Application Linking | Common Infrastructure | SMS Push

Technology Reports

Special Articles on Services for Next-Generation Smartphones

Common Infrastructure for Android Applications

*The needs of the Android™[*1] market have been increasing rapidly, and we are receiving more and more requests to support NTT DOCOMO feature phone functions and services on Android. Starting with the 2011-2012 Winter-Spring models, we have used the flexibility of application development on Android to develop application linking and SMS Push functions, providing common infrastructure functionality on Android, and to ensure feature-phone class functions and services on Android.*

Communication Device Development Department
Daisuke Motooka
Satoshi Takahashi
Core Network Development Department
Miho Kikkawa
Sachiko Kichimi

1. Introduction

Android is now spreading very rapidly, and NTT DOCOMO is shifting the focus of our terminal lineup toward smartphones, including Android terminals. Android is open source, so it provides freedom and excellent flexibility. Ordinary users can develop applications and a platform[*2] for publishing them has been built. There have been requests to provide feature phone (i-mode terminal) functions on Android, so we have used the strengths of Android and developed common infrastructure functionality, to ensure the same level of functions and services on the Android platform as we have on i-mode terminals.

One such functionality is application linking, which enables NTT DOCOMO service applications to operate in cooperation with each other. Applications installed on NTT DOCOMO terminals incorporate a more flexible and powerful linking mechanism than ordinary Android applications.

The second functionality we have developed is for development of SMS Push functions. Carrier network service nodes can send control SMS[*3] messages (SMS Push) as a mechanism to perform operations or initiate processes on a terminal. We have developed an SMS Push function for smartphones in order to provide more dynamic services with higher added value.

In this article, we describe how this functionality was implemented.

2. Application Linking Functionality

In contrast to ordinary individual Android applications, the applications installed on NTT DOCOMO terminals are designed to operate in cooperation with various other NTT DOCOMO applications. Linking between standard Android applications is implemented using Intents[*4], but we have introduced more flexible and powerful forms of

*1 **Android™**: A software platform for smartphones and tablets consisting of an operating system, middleware and major applications. A trademark or registered trademark of Google Inc., United States.
*2 **Platform**: OS or operating environment for running applications.

linking between NTT DOCOMO service applications.

2.1 Ordinary Linking on Android

Android provides several methods for links between applications, including Intents, Service Binder[*5], socket communications[*6] and shared files. The features of application linking using each of these methods are shown in **Table 1**.

Here we will focus on application linking using Service Binder. This method is a type of inter-process communication using Linux®[*7] Remote Procedure Calls (RPC)[*8]. This mechanism allows an application (the local application) to call methods[*9] of a different application (the remote application) in the same way as it calls its own methods, without concern that it is a different application.

Service Binder security is maintained using Android's permissions[*10] and signatures[*11]. An application publishing a method with Service Binder defines its permissions in a manifest file[*12]. The application calling this remote method declares that it will use this permission in its own manifest file. Having both local and remote applications agree on the permissions can prevent links between applications that are not intended by the user. Also, levels of permission can be configured based on Android signatures. By configuring the level of permission, the remote side can specify which applications are permitted access. Various permission patterns, such as signature agreement or system-level authority, are provided.

2.2 Issues with the Ordinary Linking

With ordinary linking between Android applications, for a remote system to limit access to remote methods, it requires the local application to have a matching signature or be a system application. In contrast, NTT DOCOMO service applications are linked to various other NTT DOCOMO applications in a uniform way, so they can implement more-convenient, intuitive operation. These linked applications need to be able to cooperate smoothly in all states, but existing methods were not able to support all possible cases. We introduced our own application linking mechanism to link applications and maintain consistency even in these conditions.

2.3 Improved Application Linking

With the new application linking mechanism, we have separated the remote and local applications, and implemented a new method for linking safely. An overview of this new mechanism is shown in **Figure 1**.

