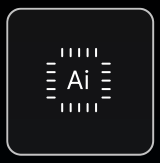
Flowable



Flowable Agentic Case Platform

One governed platform for people, systems and AI agents

Build applications visually from cases, processes, decisions, forms and data. Run them with your people, your systems and your AI agents. Govern everything from one place.

Contents:

Flowable puts cases, processes, people, systems, documents and AI agents on one platform. This datasheet describes what the Flowable Agentic Case Platform does, chapter by chapter, from modeling an application to operating it in production.

BUILD TEST PUBLISH RUN OPERATE

INTRODUCTION

Platform overview

The Flowable Agentic Case Platform is a platform for building and running work that mixes fixed sequences with steps that depend on the situation: onboarding a client, settling a claim, closing a compliance review. Parts of it repeat the same way every time, parts of it change with each case, and all of it runs for weeks, months or years across many people, documents and systems. Flowable gives it a structure: a governed case that holds the data, documents, people, rules and connected systems, with processes for the repeating parts and AI agents that work under the same permissions and audit trail as everyone else.

Flowable brings people, systems and AI agents together around your cases and processes, on open standards, with one set of rules and one audit trail for all of them.

600+
enterprises run their work on Flowable

1 bn+
events per day at the largest installations

29
model types in one modeling environment

3
open standards: BPMN, CMMN and DMN

WCAG AA
2.2, externally audited accessibility

ISO 27001
SOC 2 Type II, ISAE 3402

Who it is for

Operations and business leaders
See every case, its status and its bottlenecks, and change how the work runs by changing the model.

Business analysts and modelers
Model cases, processes, rules, forms and data visually, with AI assistance, then test them and publish them under approval.

Architects and developers
Open standards, REST APIs for every engine, extension points in Java and React, and core engines you can read on GitHub.

Security, risk and compliance
Permissions enforced on every call, a business audit trail, platform-enforced four-eye checks where necessary, and certified operations.

IT operations
One console to monitor, fix, migrate and retire running work across environments, on your infrastructure or ours.

AI and innovation teams
Put agents to work inside real cases, on the AI model of your choice, with the same permissions, audit trail and tests as everything else.

Typical solutions

Client onboarding and KYC (know your customer)

One case per client. Documents arrive by upload or email, due diligence steps activate according to risk, compliance approves under the four-eye rule, and the audit trail explains every decision. Agents read the documents and propose the next step for a person to confirm.

Claims and applications

Coverage is checked with decision tables, missing evidence is requested, payment runs through the core system over a modeled service, and every step is measured against its service level.

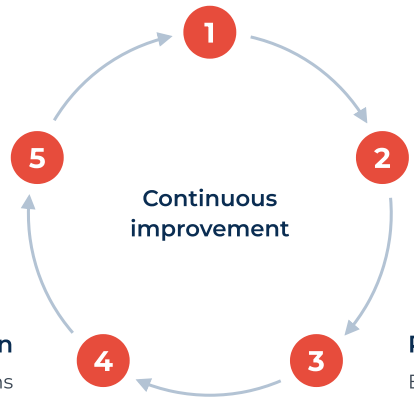
Incidents and complaints

An incident is logged, escalates on deadline, involves specialists as participants, grows with unplanned tasks as it unfolds and closes with a generated report.

How work moves through the platform

Build

Business and IT model cases, processes and decisions in Flowable Design, with AI assistance, together with forms, data, integrations and the other model types.



Test

Token simulation in the editor, tests modeled or generated by AI in Flowable Design, and step-by-step debugging catch mistakes before go-live.

Publish

Every use case ships as one versioned app, through approval, to test and production.

Run

People, AI agents and your systems work the cases in Flowable Work or behind your own user interface.

Operate

Operators use Flowable Hub to see what runs, fix what is stuck and migrate existing instances to new versions. What they learn goes into the next model version.

WHAT IS A

Model

In this datasheet, a model is something you build in Flowable Design: a case, a process, a decision table, a form, a data structure, an agent or one of the other model types. Models are the representation of your business logic, readable by business and IT, versioned together and published as an app. To model means to build them. Where an AI model such as GPT or Claude is meant, the text says AI model or LLM, short for large language model.

Where it runs

Flowable Cloud

Flowable runs the platform for you, with upgrades, patches, monitoring and backups included and an availability SLA. The fastest way to start.

Self-managed

You run Flowable in your own data center or cloud, on Kubernetes, OpenShift or virtual machines, with your database, storage and region. Everything stays inside your perimeter.

The platform

Flowable Design

Where applications are built. Analysts and developers model cases, processes, decisions, forms, data, integrations and agents together, with AI generating and refining models from the editors or from a coding assistant such as Claude Code or Codex connected through the open Model Context Protocol (MCP), then test them and publish them as one Flowable app per use case.

Flowable Work

Where the work runs. Case workers, automated steps and AI agents do their part, in the browser, in the Flowable mobile app for iOS and Android, fully headless through the REST API behind your own user interface, or through an AI assistant connected to it.

Flowable Hub

Where operations keep control. Administrators monitor running work, fix problems, migrate instances, manage configuration and watch agent activity and cost.

Everything in this datasheet is a capability you choose to use where it fits. A form carries an AI button only if a modeler adds one, and an agent proposes the next steps of a case only if you give the case one. People, systems and AI agents take part in the same cases under the same rules, and organizations that do not want AI in their workflows run the same platform without them.

Entries marked with ✦ describe the AI capabilities.

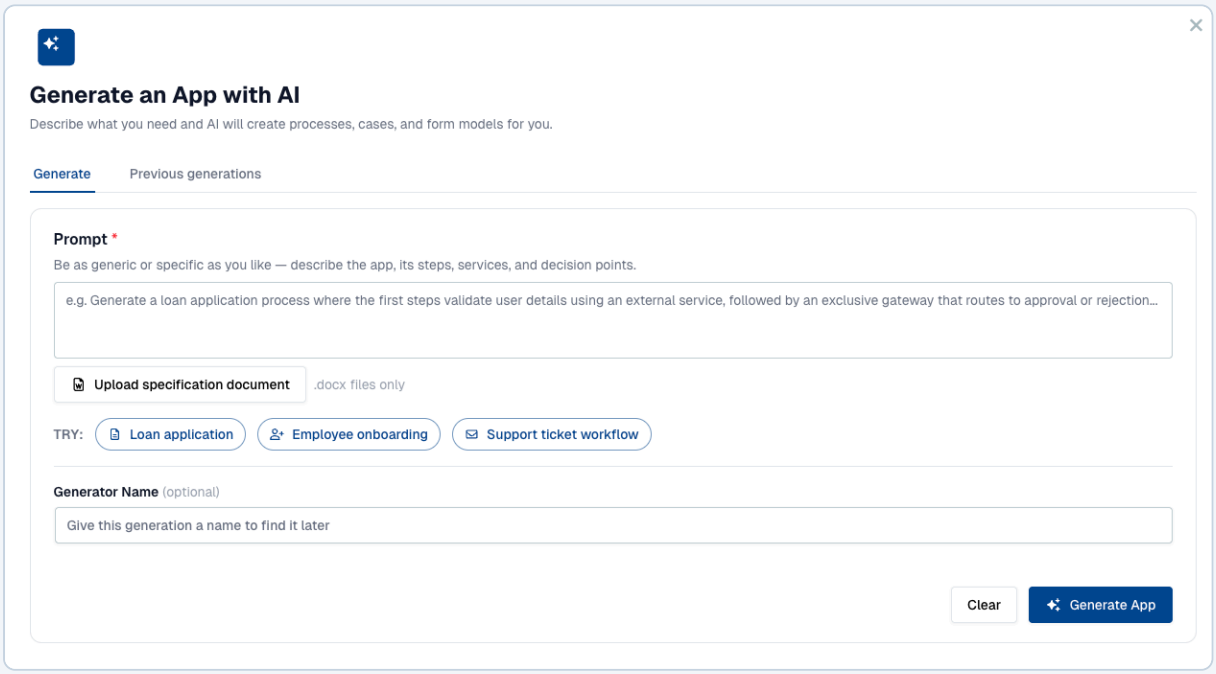
FLOWABLE DESIGN

01 Design and build applications

Most automation projects stall between the workshop and the code. In Flowable Design you build an application by modeling it, by describing it and refining the result in conversation, or both. Cases, processes, decisions, forms, data, integrations and AI agents are all models, built by business analysts and developers side by side, versioned together and published as one app, and that app is what runs in Flowable Work. Changing the business logic means changing the model, with the same controls software teams expect.

AI-assisted modeling

AI assistance runs through the whole of Flowable Design. You can generate a complete application from a description or a specification document, build or change any single model in conversation inside its editor, work from the coding assistant your developers already use, draft the instructions of an agent, generate tests and let the layout take care of itself. Your organization's own rules shape what the assistance produces, and everything it produces is an ordinary model that you review, version and publish like any other.



Step 1 of app generation: describe the app in plain language, start from an example or upload a specification document.

An application from a description or a document

Describe the business scenario in plain language, start from an example or upload a specification document, and Flowable Design builds the app in a guided wizard with seven stages: understanding your prompt, proposing the model structure, checking the services and agents involved, generating the data dictionary, processes and cases, generating the forms, testing and validation, and import. The wizard pauses after each stage for review, and the result is a normal, fully editable app that reuses your existing services and data structures.

Generated, tested, fixed, tested again

Generation does not stop at the models. The wizard also generates tests for the app, deploys everything to a connected Flowable Work runtime and lets you run the tests with Inspect, the testing and debugging toolset described in the chapter Operate, test and improve. When a test fails, you get an analysis of the failure and a fix, in the model or in the test, redeployed and ready for you to run the tests again. Each cycle pauses for you, so obvious problems are caught and corrected before the app is ever imported into your workspace.

An assistant inside the editors

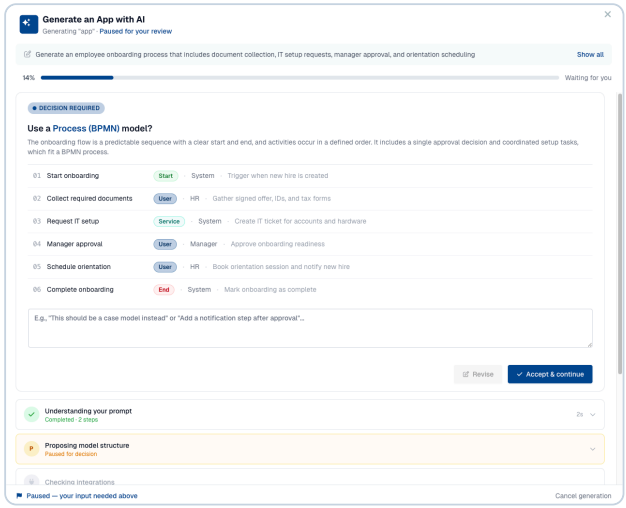
The modeling editors include an AI chat that generates a process, case, form or data dictionary from a description and refines it in conversation, adds elements, changes properties or applies bulk changes from a prompt, drafts scripts and REST and script service definitions including their request and response handler scripts, translates user-facing text, and answers questions across an app such as which forms use a given variable. The chat runs on the LLM provider you configure, including Anthropic's Claude.

Diagrams that arrange themselves

You do not tidy a generated diagram by hand. Whether a process or case comes from the chat, from a coding assistant over MCP or from your own quick draft, auto layout arranges it in one step: elements aligned, flows routed, stages sized to their content. The assistance decides the structure of the model, the layout engine decides how the diagram looks, and the result stays readable for the people who review it.

✦ The same help when building an agent

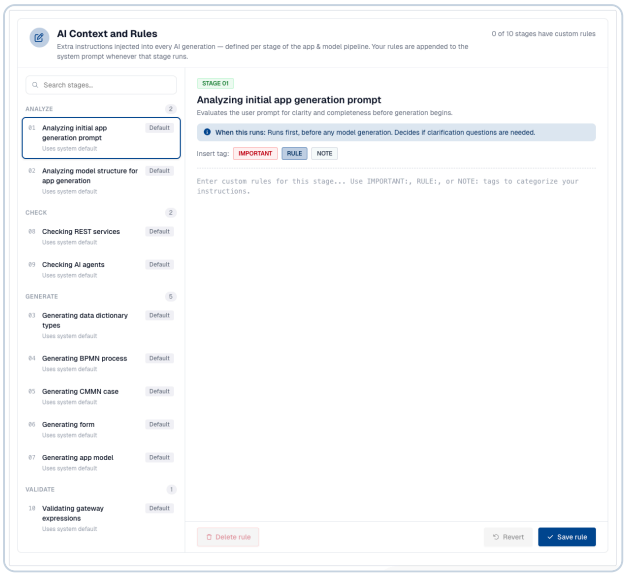
When you build an agent, you describe the job and the information it receives and returns, and you get a first version of its instructions to refine in conversation. You try out a document agent on your own sample documents in Flowable Design before anything is deployed.



Step 2: the proposed model structure is reviewed, revised or accepted before anything is generated.

AI Context and Rules

Generated models should look like your models. AI Context and Rules is where your organization tells the AI how.



AI Context and Rules in Flowable Design: ten pipeline stages in four groups, each with its own instructions that are appended to the system prompt whenever that stage runs.

Your conventions in every generation

App and model generation runs as a pipeline of ten stages in four groups: analyzing the prompt and the model structure, checking REST services and AI agents, generating the data dictionary, processes, cases, forms and the app model, and validating expressions. For every stage you write your own instructions in plain text, tagged as Important, Rule or Note: naming conventions, preferred patterns, elements to avoid, house rules for forms. They are appended to the system prompt whenever that stage runs, so the output follows your conventions instead of generic ones.

Global and per workspace

Rules are defined once for the whole organization and refined per workspace, where a workspace either extends the global rules or overrides them for its own needs. Modelers see which stages carry custom rules and which use Flowable's defaults.

The same rules for coding assistants

The rules do not stop at the Design user interface. Through the MCP guide, described below, a coding assistant working over the Design MCP server reads the same organization-wide rules, so models built from Claude Code or Cursor follow the same conventions as models built in the wizard.

The MCP guide for coding assistants

Design MCP server: build models from your coding assistant

Flowable Design exposes its modeling capabilities through the Model Context Protocol. AI coding assistants such as Claude Code, Cursor, VS Code with GitHub Copilot or Codex connect to it and browse workspaces and apps, create and edit processes, cases, decisions, forms and every other model type, validate them, write and run tests and publish apps. Edits are structural and laid out automatically, and an assistant does only what the connected user may do in Design, with the same validation and versioning.

