A Machine-learning Approach for Classifying and Categorizing Android Sources and Sinks

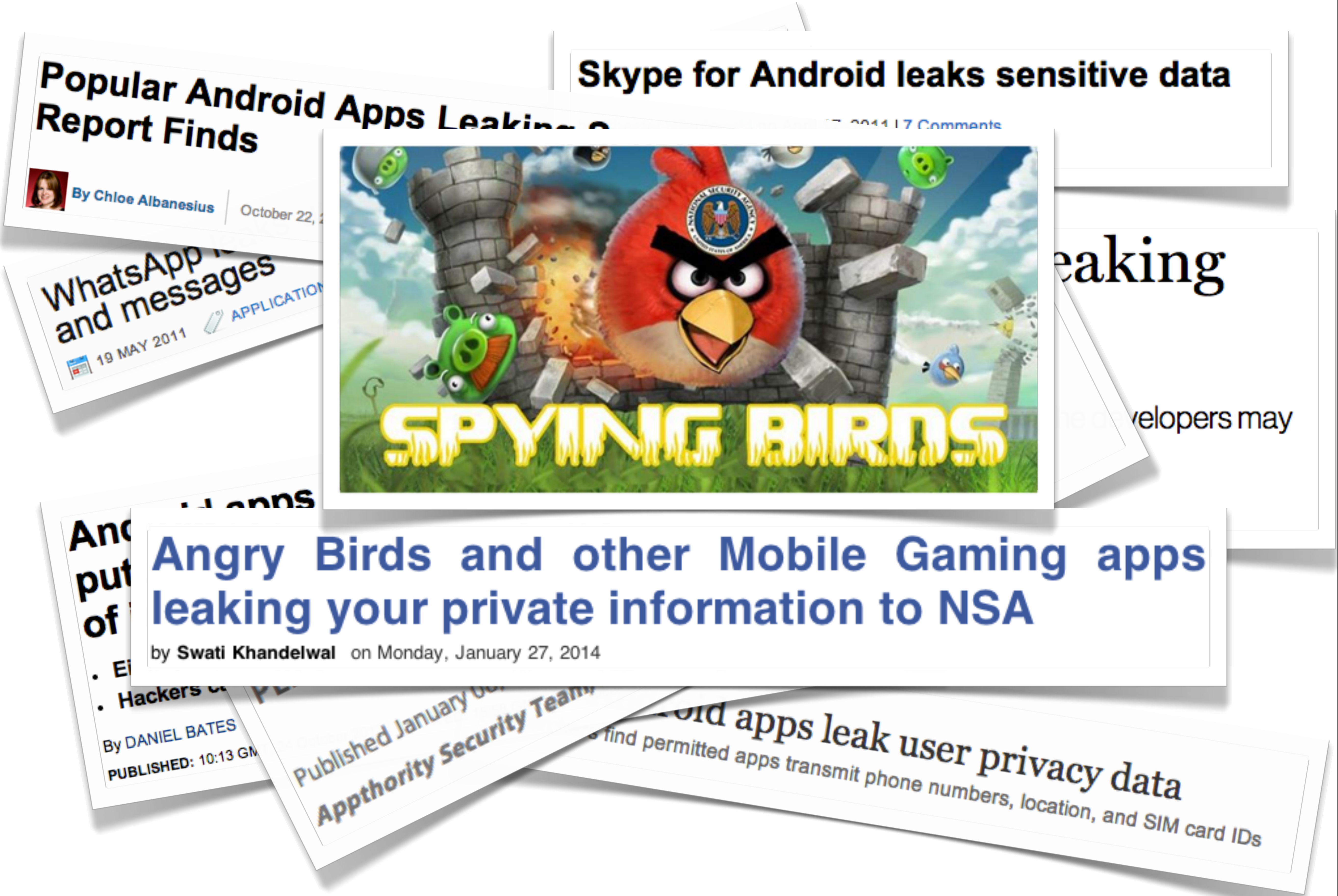
Siegfried Rasthofer, Steven Arzt, Eric Bodden

