



Flexible Event pRocessing for big dAtA aRchltectures



ICT, STREP

FERARI ICT-FP7-619491

Flexible Event pRocessing for big dAtA aRchltectures

Collaborative Project

D 1.2

Final Application Scenarios and Description of Test Environment

03.02.2015 – 11.05.2016(preparation period)

Contractual Date of Delivery:	13.05.2016
Actual Date of Delivery:	13.05.2016
Author(s):	Taji Ćurin, Damir Bogadi, Marijo Volarević, Marko Štajcer, Ana Mihalić, Michael Mock
Institution:	HT
Workpackage:	WP1
Security:	PU
Nature:	R
Total number of pages:	34



Flexible Event pRocessing for big dAtA aRchltectures



Project coordinator name: Michael Mock

Project coordinator organisation name:**Revision: 3**

Fraunhofer Institute for Intelligent Analysis
and Information Systems (IAIS)
Schloss Birlinghoven, 53754 Sankt Augustin, Germany
URL: <http://www.iais.fraunhofer.de>

Abstract:

The goal of the FERARI project is to develop a general-purpose architecture for in-stream complex event processing. This communication-efficient and flexible architecture – which is published as open source – is instantiated and validated on two specific use-cases from the telecommunication domain using real data provided by HT. These use-cases, their instantiation using the FERARI architecture and their evaluation based on real-world data is addressed in WP1. In this document, we describe the setup of a test bed (both in terms of hard- and software) and its integration with the HT operational infrastructure. Furthermore, we define the final use-cases with respect to the respective evaluation criteria and present a front-end dashboard developed for the mobile fraud detection use case that has been designed to meet the requirements of the HT fraud mining experts.

Revision history

Administration Status		
Project acronym: FERARI		ID: ICT-FP7-619491
Document identifier:	D1.2 Final Application Scenarios and Description of Test Environment (01.02.2015 – 31.01.2016)	
Leading Partner:	HT	
Report version:	3	
Report preparation date:	13.05.2016	
Classification:	PU	
Nature:	REPORT	
Author(s) and contributors:	Taji Ćurin, Damir Bogadi, Marijo Volarević, Marko Štajcer, Ana Mihalić, Michael Mock	
Status:	-	Plan
	-	Draft
	-	Working
	-	Final
	x	Submitted

Copyright

This report is © FERARI Consortium 1. Its duplication is restricted to the personal use within the consortium and the European Commission.

www.ferari-project.eu



Project funded by the European Community
under the
Information and Communication
Technologies
Programme
Contract ICT-FP7-619491

Document History

Version	Date	Author	Change Description
0.1	15/02/2016	Taji Ćurin (HT)	First draft
0.2	25/02/2016	Damir Bogadi (HT)	Second draft
1.0	29/02/2016	Taji Ćurin (HT)	Final version
2.1	01/04/2016	Michael Mock (FhG)	New abstract/Introduction/Test Bed
3.0	15/4/2016	Damir Bogadi (HT)	Complete revision according to EC comments
3.1	10/5/2016	Damir Bogadi (HT)	Update according to reviewer comments
3.2	13/5/2016	Michael Kamp (FhG)	Updates after internal revision.

Table of Contents

1	Introduction	1
1.1	Purpose and Scope of the Document	1
1.2	FERARI Setting.....	2
1.3	Workpackage 1 Description	3
1.4	Relationship with other Documents	4
2	Testing Environment	5
2.1	Test Bed in HT	5
2.2	Local Test Beds.....	7
2.2.1	Poslovna Inteligencija	7
2.2.2	Technical University of Crete	8
2.2.3	IBM	9
2.2.4	Technion.....	9
2.3	Privacy Constraints.....	10
2.3.1	Privacy Requirements	10
2.3.2	Privacy Solution.....	10
2.3.3	Data Anonymization on FERARI Server	12
3	Scenarios Setup.....	14
3.1	Success Criteria and Testing Methodology.....	14
3.2	Fraud Use Case.....	16
3.2.1	Fraud in Telecommunications.....	16
3.2.2	KPIs and Validation Criteria.....	17
3.2.3	Business Value.....	19
3.3	System Health Monitoring Use Case.....	19
3.3.1	DSLAM Definition	19
3.3.2	KPIs and Validation Criteria.....	20
3.3.3	Business Value.....	21
4	FERARI Dashboard Overview	23
4.1	Fraud Event Grid	24
4.2	Fraud statistics widget	26
4.3	Peek/offpeek statistics.....	26
4.4	Most frequently called number	27

4.5	Interactive map	28
4.6	Navigation pane	28
5	Lessons learned.....	29
5.1	FHG.....	29
5.2	HT	29
5.3	IBM.....	29
5.4	Poslovna Inteligencija	30
5.5	Technical University of Crete	30
5.6	Technion.....	31
6	Summary and future steps.....	32
7	References	33
8	Appendix A - Hardware configuration	34

List of Tables

Table 1: Acronyms.....	viii
Table 2: Data sources, periods and volumes of collection	6
Table 3: Local test bed setup in Poslovna Inteligencija	7
Table 4: Local test bed setup in Technical University of Crete	9
Table 5: Local test bed setup in IBM.....	9
Table 6: Local test bed setup in Technion.....	10
Table 7: Criteria Evaluation Matrix	14
Table 8: KPI's for fraud use case	18
Table 9: KPI's for system health monitoring use case	21

List of Figures

Figure 1: Relationship between WP1 (Prototype) and WP2 (Software Platform).....	2
Figure 2: Three Phases of the FERARI Project and WP1	3
Figure 3: Deliverables in WP1	4
Figure 4: Integration of the prototype within the FERARI architecture	7
Figure 5: Development of standalone JAVA libraries with Data Masking capabilities	11
Figure 6: Development of distributed environment.....	12
Figure 7: Sample input file used as an example of implemented data masking	13
Figure 8: DSLAM role in xDSL connectivity diagram [2]	20
Figure 9: Ferari Dashboard.....	24
Figure 10: Grid filled with events.....	25
Figure 11: Drill-down screen.....	25
Figure 12: Fraud statistic in last 24 h	26
Figure 13: Peek/offpeek statistics.....	27
Figure 14: Most frequently called number	27
Figure 15: Interactive map.....	28

Acronyms

CEP	Complex Event Processing
DSLAM	Digital Subscriber Line Access Multiplexer
FERARI	Flexible Event pRocessing for big dAta aRchItectures
PROTON	PROactive Technology Online - IBM tool
WP	Work Package

Table 1: Acronyms

1 Introduction

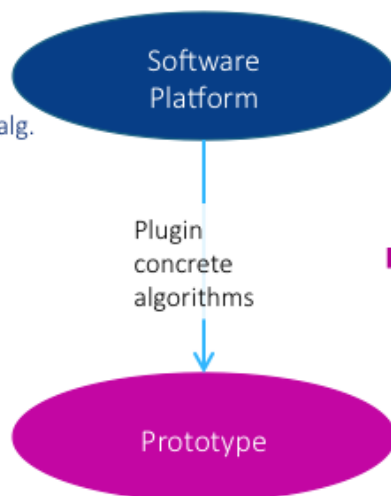
1.1 Purpose and Scope of the Document

Work package 1 (WP1) “Applications scenarios, Test Bed , Prototype” develops the basic application scenarios and use cases of the FERARI project. The WP is driven by the end users in the project, which selected and defined uses cases and will perform testing of final product based on set evaluation criteria.

FERARI aims at developing a general-purpose architecture for flexible, communication efficient distributed complex event processing on massively distributed streams of data. While the architecture, being developed in WP2 and being described in D1.2, is of general purpose and hence not restricted to specific use cases and application domains, it will be instantiated and validated in the FERARI project in two specific use cases working with real-world data. Instantiation of the FERARI architecture in those use cases and evaluation on real-world data in the scope of these use cases is addressed in WP1.

* WP2: Software Platform

- * Open source
- * General purpose for communication efficient big-data stream analysis alg.
- * Flexible event processing
- * Components as libraries
- * Interfaces to plugin concrete algorithms (learning, monitoring)
- * In stream learning
- * CEP Runtime



■ WP1: Prototype/Use Cases

- Testbed infrastructure
- Concrete algorithms
- Implementation of use cases
 - Fraud mining
 - System health monitoring
- Working on HT cluster
- Closed source

Figure 1: Relationship between WP1 (Prototype) and WP2 (Software Platform)

Figure 1 summarizes the main characteristics of WP2 and WP1, showing the differences and relationship between these two work packages. The architecture developed in WP2 is made available as open source platform in the FERARI open source repository <https://bitbucket.org/sbothe-iaais/ferari>. It is intended to

be of general purpose and usable in any application domain that works on distributed streaming data. It provides flexible mechanisms for complex event processing, and libraries, and run-time components for the distributed execution of applications, making use communication efficient protocols for in-situ processing and distributed execution of complex event processing runtimes provided by the FERARI open source platform. Application development of large-scale Big Data streaming applications is supported and significantly simplified by the FERARI open source platform. However, specific application-dependent algorithms like fraud detection or system health monitoring are not part of the open source platform, but will be developed for the purpose of validating the FERARI platform against the use cases developed and described in WP1. These specific use-case related applications running on top of the general purpose FERARI platform are working on real-world data provided by HT and will be instantiated in a test bed on cluster hardware installed at HT. As these applications are very specific and closely related to the HT data, they will not be open sourced (see also DoW p. 6 of 28 in the WP1 description).

