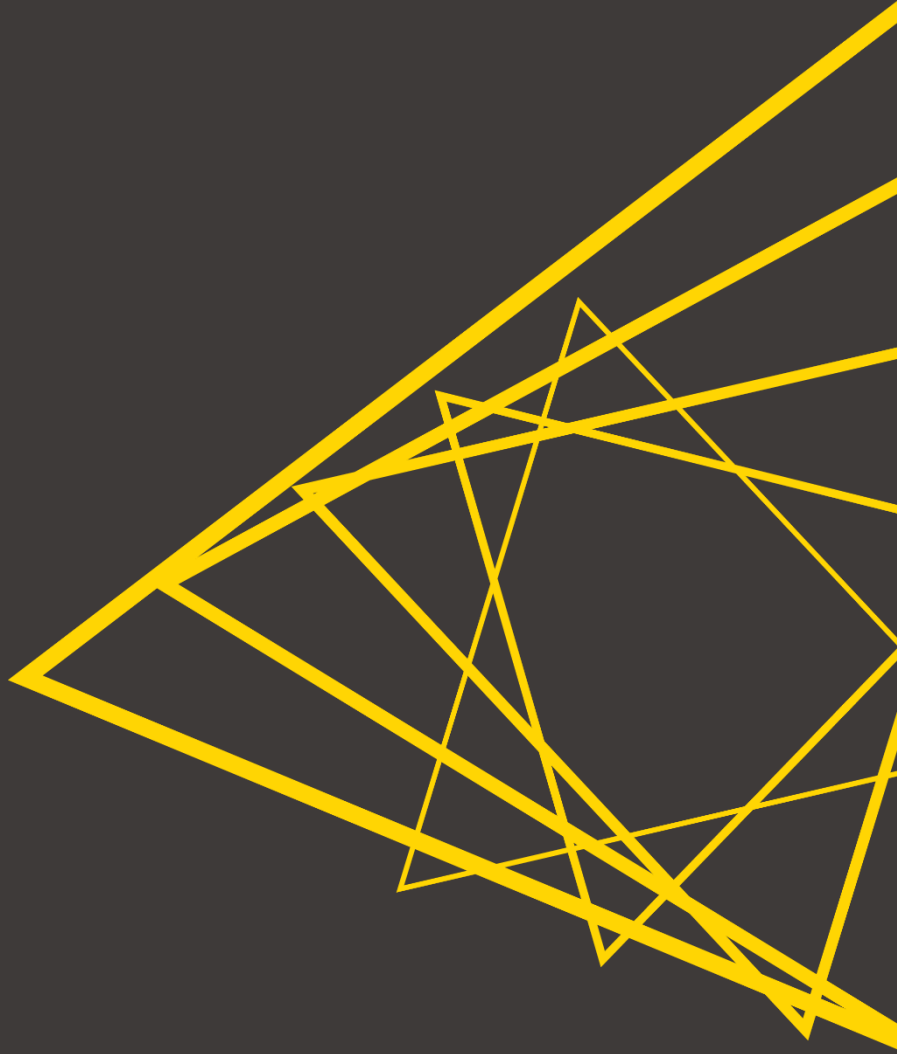
WebPortal Data App example

Today’s example

- A full web application running on WebPortal, which embeds
  - A simple guessing game
  - A feedback form
  - An interactive dashboard

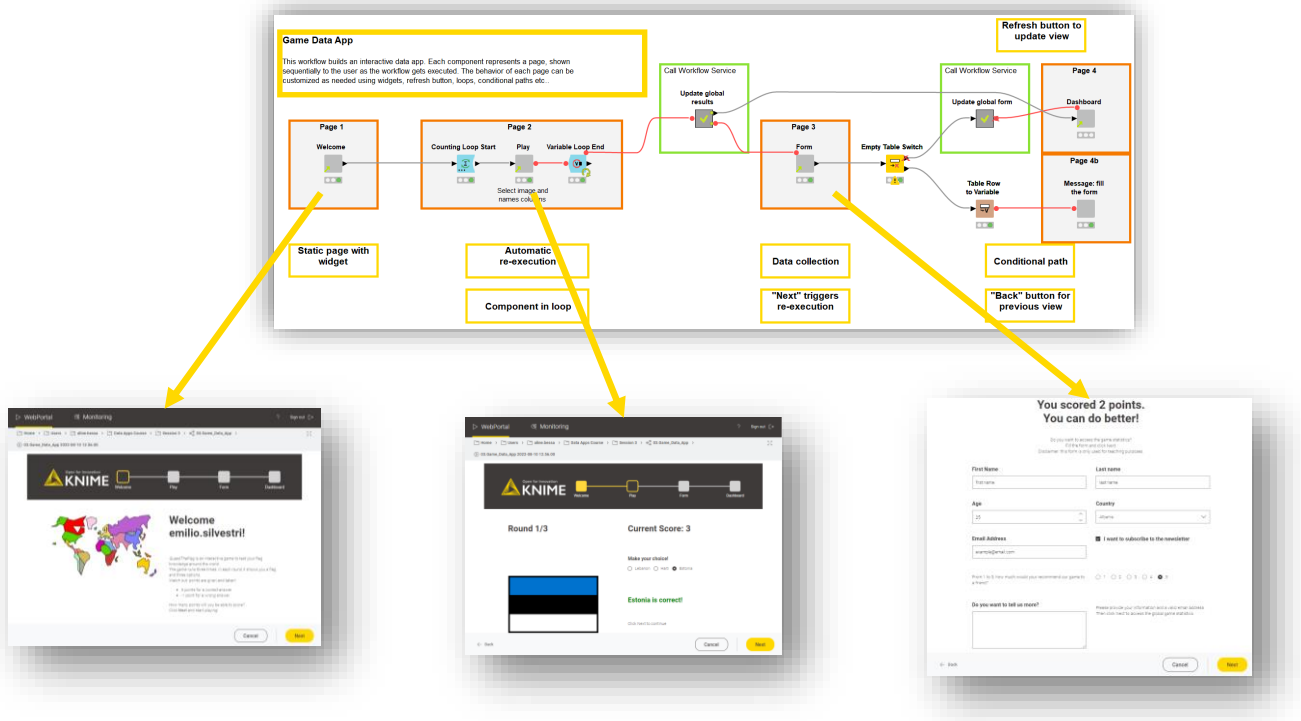


Components and WebPortal views



WebPortal paging functionality

- Component → WebPortal page
- A sequence of components generates **a sequence of WebPortal pages**



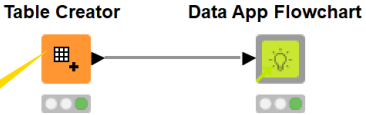
In the KNIME universe this is called **Data App**

Data App Flowchart

- Navigation made easy with the Data App Flowchart verified component

Data App Flowchart usage

- Drag and drop the data app flowchart inside each component
- Define the flowchart by changing the input table
- Customize the appearance via its dialogue
- Make sure it is placed at the top of the composite view via the "Node Usage and Layout"



user journey data

header flowchart describing the user journey in the app

Include it in every page of the Data App

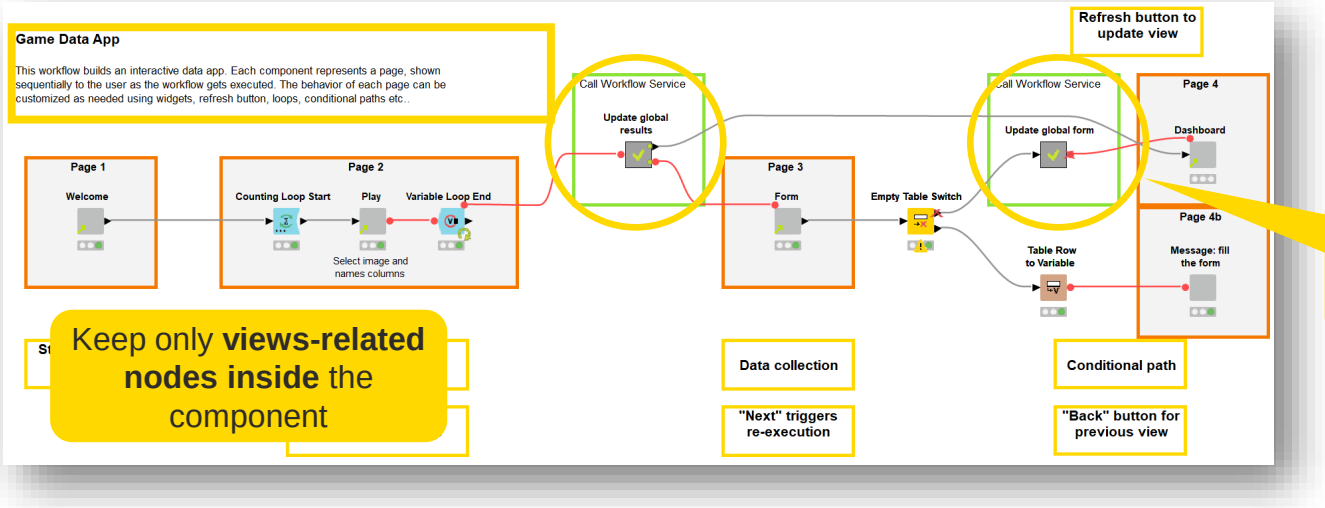
Table "default" - Rows: 4 | Spec - Columns: 2 | Prop

Row ID	S steps	I active
Row0	Welcome	1
Row1	Play	0
Row2	Form	0
Row3	Dashboard	0

Component re-execution via the “Next” button

- Components in the WebPortal get executed (at least) twice
- First time to create the displayed web page
- Second time when clicking the “Next” button

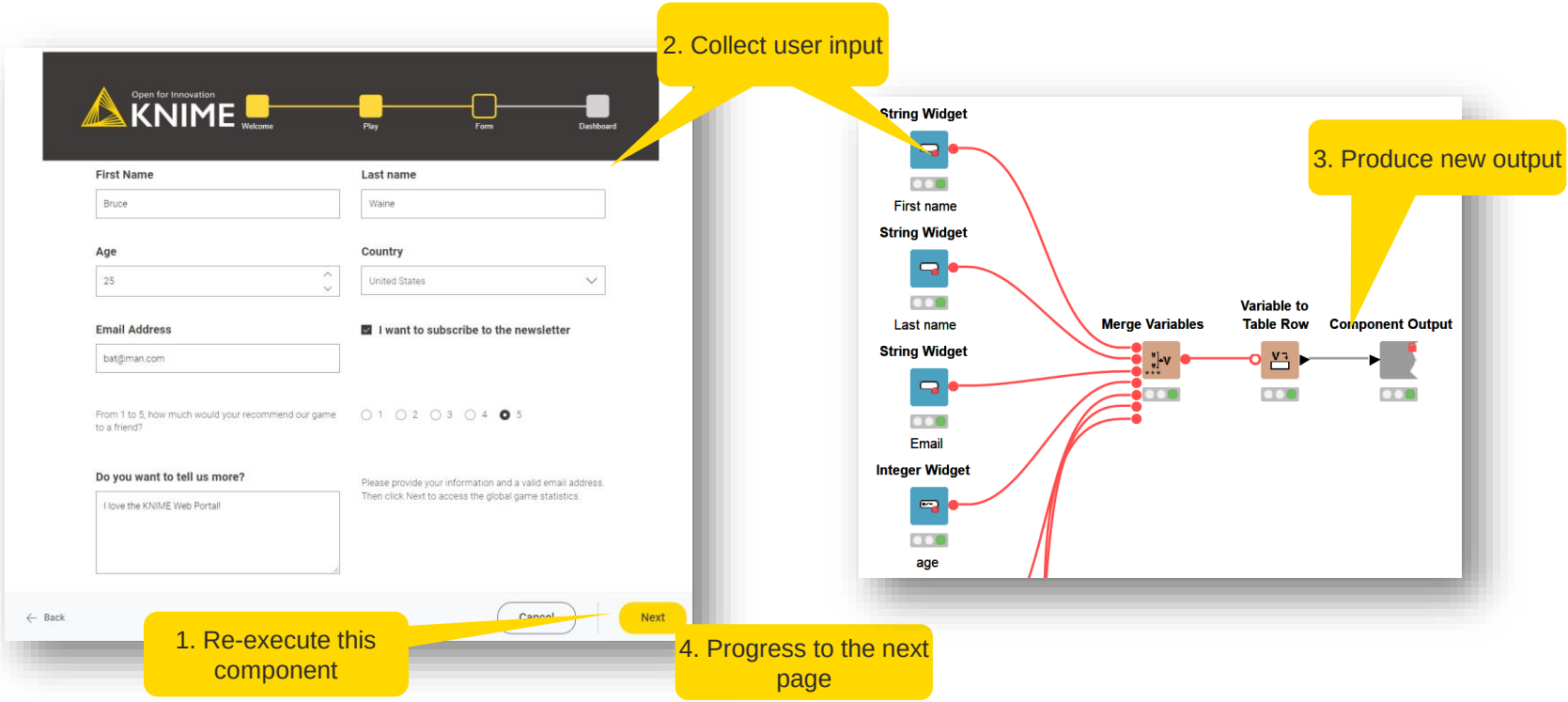
Keep **heavy computations outside** the components to avoid long execution time



e.g. data writing, model retraining etc..

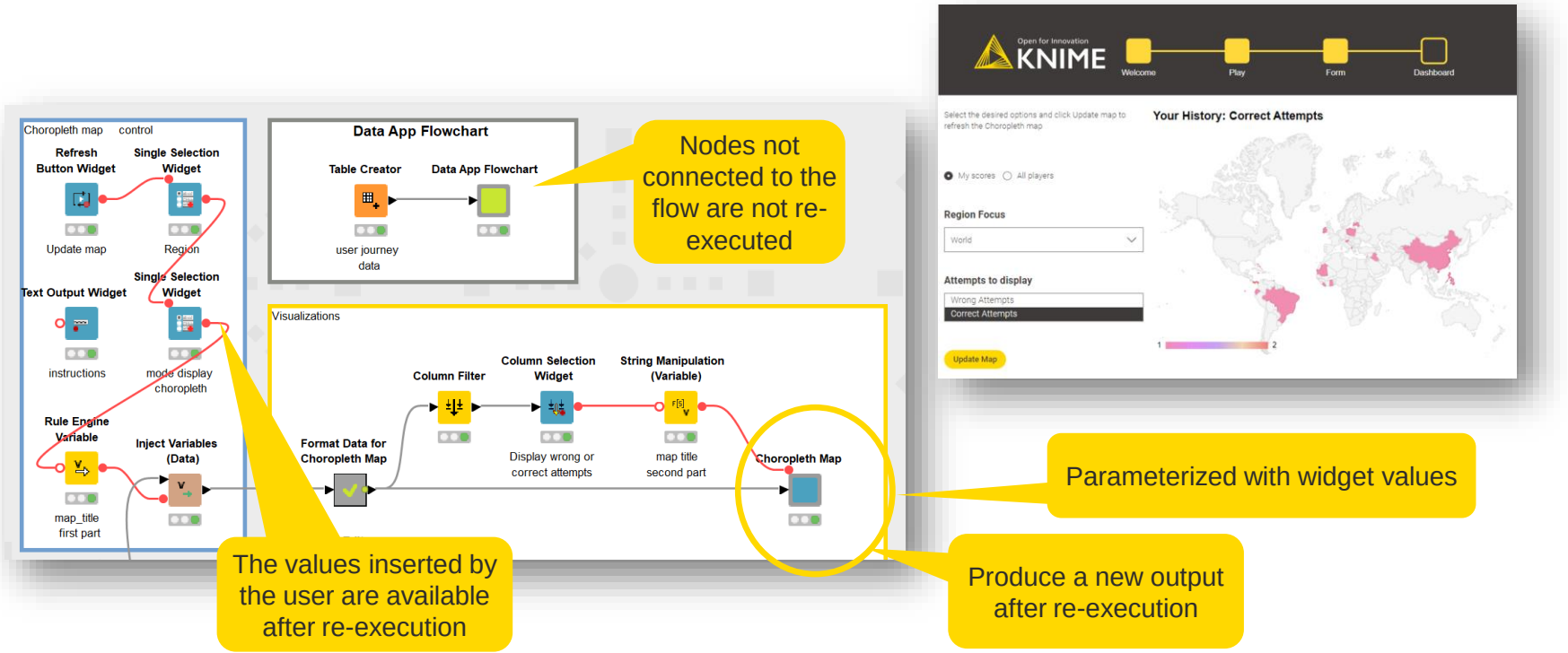
Motivations for re-execution

- Re-execution enables input collection
- The new input can be transmitted to the following page



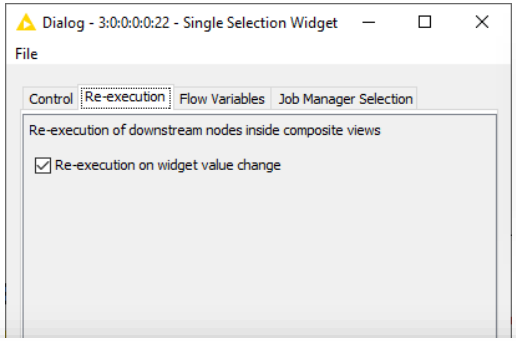
Refresh Button Widget

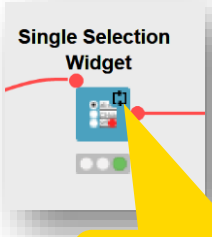
- Re-execute downstream nodes
- Remain in the same page (unlike the “Next” button)
- Get user input (e.g. from widget nodes) and send it downstream



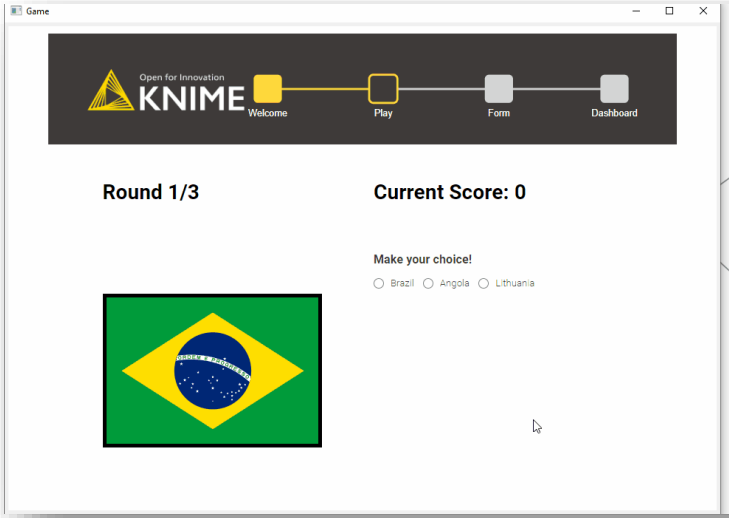
Widget automatic re-execution

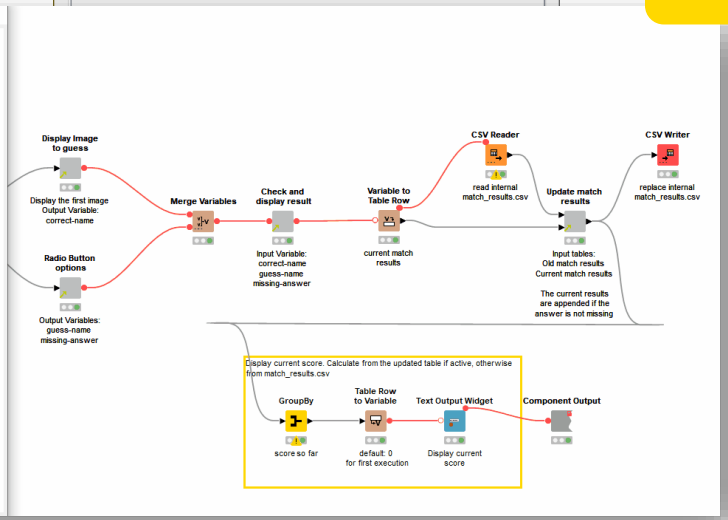
- Re-execute downstream nodes on input change
- No need for a Refresh Button





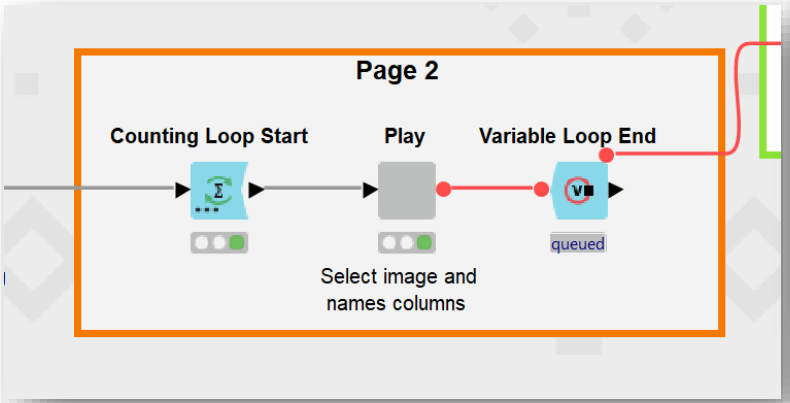
Notice the icon on the node when re-execution is enabled

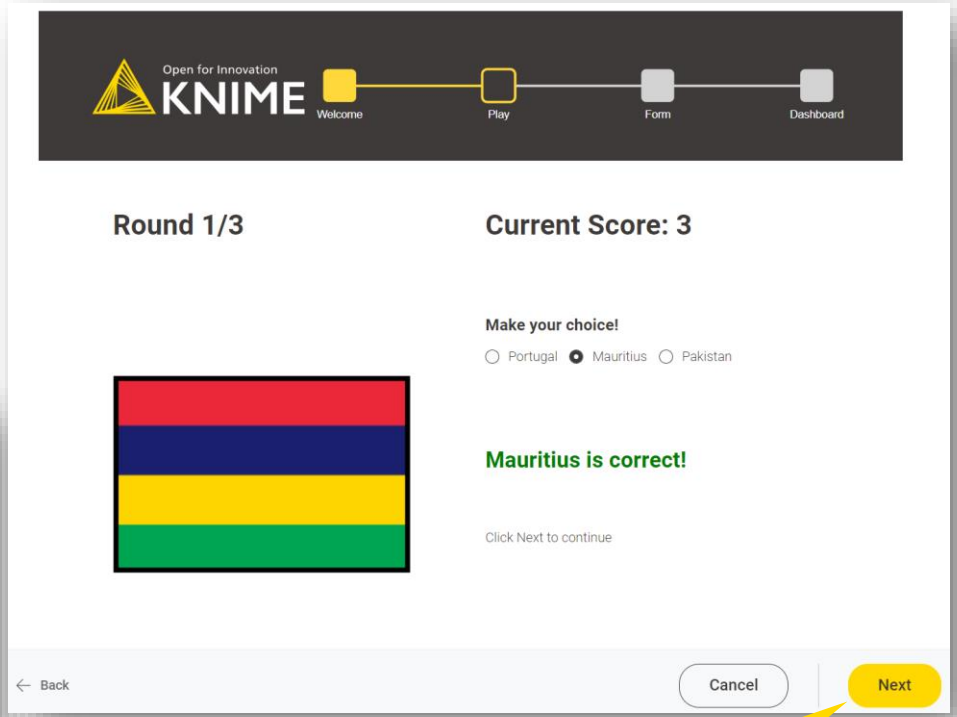




Repeating page with loops

- The same WebPortal page can be shown several times if the component is placed between loop nodes

