



INTRODUCTION TO INRULE DECISIONING



▼ IF YOUR ORGANIZATION IS CONFRONTED WITH THE UNCERTAINTY OF NOT KNOWING:

Who is maintaining decision logic,
Where it is located or how to access it,
How it is defined,
What the lifecycle is for changing it,
How quickly you can turn around changes to production, or
Whether it can be reused in other applications...

Then InRule® Decisioning might be the right solution for you.

OVERVIEW



At **InRule Technology®**
we make automation accessible.
We do it faster, with more
accuracy and transparency than
anyone else.

Determining if a software product is the right solution for your organization can be a time-consuming and sometimes confusing endeavor.

To help you with your evaluation and provide guidance on what you need to look for during that process, this Introduction to InRule will walk you through key aspects of our decision automation technology so that you can make the best choice for your organization.

Our author-first approach empowers both business and technical users to change enterprise decisions in applications—with less effort, cost, and risk than programming—to help organizations keep pace with changing marketplace conditions and customer needs.

WHY USE INRULE DECISIONING?

The challenges you currently face with your organization's decision logic may vary depending on the state of your business applications, development lifecycles, or available resources.

Since 2002, InRule Decisioning has helped over 400 customers around the world achieve measurable results and benefits that have positively impacted their organization.



Empower Subject Matter Experts to control decision logic



Centralize decision logic for easy access and reusability



Greater transparency



Gather and analyze insights into your decision logic



Faster change cycles



Standardize across the organization

TABLE OF CONTENTS

1 INRULE DECISIONING COMPONENTS

2 PERFORMANCE

3 DECISIONING ARCHITECTURE

4 DEPLOYMENT OPTIONS

5 LICENSING

6 PROFESSIONAL SERVICES

7 EVALUATION STEPS



Introduction to InRule
INRULE DECISIONING COMPONENTS



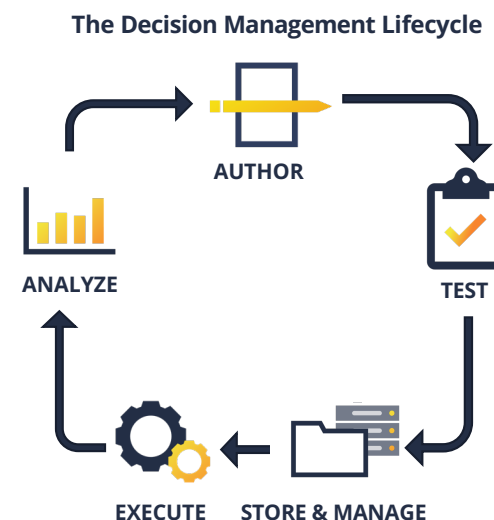
WHAT IS A DECISION PLATFORM?

InRule Decisioning is a decision platform, an end-to-end solution for everything related to the decision management lifecycle, including authoring, testing, storing, deploying and executing decision logic.

Because every organization's decisions are different, we provide a solution that can work with any business application: from insurance, healthcare, financial services, and commercial applications to public sector agencies.

While our decision platform is built with the Microsoft .NET stack, you don't have to be a .NET organization to use InRule. With today's service-based architectures, InRule integrates with almost any application (desktop, web, mobile) or platform (Microsoft Dynamics® 365, Microsoft Azure, Salesforce®).

A decision platform can be your "*single source of truth*." Instead of replicating logic and deploying it across each of your business applications, InRule allows you to define the logic once and let all your applications consume the same decisions.



AUTHOR

irAuthor®, our desktop tool, is used for the initial setup. This is where you'll establish connections to external systems, create data models—no matter how complex the relationship—author rules, and more.

Author Studio™, our web-based authoring tool, empowers business users to easily author and maintain rules from anywhere, with nothing to install. Both provide a variety of methods for authoring rules and support InRule's unique vocabulary feature, which gives rules business domain specificity and shelters newer authors from complexity.



The business language editor provides a point-and-click interface for writing rules in natural language.

The decision table editor helps users define and maintain large sets of complex rules in a compact format that is easy to visualize and work with.

Users can also build expressions and formulas using the hundreds of built-in functions available **in irAuthor's syntax editor**. Together, these editors create the business rules that represent a use case or decision. This collection of rules is referred to as a rule application

AUTHORING SKILLS

For more than two decades, we have worked with hundreds of customers on their projects. During that time, we have found that there are certain competencies that translate into being an effective Decision Author.

These are foundational skills that a Decision Architect or Decision Author should have to fully leverage the capabilities in irAuthor and Author Studio to create and maintain rule applications.

