Flowable Essentials

Introduction

Digitalization

What is digitalization all about?

In business, digitalization often gets caught up in buzzwords like AI, Blockchain, and Robotic Process Automation. However, its core principles remain consistent and vital.

At its core, digitalization addresses the **interaction** and **workflow** within the business layer, where manual activities like paper-based processes and extensive email communication still prevail. These manual processes often **operate separately** from streamlined systems like HR and ERP, leading to friction and increased transaction costs.

Moreover, digitalization presents an opportunity for **data-driven decision-making**. By digitizing processes, businesses can efficiently gather and analyze data, leading to smarter decisions and a competitive edge in the marketplace.

It's important to note the growing complexity of **compliance regulations**, which demand clearly audited business processes, adding another layer of importance to digitalization efforts.

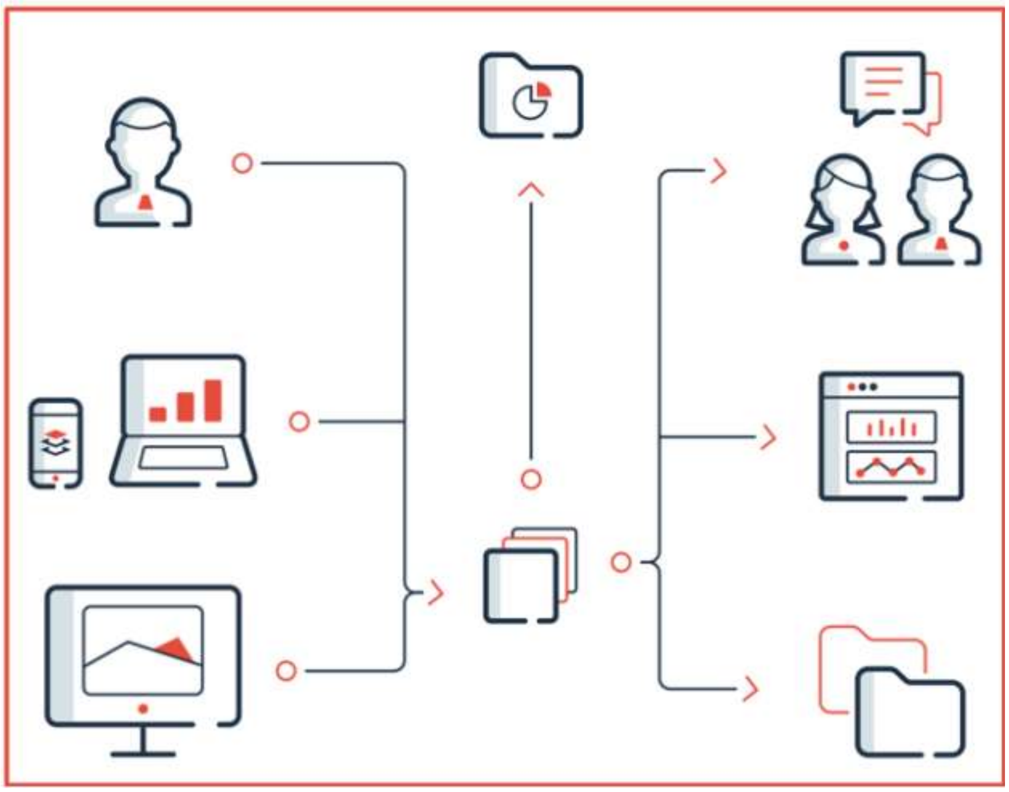
Hyperautomation

What is Hyperautomation?

You may have heard of the term "**Hyperautomation**". It represents a significant evolution in the realm of digital operations. Gartner defines it as a "business-driven, disciplined approach that organizations use to rapidly **identify**, **vet**, and **automate** as many business and IT processes as possible." It's a strategy involving the orchestration of multiple technologies, tools, or platforms, moving beyond simple automation towards a more comprehensive and integrated approach.

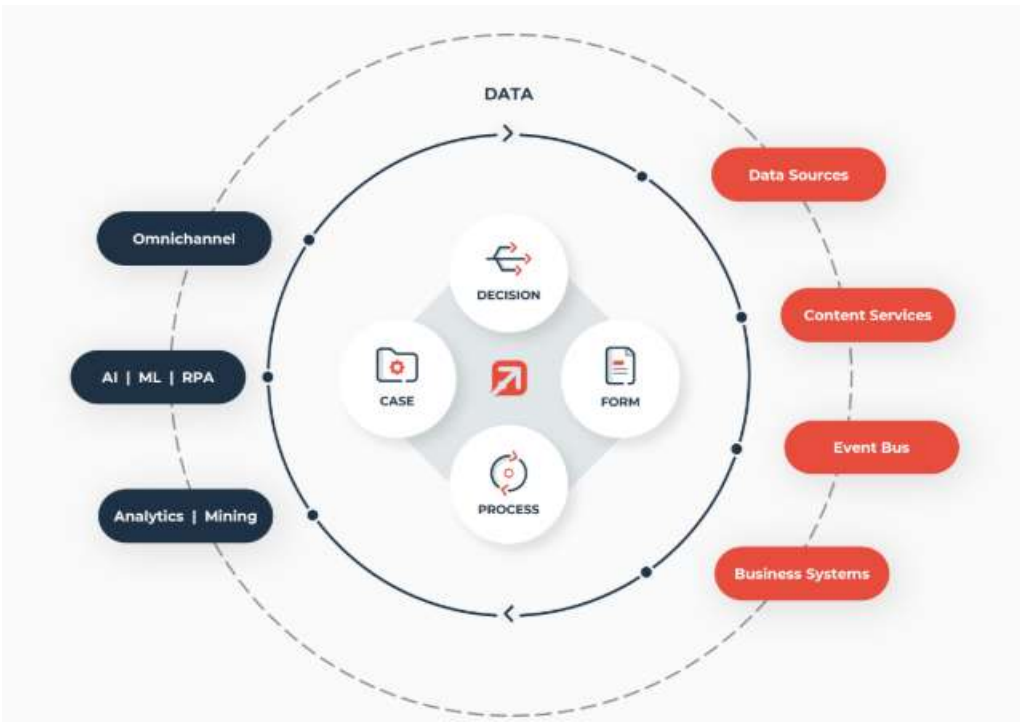
Hyperautomation is not just about automating tasks; it involves intelligently linking a variety of components like AI/ML, RPA (Robotic Process Automation), and low-code platforms. This fusion creates a more efficient and flexible system, capable of transforming and optimizing operational processes. The methodology includes understanding processes, setting automation priorities, and integrating technologies to create seamless, end-to-end solutions.

While Flowable does not implement all components of a Hyperautomation platform, it offers a variety of **connectors** to integrate with the existing ecosystem. As a **Business Process Automation low-code platform**, Flowable plays an important part of the hyperautomation ecosystem. The capabilities of the Flowable platform will allow you to integrate your business processes with a slew of tools, connecting you to the wider world of hyperautomation.



Flowable to the Rescue

What does Flowable bring to the table?



Since you are here, you are obviously interested in how Flowable can help you achieve these goals. It achieves this by enabling two critical functionalities: first, it allows knowledge workers (and systems!) to enter business data through **forms**.

Secondly, it automates the integration of various systems, ensuring that different components and systems of the business process **communicate effectively** with each other. This is achieved through functionality such as the *Service Registry*, *Event Registry* and *Connectors*.

Many people initially think of forms when considering digitalization solutions, as forms are the most visible aspect. However, Flowable emphasizes the entire business process, where forms are just one part of capturing data and driving the **workflow forward**. Everything within Flowable connects to some way or another to the overall workflow.

Flowable's approach is centered around its usage of the **BPMN** (Business Process Model and Notation), **CMMN** (Case Management Model and Notation), and **DMN** (Decision Model and Notation) standards.

Flowable's approach centers on utilizing these standards as visual modeling tools. Modeling processes with these standards makes them executable, streamlining communication with stakeholders and facilitating clearer discussions about business requirements as well.

Flow-Code

No-Code vs. Pro-Code

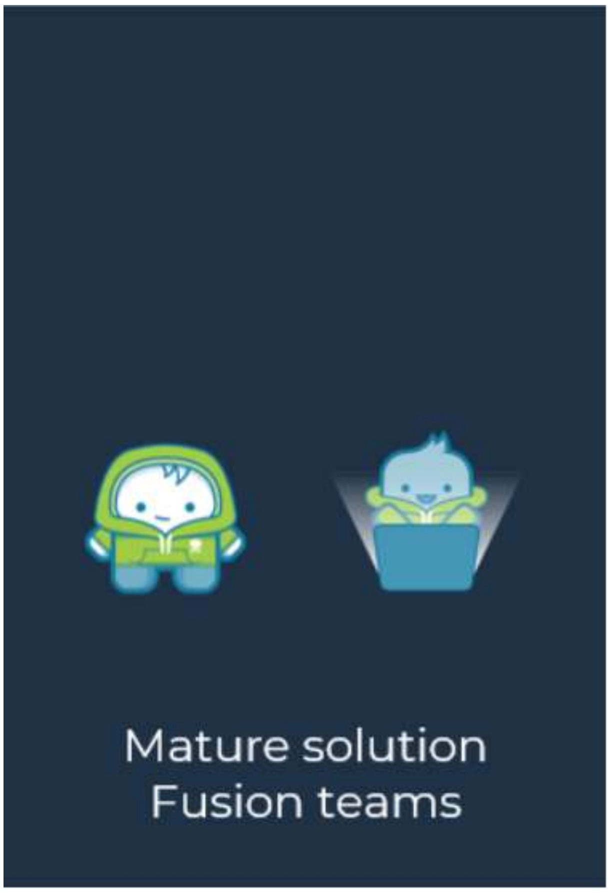
Flowable stands out in the realm of process automation by offering a versatile platform that caters to a wide range of coding skills and project needs, from no-code to pro-code solutions.

No-Code: Empowering Citizen Developers

At its no-code level, Flowable enables **citizen developers**, those without formal programming training, to participate actively in the development process. This approach is ideal for rapidly building **proofs of concept**, allowing teams to evaluate ideas swiftly and efficiently.

It encourages a hands-on approach, where business departments can collaborate directly in the application development, reducing reliance on IT resources.

This is achieved through the model-driven approach employed by Flowable. Business processes can easily be modelled with the BPMN and CMMN editors inside Flowable Design. At the same time, the Flowable Form Editor allows anyone to design dynamic forms which are the foundation for the data model of your business processes.

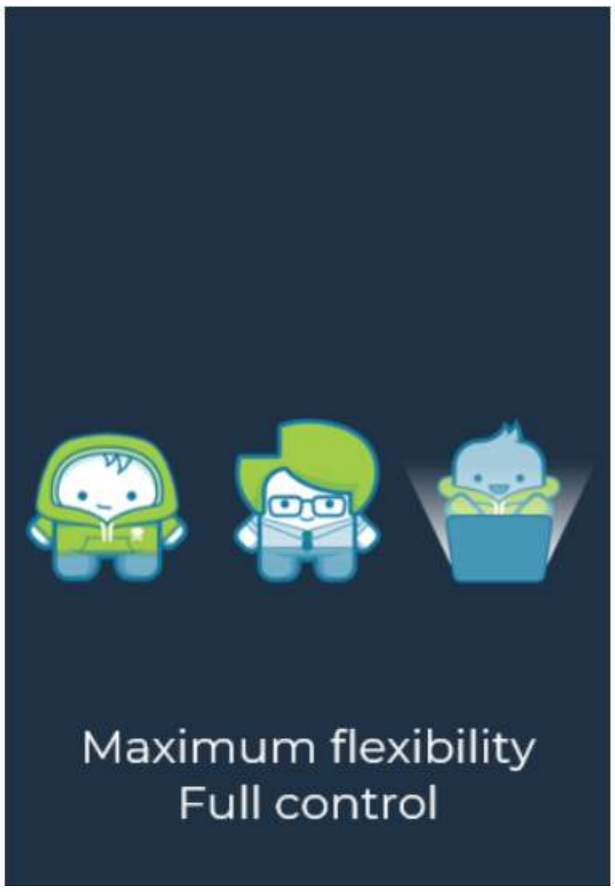


Low-Code: Bridging Business and IT

As projects evolve from proof of concept to **more mature solutions**, Flowable’s low-code capabilities come into play.

Here, business and IT teams can form **fusion teams**, working together to develop applications that may require some coding but not at an extensive level. This middle ground is crucial for integrating with other systems or extending certain functionalities, offering a balance between customization and ease of use.

At this level, modelers and developers start using more complex "Expressions". Simple scripts are used to calculate business data, modify cases, processes or tasks. Data Objects and the Event and Service Registries help you to integrate to almost any external system. Some more complex requirements may be coded entirely in the Java backend.



