uEngine6
Business Process Management



Why?

—

Business Process Management

UEngine

Paradigm Shift: As-Is vs To-Be

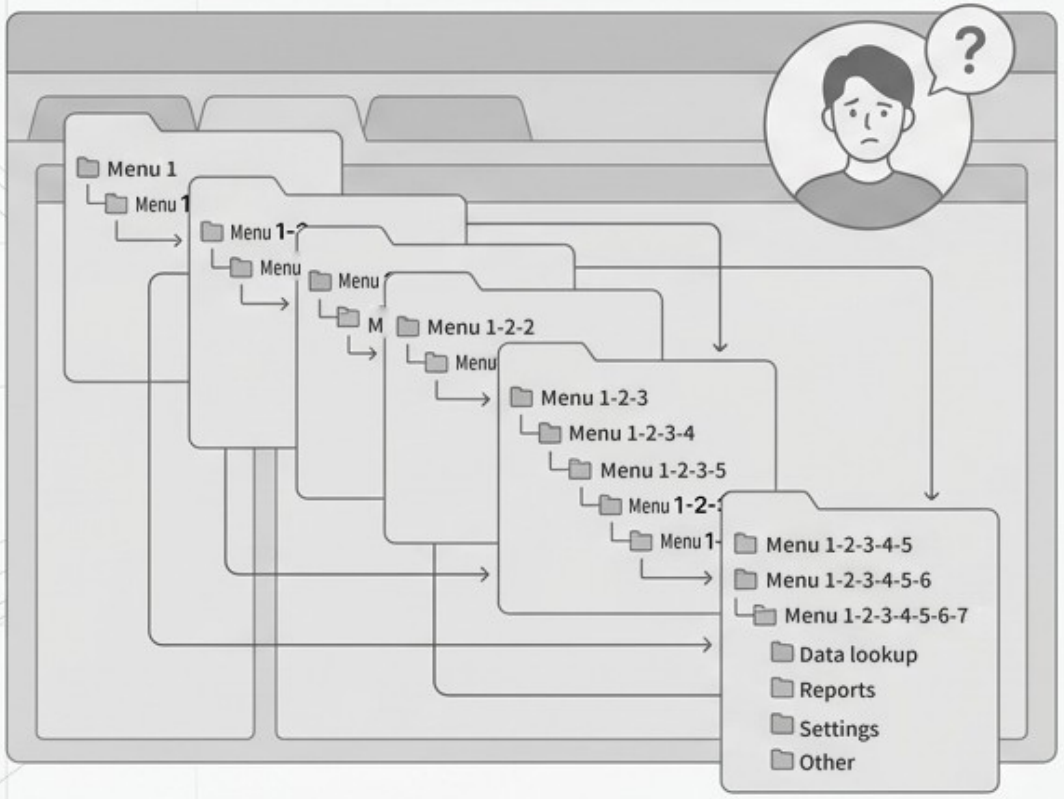
	SoR Only (As-Is)	BPM Integrated (To-Be)
Work Entry	Users search menus (Pull)	Tasks arrive in the worklist (Push)
Sequence	People remember and decide	The engine enforces the sequence
Rules	Scattered in code, Excel, heads	DMN tables, edited by the business
Integration	People open 6–7 sites	RPA/SQL fetch data into variables
Cancellation	No undo path; manual cleanup	Done steps auto-compensated in reverse
Audit	Only claims of 'done' remain	Compliance and reasons kept as evidence

iEngine

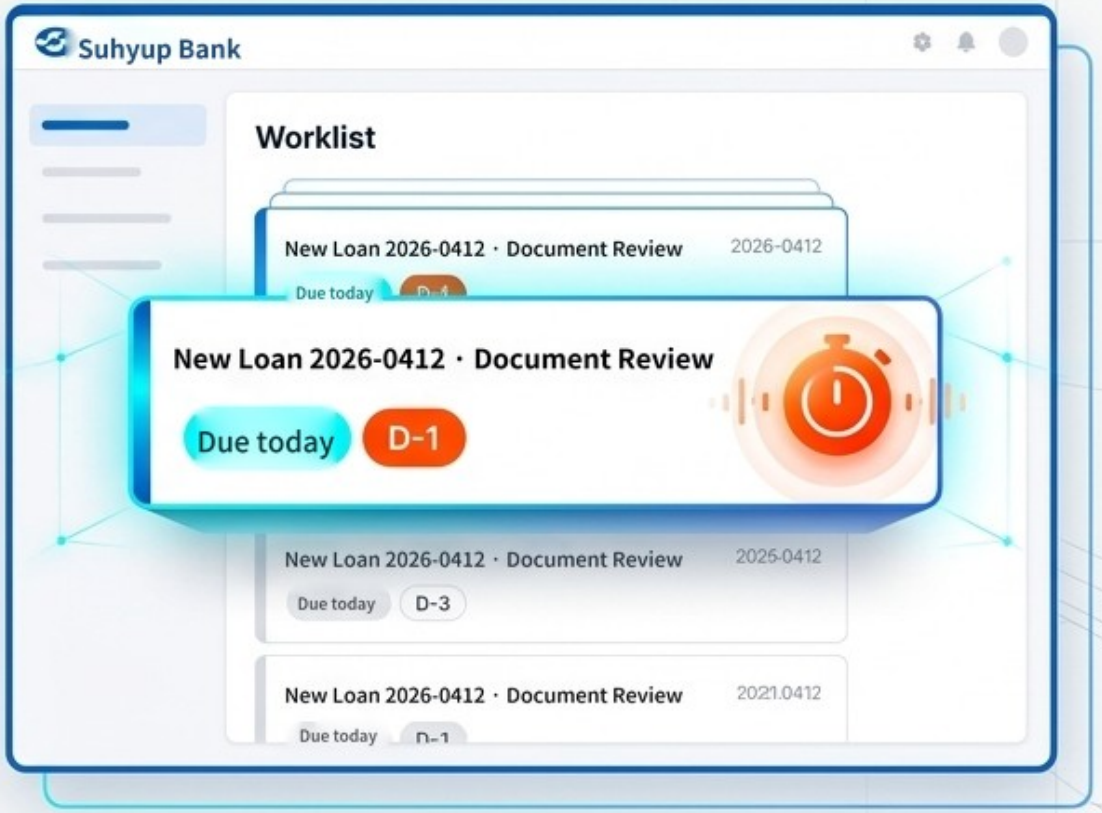
UX Shift: Tasks Come to You (Pull to Push)

The system remembers who must do what and when, and pushes items nearing their deadline to the top.

