Notifications

The Kaa Notification subsystem enables delivery of messages from the Kaa cluster to endpoints (EP). Unlike configuration data that represents the desired EP state, notifications can be thought of as calls for a dynamic EP response. For example, a notification may cause a client application to display a message on the UI (a user notification), or initiate immediate or delayed EP configuration synchronization with the Operations server (a system notification).

Notification schema

The Kaa notifications functionality allows transferring any data to endpoints. The structure of the data that is carried by notifications is defined by the *notification schema* configured on the Kaa server and built into Kaa endpoints. The notification schema is defined similarly to the endpoint profile schema. In addition to the user-defined schema and its version, notifications are characterized by their type that can be either *system* or *user*. System notifications are processed by the prepackaged endpoint functions, while user notifications are handed over to the client code via the endpoint API.

Since the data structure requirements may evolve throughout the Kaa-based system lifetime, the Kaa server can be configured to simultaneously handle notification schemas of multiple versions. In this case, a notification will have multiple schema versions associated with it, as well as multiple sets of notification data so that each schema version is covered. To deliver such a notification to an endpoint, the server chooses the schema version supported by the endpoint.

Notification topics and subscriptions

Notification topics are used for segregating notifications within the application. Every notification in the system must be associated with some topic. Thus, to receive notifications, every endpoint needs to be subscribed to corresponding topics.

Also, it is possible to make some of the topics configured within the application accessible only to some selected endpoint groups. For this purpose, each topic created in Kaa can be assigned to one or more endpoint groups (global by default). In order for an endpoint to receive notifications associated with the topic, the endpoint must belong to at least one group that supports this topic.

Topics can be either optional or mandatory. Optional topic notifications are delivered only after the endpoint has subscribed to the topic. Mandatory topic notifications are delivered in an enforced manner. It is the responsibility of the client code to set appropriate listeners for the topics the endpoint is subscribed to.

Notification pipelines

Notification pipelines manage individual notifications within a topic. A notification remains queued in the pipeline until its time-to-live (TTL) expires, after which the notification is dropped. The notification pipeline type specifies the scope of notifications delivery. It can be either multicast (targeted to an unbounded number of endpoints) or unicast (targeted to a single specific endpoint).

Multicast pipelines
Whenever a notification is sent to a topic, it gets added to the corresponding pipeline with a unique sequential index per pipeline. Endpoints independently maintain their position in every pipeline by remembering the last sequential index per pipeline that they received.

Unicast pipelines
Whenever a notification is sent to a topic, it gets added to the corresponding pipeline with a unique ID. Endpoints independently maintain their pipeline by reporting all received notification IDs. The server removes the notification from the pipeline once it receives a receipt confirmation from the endpoint. Sending a unicast notification requires specifying the endpoint and topic IDs. The endpoint must be subscribed to the specified topic, otherwise an error will be returned.