The same improvement loop, from your coding assistant

A coding assistant does not only read and write models. It generates tests for them, runs them through the Design MCP server, reads the failures, fixes the model or the test and runs again, the same loop the app generation wizard follows, driven from the assistant. Custom code is never required, but where a use case calls for it, the same assistant can also write custom Java extensions or React components together with the models that use them, in one session.

Flowable's modeling guidance, versioned with the product

The guide tool of the Design MCP server hands the assistant Flowable's own modeling guidance on demand: a discovery topic that explains how apps, workspaces and model types fit together, and one topic per model type with its schema and the rules for building it well. The assistant decides which topics to read and when, and the guidance is versioned with the product, so it always matches the release you run.

Your organization's rules included

Every guide topic ends with the organization's own AI Context and Rules for that stage: the discovery topic carries the analysis rules, each model type topic carries the generation rules for that type. A developer working in a coding assistant follows the same conventions as an analyst working in the wizard, without anyone re-explaining them.

Set up in three steps

The AI assistant setup in the Design user menu walks through assistant, configuration and verification: pick Claude Code, Claude Desktop, Cursor, VS Code with GitHub Copilot, Codex CLI or a generic MCP client, receive a scoped access token and the exact configuration snippet, and confirm the connection.

WHAT IS

MCP (Model Context Protocol)

An open standard for connecting AI assistants and agents to tools and data. An MCP server publishes what it can do; a connected assistant calls those tools on behalf of its user. Flowable provides MCP servers for Design and for Work, and Flowable agents use tools from other MCP servers.

Cases, processes and decisions

WHAT IS

BPMN, CMMN and DMN

Three open standards for describing work, maintained by the Object Management Group. BPMN (Business Process Model and Notation) describes step-by-step processes. CMMN (Case Management Model and Notation) describes cases whose steps depend on the situation. DMN (Decision Model and Notation) describes business rules as decision tables. Models in these notations are exchanged as XML and read by tools and people across the industry.

Cases and processes

A case model (CMMN) describes what may happen in a piece of work: stages, tasks, milestones and the conditions under which they become active, and the case follows the situation as it develops. A process model (BPMN) describes the structured, sequential parts, such as a payment run or an approval chain, step by step. Both are used together in one application.

Child instances

A case can start child cases and processes and follow them: one onboarding case with a sub-case per product, a claim with a payment process per settlement. Child cases and processes report their results back, problems are handled directly or handed up to the parent, and the whole hierarchy is shown as one tree in Flowable Work and Flowable Hub. When a new version of the parent model is released, Flowable Hub migrates the parent with all its child cases and processes in one operation.

Business rules as decision tables

Eligibility, pricing, routing and risk rules live in DMN decision tables that business experts read and maintain, or describe in plain words and have AI generate. Several tables combine into a decision service, and every evaluation is recorded so a case worker can later see which rules fired and why.

Exceptions are part of the business logic

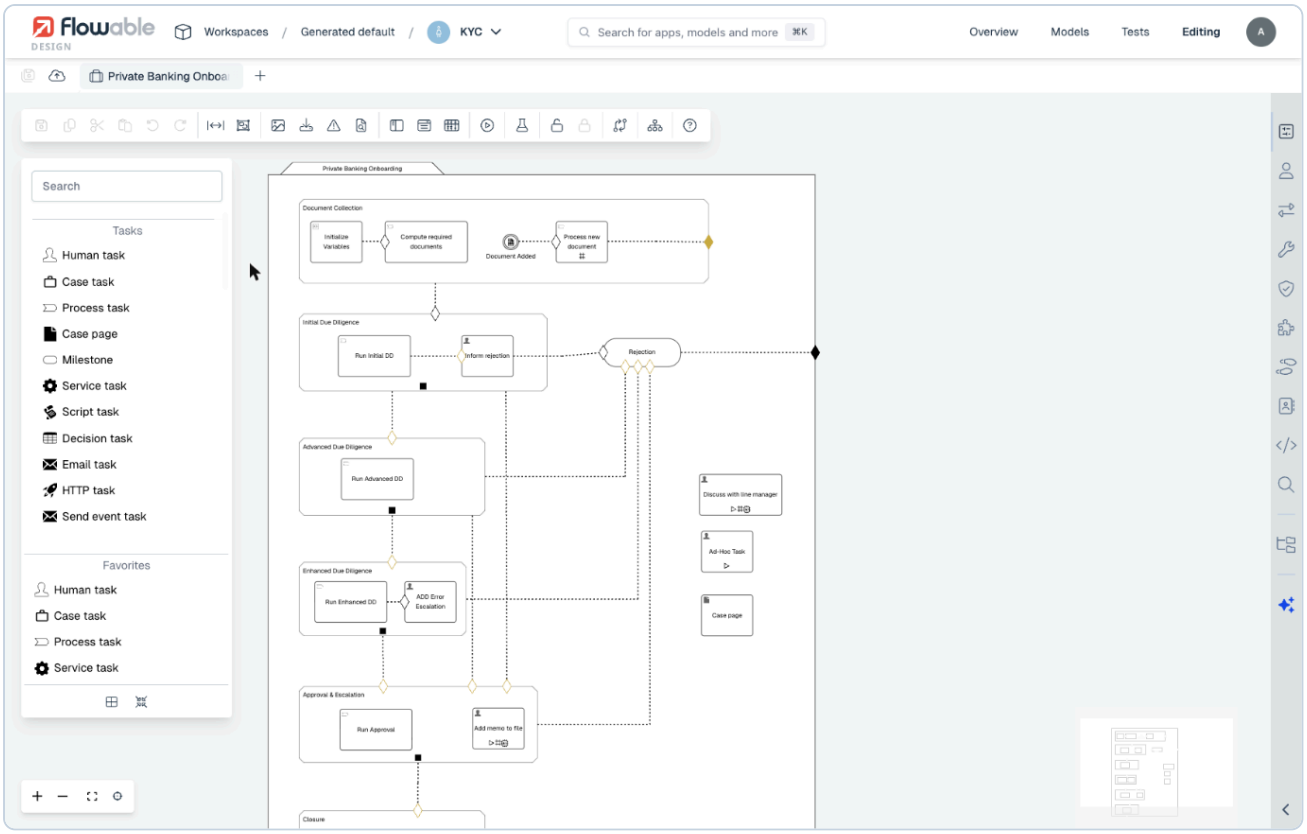
Real work is full of exceptions, and in Flowable they are part of the modeled business logic. A claim needs three of five reviewers to agree; a payment that went out must be reversed when a later check fails; a document that does not arrive in ten days triggers a reminder and then an escalation; a problem in a subprocess is handed up to the case that started it, where a person deals with it. Each of these is a standard element on the diagram, visible to everyone who reads it.

Low-code with a way out

Conditions are built with a visual expression builder that knows every variable in scope. Where a rule needs code, script tasks run JavaScript or Groovy against a typed API for data, collections, dates and business errors, and developers extend the platform with Java where needed. A variable view in the case and process editors shows which variables each step reads and writes. Models are validated on save and on demand, and the app overview shows what still needs attention before deployment.

Applications in every language

Everything shown to end users can be translated inside Flowable Design, with export to and import from Microsoft Excel. The product interface itself ships in English, German, French, Italian, Spanish and other languages and can be translated into any further language. Users work in their own language and time zone.



A client onboarding case in Flowable Design: stages for document collection, due diligence and approval, with the modeling palette on the left. The palette lists every step available for the business logic and is customizable per organization.

Build once, reuse everywhere

A library of proven building blocks

Any model can be referenced from any other. A standard subprocess, a shared form, a common decision table or a whole case becomes a building block with a declared, typed input and output contract. Other teams reuse it without having to read its internals. For each reference you choose whether running work stays on the version it started with or follows the latest version. Every model shows where it is used.

Standards for the whole organization

Model templates give new models a house style to start from. Plugins package services, processes and cases as installable components that appear in the modeling palette. The palette itself can be customized to each organization's requirements: existing elements can be removed and new ones added, so business modelers see only the elements they need, and company-specific tasks and form components sit next to the standard ones.

Governance of change

Teams work together on one application

Models are organized in workspaces, shared areas for a team's apps, with read and write permissions. Model locking prevents simultaneous edits by several people. Modelers attach comments to individual elements to record design decisions, and a search within an app finds any model or property.

Every change is versioned and comparable

Each save creates a new version, and versions of a model can be compared visually. At app level, changes across all models are aggregated as revisions, and two revisions can be compared with a diff that lists added, removed and changed models and shows the changes inside each one. A release review then covers what actually changed.

Source control and pipelines like software

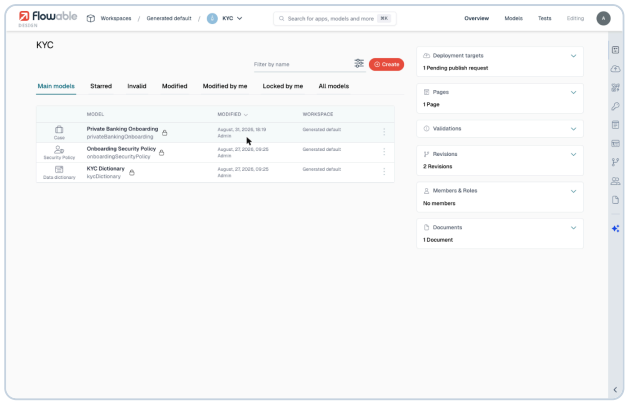
A Flowable app can be synchronized with a Git repository on GitHub, GitLab or Bitbucket. From Flowable Design you branch, pull, commit, push, resolve conflicts visually and open pull requests, and commits can trigger your CI/CD pipeline. Test definitions are stored and versioned with the models in the same repository.

Controlled path to production

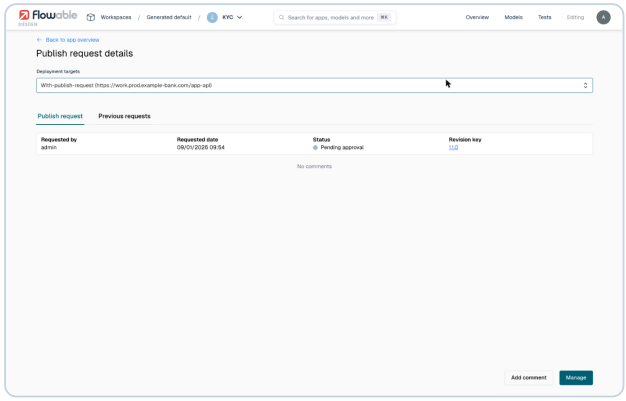
Each deployment target, such as test or production, can require an approved publish request. Nominated approvers review the changes, discuss them with the requester and approve or reject. The request records which revision went where, by whom and when; how versions coexist at runtime is described in the chapter Operate, test and improve.

Tested before it ships

Token simulation, described in the next section, shows how a model runs before anything is deployed. Tests for cases, processes, services and agents are authored inside Flowable Design and run on demand before a model reaches production. The chapter Operate, test and improve describes the testing and debugging tools in detail.



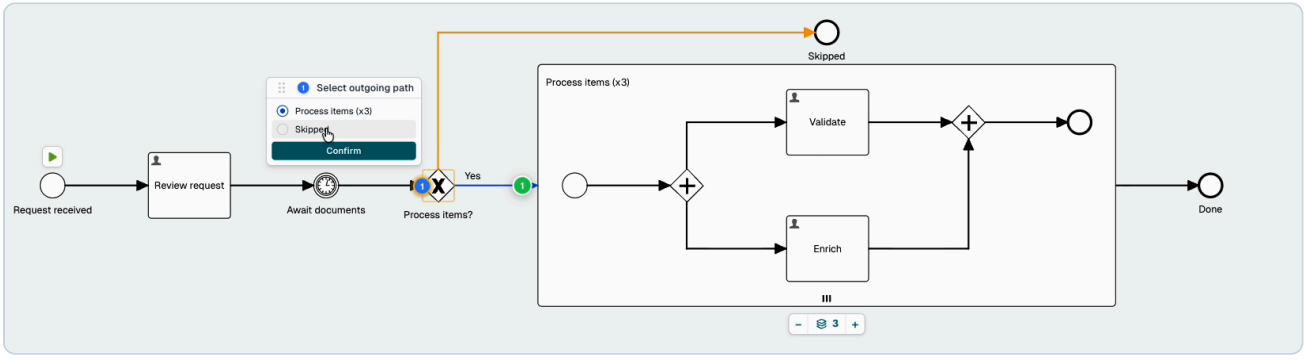
The app overview: models, validation status, revisions, pages and a pending publish request in one place.



A publish request waiting for approval before the app reaches the production environment.

Simulate the model before publishing

Token simulation animates a process or case model inside the editor. It shows where work can flow and what the model looks like while it runs, and it complements model validation and model testing.



Token simulation of a process in Flowable Design: the token waits at the gateway and the modeler picks the outgoing path, which is highlighted on the diagram.

Watch the work move

Click a start event and a token travels along the sequence flows. Tokens rest on elements as colored coins with a count, several tokens run at the same time in their own colors, and a speed control sets the pace from a slow walk to a quick overview. In a case model, Play starts the case and every step, stage and milestone shows a ring for its lifecycle state: available, active, completed, terminated or disabled.

The simulation asks you

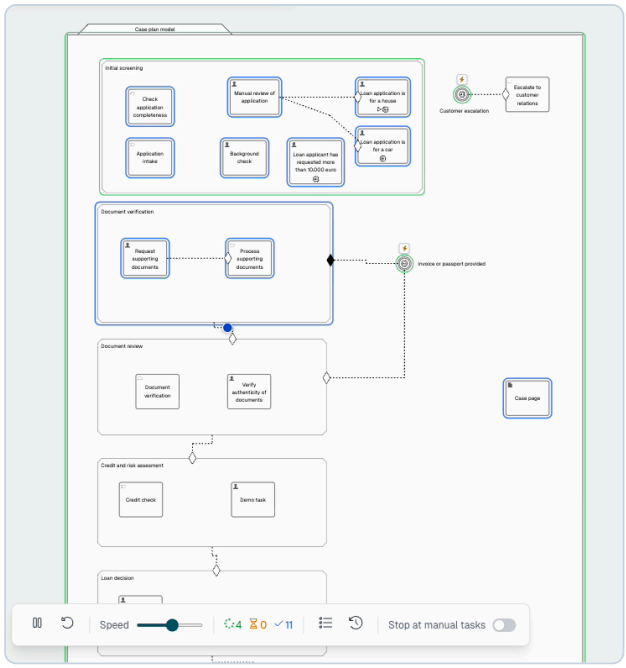
Where the diagram alone cannot decide, a popup asks the modeler. At an exclusive or inclusive gateway you pick the outgoing paths and see them highlighted before confirming. You trigger events and boundary events by hand, choose how many times a repeating activity or step runs, activate or disable optional steps, and state whether the condition for a step to open is met. The simulation has no variables and evaluates no expressions. Your choice is the decision.

