Flexible and Fine-Grained Mandatory Access Control on Android for Diverse Security and Privacy Policies

Sven Bugiel

Joint work with Stephan Heuser and Ahmad-Reza Sadeghi

Cryptography and Information Security Group
Saarland University

System Security Lab
Technische Universität Darmstadt

Cyber-physical Systems Security
Fraunhofer SIT, Darmstadt

UNIVERSITÄT DES SAARLANDES

CISPA Center for IT-Security, Privacy and Accountability

System Security Lab

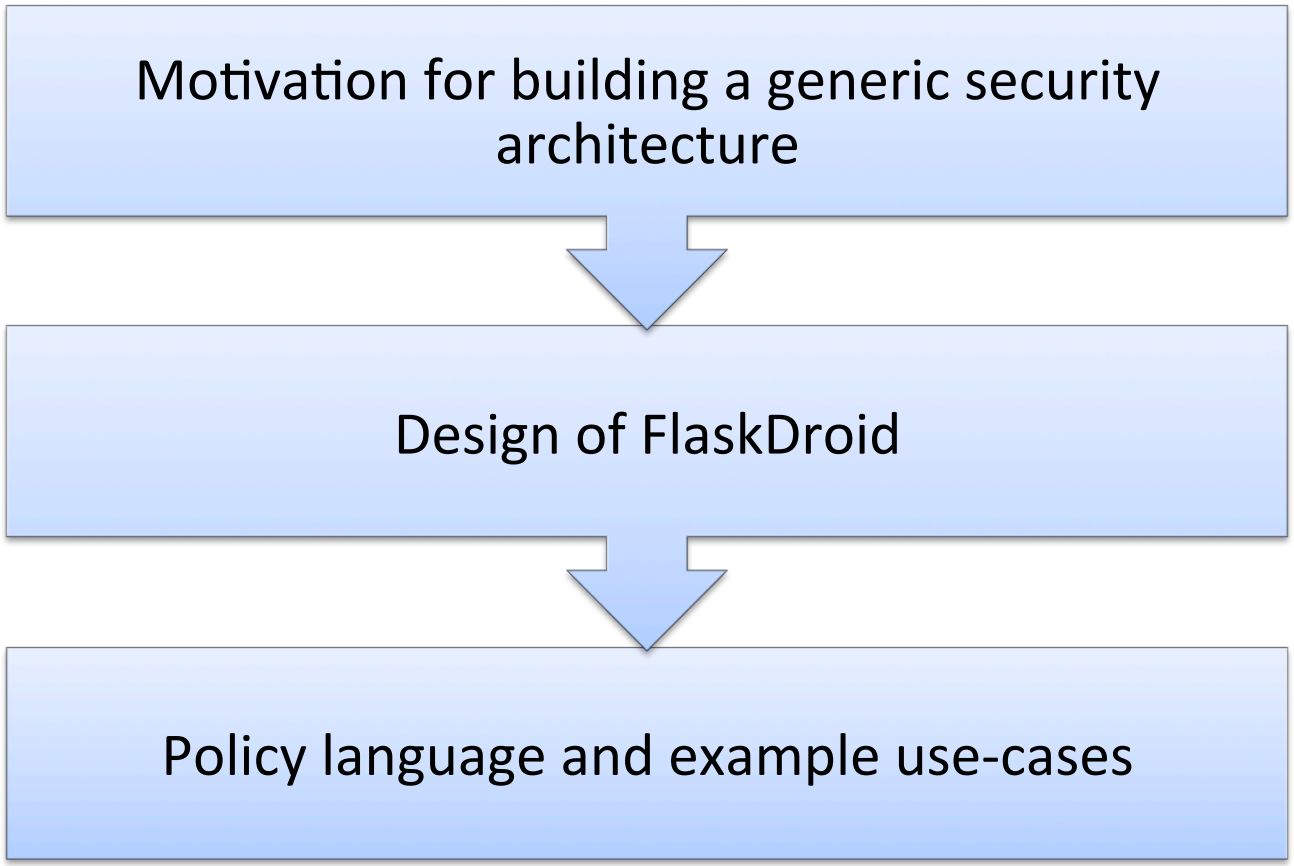
Fraunhofer SIT

TECHNISCHE UNIVERSITÄT DARMSTADT

CASED

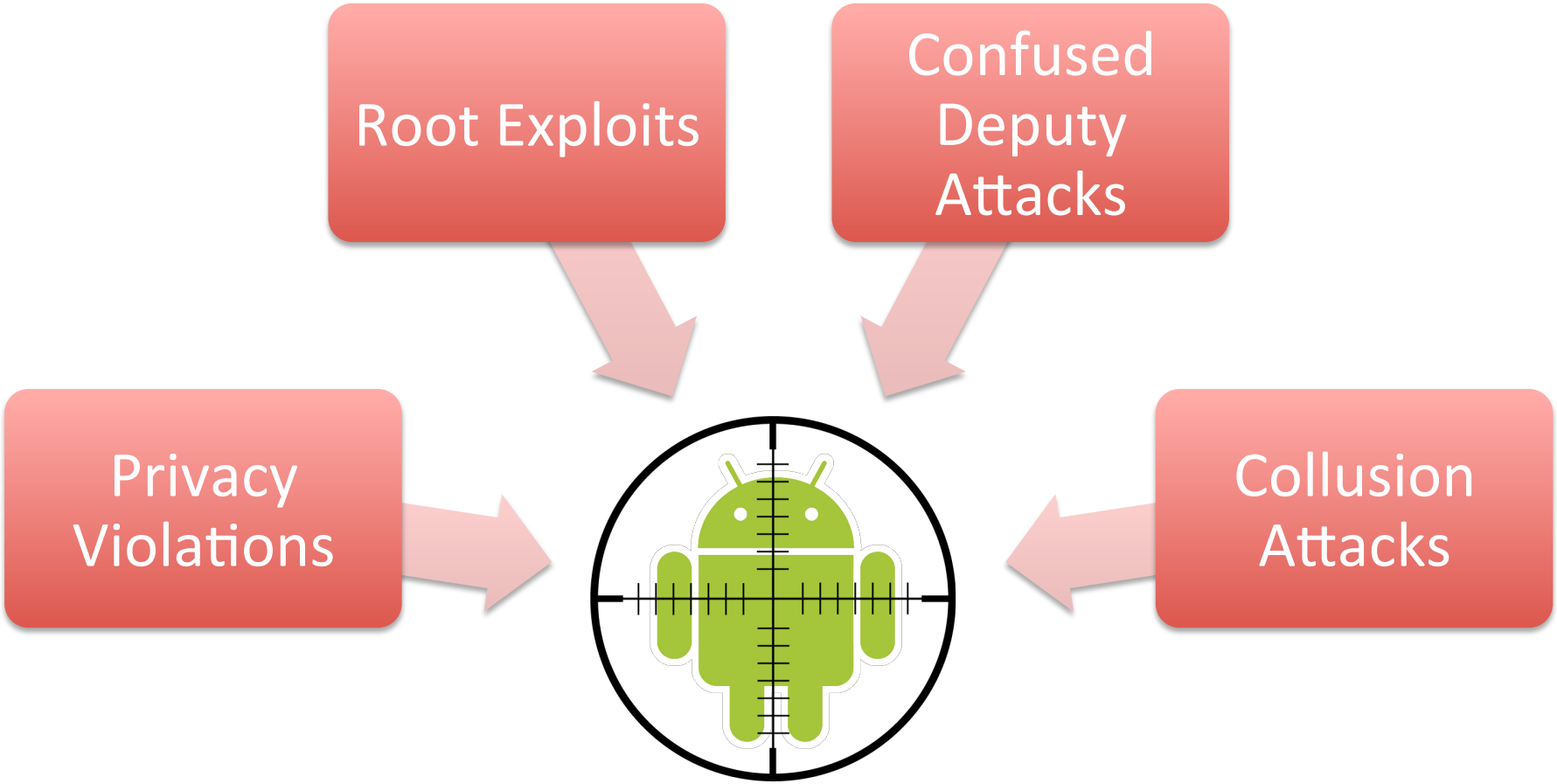
This talk is about:

