Building a Next-Generation App Store

Jason Franklin Ph.D.
jfrankli@cs.stanford.edu
Research Associate
Stanford University

App Store: Promise

Safe, Trusted, Centralized

App Stores: Reality

"**Permissions changed** in the latest update to read my phone number. **Totally unacceptable** for a puzzle game. Uninstalling." [1]

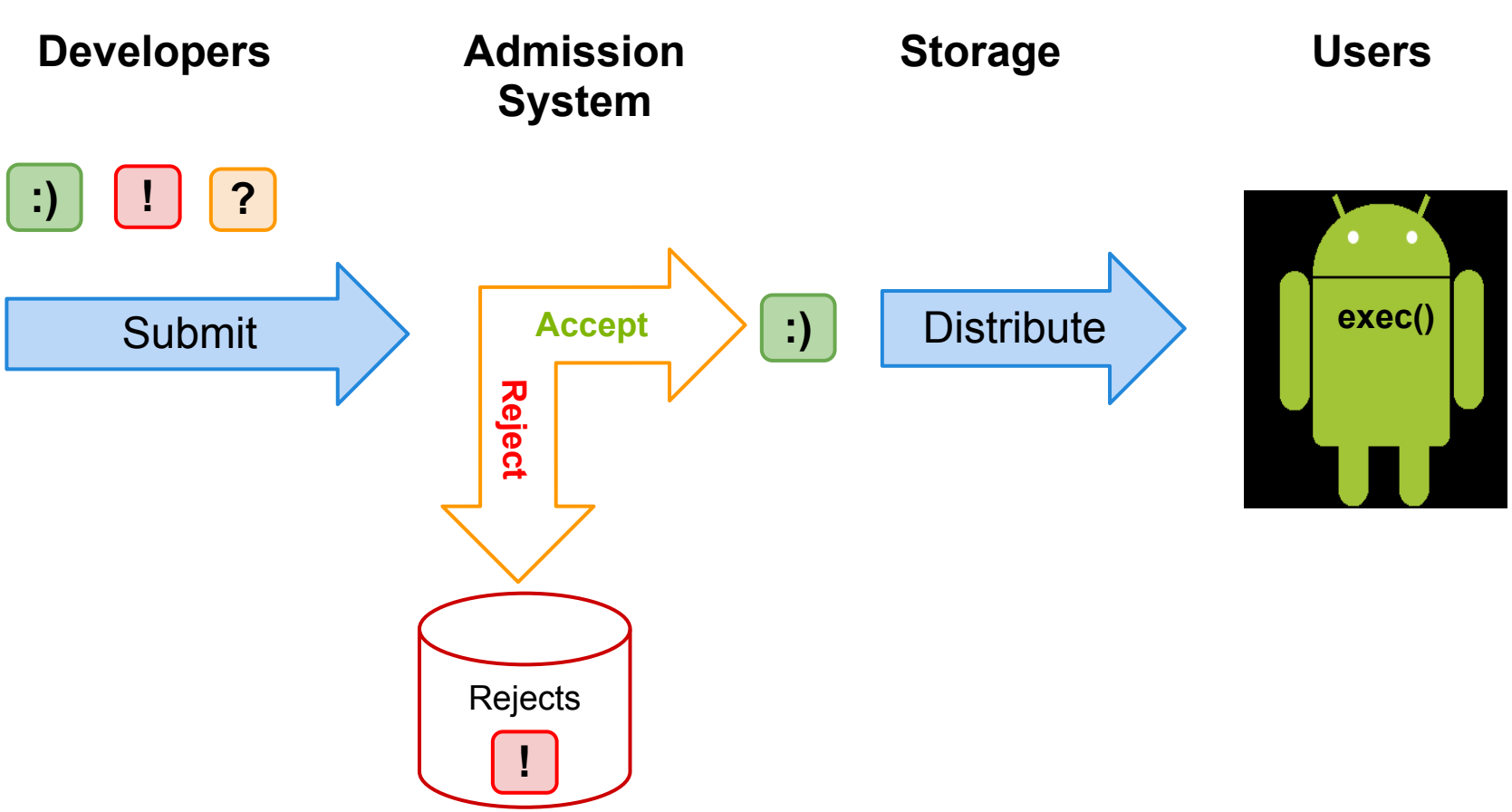
"**Uninstalling** due to the added permissions**.**" [1]

