Camunda Evolution: Versions 7 vs 8

From Embedded Workflow Engines to Cloud-Native Orchestration

- **A Decade of Transformation:** Camunda 7 provided robust embedded orchestration for traditional enterprise systems, but the shift toward distributed, cloud-native, and AI-driven workloads demanded a full rearchitecture.
- **Emergence of Camunda 8:** Camunda 8 redefines workflow automation with Zeebe, a scalable and event-driven process engine, purpose-built for Kubernetes and modern hybrid deployments.
- **Vision for Future Orchestration:** The evolution focuses on agentic orchestration, built-in AI connectors, and horizontally scalable execution—enabling intelligent, real-time automation across microservices ecosystems.

Why Two Platforms? – Evolution Drivers

The Rationale Behind Camunda’s Platform Split

- **Evolving Market Needs:** Organizations demanded real-time orchestration, AI integration, and scalability that traditional embedded architectures could not sustain.
- **Hybrid Workflow Scenarios:** Enterprises required unified orchestration across cloud, on-premise, and hybrid deployments to support distributed business processes.
- **Architectural Reimagination:** Camunda 8 was redesigned from the ground up with Zeebe to deliver cloud-native orchestration, horizontal scalability, and event-driven resilience.
- **Future-Ready Automation:** Agentic orchestration and intelligent connectors prepare Camunda 8 for integration with AI-driven decision engines and modern microservice ecosystems.

Key Architectural Differences – Engine Model

From Embedded Libraries to Remote Microservices

Camunda 7: Embedded Engine
Engine operates as a library inside the application JVM—sharing threads, transactions, and lifecycle tightly with the host app. Scaling requires deploying multiple application-engine bundles.

Camunda 8: Remote Engine (Zeebe)
Zeebe runs as an independent service communicating via gRPC or REST APIs, enabling decoupled scaling, distributed orchestration, and microservice-aligned deployments.

Deployment Flexibility
Camunda 7 thrives in monolithic or Spring Boot environments, while Camunda 8 is purpose-built for Docker, Kubernetes, and hybrid cloud setups.

Architectural Impact
The shift from embedded to remote orchestration marks the transition from traditional app-centric models to cloud-native distributed system thinking.

Data Storage & Transaction Management

From ACID Transactions to Eventual Consistency

Camunda 7 – Relational & Transactional
Uses relational databases (PostgreSQL, Oracle, MySQL) and serialized Java objects for persistence. Shared ACID transactions ensure strict consistency between the engine and application.

Camunda 8 – Event Stream & JSON
Replaces RDBMS dependency with event streaming and JSON-based variable storage. Uses Elasticsearch for secondary persistence and analytics.

Consistency Model Shift
Camunda 8 adopts eventual consistency—eliminating shared transaction managers and requiring idempotent operations and compensation logic in process design.

Developer Implications
Developers must handle custom data mappings and design workflows resilient to asynchronous execution, ensuring reliability in distributed systems.

Comparison Matrix: Camunda 7 vs Camunda 8

Key Functional and Architectural Contrasts

- **Engine & Deployment:** Camunda 7 operates as an embedded library within the application (JVM-bound), while Camunda 8 uses Zeebe as a remote microservice, deployable via Docker or Kubernetes.
- **Data & Storage:** Camunda 7 relies on relational databases and serialized Java objects; Camunda 8 uses JSON variables and event streams with Elasticsearch for persistence.
- **Scalability & Performance:** Camunda 7 scales with application instances, while Camunda 8 supports independent scaling of brokers and workers, optimized for high throughput.
- **Monitoring & Tools:** Camunda 7 integrates Cockpit and Tasklist directly, whereas Camunda 8 separates these into modular services like Operate, Tasklist, and Optimize.
- **AI & Cloud Native Capabilities:** Camunda 7 offers limited custom AI integrations, while Camunda 8 introduces native AI connectors and is purpose-built for cloud environments.

When to Choose Camunda 7

Best Fit Scenarios for Legacy and Monolithic Deployments

- **Existing Deployments:** Ideal for organizations already invested in Camunda 7 with extensive process definitions and stable production workflows.
- **Monolithic & Java-Centric Systems:** Perfect for traditional monolithic architectures and Java-heavy environments leveraging embedded transactions and serialized objects.
- **Operational Simplicity:** Camunda 7 offers straightforward setup and embedded deployment through Spring Boot, making it accessible to teams with limited distributed systems expertise.
- **Cost & Stability:** Open-source self-hosting and mature support ecosystem make it cost-effective and low-risk for smaller or stable enterprises.