FlaskDroid: Generic security architecture for Android



Android's Security Architecture Shown To Be Insufficient

Example Attacks on Android



Academic Security Extensions

http://1.androidauthority.com/wp-content/uploads/2012/03/security-notice-your-app-is-being-monitored.jpg

Key Observations

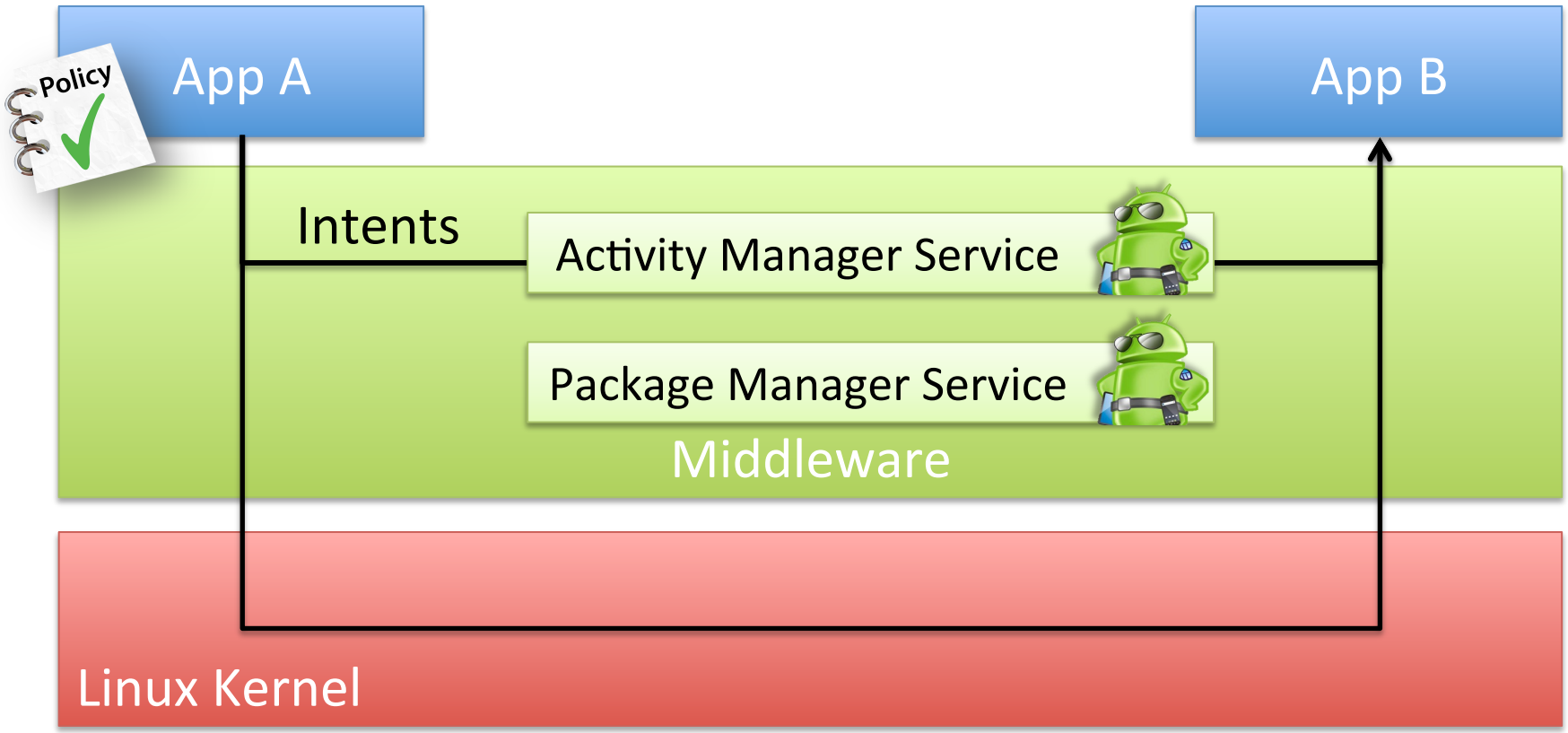
OBSERVATION #1:
MOST SECURITY EXTENSIONS ARE MANDATORY ACCESS CONTROL SOLUTIONS TAILORED TO A SPECIFIC PROBLEM

Saint

[Ongtang et al., 2009]

Runtime policy:

