ANANAS - **An**alizing **An**droid **A**pplication**s**

Dieter Vymazal
dieter.vymazal@fh-hagenberg.at
contact@malware-lab.at

11.09.2015

HAGENBERG | LINZ | STEYR | WELS

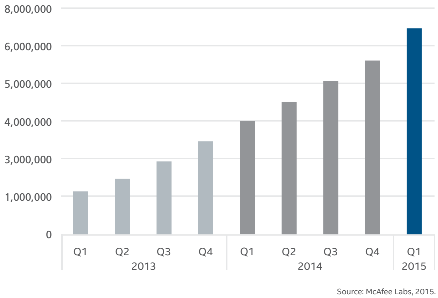
OBERÖSTERREICH
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

"ANANAS is an extensible Android malware analysis platform that performs both static and dynamic malware analysis. It consists of the ANANAS Core framework as well as several plugins."

Introduction & Motivation (1/2)

- Android is an operating system based on the Linux kernel and was launched in 2008.
- Market share of 78% (Q1 2015).[1]
- Information stored on mobile devices is getting more and more valuable:
  - Contacts (email, social networks, ...)
  - Passwords
  - Location data
  - VPN access
  - ...
- Increasing computing power and high speed internet on mobile devices.

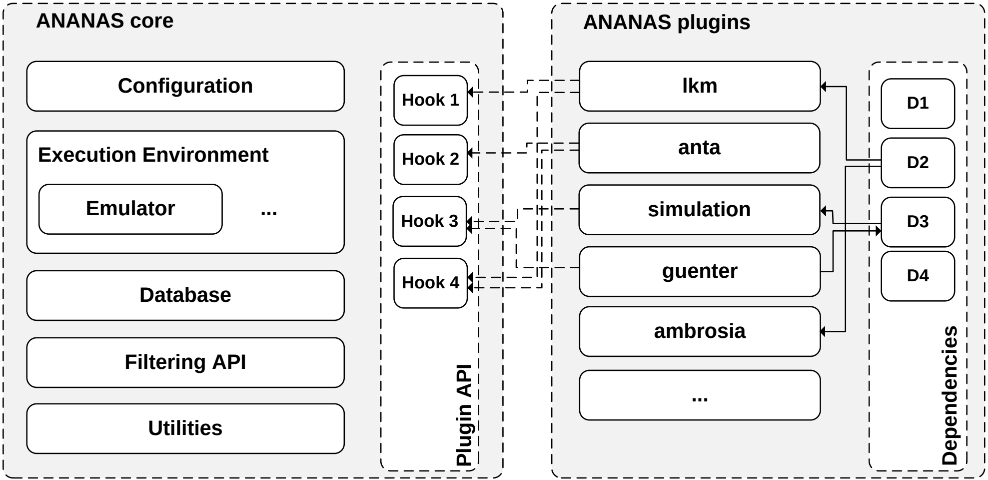
[1] http://www.idc.com/prodserv/smartphone-os-market-share.jsp



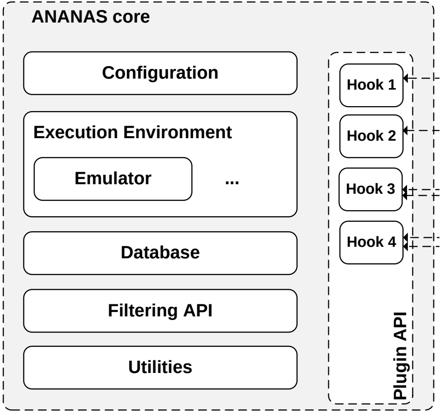
- Kaspersky Lab experts estimate that 98.05% of all existing mobile malware targets the users of Android devices.[2]
- Malware protection on mobile devices is rarely used.
- $\rightarrow$ Number of android malware is rising.

