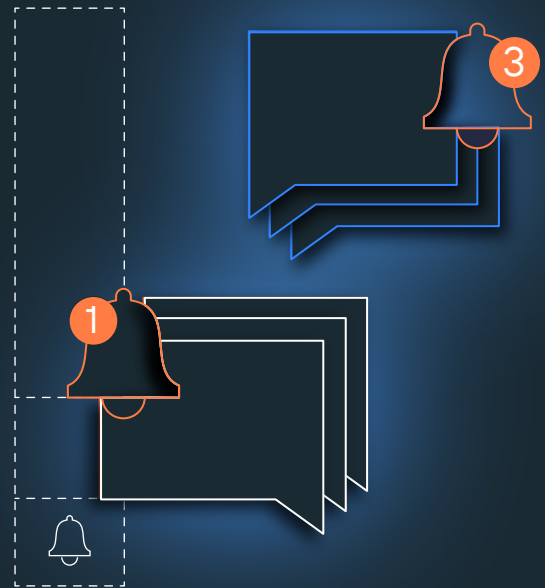


# io.Connect Notifications Server

Extend Notifications.  
Control Logic.  
Coordinate at Scale.



The io.Connect Notifications Server extends the platform's built-in notifications capabilities — giving engineering teams full control over how alerts are triggered, delivered, persisted, and resolved.

The Notifications Server transforms notifications from simple alerts into coordinated, action-ready events — powered by real-time business logic, user groups, and even AI systems.

## Create Customized, Notification-Driven Workflows

Your engineering teams are already building the workflows your users rely on. Now, they can define which apps and systems send alerts, to whom, and when — all with fully customizable lifecycle logic.

### Potential Use Cases:

**Notification groups** – Alert multiple users at once to share visibility and responsibility

**Action-based workflows** – Trigger predefined or dynamic actions when a user interacts with a notification

**AI-triggered events** – Allow AI systems to generate or respond to notifications just like a human user

**Four-eyes approval** – Require two users to review or act before a notification is dismissed

**First-touch routing** – Notify many, but assign the task to the first user who responds

## Benefits

From trading desks to AI-driven operations, the Notifications Server helps teams reduce noise and take smarter action — with the control to manage today's workflows and the flexibility to integrate tomorrow's AI-driven systems.

### **Key advantages include:**

**Groups of users share notifications and workload**

**Event history doubles as a task to-do list**

**High-priority tasks are surfaced with distinct UX to avoid being missed**

**Transparent audit trails for operational analysis**

**Designed for AI + human-in-the-loop workflows**

## Features

The Notifications Server is designed for the engineers, product leaders, and AI teams building smart, enterprise desktops — where timing, accountability, and user coordination matter. And ultimately, it delivers what end users need most: a faster, more intuitive way to work.

### **Key features include:**

**Configure persistence to apply on time basis or within/across user sessions**

**Publish events via REST API or Kafka**

**Configure retry and expiration parameters**

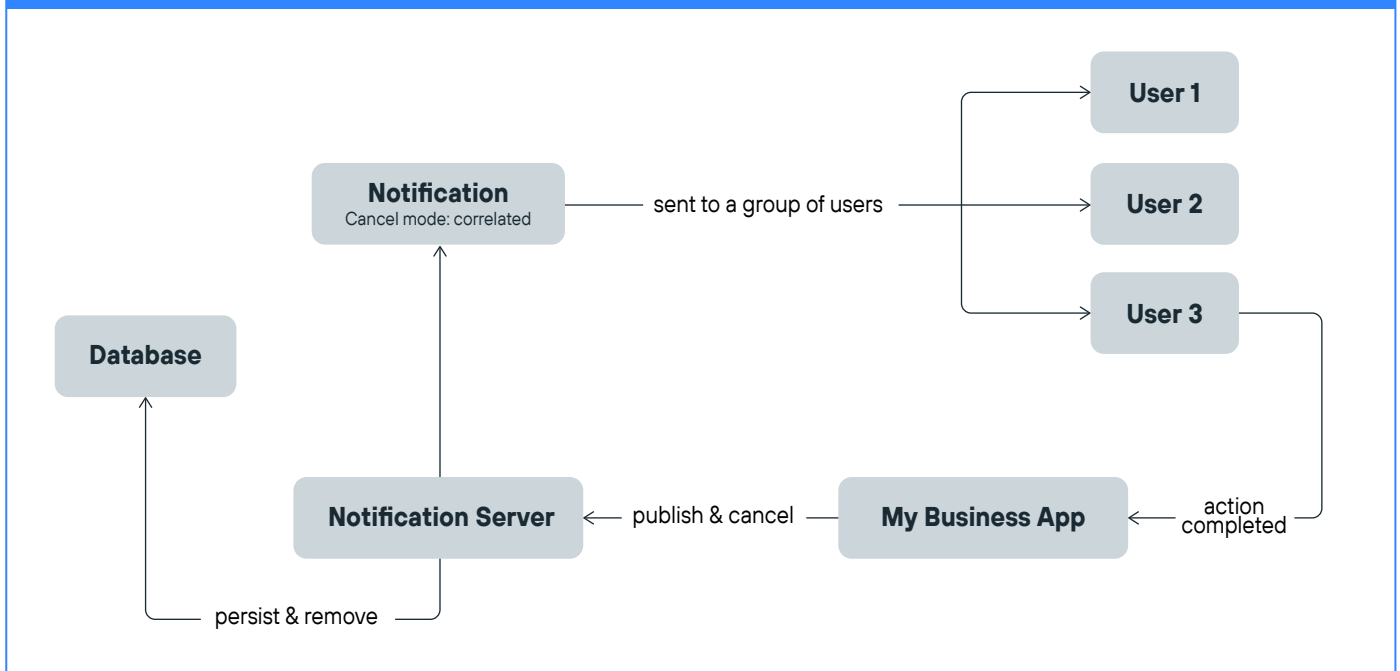
**Pre-define or dynamically assign action handlers to notifications**

**Support OpenID Connect and custom authentication**

**Deployable behind the firewall on-prem or in cloud environments**

**One Example:** A trading alert is sent to five users. When one user claims it, the alert is snoozed or removed for the others. Once action is taken, the system automatically dismisses the notification.

The notification is sent to a group and requires only one of them to perform an action before being canceled. The notification will be canceled and removed for all users when one of them completes the action - User 3 in this diagram.



The [io.Connect Notification Center](#) provides a beautiful, actionable UI for end users. But it's what happens behind the scenes that powers real-time coordination.

## Build a Better Notification System Today

This is not a plug-and-play product — it's a powerful backend capability that's custom-tailored for your specific workflows, user groups, and enterprise environment. That's why implementation is delivered in partnership with [interop.io Professional Services](#).

Our team works closely with yours to design, deploy, and optimize your notifications capabilities for maximum impact across your interoperable desktop — ensuring every notification behaves exactly as your business requires.



See how you can benefit from the io.Connect Notifications Server

**CONTACT US**