"Why suddenly Read phone state permission?" [1]

"Simple and challenging game but with new update there is too many Permissions for a simple game, will not be updating and once completed all levels I will be deleting it." [1]

[1] Oh, My Brain! Block Buzzle by mToy, https://play.google.com/store/apps/details?id=biz.mtoy.blockpuzzle&feature=related_apps#?t=W251bGwsMSwxLDEwOSwiYml6Lm10b3kuYmxvY2twdXp6bGUiXQ..

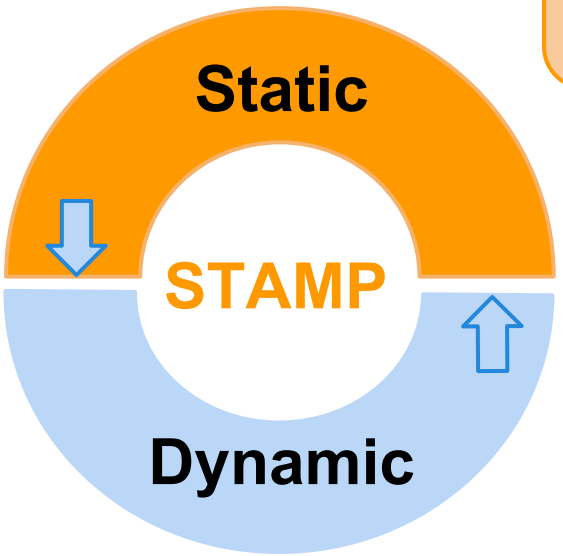
Anatomy of an App Store



STAMP Admission System

