Measure Platform Installation Guide

Measurement and Data Analysis Platform

Measure Platform Overview

The measure platform is a tool dedicated to measure, analyse, and visualise the metrics and to extract and show information of the software engineering processes.

- Implement the tools to automatically measure software engineering processes during the whole software lifecycle by executing measures defined in SMM standard and extracted from a catalogue of formal and platform-independent measurements.
- Provide methodologies and tools which allow measure tools provider to develop a catalogue of formal and platform-independent measure.
- Implement storage solution dedicated to measurements resulting of measure execution in big data context.
- Implement visualization tools to expose the extracted results in an easy-readable fashion, so allowing a quick understanding of the situation and the possible actions that can be taken to improve the diverse stages of the software lifecycle.
- Implement an extension mechanism dedicated to the integration of external analysis tools will provide long terms analysis and predictive evaluations on collected measures.
- Implement an Extended API allowing to facilitate the integration on Measure Platform with external tools and services.

The platform activity is organised around its ability to collect measurement by executing measures defined by the SMM standard. SMM measures are auto-executable component, implemented externally, which can be interrogated by the platform to collect measurements.

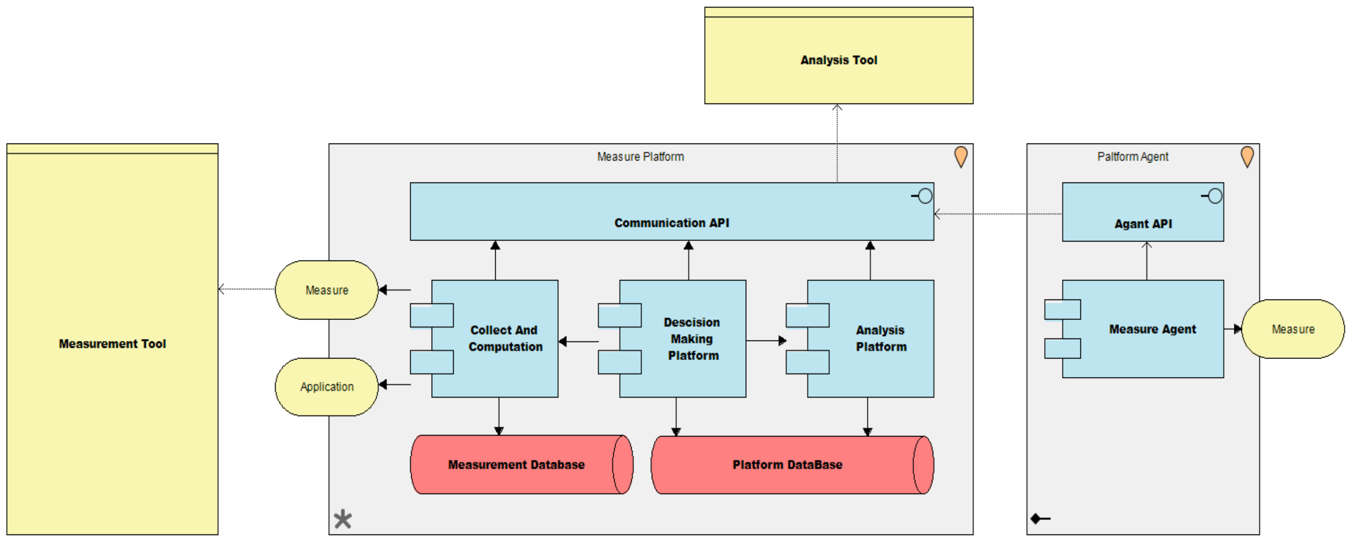


Figure 1. Architecture overview of Measure Platform

Measure Platform: Central component of this deliverable, the Measure platform provides services relate to data collection, analysis and display. It's composed of six sub-components:

Platform Agent: Measure tools on client side which collect data. The executable measure provides a way to collect data in physical world. A measure can be executed on platform side and collect physically this data through an existing measure tool. A Measure can also be directly executed on client side (a computer close to measured element) by the intermediary of a Platform Agent.

Measurement Tools: A Measurement Tool is and external tool which collects or calculates measurement from a specific source. In context of the project, 5 Measurements tools has been developed (see section 3) and a lot of existing tools in the market (such as SonarQube for code quality metrics) can be also considered as Measurement Tools.

Analysis Tools: An Analysis Tool a set of external services which work on the historical measures values in order to provide advanced and valuable analysis function to the platform. In order to support a large set of analyses services and do not limit to it a specific technology, the Analysis Tools are external processes. The analysis tool is integrated to the platform using a specific API. This integration includes embedded visualisation provided by the analysis service into the platform.