[2] http://media.kaspersky.com/pdf/Kaspersky-Lab-KSN-Report-mobile-cyberthreats-web.pdf

Architecture of ANANAS

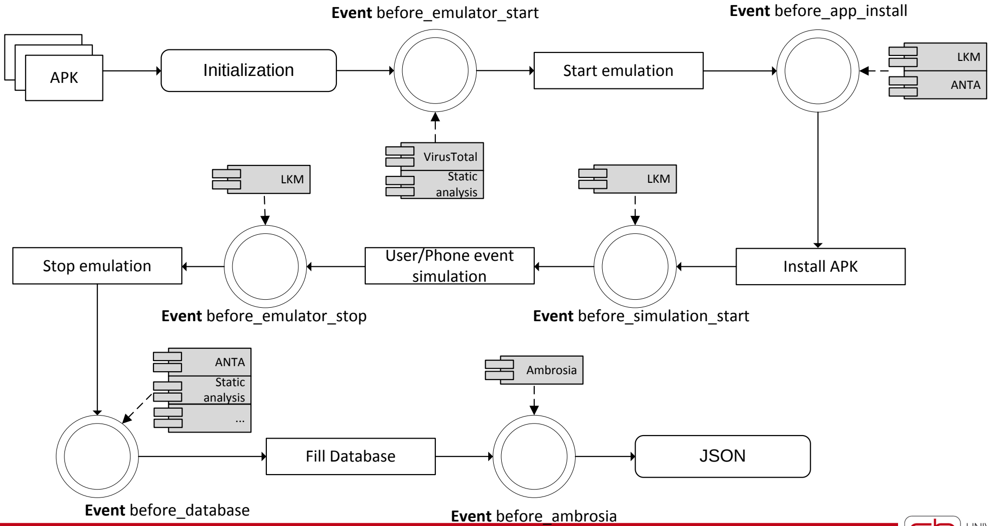


ANANAS Core (1/2)



- The ANANAS Core among others consists of an execution environment (e. g. the android emulator) and provides APIs to the plugins.
- The ANANAS Core sets up a database connection (where all the gathered information is stored).
- A configuration file allows customizing ANANAS Core.
- ANANAS Core calls so-called hooks (which represents a certain point in the analysis process).

ANANAS Core (2/2)



UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

ANANAS Plugins (1/7)

- ANANAS comes with a set of plugins that provides additional functionality not provided by the ANANAS Core:
- VirusTotal
  - The VirusTotal plugin is used to query VirusTotal.com. This site can be used to performs scans on a single sample using multiple malware scanners.
  - If the sample is unknown to VirusTotal, the VirusTotal plugin can uploads the sample.
  - The results returned by VirusTotal.com are written to the database.
  - This allows an analyst to quickly identify samples that are already know as being malware.

ANANAS Plugins (2/7)

- Static
  - The static plugin performs simple static analysis steps.
  - It disassembles the DEX bytecode of the app to the SMALI syntax
  - Afterwards, all static strings constants are being extracted from the disassembled code. Strings that represents a URL are reported separately.
  - This allows an analyst to e.g. find URLs referring to command and control servers.
  - Moreover, it calculates hashes from the APK file itself as well as from all the files in the extracted APK. This allows an analyst to find relationships between samples, e.g. if an image is contained in multiple samples.
  - The Android APK Manifest is being parsed and relevant information (e.g. required permissions) are extracted.

ANANAS Plugins (3/7)

- Simulation
  - The Simulation plugin is used to simulate normal device behavior. This is important to trigger malicious behavior in malware.
    - For example, to analyze malware that forwards received SMS to a server, SMS have to be received during analysis to trigger this behavior.
  - The sequence of the events the simulation plugin should trigger is described by simulation scripts. These scripts (see next slide) follow a very simple syntax where each line describes an event.

```
unlockscreen
sleep 3
startservices
sleep 5
startapp
screenshot
monkey 200
smsfrom '+430000000000' 'Hi there.'
screenshot
```

ANANAS Plugins (4/7)

- Guenter
  - The Guenter plugin implements a user simulation. Guenter does so by getting all the user interface components the device is showing (e.g. the view hierarchy).
  - It then iterates over all clickable items and tries to evaluate whether an item is interesting. For the purpose of malware analysis, the most interesting user interface components have been found to be clickable items that represent a positive answer to a choice /e.g. a button labelled Yes or OK).
  - Guenter then clicks on the element that was found to be most interesting.

ANANAS Plugins (5/7)

- LKM
  - The LKM plugin uses a loadable kernel module that is being inserted into the kernel of the Android system at the beginning of the analysis.
  - This module captures information from the kernel such as the running processes and threads as well as specific system calls.
  - Moreover, it is intended to also alter the normal behavior of the kernel. It hooks the `sys_unlink()` system all to save files that would otherwise be deleted. This allows an analyst to inspect the contents of temporary files created by a sample.