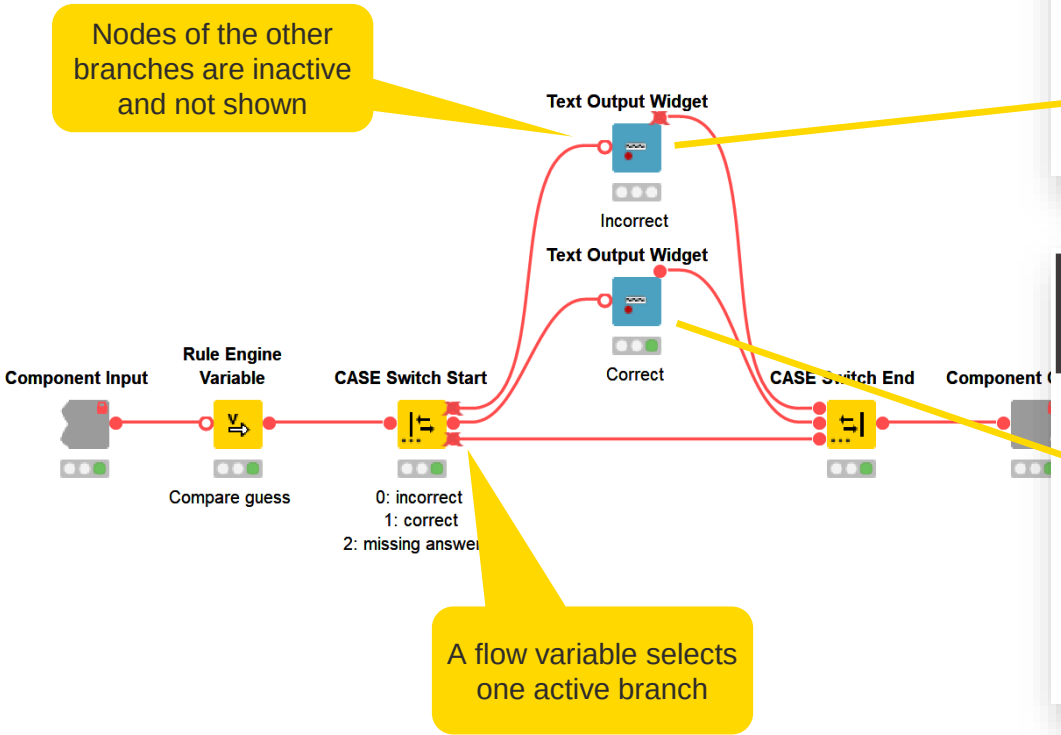


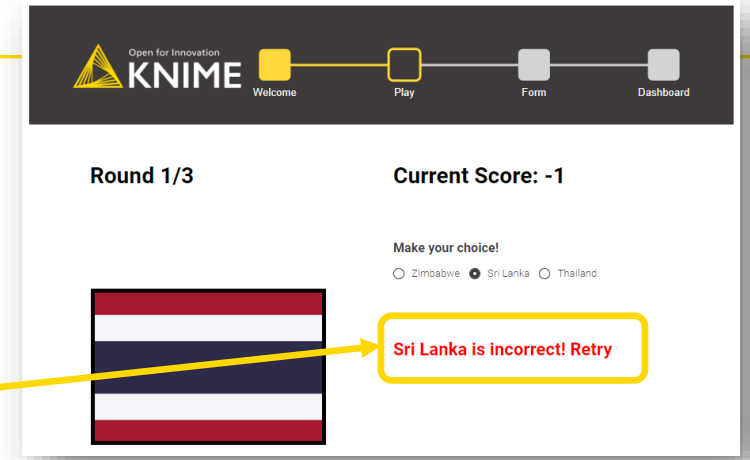


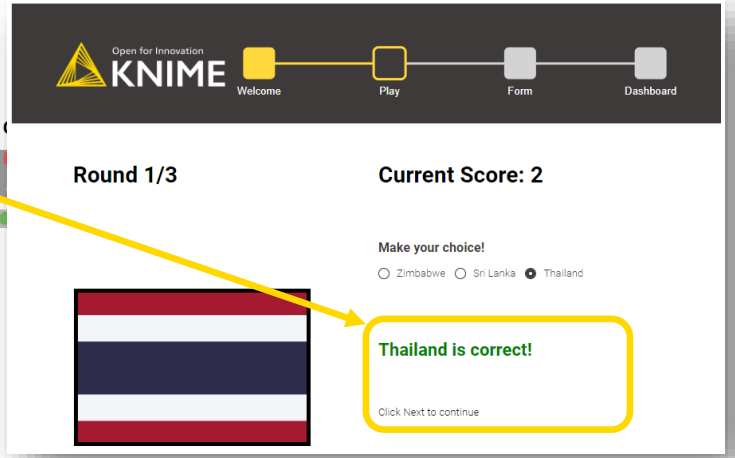
The “Next“ button is the loop iterator

Conditional Views

- Use Switch nodes **inside a component** to dynamically activate certain views

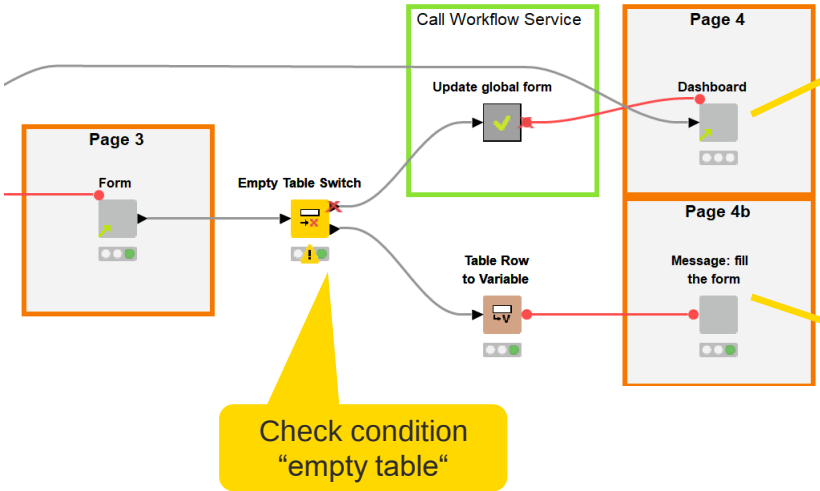


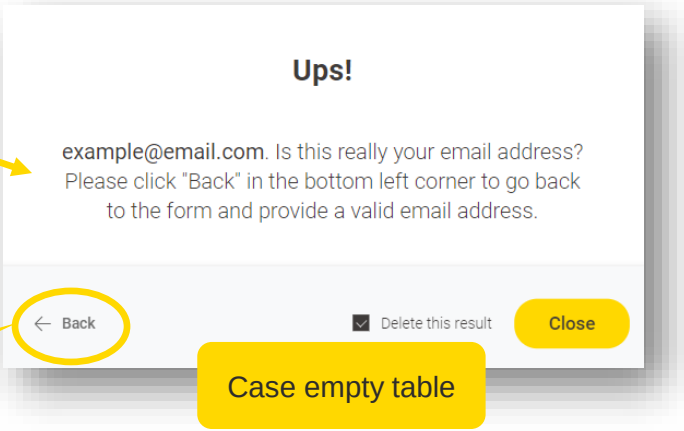




Conditional Path and “Back” button

- Use Switch nodes **in the top-level workflow** to show a different page during data app navigation





WebPortal features recap

- WebPortal paging
- Data App Flowchart

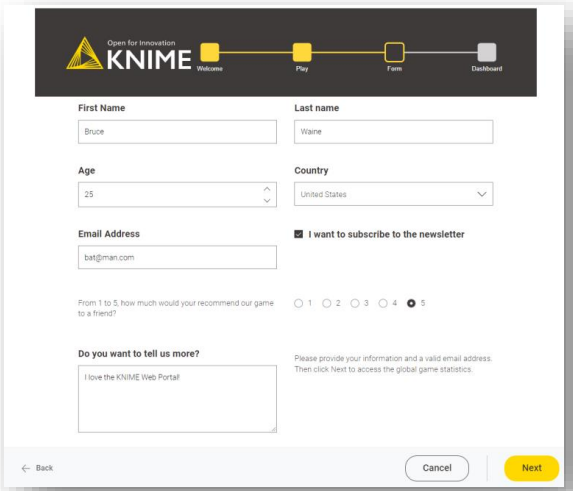
- 3 ways for re-execution
  - “Next” button
  - Refresh Button widget
  - Widgets automatic re-execution

- Repeating page with loops
- Conditional views
- Conditional paths

Guide through multiple pages

Collect and consume user input

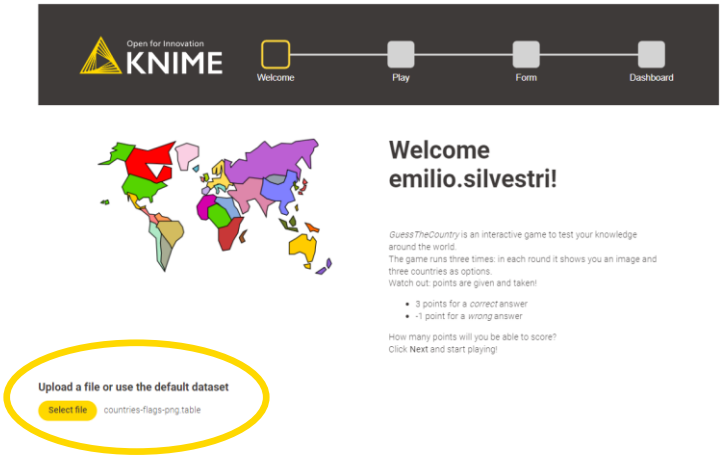
Personalize views and navigation

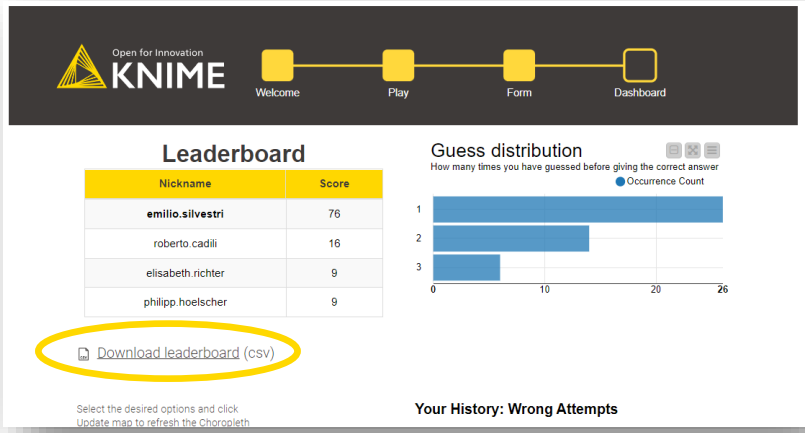


Upload and Download files via WebPortal

Upload and Download files

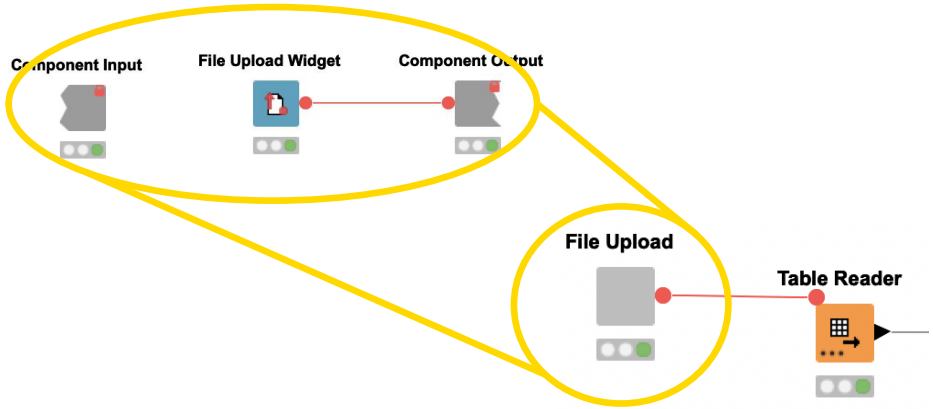
- So far, user interaction with the Data App limited to clicks and form inputs
- How can the user interact with files?





File Upload

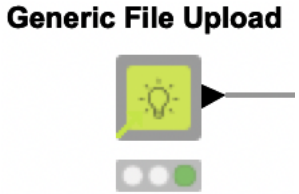
- File Upload Widget: enables the user to select a file, which is uploaded to the Server and the Path Flow Variable is created
- Reader Nodes: Path Flow Variable is passed to the Reader node to read the data into the workflow



IMPORTANT: User can only upload file types that match the selected Reader node

Verified Component: Generic File Upload

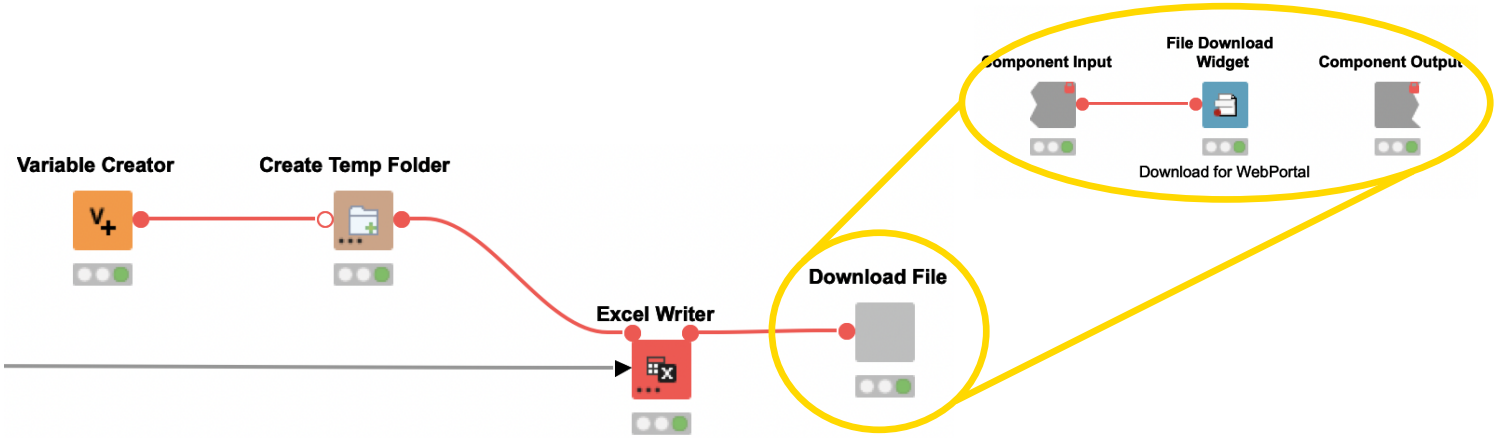
- This component can be used as a generic file upload for the KNIME WebPortal or locally in the KNIME Analytics Platform.
- The user can upload a file of different formats (.csv, .tsv, .xls, .xlsx and KNIME-native .table) and the component selects the correct node to read the data in.



IMPORTANT: This component does not cover all edge-cases for complicated file formats.

File Download

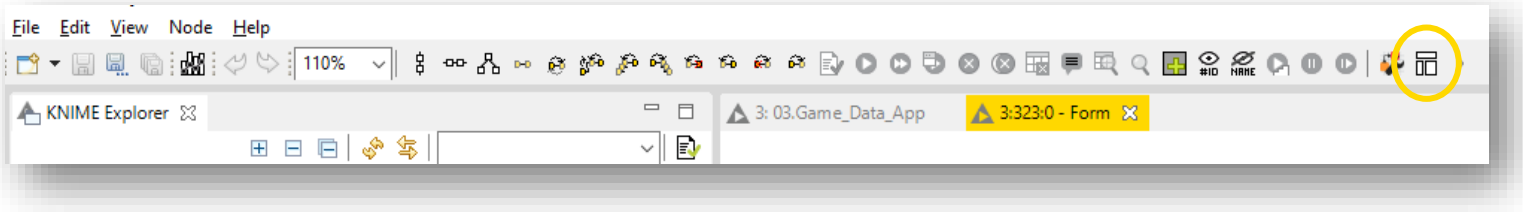
- Variable Creator: create name of file
- Create Temp Folder: create temporary folder, which is deleted with reset of workflow (Pro-Tip: use temporary folder on Server to avoid pile up of files)
- Excel Writer: write data into file in temporary folder
- File Download Widget: show a download button



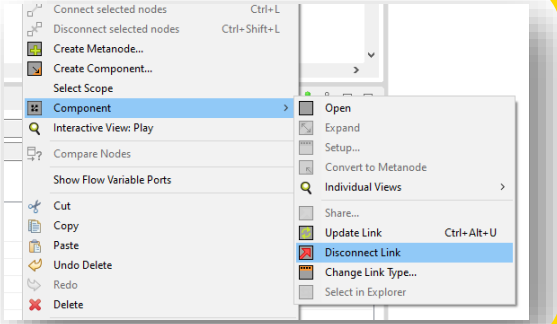
Custom Styling of WebPortal and Data Apps

Node Usage and Layout

- Customize the composite view of a component (i.e., a WebPortal page)
- Open a component and click the layout toolbar icon



- Remark: you can't change the layout of a shared component from a workflow that uses it. Shared components are read-only
- Disconnect the shared component first or use the Component editor to edit the original shared component

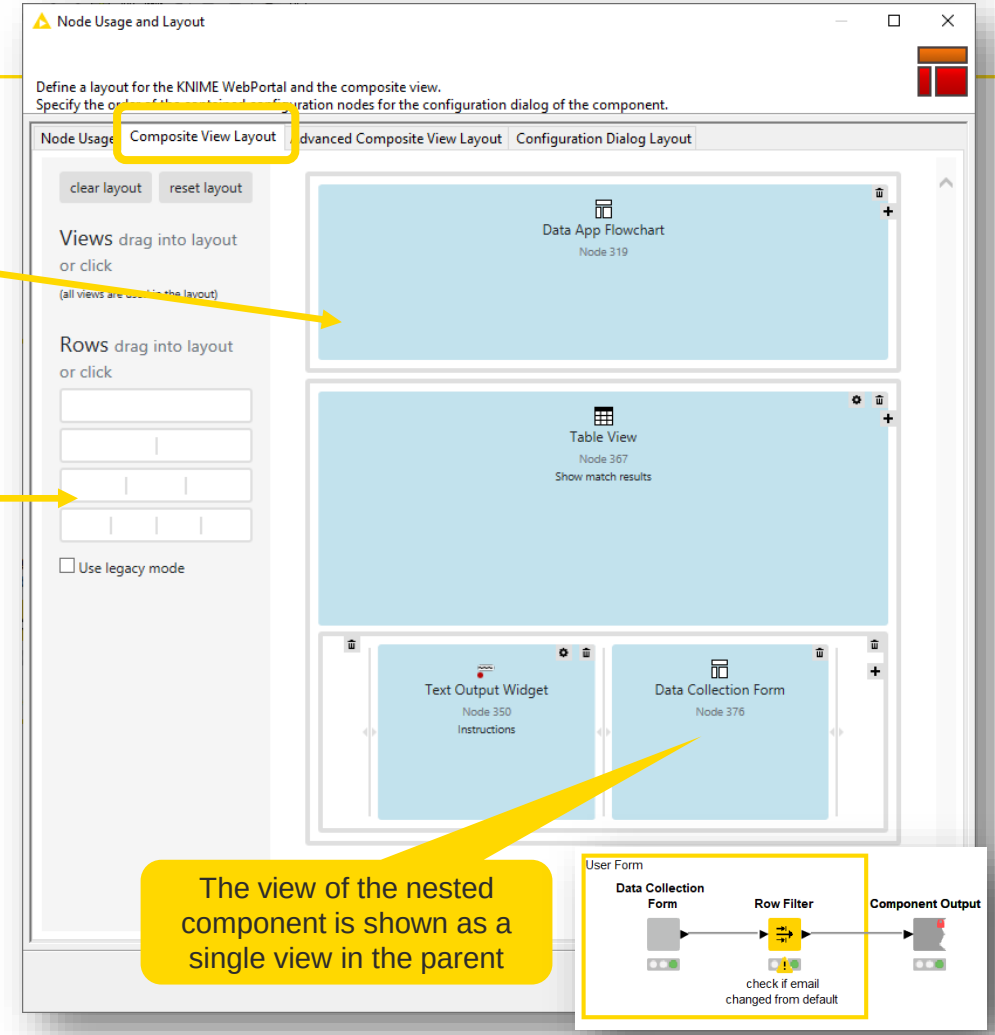


Composite View Layout

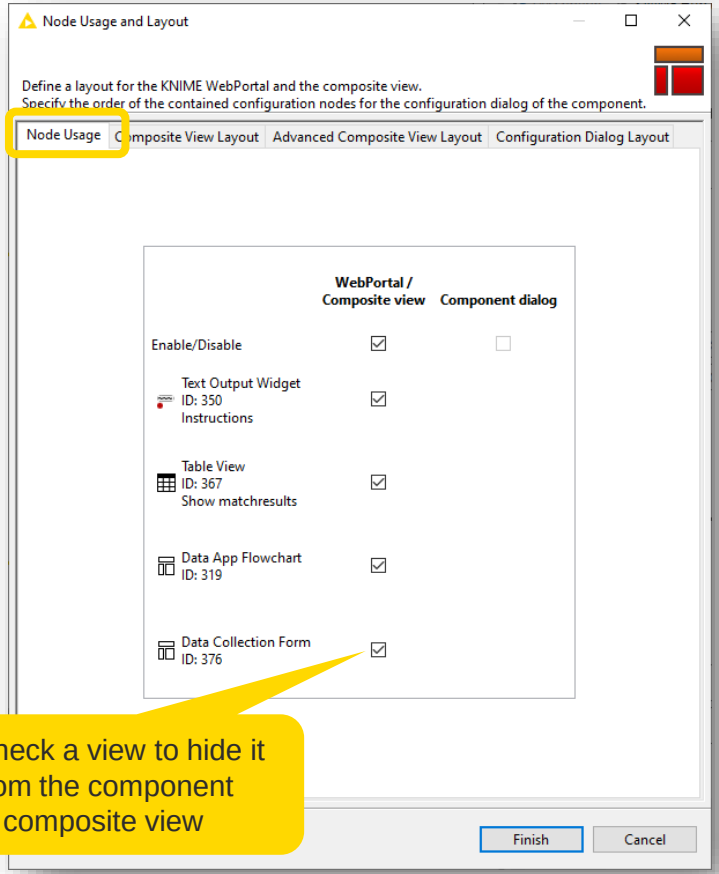
- Edit the composite view by rearranging the single views
- Group and organize views into rows and columns

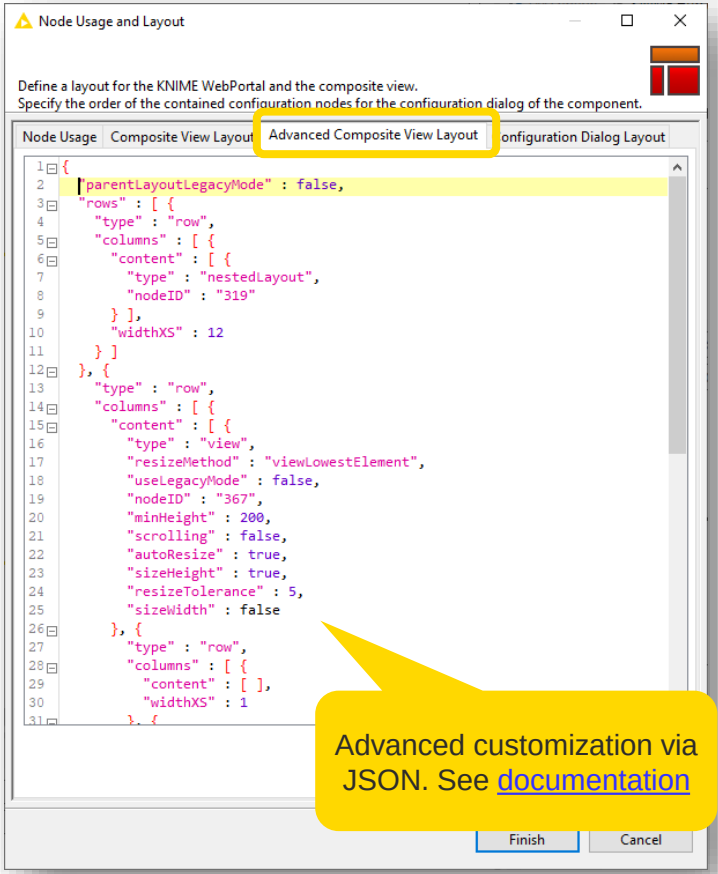
What produces a “view”?

