|  |  |  |
| --- | --- | --- |
|  | Scalable Data Analytics,  Scalable Algorithms, Software Frameworks  and Visualization ICT-2013 4.2.a | |
| |  |  | | --- | --- | | Project | **FP6-619435/SPEEDD** | | Deliverable | **D6.3** | | Distribution | **Public** | | |  |
|  | |  |
|  | |  |



http://speedd-project.eu

**The Architecture Design of the SPEEDD Prototype**

Alex Kofman (IBM)

Status: Draft (Version 0.1)

January 2015

|  |  |
| --- | --- |
| **Project** |  |
| Project Ref. no | FP7-619435 |
| Project acronym | SPEEDD |
| Project full title | Scalable ProactivE Event-Driven Decision Making |
| Project site | http://speedd-project.eu/ |
| Project start | February 2014 |
| Project duration | 3 years |
| EC Project Officer | Aleksandra Wesolowska |
|  |  |
| **Deliverable** |  |
| Deliverable type | Report |
| Distribution level | Public |
| Deliverable Number | D6.3 |
| Deliverable Title | First Integrated Prototype |
| Contractual date of delivery | M12 (January 2015) |
| Actual date of delivery | January 2015 |
| Relevant Task(s) | WP6/Tasks 6.3 |
| Partner Responsible | IBM |
| Other contributors | NCSR, CNRS, Feedzai, ETH, UoB |
| Number of pages | 42 |
| Author(s) | Alex Kofman (IBM) |
| Internal Reviewers | Feedzai |
| Status & version | Draft |
| Keywords | architecture design scalability cep decision-making proactive |
|  |  |

|  |
| --- |
| **Executive Summary** |

SPEEDD (Scalable ProactivE Event-Driven Decision making) will develop a system for proactive event-based decision-making: decisions will be triggered by forecasting events – whether they correspond to problems or opportunities – instead of reacting to them once they happen. The decisions and actions will be real-time, in the sense that they will be taken under tight time constraints, and require on-the-fly processing of “Big Data”, i.e. extremely large amounts of noisy data storming from different geographical locations as well as historical data.

The goals of WP6 (Scalability and System Integration) are to develop a highly scalable event processing infrastructure supporting real-time event delivery and communication minimization, and implement integration of the SPEEDD components into a prototype for proactive event-based decision support.

The purpose of this document is to describe the first integrated prototype and provide instructions for installing and running it.

The main goal achieved in the first integrated prototype is establishing an initial implementation of the event-driven architecture with focus on the integration and event-driven paradigm.

In addition, two demo scenarios have been prepared for traffic management and credit card fraud detection use cases to validate basic assumption and viability of the proposed design end to end.

|  |
| --- |
| **Contents** |

[1 Introduction 5](#_Toc409388935)

[1.1 History of the document 5](#_Toc409388936)

[1.2 Purpose and Scope of the Document 5](#_Toc409388937)

[1.3 Relationship with Other Documents 5](#_Toc409388938)

[1.4 Demo Scenarios 5](#_Toc409388939)

[2 Description of the Integrated Prototype 6](#_Toc409388940)

[3 Install and Execution Instructions 10](#_Toc409388941)

[3.1 Overview 10](#_Toc409388942)

[3.2 Running a virtual machine from the image 10](#_Toc409388943)

[3.2.1 Prerequisite services: Storm and Kafka 11](#_Toc409388944)

[3.3 Building from sources 11](#_Toc409388945)

[3.4 Running demo scenarios 11](#_Toc409388946)

[3.4.1 Traffic Management 11](#_Toc409388947)

[3.4.2 Credit Card Fraud Detection 11](#_Toc409388948)

[3.4.3 Stopping SPEEDD prototype 12](#_Toc409388949)

# Introduction

## History of the document

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Change Description** |
| 0.1 | 15/01/2015 | Alexander Kofman (IBM) | First draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Purpose and Scope of the Document

The purpose of this document is to describe the first integrated prototype and provide instructions for installing and running it.

## Relationship with Other Documents

The current document refers to the system requirements for the Proactive Traffic Management use case described in D8.1 and for the Proactive Credit Card Fraud Management described in D7.1.