When to Choose Camunda 8

Modern, Scalable, and Cloud-Native Automation

- **Cloud-Native & Microservices Architectures:** Camunda 8 is purpose-built for containerized, distributed applications orchestrated through Kubernetes and cloud environments (AWS, Azure, GCP).
- **High-Volume & Real-Time Workloads:** Ideal for organizations requiring low-latency, high-throughput process execution with horizontally scalable workers and brokers.
- **AI-Driven Orchestration:** Native AI connectors and decisioning frameworks enable intelligent process execution and adaptive workflows.
- **Event-Driven Integration:** Supports modern event streams (Kafka, message queues) for asynchronous orchestration across microservices and SaaS platforms.
- **Multi-Tenant SaaS and Elastic Scaling:** Provides built-in tenant isolation and independent scaling of components, suited for SaaS and large-scale enterprise environments.

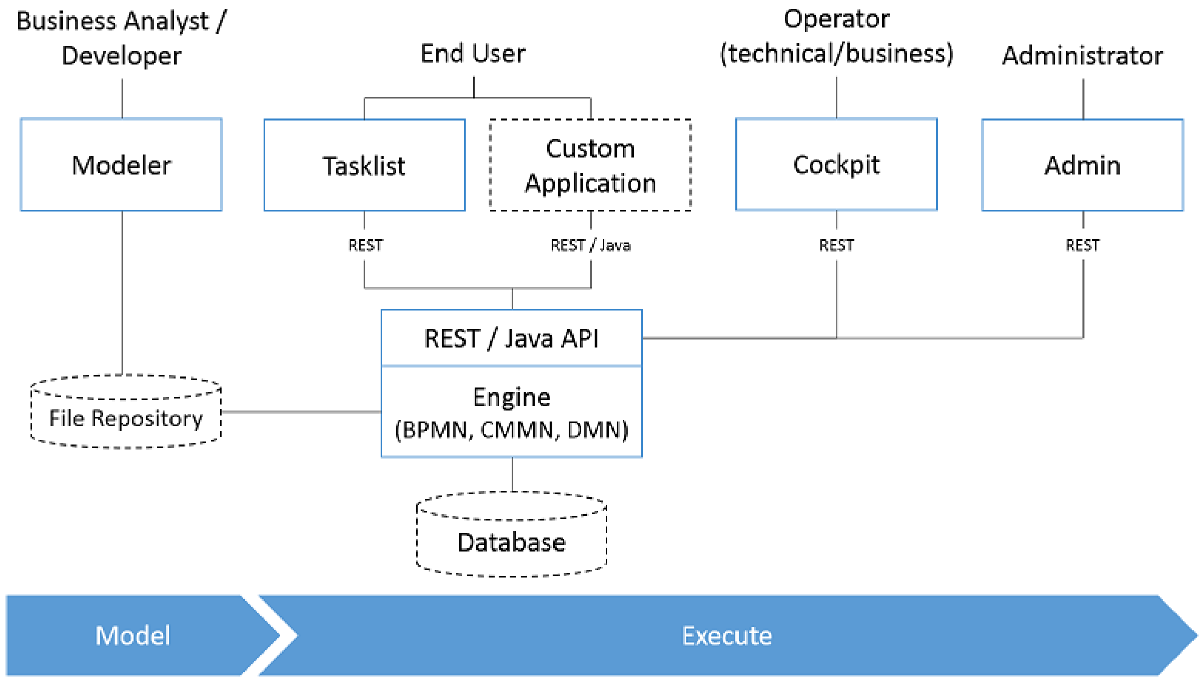
Migration Considerations

Transitioning Safely from Camunda 7 to Camunda 8

- **Not a Drop-in Upgrade:** Camunda 8 introduces a fundamentally new architecture. Migration requires process redesign, application refactoring, and testing for distributed execution.
- **Process Redesign & Refactoring:** Legacy BPMN models must be adapted to Zeebe's execution semantics, eliminating Java serialization and ensuring asynchronous task handling.
- **Data Transformation & Consistency:** Applications must handle custom data mapping and implement eventual consistency through compensation and idempotent design patterns.
- **Parallel Operation Strategy:** A side-by-side migration approach is recommended—running Camunda 7 and 8 concurrently while validating performance and reliability.
- **Validation & SLA Testing:** Comprehensive testing of throughput, latency, and fault tolerance ensures production readiness and alignment with SLAs.