Pro-Code: Full Control and Customization

For projects demanding **deep technical customization and control**, Flowable’s pro-code capabilities allow developers to delve into detailed coding.

This level is geared towards thoroughly tailoring applications to specific business and IT requirements. The modular nature of the Flowable engines provide the flexibility to optimize and customize the application. Flowable's Java and REST API and deep integration with the underlying Spring Framework allow developers to extend the Flowable platform.

Flow-Code: Evolve as You Go

Flowable adapts to these varying requirements with what we like to call "flow-code", a term that encapsulates its dynamic range from no-code to pro-code. This flexibility ensures that whether you're quickly testing a new idea or building a sophisticated, deeply integrated system, Flowable can support your journey at every step.

Summarizing the Capabilities:

- **Low-Code**: Ideal for rapid solution delivery, fostering fusion teams, adding customized logic, and ensuring easy deployment.

- **No-Code**: Empowers citizen developers, accelerates proof of concept phases, and embodies a "Just Do It" attitude for quick, iterative development.
- **Pro-Code**: Offers the freedom to customize and optimize anything, providing full control over the development process for complex, bespoke solutions.

Flowable’s multi-tiered approach aligns with the evolving needs of businesses, ensuring that no matter the stage or scale of your project, there’s a fitting and efficient way to bring your ideas to life.

Question 1
What best defines Hyperautomation?

a. A method to replace human labor with AI.

b. A business-driven approach to comprehensively automate business and IT processes

c. A technique focused on automating Service Tasks in Flowable

d. A strategy to unify REST APIs

Answer : B

Question 2
What major challenge does digitalization address in business processes?

a. The disconnect between manual processes and streamlined systems.

b. The focus on international expansion.

c. The need for new product development.

d. The reduction of employee headcount.

Answer: A

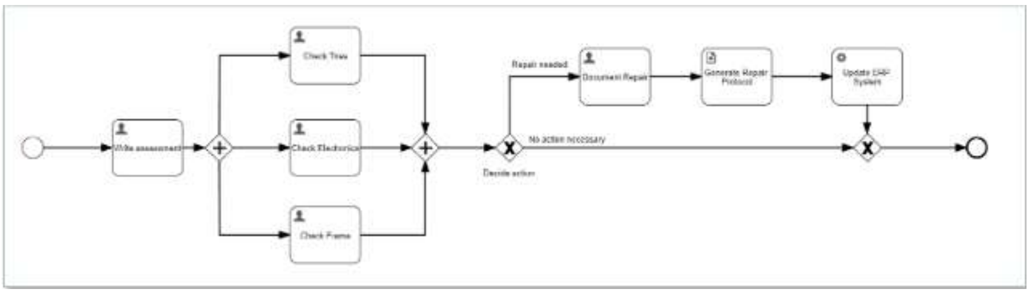
Flowable Concepts

Flowable and Standards

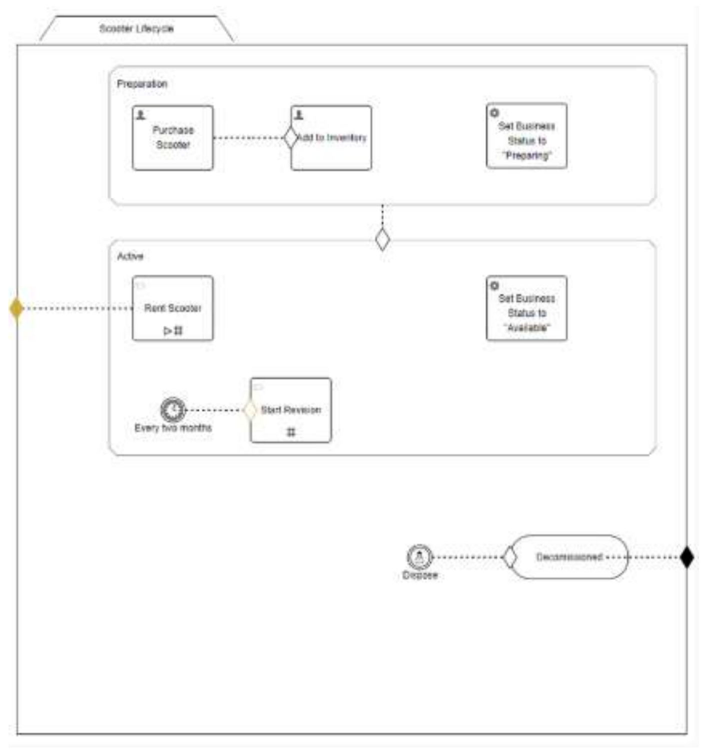
Flowable adheres to several industry standards to ensure interoperability, compatibility, and reliability: **BPMN, CMMN** and **DMN**.

These standards serve as guiding principles, shaping the design and implementation of Flowable's solutions. All these standards are not only useful for documenting the business processes but also define the **business logic** for all Flowable applications.

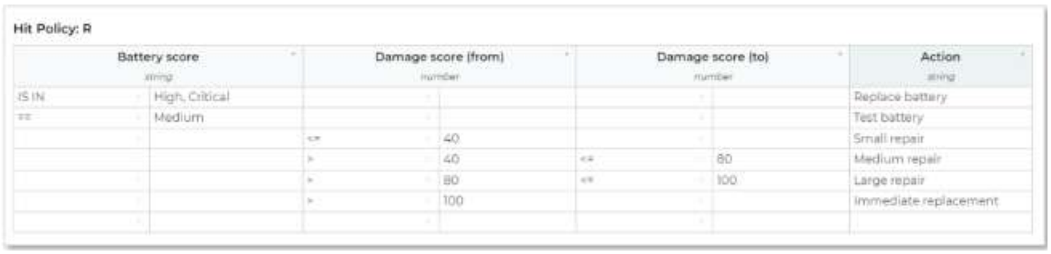
The **BPMN** (**Business Process Model and Notation**) standard is a widely recognized graphical notation for specifying business processes. By embracing BPMN, Flowable ensures that its process modeling capabilities are aligned with industry best practices, enabling users to create clear and comprehensive representations of their business workflows. BPMN is useful for clear, procedural and deterministic flows.



In addition to BPMN, Flowable also supports **CMMN** (**Case Management Model and Notation**), which provides a standardized approach to modeling case management scenarios. By supporting CMMN, companies can manage dynamic and unstructured processes with Flowable, facilitating adaptive case management and improving overall business agility. Unlike BPMN, CMMN is primarily intended for dynamic, open-ended and collaborative use cases.



Finally, Flowable also embraces **DMN** (**Decision Model and Notation**), a standard for modeling decision logic. This helps organizations to model and document complex business rules with precision and clarity.



Hit Policy: R

Battery score		Damage score (from)		Damage score (to)		Action
string		number		number		string
IS IN	High, Critical					Replace battery
==	Medium					Test battery
		<=	40			Small repair
		>	40	<=	80	Medium repair
		>	80	<=	100	Large repair
		>	100			Immediate replacement

Furthermore, from a technological point of view, Flowable is built on open source technologies such as **Java**, **Spring Boot**, and **Elasticsearch/OpenSearch**. This ensures compatibility with a wide range of platforms and systems, able to be integrated with existing IT infrastructures and applications.

By standing on the shoulders of giants, Flowable provides a **robust** or **scalable** but also **interoperable** framework and remains future-proof.

Who is responsible for standards?

The standards lie with the Object Management Group (OMG), an international consortium that develops and maintains these standards.

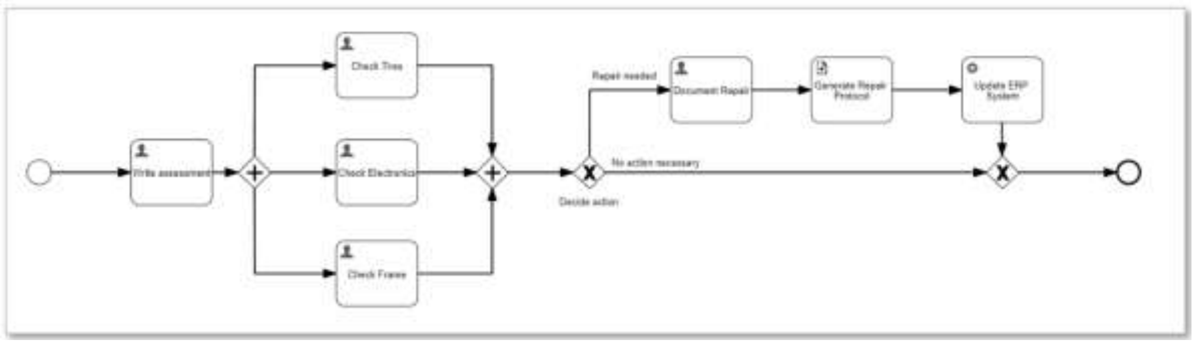
The **OMG** is a non-profit organization comprised of industry experts, software vendors, government agencies, and academic institutions. It is responsible for creating and overseeing various standards related to software and systems development, with a focus on promoting interoperability, portability, and scalability of technology solutions.

These standards undergo a rigorous review and approval process within the OMG before being finalized and published for public use. Once published, the OMG continues to maintain and evolve these standards in response to changes in technology, industry needs, and user feedback.

BPMN Models

BPMN in a nutshell

The main purpose of using BPMN (Business Process Model and Notation) is to provide a **standardized** and intuitive way to **model**, **execute**, **analyze**, and **improve** business processes within an organization. BPMN serves as a **common language** (notation) that enables stakeholders across different departments and roles to understand, communicate, and collaborate on business processes.



Main elements

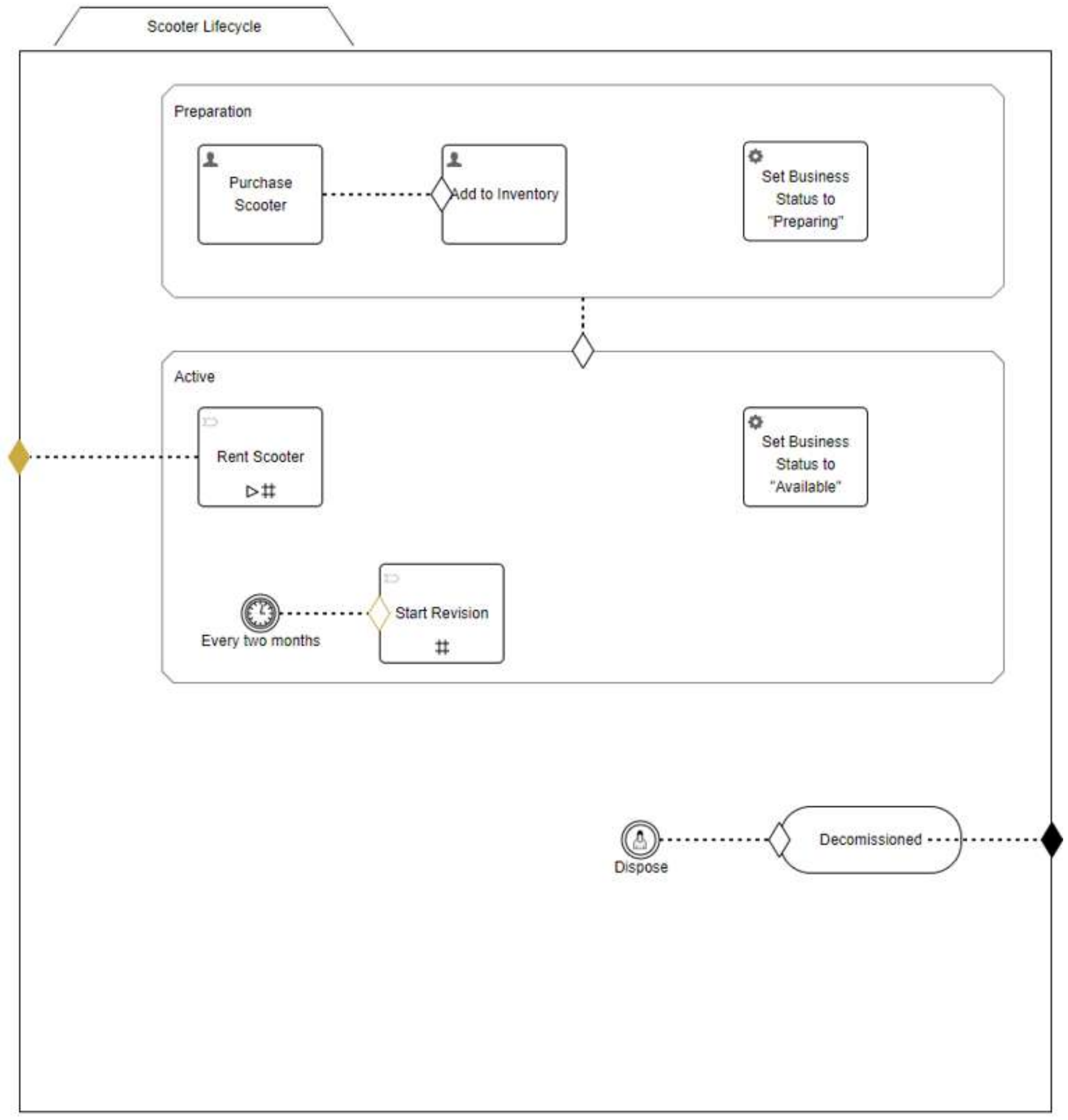
- **Start Events** represent the initiation point of a business process. They mark the beginning of a process flow and indicates the event that triggers the execution of the process.
- **Sequence Flows** represent the order in which activities, events, and gateways are executed within a business process. They visually represent the flow of control or directionality between BPMN elements, defining the path that process instances follow.
- **User Tasks** represent an activity in a business process that requires manual intervention or interaction by a human user. It means that a task must be performed by a person as part of the overall process flow. Most of the time this simply means that a (human) user must fill in a form. The data of that form is then used to control the workflow.
- **Gateways** represent points in a process where the flow can diverge, converge, or make decisions. They control the branching or merging of process flows.
- **Service Task** represent a task within a case that performs complex logic. It represents a unit of work that is executed by the system, invoking a specific piece of code in the Java backend. There are more specialized Service Tasks such as "E-Mail" tasks or "Generate Documents" tasks.
- **End Events** represent points in a process where the process flow terminates or reaches its completion.

