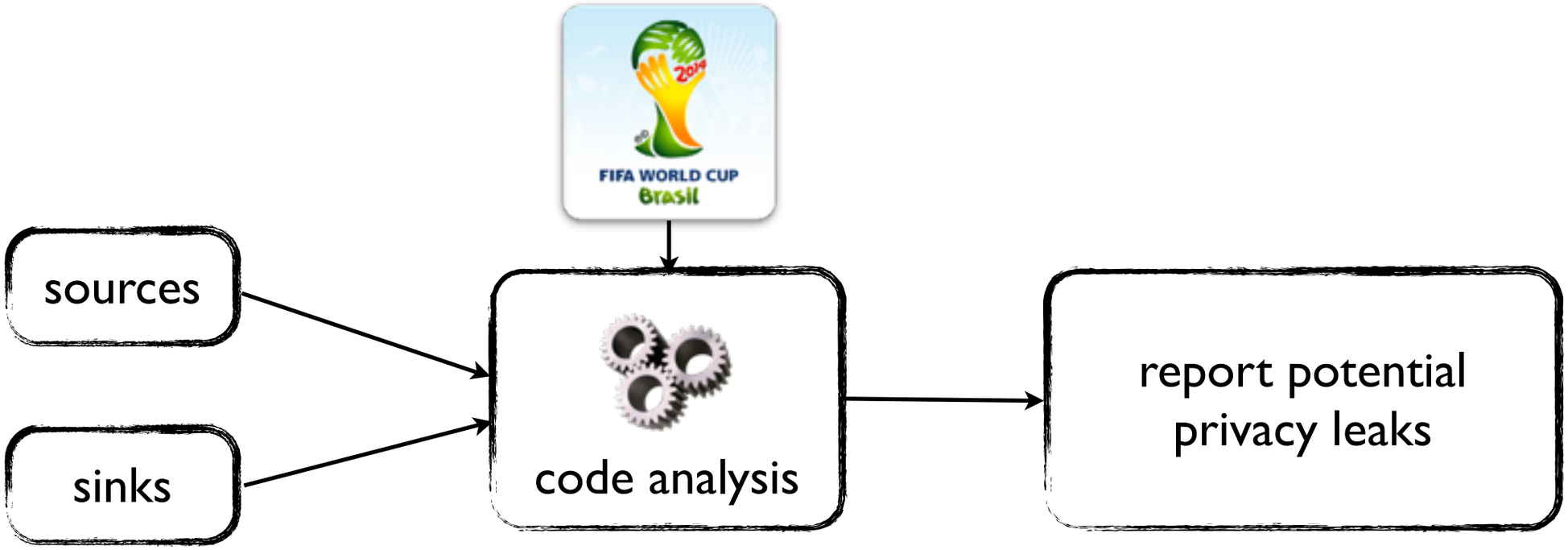
What's that app doing with my data?

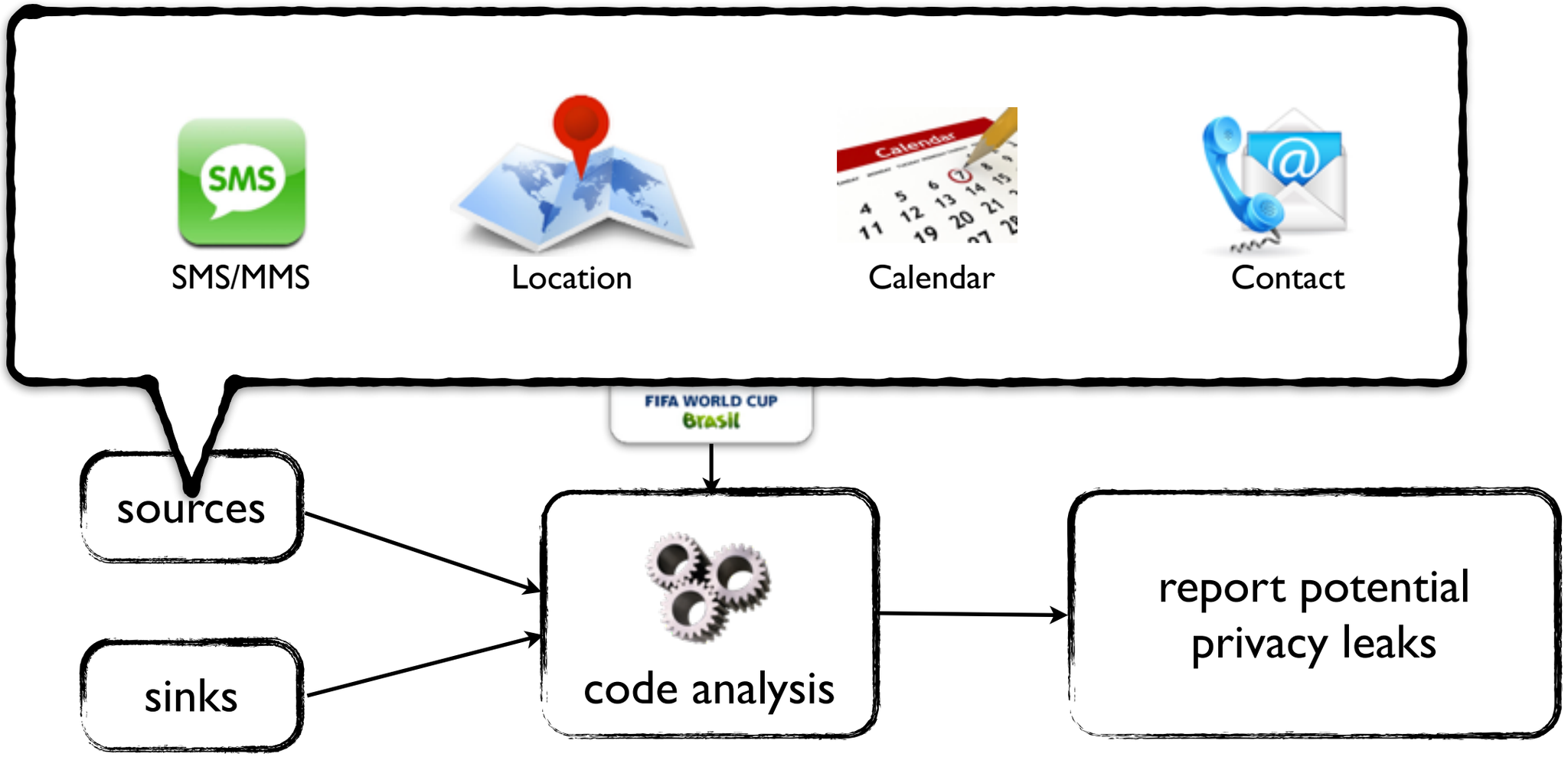
Challenges and solutions to practical taint analysis

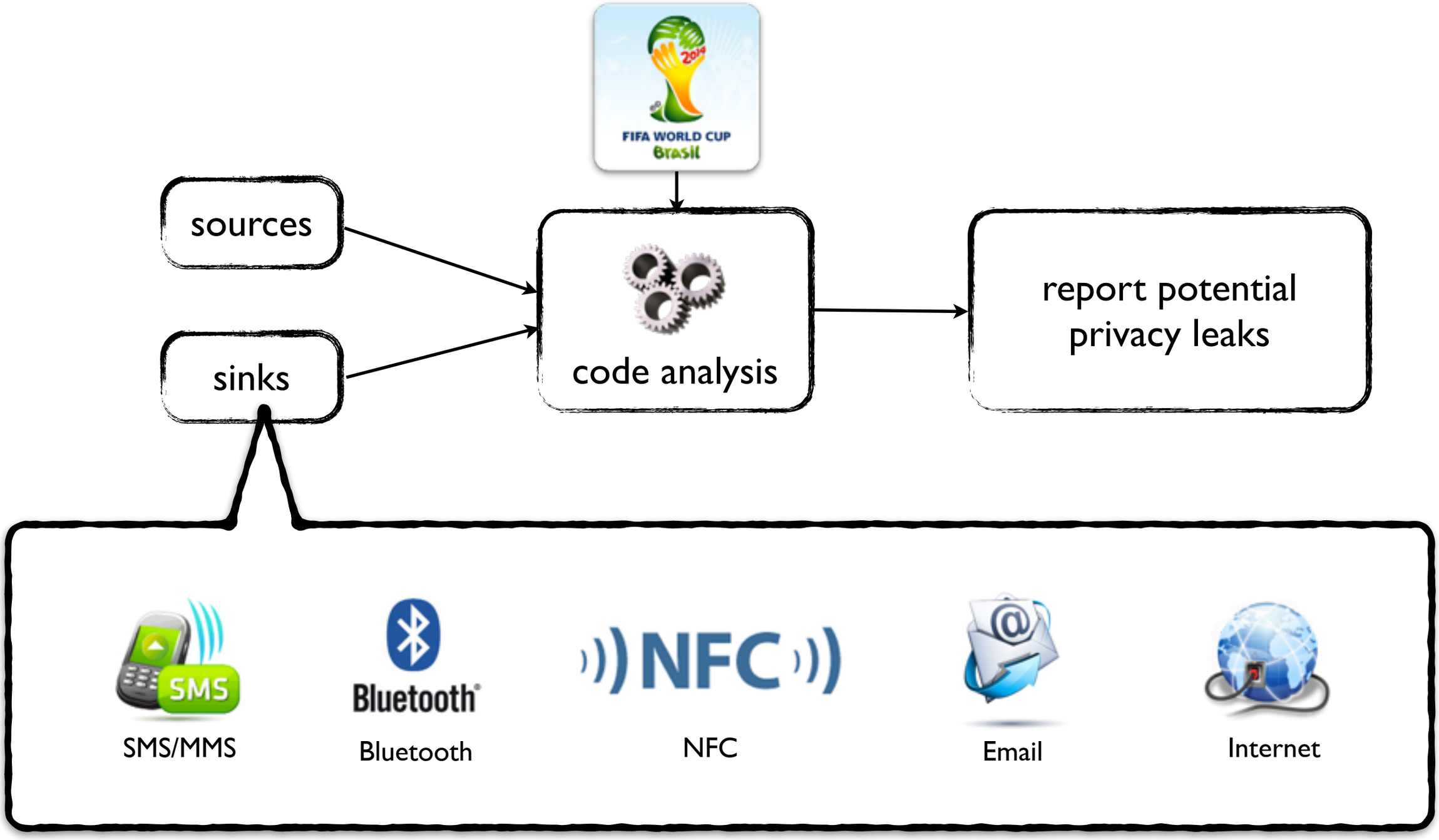
Fraunhofer SIT

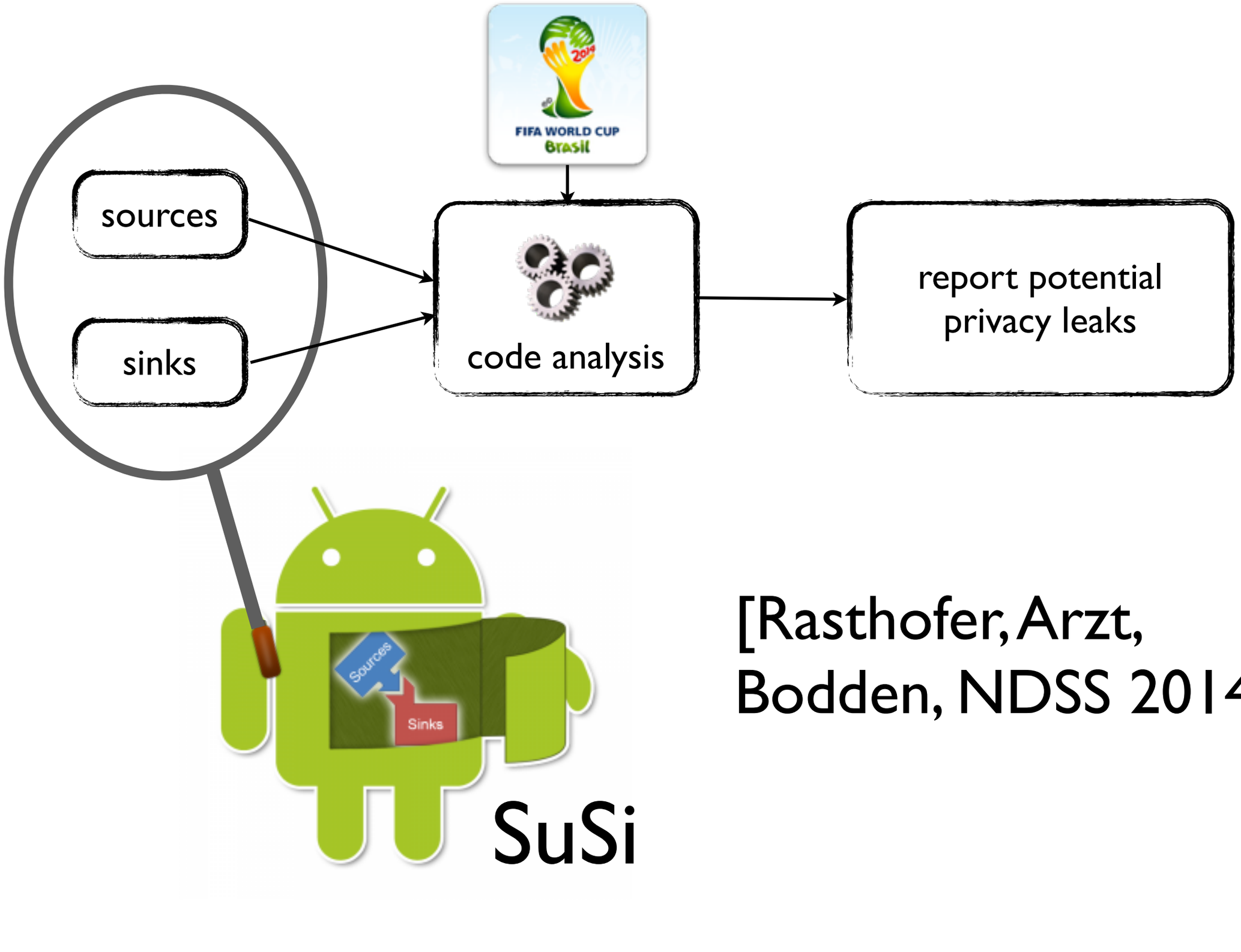
TECHNISCHE UNIVERSITÄT DARMSTADT

Eric Bodden


FIFA WORLD CUP
Brasil
sources
sinks
code analysis
report potential
privacy leaks


SMS/MMS
Location
Calendar
Contact
FIFA WORLD CUP
Brasil
sources
sinks
code analysis
report potential
privacy leaks


sources
sinks
code analysis
report potential
privacy leaks
SMS/MMS
Bluetooth
NFC
Email
Internet



SuSi

[Rasthofer, Arzt, Bodden, NDSS 2014]

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

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

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

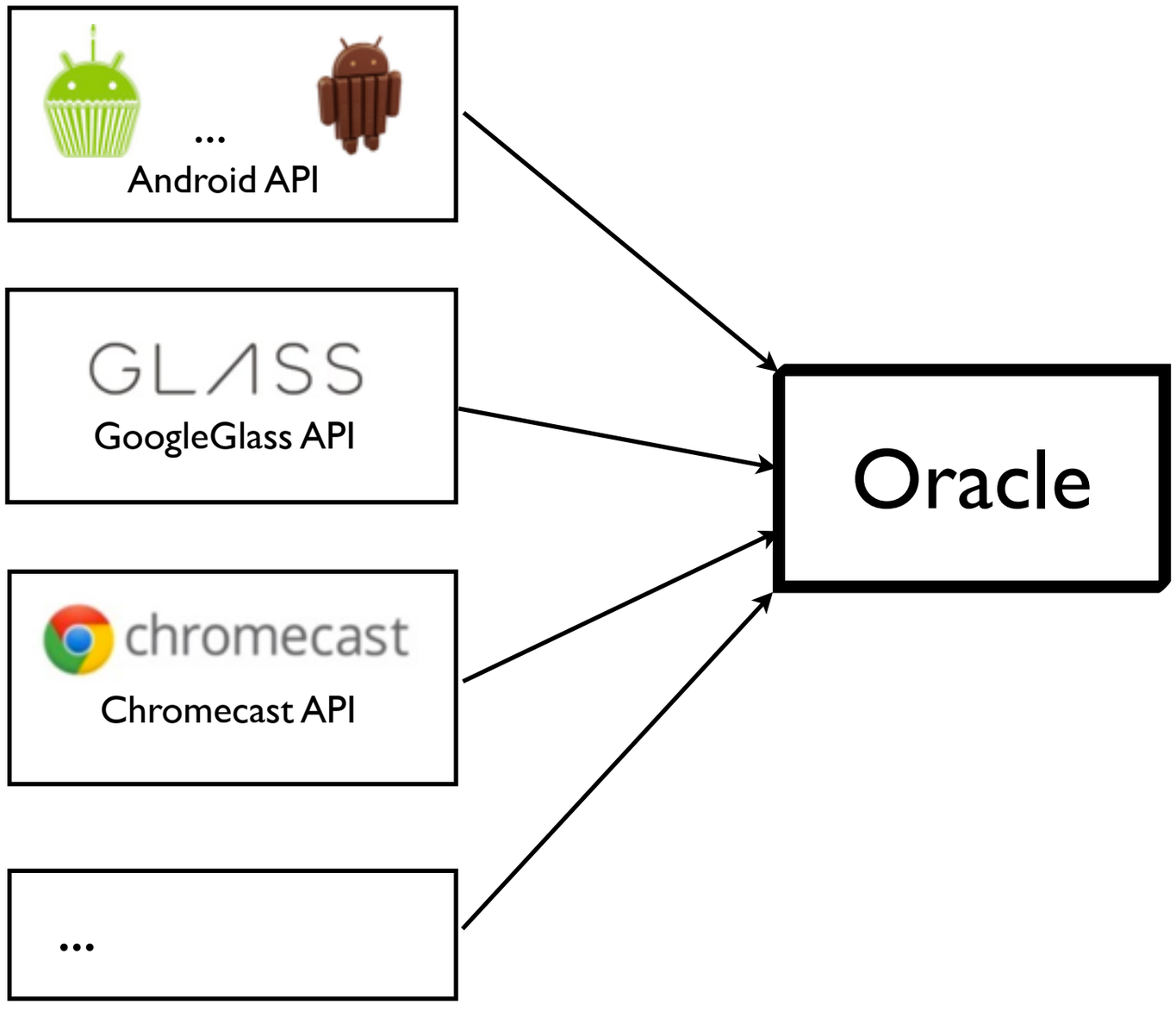
?

