ANDSF Server and Client Implementation Description

Jari Mustajärvi (NSN)
Janne Tervonen (NSN)
Sverre Slotte (Nokia)
Janne Marin (Nokia)
Jukka Reunamäki (Nokia)

ICT SHOK Future Internet Programme
(ICT SHOK FI)

Phase 3: 1.4.2011 – 31.12.2012

Tivit, Yritysten tutkimus- ja kehittämisrahoitus, Päätös 516/09, 29.5.2009, Dnro 560/31/09

TKK, Tutkimusrahoituspäätös 40212/09, 29.5.2009, Dnro 925/31/09

www.futureinternet.fi

www.tivit.fi

This work was supported by TEKES as part of the Future Internet programme of TIVIT (Finnish Strategic Centre for Science, Technology and Innovation in the field of ICT).

Executive summary / Internal release

Title: ANDSF Server and Client Implementation Description

ANDSF server and client implementations are described in high level in this document.

Content: Combined Deliverable FI3-D1.2.2 and FI3-D1.2.3 for ICT SHOK Future Internet Phase 3.

Impact: The ANDSF server and client implementations described in this document has already been showed to several internal and external parties, for example in SHOK Summit 2012 in Marina Congress Center, Helsinki. Further, a number of major, global cellular operators have seen the demo based on this ANDSF implementation.

Contact info:

Jari Mustajärvi, jari.mustajarvi@nsn.com

Sverre Slotte, sverre.slotte@nokia.com

Link: http://www.futureinternet.fi/deliverables.htm

Table of Contents

Abbreviations and Terminology

3GPP	3rd Generation Partnership Project
AAA	Authentication, Authorization and Accounting
ANDSF	Access Network Discovery and Selection Function
AKA	Authentication and Key Agreement
ANQP	Access Network Query Protocol
AP	Access Point
CCITT	Comité Consultatif International Téléphonique et Télégraphique
CDMA	Code Division Multiple Access
CS	Circuit Switched
DDF	Device Description Framework
DM	Device Management
DNS	Domain Name Server
EAP	Extensible Authentication Protocol
EPC	Evolved Packet Core
ePDG	Evolved Packet Data Gateway
GAA	Generic Authentication Architecture
HPLMN	Home PLMN
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP Secure
IFOM	IP Flow Mobility
IPSec	Internet Protocol Security
ISMP	Inter-System Mobility Policy
ISRP	Inter-System Routing Policy
JSF	Java Server Faces
LAN	Local Area Network
MAPCON	Multiple-Access PDN Connectivity
MCC	Mobile Country Code
MNC	Mobile Network Code
MO	Management Object
MVC	Model-View-Controller
OMA DM	Open Mobile Alliance Device Management
PBRM	Policy-Based Resource Management
PDN GW	Packet Data Network Gateway
PLMN	Public Land Mobile Network
PS	Packet Switched
SIM	Subscriber Identification Module
SSID	Service Set Identifier
TLS	Transport Layer Security
UE	User Equipment
URI	Uniform Resource Identifier
VPLMN	Visited PLMN
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless LAN

1 Introduction

During the course of ICT SHOK Future Internet programme, several papers related to 3GPP-defined Access Network Discovery and Selection Function (ANDSF) have been published (refer to [1], [2], [3] and [4]). As part of the programme, also implementations for both ANDSF server and client were completed in co-operation with NSN and Nokia. In this document, both server and client implementations are briefly described.

This document is a combined deliverable of initially-planned two separate deliverables, FI3-D1.2.2 and FI3-D1.2.3, i.e. both server and client implementations are described in the same document.

The document is structured in following way: section 2 gives a brief overview of the ANDSF itself. Then, section 3 describes the server side implementation and section 4 client side implementation.

2 ANDSF by 3GPP

In 3GPP, ANDSF is defined in a number of specifications: Stage2 – i.e. the requirements – are defined in 3GPP TS 23.402 [5], Stage3 – i.e. the implementation of the requirements, e.g. protocol details – are defined in 3GPP TS 24.302 [6] and the ANDSF Management Object (MO) is defined in 3GPP TS 24.312 [7].

3GPP defines ANDSF standard for managing device access network selection among the plurality of the available access networks.

Independent of the selected access network, the UE should use it to access EPC core (PDN-GW) and thereby enable seamless connectivity over different access technologies. There is only one exception – non-seamless WLAN offload - and we will come to this later in this chapter.

Wi-Fi access to EPC can be realized via trusted access network or via un-trusted access network (refer e.g. to [2] for some more details). If Wi-Fi access authentication is done using EAP-SIM or EAP-AKA, the home AAA server can provide this indication to the user. If there is no such indication, the device shall assume un-trusted access. In case of trusted access the access network itself allocates the IP address from the PDN-GW to the device and the access network has secure connection with the EPC. In case of un-trusted access the device shall make an IPSec tunnel to the ePDG and therefore secure traffic to the EPC. ANDSF does not provide this trust indication for the Wi-Fi access but it is good to understand this security scheme.

2.1 ANDSF Settings

The ANDSF information is described as special OMA Device Management (DM) Management Object (MO). The Management Object (MO) is compatible with the OMA Device Management (DM) protocol specifications, version 1.2 and upwards, and is defined using the OMA DM

Device Description Framework (DDF) as described in the Enabler Release Definition OMA-ERELD-DM-V1_2.