Camunda 7 Setup & Components Overview

Deployment, Monitoring, and Administration



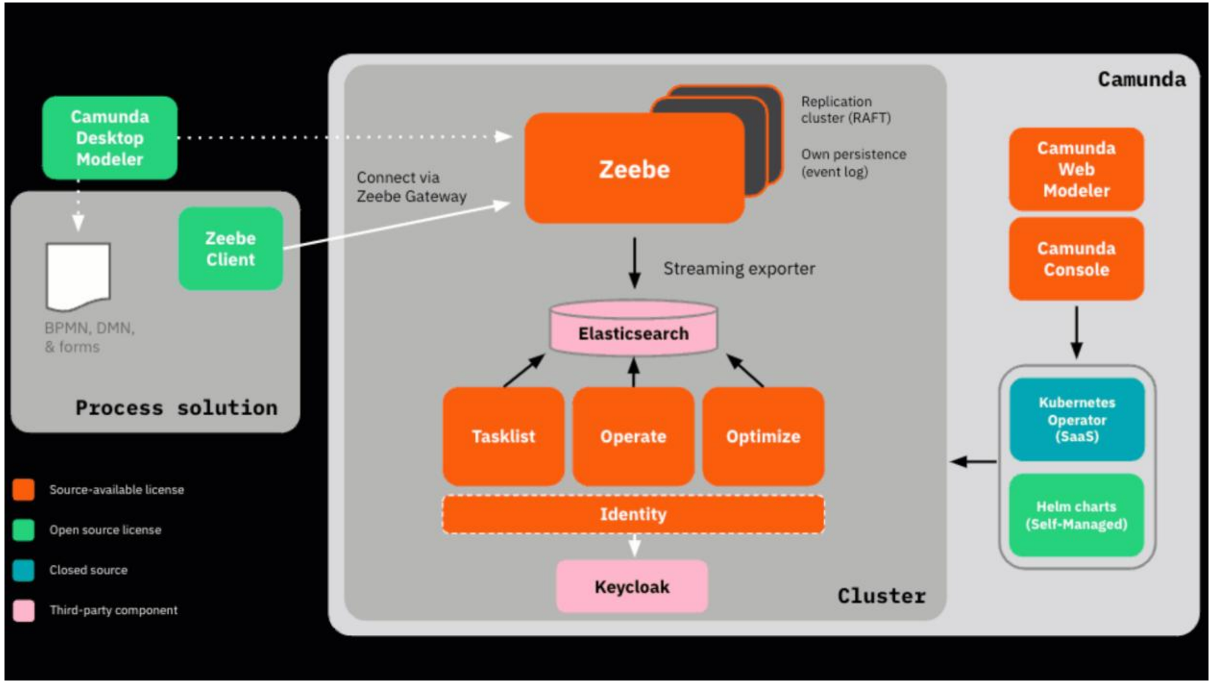
Camunda 7 Setup & Components Overview

Deployment, Monitoring, and Administration

- **Environment Requirements:** Requires Java 17+, Maven 3.6+, and 4GB RAM (8GB recommended). Typical deployment uses Spring Boot or Tomcat with embedded or external databases.
- **Core Applications:** Includes Cockpit for process monitoring, Tasklist for user task management, and Admin Console for user and access control.
- **Cockpit Capabilities:** Visualizes process instances, variables, and incidents with real-time monitoring and batch operations.
- **Tasklist & Human Workflows:** Supports task claiming, delegation, and completion through forms—ideal for user-centric workflows.
- **Admin Console:** Provides role-based access, tenant management, and system configuration for governance and compliance.

Camunda 8 Architecture Components

A Modular and Cloud-Native Orchestration Ecosystem



Camunda 8 Architecture Components

A Modular and Cloud-Native Orchestration Ecosystem

- **Core Components:** Includes Zeebe (workflow engine), Operate (monitoring), Tasklist (human task management), Optimize (analytics), Console (configuration), and Web Modeler (process design).
- **Infrastructure Dependencies:** Relies on Elasticsearch for secondary storage and analytics, PostgreSQL for metadata, and Keycloak for authentication and authorization.
- **Component Interoperability:** Each service communicates over APIs and event streams, ensuring loose coupling and independent scaling in cloud deployments.
- **Connector Ecosystem:** Extensible connectors framework enables seamless integration with external systems, APIs, and event sources like Kafka.
- **Identity & Access:** Identity service centralizes user management with OpenID Connect and Keycloak integration for secure enterprise authentication.

Camunda 8 Environment & Docker Setup

Quick Start with Docker Compose

- **System Requirements:** Docker 20.10+, Docker Compose 1.27+, 8GB RAM (16GB recommended), 10GB disk space, and active internet connection for image downloads.
- **Directory & Configuration:** Camunda 8 ships with lightweight and full-stack Docker Compose files—configurable via .env files for environment variables and secrets.
- **Startup Process:** Execute `docker compose up -d` to deploy Zeebe, Operate, and Tasklist. Initialization typically completes within 2–5 minutes.
- **Verification:** Validate container health with `docker compose ps` and logs for Zeebe, Operate, and Tasklist components.
- **Access Points:** Operate: http://localhost:8088/operate, Tasklist: http://localhost:8088/tasklist, REST API: http://localhost:8088/v2, gRPC: localhost:26500.

