ScanDal:
안드로이드 악성 앱 정적 분석기

김진영 윤용호 이광근
서울대학교 프로그래밍 연구실

제 9회 ROSAEC 워크샵
2013년 2월 2일

안드로이드 악성 앱

- 안드로이드 악성 앱 증가
  - 2012년 말 350,000건
  - 2013년 중 1,000,000건 예상
- 공식 마켓에 올라오는 경우도 많음

2012 Mobile Threat and Security Roundup (TrendLabs)
2013 정보보호 10대 이슈 전망 (한국인터넷진흥원)

목표

- 안드로이드 앱 신뢰성 자동 검증 시스템
- 정적분석 (static analysis) 기술
  - 요약해석 기반
- 안전한 (sound) 분석
  - 안드로이드 앱 안심 마켓/안심 기기

대상 신뢰성

- 민감한 개인정보의 누출여부
  - 소스: 위치정보, 기기식별번호 등
  - 싱크: 인터넷, 파일, 메시지 등
  - API 라이브러리의 리스트
- 소스로부터 싱크까지의 누출이 있는지

누출의 예

Google Wallpapers 4.2.2

ScanDal 상황

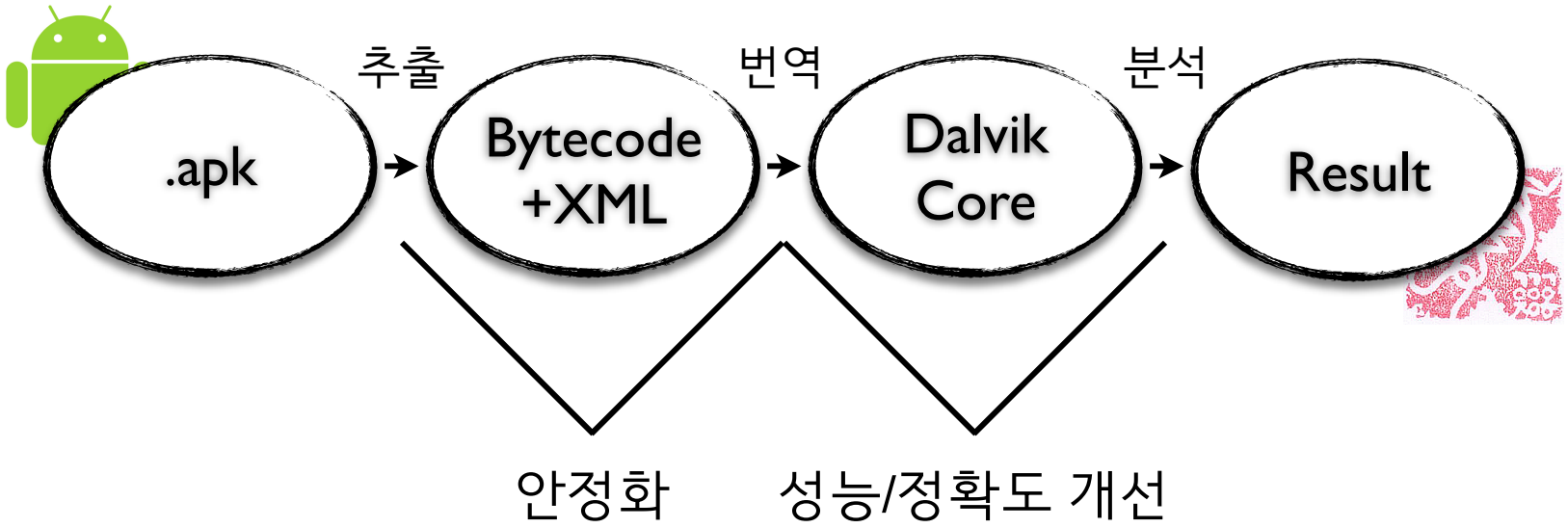
- 실제 앱에서 개인정보 누출 확인 (MoST '12)
  - 공식마켓 인기앱 11개, 비공식마켓 악성앱 8개

- 실험을 확장하며 만난 문제들
  - 갤럭시 S2 기본앱 195개 중 59개 (30%) 에서 시간초과 (30분) 또는 오류
  - 누출이 부풀려지는 경우가 많음