Static Analysis

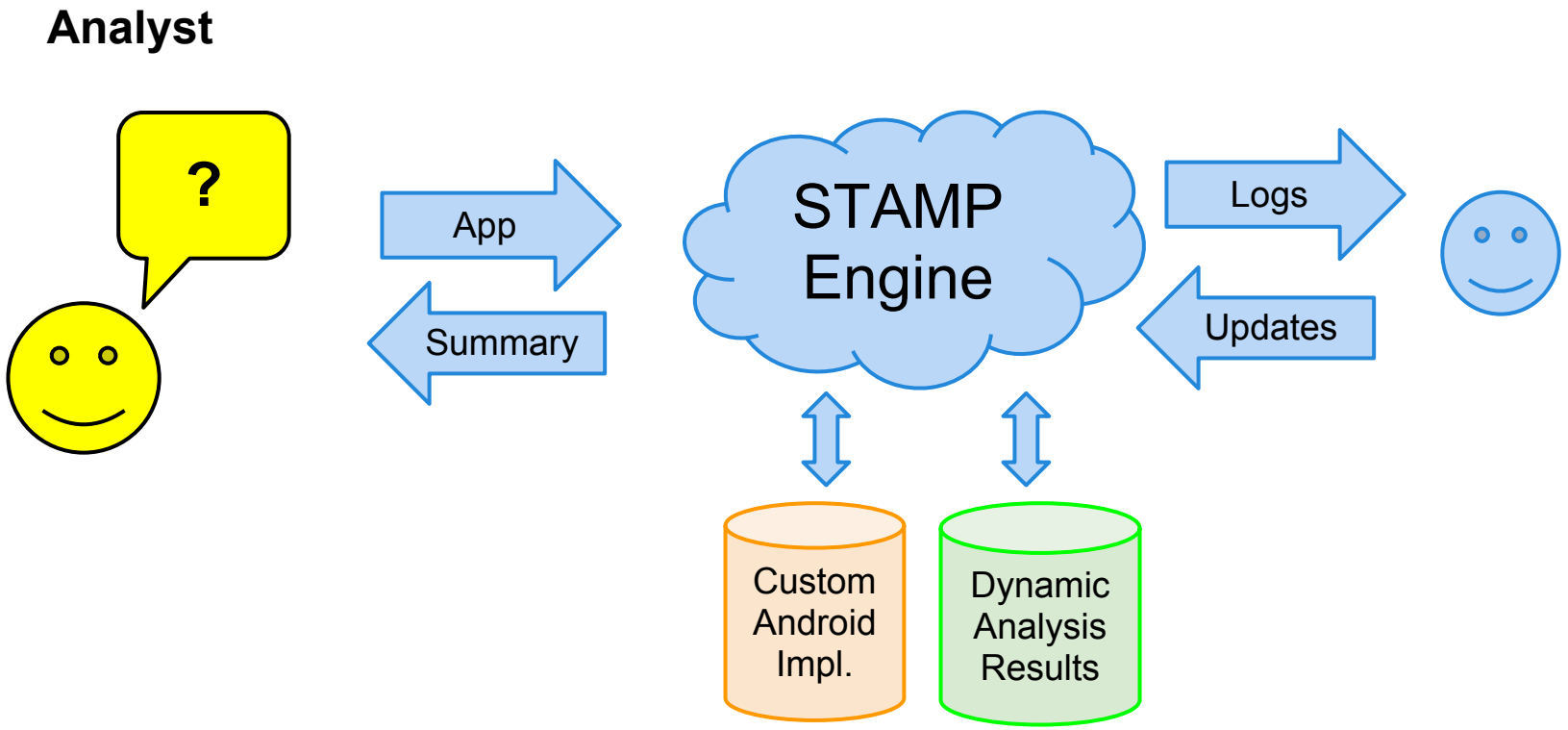
More behaviors, fewer details



Dynamic Analysis

Fewer behaviors, more details

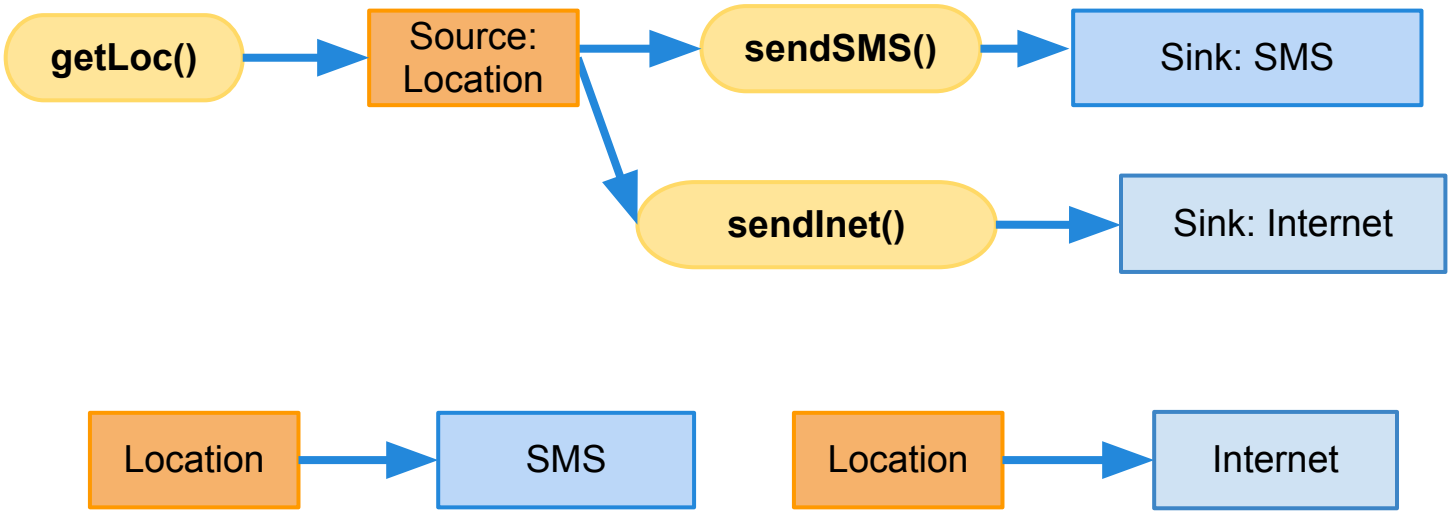
STAMP as a Service



Static vs. Dynamic Analysis

- Can be faster
  - Google Bouncer (dynamic): 300 seconds
  - Unopt. STAMP: ~40 seconds
- Detects more behaviors
  - Closer to 100% coverage
- No code execution
  - Avoid configuration issues, VMs/emulators, and input generation while always getting results

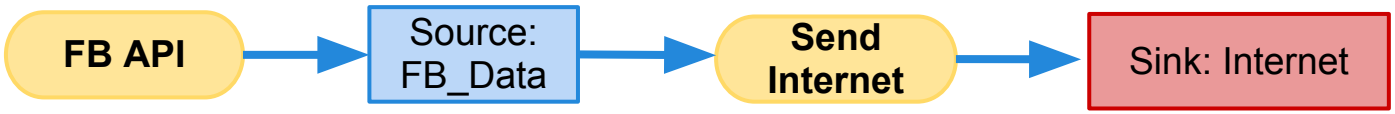