Fraunhofer
SIT

TECHNISCHE
UNIVERSITÄT
DARMSTADT

SECURE
SOFTWARE ENGINEERING
GROUP

EC SPRIDE



Popular Android Apps Leaking
Report Finds

By Chloe Albanesius | October 22,

Skype for Android leaks sensitive data

2011 | 7 Comments

WhatsApp
and messages

19 MAY 2011 | APPLICATION

eaking

elopers may

SPYING BIRDS

An apps
put
of

- Ei
- Hackers

By DANIEL BATES

PUBLISHED: 10:13 GM

Angry Birds and other Mobile Gaming apps leaking your private information to NSA

by Swati Khandelwal on Monday, January 27, 2014

Published January

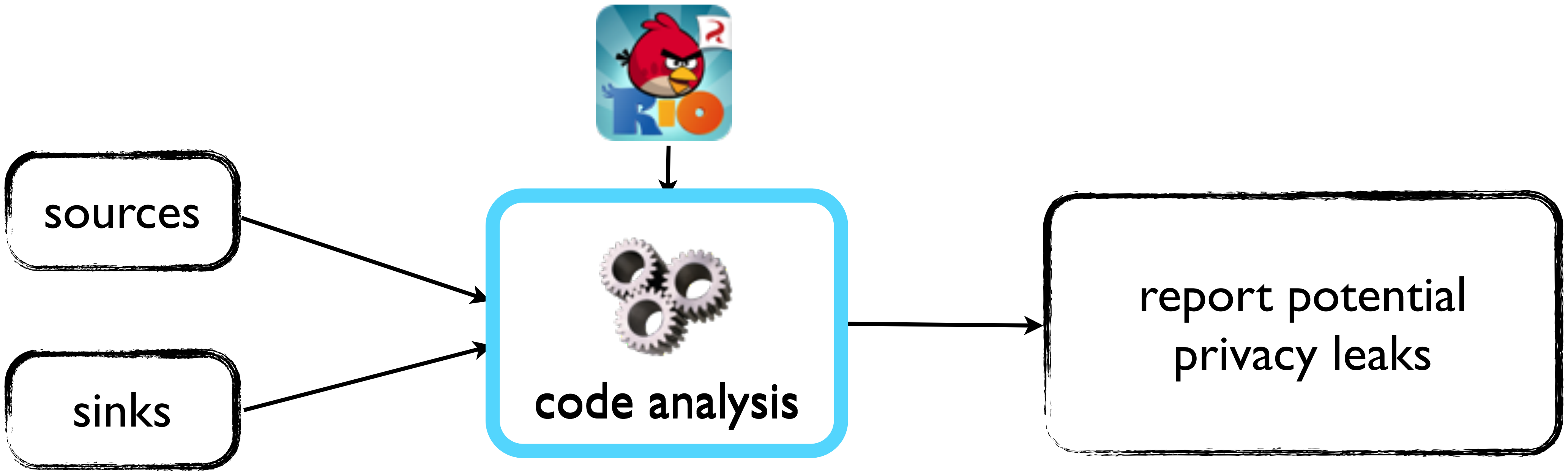
Appthority Security Team,

oid apps leak user privacy data

find permitted apps transmit phone numbers, location, and SIM card IDs

SECURE SOFTWARE ENGINEERING GROUP

EC SPRIDE



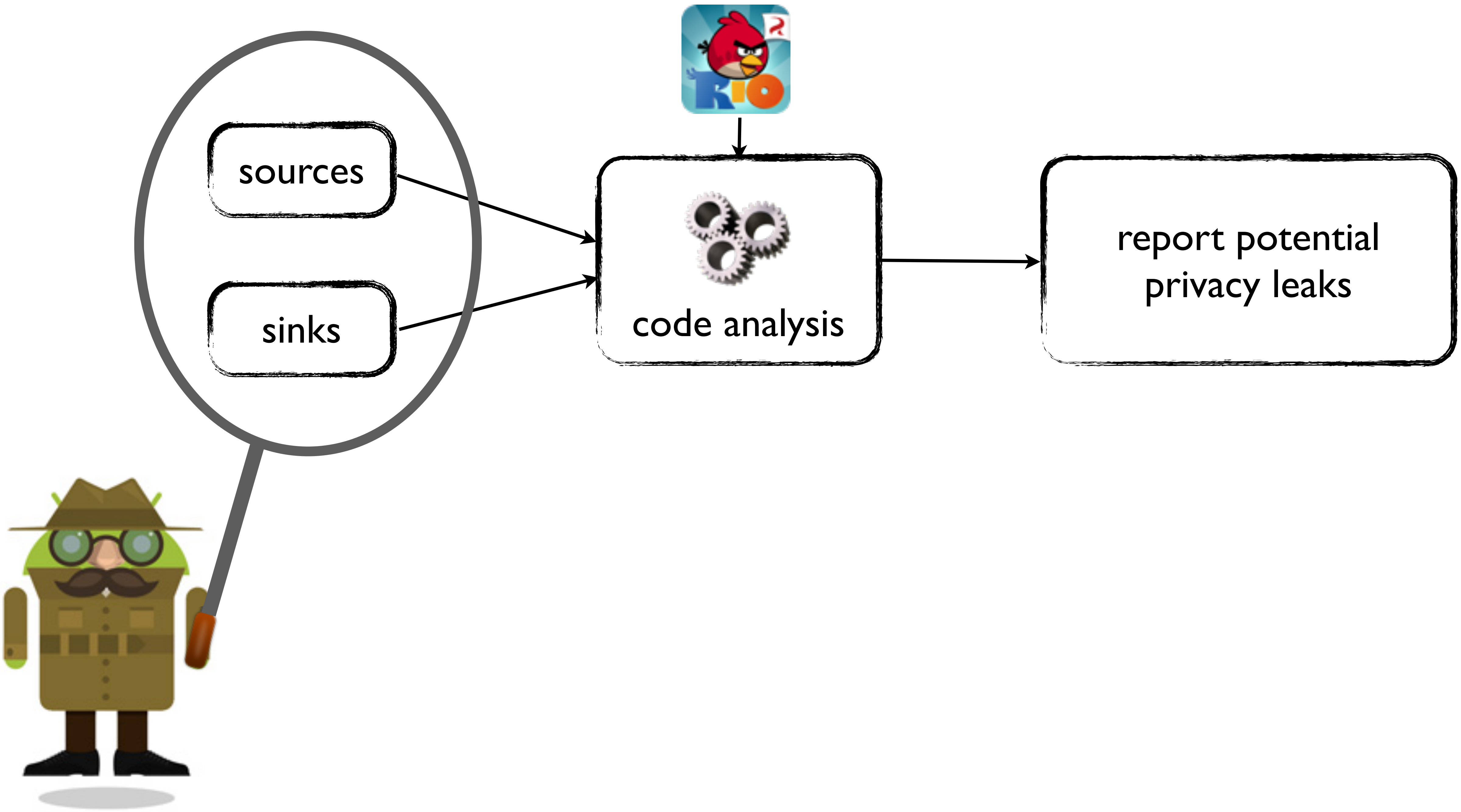
Dynamic Approaches:

TaintDroid [OSDI'10], Aurasium [USENIX'12], "Dr. Android and Mr. Hide" [SPSM'12], etc.

Static Approaches:

ScanDroid [TR 09], DeD [SEC'11], CHEX [CCS'12], LeakMiner [WCSE'12], ScanDal [Most'12], AndroidLeaks [TRUST'12], SAAF [SAC'13], FlowDroid [PLDI'14], etc.