Zeebe Architecture Overview

The Core Engine of Camunda 8

- **Event-Sourced Design:** Zeebe operates on a distributed event log architecture ensuring durability, scalability, and replayable workflows.
- **Clustered Brokers & Partitions:** Brokers are distributed across partitions for parallel execution and replication, ensuring fault tolerance and high throughput.
- **Gateways & Clients:** Gateways manage communication between brokers and clients (Java, Go, Node.js, Python) via high-performance gRPC APIs.
- **Replication & Fault Tolerance:** Zeebe employs Raft consensus and partition replication to maintain state consistency and automatic failover.
- **Exporter Framework:** Event exporters connect Zeebe with external systems like Operate, Elasticsearch, and custom analytics pipelines.

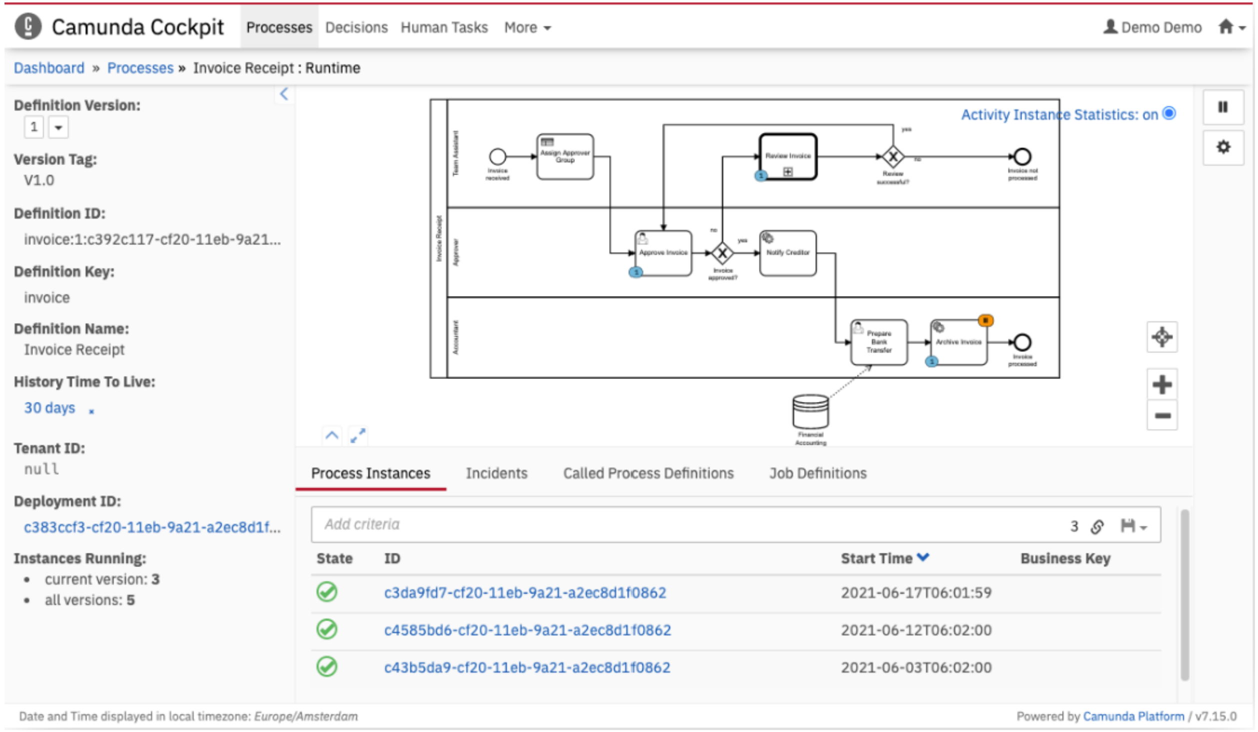
Zeebe Job Workers Explained

Executing Distributed Tasks in Camunda 8

- **Worker Role:** Job workers subscribe to specific job types, fetch tasks from Zeebe, execute business logic, and report results asynchronously.
- **Language Support:** SDKs are available for Java, Go, Node.js, Python, and C#, enabling integration across diverse technology stacks.
- **Resilience & Retry Logic:** Workers handle automatic retries and backoff strategies, maintaining system reliability during transient errors.
- **Variable Handling:** Workers read and update process variables, influencing subsequent workflow decisions and path executions.
- **Design Implications:** Encourages decoupled, stateless service designs that scale independently and align with event-driven microservice principles.