CMMN Models

CMMN in a nutshell

The main purpose of using CMMN (Case Management Model and Notation) is to provide a **standardized** framework for modeling and managing **dynamic** and **unstructured** business workflows, often referred to as **cases**. CMMN is specifically designed to address scenarios where predefined process workflows are not suitable due to their unpredictable nature, variability, and need for human intervention.

The elements used to model the business logic in CMMN are defined as **Case Plan Items**. Human tasks, events, stages, sentries, milestones among others, are examples of case plan items within a case and each one of them has its own graphic notation.



Main elements

- **Stages** represent a container for organizing and grouping related activities, tasks, events, and other case plan items within a case. They serve as structural component that help to break down complex cases into manageable chunks. Often, a stage represents a "phase" of a case, e.g. "Preparation", "Investigation" and "Completion".
- **Human Tasks** represent a type of task within a case that requires an interaction by a (human) user. Most of the time they will require users to fill in a form whose data is then stored on the case. They are the main way how human end users interact with Flowable.
- **Service tasks** represent a task within a case that performs complex logic. It represents a unit of work that is executed by the system, invoking a specific piece of code in the Java backend. There are more specialized Service Tasks such as "E-Mail" tasks or "Generate Documents" tasks.

- **Process Tasks** represent a task within a case that is performed by invoking a BPMN business process.
- **Entry and Exit Sentries** are conditional logic elements used to control the flow within a case, based on the activation or termination/completion of Case Plan Items.
- **Decorators** are used to define the behaviour of case plan items. The following decorators are supported: Required, Manual Activation, Repetition and Auto Complete. They are indicated with an icon on the case plan item.
- **User event listeners** allow to react to user-generated events within a case instance. In concrete terms, they allow a user to trigger the start or end of case plan items.
- **Milestones** represent a significant point or achievement within the lifecycle of a case instance. They are mainly used for documentation purposes.
- **Timer events** represents an event that occurs at a specific point in time or after a specified duration within the lifecycle of a case instance.

CMMN vs BPMN Case Study

Up to now you have learned a few things about the standards and notation used to model business processes with BPMN and CMMN. Maybe, at this point you are wondering what you should use to model your business processes.

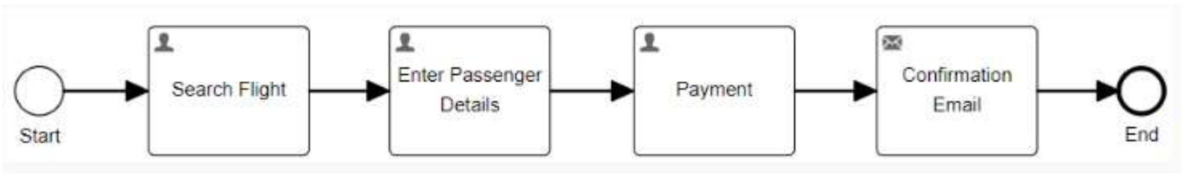
Recap

Let's not get confused by the word "business process"! It says nothing about the implementation and does not mean that we have to use BPMN. In the world of Flowable, processes are more oriented to structured flows, where tasks follow a well-defined, repeatable sequence. It's particularly useful for situations where the process flow is predictable and can be completely mapped out. This approach highlights flow control, clearly showing how tasks are connected between them.

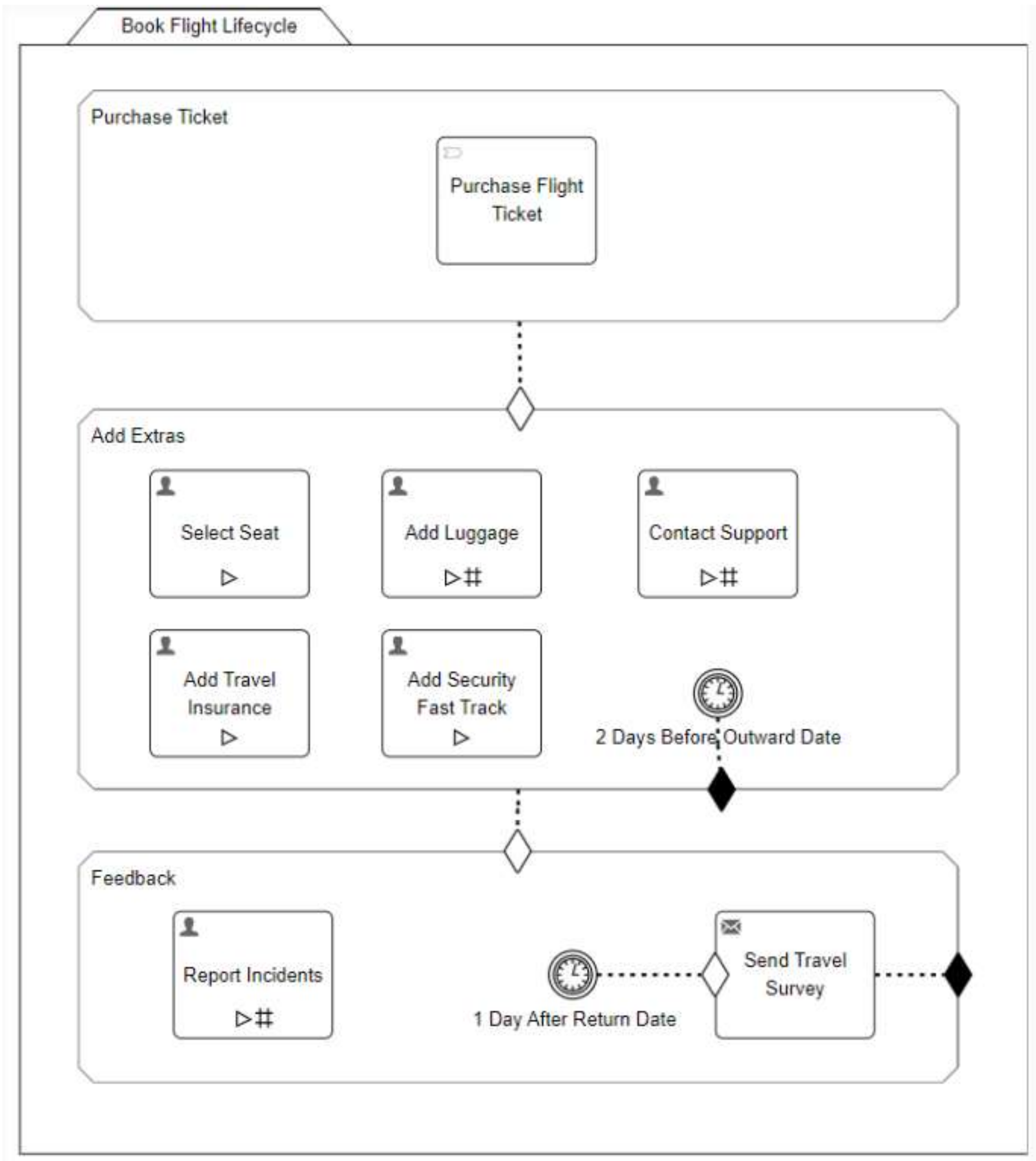
Cases are particularly effective for managing unstructured flows, where the sequence of tasks isn't always predetermined, and actions depend on changing circumstances. Unlike rigid workflows, cases support flexible execution, allowing tasks to be completed in a non-linear order based on the specific context and judgment of the user. This approach is ideal for managing the lifecycle of a case, rather than following a strict predefined process flow. Also, provides the adaptability needed for dynamic environments where traditional process modeling falls short.

“Book a flight” more a Process or Case?

If we think about booking a flight, maybe we can discuss if it is more a Process or Case, and probably we can’t get one final answer, but let’s focus on the ticket purchase part. The ticket purchase part is clearly a well-structured process. In the end, the purchase of a flight ticket is based on searching for a flight, entering the passenger details, making the payment and receiving the flight confirmation. All tasks are connected to each other and none of them can be started if the previous one is not completed.



However, if we think about everything that can be involved within a flight booking, a case can be the more suitable approach. Within the lifecycle of a flight booking there are different stages with numerous steps that the passenger can choose to complete or ignore, which shows that the flow might depend on the passenger's decision. Being able to add extras or contact the support team are examples of optional steps. Furthermore, these steps cannot be initiated if the life cycle does not allow it, i.e., in case the "Purchase Ticket" stage is not completed, the "Add Extras" stage cannot start.



As we can see in the above image, purchase ticket can be considered as a single stage in the whole lifecycle of a flight booking. Once the purchase ticket stage is completed, the case can move on to the next stage where the passenger can decide what wants to do, there is a workflow control without the need to follow a specific order. Also, considering the data entered by the passenger, the "Add Extras" stage is automatically completed depending on the departure date of the flight and moves on to the next stage. This makes sure that we can no longer book extra legroom after the flight date.

At the end of the day is the flight booking more of a Process or a Case? The answer really depends, if we are talking only about the purchase flight ticket, then it will be more of a process, but if we are talking about book a flight in general, we are talking more of a case.

In most cases, we recommend to start with a case, giving you the flexiblity to add dynamic elements later, if needed. You can always make use of Process Tasks to model most of the logic of your workflow.

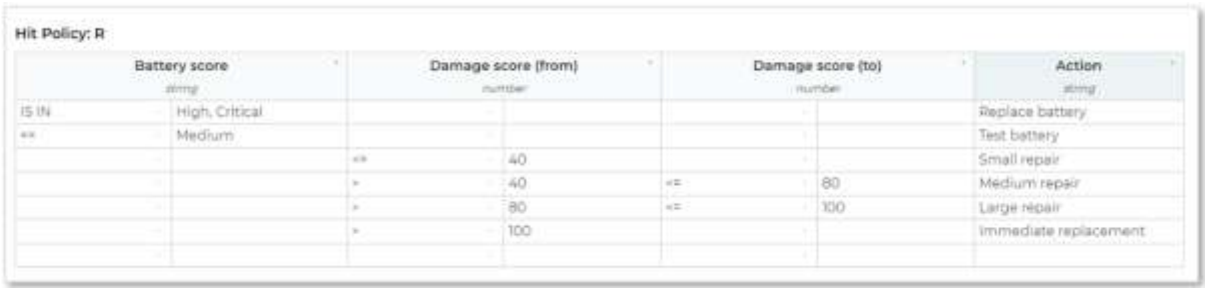
Finally, it is important to refer to the usability of both together. As we can see in this example, the purchase flight ticket remains a process but as part of the flight booking lifecycle. This capability brings flexibility and the ability to model complex use cases combining the best of both worlds.

DMN Models

DMN in a nutshell

The main propose of using DMN is for modeling and representing **business decisions** in a structured and standardized format. DMN provides a notation for expressing decision logic, decision requirements, and decision-making processes, making it easier to understand, document, and manage complex business rules and decision logic.

DMN Decision Tables can be invoked through "Decision Tasks" in both, CMMN, and BPMN.