Step by step, with breakpoints

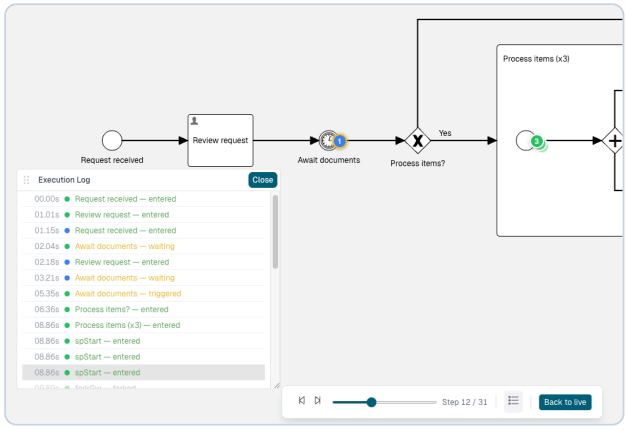
A switch stops the run at user tasks and other waiting steps, and a Complete task control moves the token on. Breakpoints on any element hold a token there and survive a reset, which makes it quick to run the same scenario again. Terminate end events, interrupting boundary events and sub-process scopes behave as in the engine, and a join that can never complete is reported as stuck. Once a case has completed, a reactivate event listener offers to reactivate it, and a dialog lets you choose which reactivation rule counts as met for each element.

Log, replay and share

An execution log lists every step with its time and the token that caused it. Step through the run reconstructs the diagram at any recorded step with previous, next and a slider, and clicking a log row jumps to that step. The same visualization replays real instances in Flowable Work, as described in the chapter Operate, test and improve.

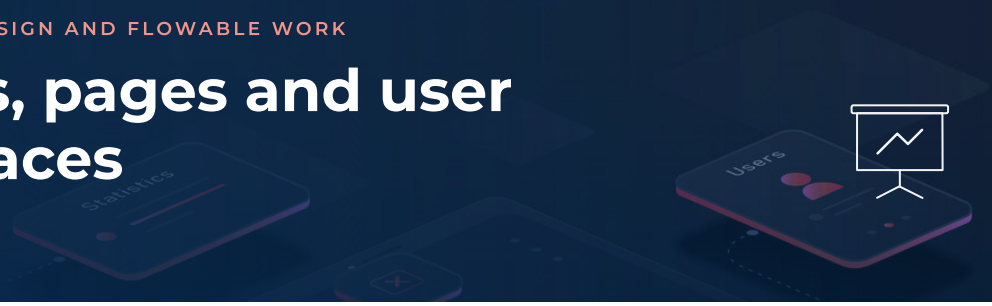


A case in simulation: every plan item carries a ring for its lifecycle state, and the control panel counts active, available and completed items.



Stepping through a run: the execution log lists every step, and the slider reconstructs the diagram at any recorded step.

02 Forms, pages and user interfaces



Most cases come with screens for the people working on them, and those screens change as often as the rules do. In Flowable, forms and pages are models: designed visually, versioned with the app and rendered the same way in the browser and in the mobile app. Organizations that already have a portal or app keep it and run Flowable behind it.

Forms without code

Forms that fit the work

Forms are composed from a large component set: standard inputs, dates and ranges, rich text, tabs, wizards and accordions, repeating subforms for lists such as beneficiaries or addresses, document and media components with previews, data tables with filtering and master-detail views. Multi-column layouts and reusable subforms keep even large forms clear and consistent across applications.

Forms that react and validate

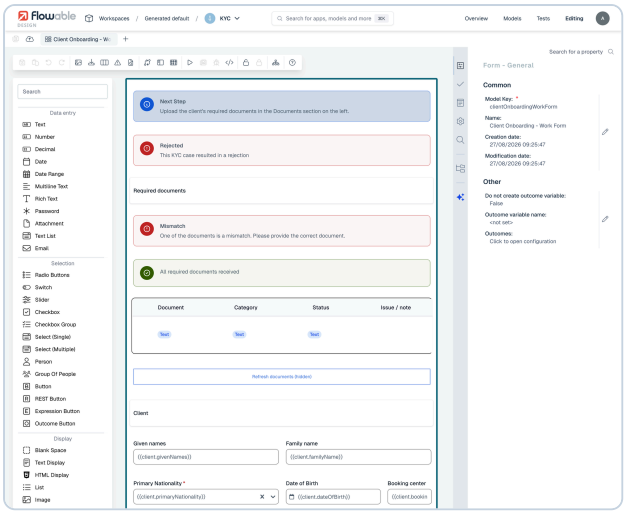
Components respond to each other: visibility, enablement, required state and prefilled values depend on what the user enters. A task is completed with outcome buttons such as Approve, Reject or Send back, and the outcome determines which path the process takes next. Selection lists and tables can be filled from live services or databases, and lists of cases, tasks or processes can be embedded directly in a form. Field constraints are enforced on the server at submission and cannot be bypassed through the API, and a task form that is not finished yet can be saved by the user, or automatically where auto-save is enabled, so work is picked up later without completing the task.

Your own components

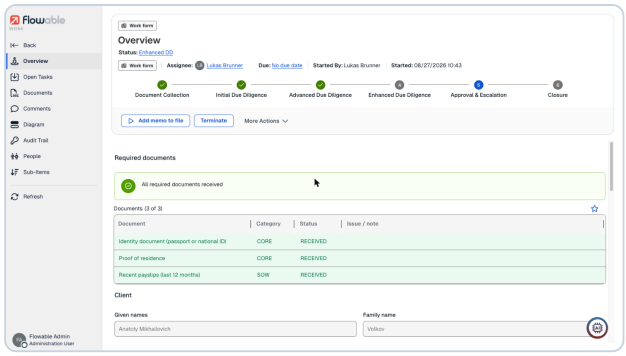
Where the standard set is not enough, developers write components in React and TypeScript. The Flowable CLI, a publicly available command-line tool, scaffolds a component library, runs it with live reload against a running Flowable environment and generates TypeScript types from the properties configured for the component in Design, where the component also gets its palette entry and property panel. A component can read and write form values, call REST services and compose existing Flowable components such as panels and tables. Built libraries are uploaded to Flowable Design with a semantic version, can be tested as drafts before they become visible to all modelers, and can be moved between environments with the same tool. Every form that uses a component decides whether it sticks to a pinned version or always uses the latest one.

✦ AI inside the form

A form can carry an AI agent button that calls a utility, knowledge or external agent with the values on the form as input and writes the result back into form fields: draft the reply to a complaint from the case data, summarize the documents of the case, check an application for plausibility or translate a text. The case worker gets the agent's help inside the form they are already working in, with the case data as context, and edits the proposal before submitting. When a user uploads a document, a document agent can classify it and propose values for the form fields, and the user accepts or rejects them field by field. Nothing is written until the user confirms, and every call is audited like any other agent run.



The form editor: the component palette on the left, a client onboarding work form on the canvas, and the property panel on the right. The alerts at the top appear only in certain situations: one when documents are still missing, one when the submitted data does not match, one when everything has been received. Each component carries its own condition on the case data and state, so the form a case worker sees changes with the case.



The same onboarding form as the case worker sees it in Flowable Work. All required documents have arrived, so only the confirmation alert is shown and the alerts for missing documents and mismatches stay hidden. The stage indicator at the top, the menu on the left and the actions on the right come from the case model.

Pages per role

The right view for each role

A case model can define named pages that appear with the case lifecycle and assembles navigation per user group from built-in views, forms and links, so a claims handler and a team lead see different views of the same case. Custom views for cases, processes and apps are assembled by drag and drop from component libraries.

Your brand, everywhere

Theme models set logos, colors and fonts without code and apply in the browser and in the mobile app; for a complete white-label, a theme is assigned per tenant or user and styled with your own stylesheet. Dashboard components, chart widgets built once, appear in the built-in dashboards, and document and content templates give generated documents and messages the same house style.

Bring your own user interface

Headless behind your portal

The whole platform is driven through its REST APIs. An existing customer portal, employee app or partner system starts work, shows tasks and submits forms without exposing Flowable Work to its users.

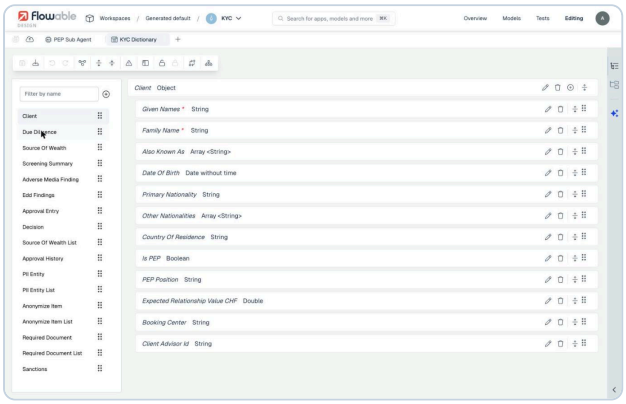
Embed the pieces you need

The case view and app view ship as drop-in React components. The form rendering SDK displays Flowable form definitions inside external web applications, a React Native form engine renders them in native mobile apps, and an Angular integration is available. Flowable Work itself can be extended with custom tabs, headers, apps and document preview handlers.

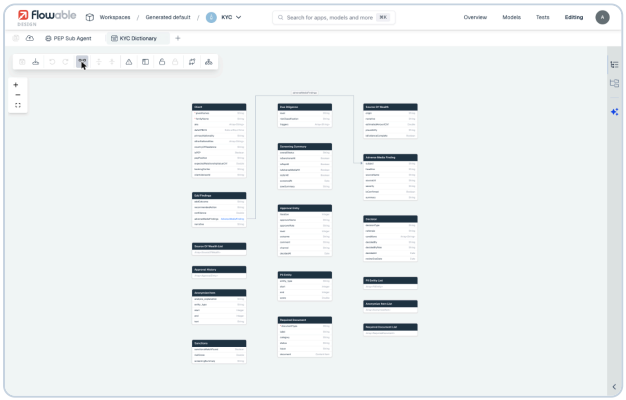
03 Data and integration

A case is only as good as the data it works on, and that data usually lives in the core system, the CRM, the document archive or a message queue. Flowable treats data structure as a model in its own right and connects to the rest of your systems through services, events and code. Case workers see current data, and other systems are informed of decisions as they are made.

Governed business data



A data dictionary: the client entity with typed, required properties.



The entity-relationship view of the same dictionary: entities, fields and relationships at a glance.

One definition of your business data

A data dictionary defines the structure, types and constraints of your business objects in one place, with an entity-relationship view. Cases and processes declare which data they read and write, and the runtime validates every value against the dictionary, with your own validation messages. Modelers, forms, services and agents all work against the same definitions.

Data lives where it fits

Most case data simply lives with the case, as variables that forms, decisions, services and agents read and write, and it stays in the history after the case closes. Where a record must outlive a single case or be shared and searched across cases, such as a client or a contract, it becomes a Data Object. Where the master copy belongs in another system, the case works on it through a service. You choose per piece of data, and the dictionary types it the same way in all three places.

Business records as first-class entities

Where the dictionary describes the shape of data, Data Objects hold the records themselves, stored in the Flowable database or mapped from REST services, as entities that are viewed, searched and edited from cases, processes and forms. For records in the Flowable database, the tables and initial schema, the create, read, update, delete and search operations and the administration pages are all generated from the structure. Custom operations are defined without code, and per Data Object you choose live synchronization in both directions or controlled refresh points. Data Object forms carry buttons that run actions, services or agents on the record.

Reference data, identifiers and search fields

Master data such as country lists or product catalogs is maintained centrally and administered in Flowable Hub. Cases and tasks can carry formatted business identifiers such as INV-00001, generated by a sequence model. Values buried in case data, such as a policy number or an amount, can be made searchable, and users then filter and sort on them in every list. Tenant and app variables hold environment settings such as endpoints and feature flags, and one app runs unchanged from development to production.

Operating parameters in business hands

Values that steer daily operations can be kept out of the models and changed without a release: thresholds, assignment rules and routing values can live in Data Objects that authorized business users edit in Flowable Work; country lists, product catalogs and other master data are maintained in Flowable Hub; endpoints, limits and feature flags are tenant variables. Where the logic itself changes, business experts edit the decision table and publish it through the normal approval, still without a development project.

✦ Agents work on the same data

Case variables and Data Objects, typed by the dictionary, act as input and output for AI agents. Agent results land as validated business data, and per-field descriptions in the dictionary tell the AI model what each value means.

Connect any service

One registry for all external systems

External systems are defined once in the service registry and called from processes, cases, forms and actions, the reusable shortcuts users trigger on a case, with responses mapped onto typed data. An OpenAPI specification is imported into a service definition together with a data dictionary of its types; database queries, scripts and Spring beans are registered the same way. External MCP servers are registered in the service registry too, and their tools are called like any REST operation. From a Salesforce object you can generate a complete integration, including data objects and screens, and SharePoint files can link directly to cases and tasks. The Salesforce and SharePoint connectors are licensed separately.

Enterprise-grade connectivity

Services authenticate with OAuth2 including On-Behalf-Of token exchange, API keys, basic authentication or custom headers; outbound calls support mutual TLS and internal certificate authorities. Credentials never sit in a model: API keys, tokens and passwords are stored as secrets in Flowable Hub, referenced by name from services, agent connections and expressions, shown once at creation and never exposed again. Awkward payloads are reshaped once, centrally, in request and response handlers.

Build and test without the real system

Every REST service can carry mock responses with match conditions, status codes and simulated errors, switched on per environment or at runtime in Flowable Inspect. Applications are built and tested while the real endpoint is unavailable or still being built.