https://www.youtube.com/watch?v=wda06hMN4mc



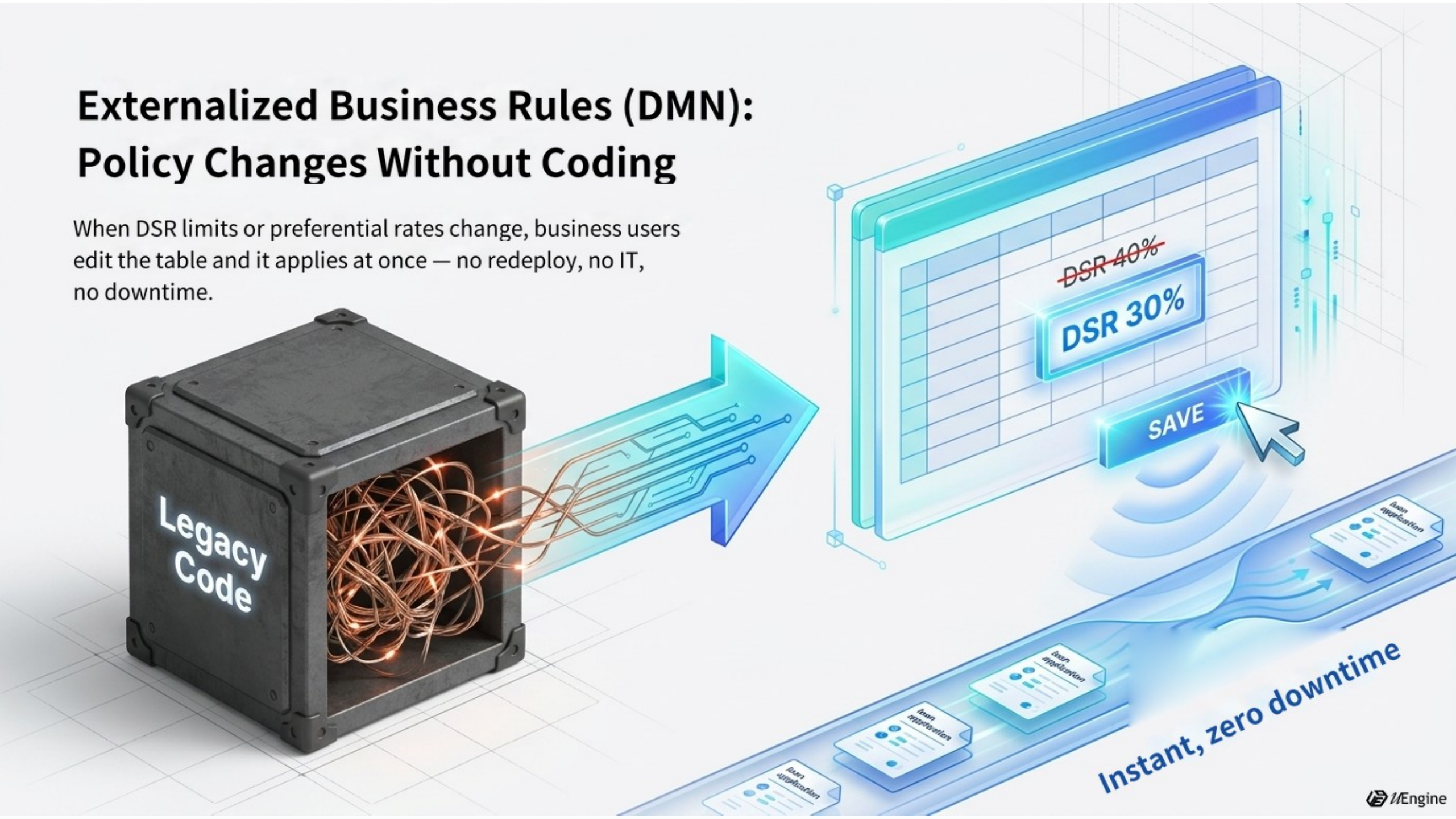
Users dig 6–7 menu levels deep



The system pushes tasks **by due date**

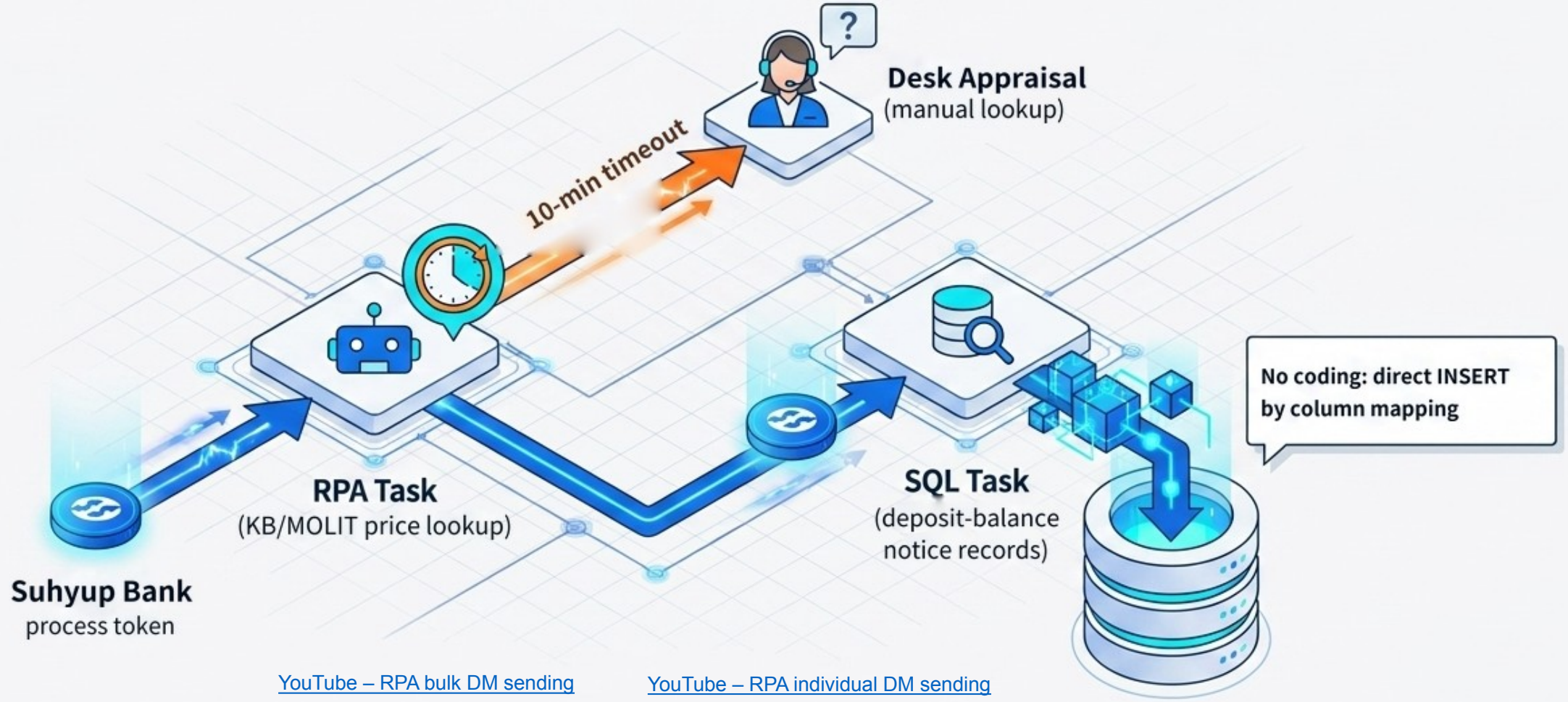
Externalized Business Rules (DMN): Policy Changes Without Coding

When DSR limits or preferential rates change, business users edit the table and it applies at once — no redeploy, no IT, no downtime.



Orchestrated Automation (RPA & SQL)

This is not fire-and-forget. The process engine waits for the bot and the database, and owns the fallback path that returns work to a person on delay or failure.



YouTube – RPA bulk DM sending

YouTube – RPA individual DM sending

Preventing Irreversible Errors: Auto-Compensation

You never draw a single cancel arrow.
On rejection, the engine finds exactly the completed steps and rolls their data back in reverse order.

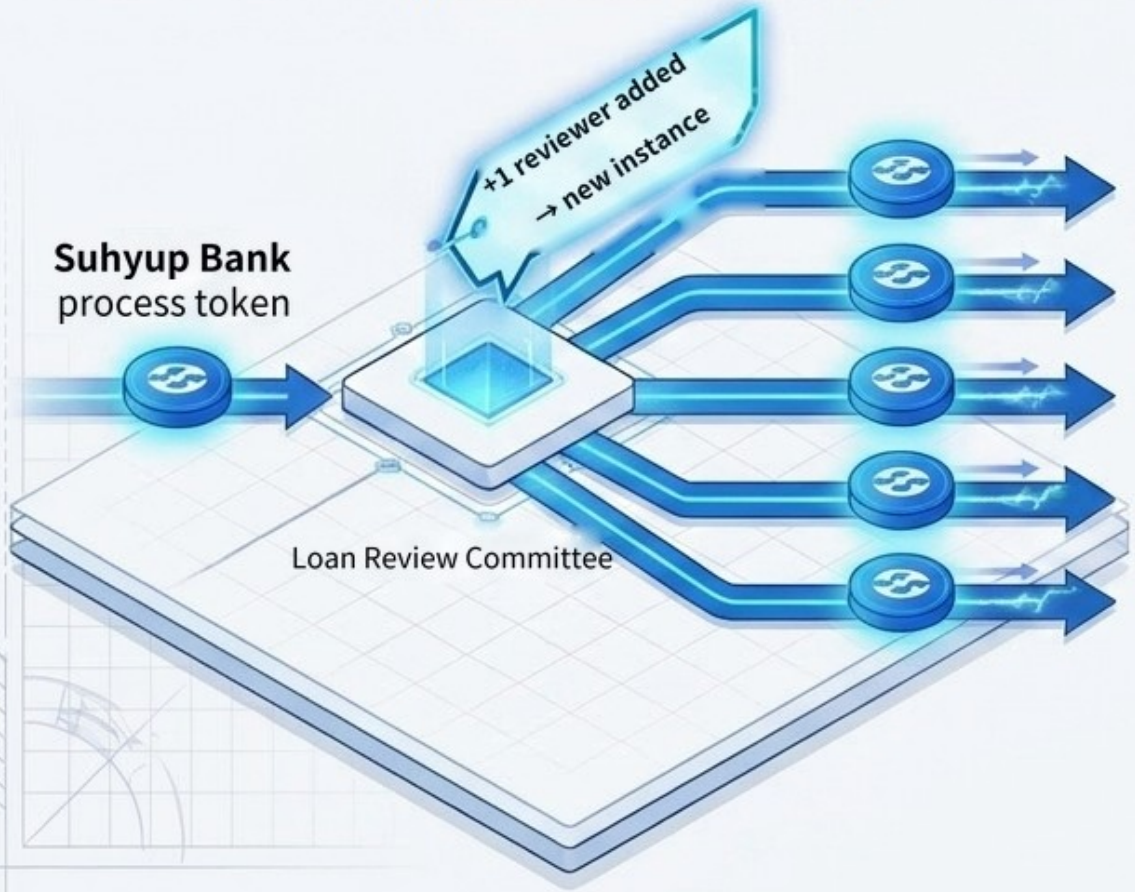


https://www.youtube.com/watch?v=wda06hMN4mc&t=110s

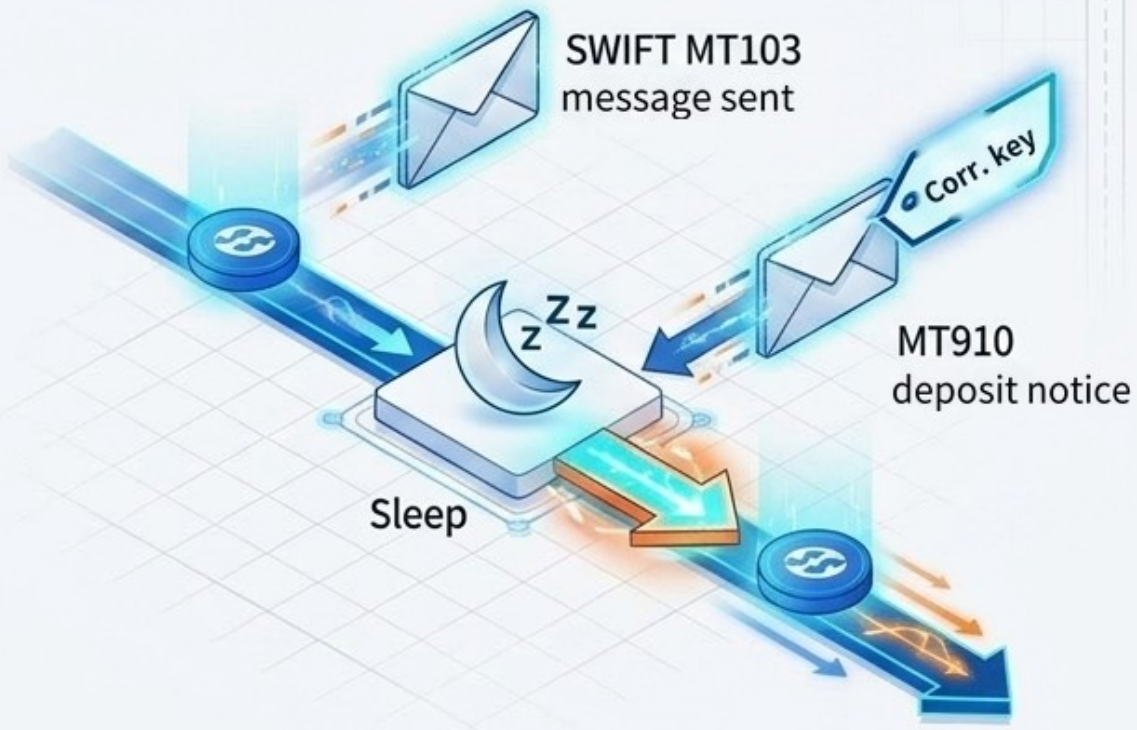
Complex Work Control: Parallel Review (Multi-Instance) and Events

The process branches once per reviewer, and the engine waits asynchronously for deposit notices or SWIFT messages, with no polling.

Multi-Instance (parallel branches)



Event-Driven (async wait)



Measurable Expansion: A 3-Phase Roadmap

We don't change the whole company at once.

Following 'you can't improve what you can't measure',
we expand safely, guided by metrics.



Phase 2 (6–9 months) – Scale Out

Complex routing (export bills, corporate cards),
parallel committee review (Multi-Instance), SLAs.

Phase 1 (3–4 months) – Validate

Simple, structured work (new accounts, balance notices).
Baseline for SQL/DMN integration.

Phase 3 (9+ months) – Advance

Core lending, automatic rollbacks,
full RPA integration, end-to-end mapping.

Phase 2 (6–9 months) – Scale Out

Complex routing (export bills, corporate cards),
parallel committee review (Multi-Instance), SLAs.