ANANAS Plugins (6/7)

- ANTA
  - The Network Analysis plugin (ANTA) generates an overview of the network communication that occurred during the analysis.
  - Before the sample app is installed, ANTA starts a network capture. After the emulator has been stopped, ANTA analyses the generated PCAP file and extracts information about DNS queries, SMTP, HTTP and HTTPS connections as well as ARP and ICMP communication.
  - The data generated by the ANTA plugin allows an analyst to e.g. identify communication to a command and control server.

ANANAS Plugins (7/7)

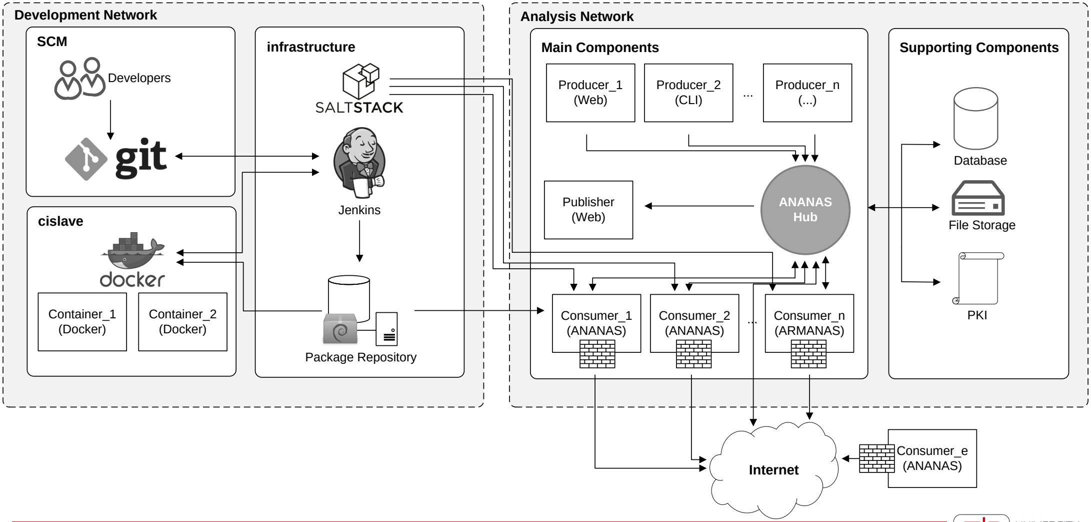
- Radiolog
  - The radio log is a log stream written by the Android system that contains information about commands sent to and received from the radio module.
  - The Radiolog plugin extracts information about sent SMS messages, SIM pin changes, outgoing and incoming phone calls, call forwarding and information about APN used.
  - Moreover, the Radiolog plugin is able to identify whether the remote phone number of incoming and outgoing phone calls as well as sent SMS is charged with additional fees.