Every irAuthor and Author Studio user will go through a training session conducted by one of our consultants.

This session will cover all the features available and ensure that your Decision Authors understand the best practices that should be followed to build stable and maintainable rule applications.



KEY SKILLS FOR RULE AUTHORIZING

- **Understand data structures with parent-child relationships**
- **Translate a use case into a list of ordered steps**
- **Combine functions to create formulas or logical expressions**
- **Implement cause and effect logic using If/Then/Else statements**
- **Implement table lookups based on matching criteria**
- **Ability to identify logic gaps**

TEST



Testing is an essential step in the decision management lifecycle. **irVerify**® is the testing tool that can be launched directly from irAuthor. When an author creates new rules or modifies existing rules, they will need to validate the decision logic to ensure they are getting the intended results.



irVerify can be used for either unit testing or regression testing. The author can create and save test data for consistent and reusable test scenarios.



irVerify includes multiple logging options that can be used to capture details related to a test scenario. These logs include: Summary Statistics, State Changes, Rule Engine Feedback, Tracing Information, and Performance Statistics.

STORE & MANAGE



Rule applications created in irAuthor and Author Studio are saved to irCatalog.



irCatalog consists of a database, a web service, and an admin utility, known as the Catalog Manager. The database is the repository, and the web service will be the mechanism used at design time or runtime to communicate with the database.



At design time, authors will retrieve a rule application from the database, check out to make modifications and then check back in.



This check-out/check-in process creates an audit record in the database. The audit records provide a means for ensuring organizational compliance.



Through the Catalog Manager, you can set up roles and permissions to grant or deny access to your organization's rule applications. The Catalog Manager is also used to deploy changes from one catalog to another as part of a decision management migration process.

EXECUTE



irServer® is our proprietary algorithm that is used to execute the decision logic defined in a rule application. Execution of decision logic is stateless. For a given decision execution request, you send the input data to irServer and specify which rule application to execute. When the process is completed, irServer will return the output state, which includes the results of the rule execution as well as optional logging information.



InRule Decisioning includes an implementation of irServer via a REST API. This REST API has been constructed to accommodate any type of transaction for any of the rule applications that you author. This REST API can be scaled out to handle larger volumes of transactions when needed.



InRule also provides alternate means for **integrating** irServer depending on your application or your organization's preferred technologies.



EXECUTION ALTERNATIVES

In addition to the out-of-the-box REST API, InRule Decisioning also includes an **SDK component** for those organizations that prefer to create a custom integration using either the .NET Framework or .NET Core.

You can also convert your rule applications into executable JavaScript files via our packaging service known as **irDistribution™**. These files can then be deployed for scenarios where decision logic needs to execute in the client application or in disconnected mode. Both the SDK and JavaScript options open the possibilities for how you can execute your decision logic. Azure Functions, Azure App Services, AWS Lambda Functions, node.js, and Containers, are just a some of the ways our clients have customized their integration architecture.



ANALYZE

For an enterprise to achieve its business goals, it is imperative to define and measure the right metrics. However, determining the appropriate mix of metrics to track in order to achieve a business objective is often an iterative process.

InRule Metrics makes it easy to select and monitor several values in the application. The result? Data-driven insights into the impact of your organization's automated decisions on vital KPIs.

InRule makes it easy to analyze and compare the impact of automated decisions on your business:

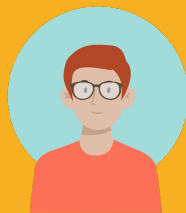
By defining and selecting rule and data metrics, applying the rules to a batch of data, and then storing the execution results in a data store, you can then develop reports and analytics to gather insights into your decision logic and identify areas of improvement.



ROLES AND RESPONSIBILITIES



InRule Decisioning consists of multiple components geared for different users. The success of the implementation depends on the right people doing the right job.



Decision Architect
(Business or IT)

Drives the overall decision design based on the organization's use cases; manages the more technical aspects of a rule application in **irAuthor** and **Author Studio**, including schema definitions, connections to external systems or creation of custom templates



Decision Author
(Business or IT)

Implements the decision logic by using the Business Language, Decision Table and Syntax editors in irAuthor and Author Studio; tests and validates decision logic with **irVerify**



Decision Integrator
(IT)

Configures business applications to call the **REST API** based on user actions, triggers or events that require execution of decision logic or leverages the **SDK** or JavaScript components when implementing a custom integration



Introduction to InRule PERFORMANCE



EXECUTING DECISION LOGIC

We are often asked how fast decision execution will be. Unfortunately, it is almost impossible for us to provide a specific number since every decision is different.