Operational Monitoring – Camunda 7 (Cockpit)

Process Instance Management and Incident Resolution



Dashboard » Processes » Invoice Receipt : Runtime

Definition Version: 1

Version Tag: V1.0

Definition ID: invoice:1:c392c117-cf20-11eb-9a21...

Definition Key: invoice

Definition Name: Invoice Receipt

History Time To Live: 30 days

Tenant ID: null

Deployment ID: c383ccf3-cf20-11eb-9a21-a2ec8d1f...

Instances Running:
- current version: **3**
- all versions: **5**

State	ID	Start Time	Business Key
✓	c3da9fd7-cf20-11eb-9a21-a2ec8d1f0862	2021-06-17T06:01:59	
✓	c4585bd6-cf20-11eb-9a21-a2ec8d1f0862	2021-06-12T06:02:00	
✓	c43b5da9-cf20-11eb-9a21-a2ec8d1f0862	2021-06-03T06:02:00	

Date and Time displayed in local timezone: *Europe/Amsterdam*

Powered by Camunda Platform / v7.15.0

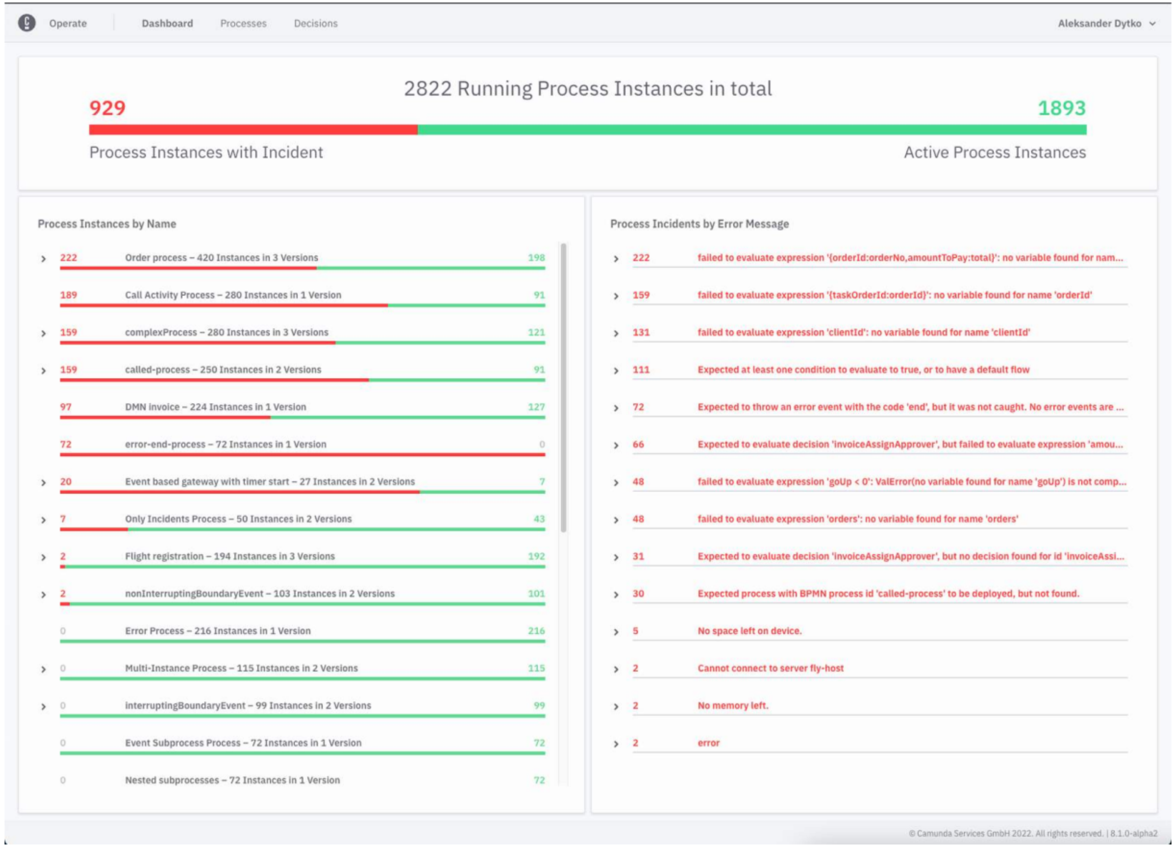
Operational Monitoring – Camunda 7 (Cockpit)

Process Instance Management and Incident Resolution

- **Process Instance Visibility:** Cockpit provides real-time insights into running, completed, and failed process instances with filtering by ID, state, or business key.
- **Execution Monitoring:** Users can trace process execution flow, inspect variables, and analyze activity timing for performance optimization.
- **Incident Management:** Cockpit highlights incidents such as failed jobs, exceptions, or missing external tasks, enabling direct resolution and retries.
- **Operational Controls:** Supports suspending, resuming, or canceling instances and modifying variables or activity states during runtime.
- **Audit & Compliance:** Provides a complete audit trail of process execution and user interventions for compliance tracking.