The improved linking method implements safe access control within the existing Android framework, which is more flexible than standard Android methods. By conforming to existing development methods and extending only NTT DOCOMO specific components, we were able to develop these improvements efficiently. This achieved more flexible linking between applications and enabled us to build a platform for NTT DOCOMO's attractive service applications.

3. SMS Push

SMS Push is a mechanism that allows service nodes on the carrier's network to initiate operations or launch processes on a terminal. Carrier services can actively prompt users to take

Table 1 Application linking features in Android

Link method	Features
Intent	A simple linking method for one-way links such as message notifications
Service Binder	Allows definition of dedicated APIs and more complex linking
Socket communication	Requires creation of a dedicated protocol, so design is complex, but provides more flexibility for data passing
Shared files	Suited for simple data sharing, making it difficult to control issues like complex timing

*3 **Control SMS**: An SMS message used to control operation of a mobile terminal, launching or operating applications.
*4 **Intent**: A mechanism provided by the Android OS for programs to exchange parameters. Used between components within an application and between applications to exchange information.
*5 **Service Binder**: A type of communication between processes on Android. Based on RPC (See *8), and implemented with a dedicated Android driver.
*6 **Socket communication**: An application interface that hides the details of TCP/IP communication, using a combination of IP address and port.
*7 **Linux®**: A registered trademark or trademark of Linus Torvalds in the United States and other countries.
*8 **RPC**: A technology that implements function calls between different applications.
*9 **Method**: Behavior of objects in object-oriented programming.

some action by sending SMS Push messages from a network service node and thus, provide more dynamic services with higher added-value. The overall configuration of the SMS Push function is shown in **Figure 2**.

The standard Android SMS Push processing mechanism is extremely simple, broadcasting an Intent to all applications in the terminal when it receives an SMS Push. An Intent can be received by any application, and application developers are free to develop applications that receive SMS Push messages.

We have developed an SMS Push infrastructure with the following additional functionalities and implemented it on our 2011-2012 Winter-Spring models.

1) SMS Push Processing on the Terminal

An application receiving SMS Push messages first decides which messages it will receive and registers them in a database. Pairs of receiving applications and associated SMS Push messages are managed in the database.

When an SMS Push message is received, the Android framework*13 analyzes it and looks for a destination for the message in the receiving-application management database. If a receiving application is already registered in the database, the application is launched (Fig. 2(1)). If such a destination application is not found, processing reverts to standard Android SMS Push processing (Fig. 2(2)). Thus, processing of SMS Push messages for NTT DOCOMO applications, which require higher reliability, is executed with higher priority, and other SMS Push messages are processed appropriately according to the standard Android implementation.