In our world, a long-running rule execution would be anything over 150ms per transaction. Most of our customers are executing rule applications in much less time than that; some are *executing decisions in single-digit milliseconds*.



Factor #1: Number of rules

The more rules in your rule application, the longer the execution can take. It is important to keep in mind that not every rule will execute every time due to conditional statements which can impact the decision execution sequence from transaction to transaction.

Factor #2: Size of record data

When the data is passed to irServer, it must map it into a data structure. Some rule applications may call for passing in minimal record data and others may require a complex hierarchy of data that consists of parent-child relationships and collections. The logging information in irVerify will capture execution times to help the author understand the contributing factors for a given rule application.



Introduction to InRule

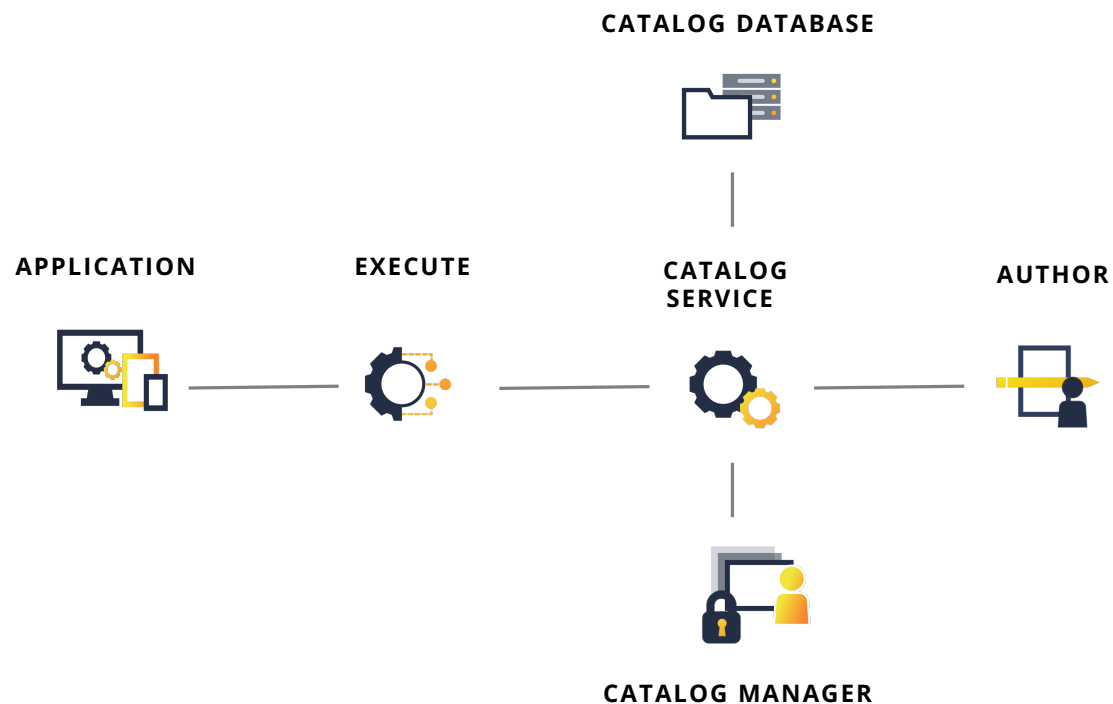
DECISION PLATFORM ARCHITECTURE



ARCHITECTURE OVERVIEW

Because **InRule Decisioning** is so flexible, it can be leveraged in almost any application through a service-based architecture so that the same decision logic can be defined once, saved to the catalog, and executed wherever and whenever it is needed.