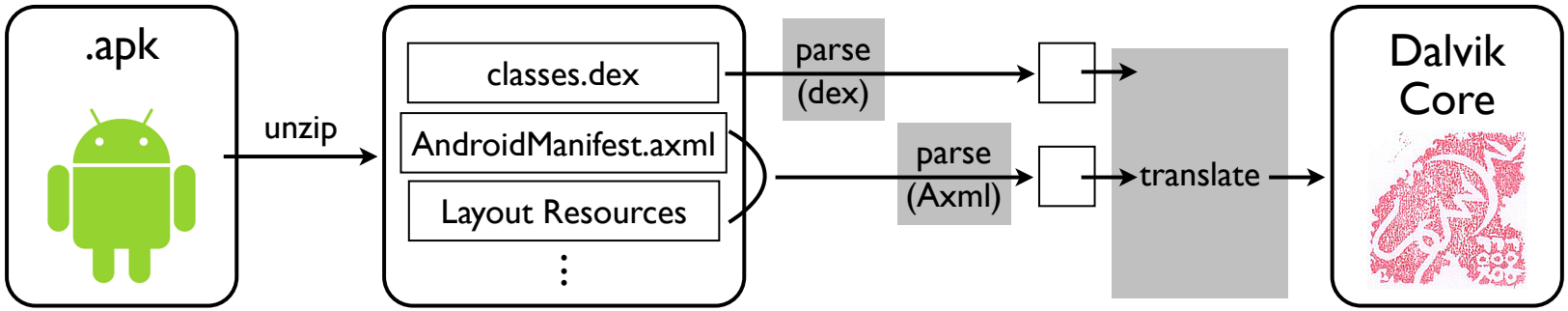
새 목표

- 안정화
- 허위 경보 줄이기
- 분석 시간, 메모리 소모 줄이기
- 앱 검수에 유용하게 활용할 수 있도록

ScanDal 큰그림



앞단



- 문제 원인
  - 오픈소스/SDK에 포함된 툴의 문제
  - 표준/가이드라인을 따르지 않는 앱
  - 악성앱의 난독화 파싱/번역에러 유발

안정화

- AXML 파싱은 여러가지 툴로 시도
  - aapt(SDK), AXMLPrinter2(Java 오픈소스)
  - 하나라도 성공하면 읽어오도록
- 발견된 각종 예외상황 처리
- 난독화에 대응

난독화 문제 예

- 실제 실행흐름이 도달하지 않는 지점에서 적법하지 않은 명령어로 변조된 악성 앱
- 별도의 명령어로 번역하고, 분석을 통해 실행흐름이 도달하지 않음을 확인

```
    8 : v0=0
    9 : v2=94
   11 : if v0<=v2 then jump 16
   13 : jump 37891 (* ?? *)
   16 : ...
        ...
37889 : v4=1
37890 : v2[v3]=v4
37892 : v2=v10.JPFreq
```

새 벤치마크

- 새 벤치마크로 안정성 확인
  - 공식 마켓 인기 앱 250개
  - 갤럭시 S2 기본앱 195개
  - 갤럭시 S 기본앱 169개
  - 악성 앱 1360개*