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

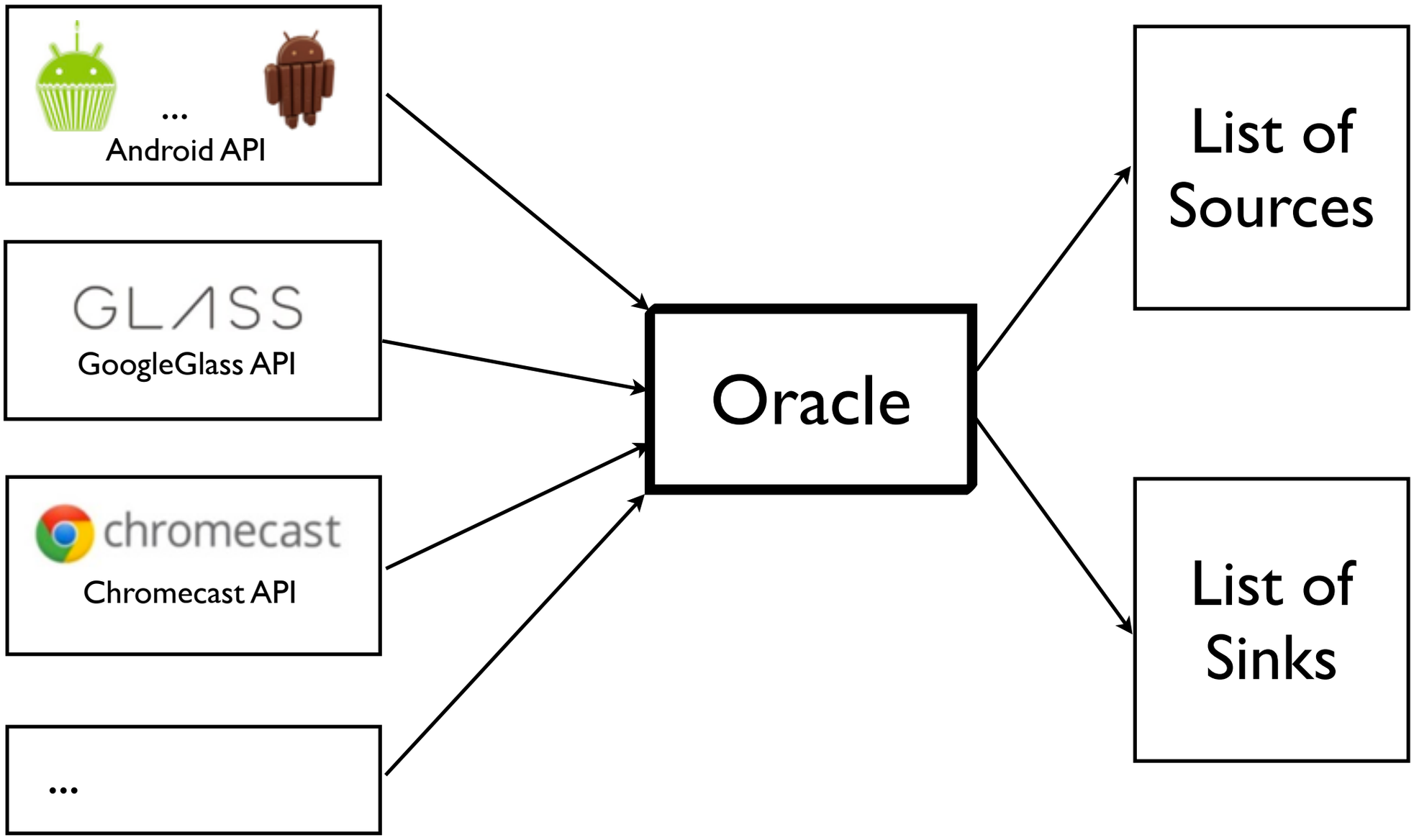
Extracting Sources/Sinks

Oracle

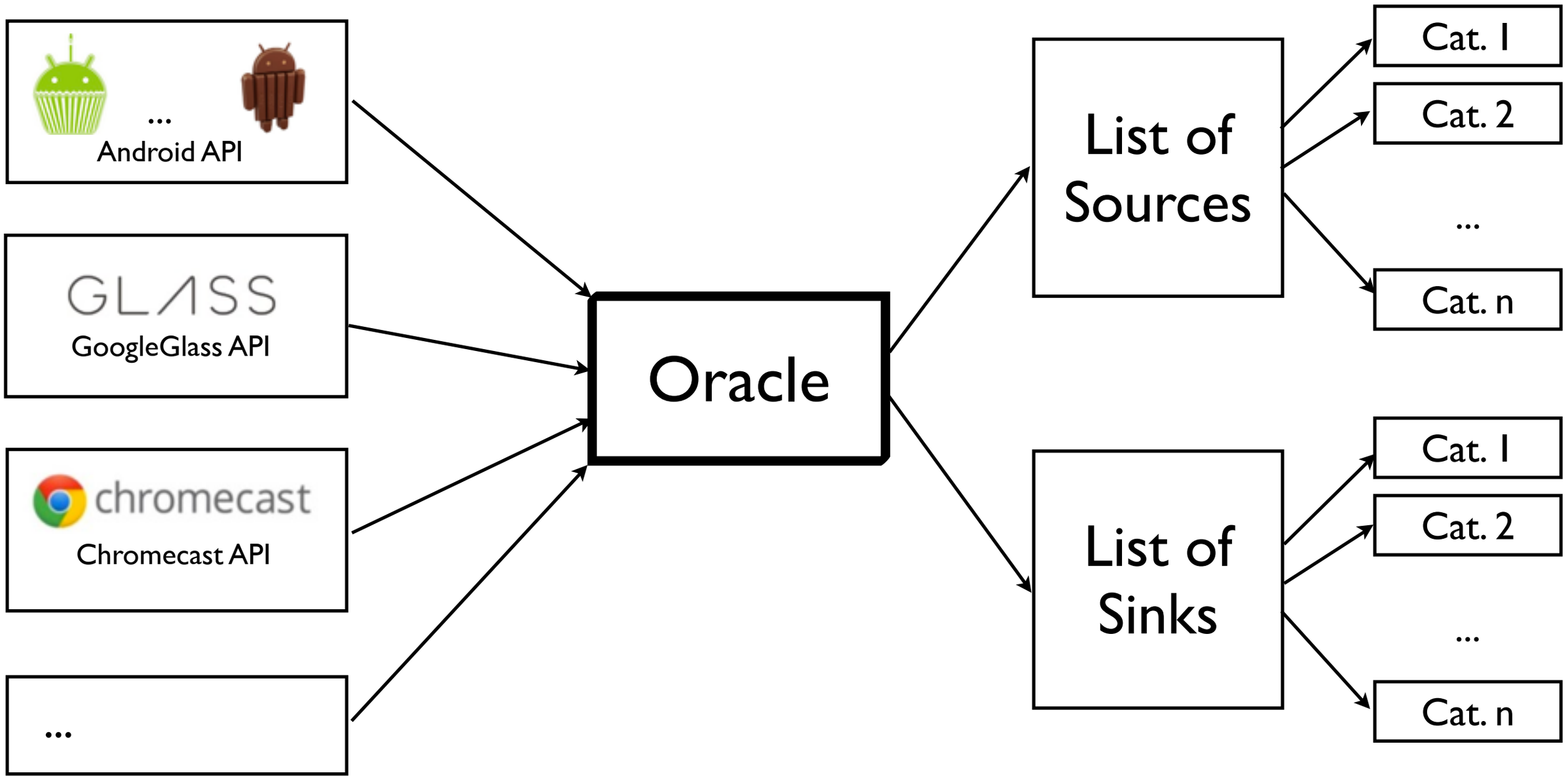
Extracting Sources/Sinks



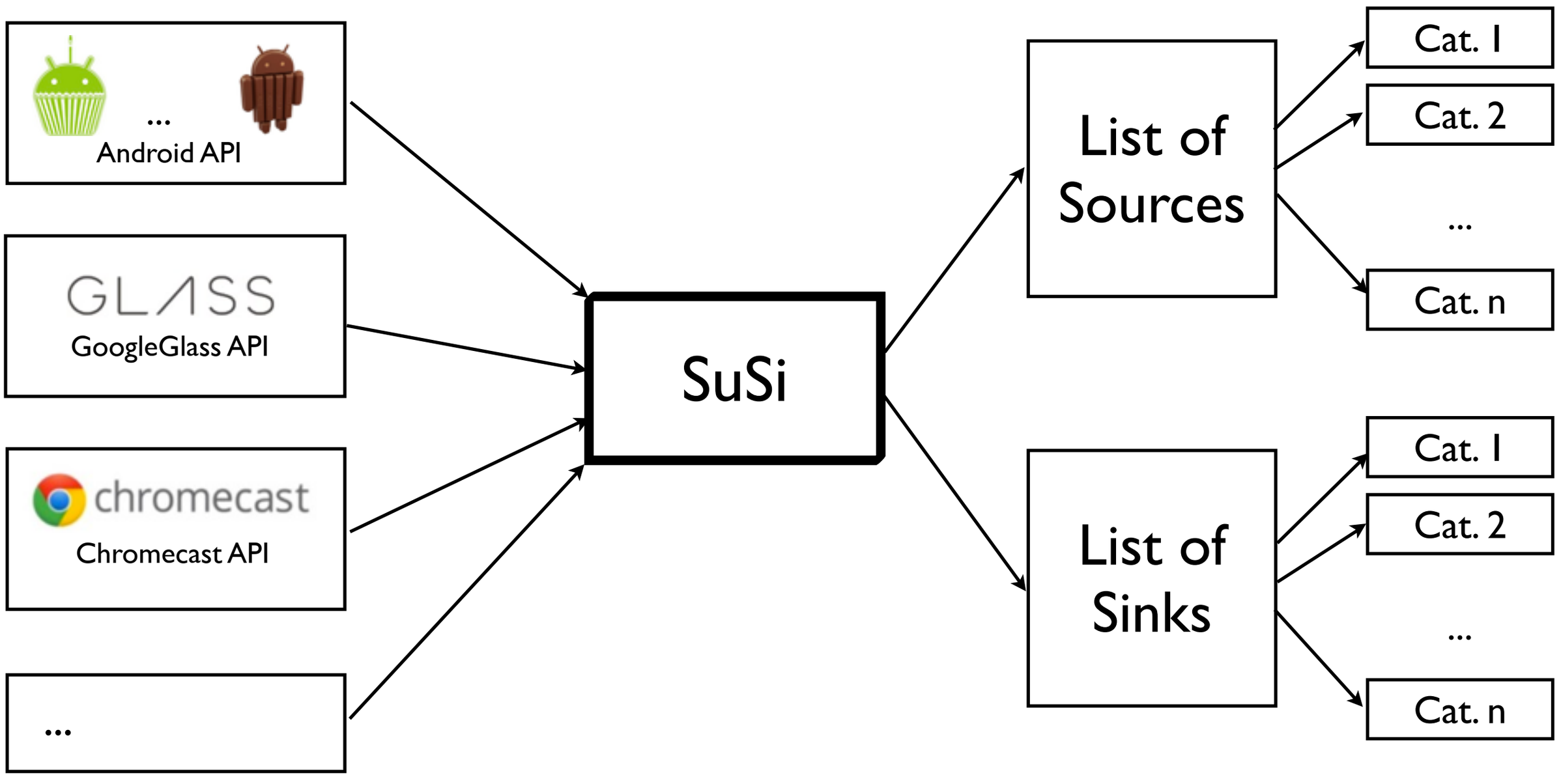
Extracting Sources/Sinks



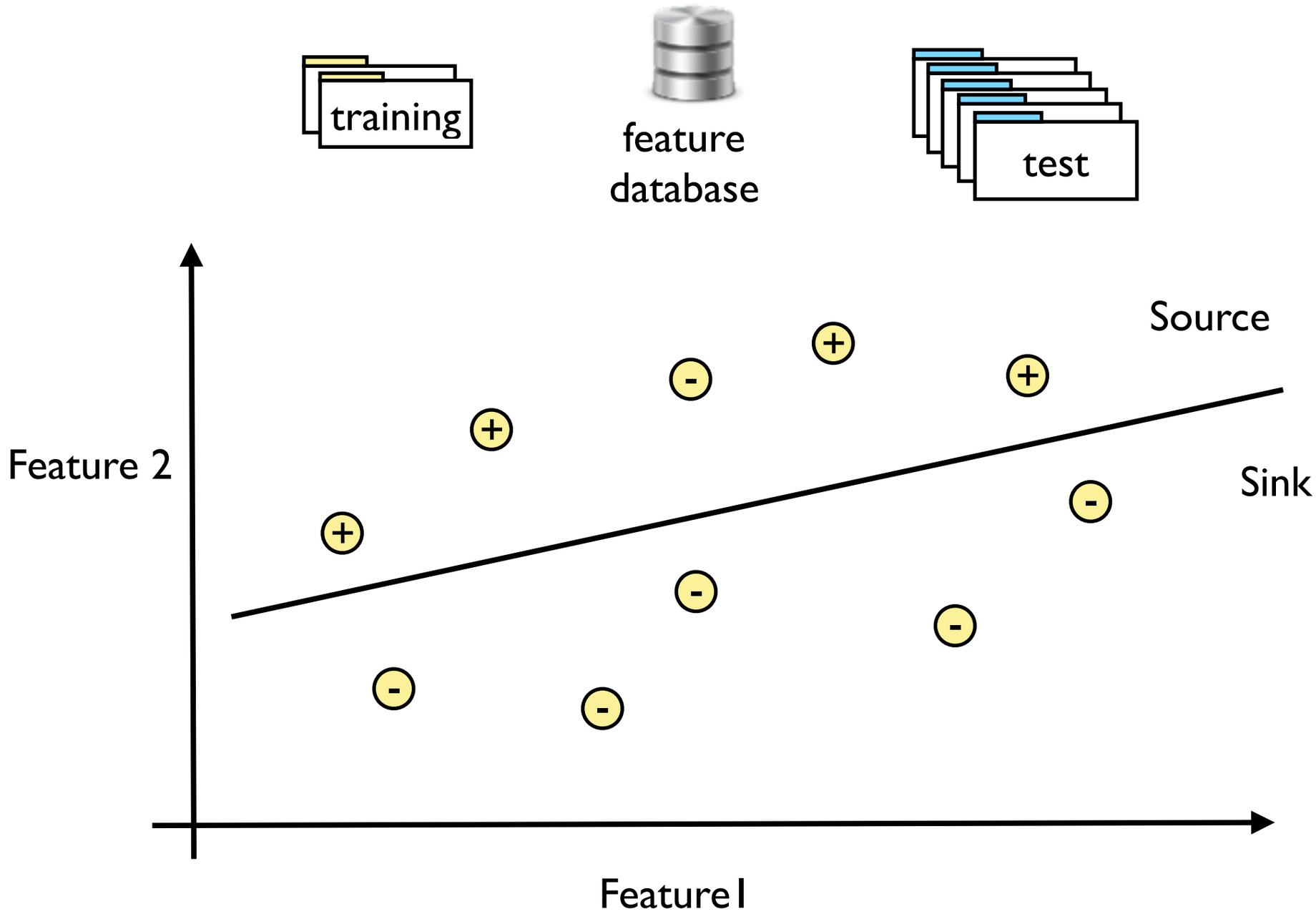
Extracting Sources/Sinks



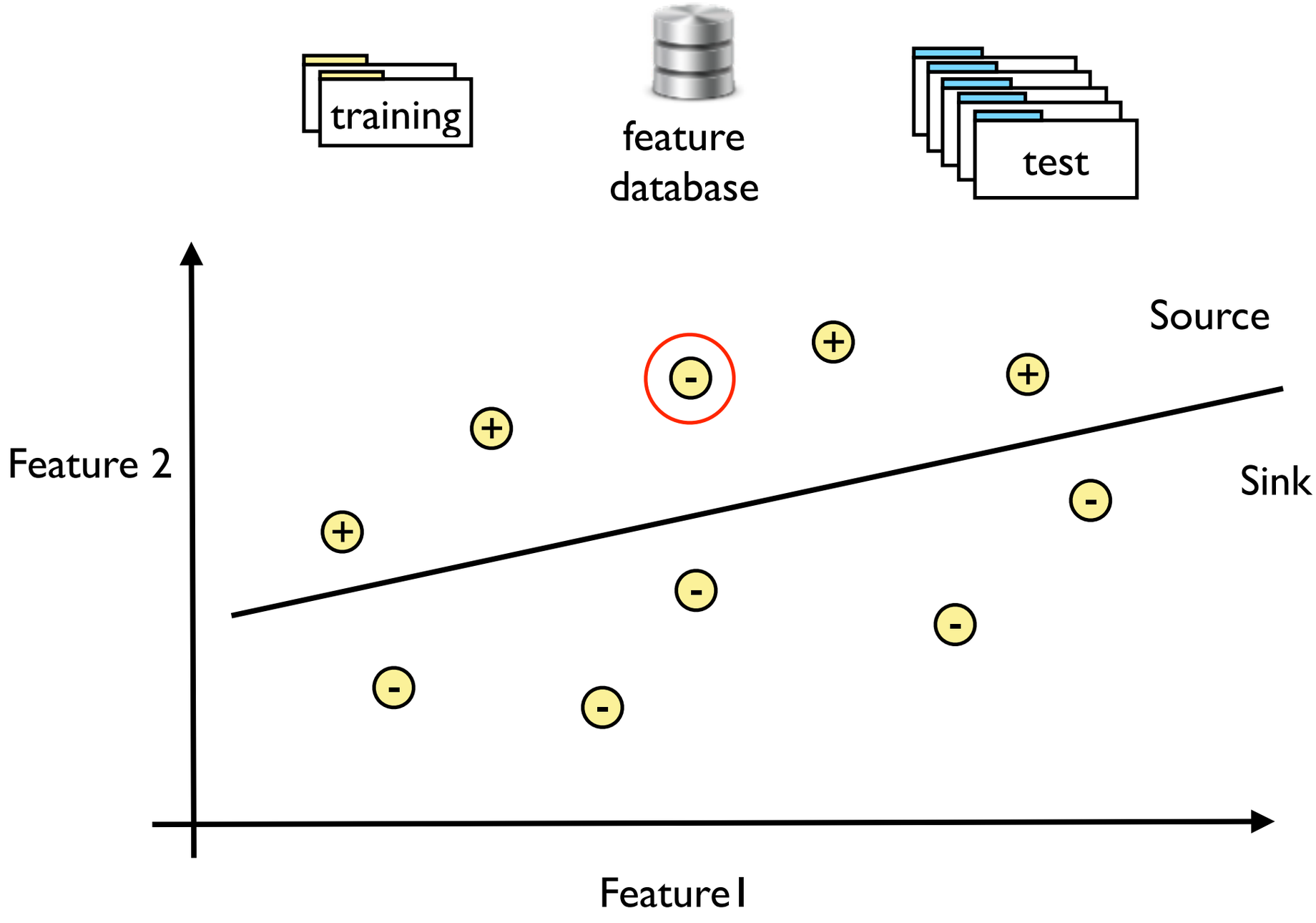
Extracting Sources/Sinks



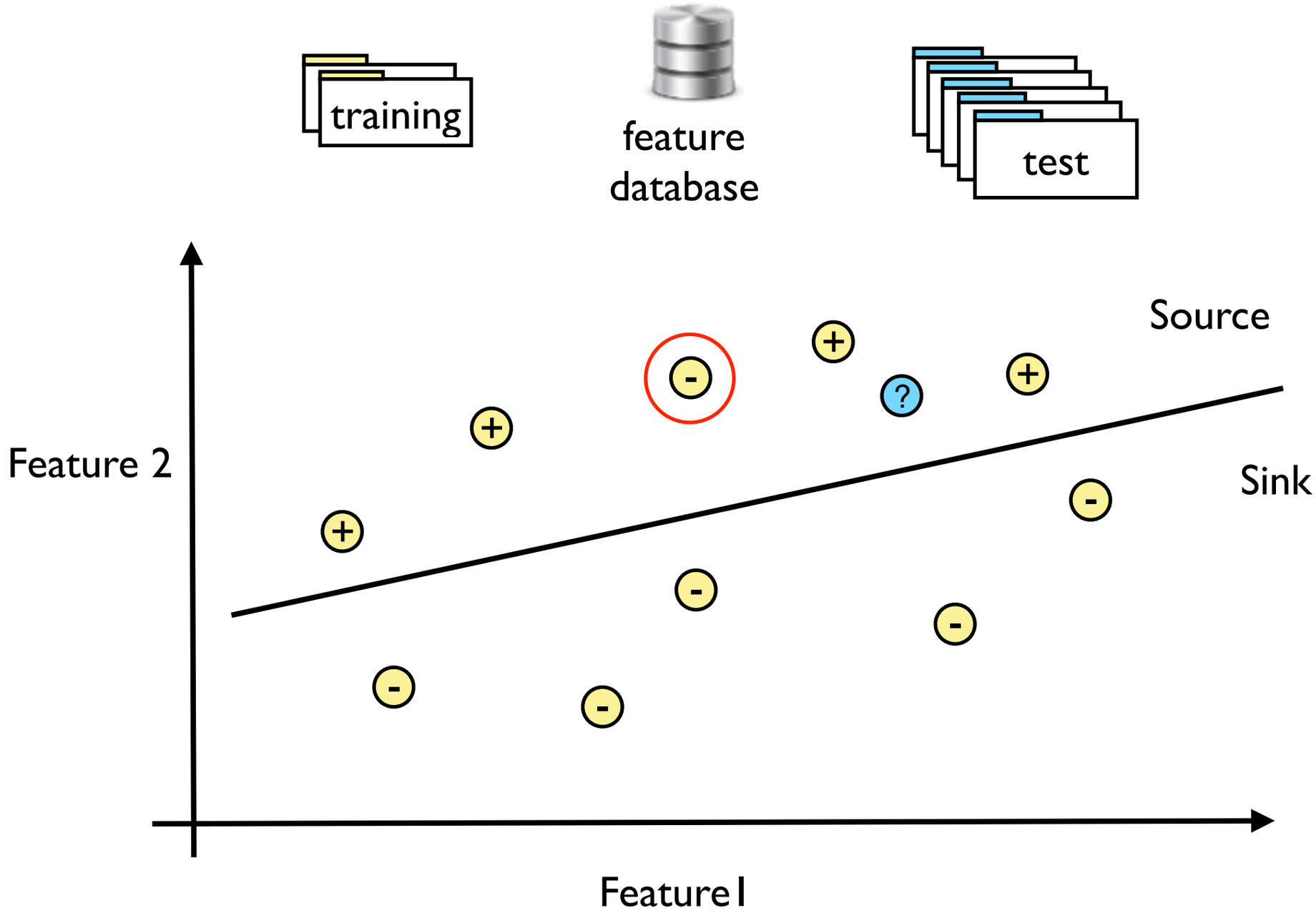
Support-Vector Machine



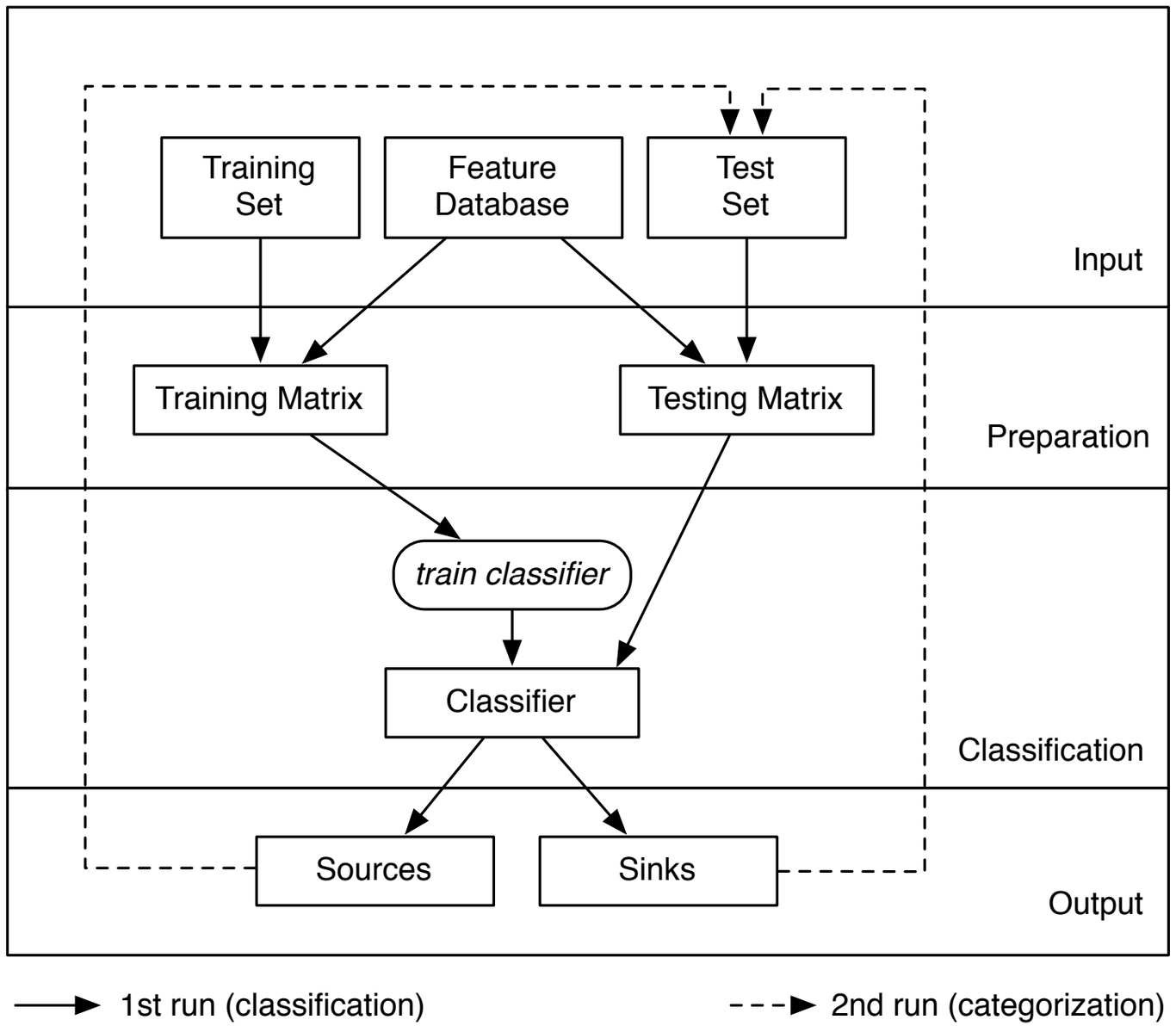
Support-Vector Machine



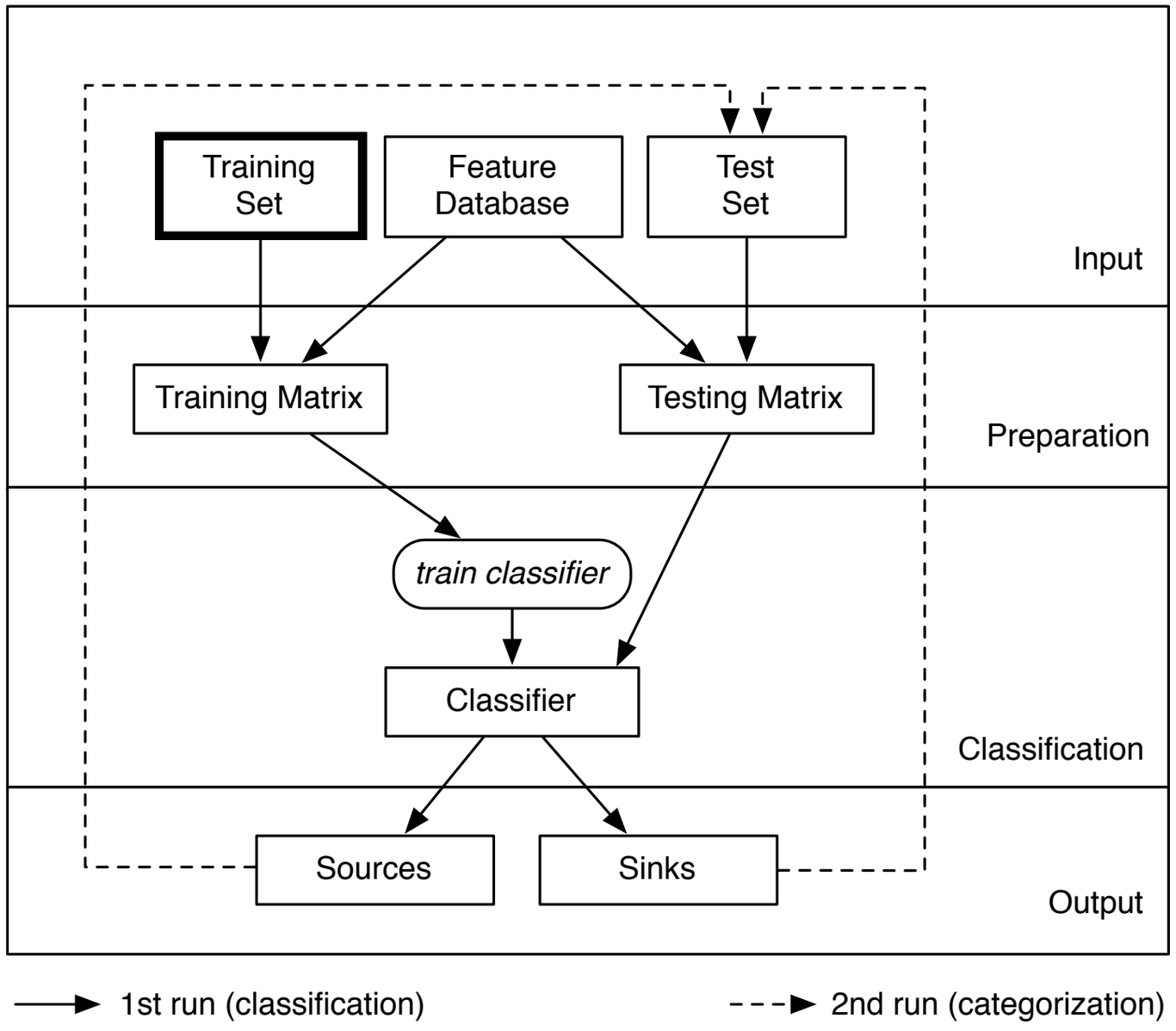
Support-Vector Machine



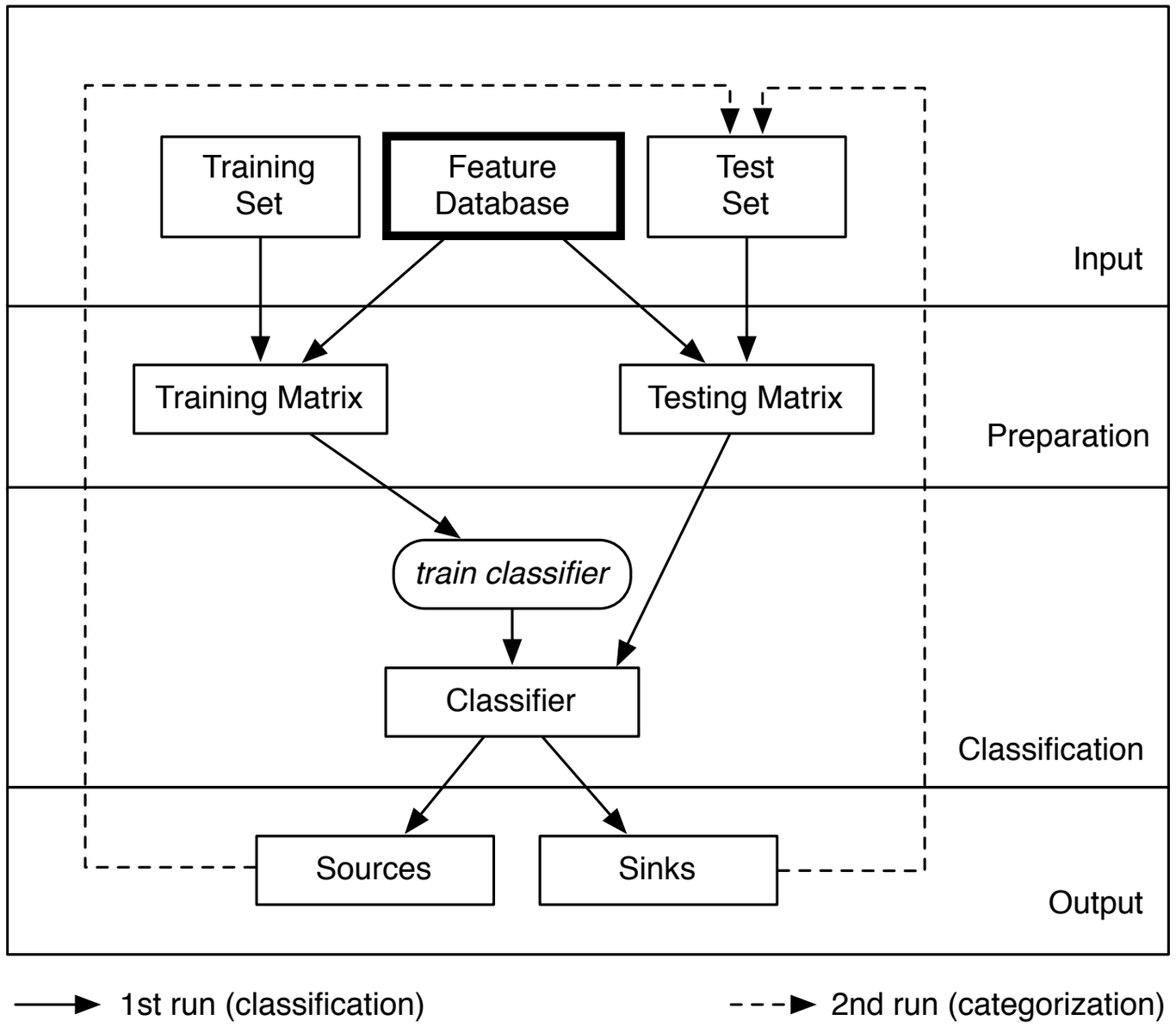
Machine-Learning Approach



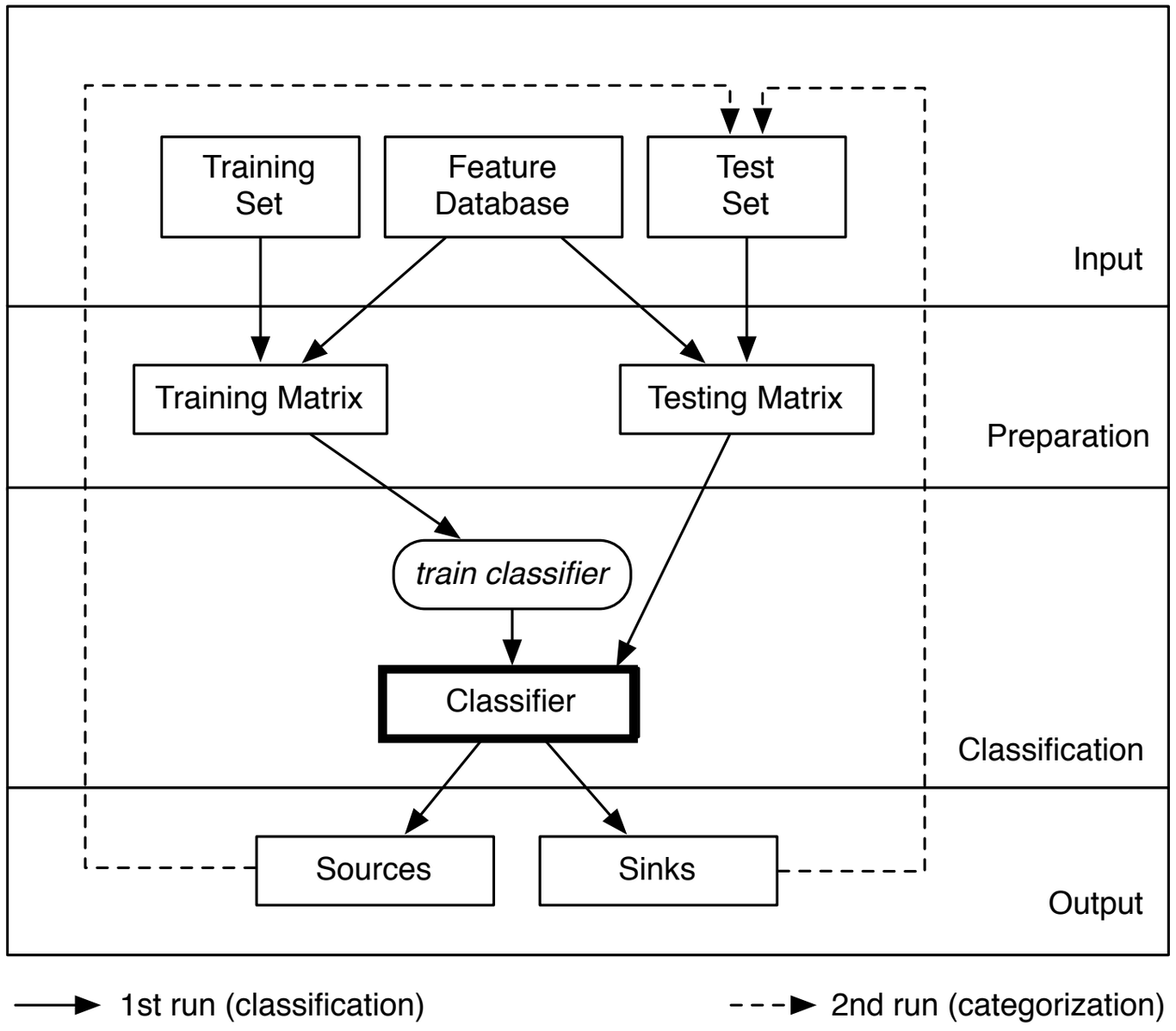
Machine-Learning Approach



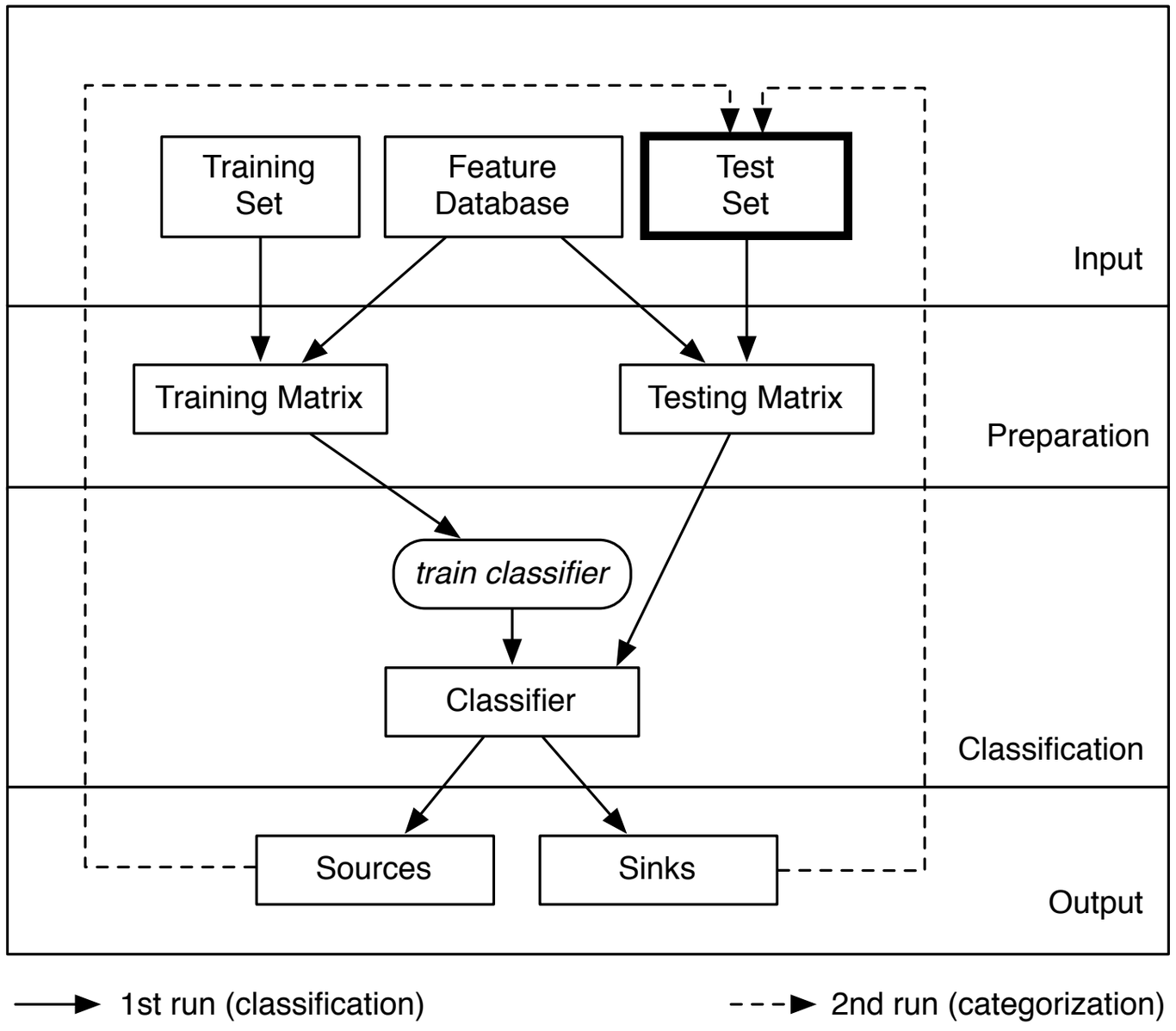
Machine-Learning Approach



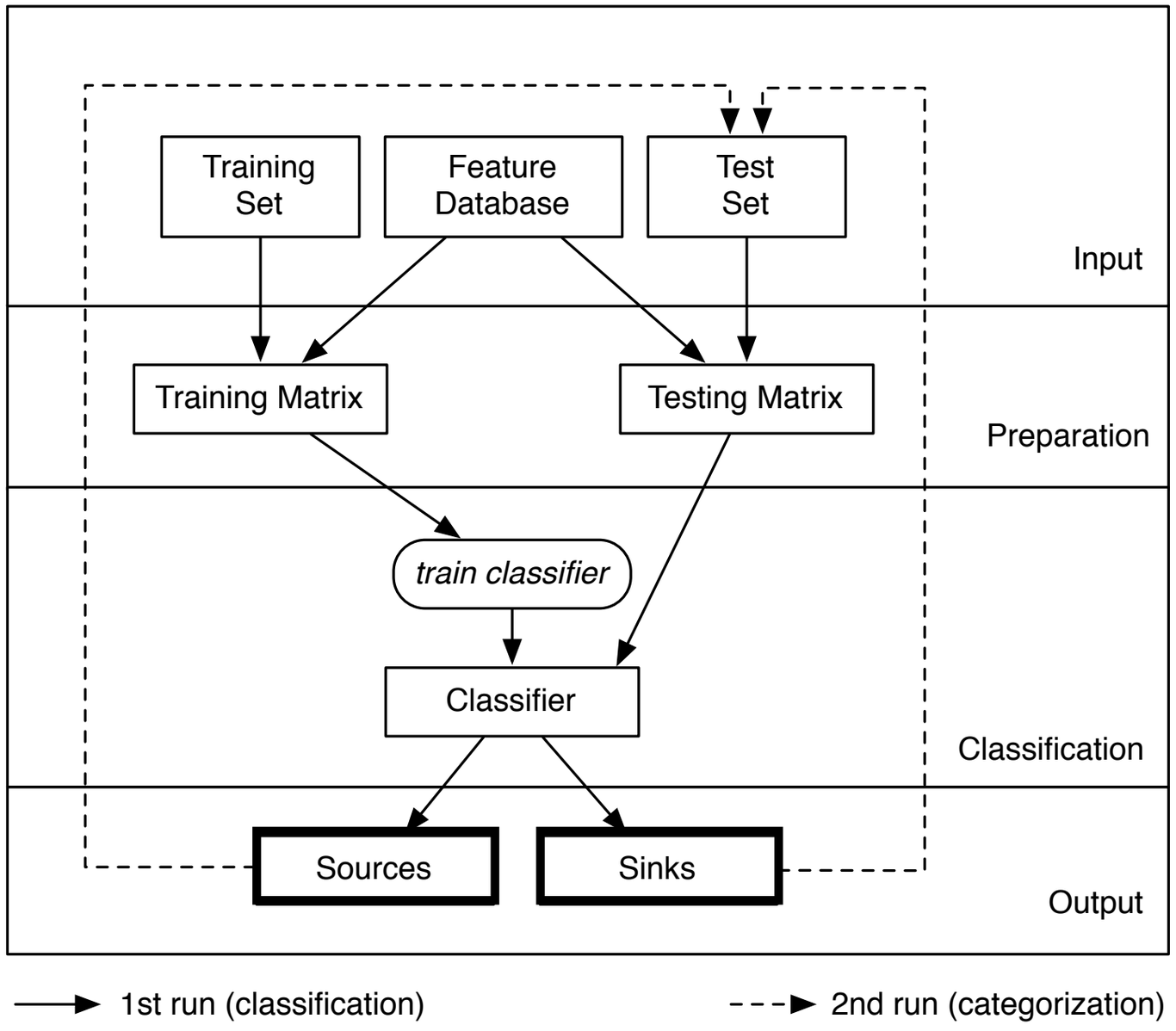
Machine-Learning Approach



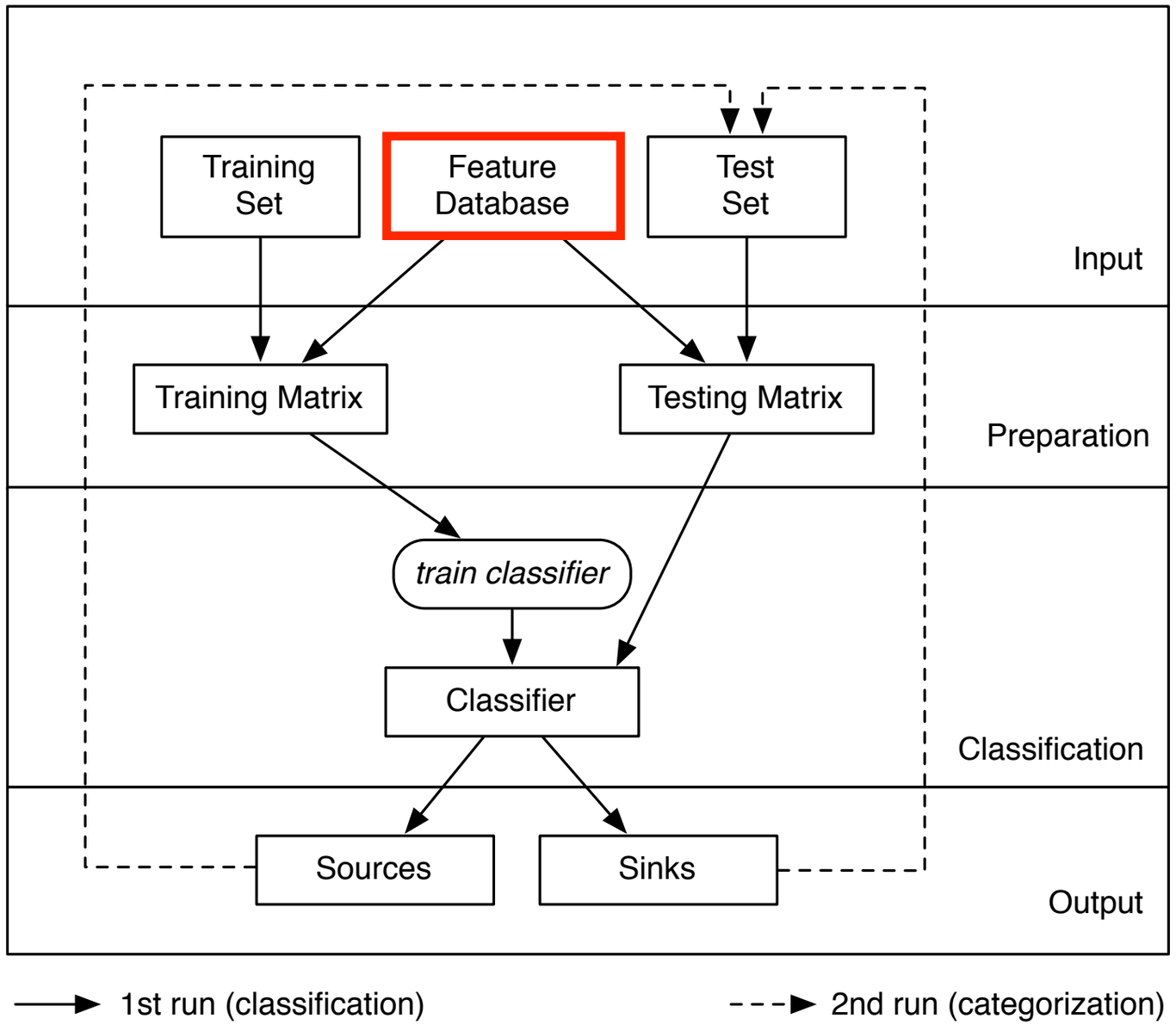
