Making Anomaly Detection for Mobile Devices Practical and Scalable: Challenges and Research Perspectives

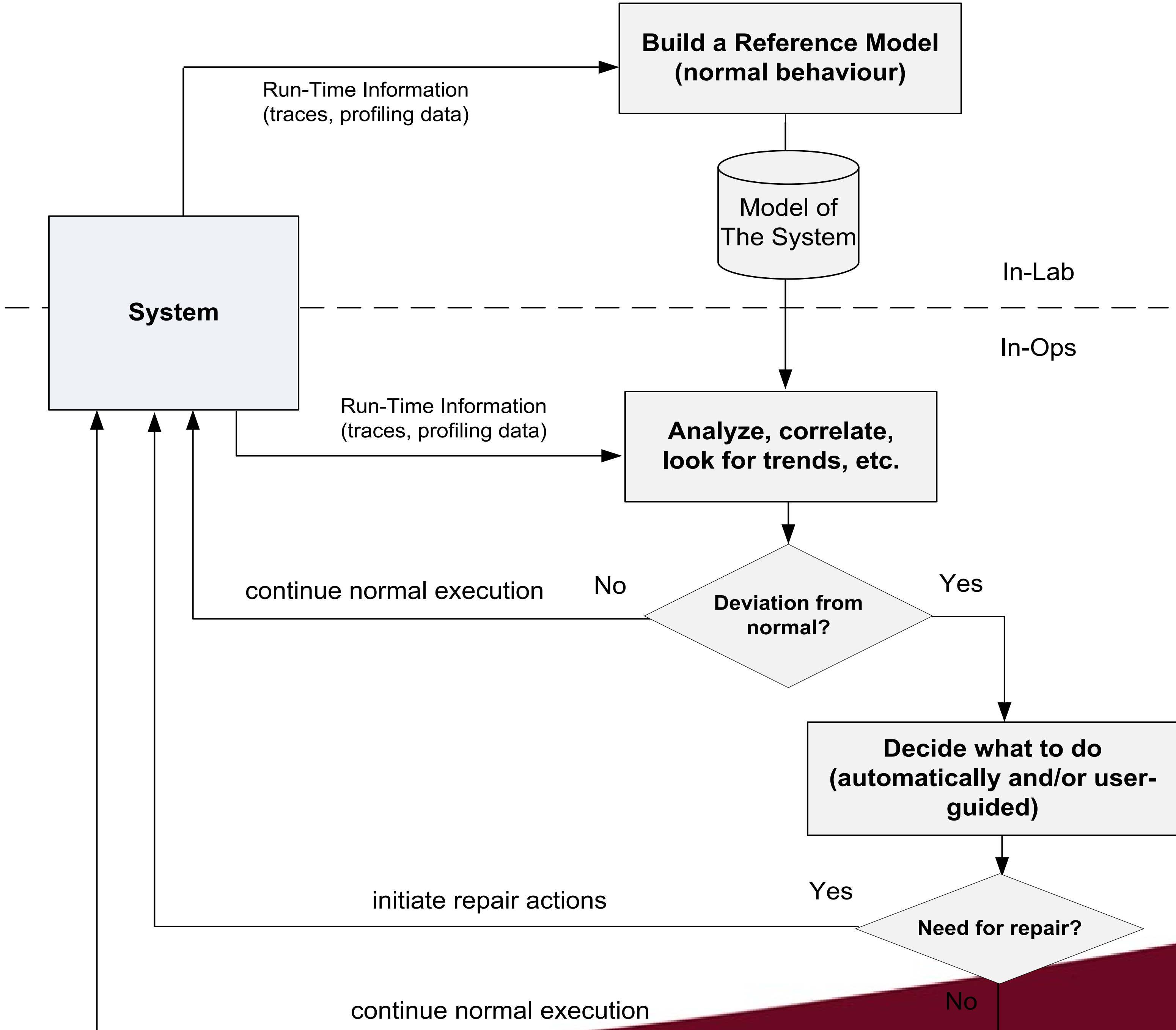
Wahab Hamou-Lhadj, PhD, ing.

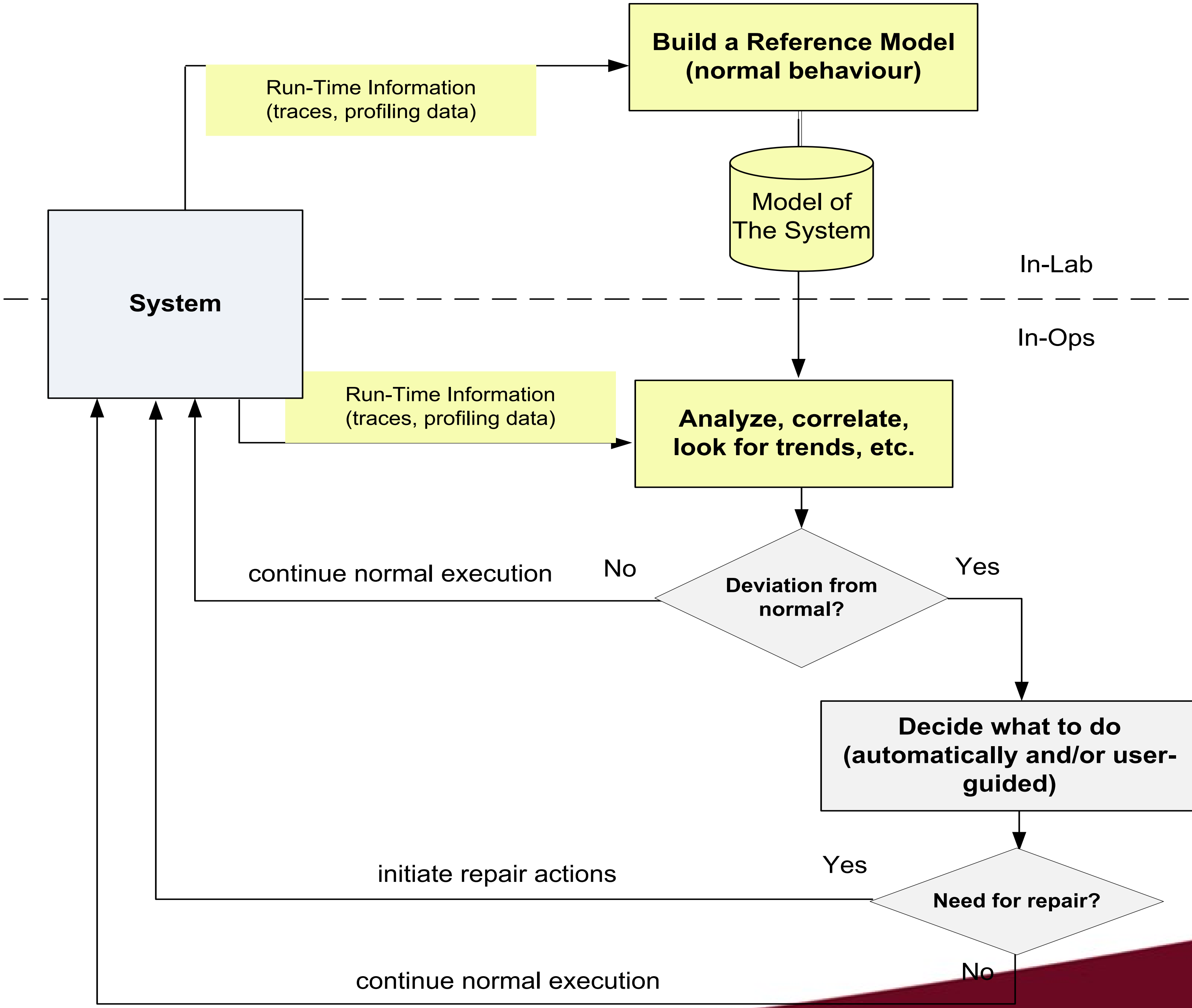
Software Behaviour Analysis Research (SBA) Lab
Concordia University
Montréal, QC, Canada
http://www.ece.concordia.ca/~abdelw

DNCMS, SRDS 2015
McGill University, Montréal, ON
Sep. 28, 2015

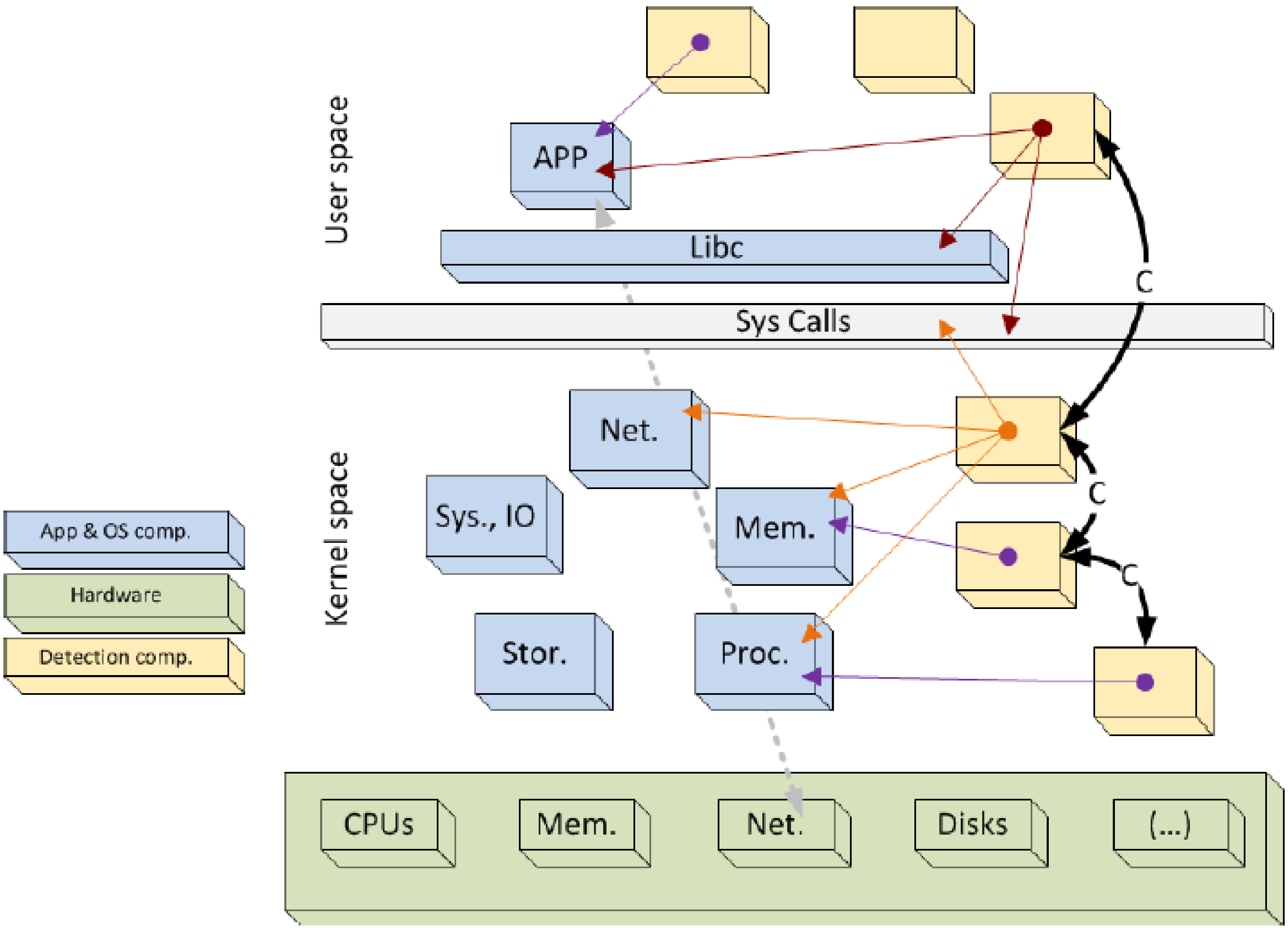
Background on Anomaly Detection Systems (ADS)

- Goal:
  - To monitor computer or network activities for signs of intrusions
- Signature-based Detection (anti-viruses)
  - Looks for known patterns
  - Detects only known attacks
- Anomaly Detection
  - Looks for deviations from normal behavior
  - Detects even unknown attacks (zero day exploits)