The purpose of this document is to explain:

- The setup of the test bed hardware and its integration with the HT operational infrastructure
- The definition of the use cases focussing on evaluation criteria (please refer to D1.1 for the definition of data – and anonymization techniques being applied – for the use cases. Please note that the use case specific evaluation criteria being describe here do not stand for their own, but extend the general purpose goals of the FERARI architecture such as scalability at very large scale, communication efficiency, and flexibility in the expressiveness of the Complex Event Processing.
- The design of a front-end for the fraud-detection use case according to the requirements of the HT fraud mining experts. Again, although FERARI is of general purpose and not restricted or related to any specific kind of GUI, the evaluation of a specific use case application makes it necessary to provide

Report is structured as follows: chapter 2 gives an overview the test bed setup, chapter 3 and 4 give an overview on the fraud detection and system health monitoring use case and their evaluation criteria, respectively, and chapter 5 shows the requirements and design of the Fraud GUI.

1.2 FERARI Setting

The FERARI project aims to develop a highly scalable distributed streaming architecture supporting complex event processing in a communication efficient manner. A key element of the architecture will be communication efficient distributed methods for monitoring global functions on globally distributed states by partitioning of the global function to distributed local functions that communicate only if needed. The general applicability of these methods will be demonstrated in various application scenarios, including distributed online machine learning. The use cases for the evaluation of the

framework and the machine learning algorithms are real world use cases from Hrvatski Telekom. One use case focuses on fraud discovery in mobile networks, which includes SIMbox fraud, premium rate service fraud and roaming fraud amongst others. The other use case is system health monitoring, a problem attracting more and more attention as current failure detectors model normal behavior usually from historical data. This approach is getting more and more challenging, as the network components are becoming more complex and very dynamic as technologies evolve and data consumption grows.

1.3 Workpackage 1 Description

WP1 will develop the basic application scenarios on which all further development crucially depends. This WP is driven by the end user (HT) in the project.

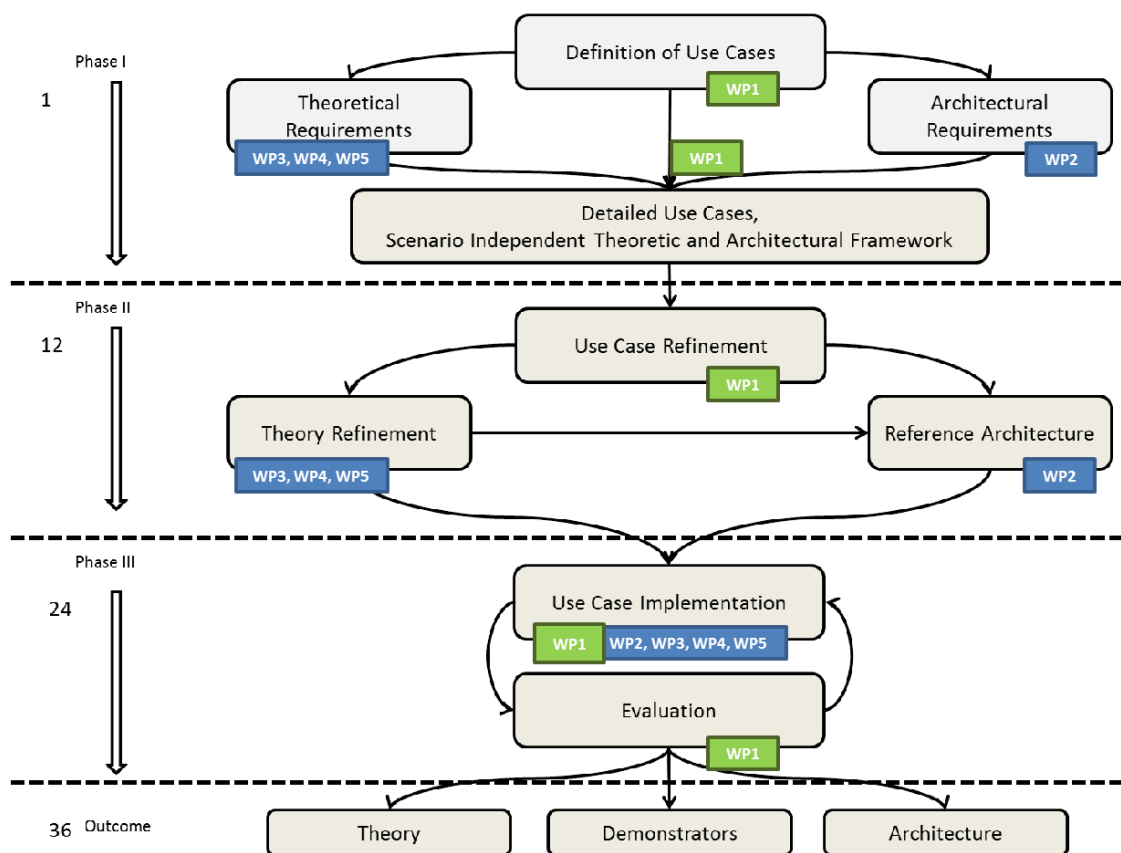


Figure 2: Three Phases of the FERARI Project and WP1

Work package 1 (WP1) “Applications scenarios, Test Bed, Prototype” develops the basic application scenarios and use cases of the FERARI project. The WP is driven by the end users in the project, which selected and defined uses cases and will perform testing of final product based on predetermined evaluation criteria.

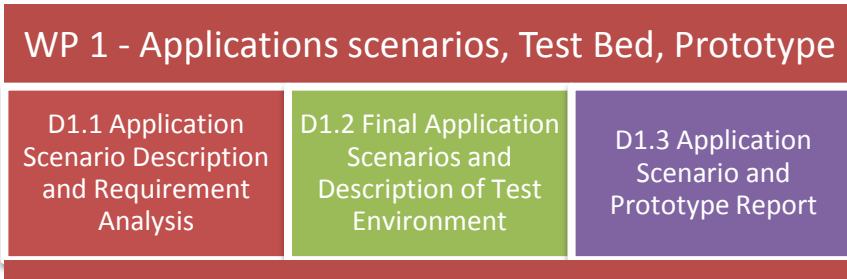


Figure 3: Deliverables in WP1

Specific objectives as per Description of Work are:

1. Selecting and defining the application scenarios in the telecommunication scenario;
2. Definition of testing and evaluation criteria for the end users at HT;
3. Setting up of a test bed both at HT and at the project partner's local sites; and
4. Implementation and evaluation of scenarios to demonstrate the advantage of FERARI with respect to the state of the art as well as to demonstrate its business value.

1.4 Relationship with other Documents

The general purpose architecture described in D2.2 will be instantiated in WP1 to handle the use cases described in D1.2. Fraud detection will be implemented via the instantiation of specific fraud rules in the Complex Event Processing – see detailed description in D4.1 and its application in the flexible event model described in D4.2. D3.2 evaluates a communication efficient in-situ implementation of a fraud rule on the fraud data described in D1.1 and D1.2. The distributed CEP optimizer described in D5.2 will be used to find optimal distributed placements for the complex event expressions that implement the fraud rules.

2 Testing Environment

2.1 Test Bed in HT

HT has installed a dedicated cluster hardware as test bed for the FERARI project. This test bed serves two main purposes:

- **Data repository for real-world data:** Real-world data from HT is copied from the HT operational infrastructure on the FERARI test bed. It is anonymized on the test bed by the partner PI. Anonymized data is made available to rest of the consortium.
- **Installation of FERARI use cases:** Use-case specific application running on the FERARI architecture will be installed at the test bed. Recall that one final goal of the FERARI architecture is to support communication efficient in-situ operations, which implies that the data is process close to where it is created (i.e. in the HT infrastructure). However, as a research project, it is impossible to experiment with the HT operational infrastructure.

The test bed environment of FERARI project is located in Zagreb, Data center Utrine, where FERARI Server is located. All Consortium partners received remote access to the environment. All partners have access to the anonymized data, while only Poslovna Inteligencija has approach to raw data, which anonymizes it and stores it into separate file directory for other consortium members to use.

FERARI project will primary use DSLAM Line data for the System Health use case, Cloud Health logs are collected and available for backup or additional analysis, depending on project needs.

The Hard- and Software configuration of the FERARI test bed is as follows:

- Testing environment is placed in Data Center Utrina, Kombolova 2, 10 000 Zagreb
- Physical Sever; CentOS Linux release 7.1.1503 (Core)
- Hardware Components : HP configuration – details in Appendix B
- System SW installed :
 - OS CentOS,
 - Backup App. Networker
- Application SW installed:
 - NodeJS witch attached modules
 - Redis
 - MySQL
 - Java

- Anonymization engine
- Additionally: Storm/FERARI platform will be installed and configured
- Scheme of File System directories with volume of data collected:
 - System directories
 - oriData filesystem – contains raw data
 - anonData filesystem – contains anonymized data

In a similar way, data from the HT network infrastructure is collected and copied into the FERARI test bed hardware. The following Table summarizes the data connections and amount of data being transferred between HT infrastructure and the FERARI test bed:

Data Sources and Volumes
HT data sources with period of collecting
• Data Warehouse (CDR) – batch load
• Fraudsters data – batch load (weekly)
• VMware – Cloud Health logs – on daily basis
• DSLAM (POLraw, POLraw_1 data bases) –On daily basis
• DSLAM HW measurement (Netcool) – On daily basis
Data Volumes
• DWH CDR (currently): 0.4 TB – planned additional 0.6 TB in next batch
• Cloud Health logs (currently): 0.5 TB
• DSLAM Line data: 1.9 TB
• Total: 2.8 TB

Table 2: Data sources, periods and volumes of collection

The following Figure shows the relationship between the FERARI test bed and the FERARI architecture described in D2.2 and how the data for fraud detection and system health monitoring is collected and processed on the FERARI test bed. Note that the data from the operational HT infrastructure is copied into the FERARI test bed.