## Demo Scenarios

For demonstration purposes, demo scenarios were prepared for the traffic management and for the credit card fraud detection use cases. The scenarios are available in a form of storyboards [here – TBD – URL to where the scenario presentations can be downloaded].

# Description of the Integrated Prototype

The diagram in Figure ‎2.1 presents the architecture of the SPEEDD prototype for the traffic management use case (as it is described in details in the Architecture Design Document D6.1).

The first version of the integrated prototype provides partial implementation of the architecture (see Figure ‎2.2). The input events are replayed from a csv file. The derived and predicted events are processed by the Decision Maker component and are displayed in the dashboard. The current release does not include connection to the micro-simulator; therefore automatic actions are not implemented.

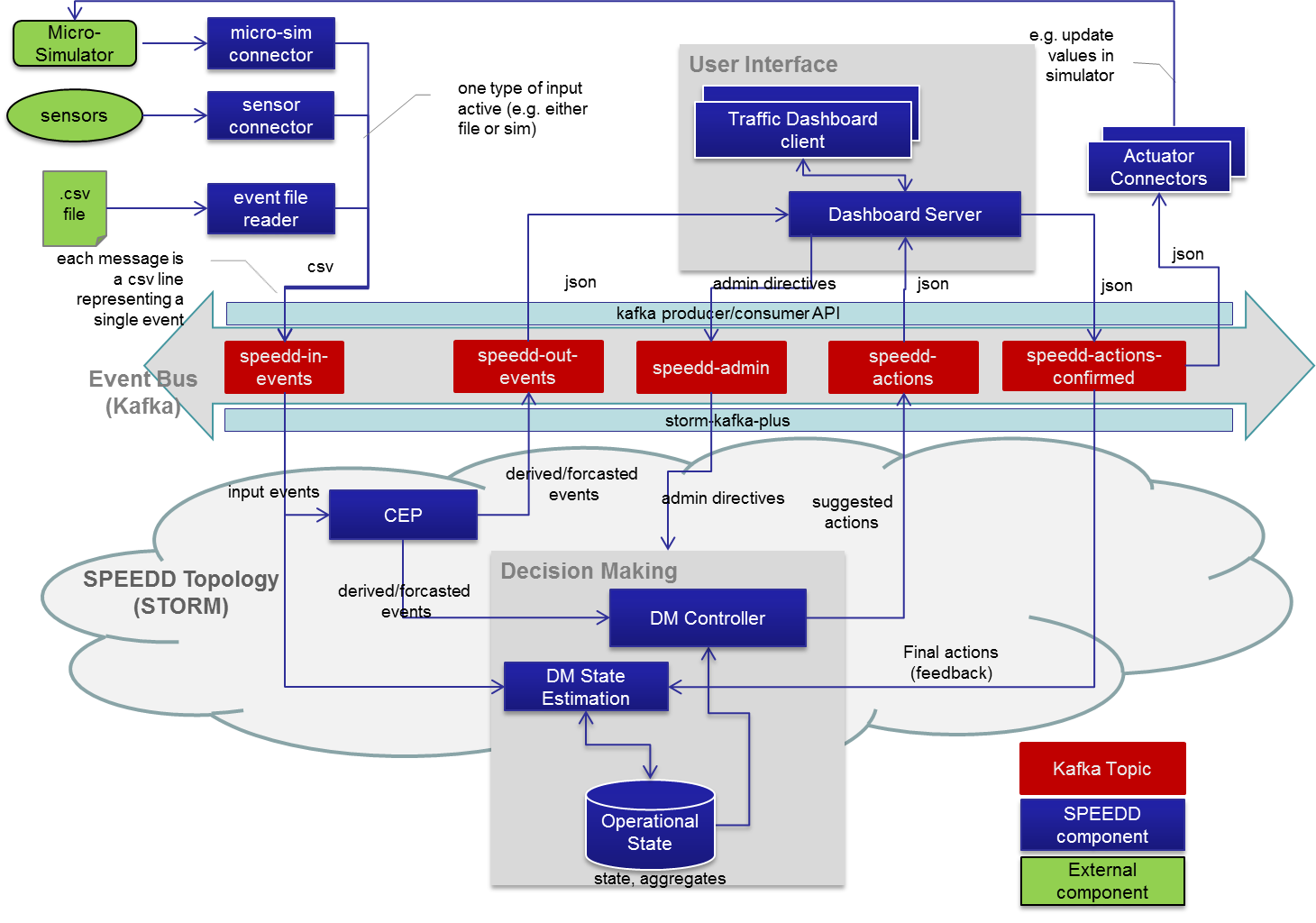


Figure ‎2.1 - SPEEDD Runtime - Event-Driven Architecture (Traffic Use Case)