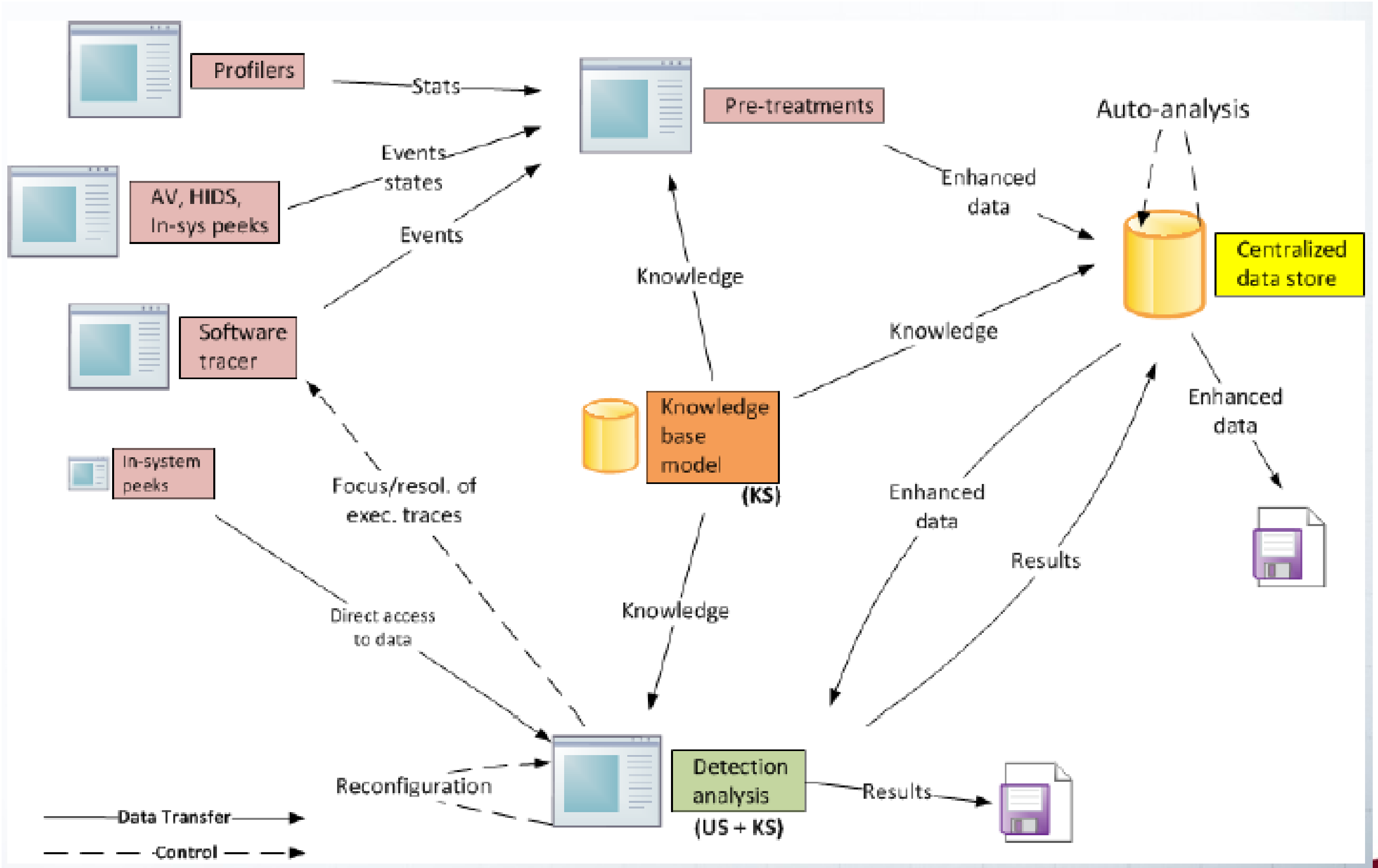

Build a Reference Model
(normal behaviour)
Run-Time Information
(traces, profiling data)
Model of
The System
In-Lab
System
In-Ops
Run-Time Information
(traces, profiling data)
Analyze, correlate,
look for trends, etc.
continue normal execution
No
Deviation from
normal?
Yes
Decide what to do
(automatically and/or user-
guided)
initiate repair actions
Yes
Need for repair?
No
continue normal execution


Build a Reference Model
(normal behaviour)
Run-Time Information
(traces, profiling data)
Model of
The System
In-Lab
System
In-Ops
Run-Time Information
(traces, profiling data)
Analyze, correlate,
look for trends, etc.
continue normal execution
No
Deviation from
normal?
Yes
Decide what to do
(automatically and/or user-
guided)
initiate repair actions
Yes
Need for repair?
No
continue normal execution

What can we monitor?



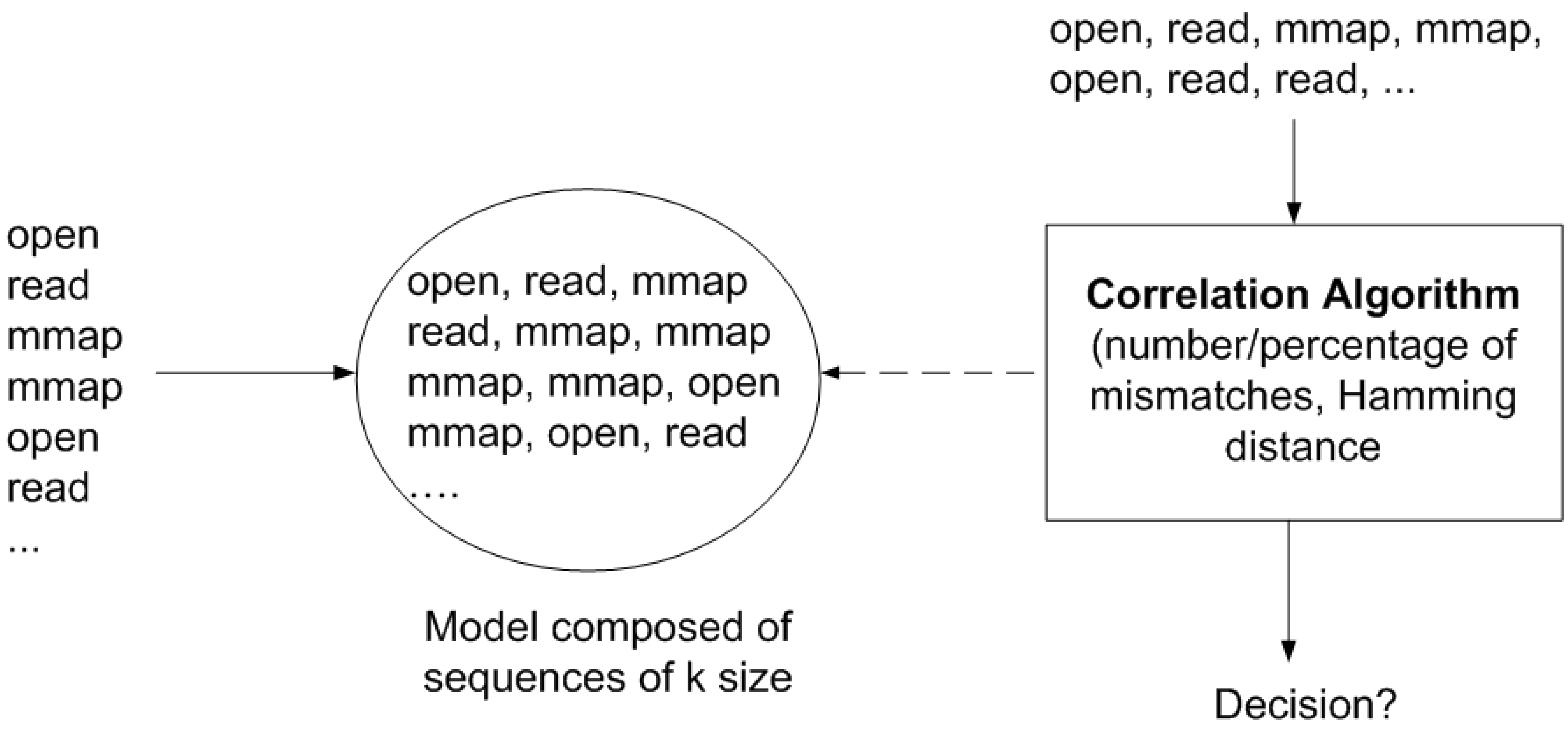
A Complex Monitoring Framework



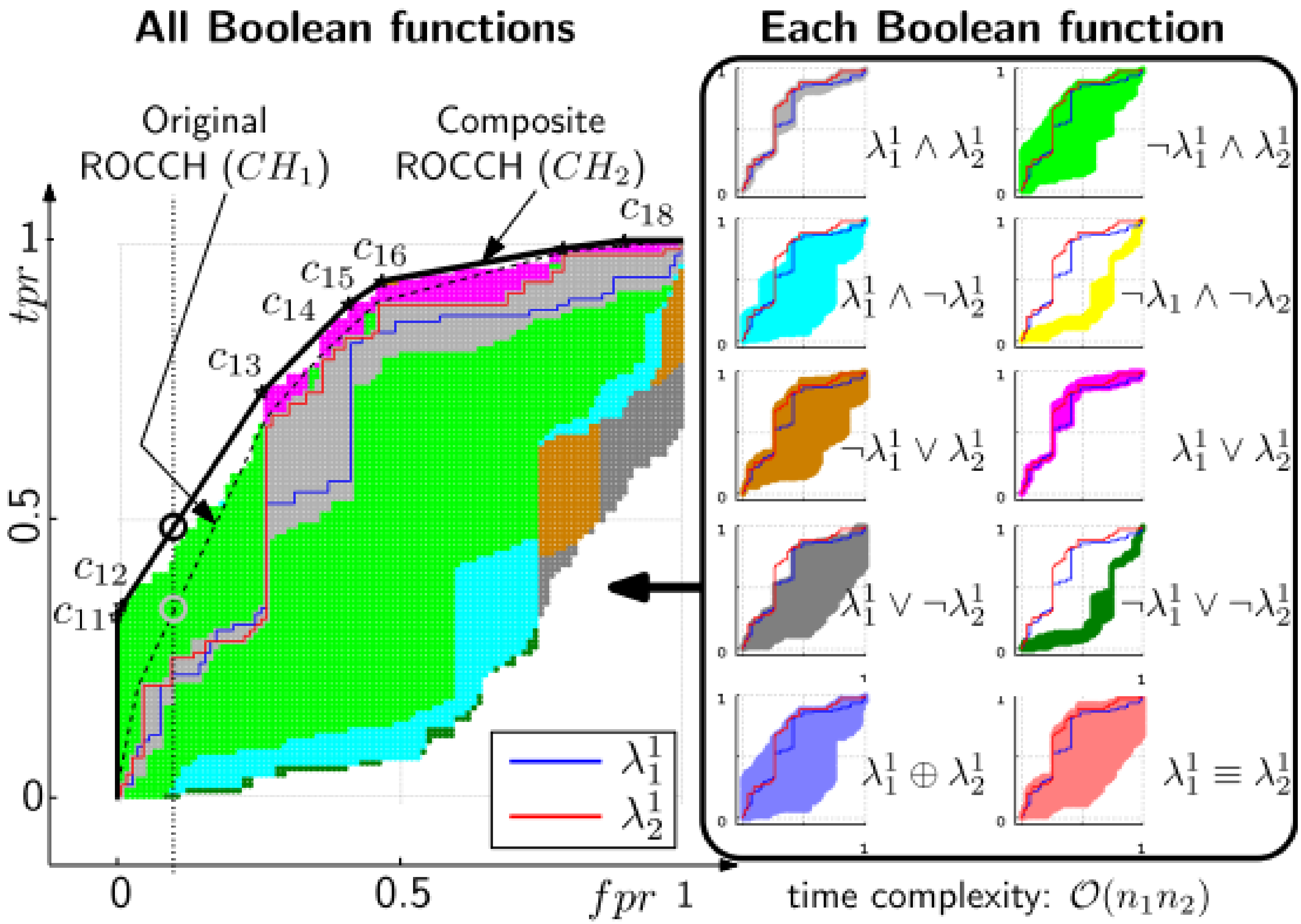
Existing Work

- Several techniques have been used to model the normal behavior of a system
  - Sliding window technique
  - HMM
  - Neural networks (two-class)
  - Clustering
  - Varied length n-gram technique
  - Context Free Grammar
  - Data fusion
  - Computational intelligence

Example: Sliding Approach (STIDE)



Incremental Boolean Combination of HMMs



Advanced Host-Level Surveillance

DEFENCE R&D DÉFENSE

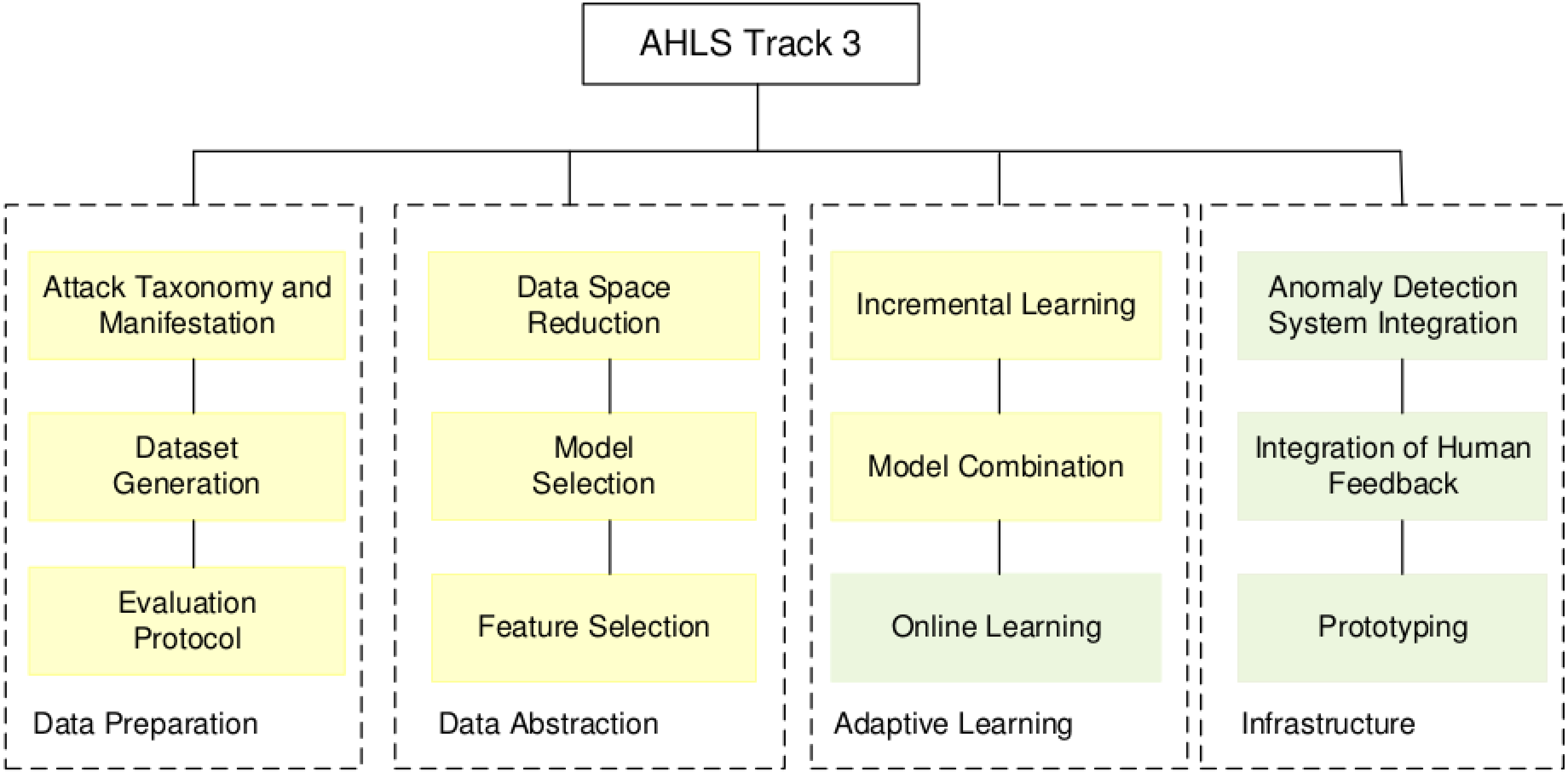
POLYTECHNIQUE MONTRÉAL
LE GÉNIE EN PREMIÈRE CLASSE
UT TENSIO SIC VIS

