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

First Integrated Prototype

Alex Kofman; Anastasios Skarlatidis; Anton Andreev; Elias Alevizos; Fabiana Fournier; Inna Skarbovsky; Ivo Correia; Marius Schmitt; Natan Morar; Rohit Singhal

Status: Final (Version 1.0)

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	Alina Lupu

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	18
Author(s)	Alex Kofman; Anastasios Skarlatidis; Anton Andreev; Elias Alevizos; Fabiana Fournier; Inna Skarbovsky; Ivo Correia; Marius Schmitt; Natan Morar; Rohit Singhal
Internal Reviewers	Feedzai
Status & version	Final, 1.0
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 management use cases to validate basic assumption and viability of the proposed design end to end.

Contents

List of Figures

1 Introduction

1.1 History of the document

Version	Date	Author	Change Description
0.1	15/01/2015	Alexander Kofman (IBM)	First draft
1.0	29/01/2015	Alexander Kofman (IBM)	Incorporate internal review comments

1.2 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.

1.3 Relationship with Other Documents

The architecture of the SPEEDD software is described in details in the Architecture Design document (D6.1). 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.

1.4 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: http://www.speedd-project.eu/sites/default/files/credit_card_fraud_demo_v2.pdf, http://www.speedd-project.eu/sites/default/files/traffic_management_demo.pdf.

2 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.

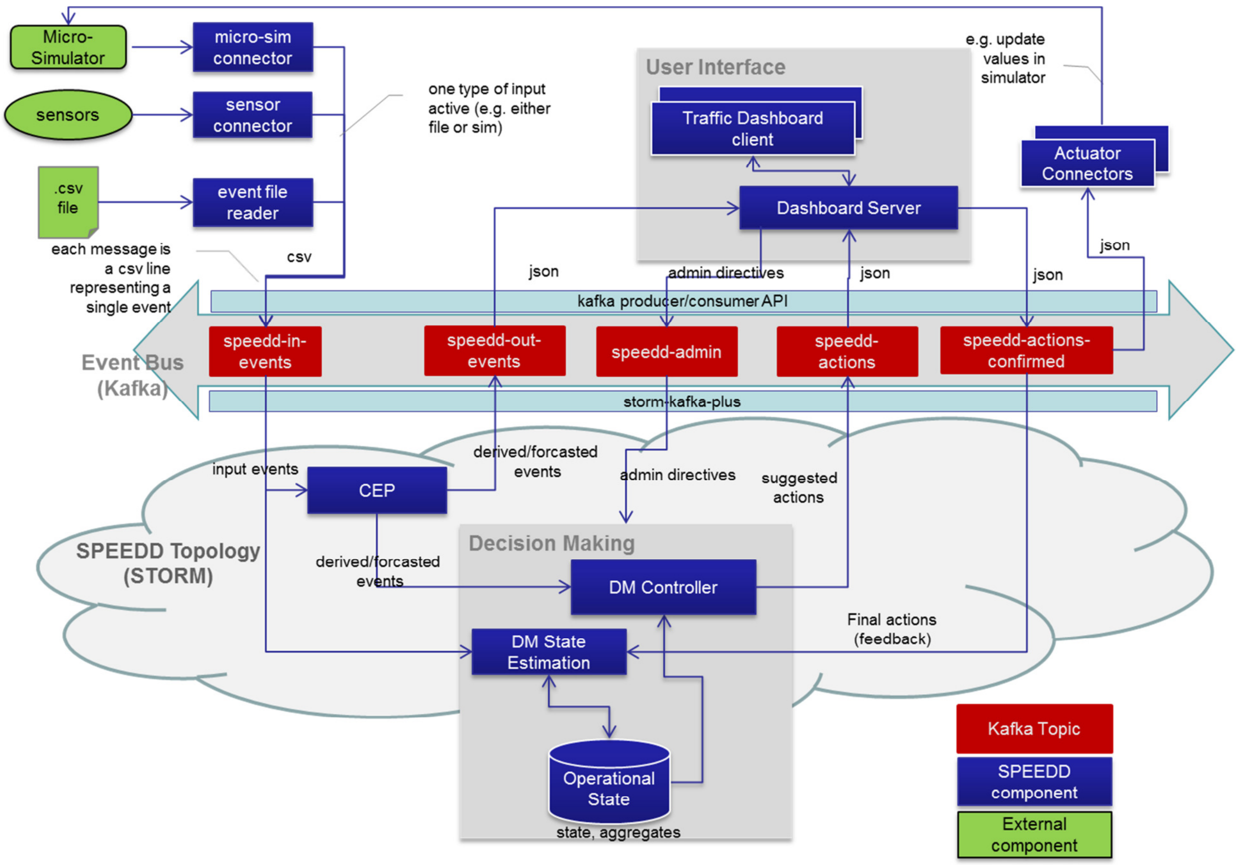


Figure 2.1 - SPEEDD Runtime - Event-Driven Architecture (Traffic Use Case)