Piña

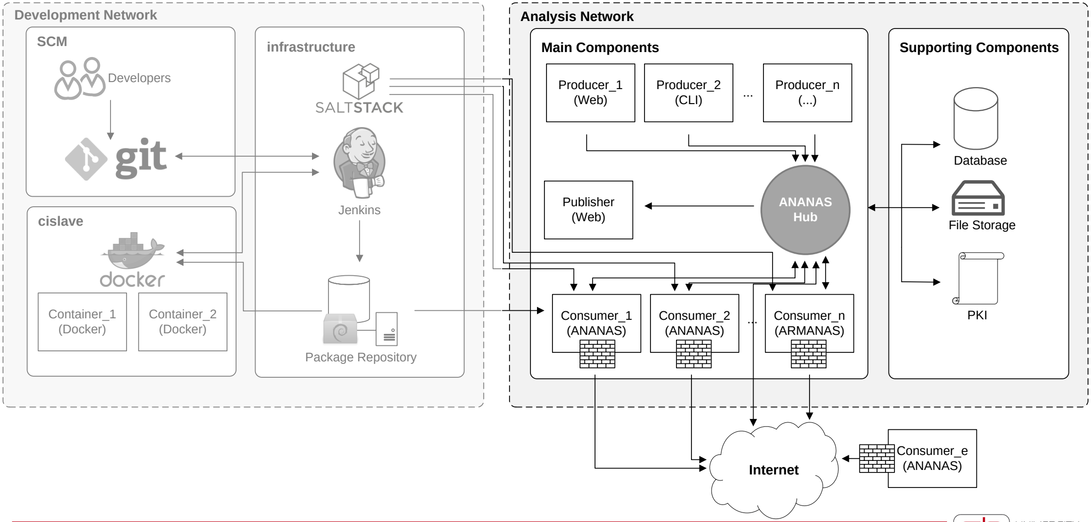
Introduction & Motivation

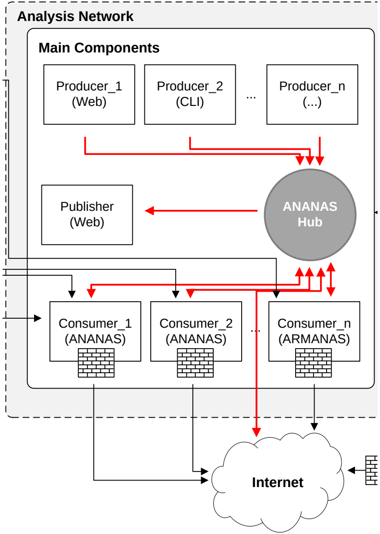
- Piña is an environment for automated analysis of Android applications and meets the following requirements.
  - Usage of ANANAS
  - ANANAS as a service
  - Access to analysis results and APK files
  - Queuing and prioritization
  - Recognition of applications which have been analyzed already
  - Scalability and parallelism
  - Error handling
  - Isolation of the analysis environment (the sample is not able to communicate with the underlying system)
  - Authentication and authorization of each component
  - Secure communication between the components

Piña Architecture

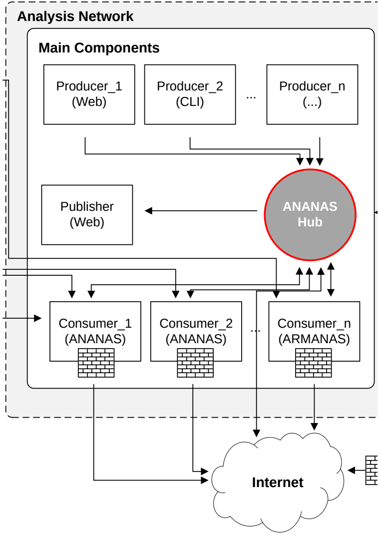


Analysis Network (1/7)

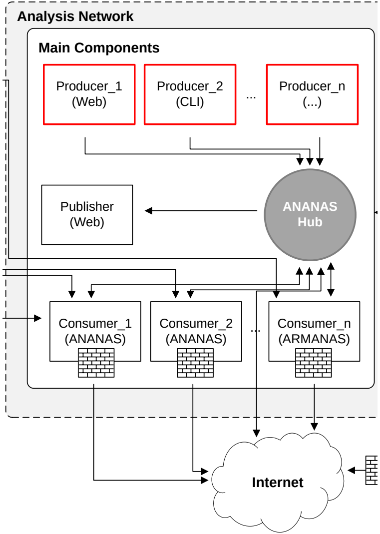




- NRMQ
  - All components of the analysis environment communicate via the NRMQ protocol which was specially designed for Piña.
  - NRMQ is a client-server protocol. The clients send a request to the server (hub) and the server responds with the requested data and a status code (success/failure).
  - The communication is secured using TLS.



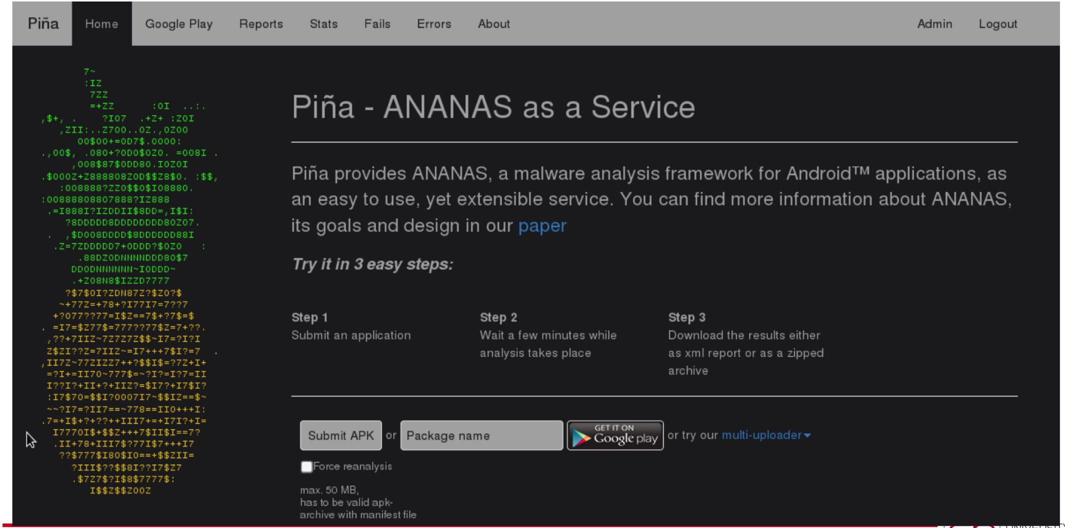
- ANANAS-Hub
  - The ANANAS hub coordinates the analysis jobs as well as the available analysis resources, called consumers.
  - When receiving an analysis job, the hub forwards it to an available consumer.
  - The Hub acts as a server (within the NRMQ protocol) to which the clients, divided into producers, consumers, and publishers, connect to.