Machine-Learning Approach



Machine-Learning Approach



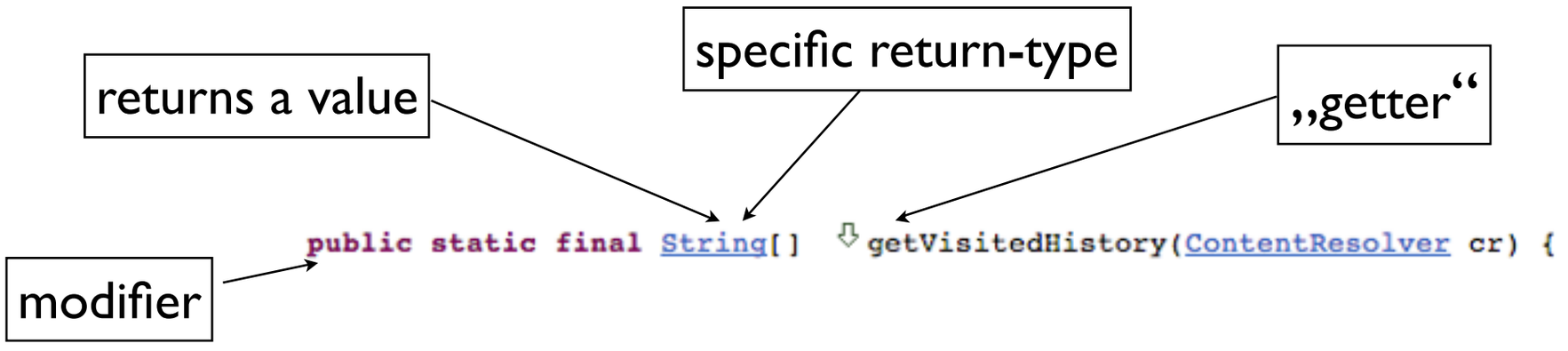
Machine-Learning Approach



Feature-Database: Classification

```
public static final String[] ⇩getVisitedHistory(ContentResolver cr) {
}
```

Feature-Database: Classification



Feature-Database: Classification

returns a value

specific return-type

„getter“

modifier

```
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;
}
```

Feature-Database: Classification

returns a value

specific return-type

„getter“

modifier

```
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;
}
```

data flow 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

- Data flow to return value
- Data flow from parameter to (abstract) sink

Feature-Database: Categorization

SMS/MMS

Location

Calendar

Contact

...

SMS/MMS

Bluetooth

NFC

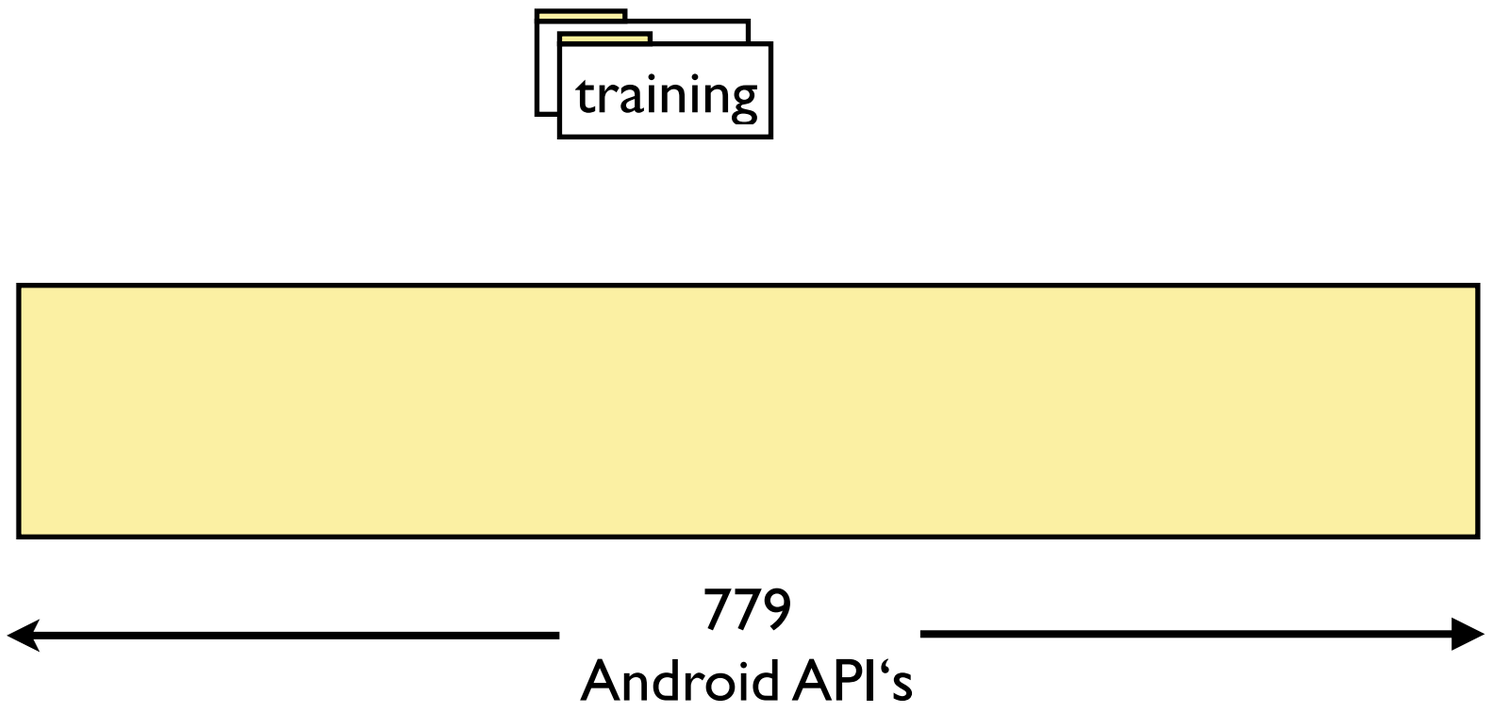
Email

Internet

...