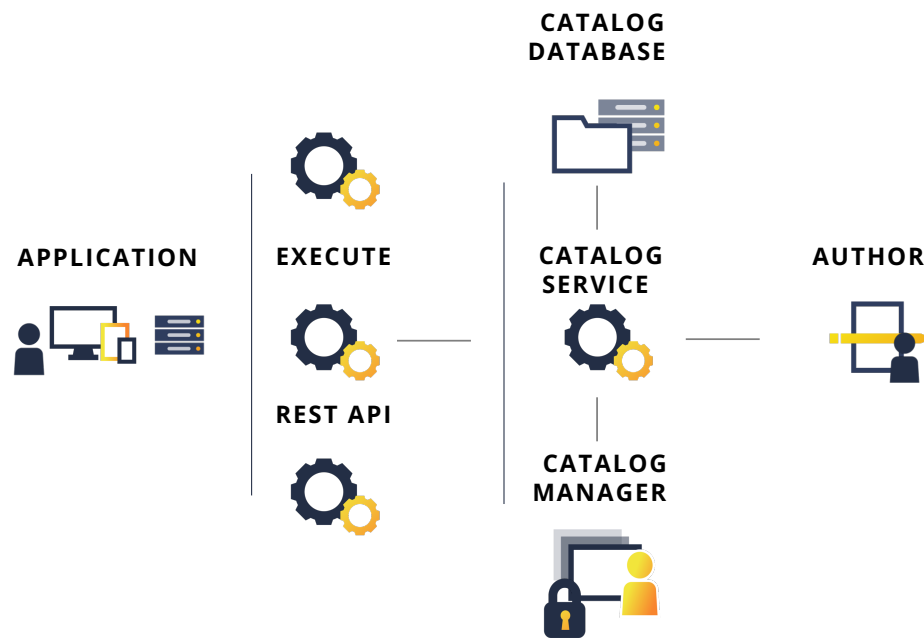
The **Catalog Database** acts as your organization's "single source of truth".



ARCHITECTURE OVERVIEW

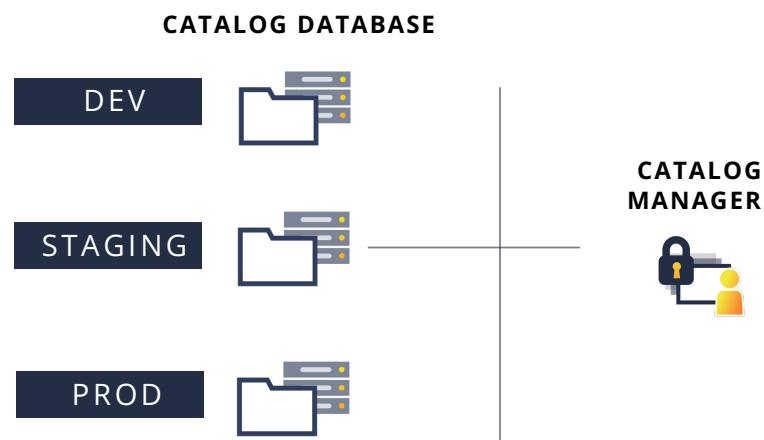
If you have a large volume of transactions that exceed the capacity of a single implementation of the REST API, it can be scaled out to accommodate the additional throughput.

Our **Decision Platform** does not include a load balancer, but most organizations have an existing load balancing protocol which can typically be reused for your decision execution architecture. In this scenario, all execution points share the same catalog, thereby executing the same decision logic.



DEPLOYMENT ARCHITECTURE

Since most organizations maintain multiple environments, our platform requires a unique catalog and REST API for each one. This allows you to isolate your decision logic so that your authors can safely make changes without impacting production. **The Catalog Manager** contains all the functionality required to migrate changes from one catalog to another as part of a rollout process without requiring any changes or restarts of the REST API.





Introduction to InRule
DEVELOPMENT OPTIONS



DEPLOYMENT OPTIONS

InRule provides multiple options for deploying our decision platform.

InRule SaaS supports the execution of decision logic using the REST API.

With the self-hosted cloud and self-hosted on-prem options, you can choose either the REST API or a custom execution model.





Introduction to InRule LICENSING



INRULE DECISIONING LICENSING MODEL

InRule Decisioning is licensed as a subscription, with overall pricing based on configurable components.

Your subscription will include access to the InRule support site for forums, videos, phone and email support, and unlimited use of our computer-based training for *irAuthor* and *Author Studio*. If you opt for a self-hosted deployment, you will also have access to all platform updates, including minor and major releases.

Licensing of irServer and irCatalog is based on production requirements. Our license model includes the equivalent configuration for a Development and Production environment. Additional environments beyond Production and Development will incur an additional cost.

Support for OEM Licensing InRule can be embedded within your product or solution which gives you the option of letting your customers tailor their own business logic in your packaged solution and devices.

COMPONENT	CONFIGURATION	LICENSED BY
irAuthor and Author Studio	Required	# of named users
irSDK	Optional	# of named users
irServer/irCatalog	Required	# of cores or equivalent required for production
Metrics	Optional	Yes / No



DETERMINING SIZING FOR irServer®

Estimating the appropriate number of cores for **irServer®** is an important decision point both for the success of your application and for your project budget.

However, the performance factors, number of rules and data payload vary from rule application to rule application which is why we strongly recommend the following to help you arrive at a number that will meet your needs:

When evaluating InRule Decisioning, identify one or more use cases as candidates for a proof of concept. An InRule Sales Engineer can work with you to implement those use cases as rule applications.