Hit Policy: R

Battery score		Damage score (from)		Damage score (to)		Action
string		number		number		string
IS IN	High, Critical					Replace battery
==	Medium					Test battery
		<=	40			Small repair
		>	40	<=	80	Medium repair
		>	80	<=	100	Large repair
		>	100			Immediate replacement

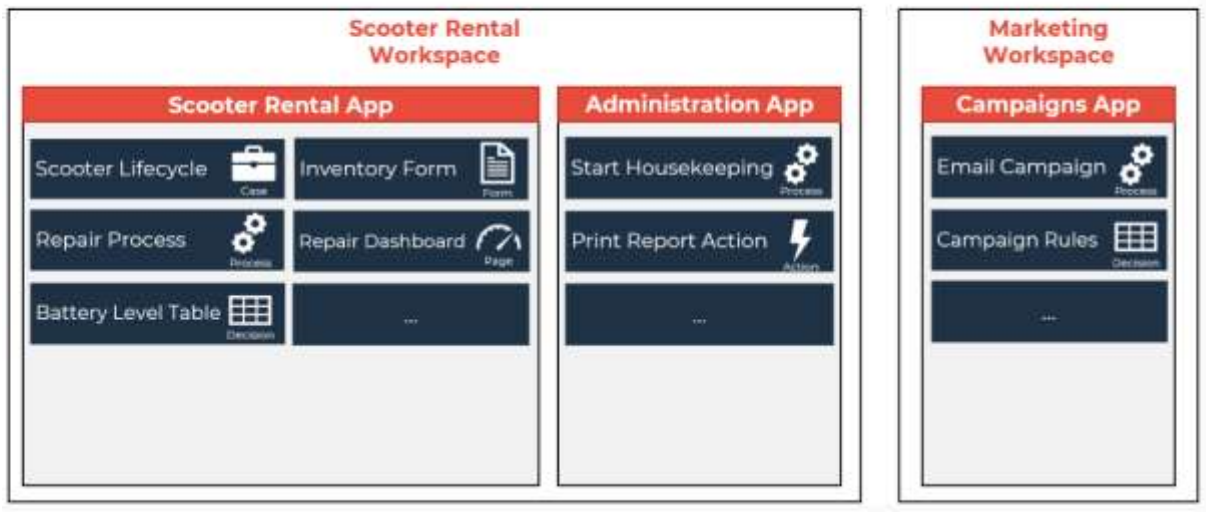
Main elements

- **Hit policies** specify how the decision table should handle multiple matching rules and select the appropriate result(s) to return as the outcome of the decision.
- **Input columns** represent the variables that are used to evaluate conditions and determine the outcome of a decision table.
- **Output columns** represent the variables that are used as output of a decision table.
- **Operators** are symbols or keywords used to perform operations or comparisons on data values within decision tables.
- **Input data** represents the information provided to the decision table for evaluation. It serves as the basis for making decisions.
- **Output data** represents the results produced by the decision table based on the input data and decision logic.

Structure

Inside Flowable Platform, the content created by modelers is structured according to the guidelines set by **Flowable Design**. This structure aids modelers to organize their implementation according to their specific requirements.

In Flowable Design, the hierarchy is delineated into three levels: **Workspaces**, **Apps**, and **Models**. **Workspaces** represent the highest level in this hierarchy and it is possible to have one or multiple Apps. **Apps** constitute the second level and can include one or multiple models, often several. Lastly, **Models** comprise the lowest level and contain all the necessary configurations as per the business requirements.



Workspaces

- A Workspace is an entity that **facilitating the grouping of apps**, enabling users to organize their projects based on their needs. Multiple Workspaces can be created to accommodate different projects or functional areas within an organization. Each Workspace can be configured to suit specific needs, with options to designate it as either **public** or **private**. This flexibility allows users to control access and visibility, ensuring that the content remains secure.
- Furthermore, Workspaces can have different permissions, **Owner**, **Modeler** or **Reader**, for certain **groups** of users or **specific users**. This feature provides an additional layer of control, allowing to restrict editing privileges and maintain integrity.
- Moreover, the capability to **share Models across Workspaces** enhances collaboration and speeds up the development. By designating a Workspace as shared, Models can be accessed by multiple teams or projects, promoting cross-functional synergy.

Apps

- In Flowable, the approach to **organizing business logic** revolves around the use of **Apps**. These Apps serve as containers for Models, such as processes, cases, or forms among others. It's important to note that Apps are exclusively dedicated to have **no-code** or **low-code** content and do not include any Java code. Apps can easily be shared by downloading an app bundle (business archive, BAR).

Models

- Models are the main way to build Flowable applications. In contrast to the traditional way to write software, using models allows to **speed up the implementation** process, since they are based on the no-code and low-code approach. There are several types of models, e.g. process models, case models, form models or action models. Each concrete Model has its unique **properties** configuration options tailored to suit specific business requirements.

Model to instance

One of the most important concepts in Flowable is the one of "models" and "instances". There are many types of models and thus instances. To explain the concept will concentrate on case instances in this module.

A process or case instance refers to an **individual**, **concrete representation** of a process or case within a system. It represents the execution of a defined set of tasks, actions or operations that are carried out to achieve a specific goal. Each process instance usually **follows a predefined sequence of steps**. This does not mean that every instance is the same, they can have different **variable values**, follow **different paths** etc.

A real-life analogy

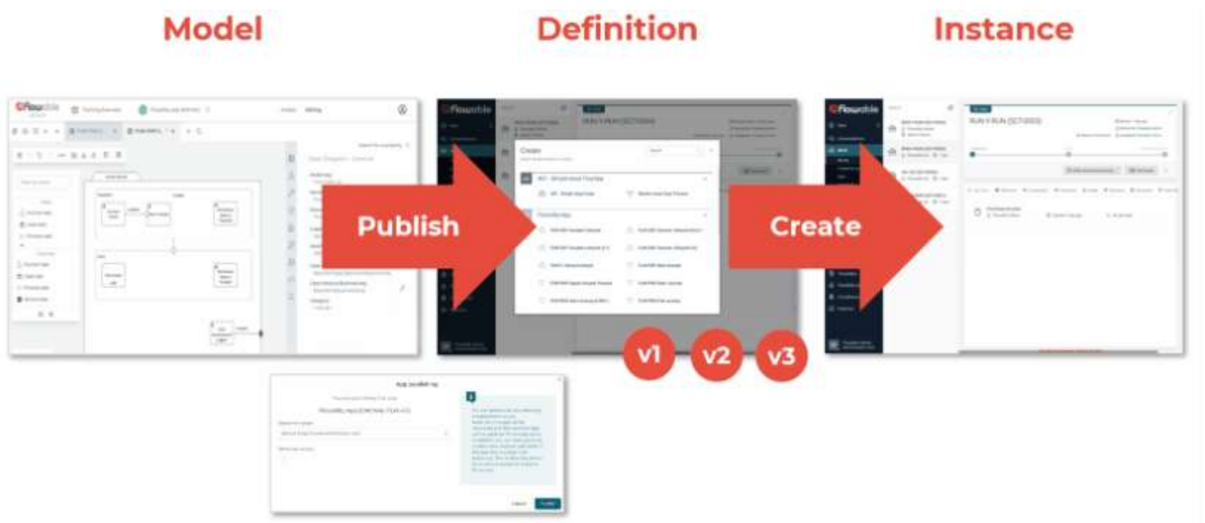
Complaints Case Lifecycle Model
Definition
Complaints Instance

Imagine a typical real-life scenario where the airline "*Flyable*" needs to implement a complaint management system. In this case, the first step for

Flyable is to model the Complaint's Case Lifecycle. This lifecycle should define all states of a complaint regardless of the type of complaint and the type of customer.

Once the design phase is completed, this model should be available for Flyable's customers to create complaints. When available, **each new complaint becomes a new instance** of the case lifecycle complaint. Each instance has an **independent lifecycle** and contains **unique data (variables)**. If the same customer creates two complaints, there will be two independent instances, where there may be repeated data, such as customer information, but there will also be different data, such as the creation date and time, instance identifier or even the claims' reason.

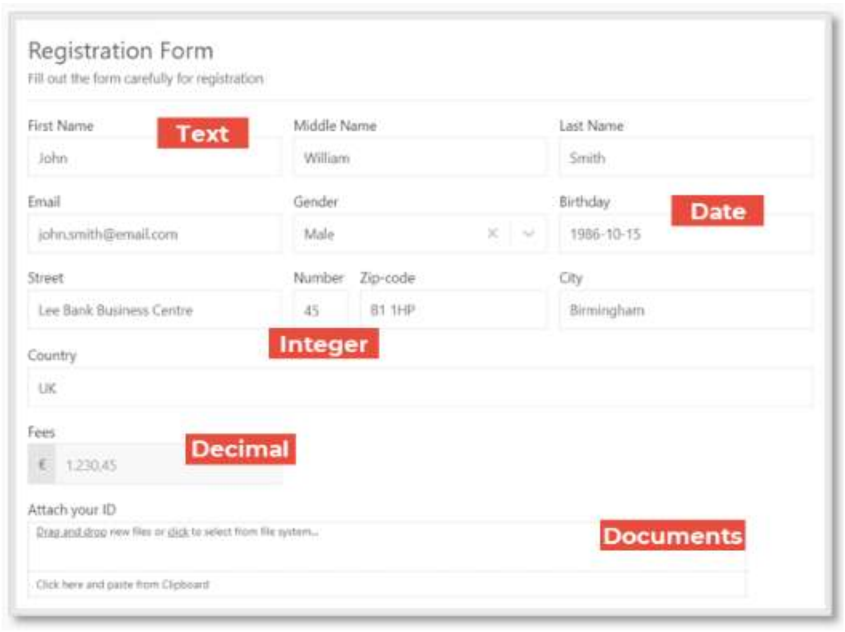
What does this mean in Flowable?



In Flowable the previous use case should be organized in the same way. In a first step we need to create a **case model**. When publishing that model and make it available to the users, a new **definition** will be created. Finally, users can now open new claims **instance** based on the last deployed definition.

Forms and Data

In today's environment data plays a crucial role. Its quality and accuracy have a direct impact on decision making, process optimization, compliance and governance, performance measurement and continuous improvement. By effectively leveraging data, organizations can **improve the efficiency**, **effectiveness** and **agility** of case management, processes and decision processes.


Registration Form
Fill out the form carefully for registration

First Name: John	Middle Name: William	Last Name: Smith
Email: john.smith@email.com	Gender: Male	Birthday: 1986-10-15
Street: Lee Bank Business Centre	Number: 45 Zip-code: B1 1HP	City: Birmingham
Country: UK		
Fees: € 1.230,45		

Attach your ID
Drag and drop new files or click to select from file system...
Click here and paste from Clipboard

In Flowable, there are different ways to enter data into processes, which will later be used to make decision, generate documents or integrate with external systems. Data entry happens by setting, updating and deleting **case and process variables**. Variables can be added at different points in a process, either through **user-filled forms**, through **integrations with third-party** systems, or **automatically generated** based on context.

Depending on the context, variables be of a different **type** which also offers a basic validation layer. Common variable types include **string** (text), **integer**, **decimal**, or **date**, among others. To model more complex data structures variables of type **JSON (JavaScript Object Notation)** or **Data Objects** can be used.

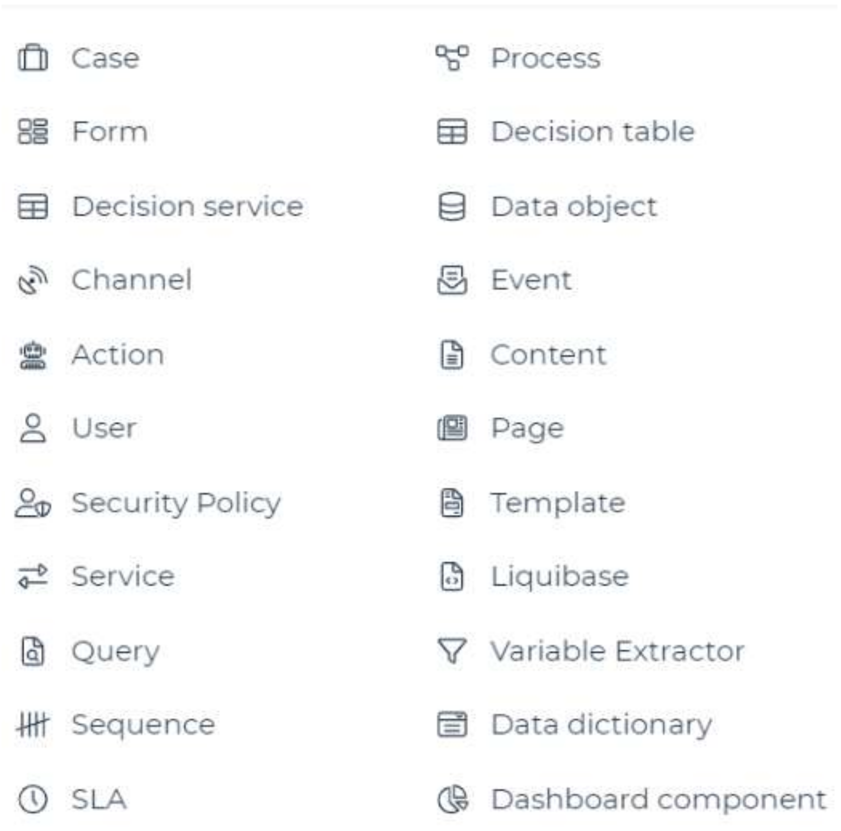
The proper use of variable types can **help to prevent errors propagation** to other systems. For instance, if an external system expects to receive a numerical value and that variable is set as string (text) in Flowable, when Flowable attempts to send a text value to the external system, an error will occur. Such situations lead to a poor user experience and undermine trust in the system.