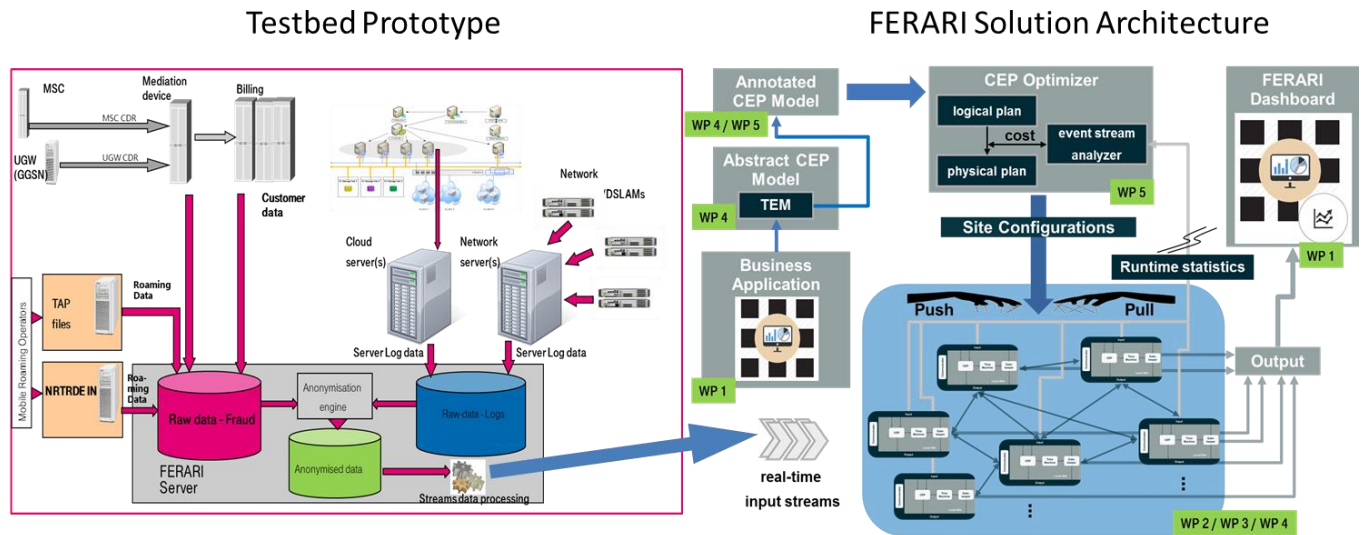


Figure 4: Integration of the prototype within the FERARI architecture

The complete FERARI runtime system is installed on the FERARI test bed. The (anonymized) real-world data, previously being collected and copied from the HT infrastructure to the FERARI test bed, is being fed as streaming data into the FERARI Architecture. Use case specific algorithms implemented in the FERARI architecture evaluate the streaming data and generate use case specific output, which can be fed into a use case specific GUI as presented in section 4.

2.2 Local Test Beds

2.2.1 Poslovna Inteligencija

Local Test Bed Setup	
Category	Description
Hardware used	Intel i7 quad core with 8 GB memory
System software	Linux, Debian 64 bit
Application software used	Proton standalone and ProtonOnStorm, Tomcat, Java 7
Purpose and connection with WP	Testing of system prototype application for WP1
Consortium member	Poslovna Inteligencija

Table 3: Local test bed setup in Poslovna Inteligencija

2.2.2 Technical University of Crete

Local Test Bed Setup	
Category	Description
Hardware used	<u>Single Virtual Machine Testbed</u> Single Laptop PC with: - Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz 2.59GHz with 8GB RAM and 1TB HDD <u>Cluster Infrastructure - SoftNet Cluster</u> A cluster with 23 machines: 13 Dell PowerEdge R300 Quad Core Xeon X3323 2.5GHz machines with 8GB RAM and 500GB HDD; 3 Dell PowerEdge R310 Quad Core Xeon X3440 2.53GHz machines with 8GB RAM and 500GB HDD; 3 Dell PowerEdge R310 Quad Core Xeon X3440 2.53GHz machines with 16GB RAM and 500GB HDD; 3 Dell PowerEdge R320 Intel Xeon E5-2430 v2 2.50GHz machines with 32GB RAM and 1TB HDD; - One Dell PowerEdge R320 Intel Xeon E5-2430 v2 2.50GHz machine with 32GB RAM and 2TB HDD.
System software	<u>Single Virtual Machine Testbed</u> Linux Alkor 3.16.0-4-amd64 #1 SMP Debian 3.16.7-ckt9-3~deb8u1 (2015-04-24) x86_64 GNU/Linux <u>Cluster Infrastructure - SoftNet Cluster</u> Linux clu04 3.2.0-95-generic #135-Ubuntu SMP Tue Nov 10 13:33:29 UTC 2015 x86_64 x86_64 x86_64 GNU/Linux
Application software used	<u>Single Virtual Machine Testbed & Cluster Infrastructure - SoftNet Cluster</u> Redis: Redis server v=2.8.17 sha=00000000:0 malloc=jemalloc-3.6.0 bits=64 build=

	5937320cdd80c1e4 • Eclipse: Luna 4.4.1 • Storm : 0.9.5 • ProtonOnStorm: (Protonj2SE 0.0.1)
Purpose and connection with WP	• Single virtual machine testbed was used for elementary debugging and integration testing purposes in WP5 and WP2, respectively. • SoftNet cluster was used for implementation and testing of the algorithms incorporated in the FERARI CEP optimizer (WP5). • SoftNet cluster was used for developing adaptors and integrating the various FERARI modules on the common architectural framework (WP2).
Consortium member	Technical University of Crete (TUC)

Table 4: Local test bed setup in Technical University of Crete

2.2.3 IBM

Local Test Beds Setup	
Category	Description
Hardware used	Intel i7 dual core with 16 GB memory
System software	Windows 7
Application software used	Proton standalone and ProtonOnStorm, Tomcat, Java 7
Purpose and connection with WP	Testing of CEP applications for WP4
Consortium member	IBM

Table 5: Local test bed setup in IBM

2.2.4 Technion

Local Test Bed Setup	
Category	Description

Hardware used	Four 6 core machines with 8G RAM each
System software	Linux
Application software used	-
Purpose and connection with WP	Testing in-situ methods develop in WP3
Consortium member	Technion

Table 6: Local test bed setup in Technion

2.3 Privacy Constraints

2.3.1 Privacy Requirements

Sensitive customer data is used as input for FERARI and therefore strict privacy policies and anonymisation rules have to be applied. In month 3 of the project, new EU Opinion on Anonymisation caused re-evaluation of anonymisation techniques previously in use.

New anonymisation techniques were developed by Poslovna Inteligencija and evaluated by Hrvatski Telekom privacy team. Approval of new anonymisation techniques was required (and successfully obtained) from Deutsche Telekom Headquarter in Bonn.

Benefits from newly developed anonymisation go well beyond FERARI project; other projects in Hrvatski Telekom and DT Group are interested in using the same techniques because the same strict regulations apply to all projects which are dealing with sensitive customer data.

2.3.2 Privacy Solution

New EU Opinion on Anonymization (*Article 29 Data Protection Working Party, Opinion 05/2014 on Anonymisation Techniques*) was adopted on 10 April 2014.

New methods needed to be adopted and implemented with focus on:

1. Security Enforcement
 - a. Most companies have no way to enforce security practices over production data
2. Production Impact
 - a. Unsecured Access and Choices can lead to Fatal Mistakes in Production
3. Monitor Everyone, Everywhere All the Time
 - a. How can you track who and what masked in different locations and based on which data set?

Protecting sensitive information by hiding or altering data so that an original value is unknown is imperative for HT and the project.

Requirements:

- Copy of the production dataset without sensitive information;
- Use of realistic test data;
- The ability to replace existing values with new values that follow the format of the original;
- Cascade masked values to all related tables and fields;
- Resistance to Linking attacks and compliance with EU opinions; and
- In-situ Data Masking methods/algorithms.

Solution:

- First iteration – development of standalone JAVA libraries with Data Masking capabilities; and
- Second iteration – development of distributed environment.