UNIVERSITÉ Concordia UNIVERSITY

ERICSSON

UNIVERSITY OF TORONTO

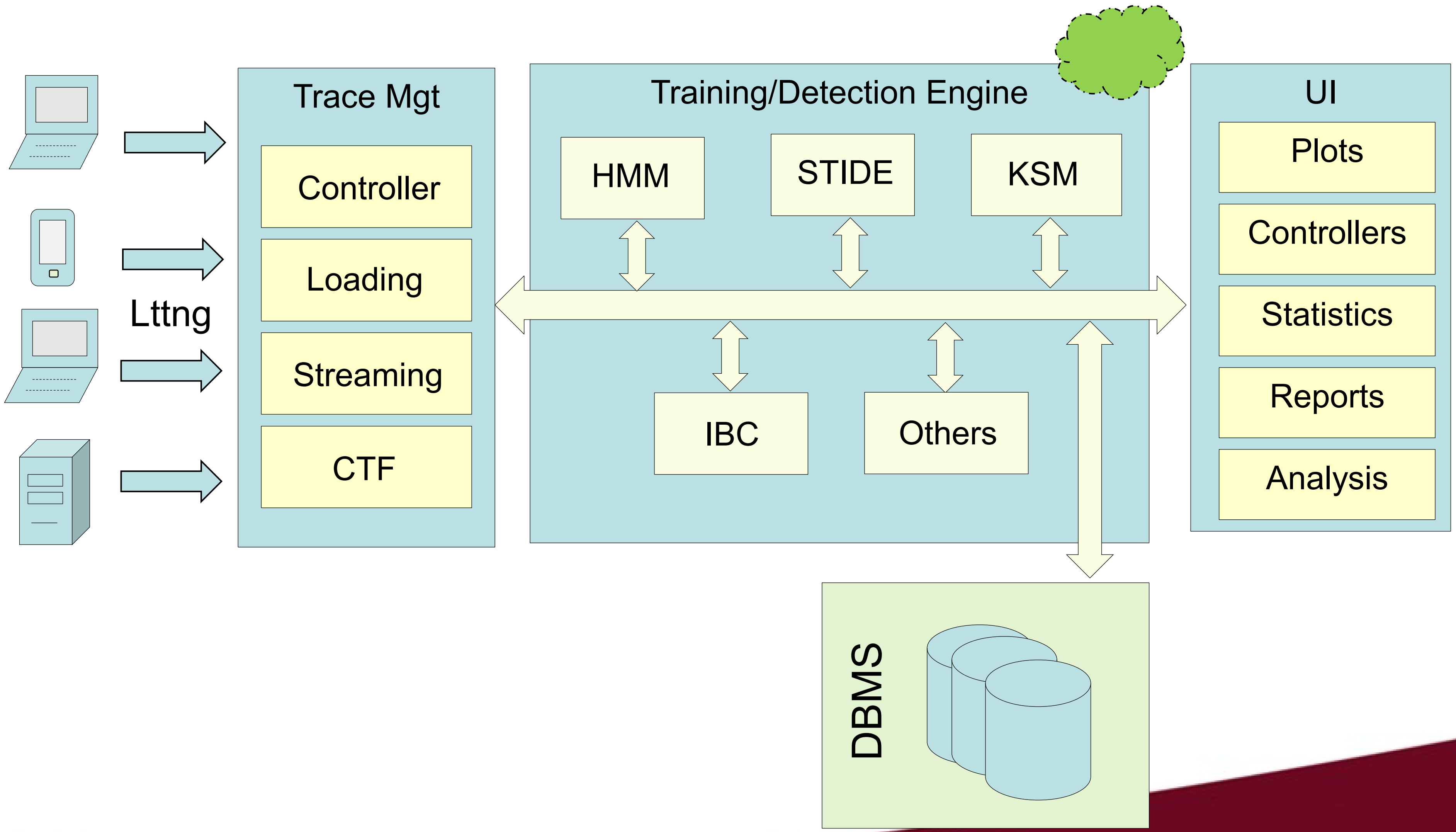
Advanced Host-Level Surveillance



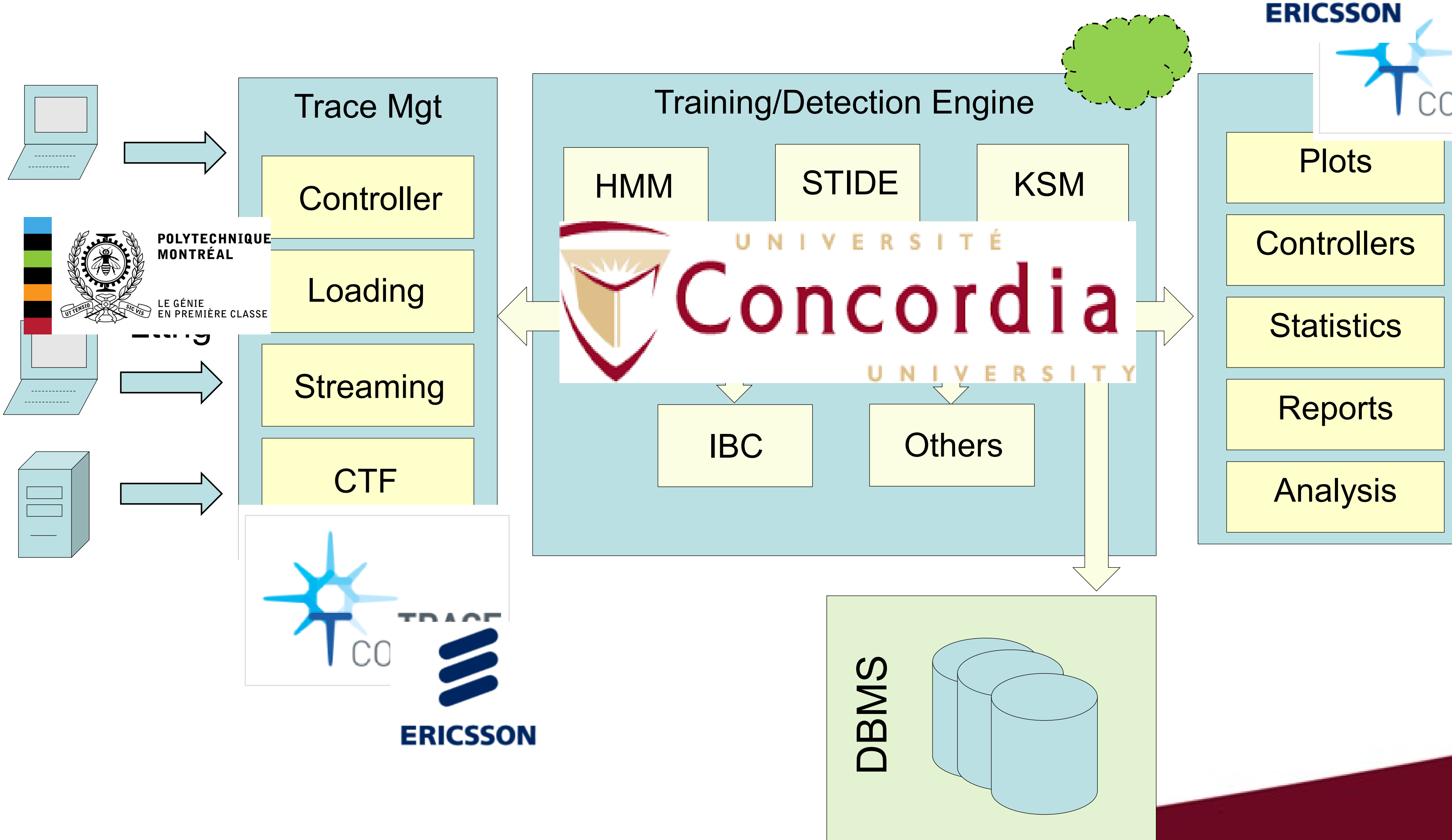
TotalADS: An Integrated Anomaly Detection System

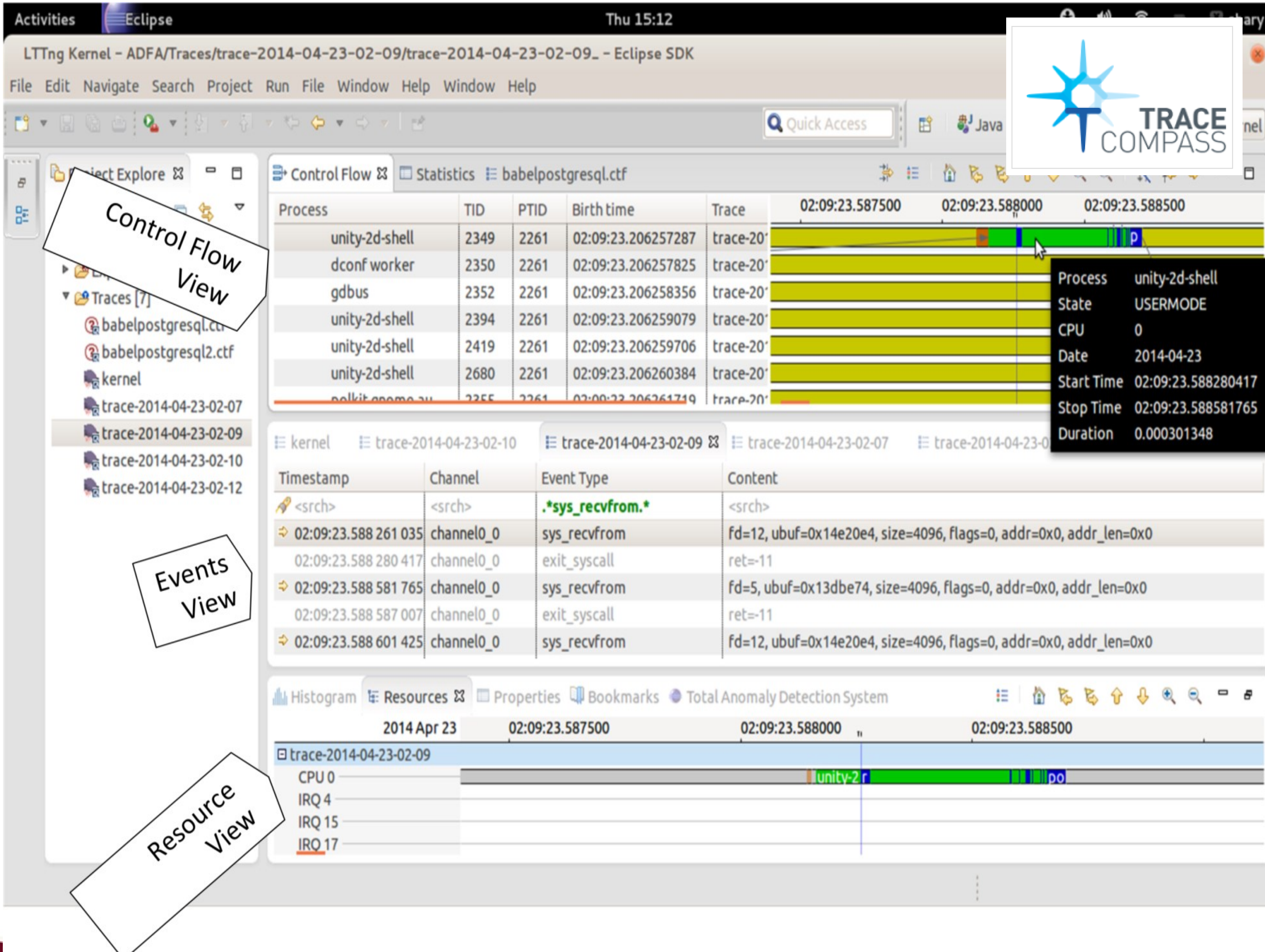
- Eclipse Plug-in
- Open Source
- Based on TraceCompass (A powerful tracing infrastructure)
- Supports STIDE, HMM, KSM, IBC, and others
- Supports a combination of classifiers
- Supports trace analysis and forensic analysis
- Supports CTF (Common Trace Format), standardized by the Linux Foundation