...but wait



sources

sinks

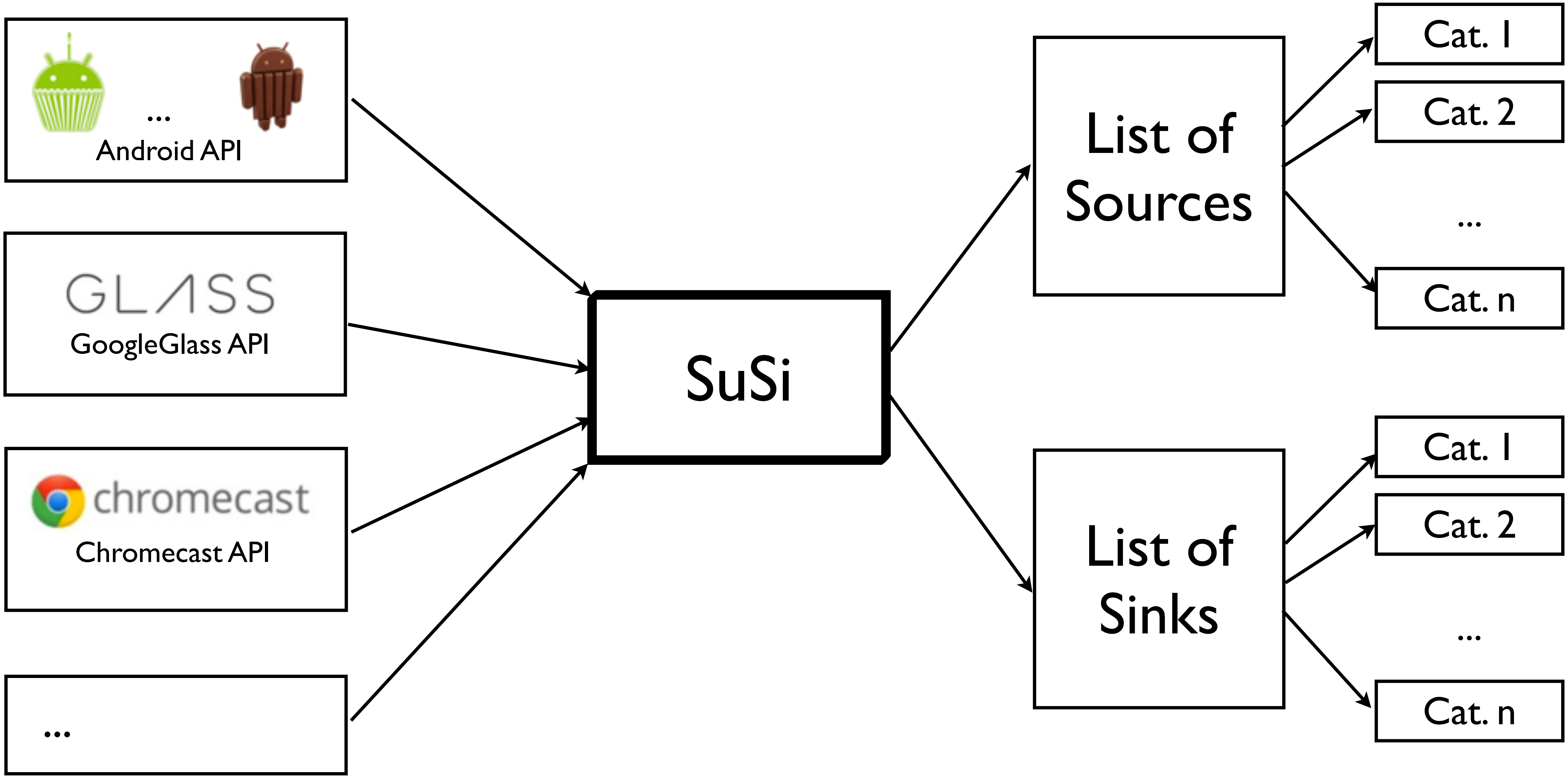
code analysis

report potential privacy leaks

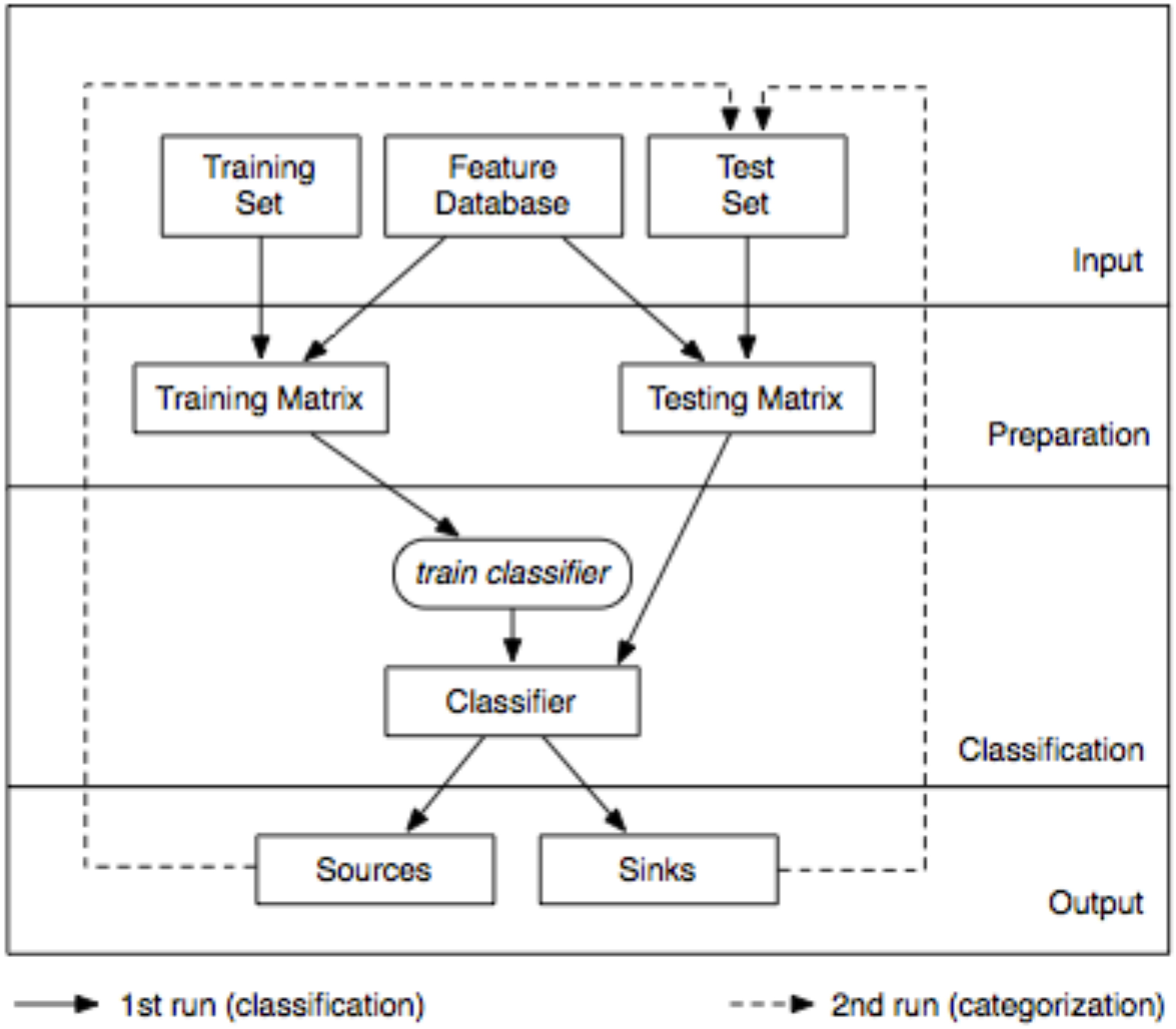
Method	TaintDroid	SCanDroid	DeD
	?		
Location.getLongitude()	✓	✓	✓
Location.getLatitude()	✓	✓	✓
Browser.getAllBookmarks()	✓		

