

Glossary

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Term	Description
A	
Administration UI (also, Admin UI and web UI)	An online tool for managing Kaa users, applications, schemas, etc. Refer to Administration UI guide for more information.
Application	The application in Kaa represents a family of available implementations of a specific software application used by endpoints. For example, two versions of a sound frequency measuring application which differ by their implementation for, respectively, Arduino and STM32 platforms would be considered the same application in Kaa. Refer to Design reference main page for more information.
Avro UI form	A GUI component in Admin UI that allows the user to either create Kaa schemas or enter corresponding data records, in both cases without using the Avro/JSON syntax. Refer to Avro UI forms for more information.
B	
Base schema	A derivative schema obtained from the configuration schema and used within the given application for updating configuration of all endpoints belonging to that configuration schema. Refer to Configuration and its Schema-specific configuration management section for more information.
Bootstrap server	One of the three server types in Kaa, with the other two being the Control server and Operations server. A Bootstrap server is responsible for directing endpoints to Operations servers. On their part, Kaa endpoints have a built-in list of Bootstrap servers set up in the given Kaa deployment. The endpoints use this list to query the Bootstrap servers and retrieve from them a list of currently available Operations servers, as well as security credentials. Bootstrap servers maintain their lists of available Operations servers by coordinating with the ZooKeeper service. Refer to Design reference main page for more information.
C	
Client (also, Kaa client)	A particular application or piece of software which uses the Kaa endpoint SDK and resides on a particular connected device. It is the responsibility of the Kaa client to process structured data provided by the Kaa server (configuration, notifications, etc.) and to supply data to the return path interfaces (profiles, logs, etc.). Refer to Design reference main page for more information.
Configuration schema	A user-defined specification of the application data model that Kaa Configuration subsystem uses to configure endpoints registered under the application. In other words, the configuration schema defines in which format the actual configuration data should be entered by the user/developer and then transferred to the endpoints. Note that there can be several schemas created for a single application. The version of the schema for the endpoint is selected during SDK generation . Refer to Configuration and Administration UI guide for more information.
Configuration data (in short, configuration)	A set of configuration values specified by the user based on the corresponding configuration schema. Having been specified on the server, these values are then distributed to the endpoints that belong to the corresponding application and support the corresponding configuration schema. Refer to Configuration for more information.
Control server	<p>One of the three server types in Kaa, with the other two being the Bootstrap server and Operations server. It is responsible for managing overall system data, processing API calls from the web UI and external integrated systems, and delivering notifications to Operations servers. A Control server manages data stored in a database (independently for each tenant) and notifies every Operations server on most data updates via a Thrift-based protocol. A Control server maintains an up-to-date list of available Operations servers by continuously obtaining this information from ZooKeeper.</p> <p>To support high availability, a Kaa cluster must include at least two Control servers, with one of them being active and the other(s) being in a standby mode. In case of the active Control server failure, ZooKeeper notifies one of the standby Control servers and promotes it to the active Control server.</p> <p>Refer to Design reference main page for more information.</p>
D	