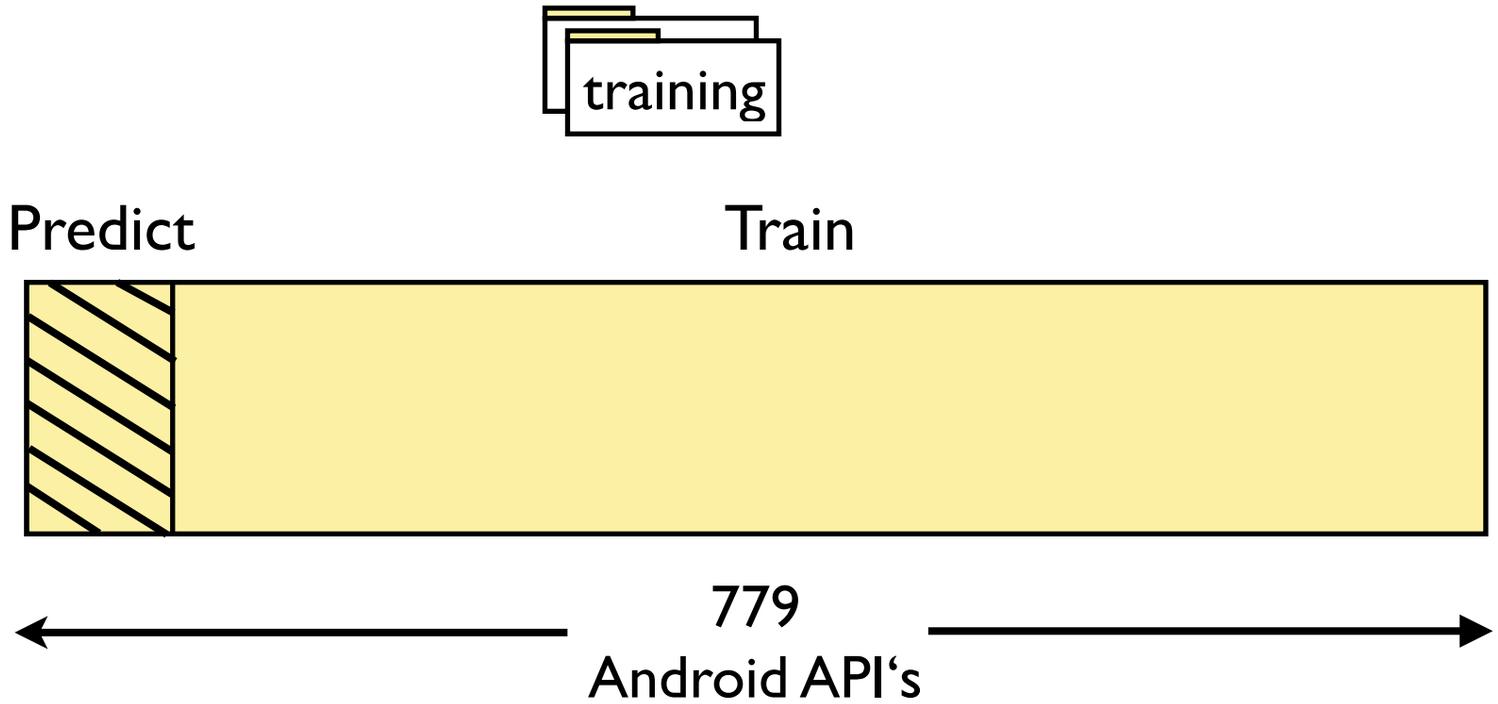
Evaluation

Ten-fold cross validation:



Evaluation

Ten-fold cross validation:



Evaluation

Ten-fold cross validation:

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

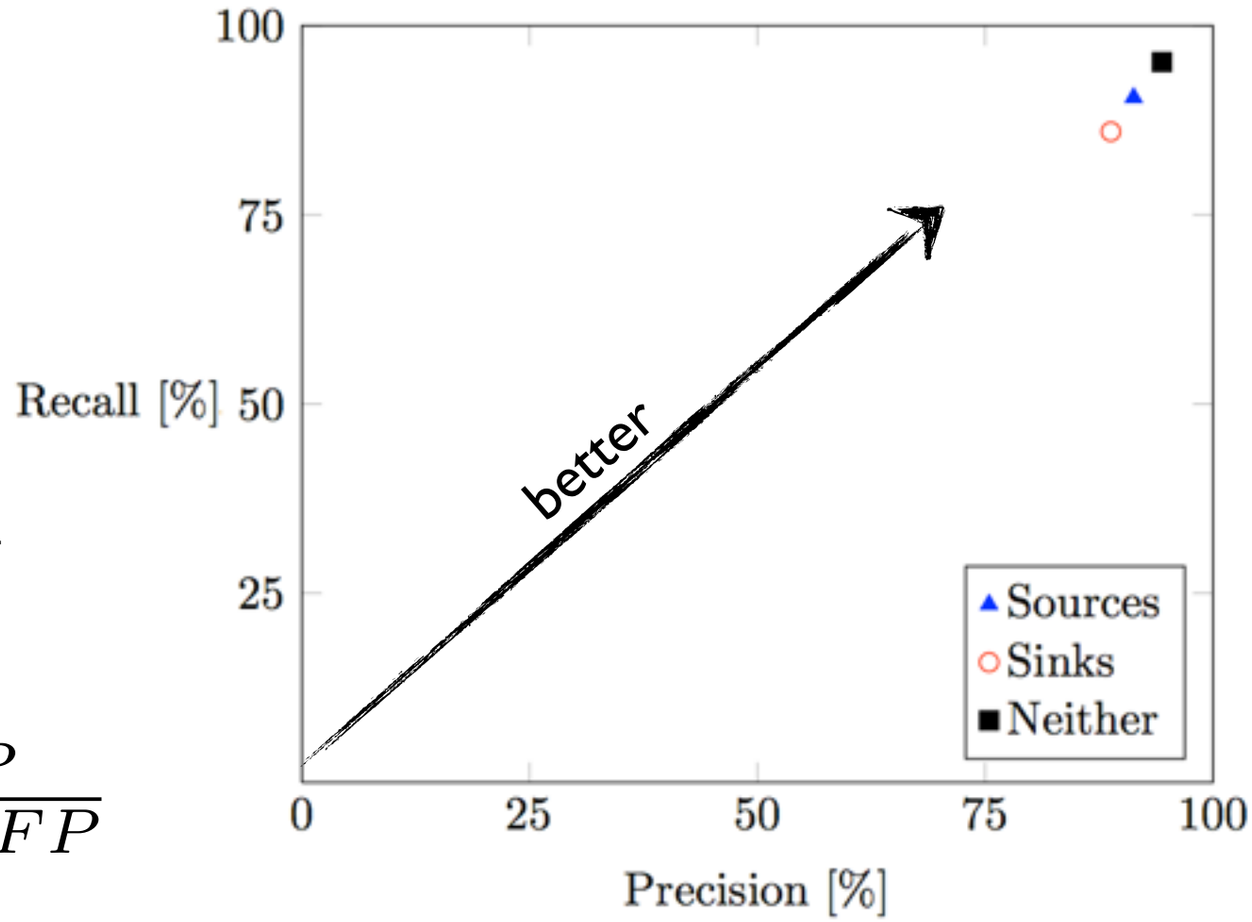
$$Precision = \frac{TP}{TP+FP}$$

Evaluation

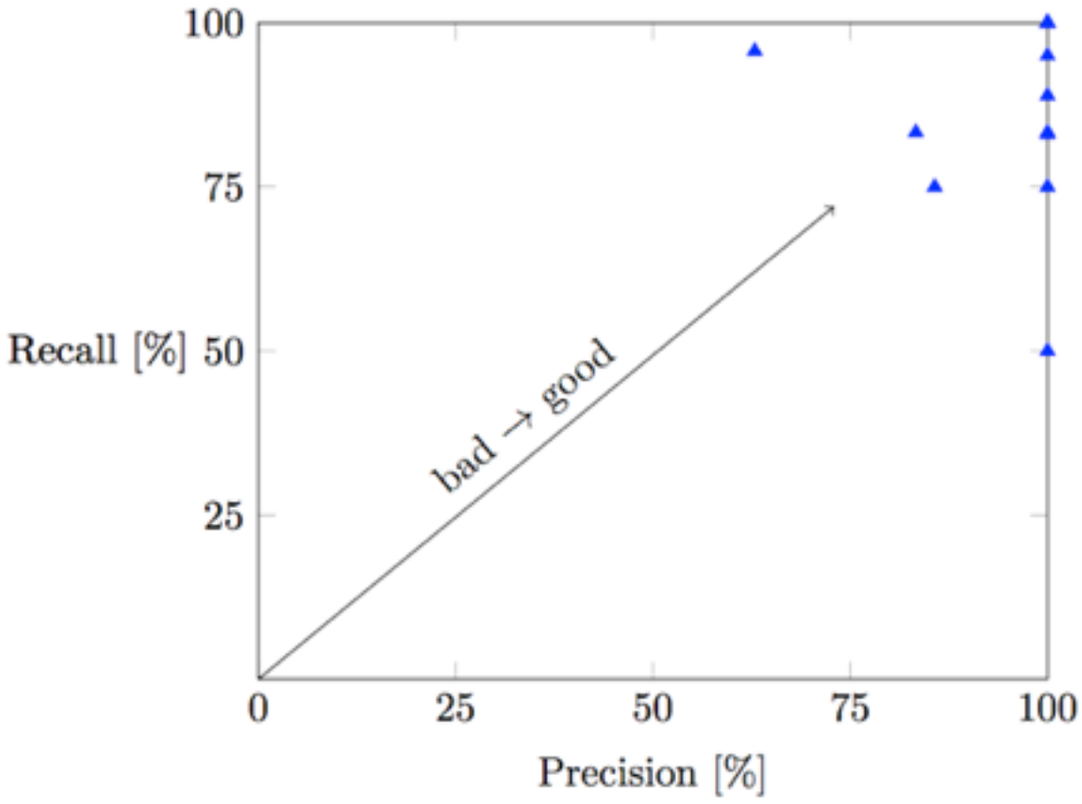
Ten-fold cross validation:

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

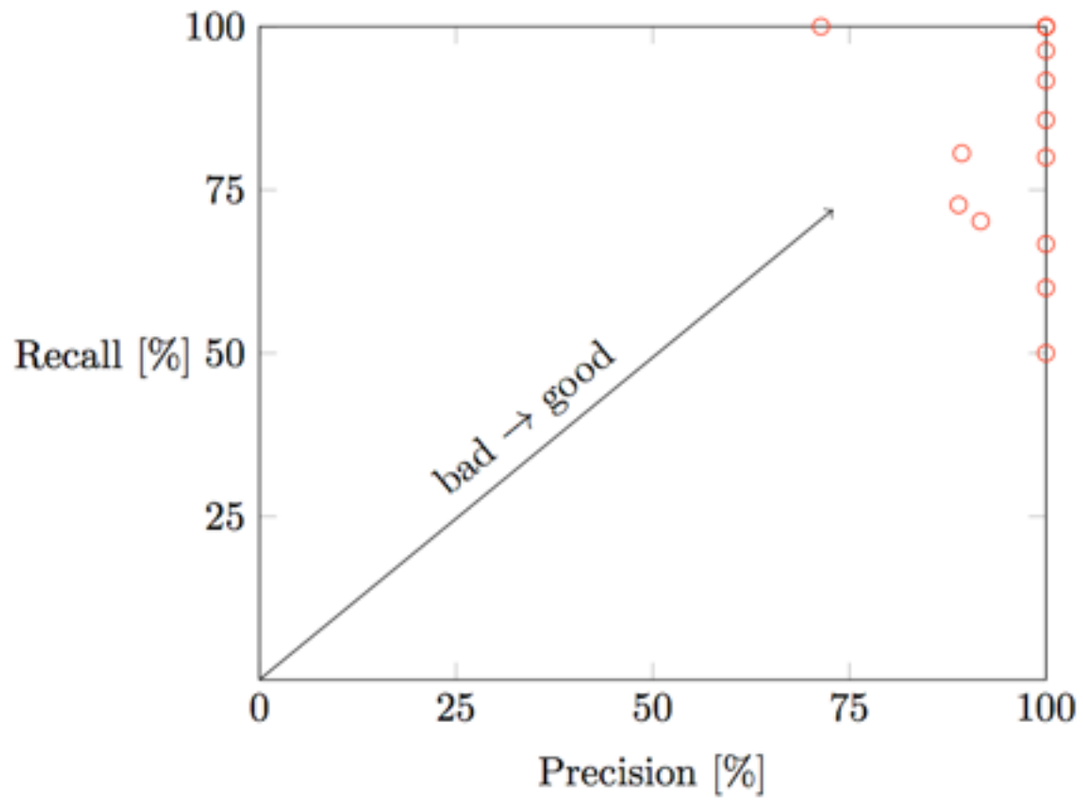
$$Precision = \frac{TP}{TP+FP}$$



Results for categorization

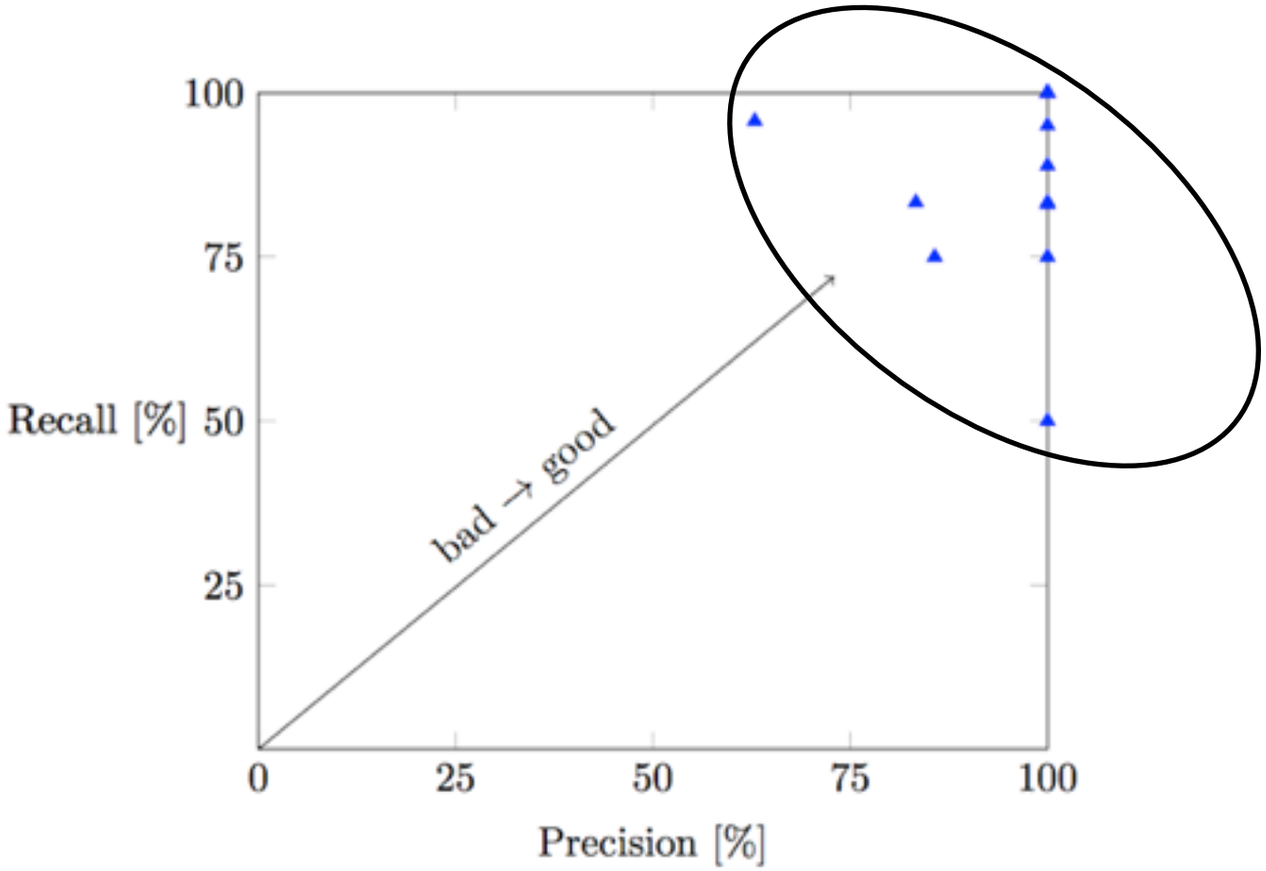


Categorized Sources

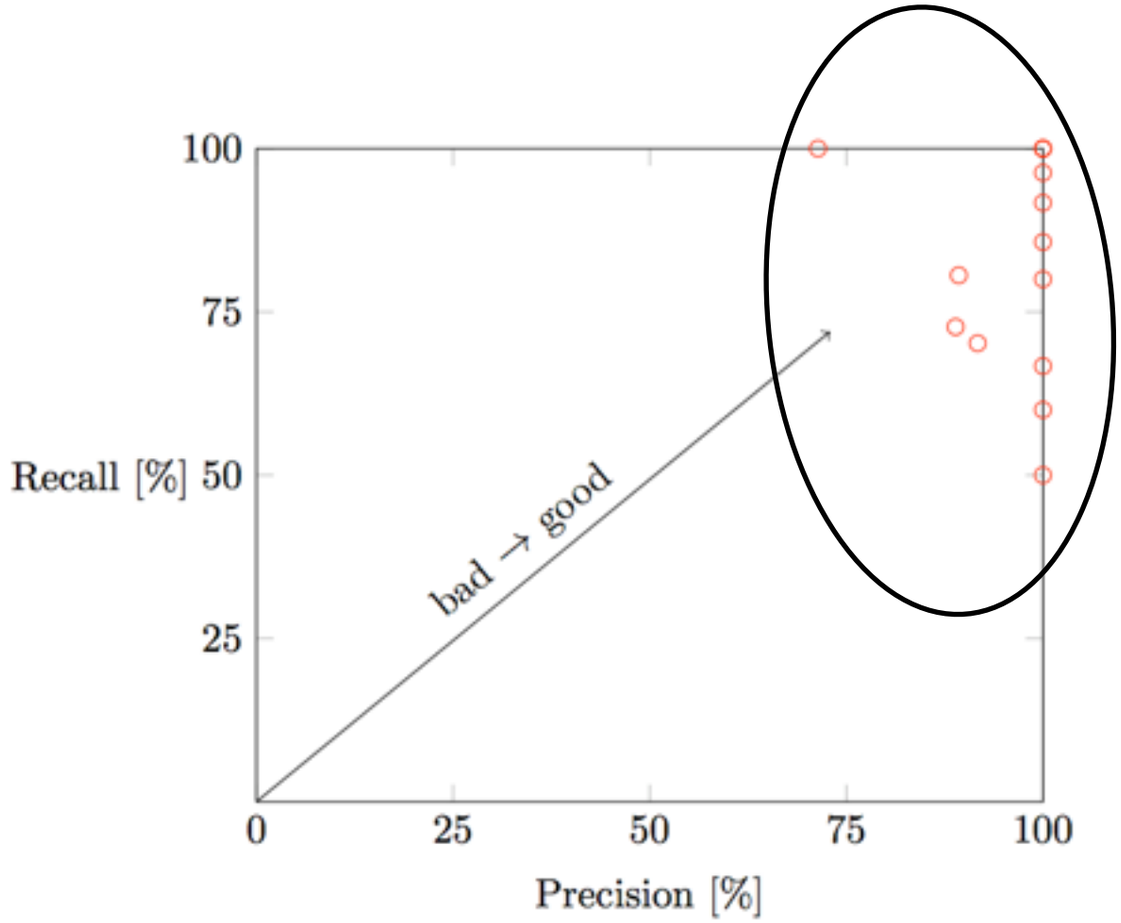


Categorized Sinks

Results for categorization

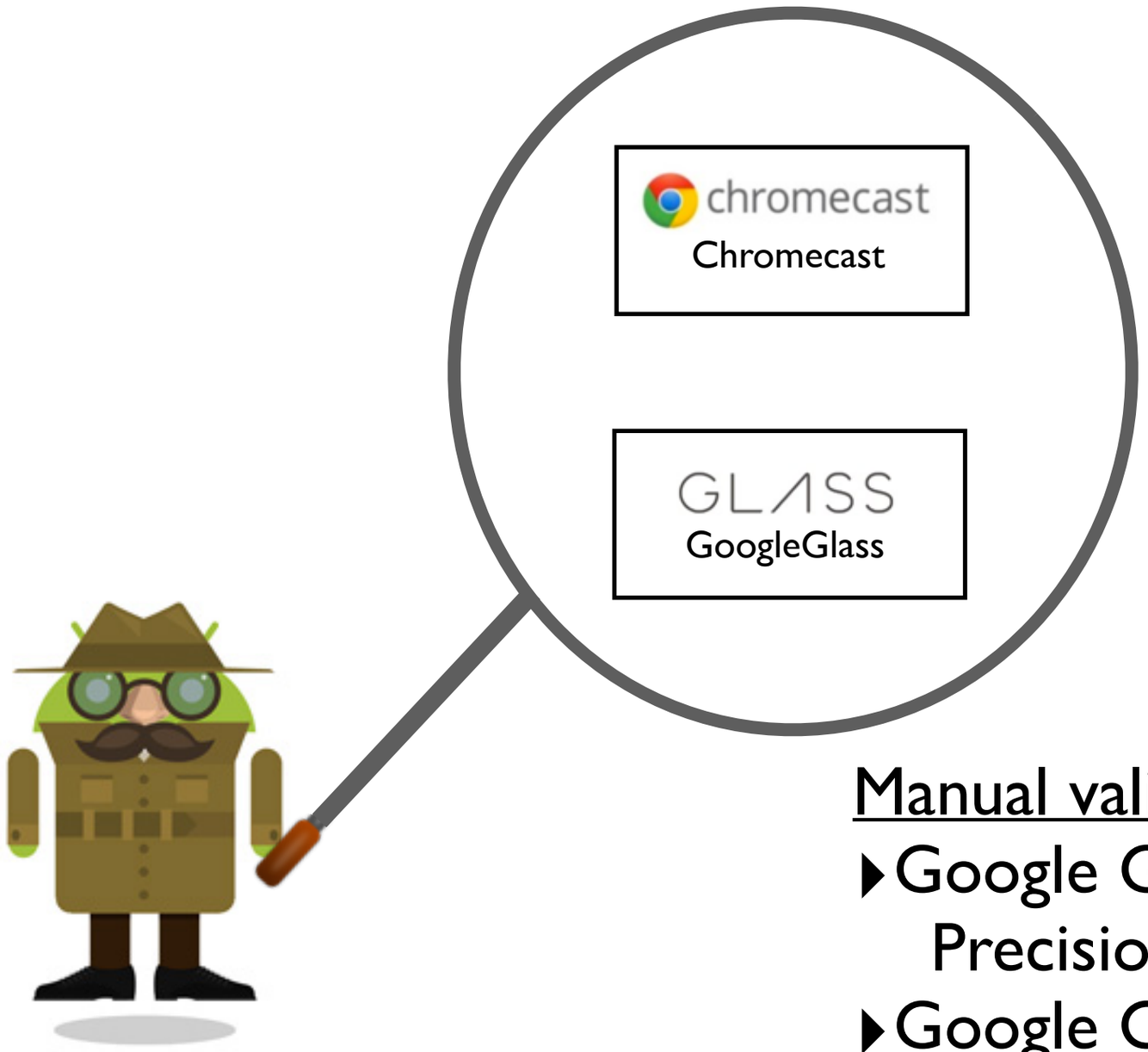


Categorized Sources



Categorized Sinks

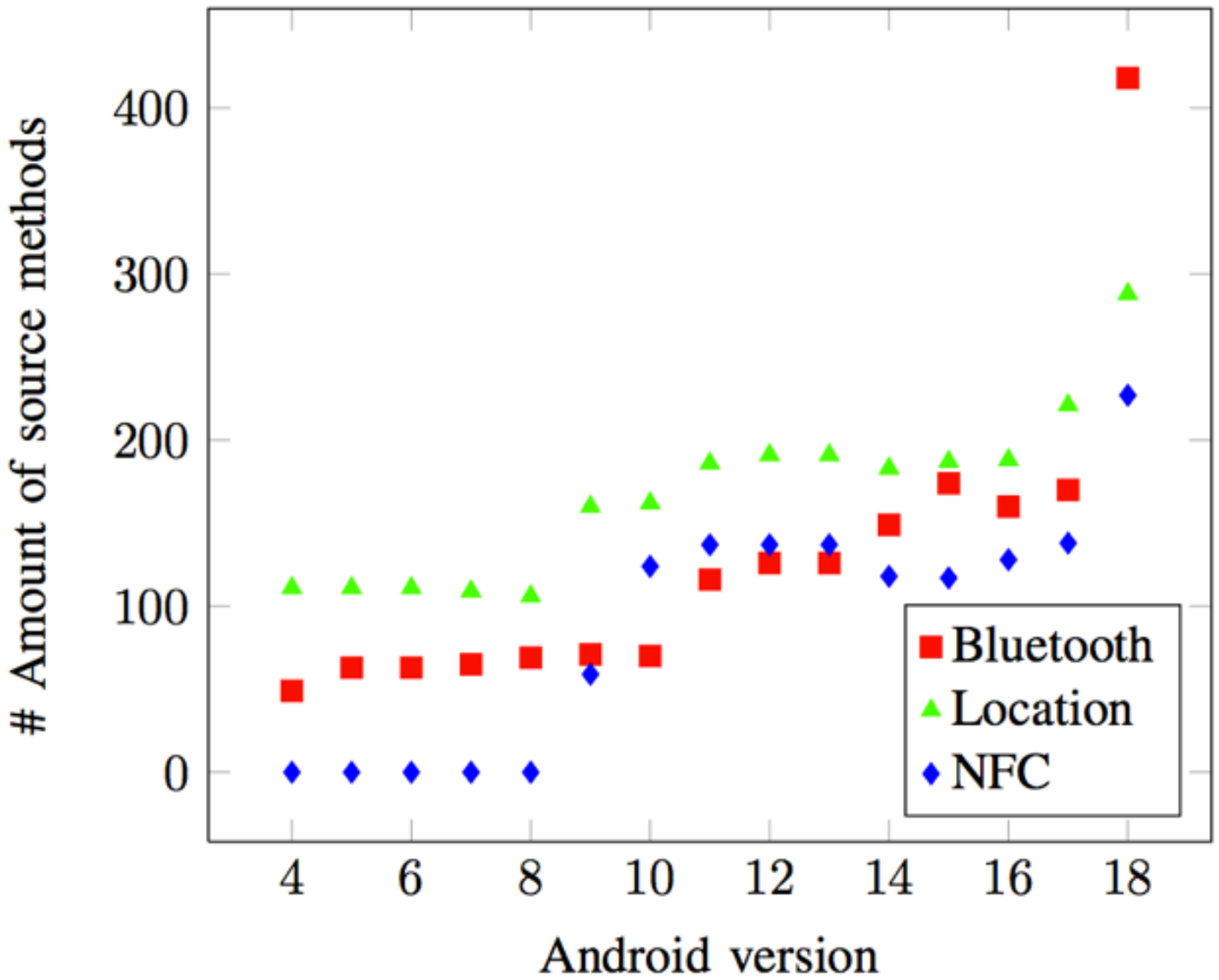
On untrained APIs



<u>Manual validation:</u>

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

Evolution

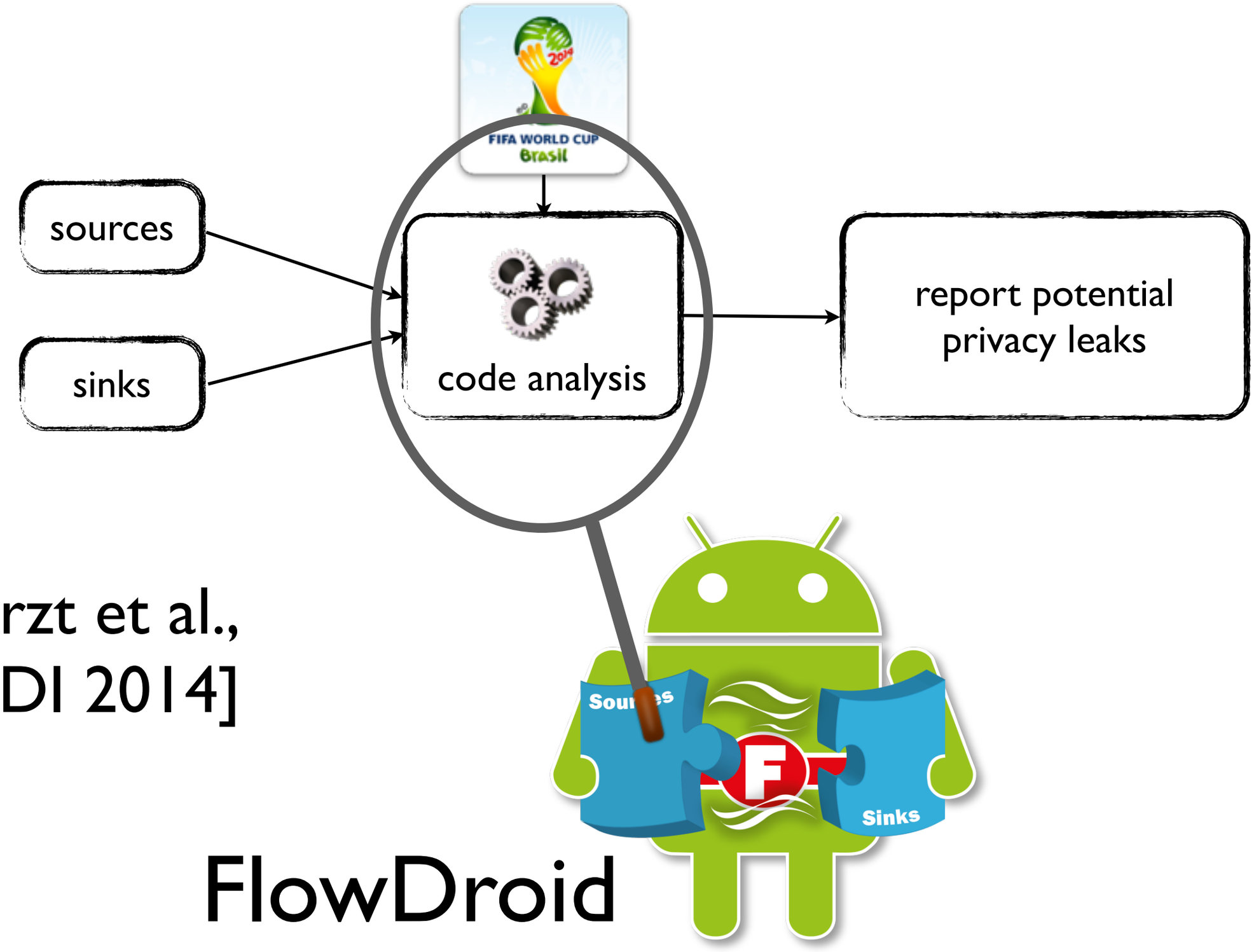

400
300
200
100
0
Amount of source methods
4
6
8
10
12
14
16
18
Android version
Bluetooth
Location
NFC

Top Source/Sink Methods in Android-Malware

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()	✓	✗	✗



[Arzt et al.,
PLDI 2014]

FlowDroid

joint work with…

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Steven Arzt, Siegfried Rasthofer,
Christian Fritz

UNIVERSITE DU
LUXEMBOURG

Alexandre Bartel, Jacques Klein,
Yves Le Traon

Damien Octeau, Patrick McDaniel

joint work with…

TECHNISCHE UNIVERSITÄT DARMSTADT

Steven Arzt, Siegfried Rasthofer, Christian Fritz

uni.lu UNIVERSITÉ DU LUXEMBOURG

Google Faculty Research Award

Alexandre Bartel, Jacques Klein, Yves Le Traon

THE PENNSYLVANIA STATE UNIVERSITY · 1855 ·

Damien Octeau, Patrick McDaniel

joint work with…

TECHNISCHE UNIVERSITÄT DARMSTADT

Steven Arzt, Siegfried Rasthofer, Christian Fritz

Google Faculty Research Award

uni.lu UNIVERSITÉ DU LUXEMBOURG

Alexandre Bartel, Jacques Klein, Yves Le Traon

THE PENNSYLVANIA STATE UNIVERSITY · 1855 ·

Damien Octeau, Patrick McDaniel

FlowDroid

- Can analyze Android binaries or source
- Built on top of Soot
- Specialized to taint analysis but can be extended with other kinds of Android analyses

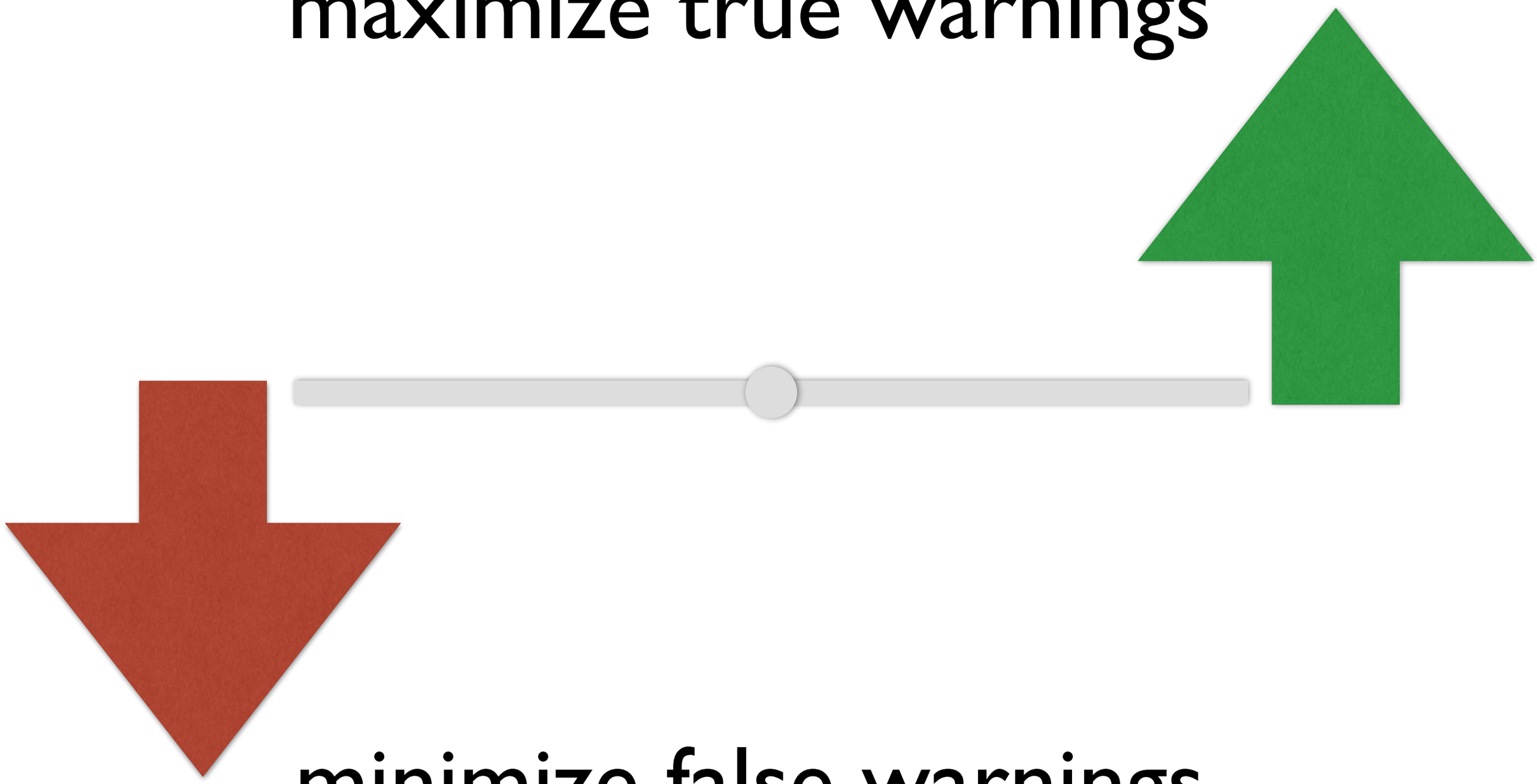
```
void main() {
        a = new A();
        b = a.g;
        foo(a);
        sink(b.f);
}
```

```
void foo( z ) {
        x = z.g;
        w = source();
        x.f = w;
}
```

Will it leak?

Two main requirements

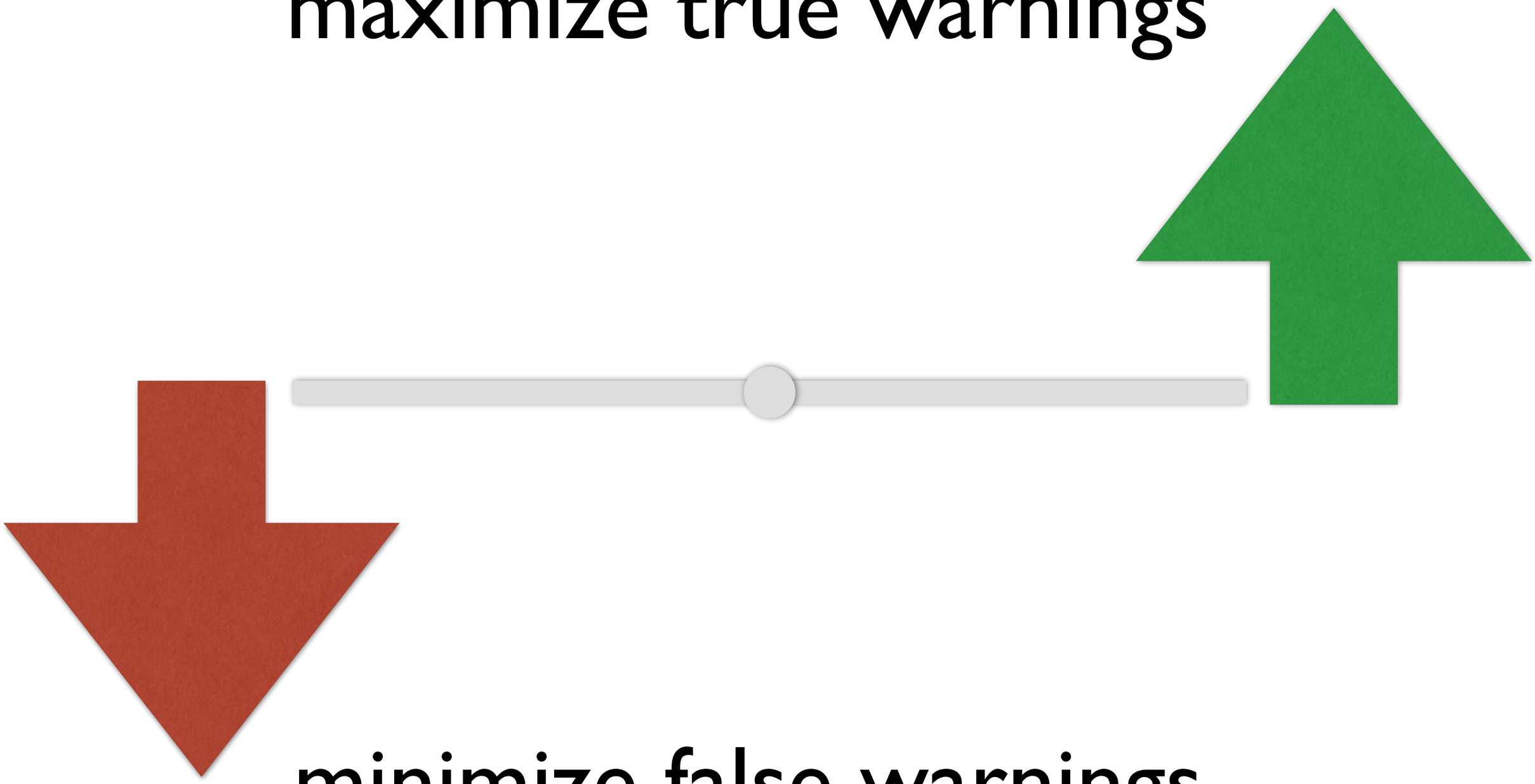
maximize true warnings



minimize false warnings

and do it fast!