TotalADS Architecture

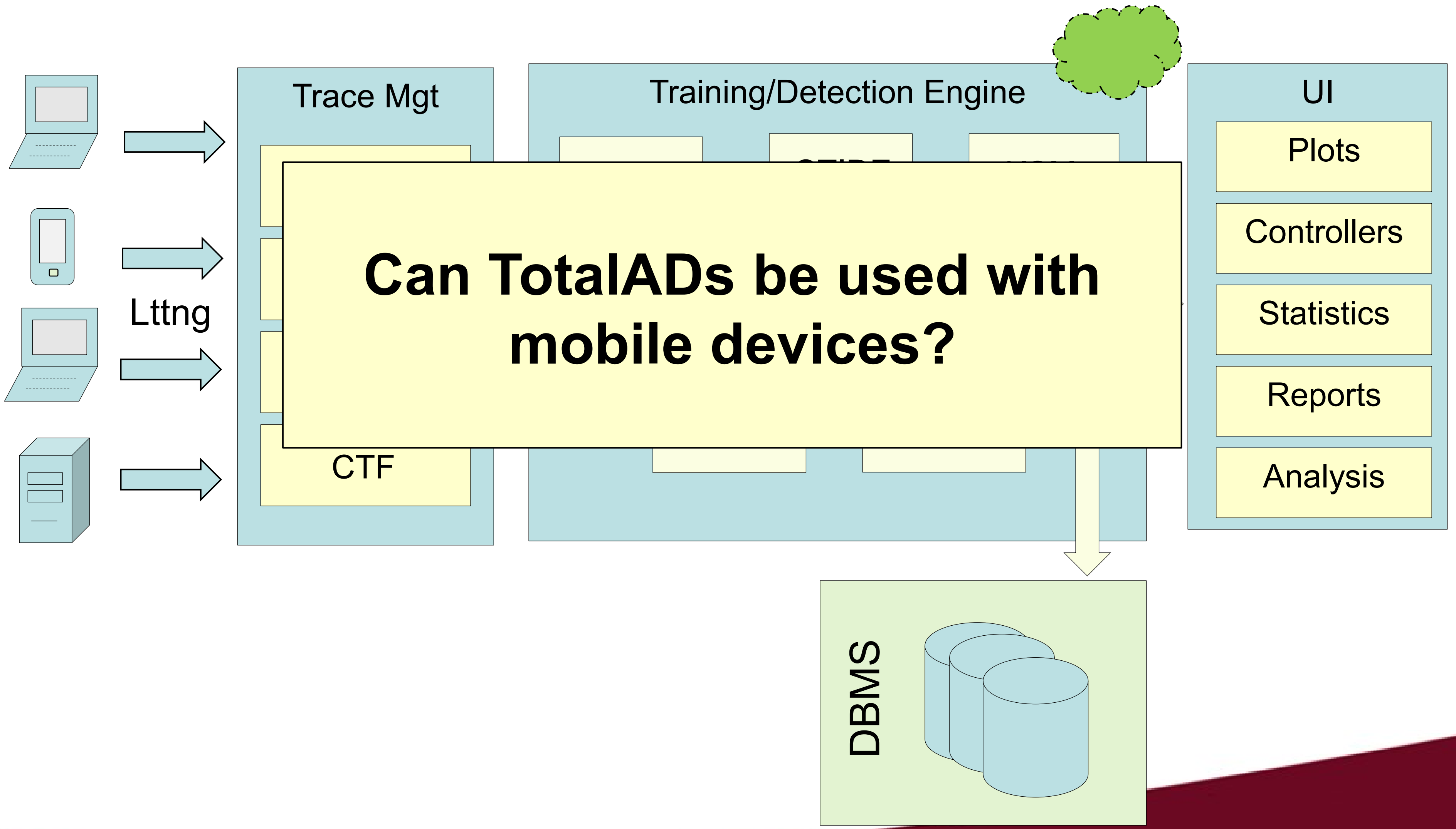


TotalADS Architecture




TRACE COMPASS
Control Flow View
Events View
Resource View
LTTng Kernel - ADFA/Traces/trace-2014-04-23-02-09/trace-2014-04-23-02-09_ - Eclipse SDK
Control Flow
Statistics
babelpostgresql.ctf
Process
unity-2d-shell
State
USERMODE
CPU
0
Date
2014-04-23
Start Time
02:09:23.588280417
Stop Time
02:09:23.588581765
Duration
0.000301348
Timestamp
Channel
Event Type
Content
Histogram
Resources
Properties
Bookmarks
Total Anomaly Detection System
trace-2014-04-23-02-09
CPU 0
IRQ 4
IRQ 15
IRQ 17

TotalADS Architecture



Can TotalADs be used with mobile devices?

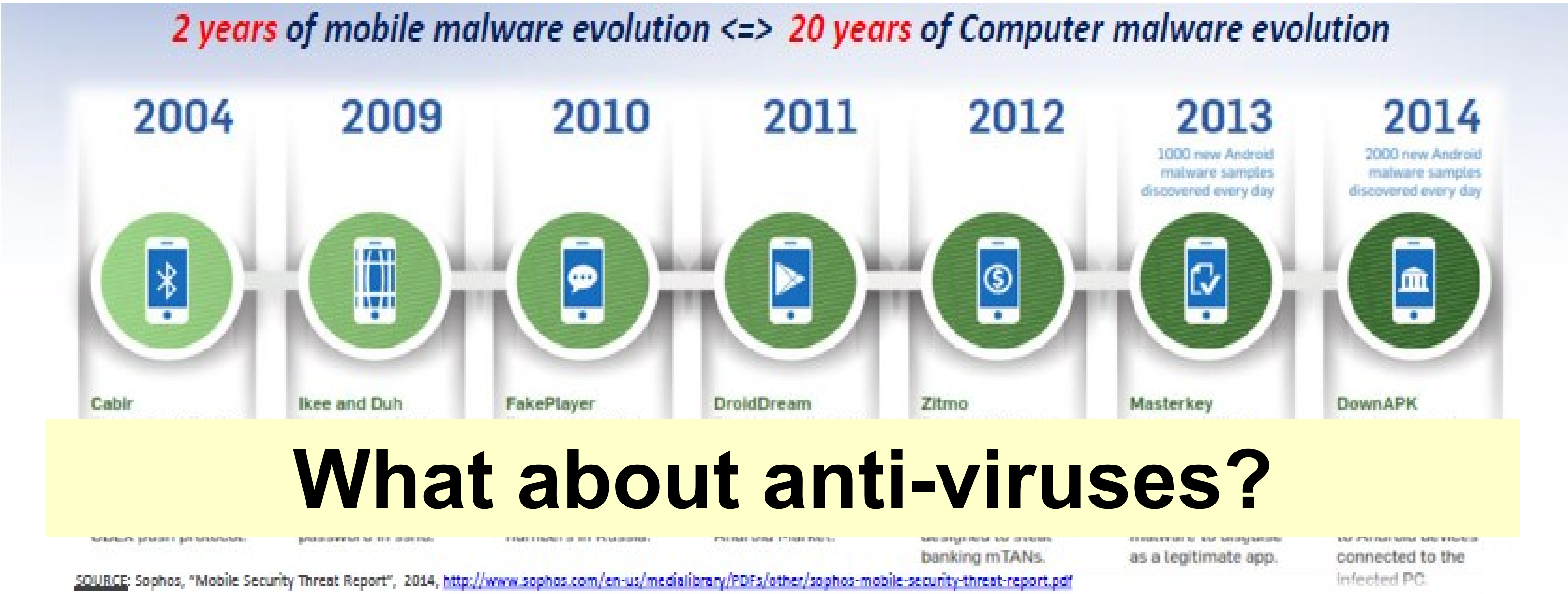
Why mobile devices?

F-Secure 2014: "Android devices are the more popular target for attacks with 294 new threat families or variants"

General-purpose small devices

- **2009:Psyb0t**
  - Linux-based routers, DSL modems
- **2010:Stuxnet**
  - Industrial control systems (ICS)
- **2012: DNSChanger**
  - Computers and routers
- **2013:GPS attack**
  - GPS based systems
- **2013:Linux.Darlloz**
  - Cameras, set-top boxes
- **2014:The Moon**
  - Linksys routers

Why mobile devices?



What about anti-viruses?

F-Secure 2014: *"Android devices are the more popular target for attacks with 294 new threat families or variants"*

General-purpose small devices

- **2009:Psyb0t** — Linux-based routers, DSL modems
- **2010:Stuxnet** — Industrial control systems (ICS)
- **2012: DNSChanger** — Computers and routers
- **2013:GPS attack** — GPS based systems
- **2013:Linux.Darlloz** — Cameras, set-top boxes
- **2014:The Moon** — Linksys routers

Fraunhofer

AISEC

FRAUNHOFER RESEARCH INSTITUTION FOR
APPLIED AND INTEGRATED SECURITY

ON THE EFFECTIVENESS OF MALWARE PROTECTION ON ANDROID

AN EVALUATION OF ANDROID ANTIVIRUS APPS

RAFAEL FEDLER, JULIAN SCHÜTTE, MARCEL KULICKE

04/2013

Fraunhofer

AISEC

FRAUNHOFER RESEA N FOR

APPLIED AND INTEG

Tools

avast
AVG
BitDefender
ESET
F-Secure
Kaspersky
Lookout
McAfee
Norton
Sophos
Trend Micro

Family of attacks

- AnserverBot
- DroidKungFuUpdate
- FakeInst
- GingerMaster
- JiFake
- OpFake
- Plankton
- SpyEye-in-the-Mobile (SpitMo)
- SuperClean
- Zeus-in-the-Mobile (ZitMo)
- Our own proof of concept malware

ON THE EFF S OF PROTECTION ROID

AN EVALUATION OF ANDROID ANTIVIRUS APPS

RAFAEL FEDLER, JULIAN SCHÜTTE, MARCEL KULICKE

04/2013

Fraunhofer

AISEC

FRAUN

APPLI

	1	2	3	4	5	6	7	8	9	10	Total
avast	✓	–	✓	✓	–	✓	✓	–	✓	–	6/10
AVG	–	–*	–	✓	–	✓	–*	–	–	–	2/10
BitDefender	✓	–	–	✓	–	✓	✓	–	–	–	4/10
ESET	✓	✓	✓	✓	✓	✓	✓	✓	✓	–	9/10
F-Secure	✓	–	✓	✓	–	–	✓	✓	✓	–	6/10
Kaspersky	✓	–	–	✓	–	–	✓	–	–	–	3/10
Lookout	✓	–	–	✓	–	✓	✓	✓	✓	✓	7/10
McAfee	✓	–	–	✓	–	–	–	–	–	–	2/10
Norton	✓	–	–	✓	–	✓	–	–	–	–	3/10
Sophos	–	–	–	–	–	–	–	–	–	–	0/10
Trend Micro	✓	–	–	✓	–	–	–	–	✓	–	3/10

Table 3.1: Detection rates for Test Case 1 (–* denotes that the sample has been detected as aggressive adware, not as malware)

ON
PRO
AN E

RAFAEL FEDLER, JULIAN SCHÜTTE, MARCEL KULICKE

04/2013

Fraunhofer
AISEC

	1	2	3	4	5	6	7	8	9	10	Total
avast	✓	–	✓	✓	–	✓	✓	–	✓	–	6/10
AVG	–	–*	–	✓	–	✓	–*	–	–	–	2/10
BitDefender	✓	–	–	✓	–	✓	✓	–	–	–	4/10
Trend Micro	✓	–	–	✓	–	–	–	–	✓	–	3/10

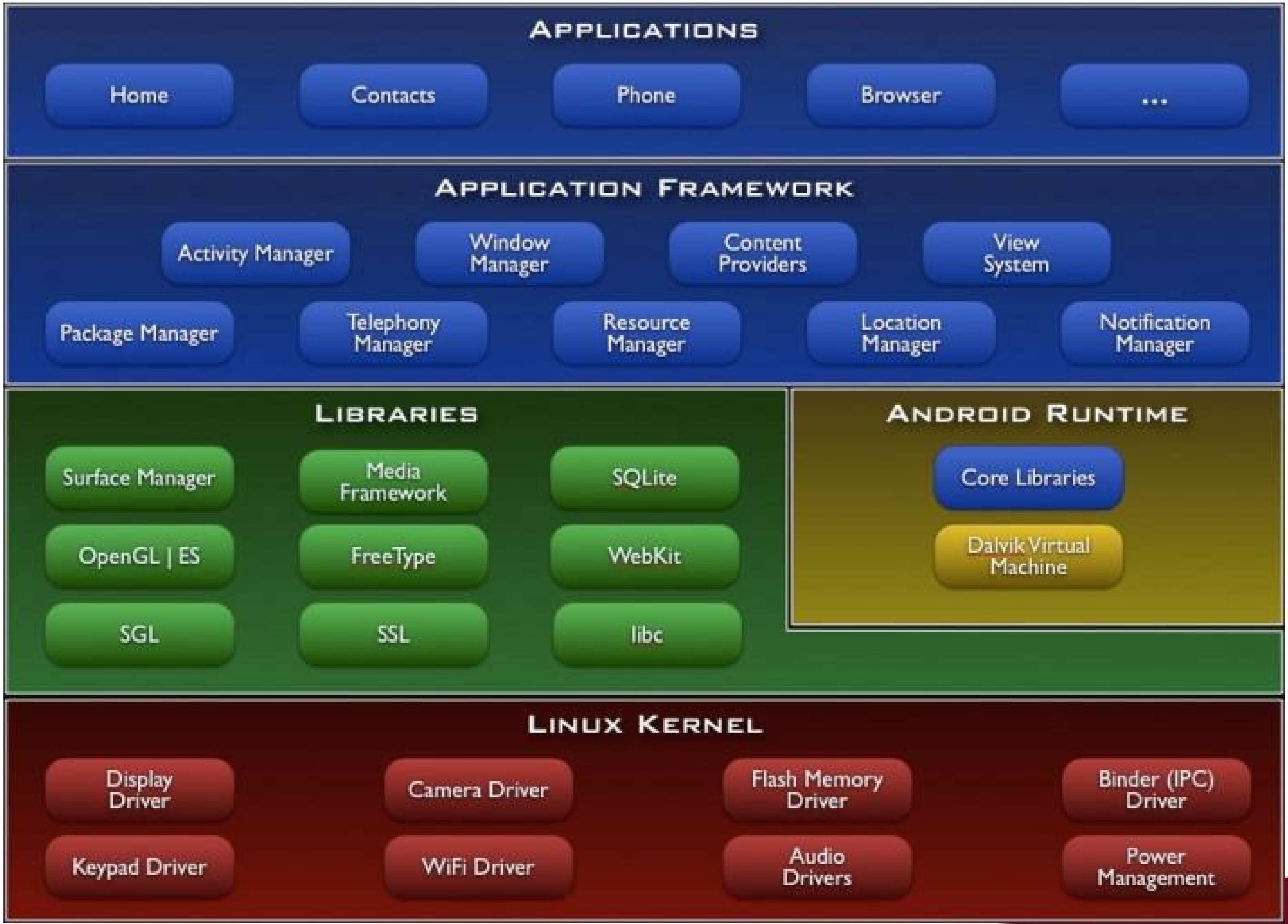
Table 3.1: Detection rates for Test Case 1 (–* denotes that the sample has been detected as aggressive adware, not as malware)

antivirus software is tested for recognizing known malware samples, our test setup considers the ability to cope with typical malware distribution channels, infection routines, and privilege escalation techniques. We found that it is easy for malware to evade detection by most antivirus apps with only trivial alterations to their package files.