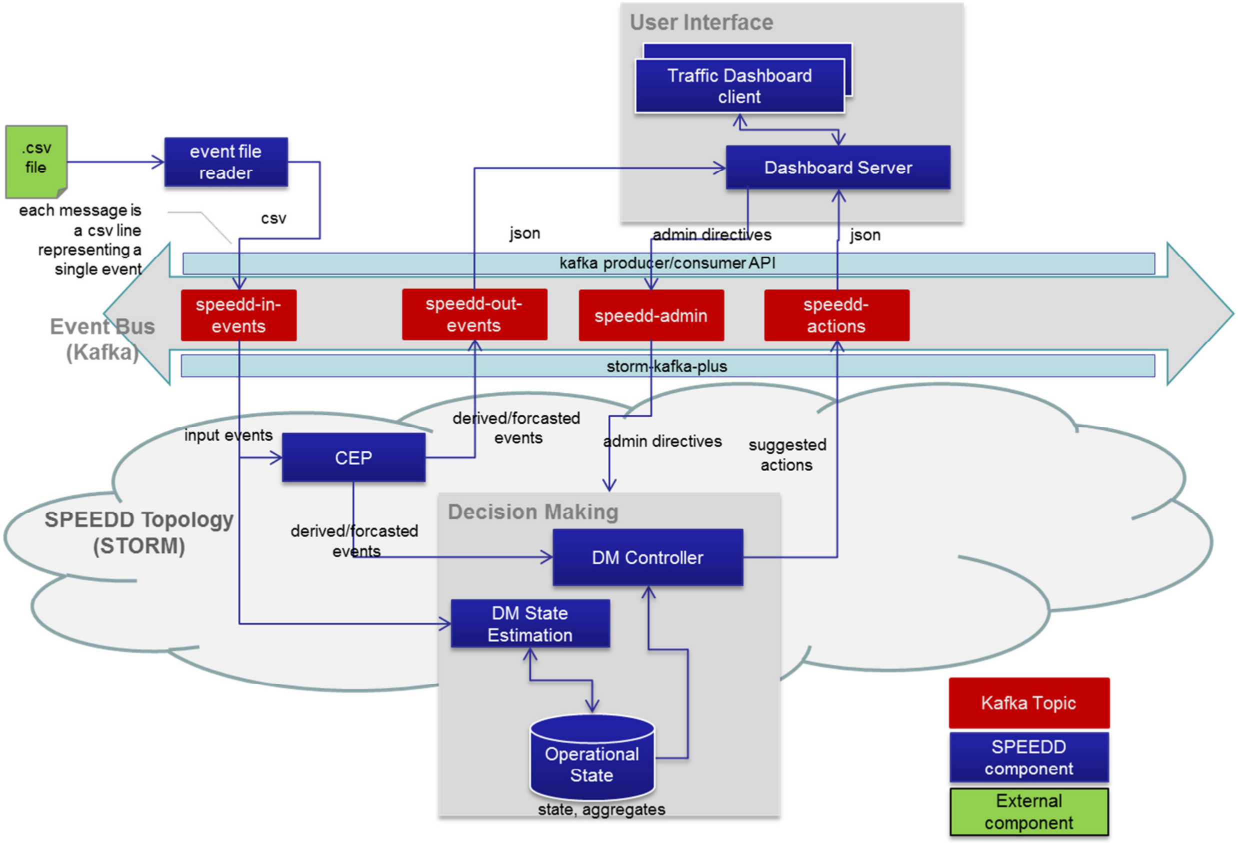


Figure 2.2 - SPEEDD Runtime Architecture for Traffic Management Use Case (as implemented in v1)

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).

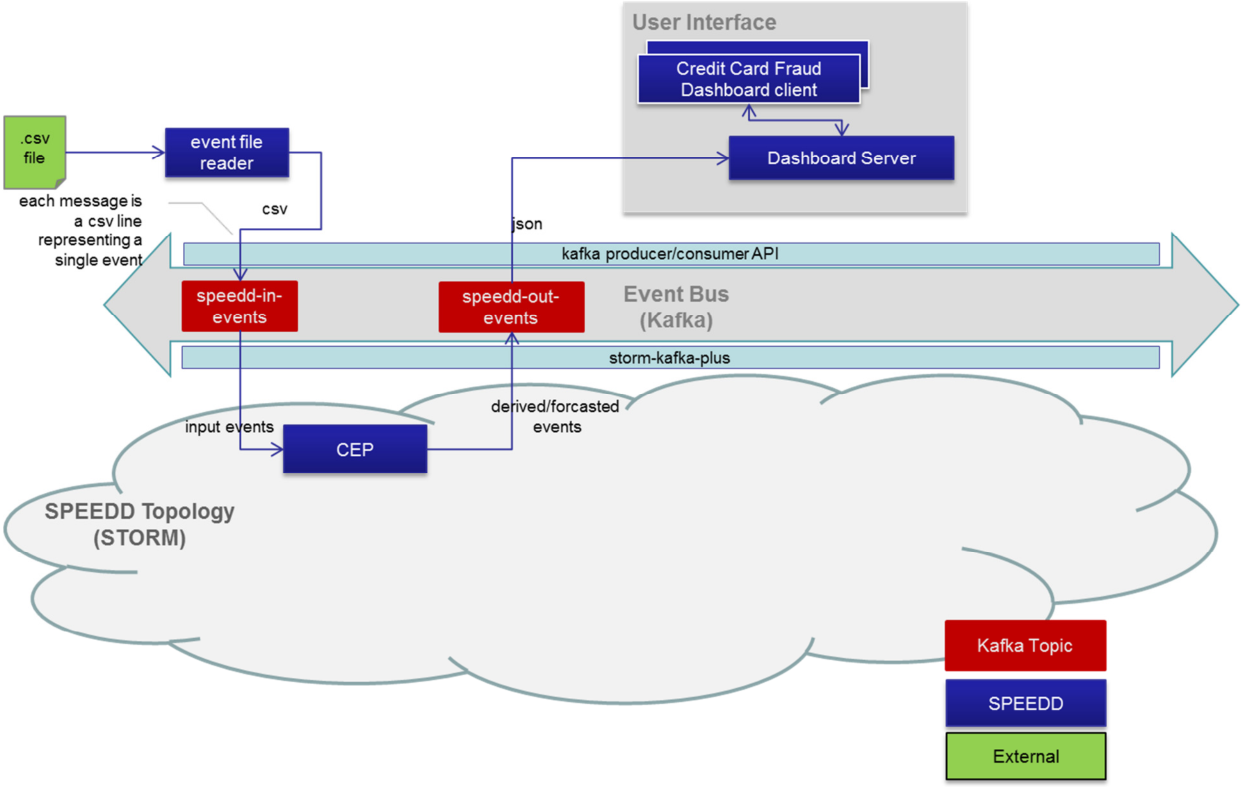
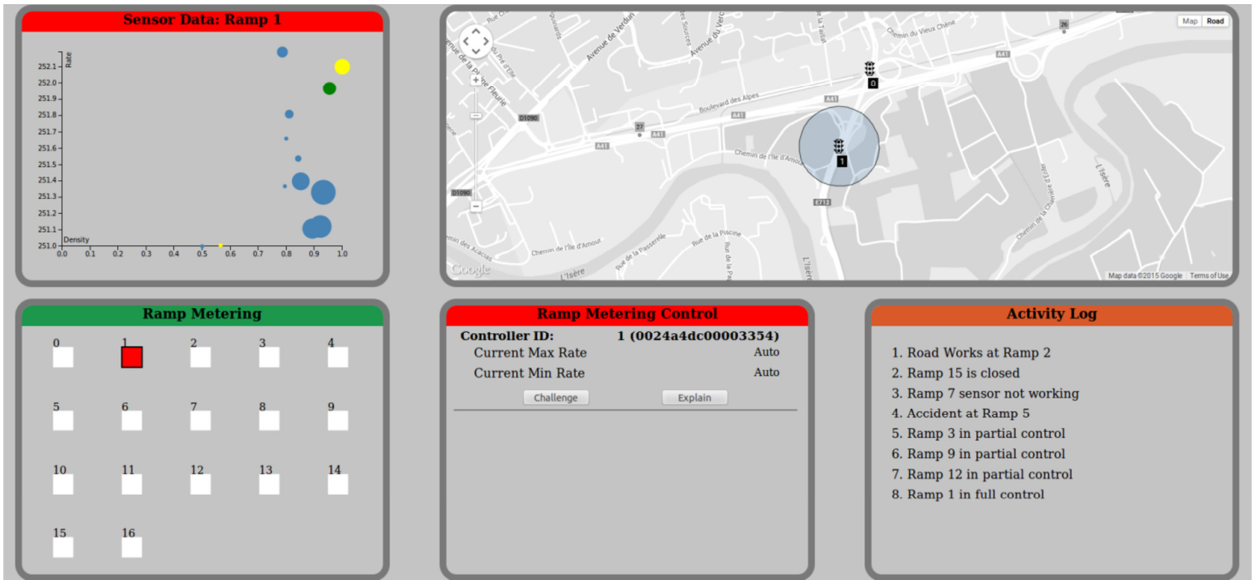