Operational Monitoring – Camunda 8 (Operate)

Real-Time Observability and Process Management



2822 Running Process Instances in total

929 Process Instances with Incident

1893 Active Process Instances

Process Instances by Name

222	Order process – 420 Instances in 3 Versions	198
189	Call Activity Process – 280 Instances in 1 Version	91
159	complexProcess – 280 Instances in 3 Versions	121
159	called-process – 250 Instances in 2 Versions	91
97	DMN invoice – 224 Instances in 1 Version	127
72	error-end-process – 72 Instances in 1 Version	0
20	Event based gateway with timer start – 27 Instances in 2 Versions	7
7	Only Incidents Process – 50 Instances in 2 Versions	43
2	Flight registration – 194 Instances in 3 Versions	192
2	nonInterruptingBoundaryEvent – 103 Instances in 2 Versions	101
0	Error Process – 216 Instances in 1 Version	216
0	Multi-Instance Process – 115 Instances in 2 Versions	115
0	interruptingBoundaryEvent – 99 Instances in 2 Versions	99
0	Event Subprocess Process – 72 Instances in 1 Version	72
0	Nested subprocesses – 72 Instances in 1 Version	72

Process Incidents by Error Message

222	failed to evaluate expression '{orderId:orderNo,amountToPay:total}': no variable found for nam...
159	failed to evaluate expression '{taskOrderId:orderId}': no variable found for name 'orderId'
131	failed to evaluate expression 'clientId': no variable found for name 'clientId'
111	Expected at least one condition to evaluate to true, or to have a default flow
72	Expected to throw an error event with the code 'end', but it was not caught. No error events are ...
66	Expected to evaluate decision 'invoiceAssignApprover', but failed to evaluate expression 'amou...
48	failed to evaluate expression 'goUp < 0': ValError(no variable found for name 'goUp') is not comp...
48	failed to evaluate expression 'orders': no variable found for name 'orders'
31	Expected to evaluate decision 'invoiceAssignApprover', but no decision found for id 'invoiceAssi...
30	Expected process with BPMN process id 'called-process' to be deployed, but not found.
5	No space left on device.
2	Cannot connect to server fly-host
2	No memory left.
2	error

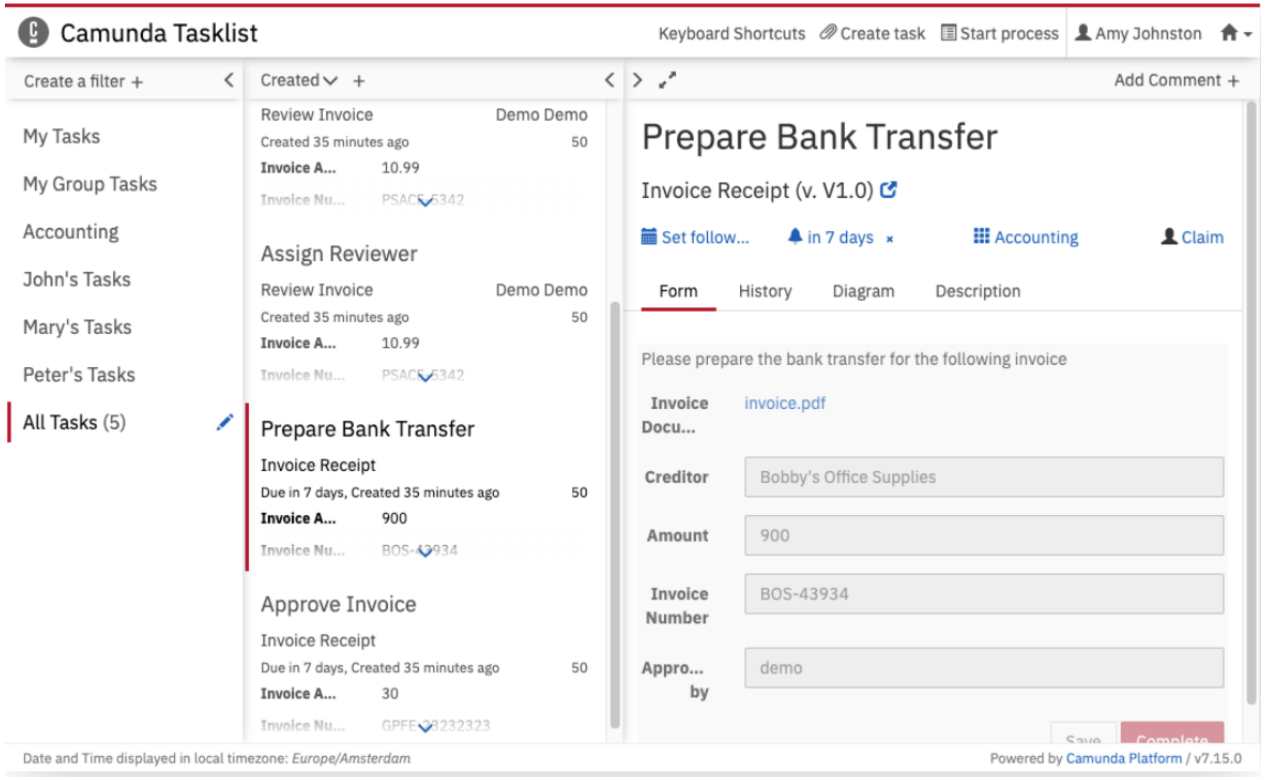
Operational Monitoring – Camunda 8 (Operate)