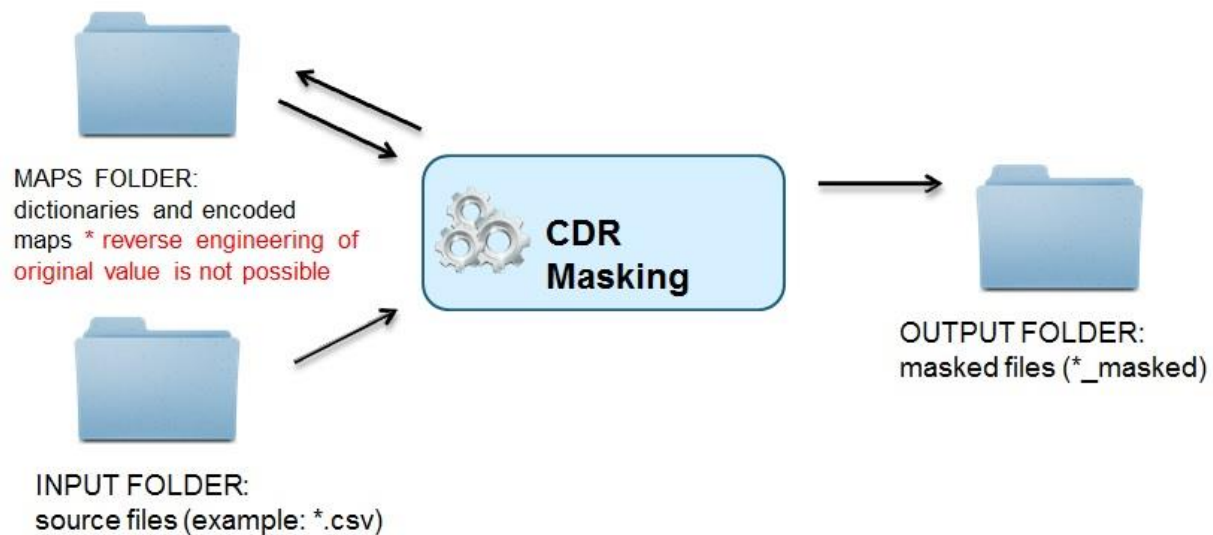


Figure 5: Development of standalone JAVA libraries with Data Masking capabilities

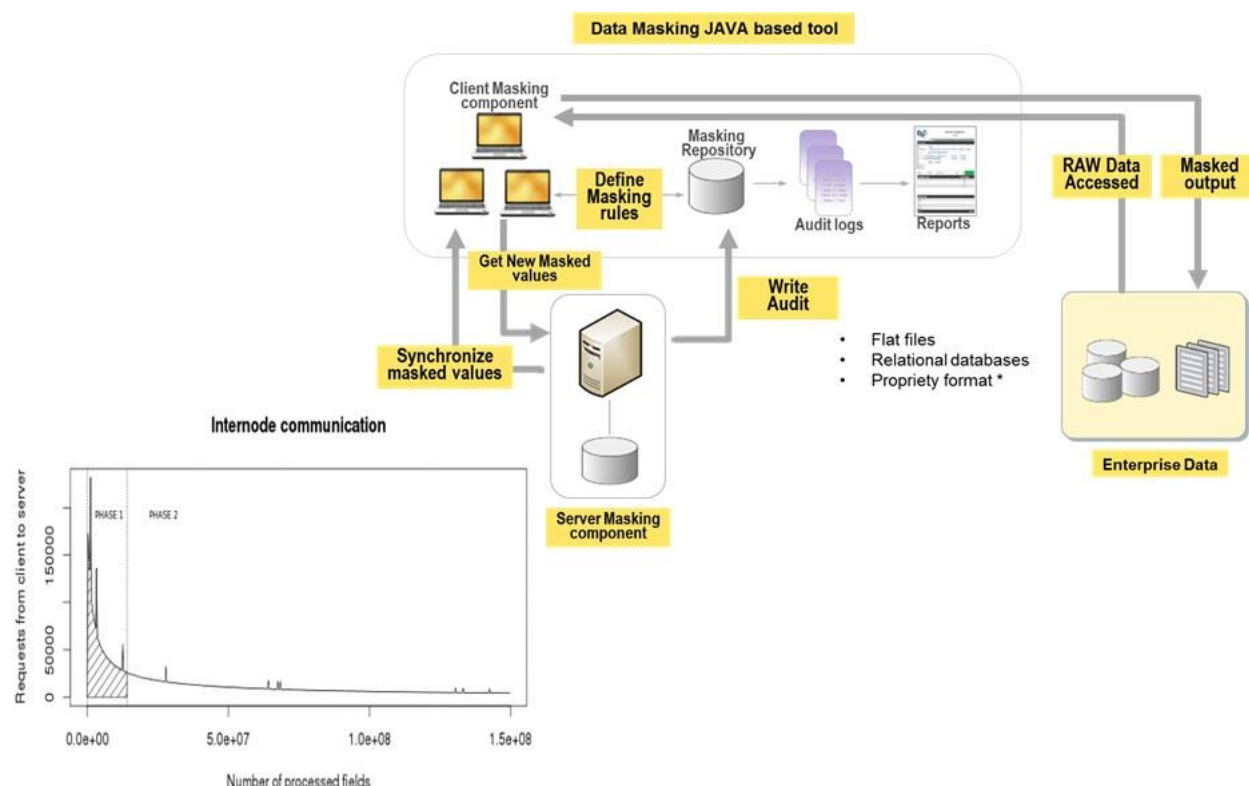


Figure 6: Development of distributed environment

2.3.3 Data Anonymization on FERARI Server

Our solution developed by PI for FERARI project uses randomization approach to retain data structure and information, as well as data distribution if needed. At the moment, following methods are implemented for masking data for project purposes:

- **Substitution technique** which replaces the original value with some other fake masked value. This fake value can be either randomly chosen from the given list (pseudonymization) of fake values or generated randomly.
- **Partial Substitution** technique which replaces part of the original value with some other fake masked value thus preserving one part of original information (this is suitable for phone number masking when we want to preserve important phone prefix but not specific number)
- **Nullify** – technique which simply deletes the original value. Normally, the technique cannot be used on non-nullable columns of the database.

In the figure below there is a sample input file which is used as example of data masking. Actually, the file content are call detail records, which are used as source data for evaluation of Mobile Fraud use-case, whose columns contains following sensitive information:

- BILLED_MSISDN – the MSISDN, which will be charged;
- CALLING_NUMBER – the MSISDN, which is calling;
- CALLED_NUMBER – the MSISDN, which is called; and
- OTHER_PARTY_TEL_NUMBER – Telephone number.

CALL_ID	BILLED_MSISDN	CALL_START_DATE	CALL_PRICE	CALLING_NUMBER	CALLED_NUMBER	OTHER_PARTY_TEL_NUMBER
10826720546	3859932JVBLK	5.5.2014 15:22	BG0 XB3	3859932JVBLK	web.htgprs	web.htgprs
10859401789	3859970XCXYX	7.5.2014 20:12	T22 E4B	+3859970UHINA	+387658YUBNP	+387658YUBNP
10935693759	3859896ALZRT	13.5.2014 12:09	BG0 E4B	3859896ALZRT	web.htgprs	web.htgprs
10965354455	3859890TCKRO	15.5.2014 12:43	T11 XR3	+3859890OTRSM	003859891EZKVB	003859891EZKVB
11031975043	3859118TGLKM	20.5.2014 11:42	T21 XB4	+3859945FCFDD	+3859118DFGTR	+3859945FCFDD
11084511210	3859816KIXMU	23.5.2014 17:22	BG0 XR4	3859816KIXMU	web.htgprs	web.htgprs
11096164884	3859943RVTGQ	24.5.2014 16:23	T21 XR5	+3859919BIZDO	+3859943SJWYP	+3859919BIZDO
11158223819	3859818UHWFX	29.5.2014 6:21	T21 XB4		+3859818ERUQK	
10798671299	3859891GSNMT	3.5.2014 8:29	T11 XR2	+3859891YZCSJ	003859953SKIED	003859953SKIED
10869735267	385982FJUTD	8.5.2014 13:25	T11 XB4	+385984HZTVL	+385982ZPBBP	+385984HZTVL
10911754896	3859818JZFGD	11.5.2014 19:15	T21 XR4	+3859892FRWYE	+3859818IQJSB	+3859892FRWYE
10941921853	3859957KQOOD	13.5.2014 20:47	T21 XR4	+3859221HFRPB	+3859957KTVJL	+3859221HFRPB
10967429087	3859936BFKTS	15.5.2014 15:31	T11 XR5	+3859150CFCIY	+3859936NPAMJ	+3859150CFCIY
11002199784	3859921VZUOQ	18.5.2014 10:01	T21 XB5	+3859173YLGBl	+3859921PXBHB	+3859173YLGBl
11037688983	3859898THQWR	20.5.2014 17:38	T11 XR3	+3859963ATBWJ	+3859898RXUDA	+3859963ATBWJ
11114595297	385982PKKGS	26.5.2014 9:29	T11 XB4	+385982RSQFR	003859161NVQUT	003859161NVQUT
11183540627	3859980WXMEG	30.5.2014 13:06	T11 XR1	+3859967ZTVVR	+3859980NZVJE	+3859967ZTVVR
11201411018	3859898LYVVQ	31.5.2014 19:41	T11 XR2	+3859151WYRUW	+3859898NGKPA	+3859151WYRUW
10808757615	385985TKQFG	4.5.2014 9:03	BG0 XR5	385985TKQFG	web.htgprs	web.htgprs
10925712419	3859896JXDPU	12.5.2014 16:50	BG0 E5B	3859896JXDPU	web.htgprs	web.htgprs
11014582728	3859890OZJWI	19.5.2014 10:08	T11 XR3	+3859890QVQCW	003859178VOBPR	003859178VOBPR
11044351474	3859112PQZPK	21.5.2014 8:51	T11 XB4	+3859121EBAFK	+3859112IMZBA	+3859121EBAFK
11068712765	3859153KVG TG	22.5.2014 17:30	T11 E4B	+3859590NWPTF	+3859153ONLGB	+3859590NWPTF
11085855519	3859899HPXEF	23.5.2014 18:44	T21 E5B	+385987WJCTD	+3859899AHIZT	+385987WJCTD
11105409324	3859898GDIPH	25.5.2014 14:09	T11 XB4	+3859232FLROK	+3859898KLEJI	+3859232FLROK
11158529570	3859157LKAVM	29.5.2014 6:47	T11 XB3	+3859157IDORC	00385982ECKDV	00385982ECKDV
10778448234	3859771VNXBO	1.5.2014 13:40	T11 F44	+3859771JERFS	003859193IGTAC	003859193IGTAC
10807269405	385984AYZOT	4.5.2014 4:48	T11 V50	+385317AFCRO	+385984BHNWI	+385317AFCRO
10831101026	3859926KACUV	6.5.2014 5:06	T21 F38		+3859926AXUMF	