SmsManager.sendTextMessage	✓	✓	✓
Log.d()			✓
URL.openConnection()	✓		

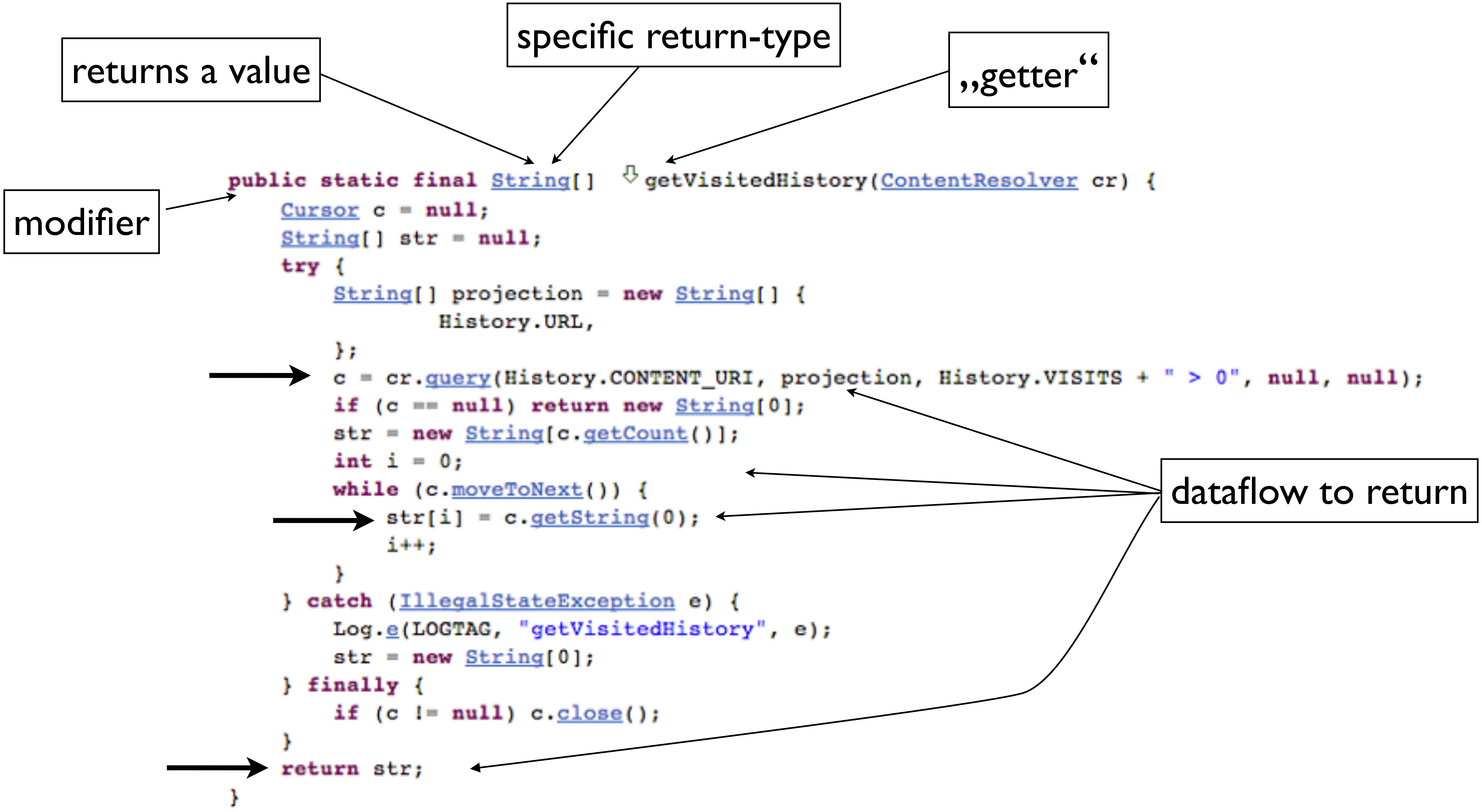
Extracting Sources/Sinks



Machine-Learning Approach



Feature-Database: Classification



```
public static final String[]  getVisitedHistory(ContentResolver cr) {
    Cursor c = null;
    String[] str = null;
    try {
        String[] projection = new String[] {
                History.URL,
        };
        c = cr.query(History.CONTENT_URI, projection, History.VISITS + " > 0", null, null);
        if (c == null) return new String[0];
        str = new String[c.getCount()];
        int i = 0;
        while (c.moveToNext()) {
            str[i] = c.getString(0);
            i++;
        }
    } catch (IllegalStateException e) {
        Log.e(LOGTAG, "getVisitedHistory", e);
        str = new String[0];
    } finally {
        if (c != null) c.close();
    }
    return str;
}
```