Real-Time Observability and Process Management

- **Unified Dashboard:** Operate presents a centralized view of running and completed process instances with metrics for throughput, duration, and incidents.
- **Process Visualization:** Visualizes BPMN flows, highlighting active and completed paths, variable states, and execution timelines.
- **Incident Analysis:** Supports investigation of service task failures, variable mapping errors, and retries directly within the interface.
- **Process Modification:** Allows runtime modifications such as adding or moving tokens, adjusting variables, and retrying failed steps.
- **Integration & Scalability:** Integrates with Zeebe and Elasticsearch for scalable event tracking and data analytics across distributed systems.

Tasklist & Human Task Management

Empowering Human-Centric Workflow Execution



Camunda Tasklist

Keyboard Shortcuts · Create task · Start process · Amy Johnston

Create a filter +

- My Tasks
- My Group Tasks
- Accounting
- John's Tasks
- Mary's Tasks
- Peter's Tasks
- All Tasks (5)

Created +

Review Invoice — Demo Demo
Created 35 minutes ago — 50
Invoice A... 10.99
Invoice Nu... PSACE-5342

Assign Reviewer
Review Invoice — Demo Demo
Created 35 minutes ago — 50
Invoice A... 10.99
Invoice Nu... PSACE-5342

Prepare Bank Transfer
Invoice Receipt
Due in 7 days, Created 35 minutes ago — 50
Invoice A... 900
Invoice Nu... BOS-43934

Approve Invoice
Invoice Receipt
Due in 7 days, Created 35 minutes ago — 50
Invoice A... 30
Invoice Nu... GPFE-23232323

Add Comment +

Prepare Bank Transfer

Invoice Receipt (v. V1.0)

Set follow... · in 7 days × · Accounting · Claim

Form · History · Diagram · Description

Please prepare the bank transfer for the following invoice

Invoice Docu...	invoice.pdf
Creditor	Bobby's Office Supplies
Amount	900
Invoice Number	BOS-43934
Appro... by	demo