Other Models

You've already discovered that **models can accelerate your development process** since they're built on **no-code** or **low-code** principles.

Configuring these models can vary depending on the type, with some allowing drag-and-drop configuration while others require adjusting various properties, like using a settings template.

Each model serves to facilitate the implementation of your business requirements. Sometimes, a single model be sufficient, while at other times, a combination of different models may be necessary. Additionally, you've learned that each model has its own set of properties, enabling you to tailor its behavior to meet your specific requirements. This section will provide an overview of all the models.



Basics about other models

- **Form** models represents a structured definition of a user interface that captures input data from users.
- **Decision table** models serve as a structured way to model complex decision rules, allowing organizations to automate business decisions.
- **Decision service** models offer a method for managing more complex decisions. Within a decision service, it's possible to combine various decision tables and establish dependencies between them.
- **Data objects** models represent structured data entities, allowing to define and manipulate data even outside of the lifecycle of a process or case. In Flowable there are two types, Database Data Objects and REST Data Objects.
- **Channel** models serve as connector that enable integration and communication between Flowable and external systems, services, or message brokers. In addition to channel models, there is an internal channel.
- **Event** allow to define small data structures which can be sent and received from and to channels.
- **Action** models allow to invoke scripts and Java logic from Flowable Forms and the Flowable Work frontend.
- **Content** models are used to define a type for uploading of a document or other file type. Content models can also be used to define a versioning scheme, or define metadata values, or define the type or state values for the content item.

- **User** models allow to define "user definition" which define important aspects of a cohort of users, e.g. groups, allowed features etc.
- **Page** models define the layout of UI pages, enabling organizations to design and customize user interfaces. Pages are very similar to Forms, so the user interface and behavior are almost the same.
- **Security policy** models provide a framework for managing and enforcing security measures to protect tasks cases, processes and tasks from unauthorized access.
- **Template** models define the structure, and content placeholders of templates to standardize and simplify the creation of documents, reports, or emails. Freemarker is the technology that Flowable uses behind the scenes.
- **Service** models represents a structured definition of reusable services or operations that perform specific tasks or functionalities within the platform. Services can be standard (types: expression, script, or REST), or data model based for data object models.
- **Liquibase** models define the changesets and configurations used to manage database schema evolution. Liquibase changelog is an external technology used internally in Flowable and used to manage Database Data Objects.
- **Query** models is a structured way for creating searches or criteria to fetch data from Elasticsearch/OpenSearch. It lays out the parameters, filters, and sorting preferences for retrieving data from processes or cases. It is possible to build a query using either a straightforward wizard-driven editor (Builder) or manually craft one query (Custom) using the Elastic Query syntax, with no limitations.
- **Variable extractor** models allow to control how complex variables are indexed into the Elasticsearch index. In addition, they allow to add variables to the full-text search.
- **Sequence** models provide a mechanism for generating unique identifiers for various purposes, such as process instance or case instances business identifiers, or sequences with a custom pattern.
- **Data dictionary** models are structured definition that manages the metadata, attributes, and definitions of data entities used across different processes or cases. In addition, it is also possible to define constrains to ensure data quality within processes or cases.
- **SLA (Service Level Agreement)** models are a structured definition that specifies the criteria, thresholds and rules that govern the service levels for a particular task within a process or case.
- **Dashboard component** models are a user interface element that presents summarized and interactive data visualizations related to processes, cases, or tasks. Flowable provides some ready-to-use dashboard components, but it is also possible build custom Dashboard components. Each dashboard component has its own type and query and can be used by business users in their own dashboards.

Quiz: Important Flowable Concepts

Question 1

BPMN is a common programming language that helps to implement processes.

○ True

○ False

Answer: False

Question 2

Which types of permissions can be defined in a Workspace?

- ☐ a. Modeler
- ☐ b. Reader
- [✓ c. Owner]
- ☐ d. Watcher

Answer: A, B, C

Question 3
In CMMN case models you can use sequences flows to control the flow within a case.

- ○ True
- ○ False

Answer: False

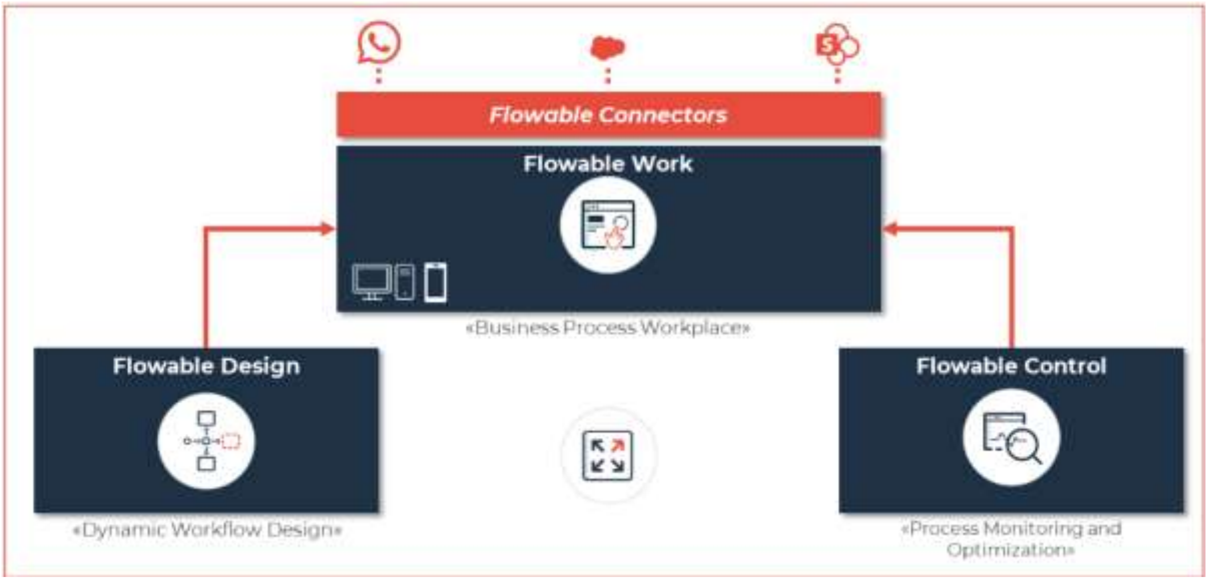
Question 4
DMN stands for...

- ○ a. Dynamic Measures and Notation
- [✓ b. Decision Model and Notation]
- ○ c. Dynamic Model and Notation

Answer : B

Flowable Platform

Flowable Platform Tools


Flowable Connectors
Flowable Work
«Business Process Workplace»
Flowable Design
«Dynamic Workflow Design»
Flowable Control
«Process Monitoring and Optimization»

Flowable offers a comprehensive suite of tools that synergize to create, deploy, and manage business applications. Understanding how these components interact is crucial:

1. **Flowable Design**: The starting point for creating applications. Use this tool to design and model your processes. It provides a user-friendly interface for defining workflows, forms, decision tables and many other types of models.
2. **Flowable Work**: Once your applications are designed in Flowable Design, they can be published to Flowable Work. This platform is where business users interact with the business processes defined in your apps.
3. **Flowable Control**: This tool is essential for ongoing maintenance and troubleshooting. It allows you to monitor existing instances and data. If there are any issues or bottlenecks in your processes, Flowable Control helps in identifying and resolving them.

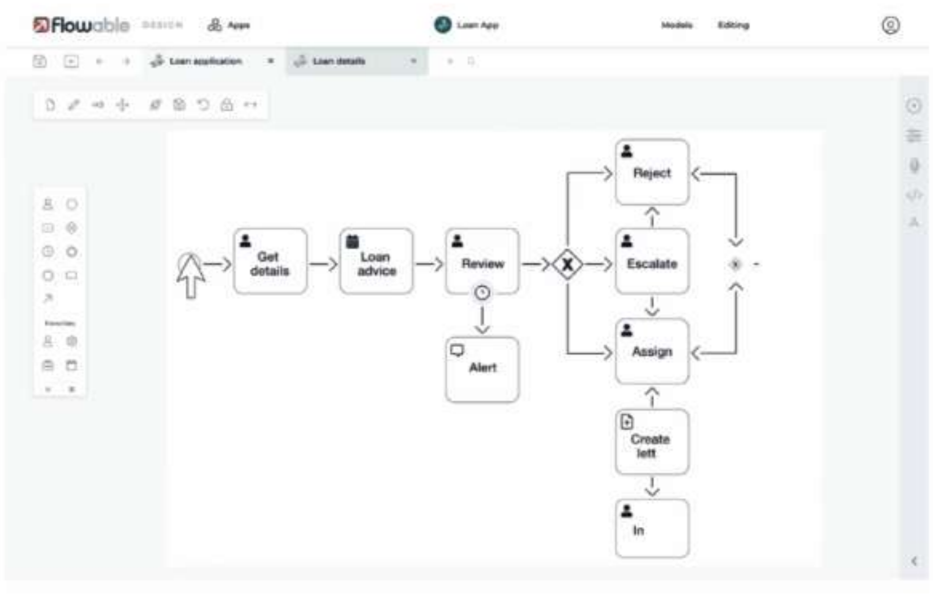
Each of these tools plays a distinct role as part of the Flowable Platform, they form a cohesive environment for developing, deploying, and managing business process applications.

Flowable Design

Introduction

Flowable Design is the main way to create and manage Flowable Apps. This is done through **models**. Think of them as the way how Flowable applications are configured.

Flowable supports three key standards: **BPMN** for business processes, **CMMN** for case management, and **DMN** for decision management. In addition, Flowable Design allows you to work with a number of other model types, e.g. Forms, Data Dictionaries or Data Object Models.



Key Features

1. **Visual Model Editing**: Flowable Design provides a user-friendly interface for creating process and case workflows. While simple to get started with, experienced modelers can craft robust models capable of handling complex scenarios.
2. **Workspaces and Apps**: Models are organized into Apps, serving as units of deployment. Workspaces facilitate the organization and separation of apps, with options for public, private, or shared settings, promoting collaboration and sharing across different workspaces.
3. **Data Dictionaries and Autocompletion**: Flowable Design offers a comprehensive view of all model data variables, crucial for design and review processes. Smart auto-completion features further assist modelers in efficiently managing process variables. This can further be enhanced by making use of Data Dictionary models which allow modelers to pre-define

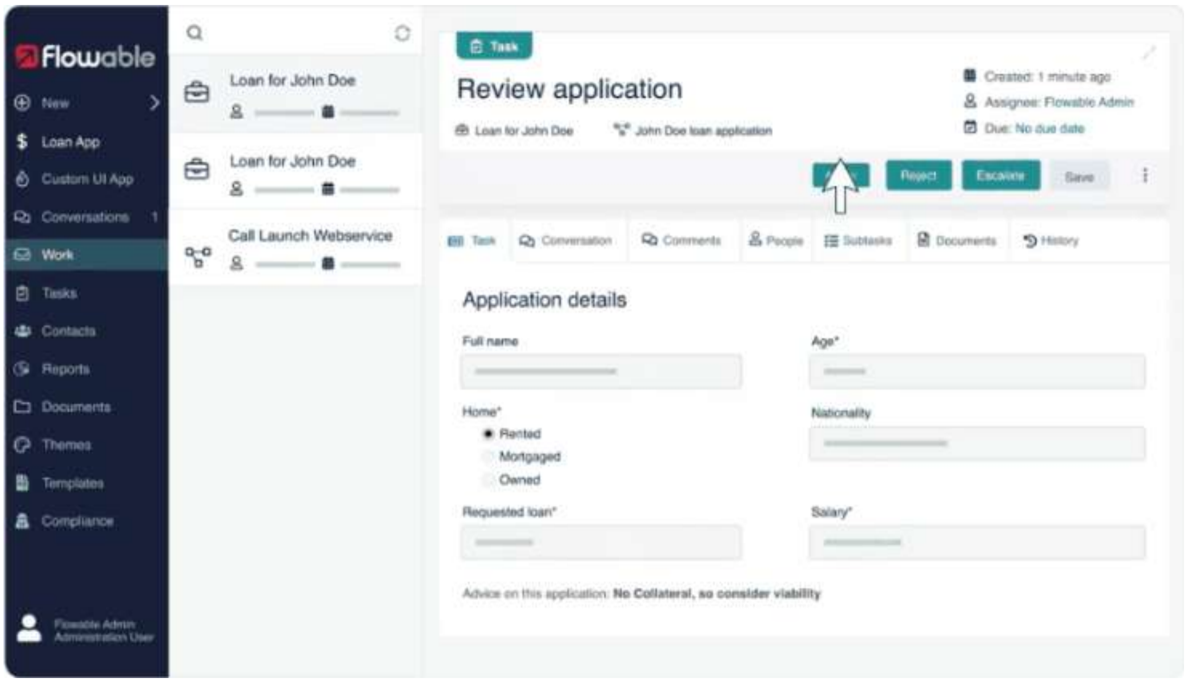
dynamic data models.

