i3 SDK (Software Development Kit)

The i3 software supports governance and management of information in federated operating environments. At the core of the i3 system there is a message distribution fabric. i3 accepts messages from data systems such as IoT devices and other data processing appliances, applies information processing rules to these data messages, and distributes the information in accordance with the information owner's direction.

The i3 core delivers information to one or more destinations which have applied for access to the information and been accepted by the device managers that own the information associated with the devices they support.

If an application is being developed for use in an i3 environment, the developers can incorporate the i3 APIs directly into their software. However, if the objective is to use an established application with proprietary APIs in an i3 environment, an application wrapper can needed to link the pre-defined application into i3.

Similarly, if a device is being built for use in an i3 environment, the developers can incorporate the i3 APIs directly into the device's software system. However, if the objective is to use a commercially available device with proprietary APIs in an i3 environment, a device wrapper can be used to link the device into i3.

Device wrappers can also be used to equalize information flows when different but similar data is being generated by various devices. The device wrappers can be used to transform the messages from these different devices into a common information format.

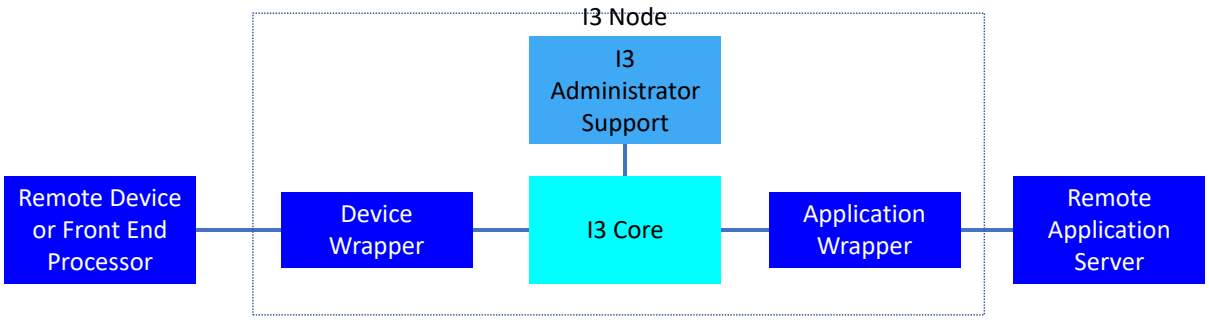
The i3 SDK (Software Development Kit) provides software tools and uses examples to describe the process of developing application and device wrappers. The most simplistic of application wrappers accepts information from the i3 core information fabric using the i3 APIs and write that information to an application specific API. The most simplistic of device wrappers accepts information from the data device and write those messages to i3's API. For these simplistic cases, the primary purpose of the wrapper is to login and authenticate the device or application with i3 and with the application or device respectively.

However, these simplistic wrappers can be more complex if the situation warrants. For example, a device wrapper can break a single message from a remote device into two information types giving the appearance that the device has issued two messages. Terms and parameter names within a message can also be changed in a device wrapper in order to equalize messages that come from devices supplied by different suppliers. Device wrappers can also be used to create messages that are not native to the device or the wrapper can be used to remove messages coming from a device from the upstream information flow.

Application wrappers are similar to device wrappers in that they serve as flex points that serve targeted applications. While device wrappers serve to accept messages from information producers and convert them to normalized information network messages, the application wrappers accept standardized messages from the information network and convert them to application specific APIs. Like device wrappers, application wrappers can transform information message structures and formats. They can also combine information messages for submittal to the application's API or they can serve to reduce the information flowing to the application as the situation warrents.

These potential transformational information processing functions of a wrapper can be useful to equalize the appearance of a device's messages in a multivendor environment. It can also improve management granularity by allowing applications to request access to a subset of device information that might not natively be supported by the device generating the information.

The wrapper structure can be made to periodically poll state-drive devices or it can be used to accept autonomously generated messages and submit the moderated messages to the i3 core.



There may be a unique wrapper for each device.

The messages created by the device wrapper corresponds to a data type defined in the I3 system.