Measures: A Measure is a small and autonomous Java program based on the SMM specification which can collect measurements. A Measure can be Direct (Collect of measurement in physical world), a Proxy (Ensure communication between a Measurement Tool and the Platform) or Derived Measure (Measure calculated by the aggregation of existing Measures).

Applications: An Application is a set of Measures aggregated together in order to address a functional requirement. The application is associated with a visual dashboard which is directly integrated into the Decision-Making platform when the Application is deployed on a project.

Hardware and Software Requirements

System	**Linux, Windows**		
Installation Scenario	**Minimum Requirement**	**Standard Configuration (500 Metric Collection, Basic Analysis, 50 Users)**	**Advanced Analysis (+2000 Metrics, All Analysis Services, 200 Users)**
RAM	4 Go	8 Go	16 Go
Processor	32-bit, 4 cores	64-bit, 4 cores	64-bit, 8 cores
Hard Disk	80 GB for system drive	80 GB for system drive	200 GB for system drive

Prerequisite

Install MySQL Database

- Download MySQL Community Server ver. 5.7 or above:

  https://dev.mysql.com/downloads/mysql/

- Install MySQL using these instructions:

  https://dev.mysql.com/doc/refman/5.7/en/installing.html

- Create a new database named "measureplatform".

Using MySQL Command Line Client:

```
CREATE DATABASE measureplatform;
```

Install Elasticsearch

- Download Elasticsearch ver. 6.5.4

  https://www.elastic.co/downloads/elasticsearch

- Unzip the application in your tool directory.

Install Kibana

- Download Kibana ver. 6.5.4

  https://www.elastic.co/downloads/kibana

- Unzip the application in your tool directory

Java 1.8 Installation

- Download and install the jdk8 in your environment:

  http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

Measure Platform Installation

Download

- Download the last released version of the MeasurePlatform

  https://github.com/ITEA3-Measure/MeasurePlatform/releases

- Unzip the platform in your tool directory.

Installation and Configuration

Platform is configured using a property file named **application.properties**. This property file has to be put in the same folder of the measure-platform-1.0.0.jar binary application.

Edit the **application.properties** file:

Table 1 : Local Storage configuration

Property	**Description**	**Default Value / Example**
measureplatform.storage.measure	path of the local storage directory for measures.	/home/.../storage/measures
measureplatform.storage.application	path of the local storage directory for measurement application. .	

Table 2 : Data Base configuration

Property	**Description**	**Default Value**
spring.datasource.url	JDBC URL of the database	jdbc:mysql://localhost/measureplatform
spring.datasource.username	Login of the MySQL database.	
spring.datasource.password	Password of the MySQL database.	
spring.datasource.driver-class-name	Driver JDBC for MySQL	com.mysql.jdbc.Driver

Table 3 : Platform configuration

Property	**Description**	**Default Value**
measure.kibana.api	Ip of the Elasticsearch installation.	localhost:9200
measure.kibana.adress	Ip of the Kibana installation.	localhost:5601
server.port	Port of the MeasurePlatform web application	80

Table 4 : Mail Service Configuration of Measure Platform

Property	Description	Default Value
spring.mail.host	Url of the mail service	smtp.gmail.com
spring.mail.port	Port of the mail service	587
spring.mail.username	Login of the mail account	
spring.mail.password	Password of the mail account	
spring.mail.protocol	mail protocole	smtp
spring.mail.tls		true
spring.mail.properties.mail.smtp.auth		true
spring.mail.properties.mail.smtp.starttls.enable		true
spring.mail.properties.mail.smtp.ssl.trust		smtp.gmail.com

Example of **application.properties** file:

```
##Local Storage
measureplatform.storage.measure=/home/.../storage/measures
measureplatform.storage.application=/home/.../storage/measures

##DataBase Configuration
spring.datasource.url=jdbc:mysql://localhost:3306/measureplatform
spring.datasource.username=
spring.datasource.password=
spring.datasource.driver-class-name=com.mysql.jdbc.Driver

##Platform Configuration
server.port=80
measure.kibana.adress=localhost:5601
measure.kibana.api=localhost:9200

##MailService Configuration
spring.mail.username=
spring.mail.password=
spring.mail.host=smtp.gmail.com
spring.mail.port=587
spring.mail.protocol=smtp
spring.mail.tls=true
spring.mail.properties.mail.smtp.auth=true
spring.mail.properties.mail.smtp.starttls.enable=true
spring.mail.properties.mail.smtp.ssl.trust=smtp.gmail.com
```