4. **Multilingual Support**: Flowable Design allows for defining user interface labels and model elements in multiple languages. Translations can be exported and imported as Excel files for efficient collaboration.
5. **Hot Deployment**: Flowable Design supports hot deployment ("publishing") of models. This allows business process modelers to immediately test and tweak their changes in Flowable Work during the development phase.

Flowable Work

Introduction

Flowable Work is a robust platform designed to efficiently manage complex business workflows, facilitating effective connections between customers and employees. Notable for its frontend flexibility, it offers customizable options to suit diverse use cases, while also providing a straightforward interface for immediate use.



Review application

Loan for John Doe   John Doe loan application

Created: 1 minute ago
Assignee: Flowable Admin
Due: No due date

Application details

Full name
Age*
Home*
[✓ Rented]
Mortgaged
Owned
Nationality
Requested loan*
Salary*

Advice on this application: **No Collateral, so consider viability**

Key Features

1. **Customizable Frontend**: Flowable Work offers an adaptable frontend, catering to a wide range of use cases. While it provides a generic interface for immediate feedback, it also allows for extensive customization to meet specific needs. Additionally, organizations can leverage Flowable Work's REST APIs for developing fully customized frontends tailored to their unique requirements.

2. **Mobile App**: The mobile app version of Flowable Work ensures an omnichannel presence, enabling users to stay updated and manage processes seamlessly from any location. Through the Flowable Engage connectors, users can furthermore be reached through a number of channels such as WhatsApp, WeChat or Line.

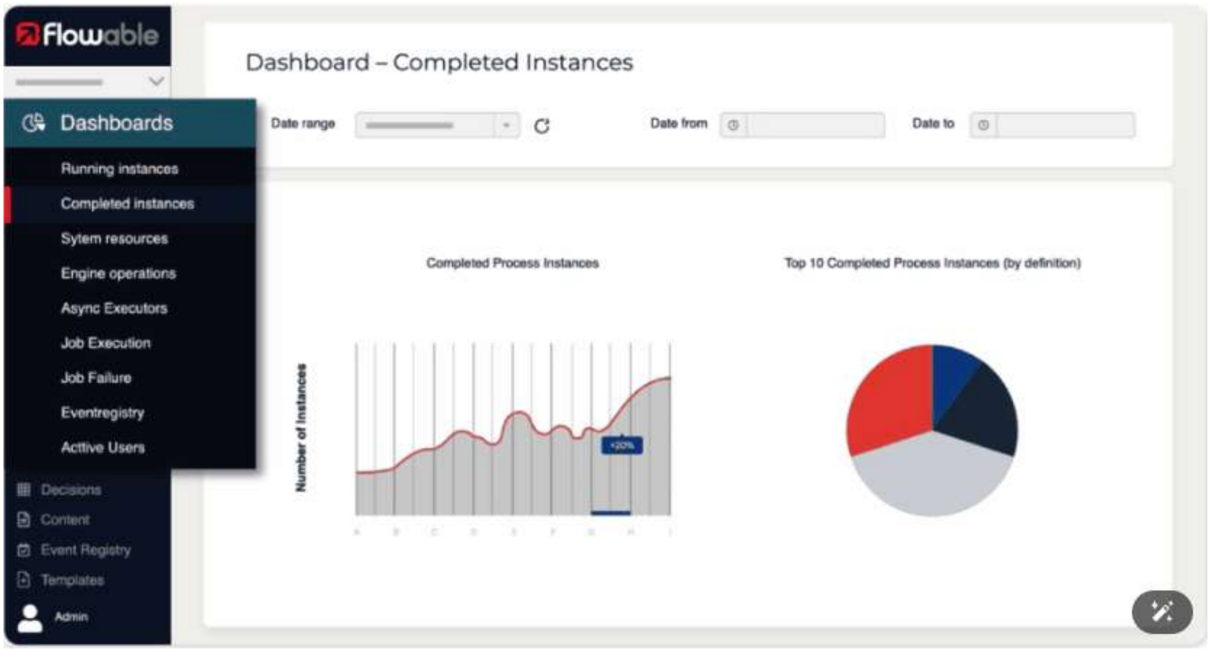
3. **Analytics, and Dashboards**: Flowable Work offers tools for creating customized dashboards, allowing users to track open tasks and monitor progress using various chart types like bar charts and pie charts.
4. **Enterprise Security**: Designed with a focus on security, Flowable Work allows the creation of user models and security policies to control data access and ensure compliance. Users and group management can be tailored precisely to each organization's needs.
5. **Task Management**: One of the cornerstones of the BPMN and CMMN standards is the interaction with humans. In Flowable Work, task forms can be claimed, filled, saved, reassigned.

6. **Content Management**: Flowable Work offers a simple content management system that allows users to view, update and organize their business process-relevant documents.

Flowable Control

Introduction

Flowable Control is a versatile tool designed for both administrators and developers, offering a range of features for effective monitoring and management of Flowable's runtime engines. It is essential for addressing issues and adapting to unforeseen changes or errors.



Key Features

1. **Application Monitoring and Real-time Engine Insights**: This encompasses in-depth inspection of Flowable engines for system health, alongside real-time monitoring with detailed metrics like CPU and memory usage, database connections, and user activity.
2. **Process and Case Management**: This includes the ability to change the state of processes and cases, particularly useful for correcting paths and rerunning processes, and managing failed jobs by listing and restarting operations as needed.
3. **Instance Inspection and Migration**: Users can inspect and manage a range of instances (processes, cases etc.) and migrate these instances to updated model versions facilitating migrations.
4. **Detailed Data Analysis and Server Management**: Flowable Control offers comprehensive data exploration of cases, processes, tasks, and more, with capabilities to manage multiple Flowable servers and tenants for configuration and state modifications.
5. **Enhanced Security with Fine-Grained Permissions**: A robust permission system underpins all features, ensuring secure access and modification of business data by authorized personnel only.

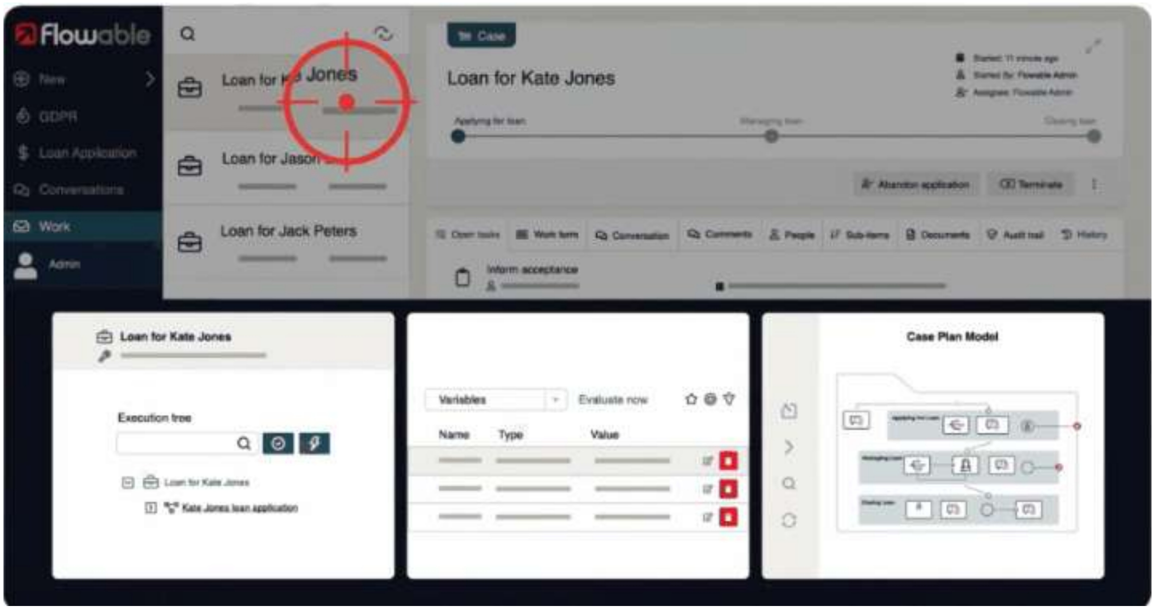
Target Audience and Applications

Flowable Control's dual functionality makes it invaluable for both administrators and developers. It enables real-time monitoring and modifications, providing critical insights and control over Flowable-based applications through all phases, from deployment to maintenance. This dual utility underscores its role as a critical component for effective application lifecycle management.

Flowable Inspect

Introduction

Flowable Inspect is a productivity-boosting tool within the Flowable Work UI designed for testing and debugging form, process and case models. It addresses the usual challenges in model development by providing a sophisticated environment to step through executions, manage variables, and test various scenarios.



Key Features

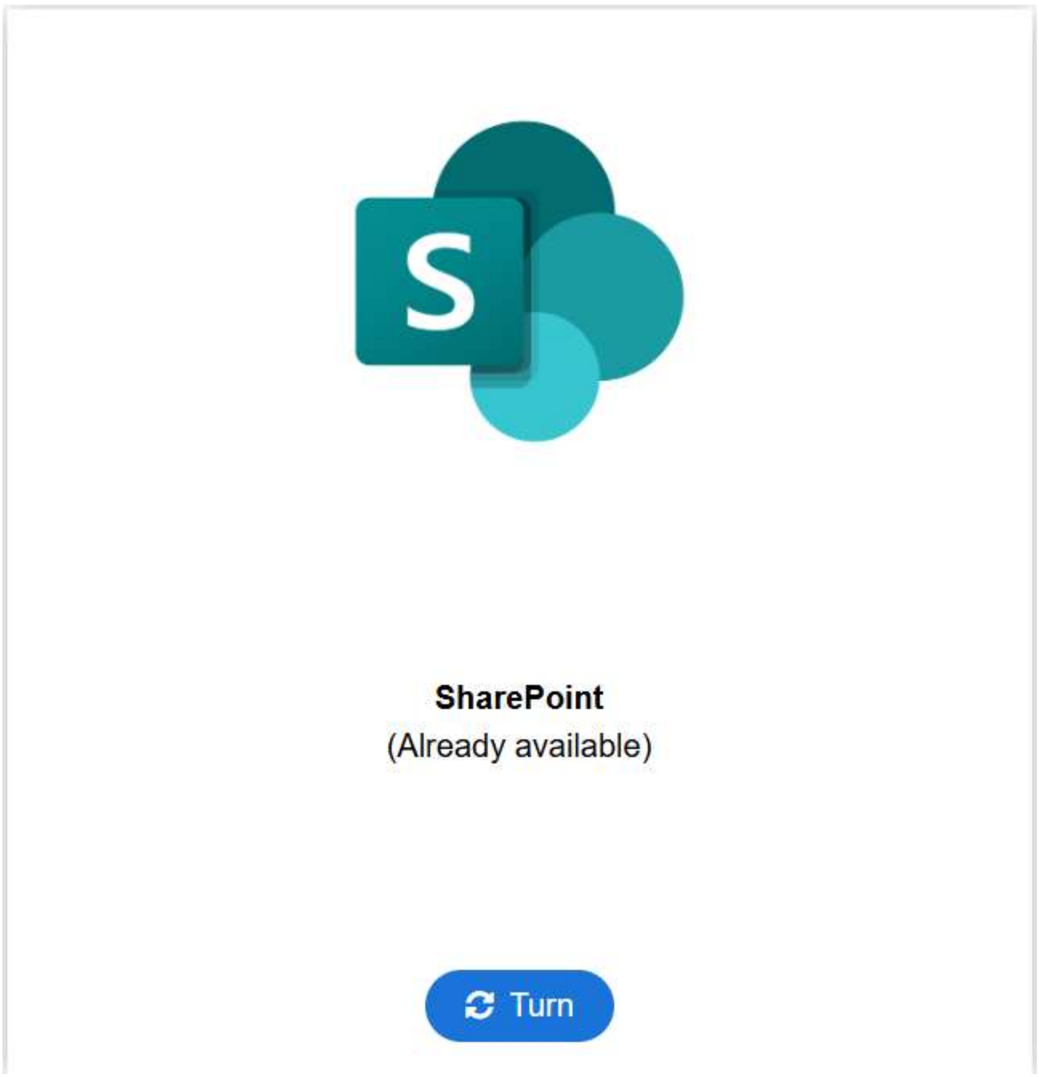
1. **Execution Inspection and Management**: At any stage of case or process execution, users can open the Inspect panel to navigate the execution tree, alter variable values, or trigger events like timers.
2. **Recording and Replaying Test Suites**: Flowable Inspect allows for the recording of different execution paths, including user interactions and manual events. These recordings can be replayed to validate changes in models, ensuring consistent functionality.
3. **Step and Breakpoint Debugging**: Users can add breakpoints during execution, enabling them to jump through the process or step through it incrementally. This feature is vital for in-depth analysis and troubleshooting of specific segments of a model.
4. **Validation through Test Actions**: Test actions, including variable value checks and condition expressions, can be added to test definitions. When replayed, any failure in these test actions is reported, aiding in pinpoint accuracy in debugging.
5. **Form Inspector and Debugger**: Flowable Inspect provides insights into the structure of user forms, including components and subforms. Users can modify attributes and values to test dynamic interactions and assess the form's functionality and appearance.