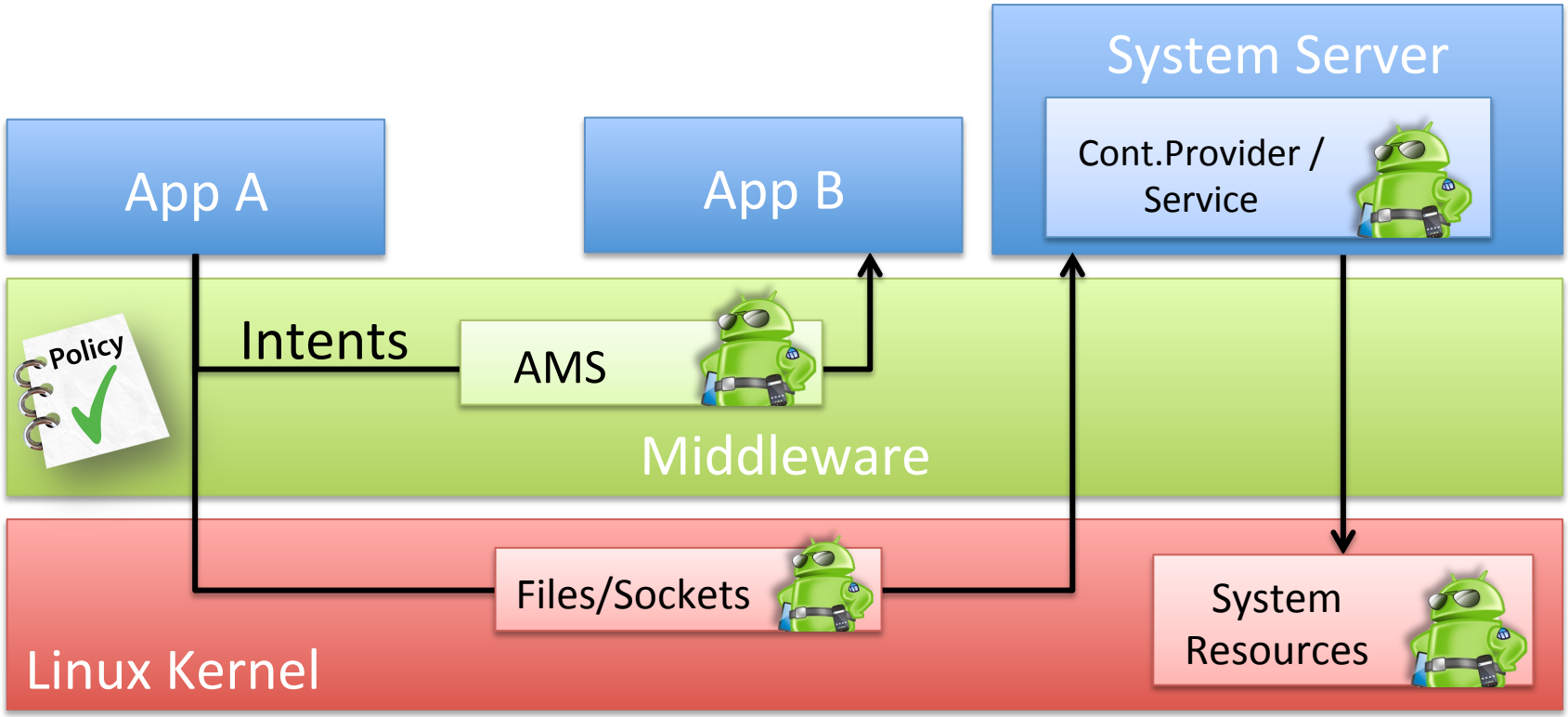
(expose|access) (source app, type, action) (destination app, component) conditions



XManDroid / TrustDroid

[Bugiel et al., 2012/2011]

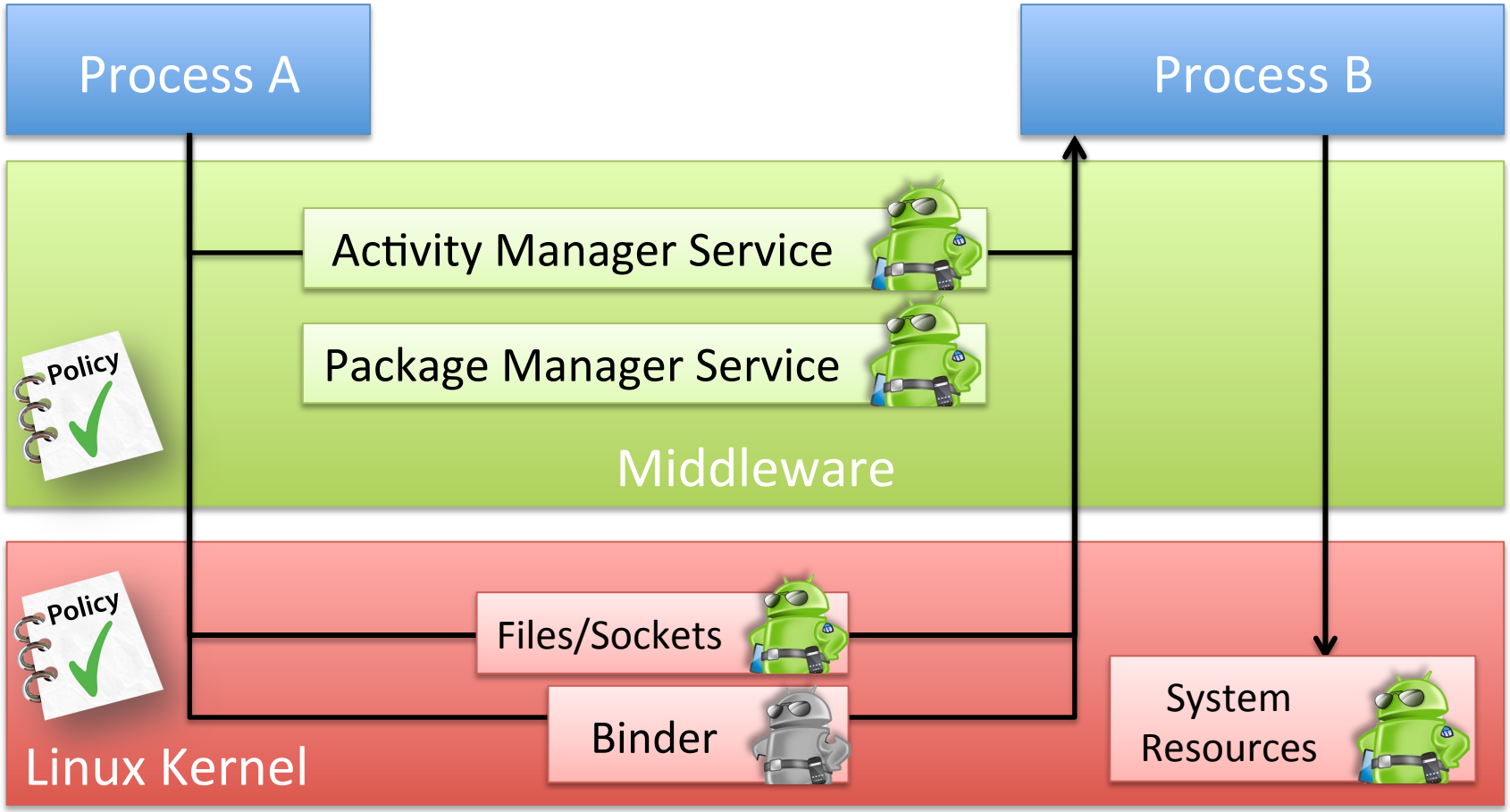
VALID policy language with Android-specific extensions