Data Flow



- Source-to-sink flows
  - Sources: Location, Calendar, Contacts, Device ID etc.
  - Sinks: Internet, SMS, Disk, etc.

Data Flow Analysis in Action

- Malware/Greyware Analysis
  - Data flow summaries enable enterprise-specific policies
- API Misuse and Data Theft Detection



- Automatic Generation of App Privacy Policies
  - Avoid liability, protect consumer privacy

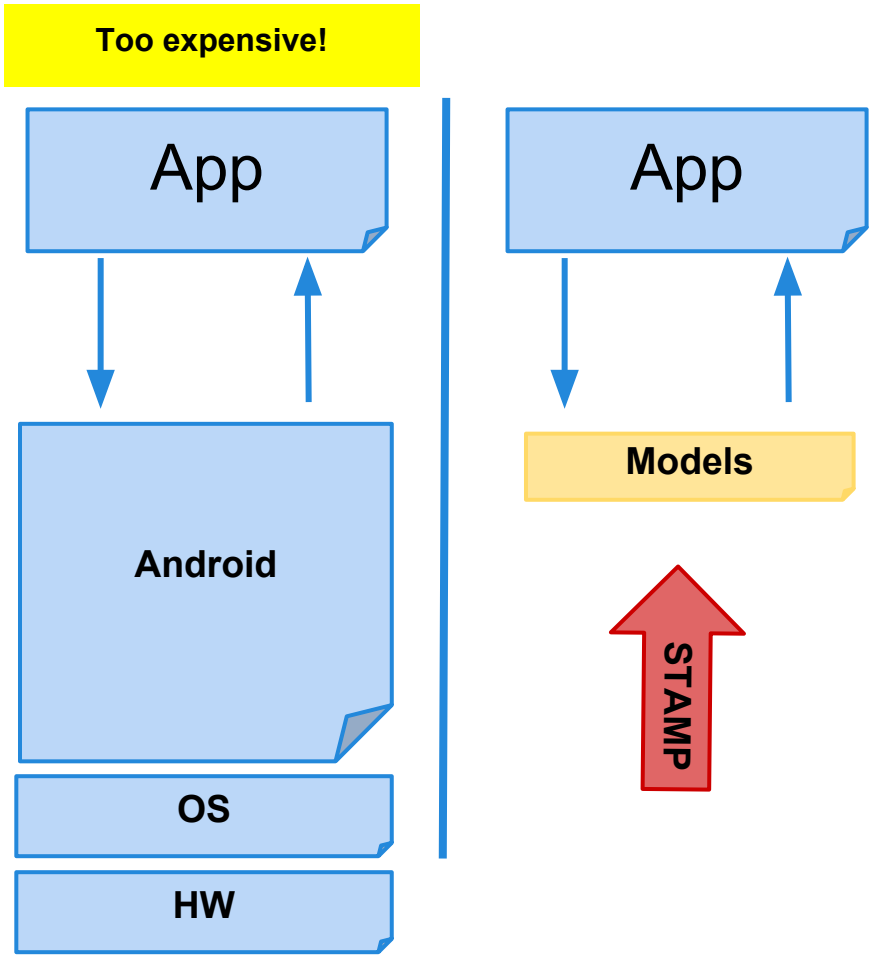
Privacy Policy
This app collects your:
Contacts
Phone Number
Address