Figure 2.3 - SPEEDD Runtime Architecture for Credit Card Use Case (as implemented in v1)

The screenshot in Figure 2.4 presents a snapshot of the traffic management 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. For a more detailed overview of the traffic management dashboard see chapter 4.



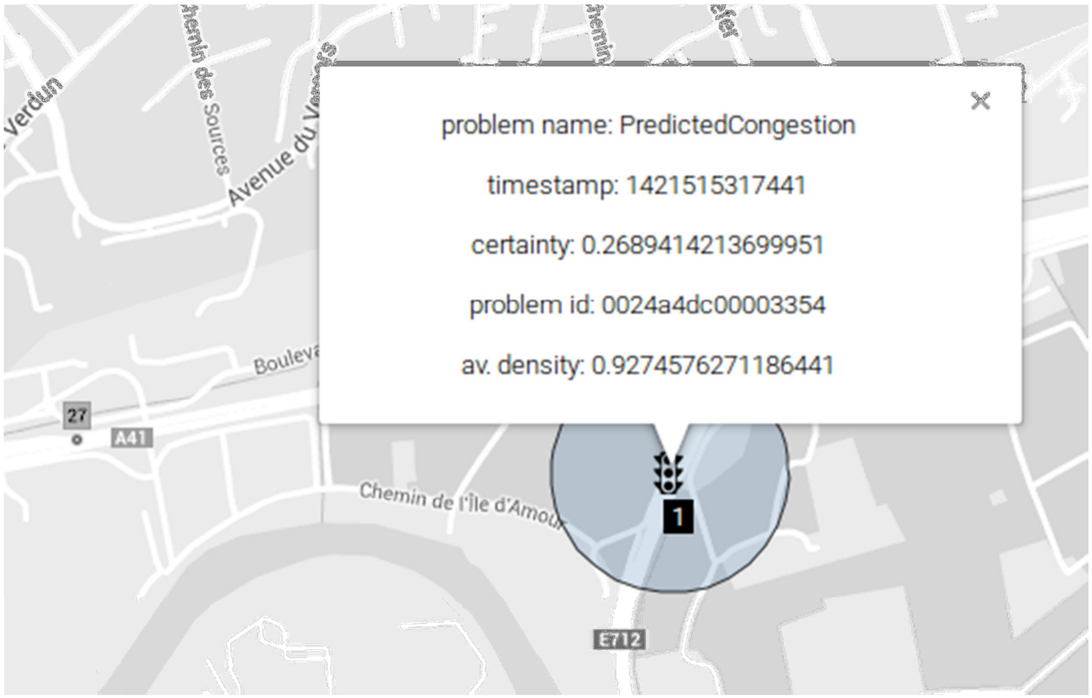
Ramp Metering Control

Controller ID: 1 (0024a4dc00003354)
Current Max Rate Auto
Current Min Rate Auto

Activity Log

1. Road Works at Ramp 2
2. Ramp 15 is closed
3. Ramp 7 sensor not working
4. Accident at Ramp 5
5. Ramp 3 in partial control
6. Ramp 9 in partial control
7. Ramp 12 in partial control
8. Ramp 1 in full control

