

Transports

- [Kaa services and data transfer](#)
- [Assigning channels to services](#)
- [Default transport channels](#)

One size does not fit all. Depending on the project, data transport requirements may vary drastically. The transport protocol reliability, security, data consumption, transport delay, computational complexity, energy budget, and many other criteria have to be factored in when considering how to establish effective communication between the server and endpoints.

This is why Kaa is designed to support virtually any data transport protocols. Moreover, Kaa allows developers to use different transport protocols for different actions performed over the same endpoint. For example, notifications can be delivered as SMS messages, whereas configuration and profile data - as TCP traffic. This ability is achieved by the abstract implementation of transport layer so that the Operations server can use any specific transport method when establishing a transport channel. Each channel supports a specific transport protocol and is responsible for data encoding, compression, and delivery.

Kaa provides default transport channel implementations for all its services. However, developers can create custom implementations of transport channels for any of the Kaa services and thus override the default data channels. Both the endpoint and the Operations server are able to differentiate between the transport channel instances and select an appropriate channel for sending data during a specific session based on predefined rules.

Kaa services and data transfer

Each of the Kaa services is responsible for specific data exchange functionality, as follows:

- The [Bootstrap](#) service supplies endpoint with a list of currently available Operation servers.
- The [Configuration](#) service supplies endpoints with the configuration schema and configuration data.
- The [Notifications](#) service delivers notifications from the server to endpoints.
- The [Events](#) service handles the messages exchanged between endpoints.
- The [User Feature](#) service registers a new user and adds new endpoints to the server.
- The [Logging](#) service sends recorded data (logs) from endpoints to the server.
- The [Profile](#) service updates the endpoint profile on the server side.

Assigning channels to services

There are two methods of assigning a transport channel to a service. The first is to assign a separate transport channel to each service. The other method is to group all or some of the services and assign them a common transport channel. However, it is important to remember that one channel can maintain only one open session at a time.

Each transport channel is capable of transferring data in one of the following modes.

- From the endpoint to the server (upstream)
- From the server to the endpoint (downstream)
- Both ways (bi-directional)

In case there are several upstream channels created for some service, Kaa uses the most recent channel. For example, let's assume that in addition to a channel that works upstream for three services (configuration, notification, and events), a new channel was created and registered to work upstream for one of those services (let's say, notification). As a result, the notification service will communicate upstream through the second channel only.

However, Kaa can use multiple transport channels in the downstream mode for the same service. Moreover, if the channel which is currently in use for upstream communication supports the downstream mode for the same service, the server assigns that channel to participate in downstream communication as well.

Default transport channels

Kaa provides four default transport channels that cover data exchange needs of all the Kaa services. Each channel is characterized by the transport protocol, server type, transfer mode (upstream, downstream, or bi-directional), and one or multiple services. The default channels available together with Kaa are as follows:

Channel name	Target server	Services	Supported modes	Based on
Default bootstrap	Bootstrap	Bootstrap	Bi-directional	HTTP 1.1
Default operation long poll	Operations	All except bootstrap	Bi-directional for profiling, configuration, notifications, and user association. Downstream for events and logs.	HTTP 1.1 long poll
Default operation HTTP	Operations	Events, logging	Upstream for events and logs.	HTTP 1.1
Default KaaTCP channel	Operations	All except bootstrap	Bi-directional	mqtt v3.1

If a custom channel was created to work upstream for some service, Kaa will always use the custom channel for that service, because the custom channel is always more recent than the default channel.

Copyright © 2014, [CyberVision, Inc.](#)