- View nodes
- Widget nodes
- **Nested components**



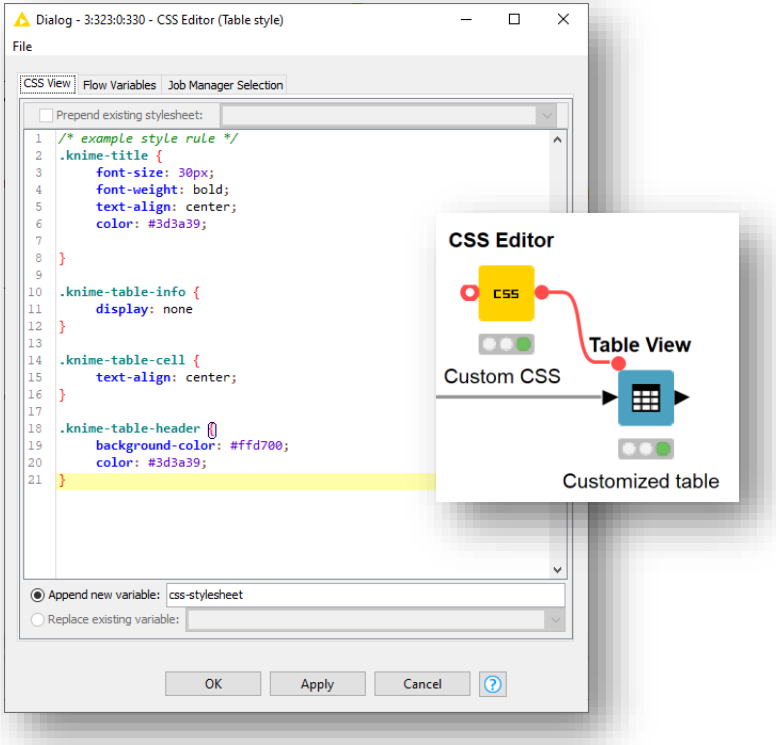
Node Usage and Advanced Composite View Layout

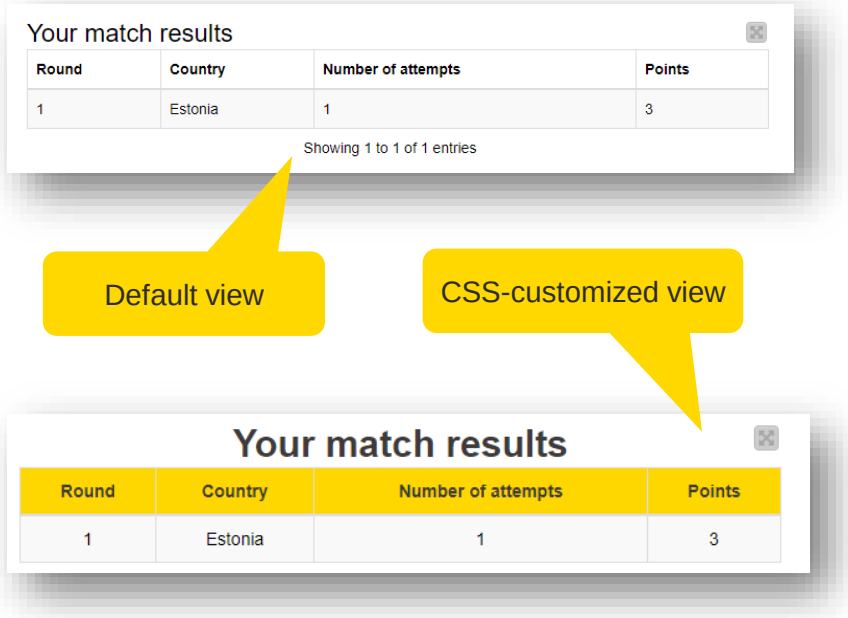




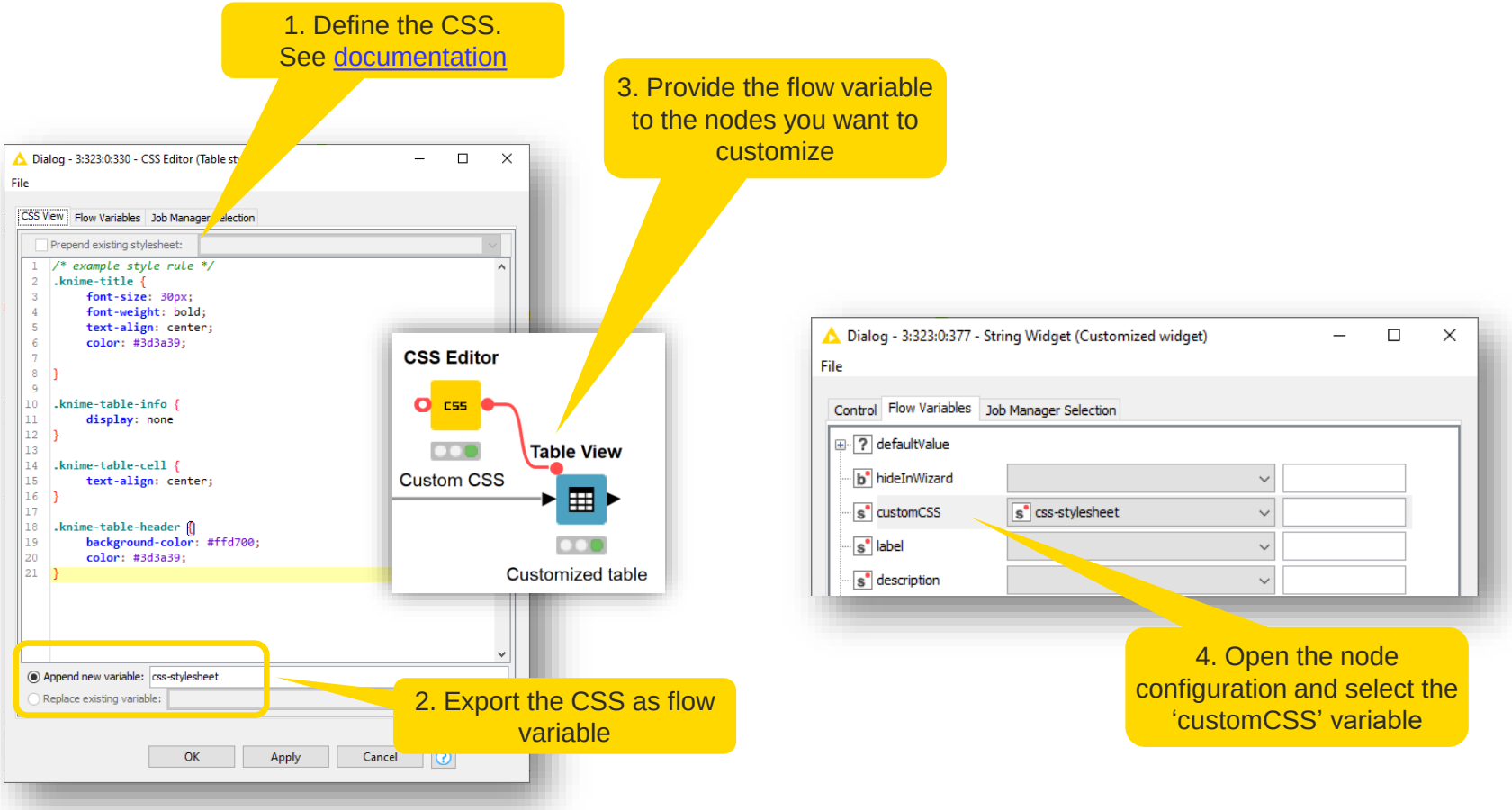
CSS Editor

- Create custom CSS styling of JavaScript views and Widgets
- The same style can be reused in multiple views





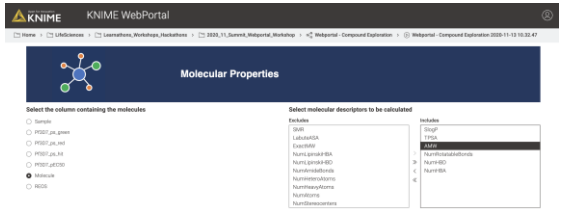
CSS Editor usage

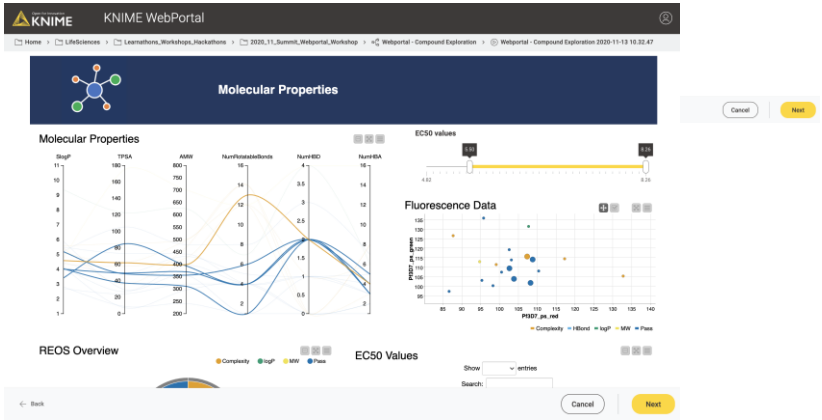


Theming

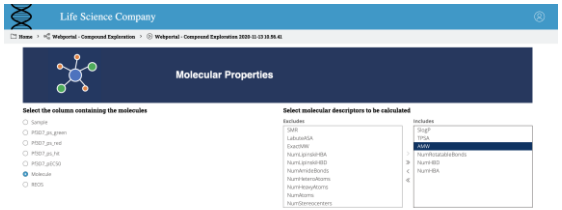
- Apply a customized theme to the WebPortal (see Documentation)
- Admin rights necessary

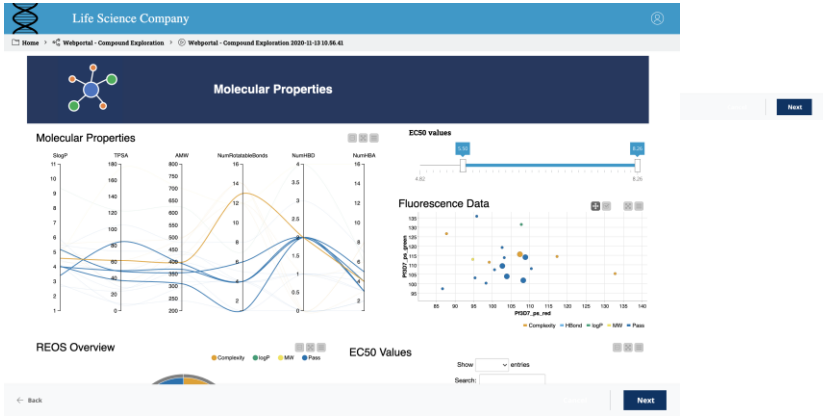
Default Theme



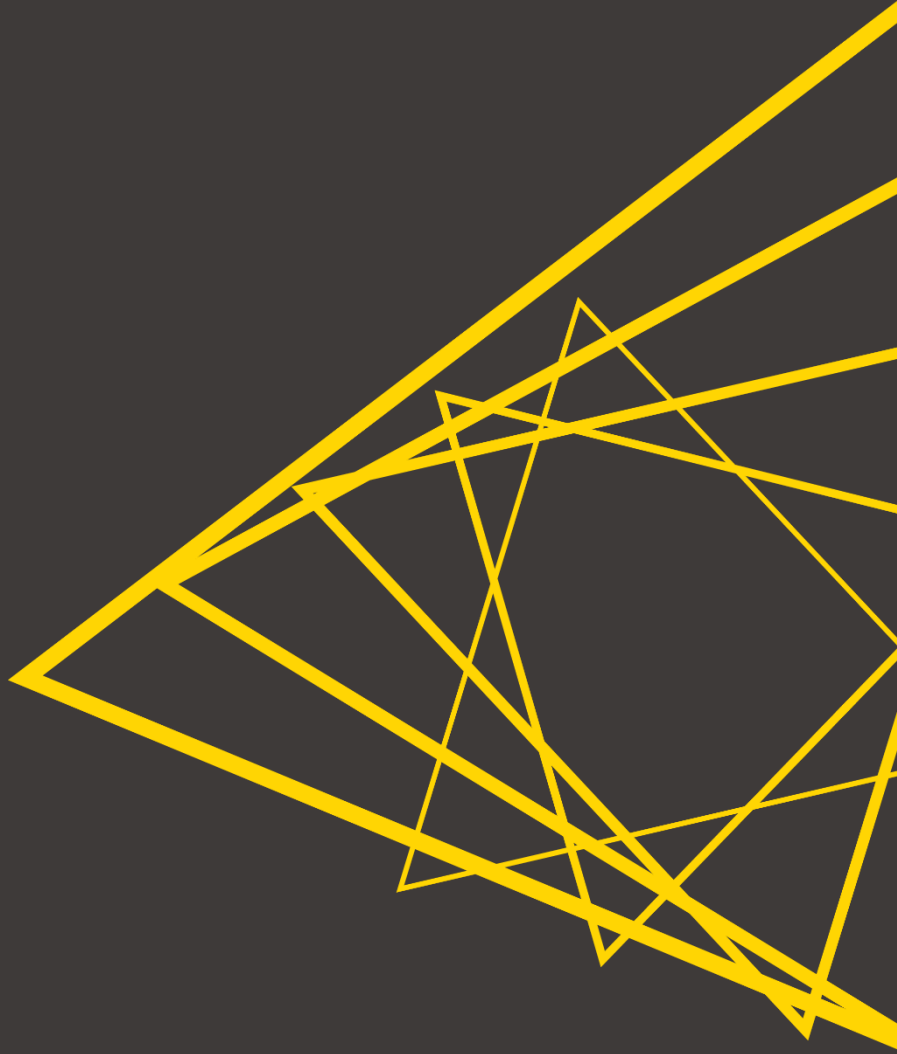


Customized Theme

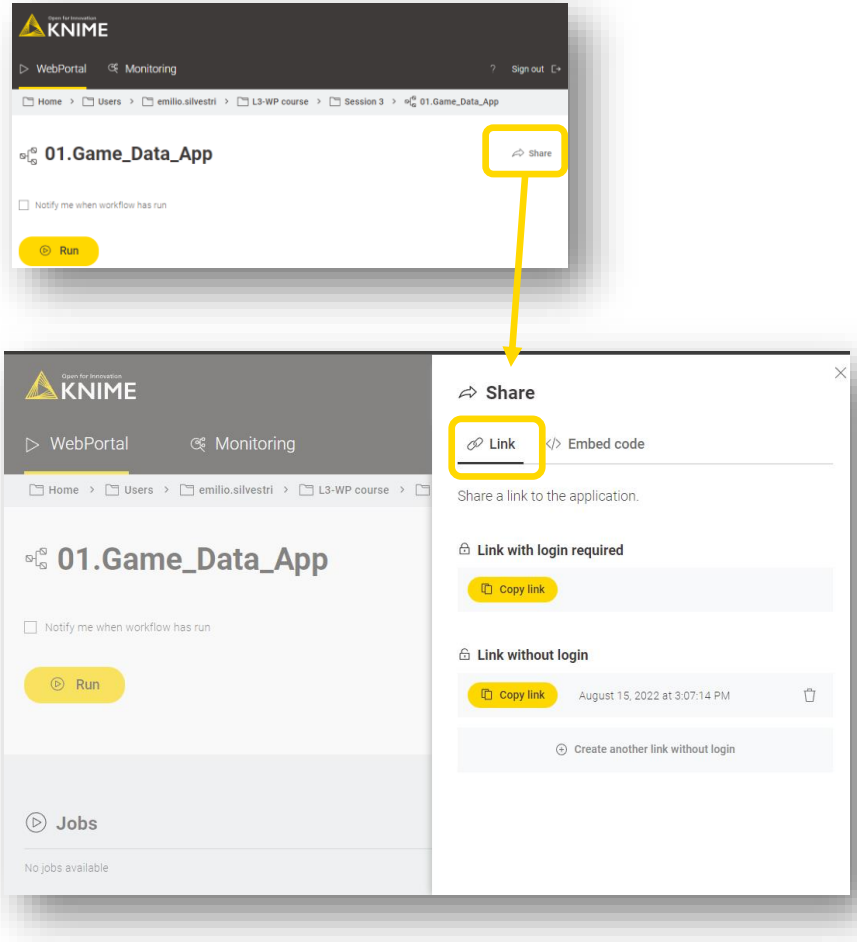




Sharing Data Apps



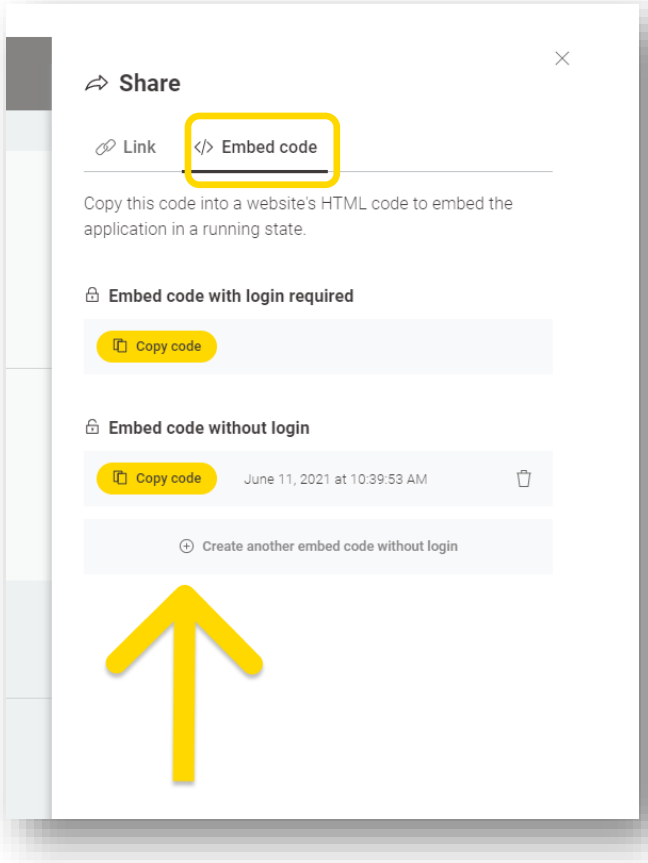
Share link to Data App



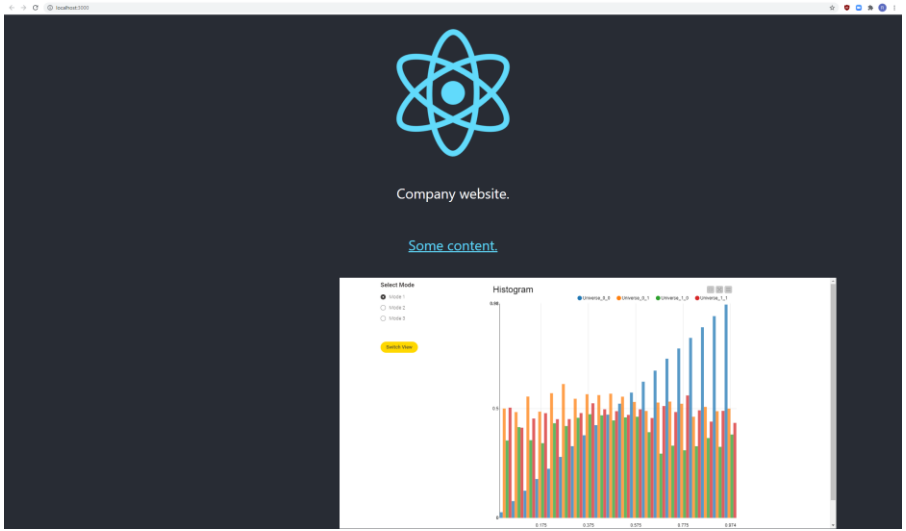
- Generate a link to the Data App directly from the WebPortal
- Possibility to generate a link that does not require user authentication
- The links act as access token

- Important: **use with caution!**
  - Everyone with the link will be able to execute the workflow.
  - The jobs started with a token link are associated to the user who generated the token

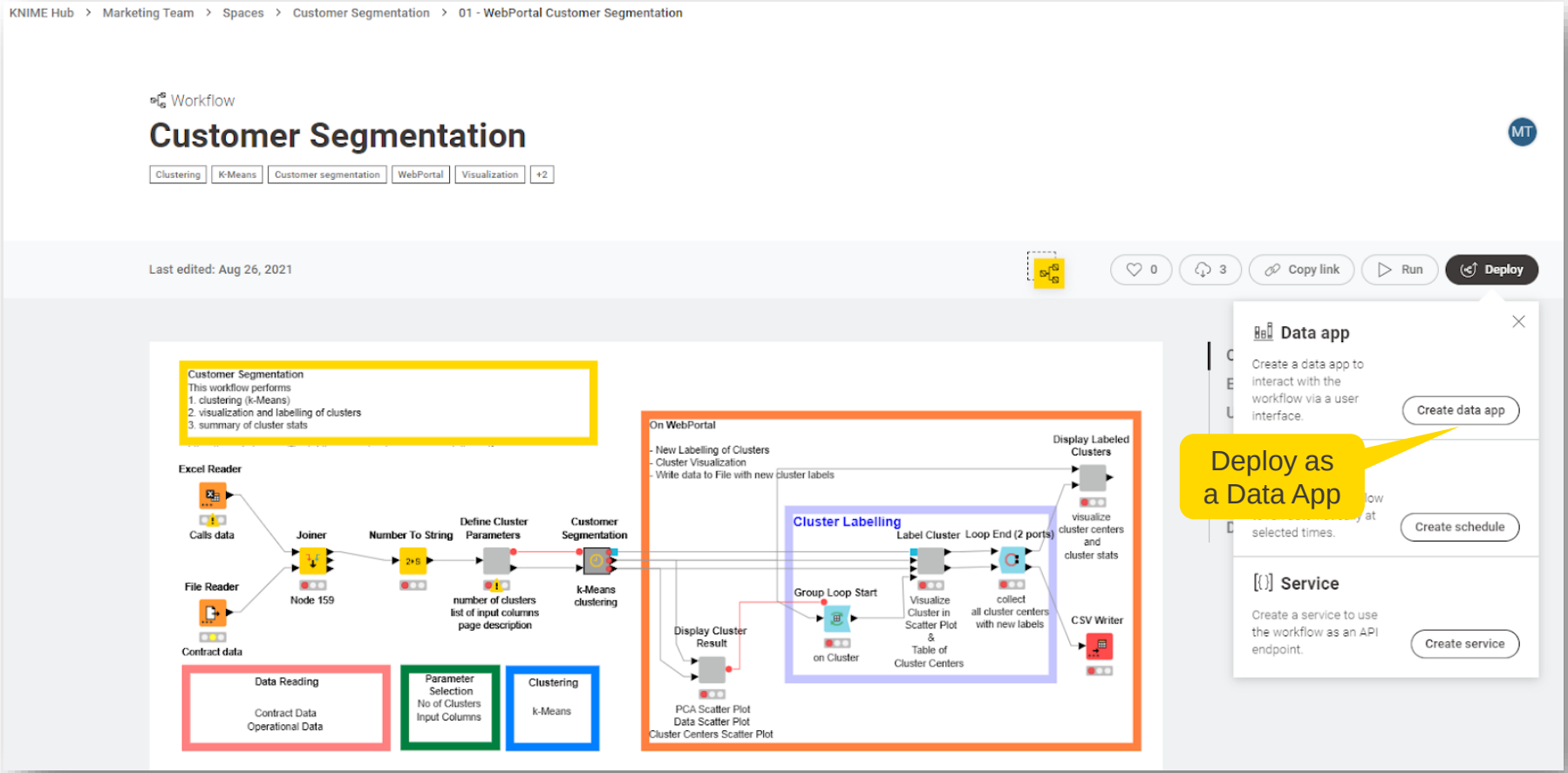
Embed Data App



- Embed Data App HTML code in an iframe on any website
- As with links, choose if login required or not