*http://www.malgenomeproject.org

정확도/성능 개선

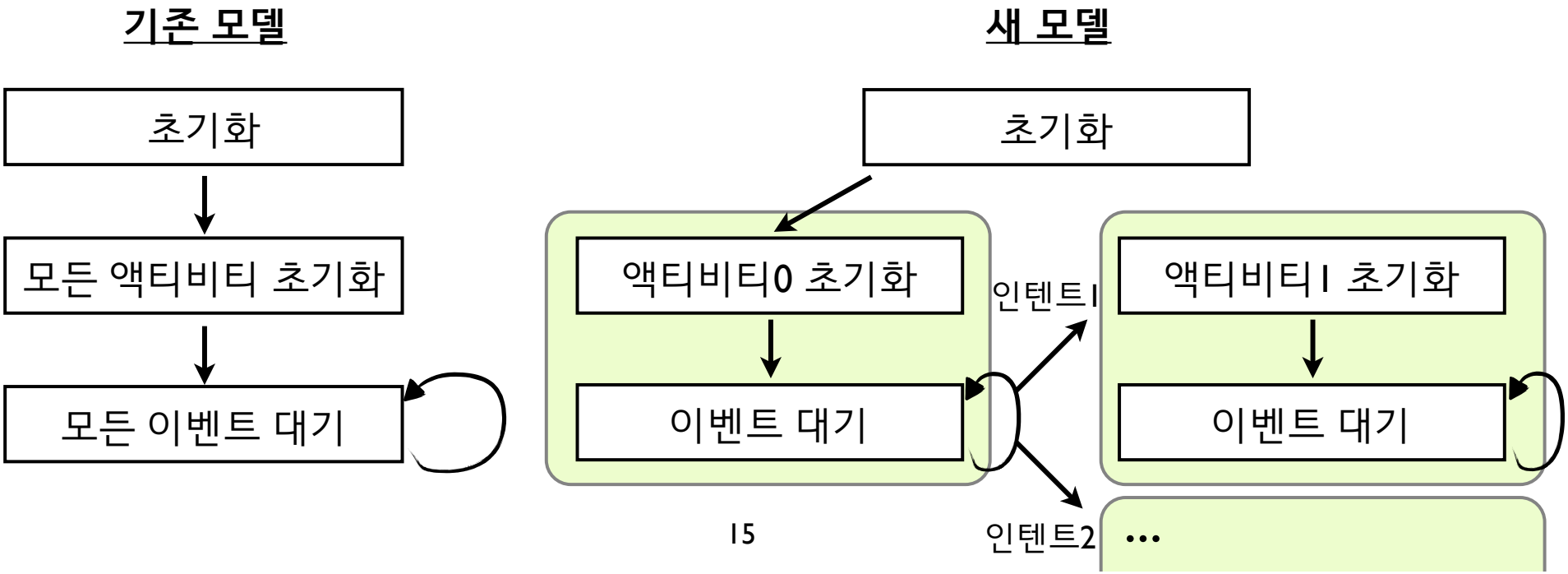
- 함수호출 분석시
  - 불필요한 메모리는 잘라내고 전달
  - 일부 함수는 문맥을 구별해서 분석
- 실제 실행에 가깝도록 실행모델 개선
- Object 타입을 더 정교하게 분석

Sparse 분석은요?

- 하고 싶지만..
- 값싸면서도 쓸모있는 전분석이 어려움
  - 레지스터 기반의 기계어
  - 빈번한 API 라이브러리 호출
- 적절한 균형의 전분석을 찾거나
- 전분석 없는 방식을 고민하거나