Delta (also, <i>delta update</i>)	The difference between the new configuration created on the server (and due to be applied to the endpoint) and the current configuration used by the endpoint. The delta is sent to the endpoint and then merged with the endpoint current configuration to achieve the required up-to-date configuration. Using deltas instead of full configuration resets is an effective way to reduce load on available data channels. Refer to Configuration and its Endpoint data synchronization section for more information.
E	
Endpoint (EP)	An abstraction which represents a separate managed entity within a Kaa deployment. Practically speaking, an endpoint is a specific Kaa client (see <i>Client</i>) registered (or waiting to be registered) within a Kaa deployment. For example, a news application installed on your mobile phone, the same news application installed on your tablet, and the same news application on your WiFi-enabled fridge would be considered three different endpoints in Kaa. Refer to Design reference main page for more information.
Endpoint group	An independent managed entity which is defined by the profile filters assigned to it. Those endpoints whose profiles match the profile filters of the specific endpoint group become automatically registered as members of this group. There is no restriction for endpoints on having membership in more than one group at a time. Refer to Endpoint grouping for more information.
Endpoint group profile filter (in short, <i>profile filter</i>)	A predicate expression based on the Spring Expression Language which defines characteristics of the corresponding group members (endpoints). These filters are executed against the endpoint profile to figure out whether or not the endpoint belongs to the group. Refer to Endpoint profiling for more information. Refer to Endpoint grouping for more information.
Endpoint profile	A structured data set of custom-defined complexity that describes specific characteristics of the endpoint. Endpoint profiles are used to classify endpoints into endpoint groups. Refer to Endpoint profiling for more information.
Endpoint profile schema (in short, <i>profile schema</i>)	Defines an endpoint profile structure. A profile schema can be used across multiple endpoints. It is defined by the Apache Avro schema format and supports all of Avro features: primitive types, complex types, arrays, maps, etc. Refer to Endpoint profiling for more information.
Endpoint SDK (also, <i>Kaa SDK</i>)	A library which provides communication, data marshaling, persistence, and other functions available in Kaa for specific type of an endpoint (e.g. Java endpoint SDK, C++ endpoint SDK, C endpoint SDK). This SDK can be used to create Kaa clients, which are any pieces of software that utilize Kaa functionality and are installed on some connected devices. Refer to Design reference main page for more information.
Event	A message generated by the endpoint based on a particular event class and in accordance with the corresponding event class schema. After being generated, an event is sent to the Kaa server for further processing. Refer to Events for more information.
Event class (EC)	A description of an event structure in the form of the event class schema. An EC is uniquely identified by a fully qualified name (FQN) and a tenant. In other words, there can be no two ECs with the same FQN within a single tenant. Refer to Events for more information.
Event class schema	An Avro-based schema used for specifying an event structure. An event class schema contains the classType attribute that supports two values: <i>event</i> and <i>object</i> . Kaa uses the classType attribute to distinguish actual events from objects, which are reusable parts of events. This is useful for avoiding redundant methods in SDK API. Refer to Events for more information.
Event class family (ECF)	A group of events related together by the event subject. ECFs are registered within the Kaa tenant together with the corresponding event class family schemas. An ECF is uniquely identified by its name and/or class name and tenant. In other words, there can be no two ECFs with the same name or same class name within a single tenant. Refer to Events for more information.
F	
G	
Group "all"	A default, non-editable group created for each Kaa application. The profile filter of this group is automatically set to "true" for every profile schema version in the system. As a result, the "all" group contains every endpoint registered in the application. The "all" group is used to define the default configuration, default notification topics access list, and for some other special functions. Refer to Endpoint grouping for more information.
H	