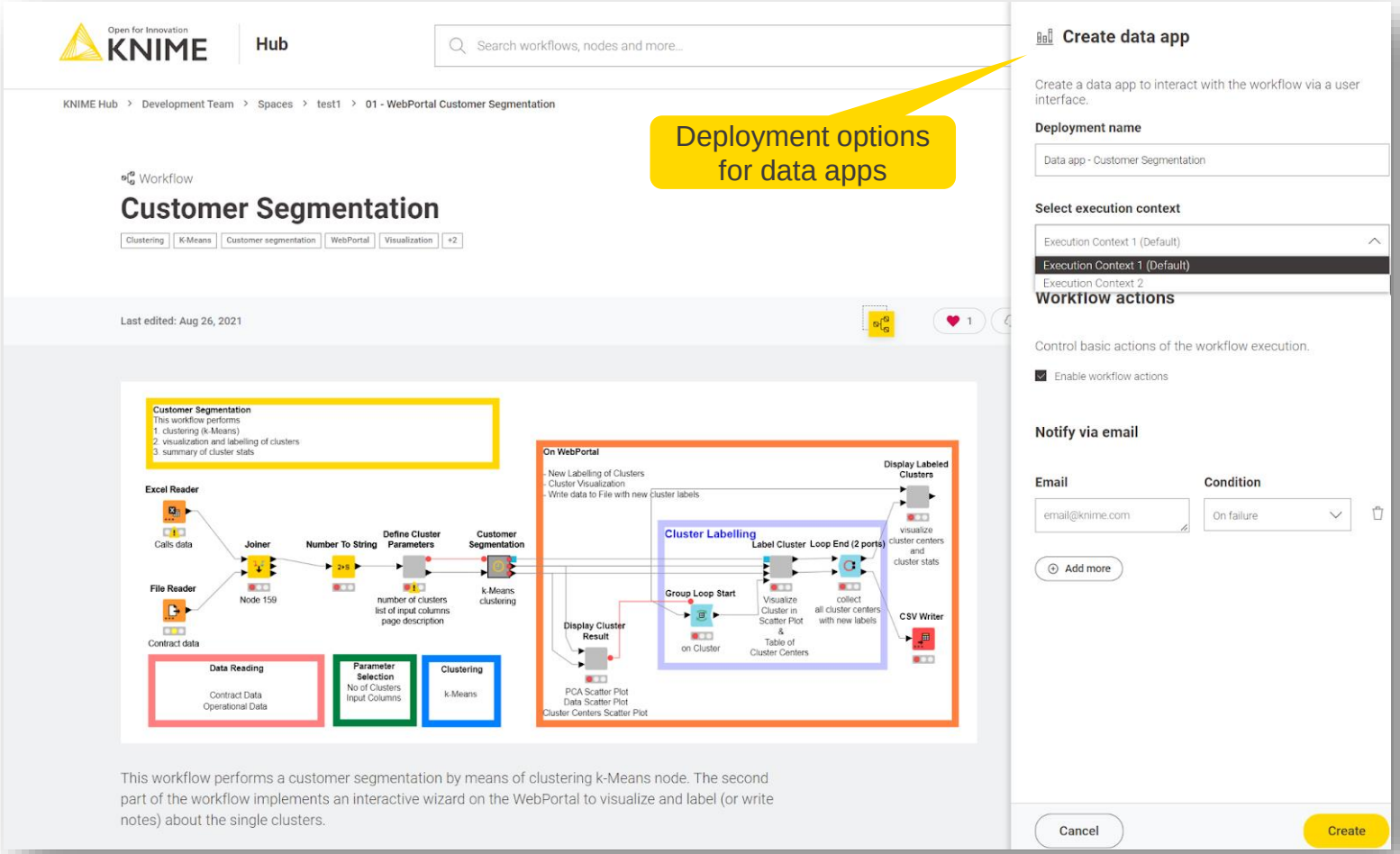


Preview: Data Apps in KNIME Business Hub



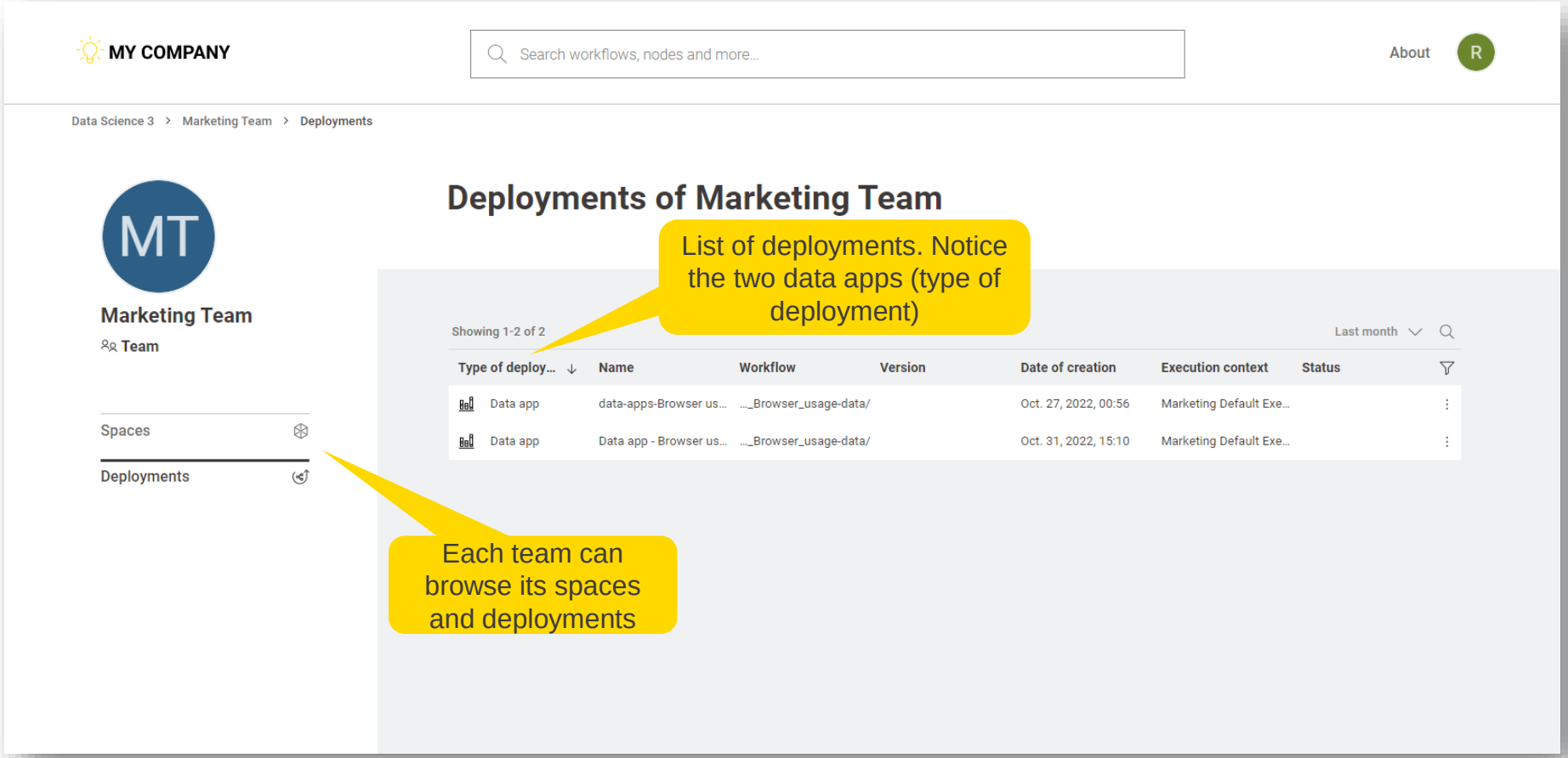
Open for Innovation KNIME

Preview: Data Apps in KNIME Business Hub



Open for Innovation KNIME

Preview: Data Apps in KNIME Business Hub

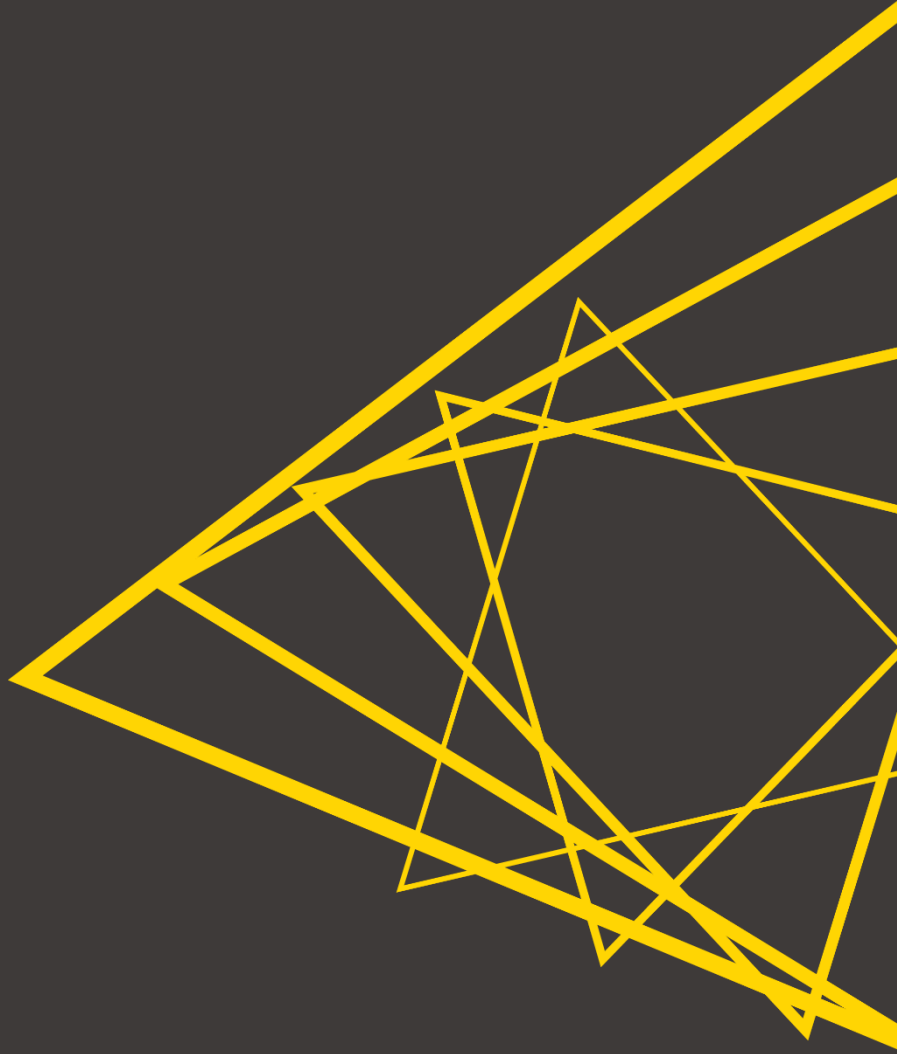


Session 3: Summary

Now you should be able to:

- Define the concepts of WebPortal and Data App
- Identify and apply the main WebPortal functionalities
- List and use the tools to customize a Data App
- Outline the steps to deploy and share a Data App

Exercises



Exercises

- **Exercise 01** – Build a simplified, standalone version of the Game Data App. Test your knowledge on
  - CSS customization
  - The Data App Flowchart verified component
  - Widgets
  - Composite View Layout
  - Conditional path
- Deploy the Data App and play on KNIME WebPortal.

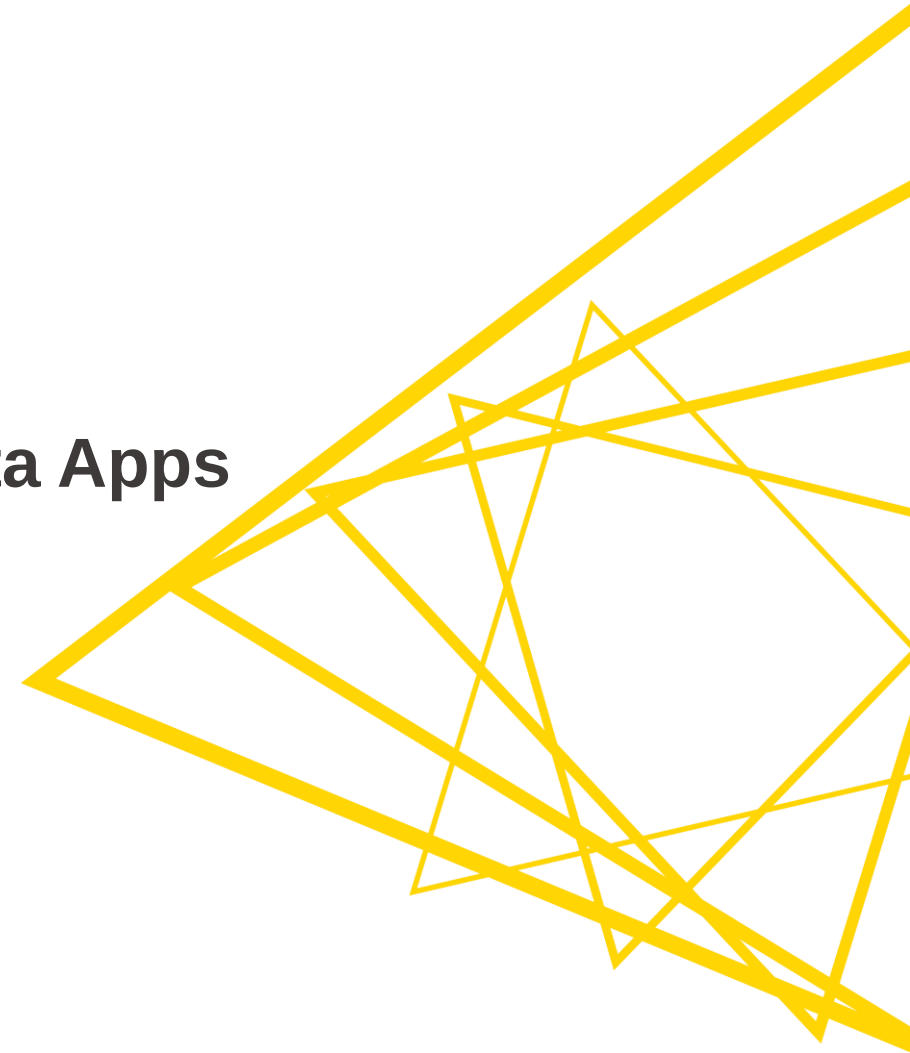
Open for Innovation
KNIME

Thank You!

Open for Innovation
KNIME

[L3-WP] Productionizing Data Apps

KNIME GmbH



Structure of the Course

Session	Topic	Duration
Session 1	What happens after the prototype is ready for deployment?	75 min
Session 2	Introduction to KNIME Server	75 min
Session 3	Deploying WebPortal Data Apps	75 min
Session 4	**Performance Optimization, orchestration, error handling, and KNIME Edge**	75 min
Session 5	Wrap-up Session	15 min

Structure of each session

- Discussion of past exercises
- Course
- Introduction of next exercises

Exercises Session 3

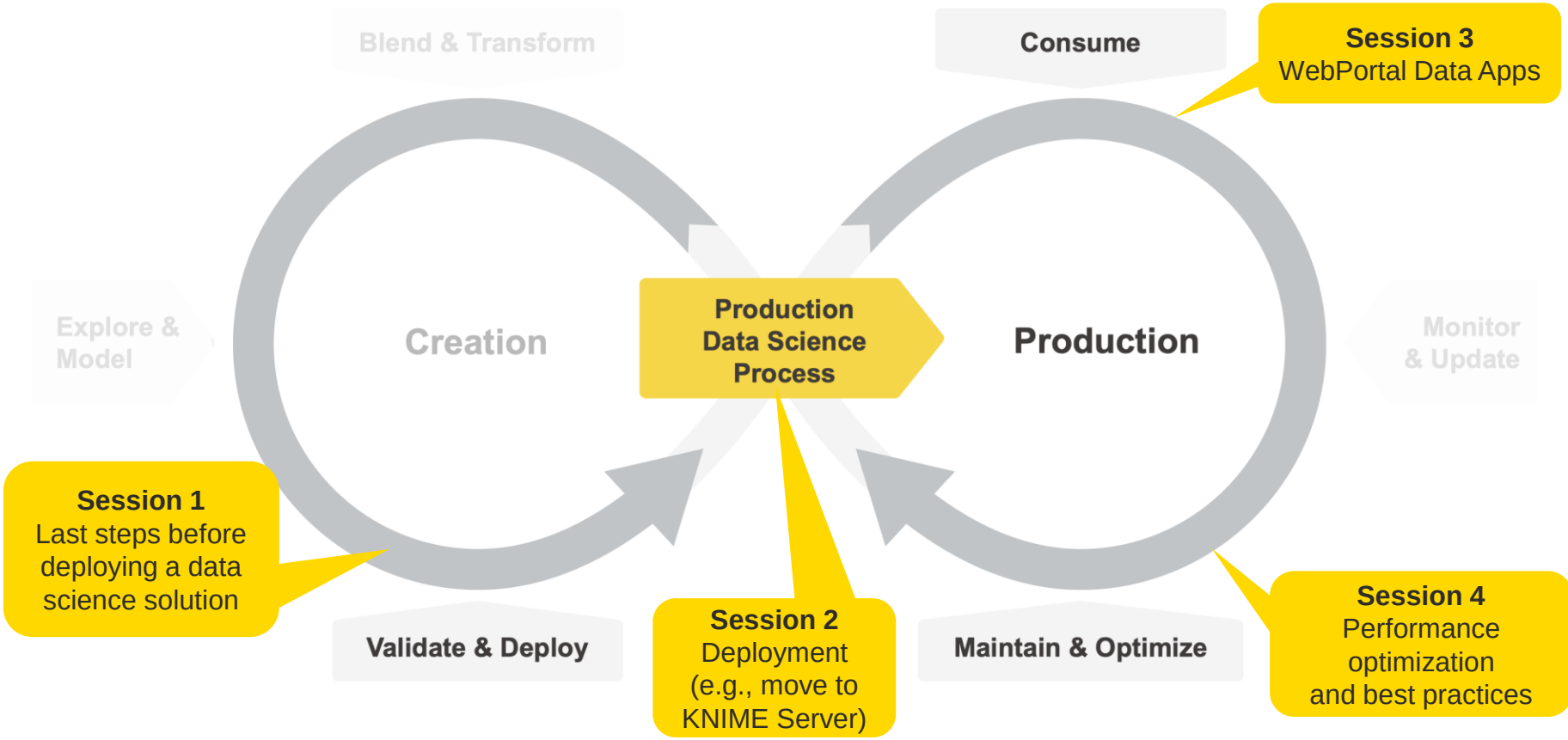
- **Exercise 01** – Build a simplified, standalone version of the Game Data App. Test your knowledge on
  - CSS customization
  - The Data App Flowchart verified component
  - Widgets
  - Composite View Layout
  - Conditional path
- Deploy the Data App and play on KNIME WebPortal

Open for Innovation
KNIME

Session 4

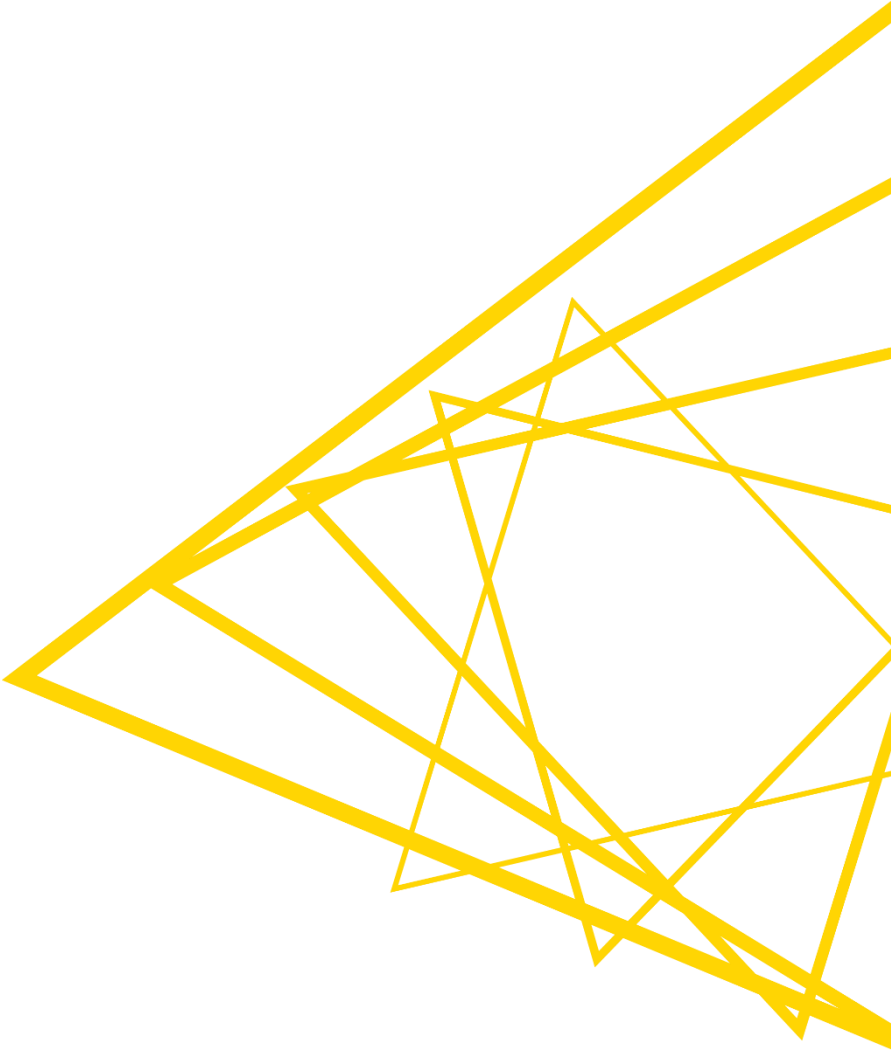
Performance optimization, orchestration, error handling, and KNIME Edge

