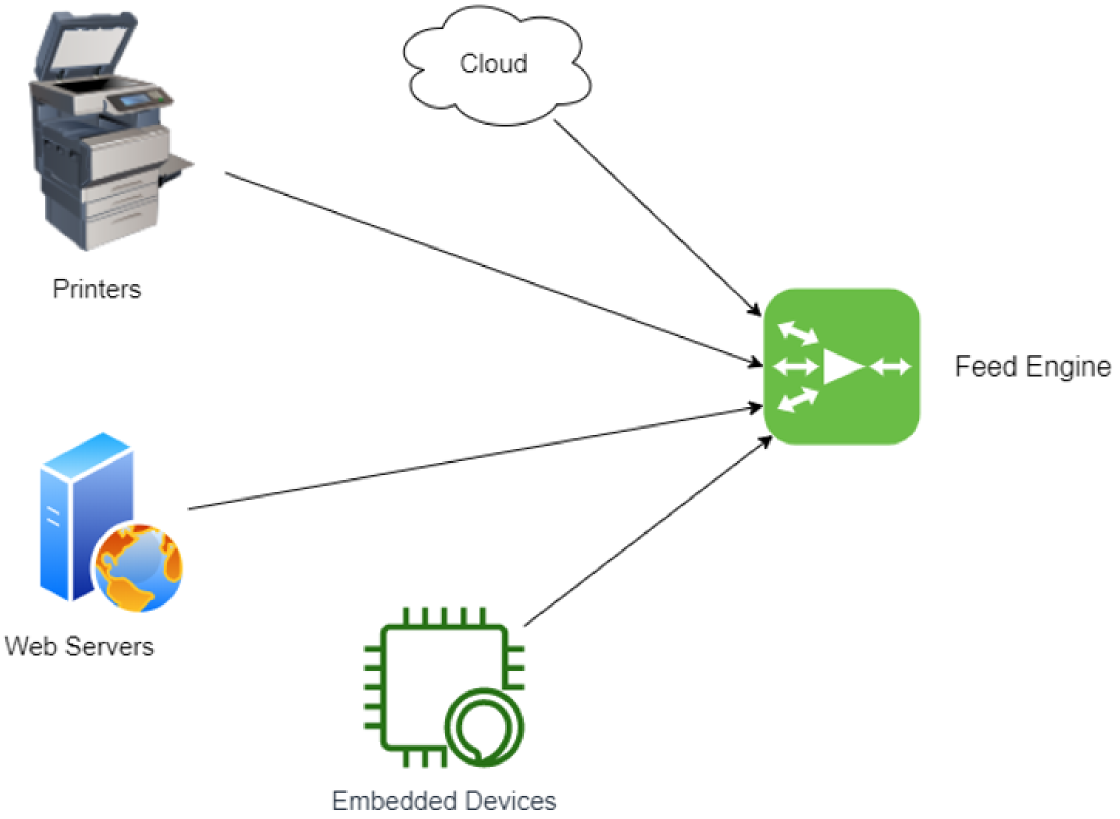
Feed Engine v1.0

(Progress Update : 18-06-2020)



Description

Used to parse the feeds from various sources (Printers, Network devices, Web servers and other connected devices). These feeds can be used in checking load balancing, health status, tracing.

Usage

To streamline the process we are utilising the `Protocol Buffers` and `gRPC` framework.

The engine runs on `9000` port by default. All devices should submit the feeds in serialized format such that data transmission is fast and accurate across network.

We defined a `Print` service which has a `RPC` method called `Feed`. This method takes `Content` as input parameter and returns `Data` from the server.

The `Content` message definition specifies a field `data` and `Data` message definition specifies a field `feed`.

On successful data transmission you should see a message.

```
...
return service_pb2.Data(feed='Pushing feeds')
...
```

Here is how a sample feed information looks like.

```
{
    "version": "v1.0",
    "title": "Printer Feed",
    "home_page_url": "http://printer.laserinternal.htb/",
    "feed_url": "http://printer.laserinternal.htb/feeds.json",
    "items": [
        {
            "id": "2",
            "content_text": "Queue jobs"
        },
        {
            "id": "1",
            "content_text": "Failed items"
        }
    ]
}
```

QA with Clients

> Gabriel (Client) : What optimisation measures you've taken ?

> Victor (Product Manager) : This is main aspect where we completely relied on gRPC framework which has low latency, highly scalable and language independent.

> John (Client) : What measures you take while processing the serialized feeds ?

> Adam (Senior Developer) : Well, we placed controls on what gets unpickled. We don't use `builtins` and any other modules.

Release Info

Currently we are working on v1.0 with basic feature which includes rendering feeds on dashboard.

Bugs

> 1. Error handling in _InactiveRPCError

> 2. Connection timeout issues

> 3. Forking issues

> 4. Issue raised by clients in last update

Todo

> 1. Fork support to increase efficiency for more clients

> 2. Data delivery in more formats

> 3. Dashboard design and some data analytics

> 4. Merge staging core to feed engine