RAFAEL FEDLER, JULIAN SCHÜTTE, MARCEL KULICKE

04/2013

Android Architecture



nikhilsalunkedev. 2014." What is Android?". En ligne. <https://nikhilsalunkedev.wordpress.com/2014/03/21/what-is-android/>. Consulté le 12 mai 2014.

DroidTrace: A Ptrace Based Android Dynamic Analysis System with Forward Execution Capability (Zheng 2014)

- An approach for exploring the behaviour of apps with dynamic payloads (reflection, native code, etc.)
  - Aimed to detect code injection and manipulation attacks using repacking and dynamic payload
- Among the analyzed 1260 malware samples, 1083 of them (86%) were repackaged versions of legitimate apps with malicious payloads (Zhou 2012)
  - indicating repackaging is a favorable vehicle for mobile malware propagation.
- Even without code obfuscation, it has been found that about 5% to 13% of apps in third party app markets are the plagiarism of legitimate applications

DroidTrace: A Ptrace Based Android Dynamic Analysis System with Forward Execution Capability (Zheng 2014)

Reverse engineer .dex files to uncover functions with dynamic loading

Generate Application Control Flow Graphs for each function with dynamic loading

Execute the apps and generate Strace syscall traces with selected syscalls

Analyze the behaviour of apps with dynamic loading

A Framework for Evaluating Mobile App Repackaging Algorithms (Huang 2014)

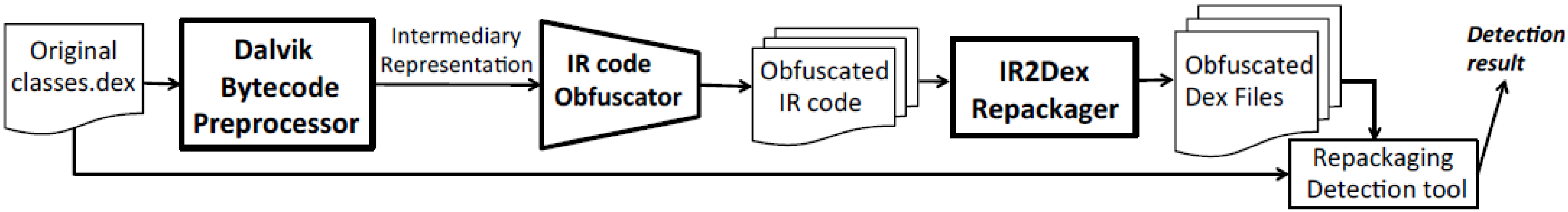
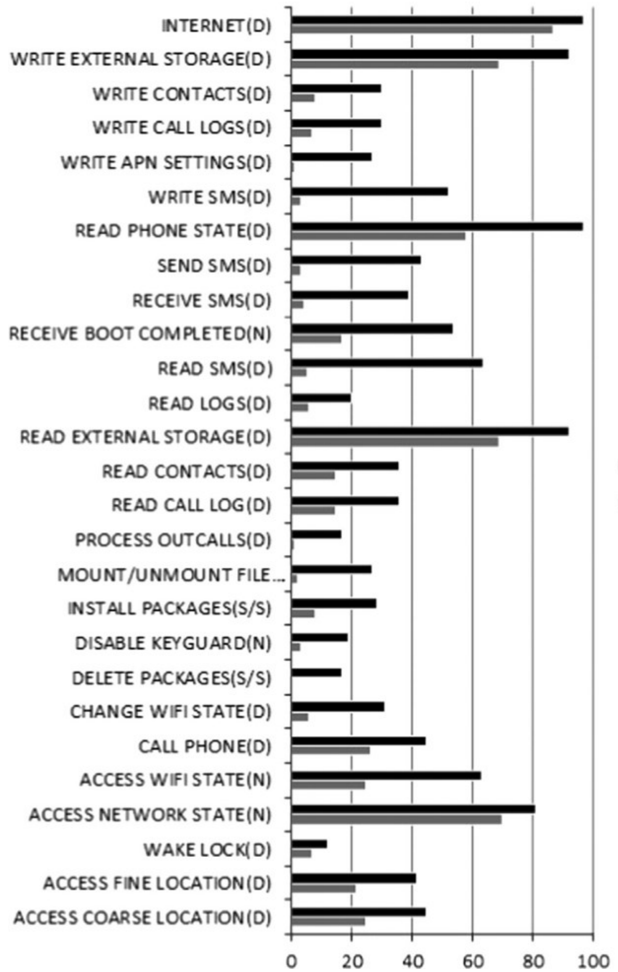


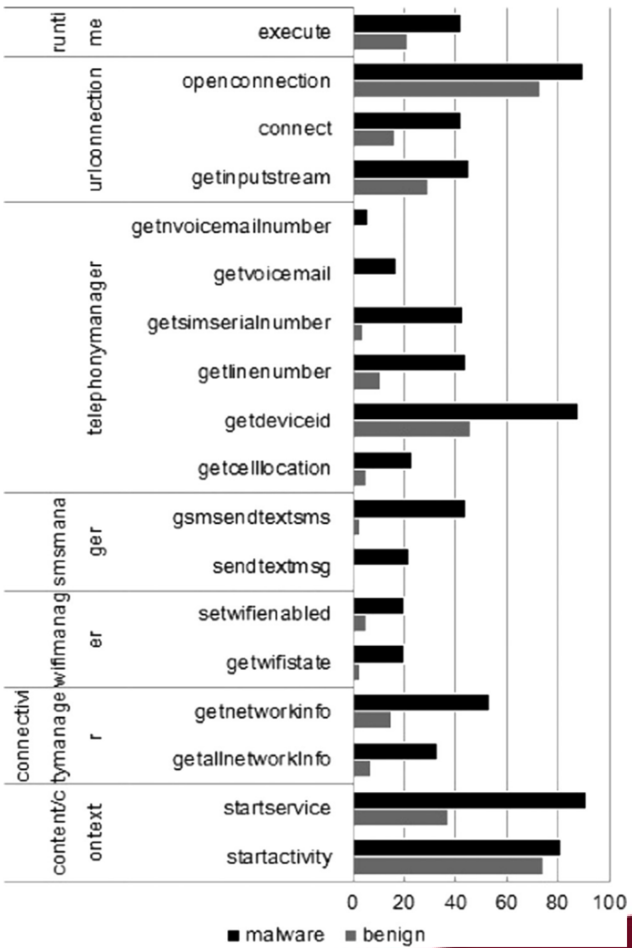
Fig. 3. A Framework for evaluating the obfuscation resilience of repackaging detection algorithms

- SandMarks is used to simulate various obfuscation techniques
- Case study on AndroGuard,an Android application repackaging algorithm shows that the tool is not resilient to control-flow manipulation

Android Malware Detection Using a Multifeature Collaborative Decision Fusion (Sheen 2014)

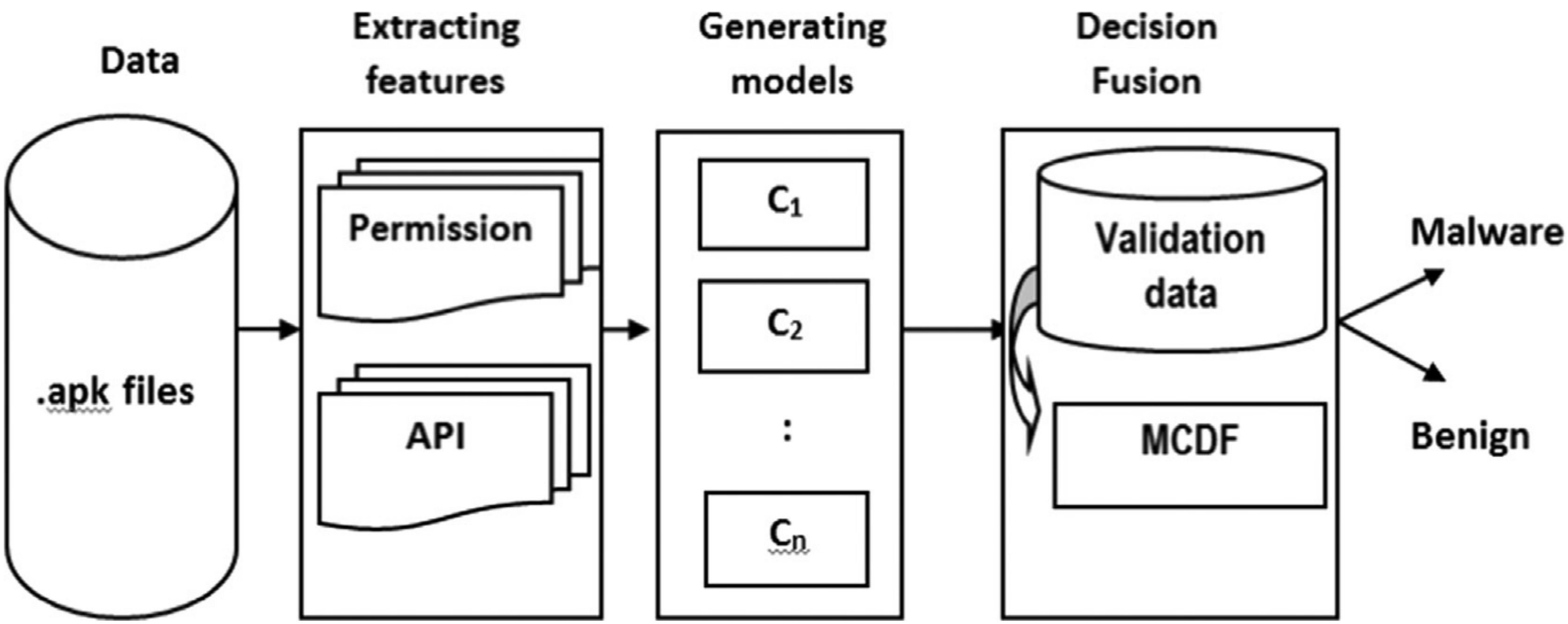


Most frequently accessed permissions

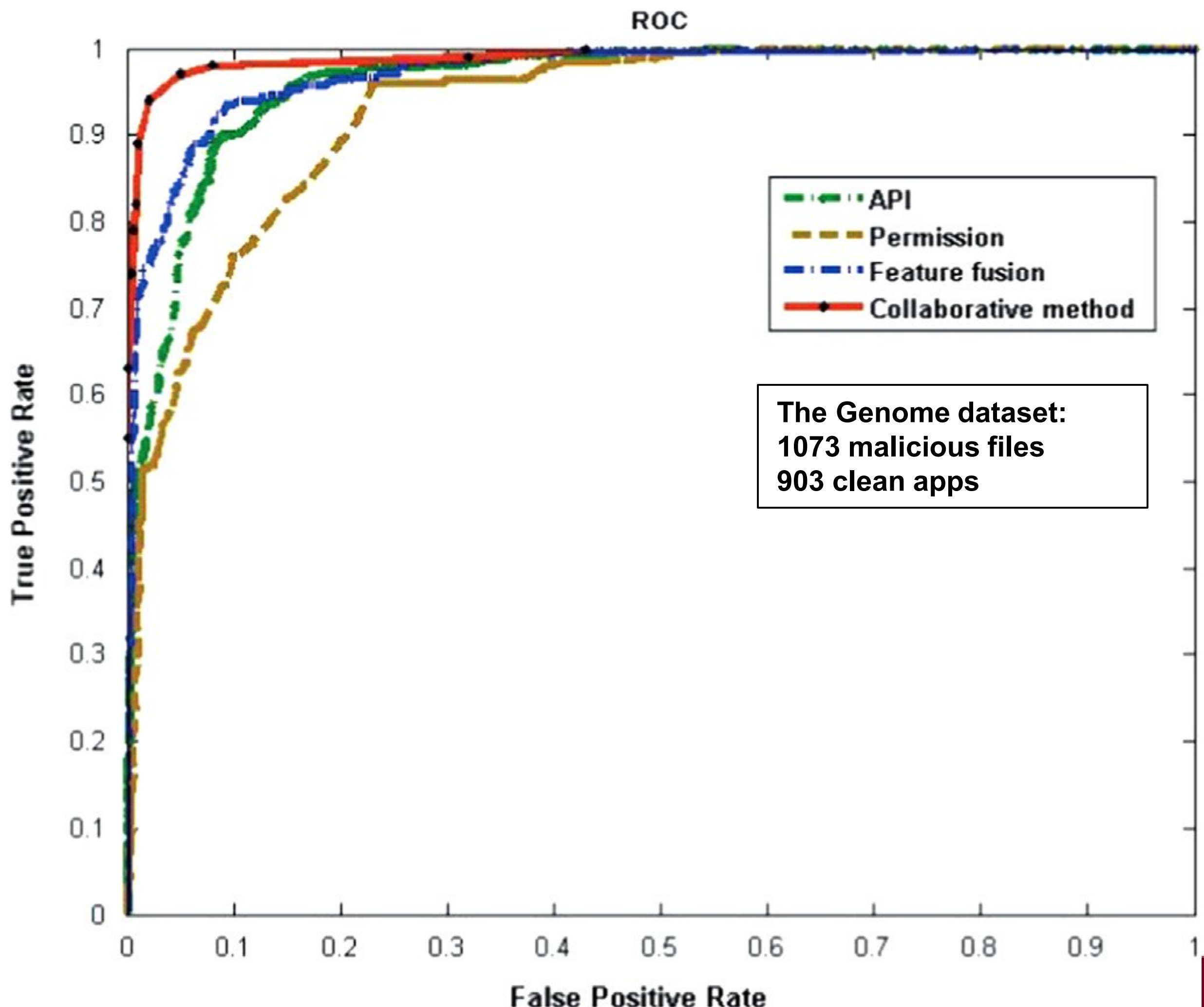


Most frequently accessed API calls

Android Malware Detection Using a Multifeature Collaborative Decision Fusion (Sheen 2014)