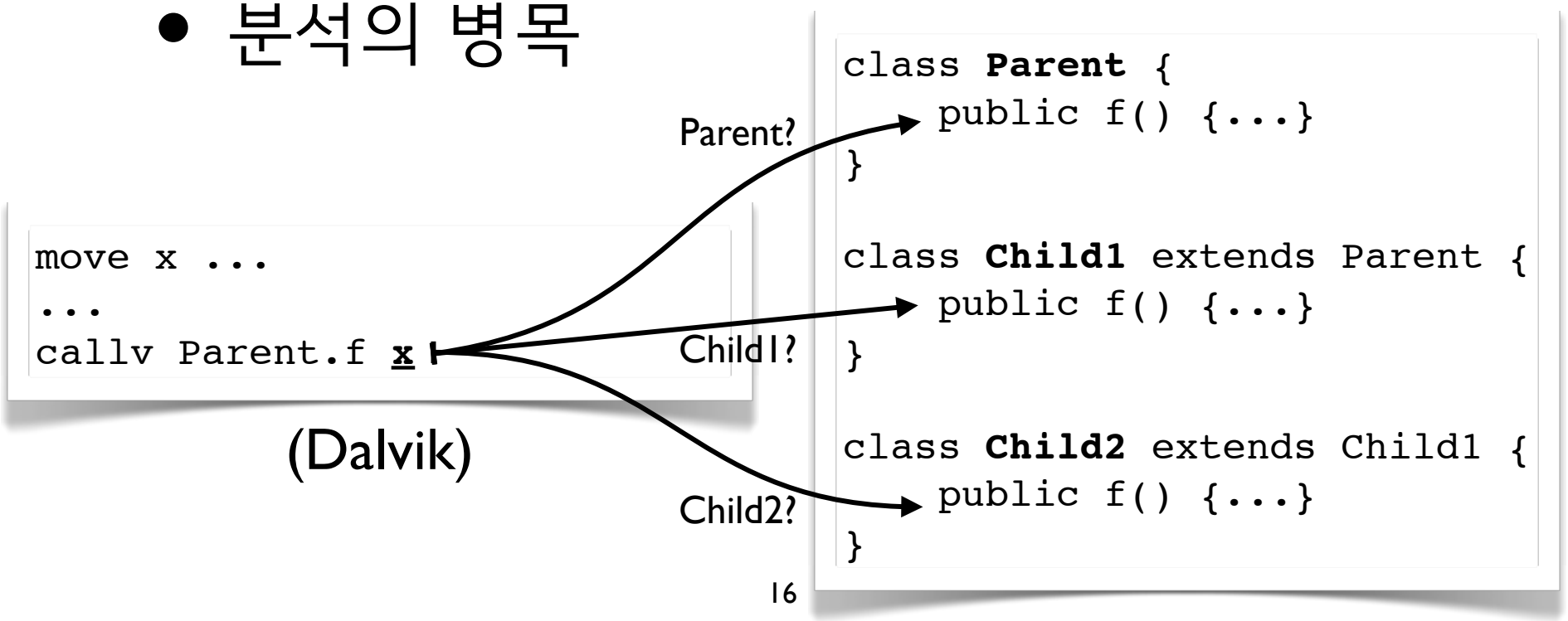
실행모델 개선

- 앱을 중간언어로 번역시 적절한 실행모델 필요
- 기존 모델: 모든 초기화 후 모든 이벤트 대기
- 새 모델: 인텐트를 통한 액티비티 전환을 파악



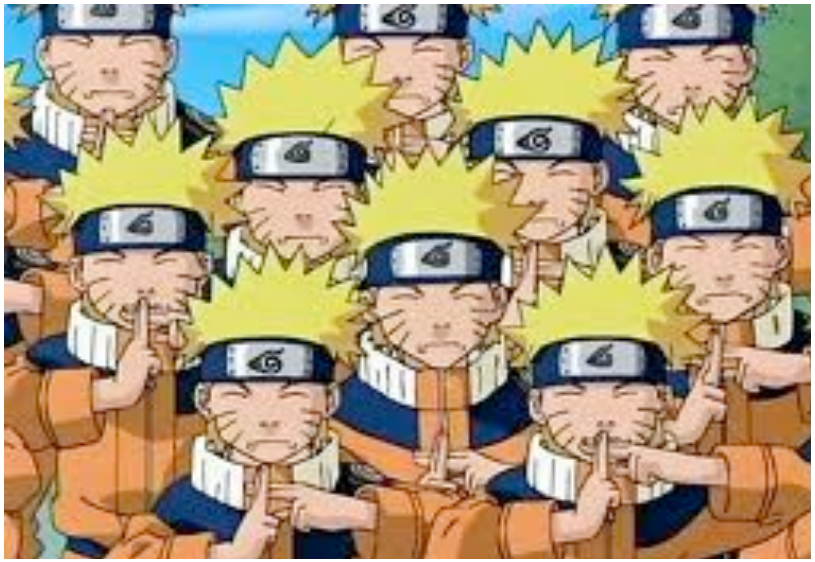
Virtual Call

- 상속을 활용해, Object의 타입에 따라 다른 메소드가 호출되도록 작성된 코드
- 분석의 병목



같은 이름의 메소드가 많음

- Java API 클래스의 흔한 메소드
  - toString(), equals(), run(), ...
- 코드의 난독화
  - a(), b(), c(), ...
- 실제로 일어나지 않는 재귀호출이 생기기도



