

# e-Me

## Prototype – Consumer Services



# 1 Consumer Services

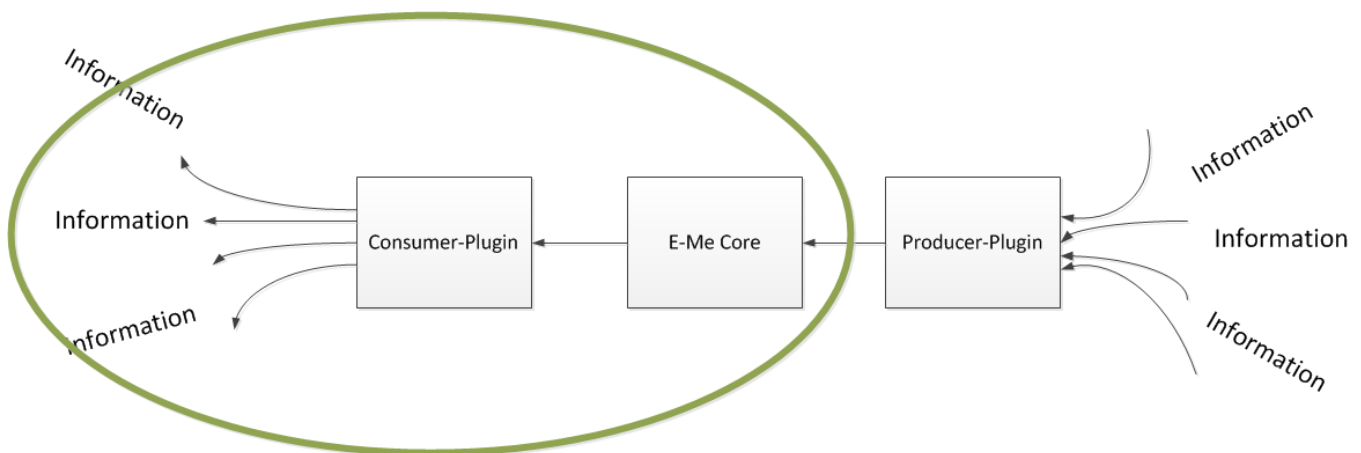
## 1.1 What is a consumer service?

A Consumer service is a plugin. This plugin is responsible of delivering information to an end user. A consumer plugin targets a single source. This can be anything from posting updates on twitter to sending a SMS.

The consumer plugin is started by e-Me Core and is provided with three sets of information, a collection of messages, files and events. These are the three formats of data that e-Me handles. The plugin chooses what types should be delivered. This is entirely down to the plugin to decide what to do.

Once the consumer plugin has sent the messages it should, ID's of the successfully sent items are returned to e-Me Core. That way Core can clean up data in the database and remove stored messages that have been delivered. The plugin is also responsible of returning a result state. This state can be either Successful or Failed.

The image bellow illustrates the consumer plugin, highlighted with a green ellipse.



## 1.2 Consumer service architecture

A consumer plugin is a Java Web Archive or WAR. This war is built using Maven, a technique commonly used to manage dependencies with libraries and projects. This way we have set up a maven repository that enables automatic download of the dependencies for the project without any manual interference. And the developer does not need any external code or resources to create the plugin.

## 1.3 Software

Some of the techniques and libraries used in consumer plugins are:

- Java
- Maven
- Tuckey urlrewritefilter
- Spring framework
- JBoss
- JUnit

## 1.4 Dependencies

All plugins are dependent on e-Me Core. When a developer downloads the plugin template and builds it, it automatically downloads parts of e-Me core that it needs to run. This is done to make sure that the plugin follows the forms of data and communication that e-Me Core uses. (This is only for the communication between Core and the plugin)