

Automated Detection and Classification for Packed Android Applications

Yibin Liao, Jiakuan Li, Bo Li, Guodong Zhu, Yue Yin,
Ruoyan Cai

Network System and Security (NSS) Lab
University of Georgia (UGA)

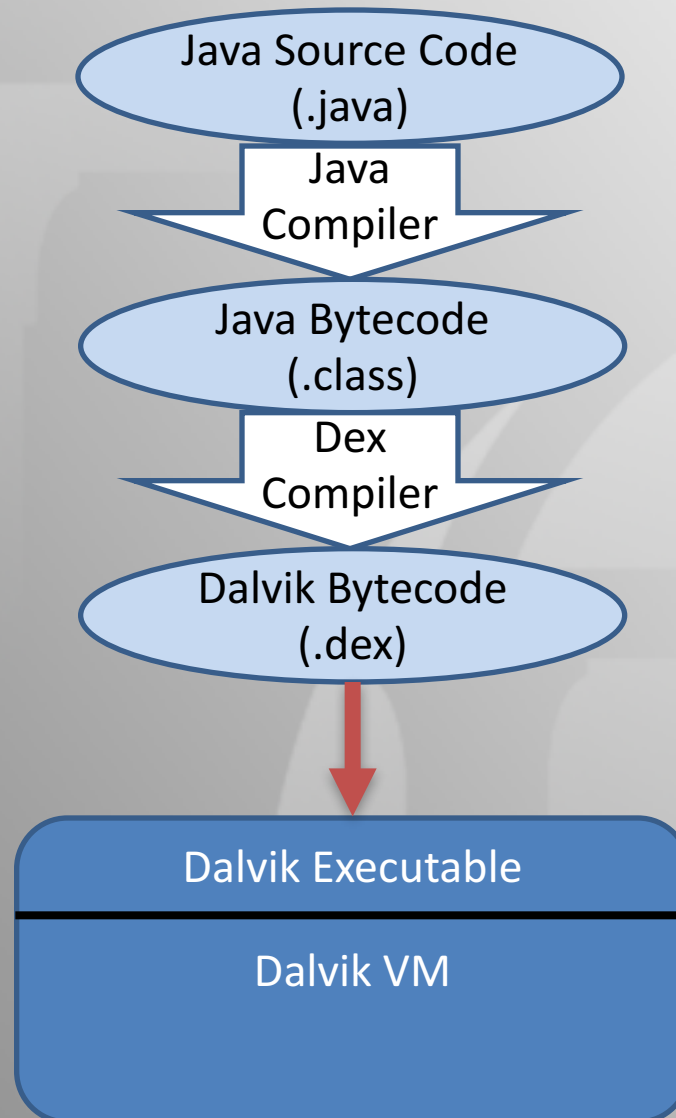


Goal

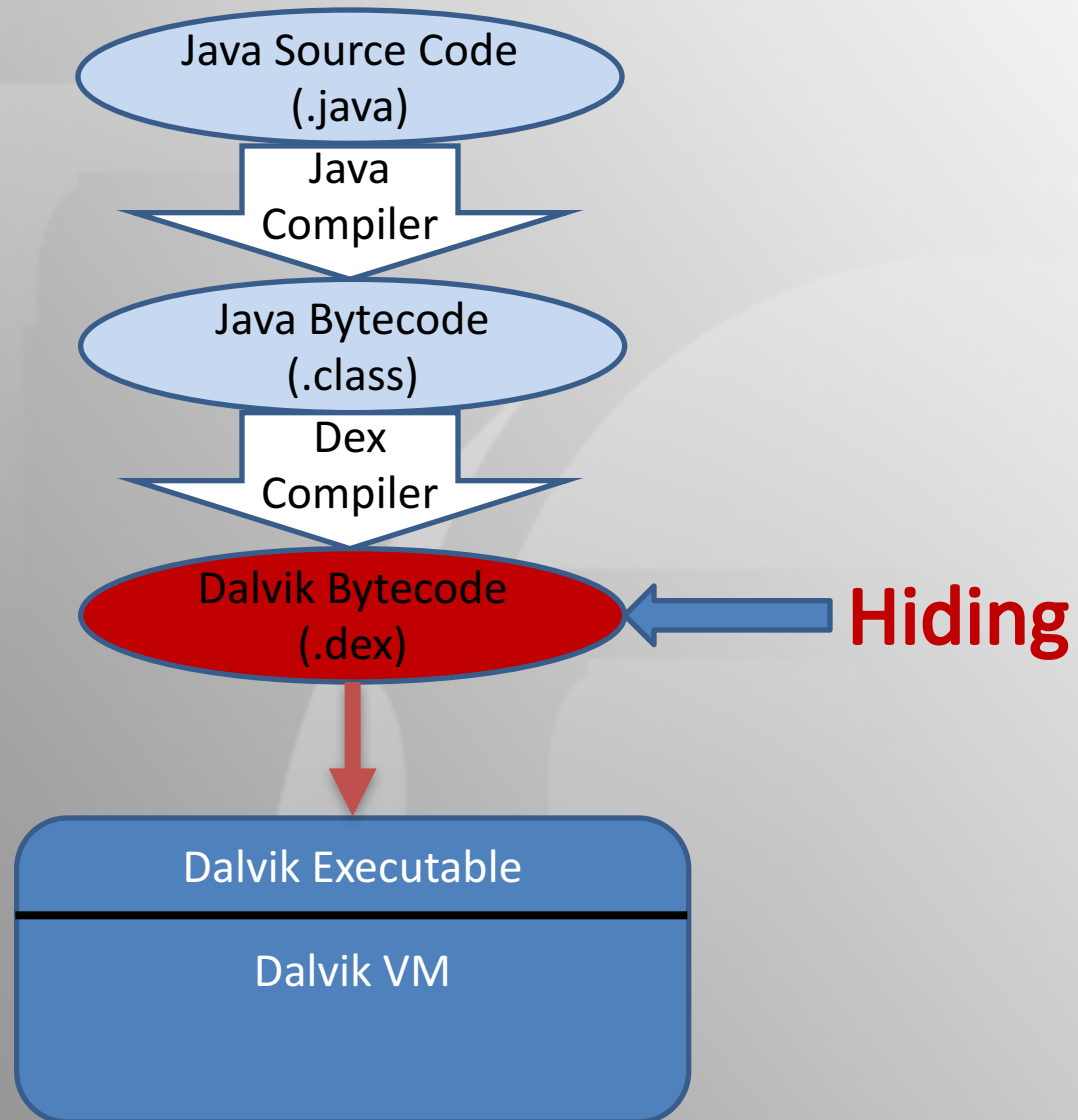
- Detection
 - Automatically identify **packed Android apps**
- Classification
 - Automatically classify different packers into different groups