I	
J	
K	
Kaa admin	The highest-level administrator of Kaa. He is able to create, edit, and delete tenant admins. See also <i>tenant admin</i> and <i>tenant developer</i> . Refer to Administration UI guide for more information.
Kaa framework	The Kaa framework consists of the Kaa server and endpoint SDKs. The Kaa server implements the back-end part of the framework, exposes integration interfaces, and offers administrative capabilities. An endpoint SDK is a library which provides communication, data marshaling, persistence, and other functions available in Kaa for specific type of an endpoint (e.g. Java-based, C++-based, C-based). This SDK can be used to create <i>Kaa clients</i> , which are any pieces of software that utilize Kaa functionality and are installed on some connected devices. It is the responsibility of the Kaa client to process structured data provided by the Kaa server (configuration, notifications, etc.) and to supply data to the return path interfaces (profiles, logs, etc.). Refer to Design reference main page for more information.
Kaa instance (also, Kaa deployment)	A particular implementation of the Kaa framework and it consists of a Kaa cluster and endpoints. A Kaa cluster represents a number of interconnected Kaa servers. An endpoint is an abstraction which represents a separate managed entity within a Kaa deployment. Practically speaking, an endpoint is a specific Kaa client registered (or waiting to be registered) within a Kaa deployment. For example, a news application installed on your mobile phone, the same news application installed on your tablet, and the same news application on your WiFi-enabled fridge would be considered three different endpoints in Kaa. Refer to Design reference main page for more information.
L	
Load balancing strategy	A particular way of re-balancing workload between the Operations servers within a Kaa cluster to achieve more or less equal load for each server. Kaa implements a number of load balancing strategies and automatically uses them at run time.
Log appender	A service utility which resides on the Operations server. This utility is responsible for writing logs (received by the Operations server from endpoints) to a single specific storage, as defined by the log appender's type. Each Kaa application may use only one log appender at a time. A Kaa developer is able to add, update and delete log appenders using Admin UI or REST API . Kaa provides several default implementations of log appenders. It is also possible to create custom log appenders. Refer to Log appenders for more information.
Log schema	Defines the structure of logs collected by endpoints and subsequently transferred to the server. A log schema is fully compatible with the Apache Avro schema . There is one log schema defined by default for each Kaa application. This schema supports versioning, therefore, whenever a new log schema is configured on the Kaa server for the application, this new schema gets a new sequence version assigned. The Kaa server maintains compatibility with the older versions of the log schema to ensure proper functioning of the clients that for some reason are not yet upgraded to the latest schema version. Refer to Logging for more information.
M	
Multitenancy	A capability of Kaa to serve multiple user groups within a single Kaa instance by providing separate functionality to each of those groups. Refer to Design reference main page for more information.
N	
Notification pipeline	Manages individual notifications within a notification 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). Refer to Notifications for more information.
Notification schema	Defines the structure of data transferred by corresponding notifications from the server to endpoints. Refer to Notifications for more information.
Notification topic	Allows grouping related notifications within the application. It is required that every notification in the system be associated with some topic. Thus, to receive notifications, every endpoint needs to be subscribed to corresponding topics. Refer to Notifications for more information.
O	

Operations server	<p>One of the three server types in Kaa, with the other two being the Bootstrap server and Control server. An Operations server is a "worker" server that is responsible for concurrently handling multiple requests from multiple clients. Most common Operations server tasks include endpoint registration, endpoint profile updating, configuration updates distribution, and notifications delivery.</p> <p>Multiple Operations servers may be set up in a Kaa cluster for the purpose of horizontal scaling. In this case, all the Operations servers will function concurrently. In case an Operations server outage happens, the corresponding endpoints switch to the other available Operations server automatically. A Kaa cluster provides instruments for the workload re-balancing at run time, thus effectively routing endpoints to the less loaded Operations servers in the cluster.</p> <p>Refer to Design reference main page for more information.</p>
Override schema	A derivative schema obtained from the configuration schema and used within the given application for updating configuration of a specific endpoint group belonging to that configuration schema. Refer to Configuration and its Group-specific configuration management section for more information.
P	
Profile filter	See <i>endpoint group profile filter</i> .
Profile schema	See <i>endpoint profile schema</i> .
Protocol schema	Determines the structure of the data updates that Kaa server sends to the endpoints and possible update actions (change, add, reset, and leave unchanged). A protocol schema is automatically generated by the Control server for each configuration schema (when the configuration schema is loaded by the user). Refer to Configuration for more information.
Q	
R	
S	
T	
Tenant admin	A Kaa user who is responsible for managing applications, users and event class families. See also <i>Kaa admin</i> and <i>tenant developer</i> . Refer to Administration UI guide for more information.
Tenant developer	A user that creates SDKs based on customer requirements. Tenant developers set schemas, group endpoints, and control notification processes. See also <i>Kaa admin</i> and <i>tenant admin</i> . Refer to Administration UI guide for more information.
Transport	A Kaa server component responsible for communication with a Kaa client. Refer to Transports for more information.
Transport channel	A Kaa client component responsible for communication with a Kaa server. Refer to Transports for more information.
U	
User verifier	A server component that handles the user verification. Refer to User verifiers for more information.
V	
W	
web UI (also, Administration UI)	An online tool for managing Kaa users, applications, schemas, etc. Refer to Administration UI guide for more information.
X	

Y

Z