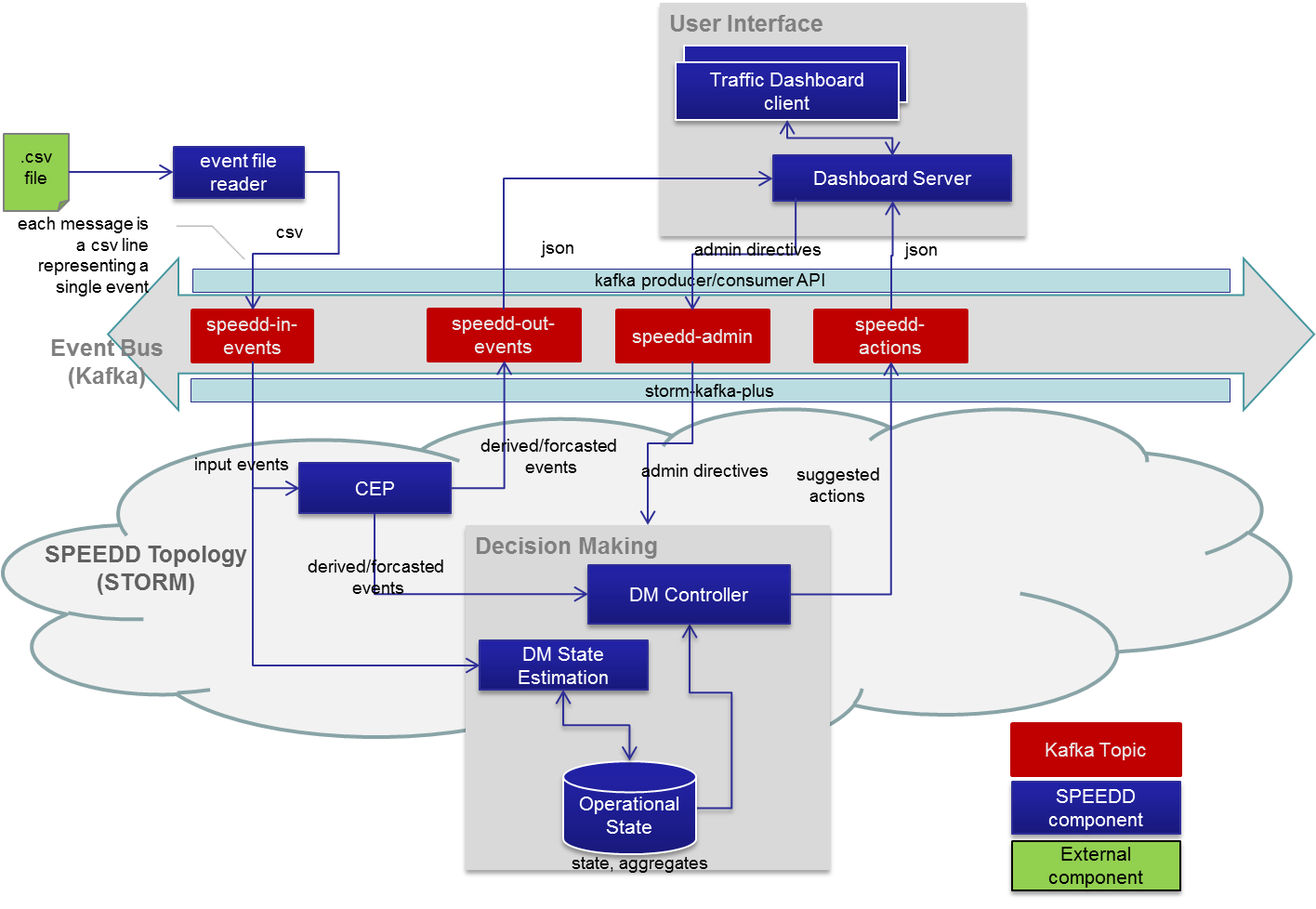


Figure ‎2.2 - SPEEDD Runtime Architecture for Traffic Management Use Case (as implemented in v1)