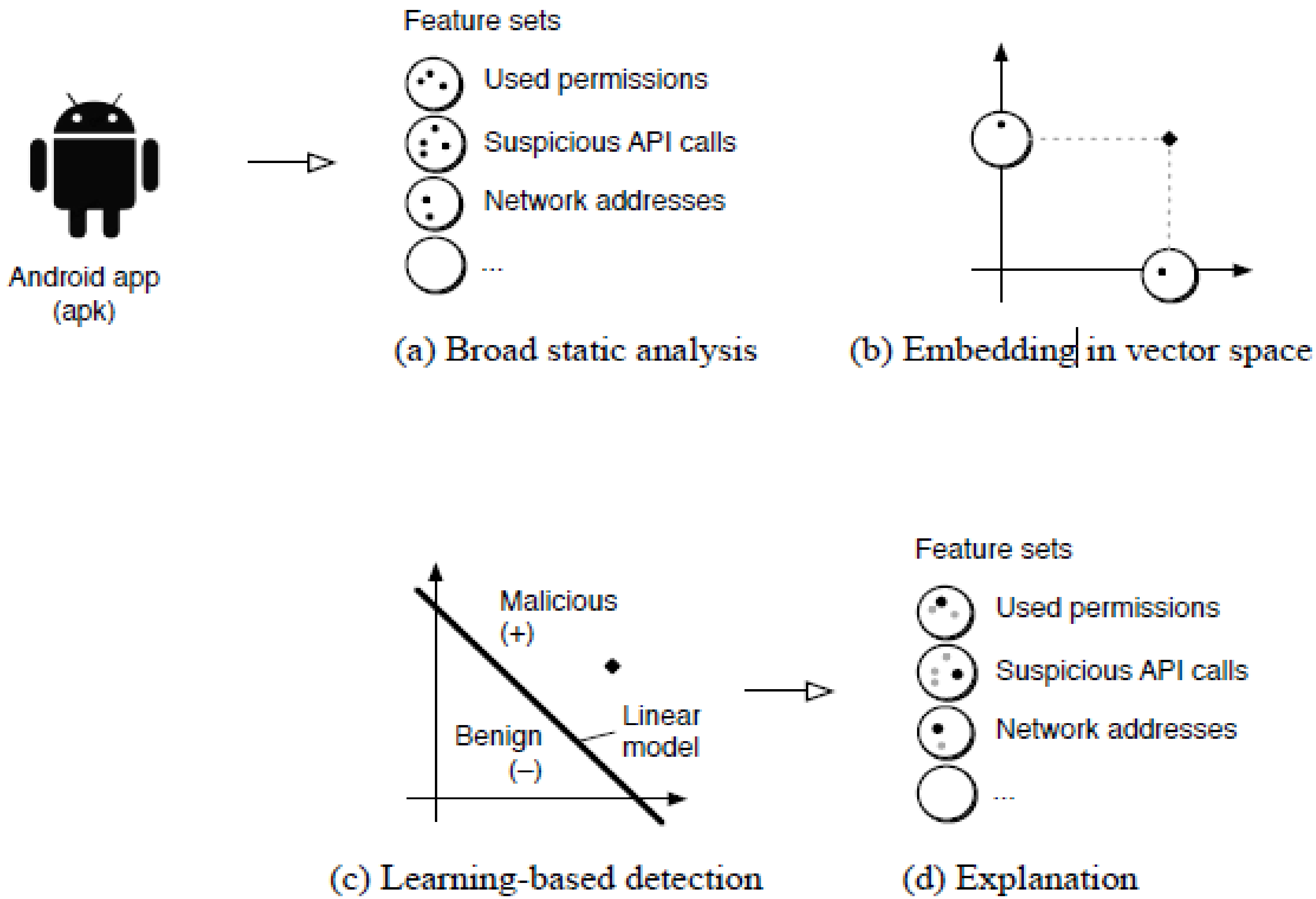


Android Malware Detection Using a Multifeature Collaborative Decision Fusion (Sheen 2014)



The Genome dataset:
1073 malicious files
903 clean apps

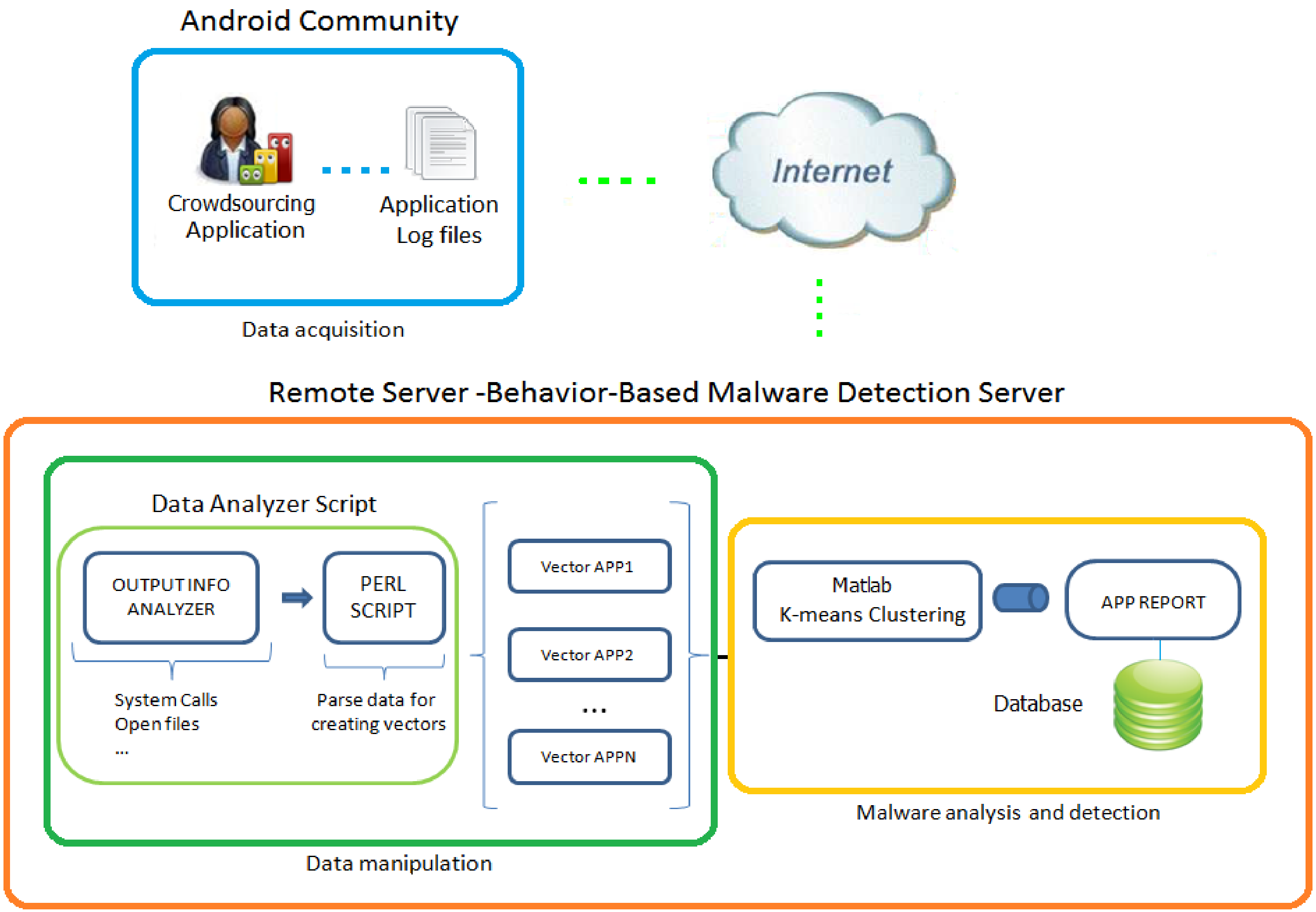
DREBIN: Effective and Explainable Detection of Android Malware in Your Pocket (Arp, 2014)



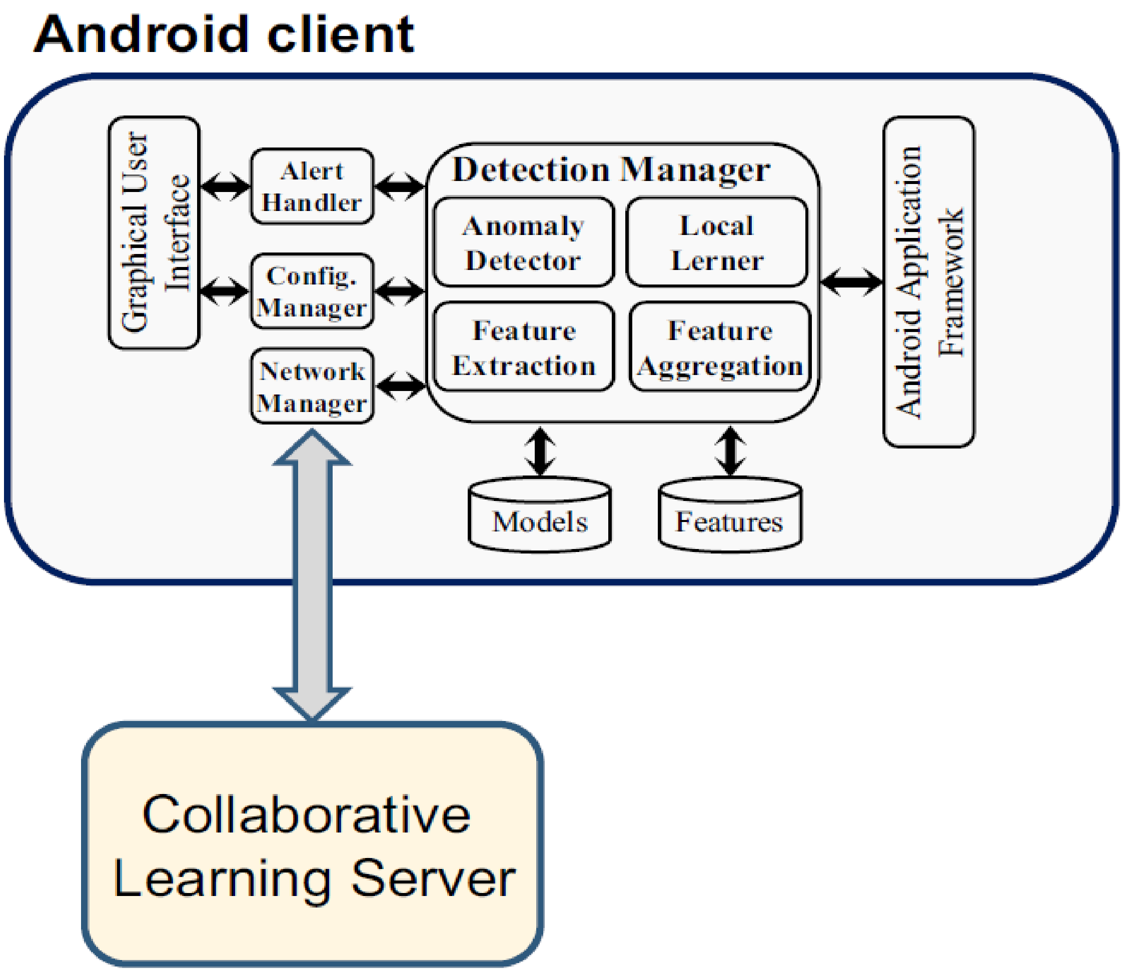
(a) Broad static analysis (b) Embedding in vector space

(c) Learning-based detection (d) Explanation

CrowDroid: Behavior-Based Malware Detection System for Android (Burguera 2011)



Analysis of Deviations in Application Network Behavior (Shabtai 2015)



Author	Approach	Detection Method	Platform	Description
Schmidt et al.(2008)[35]	HIDS, NIDS	Anomaly Detection	Android OS	Analyzes the security on Android smartphones from Linux-kernel view. Uses Network traffic, Kernel system calls, File system logs and Event detection modules to detect anomalies in the system.
Schmidt et al.(2009)[32]	HIDS	Signature-Based Detection	Android OS	Performs static analysis on the executables to extract function calls in Android OS using the command readelf. Function calls are compared with malware executables for classification.
Blï£¡sing et al.(2010)[3]	HIDS	Signature-Based Detection	AndroidOS	Uses an Android Application Sandbox to perform Static and Dynamic analysis on Android applications. Static analysis scans Android source code to detect Malware patterns. Dynamic analysis executes and monitors Android applications in a totally secure environment.
Enck et al.(2010)[15]	HIDS,NIDS	Anomaly Detection	Android OS	TaintDroid is a real time monitoring system for Android OS. TaintDroid monitors Android applications and alerts the user whenever a sensitive data of the user is compromised. Uses "taint tracking" analysis to monitor privacy sensitive information.
Portolakidis et al.(2010)[29]	HIDS,NIDS	Anomaly Detection	Android OS	A remote security server in the cloud performs the Malware detection analysis. Virtual environments will be used to analyze Android mobile phone replicas.
Shabtai et al.(2010)[37]	HIDS	Anomaly Detection	Android OS	Intrusion detection for mobile devices using the knowledge-based, temporal abstraction method (KBTA) methodology. Detects suspicious temporal patterns and to issues an alert if an intrusion is found. These patterns are compatible with a set of predefined classes of malware as defined by a security expert.
Shabtai et al.(2011)[38]	HIDS	Anomaly Detection	Android OS	Host-based malware detection system that continuously monitors smartphone features and events and applies machine learning to classify the collected data as normal (benign) or abnormal (malicious) based on a already known malware and behavior.

Burguera, Iker, Urko Zurutuza et Simin Nadjm-Tehrani. 2011. « Crowdroid: behavior- based malware detection system for Android ». In Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices. (Chicago, Illinois, USA), p. 15-26. 2046619: ACM.