This process helps to uncover the intricacies and challenges of your decision logic, provides decision modelling solutions using the different rule editors, and acts as a starting point to understand the execution times.

When we can estimate the execution time for a given decision transaction, and we also know the anticipated number of transactions your application(s) will process in a given period, we can then estimate the number of cores you will need for production.



Introduction to InRule
PROFESSIONAL SERVICES



RULE-ORIENTED APPLICATION DEVELOPMENT

Our Professional Services team, known as **R.O.A.D. (Rule-Oriented Application Development)** is involved in every customer project. R.O.A.D. consists of Decision Authoring Consultants and Decision Integration Consultants.



TRAINING



WORKSHOPS



AUTHORING



INTEGRATION



Decision Authoring Consultants train your team on *irAuthor* and *Author Studio* to ensure they follow the best practices and guidance for proper authoring techniques. They can also perform rule application reviews or provide guidance and samples through the design phase of your project. If your timelines are short or if you simply don't have the resources, our consultants can also be used for project work to author your rule applications.



Decision Integration Consultants will train your IT team on proper techniques for using the REST API, the *InRule SDK*, or JavaScript. They handle the more technical aspects of the architecture and integration and can provide guidance on deployment practices, performance optimization, capacity planning, and state management.

INRULE

CONFIDENTIAL · COPYRIGHT ©2023 · INRULE TECHNOLOGY, INC. · ALL RIGHTS RESERVED.

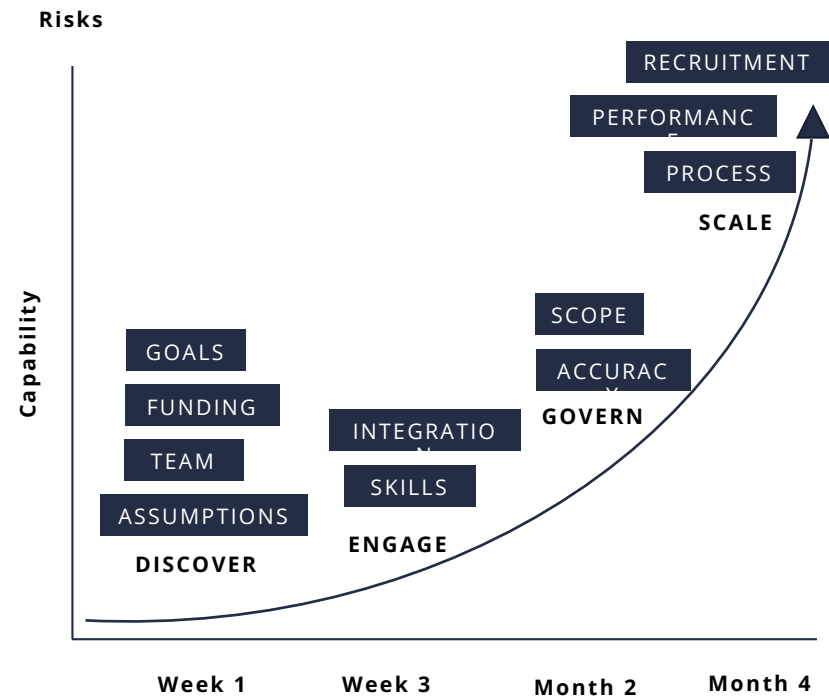
WORKSHOPS

For each project, it's important for stakeholders to agree on why they are doing it, the problem they aim to solve and how success will be defined.

At "go-live" everyone will know why something is better and how it impacts the goals of the business.

Our workshops are designed to create a foundation for project success.

Visit our [website](#) for more details on our workshop sessions.





Introduction to InRule

GET TO KNOW INRULE



RECOMMENDED EVALUATION STEPS

STEP 1: INTRODUCTORY DEMONSTRATION

An InRule sales engineer will take you through an overview of InRule Decisioning components, provide a demonstration of **irAuthor** and **Author Studio** to show how to write and test decision logic, and answer any questions you may have as it relates to the product or your project.

STEP 2: PROOF-OF-CONCEPT

Once you have a basic understanding of how our decision platform works from the intro demo, a proof of concept will provide better context for how your decision logic would look implemented using InRule's authoring tools.

An InRule Sales Engineer can take your use cases and create sample rule applications that demonstrate different decision modelling techniques and highlight key considerations for how to properly author your decision logic.

STEP 3: TRIAL

Our trial is the easiest and most recommended approach to familiarize yourself with InRule. We stand up a hosted environment, so you have access to all **the InRule Decisioning components**.