D3.2.4

Smart Lighting System Final Prototype

Project number:	257243
Project acronym:	TClouds
Project title:	Trustworthy Clouds - Privacy and Resilience for Internet-scale Critical Infrastructure
Start date of the project:	1st October, 2010
Duration:	36 months
Programme:	FP7 IP

Deliverable type:	Prototype
Deliverable reference number:	ICT-257243 / D3.2.4 / 1.0
Activity and Work package contributing to the deliverable:	Activity 3 / WP 3.2
Due date:	30th September 2013 – M36
Actual submission date:	30th September 2013

Responsible organisation:	EFACEC ENG
Editor:	Paulo Santos
Dissemination level:	Public
Revision:	1.0

Abstract:	This document describes how to install, configure and test the integration environment between the Smart Lighting System and TClouds security component BFT-SMaRt
Keywords:	Smart Lighting, SMR, H2 database

Editor

Paulo Santos (EFACEC ENG)

Contributors

Paulo Viegas, Paulo Santos (EFACEC ENG)

Executive Summary

This document describes the installation and configuration of the integration between the Smart Lighting System (SLS) and TClouds security component BFT-SMaRt.

Contents

Chapter 1

Introduction

This document describes the process of installing and configuring the runtime integration environment between the SLS and BFT-SMaRt.

There is also the installation and configuration of a runtime that validates all the steps described on *Integration_4* and *Integration_6* validation activities of D3.3.4 - Final Report On Evaluation Activities[5].

All the supplied software can run in any Windows/Linux server as long as they comply with minimum requirements described in each chapter.

The installation procedure of the several modules refers to a `<WORKING_FOLDER>` as the folder where the installation package for the Smart Lighting and validation modules is unpacked to. The installation package containing the BFT-SMaRt is referred as `<BFT_FOLDER>`.

A Linux operating system is assumed.