Solution

—

uEngine6
Key Features

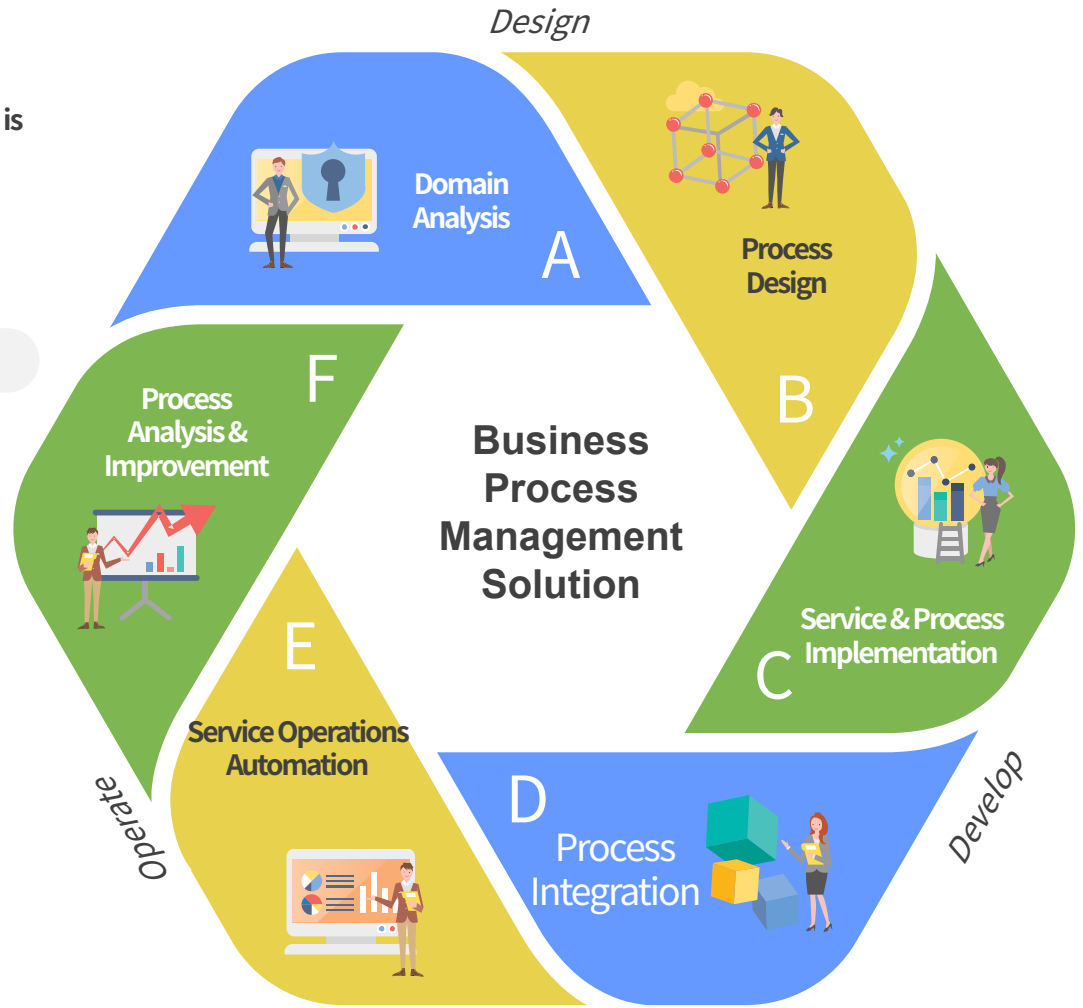
uEngine

Dedicated BPM Solution – uEngine BPM6

From architecture analysis to operations automation, uEngine BPM is

the platform built for SOA MM Level 7

http://github.com/uengine-oss/



- Compliant with the BPMN 2.0 industry standard
- uEngine6 BPM is a business process management system (BPMS) built on a modern cloud-native architecture
- Event-driven architecture (EDA) minimizes dependencies between services
- LLM-based AI automates process modeling, form generation and more from natural language

uEngine BPM Overview

SOA MM Level 7 | BPMN 2.0 | Process GPT | Event-Driven Architecture | Open Source (LGPL)

Next-Generation BPMS

uEngine6 BPM is a next-generation BPMS supporting BPM SOA MM Level 7, a complete solution for automating and optimizing business processes.

AI-Powered Process Automation

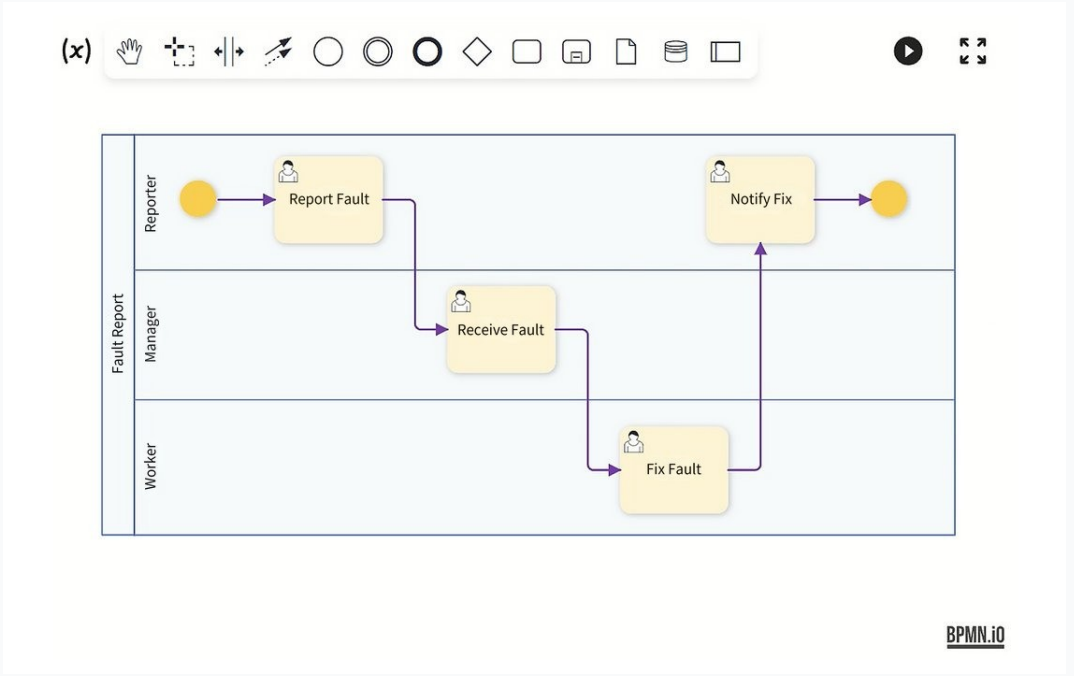
Process GPT enables prompt-based process modeling and automatic execution. LLM-based AI completes the model and suggests variables and API integrations.

Event-Driven Architecture (EDA)

Loosely coupled event integration connects services flexibly, with advanced features such as retries, compensation and multi-instance.

Full BPMN 2.0 Support

Fully supports the OMG BPMN 2.0 standard, with extensibility and compatibility through the OCE REST API.



Key Implementation Examples

Parallel approval by several approvers with multi-instance (MI)

External system integration through message events

Fully automated fault report → intake → repair workflow

Branching by fault type (software / hardware)

Key Feature 1: Multi-Instance Processing

Parallel and Sequential Loop Execution

Fully supports the BPMN 2.0 multi-instance marker, set declaratively in the UI. Inputs are split per instance and results aggregated automatically.

Implementation Highlights

Set MI on a subprocess to create multiple instances

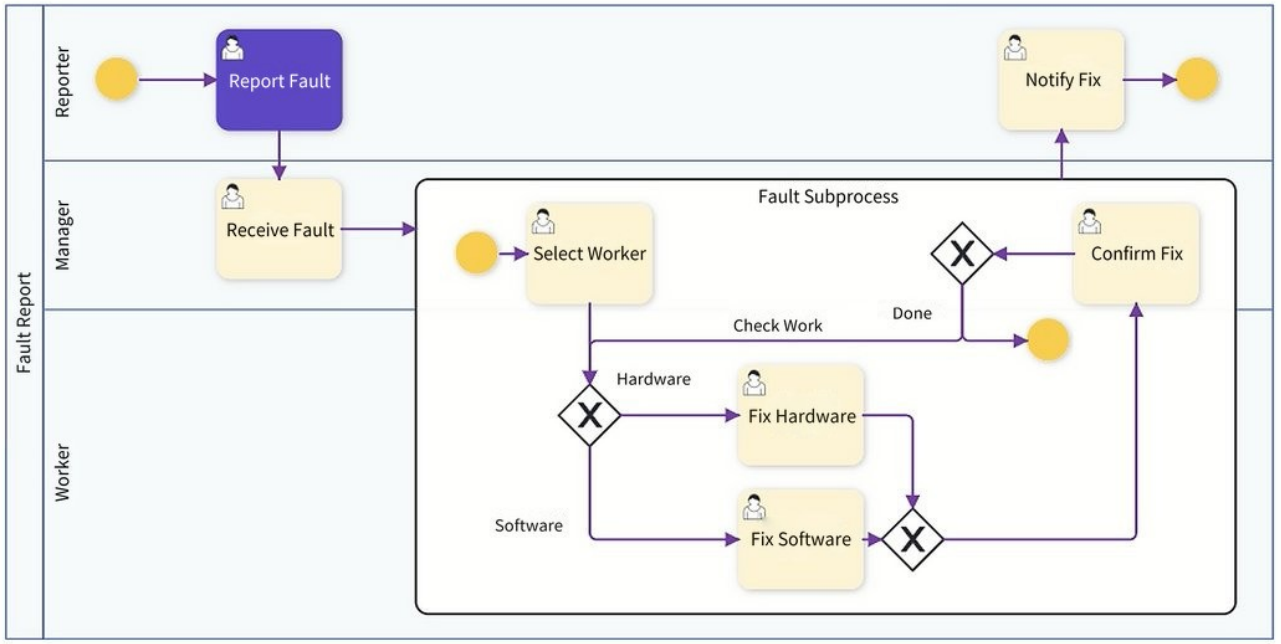
Handle many approvers at once with a Parallel Loop

Variables and state kept separately per instance

Results aggregated and merged back into the process

How It Works

Set a forEach variable on the subprocess and map a predefined collection; the instances are created automatically



Multi-instance subprocess in uEngine BPM

Parallel Execution

Instances run at the same time, cutting processing time

Sequential Execution

Instances run in order, for steps that depend on each other

Key Test Scenario 1: Multi-Instance Parallel Approval

Parallel Processing | Multi-Instance | Concurrent Work | Result Aggregation | Subprocess

Test Objective

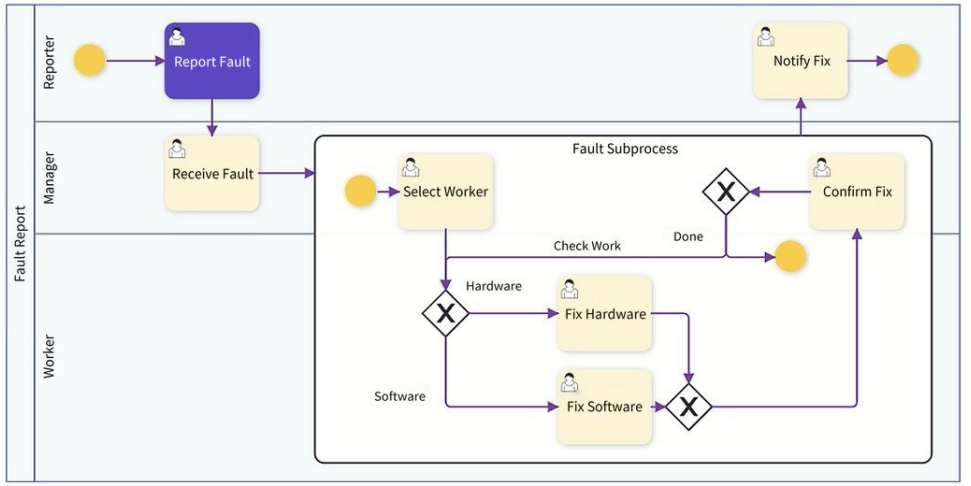
Measures how parallel review and approval performs, verifying parallelism and result consolidation when five reviewers work at once.