Figure 7: Sample input file used as an example of implemented data masking

As shown in picture, substitution method is applied on columns marked red, where phone number prefix is retained in order to be able to identify called zones and destinations, and the rest of the digits are masked, telephone number cannot be recognized and it cannot be related to specific person or customer.

3 Scenarios Setup

This chapter describes the use cases investigated in FERARI and testing methodologies and business values in detail. Two real world use cases from Hrvatski Telekom have been selected to test the solutions to be developed within FERARI. These use case are:

- Fraud Mining; and
- System health monitoring.

Each use case is described in a dedicated section.

3.1 Success Criteria and Testing Methodology

When evaluating FERARI solution, we take into consideration several criteria's:

Criteria Evaluation Through KPIs		
CRITERIA	FRAUD USE CASE	SYSTEM HEALTH MONITORING USE CASE
C1. REDUCTION OF COMMUNICATION	N/A	
C2. PROCESSING TIME	Fraud Case Detection Time	Failure Prediction Time
C3. NUMBER OF FALSE ALARMS/VALUE OF THE PROPOSED SOLUTION	Number of Fraudsters	False Alarm Ratio
	Revenue at Risk Due to Fraud	Failure Precision Percentage
C4. NUMBER OF DOMAINS TO WHICH THE FERARI SOLUTION CAN BE APPLIED	N/A	
C5. SOLUTION ADAPTABILITY TO NEW AND UNFORESEEN CIRCUMSTANCES	N/A	

Table 7: Criteria Evaluation Matrix

Backtesting is an important model-monitoring activity that aims at contrasting exante made predictions with ex-post realized outcomes (Baesens 2014). [1]

For each use case testing we focus on:

- Correct implementation of theory in the solution;
- Testing of the solution automation;

- The ability of key components to communicate among each other;
- Correctness output data format and content; and
- The defined KPIs.

A usual approach consists of three standard levels of tests:

1. Unit testing - performed on local test beds

Unit testing covers testing of each of the key FERARI components and the correctness of the format and content of the provided input data. This has been performed locally by the Consortium members on their local test beds and shall be continued to be performed in Year 3 as the FERARI system continues to be further developed and fine-tuned.

2. Integration testing - performed on local test beds and HT's test bed

Integration testing focuses on testing the solution automation capabilities, the ability of key components to communicate among each other and correctness of the output data format. This has been performed locally by the Consortium members on their local test beds and shall be continued to be performed in Year 3 as the FERARI system continues to be further developed and fine-tuned. Also, this shall be tested during System testing on the HT's test bed.

3. System testing - performed on HT's test beds

A system test will involve testing the interface, executing actions, reviewing entries, results drilldown, followed by summary processing of entries. This is the focus of HT in Year 3.

3.2 Fraud Use Case

This section defines the problem of the fraud mining use case, lists some typical fraud examples in telecommunications, the fraud discovery process, KPIs and business value from fraud detection systems.

3.2.1 Fraud in Telecommunications

Communication service providers are also targets of fraud schemes that can significantly impact their revenues and service performance. 2015 Global Loss Survey states that loss from fraud in telecommunications accounts for 1.69 percent of lost revenues.

Usual fraud detection systems in telecommunications utilize billing and usage information, network data, location data, CRM Data and some sources of external data in order to build systems that can detect certain patterns that correspond to fraudulent behavior.

The goal in fraud mining is to identify users, which use a network service without the intention to pay for that use. Many fraud mining systems in telecommunications use some form of rules Rosset et al. (1999), often defined by fraud experts or automatically by some software, to raise alarms. These alarms are checked by fraud investigators on a case-by-case basis. During night times when no fraud investigators are present the software may automatically block certain calls to prevent damage. During day times the fraud investigators take actions after they have investigated a case. It is their duty to decide whether a suspicious behavior is fraudulent or legal. This depends on the current call, the call history, the customer history and the subscription plan of the customer. The focus within FERARI lies on the identification of suspicious calls and users and the design of distributed communication efficient systems for this task. Within this coarse definition of telecommunication fraud several well known patterns exist, each with its own characteristics.

Fraud examples include:

- Subscription Fraud;
- SIM Cloning;
- Roaming Fraud;
- Abnormal Usage Fraud;
- Interconnect Bypass;
- False Answer Supervision; and
- Internal Fraud, etc.

3.2.2 KPIs and Validation Criteria

System test in the Fraud Use Case will consist of testing the solution automation and ability of key components to communicate among each other. Correctness of the solution's output format and content will also be tested. Result of the test should be subscribers that are flagged as fraudsters.

Our aim is to perform benchmarking of HT's fraud detection system against results derived from the FERARI fraud use case. In order to perform comparison of the results we shall set the real thresholds used in HT's fraud detection system in FERARI implementation. For the purpose of results evaluation, the following KPIs have been determined:

Fraud Use Case KPIs		
KPI	Description	Validation Criteria
Fraud case detection time	Time spent between moment when fraud case happened and moment when HT detected fraud case. We aim to detect fraudulent behavior as soon as possible and to take measures in preventing such behavior. Expectation from FERARI project is to decrease fraud detection time , meaning we will be able to detect fraud cases sooner than now.	Velocity improvement. Decrease fraud detection time by 10 percent. Current fraud case detection time is 20 minutes.
Number of fraudsters	Number of customers where fraudulent behavior is detected. Our goal is to detect as many as possible customers who exhibit fraudulent behavior. Expectation from FERARI project is to increase number of detected fraudsters. While detecting the number of fraudsters in a data population, we have to take notice on the false positive and false negative alarms ratio . While we would have perfect result if we declare the whole population as fraudulent we would also have the highest percentage of false positives. Generally, the false positive rate should be no more than produced by the current fraud mining system deployed in HT.	Increase in effectiveness or value with the proposed solution. At least as many fraudsters are detected as before.
Revenue at risk due to the fraud	Estimated sum of charges generated by fraudsters which will never be collected by HT. Fraudsters generate significant charges through fraudulent behavior; fraudsters never pay those invoices which represents direct revenue loss for the service provider. We aim to detect all fraudsters and to detect fraudulent behavior as soon as possible; therefore fraudsters will have limited time to generate fraudulent charges. This will in return decrease the revenue at risk and bring more value .	Increase in value with the proposed solution. Decrease revenue at risk due to the fraud by 15 percent.

Table 8: KPI's for fraud use case

3.2.3 Business Value

Fraudsters are generally hard to be recognized during the credit check process and they cause a significant financial damage. According to market analysis conducted by TM forum, despite operator's continuous battle to prevent fraud, it still accounts for approximately 3 percent of revenue lost. To minimize exposure, it is crucial to build fraud prevention models that can detect fraudulent pattern beforehand or within a very short timeframe after fraud happened. In times of ever increasing margin pressures, fraud prevention can deliver improvement to the bottom line and a quick ROI.

Fraud often impacts the customer as much as it impacts the provider's bottom-line. In case of SIM cloning, fraudsters "take over" a legitimate account. Original customer will follow his regular usage pattern, but fraudster may simultaneously use the cloned card to superimpose his use on the legitimate user. Reducing customer's involvement in fraud by a third party through the operator prevents damage to customer experience and customer loyalty and thus decreases customer's likelihood to churn. Similar, in case of SIMbox fraud, where large number of SIM cards from national and foreign operations is used to avoid international routing cost, network degradation occurs as the cells are often overloaded. Poor network quality leads to customer's dissatisfaction and decreases customer's loyalty. In case of subscription fraud, it is important to have highly accurate model in order to minimize possibility of blocking a customer who does not have intention to behave fraudulent. In opposite case, provider loses potential customer and revenue. On the other hand, customers who are detected early as fraudsters can then be offered an alternative lower risk product. In case of premium rate service fraud, where fraudster generates calls to the premium rate service and gets a share of the profits from the premium rate service provided, and roaming fraud, customer's satisfaction is less in danger than in previously mentioned fraud types, but provider's financial exposure rather exists.