Producer

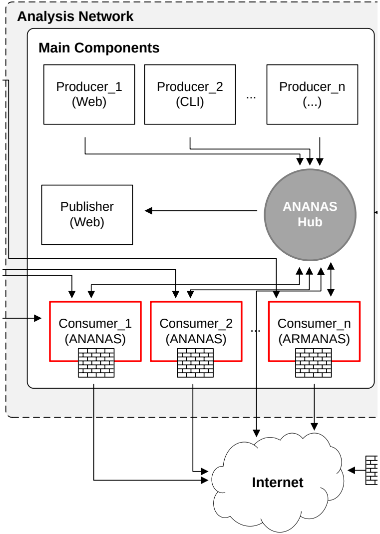
- A producer is responsible for creating analysis jobs and sending them to the ANANAS hub. Currently two different kinds of producer components are implemented.
  - **Web** The web interface allows to upload APK files, start an analysis and fetch the generated results.
  - **CLI** The CLI is intended to be an administration tool which comes with producer functions and also offers the possibility to query different information.

Analysis Network (5/7)



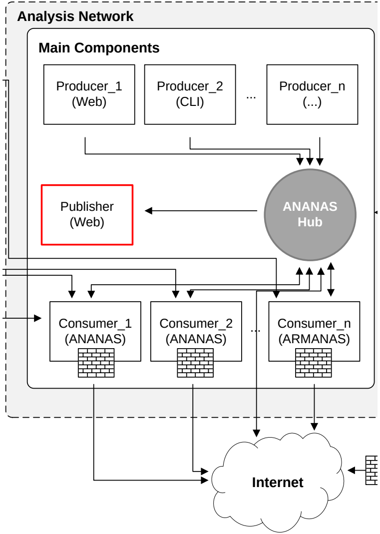
UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA

Analysis Network (6/7)



- Consumer
  - The consumers are responsible for analyzing the samples contained within the analysis jobs by using ANANAS.
  - A consumer establishes a connection to the hub and requests an analysis job. Subsequently ANANAS is being started and the requested file analyzed.
  - After an analysis has finished, a consumer sends the results back to the hub where they are stored.

Analysis Network (7/7)