How Android App is Built and Run



Packed Android App



Code Tree of Decompiled Dex

- ▶ android.support
- ▶ butterknife
- ▼ com
 - ▶ bmpak.anagramsolver
 - ▶ squareup.javawriter
- ▼ io.realm
 - ▶ annotations
 - ▶ exceptions
 - ▶ internal
 - ▶ processor
 - ▶ BuildConfig.class
 - ▶ EnglishWordRealmProxy.class
 - ▶ FranceWordRealmProxy.class
 - ▶ GermanWordRealmProxy.class
 - ▶ GreekWordRealmProxy.class
 - ▶ Realm.class
 - ▶ RealmBaseAdapter.class
 - ▶ RealmChangeListener.class
 - ▶ RealmList.class
 - ▶ RealmMigration.class
 - ▶ RealmObject.class
 - ▶ RealmQuery.class
 - ▶ RealmResults.class
 - ▶ ValidationList.class

Original

- ▼ com.shell
 - ▶ NativeApplication.class
 - ▶ SuperApplication.class

ijiami

- ▼ com.secneo.guard
 - ▶ ACall.class
 - ▶ ApplicationWrapper.class
 - ▶ FirstApplication.class
 - ▶ MyClassLoader.class
 - ▶ Util.class
 - ▼ neo.proxy
 - ▶ DistributeReceiver.class

Bangcle

- ▼ com.ali.mobisecenhance
 - ▶ StubApplication.class

Ali

Current problems

- Packed Android Malware
- Manual effort for analysis
 - Tedious
- Packers are evolving.
 - Unpacking approaches only works for a limited time, or particular type of packers.

Current packing techniques

- Code obfuscation
- Anti-debugging
- Bytecode hiding
- Dynamic code modification
- Dynamic loading