SE Android

[Smalley and Craig, 2013]

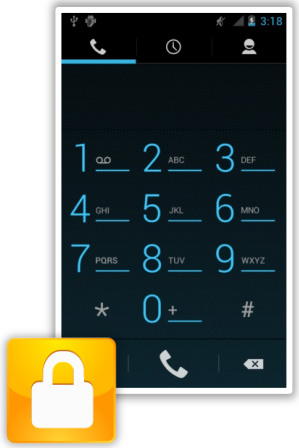
SELinux policy language (kernel) and MMAC extensions (middleware)



Nice to have: Policy-driven instantiations of use-cases

Existing work
(e.g. Saint, TrustDroid,…)

Generic security architecture

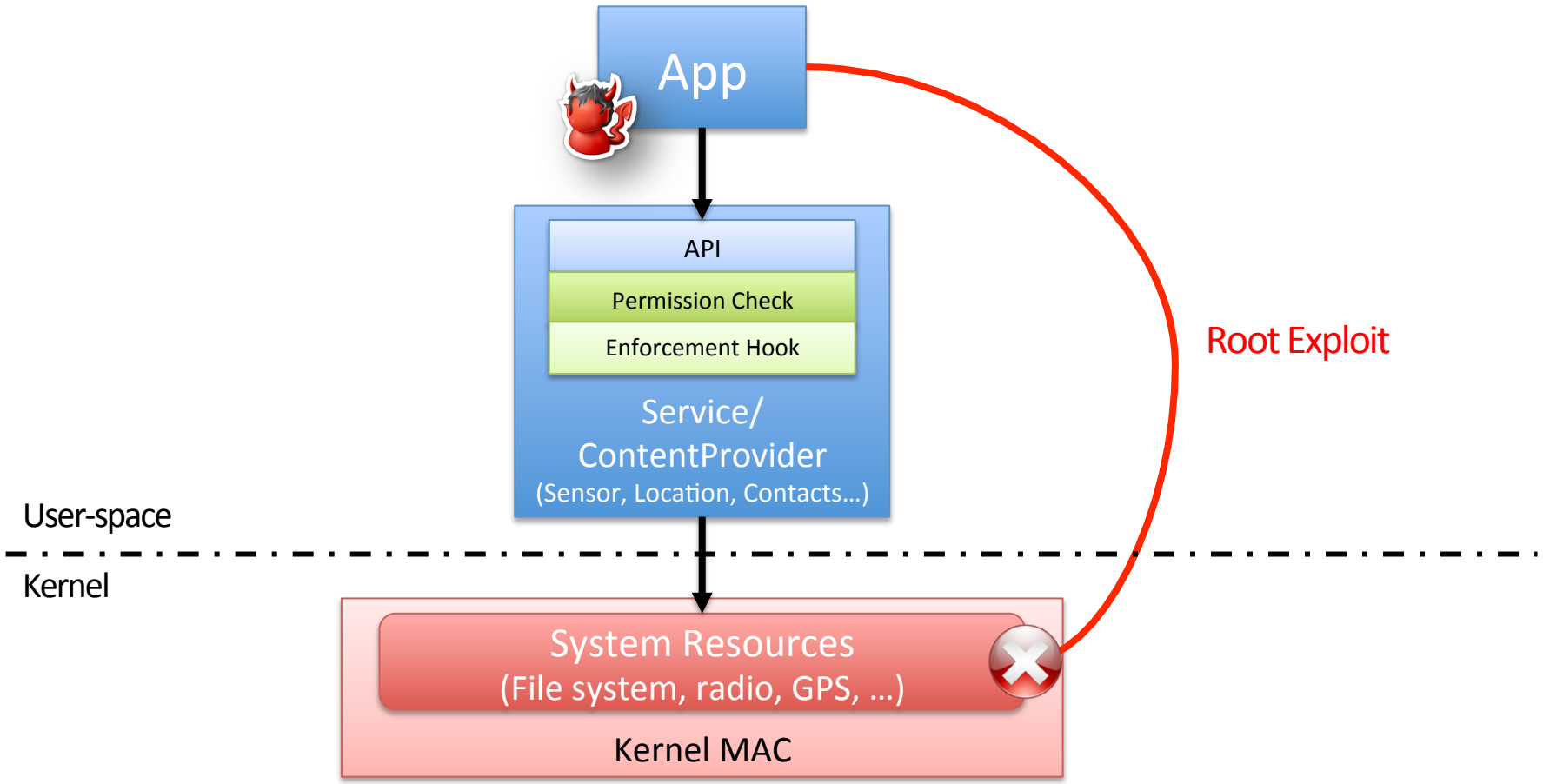


New use-cases
(e.g. Phone Booth Mode)

OBSERVATION #2:

ACCESS CONTROL REQUIRED AT BOTH USER-SPACE AND KERNEL LEVEL

Observation #2:
Access Control required at user-space and kernel level



FlaskDroid

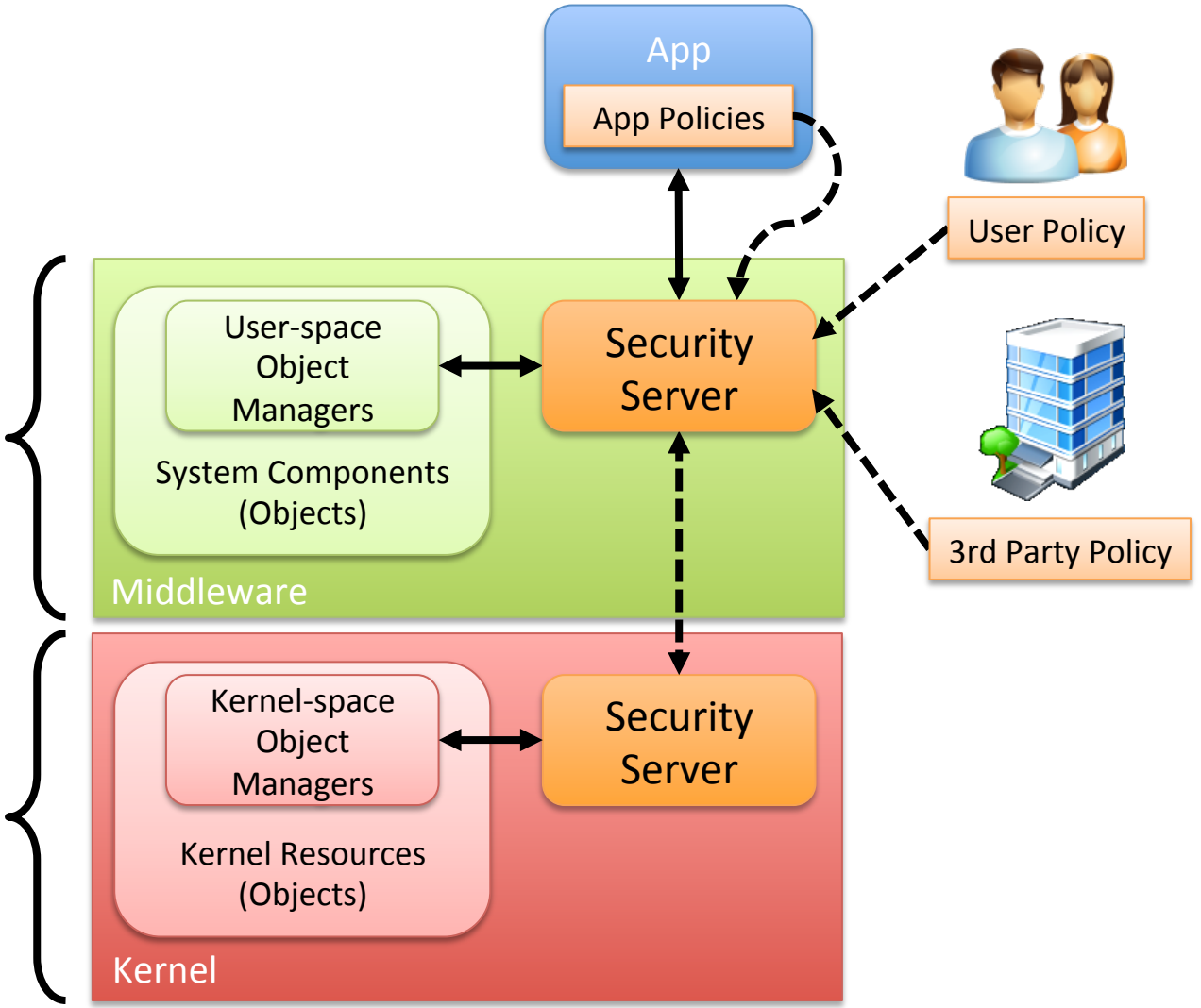
Main Contributions

- System-wide security framework operating on both middleware and kernel layer
- Policy language specifically designed for the rich semantics at middleware layer
- Policy-driven instantiations of use-cases and related work

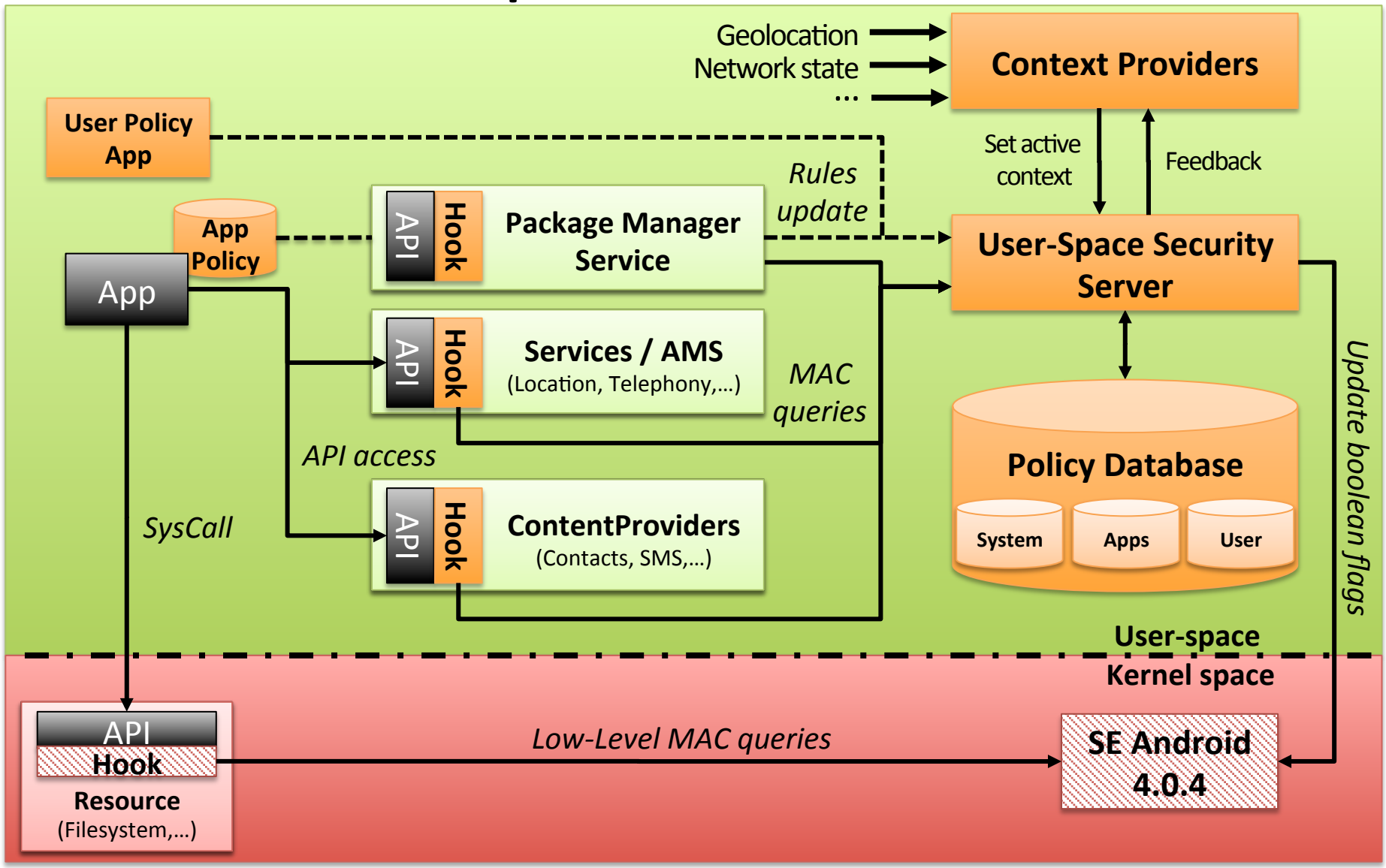
Design

- Dynamic, system-state aware policies
- Support multiple stakeholder

- Preserve security invariants (e.g., no root)
- Low-level enforcement in alignment with middleware