Figure 2.4 - Dashboard UI for Traffic Management Use Case



problem name: PredictedCongestion

timestamp: 1421515317441

certainty: 0.2689414213699951

problem id: 0024a4dc00003354

av. density: 0.9274576271186441

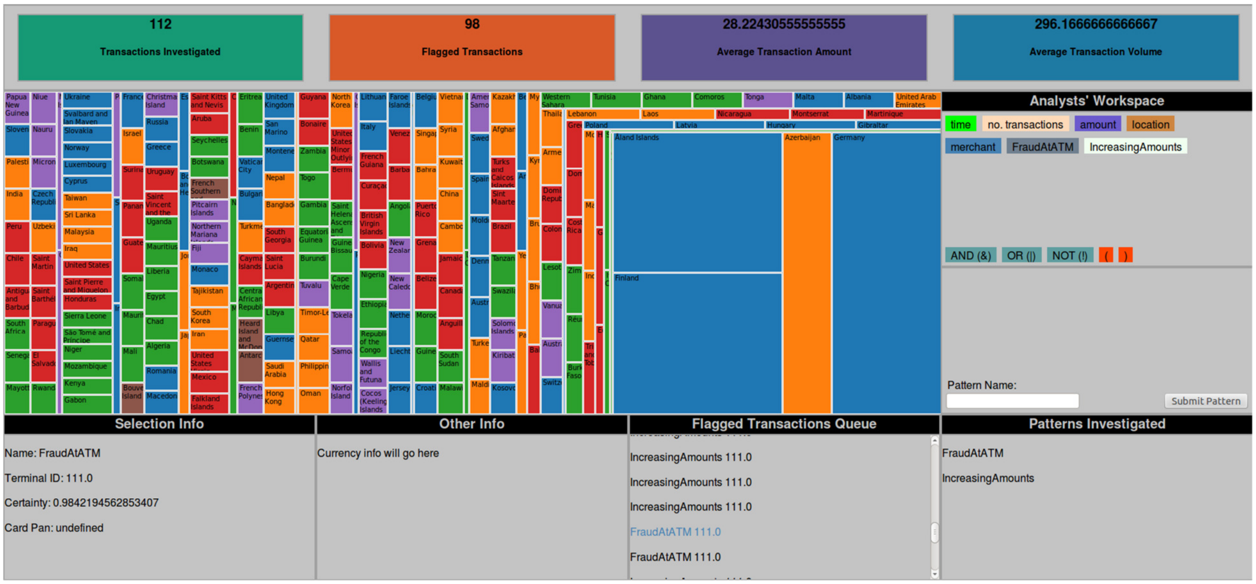
Figure 2.5 - Traffic Management Dashboard - Details of the Predicted Congestion

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.



Name: FraudAtATM

Terminal ID: 111.0

Certainty: 0.9842194562853407

Card Pan: undefined

Currency info will go here

IncreasingAmounts 111.0

IncreasingAmounts 111.0

IncreasingAmounts 111.0

FraudAtATM 111.0

FraudAtATM 111.0

FraudAtATM

IncreasingAmounts

Figure 2.6 - Credit Card Fraud Detection Dashboard

For a more detailed overview of the credit card fraud management dashboard see chapter 5.

3 Install and Execution Instructions

3.1 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 https://github.com/speedd-project/speedd/wiki/Setting-Up-Development-Environment#setup-development-environment-on-your-machine)

In the following we provide instructions for each option.

3.2 Running a virtual machine from the image

The virtual machine image is available for download on the project's FTP server: ftp://143.233.226.91/pub/projects/SPEEDD/VMs/speedd-dev-v0.2.ova.

The VM image was produced and tested on Oracle VM Virtual Box software v4.3[1].

Create a new virtual machine using the provided VM image by importing the virtual appliance file[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 in the next section (3.2.1).

3.2.1 Build the SPEEDD runtime software

Before running the demo pull and build the SPEEDD runtime software from the sources by performing the following steps:

1. cd ~/speedd/speedd-runtime
2. git pull
3. mvn clean install assembly:assembly -DskipTests
4. cd ~/speedd/speedd-runtime/scripts
5. sudo ./cleanup-all.sh

[1] https://www.virtualbox.org/

[2] https://www.virtualbox.org/manual/ch01.html#ovf – section in the virtualbox user manual with description of the import process

3.3 Building from sources

Although we provide a virtual machine image for convenience, it is possible to run the software on any machine that satisfies the system and software requirements. For detailed instructions see the README files as follows:

- SPEEDD runtime: https://github.com/speedd-project/speedd
- Traffic Management Dashboard installation instructions: https://github.com/speedd-project/speedd/tree/master/speedd-ui
- Credit Card Fraud Management installation instructions: https://github.com/speedd-project/speedd/tree/master/speedd_ui_bf

3.4 Running demo scenarios

In the following sections we assume that you run the demo in the virtual machine created from the provided image. In case you run your own environment, the paths have to be adjusted.

3.4.1 Traffic Management

1. cd ~/speedd/speedd-runtime/scripts/traffic
2. sudo ./start-speedd-runtime
3. start sending input events by running ./playevents-traffic
4. open a new terminal shell
5. cd ~/speedd/speedd-ui/bin
6. Run the dashboard by starting ./run.sh
7. Open the following URL in your browser to work with the dashboard UI: http://localhost:3000.

3.4.2 Credit Card Fraud Detection

1. cd ~/speedd/speedd-runtime/scripts/ccf
2. sudo ./start-speedd-runtime
3. start sending input events by running ./playevents-fraud
4. open a new terminal shell
5. cd ~/speedd/speedd_ui_bf/bin
6. Run the dashboard by starting ./run.sh
7. Open the following URL in your browser to work with the dashboard UI: http://localhost:3000.