Our approach

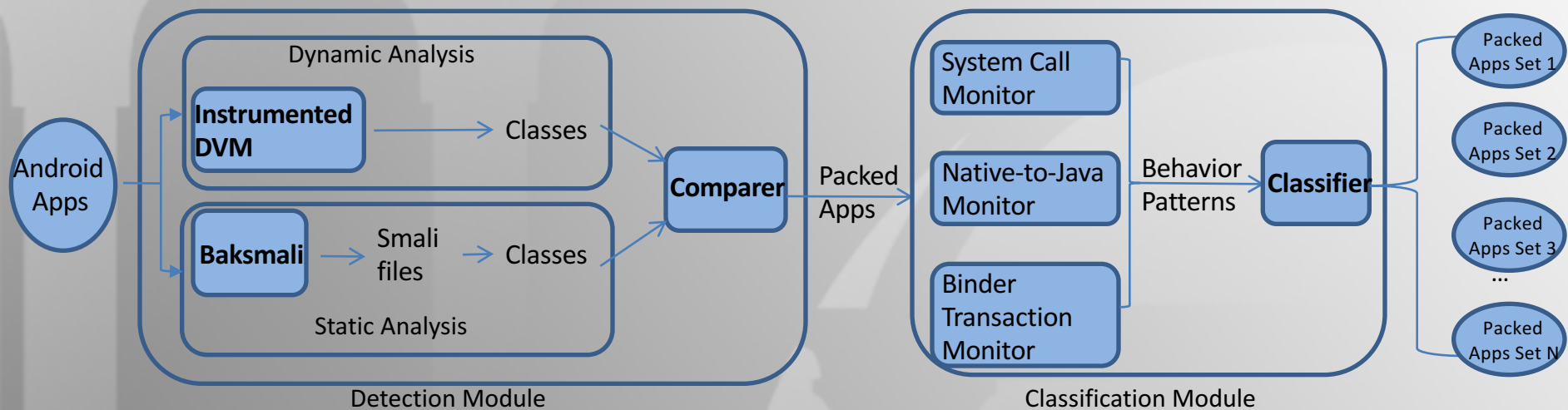
Detection

- Combined Static and Dynamic analysis
 - Static
 - Static analysis tools (baksmali)
 - Dynamic
 - DVM instrumentation
 - Compare classes from static and dynamic analysis

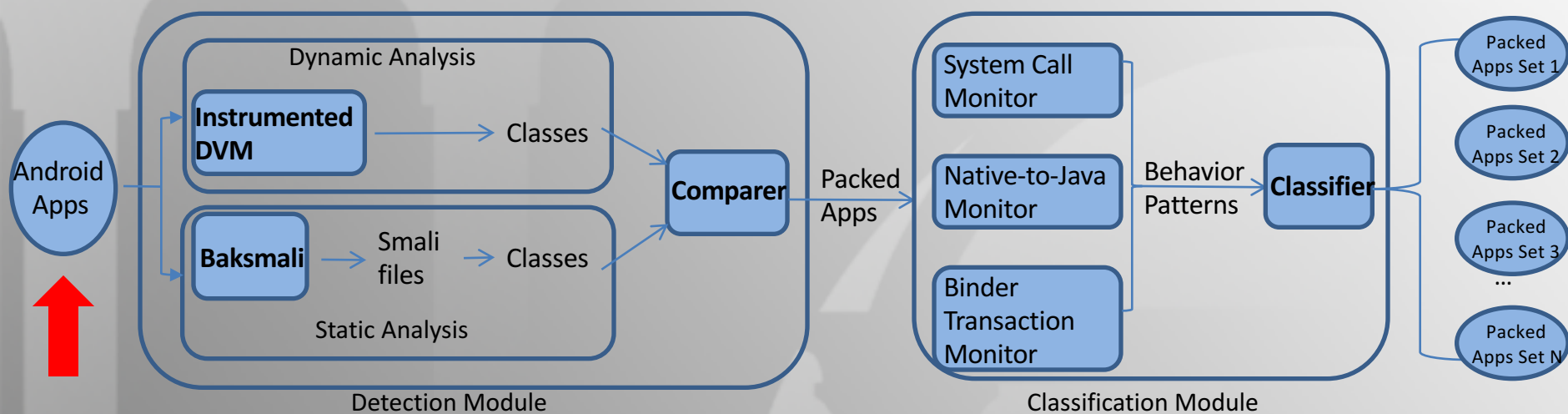
Classification

- Runtime environment monitoring to capture the execution behavior pattern
 - System calls
 - Kernel modules
 - IPC transaction
 - Binder trace
 - Native-to-Java interaction
 - JNI trace