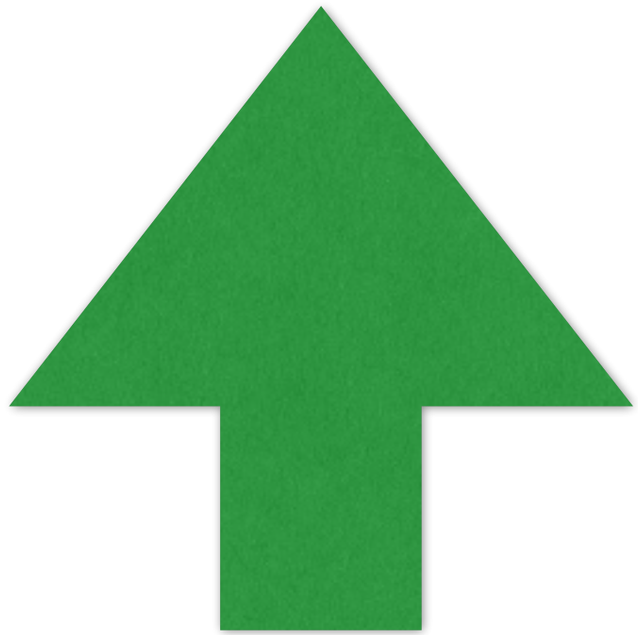
maximize true warnings



minimize false warnings

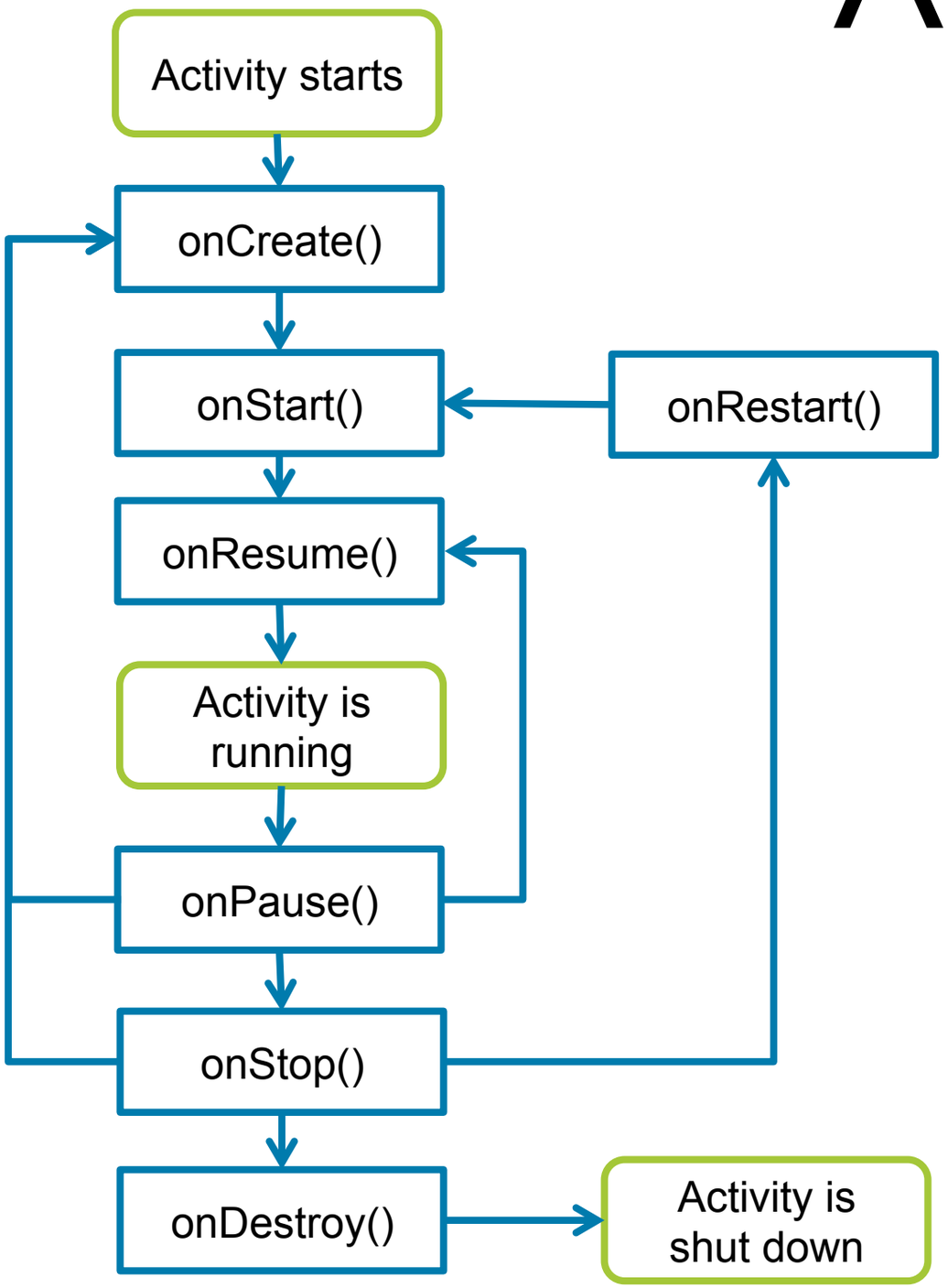
maximizing

true warnings

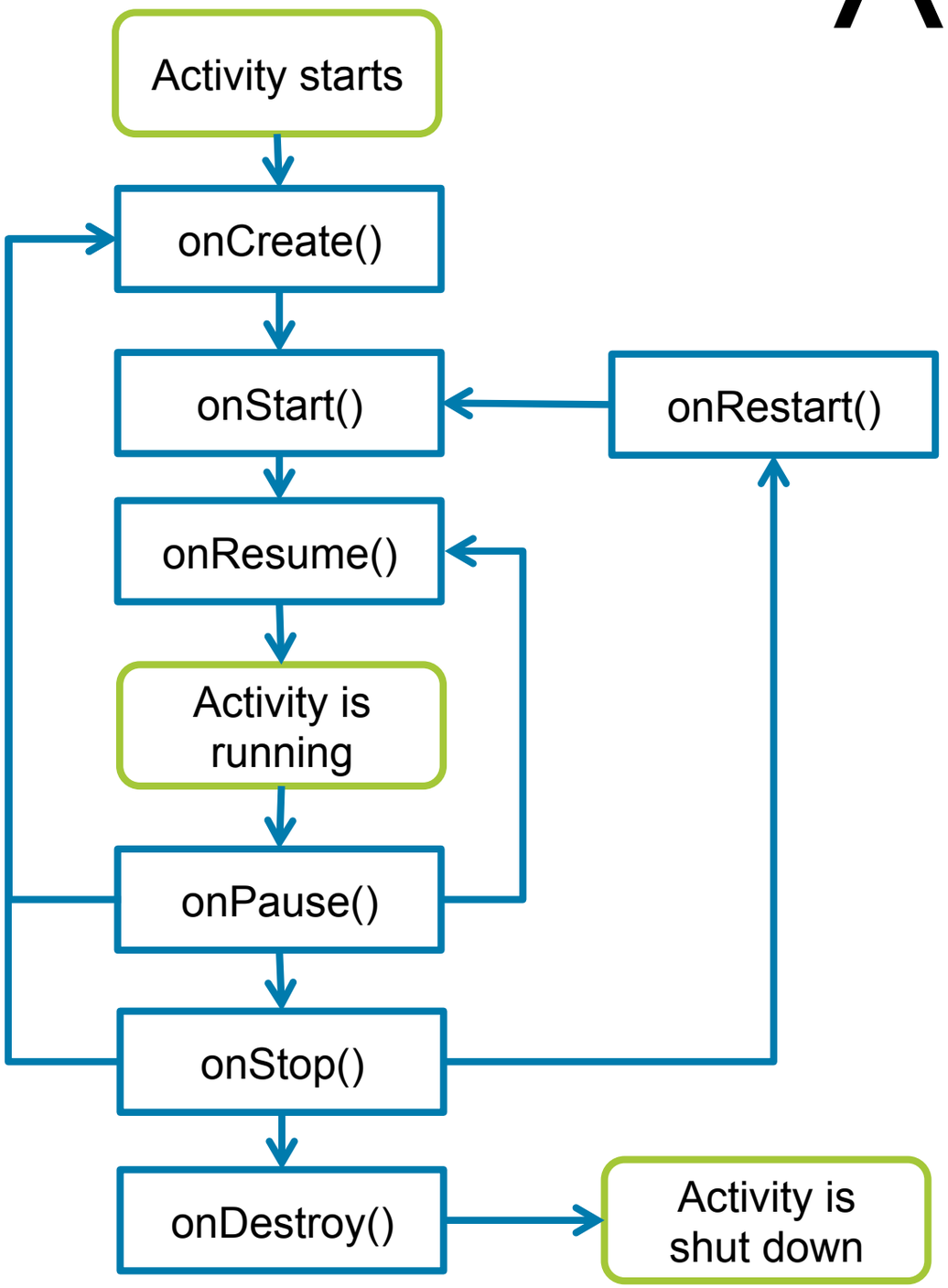


Recall

Activity lifecycle

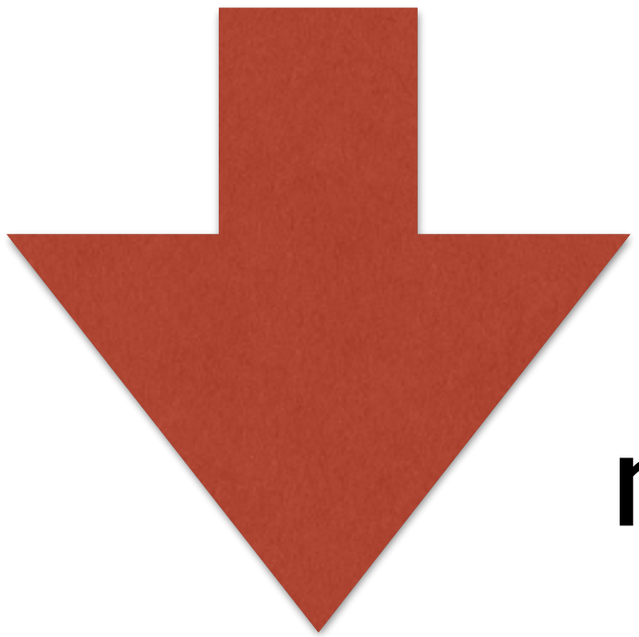


Activity lifecycle



Also:

- services
- broadcast receivers
- content providers
- fragments



minimize false warnings

Options for highly precise analysis

- **Flow-sensitive:** respect program order
- **Context-sensitive:** re-analyze calls to the same methods but with different parameters
- **Field-sensitive:** distinguish different fields of the same object
- **Object-sensitive:** distinguish different "owner objects"
- Optional: on-demand alias analysis with same level of precision

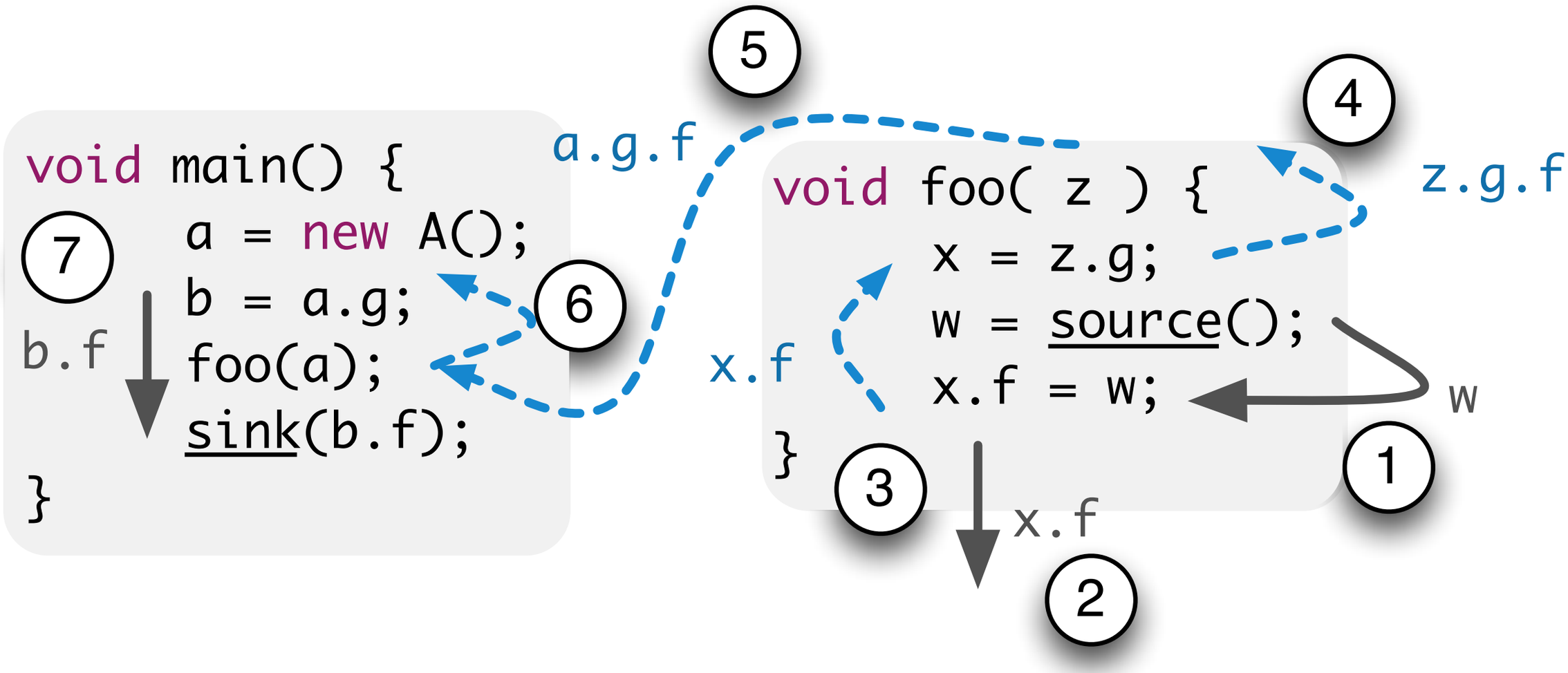
Intertwined analyses: forward taint analysis, backward alias analysis

```
void main() {
        a = new A();
        b = a.g;
        foo(a);
        sink(b.f);
}
```

```
void foo( z ) {
        x = z.g;
        w = source();
        x.f = w;
}
```

Inspired by **Tripp et al., FASE 2013:** ***Andromeda: Accurate and scalable security analysis of web applications***

Intertwined analyses: forward taint analysis, backward alias analysis



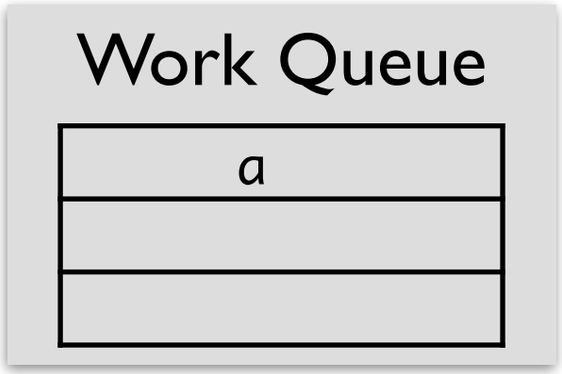
Inspired by **Tripp et al., FASE 2013:** ***Andromeda: Accurate and scalable security analysis of web applications***

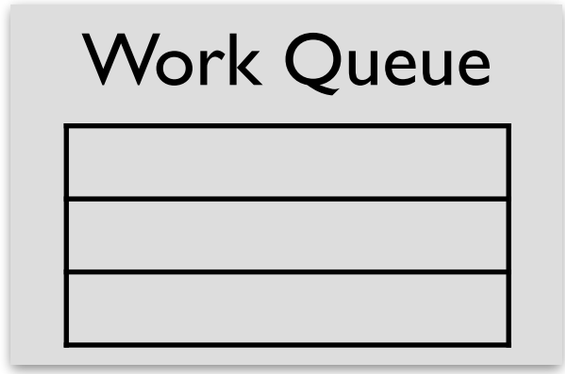
```
a = source();
x = y;
x.f = a;
leak(y.f);
```

Forward
Taint Analysis

Backward
Alias Analysis

```
a = source();
x = y;
x.f = a;
leak(y.f);
```





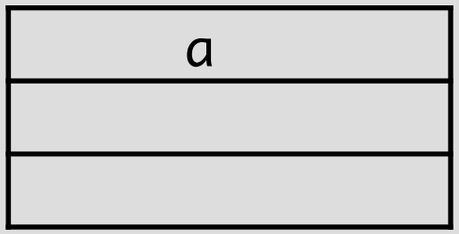
```
a = source();
x = y;
x.f = a;
leak(y.f);
```

Forward
Taint Analysis

Backward
Alias Analysis

x.f=a

Work Queue



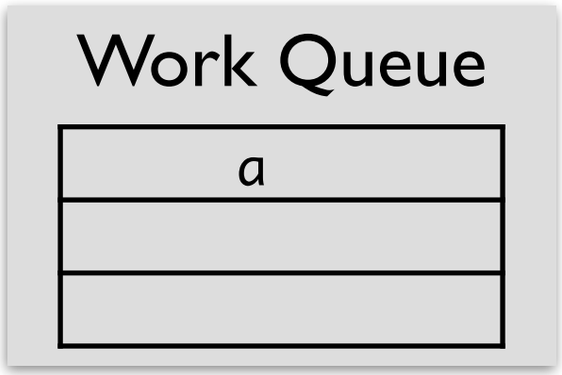
Work Queue

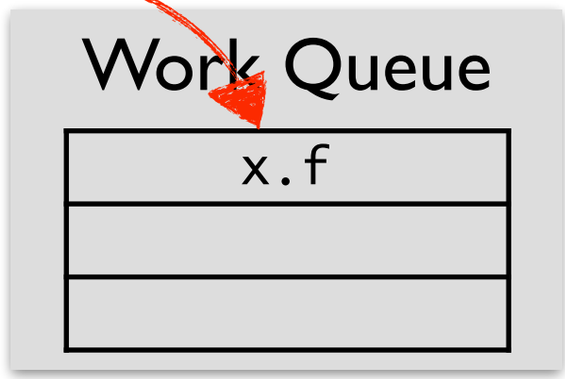
```
a = source();
x = y;
x.f = a;
leak(y.f);
```

Forward
Taint Analysis

Backward
Alias Analysis

x.f=a





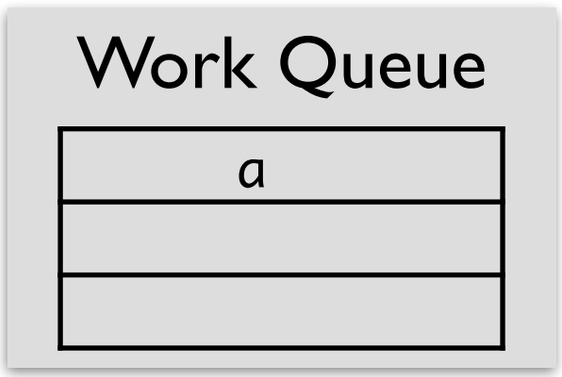
```
a = source();
x = y;
x.f = a;
leak(y.f);
```

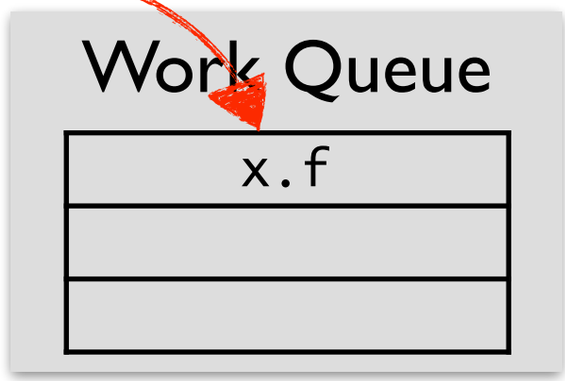
Forward
Taint Analysis

Backward
Alias Analysis

x.f=a

x=y





```
a = source();
x = y;
x.f = a;
leak(y.f);
```

Forward Taint Analysis

Backward Alias Analysis

x.f=a

x=y

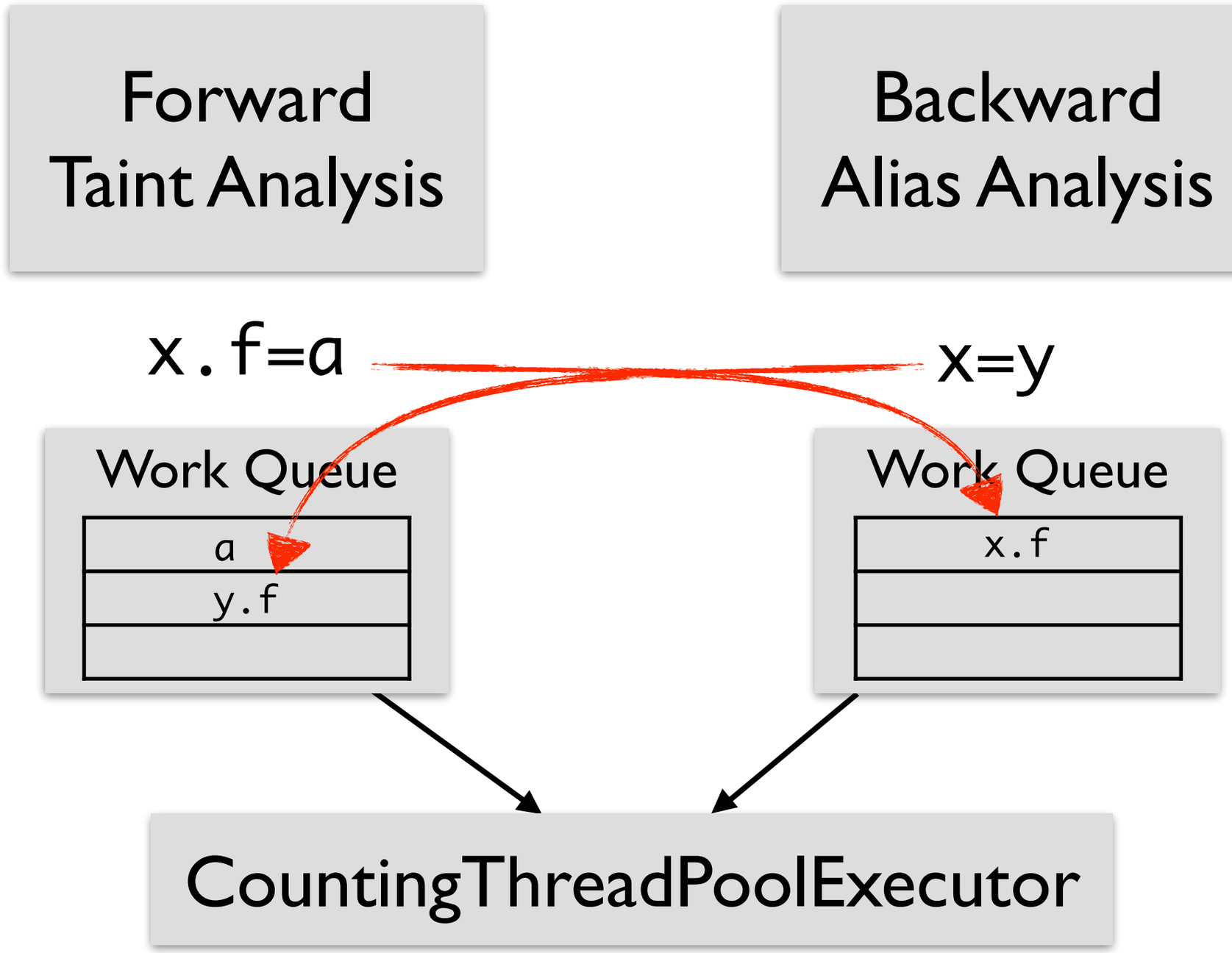
Work Queue

a
y.f

Work Queue

x.f

```
a = source();
x = y;
x.f = a;
leak(y.f);
```



terminates not before *both* queues are empty

Demand-driven alias analysis

Demand-driven alias analysis

Devil in in the details:

Demand-driven alias analysis

Devil in in the details:

- How exactly to coordinate the interplay between both analysis?

Demand-driven alias analysis

Devil in in the details:

- How exactly to coordinate the interplay between both analysis?
- How to maintain context- and flow sensitivity?
  - Both analyses use different (value) contexts

Demand-driven alias analysis

Devil in in the details:

- How exactly to coordinate the interplay between both analysis?
- How to maintain context- and flow sensitivity?
  - Both analyses use different (value) contexts
- How to keep summaries concise? (important for performance and memory footprint)

Demand-driven alias analysis

Devil in in the details:

- How exactly to coordinate the interplay between both analysis?
- How to maintain context- and flow sensitivity?
  - Both analyses use different (value) contexts
- How to keep summaries concise? (important for performance and memory footprint)

Details: see PLDI'14 paper

… and do it fast!

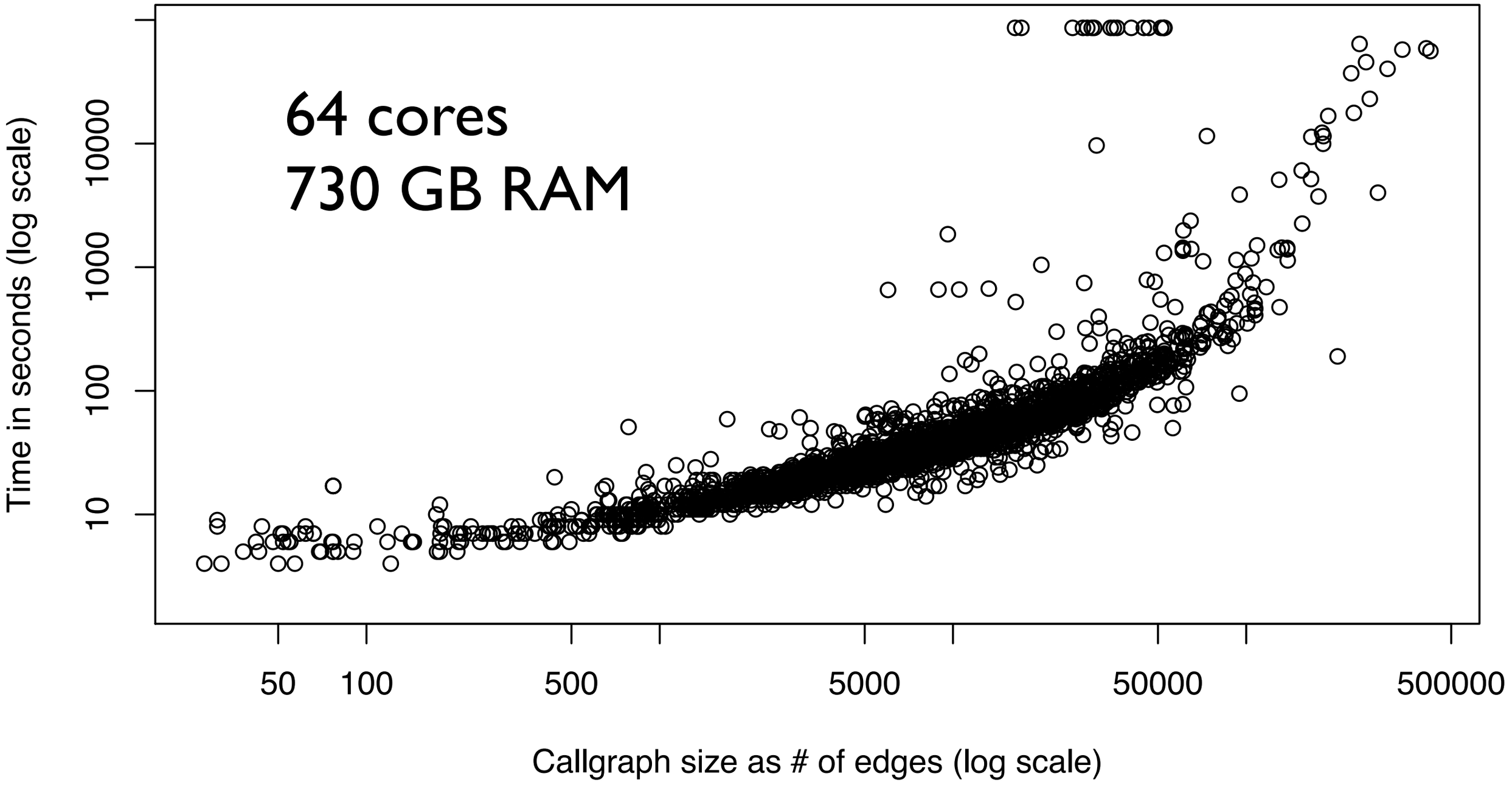


Chart from joint work with Gorla et al., under submission

secure-software-engineering / soot-infoflow-android

Android-specific components of FlowDroid — Edit

438 commits | 4 branches | 1 release | 5 contributors

branch: develop soot-infoflow-android / +

made the Android platform JAR directory a normal configuration option

StevenArzt authored on Jul 17 latest commit d48712c20e

PLDI14 Distribution	added a self-contained version for the PLDI'14 artifact evaluation	7 months ago
insecureBank	insecure Bank app (does not work at the moment, ArrayIndexOutOfBounds…	2 years ago
lib	removed old library	5 months ago
src/soot/jimple/infoflow/android	add disable callback sources option	2 months ago
test/soot/jimple/infoflow/andr…	added a new test case	4 months ago
testAPKs	added a new test case for strong updates	4 months ago
.classpath	modernized the layout file parser and migrated it to axml 2.0	5 months ago
.gitattributes	added git attribute file	a year ago

Code
Issues 21
Pull Requests 1
Wiki
Pulse
Graphs
Settings

HTTPS clone URL

https://github.com/:

You can clone with HTTPS, SSH, or Subversion.