Usage and Applications

Flowable Inspect is particularly useful during the development and testing phases of process and case models. Its capabilities to step through executions, record and replay test scenarios, and debug at a granular level make it an essential tool for developers looking to streamline and refine their models within the Flowable environment. By enabling thorough testing and debugging, Flowable Inspect ensures that models are robust, efficient, and error-free before deployment.

Flowable Connectors

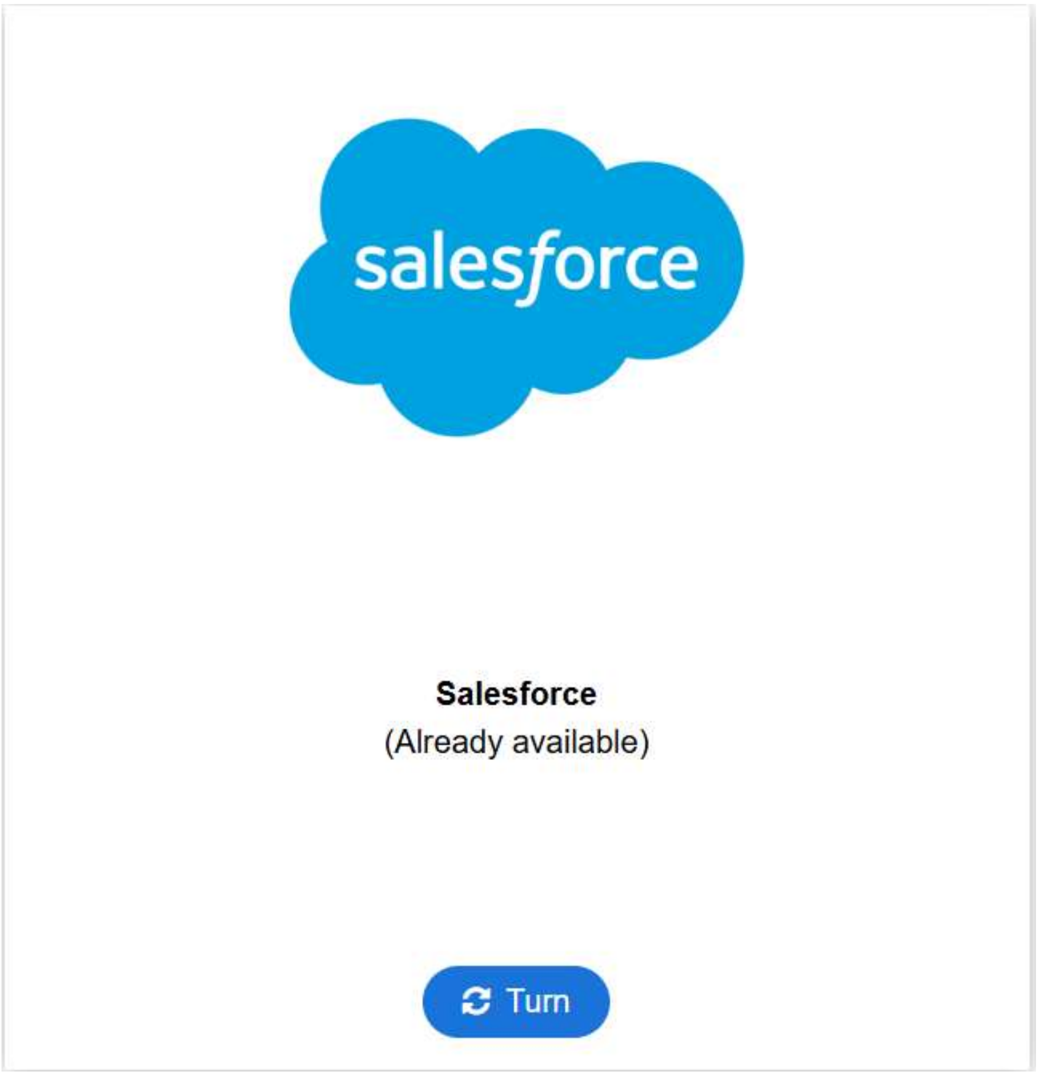
Flowable provides **connectors** that allow users to integrate with different market applications quickly and easily in a way to increase the level of process automation. These out-of-the-box connectors provide a tailored experience that goes beyond the generic integrations already possible through the Flowable Service Registry.



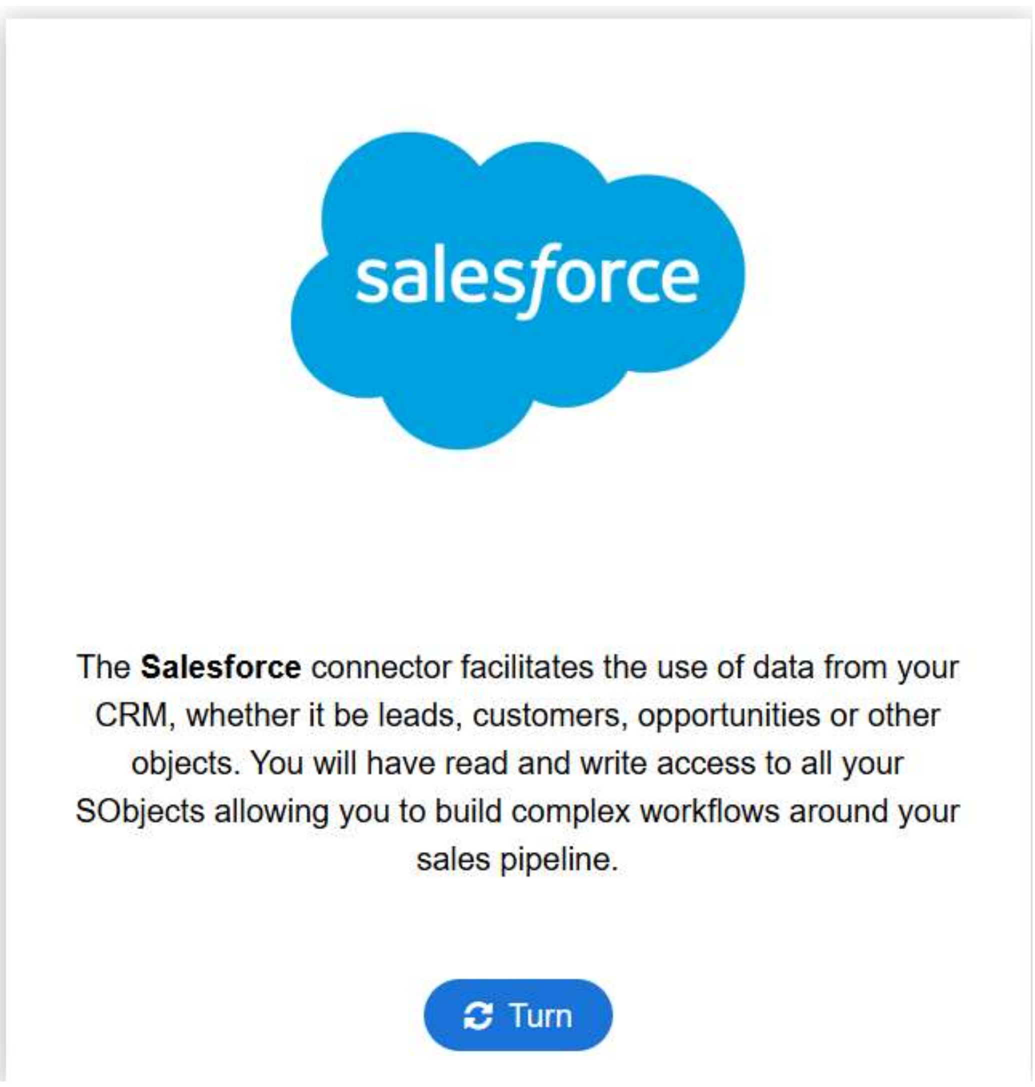
SharePoint
(Already available)

The **SharePoint** connector makes it possible to integrate the Flowable Forms **attachments component** with a SharePoint server. Using this connector, users can decide where they want to save the documents of a particular case or process, so it is possible to ensure a **centralized document repository** in an straightforward way.

