## Purpose

This document is dedicated to description of offer to design and implementation of information system for retrieving and collecting information from social networks about different media content - **Metrics Collector** (or MC, name can be discussed). Such information will contain some metrics (number of visitors, number of likes, posts, etc). MC does not have any analytics blocks or modules, MC just provides information for other Analytics Systems. Information is provided via Storage.

## Basic Definitions

**Source** - social network or any other data source which can be used to solve business goals. As a rule, source provides an own API to communicate with him.

**Connector** - module which communicates with Source using corresponding API. Connector retrieves *Metrics Data* from *Source* and returns it back.

**Metrics Data** - data which contains metrics. Metrics will be defined in design in accordance with business requirements and possibilities of each particular *Source*.

**Data Saver** - module which saves *Metrics Data* received from *Connector* into *Storage*.

**Storage** - SQL DB (or any other).

**Task** - module which combines *Connector*(s) and *Data Saver*(s) to process tasks. *Task* can be considered also as container of *Connectors* and *Data Savers* chain. It is assumed that *Task* will contain one *Connector* and one *Data Saver* (at least on first phase).

**Scheduler** - module which starts *Task* in accordance with configured schedule. One *Engine* will contain many *Schedulers*, one *Scheduler* can start only one type of *Task*. Type of *Task* usually related to *Connector*.

**Engine** - container of *Schedulers*, *Engine* also has functions of *Engine Configuration* management, logging. Each *Engine* has an own identifier, *Engine identifier* is used to separate configuration of different Engines deployed into one *Cluster*.

**Engine Configuration** - set of configuration data for *Engine* functioning.

**Cluster** - container of *Engines*, Cluster also can has *Cluster Dashboard* module.

**Cluster Dashboard** - set of REST services which provides functionality to configure *Cluster* (i.e. deployed Engines). *Cluster Dashboard* provides possibility to create *UI*.

The system can contain other additional modules, for example *Processing Monitor*, *Data Replication*, etc.

**Cluster Administrator**  - a responsible person who manages *Cluster*, different *Clusters* can have different *Cluster Administrators*.

Schematically:

**Cluster**

Engine 1

...

Engine 2

Engine N

Cluster Dashboard

**Engine**

Scheduler 1

...

Scheduler 2

Scheduler N

Engine Configuration

**Scheduler**

Connector

Data Saver

Task

The common steps of processing is the following:

1. Engine reads configuration and starts all configured Schedulers.
2. Scheduler in accordance with configuration starts Task (*TIMESTAMP*)
3. Task performs an internal logic (calls Connector and Saver which are contained inside Task).
   1. Connector calls Sources and returns Metrics Data to Task.
   2. Task passes Metrics Data to Data Saver.
   3. Data Saver saves data into Storage.
4. Scheduler calculates next start of Task taking into consideration existing limitations and *TIMESTAMP*.
5. Scheduler performs step #1.

## Deployment

Apache Tomcat as deployment environment is proposed to use, there are reasons:

* Each Engine is deployed as separated WAR, name of WAR file is used for Engine identification (Engine identifier).
* Separately deployed Engines work isolated from each other.
* Tomcat provides access through HTTP/HTTPS, this access can be used for UI.

**Apache Tomcat Webapps**

Engine1.war

Engine2.war

Engine3.war

*In this example 3 engines with identifiers Engine1, Engine2 and Engine3 and Cluster Dashboard are deployed.*

cluster-dashboard.war