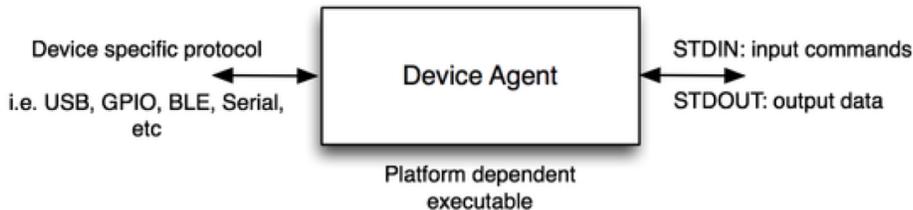


Architecture

Device Agent

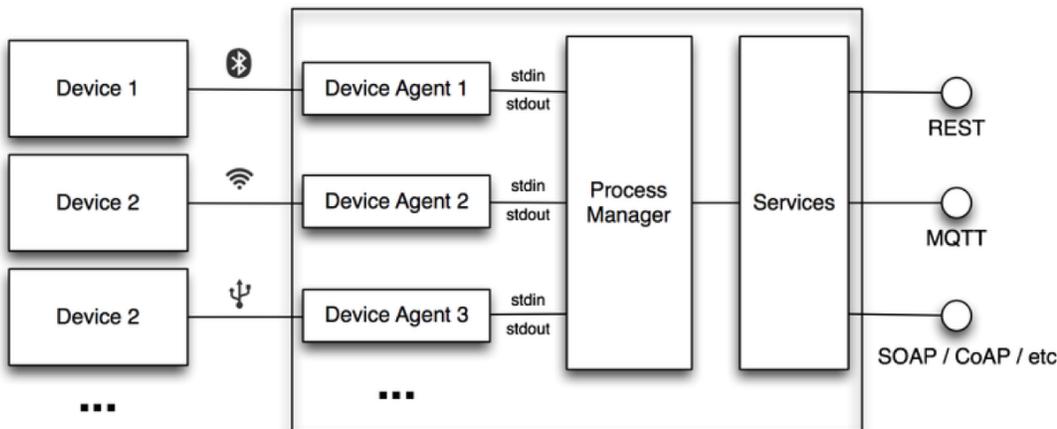
One of the core concepts of the DGW is the *Device Agent* - an executable (preferably platform independent, but in most cases they are dependent due to the different libraries for accessing the hardware), which implements the low-level communication with the actual device using its interface and protocol (e. g., talking via GPIO to temperature sensor, reading data via USB or detecting beacons via BLE). This is the left side of the communication in the Figure below.



The right side of it describes how the *Device Agent* communicates with the DGW managing its execution: it is done using the most universal and ubiquitous interface - standard system input and output streams (*stdin*, *stdout* correspondingly). The standard system error stream (*stderr*) is used for logging purposes (data written into *stderr* will be forwarded to DGW for output in the debug console).

Process Manager

The *Device Agent* executable is not aware of the DGW and its interfaces. It is a standalone program that can be executed manually from a command line. In the DGW context, this program is executed and managed by the DGW's *Process Manager*, as depicted in the Figure below (a high-level overview of the DGW architecture):



The core of the system is the *Process Manager* that reads devices/resources configuration and executes corresponding device agents and redirects the system streams (*stdin*, *stdout*, *stderr*). *Process Manager* supports 3 types of agent execution: *task*, *timer* and *service*.

- *task* execution means the *Device Agent* is executed only once per request coming from *Services* component. The last value is cached for a given TTL period.
- *timer* execution is similar to the task execution, but initiated by the *Process Manager* periodically (using interval value from the configuration). The value is cached in between the executions.
- *service* execution means the *Device Agent* is running as a process and producing output into *stdout* constantly. The last value is cached as well.

Services

The *Services* component is responsible for establishing communication with the devices connected to the DGW by applications and services over the network via standardised protocols. In current implementation, REST and MQTT protocols are supported.

The *Services* component creates a corresponding RESTful endpoint for each devices/resources configured with REST protocol and establishes a MQTT publication connection for devices/resources configured to use MQTT protocol. This component passes data from POST/PUT/DELETE HTTP requests to the corresponding *Device Agents* by writing it to their *stdin* and returning the (cached) value received from *stdout* to the HTTP GET requests. For resources configured to use MQTT, all messages written to *stdout* by the corresponding *Device Agents* are published to the configured MQTT broker.