How uEngine Implements It

Set the fault variable as the subprocess 'forEach Variable' and map the 'multi data' field to enable multi-instance processing. It is configured declaratively in the UI, with no code to write.

Multi-Instance Implementation Steps

1. Define the variable to pass into the subprocess (e.g., approver)
2. Bind the approver list to the subprocess forEach variable
3. In the previous activity, map the multi-data field to the approver list
4. Results are aggregated automatically when all instances finish



Multi-Instance Features

Supports both parallel and sequential execution

Inputs split automatically per instance

Results from all performers aggregated and returned

Configurable completion condition (completionCondition)

Key Feature 2: Compensation Handling

Compensation for Long-Running Transactions

Define and run compensation events in long-running transactions. Explicit compensation triggers in the BPM model keep long business flows under control.

Compensation Highlights

Define rollbacks for long-running transactions

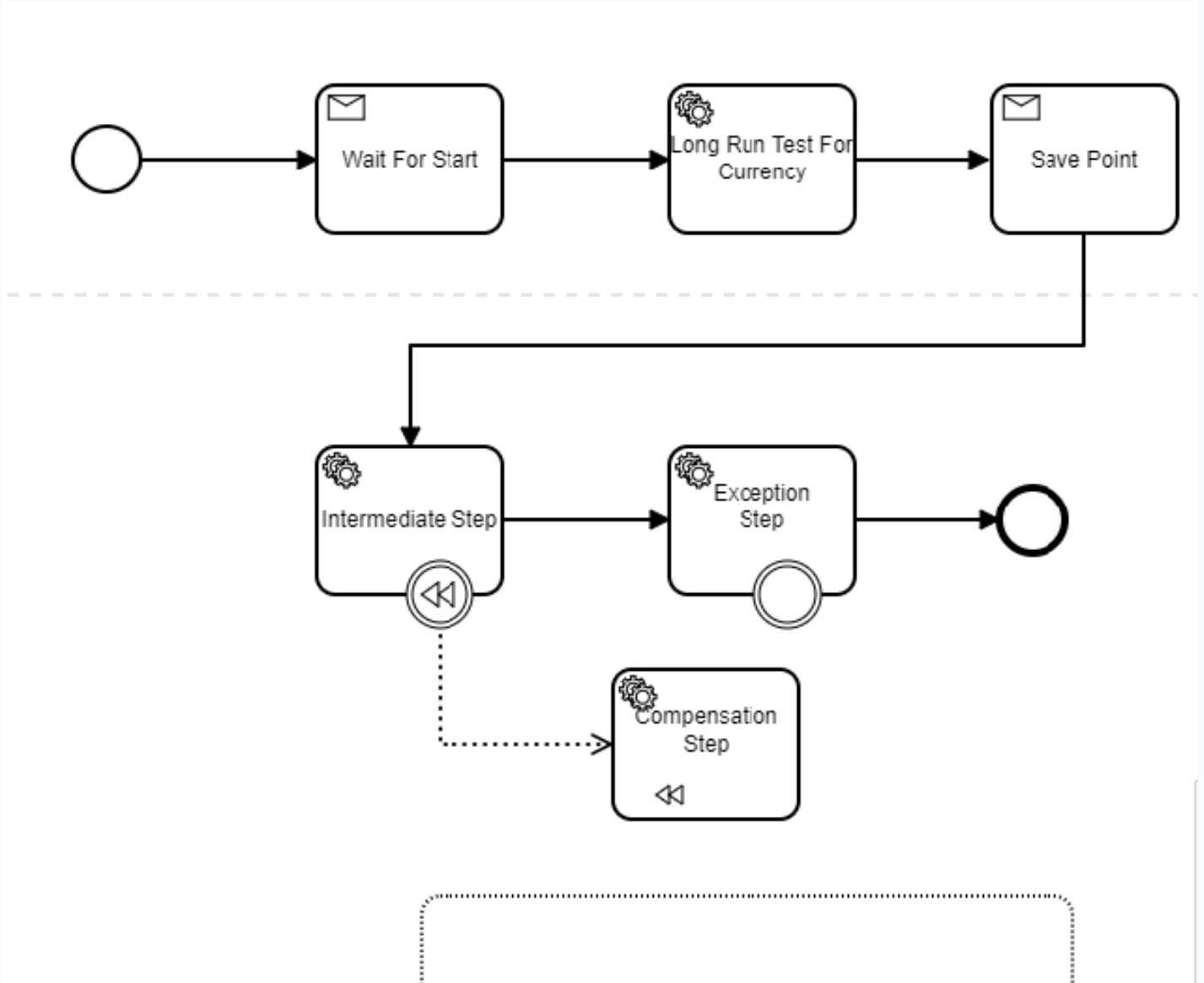
Compensation handlers shown clearly in BPMN

Nested compensation inside transaction boundaries

Automatic compensation on failure or timeout

How It Works

Attach a compensation event in the BPMN model and assign a handler; failures roll back automatically



Test Scenario 2: Refund Process with Compensation

BPMN Compensation | Long-Running Transaction | Compensation Flow | Cancel & Refund

Scenario Objective

Verifies that a process with cancellation and compensation logic runs correctly through a long flow from order and payment to cancellation and refund.

Compensation Flow Steps

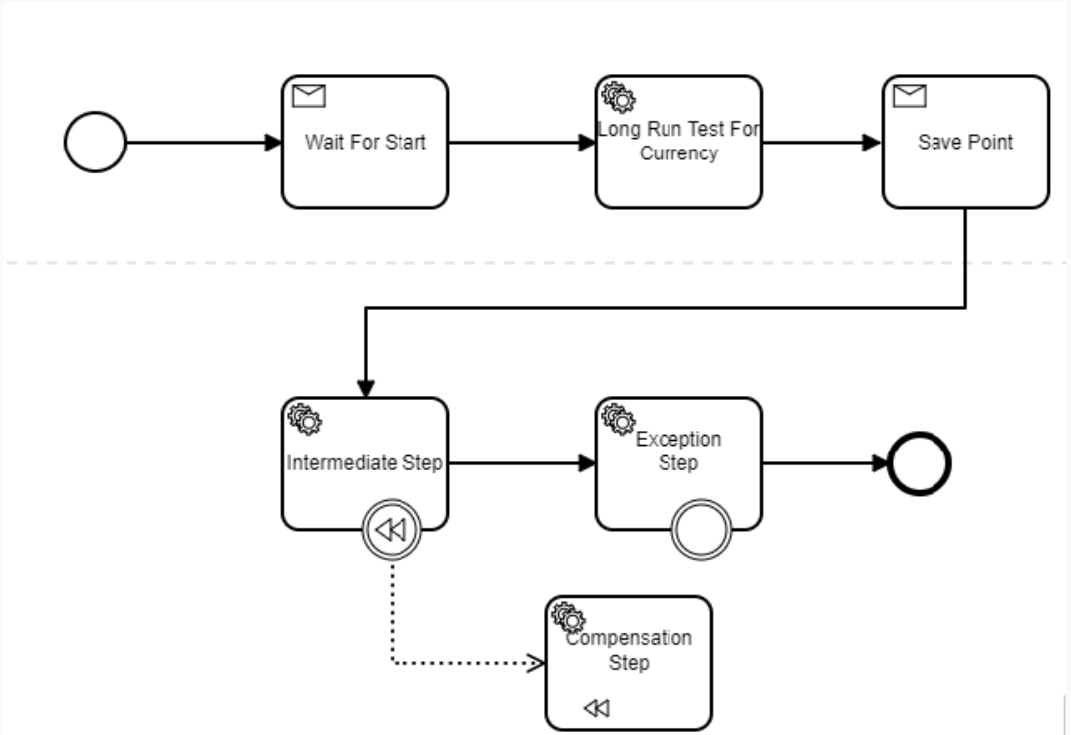
Order cancelled → refund triggered → compensation event → handler runs → related work rolled back → state restored. Business rules can apply at each step.

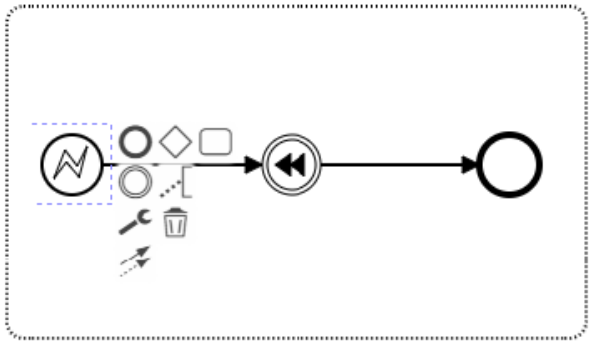
Long-Running Flow Control

Declare and run compensation events in long-running transactions, with rollback paths for long processes that cross transaction boundaries.

uEngine Advantage

Design compensation flows explicitly in BPMN and assign handlers. Even complex compensation logic from real requirements is built visually.





Key Feature 3: Event-Based Exception Handling

Event Handling for Exception Cases

Design exception scenarios with boundary and intermediate events. External triggers such as Kafka let the process listen for events and run alternative flows.