The implemented architecture for the credit card fraud detection use case is represented in Figure ‎2.3. The input events representing credit card transactions are replayed from a csv file. The events are processed by the CEP component which detects important trends and suspicious transactions. The dashboard presents the current operational situation to the operator, indicating the trends and situations requiring operator’s attention. The Decision Maker is not currently implemented as a separate component (the implementation will be addressed in the next prototype release).

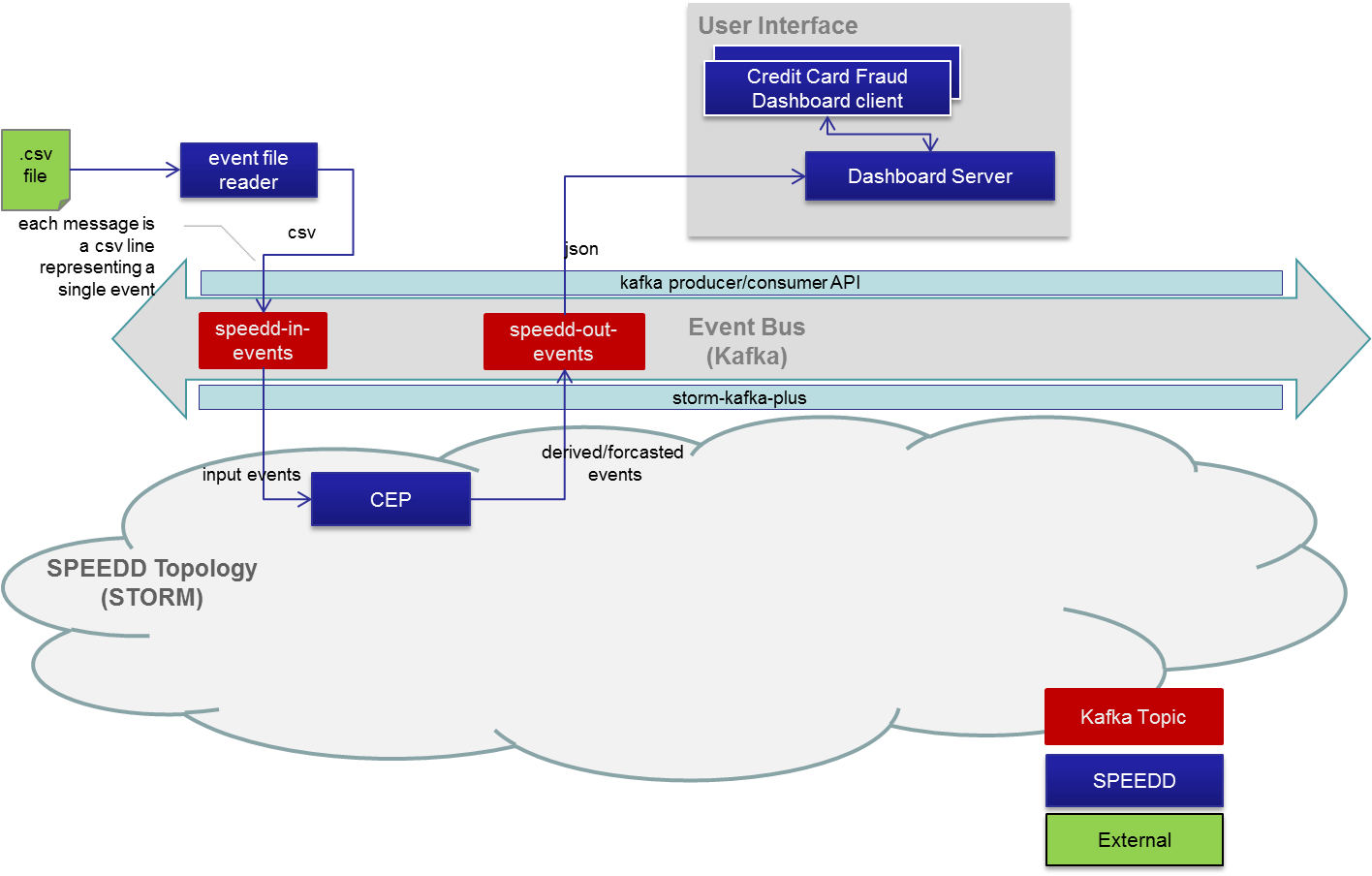


Figure ‎2.3 - SPEEDD Runtime Architecture for Credit Card Use Case (as implemented in v1)

