Component-based Decoupling of Mobile Applications using RemoteBinder

Yutao Liu ***Yubin Xia*** ***Haibo Chen***

{ytliu.cc, xiayubin, haibochen}@sjtu.edu.cn

INSTITUTE OF PARALLEL AND DISTRIBUTED SYSTEM

http://ipads.se.sjtu.edu.cn

Motivation:

Enhance mobile security utilizing mobile cloud computing

* **Key issue:** make cloud and mobile cooperate with each other
  - Approach 1: Running **replica** of mobile image in cloud[1].
    * Too strict consistency, consume high network bandwidth
  - Approach 2: Function-level[2][3]/instruction-level[4] **application partition**.
    * Not transparent for application, or require tight coupling

Observation:

Android framework for application development

* Adopt a **component-based** approach
  - Each application consists of multiple **loosely coupled** components
    * Service, Activity, Broadcast Receiver, Content Provider
  - Each component communicate with others using Android binder mechanism

Mobile malware characteristics

* Malicious code usually runs in **background Service**
  - Most malware is repacked, Services are loosely coupled with others
  - Service is the common component to run maliciously without users noticing

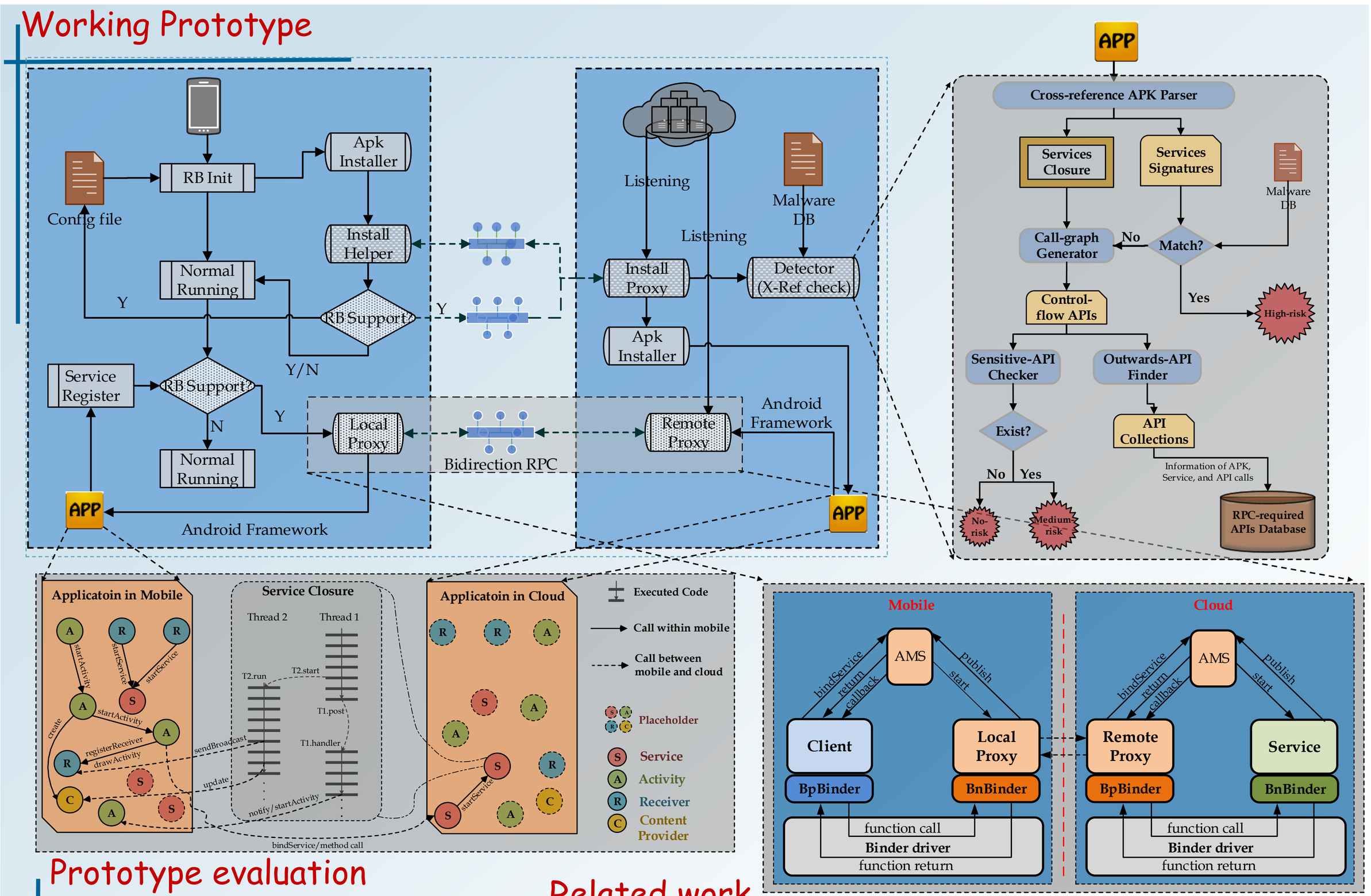
Solution:

A transparent Service component-based decoupling approach

* An **integrated** mobile-cloud protection **framework**
  - Define the whole protection process in mobile and cloud
* **Cross-reference** tool for static analysis
  - Find the **closure** of a Service and all of its **outward API** invoking points
* **Behavior**-based suspicious Service **detector**
  - Detect Service as suspicious based on its behavior
* **Proxy**-based mechanism for **bidirectional RPC**
  - Communicate the decoupled Service in cloud with other components in mobile
* **Asymmetric** protection policies in mobile and cloud
  - Lightweight protection in mobile and stronger protection in cloud for suspicious Services

Our solution requires Android framework modification: including AMS, apk installer, permission system, etc.

Working Prototype



Prototype evaluation

* **Environment Setup**
  - Samsung Galaxy Nesus with Android 4.2
  - Cloud emulator in Debian 6.0
* **Prevent 7 types of malicious behavior**
  - C&C Server, SMS Trojan, etc.
* **Performance evaluation**
  - Remote method call delay is 45ms
  - Extra network data is less than 100 bytes/RPC

Related work

* **Replica-based mobile cloud computing for security**

[1] PORTOKALIDIS, G., HOMBURG, P., AND ANAGNOSTAKIS, K. Paranoid Android: Versatile protection for smartphones. In ACSAC (2010).

* **Application partition approaches for mobile cloud computing**

[2] CHUN, B. G., IHM, S., MANIATIS, P., NAIK, M., AND PATT, A. CloneCloud : Elastic Execution between Mobile Device and Cloud. In Eurosys (2011).

[3] CUERVO, E., BALASUBRAMANIAN, A., CHO, D., WOLMAN, A., SAROIU, S., CHANDRA, R., AND BAHL, P. MAUI: Making smart- phones last longer with code offload. In Mobisys (2010).

[4] GORDON, M. S., JAMSHIDI, D. A., MAHLKE, S., MAO, Z. M., AND CHEN, X. COMET : Code Offload by Migrating Execution Transpar- ently. In OSDI (2012).