Implementation



Policy Language

```
type android_t;
type contacts_email_v2_t;
```

Type defintions for Type Enforcement

```
class contentProvider_c
{
  query insert update delete
  readAccess writeAccess
};

class contactsProvider_c
inherits contentProvider_c;
```

New classes for middleware-specific objects

Policy Language (cont.)

```
bool phoneBooth_b = false;
kbool app_network;
```

Boolean definitions for middleware and kernel

```
if(~phoneBooth_b)
{
 allow app_telephony_t any: contactsProvider_c {query};
};
```

Conditional policies

```
context phoneBooth_con;
switchBoolean
{
    context=phoneBooth_con;
    auto_reverse=true;
    phoneBooth_b=true;
};
```

Context definitions and mapping to boolean values

Policy Language (cont.)

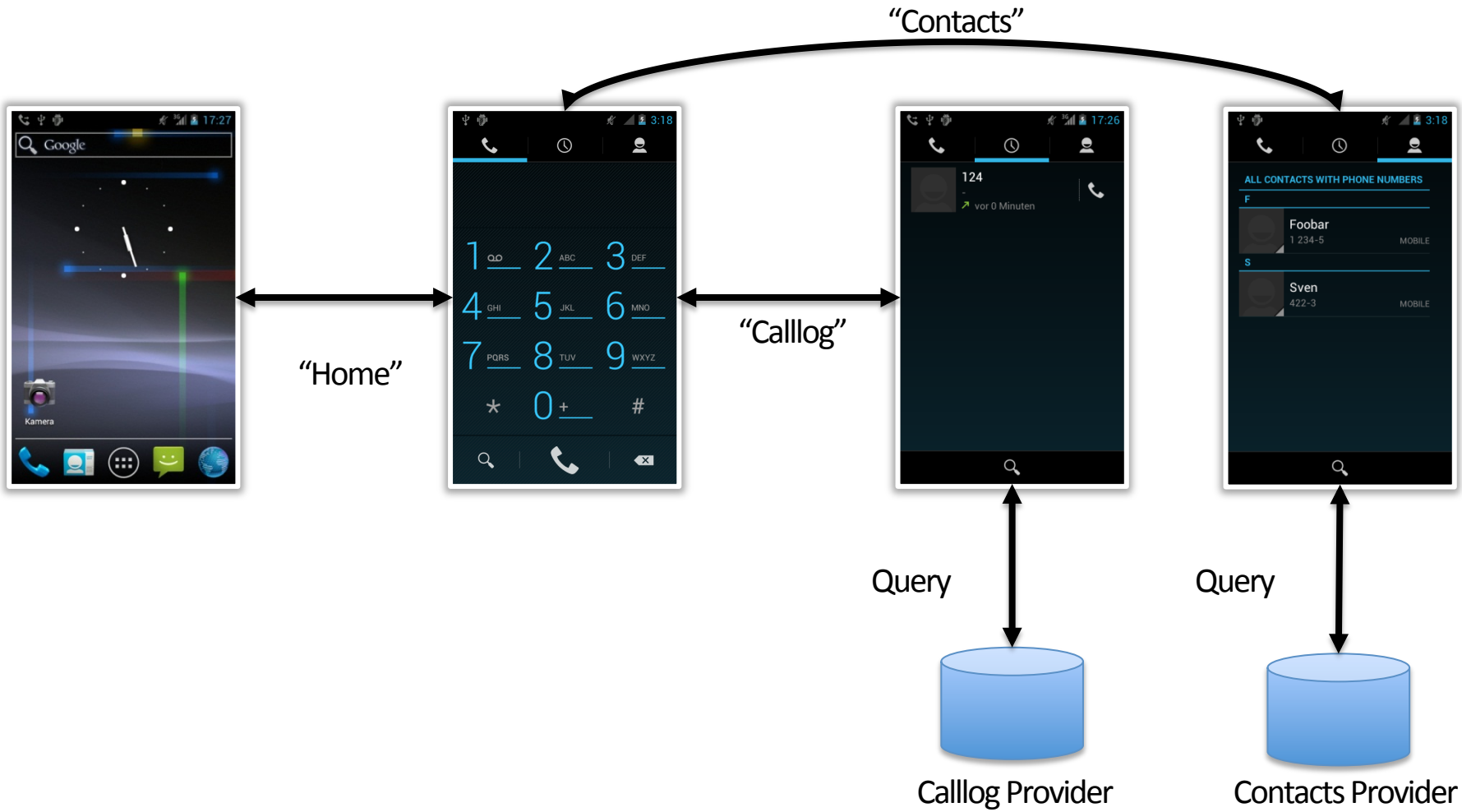
```
appType app_contacts_t
{
  Package:package_name=com.android.contacts;
};

intentType intentLaunchHome_t
{
  Action:hasAction=android.intent.action.MAIN;
  Categories:hasCategory=android.intent.category.HOME;
};
```