This Course's Sessions

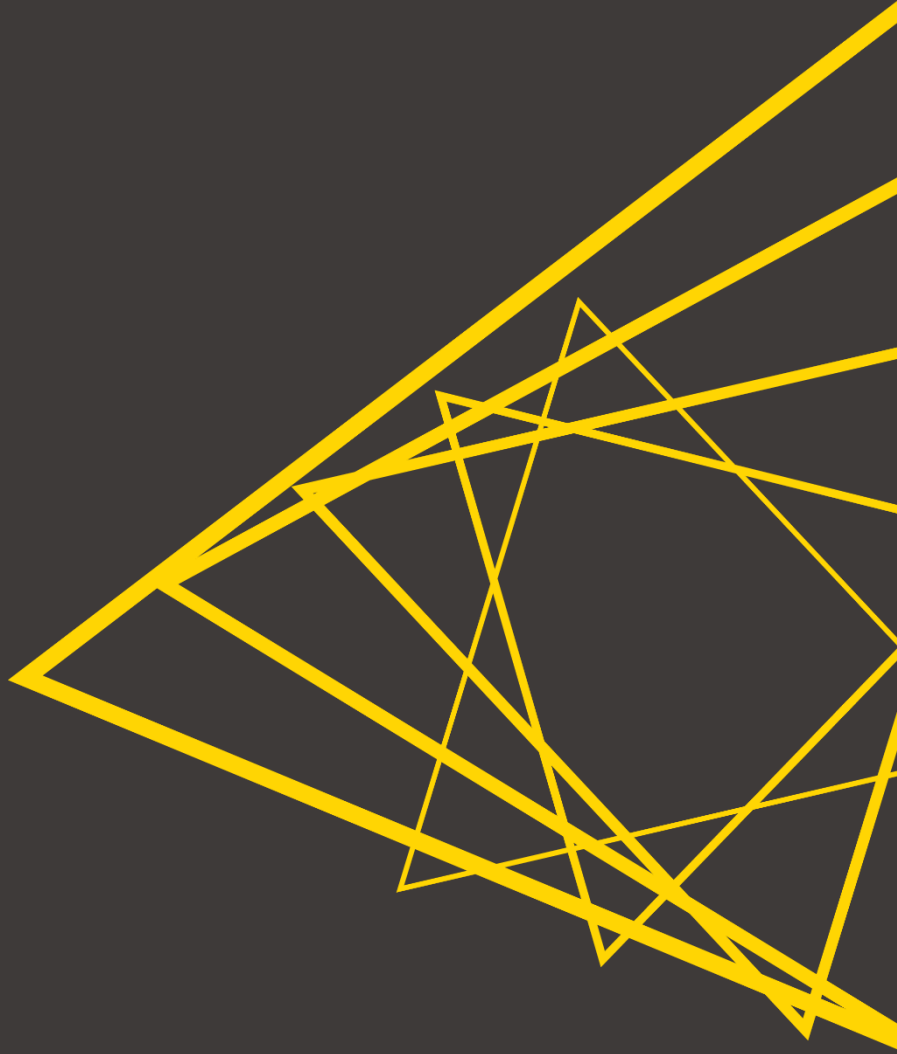


Learning Objectives

1. List the possible performance optimization operations
2. Recall and apply the tools for orchestration offered by KNIME Server
3. Identify the techniques for error handling



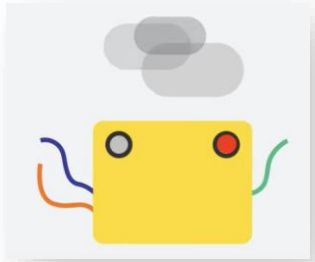
Performance Optimization



Motivation for Performance Optimization

- Will your deployed workflow execute with good performance?
  - Performance matters for better interactivity (data apps) and to reduce the chances of experiencing time out issues (e.g., with web services)
- Stress test your deployed workflow before going public
  - Feed it with large amounts of data (e.g., standalone applications)
  - Simulate interaction with many concurrent users (e.g., data apps)
  - Simulate interaction with many requests (e.g., web services)
- If workflow crashes or takes too long to execute, optimize its performance
  - Modify the workflow explicitly for better runtime
  - Move to a better infrastructure (e.g., a bigger KNIME Server instance)
  - Rely on KNIME Server's functionalities

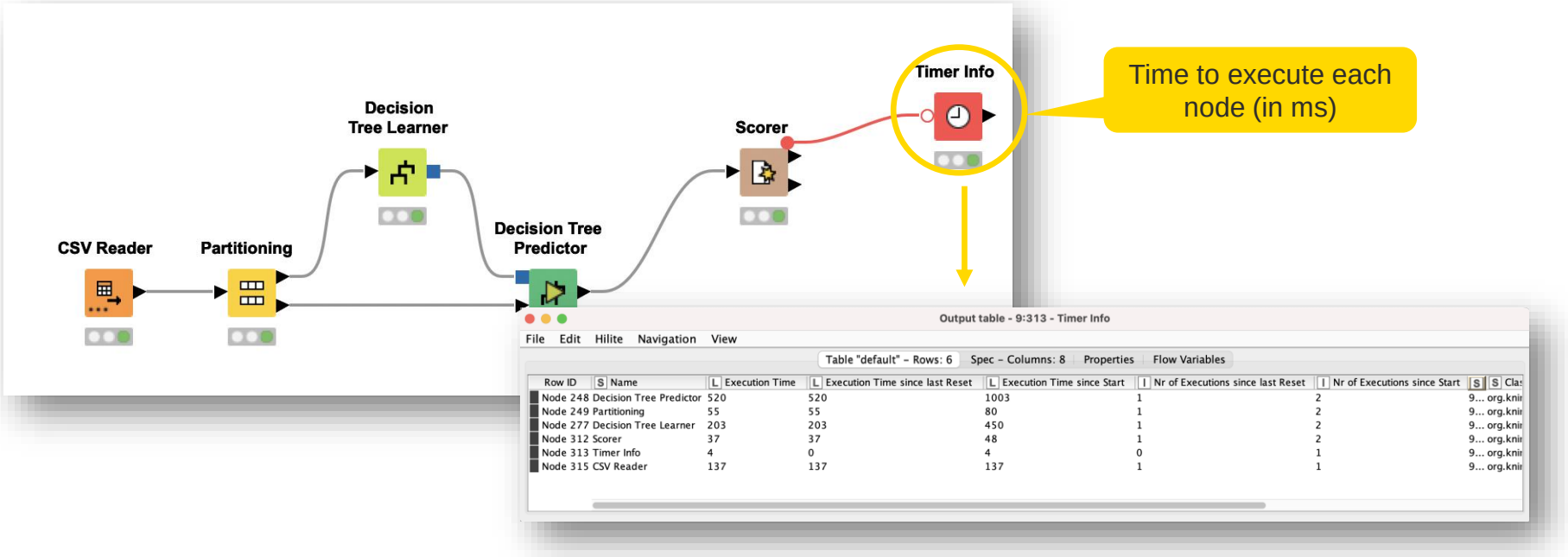
Large volumes of data →



Deployed workflow taking too long to run

Finding Performance Issues with the Timer Info Node

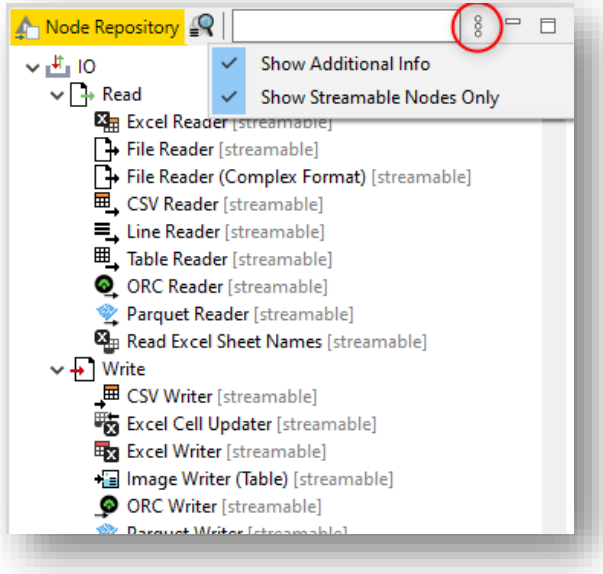
- The Timer Info node allows you to see which nodes in your workflow take the longest to execute
- It helps decide which parts of the workflow could benefit the most from some optimization



Streaming Execution for Better Performance

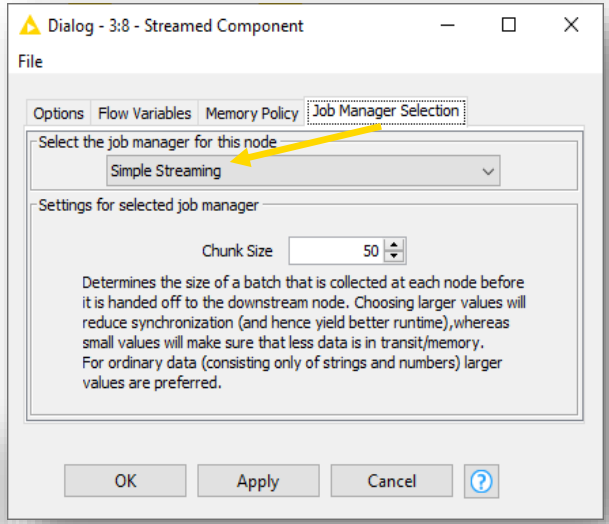
- Standard execution: node-by-node
  - Node processes all data, finishes, then passes data to next node etc.
- Streaming execution: nodes executed concurrently
  - Each node passes data to the next as soon as it is available, i.e., before node is fully executed
  - Faster execution, especially for reading/preprocessing data

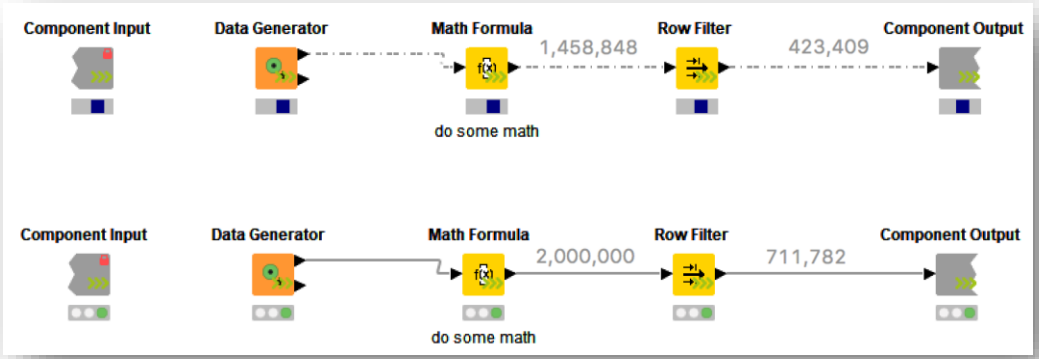
- Advantages
  - less I/O overhead and faster runtime
- Disadvantages
  - Intermediate results not available since nothing is cached
  - Available for a limited number of nodes



Streaming Execution for Better Performance

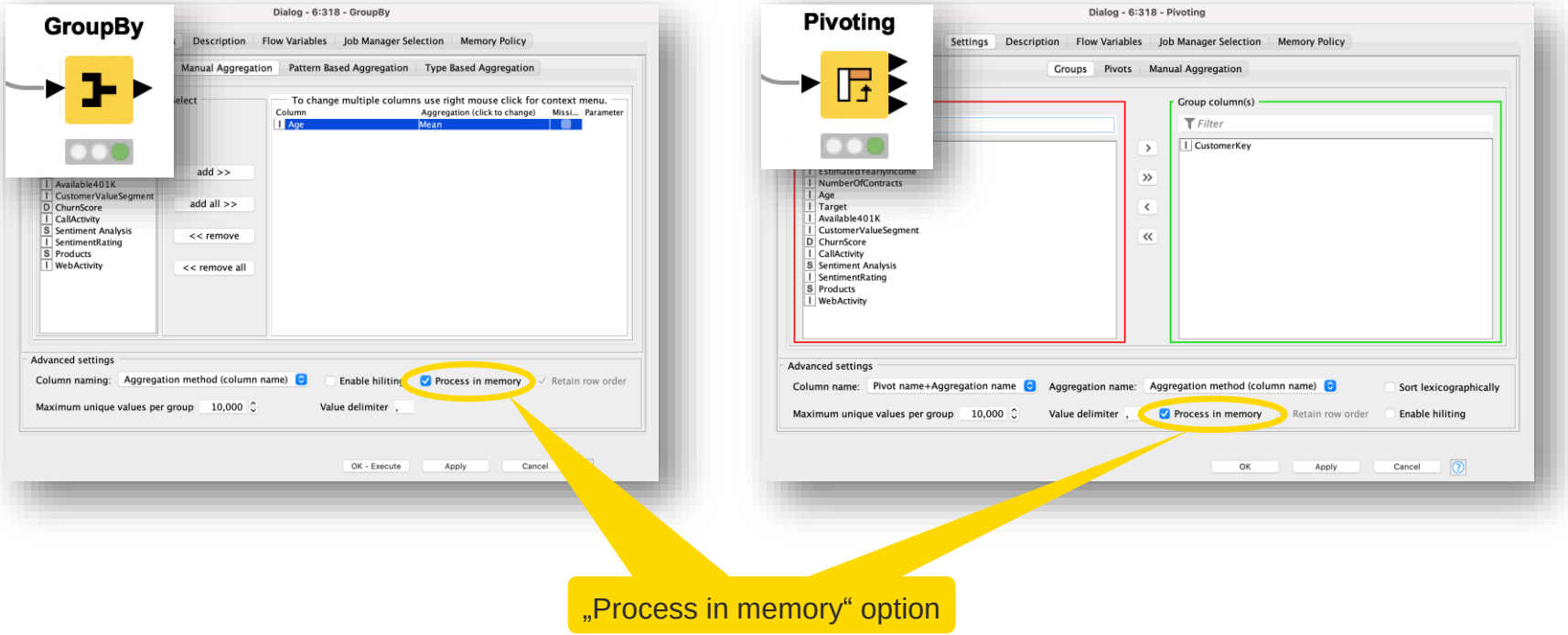
- Beta feature
  - Install KNIME Streaming Execution (Beta) extension
- Enable streaming execution in components
  - Create Component -> Configure -> Job Manager Selection -> Simple Streaming





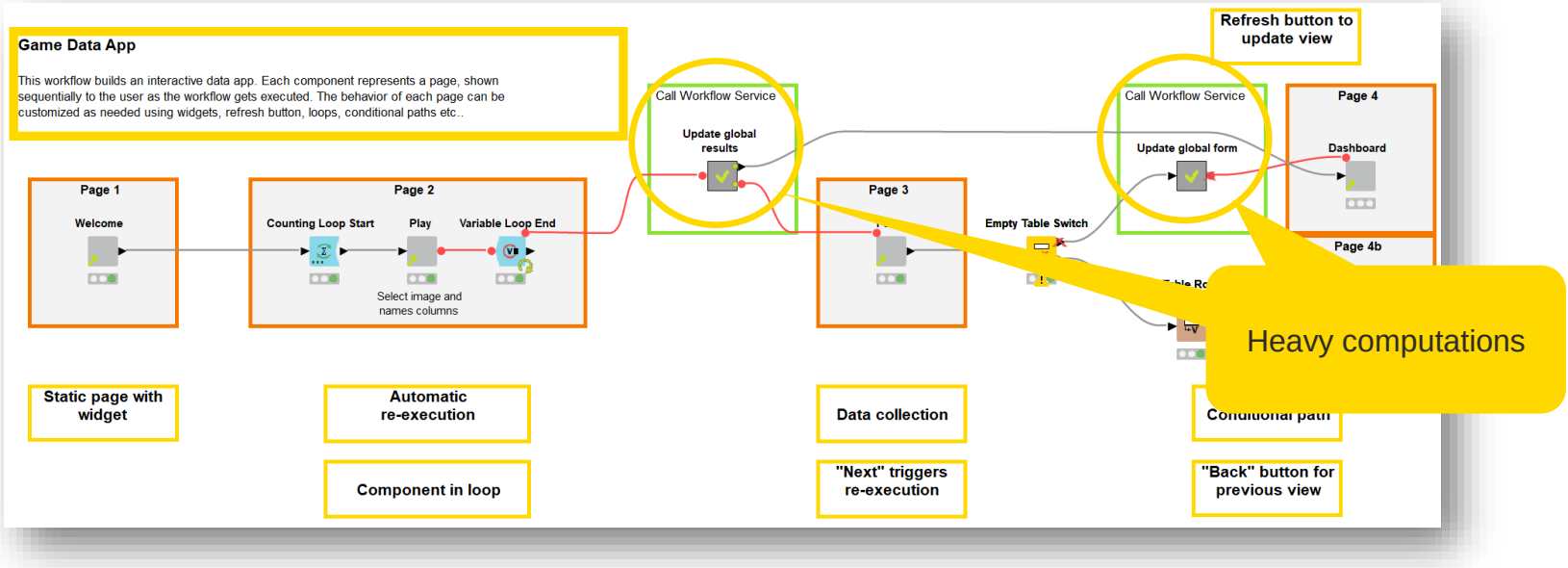
„Process in Memory“ Option

- Option for GroupBy and Pivoting nodes
  - The “Process in memory” option uses up a lot of memory, but it brings a significative performance improvement by reducing swaps to disk



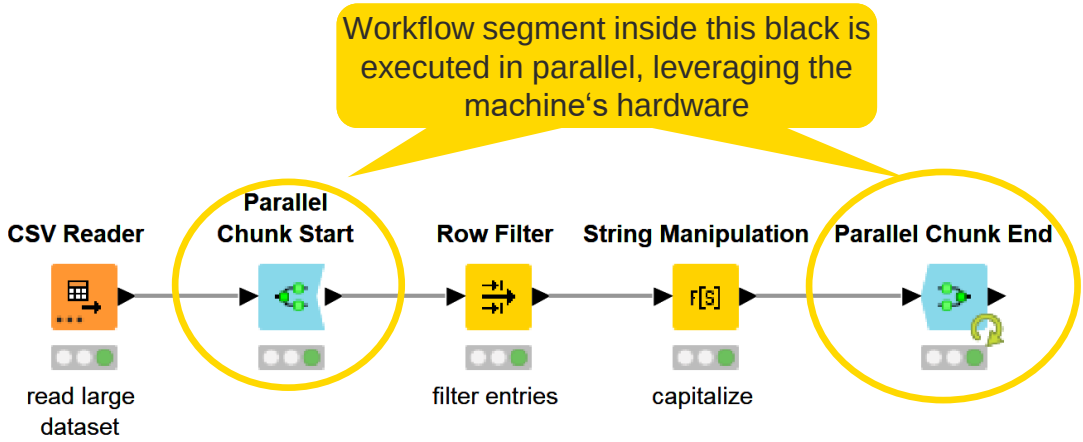
Data Apps: Heavy Computations Outside of Components

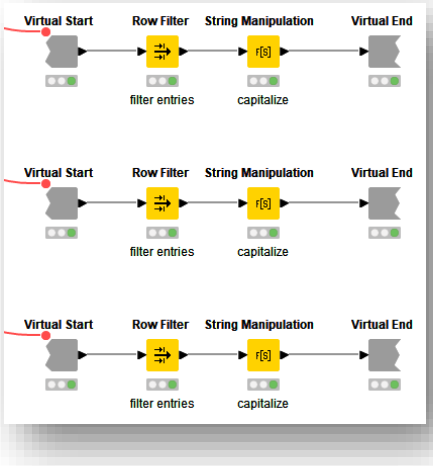
- Components in the WebPortal get executed (at least) twice
  - The first time to create the displayed web page, and the second time when one clicks the “Next” button
- To optimize performance, keep heavy computations outside of components
  - e.g., data writing, model training, large data reading



Parallel Chunk Loop Nodes

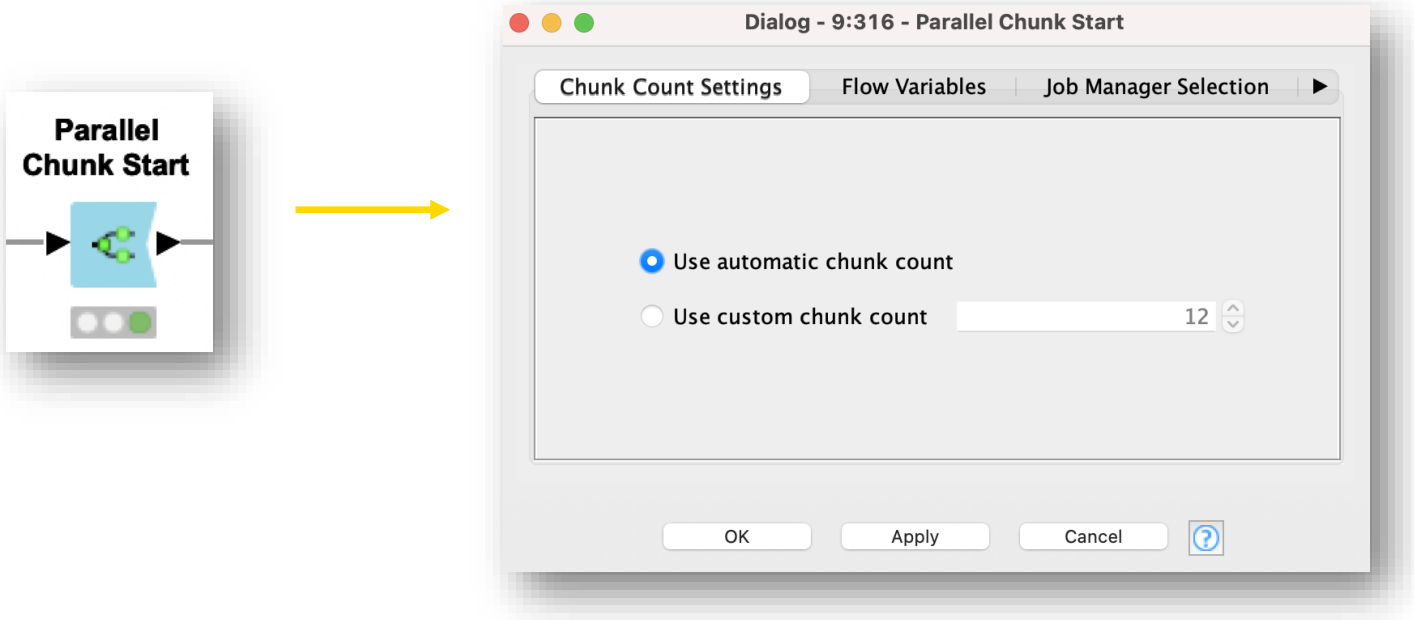
- Parallel Chunk Start node
  - It splits the input data into equally sized chunks and executes the body in parallel, each processing one chunk
- Parallel Chunk End node
  - It represents the end of a parallel chunked block. Results from all parallel executions are collected and concatenated
- Install KNIME Parallel Chunk Loop Nodes extension