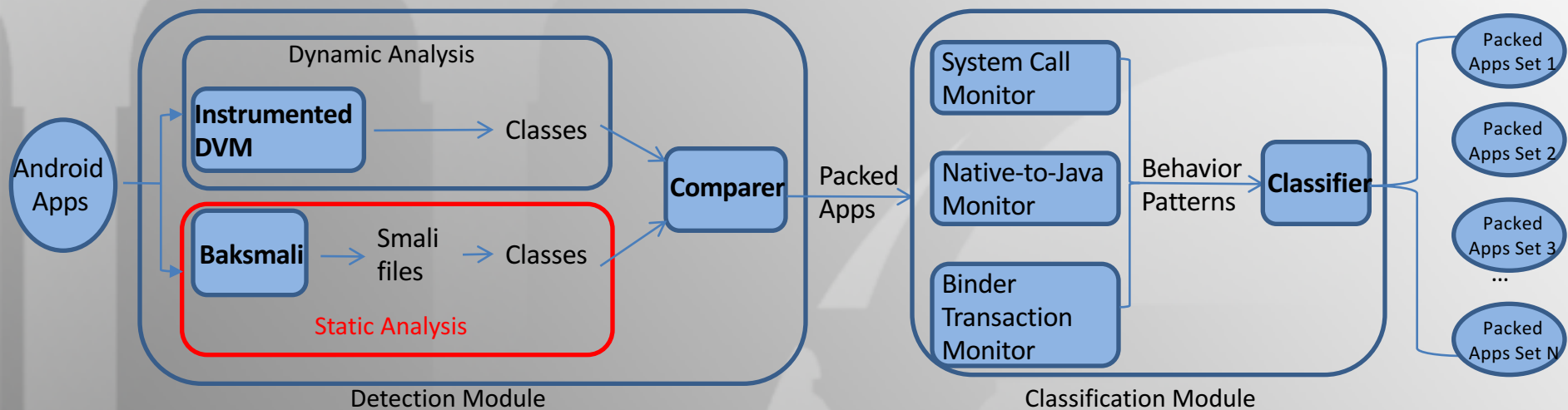
Overview



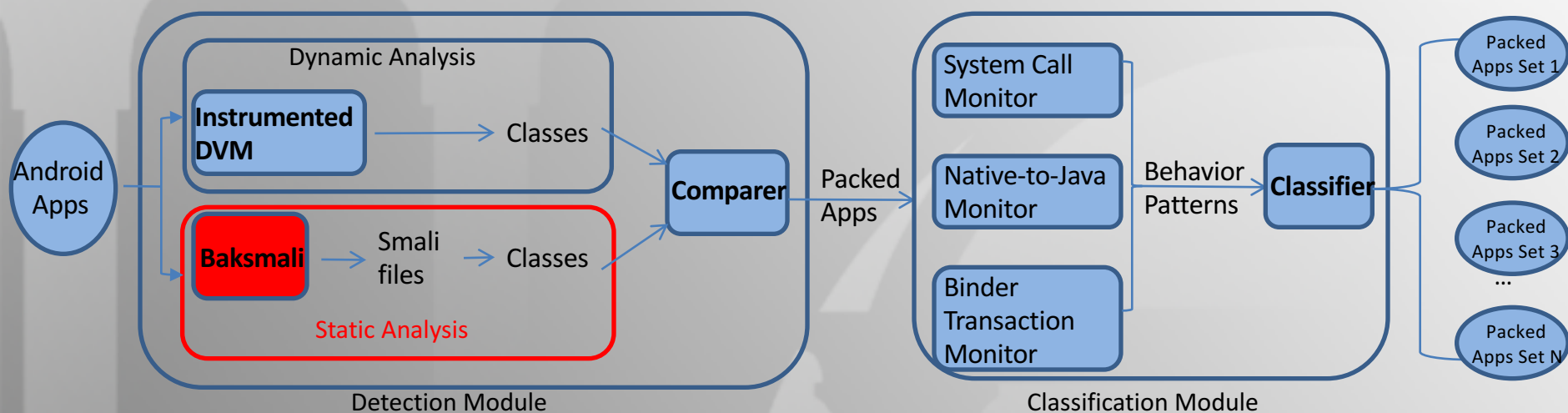
Overview



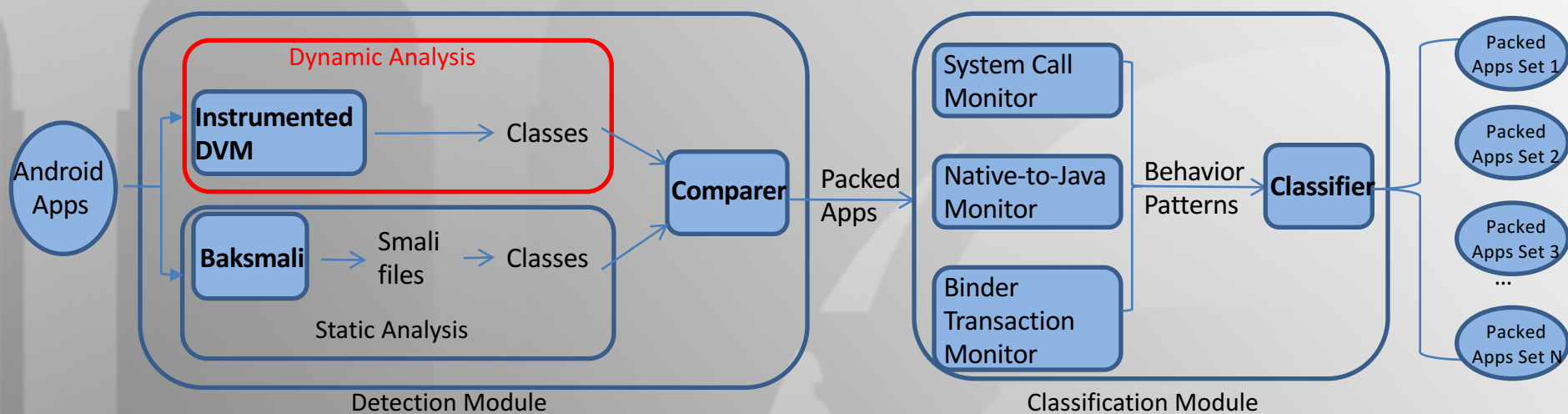
Overview