The screenshot in Figure ‎2.4 presents a snapshot of the dashboard displayed to the operator. The derived and predicted events (e.g. predicted congestion) are displayed on the map view. User can click on the circle representing the congestion to see the event details (see Figure ‎2.5).

The decision maker component currently implements a single type of automatic action – setting a ramp metering rate. The history of rates is displayed on the “sensor data” view (top-left view in Figure ‎2.4). This view presents history of metering rate changes vs. traffic density values for a single ramp selected in the “ramp metering” view (the bottom-left view in Figure ‎2.4).

A ramp can be under fully automatic control (all the metering rate values are set by the automatic decision maker), partially controlled by the user (the user can set upper and lower limits for the ramp rate), or fully controlled by the user (all the rates are set by the user). The mode of control is managed using the “ramp metering control” view (central bottom view in Figure ‎2.4). The colors of the points in the “sensor data” view indicate the mode of control: blue – fully automatic, yellow – partial, green – full user control.

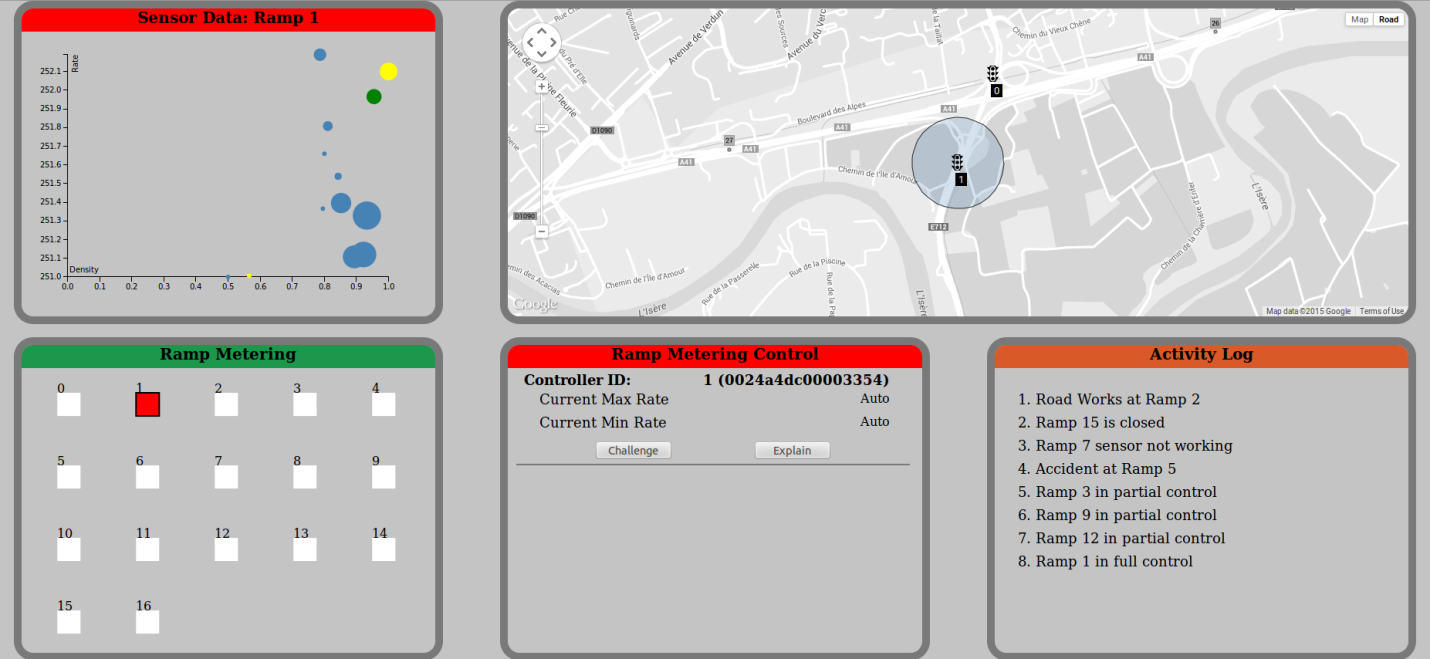


Figure ‎2.4 - Dashboard UI for Traffic Management Use Case

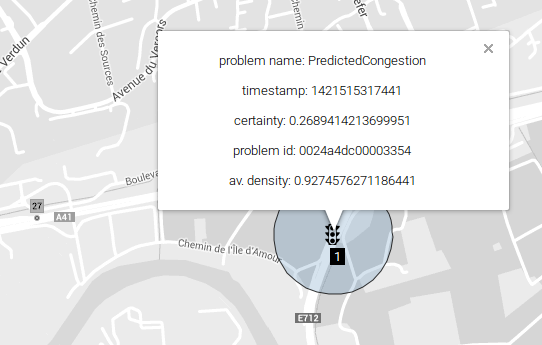


Figure ‎2.5 - Traffic Management Dashboard - Details of the Predicted Congestion

The screenshot in the Figure ‎2.6 represents a snapshot of the credit card management dashboard. The top bar shows transaction statistics. Clicking on any area in that bar renders the tree map in the center according to the selected criteria (e.g. “average transaction amount”).

The “Flagged Transaction Queue” view shows the list of detected situations that represent a trend or a suspected fraud attempt. The current version of the SPEEDD prototype detects two types of patterns – “Increasing Amounts” and “Fraud at ATM” (refer to D3.1 for detailed description of these patterns and discussion of the implementation).

Selection of an item in the “Flagged Transaction Queue” shows the details of the selected pattern in the “Selection Info” view.