returns a value

specific return-type

„getter“

modifier

dataflow to return

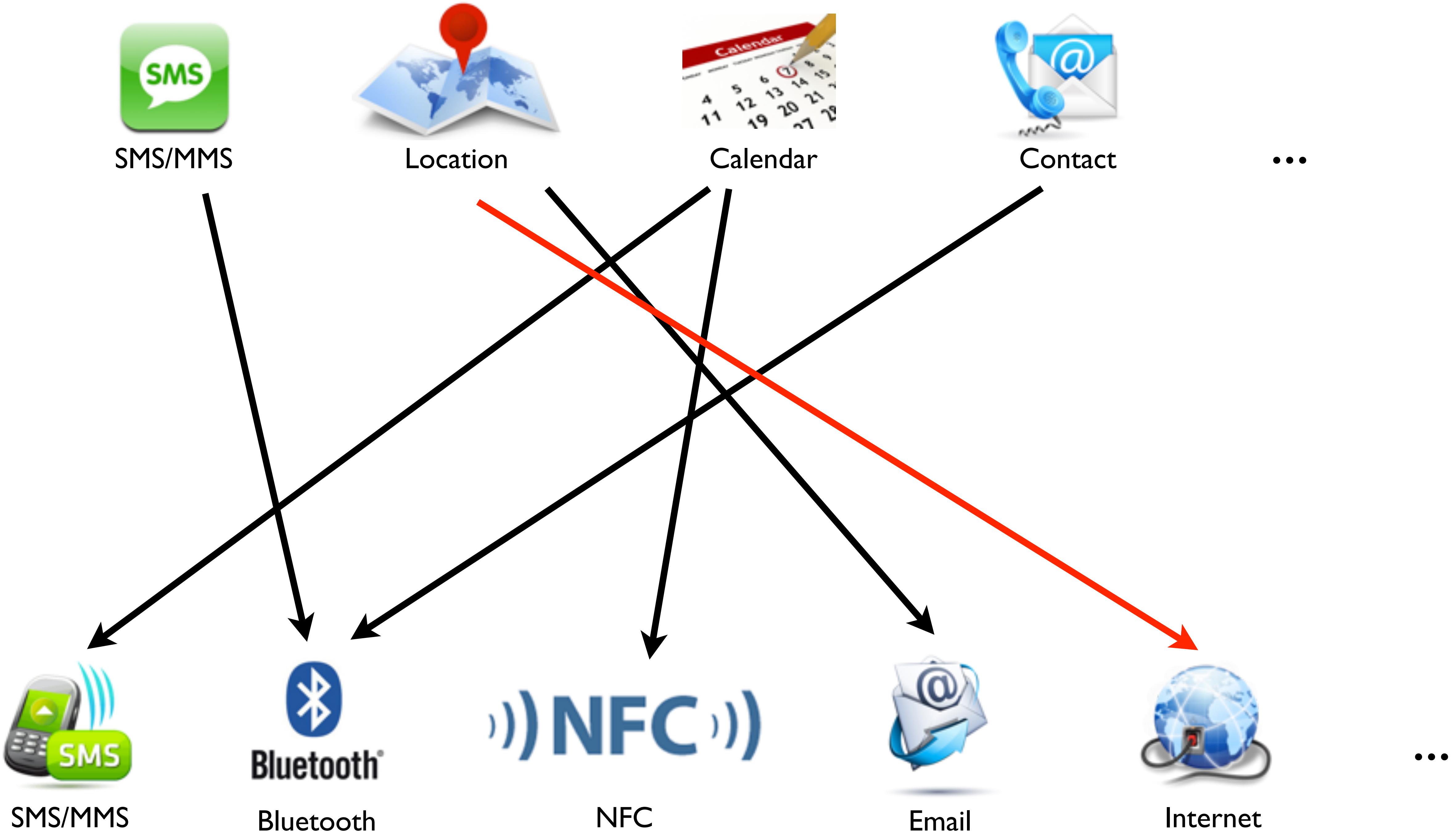
Feature-Database: Classification

Feature-Categories:

- Method name
- Method has parameters
- Method’s return type
- Parameter type
- Method modifiers
- Modifiers of declaring class
- Name of declaring class