- Vulnerability Discovery

Challenges

- Android is 3.4M+ lines of complex code
  - Uses reflection, callbacks, native code
- **Scalability:** Whole system analysis impractical
- **Soundness:** Avoid missing flows
- **Precision:** Minimize false positives

STAMP Approach



- Model Android/Java
  - Sources and sinks
  - Data structures
  - Callbacks
  - 500+ models
- Whole-program analysis
  - Context sensitive

Building Models

- 30k+ methods in Java/Android API
  - Reimplement w. minimum necessary details

- Follow the permissions
  - 20 permissions for sensitive sources
    - ACCESS_FINE_LOCATION (8 methods with source annotations)
    - READ_PHONE_STATE - (9 methods)
  - 4 permissions for sensitive sinks
    - INTERNET, SEND_SMS, etc.

Identifying Sensitive Data

```
android.Telephony.TelephonyManager: String getDeviceId()
```

- Returns device IMEI in String
- Requires permission GET_PHONE_STATE

```
@STAMP(
   SRC ="$GET_PHONE_STATE.deviceid",
   SINK ="@return"
)
```

Data We Track (Sources)

- Account data
- Audio
- Calendar
- Call log
- Camera
- Contacts
- Device Id
- Location
- Photos (Geotags)
- SD card data
- SMS

30+ types of sensitive data

Data Destinations (Sinks)

- Internet (socket)
- SMS
- Email
- System Logs
- Webview/Browser
- File System
- Broadcast Message

10+ types of exit points

Currently Detectable Flow Types

396 Flow Types

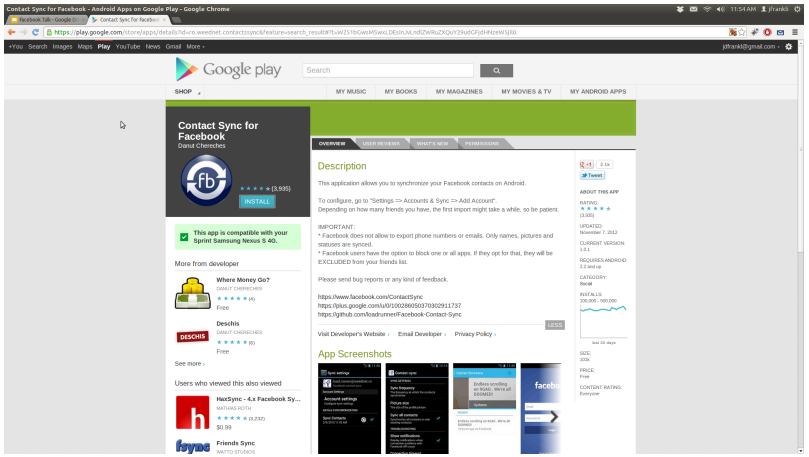
Unique Flow Types = Sources x Sink

Example: Facebook Contact Sync

Contact Sync for Facebook (unofficial)

Description:

This application allows you to synchronize your Facebook contacts on Android.