Pay as you go…

`--aliasflowins`
use flow-insensitive ahead-of-time alias analysis (Spark)

`--aplength n`
restrict length of "access paths" (less object-sensitive)

`--nocallbacks`
don't model callbacks in Android lifecycle

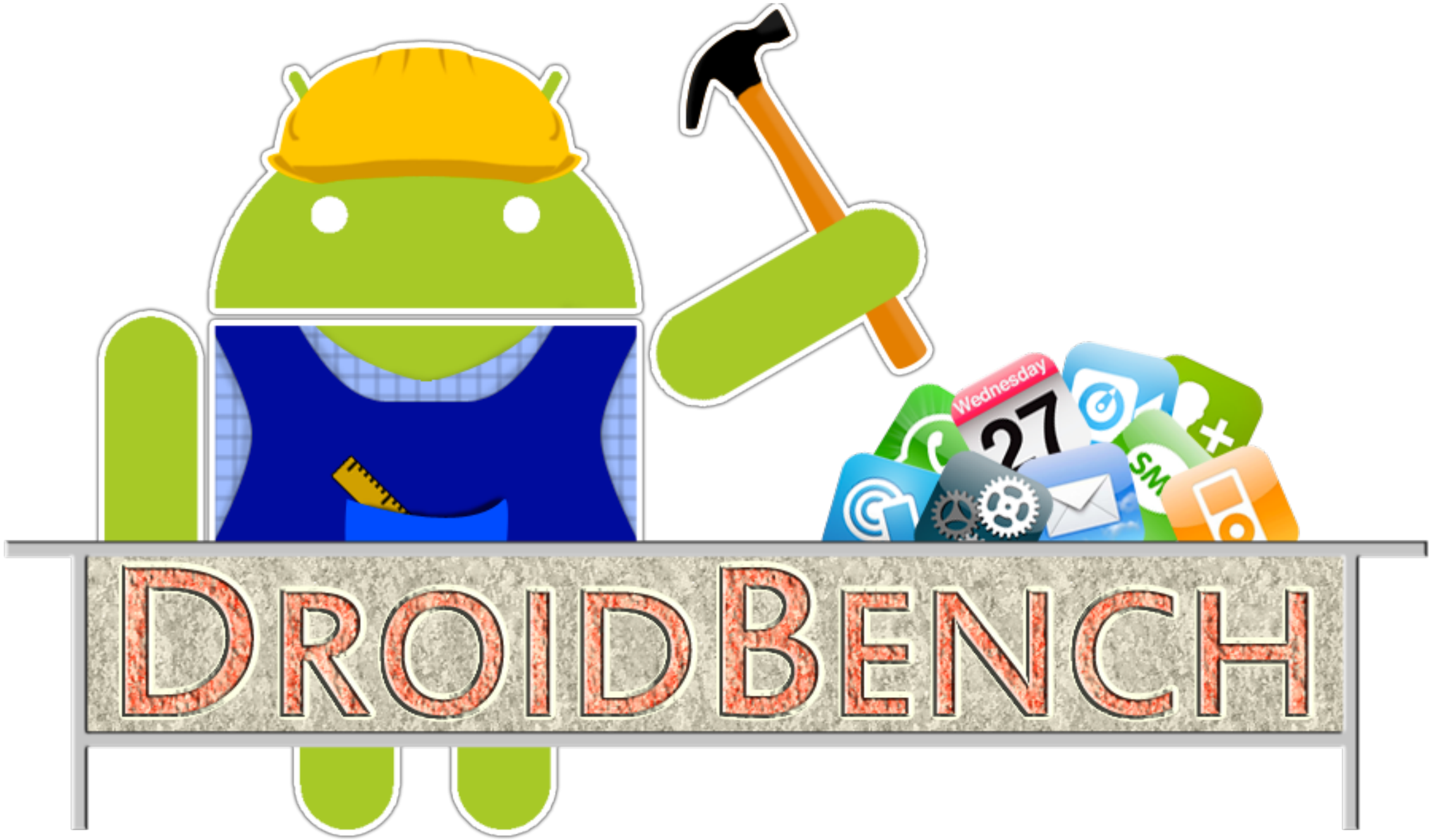
`--nopaths`
don't record path along which taint flows

`--nostatic`
don't track static fields

Open question:

How to enable full precision efficiently on *all* practical apps?

- Generalize concept of inter-connected flow-sensitive analyses
- Fully automatic recovery of runtime values in obfuscated apps
  - reflective calls, phone numbers, C&C commands, …
- Effective visualization of data leaks
- Tool-assisted “debugging” and reverse engineering on the binary level


Wednesday
27
DroidBench

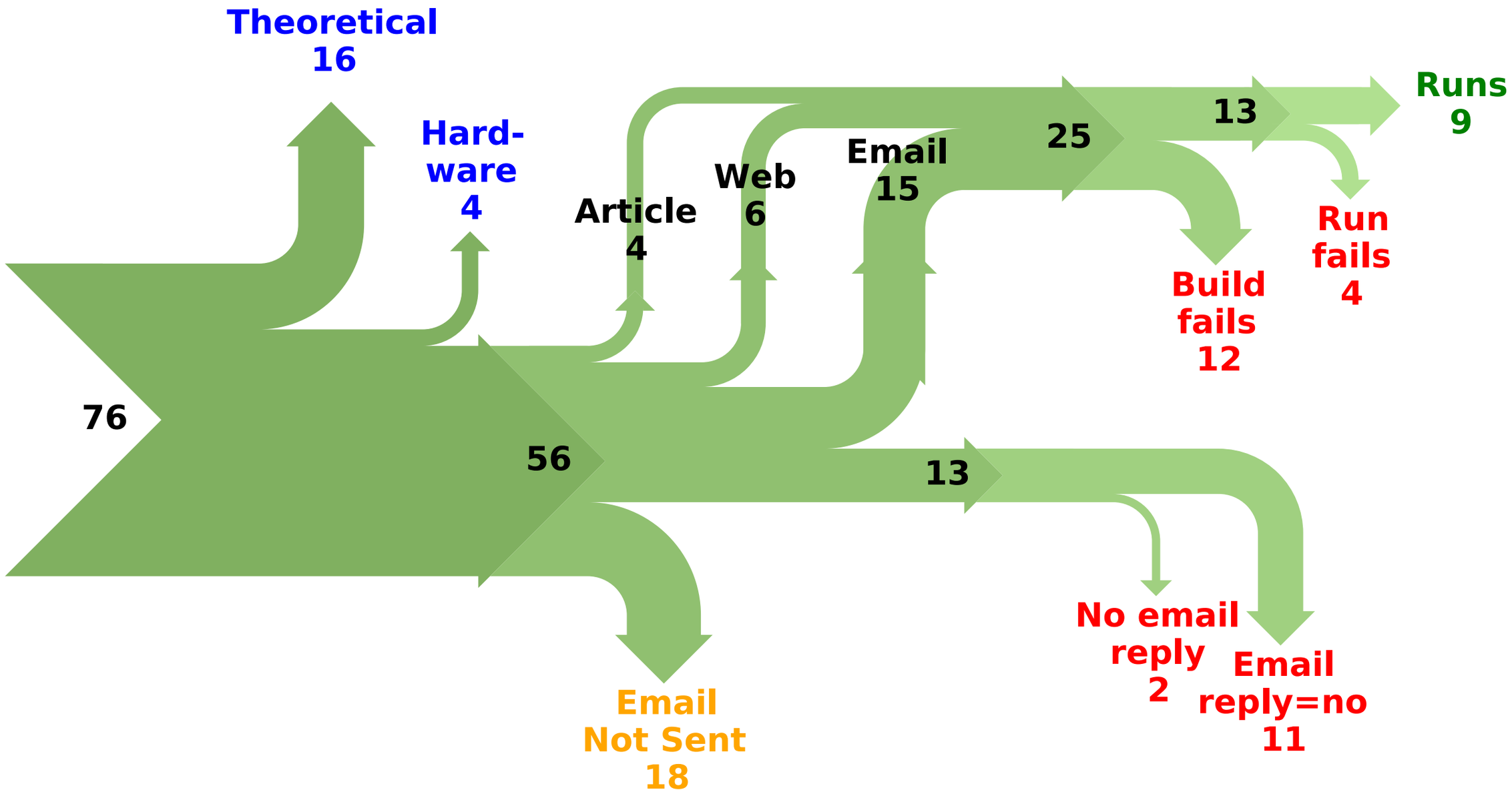
64 Test apps

- Complex data structures
- Callbacks, Lifecycle
- Field and Object Sensitivity
- Inter-app communication
- Reflection
- Implicit flows

Other Static Approaches:

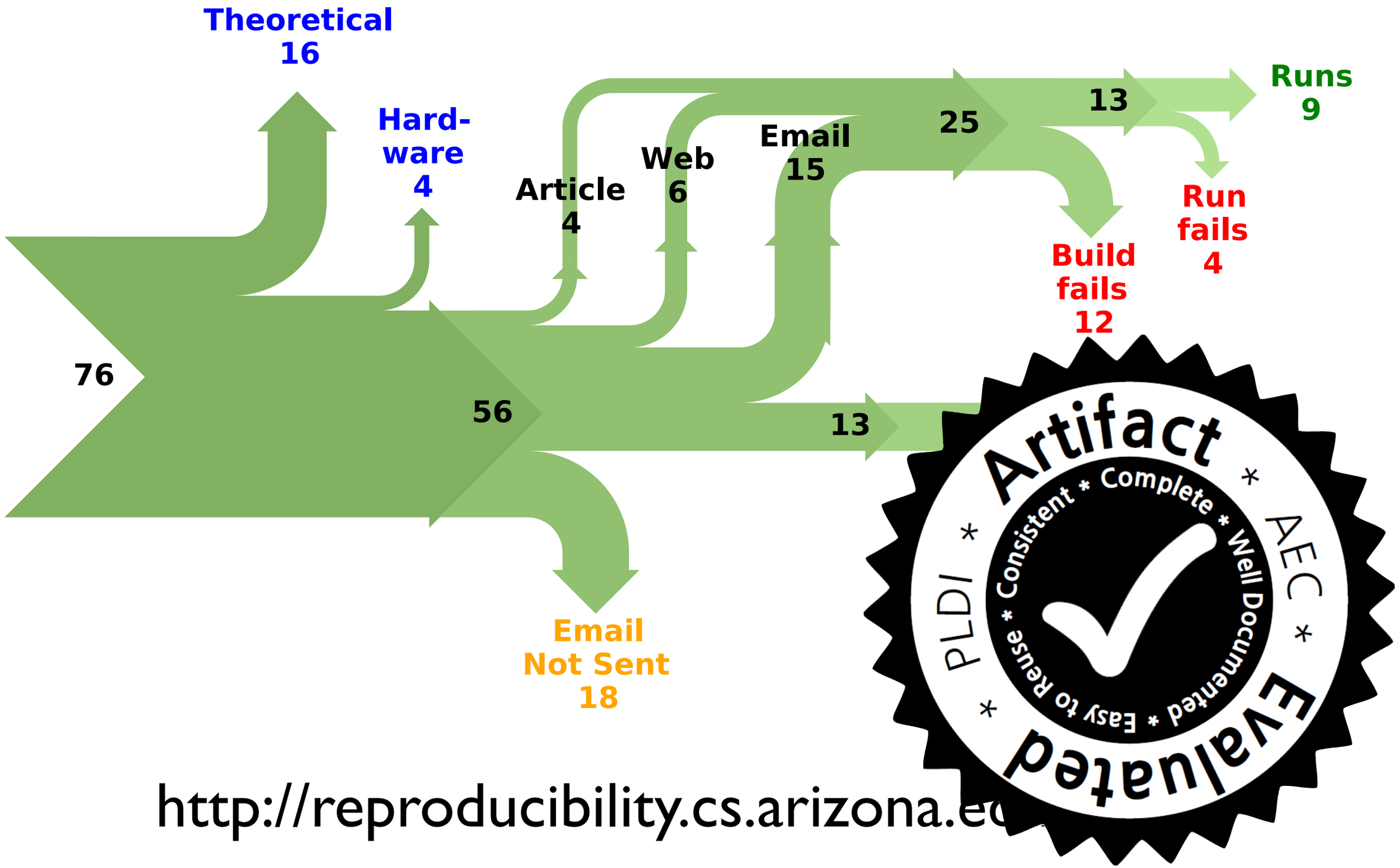
ScanDroid [TR 09], DeD [SEC'11], CHEX [CCS'12], LeakMiner [WCSE'12], ScanDal [Most'12], AndroidLeaks [TRUST'12], SAAF [SAC'13],…

CCS



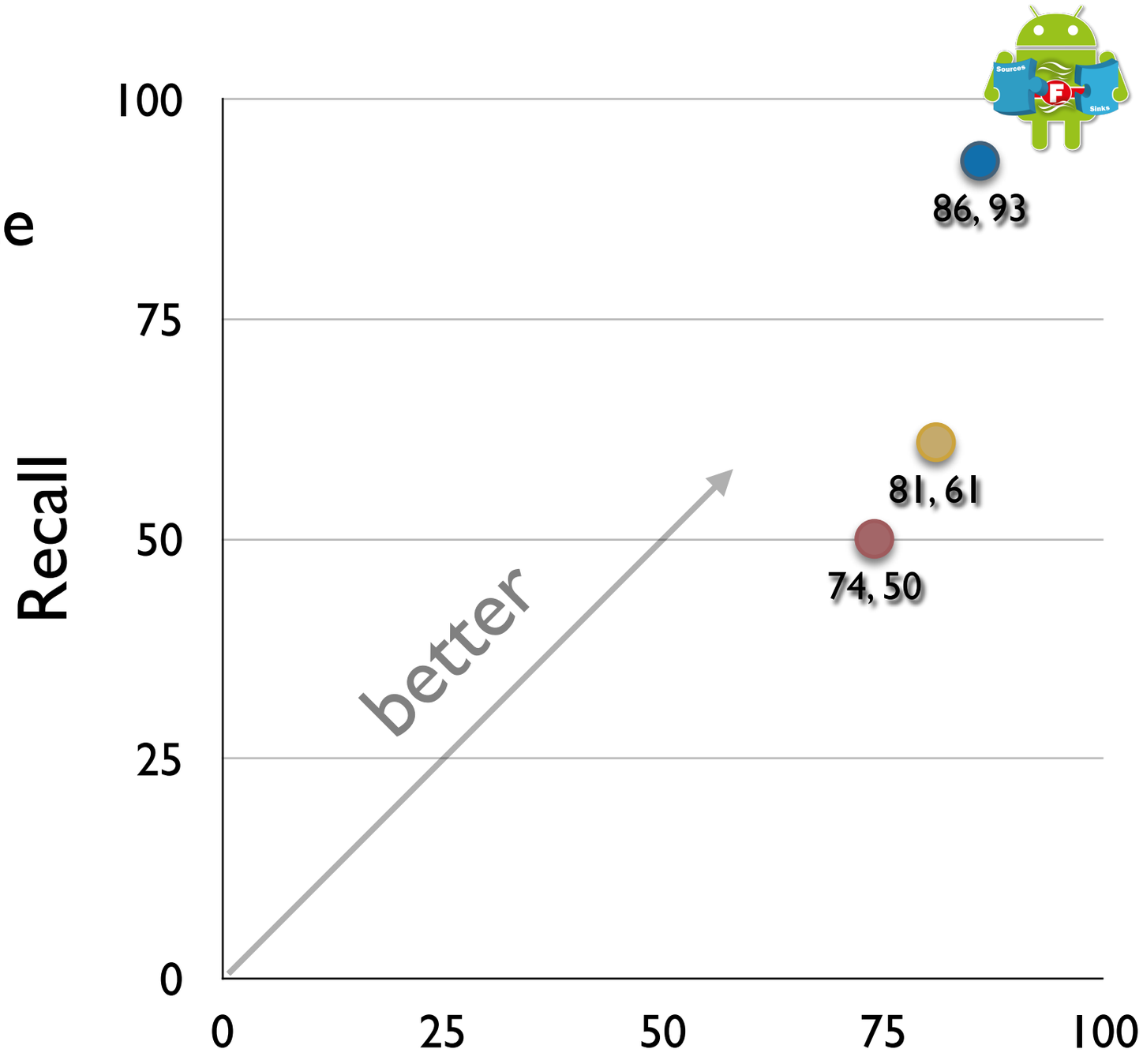
http://reproducibility.cs.arizona.edu/

CCS

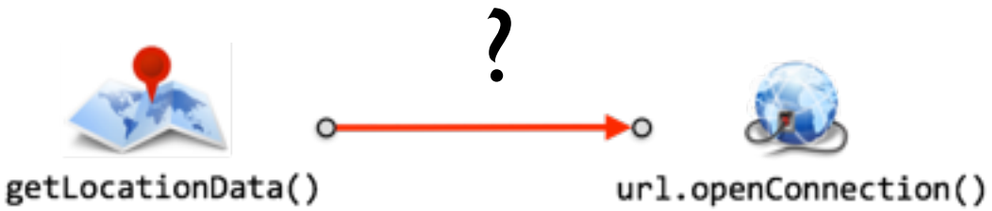


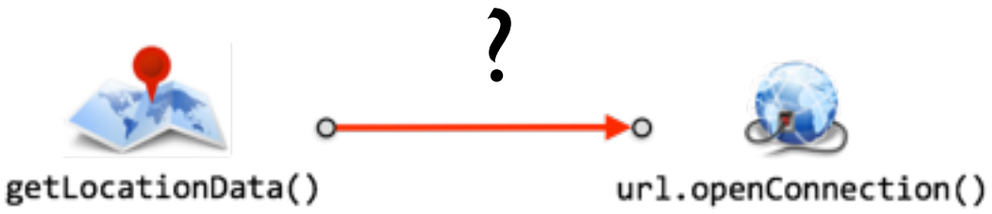
http://reproducibility.cs.arizona.ed

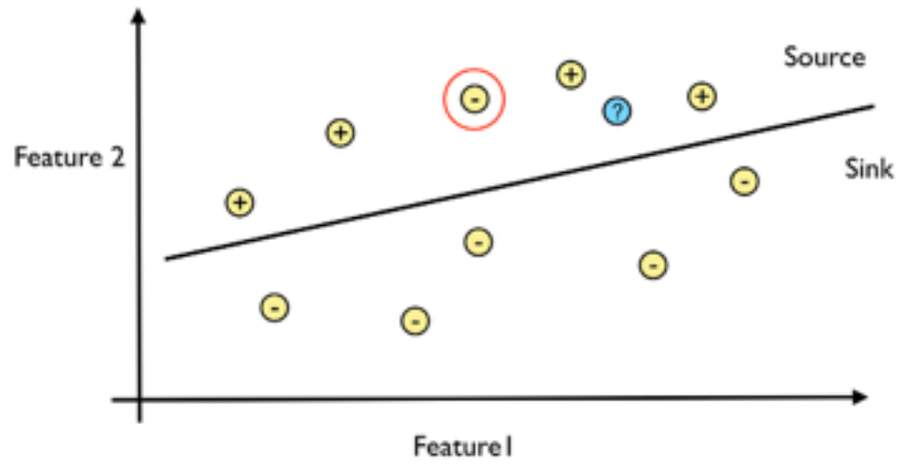
- FlowDroid
- IBM AppScan Source
- HP Fortify

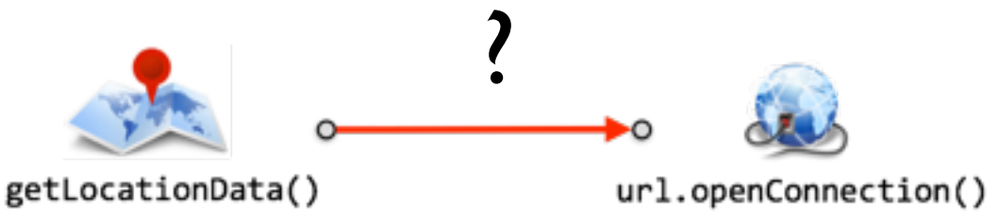


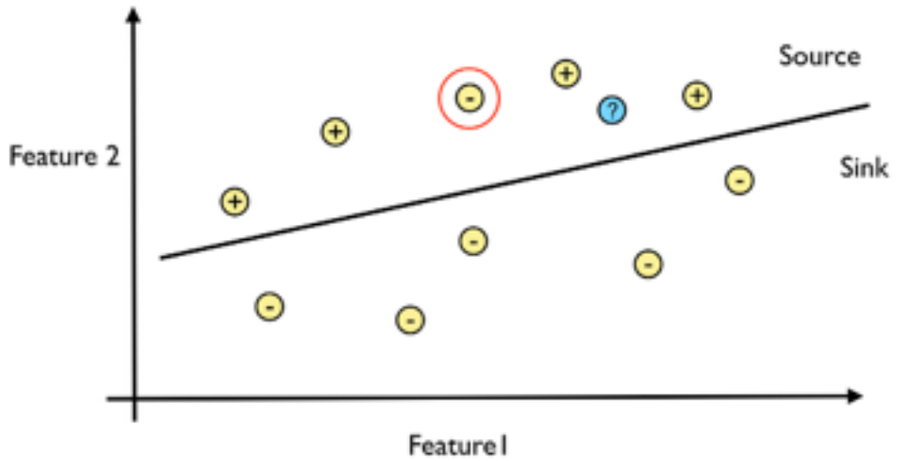
on DroidBench 1.0

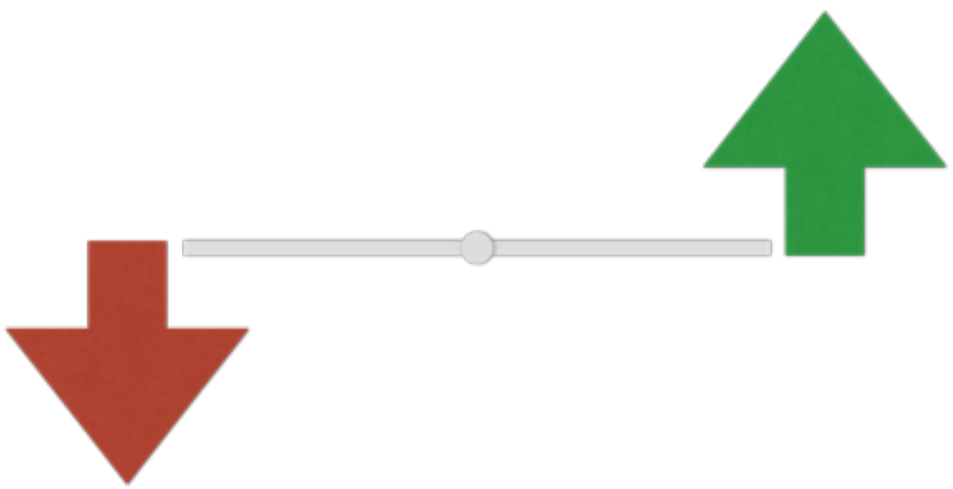

?
getLocationData()
url.openConnection()

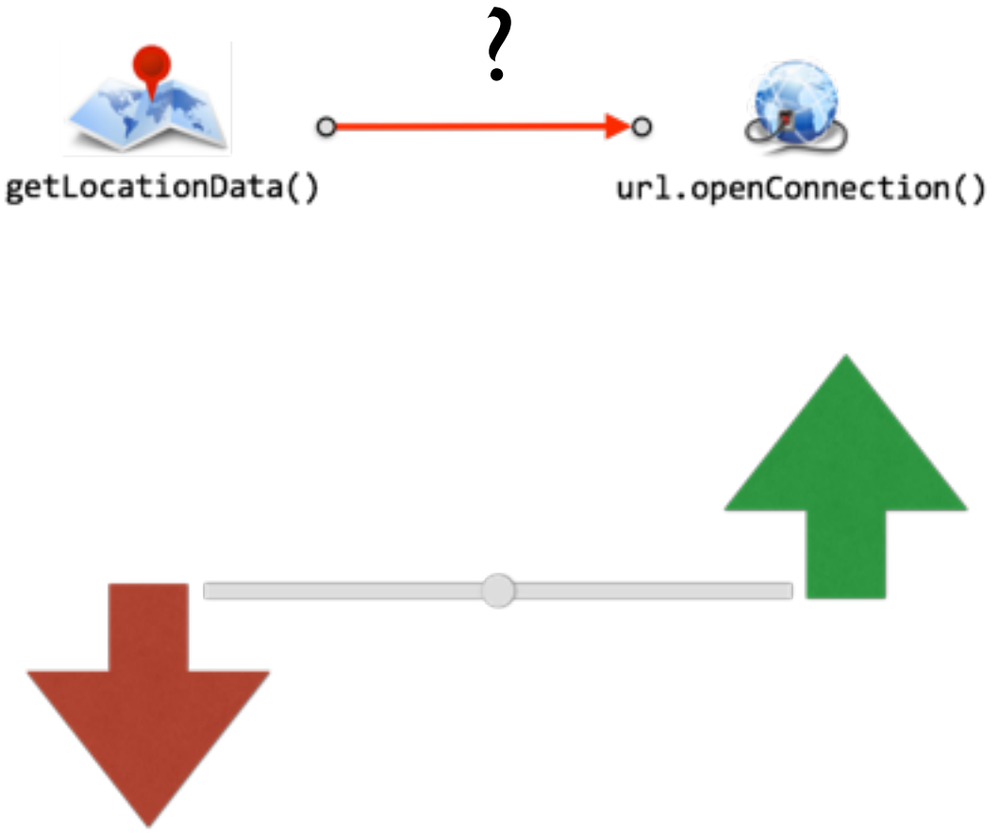

?
getLocationData()
url.openConnection()