Event Types

Boundary Event – handles an event raised during a task

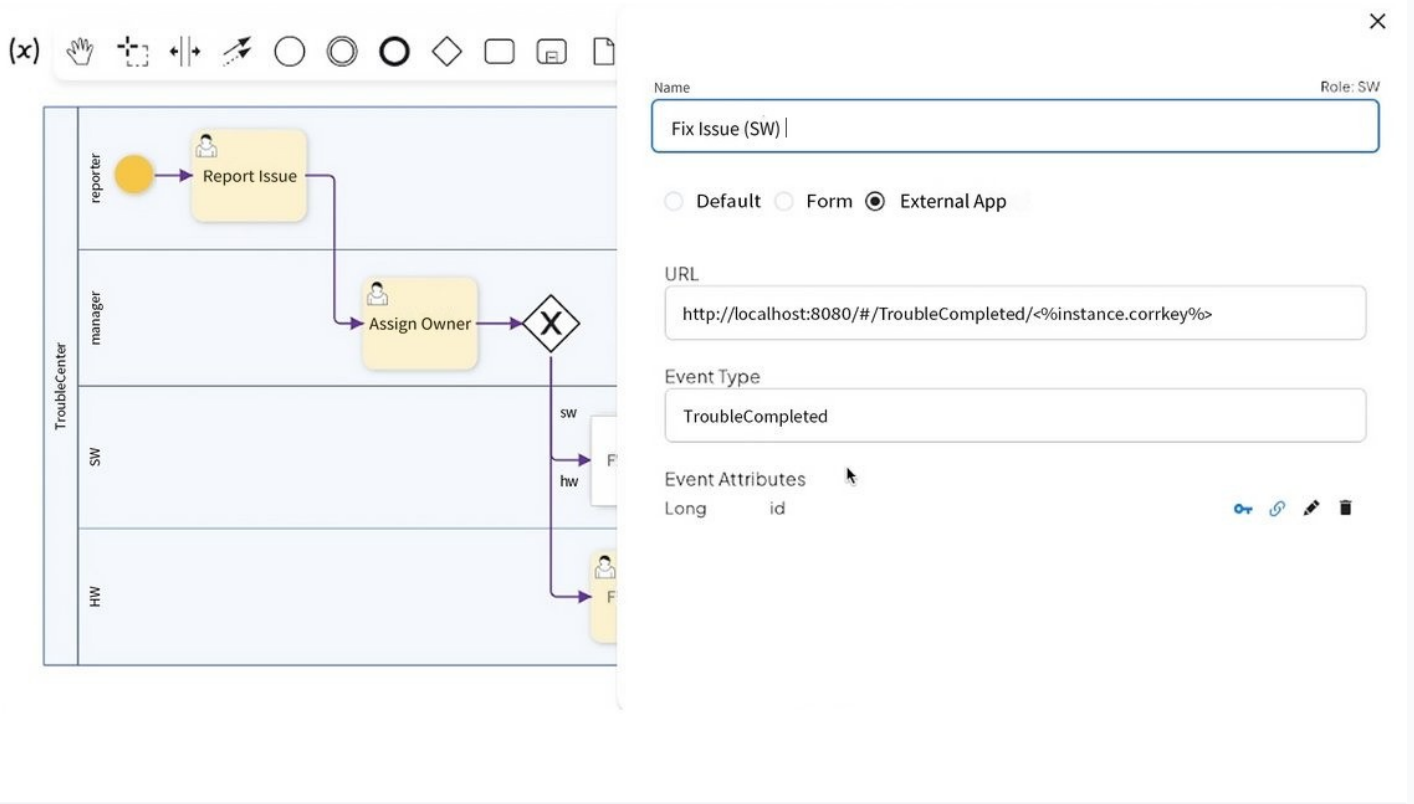
Timer Event – runs an alternative flow on timeout

Error Event – branches to an alternative flow on error

Message Event – handles messages from external systems

How It Works

Set the event type, URL and mapping to receive and trigger external events; the defined alternative flow runs automatically



Event-based exception settings in uEngine BPM

Interrupting Events

Stops the task and switches to the exception flow

Non-Interrupting Events

Task keeps running; exception flow runs in parallel

Test Scenario 3: Event-Based Exception Handling

Boundary Event | Intermediate Event | Timer Event | Exception Flow | Alternative Process

Scenario Objective

Verifies that an event trigger runs an alternative flow when a task times out or fails, with automatic alerts on timeout and compensation on error.

Timer-Based Exceptions

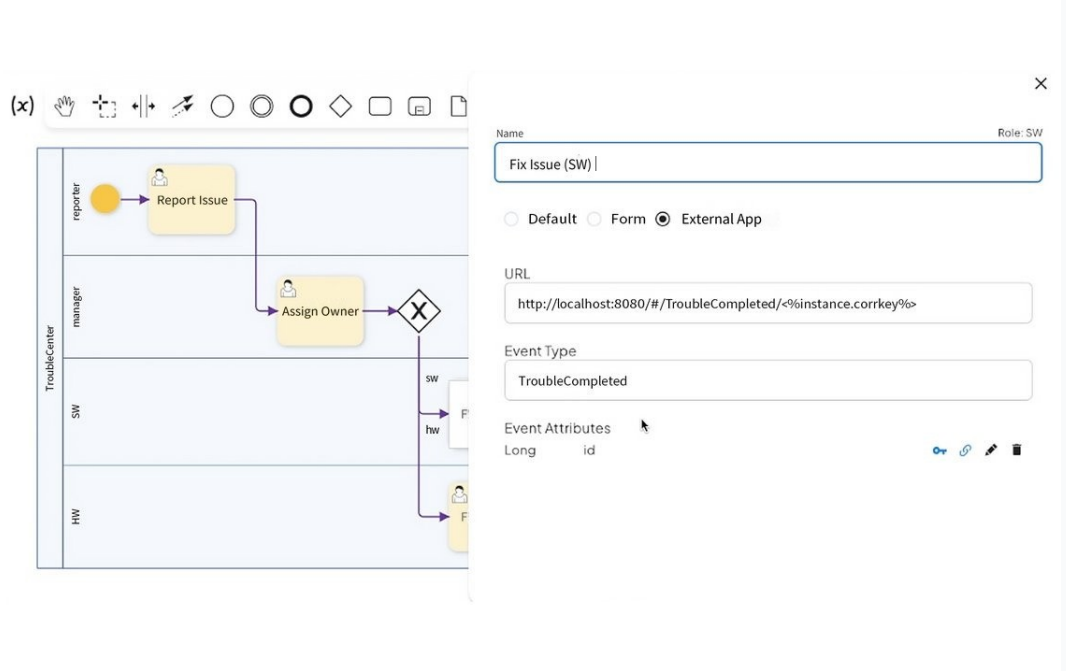
Attach a boundary timer event to a task to run an alternative flow if it is not done in time. uEngine models time-based alerts and escalations intuitively.

Event-Based Branching

With an event gateway, the process branches on whichever event comes first (message, signal, condition, etc.), fitting naturally into an event-driven architecture.

Business Use Cases

Handles critical real-world exceptions: auto-escalation on customer-service SLA breaches, payment time limits, approval expiry and external API timeouts.



Implementing Exception Handling

Define exceptions by adding a boundary event to a task

Set the event type (Timer, Error, Message, etc.)

Connect external systems through the event URL (webhook)

Define variable mappings and conditions for the exception flow

Key Feature 4: Loosely Coupled Event Integration

Loose Coupling Between Services

Event integration over brokers such as Kafka and RabbitMQ; incoming events can start a process or steer it.

Implementation Highlights

Full support for receiving and sending message events

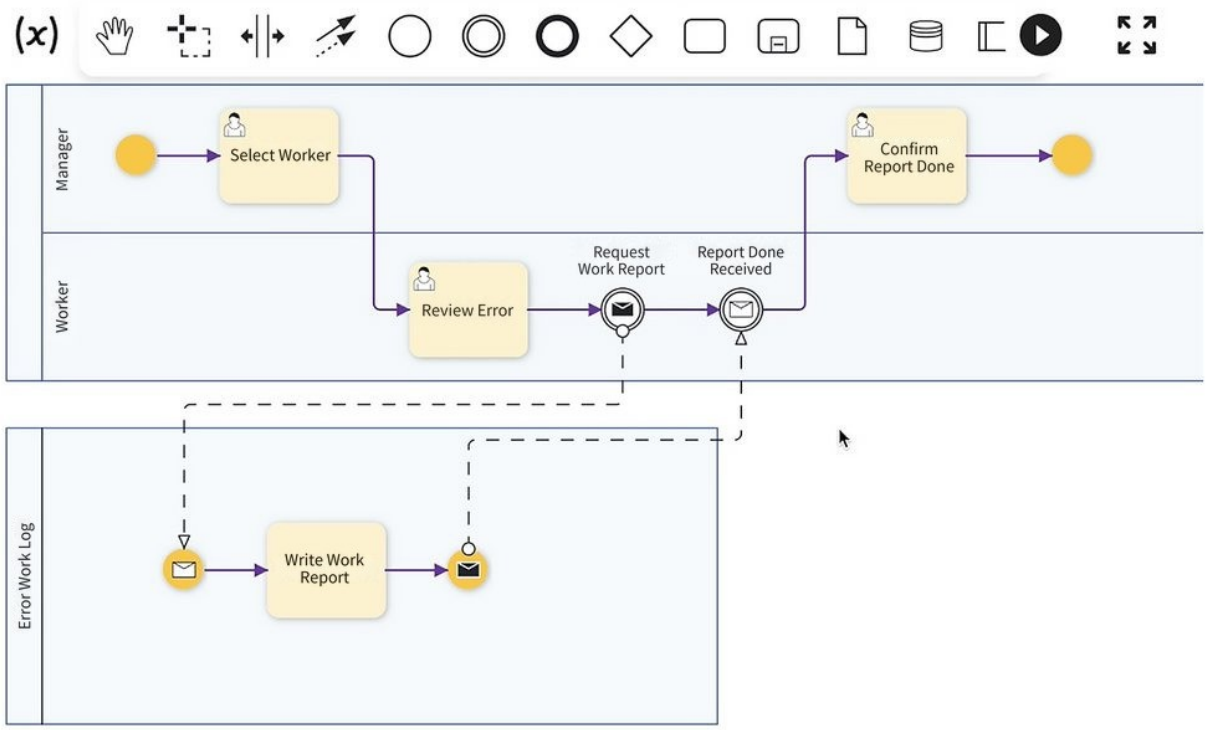
REST API integration with external systems

Event-driven process execution and control

Scalability through asynchronous messaging

How It Works

Services exchange messages through message start and intermediate events; external events can steer the flow mid-process



Message events in uEngine BPM

Event Publishing

Publishes events to external systems at set steps

Event Subscription

Subscribes to external events to start or advance processes

Test Scenario 4: Kafka-Based Loosely Coupled Integration

Kafka | RabbitMQ | Event-Driven Architecture | Async Messaging | Loose Coupling

Scenario Objective

Verifies that a Kafka event from an external system starts a BPM instance automatically and that events can steer the flow mid-process