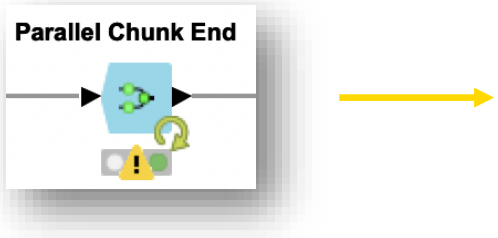
Parallel Chunk Loop Nodes

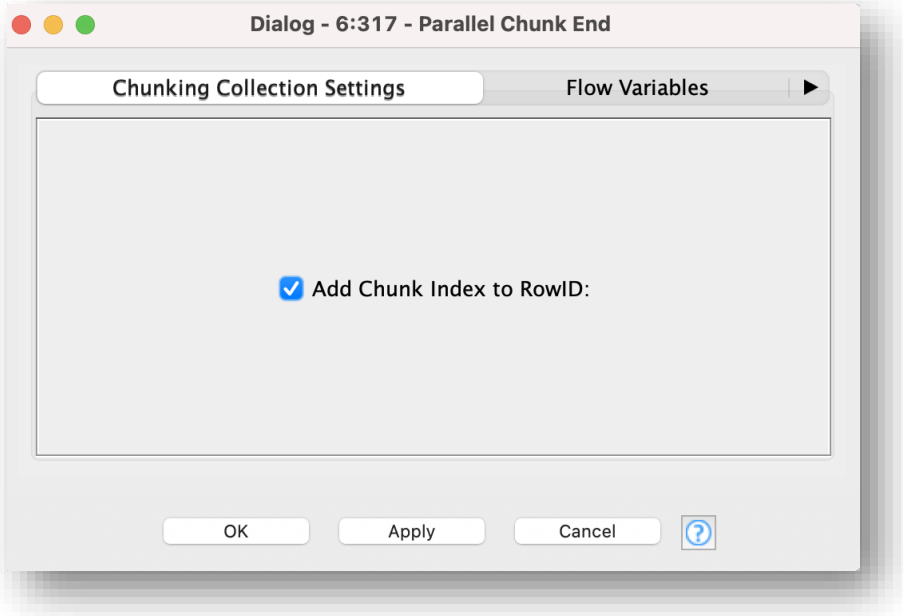
- Parallel Chunk Start node
  - Split the input data table into equal-size chunk
  - Configure the number of chunks in the configuration dialog



Parallel Chunk Loop Nodes

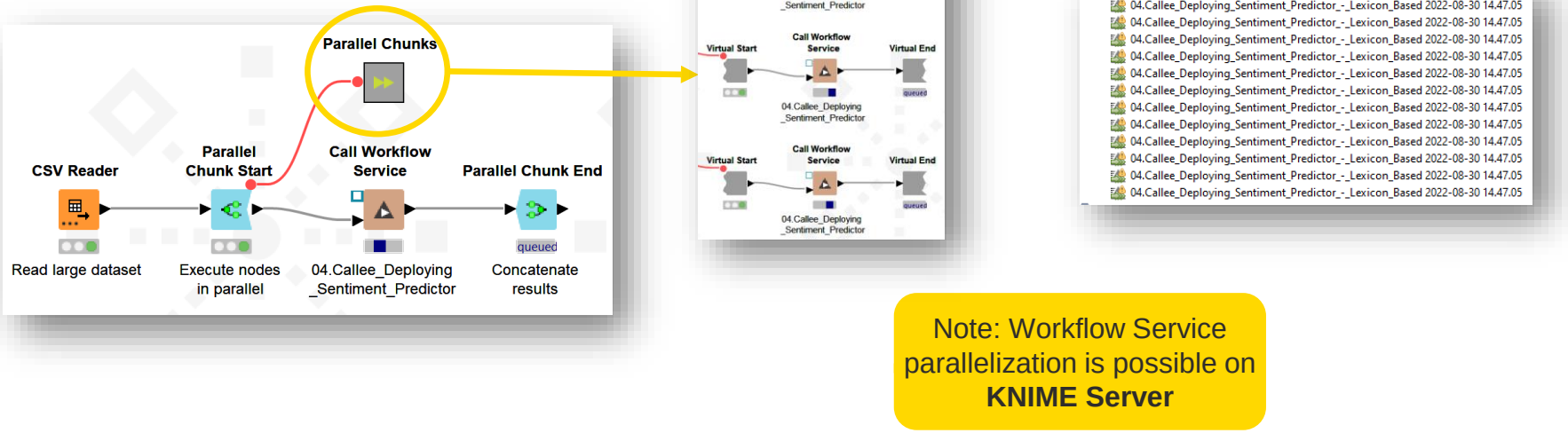
- Parallel Chunk End node
  - Concatenate the rows produced by each parallel execution
  - Add Chunk index to RowID if the segment in the loop generates tables with repeated RowIDs.





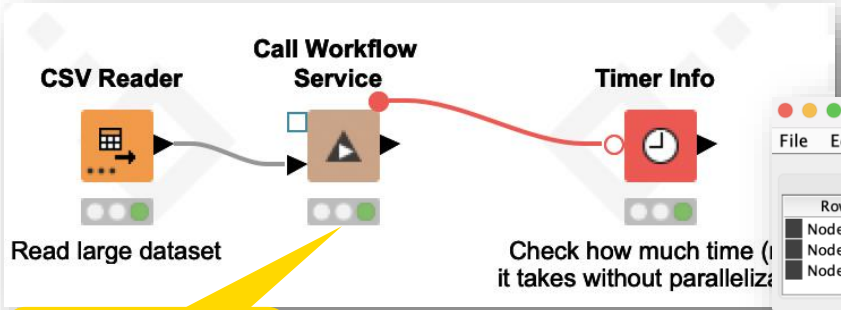
Parallelize Workflow Services in KNIME Server

- Parallelization at his best using
  - Parallel Chunk Loops
  - Workflow Services
  - KNIME Server execution
- KNIME Server creates one job for every parallel Workflow Service



Note: Workflow Service parallelization is possible on **KNIME Server**

Parallel Chunk Loop Nodes - Example



Serial execution on **KNIME Server**: ~392 s

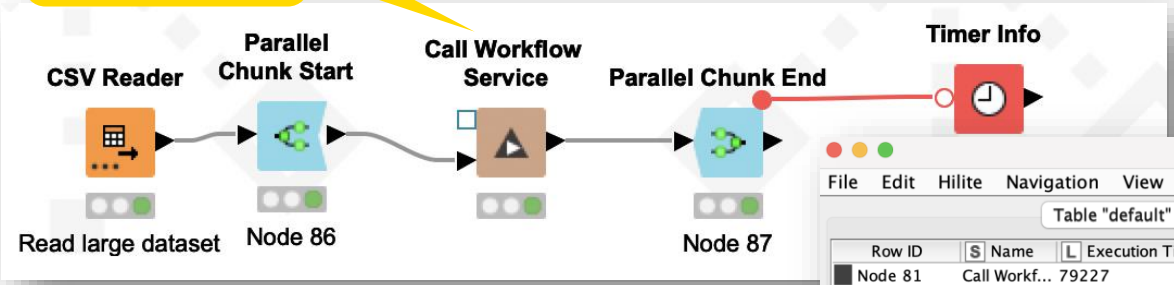
Output table - 1:82 - Timer Info (job)

File Edit Hilite Navigation View

Table "default" | Spec - Columns: 8 | Properties | Flow Variables

Row ID	Name	Execution Time	Execu...	Execu...	Nr of ...	Nr of ...	NodeID	Classn...
Node 81	Call Workflow Service	392604	392604	392604	1	1	39:81	org.knime...
Node 82	Timer Info	?	0	0	0	0	39:82	org.knime...
Node 83	CSV Reader	1114	1114	1114	1	1	39:83	org.knime...

Lexicon-based Sentiment Analysis workflow



Serial execution on **KNIME Server**: ~79 s

Output table - 2:82 - Timer Info (job)

File Edit Hilite Navigation View

Table "default" | Spec - Columns: 8 | Properties | Flow Variables

Row ID	Name	Execution Time	Executio...	Execu...	Nr of ...	Nr of ...	NodeID	Classn...
Node 81	Call Workf...	79227	79227	215281	1	2	40:81	org.knime...
Node 82	Timer Info	?	0	0	0	0	40:82	org.knime...
Node 85	CSV Reader	329	329	1706	1	2	40:85	org.knime...
Node 86	Parallel C...	819	819	2196	1	2	40:86	org.knime...
Node 87	Parallel C...	7353	7353	7353	1	1	40:87	org.knime...

For this example, execution in a fraction (~20%) of the time

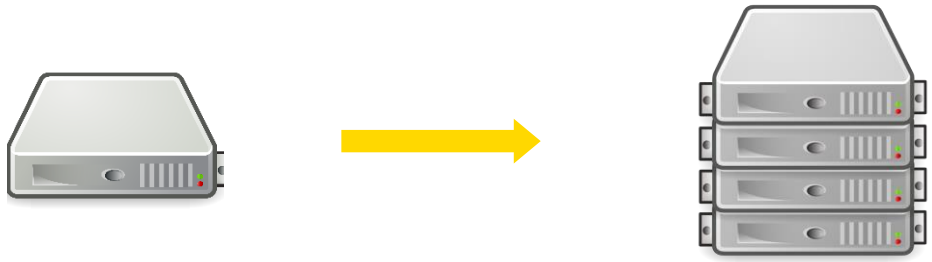
Open for Innovation KNIME

Scaling Up: Better Infrastucture Also Helps in Performance

- Even optimized workflows can crash, or have slow performance, due to poor infrastructure
  - E.g., they can be deployed in a machine that cannot handle enough concurrent users or requests
- KNIME Server supports ”Scale Up”, which usually helps
  - More powerful infrastructure (e.g., a better cloud instance)
  - Having all components in a same machine makes communication among them faster and more reliable

Scaling Out: KNIME Distributed Executors

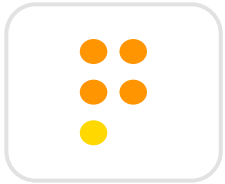
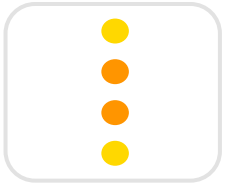
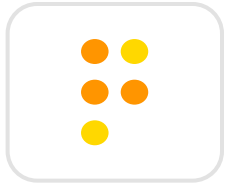
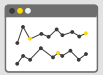
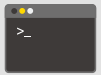
- Distributed executors can also improve the performance of your workflows
  - Depending on the workflow or on the amounts of data, a single executor in a single machine can become a bottleneck
  - Solution: distributed executors in different machines (“Scale Out”)
  - Disadvantage: overhead from communication across different machines
- “Scale Out” is a functionality of KNIME Server Large



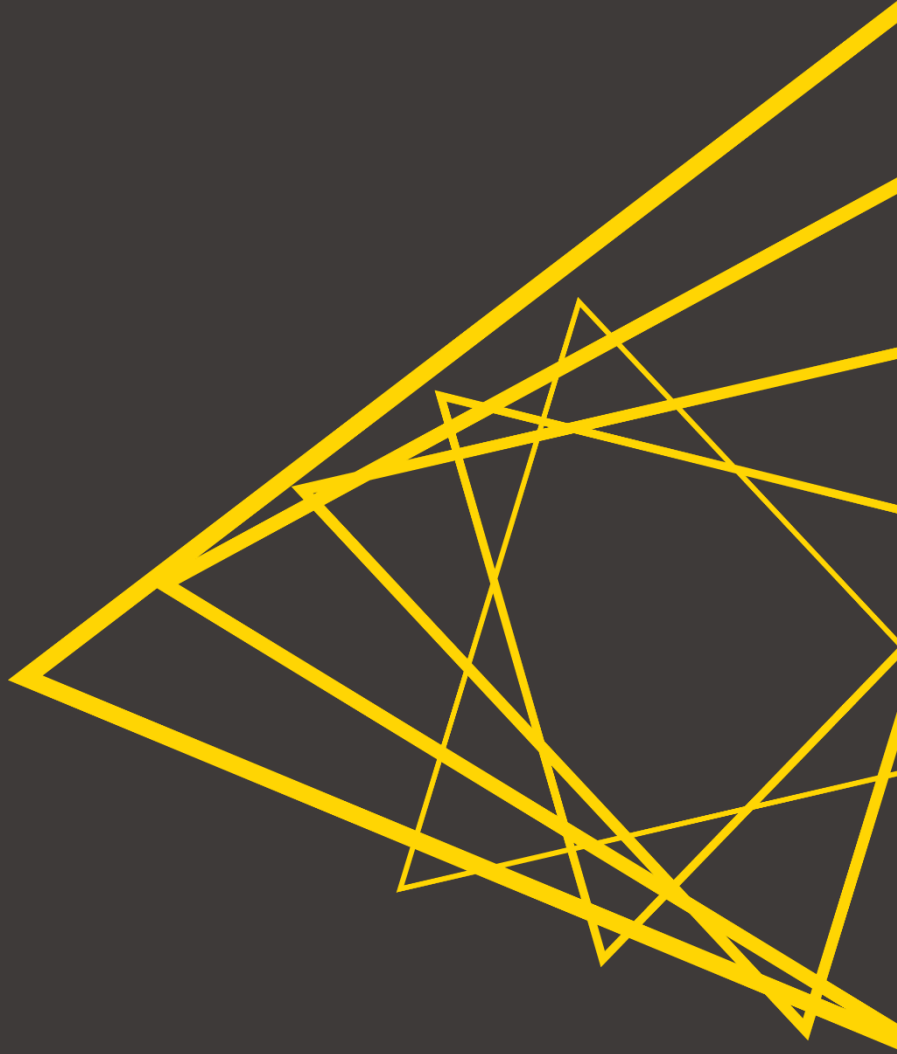
Preview: execution in KNIME Business Hub

KNIME Business Hub in the Organization

IT only allocates **seats**
& **execution cores** per team.

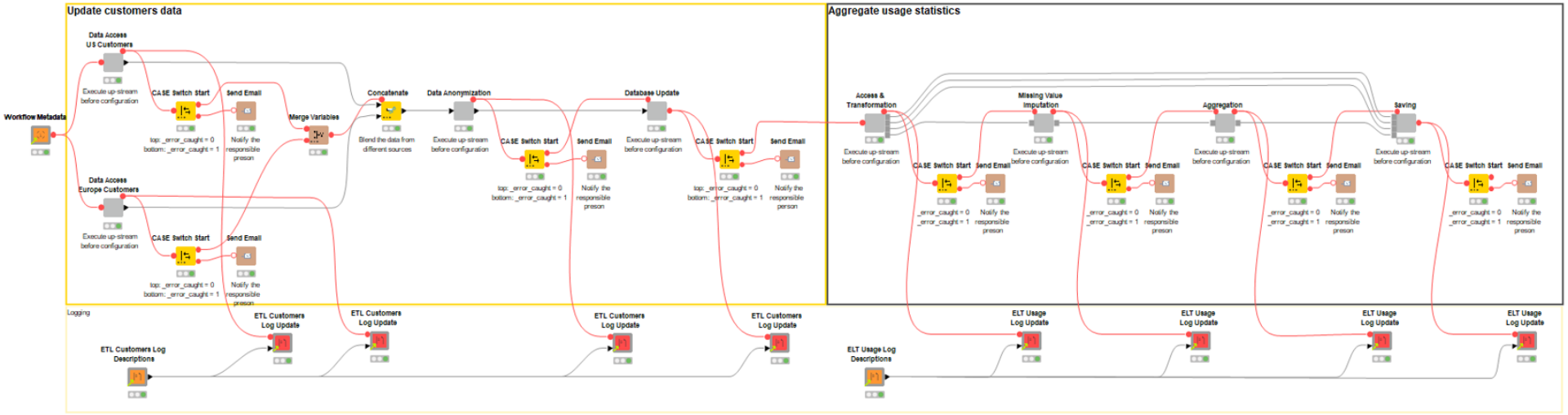
	Marketing team	R&D team	CoE Team
Seats allocated by teams			
Read, write or execute permissions determined by team	Marketing spaces 	Knowledge sharing  Development 	Knowledge sharing  Development 
Deployment & monitoring by team	Scheduled execution for Monthly Reporting 	REST API  Data App 	Data App for Leadership  Data App for Marketing  REST API for Production 
Execution context determined by team	Marketing execution context (8 cores, no GPU)	KNIME expert execution context (8 cores, GPU)	Dev & Test execution context (4 cores, GPU)	Prod execution context (8 cores, GPU)

Orchestration



Motivation behind Orchestration

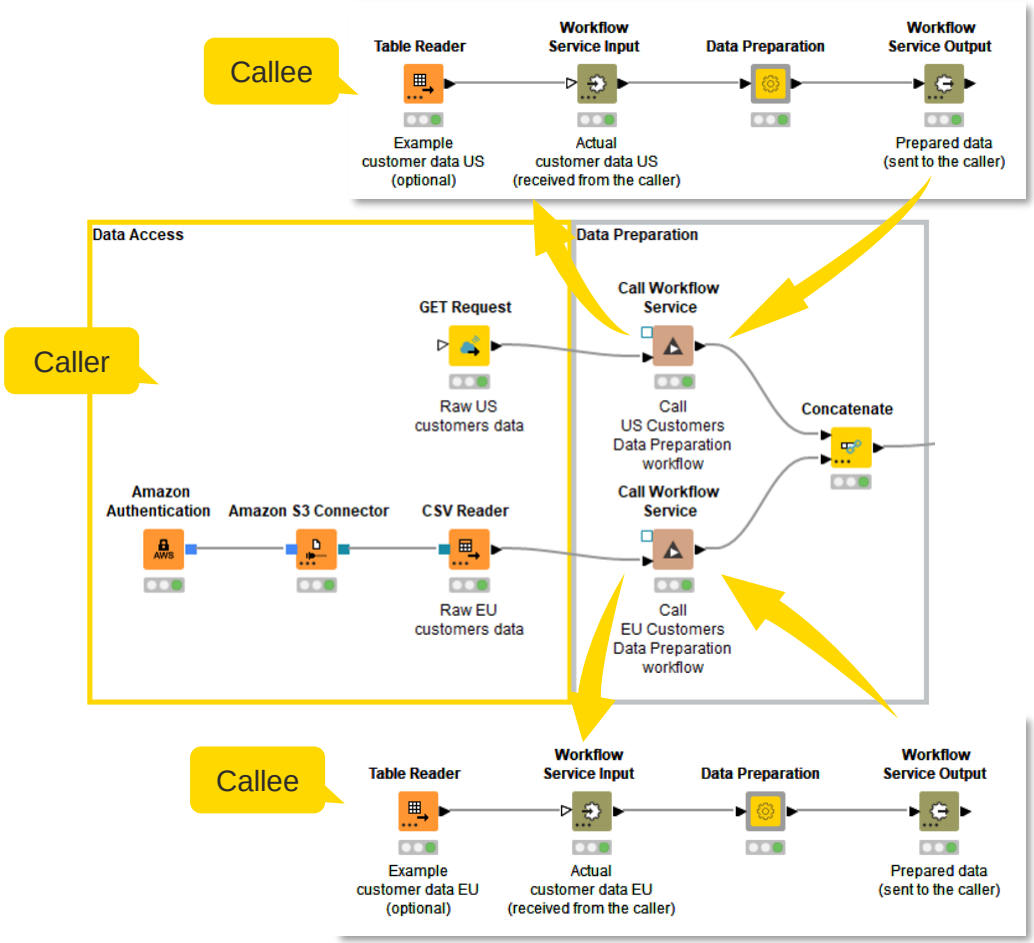
- A large workflow…
  - …combines many processes and purposes in one
  - …is difficult to maintain
  - …is difficult to test
  - …loads slowly



Orchestration

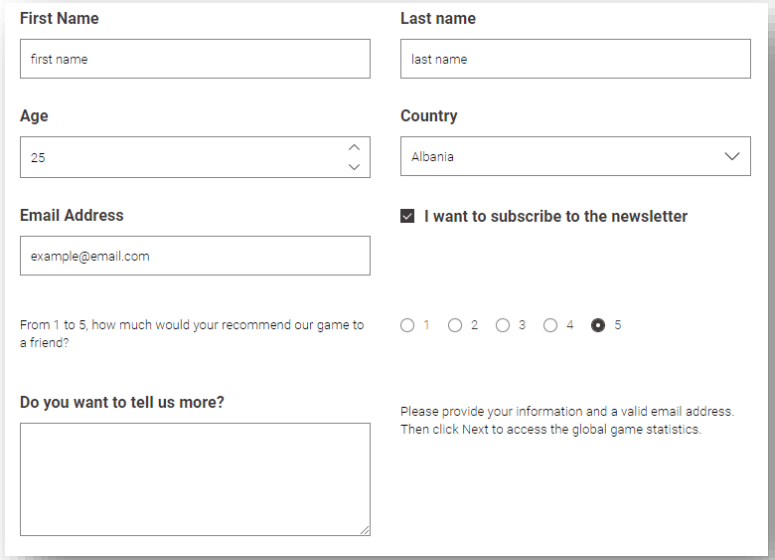
- Create modular workflows
  - Split workflows that perform different processes (*callee* workflows) and orchestration workflow (*caller*)
  - Callee can get data from and expose data to a caller
  - Call and orchestrate in the caller workflow
- Execute in parallel or setup workflow dependencies or cascades

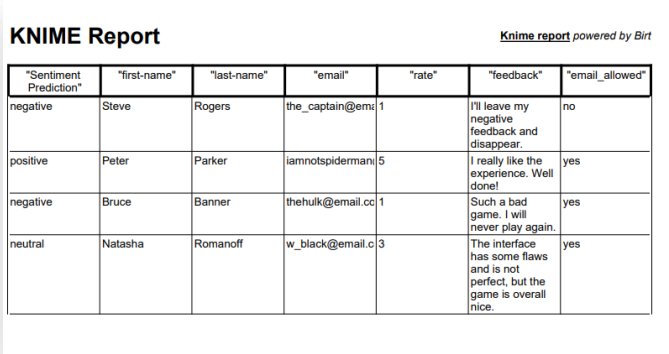
Classic approach:
one workflow call the other pieces



Orchestration Example

- Collect user feedback via WebPortal Data App
- Get the sentiment of the feedback (positive, negative, neutral)
- Generate a daily report with the latest feedback