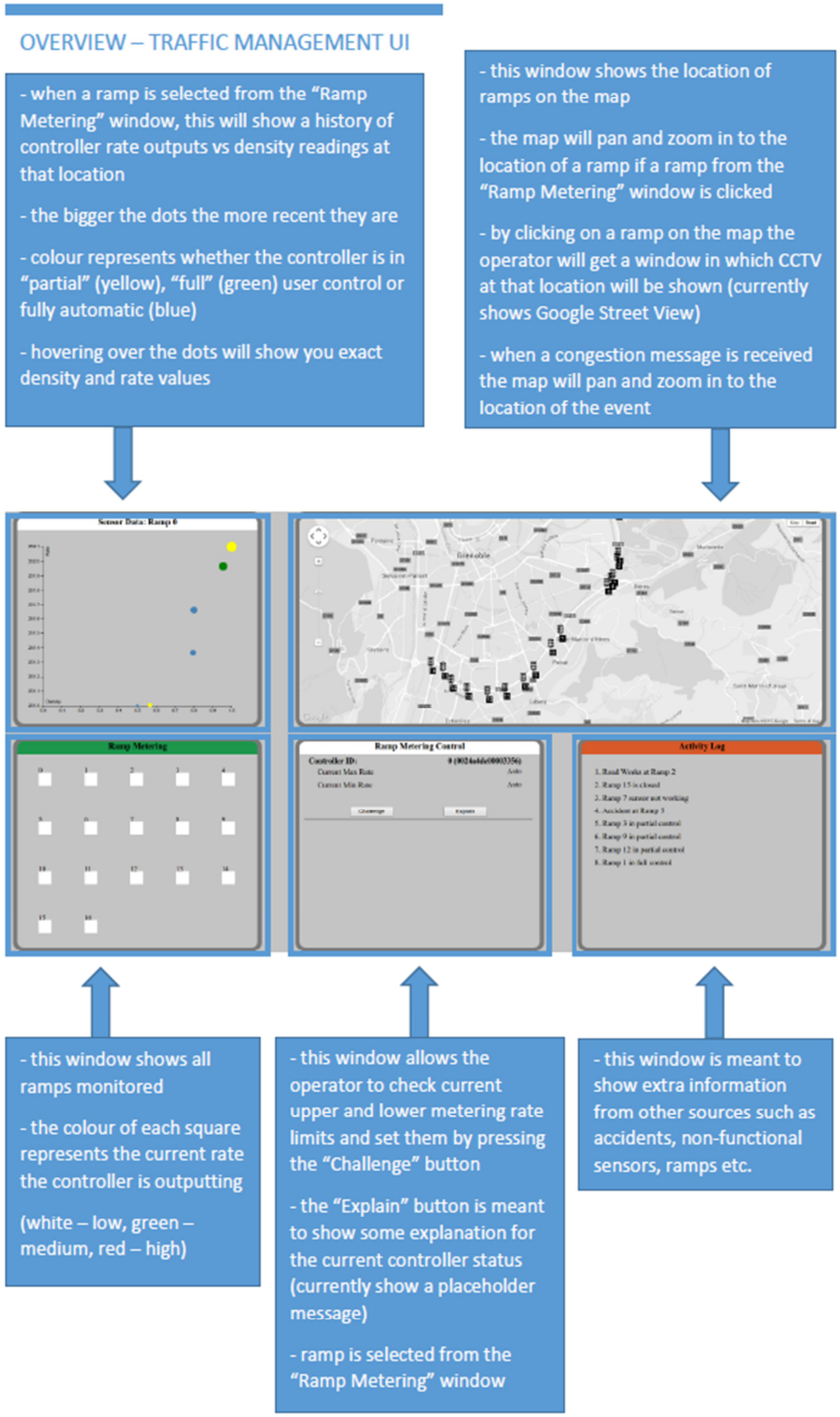
3.4.3 Stopping SPEEDD prototype

1. cd ~/speedd/speedd-runtime/scripts
2. Kill the topology by running 'sudo ./kill-speedd-runtime'
3. 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)[3].

[3] This is a temporal limitation caused by inability to run two topologies with same name on a single STORM cluster. This issue will be fixed in the future versions of SPEEDD software.

4 Traffic Management Dashboard



OVERVIEW – TRAFFIC MANAGEMENT UI

- when a ramp is selected from the “Ramp Metering” window, this will show a history of controller rate outputs vs density readings at that location

- the bigger the dots the more recent they are

- colour represents whether the controller is in “partial” (yellow), “full” (green) user control or fully automatic (blue)

- hovering over the dots will show you exact density and rate values

- this window shows the location of ramps on the map

- the map will pan and zoom in to the location of a ramp if a ramp from the “Ramp Metering” window is clicked

- by clicking on a ramp on the map the operator will get a window in which CCTV at that location will be shown (currently shows Google Street View)

- when a congestion message is received the map will pan and zoom in to the location of the event

Controller ID: 0 (0024a4dc0000356)
Current Max Rate Auto
Current Min Rate Auto

1. Road Works at Ramp 2
2. Ramp 15 is closed
3. Ramp 7 sensor not working
4. Accident at Ramp 5
5. Ramp 3 in partial control
6. Ramp 9 in partial control
7. Ramp 12 in partial control
8. Ramp 1 in full control

- this window shows all ramps monitored

- the colour of each square represents the current rate the controller is outputting

(white – low, green – medium, red – high)

- this window allows the operator to check current upper and lower metering rate limits and set them by pressing the “Challenge” button

- the “Explain” button is meant to show some explanation for the current controller status (currently show a placeholder message)