Step 1: The external order system publishes an 'order event' to Kafka

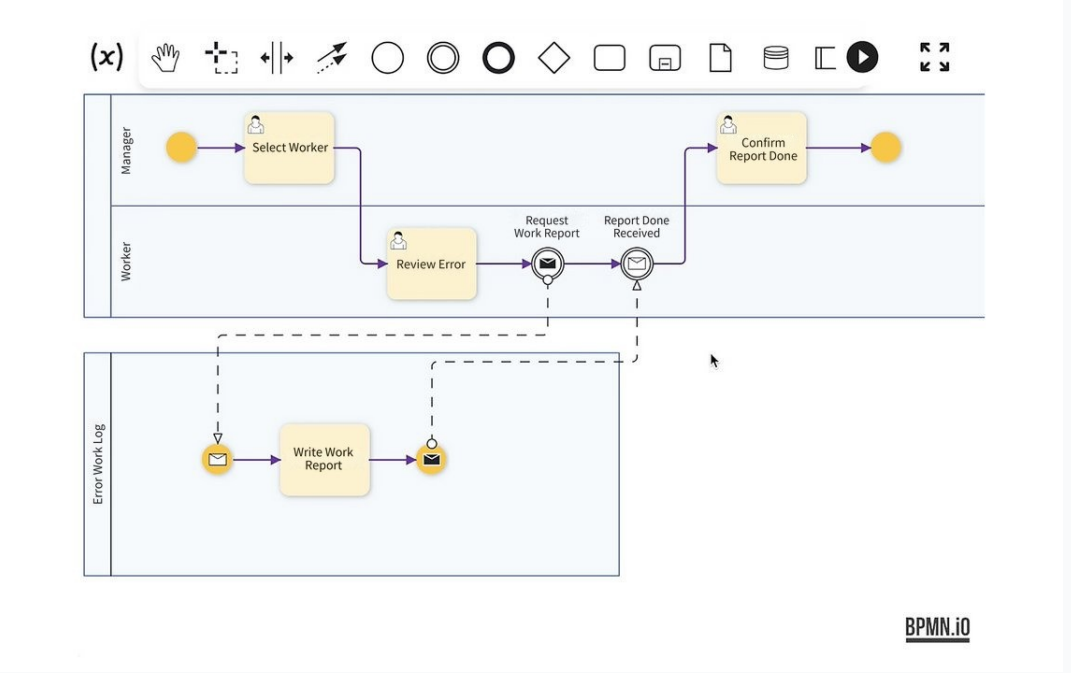
Step 2: uEngine BPM's Kafka connector receives it and starts an 'order processing' instance

Step 3: The flow branches on external events (payment done, shipped, etc.)

Step 4: On completion, a result event goes back to Kafka for other systems

Verification Points

Lower coupling between services through event integration, scalable and resilient execution, and asynchronous processing in containers



How uEngine BPM Implements It

Configure Kafka integration through message events

Link events by service path and correlation key

Bind a Kafka topic to the start event to auto-start

Control async messaging with catch/throw message events

Key Feature 7: Dynamic Forms and UI Integration

Dynamic UI for Each Process Step

uEngine BPM generates dynamic forms and input screens automatically. Binding work screens tightly to process logic maximizes development efficiency.

Implementation Highlights

Data Mapper links form data to process variables

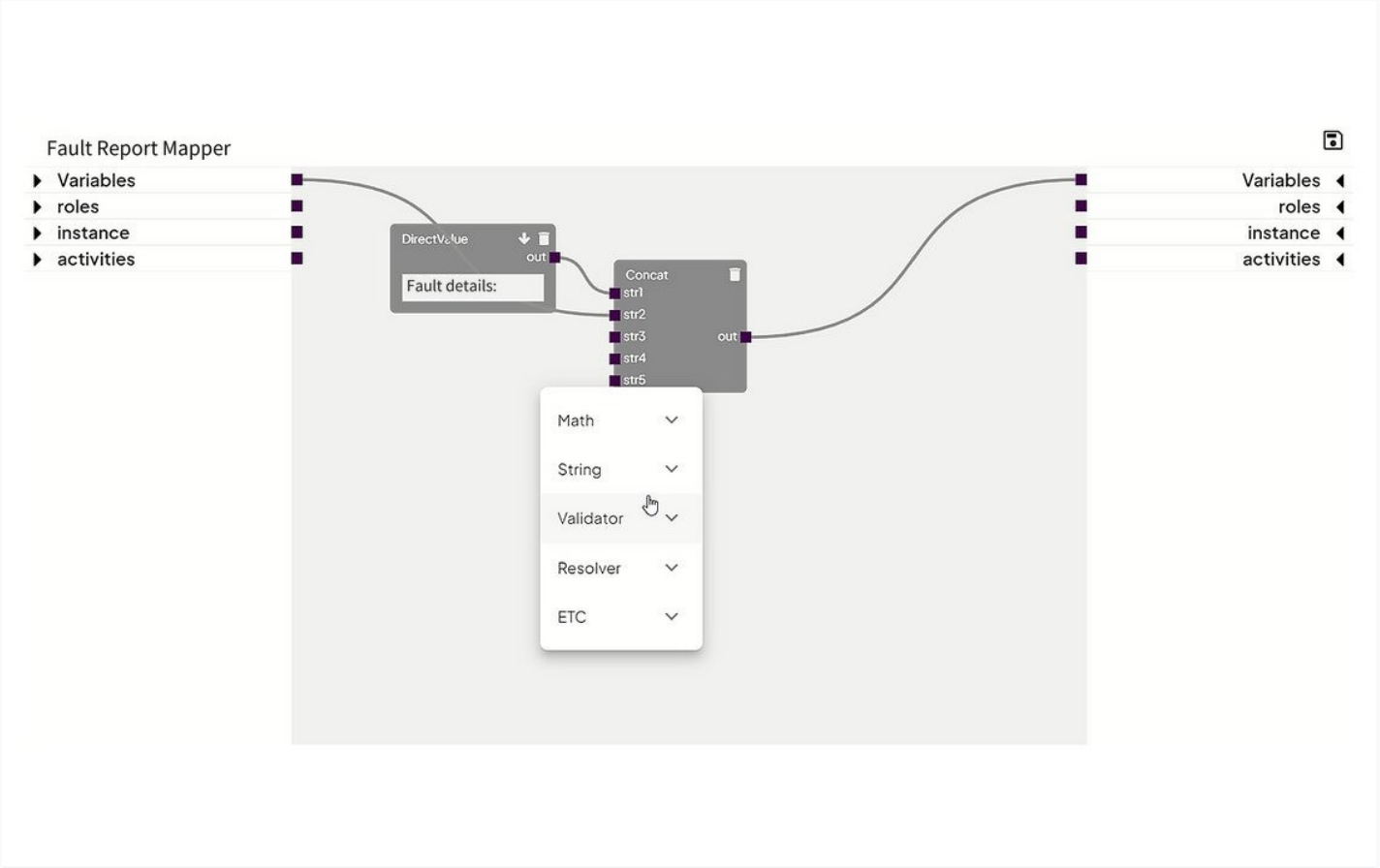
Form fields and styles defined declaratively in the UI

Data transformation and legacy system integration

A tailored form generated for each process step

How It Works

Add a form under process data, then use field mapping to connect transformation blocks such as DirectValue and Concat; the dynamic form is generated automatically



uEngine BPM Data Mapper – data mapping and transformation

Automatic Form Generation

Input forms built from process variables, cutting development time

Data Transformation

Converts input and system data automatically

Key Feature 8: Lightweight Operation on Kubernetes

Container-Based Scalability and Autoscaling

Deploys easily to Kubernetes with a Helm chart. Distributed multi-instance execution and workload-based autoscaling make it a natural fit for cloud-native environments.

Kubernetes Deployment Benefits

Fully integrated with microservice architecture

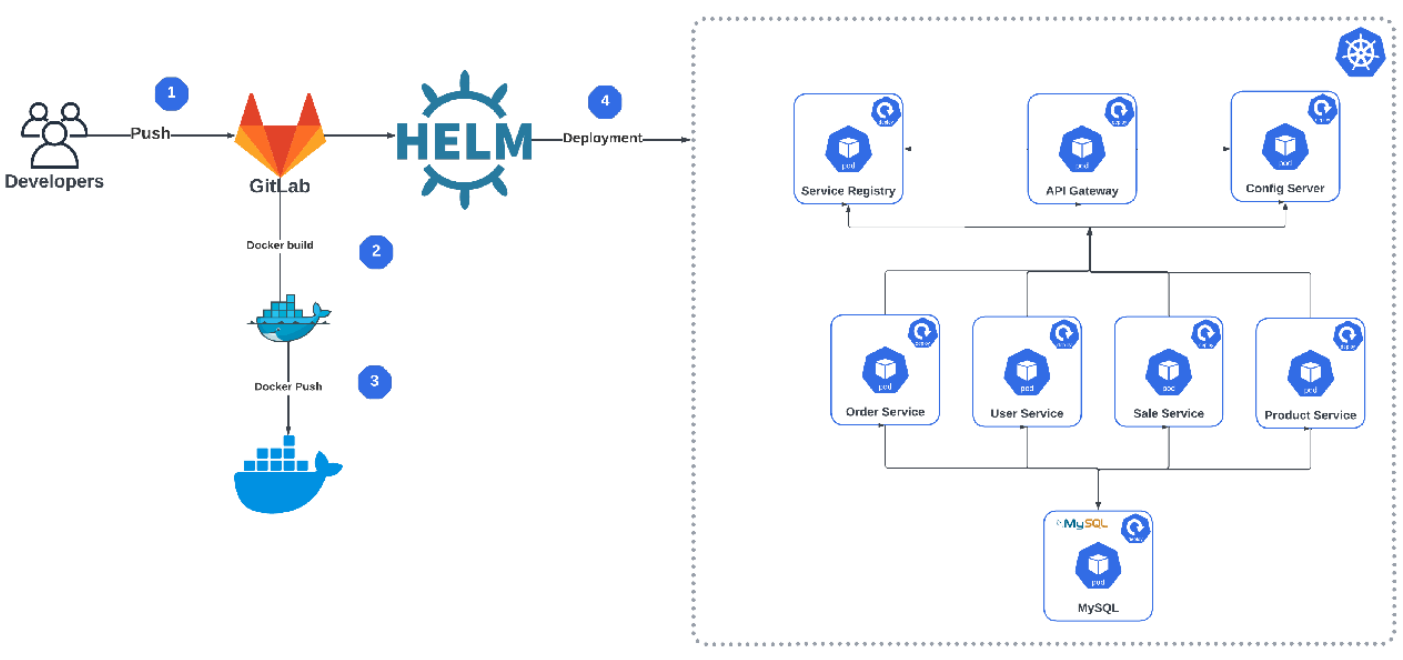
Workload-based autoscaling (HPA support)

Zero-downtime deploys and rollbacks

Efficient container resources (CPU/memory limits)

uEngine Helm Chart Deployment

Deploy to Kubernetes with one command: helm install uengine-bpm uengine/bpm-chart



uEngine BPM microservice architecture on Kubernetes

Orchestration

Automated deployment, scaling and recovery

Lightweight Containers

Tuned JVM settings for lean containers

DevOps Integration

Key Feature 9: Open Source, Easy to Customize

Fully Open Source, Extensible per Customer

uEngine BPM is fully open source (LGPL), so the source can be accessed and modified freely. Built on Spring Boot, it is easy to extend for each customer's business.