Metrics for dyanmically assigning application and Intent types

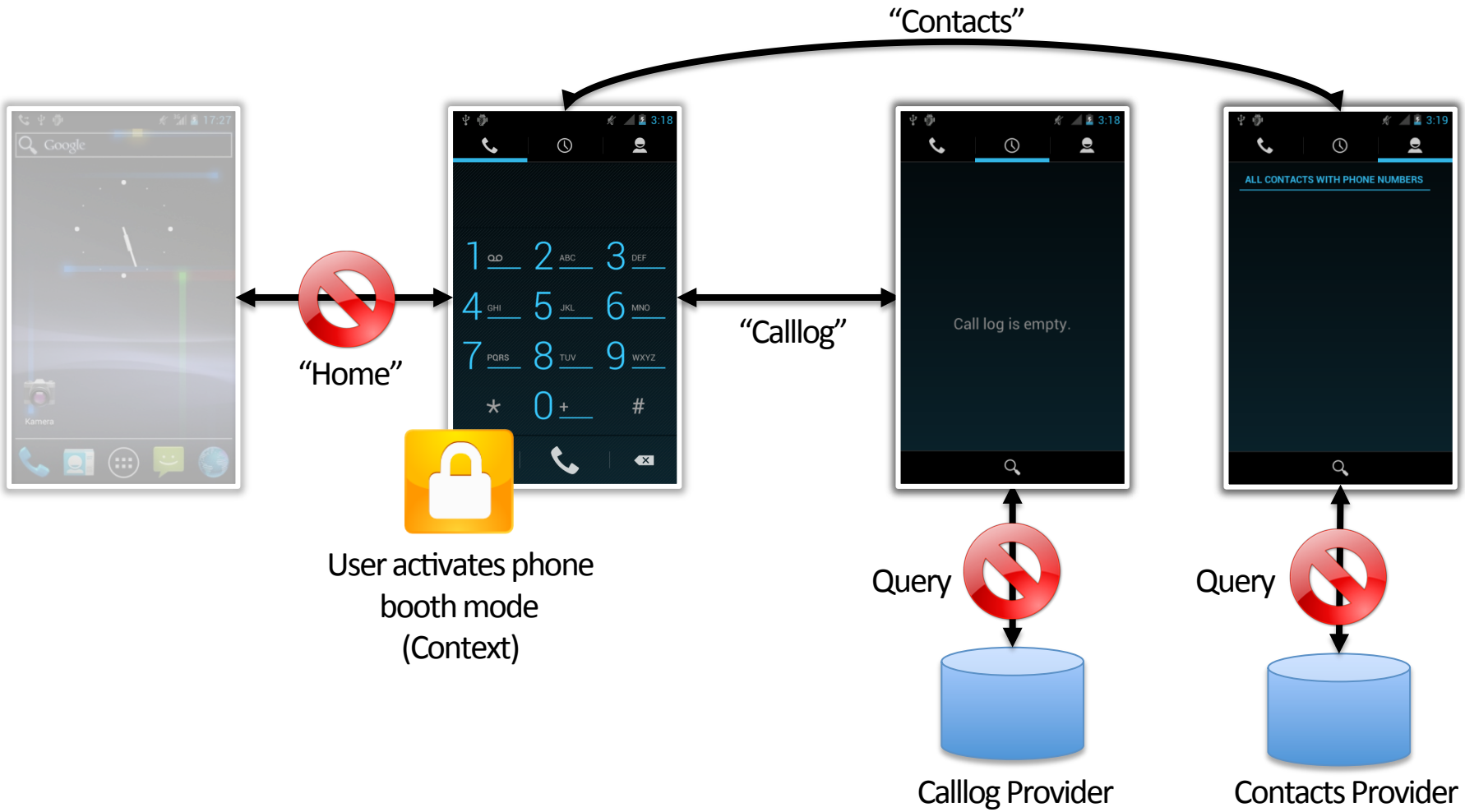
Base Policy

Policy	#Types	#Classes	#Rules
FlaskDroid Middleware MAC (base policy from 12/04/2012)	111	18	109
SE Android 4.0.4 (master branch, 12/04/2012)	232	84	1359
SELinux Fedora 17 (targeted, policy.27 from 12/04/2012)	3900	83	103235

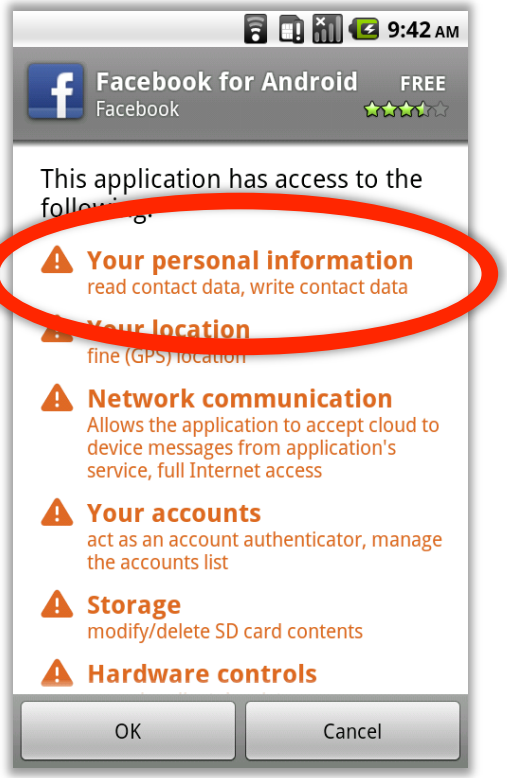
Use-case: “Phone booth mode”

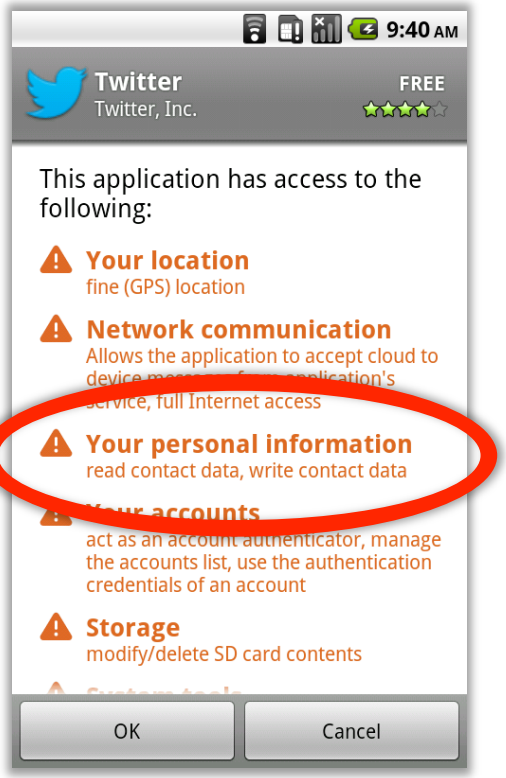


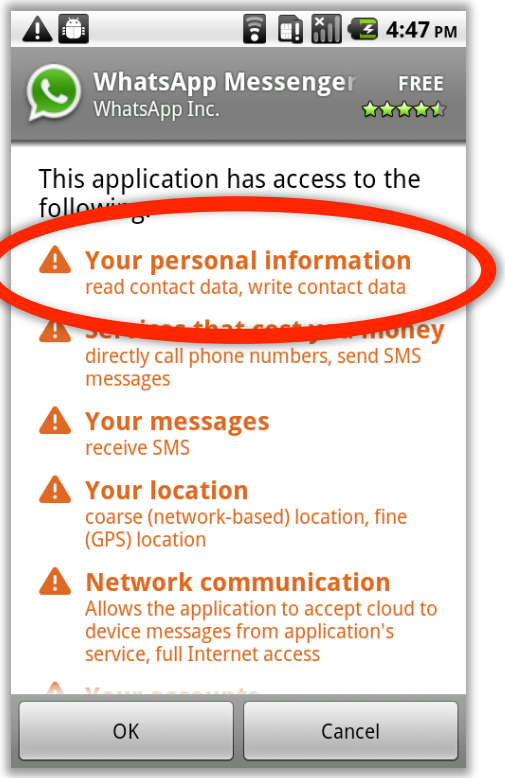
Use-case: “Phone booth mode”