IMPORTANT:

* "Facebook does not allow [sic] to export phone numbers or emails. Only names, pictures and statuses are synced."

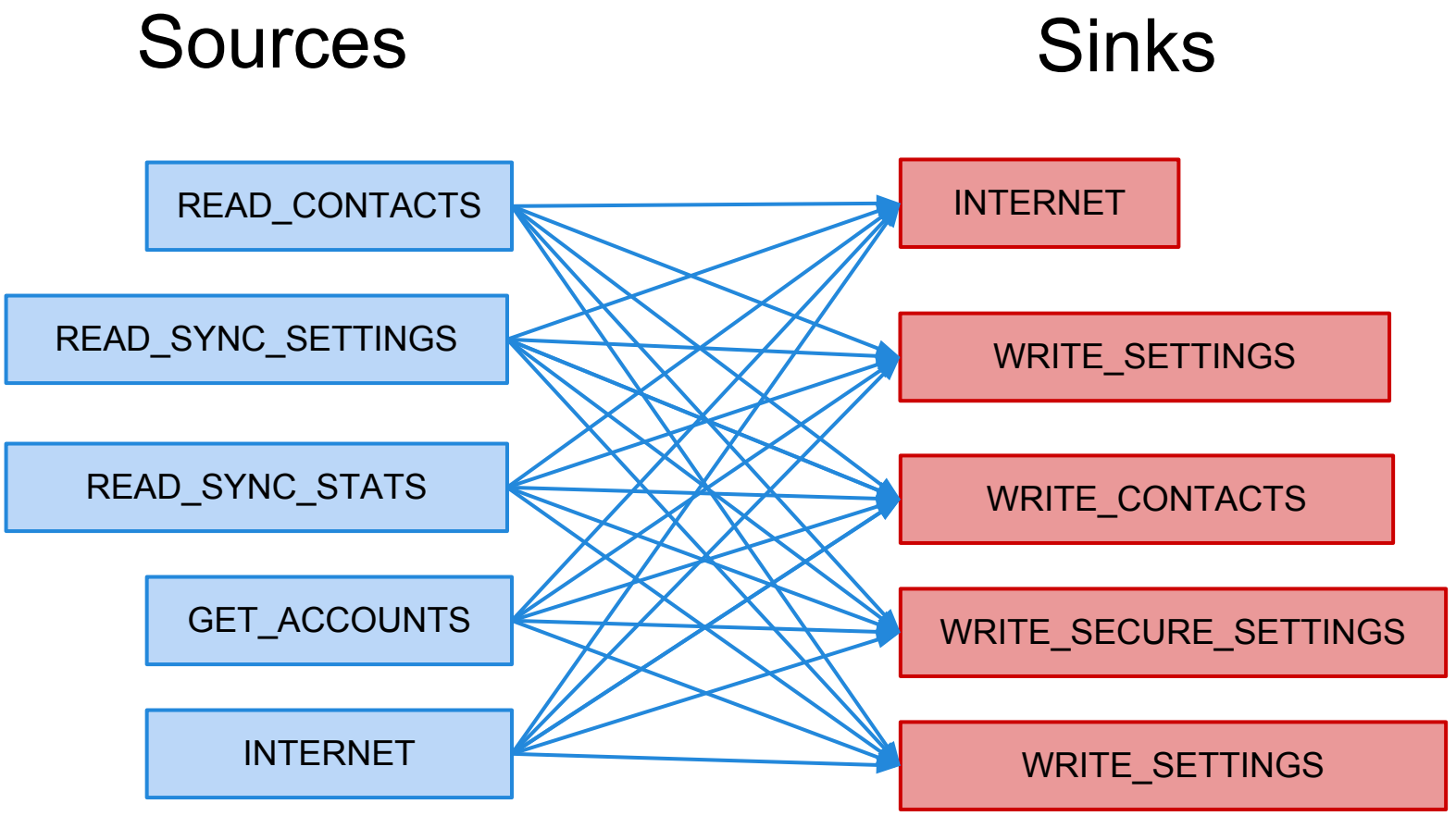
* "Facebook users have the option to block one or all apps. If they opt for that, they will be EXCLUDED from your friends list."