LGPL License Terms

Source disclosure obligations apply

Can link with proprietary software (commercial libraries)

Patent rights can be held and exercised

Source must be disclosed for redistributed derivatives

How to Customize

Spring Boot service components keep the core intact while business logic, UI and integrations are extended per customer

Open Source License Comparison

License	Source Disclosure	Copyright Notice	State Changes	Derivative Disclosure	Patent Rights	Proprietary Linking
GPL	O	O	O	O	O	X
LGPL	O	O	O	O	O	O
MPL	O	O	O	O	O	O
BSD Licence	O	O	O	O	X	O
Apache Licence	O	O	O	O	X	O

* Based on license information at bpm-intro.uengine.io

Spring Boot Extensibility

- Component-based microservice architecture
- Easy plug-in style feature additions
- Separate service layers for scalability

Per-Customer Customization

- Process-specific components
- UI themes and interface changes
- Connectors for legacy systems

Key Feature 10: Visualizing and Tracking Complex Processes

Intuitive BPMN-Based Visualization

Visualize complex processes with parallel flows, branches, events and exceptions in standard BPMN and track them in real time. The state of running processes is clear at a glance.

Real-Time Tracking

Visual execution history and current progress

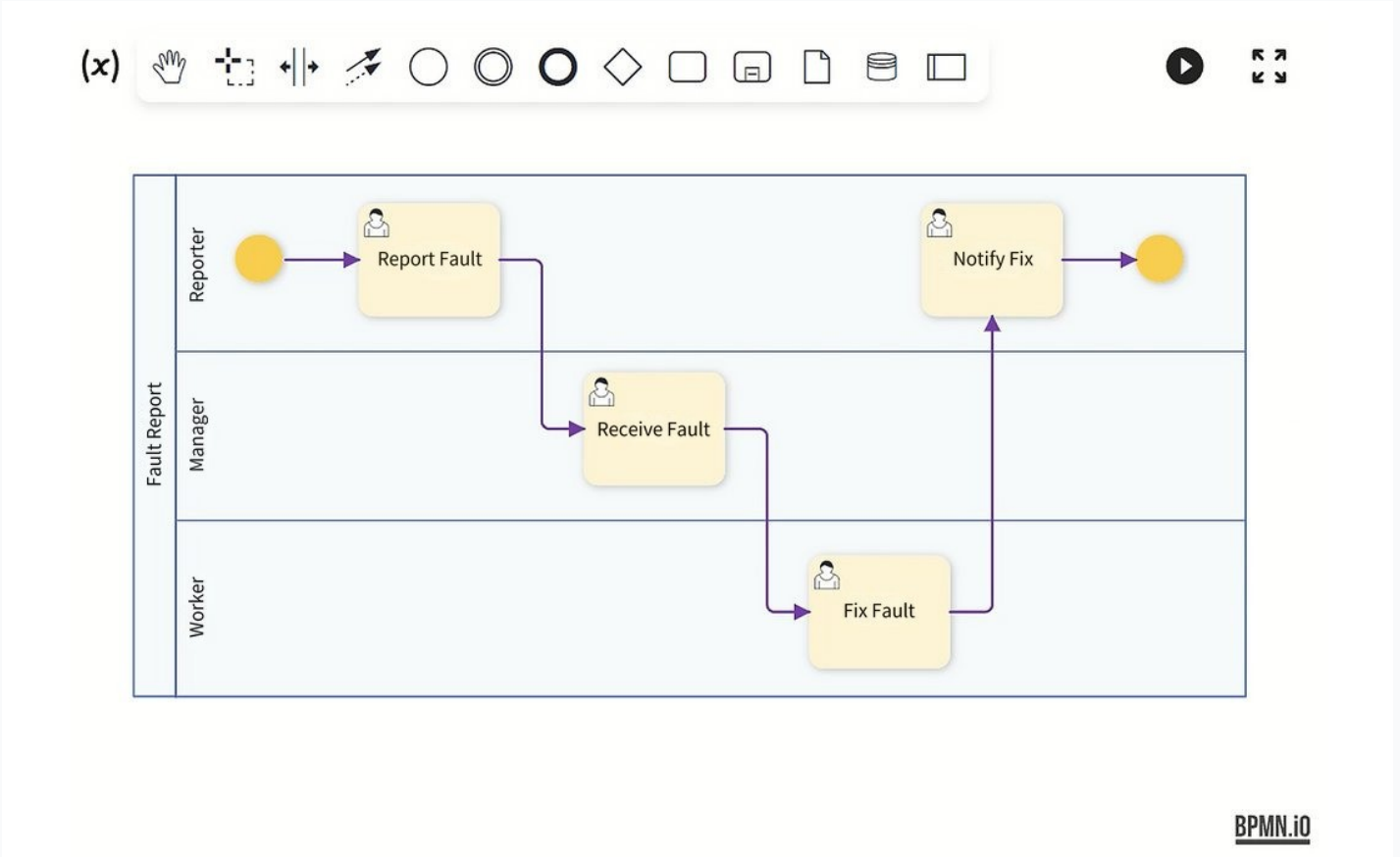
Find bottlenecks and trace issues in complex flows

SLA monitoring and timeline-based analysis

Trace execution paths and anticipate next steps

In Practice

Even in processes with many branches and parallel paths, users see the current state at a glance in the BPMN navigator and can trace its history.



Real-time BPMN process view in uEngine BPM

Real-Time Dashboard

Monitor instances, progress and bottlenecks visually

Execution History Analysis

Analyze paths and durations of completed processes

Test Scenarios 5–6: AI Modeling and Complex Process Visualization

Process GPT | LLM Prompts | Real-Time Views | Dashboard | Auto Model Generation

Scenario 5: Generate a Process from an AI Prompt

A simple prompt such as a travel expense request generates a complete BPMN model that runs right away. Process GPT extracts the variables and recommends suitable API integrations.

Scenario 6: Real-Time Visualization and Tracking of Complex Processes

Complex processes with parallel flows, conditional branches, events and exceptions are tracked live in the BPMN navigator and dashboard, with SLA analysis and timelines.

LLM-Driven Process Design

From a plain text description, Process GPT generates a complete BPMN 2.0 model. A request like "Create a travel expense request process" produces a runnable process.

Visualizing Complex Parallel and Conditional Flows

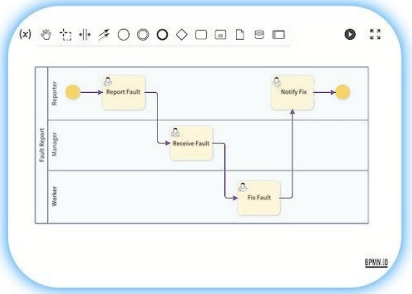
Flows with multi-instance, branching and event-based exceptions are visualized clearly and execution is tracked in real time.

Real-Time Monitoring and Dashboard

Monitor running instances live, with a dashboard for SLA analysis, bottleneck

uEngine6 BPM

01/ BPM Process Basics
02/ Subprocesses
03/ Conditional Flows
04/ Form Mapping
05/ Event-Based Integration
06/ Message-Based Integration



Test Scenario Results

A runnable BPMN model from a prompt in just 30 seconds

AI finds the variables and form fields and builds the input form

Color-coded execution status makes complex processes easy to read

Live per-instance progress, even at large parallel scale

Company

—

About Us

UEngine

uEngine Solutions

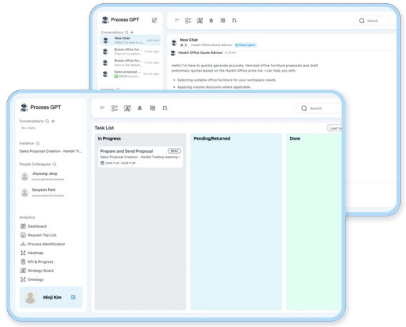
With 19 years of building and supplying BPM, SNS, Cloud and Agentic BPM solutions, and deployments in manufacturing and finance, uEngine Solutions is

a technology company that delivers BPM services successfully.

History	• Founded March 20, 2007
Key Customers	• Hanwha Group, IBK Industrial Bank, Samsung Electro-Mechanics, SKT, SK C&C, Hyundai Kia Motors, Woori Bank, Hana Bank, Daewoo E&C, KITECH and more
Business Areas and Technologies	• Cloud computing / SOA consulting and systems; SaaS platform research with ETRI (2010) • Runs global open-source communities (uEngine.org, Open Cloud Engine) • BPM (business process management), BRE (business rule engine) • Enterprise social-network collaboration system • Agentic BPM – Process GPT GS (Good Software) certified • Ontology-based graph DB and data intelligence platform • Legacy code / DB analysis; spec-driven AI design and build automation

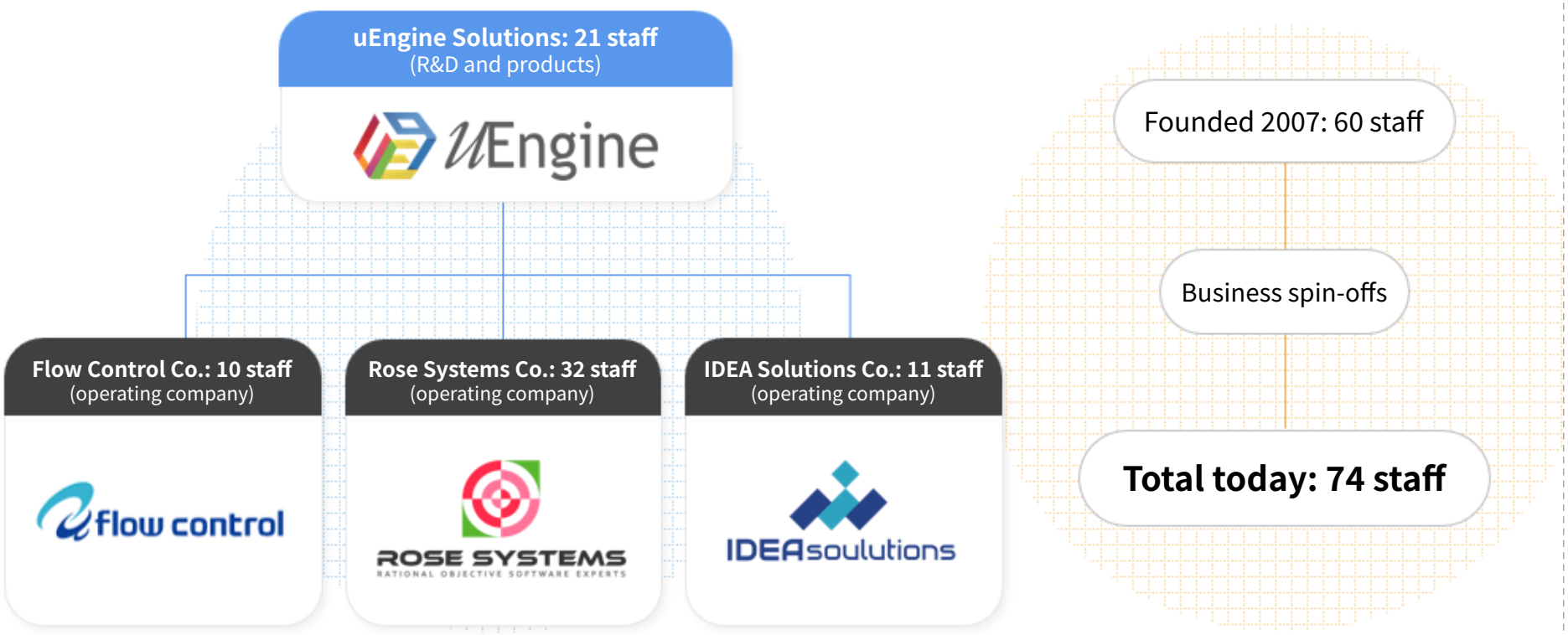
Related Solutions

uEngine6 BPM + = Process GPT



- Process GPT – an agentic process orchestrator that adds AI (LLM) to uEngine6 BPM
- One sentence creates a process
- Learns as it runs
- People and AI work together

Our Specialists



Since its founding in 2007, **uEngine Solutions** and its spin-offs —
four companies, 74 people in total — handle **R&D, product supply and maintenance**.