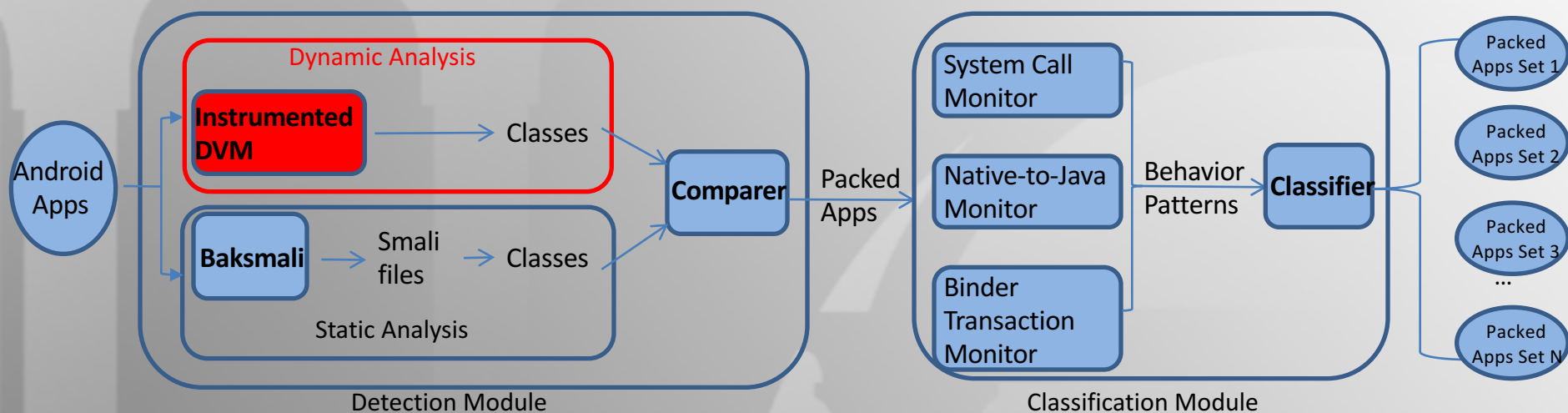
Overview



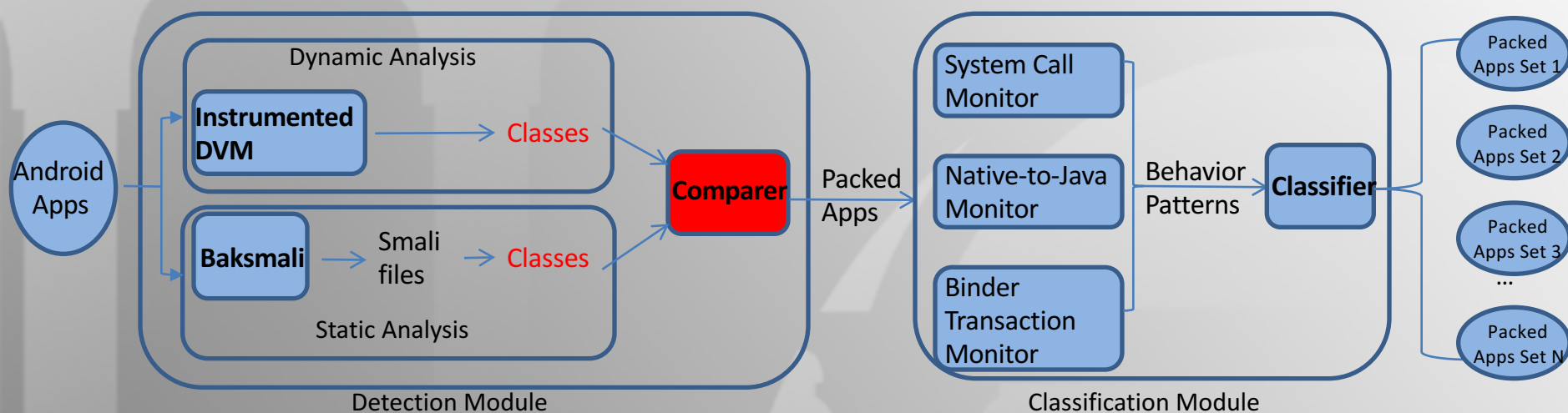
Overview



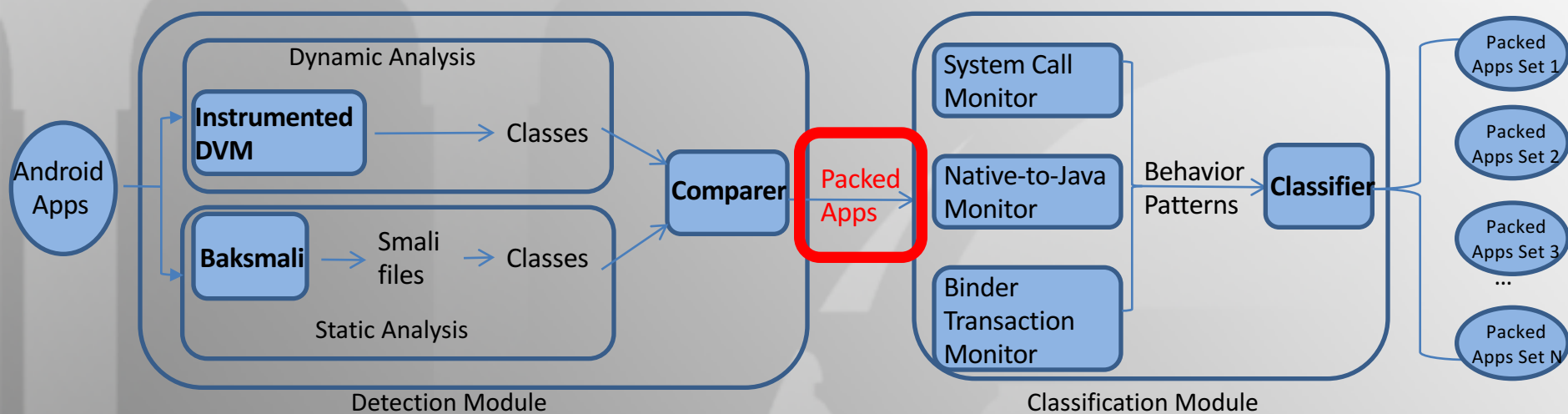
Overview