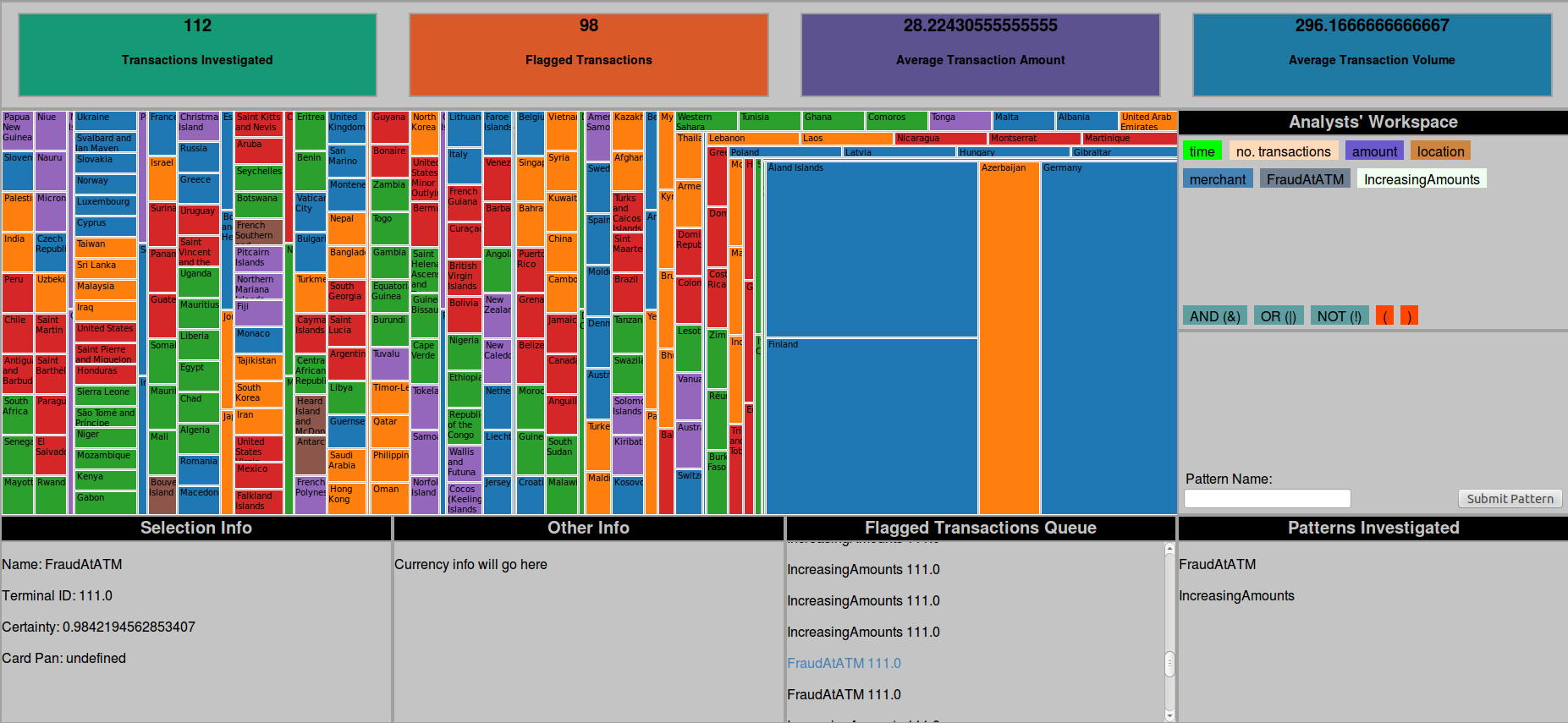


Figure ‎2.6 - Credit Card Fraud Detection Dashboard

# Install and Execution Instructions

## Overview

There are two options for running the first version of the SPEEDD prototype:

1. running a virtual machine from the image provided as part of this deliverable
2. building the prototype from source code and running on any machine that satisfies the system and software prerequisites (see )

In the following we provide instructions for each option.

## Running a virtual machine from the image

The virtual machine image is available for download on the project’s FTP server: [sftp://speedduser@143.233.226.91/VMs/speedd-dev-v0.2.ova](http://speedduser@143.233.226.91/VMs/speedd-dev-v0.2.ova).

The VM image was produced and tested on Oracle VM Virtual Box software v4.3[[1]](#footnote-1).

Create a new virtual machine using the provided VM image by importing the virtual appliance file[[2]](#footnote-2).

Use the following credentials to log into the VM:

Username: livlab

Password: livlab

After starting and logging into the VM follow the instructions for running the demo.

### Prerequisite services: Storm and Kafka

Before running demos make sure that the prerequisite services are up and running. SPEEDD prototype depends on Kafka and Storm platforms. Run the following commands in a terminal shell to validate that the corresponding services are up:

* service kafka-server status
* service storm-nimbus status
* service storm-supervisor status