✦ Any LLM, centrally managed

Agents run on standard AI models, frontier or open-weight: OpenAI, Anthropic, Azure OpenAI, Amazon Bedrock, or any endpoint compatible with the OpenAI or Anthropic APIs, including self-hosted AI models. Flowable ships no models of its own and does not train on your data. Providers, AI models, prices and credentials are administered as agent connections in Flowable Hub. Agents reference a connection by name, and an AI model swap or credential rotation needs no redeployment.

Event-driven at scale

Business events instead of message plumbing

Channel and event models turn messages from Kafka, RabbitMQ, JMS, AWS SQS and SNS, email, REST webhooks, files, FTP and SFTP, SQL and Azure Service Bus into named business events with typed payloads. Cases and processes send events, wait for them, start on them and react to them in running work, while the transport stays out of the business models and can be exchanged.

Millions of events a day

Events are matched to the right running instance on several fields with an algorithm built for high volume. In a multi-tenant setup the tenant is detected from the message, and one topic serves all tenants. Combined send-and-receive activities make request-response interaction with microservices safe.

Extend with code

Your developers, your languages

Task logic runs outside the platform where you prefer: official external worker libraries for Java and Spring Boot, Python, JavaScript and TypeScript, .NET and Go poll for work over REST and run anywhere, including behind firewalls. RPA Framework robots and Power Automate Desktop flows bring screen-based legacy systems into the same end-to-end flow. Java extension points add custom tasks, expression functions, REST controllers and user interface actions, packaged with the platform's Spring Boot starters.

Everything is an API

Everything the user interfaces do is also available through the REST API. Any system can start work, complete tasks, upload documents and query state. The public documentation describes every REST API with its OpenAPI specification and a Swagger UI to try it out: processes, cases, decisions, forms, content, data objects, identity, events, external workers, agents, Design and more.

04 Agents in cases and processes

An AI agent on its own answers questions. An agent inside a Flowable case does work: it reads the documents that arrived, checks them against the rules, summarizes for the approver and proposes or takes the next step, within the boundaries the case sets. Agents are modeled, versioned, tested, governed and audited like every other part of the application, and they work inside the boundaries the case sets. A dedicated agent engine runs beside the case, process and decision engines, on the LLM provider you choose.

Who does what in a case

Start with a case

A loan, a claim or a new client opens a case. Its stages, data and rules set what may happen and when.

People work the case

Case workers review and approve, and every step they take is kept in the audit trail.

Systems do the routine work

Processes run the fixed steps. They look up data, apply decision tables and generate documents.

Agents read, draft and judge

Agents read documents, check them against the rules and draft the assessment, seeing only what the case lets them see.

The orchestrator proposes, people decide

It tracks the case and proposes or takes the next step. Steps with high stakes wait for a person, and people can step in at any time.

Eight agent types

Each agent is a versioned model in Flowable Design with typed operations, its own tools, audit settings and tests, and it runs on the LLM you assign to it. The type decides what the agent is for.

WHAT IS

An agent in Flowable

A role, a job and its context. The eight agent types are roles: each defines what kind of work an agent does and what is already built for it. An agent is a role with a job description, written in Flowable Design. Flowable supplies the context for every call to the AI model: the case data, the documents, the conversation and the tools the agent may use. Your team decides where that agent works: in which cases, processes and forms it takes part. You build an agent once and use that agent in as many cases, processes and forms as you like. When you change the agent, every place that uses the agent gets the change.

Configured in minutes, reused everywhere

A business analyst picks a role, describes the job, receives a draft of the instructions, inputs and outputs, corrects it and tests it against real documents and cases. You can have as many agents of one role as you have jobs: a classifier for onboarding documents and another for claims are two agents of the same role. Once published, the same agent can be placed in any case, process or form where the modeler wants it, and offered to other systems through a REST endpoint. A change is made once and reaches every place the agent is used.

WHAT IS

LLM (large language model)

The AI model that reads and writes text for an agent, such as GPT, Claude, Gemini or an open-weight model you host yourself. Flowable agents run on the LLM you assign; the AI model reasons and drafts, while Flowable decides what it may see and do.

Utility agent

Runs a focused prompt with defined, typed input and output: classify a request, draft a reply, extract a value, assess plausibility. The workhorse for single steps in a process or case, small and reusable across applications.

Knowledge agent

Answers questions and produces content grounded in your knowledge bases. Relevant passages are retrieved at runtime and the answer records which knowledge items it used.

Document agent

Understands documents: recognizes the document type from the types you defined, extracts structured fields, summarizes and translates. The engine behind intelligent document processing in forms, inbound email and content workflows.

Orchestrator agent

Reads the whole case, recognizes intents and decides which of the steps currently allowed by the case should be started or suggested. It coordinates sub-agents, services and human tasks, and case workers chat with it about their case.

External agent

Brings an agent hosted on a third-party platform such as Salesforce Agentforce, Azure AI Foundry or Amazon Bedrock into the case, with multi-turn sessions and asynchronous execution, under Flowable's audit.

A2A agent

Talks to any agent that implements the Agent2Agent protocol, on any vendor's platform. When the remote agent needs more input, the step turns into a human task.

WHAT IS

A2A (Agent2Agent protocol)

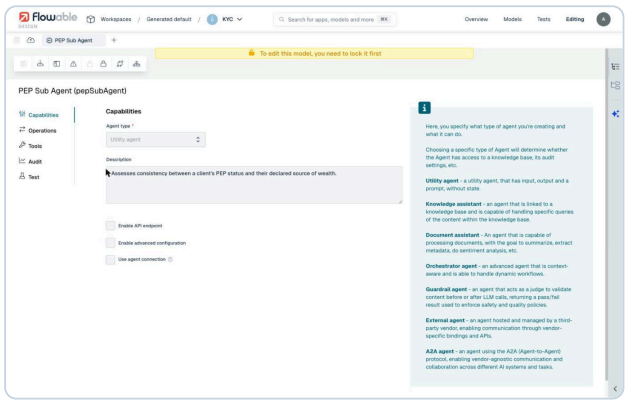
An open protocol for agents on different platforms to hand each other tasks and results. A Flowable agent can call an A2A agent on another vendor's platform, and Flowable agents can be called the same way.

Guardrail agent

Acts as a content judge before or after an LLM call and returns pass or fail against a policy written in plain language, for safety, tone, scope and compliance.

Evaluator agent

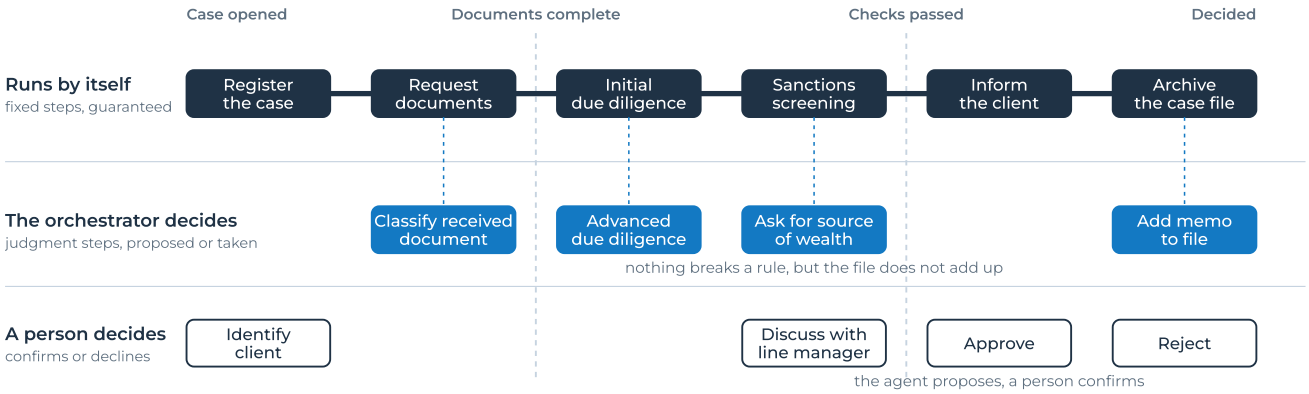
Acts as a quality judge after another agent has run and records a verdict, a score and a reason. Agent quality is measured continuously in production.



An agent model in Flowable Design: the agent type and its capabilities, with operations, tools, audit and tests in the side navigation.

Deep dive: the orchestrator and the case as its harness

Of the eight agent types, the orchestrator is the one that decides what happens next in a case. This section explains how it works. A case mixes two kinds of steps. Fixed steps always happen the same way once their conditions are met: a process runs, a decision table is applied, a letter goes out. Judgment steps need someone to decide whether and when they should happen. Both are drawn in the same case model, which holds every action that can ever be taken in the life of a piece of work and the conditions for each. The fixed steps run by themselves. For the judgment steps, the orchestrator agent brings the judgment.



One client onboarding case, read left to right. The top lane is the fixed backbone: these steps run in this order whenever their conditions are met. The steps below hang off the backbone at the moment they become possible, and the case, through its conditions, decides when that is. The orchestrator decides only in the middle lane. Approve and reject stay with a person.

Case opened

Documents are missing, the client is not identified yet.

- Register the case
- Request documents
- Identify client
- Classify received document
- Initial due diligence
- Advanced due diligence
- Approve
- Reject

Due diligence

Documents are complete, a risk indicator was raised.

- Initial due diligence
- Sanctions screening
- Advanced due diligence
- Ask for source of wealth
- Discuss with line manager
- Reject
- Approve

Completed: Register the case, Request documents, Identify client

Decision

All checks have passed.

- Approve
- Reject
- Add memo to file
- Inform the client
- Archive the case file
- Reopen after closure

Completed: Register the case, Request documents, Identify client, Initial due diligence, Sanctions screening

Open in every stage — Add a note · Upload a document · Ask the orchestrator

Fixed step: runs by itself the moment it opens

Judgment step: the orchestrator or a person decides whether and when

Not open: its conditions are not met

The same client onboarding case at three moments. Fixed steps run deterministically when their conditions are met. The open judgment steps are the whole space of actions available to the orchestrator agent and to the case workers at that moment.

Fixed where it must be, judgment where it helps

A large part of a case is deterministic. When the documents are complete, the initial due diligence process always runs. When the decision is made, the client is always informed. These steps happen by themselves, in the order the model says, with no LLM call and no token spent, and that is guaranteed. Between them sit the steps where judgment is needed: whether a question about the source of wealth is warranted, whether the line manager should be involved. Only these are handed to the orchestrator agent. Agent tasks can also sit inside the fixed part, one or several, in sequence or in parallel, and the case orchestrates people, processes and agents together.

Rule or judgment?

If you can tell a new colleague exactly when a step should happen, write it as a condition in the case model. It then runs every time, for free. If your answer begins with "it depends", because the trigger lies in a pattern across the file, in the wording of an email or in a situation nobody foresaw, hand the decision to the orchestrator. It reads the whole case the way an experienced colleague would, proposes the step and records why. Most of a case stays rules. Judgment is bought only where rules cannot be written.

One model holds every possible action

The case model describes the whole life of a piece of work: its stages, every step that may be taken in each of them, and what must be in place before a step opens, such as a document that has arrived, a check that has passed or a decision that has been made. Nothing an agent or a person does in the case exists outside this model. Business and IT draw it together, and everyone can read it.

Where the case stands decides what is open now

At any moment the case engine knows which steps are open. At the start of an onboarding these are collecting documents and identifying the client. Later they are the checks and an escalation. At the end they are approve, reject or add a memo to the file. Steps open and close as data arrives, documents are classified and decisions are made. Fixed steps start the moment they open. Judgment steps wait: case workers see them as the actions on their case, and the orchestrator receives the same list.

The agent chooses the next best step

Whenever something changes, the orchestrator reads the case data, the documents, the conversation and the history and decides which of the open judgment steps should happen next, or that none should. Per step the modeler decides whether the agent may take it on its own or must propose it to a person with the right permissions. The agent does not invent steps and cannot reach an action the case does not offer at that moment.

Every step it takes is a modeled step

Whatever the agent starts is an ordinary case step: a human task, a process, a decision table, a service call or another agent. It runs with the same permissions, service levels, audit trail and history as when a person starts it. Its result changes the case, the case changes the list of open steps, and the agent looks again. That loop is how a case moves forward with agents and people taking turns.

The plan is a diagram you can review

The plan the agent follows is the case model itself. It is reviewed by the business, versioned, simulated and tested in Flowable Design, migrated when it changes and inspected in Flowable Work and Hub. Changing what an agent may do is a change to the case model, with the same review and approval as any other change. Switching to another LLM does not change what the agent is allowed to do.

Agents as participants in the case

Agents where the work needs them

An agent task invokes an agent at any point in a process or case. Agents chain, with one agent's output feeding the next, run in parallel and steer the flow based on their results, and an orchestrator agent composes sub-agents, services and human tasks into multi-agent work. Approval gates, escalations and SLAs surround agent work with the same modeling elements used for people, and any agent can be published as a REST endpoint for other systems, restricted to named user groups.

Autonomy you dial in per step

When a case has an orchestrator, it looks at the steps the case allows whenever something changes and decides which should be started. Per step the modeler chooses automatic activation, where the agent proceeds on its own, or suggestion, where it proposes the step and a person with the right permissions confirms it. Intent listeners describe situations that matter, such as an applicant withdrawing, and the orchestrator triggers them when a message, a document or a data change implies that intent.

Ask the case

When chat is enabled on the orchestrator of a case, case workers chat with it about their case, with the conversation memory the agent keeps configurable per agent. A regular question is private and changes nothing; switching to acting mode lets the agent propose or perform steps visible to all participants. The agent sees and does only what the chatting user is allowed to.

Knowledge, tools and MCP

Grounded in your knowledge

Knowledge bases are Flowable's implementation of retrieval-augmented generation (RAG). They index documents, emails, Office files and text from content folders through a configurable pipeline: text extraction, semantic chunking and embeddings from OpenAI or Azure OpenAI, stored in Elasticsearch, with other stores and embedding providers pluggable in Java. A search-only mode queries a corporate index that is maintained elsewhere. Agents retrieve from them at runtime, and every answer records the exact knowledge items it used, with relevance scores.

WHAT IS

RAG (retrieval-augmented generation)

Before an agent answers, the relevant passages from your documents are retrieved and handed to the AI model together with the question. Answers are grounded in your content, and each answer lists the items it used.

Tools under your control

Agents call services from the service registry, other agents and tools on external MCP servers, which are registered as services: their tools are discovered once at design time and become ordinary operations that processes, cases and agents can use. The agent model lists which tools an agent may use; within that list the agent decides when to call which tool and with what input. Nothing outside the list is reachable, and every call is recorded.