Object 타입 도메인

- 하나의 타입으로 정해지는 경우 - “ty”
  - 새로 만들어지는

```
new r0 ArrayList ..
```

- 어떤 타입의 서브타입일 수 있는 경우 - “ty*”
  - API 라이브러리 호출이 되돌려주는

```
callv List.get(I)Object r0 ..
```

타입정보로 가지치기

```
move x ...
...
callv Parent.f x
```

```
class Parent {
    public f() {...}
}

class Child1 extends Parent {
    public f() {...}
}

class Child2 extends Child1 {
    public f() {...}
}
```

Reg -- x: {pc}

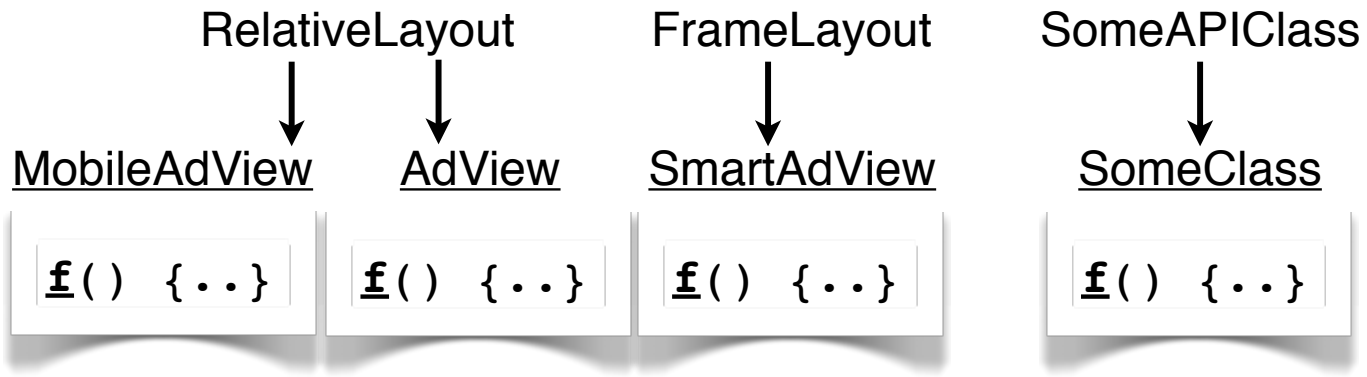
Mem -- pc.ty = Child1*

상속관계를 알아야

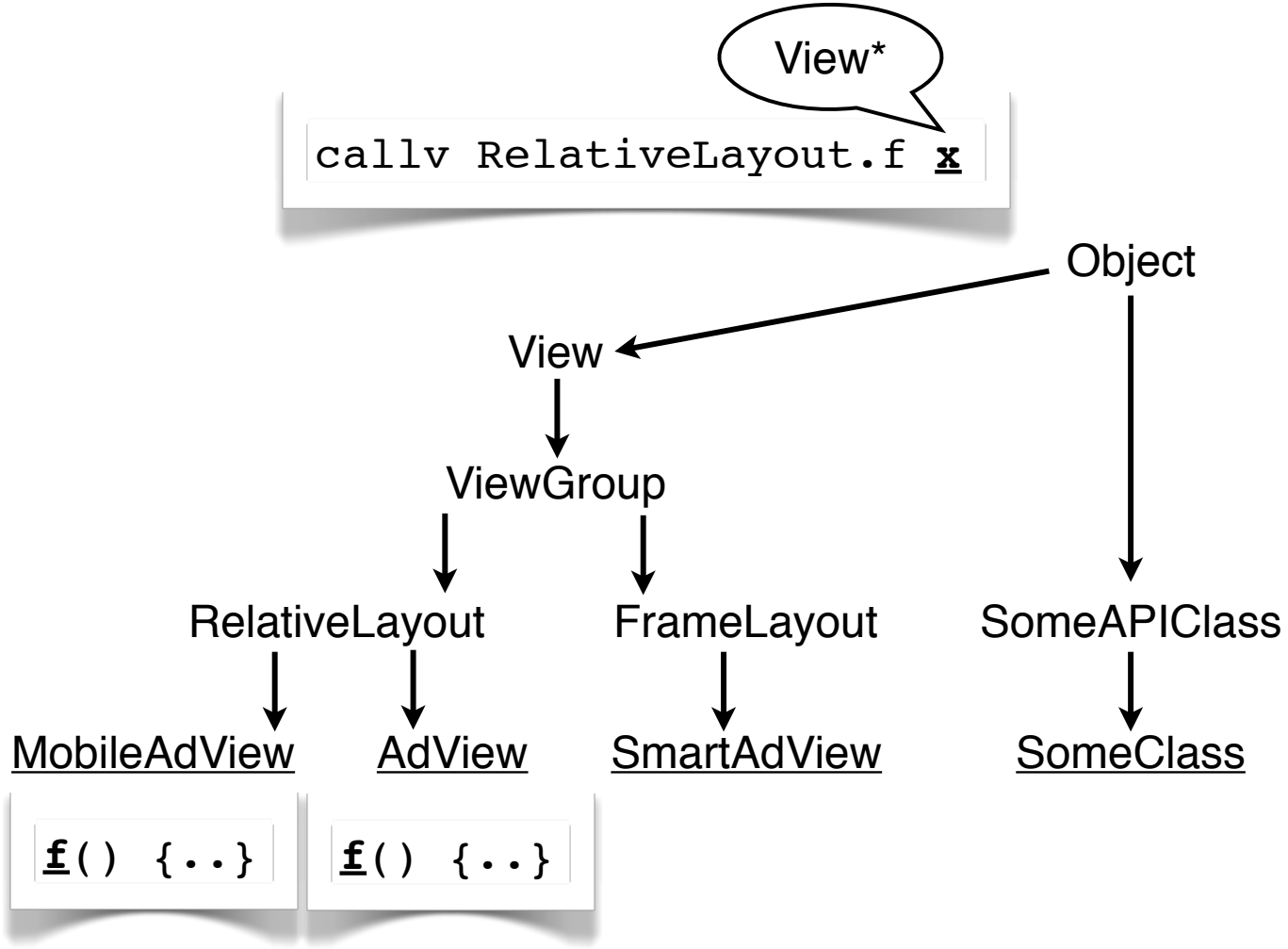
- 3000개가 넘는 API 클래스
  - 자바, 안드로이드, ...
- 상속관계를 플랫폼으로부터 읽어옴
- Dalvik Core로 번역시 장착

상속관계로 가지치기


View*
callv RelativeLayout.f x


RelativeLayout
FrameLayout
SomeAPIClass
MobileAdView
AdView
SmartAdView
SomeClass
f() {..}
f() {..}
f() {..}
f() {..}

상속관계로 가지치기


View*
callv RelativeLayout.f x
Object
View
ViewGroup