Privacy Policy: (page not found)

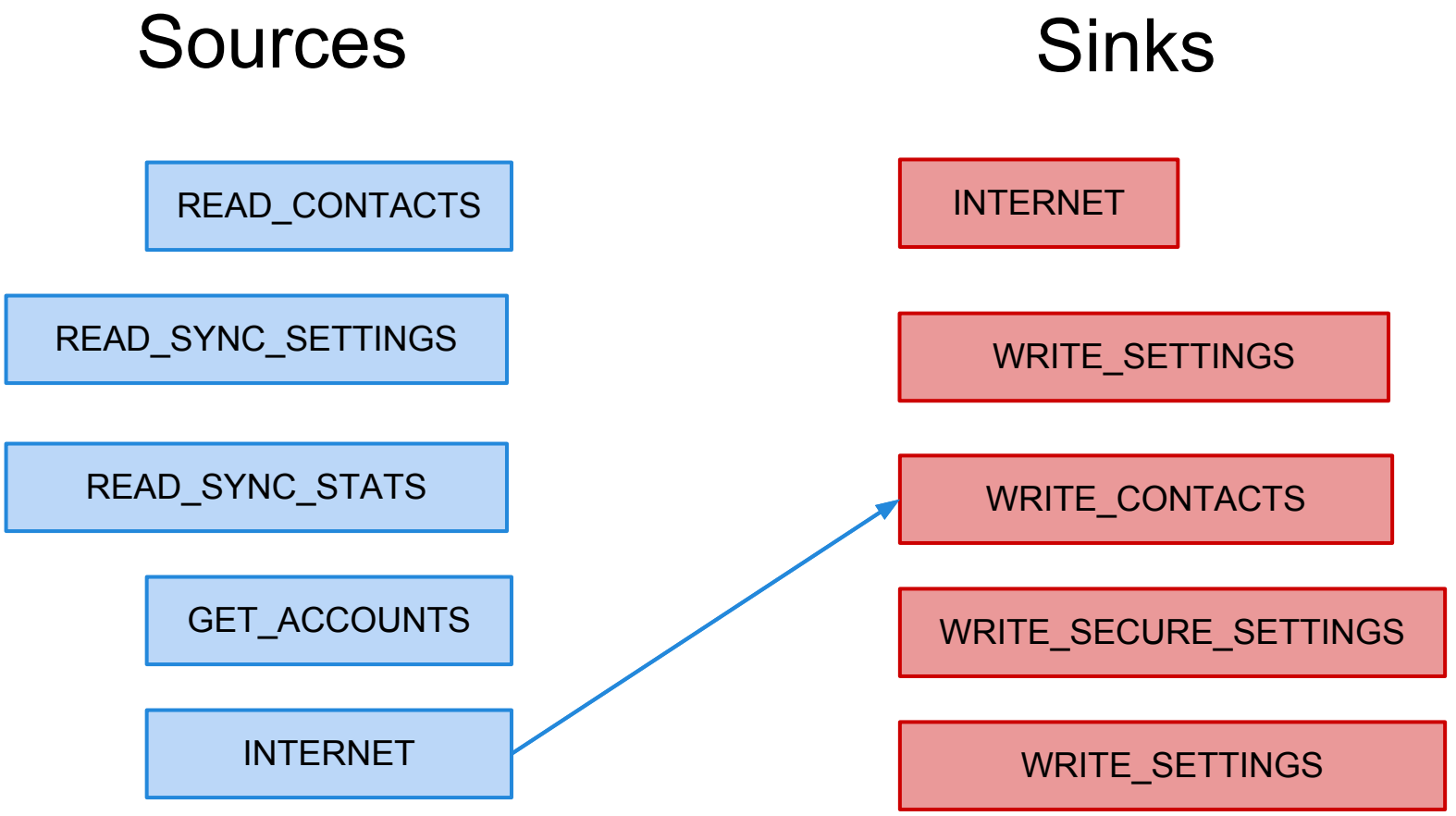
Contact Sync Permissions

Category	Permission	Description
Your Accounts	AUTHENTICATE_ACCOUNTS	Act as an account authenticator
	MANAGE_ACCOUNTS	Manage accounts list
	USE_CREDENTIALS	Use authentication credentials
Network Communication	**INTERNET**	**Full Internet access**
	ACCESS_NETWORK_STATE	View network state
Your Personal Information	**READ_CONTACTS**	**Read contact data**
	WRITE_CONTACTS	Write contact data
System Tools	WRITE_SETTINGS	Modify global system settings
	WRITE_SYNC_SETTINGS	Write sync settings (e.g. Contact sync)
	READ_SYNC_SETTINGS	Read whether sync is enabled
	READ_SYNC_STATS	Read history of syncs