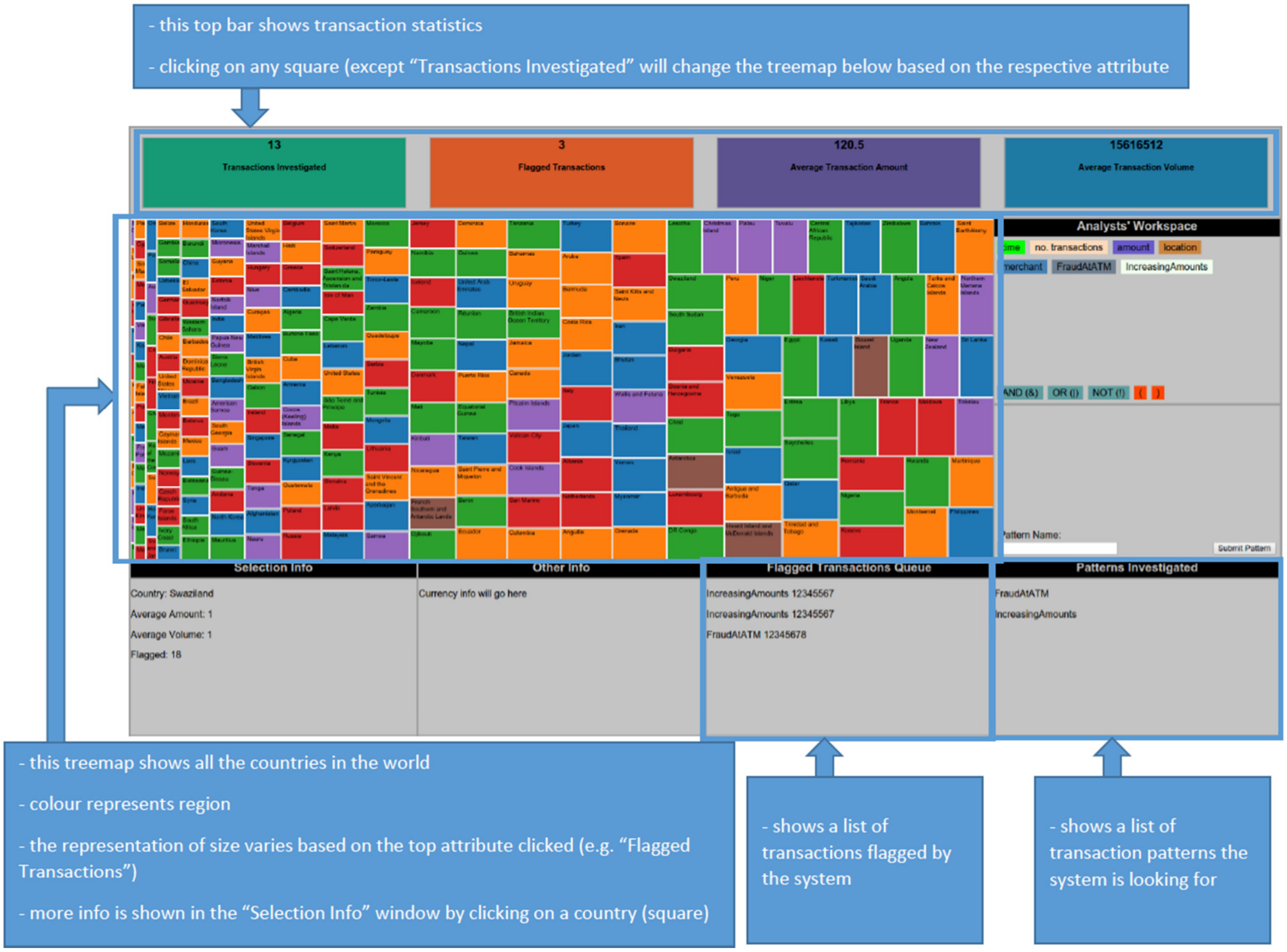
- ramp is selected from the “Ramp Metering” window

- this window is meant to show extra information from other sources such as accidents, non-functional sensors, ramps etc.

Figure 4.1 - Traffic Management Dashboard Overview

5 Credit Card Fraud Management Dashboard

OVERVIEW – CREDIT CARD FRAUD UI



- this top bar shows transaction statistics

- clicking on any square (except “Transactions Investigated” will change the treemap below based on the respective attribute

13 Transactions Investigated

3 Flagged Transactions

120.5 Average Transaction Amount

15616512 Average Transaction Volume

Analysts' Workspace

Pattern Name:

Selection Info

Country: Swaziland

Average Amount: 1

Average Volume: 1

Flagged: 18

Other Info

Currency info will go here

Flagged Transactions Queue

IncreasingAmounts 12345567

IncreasingAmounts 12345567

FraudAtATM 12345678

Patterns Investigated

FraudAtATM

IncreasingAmounts

- this treemap shows all the countries in the world

- colour represents region

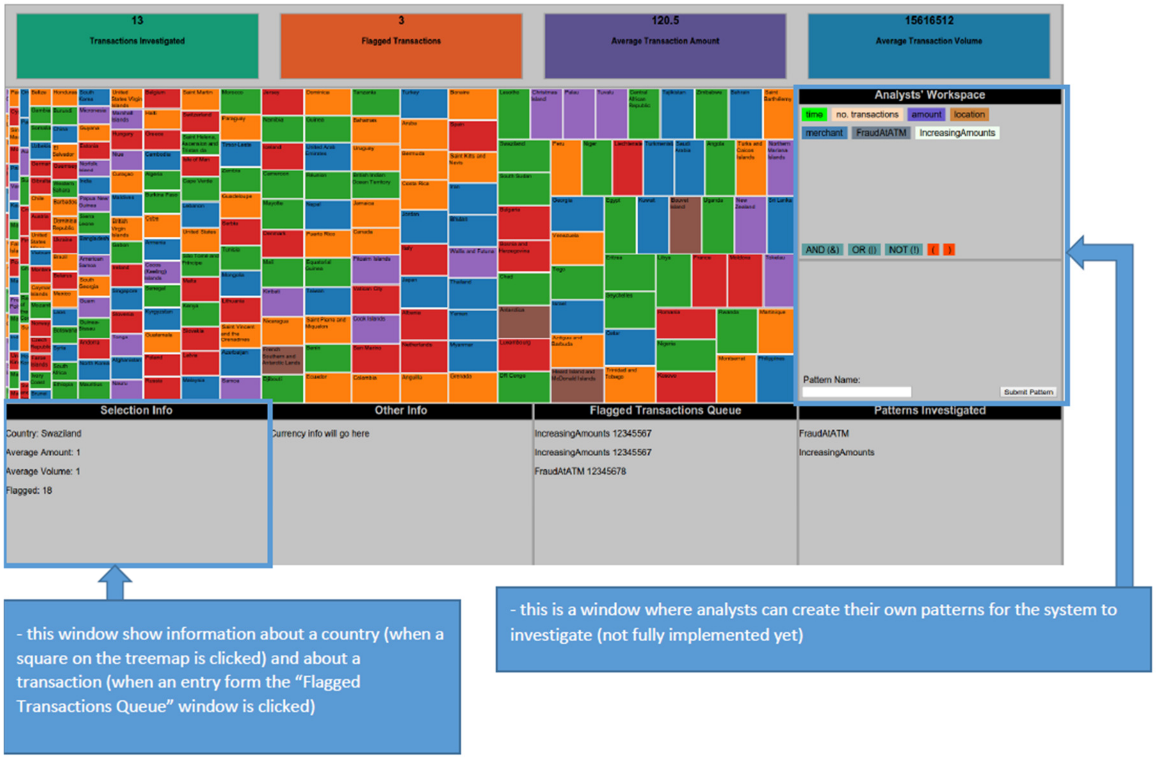
- the representation of size varies based on the top attribute clicked (e.g. “Flagged Transactions”)

