Realizing API Virtualization on Android
Taeyeon Ki, Alexander Simeonov, Bhavika Pravin Jain, Chang Min Park, Keshav Sharma
Karthik Dantu, Steven Y. Ko, Lukasz Ziarek

I. Motivation

Mobile vendors control platform distribution channels.

Unlike mobile platforms, mobile apps are open for third parties.

II. Propose

- Injecting a replacement class that contains custom code for each platform API class that a third-party developer wants to replace
- Rewriting the binary of an app so that the app code uses replacement classes instead of platform API classes
- Distributing the new implementation through standard app distribution channels

III. Contribution

- Mitigating the lack of openness in mobile systems by proposing API Virtualization
- Exploring and addressing a unique set of challenges that API Virtualization brings in order to correctly and completely handle all features of Android and Java.
- Realizing API Virtualization prototype, and showing its feasibility and practicality

IV. Initial Results

- Samsung Galaxy Nexus Devices running Android 4.4
- Enabling stay-awake mode
- Set CPU governor to “Performance”

V. Use Cases

1. Vendor-Tied Library Switching: Google Maps to Amazon Maps

2. Runtime Permissions: Runtime Internet Permission

VI. Conclusion

API Virtualization enables open innovation in Android. Also, it allows third-party developers to inject custom code into an app binary. Through API virtualization prototype on Android, we show the feasibility and practicality of our API virtualization, and the low overhead that API virtualization imposes.