Save · Complete

Date and Time displayed in local timezone: *Europe/Amsterdam*

Powered by Camunda Platform / v7.15.0

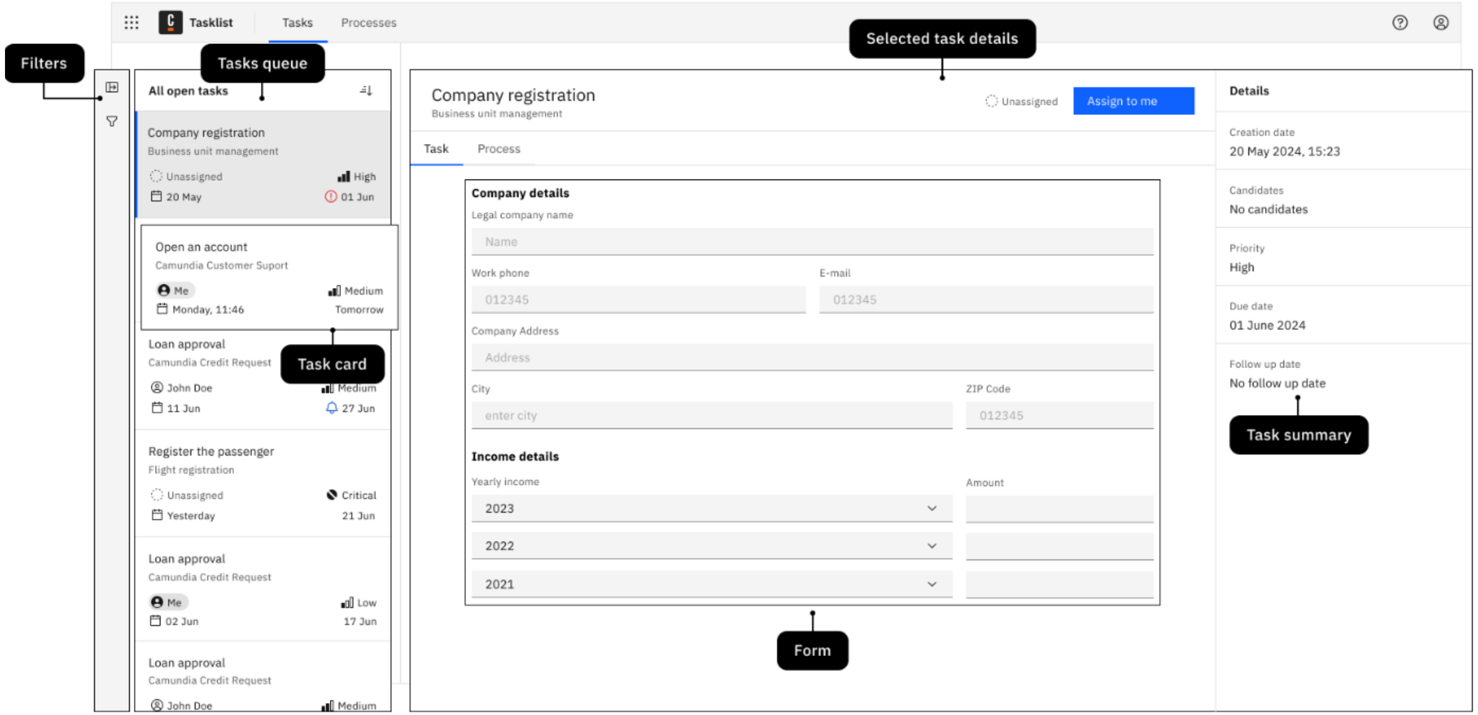
Tasklist & Human Task Management

Empowering Human-Centric Workflow Execution

- **Unified Task Management:** Tasklist enables users to view, claim, and complete assigned tasks across workflows, supporting both personal and group queues.
- **Flexible Task Views:** Camunda 7 offers basic views for ‘My Tasks’ and ‘Group Tasks,’ while Camunda 8 adds advanced filtering by variables, dates, and priorities.
- **Form Integration:** Embedded and external forms allow users to provide input directly, linking user actions to process transitions in real time.
- **Collaboration Features:** Supports delegation, escalation, and reassignment for collaborative task management and exception handling.
- **User Experience Evolution:** Camunda 8 Tasklist introduces improved UI/UX, real-time synchronization, and tenant-aware access for multi-tenant deployments.

Tasklist & Human Task Management

Empowering Human-Centric Workflow Execution



All open tasks

Company registration
Business unit management
Unassigned High
20 May 01 Jun

Open an account
Camundia Customer Suport
Me Medium
Monday, 11:46 Tomorrow

Loan approval
Camundia Credit Request
John Doe Medium
11 Jun 27 Jun

Register the passenger
Flight registration
Unassigned Critical
Yesterday 21 Jun

Loan approval
Camundia Credit Request
Me Low
02 Jun 17 Jun

Loan approval
Camundia Credit Request
John Doe Medium

Company registration
Business unit management
Unassigned Assign to me

Task Process

Company details

Legal company name: Name

Work phone: 012345
E-mail: 012345

Company Address: Address

City: enter city
ZIP Code: 012345

Income details

Yearly income	Amount
2023	
2022	
2021	

Details

Creation date
20 May 2024, 15:23

Candidates
No candidates

Priority
High

Due date
01 June 2024

Follow up date
No follow up date

Summary – The Road Ahead with Camunda 8

Embracing Intelligent, Scalable Process Orchestration

- **Architectural Transformation:** Camunda 8 represents a complete evolution from monolithic embedded engines to distributed, event-driven orchestration systems.
- **Scalability and Flexibility:** Zeebe’s microservice-based architecture enables horizontal scaling, fault tolerance, and independence across components and tenants.
- **AI-Driven Automation:** Native AI connectors and decisioning capabilities empower dynamic, context-aware business workflows.
- **Operational Visibility:** Enhanced observability through Operate, Optimize, and Tasklist ensures data-driven insights and continuous improvement.
- **Future-Ready Ecosystem:** Camunda 8 lays the foundation for intelligent, cloud-native automation at enterprise scale—aligning with hybrid, multi-cloud, and SaaS strategies.