To summarize, reducing fraud can greatly improve profitability and solidify customer experience and customer loyalty. There may be perceived short-term loss of revenue where fraudulent customers are cut-off, but as the fraud would impact on operator's cost, result is net profit. Fraud managements is primary a cost saving oriented use case but it may result in revenue stability through maintenance of customer experience and loyalty.

3.3 System Health Monitoring Use Case

This section defines the problem of the system health use case, lists some typical network monitoring problems in telecommunications, KPIs and business value from system health monitoring system.

3.3.1 DSLAM Definition

DSLAM is a telecommunications device usually located in telephone exchanges or a central office and enables interconnection of digital subscriber lines (DSL, ADSL) with the internet.

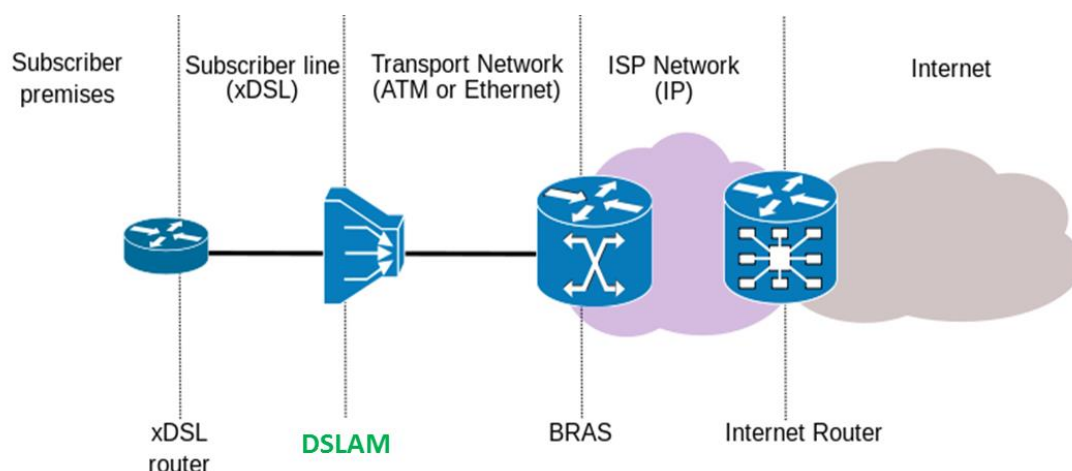


Figure 8: DSLAM role in xDSL connectivity diagram [2]

3.3.2 KPIs and Validation Criteria

System test in the System health monitoring use case will consist of testing the solution automation and ability of key components to communicate among each other. Correctness of the solution's output format and content will also be tested. Result of the test should be system components that are likely to fail.

Our aim is to see if the implemented System health monitoring use case can be used in HT in order to detect anomalies in equipment parameters and signal a preemptive alarm if there is a high likelihood that a part of the equipment will fail. For the purpose of results evaluation, the following KPIs have been determined:

System Health Monitoring Use Case KPIs		
KPI	Description	Validation Criteria
False alarm ratio	Ratio between predicted failures and actual failures. The aim is to have prediction mechanisms as accurate as possible, meaning that number of false alarms should be as low as possible. False alarm ratio ~20% would be acceptable for HT and can provide significant improvements compared	Veracity and effectiveness of the proposed solution. False alarm ratio up to 20 percent.

	to current practices.	
Failure prediction time	The aim is to have this value reasonably high which would give HT enough time to react and take measures in preventing actual equipment failure. Failure prediction time less than 24 hours would be sufficient for HT to react in case of line performance corruption, to be able to correct line and increase line stability and/or quality.	Velocity improvement. Failure prediction time less than 24 hours.
Failure precision percentage	Aim is to have DSLAM predicted failure precision in 48 hours above 50%. This in return should bring more value to HT since the equipment will be repaired before it fails and disrupts network performance and thus HT's customers.	Increase in value with the proposed solution. Predicted failure precision in 48 hours above 50%.

Table 9: KPI's for system health monitoring use case

3.3.3 Business Value

Modern datacenters have large number of machines, both physical and virtual, and automated health monitoring systems are crucial for their proper functioning. However, current failure detectors model normal behavior usually from historical data. This approach is getting more and more challenging as the environment in which cloud services must live in is becoming unpredictable and very dynamic as technologies evolve and data consumption grows. This makes workload constantly changing and often difficult to predict. Frequent software and hardware changes can make old data and previously learned models obsolete. Furthermore, obtaining expertly-labeled historical data is expensive since it requires manual inspection by experts with deep insight into how model works. And last, supervised techniques can often only detect problems that have been foreseen or encountered before.

DSLAM failure affects operator's business from multiple angles. First, DSLAM failure affects the network performance customers are experiencing. Quality of service is tightly related to customer's satisfaction. Customers with low index of satisfaction are less loyal to their service provider are more like to churn. Second, if service cannot be consumed due to DSLAM failure, operator is unable to charge this service to a user and experiences revenue lost. Third, false alarms can be costly since they require engineers responding to the alarm. Frequent false alarms also give rise to "alarm fatigue", where personnel start ignoring alarms since most of them are false.

Business value coming out of a suitable anomaly detection approach, as described in previous paragraphs, is very evident. By improving false alarm ratio, failure prediction time and failure precision

percentage, operator can impact bottom line from two directions. First is through revenue increase that comes from higher customers' satisfaction and loyalty, and second, through cost reduction by reducing interventions related to false alarms.

4 FERARI Dashboard Overview

A dashboard is a control panel in which information has to be displayed visually with combination of text and graphics. The focus is on the graphical display because it is easier to „read“ and faster to understand displayed information in graphical form. In other words, the goal is to present the information in such a way that human eyes and human brains can absorb it quickly, and easily extract the correct and the most important meanings from it.

FERARI Dashboard presents the front-end interface for displaying the most important information in a single-screen. It's an example of Graphical User Interface for helping potential users to track, monitor and analyze the information about the fraud in order to detect fraudulent and make decision about future steps. FERARI Dashboard provides a graphical user interface built with a couple of cutting – edge front-end technology such as AngularJS (with supporting components), Bootstrap, D3.js and Leaflet. The Information on the Dashboard are separated into panels called „widgets“ which display high-level information for quick overview.

FERARI Dashboard consists of following widgets (Figure 5):

- Fraud event grid
- Fraud statistics widget
- Peek/offpeek statistics
- Most frequently called number
- Interactive map
- Navigation pane