Technical Support and Training

Course	Process Management Fundamentals	Advanced Process Modeling	Operator Course	Developer Course (Customization)
Audience	Operations team, dev team (on request)	Operations team, Dev team (on request)	Operations team, dev team (on request)	Dev team (on request)
Topics	• Agile and cloud native • Creating and running processes • Process management • Workflow development • Process analysis	• Complex flow control: subprocesses and multiple instances • Event handlers • Building flows - External integration and scripting - Database integration • Process simulation	• Model management Version control Multi-resource management • Instance management: force stop, recovery, variables, reassignment • Organization management - Role picker, group/role management • Process operations	• Form application management • Custom activity and work item handler development • Automatic app integration through AI-generated OpenAPI • Loosely coupled system integration with event-driven architecture
Prior knowledge	None	JavaScript, SQL	None	Spring boot, VueJS

Track Record: Large-Scale Transactions and Concurrent Users

Measured in production at financial institutions — three key metrics: concurrent users, throughput (TPS) and response time

Concurrent Sessions
approx. 3,100 users
Case ①: peak day of 1,404,949 transactions · 1,041 active users [derived]

34.69 TPS
Effective Business-Hour Throughput
[Derived] 9 hours, 80% concentration

11.021 ms
Average Response Time
Measured on the peak day

4.29 %
CPU Utilization
95.7% headroom · measured

How concurrent users are derived — Little's Law applied to measured volume (Case ①)

① **Peak-day volume**
1,404,949 txns
Measured by customer monitoring

▶

② **Effective throughput**
34.69 TPS
× 80% ÷ 32,400 s (9 hours)

▶

③ **Active users**
1,041 users
× (response 0.011 s + think 30 s)

▶

④ **Concurrent users**
approx. 3,100 users
× 3 (sessions : active = 3 : 1)

Case ① BPM AP, 4 active nodes

28,165,134 per month · 908,552 daily avg · 2,227,004 peak-day total · 31 consecutive days

Node	Daily avg	Daily peak	Monthly total
bpmSvr101	274,514	557,833	8,509,948
bpmSvr102	204,840	573,393	6,350,053
bpmSvr201	139,010	517,885	4,309,300
bpmSvr202	290,188	577,893	8,995,833

~3,100 concurrent users 34.69 TPS · response 11.021 ms · CPU 4.29%

Case ② 8-core configuration

Business processes at a major commercial bank · derived on 20 business days a month

Item	Value	Note
Completed instances per month	170,301	Measured ('completed')
Avg tasks per instance	15	Measured
Tasks processed per month	2,554,515	[Derived] 170,301 × 15
Tasks per core per month	**319,314**	[Derived] ÷ 8 cores

~280 concurrent users 127,726 tasks/day avg · 3.15 TPS · ~35 per core

※ Concurrent-user formula — effective throughput = daily transactions (tasks) × 80% ÷ 32,400 s (9 business hours); active users = effective throughput × (avg response time + 30 s think time); concurrent users = active users × 3 (sessions : active = 3 : 1). ※ With a 20 s think time Case ① is about 2,100 users; with 60 s, about 6,200. ※ Items marked [Derived] are calculated with this formula; everything else (transaction counts, response time, CPU utilization, instance counts) is measured by customer monitoring. ※ 'Transactions' in Case ① and 'tasks' in Case ② are counted differently, so the two cases cannot be compared directly.

Maintenance Service Policy

○ : online, ◎ : online/on-site

Category	Warranty Service Item		Warranty Service Level		
			Silver	Gold	Platinum
Product Support	Installation & Enhancements	Installation support	◎	◎	◎
		Major enhancements	N/A	N/A	N/A
		Minor enhancements	○	○	◎
	Fixes & Updates	Patches/Hotfixes	○	○	◎
	Open-source SW license warranty		○	○	○
Maintenance	Basic maintenance		○	○	○
	Emergency incident support		○	◎	◎
	Training		-	◎	◎
	Performance tuning		-	-	◎
Service Fee	Annual maintenance		15% of license fee		
Custom Consulting	Architecture redesign		By agreement		
	Other professional services		KRW 1,000,000/day		

〈 Emergency Support Service Levels 〉

Warranty Service Item	Warranty Service Level	Support Hours	Response Time (business hours)	Support Visits
Emergency incident support	Silver	8h × 5 days/week	Within 24 hours	Online: unlimited
	Gold	24h × 5 days/week	Within 8 hours	Online: unlimited On-site: up to 40 hours/year
	Platinum	24h × 5 days/week	Within 4 hours	Online: unlimited On-site: 1 visit/month

Key References and Success Stories

uEngine BPM deployments across finance, insurance, public sector, manufacturing/construction, IT and telecom

6 Industries
Finance · Insurance · Public · Manufacturing · IT · Telecom

18+ Years
Building and deploying BPM

10,000 Users
Largest single-site deployment

Finance

한국증권금융 Korea Securities Finance Corp.
KDB산은금융그룹 대우증권
우리에프아이에스
참! 좋은 은행 IBK기업은행
하나금융그룹
KCB
코람코자산신탁

Insurance

대한생명
NH농협생명
PCA LIFE

Public Sector

ETRI
산림청 KOREA FOREST SERVICE
식품의약품안전처
KITECH
NTS 국세청
ifez
KORES

Manufacturing / Construction / Logistics

LG Energy Solution
LG디스플레이
LG화학
SAMSUNG 삼성전기
KIA
HYUNDAI
대우건설
경동나비엔

IT

Hanwha
SK C&C
posco ICT
Union&EC

Telecom

SK telecom
LG U+

Success Stories

Finance

IBK Industrial Bank
POST next-generation BPM

Scope
Core banking (deposits and other), BizHUB consultative sales, credit cards — bank-wide BPM

Period / Scale
EAI links across core banking, cards, Biz-Hub and subsystems; architecture principles set across many systems

Results
- Process-centric system architecture
- Easier, more accessible process changes
- Faster development by separating process and business logic

Insurance

Hanwha Life
(formerly Korea Life)

Next-gen loan BPM underway (2026.03~)

Scope
SR (IT projects), one-stop branch open/close/move, new products, approvals (BPM · GW · Portal · EAM · EDM · SSO · DRM)

Period / Scale
Built 2005.09–2006.04, upgraded 2012 · 10,000 users, 3,000 instances a day

Results
- One unified, worklist-based work environment
- Handles ad hoc, dynamic-parallel and event-branching processes
- More flexible and scalable than foreign vendors; still in use

Public

KITECH
Next-gen enterprise information system

Scope
Next-gen enterprise information portal (EIP), BPM-based e-approval, standard management processes

Period / Scale
2010.12–2011.08 · 20,000+ approvals a month, 6.53 million menu uses in total

Results
- More efficient public research projects
- Know-how from variable processes kept as assets
- Users up 27.1% (1,851; 2014 vs. 2012)

Manufacturing

Samsung Electro-Mechanics
Next-gen anomaly handling system

Scope
Next-gen anomaly handling system, BPM-based production process automation, standard management processes

Period / Scale
2015.12–2016.03 · anomaly processes in production control

Results
- More efficient anomaly handling with BPM
- Variable processes automated and kept as assets
- Faster collaboration between system stakeholders

Incident Response Manual

1. Severity Levels and Recovery Time Objectives (RTO)

Severity	Definition	Typical Incidents	Target Recovery Time
Sev 1 (Critical)	No user can do process-based work	Engine or process DB failure blocks all BPM-based work	Within 4 hours
Sev 2 (Major)	Delays or errors in some core processes	A workflow stops; external integration fails	Within 8 hours
Sev 3 (Minor)	Errors limited to a user or screen	Form display error; one instance fails	Within 1 business day

2. Step-by-Step Response

1. Detect · Monitor

Engine, DB and link alarms; auto-detection and alerts

2. Log · Notify

Help desk logs the incident; severity set and shared

3. Find the Cause

Trace logs and instances; confirm the impact

4. Recover

Restart, roll back or hold; reprocess instances

5. Follow Up

Root-cause report; prevention measures logged

3. Escalation Rules

- Sev 1: alert operations → support channel → uEngine support at once
- Sev 2: dev team joins if unresolved in 8 hours; handled as Sev 1 after that
- Sev 3: handled routinely, included in the weekly incident report

4. Prevention

- Threshold alerts on instance backlog and integration latency
- Monthly preventive maintenance (engine restart, data recovery drills)
- Incident history database; prevention measures checked

Summary and Conclusion

uEngine BPM's Key Competitive Advantages

- **Full BPMN 2.0 support** — a complete OMG BPMN 2.0 implementation ensures compatibility and portability
- **Strong multi-instance and compensation** — reliable control of complex parallel work and long-running flows
- **AI plus event-driven architecture** — intelligent automation with Process GPT and Kafka-based events
- **Kubernetes-ready and open source** — cloud-native support and freedom to customize under the LGPL

Benefits

- Faster work processing
- Higher automation
- Process transparency
- Lower operating costs

Key Differentiators vs. Other BPMS

Multi-instance: declarative parallel/sequential processing

Compensation: reliable rollback for long-running transactions

AI modeling: complex processes designed from a text prompt

Kubernetes: scalable, highly available cloud deployment

Support and Sales Inquiries

bpm-intro.uengine.io

info@uengine.org

+82-2-567-8301

Technical consultation

The right BPMS for your business — partner with uEngine.

© 2025 uEngine | bpm-intro.uengine.io