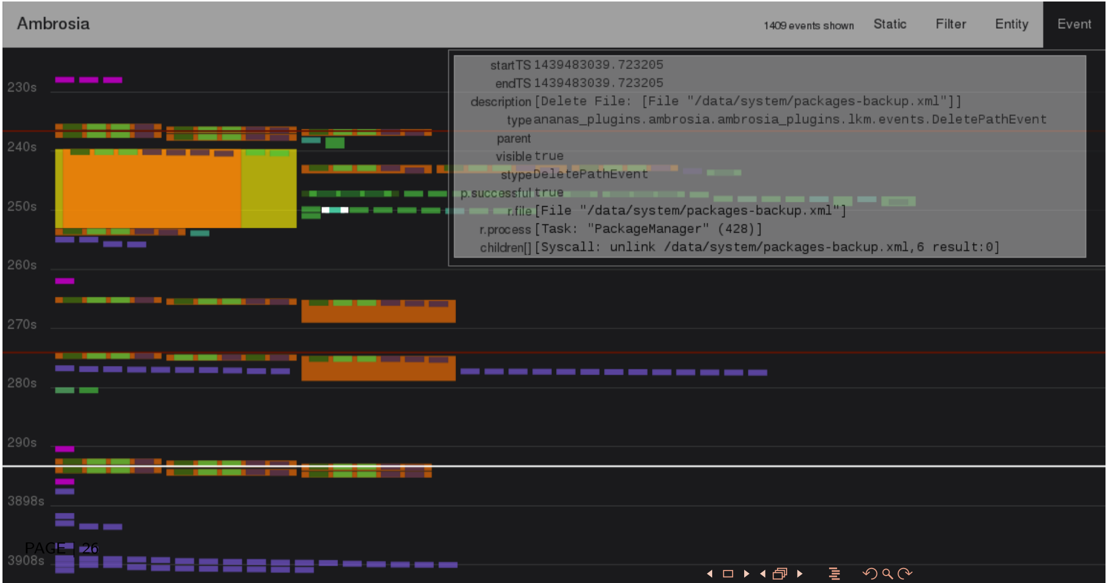


- Publisher
  - The publisher's task is to provide the users with the analysis results.

Ambrosia (1/2)

- Ambrosia is a framework that visually present the results generated by the ANANAS analysis framework.
- The first stage fetches the results stored in the database and generates a JSON file (preprocessing).
- In the second stage the JSON file is loaded by the web component (written in javascript – means executed in the browser) of ambrosia and presents the results.
- The rendering component allows an analyst to navigate through the events and the entities.
- A comprehensive filter language allows to query events by its type, content and other characteristics.

Ambrosia (2/2)



Ambrosia
1409 events shown
Static
Filter
Entity
Event
startTS 1439483039.723205
endTS 1439483039.723205
description [Delete File: [File "/data/system/packages-backup.xml"]]
type ananas_plugins.ambrosia.ambrosia_plugins.lkm.events.DeletePathEvent
parent
visible true
stype DeletePathEvent
p.successful true
r.file [File "/data/system/packages-backup.xml"]
r.process [Task: "PackageManager" (428)]
children[] [Syscall: unlink /data/system/packages-backup.xml,6 result:0]
230s
240s
250s
260s
270s
280s
290s
3898s
3908s

Current Work

ARMANAS (1/2)

- ARMANAS allows running a dynamic analysis workflow on a physical hardware with ARM CPUs (similar to regular off-the-shelf Android devices).
- Analysis workflow and plugins are modified and adapted to fit the needs of the physical environment.
- Improved resilience against evasion techniques.
- Faster boot and execution speed throughout the dynamic analysis.

ARMANAS (2/2)

- ARM architecture is the most widely used architecture in mobile devices.[3]
- In order to produce reliable results, the environments footprint has to be minimized.
  - This can be achieved much more easily using actual hardware as virtualized hardware is more susceptible to detection.

[3] http://dl.acm.org/citation.cfm?doid=1941487.1941501

Current Work (1/5)

- DAMIAN – Detecting Android Malware in ANANAS
  - DAMIAN correlates various ANANAS results with the purpose of gaining new information.
  - For this purpose, ANANAS results will be consolidated with previously chosen analyzing methods (e.g. classification in malware families).
  - The analyzing methods will be picked out according to a literature research on already existing publications on this topic.
  - Finally a framework will be deployed to adapt various methods to the ANANAS results.
  - The final framework should be able to preprocess the ANANAS results, apply the chosen analyzing methods to the results and post-process them. Furthermore the framework is easily expandable, means new analyzing methods can be added without any trouble.

Current Work (2/5)

- Guenter-NG – A framework for user simulation for ANANAS
  - Evaluating current Guenter version.
  - What are characteristics of a real user?
  - Recognition of input fields and providing suitable input data.
  - Creating templates of known user interactions.

Current Work (3/5)

- LEA – Lightweight Emulation of Android
  - ANANAS currently uses the official Android Emulator which doesn't run very smoothly.
  - On one hand it runs very slow in virtualized environments, on the other hand there are several stability issues while emulating Android applications.
  - The first step is to find out what requirements on an emulator ANANAS has.
  - After that the qualified Android emulation products will be evaluated.
  - Finally the best fitting solution for ANANAS will be implemented.

Current Work (4/5)

- Hardware Emulation in Android
  - Provide a more realistic (virtual) hardware environment
  - Realistic emulation of sensors and auxiliary components (camera, gps, . . .)
  - Be as close as possible to a real mobile phone in terms of available hardware
  - Make it harder for apps to detect our analysis environment through hardware investigation

Current Work (5/5)

- ANNE – ANANAS Networking Environment
  - ANANAS has only limited access to the internet.
  - Some Android applications might not show their full functionality.
  - A more realistic networking environment has to be set up.
  - This can be done via (e. g.) service simulation and/or proxies.
  - Also, legal issues concerning the ANANAS analysis environment are to consider.