In case any of the above service is not running, run the following commands in the terminal shell to start the corresponding service:

* sudo service kafka-server start
* sudo service storm-nimbus start
* sudo service storm-nimbus start

## Building from sources

The source code of the project is available here: <https://github.com/speedd-project/speedd>. To build and run the prototype follow instructions in the README file.

## Running demo scenarios

In the following sections we assume that your current working directory is 'speedd-runtime'.

### Traffic Management

1. cd ./traffic
2. sudo ./start-speedd-runtime
3. Run the dashboard by starting ../../speedd-ui/bin/run.sh
4. start sending input events by running ./playevents-traffic

### Credit Card Fraud Detection

1. cd ./ccf
2. sudo ./start-speedd-runtime
3. Run the dashboard by starting ../../speedd\_ui\_bf/bin/run.sh
4. start sending input events by running ./playevents-fraud

### Stopping SPEEDD prototype

1. Kill the topology by running 'sudo scripts/kill-speedd-runtime'
2. Stop the UI by killing the process (or Ctrl-C in the terminal shell where the UI process has been started)

***Note: It is important to stop the prototype running current demo scenario before running another demo scenario (for example, if the traffic management demo is running, it is important to stop it before running the credit card fraud demo).***

1. https://www.virtualbox.org/ [↑](#footnote-ref-1)
2. <https://www.virtualbox.org/manual/ch01.html#ovf> – section in the virtualbox user manual with description of the import process [↑](#footnote-ref-2)