Network structure

Paolo Bosetti

2025-06-04

Illustrate a typical MADS network structure and its requirements.

Table of contents

Architecture

The typical architecture of a MADS network can be represented as:

> **!** Important
>
> Remember that the above schematic represent *processes*, regardless the physical machine on which they are being executed.
> For example, the whole network could run on a single workstation, or it could be conversely distributed over multiple devices connected to the same IP network, each device running a single process/node.

In the figure Figure 1, the solid lines represent a ZeroMQ connection over TCP/IP, which uses compressed JSON as a data encoding protocol. Compression is preformed with the snappy

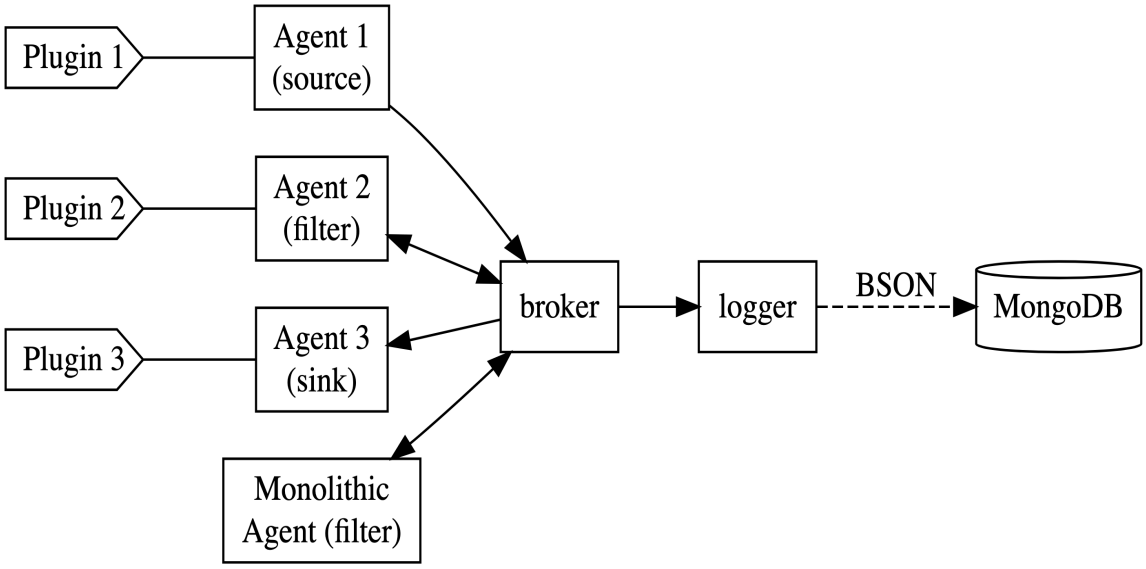

Figure 1: MADS Network

library. The dashed line, conversely, represents the proprietary MongoDB protocol, with data serialized as BSON (Binary-JSON).

The broker

What is the broker purpose?

The broker solves the issue of knowing multiple network addresses when you have a number of devices participating to the same distributed system.

With the aid of the broker, any separate device partaking to the MADS network only needs to know a single hostname/IP address: that of the machine running the broker.

> **Warning**
>
> There can only be a single broker per network.

Running the broker is quite simple:

```
mads broker
```

The agents

Agents can be:

- **monolithic**: implemented as a single executable inheriting the `Mads::Agent` C++ class.
- **plug-in**: a single executable that on runtime loads a proper plug-in (i.e. a dynamically loaded library)

Regardless the type, agent can have three different behaviors:

- **source**: they provide information to the network (e.g. by reading sensors)
- **filter**: they operate and transform received information
- **sink**: they consume information received from the network (e.g. to store or visualize)

The MADS installer provides three general purpose agents, aptly named `source`, `filter`, and `sink`, that are designe do load proper plugins. The command `mads plugin` can be used to generate a suitable template for a new plugin to be developed.

Layout and requirements

> **Note**
>
> To be done.

How to develop a plugin

> **Note**
>
> To be done.

How to deal with time

> **Note**
>
> To be done.