Work MCP server: Flowable as a tool for any assistant

Flowable Work exposes its runtime through the Model Context Protocol. A user connects the AI assistant they already use, and it browses the available processes and cases, starts instances, reads running and completed work and tasks, completes tasks with form data, invokes actions and service-backed form fields, looks up users and reads master data, all without leaving the assistant. Everything an assistant does through the Work MCP server runs under the same permissions and tenant scoping as the Work REST API and returns the same data the user interface shows, so an assistant is limited to what its user may do in Flowable Work. The Design MCP server, described in the chapter on building applications, does the same for modeling.

Open standards for interoperability

Agents on other platforms take part in a case through the Agent2Agent protocol, and MCP works in both directions: Flowable consumes tools from MCP servers and exposes its own runtime and modeling environment as MCP servers. No proprietary protocol is needed between your agents and other platforms.

Trust and control

Guardrails on every call

Rules can check what goes into and comes out of every LLM call: deterministic constraints such as ranges, patterns and block lists, guardrail agents that judge content against a policy written in plain language, checks by registered services, and custom logic. Ready-made connectors plug in specialized checkers for harmful content, personal data detection and prompt injection (OpenAI Moderation, Microsoft Presidio, LLM Guard, Guardrails AI), and a sanitizing guardrail can redact personal data before it reaches the model. A triggered guardrail warns, skips the call or raises a business error that the case or process handles like any other exception: a retry with corrected input, a review by a person, or a stop. The remediation is drawn in the model, where everyone can see it.

Data minimization by design

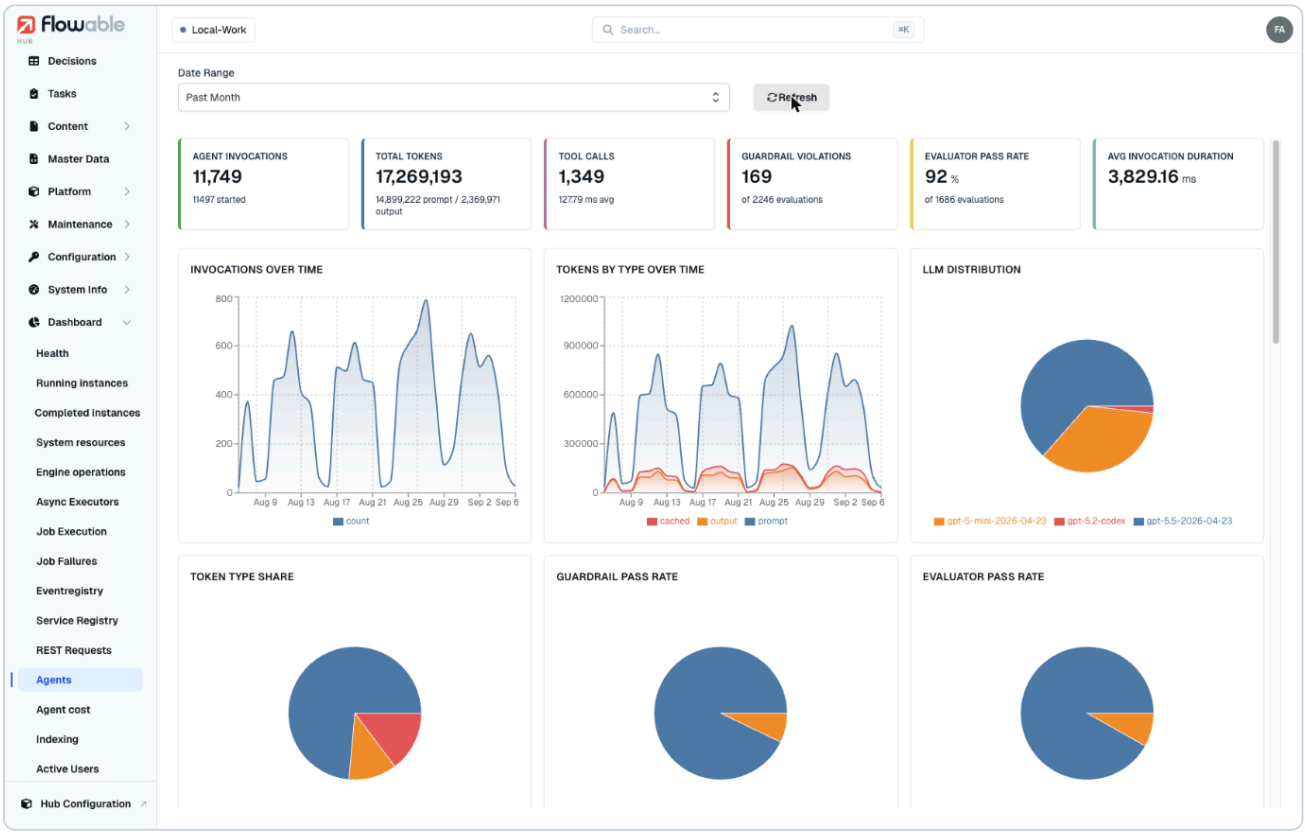
Every agent operation declares which case data it reads and which it writes. An agent receives only the data its step needs, and its reach is visible field by field. In case chat the agent acts inside the chatting user's permission set. Agents inherit the platform's permissions.

Quality that is measured

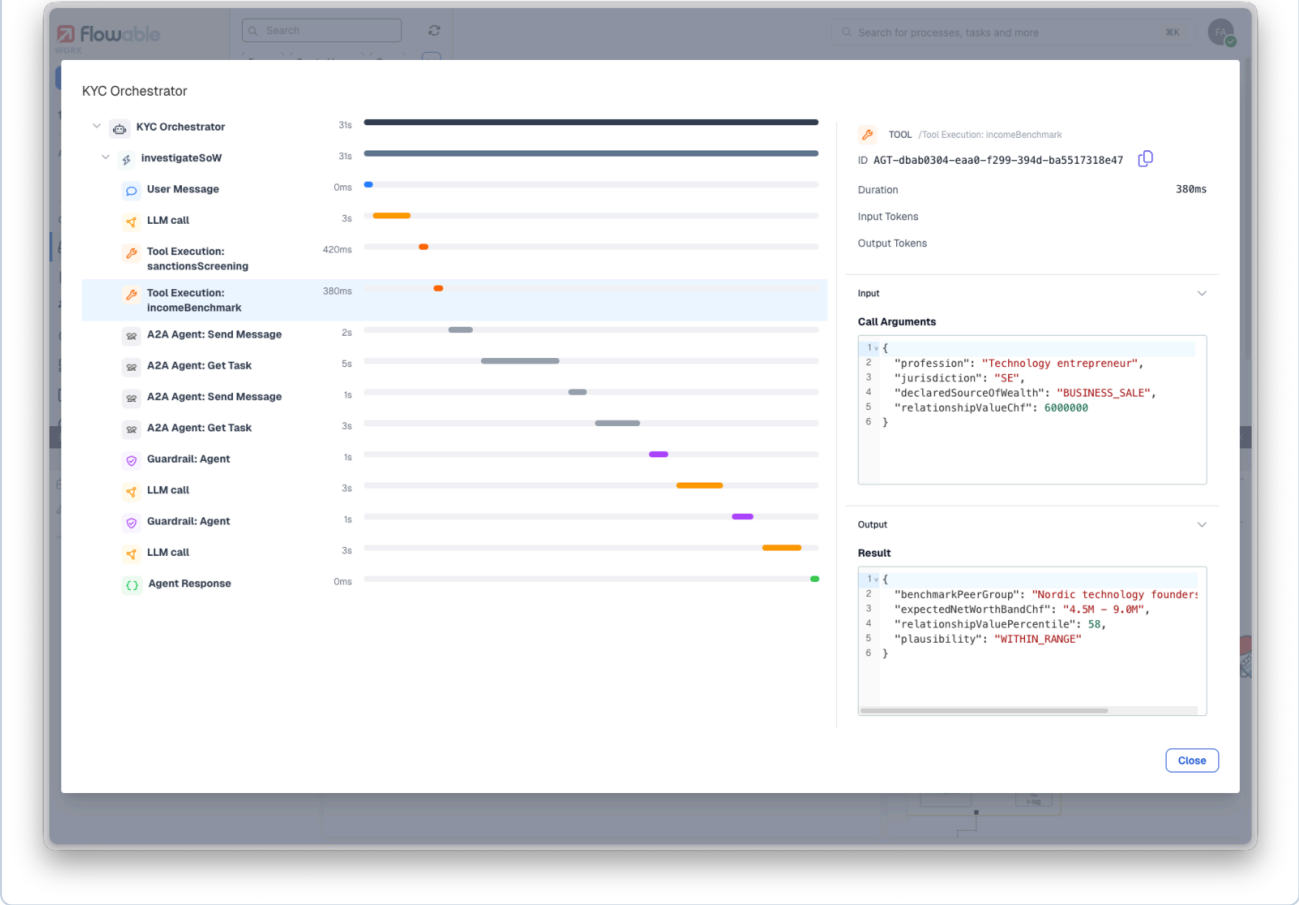
Evaluators can score agent output after every run on dimensions such as correctness, tone, faithfulness to the provided context and format. Evaluators do not block or alter a response. They measure, asynchronously, through an evaluator agent with your criteria, a registered service or custom code, each returning pass or fail, a score and a reason. Repeatable tests in Design assert on responses, guardrail outcomes and evaluator scores, with services mocked for repeatable runs, and document agents are tested against real files. Pass rates per operation and version show when a prompt change or a new AI model makes an agent worse.

Audit and cost per run

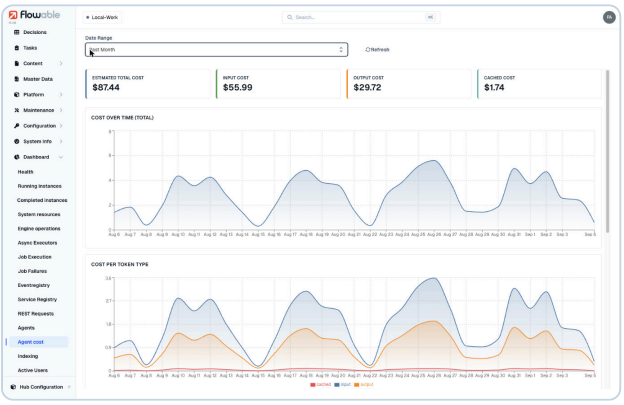
Every interaction is recorded: prompts, responses, tool calls, guardrail results, retrieved knowledge, the vendor and AI model that produced each result, and execution metadata, with audit granularity configurable per agent. The agent timeline, in Flowable Work with Inspect and in Flowable Hub, replays any run, including agents that ran inside child processes and cases. Token usage per invocation, agent and AI model is tracked automatically and, with prices maintained per AI model, turned into cost on dashboards that show volume, quality and spend side by side. Agent data follows the same retention rules as case data.



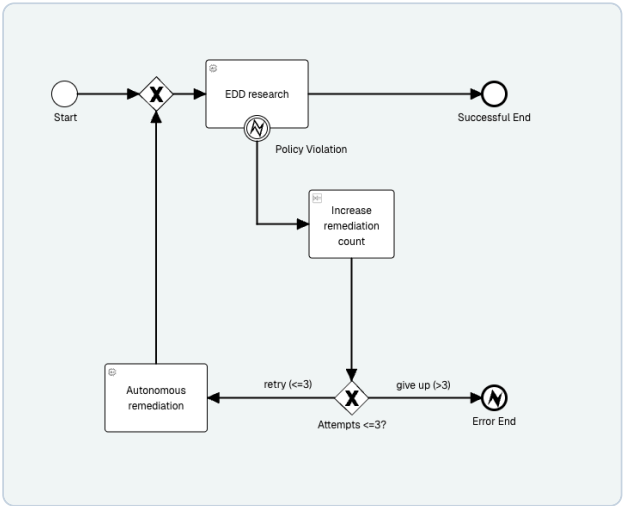
The agent dashboard in Flowable Hub for one month: invocations, tokens, tool calls, guardrail violations and evaluator pass rate, with the weekly rhythm of the business.



The agent timeline: each LLM call, tool execution, A2A call and guardrail check with its duration, and the selected step's input and output.

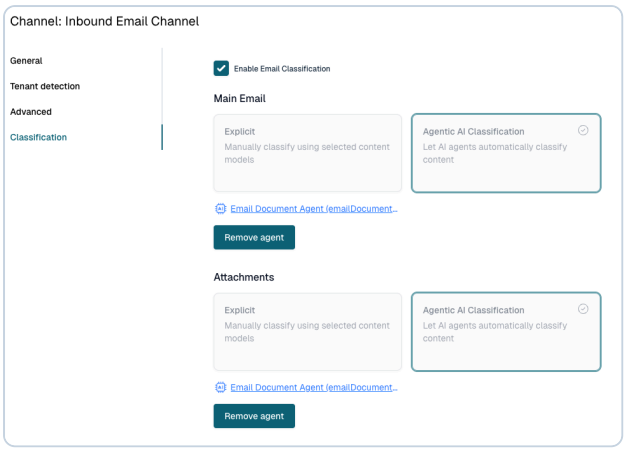


Agent cost in Flowable Hub: token usage turned into money per day and per token type.



Guardrail remediation in a process: a violation raises a business error, and the process model decides whether to retry with corrected input or hand the step to a person.

Documents and email understood on arrival

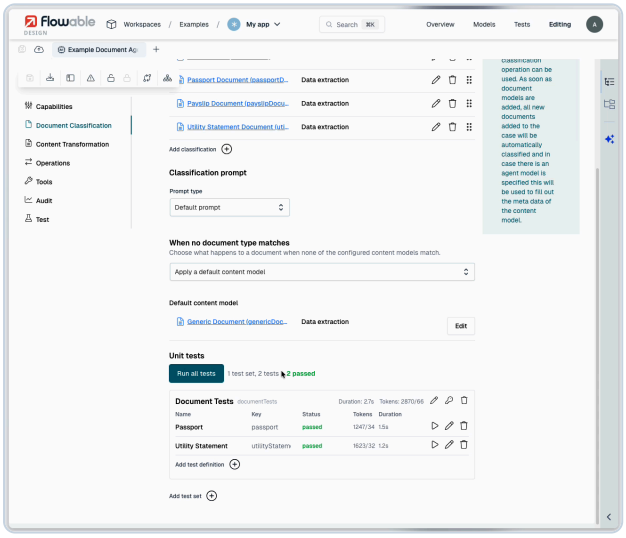


An inbound email channel: the body and each attachment are classified by a document agent, or assigned a fixed document type.

