NetInf Service Specifications

January 9, 2013

1 Components

The frontend team's product consists of two Android applications:

1. NetInfService, providing a NetInf node.
2. A simple browser, henceforth called "Application", using NetInf through the NetInfService to cache and share web pages.

1.1 NetInfService

The NetInfService provides a RESTful API that allows any application running on the same device as the NetInfService to access NetInf functionality through simple HTTP requests. The interface is described in detail in Section **??**. NetInfService uses parts of OpenNetInf to provide this functionality. An overview of how the NetInfService works can be seen in Figure 1.

Figure 1: NetInfService Overview

1.1.1 RESTful API

The RESTful API receives HTTP requests. Depending on the type of request the ResolutionController, SearchController or TransferDispatcher is called. Publish requests are handled by the ResolutionController. Retrieve requests first do a get using the ResolutionController and then, if the get response contained locators, transfer the file using the TransferDispatcher. Search requests are handled by the SearchController.

1.1.2 ResolutionController

The ResolutionController handles a number of ResolutionServices. Each ResolutionServices should provide publish, get, and delete functionality using some convergence layer or equivalent. Currently there are two ResolutionServices. The LocalResolutionService which uses a local SQLite database and the NameResolutionService which communicates with a specific NRS using the HTTP convergence layer.

1.1.3 SearchController

The SearchController handles a number of SearchServices. Each SearchService should provide search functionality using some convergence layer or equivalent. Currently there is one SearchService. The UrlSearchService which provide search functionality using a single string which is assumed to be a URL.

1.1.4 TransferDispatcher

The TransferDispatcher handles a number of ByteArrayProviders. Each ByteArrayProvider should provide functionality to retrieve a file (as a ByteArray) in some way. Currently there is one ByteArrayProvider. The BluetoothProvider which uses Bluetooth to transfer files from other Bluetooth enabled devices.

1.2 Application

The idea behind the browser Application is simple. When a traditional web URL is entered into the address bar and you click the refresh button, instead of just downloading the webpage from the Internet the application first tries to use NetInf to retrieve the webpage.

This is done by using the NetInfService. For the entered URL and each resource it links to:

1. Search for the hash of the URL/resource.
2. Get the file with the given hash.
3. Possibly publish the file so that others devices can get the file from this device.

If the search or get fails for any reason, be it a timeout, no matches found or something else, the webpage is downloaded using the Internet.

1.3 Control Flow

Figure 2 shows a more detailed picture of the control flow of the Application and NetInfService. The picture provides the packages and/or source files that are involved in different parts of the program, as well as some text and arrows describing the control flow.

The NetInfService mainly does its work by passing around Identifiers and InformationObjects (which encapsulates an Identifier).

An Identifier contains information about an InformationObject. The most important pieces of information in these applications are:

- Hash Algorithm
- Hash
- Content-Type
- Metadata (as a JSON string)

InformationObjects can contain attributes. In these applications the only used attributes are locator attributes. More specifically locators pointing to other Bluetooth devices and locators pointing to the local file system.

Figure 2: Control Flow

2 HTTP RESTful Interface

In order to communicate any application that wishes to use the NetInf service, there is a HTTP RESTful interface that can be used to send requests to the NetInf node. There is also a specification on what format this messages should have. In every message, regarding of the purpose of it, begins in the following way:

http://{Host}:{Port}/{Prefix}?hash={Hash}&hashAlg={Hash Alg}

Where:

- Host: the IP address of the NetInf node. If the node is located in the same devices then the host could be *localhost*.
- Port: the port of the NetInf node.
- Prefix: the name of the operation the application is requesting.
- Hash: the hash of the object.
- Hash Alg: the hash algorithm used to hash the content e.g. sha-256

2.1 Publish

To publish a content, the prefix used is "publish". And the message will look like this:

http://{Host}:{Port}/publish?
hash={Hash}&hashAlg={Hash Alg}&ct={Content Type}
&btmac={Bluetooth MAC Address}
&filepath={File Path}&meta={Meta-data}

- Content Type: The MIME content type of the object that is being publish.
- Bluetooth MAC Address: the MAC Address of the device, so it will be added as a locator.
- File Path: the file path of the object that is being published. This is used when a full put is done to the Name Resolution Service. In the case that the application does not want to do a full put then this field can be empty, ignoring also the header "&filepath=".
- Meta-data: the meta-data for the object that is being published. The meta-data is a JSON object has to be encoded for URLs and it has the following format:
  {meta}:{...}

2.2 Retrieve

To retrieve content, the prefix used is “retrieve”. And the message should look like this:

http://{Host}:{Port}/retrieve?hash={Hash}&hashAlg={Hash Alg}

2.3 Search

To search content the prefix used is “search”. And the message contains tokens for the search:

http://{Host}:{Port}/search?tokens={Tokens}&ext={Ext}

Right now, the NetInf Service only supports one token and the ext field is not used although it has been defined to satisfy the NetInf specification.

3 Components

4 Installation

The NetInf service runs as a regular Android application. It has to be installed either through Eclipse by running while the phone is connected to the computer or installing the .adk directly in the phone. It is important to have the NetInf service up and running for using any type of application that wants to communicate with it.

5 Settings

The NetInf service has a settings menu. There you can set the address and the port of the Name Resolution Service. Also, there is a Bluetooth Server switch that turns on and off the function of sharing objects through Bluetooth.