Safe Execution of Dynamically Loaded Code on Mobile Devices

Author: Glen Pink, gpin7031@uni.sydney.edu.au
Supervisor: Assoc. Prof. Bob Kummerfeld
School of Information Technologies

1. Aims of the Project

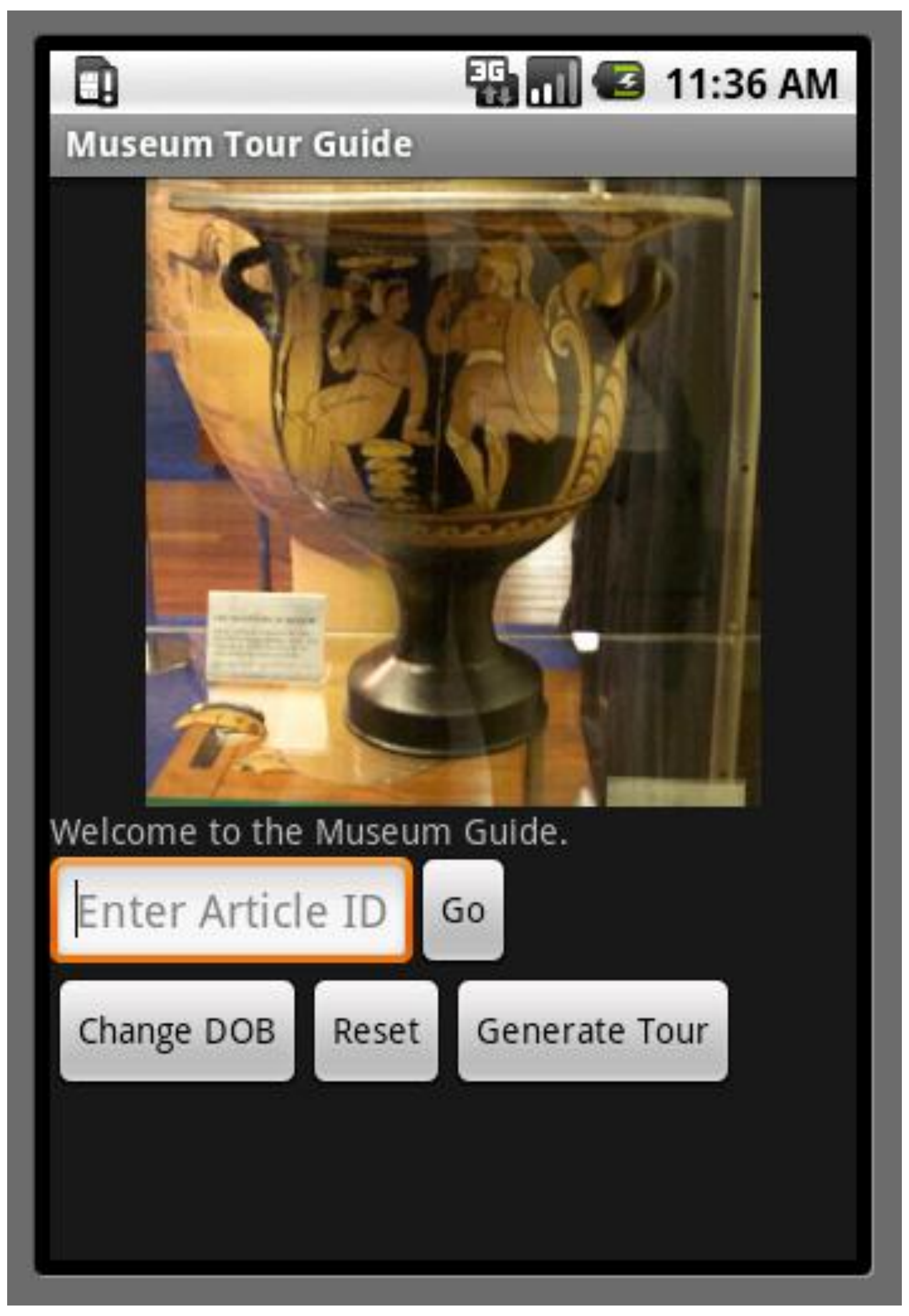
- To provide a framework for the safe dynamic loading of code onto mobile devices.

2. Introduction

As the use of smart phones grows, so does the potential for personalised, context aware applications on mobile devices. We can personalise, transfer and display data for a wide variety of contexts.