Documents become structured data

Where you add a document agent, it recognizes the type of an incoming document from the types you defined and extracts structured fields, summaries and translations that feed the case as typed data. A fallback type catches documents that match nothing else, and you control how documents reach the model, as text or as images for vision models. Content arriving through an inbound email channel can be classified on arrival, the body and each attachment separately, and intents in the text trigger case logic. When a form upload is connected to a document agent, the extracted values are proposed field by field and committed only when the user confirms.

Testing document agents



Unit tests on a document agent operation in Flowable Design: test documents with their expected classification, run in one click, with pass or fail, tokens and duration per test.

✦ Unit tests for classification and extraction

Each document agent operation has its own unit tests in Flowable Design. Upload your real documents into test sets, state for each which type it should be recognized as and which fields it should extract, and run the set with one click. Every test reports pass or fail with the tokens and time it used.

✦ Evaluate a single run

The Evaluate tab runs the operation once on a chosen document and shows the response together with the full execution trace, so a wrong classification is traced to the prompt, the content model or the document itself.

✦ Change with confidence

Re-run the tests whenever a content model, the prompt or the fallback model changes, and before switching to another LLM. Because the tests use your actual documents at scale, you know the agent still classifies and extracts correctly before the change reaches production.

Three principles behind the design

Model it, do not prompt it

If the same input always produces the same output, it belongs in a process, a decision table or a service, not in a prompt. Tokens are spent where judgment is needed.

Rent the intelligence, own the context

Every organization rents the same AI models on the same terms. Your processes, data, rules, history and permissions are what make the result yours.

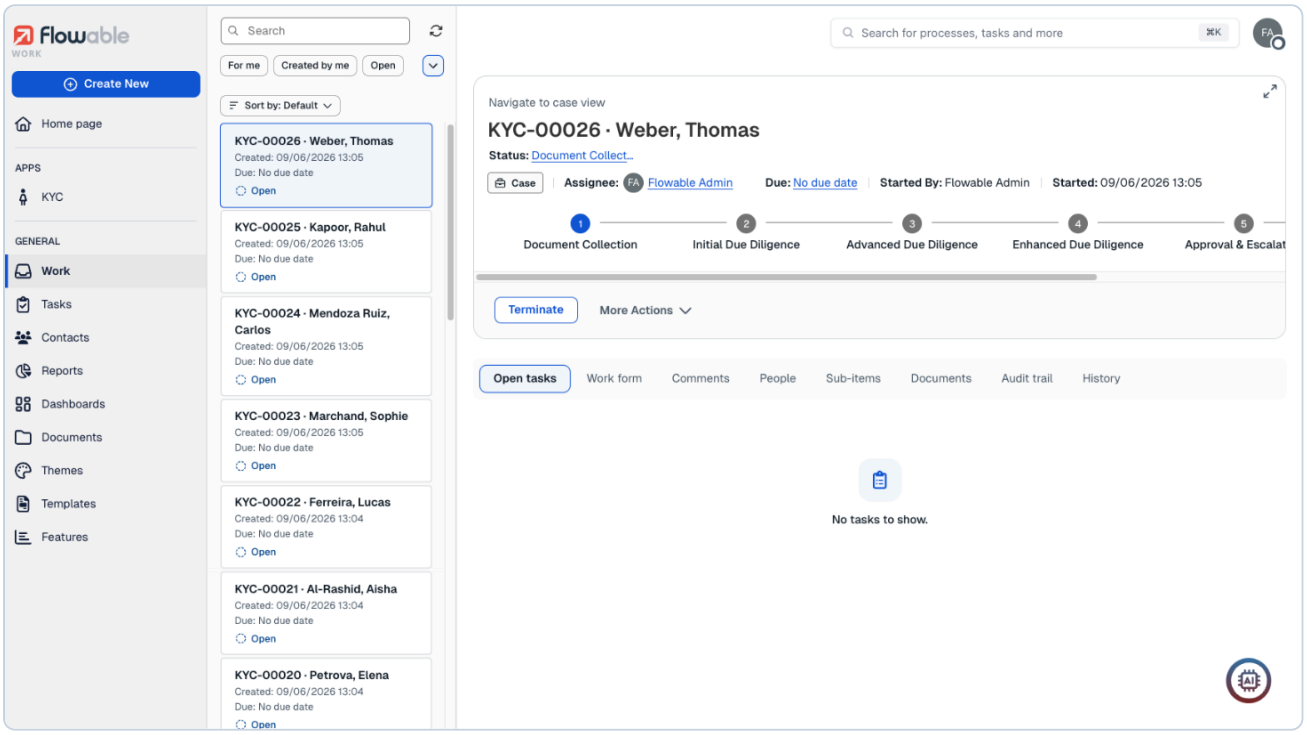
Business knowledge as auditable artifacts

Cases, processes, decisions and agents are human-readable models, versioned and reviewed like software. An automated process is an asset; treat it like one.

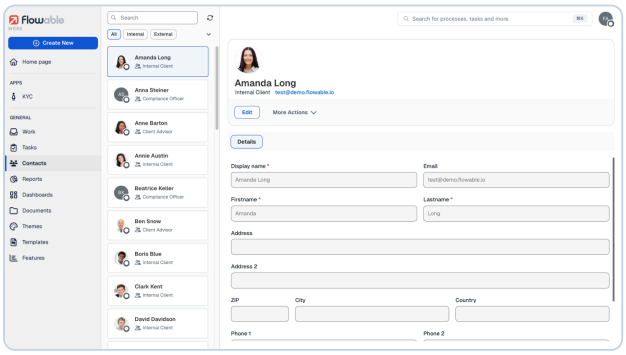
05 Case work in Flowable Work

Flowable Work is where case workers, team leads and specialists spend their day. Every case and task from every application arrives in one place, with the documents, the conversation and the history attached, in the browser and on mobile. What the model requires and how people handle the exceptions are both recorded.

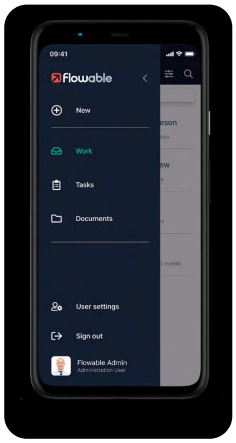
One place for all work



The work inbox: all cases of a client advisor on the left, the selected case with its stage indicator and tabs on the right.



Contacts: internal and external contacts with their own forms and permissions.



The Flowable mobile app with the same work, tasks and documents as the browser application.

Inbox, tasks and assignment

A work inbox shows everything a user is assigned to or has started, and a task inbox collects tasks from all applications with filters for personal work, team queues and completed items. Tasks go to individuals or to candidate groups, where a team member claims the task from the queue; cases carry their own candidates and rules for who may take them over. Priorities and due dates are set per task, and where you use them, business calendars count working time and deadlines drive service-level (SLA) timers and escalations. New cases and processes start from the inbox.

Collaboration around a task

The assignee owns completion, but colleagues are involved to fill in part of a form or advise in comments. Delegation is modeled as part of the process where the business requires it, and a user with the permission hands a task to another candidate directly in Flowable Work. Any task breaks into ad hoc subtasks with their own assignees and due dates, as deep as the work requires. Users also create standalone tasks outside any case, as a to-do list or as work handed to others. Files attach to tasks and cases independently of what the forms require.

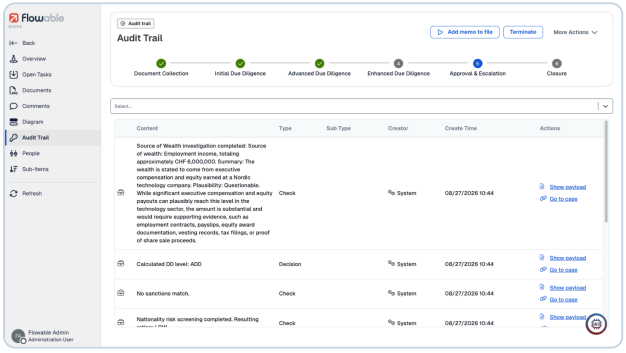
Clients and contacts

Internal and external contacts are records in their own right. User definition models define per type of person, such as client, advisor or compliance officer, which fields, forms and permissions apply, and an onboarding case and a later complaint case work with the same client record.

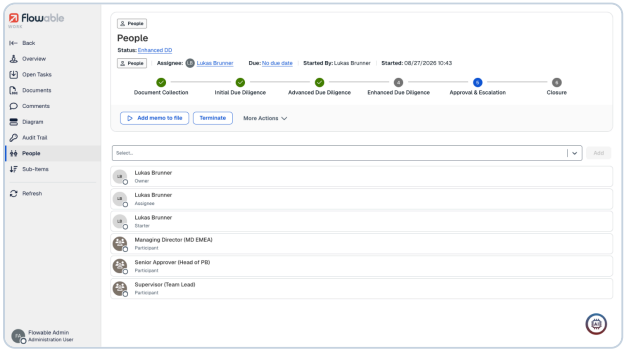
Also on mobile

The Flowable mobile app, available for iOS and Android, brings cases, tasks, documents and forms to the phone with the same forms and theming as the browser application. Camera and microphone can capture photos, video and audio straight into a case, for inspections, incident reports and identity checks in the field. Organizations that build their own app use the React Native forms library to render Flowable forms inside it.

See where every case stands



The audit trail of a case: every check, decision and action with type, creator and time.



People on a case: owner, assignee and participant groups, the basis for permissions.

Status, diagram and history

A stage indicator and a business status in your own vocabulary can show at a glance where a case stands. The live diagram highlights completed and active steps, and a history view summarizes what happened so far.

The audit trail

The audit view records every action with timestamps and user details, including which decision rules fired and why. Audit events can be published to external audit systems as they occur.

Conversations in context

Case workers collaborate in real-time chat directly on the case, in context and fully audited, and comments cover asynchronous exchanges between everyone involved.

Handle what the model did not foresee

Adapt running work

A completed case is reopened along rules defined in the model: which steps become active again, which data is reset first, and what the user may choose to redo. Rework in a regulated setting follows the same rules as the first pass. Work that is no longer needed is canceled at any point.

Actions tailored to your operation

Custom actions such as "escalate to legal" or "request documents" are defined in Design without custom code, attached to cases, tasks, documents or contacts, and offered in the right context in the user interface or invoked through the API. An action starts a case or process with its start form, returns instructions such as navigating to a page, and is tested from the editor.

Find anything

Cases, processes and tasks are found by name, description, business attributes and other fields, backed by Elasticsearch. New search queries and lists are defined in a low-code query editor and deployed without a restart.

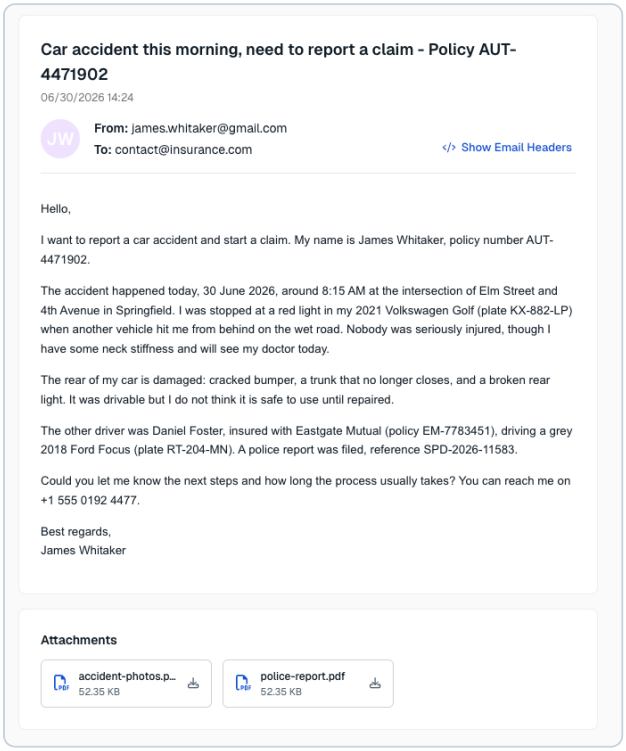
Accessible to everyone

Flowable Work conforms to WCAG 2.2 AA, confirmed by an external audit, with full keyboard navigation and screen reader support. An Accessibility Conformance Report (VPAT) is available through your account manager.

FLOWABLE WORK

06 Documents and content

Much case work starts with a document and ends with one: an application form arrives, evidence is collected and reviewed, a contract or a decision letter goes out. Content services are part of the platform. Documents live with the case they belong to, trigger the work that has to follow, and are generated from the case data when the work is done.



An inbound email in Flowable Work: the message rendered as received, each attachment captured as its own document.

Documents live with the case

Documents are organized in folders and linked to the case, process or task they belong to, and every user has a personal folder. Document types with their own properties are defined as simply as a form; metadata can classify documents and drive automation such as routing by document type. Users upload new versions, open any earlier version and download all documents of a case in one go, and Word, Excel and PowerPoint open documents stored in Flowable directly.

Documents drive the work

Cases and processes react to document events: a document created, updated, renamed, moved or deleted, a document type assigned or metadata changed, filtered by folder, type and filename. A review process can start the moment a contract arrives. Emails received through an inbound channel become documents too, the message and each attachment separately, attached to the case and shown in a dedicated email view.

✦ Documents understood by agents

A document agent can recognize the type of every document as it arrives, whether uploaded to a form, added to a case folder or received as an email attachment, extracts the fields the case needs and writes them as typed case data for a person to confirm. Document events then start the work that follows, so a signed contract triggers its review the moment it lands. The chapter AI agents in your operations describes document agents in detail.

Documents generated from the case

Word and PDF documents can be generated from managed, versioned templates with case data and language variants, existing PDF forms are pre-filled, and output can be PDF/A for archiving. Office documents convert to PDF renditions for in-browser reading, thumbnails give visual cues, and several PDFs merge into one to assemble a complete file from generated and standard pages. Emails and messages can draw their standard texts from content templates.

OCR and extraction with ABBYY

Processes can send documents to ABBYY for text recognition (OCR) and data extraction, and a form component embeds the ABBYY review screen in a Flowable task so extracted values are corrected in place. The ABBYY connector is licensed separately.