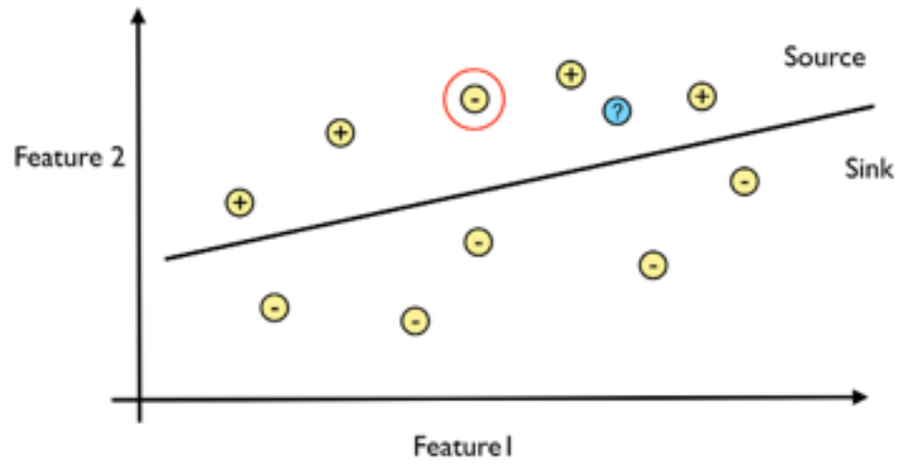

Source
Sink
Feature 2
Feature1

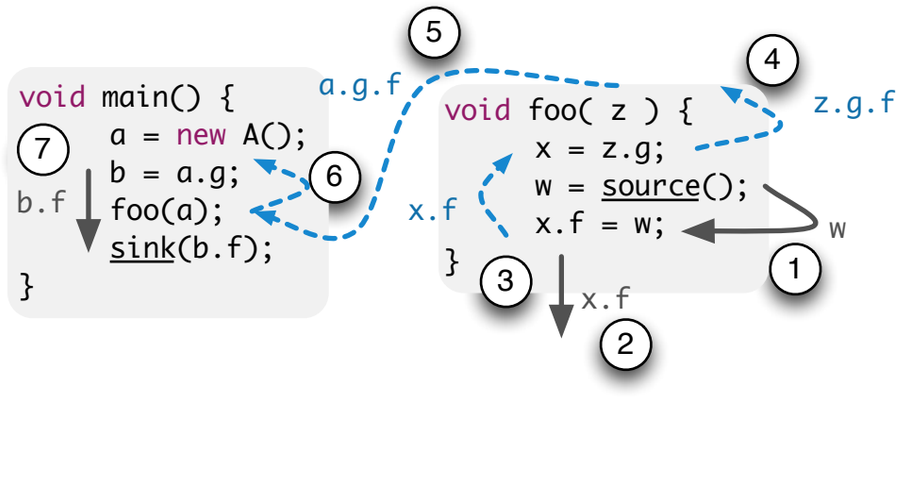

?
getLocationData()
url.openConnection()

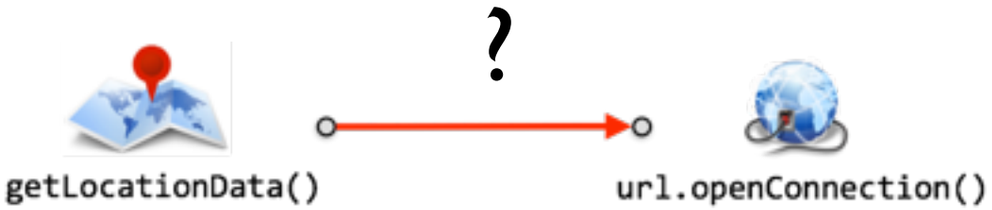

Source
Sink
Feature 2
Feature1

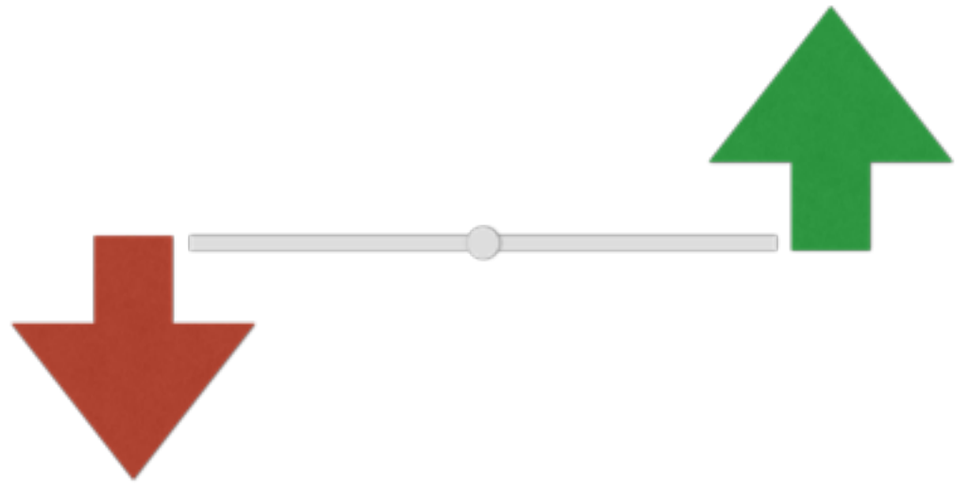


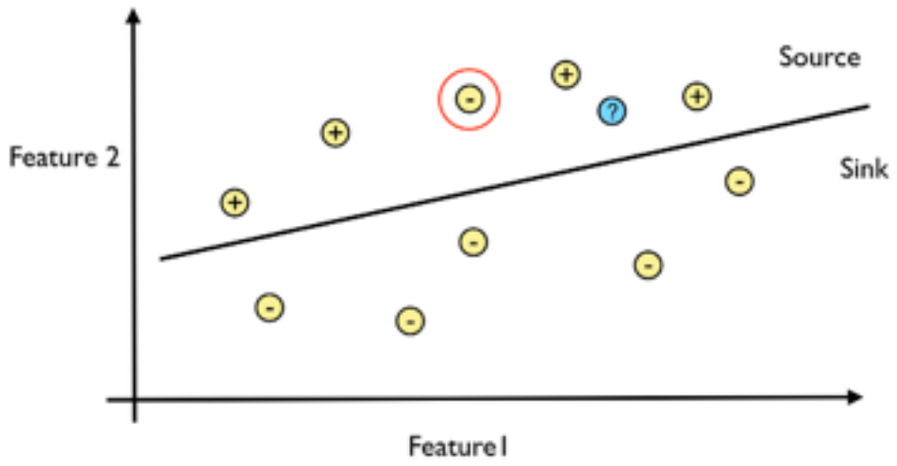

?
getLocationData()
url.openConnection()

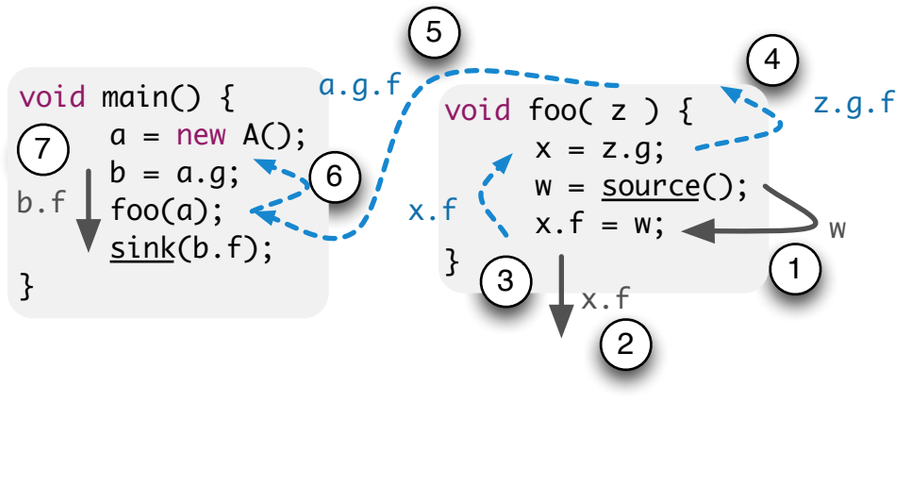

Source
Sink
Feature 2
Feature1


void main() {
a = new A();
b = a.g;
foo(a);
sink(b.f);
}
void foo(z) {
x = z.g;
w = source();
x.f = w;
}
a.g.f
z.g.f
x.f
b.f
w
x.f
1
2
3
4
5
6
7

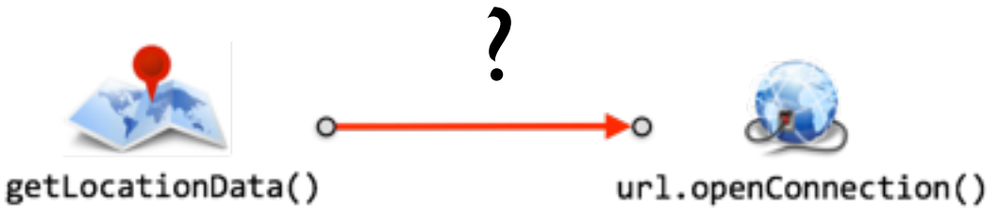

?
getLocationData()
url.openConnection()

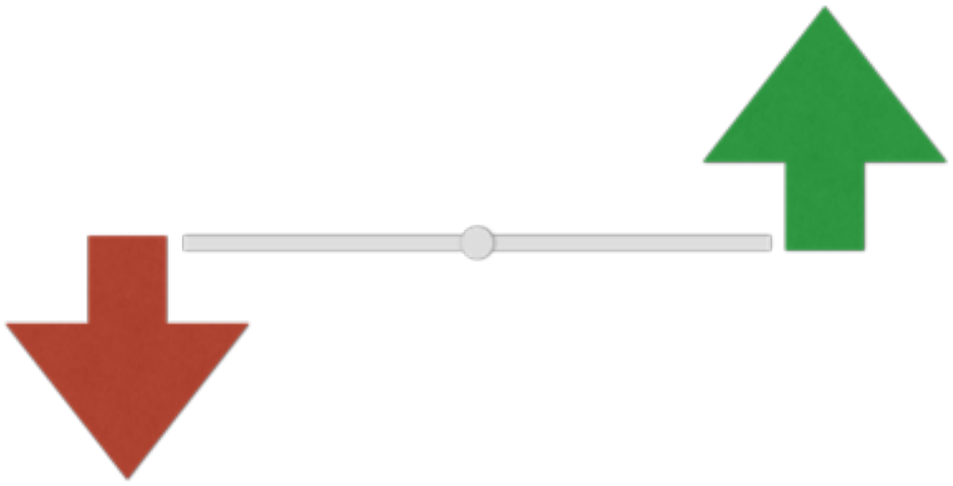


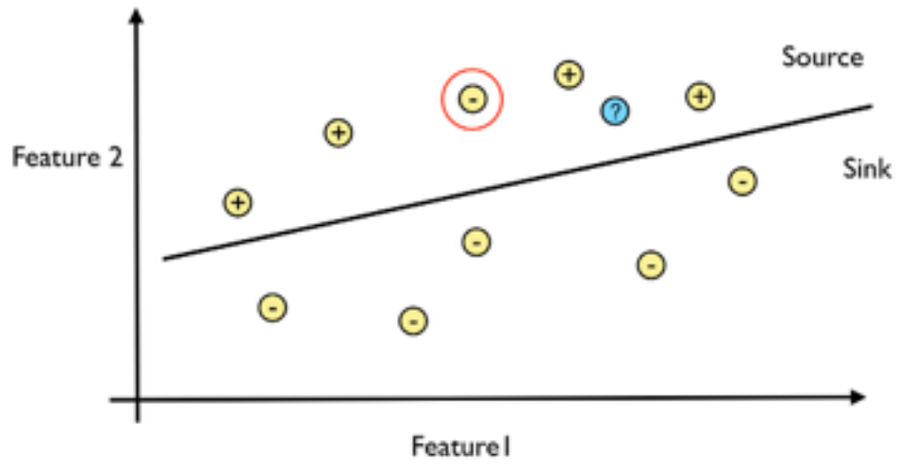

Source
Sink
Feature 2
Feature1

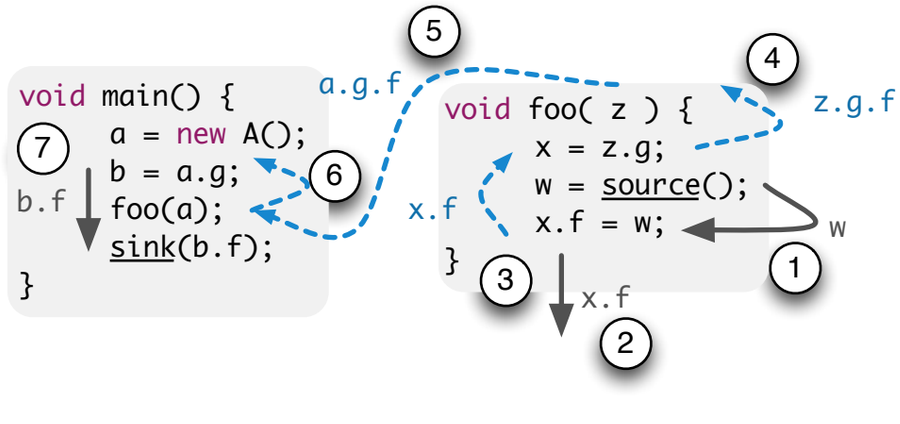

void main() {
a = new A();
b = a.g;
foo(a);
sink(b.f);
}
void foo(z) {
x = z.g;
w = source();
x.f = w;
}
a.g.f
z.g.f
x.f
b.f
w
x.f
1
2
3
4
5
6
7


DROIDBENCH

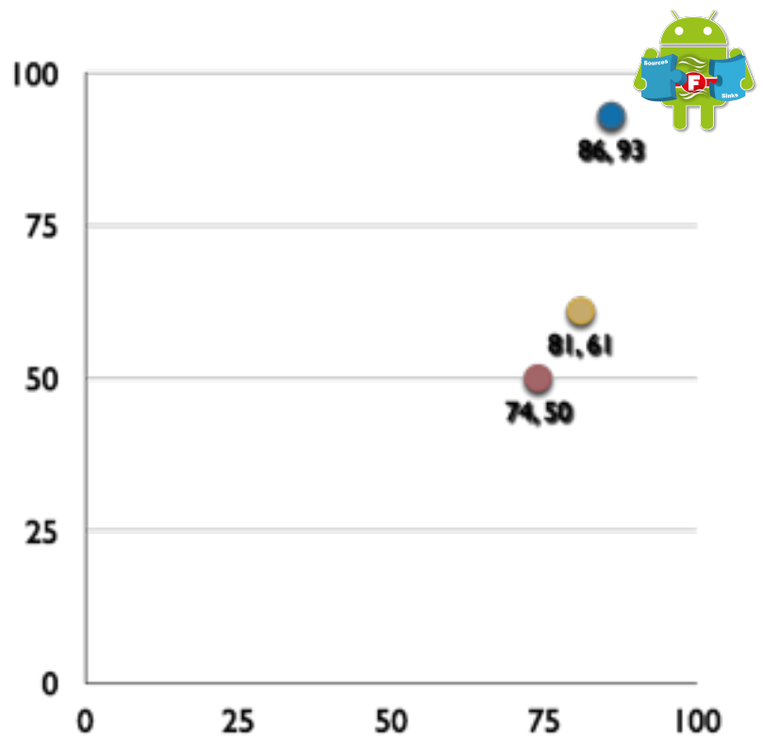

?
getLocationData()
url.openConnection()

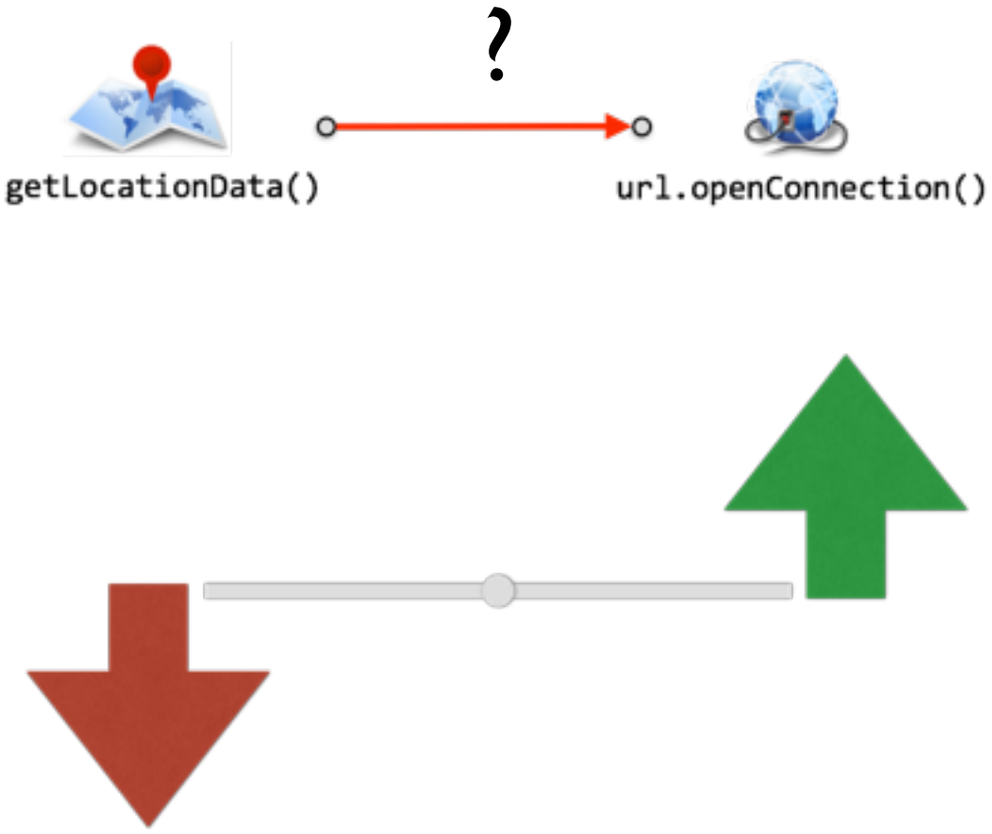


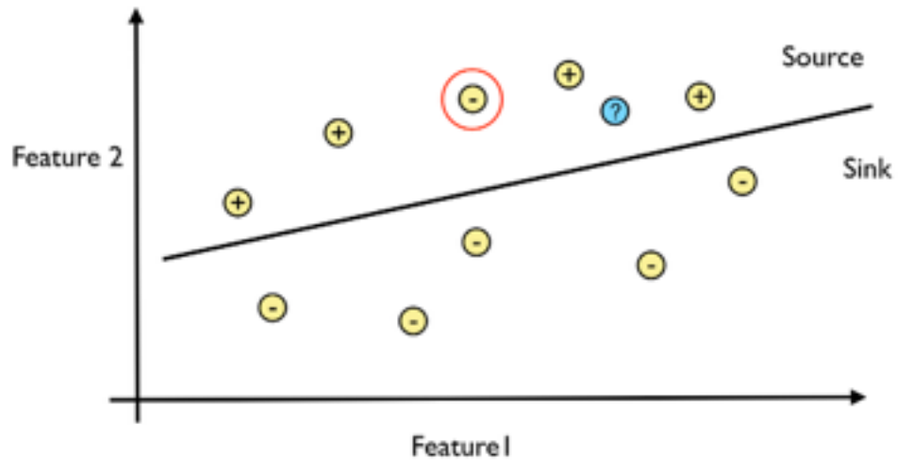

Source
Sink
Feature 2
Feature1

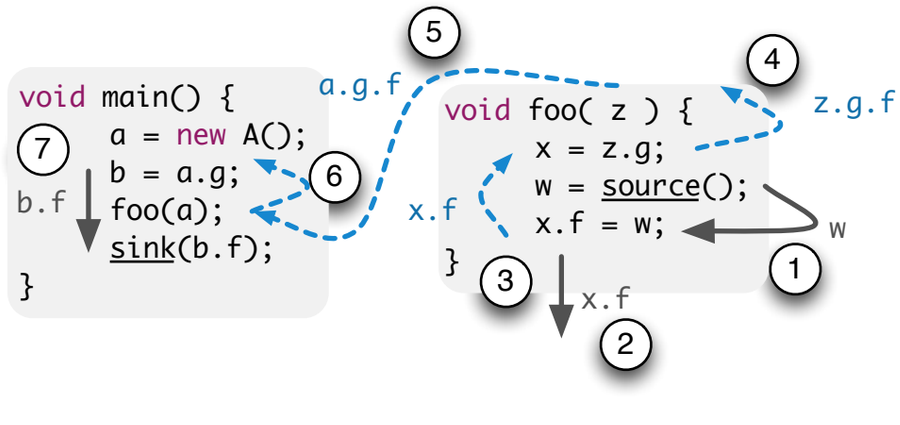

void main() {
a = new A();
b = a.g;
foo(a);
sink(b.f);
}
void foo(z) {
x = z.g;
w = source();
x.f = w;
}
a.g.f
z.g.f
x.f
b.f
w
x.f
1
2
3
4
5
6
7

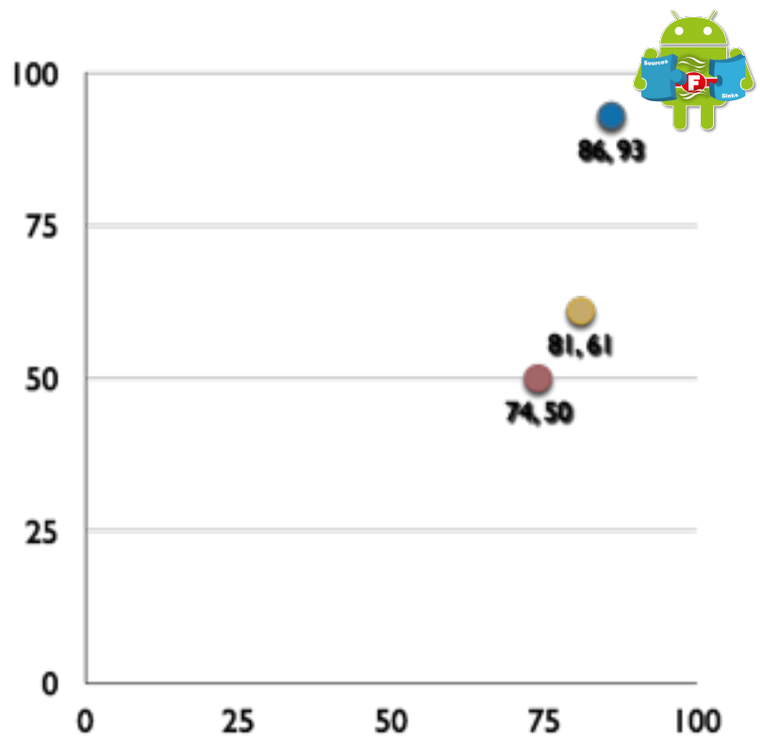

DROIDBENCH


100
75
50
25
0
0
25
50
75
100
86, 93
81, 61
74, 50









sseblog.ec-spride.de/tools/