- more info is shown in the “Selection Info” window by clicking on a country (square)

- shows a list of transactions flagged by the system

- shows a list of transaction patterns the system is looking for

Figure 5.1 - Credit Card Fraud Management Dashboard Overview



13 Transactions Investigated

3 Flagged Transactions

120.5 Average Transaction Amount

15616512 Average Transaction Volume

Analysts' Workspace

time | no. transactions | amount | location

merchant | FraudAtATM | IncreasingAmounts

AND (&) | OR (|) | NOT (!) | (|)

Pattern Name:

Submit Pattern

Selection Info

Country: Swaziland

Average Amount: 1

Average Volume: 1

Flagged: 18

Other Info

Currency info will go here

Flagged Transactions Queue

IncreasingAmounts 12345567

IncreasingAmounts 12345567

FraudAtATM 12345678

Patterns Investigated

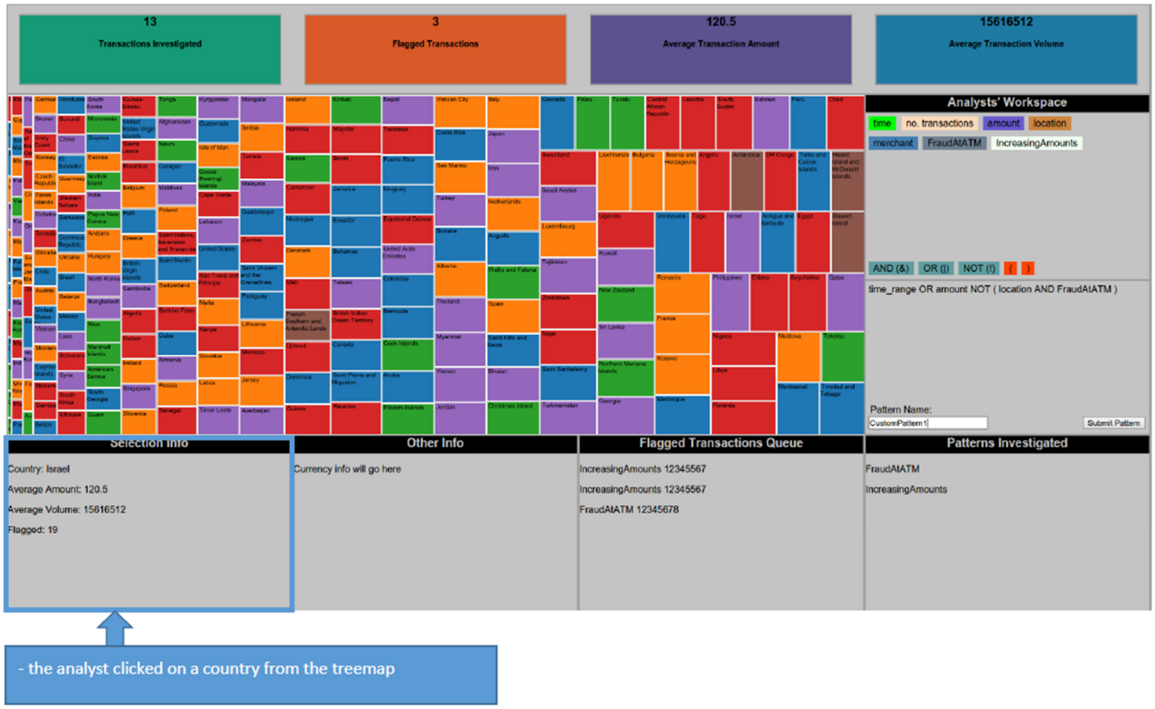
FraudAtATM

IncreasingAmounts

- this window show information about a country (when a square on the treemap is clicked) and about a transaction (when an entry form the "Flagged Transactions Queue" window is clicked)

- this is a window where analysts can create their own patterns for the system to investigate (not fully implemented yet)

Figure 5.2 - Credit Card Management Dashboard overview - cont.