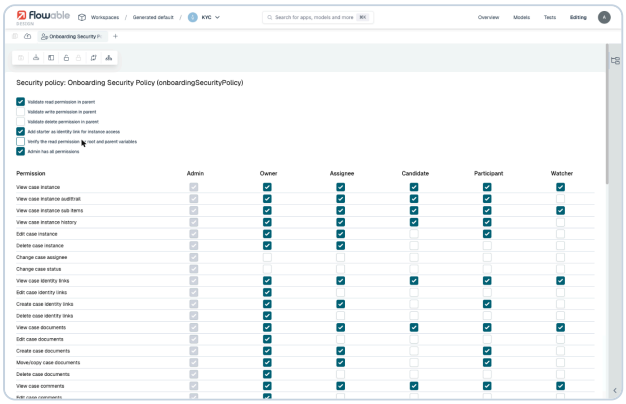
✦ Ask AI about a document

Users ask questions or run predefined prompts against a document attached to their work: summarize, extract insights, check grammar, translate, or examine the sentiment of an inbound customer email. Document agents, described in the chapter on agents, can classify and extract from documents the moment they arrive.

07 Security and compliance

Flowable is built for organizations whose work is regulated. Security is about what each person, each system and each agent can see and do: at runtime for case workers, at build time for modelers and in administration for operators. Every action leaves a trace that an auditor can follow.

Runtime security



A security policy in Flowable Design: which roles may see and perform which operations on a case, its processes and its tasks.

Your identity provider, your users

Users log in once through any OAuth2, OpenID Connect or SAML provider such as Microsoft Entra, Keycloak or Okta, or through LDAP and Active Directory with user and group synchronization; authentication is pluggable through Spring Security. Systems authenticate with access tokens, technical users or OAuth2 client credentials, and a user's identity can be passed on to downstream systems so a call runs on that user's behalf. Secrets for outbound connections are kept in Flowable Hub, referenced by name and never shown again after creation.

Permissions down to the operation

Security policies define who sees and does what on cases, processes and tasks, per operation, based on the role a person holds on that specific case or task: the owner who is accountable, the assignee doing the current step, participants who contribute, candidates who may pick it up and watchers who only follow it. Someone with no role on a case does not see it. A four-eye principle can keep the person who performed a step from approving it.

One installation, many tenants

Business units, subsidiaries or clients run as tenants on one installation. Every record carries its tenant, every query and API call is scoped to it, and tenant administrators see only their own. Each tenant has its own users, identity settings, theme, variables and secrets; apps published to the default tenant are shared with all, and inbound events are routed to the right tenant. Runtime limits on loops, transaction duration, request rates, expression access and uploads keep one faulty model from affecting the others.

Retention and the right to be forgotten

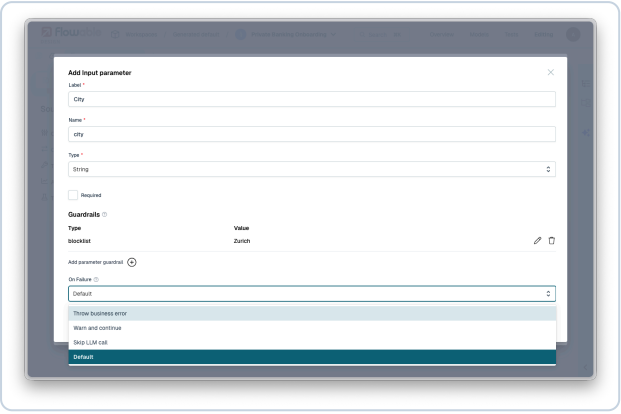
Housekeeping configuration defines how long completed work is kept, and a housekeeping task inside a process or case deletes one specific completed case with its documents, comments, audit entries and index data on demand, which is how a right-to-be-forgotten request is handled.

One audit trail, end to end

One trail records everything that happens to a case: every user and system action with time and identity, which decision rules fired and why, business entries that audit tasks write against a client or contract and readable only by those allowed to see that object, every service-level measurement, and every step of every AI agent from prompt to tool call to guardrail result. The same trail covers administrative changes in Flowable Hub and, in Flowable Design, which revision was published where, by whom and with whose approval. Entries can stream to an external audit system as they occur.

AI under the same governance

Agents add no permission rules of their own. They act inside the permissions, data rules and audit trail described above, with four controls on top.



A deterministic input guardrail on an agent operation in Flowable Design: ranges, patterns and block lists are checked before anything reaches the AI model.

✦ Agents inherit the platform's permissions

An agent works inside the case that invoked it and sees only what that case and its operation declare. In case chat it acts inside the chatting user's permission set, and through the Work MCP server an assistant is limited to what its user may do. Which LLM an agent may use is set centrally in Flowable Hub, per agent.

✦ Data minimization by design

Every agent operation declares which case data it reads and which it writes, so an agent receives only the data its step needs and its reach is visible field by field. A sanitizing guardrail can redact personal data before it reaches a model, and AI models hosted in your own cloud tenant keep the data where it is.

✦ Guardrails on every call

Deterministic checks, guardrail agents judging against a policy in plain language, checks by registered services and ready-made connectors for harmful content, personal data and prompt injection can inspect what goes into and comes out of every LLM call. A violation is handled in the model like any other exception: retry with corrected input, review by a person, or stop.

✦ An audit trail for every agent step

Prompts, responses, tool calls, guardrail results, retrieved knowledge and the AI model and vendor that produced each result are recorded, with granularity set per agent. The agent timeline in Flowable Work and Flowable Hub replays any run, evaluators score quality continuously, and agent data follows the same retention rules as case data.

Build-time and administration security

Who may change what

In Flowable Design, access is set per workspace and per app, and custom roles bundle fine-grained permissions: create, edit, delete, publish, approve publish requests, force-unlock and more, tenant-wide or scoped to a workspace or app. Together with deployment approval this governs the whole path from model to production.

Administration with separation of duties

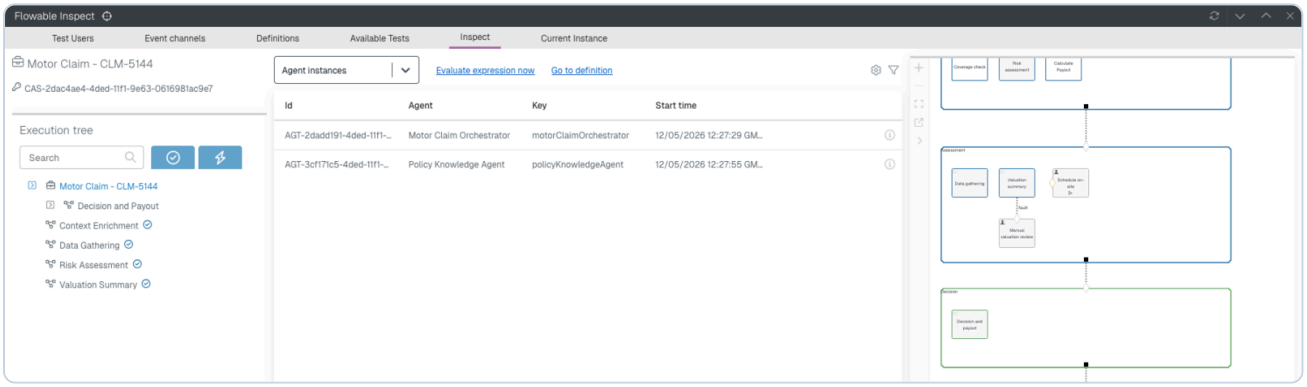
Flowable Hub has its own role model with built-in and custom roles granted per connected environment, so support staff get read-only access to production while operators keep full control. Every administrative change is written to an audit log.

08 Operate, test and improve

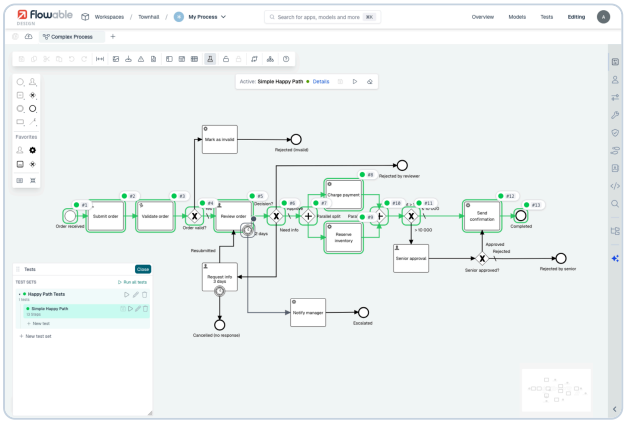
Applications that run for years need to be tested before go-live, kept healthy in production and changed while thousands of instances are still running. Flowable builds testing and debugging into Work and Design as the Inspect feature set, provides Flowable Hub for administration, and indexes all history so that operations and management see how the work performs.

Test and debug with Inspect

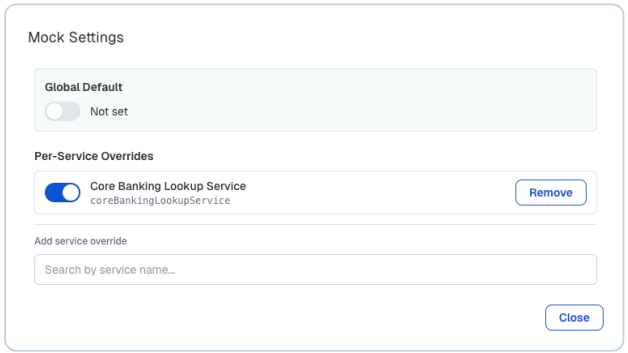
Inspect is a feature set inside Flowable Work and Flowable Design for development and test environments. It takes the guesswork out of testing case and process models: step through executions and replay the tests modeled in Design on every change.



Inspect in Flowable Work: the execution tree, the case diagram and the agent instances of the current case.



A test in Flowable Design: the steps of the passed test are highlighted on the process diagram, with the test sets and their results beside it.



Switching service mocks on globally or per service from Inspect.

Debug a running case like code

Open the Inspect panel on any instance to walk the execution tree, read or change variables, evaluate expressions in the live scope, trigger timers and events, and step through the execution or jump between breakpoints. Every exception is stored on the instance with its stack trace, and a form inspector shows a form's structure and the dependencies between its components.

Tests that travel with the model

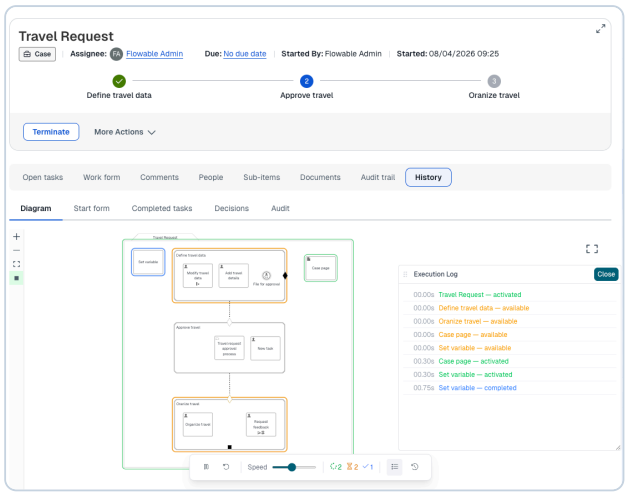
Tests are modeled in Flowable Design for cases, processes, services, actions and agents, or generated by AI from the model, with assertions on variables and expressions, including structured JSON comparison. They are replayed on demand or through the API in automated pipelines and run in bulk with a pass and fail report. A test can also be recorded by running through a case or process in Inspect. Test sets move between Design and Inspect by export and import.

Replay what an instance actually did

The token simulation from Flowable Design also replays real process and case instances in Flowable Work. On the History tab of a work item, the Diagram view offers Replay execution: tokens follow the recorded history, gateway paths are the ones the instance took, and the run stops at the activities the instance still waits on. The execution log and the step view work as in the editor, and when the model has changed since the instance ran, the replay says so.

Test without the outside world

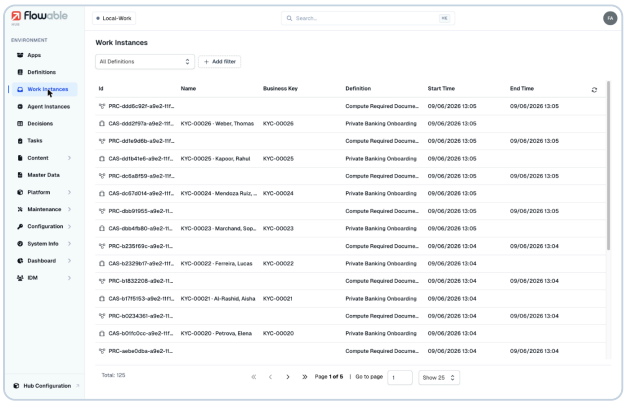
Service mocks replace external systems and are switched on globally or per service from Inspect, events are triggered by hand as if a broker had delivered them, and testers switch users to see the application exactly as another person would.



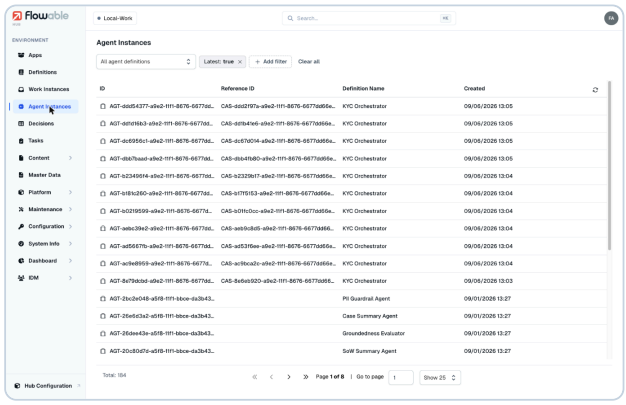
Replaying a real case from the History tab in Flowable Work: tokens follow the recorded history and the execution log lists what happened.

Administer with Flowable Hub

Flowable Hub is the administration console: apps, definitions, instances, jobs, configuration and system health in one place, across several Flowable environments.



Work instances in Flowable Hub: cases and processes across all apps, with filters and drill-down.



Agent instances in Flowable Hub: every agent run linked to its case, with details, invocations, evaluations and jobs.