Your Accounts	GET_ACCOUNTS	Discover known accounts
Extra/Custom	WRITE_SECURE_SETTINGS	Modify secure system settings

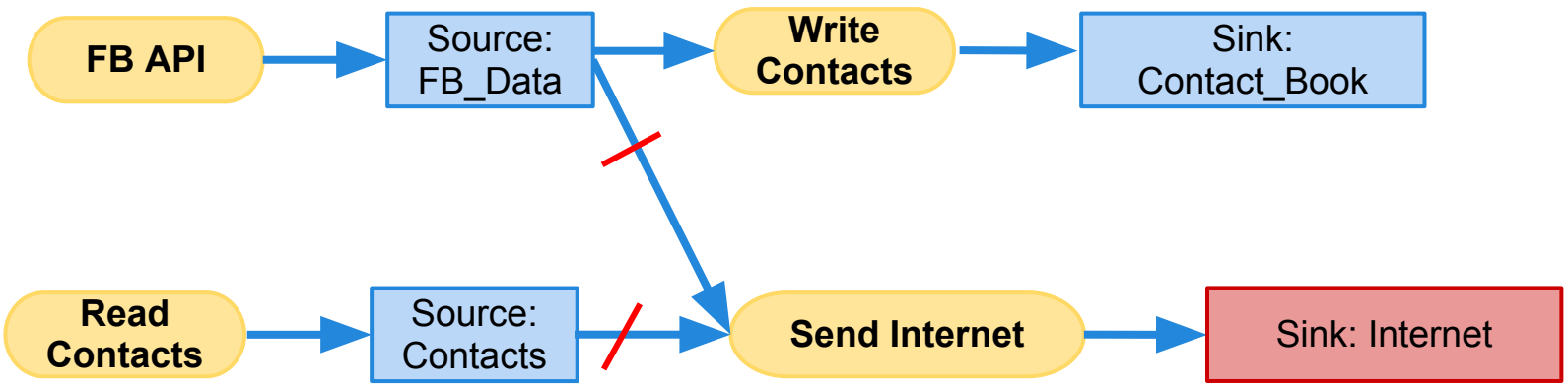
Possible Flows from Permissions



Expected Flows



Observed Flows



Conclusion

- Exploring space of admission systems
- Fast, practical static data flow analysis
- Dynamic analysis collects concrete values
  - Users test drive apps, we collect data
- Warning system identifies violated assumptions
  - Dynamic code loading, reflection, and anti-analysis techniques

Interested in STAMP?

Contact:
Jason Franklin
jfrankli@cs.stanford.edu

Credits:
Alex Aiken,
John Mitchell,
Saswat Anand,
Osbert Bastani,
Lazaro Clapp,
Patrick Mutchler,
Manolis Papadakis