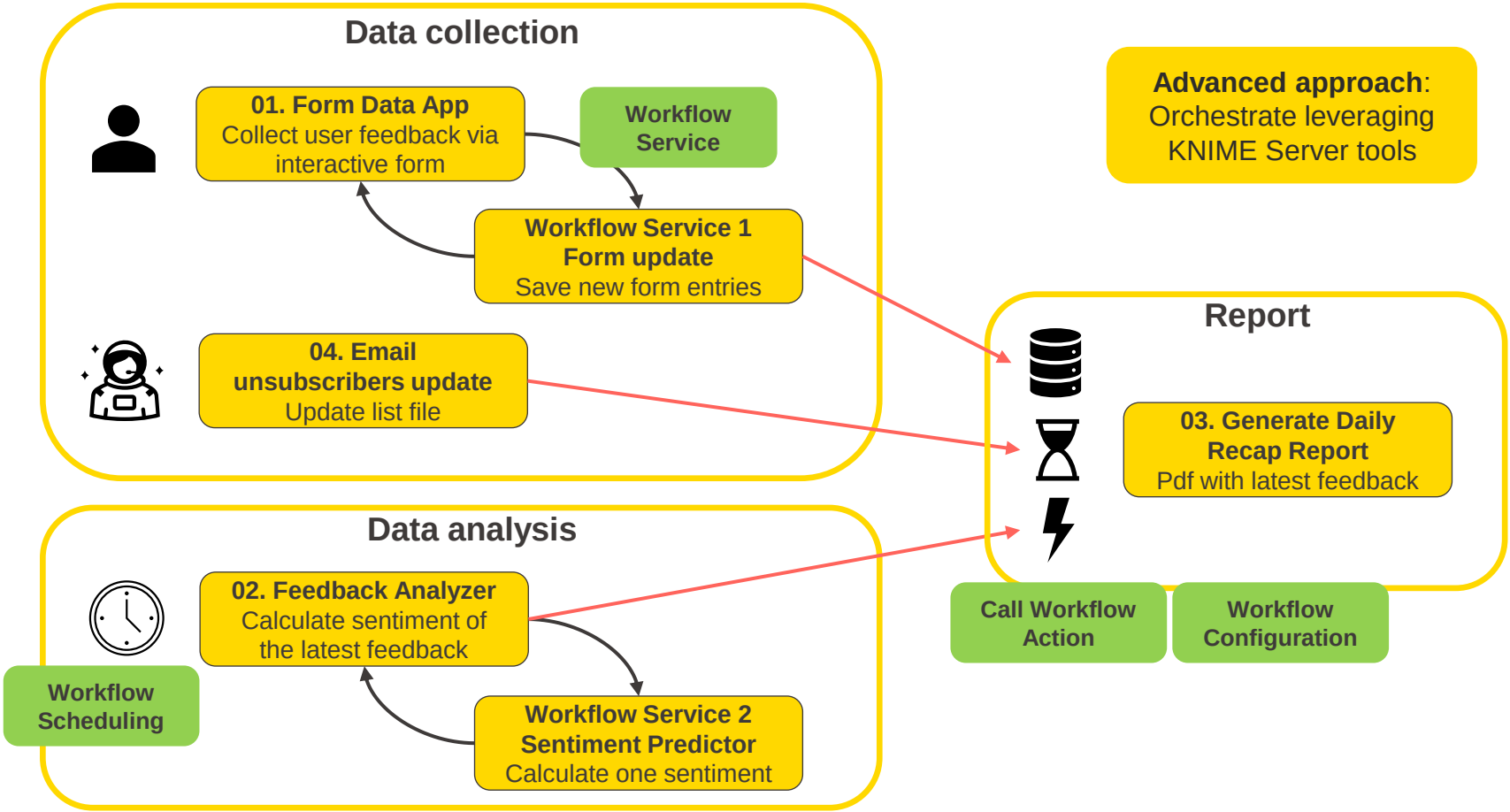


KNIME Report

Knime report *powered by Birt*

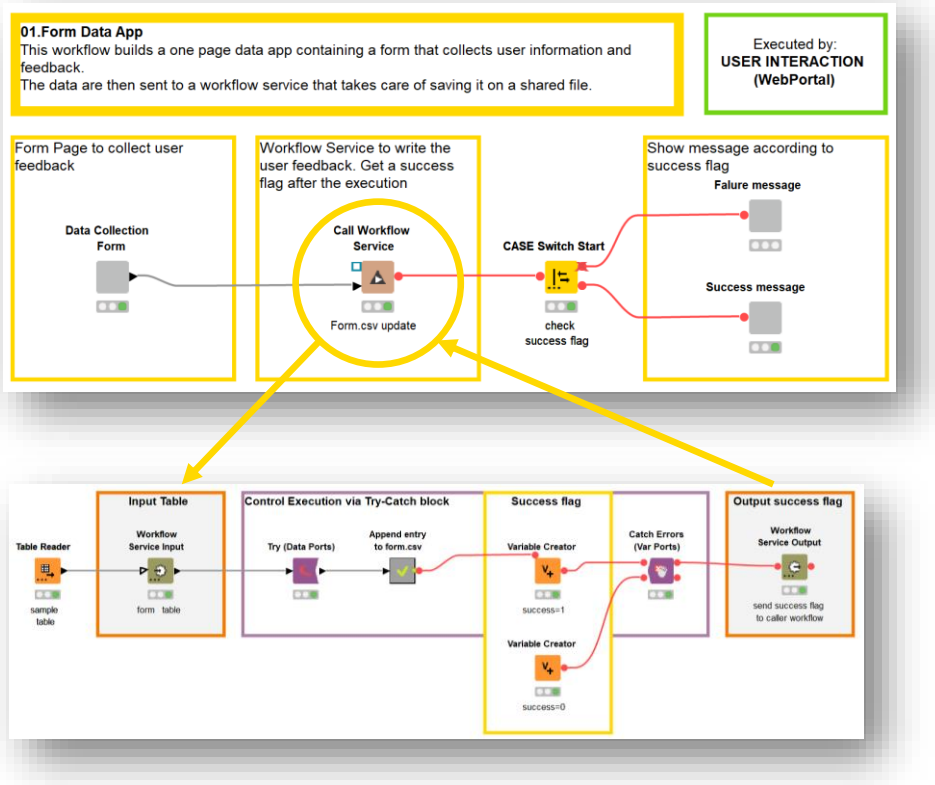
"Sentiment Prediction"	"first-name"	"last-name"	"email"	"rate"	"feedback"	"email_allowed"
negative	Steve	Rogers	the_captain@ema	1	I'll leave my negative feedback and disappear.	no
positive	Peter	Parker	iamnotspiderman	5	I really like the experience. Well done!	yes
negative	Bruce	Banner	thehulk@email.cc	1	Such a bad game. I will never play again.	yes
neutral	Natasha	Romanoff	w_black@email.c	3	The interface has some flaws and is not perfect, but the game is overall nice.	yes

Orchestration Example



Workflow Service

- Delegate operations to external workflow
- Enhance modularization
- Enable workflow dependencies
- Can introduce parallel execution

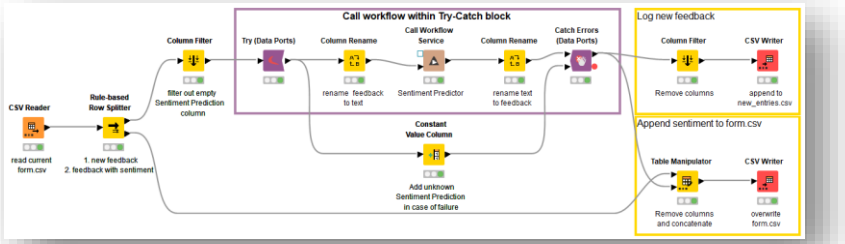


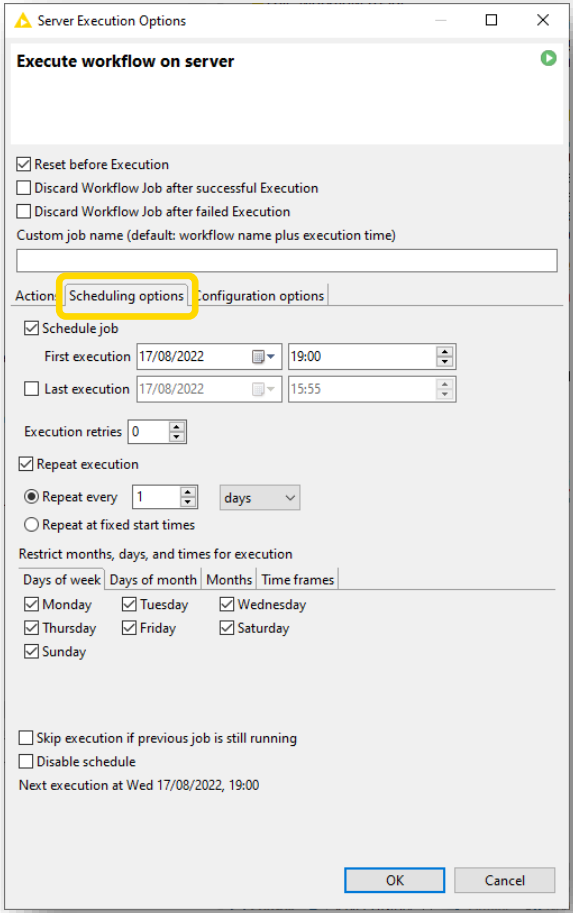
In our example:

- Delegate writing operation of collected data
- Get sentiment of one feedback

Workflow Scheduling

- Execute a workflow on the KNIME Server at a specified time/interval
- Enable automation



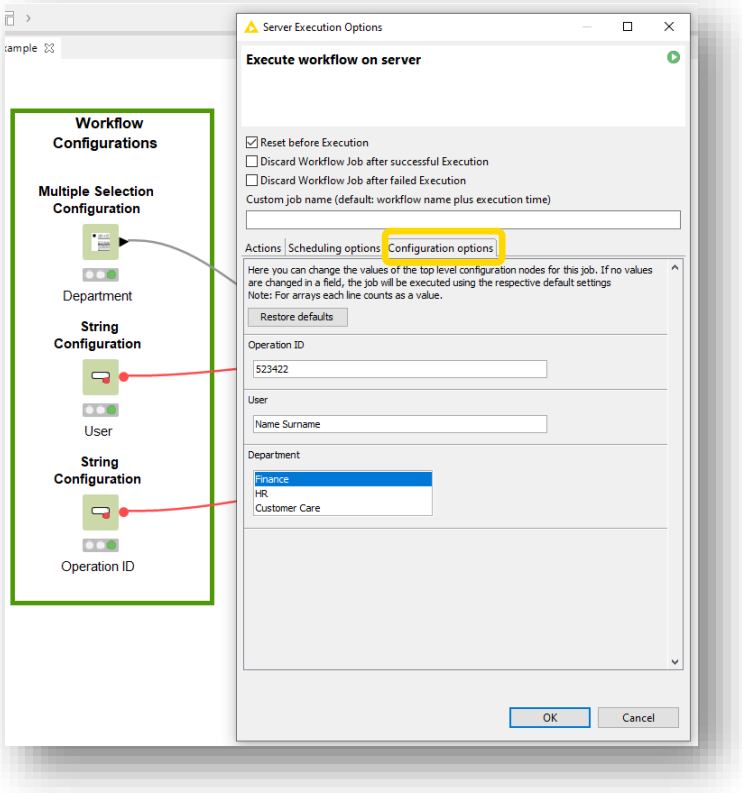


In our example:

- Once a day, append sentiment to the latest feedback

Workflow Configuration

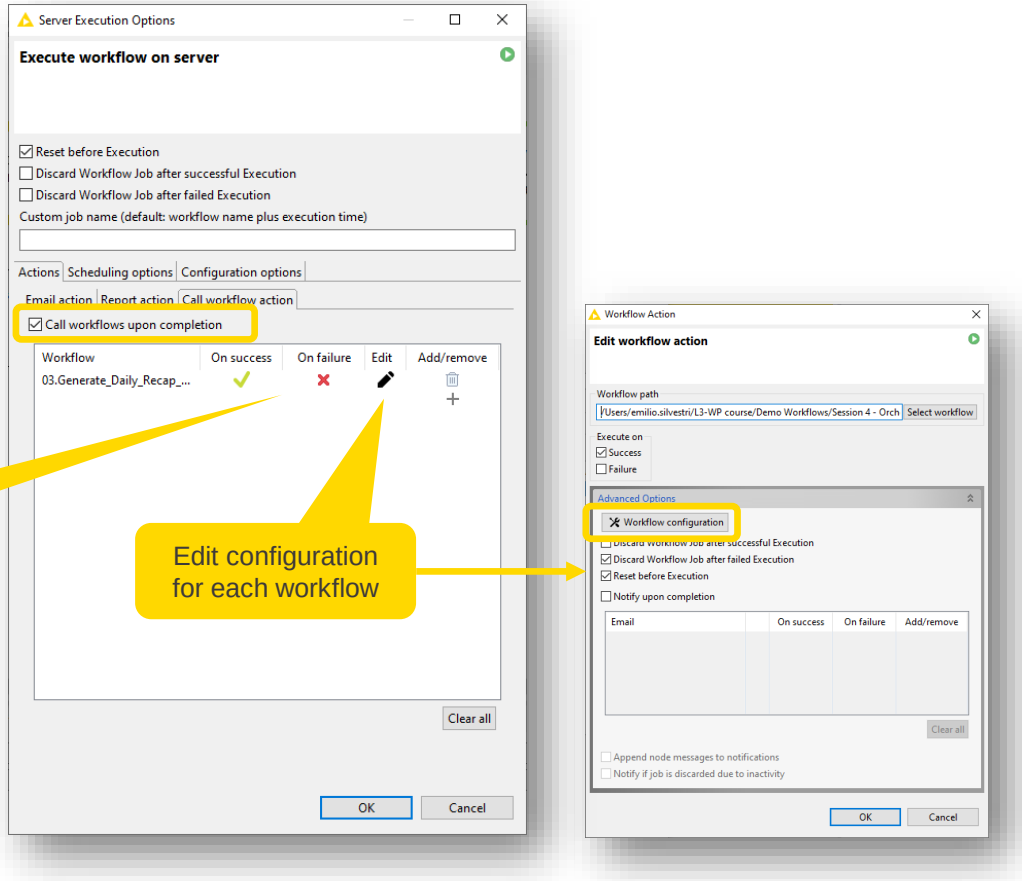
- Configuration nodes in the top-level workflow enable workflow configuration



Example application
Change the behavior of a workflow upon execution, no need to open it

Call Workflow Action

- Trigger the execution of another workflow after this one is completed (only KNIME Server)
- Enable sequential execution
- Decide what to execute next in case of success/failure

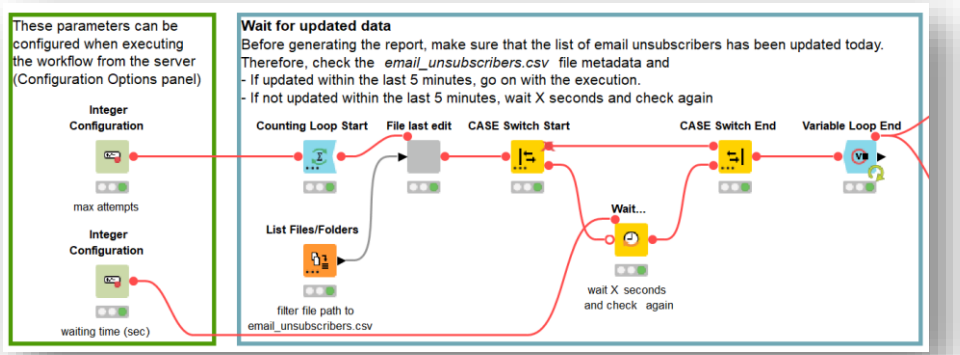


In our example:

- Generate report *after* calculating sentiment feedback

The Wait… node

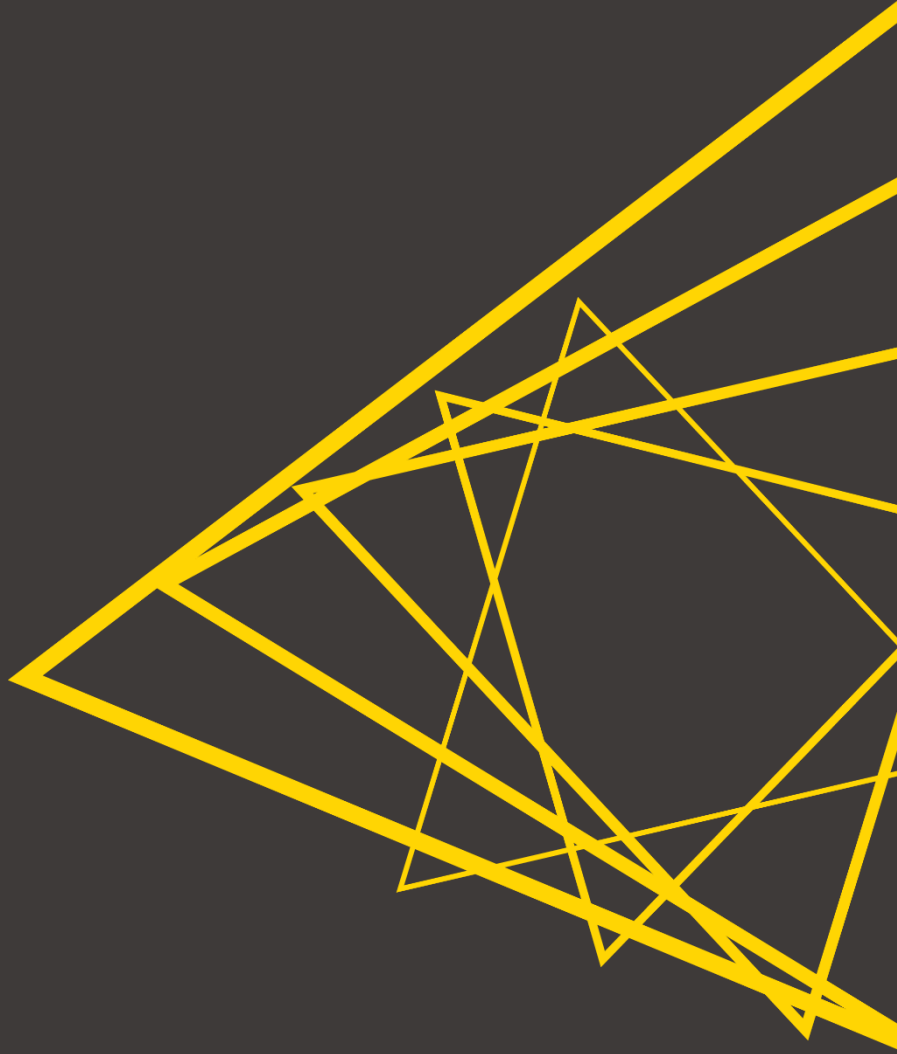
- The Wait… node allows pausing the execution of the workflow
  - Until a specified time
  - For a time interval
  - Until file creation, modification or deletion
- Powerful to model execution dependencies



In our example:

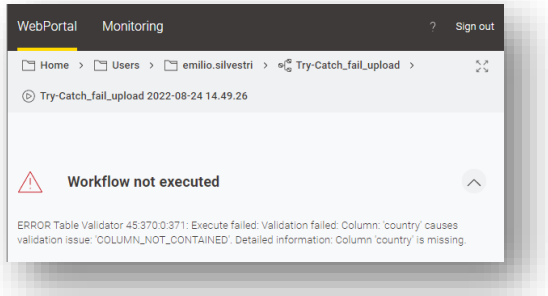
- Wait for data updated by an external department

Error Handling

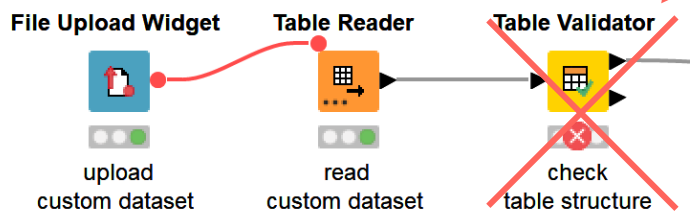


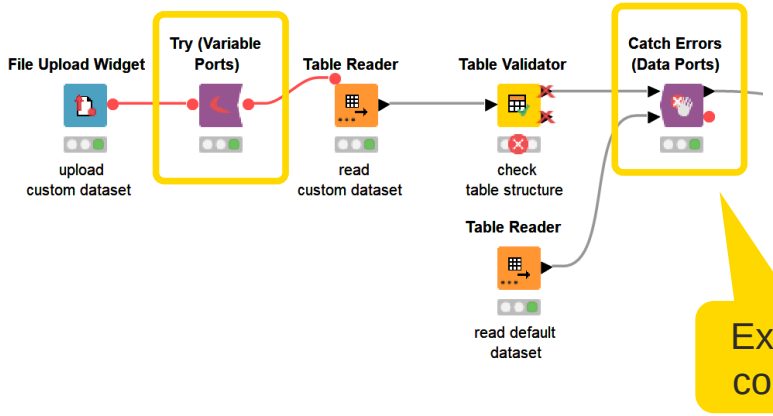
Handling Errors

- Data App fails if just **one** node fails
  - (Non-expert) end users run into abrupt interruption of Data App execution



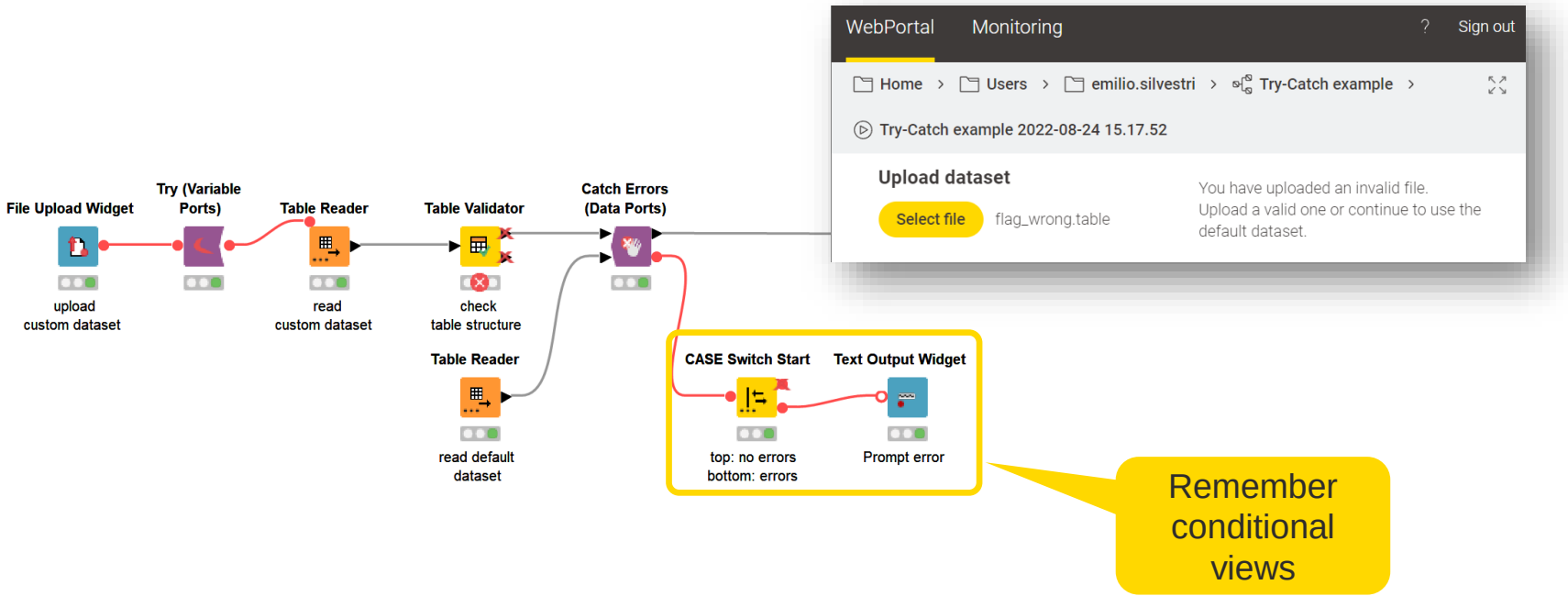
- Try & catch nodes let us
  - Continue the execution
  - Nicely notify the user about the error
  - Report failure