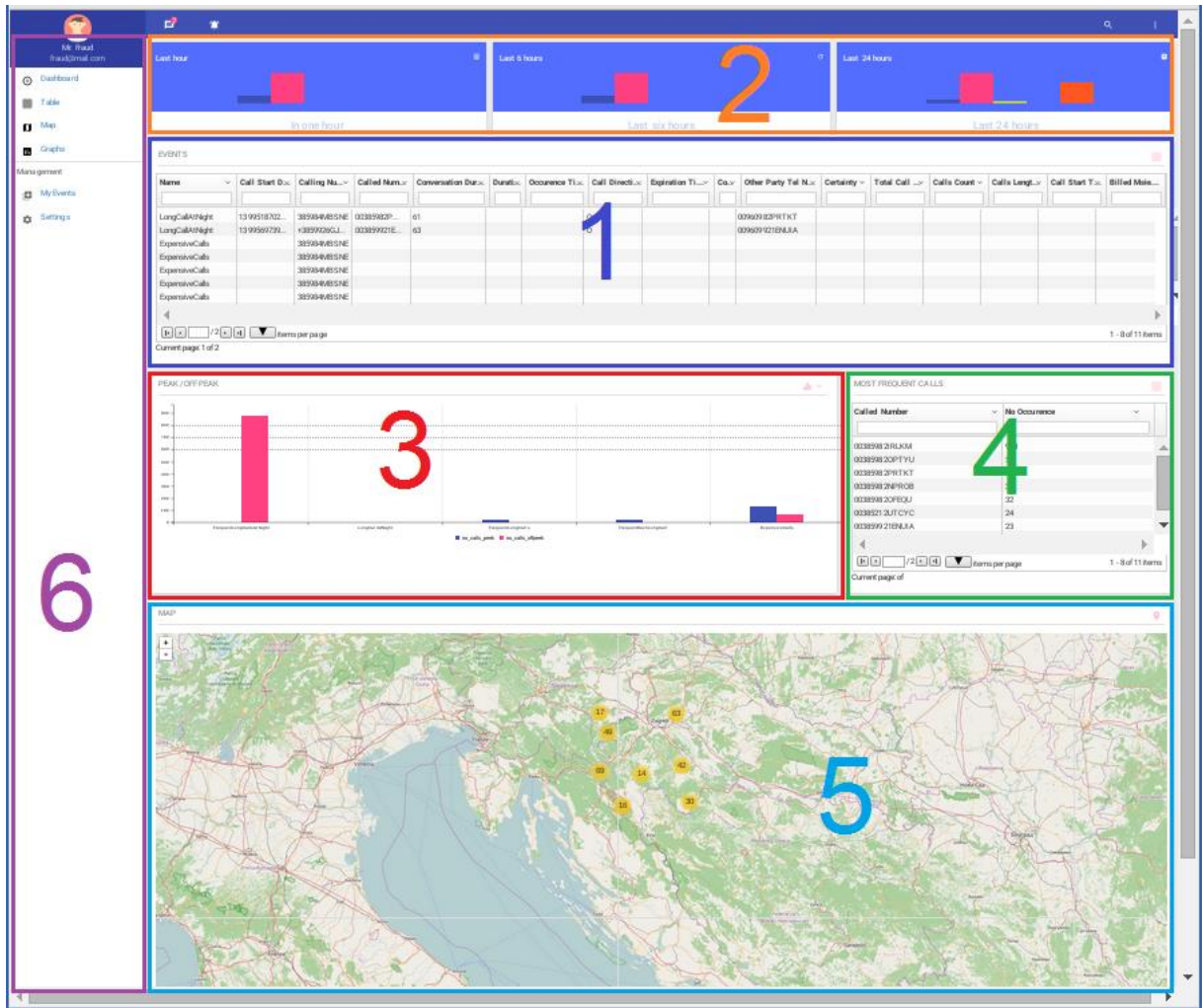


Figure 9: Ferari Dashboard

4.1 Fraud Event Grid

Fraud event grid (Figure 9) is the main widget which represents a grid filled with derived fraud events. Records in grid are shown in real-time as fraud events are detected by Proton system. Every record in the grid is actually a high level information on which user may drill down into low level detailed data (Figure 10). High level overview contains information about event type, call start date and time if event is derived from single call, calling number, called number, conversation duration, event occurrence time, call direction, detection time, total cost, sum of length for all calls that participate in fraud event, count of all calls that participate in fraud event and certainty of fraud event calculated by Proton engine.

EVENTS

Name	Call Start Date	Calling Number	Called Number	Conversation D.	Duration	Occurrence Time	Call Direction	Detection Time	Cost	Other Party Tel.	Certainty
LongCallAtNight	1399003543000	+385984MBSNE	00385983ELZFS	59	0	1446034604230	O	1446034604230	0	00960983ELZFS	1
LongCallAtNight	1400868355000	+3859988YJZBP	00385212UTCYC	2858	0	1449761916665	O	1449761916665	0	00960212UTCYC	1
LongCallAtNight	1399518702000	385984MBSNE	00385982PRTKT	61	0	1449761952786	O	1449761952786	0	00960982PRTKT	1
LongCallAtNight	1399569739000	+3859926GJQKC	003859921ENUJA	63	0	1449761955801	O	1449761955801	0	009609921ENUJA	1
ExpensiveCalls		385984MBSNE			0	1449761973856		1449761973856	0		1
ExpensiveCalls		385984MBSNE			0	1449761973866		1449761973866	0		1
ExpensiveCalls		385984MBSNE			0	1449761973869		1449761973869	0		1
ExpensiveCalls		385984MBSNE			0	1449761973875		1449761973875	0		1

1 / 2 8 Items per page
Current page: 1 of 2
1 - 8 of 16 items

Figure 10: Grid filled with events

Fraud events can be explored in more detail by click on specific event, which opens drill-down screen. Drill-down screen consists of two widgets (Figure 11):

1. Subscriber details
2. Fraud event details

User details

Subscriber Name	Subscriber Address	Subscriber Activation Date	Subscriber Vat Code	Subscriber Status	Subscriber Rate Plan	Rate Plan Change Date	Payment Type	Payment Behavior	Credit Score
Kurt Frank	9483 Riverside	2014-01-1 07:11:00:00:00Z	65540A488075406Y335	IN	27	2013-12-31 17:05:51:43:00Z	Voucher	Regular	1

1

1 - 1 of 1 items
Current page of

Derived Event Name	Billed Missdn	Call Start Date	Calling Number	Called Number	Call Direction	Other Party Tel Number	Other Party Tel Number Prefix	Conversation Duration	Total Call Charge Amount
LongCallAtNight	3859157JLGEK	2014-05-2 07:04:47:07:00Z	+3859157UQHNI	00385982NPRGB	O	00960982NPRGB	960	291	0.23
LongCallAtNight	3859157JLGEK	2014-05-2 07:03:40:07:00Z	+3859157UQHNI	00385982PRTKT	O	00960982PRTKT	960	487	0.75
LongCallAtNight	3859157JLGEK	2014-05-2 07:01:40:07:00Z	+3859157UQHNI	00385982OPTYU	O	00960982OPTYU	960	365	0.43
FrequentLongCallAtNight	3859157JLGEK	2014-05-2 07:01:40:07:00Z	+3859157UQHNI	00385982OPTYU	O	00960982OPTYU	960	365	0.43

2

1 - 4 of 4 items
Current page of

Figure 11: Drill-down screen

Subscriber details widget contains personal information about subscriber such as:

- Name and address
- Subscriber activation date
- VAT code

- Subscriber status
- Subscriber rate plan and last change date of subscriber rate plan
- Payment type, behaviour and subscriber credit score

Source for subscriber data is CRM system, and data is paired with fraud event using Calling number detected by Proton system.

Since fraud event can be derived from multiple calls, fraud events details widget contains list of all calls that participate in specific fraud event and their details:

- Calling number and billed MSISDN
- Called number
- Call start date
- Called number prefix
- Destination zone
- Conversation duration and total call charge amount

All grids are interactive and allow sorting and filtering data by any attribute.

Other widgets contain quantitative and other types of charts that give some general information about fraud events and are described below.

4.2 Fraud statistics widget

Fraud statistics widget shows occurrence of fraud events by type in last hour, last 6 and last 24 hours. It allows end-user to filter desired events for analysis selecting event type by click. Filtered events are shown in main grid.

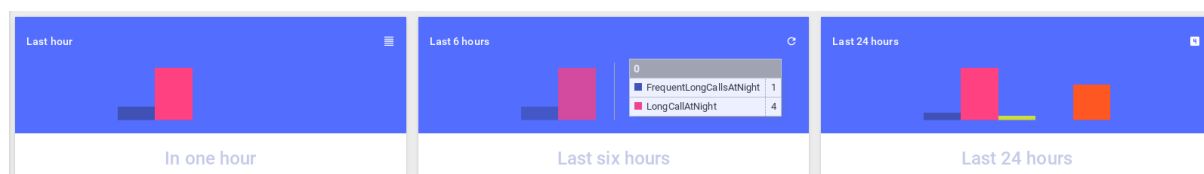


Figure 12: Fraud statistic in last 24 h

4.3 Peek/offpeek statistics

Peek/offpeek statistics chart helps end-user to identify peek/offpeek period within the day when different fraud event types occurs. Statistic is calculated for events that occurred in last 30 days.

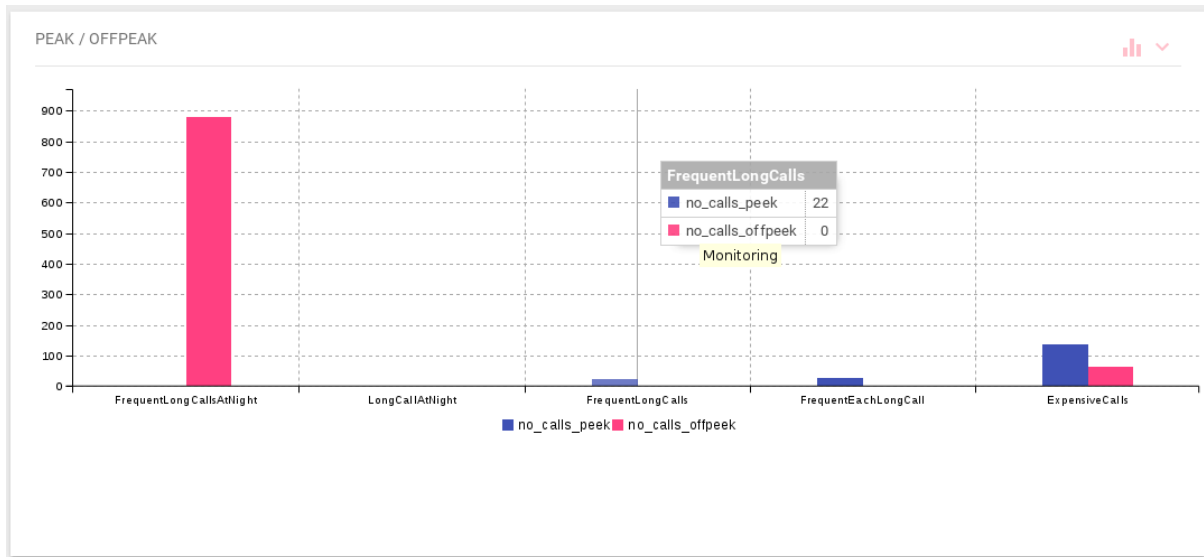


Figure 13: Peek/offpeek statistics

4.4 Most frequently called number

This widget shows information about most frequently called number that was part of derived events in last 24 hours. This information enables end user to consider blocking all future outgoing calls towards specific number.

MOST FREQUENTLY CALLED NUMBER

Called Number	No Occurrence
00385982IRLKM	49
00385982PRTKT	2
00385982OPTYU	2
00385982OFEQU	2
003859585PLXQI	1
003859816OHYYS	1
003859921ENUIA	1
00385983ELZFS	1

1 - 8 of 9 items

Figure 14: Most frequently called number

4.5 Interactive map

Interactive map is widget with markers and marker clusters whose role is to show fraudulent positions on a map. Marker positions are calculated from cell identifier from call details, paired with cell coordinates in database. The Leaflet Markercluster is a JS library for interactive maps that set clustering based on the collection of the markers, which allows zoom-in functionality.