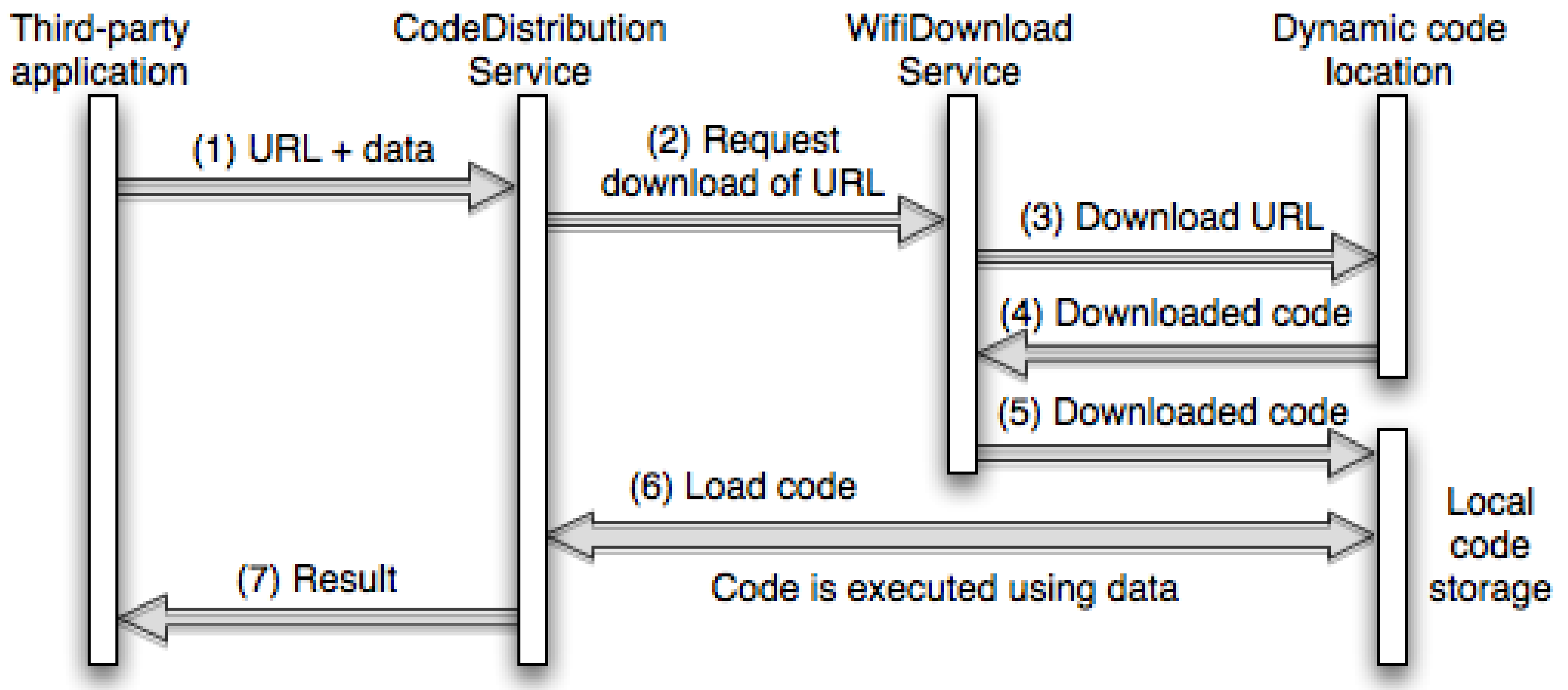
The optimal personalisation of data requires the use of potentially private data. In addition to this, any personalisation code must traditionally be run on servers and be verified to be safe (not malicious) before being executed.

However, if we allow code to be dynamically loaded and executed safely on to mobile devices we can avoid these problems and allow for greater personalisation to take place. For example, a museum tour (pictured) could be personalised on a person's device by using their private data on their mobile device.



11:36 AM
Museum Tour Guide
Welcome to the Museum Guide.
Enter Article ID

This project proposes a framework for the dynamic loading of code onto a mobile device while preventing dynamically loaded code from attacking or inadvertently damaging the device.



3. Android

This framework is built on the Android platform. Android applications are written in Java, and Android itself is based on Linux. Android provides a solid security foundation on which to build this safe loading framework. Applications are provided limited access to phone functionality by the Android permission mechanism. If an application requires a particular piece of phone functionality (such as the ability to make phone calls) it must explicitly request access to this at install time.

While this does provide a good level of security, it is not sufficient if dynamic code is to be loaded. Significantly, to actually download a piece of dynamic code requires the *Internet* permission, which would allow any malicious code to freely send and receive data (including further dynamic code to load).

We must also prevent the dynamically loaded code from having unrestricted access to the data of the loading application (and other applications) and from accessing the memory of other process. Android prevents direct process memory modification, however processes may still communicate via "Intents" (broadcast messages) or via the Android Inter-Process Communication (IPC) mechanism. We may still want to provide dynamically loaded code with data provided by other sources on the device.

4. The Framework

The dynamic code loading framework provides two proxy layers for dynamic code loading. These layers communicate using the Android IPC mechanism.

WifiDownloadService (WDS) manages the storage of dynamic code. CodeDistributionService (CDS) manages the execution of code and provides an interface for third party applications. CDS has no Android permissions thereby preventing most malicious attacks from dynamic code

The way in which the system works is as follows (refer to diagram). An application wants to load a piece of code dynamically and has a URL giving the location of that code: perhaps acquired from a QR code (below). It also has data to pass to the dynamic code.

5. Demonstration Applications

Dynamic Locator: People trying to locate them scan QR codes which encode the location of the code and the name of the person. This downloads code to determine their location, this is executed on the device and the location of the individual is displayed.

Museum Guide: This application generates museum tours personalised to the user (using their private data) when they enter a museum.

6. Future Work

- To add the ability for dynamic code to have some Android permissions as opposed to none.
- To extend the framework to other mobile environments.

This research is supported in part by the Smart Services CRC

The University of Sydney