Running Measure Platform

1. **Start MySQL**

2. **Start Elasticsearch**

```
./elasticsearch-6.5.4/bin/elasticsearch
```

3. **Start Kibana**

```
./kibana-6.5.40/bin/kibana
```

4. **Start the Measure platform**

```
java -jar measure-platform-{version}.war
```

To access to the Platform: *http://localhost:80/#/*

Connection using the default Platform Account

At deployment, the Measure Platform is created with 1 default administrator account:

Administrator account	
Login	admin
Password	admin

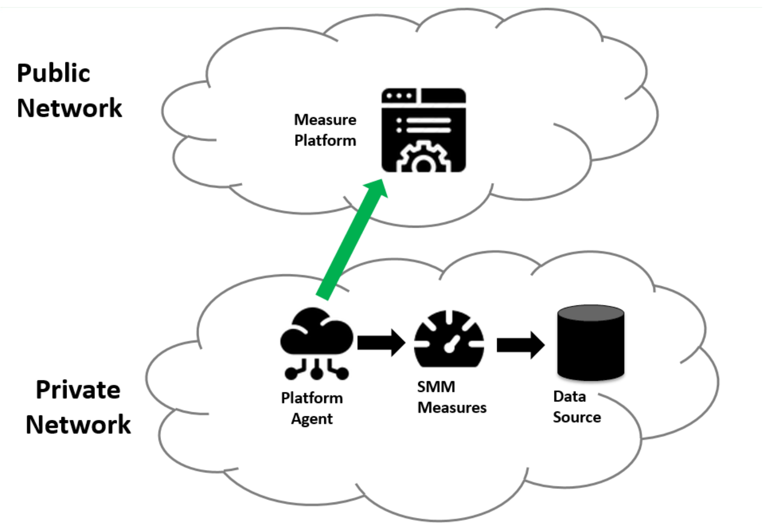
It is highly recommended to change this password during the first launch of the platform

Measure Agent Installation

The Platform Agent in an autonomous application connected to the Measure Platform via the Communication API. It allows to deploy and execute measures on client side and the configuration of measure execution remains administrated remotely by the platform.

In most of the cases, measures are executed on platform side and they manage a connection with external data sources to collect required data. But it's not always possible to execute measures on platform for various reasons like network configuration, security policy, and scalability or because of the mechanisms used by the measure to collect data. To address this issue, we have developed an autonomous agent which provides the following services:

- Allow to execute direct measure on client side, closer of measured element instead of executing it on platform side.
- Ensure the scalability of the Measure platform by providing a way to execute time-consuming measures on remote computers while maintaining a centralized control of measure's execution.
- Solve common issues related to Network Security rules in Industrial context by allowing to deploy the Measure Platform different networks than where the monitored data sources is stored.



Download

- Download the last packaging of the Agent:

  https://github.com/ITEA3-Measure/MeasureAgent/releases

- Unzip the platform in your tool directory.

Installation and Configuration

Platform is configured using a property file named **application.properties**. This property file has to be put in the same folder of the measure-platform-1.0.0.jar binary application.

Edit the **application.properties** file:

Property	**Description**	**Example**
measure.repository.path	Path of local directory containing measures	C:/work/MEASURE/Agent/storage
measure.server.adress	IP of MeasurePlatform Server	localhost:80
measure.agent.name	Name of the Agent (*Must be Unique*)	Agent1

Example of **application.properties** file:

```
measure.repository.path=/home/measure/storage
measure.server.adress=xxx.xxx.xxx.xxx/measure
measure.agent.name= MyAgent
```

Deploy Measure on Agent

The Measure which can be executed by the agent must be manually deployed. During the configuration, you have defined the path of the storage folder, a directory which will contain the measures deployed on your agent.

- Unzip the packaged Measure
- Copy the unzipped Measure on storage folder.

Expected Directory Organisation Example:

```
/agent/storage
/agent/storage/MyMeasure
/agent/storage/MyMeasure/MyMeasure-1.0.0.jar
/agent/storage/MyMeasure/MeasureMetadata.xml
/agent/storage/MyMeasure/lib
```

Running Measure Platform Agent

```
java -jar measure-agent-1.0.0.jar
```

Visualise Agent on Server Side

When an agent is started, it is automatically registered on the Measure Platform. The list of agents actually registered on the Measure Platform can be consulted on the "Platform > Remote Agent" page.

Visualise Available Measure

The measures deployed on agents can be visualised on the Measure Catalogue page like other measures deployed on server. The Host indicates the name of the agent which provides this measure.

Instantiate Client-Side Measure

Like for server-side measures, you have to create an instance of a client-side measure in order to collect it. The only differences to a server-side measure is that, at this time it is not possible to execute once a client-side measure. The client-side measure instance has to be scheduled.