2) Changes to Applications Receiving SMS Push

A management Application Pro-

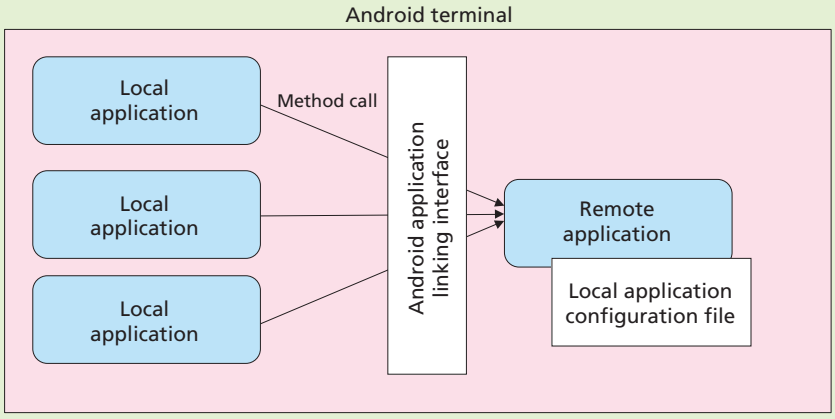

Figure 1 Overview of application linking

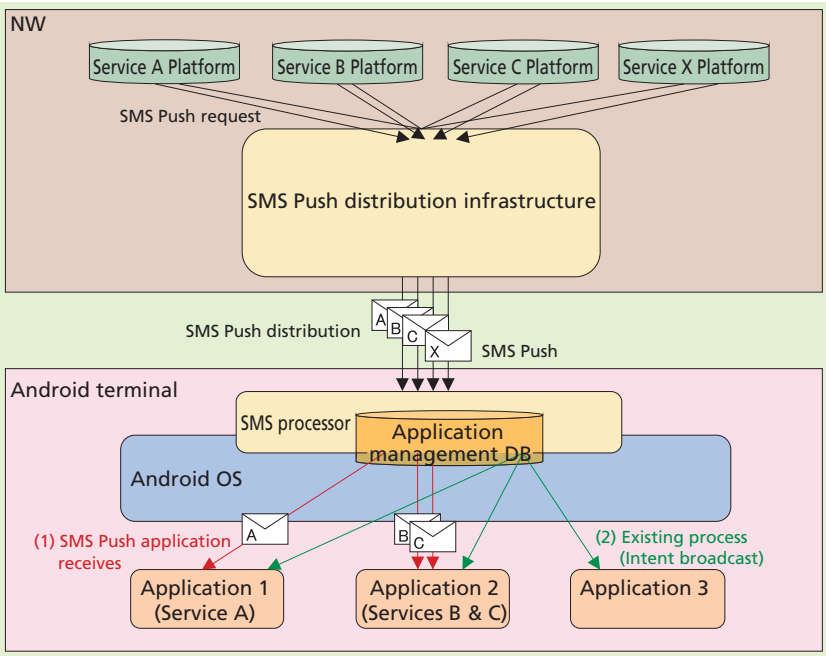

Figure 2 SMS Push function configuration

*10 **Permissions**: Settings required for linking between Android applications. By configuring permissions on the systems being linked, both developers and users can clarify how the applications will be used.

*11 **Signature**: A digital signature required when distributing Android applications, certifying the developer of the application.

*12 **Manifest file**: A configuration file that declares the functions used by an Android application. A manifest file must be created for all Android applications, according to application.

*13 **Android framework**: The Android system itself. Android applications run on this framework.

gramming Interface (API)[*14] is provided for the SMS-Push-receiving application management database. Only legitimate NTT DOCOMO applications can call this management API.

NTT DOCOMO service applications are registered in this receiving application management database in its initial state. This management API will be used to update the SMS Push receiving application database in the future, when NTT DOCOMO services using SMS Push are added, when application names change or other such cases. This approach enables us to respond flexibly even for the Android platform, with its very fast development cycle and potential for sudden large changes. As a result, we are able to minimize additional development and impact on the design after a terminal is released.

3) Extension of the SMS Distribution Infrastructure

In distribution of SMS messages, the following cases must be handled. Since there are several possible routes for SMS distribution, these issues must be resolved with common functions that can be developed efficiently.

- Case 1

When deciding terminal applicability on the Push server, which determines whether a terminal supports the service, the Subscriber Identity Module (SIM)[*15] in the destination terminal could have been exchanged before the SMS is delivered to the terminal. Thus, regardless of whether the Push server decided the terminal applicability, there will be cases when a terminal that does not support the service receives an SMS Push message. As a result, erroneous or other unexpected operation could occur.

- Case 2

There are cases when the Push server, whose functionality is allocated at a higher-level in the network, cannot get terminal information, and SMS Push messages will be sent without deciding terminal applicability. In such cases, there is a risk that terminals that do not necessarily support the service will receive SMS messages.

To resolve these issues, we consolidated decision of terminal applicability in the SMS distribution infrastructure, so it passes through SMS Center (SMSC)[*16], which provides a variety SMS distribution functions. An overview of the organization of this new terminal applicability function is shown in **Figure 3**.