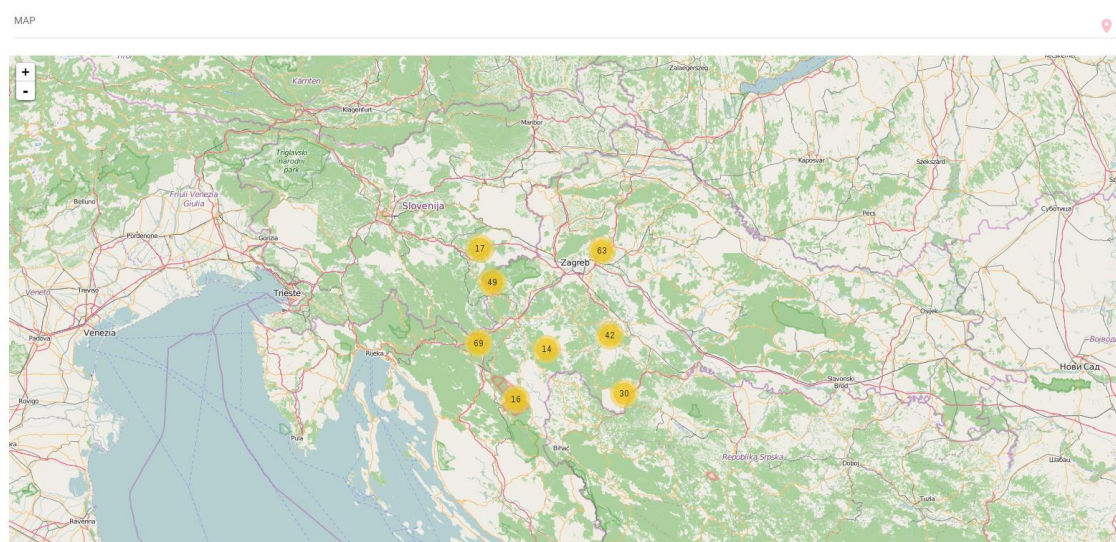


Figure 15: Interactive map

4.6 Navigation pane

Navigation pane allows easy navigation through application and enables end-user to personalize dashboard.

5 Lessons learned

5.1 FHG

We have applied our batch and memory based commercial fraud mining tool to the CDR data in order to see whether new rules for fraud detection could be generated easily from the CDR data. From this the two following lessons were learned.

First, processing larger call volumes to mine fraud patterns requires a streaming solution. With 500GB of RAM the batch based system managed to process calls from up to five days only. Thus the analysis of calls from several weeks for all users at the same time is infeasible with a batch based system with a few TB of RAM. A streaming solution which will store only the essential information from each call will require far less memory and thus allow the analysis of longer time periods of call histories with current hardware.

Second, fraud labels at the aggregation of days hinder the detection of precise rules. Fraud is labeled at subscriber level at the day of detection by human experts. While these labels are precise enough for human experts to communicate the fraud information amongst colleagues, the learning algorithm does not know whether the fraud occurred the same day, a few days earlier or both. In particular all the genuine calls if any prior to fraudulent calls cannot be distinguished by the algorithms, if the fraud label refers to the moment of discovery by experts rather than to the occurrence of fraud.

5.2 HT

Analysis of cloud monitoring data showed that HT's cloud is already very reliable, no failures occurred, cloud growth is lower than expected, and data size is still fairly small.

Due to the constraints of cloud data detected during data analysis, project decided to investigate logs and failure reports from Network Elements; large amount of nodes to be monitored, bigger data quantities, better input to use case, high value for HT.

5.3 IBM

5.3.1.1 *In the scope of WP4*

TEM model is an ongoing and evolving work - Although we have modeled an entire application, there will be more scenarios in the future of unforeseen conditions that will require extensions to the syntax or maybe even more tables. The applicability of the model requires testing on more scenarios. We have the first complete version of the model but future versions will be revisions resulting from experimenting with the model.

TEM validation requires research work - Validation of the model requires research on different techniques and approach. We have started to look into constraint satisfaction problem (CSP) techniques but this is a complete different piece of work and out of the scope of the project.

5.3.1.2 *In the scope of the demo*

Generating sample data for testing is not an easy job - Fraud is a rare occurrence and difficult to find in real data, therefore in order to show different patterns there is a need to manually generate events that match the specific patterns. This requires certain amount of skills and effort.

5.4 Poslovna Inteligencija

In the struggle with Big Data challenges, we were aware of importance how to receive large amount of data arriving very quickly and even how to process them fast. Also, in data stream processing it is desirable to detect important events (event processing) and using existing Big Data streaming platform, it's possible to implement event-stream processing.

According to the EU Data Protection Directive and the architecture of a project, it was necessary to mask data and that was our first focus. Working on the data anonymization led to insights regarding methods and techniques of data masking and further development in commercial purpose.

In fraud detection HT use empirical knowledge of their experts and implications of fraud detection use case will have significant impact on revenue, expenses or the profits of the company. This will be achieved throughout implementation of the in-situ data processing and machine learning algorithms that will help company to detect fraud in real-time, reducing false alarms and manual work of experts.

5.5 Technical University of Crete

Distributed timing issues are critical for inter-cloud CEP: due to the fact that reducing communication during inter-cloud CEP entails lower frequency events being transmitted before higher frequency ones, patterns are not evaluated based on the order of event occurrence as they should. Moreover, events arrive out of order anyway due to transmission lags. These are critical details in determining the detection of complex events within certain time contexts, for instance, when only events belonging in a certain time window need to be taken into consideration.

Coping with inter-cloud pattern evaluation under different policies is a challenging task: for similar reasons with the lesson above, the implementations of certain policies need to be rethought and adapted for distributed event monitoring. Neglecting such timing details for selective policy types affects the output of the distributed event detection compared to a given ground truth.

Distributive, event query execution plans in the probabilistic context are only affected by the confidence threshold: should it exist. The rest of the probabilistic context and (un)certainity flavors only affect the corresponding derived event attributes. The major effect in the production of query execution

plans comes from a potential confidence threshold. This is because such a threshold affects the qualifying pattern matches and thus affects the selectivity of a query operator. In turn, Operator's (complex event) selectivity is statistic information that the CEP optimizer takes into account upon examining alternative query execution plans.

5.6 Technion

Working on implementing the distributed counter algorithm on real-world CDR data provided by HT has led to several important insights regarding the application of in-situ monitoring to real-world telco data. First, as opposed to laboratory settings, which consist of at most several hundred sites, the data provided by HT consists of over 18,000 cell towers that where associated with over 6000 base stations. Only a small subset of these sites, however, are relevant to any given subscribers. Furthermore, the set of sites that servers a given subscriber in a given day is not fixed. This has led us to adjust our monitoring algorithm to efficiently handle a dynamic set of site.

A second insight gained while working on this data is when resolving constraint violations, it is worthwhile retaining some slat at the coordinator. This enables resolving a constraint violation at a certain site without involving other sites, and leads to improved performance in terms of communication reduction.

6 Summary and future steps

In Year 2 of the FERARI project we have prepared the test bed, delivered data for development and started collecting data that shall be used in the testing procedure.

Our goal in FERARI is to bring event processing much closer to the business users, to enable them to use benefits of complex event processing, and get results of data analysis much faster than they have right now – to be able to react more adequately. On the other hand, users will have information about possible failure in advance, which will give them ability to react proactively. Combination of those two approaches will save significant time and resources in process.

During year three of the project we will concentrate for detailed system setup, preparation of test procedure, detailed testing and evaluation of FERARI ecosystem.

7 References

- [1] Baesens B., Vlasselaer V.V., Verbeke W. (2015), *Fraud Analytics Using Descriptive, Predictive, and Social Network Techniques*, Wiley Publishing, Inc., 2015, pp.
- [2] Ferre, L. URL: https://commons.wikimedia.org/wiki/File:XDSL_Connectivity_Diagram_en.svg

8 Appendix A - Hardware configuration

HP ProLiant DL380p Gen8 12 LFF Configure-to-order Server
HP DL380p Gen8 Intel Xeon E5-2660v2 (2.2GHz/10-core/25MB/95W) FIO Processor Kit
HP DL380p Gen8 Intel Xeon E5-2660v2 (2.2GHz/10-core/25MB/95W) Processor Kit
HP 16GB (1x16GB) Dual Rank x4 PC3-12800R (DDR3-1600) Registered CAS-11 Memory Kit
8 x HP 3TB 6G SATA 7.2K rpm LFF (3.5-inch) SC Midline 1yr Warranty Hard Drive
HP 2GB P-series Smart Array Flash Backed Write Cache
HP Ethernet 1Gb 4-port 331FLR FIO Adapter
HP Ethernet 10Gb 2-port 561T Adapter
HP 750W Common Slot Platinum Plus Hot Plug Power Supply Kit
HP iLO Advanced including 1yr 24x7 Technical Support and Updates Single Server License
HP Insight Control including 1yr 24x7 Support ProLiant ML/DL/BL-bundle Tracking License
HP 2U LFF BB Gen8 Rail Kit with CMA