RelativeLayout
FrameLayout
SomeAPIClass
MobileAdView
AdView
SmartAdView
SomeClass
f() {..}
f() {..}

개선 결과 (1)

앱	개선 전			개선 후		
	time(sec)	mem(MB)	#alarm	time(sec)	mem(MB)	#alarm
Kids Preschool Puzzle	1	62	29	1	29	6
Job Search	1	95	7	1	45	3
Kids Shapes	2	137	36	4	77	6
Kids ABC Phonics	3	109	30	1	47	6
Backgrounds HD Wallpapres	4	133	10	1	30	3
Bible Quotes	8	265	3	1	40	2
ES Task Manager	20	424	3	29	302	2
Multi Touch Paint	42	718	53	56	431	38
Adao File Manager	67	1143	14	116	819	1
(D-Day) The Day Before	225	2648	14	84	976	1
Kids Numbers and Math	559	176	29	1	41	6

개선 결과 (2)

- 시간제한: 30분
- 갤럭시 S2 기본앱 195개
  - 시간초과 50개 (26%) -> 28개 (14%)
- 공식 마켓 인기앱 250개
  - 시간초과 160개 (64%) -> 60개 (24%)
- 일부 큰 앱, 라이브러리 호출이 빈번한 앱은 더 개선 필요

결론 및 앞으로

- ScanDal 안정화 및 성능, 정확도 개선

- 성능 계속 개선 필요
  - 전분석 고민
- JNI를 포함하는 분석
- 일반적인 악성행동으로 확장

감사합니다