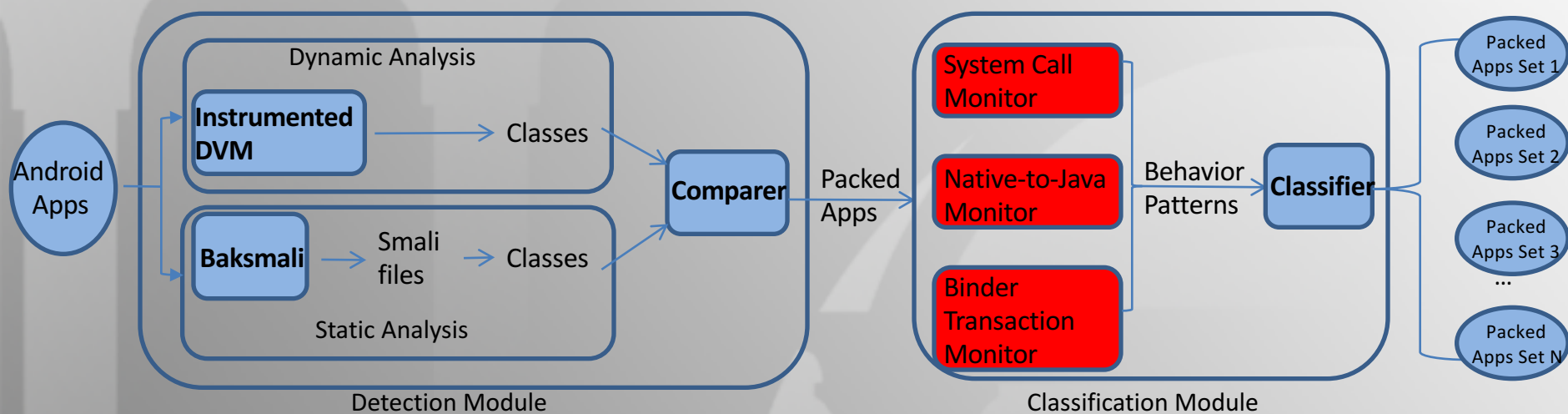
Overview



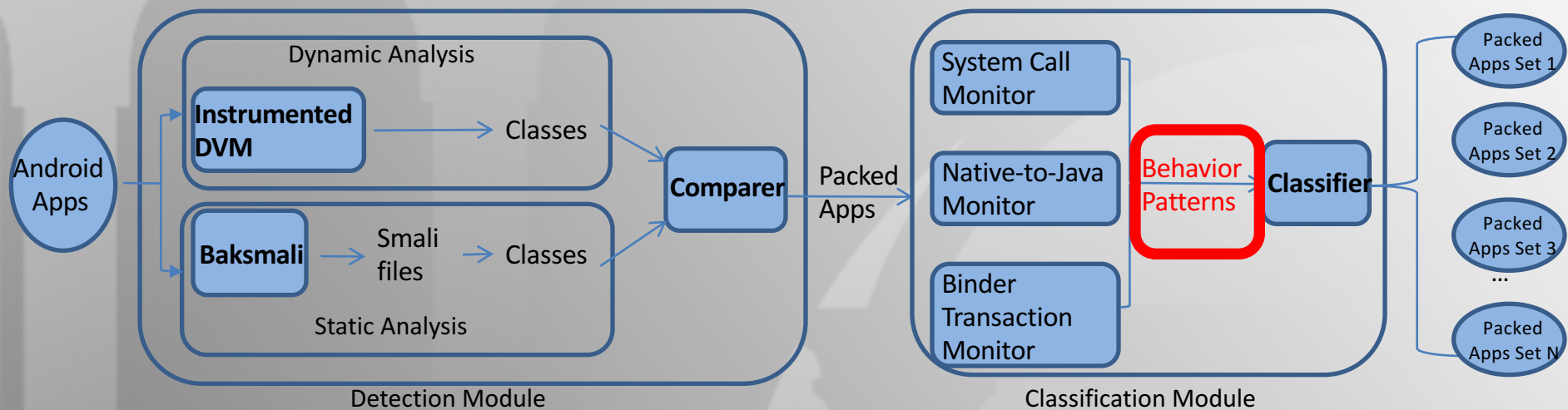
Overview



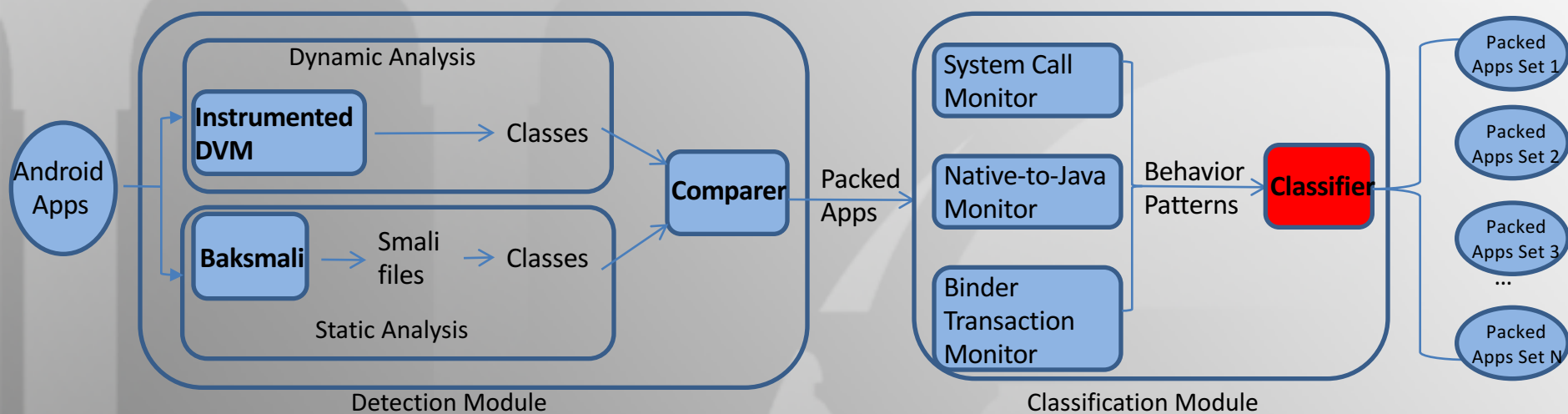
Overview