Chapter 2

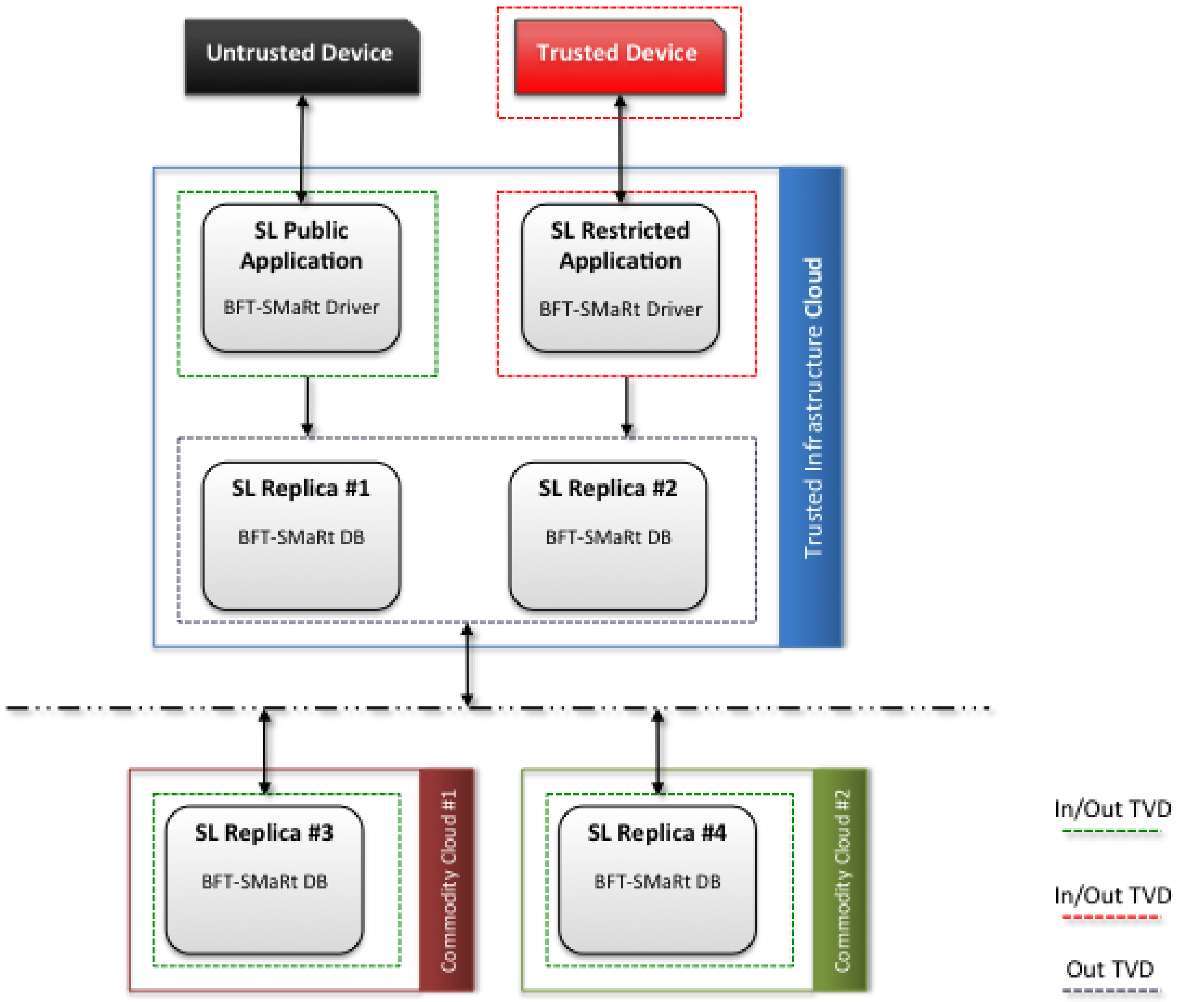
Architecture

Two instances of the Smart Lighting Web application and several BFT-SMaRt nodes constitute the integration environment. Each BFT-SMaRt node has a H2 database instance for data persistence and a BFT-SMaRt driver responsible for all communication and resilience logic between the nodes and their local H2 database.

To enhance security as a whole, one Smart Lighting Web application is only available with view only privileges in mind (Client users), and the other protected by a secure TVD to allow operation/administration privileges (Operator and Administrator users).

All elements can run locally on one cloud, or the replica nodes distributed through several other clouds, thus enhancing redundancy.

The following picture depicts this environment:



Chapter 3

BFT-SMaRt

This chapter describes the installation, configuration and runtime organization of a replica node with BFT-SMaRt.

3.1.1 *Minimum Requirements*

- 2GB of available RAM
- 100MB of free disk space (for demonstration purposes)
- Java™ 1.7 Runtime[1] or higher

3.1.2 *Installation*

Extract `<BFT_FOLDER>/smr` to the machine that will be used as a node server.

Inside the newly created `smr` directory execute:

```
chmod +x *.sh
```

With the **root** user, create the file `/etc/profile.d/smr.sh` with:

```
export REPLICA=0

export SMR_HOME=/home/tclouds/smr
```

Set REPLICA to the correct replica number ID.

Set the SMR_HOME to where you extracted the smr.tar.gz concatenated with /smr.

For example, if you extracted to /home/tclouds the SMR_HOME will be set to `/home/tclouds/smr`

Configure the nodes participating in the SMR at `<SMR_HOME>/config/hosts.config`.

3.1.3 *Running*

Open a command line shell positioned at `<SMR_HOME>` and execute the following command:

```
./startAll.sh
```

Inside `<SMR_HOME>` directory there are a few scripts to automate some repetitive tasks. The following is a brief description of them.

stopAll.sh	Kill the SMR replica process and the underlying H2 database
copyDBs.sh	Resets the H2 database
startAll.sh	Starts the SMR replica process and the underlying H2 database

3.1.4 Runtime

The directory structure of the SMR binaries is represented by the following tree:

```
\---smr
    |              // start/stop scripts and main binary file
    +---config     // configuration files
    +---data       // underlying database
    \---lib        // jar files dependencies
```

Chapter 4

Smart Lighting Web Application

This chapter describes the installation, configuration and runtime organization of the Smart Lighting Web application.

4.1.1 Requirements

- 2GB of available RAM
- 100MB of free disk space
- Java™ 1.7 Runtime[1]
- Apache Tomcat 7.0[2]

4.1.2 Installation

With the **root** user, create the file `/etc/profile.d/tomcat.sh` with:

```
export CATALINA_HOME=/opt/apache-tomcat-7.0.41

export SLCONFIG=/home/tomcat/smartlighting

export REPLICA=X
```

With the **tomcat** user, create the file `/opt/apache-tomcat-7.0.41/bin/setenv.sh` with:

```
CATALINA_OPTS="-Dsmartlighting.folder=$SLCONFIG -
Djavax.net.ssl.trustStore=$SLCONFIG/tclouds.jts -
Djavax.net.ssl.trustStorePassword=efacec -Duser.timezone=UTC -
Ddivdb.folder=$SLCONFIG -Ddivdb.firstclient=10000"

export CATALINA_OPTS
```

and execute:

```
chmod +x /opt/apache-tomcat-7.0.41/bin/setenv.sh
```

If several servers are installed with this web application, the value of `divdb.firstclient` must be configured accordingly.