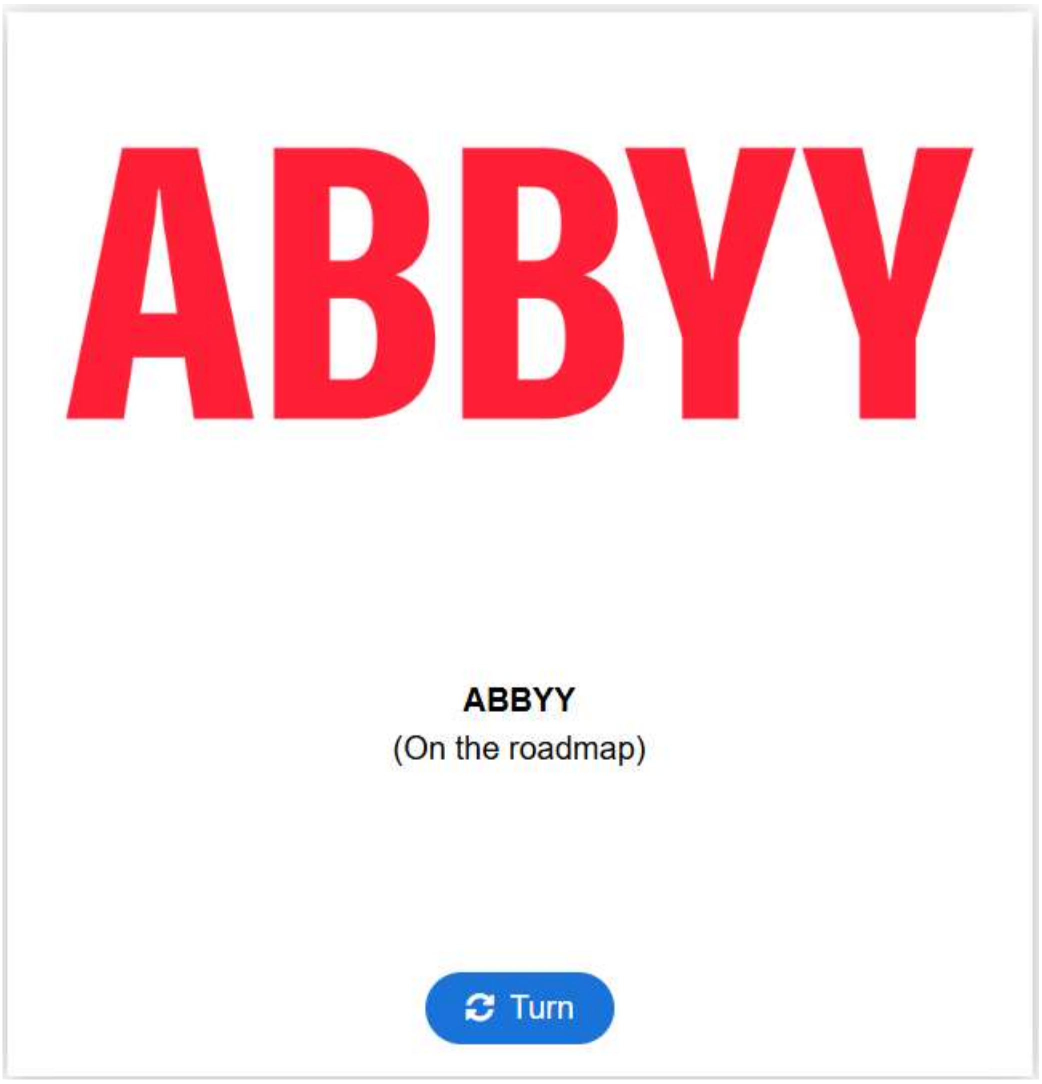
Turn



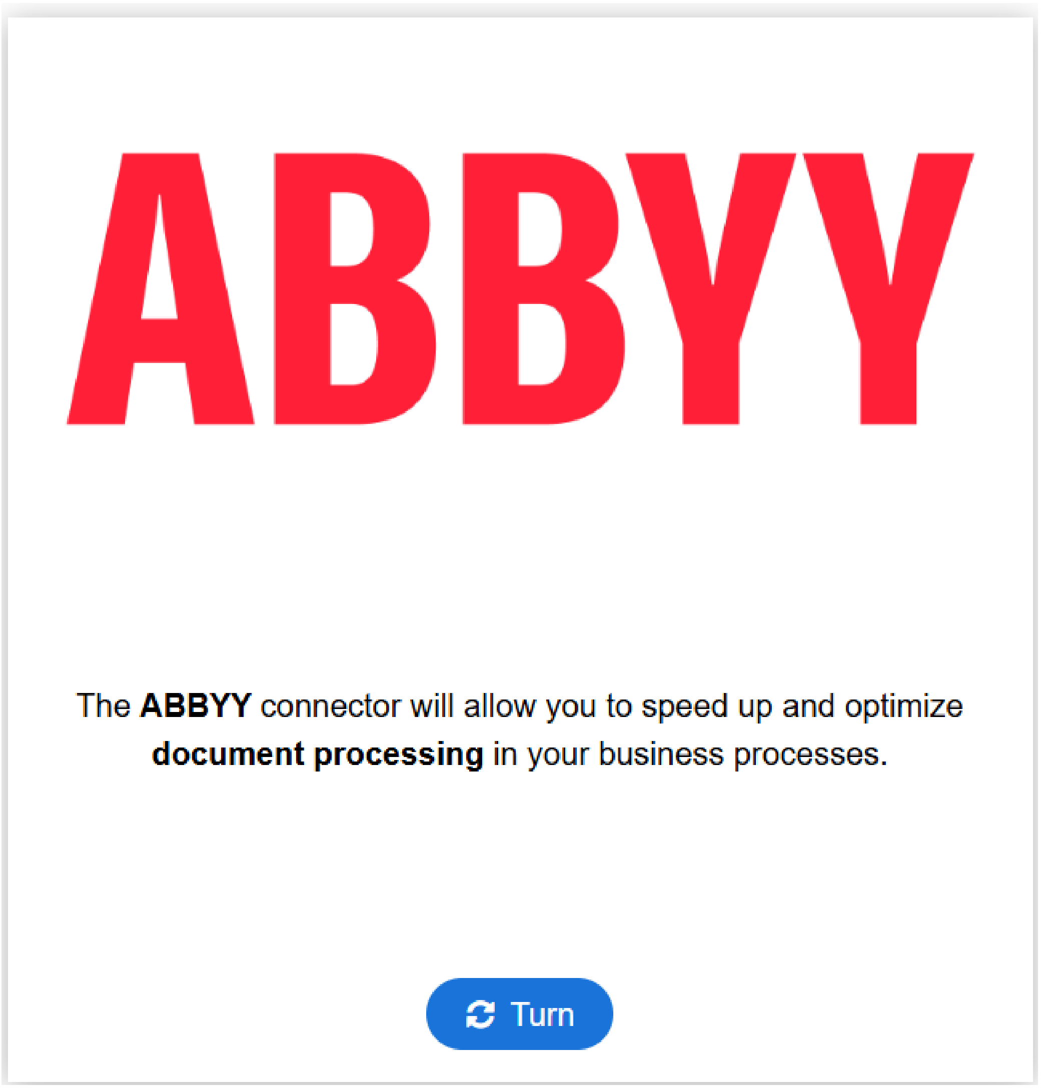
Salesforce
(Already available)



The **Salesforce** connector facilitates the use of data from your CRM, whether it be leads, customers, opportunities or other objects. You will have read and write access to all your SObjects allowing you to build complex workflows around your sales pipeline.



ABBYY
(On the roadmap)

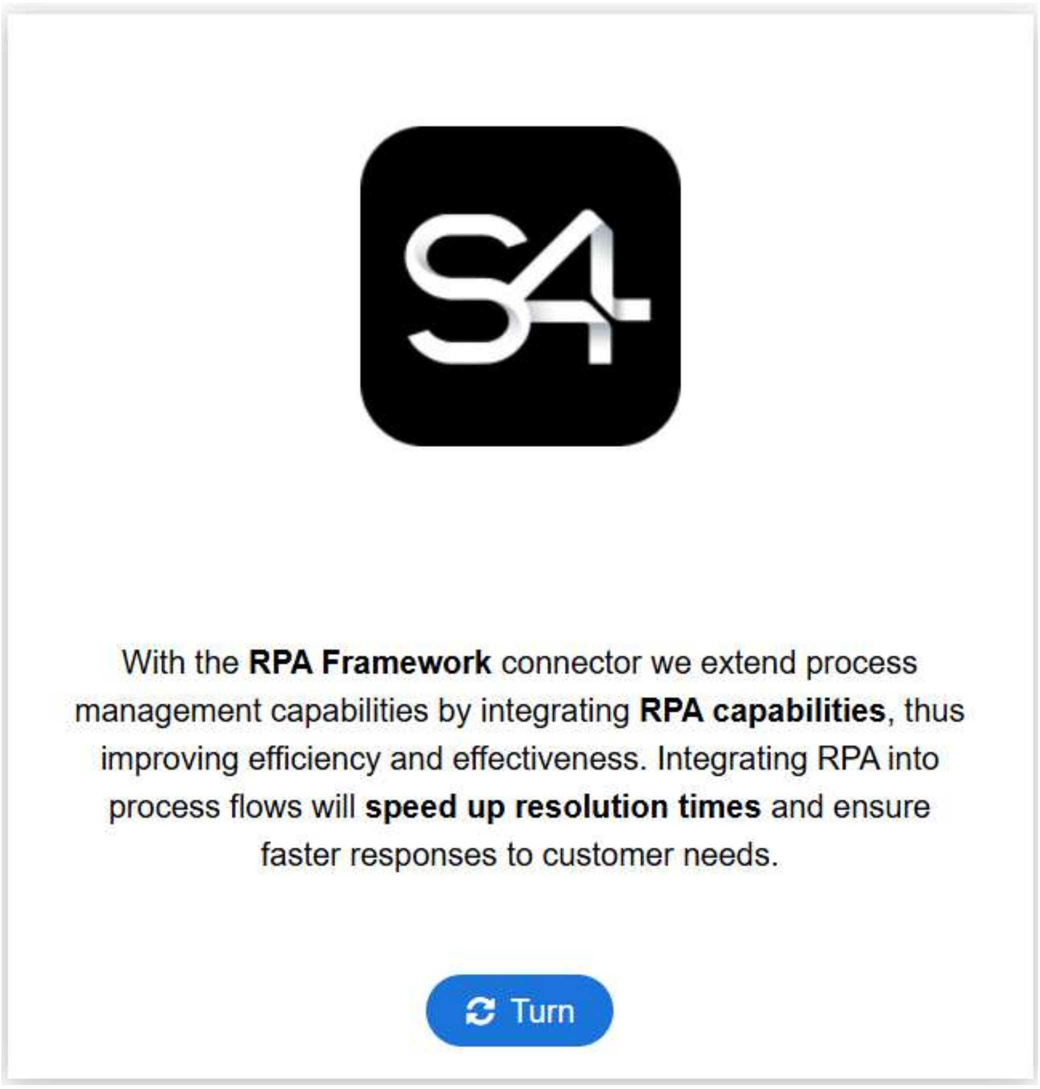


The **ABBYY** connector will allow you to speed up and optimize **document processing** in your business processes.

UiPath

(On the roadmap)

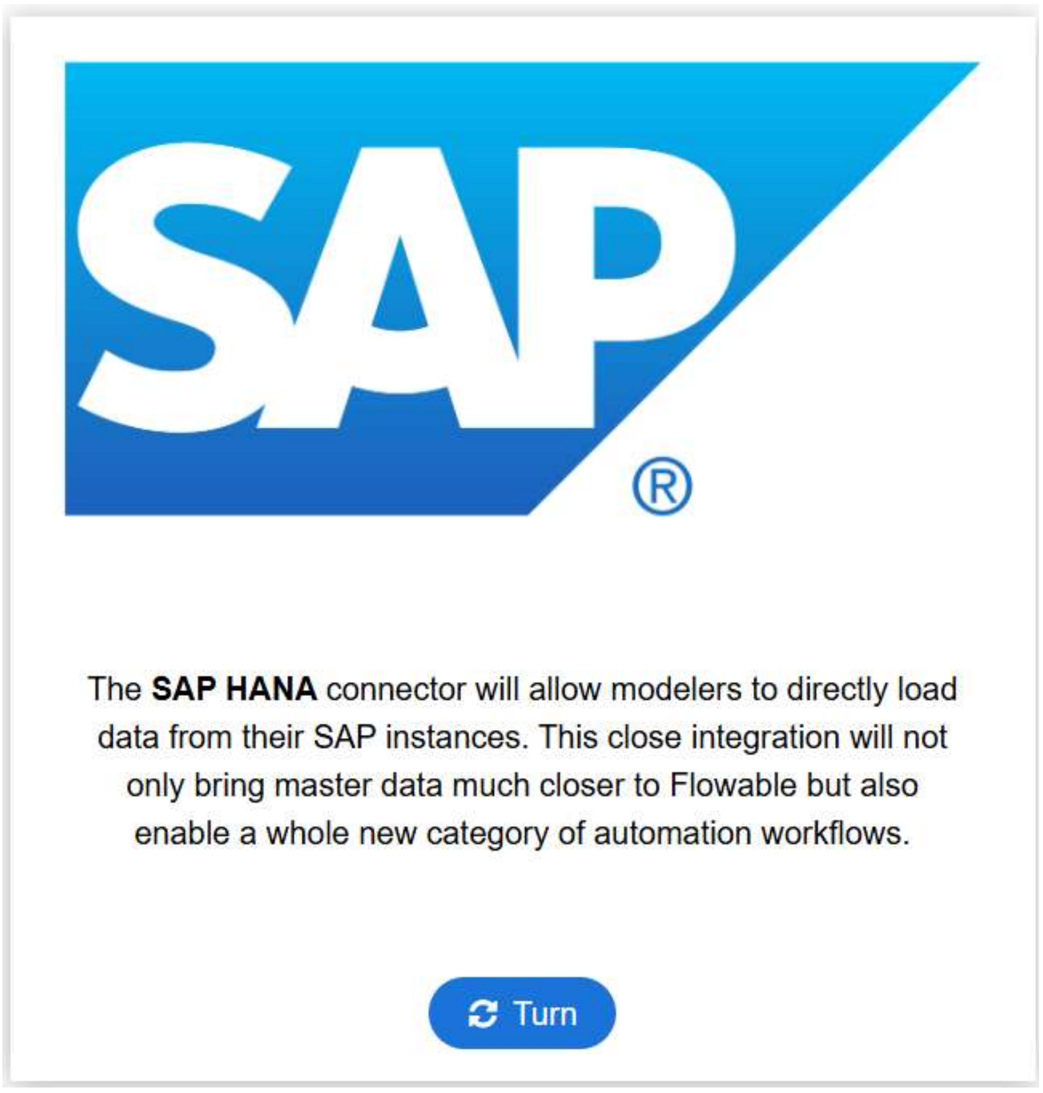
Turn



With the **RPA Framework** connector we extend process management capabilities by integrating **RPA capabilities**, thus improving efficiency and effectiveness. Integrating RPA into process flows will **speed up resolution times** and ensure faster responses to customer needs.

SAP
(On the roadmap)

Turn



The **SAP HANA** connector will allow modelers to directly load data from their SAP instances. This close integration will not only bring master data much closer to Flowable but also enable a whole new category of automation workflows.

Flowable Engage

Introduction

The Flowable Engage Connectors are designed to enhance customer engagement by integrating with popular messaging applications. They allow seamless, frictionless conversations with customers through platforms like **WhatsApp**, **WeChat**, **Line**, and more.

Key Features:

1. **Multichannel Customer Interaction**: Engage customers on platforms like WhatsApp, WeChat, Line, etc., while ensuring secure and compliant communication, including e-Discovery for complete searchability and legal compliance.
2. **Tight Integration with BPMN and CMMN**: Start, update and interact with BPMN processes and CMMN cases from your conversations. Further enhance interactivity through "Slash actions" which can trigger scripts and Java logic.
3. **Enhanced Group Communication**: Open and moderate group conversations, allowing for controlled multi-participant discussions.
4. **Efficiency and Insight Tools**: Combine user lifecycle management with productivity-enhancing features like message templates, slash commands, and comprehensive reporting & analytics for a clear overview of customer interactions.