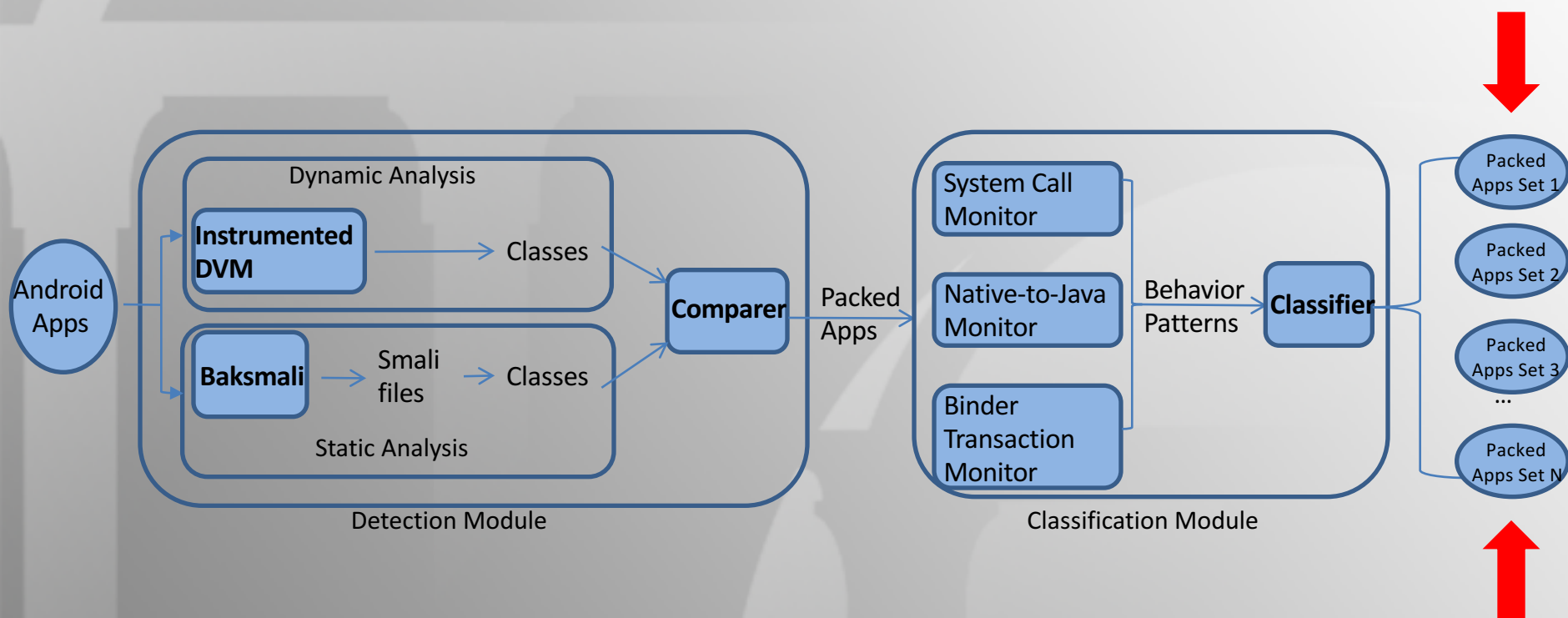
Overview



Overview

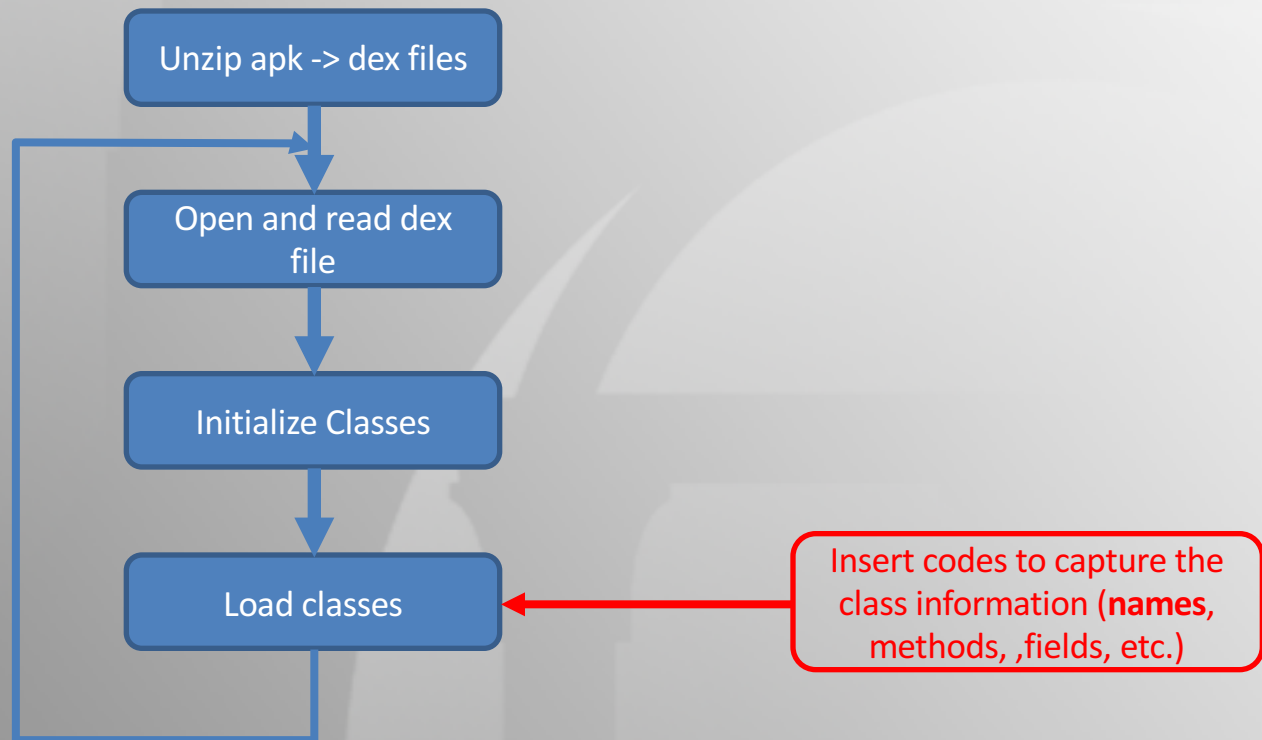


Overview





VM instrumentation (DVM)



Conclusion

- Implemented a detection module to identify packed Android Apps
- Proposed approaches to extract the execution behavior from different packers



Thank you

QA?