With the **root** user, create the file `/etc/rc.d/init.d/tomcat` with:

```
#!/bin/bash

# description: Tomcat Start Stop Restart

# processname: tomcat
# chkconfig: 234 20 80
JAVA_HOME=/usr/java/jdk1.7.0_25
export JAVA_HOME
PATH=$JAVA_HOME/bin:$PATH
export PATH

. /etc/profile.d/tomcat.sh

case $1 in
start)
cd $CATALINA_HOME/bin
/bin/su tomcat ./startup.sh
;;
stop)
/bin/su tomcat $CATALINA_HOME/bin/shutdown.sh
;;
restart)
/bin/su tomcat $CATALINA_HOME/bin/shutdown.sh
/bin/su tomcat $CATALINA_HOME/bin/startup.sh
;;
esac
exit 0
```

and execute:

```
chmod +x /etc/rc.d/init.d/tomcat
```

Copy the `<WORKING_FOLDER>/config` directory to the directory defined by the variable `SLCONFIG`.

Open the file `<SLCONFIG>/application.properties` and change the properties according to the installation environment. The properties are self explanatory.

The most important properties are the ones that define the access to the SMR and `system.onlyClients`. The latter if true, only Client Users are allowed, otherwise the application is assumed to be running in a secure TVD environment allowing only user with operation/administration privileges (Operator and Administrator users).

Configure the nodes of the SMR at `<SLCONFIG>/hosts.config`.

Copy the file `<WORKING_FOLDER>/smartlighting.war` to `<TOMCAT_DIR>/webapps`

Reboot the application server.

4.1.3 Running

To start the tomcat service execute

```
sudo service tomcat start
```

To stop the tomcat service execute

```
sudo service tomcat stop
```

4.1.4 Runtime

After the tomcat service starts up there will be a folder in `<TOMCAT_DIR>/webapps` with the name `smartligthing` witch holds the application. This folder follows the specification for J2EE 6 Web Module[4].

```
\---smartlighting
    +---css             // css files and supporting files
    +---images          // application images
    +---jQuery          // jQuery related javascript
    +---js              // several javascript libraries
    +---META-INF
    \---WEB-INF
        +---classes     // java compiled classes, html and other resources
(by default wicket has them together)
        +---config      // spring configuration files
        +---dynimg      // images generated at start up
        +---lib
        \---reports     // jasper reports
```

Chapter 5

Validation Tests

This chapter describes the installation, configuration and runtime organization of the validation tests.

5.1.1 Minimum Requirements

- 2GB of available RAM
- 100MB of free disk space (for demonstration purposes)
- Java™ 1.7 Runtime[1] or higher

5.1.2 Installation

Extract `<WORKING_FOLDER>/validation` to the same machine where is located the web application.

Inside the newly created `validation` directory execute:

```
chmod +x *.sh
```

If needed, edit `<VALIDATION_DIR>/startValidation.sh` and change the value of the runtime variables `divdb.folder` and `divdb.firstclient` to reflect the execution environment. This validation execution will use the same configuration defined for the web application.

5.1.3 Running

To execute the validation it is necessary to pass a property file with the execution parameters. These property files represent the configuration needed to evaluate the steps of *Integration_4* and *Integration_6* validation activities of D3.3.4[5].

The name of each file indicates the step that it refers to. The name format is:

```
<section>-step<step number>-<number of sessions>sessions.properties
```

For example `Integration6-step4-20sessions.properties` has the properties to run *Integration_6*, step 4 with 20 simultaneous sessions.

Although the majority of the properties are self explanatory and have accompanying comments, there are some that need a more detailed explanation, such as *warmup* and *compromise.**

The *warmup* property defines the number of interactions to exercise java for better performance. Only makes sense for steps in *Integration_6*, since they are the only ones with performance concerns.

The *compromise.** properties, if present, define the underlying database of a node to compromise and what SQL to use to do so. This “attack” is made, for each session, after the modification of the schedules.

Before executing the validation make sure that all nodes of the SMR are up and running with a clean database.

To run the validation tests, open a command line shell positioned at `validation` directory and execute the following command:

```
./startValidation.sh <property file>
```

5.1.4 Runtime

The directory structure of Validation tests binaries is represented by the following tree:

```
\---validation
    |                      // configuration, start/stop scripts
    |                      // and main binary file
    \---validation-lib     // jar files dependencies
```

Chapter 6

List of Abbreviations

BFT-SMaRt	Byzantine Fault Tolerant State Machine Replication
J2EE	Java 2 Enterprise Edition
SLS	Smart Lighting System
SMR	State Machine Replication

Chapter 7

References

1. http://www.oracle.com/technetwork/java/javase/downloads/jre7u7-downloads-1836441.html
2. http://tomcat.apache.org/download-70.cgi
3. http://h2database.com/html/main.html
4. http://docs.oracle.com/javaee/6/tutorial/doc/bnadx.html
5. D3.3.4 - Final Report On Evaluation Activities