Main Challenges

Effectiveness

- Data collection
- False alarms
- Adaptability

Efficiency

- Storage
- CPU
- Battery Usage

Feasibility

- Deployment
- Capacity
- Configuration
- Administration

Effectiveness: Data Collection

- Static analysis:
  - Obfuscation and dynamic libraries are (and will remain) a serious problem
- Dynamic analysis:
  - Tracing overhead
  - Trace volume
  - Trace format
  - Trace types
  - Availability of tracing tools

Effectiveness: False alarms

- High false alarms **reduce confidence** and could lead to deactivation of the ADS
- Causes:
  - Unrepresentative normal data for training and attack data for validation and testing
  - Inappropriate model or feature selection
  - Poor optimization of models parameters
  - Over fitting (leads to poor generalization)
  - Inadequate assumptions such as static environments

Kernel State Modeling (KSM)

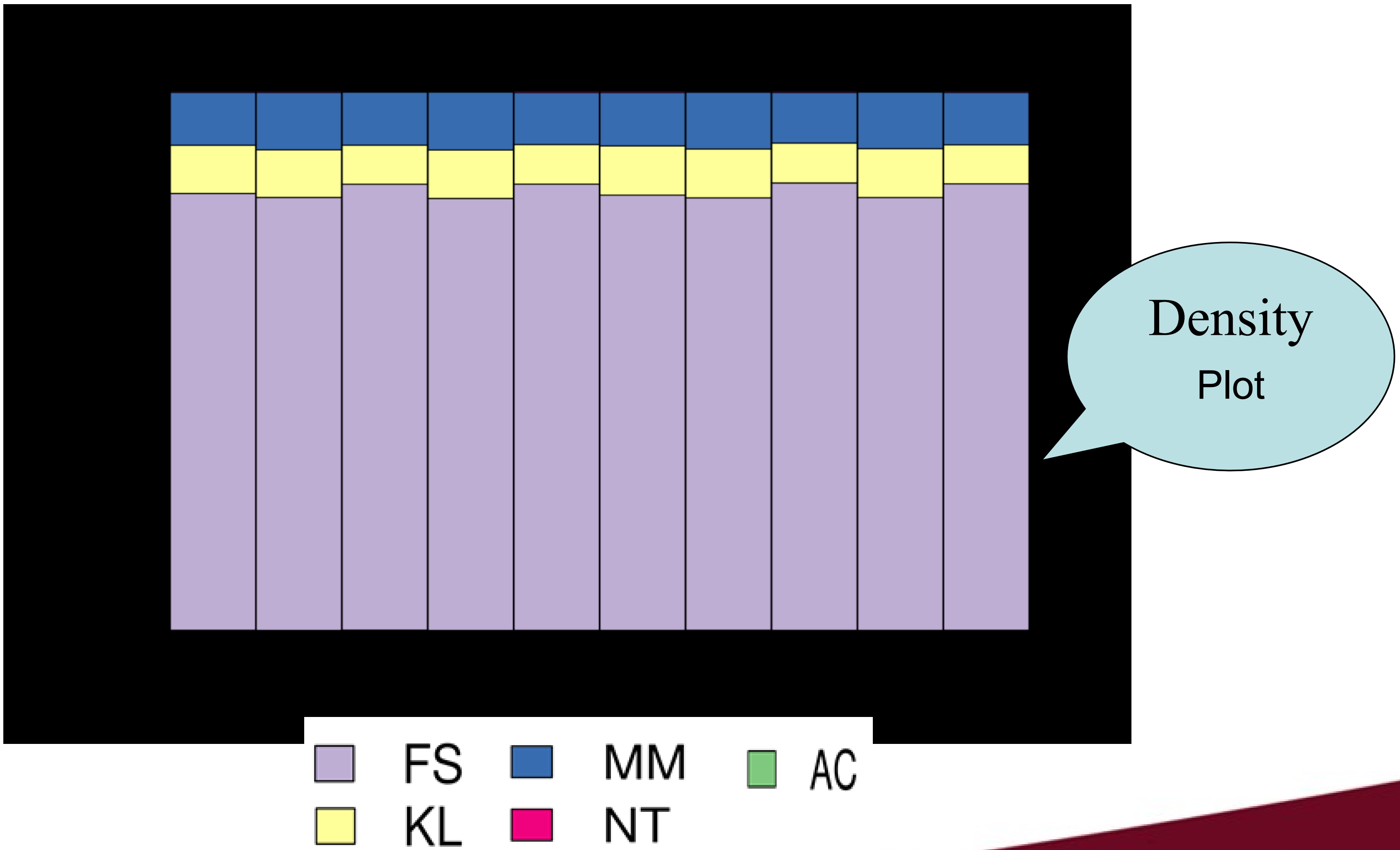
- KSM is an anomaly detection technique
  - Transforms system calls into kernel modules, called states
  - Detect anomalies at the level of interaction of kernel states
  - Reduces data space used in training and testing
  - Favors efficiency while keeping accuracy

Transforming System Calls into States of Kernel Modules

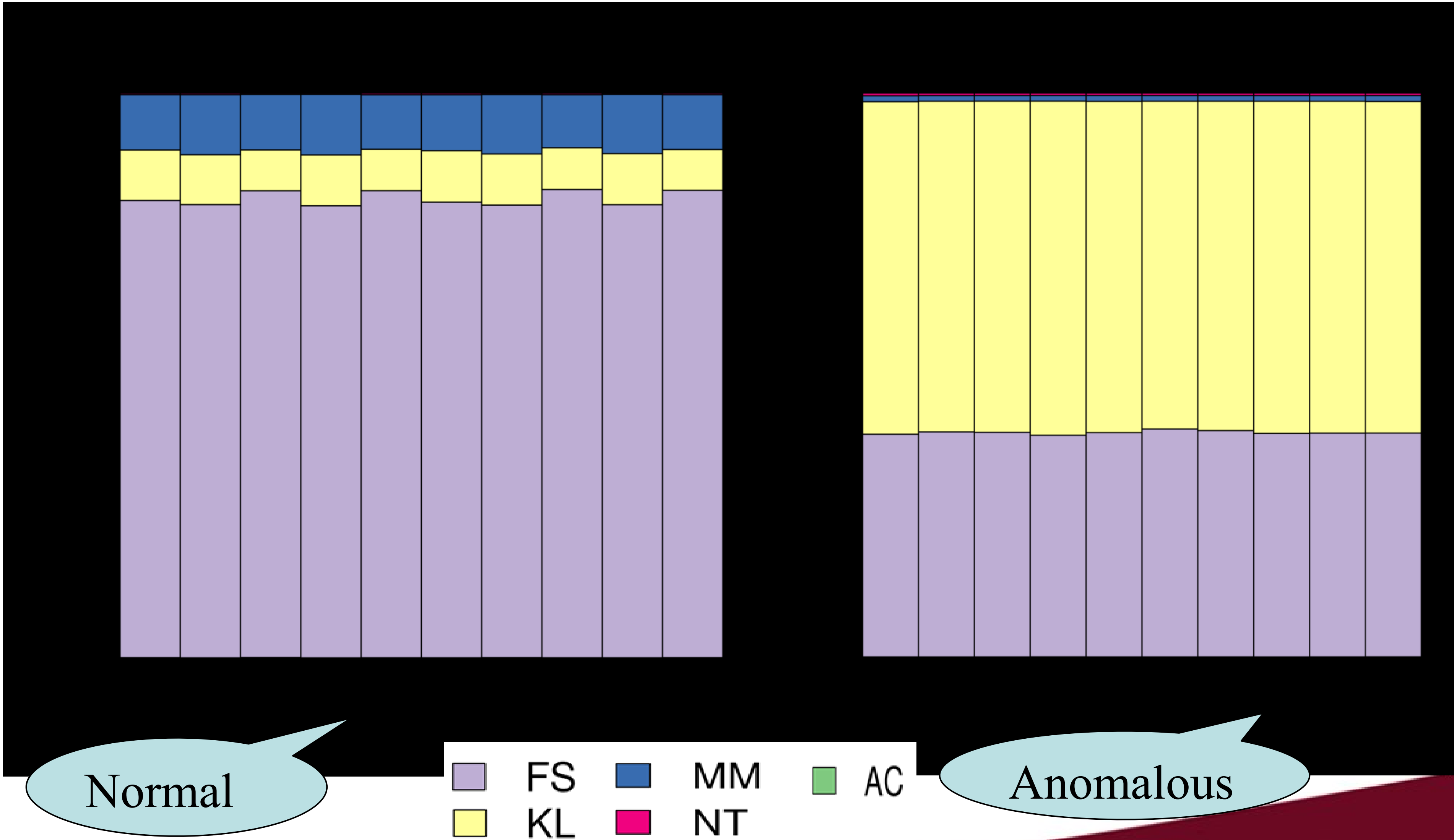
State	Module in Linux Source Code	# of System Calls
AC	Architecture	10
FS	File System	131
IPC	Inter Process Communication	7
KL	Kernel	127
MM	Memory Management	21
NT	Networking	2
SC	Security	3
UN	Unknown	37

[Source]: http://syscalls.kernelgork.com

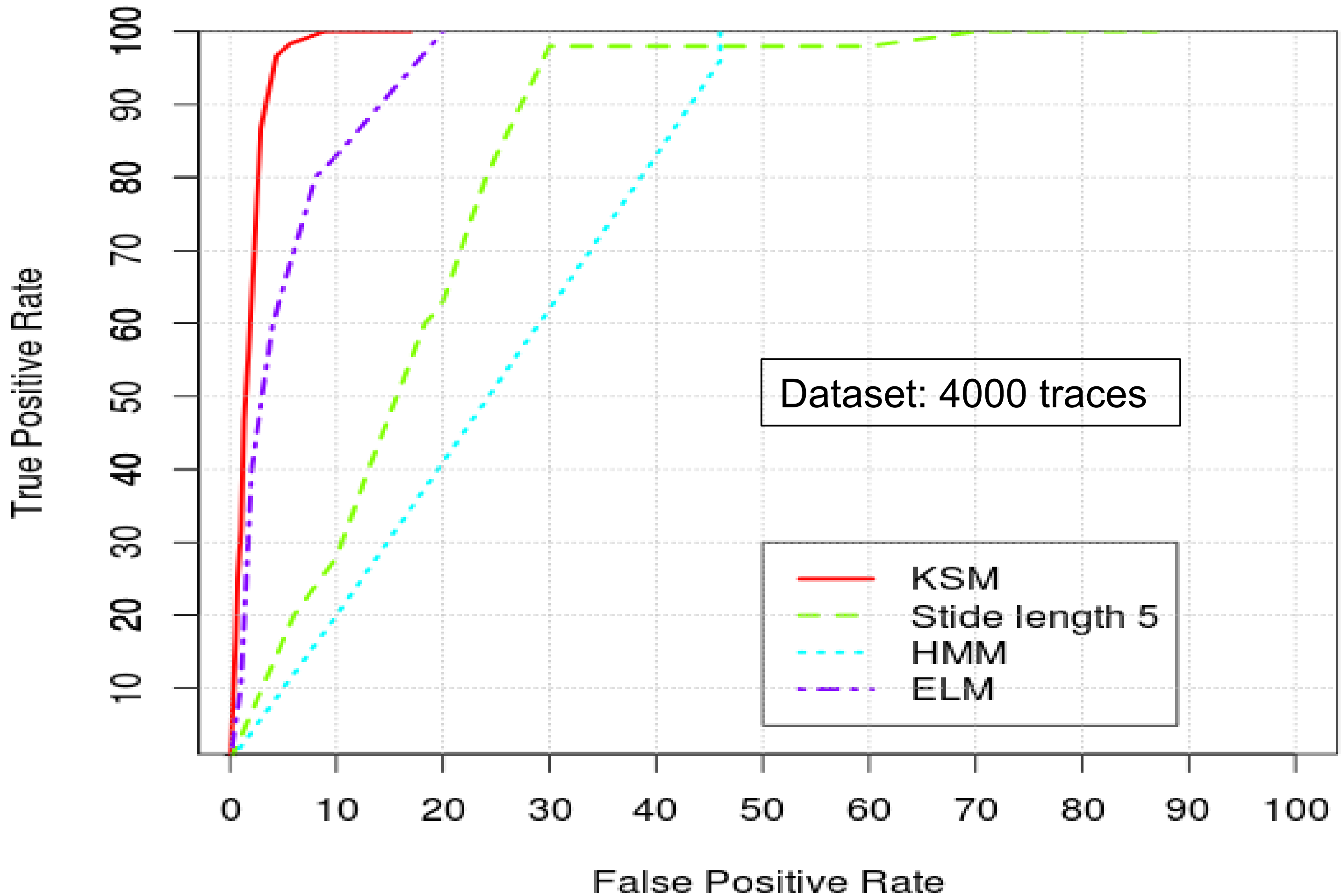
KSM and Density Plots



Anomaly Detection



Results of KSM on ADFA Linux Dataset

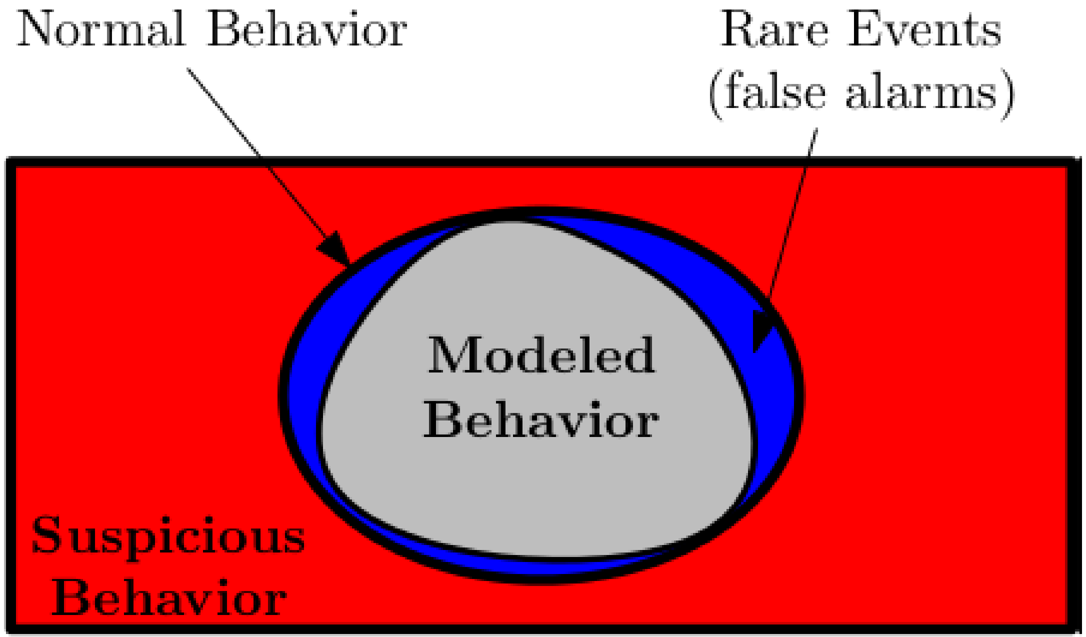


KSM Execution Time

	Size of All Traces	KSM	Stide	HMM
Login	26.2KB	4.46 sec	0.03 sec	56.43 min
PS	29.6KB	5.14 sec	0.11 sec	46.24 min
Xlock	47.4MB	1.51 min	12.3 min	13.37 hr
Stide	36.2MB	5.85 min	8.53 min	2.3 day
Firefox	270.6MB	9.35 min	4.17 hr	4.03 day