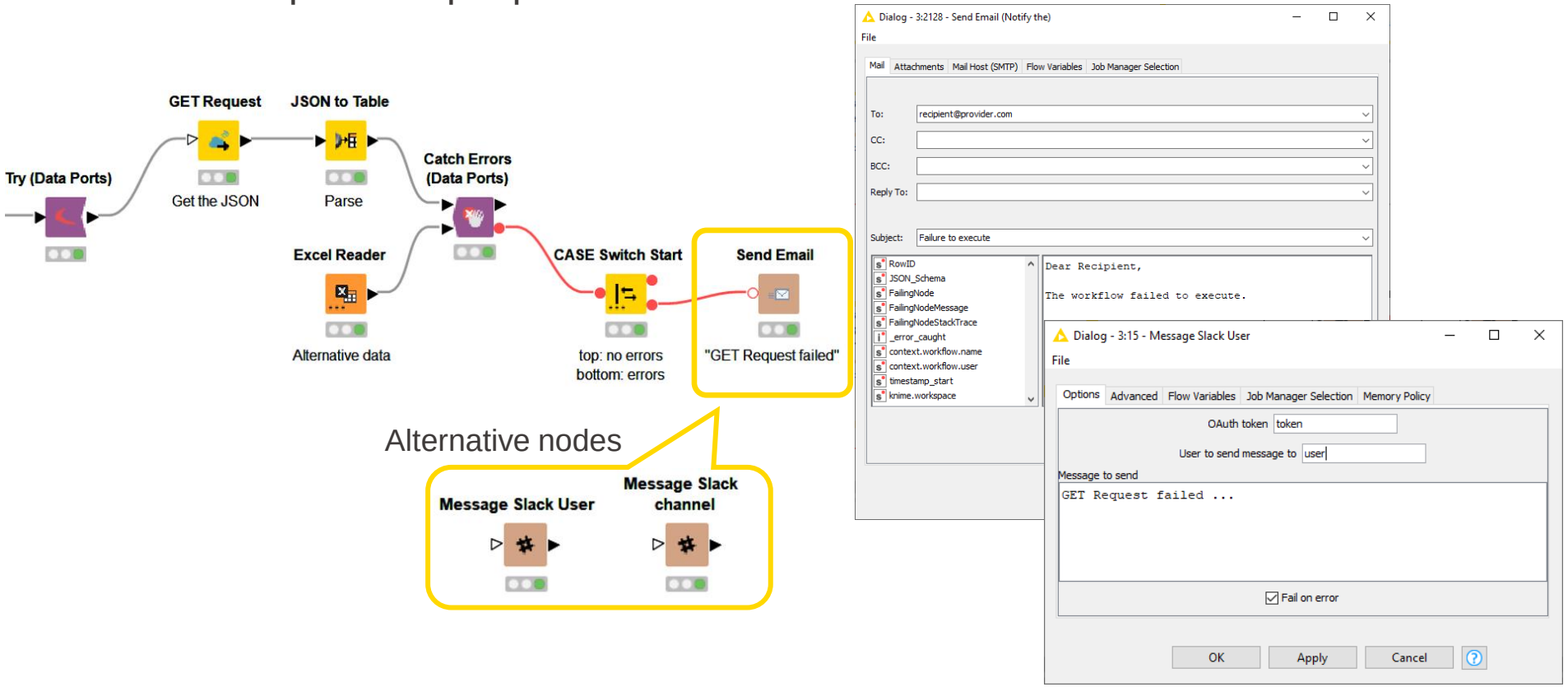
Notify the user

- Inform the user about the error
- Useful if the user can solve the error e.g. by changing input



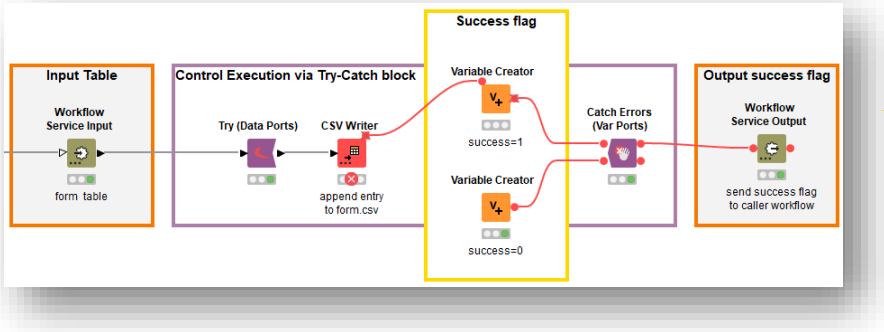
Reporting Failures

- Inform responsible people about a failure



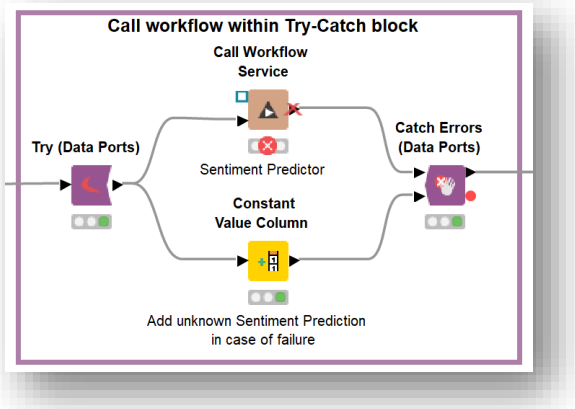
Errors in Workflow Services

- Dealing with Workflow Services, errors can be handled at different levels



In the **callee** workflow (i.e. the service)

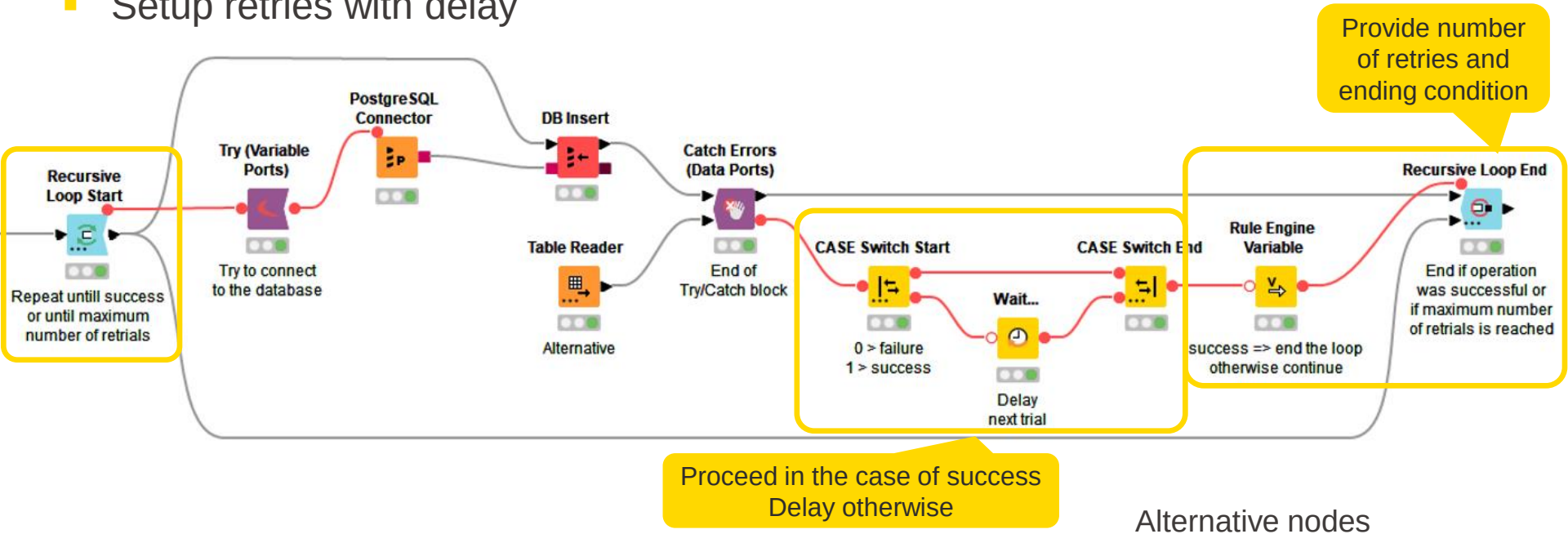
In the **caller** workflow



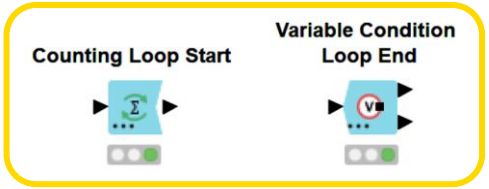
- The choice depends on the use case

Handling Errors with Retries and Delay

- Setup retries with delay

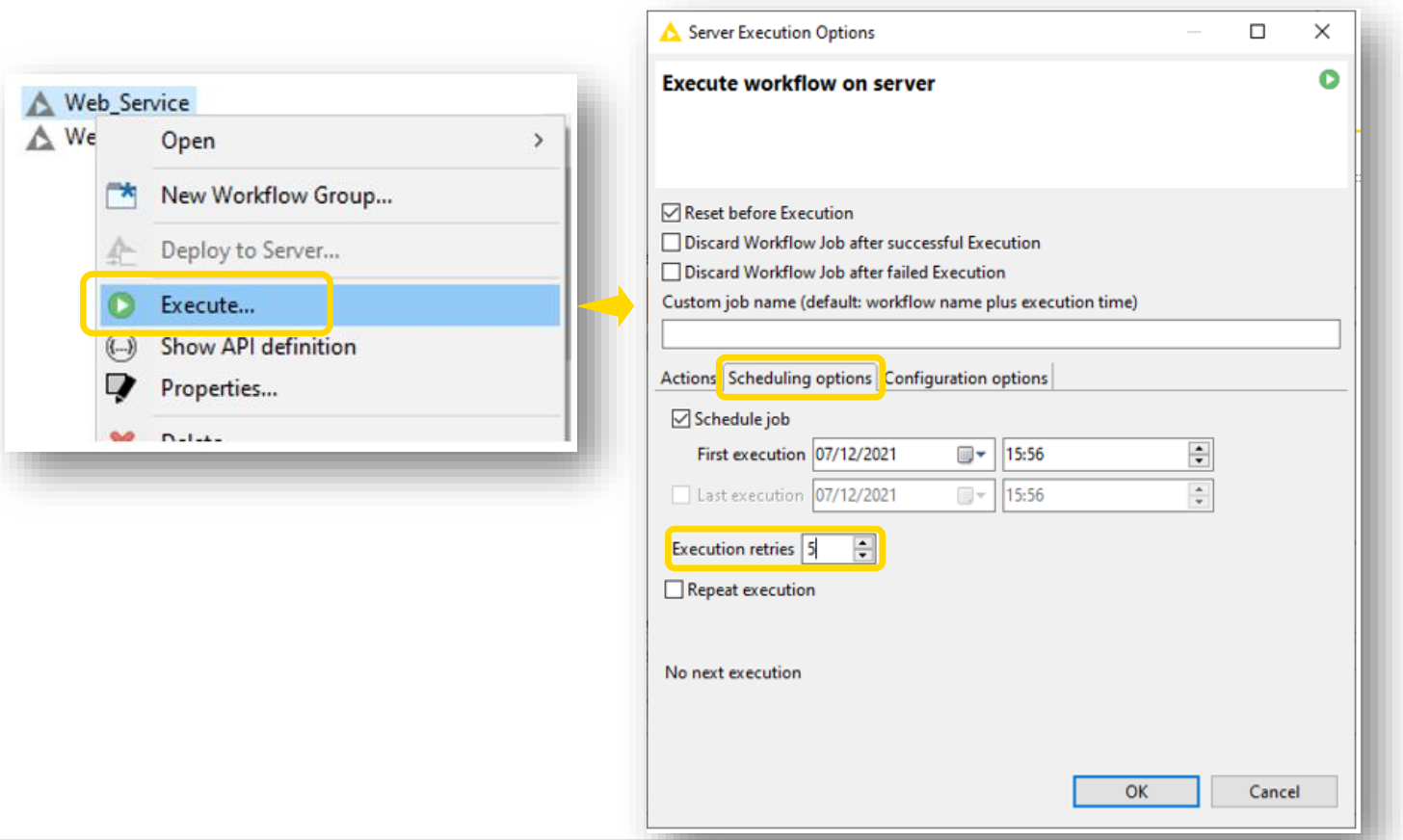


Alternative nodes

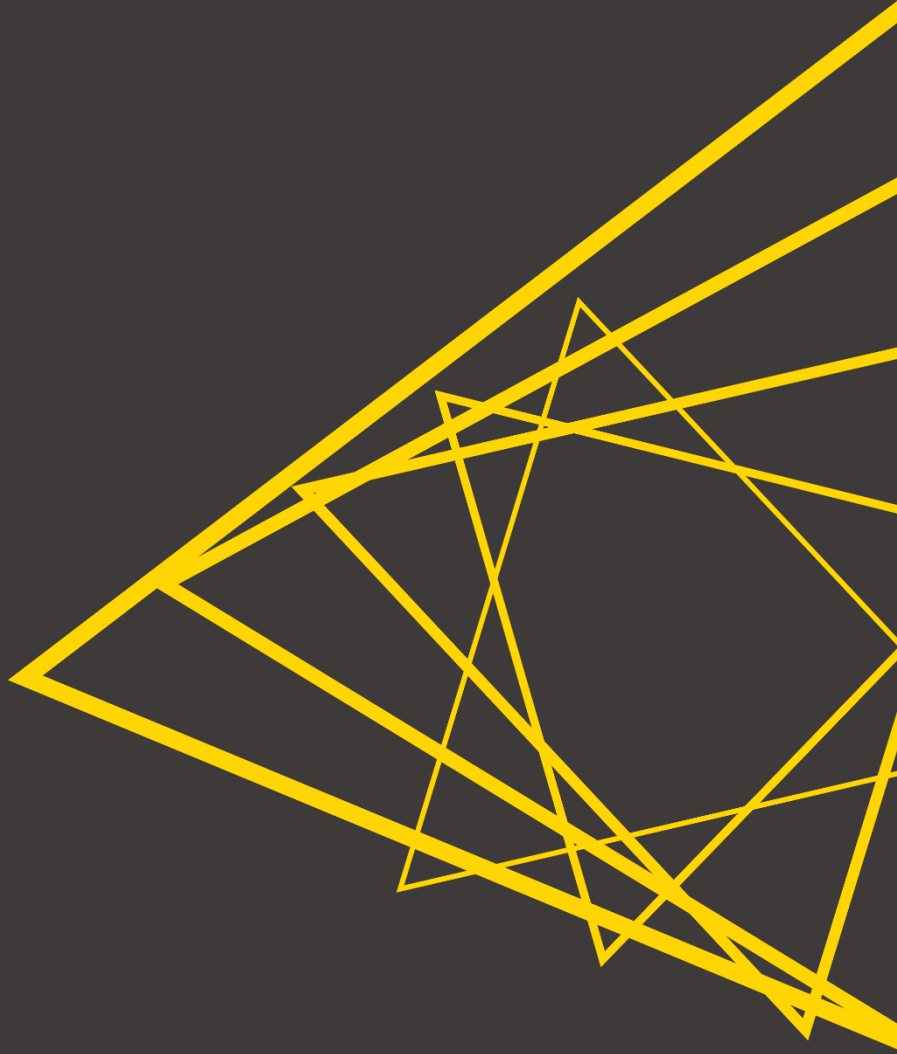


Handling Errors with Retries on the KNIME Server

- Setup several retries when scheduling on KNIME Server

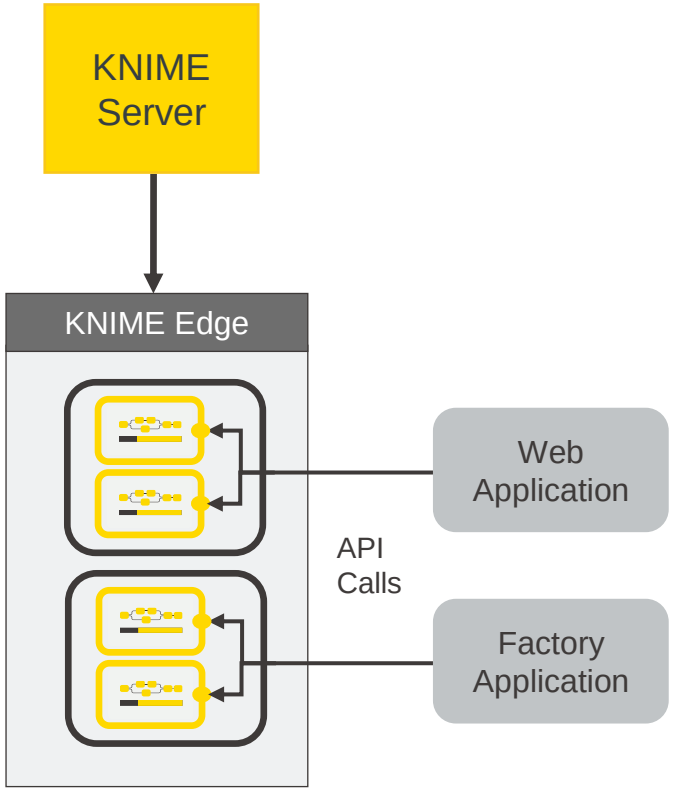


KNIME Edge



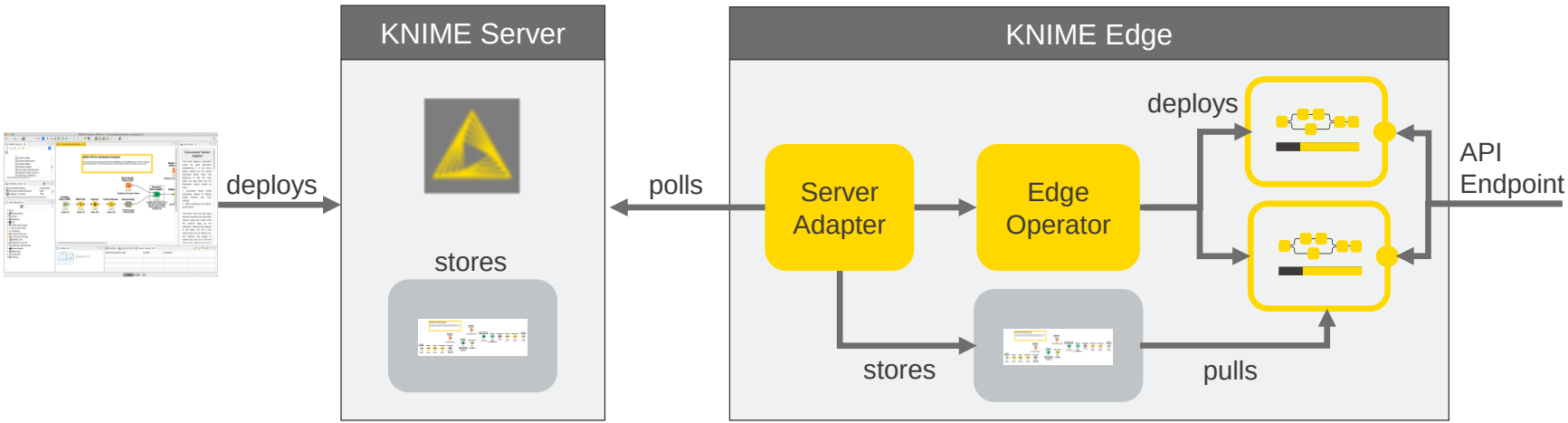
KNIME Edge: a KNIME Server Extension

- Decentralised Kubernetes cluster to:
  - **Run workflows needing:**
    - High request throughput capability
    - Low and constant latency
    - Scaling on demand
    - To stay close to production where data is produced
  - **While using KNIME Server as a control plane for:**
    - Monitoring and management
    - Rolling updates with no downtime
    - Rolling updates to many, concurrent users
    - Replicate deployments across geographical regions



Deploying a KNIME Inference Workflow

- Create a workflow in KNIME and deploy to Server
- Edge(s) poll Server for inference workflows
- Edge deploys workflows at scale with an API Endpoint
- Applications use endpoint to invoke inference workflows

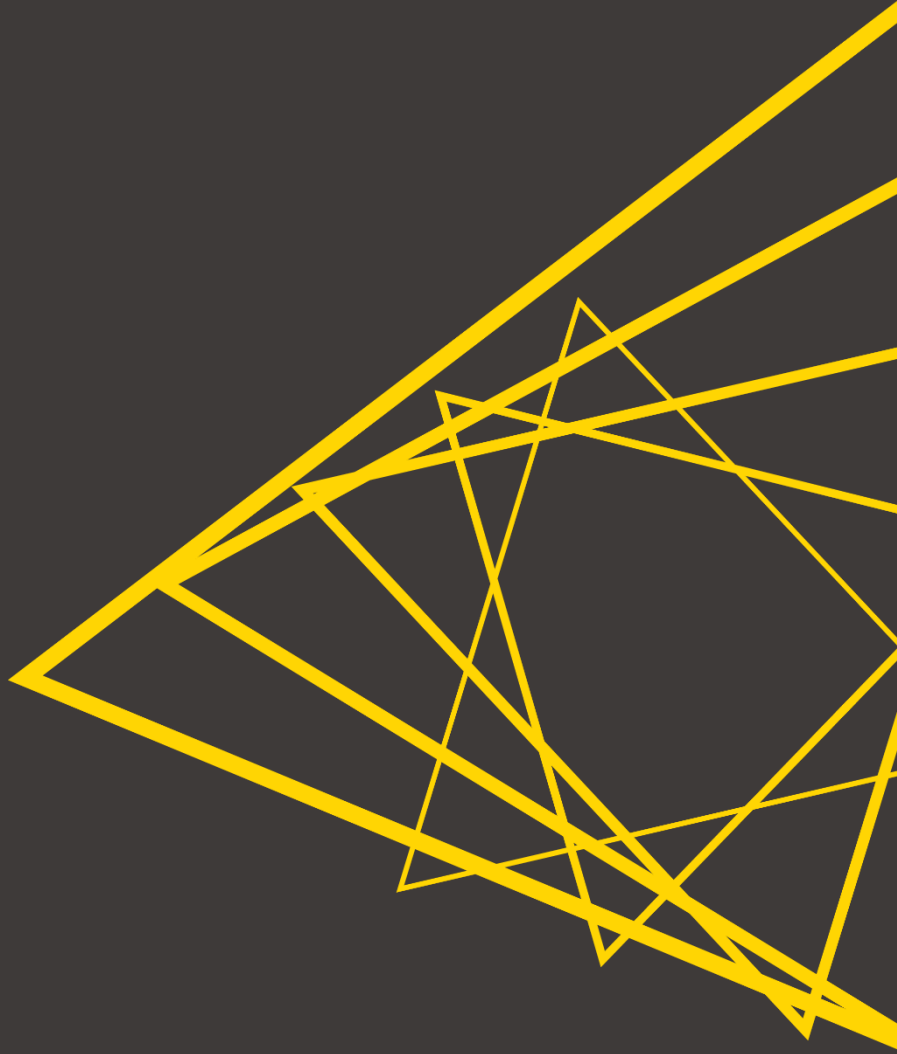


Session 4: Summary

Now you should be able to:

- List the possible performance optimization operations
- Recall and apply the tools for orchestration offered by KNIME Server
- Identify the techniques for error handling

Exercises



Exercises

- **Exercise 01 -** Use the Parallel Chunk Loop nodes to execute a lexicon-based sentiment predictor in parallel on KNIME Server
- **Exercise 02** - Move the workflows to production by deploying on the KNIME Server and configuring execution dependencies. Practice workflow scheduling, call workflow actions and workflow configuration execution.

Open for Innovation
KNIME

Thank You!