13 Transactions Investigated

3 Flagged Transactions

120.5 Average Transaction Amount

15616512 Average Transaction Volume

Analysts' Workspace

time | no. transactions | amount | location

merchant | FraudAtATM | IncreasingAmounts

AND (&) | OR (|) | NOT (!) | (|)

time_range OR amount NOT (location AND FraudAtATM)

Pattern Name: CustomPattern1

Submit Pattern

Selection Info

Country: Israel

Average Amount: 120.5

Average Volume: 15616512

Flagged: 19

Other Info

Currency info will go here

Flagged Transactions Queue

IncreasingAmounts 12345567

IncreasingAmounts 12345567

FraudAtATM 12345678

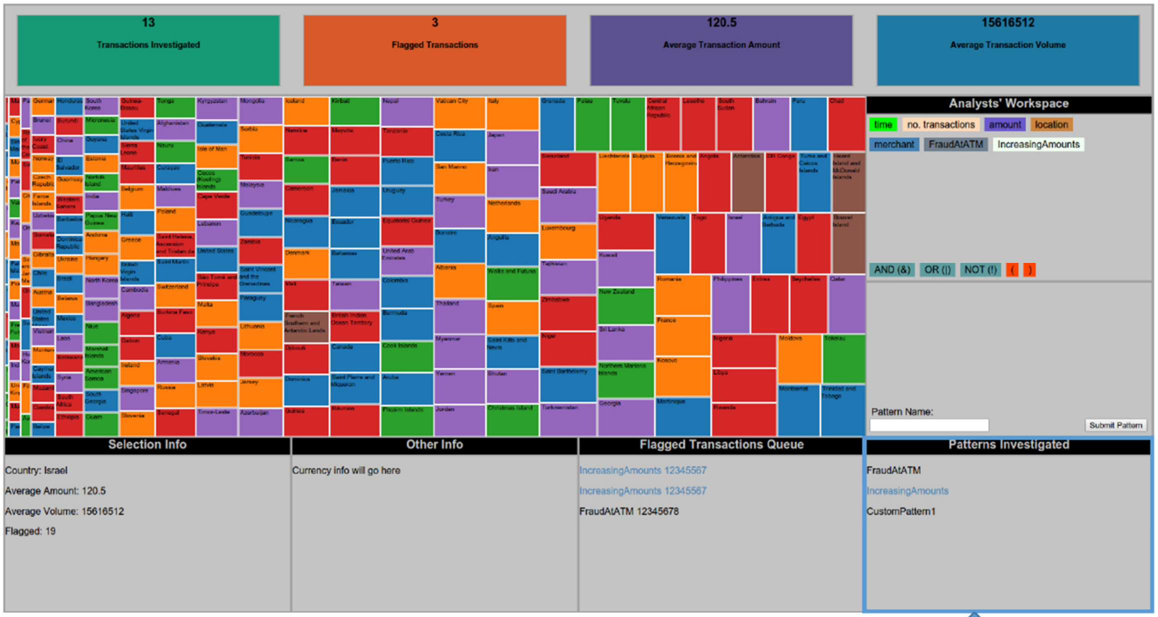
Patterns Investigated

FraudAtATM

IncreasingAmounts

- the analyst clicked on a country from the treemap

Figure 5.3 - Exploring Country Details



Country: Israel
Average Amount: 120.5
Average Volume: 15616512
Flagged: 19

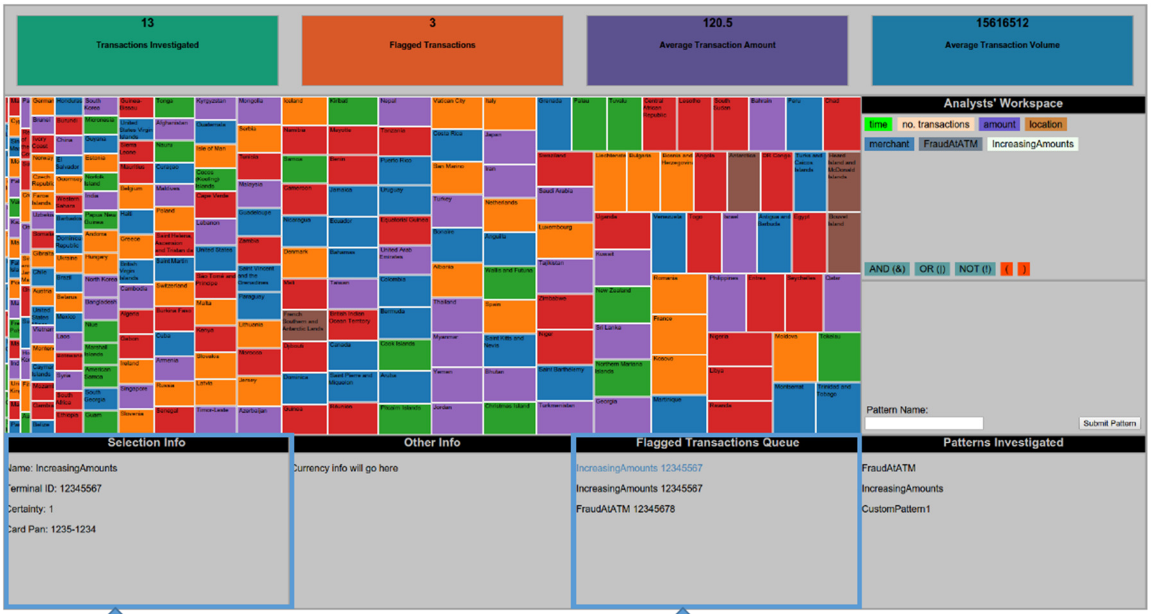
Currency info will go here

IncreasingAmounts 12345567
IncreasingAmounts 12345567
FraudAtATM 12345678

FraudAtATM
IncreasingAmounts
CustomPattern1

- the pattern has been submitted by pressing the "Submit Pattern" button
- "IncreasingAmounts" pattern is selected and corresponding flagged transactions in the queue get highlighted

Figure 5.4 - Highlight Specific Patterns



ame: IncreasingAmounts
erminal ID: 12345567
ertainty: 1
ard Pan: 1235-1234

Currency info will go here

IncreasingAmounts 12345567
IncreasingAmounts 12345567
FraudAtATM 12345678

FraudAtATM
IncreasingAmounts
CustomPattern1

- the first flagged transaction is the queue has been selected and information about the transaction is shown in the "Selection Info" window

Figure 5.5 - Explore details for a flagged transaction