For case 1, a final check that the terminal information is the same as that used when deciding applicability on the

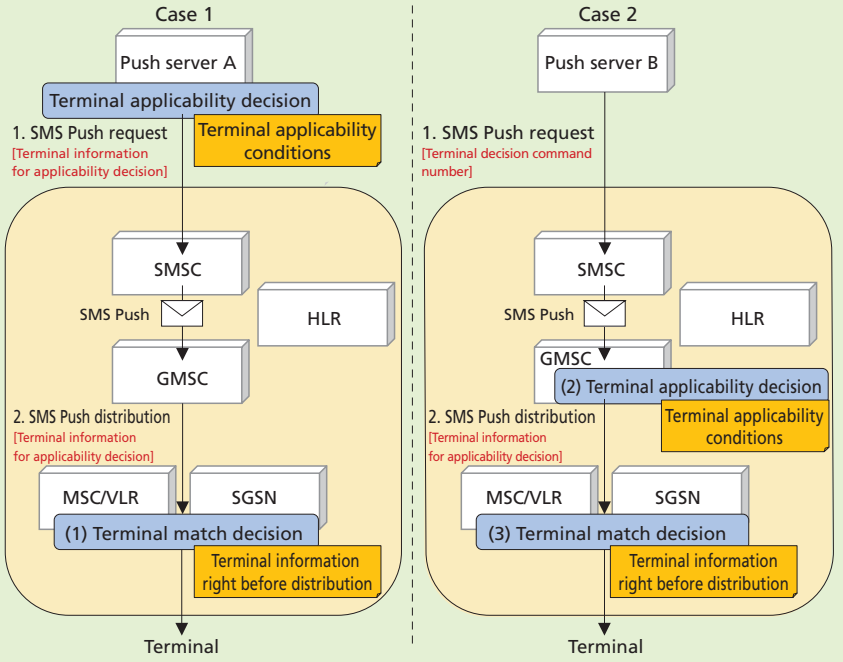

Figure 3 Terminal decision functionality

*14 **API**: An interface to use functions supported by the OS or middleware for other software.

*15 **SIM**: An IC card on which the phone number and other information about a user are stored in subscribing to a mobile communication company. The subscriber identification module in GSM is referred to as SIM.

*16 **SMSC**: The SMS Center server, as standardized by the 3GPP. Stores and retransmits SMS messages.

Push server is needed in the SMS distribution infrastructure. This check needs to happen in real-time, so we created a function on the visiting switch[1], closer to the user, which compares terminal information used for deciding terminal applicability with the latest information immediately before delivery (Fig. 3(1)).

For case 2, deciding terminal applicability must be done inside the SMS distribution infrastructure instead of in the Push server. Assuming that this function would be used on various SMS Push routes, we decided to perform the decision on the GMSC, where functionality can be concentrated and it can also be applied for overseas distribution routes (Fig. 3(2)). The GMSC controls how terminal applicability is decided with reference to decision conditions associated with a terminal applicability instruction code in the SMS Push distribution signal, so it can enforce special controls without knowing about the service. After deciding terminal applicability, as in Case 1, a final check that the terminal information has not changed from that used to decide applicability in the SMS distribution infrastructure, in case the SIM has been changed in the meantime (Fig. 3(3)). We plan to make use of this case in the future.

For improvements to SMS Push functions, we used Android Framework code approved by the Android Open Source Project (AOSP)[*17], enabling us to provide safer services than using the carrier's SMS Push. Also, the management API provides more flexibility, allowing services to be added after terminals are released, among other possibilities.

By extending functionality to concentrate routes in the SMS distribution infrastructure, terminal applicability can also be decided for SMS Push other than that using the SMS distribution architecture, so we can avoid delivering SMS Push messages to terminals that do not support the service.

4. Conclusion

In this article, we have described development of common application infrastructure functionality for Android, including application linking functionality and SMS Push functionality. Introduction of these functions enables provision of the same level of services on the Android platform as we provide on i-mode, while maintaining flexibility and security. Android is open source, so each time the next release of the OS source code or a Software Development Kit (SDK)[*18] is released, the effects of the release must be investigated and handled.

*17 **AOSP**: The Android open source project, hosted by Google.

*18 **SDK**: A set of documents, tools, libraries, sample programs, etc. needed to create applications.