Effectiveness: Adaptability

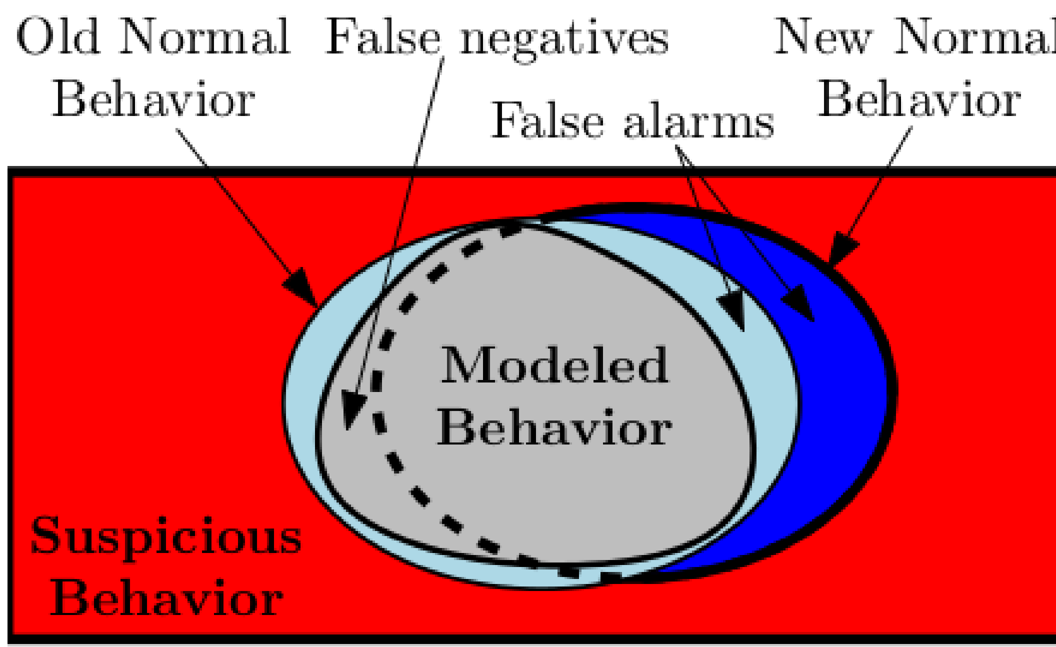
- ADSs are often designed using limited data
  - collection and analysis of representative data from each process (different versions, OS, etc.) is costly



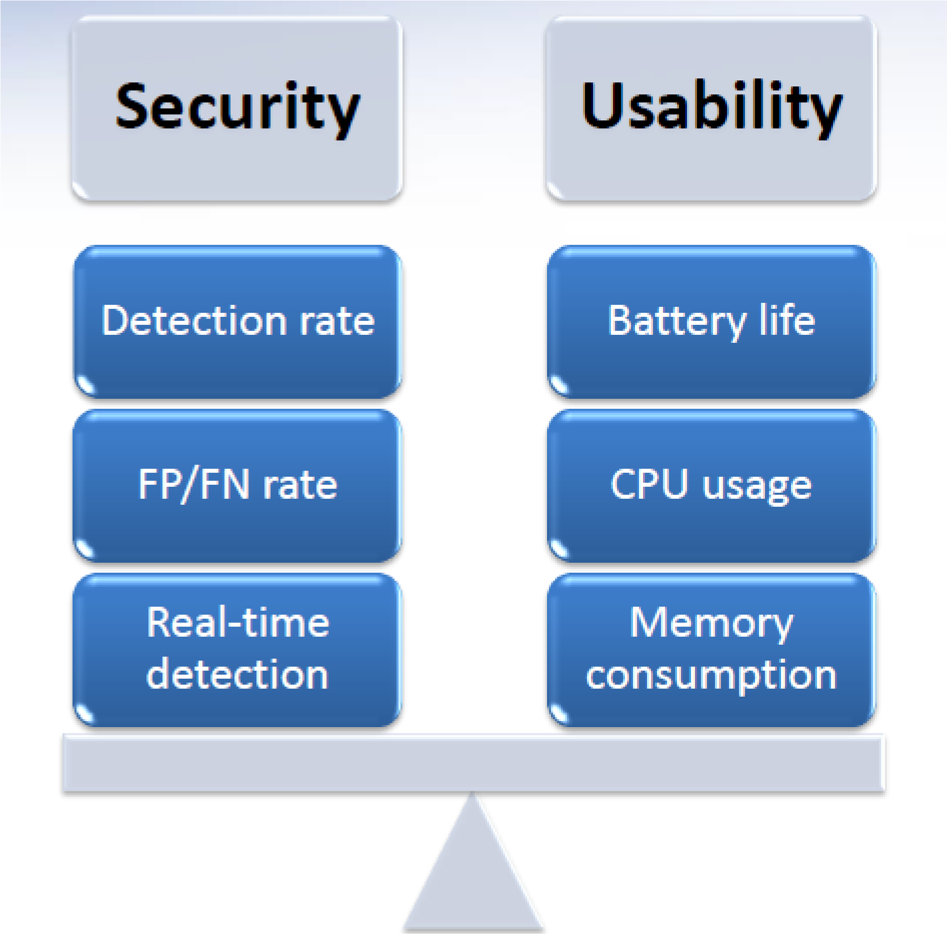
Effectiveness: Adaptability

- Dynamic environment
  - Changes in normal process behaviour due, for instance, to application update

Internal model of normal behavior **diverges** with respect to the underlying data



Efficiency of Anomaly Detectors on Mobile Devices



Experimenting with Known ADS Models

1 2 3

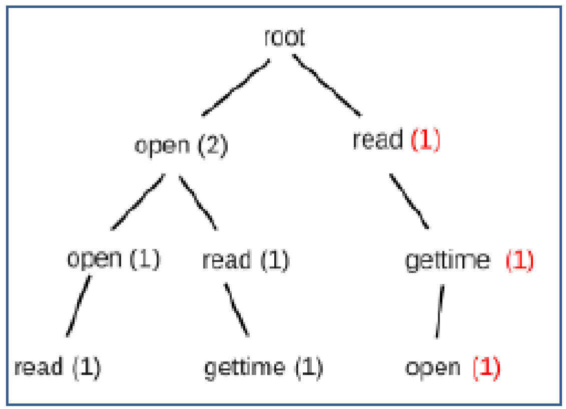
Open, open, read, gettime, open, read, close

Lookahead pairs

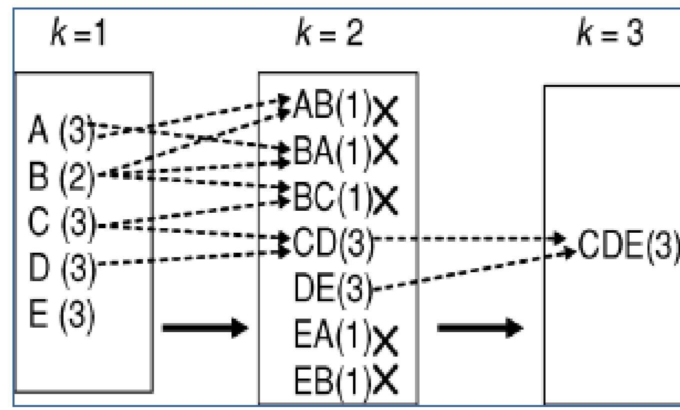
W=3	syscall	1 after	2 after
w1	open	open	read
w2	open	read	gettime
w3	read	gettime	open
w4	gettime	open	read
w5	open	read	close

W=3	syscall	1 after	2 after
Call 1	open	Open, read	Read, gettime, close
Call 2	read	gettime, close	open
Call 3	gettime	open	read

N-gram Tree

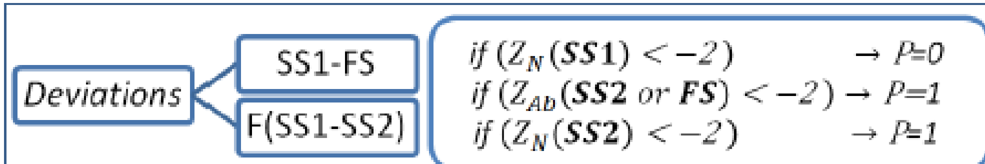


Varied-length N-grams



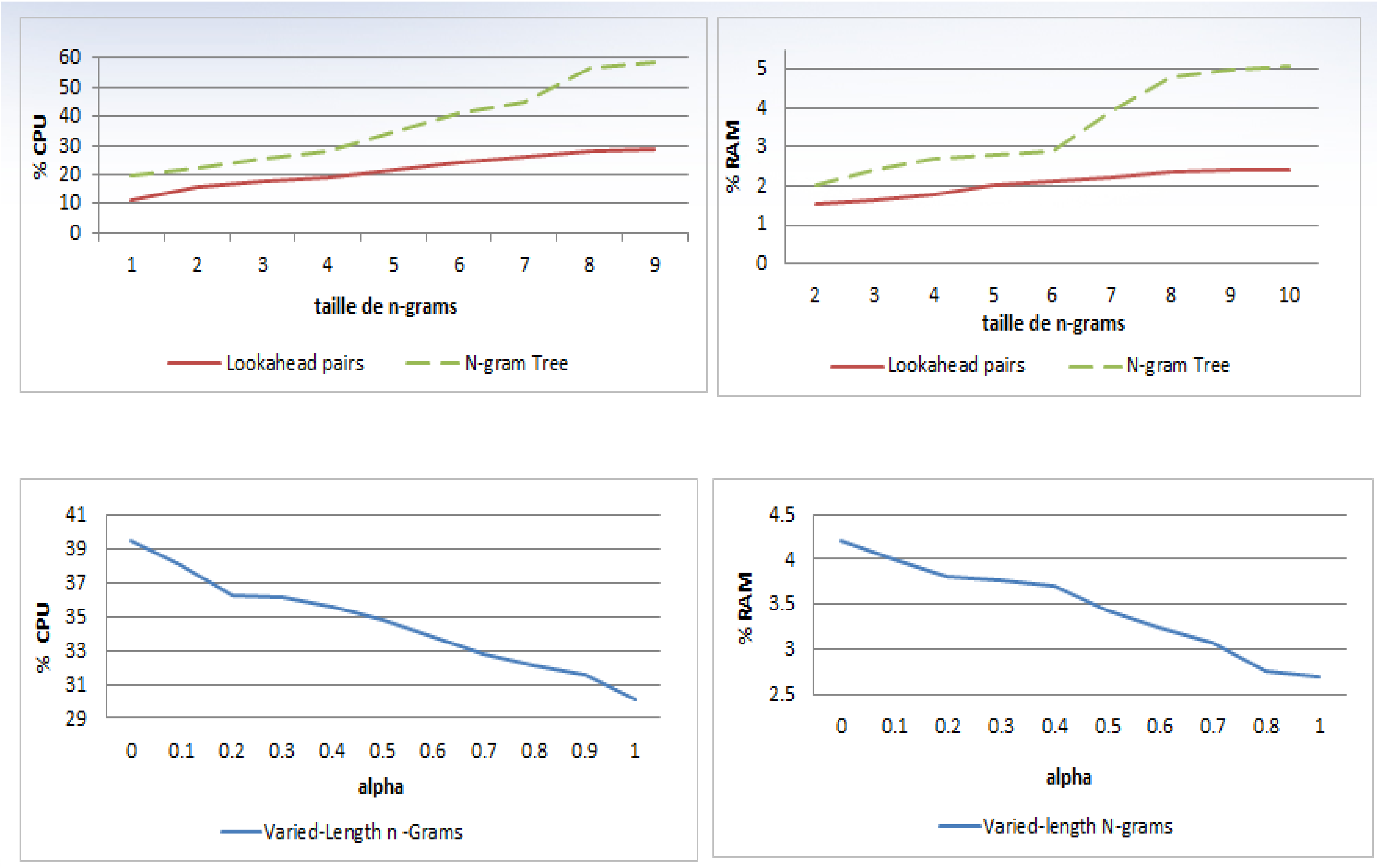
$$f(p_{k+1}) > \alpha \min(f(r_k), f(q_k))$$

Finite State Machines



$$P(Anomaly) = \frac{(1 + SD_N(SS1)) * (\alpha * Z_N(SS1) + 3)}{(Br(SS1)) * (Z_{Ab}(SS2 \text{ or } FS) + 3) * (Z_N(SS2) + 3)}$$

CPU and Memory Usage



Feasibility

- On-device vs. remote detection
- Dynamic configuration (tracing, algorithm selection, thresholds)
- Combining multiple data sources
- Combining multiple heterogeneous detectors
- Human in the loop
- Governance of app markets

Thank You

Wahab Hamou-Lhadj, PhD, ing.

Software Behaviour Analysis (SBA) Research Lab

Concordia University

Montreal, QC, Canada

www.ece.concordia.ca/~abdelw/sba