- Dataflow to return value
- Dataflow from parameter to (abstract) sink

Feature-Database: Categorization

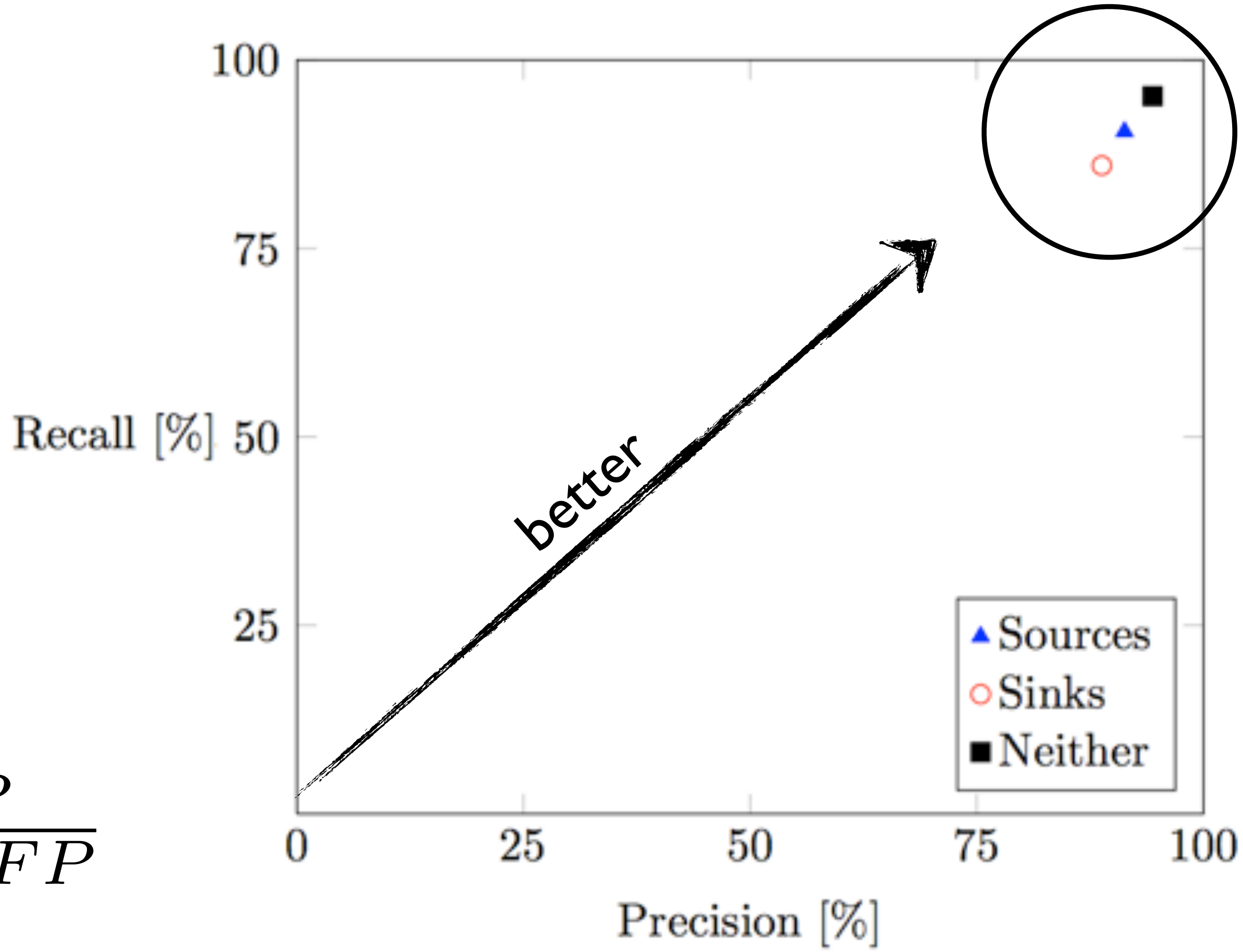


Evaluation

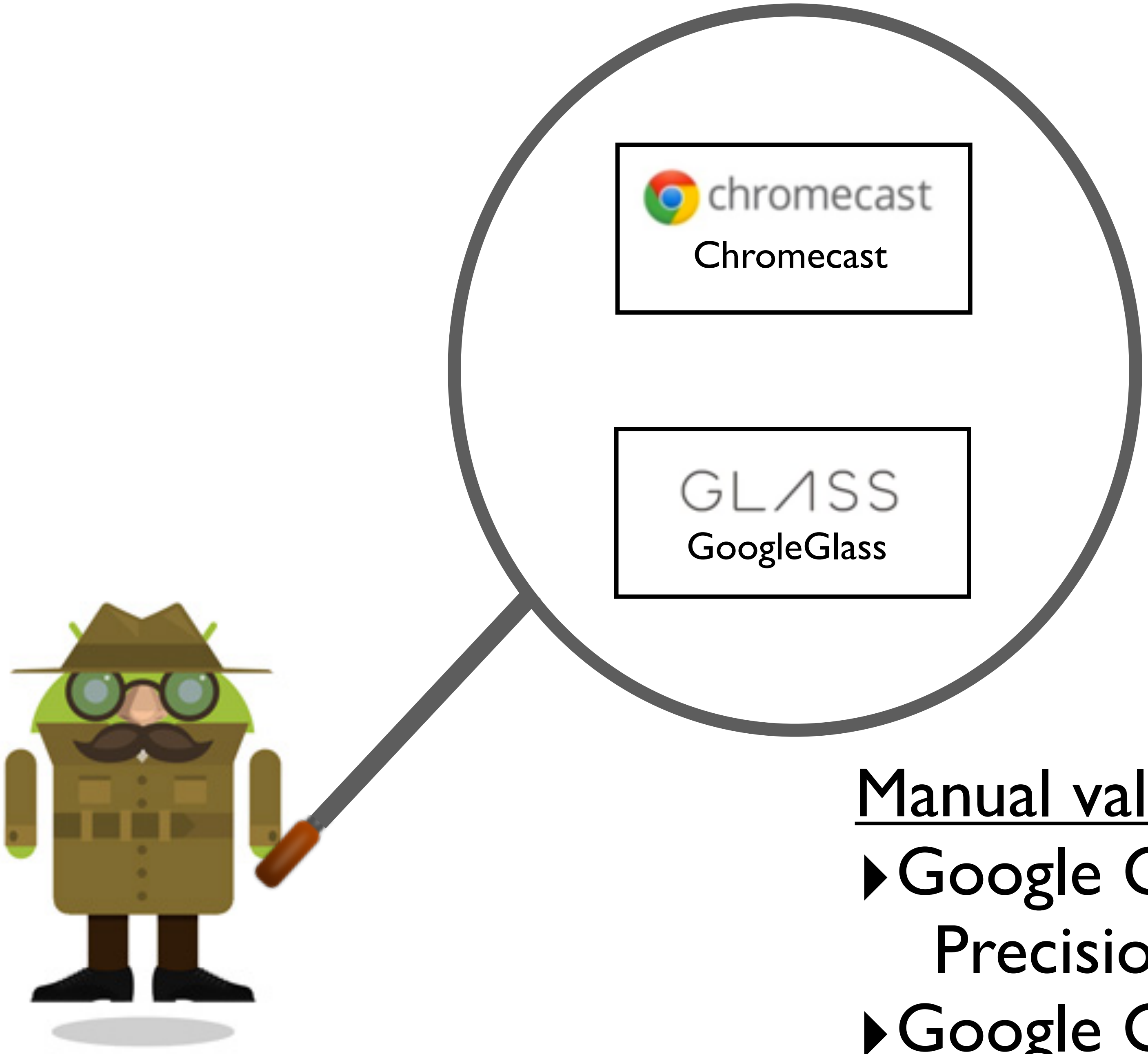
Ten-fold cross validation:

$$Recall = \frac{TP}{TP+FN}$$

$$Precision = \frac{TP}{TP+FP}$$



Evaluation



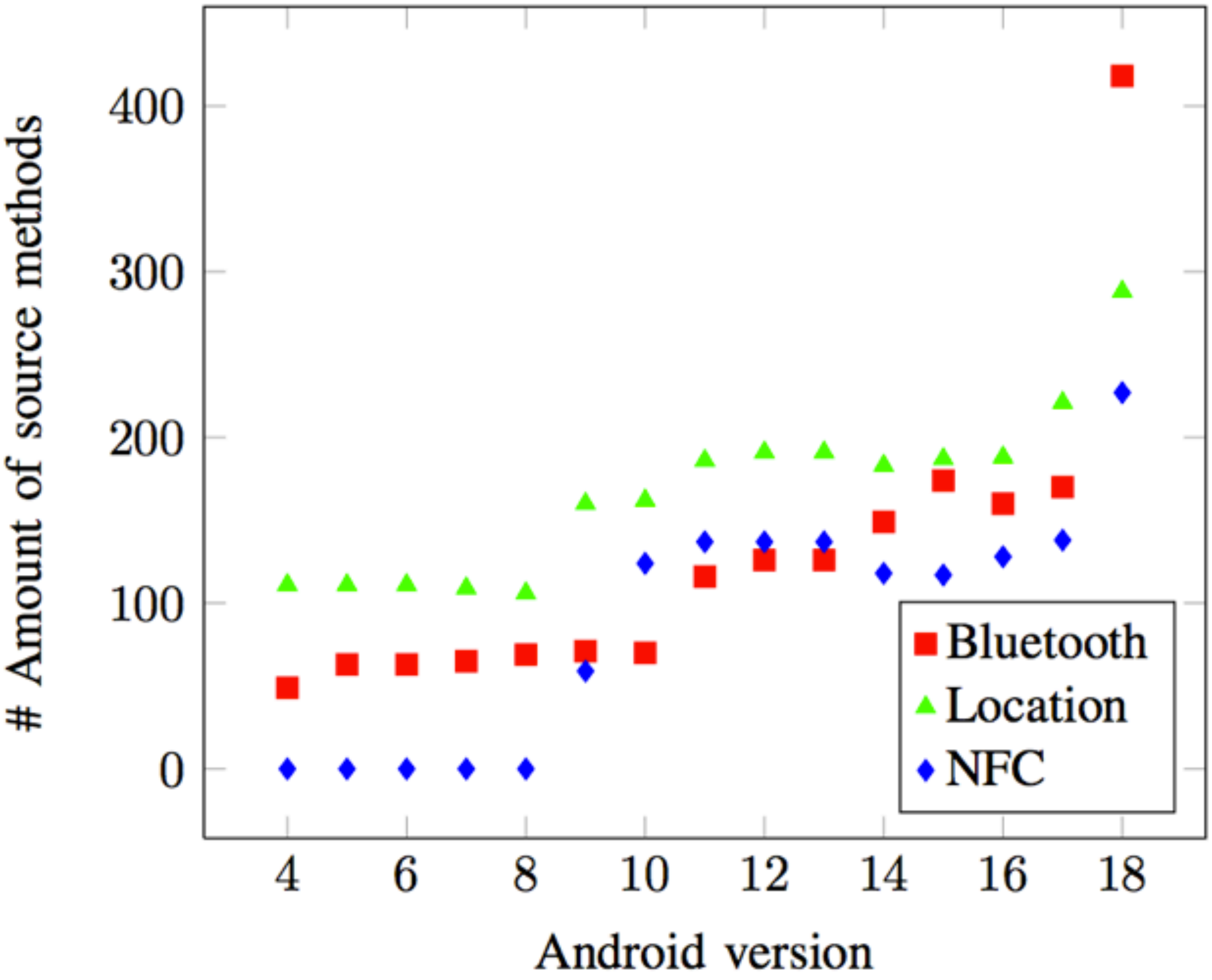
Chromecast

GoogleGlass

<u>Manual validation:</u>

- Google Glass API:
  Precision: 98% and Recall: 100%
- Google Chromecast API:
  Precision and Recall: 100%