See and fix everything that runs

Searchable lists cover processes, cases, tasks, forms, decisions, jobs and events, active and completed, with drill-down into every state, variable and role assignment. Failed jobs are restarted, stuck tasks reassigned, delegated or completed, variables corrected and the state of a case or process changed directly, for example jumping back after a wrong path.

Health, configuration and housekeeping

Real-time charts show CPU and memory, database connections, job throughput and failures, events and active users. A health dashboard summarizes the state of the platform and turns findings into advice, such as jobs that failed repeatedly and wait for an operator or slow-job incidents per process or case definition, each linked to the screen where operators look further. Tenant variables, secrets, OAuth2 clients, master data and sequences are administered without redeployment, and a housekeeping wizard removes completed instances with a live preview of what will be deleted.

✦ Agent operations

Agent connections with providers, AI models and prices are configured centrally. Agent dashboards show invocations, tokens, cost, guardrail violations and evaluator pass rates per agent, version and model, and the agent instance view opens the full timeline of any run: every LLM call, tool call and guardrail check with its input, output and duration. The same timeline is available in Flowable Work with Inspect.

Change while the work is running

A case that runs for months will see several releases of its own model. Flowable treats that as normal: versions coexist, and moving work between them is a routine operation.

Versions run side by side

Every publish creates a new version of the app without downtime. Running instances continue on the version they started with, new instances start on the latest, and both are served by the same nodes. Where one model references another, the modeler decides whether running work is pinned to the referenced version or follows the newest.

Migrate when you are ready

Running process and case instances migrate to a new model version with visually defined mappings. Flowable Hub analyzes the instances first, proposes the mapping and migrates a parent with all its child processes and cases in one operation. Migrations, reindexing and bulk deletions run as monitored batches with per-part results, also for hundreds of thousands of instances.

Adjust one instance without a release

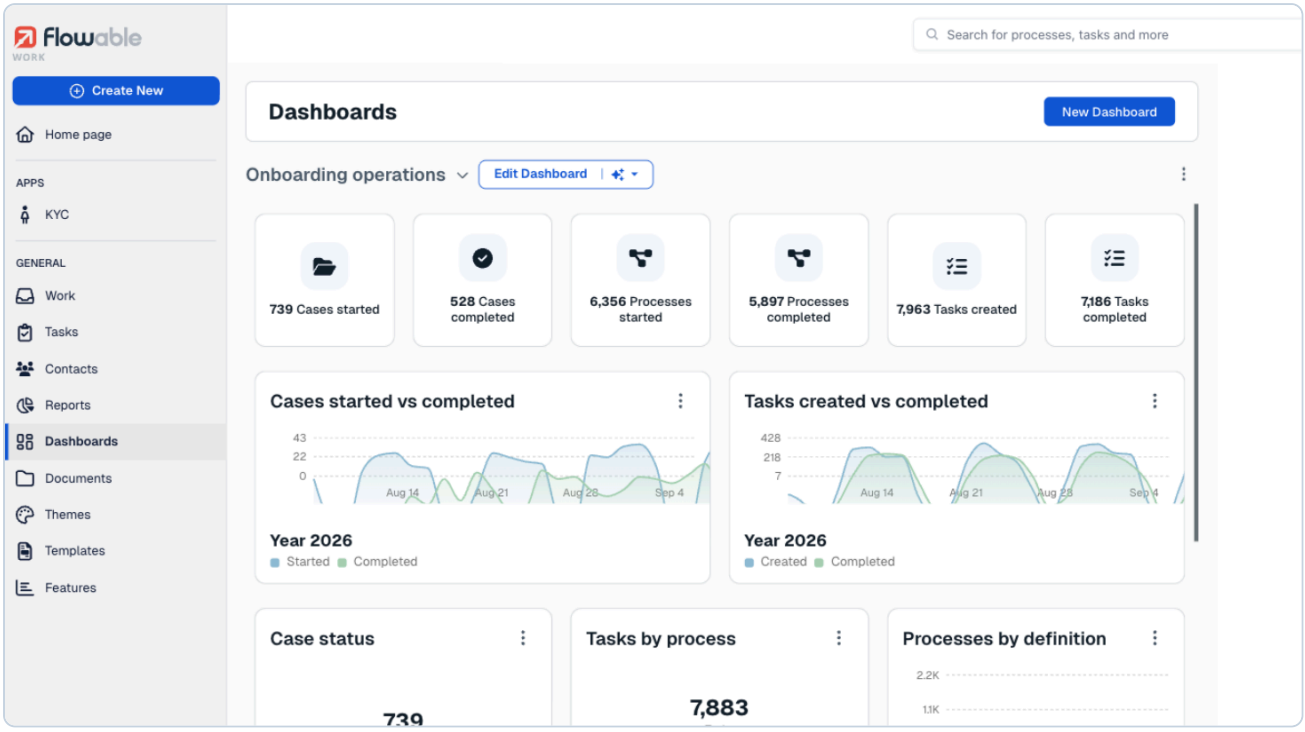
A single running instance is adapted in place: its state is moved forward or back by an operator, and a completed case is reopened along rules in the model. The chapter Running the work describes these from the case worker's side.

Patches without a maintenance window

Service packs focus on bug fixes and are applied while the platform keeps running. The major releases are described in the chapter Architecture, deployment and openness.

Metrics, reports and dashboards

Every case, process, task and agent run is indexed as it happens. The numbers operations and management need are available without a separate reporting system: in dashboards that business users assemble themselves, in built-in reports, and in the monitoring views of Flowable Hub.



A dashboard in Flowable Work built from the component catalog: cases, processes and tasks started and completed, throughput over time with the weekly rhythm of the business, and breakdowns by status and definition.

Dashboards business users build themselves

In Flowable Work a user creates a dashboard, picks components from the catalog and arranges them on a canvas: counters, pie, bar and line charts and drill-down tables for cases, processes and tasks, filtered by definition and time period, with shared filters in the header. Every user has their own dashboards, and a description in plain language is enough to have AI generate one.

The component catalog

Out of the box the catalog covers instances started and completed, status breakdowns, distribution and comparison by definition, throughput over time and sortable detail tables for cases, processes and tasks, plus agent components for invocations, tokens, cost, tools and guardrails. Custom dashboard components are modeled in Flowable Design from query models, and any indexed business data can become a chart.

Reports on frequency and duration

Built-in reports show how many cases, processes and tasks were created and their average, minimum, maximum and total durations, grouped by definition, hour, week, month or year, as bar, line or pie charts. A heat map projects frequency and time spent onto the process diagram and shows the bottleneck on the model itself.

Service levels and searchable business metrics

SLA models track reaction and completion times per task in working hours or around the clock, escalate by email or into the case conversation, and write every measurement to an SLA audit table and index, so service-level reporting has an auditable source. Values from case data, such as an amount or a risk class, can be made available to dashboards and lists, which then slice by them.

Operational metrics in Flowable Hub

Hub adds time-series dashboards over a chosen period for instances, jobs, events, service calls, REST requests, indexing, active users and system resources, and on the Agentic Case Platform for agents and agent cost. A live health dashboard summarizes the state of the platform, and the same metrics can be exported to your own monitoring tools such as Grafana or Datadog.

Analytics that respect permissions and retention

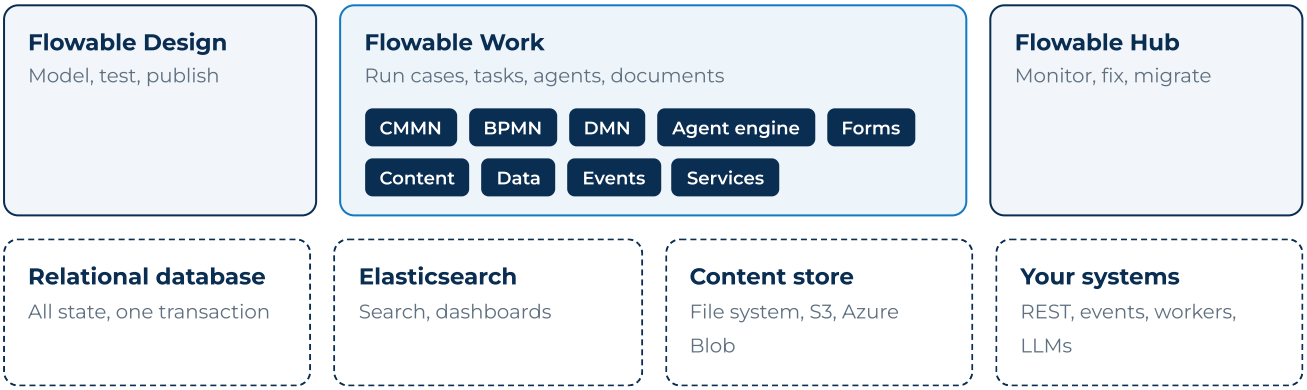
Dashboards and reports show a user only the data that user is allowed to see. Per model you set how much history is recorded, from every variable change down to start and end only.

PLATFORM

09 Architecture, deployment and openness

The platform is built for long-running work at enterprise volume: it scales out across many application instances, keeps all state in one database with no single point of coordination, and has a clear answer to what happens when something fails. Flowable runs it for you in Flowable Cloud, or you run it yourself.

Performance and reliability



Coordination-free clustering

All case and process state lives transactionally in the database and nodes coordinate only through it. There is no cluster membership protocol and no session replication. Any node executes the next step of any instance, and throughput grows with the number of nodes to thousands of process steps per second.

Failures never lose work

Transaction boundaries are part of the model, and a failure never rolls back completed work. A failed step retries with backoff; a step that keeps failing lands in a dead-letter queue where operators inspect and replay it in Flowable Hub. Work in flight on a node that stops, or with an external worker that crashes, is released after a lock timeout and continued elsewhere.

Backup and recovery

Flowable Cloud is backed up with a recovery time objective of 2 hours and a recovery point objective of 1 hour. The procedure covers the three stores that hold your work: database, search index and content store.

Built for years of history

Completed work is stored separately from active work, and response times stay stable as history accumulates. History is indexed for search and analytics.

Deployment options and openness

ISO/IEC 27001:2022 · ISO 9001:2015 · SOC 2 Type II · ISAE 3402 · GDPR · CCPA · WCAG 2.2 AA

Flowable Cloud

Flowable runs the platform for you on Microsoft Azure, in EU or US regions with others on request, on a geo-redundant, multi-zone architecture with an availability SLA. Upgrades, patches, monitoring and backups are Flowable's job, and development, test and production environments are available. Where your requirements call for it, a dedicated single-customer environment gives you physical data separation, the region of your choice, your own release schedule and full freedom for custom code.

Certified operations

Flowable is certified to ISO/IEC 27001:2022 and ISO 9001:2015, holds a SOC 2 Type II attestation and an ISAE 3402 report, and operates under GDPR and CCPA. Flowable Cloud encrypts data in transit and at rest, is penetration tested and protected against DDoS and malware. Reports, policies and subprocessors: trust.flowable.com.

Self-managed, with full data sovereignty

Where your requirements demand it, you run the platform in your own data center or cloud account and keep control over where every case, document and index lives, who operates them and which region they never leave. An official Helm chart deploys the platform on Kubernetes and OpenShift, and Terraform blueprints for AWS and Azure set up development, test and production environments. The platform runs on PostgreSQL, Oracle, SQL Server, MySQL, MariaDB, DB2, Azure SQL or Amazon Aurora, stores content on the file system, in the database, in Amazon S3 or Azure Blob Storage or in a custom store, uses Elasticsearch for search and analytics, and needs a load balancer and a shared or object store for content, with published sizing guidance. A health dashboard in Flowable Hub rates each node and its job executors and turns findings into advice, from a stalled executor to a JVM heap below the recommended size. The runtime is a Spring Boot application on Java 17, 21 or 25.

✦ Your models, your data, your rules

Flowable ships no AI models of its own; each agent runs on a standard AI model, frontier or open-weight, that you pick: hosted, in your own cloud tenant or on your own hardware through any OpenAI- or Anthropic-compatible endpoint, switched centrally in Flowable Hub. Knowledge bases, documents and every recorded prompt and response stay inside your Flowable environment, and guardrails can redact what must not reach an AI model at all.

Releases, upgrades and compatibility

Flowable publishes two major releases a year and monthly service packs that focus on bug fixes. Service packs are applied without downtime, and a major release upgrades the database automatically, with the release notes listing every change that needs attention. The public REST API follows a written compatibility policy: a breaking change is deprecated first and removed at the earliest in the next major release.

Open standards, open source core

Process, case and decision models follow BPMN 2.0, CMMN 1.1 and DMN 1.3 and are exchanged as standard XML. Notation, skills and tooling are industry-wide. The execution engines are open source under the Apache 2.0 license, maintained by Flowable engineers and used by a large community. The code that executes your models is open for anyone to read. The same openness applies to AI: any LLM vendor through standard APIs, MCP for tools and A2A for agents on other platforms.

WORKING WITH FLOWABLE

Support

A Flowable subscription covers the platform and the people behind it, from the first development steps through production. The subscription includes the following services.

Production support

Assistance for problems in production with guaranteed response times.

Service packs

Service packs focus on bug fixes, and production installations stay current with minimal functional change.

Community

The open source and enterprise forums provide an active community for informal help and exchange.

WORKING WITH FLOWABLE

Enablement

Flowable's professional services team offers these engagements in addition to your subscription.

Flowable Kickstart

Before the first project, Flowable architects work out the platform setup with your team in three to five workshops: deployment type, tenant strategy, single sign-on, monitoring and, for self-managed installations, sizing and database.

Flowable Permanent Mentor

A dedicated Flowable specialist embedded in your project at a fixed weekly allocation, for continuous guidance, hands-on coaching and knowledge transfer.

Flowable Health Check

Before go-live or when something misbehaves, Flowable experts review your platform and applications against best practices, from architecture and modeling to security and monitoring, and report their recommendations.

Flowable Scheduled Advisor

Prepaid days with experienced Flowable consultants, called on when you need them for particular phases of a project.

Next steps

See it running

Ask your Flowable contact for a demonstration on your own use case, or start with a trial environment on Flowable Cloud.

Check the trust center

trust.flowable.com publishes certifications, reports, policies and the subprocessor list for Flowable Cloud.

Read the documentation

documentation.flowable.com covers modeling, operations, security and the APIs in depth; release.flowable.com describes what is new in 2026.1.

Talk to us

flowable.com and info@flowable.com for licensing and a conversation about your portfolio of work.

Flowable

flowable.com
info@flowable.com