Use-case: Fine-grained Access to Contacts Data





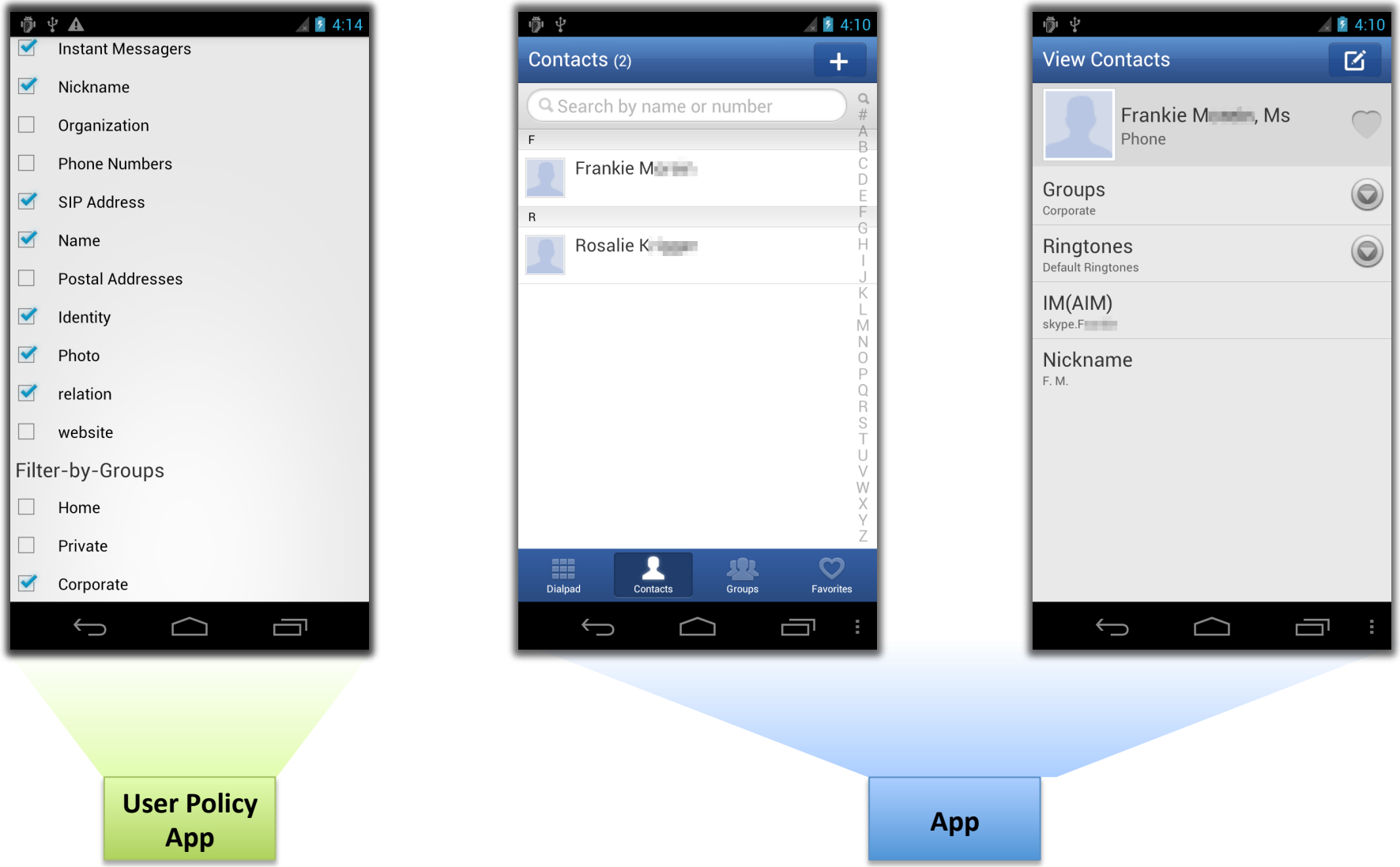


Do these apps *really* need *all* my contacts data?

Or are just the telephone numbers or email addresses enough?

Use-case: Fine-grained Access to Contacts Data

Use-case: Fine-grained Access to Contacts Data



Further use-cases

- *App developer policies*
  (Saint)
- *Secure integration of higher privileged 3rd party apps*
  (Firewall and Anti-Virus apps, no root required)
- *Multi-level security*
  (private vs. business domain)
- *Context-aware policies*
  (prevent reading sensor data while keyboard in foreground)

Quo Vadis?

- Port to SE Android 4.3
  - Integration with SE Android MMAC
- Towards *completeness*
  - Static analysis of API for hook placement
  - Formal analysis of policy subspaces
- More fine-grained types
  - Currently assigned to application sandboxes

Thank You!
Questions?

www.flaskdroid.org

References

- **[Ongtang et al., 2009]** M. Ongtang, S. McLaughlin, W. Enck, and P. McDaniel, *"Semantically Rich Application-Centric Security in Android,"* in ACSAC 2009
- **[Bugiel et al., 2011]** S. Bugiel, L. Davi, A. Dmitrienko, S. Heuser, A.-R. Sadeghi, and B. Shastry, *"Practical and lightweight domain isolation on Android,"* in SPSM 2011
- **[Bugiel et al., 2012]** S. Bugiel, L. Davi, A. Dmitrienko, T. Fischer, A.-R. Sadeghi, and B. Shastry, *"Towards taming privilege-escalation attacks on Android,"* in NDSS 2012
- **[Smalley and Craig, 2013]** S. Smalley and R. Craig, "*Security Enhanced (SE) Android: Bringing Flexible MAC to Android*," in NDSS 2013.