Evaluation

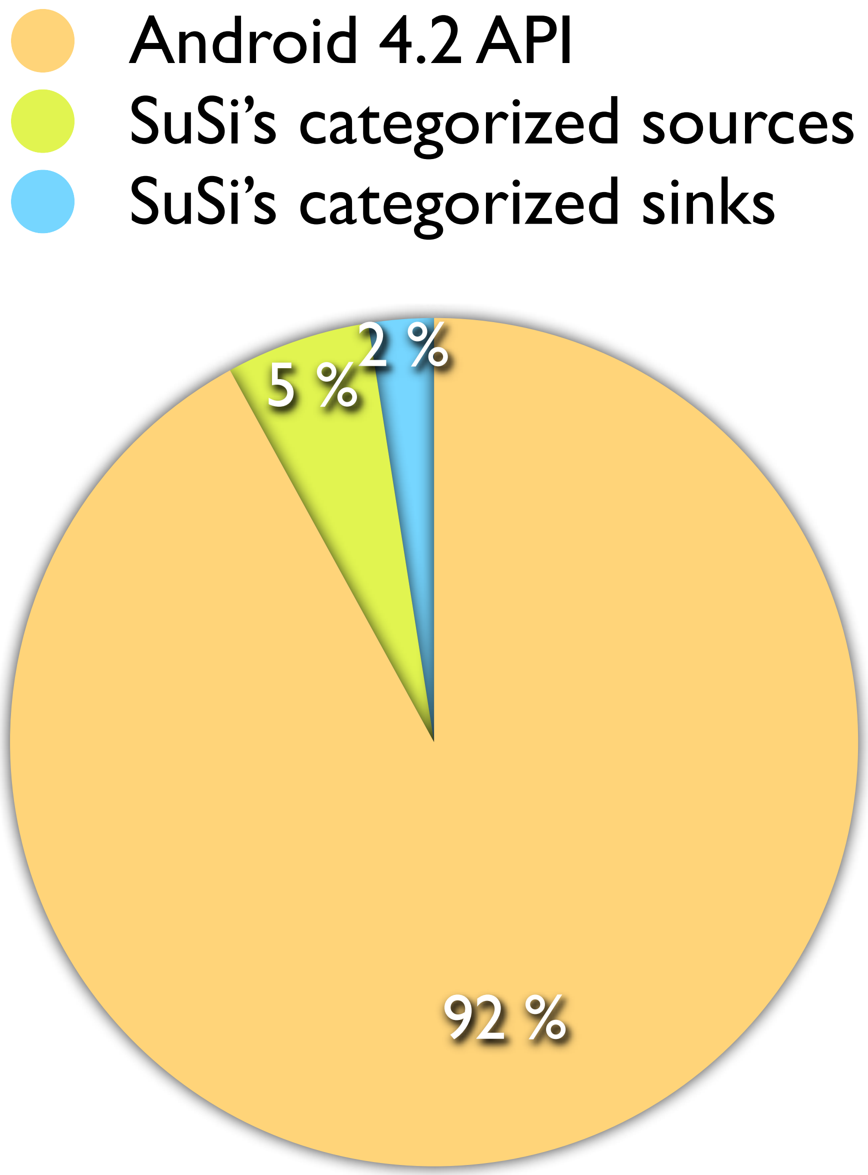


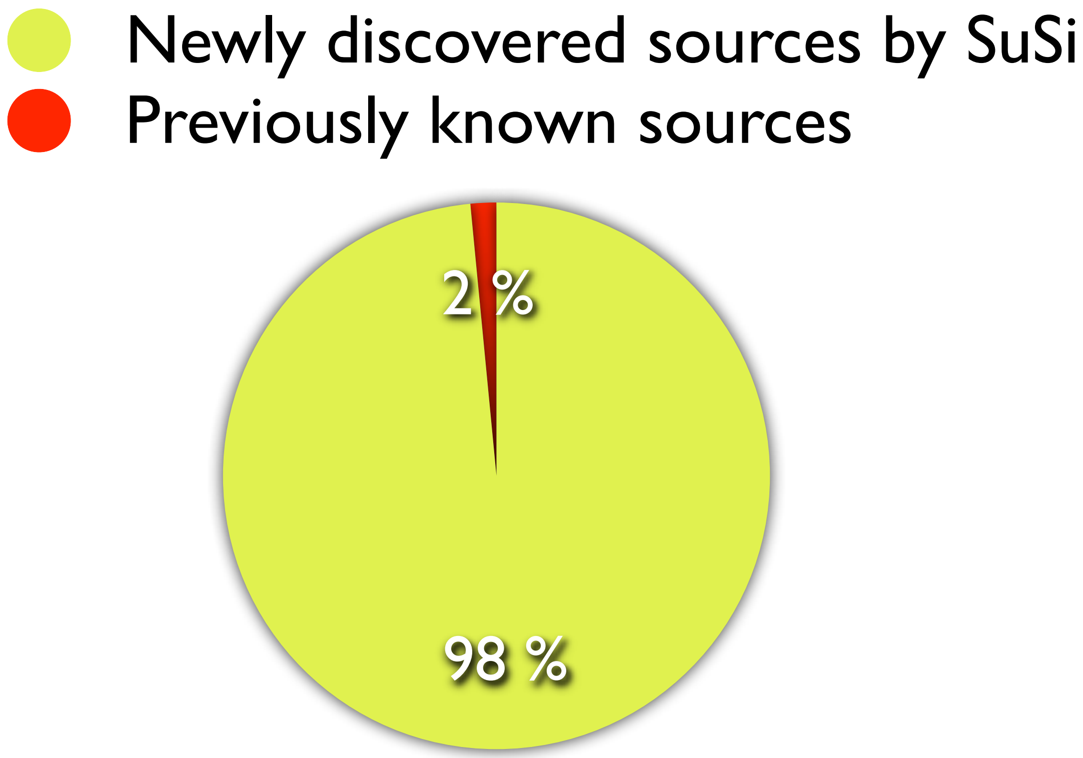
# Amount of source methods
400
300
200
100
0
4
6
8
10
12
14
16
18
Android version
🟥 Bluetooth
▲ Location
◆ NFC

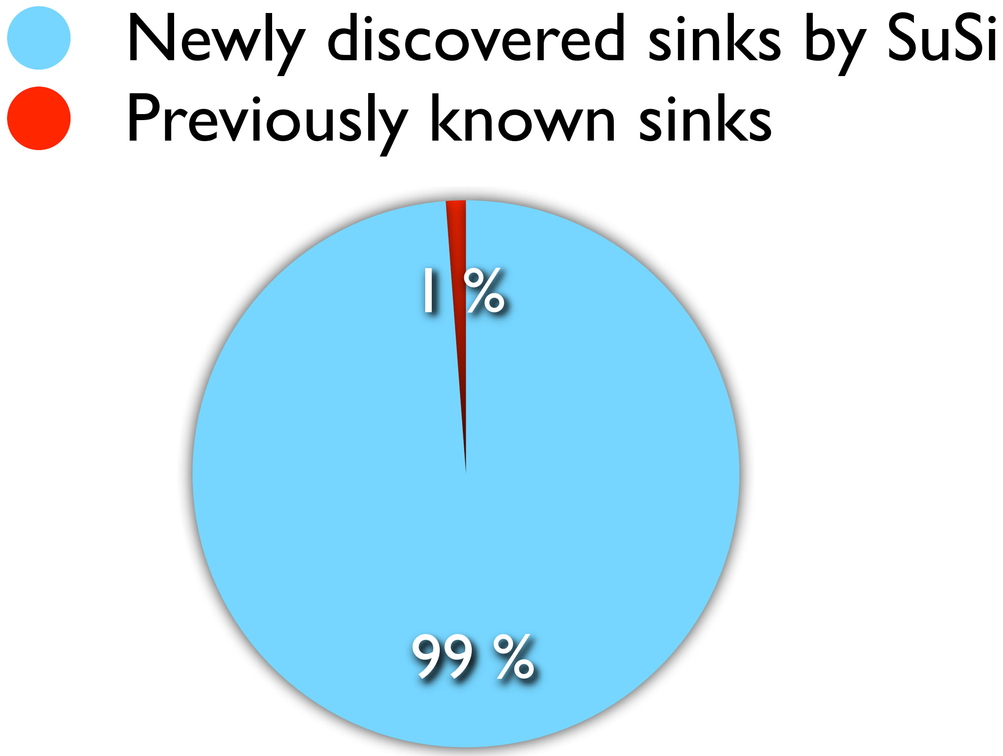
Top Source/Sink Methods in Android-Malware

Method	TaintDroid	SCanDroid	DeD
BluetoothAdapter.getAddress()	✗	✗	✗
WifiInfo.getMacAddress()	✗	✗	✗
Locale.getCountry()	✗	✗	✗
WifiInfo.getSSID()	✗	✗	✗
GsmCellLocation.getCid()	✗	✗	✗
GsmCellLocation.getLac()	✗	✗	✗
Location.getLongitude()	✓	✓	✓
Location.getLatitude()	✓	✓	✓
Browser.getAllBookmarks()	✓	✗	✗

SmsManager.sendTextMessage	✓	✓	✓
Log.d()	✗	✗	✓
URL.openConnection()	✓	✗	✗


Android 4.2 API
SuSi's categorized sources
SuSi's categorized sinks
5 %
2 %
92 %


Newly discovered sources by SuSi
Previously known sources
2 %
98 %


Newly discovered sinks by SuSi
Previously known sinks
1 %
99 %

Open-Source on GitHub:
https://github.com/secure-software-engineering/SuSi

Siegfried Rasthofer
Secure Software Engineering Group (EC-SPRIDE)
Email: siegfried.rasthofer@cased.de
Blog: http://sse-blog.ec-spride.de