The ANDSF MO is used to manage Inter-System Mobility Policy (ISMP) and Inter-System Routing Policy (ISRP) as well as access network discovery information stored in a UE supporting provisioning of such information from an ANDSF server. ANDSF server installs an ANDSF MO into the device and manages this information dynamically via OMA DM Procedures. The device can then opt to obey this information or use it as a hint for suitable network selection if device wishes to engage a data transfer session with the network. ANDSF therefore relates to PS connections only and not to CS connections.

ISMP contains network selection policies for given location and optionally for given time. Device may have any number of ISMP policies but only one of them – the active policy - is followed at any given time based on the validity conditions set by the operator. Device shall select the access network to use according to access network type (Wi-Fi, 3GPP, WiMAX, CDMA [defined by 3GPP2]) and optionally also access network identity (like SSID) according to operator preferences in the active ISMP policy.

The ISRP information contains rules for traffic distribution over different access networks for UEs that are configured for IFOM, MAPCON or non-seamless WLAN offload (features defined by 3GPP). While ISMP only concerns on one access network selection for all the traffic ISRP allows defining the access network selection rules at traffic profile level. Specific type of traffic is sent via one access network while some other traffic via another network. It of course will be limited by devices capabilities to use multiple access networks at the same time. As with ISMP the ISRP rules also have validity conditions.

IFOM – IP Flow Mobility – contains traffic profiles and access network selection rules according to these traffic profiles.

MAPCON allows access network selection based on used APN.

Non-seamless WLAN offload internally is identical to IFOM rules except there is no access network type selection as WI-FI is mandatory access network in this. The traffic profiles in the non-seamless WLAN offload rules define traffic which can be sent directly to internet without allocating a PDN connection for them in the PDN-GW. There is no service continuity by preserving IP address if such traffic is moved to a new connection. If non-seamless offload rule is followed, non-matching traffic is implicitly assumed to be routed via tunnel to ePDG.

As part of the ANDSF setting provisioning, the ANDSF server can also provide actual access network settings to the device. It could include credentials to the access network if EAP-SIM or EAP-AKA is not used in the network.

2.2 ANDSF Security

ANDSF relies on secure communication with the device and ANDSF servers. This security is built on USIM card capabilities to generate shared secrets between the UE and server without exchanging this secret over the air. The server shall request the session key from home operator directly. A TLS session shall be created based on this key. This concept is called Generic Authentication Architecture (GAA) in 3GPP. Both the UE and the server can

authenticate each other. ANDSF servers can be discovered automatically using DNS based on the MCC and MNC values of HPLMN or VPLMN operator.

Optionally, the security can be based on the HTTPS connection where only server is authenticated by the server certificate. The user authentication will be conveyed as a simple username/password based digest information exchanged between the server and UE in the initial OMA DM session setup phase. Since this method requires the UE learns the credentials somehow, it is not as secure as GAA method. Typically, the credentials are either preconfigured into the device by operator or they are delivered to the device over SMS using OMA DM Bootstrapping. As such, this OMA DM Bootstrapping based security is weak as this initial step can be forged. Anyone can setup an ANDSF server and send legitimate looking OMA DM Bootstrap SMS to anyone, and if the recipient decides to trust to the message security is gone. OMA DM Bootstrap information contains the ANDSF server address.

2.3 OMA DM Session

The OMA DM Protocol uses a messaging sequence that consists of three parts:

- Alert Phase – used only for server initiated management sessions
- Setup Phase – authentication and device information exchange
- Data (Management) Phase

The alert phase is optional and data flows only from server to client. If server wishes to push settings to the device it has to send a notification package to the device requesting it to start a new management session. This alert phase is done with a WAP push message over SMS bearer.

The setup phase and data phase are carried over a TLS session with the UE and server. The session is started always by UE, but server may request it with the alert notification SMS. TLS session is secured according to the used security mode. In GAA, the TLS session is established using negotiated PSK key while in generic OMA DM mode it is a normal HTTPS session.

In the setup phase, the client issues a setup request, with data flowing from client to server, followed by a server response. The initial client request contains 3 primary pieces of information:

1. Device information e.g. device id, manufacturer, model tag, phone language and DM protocol version

2. Client credentials used for authentication purposes unless GAA is used.

3. Indication whether the incoming session is client or server initiated

If GAA is not used the server responds with its credentials, to identify the server to the client for authentication and identification purposes. The server shall also include the first data command. In the OMA DM it is always the server which shall issue commands and the client can only follow those.

In the data management phase, the device initiated messages (inside HTTP POST Request) consists of status information and results that it provides in response to commands from the server. The first message is the response to commands included in the final setup phase message by the server (inside the HTTP ACK response). Afterward, the server can issue new

commands in next HTTP ACK message to previous HTTP POST Request message or it can simply indicate there are no more operations. In that case, the client will stop sending responses in the HTTP POST Request messages and closes the session.

Device configuration data is organized in a hierarchical structure called the device management tree. Subtrees are called device management nodes and a leaf, usually a single configuration parameter, is called a manageable object. ANDSF MO is the subtree under ./ANDSF node.

Management objects can be manipulated via SyncML messages using following server commands:

- Add: Add an Object (Node) to a tree
- Get: Returns a Node name based on the URI passed with the GET request
- Replace: Replaces an Object on the tree
- Delete: Deletes an Object on the tree
- Copy: Copies an Object on the tree
- Exec: Execute a device-defined command on an Object on the tree

ANDSF MO management typically uses only Add/Get/Replace/Delete commands.

2.4 Hotspot 2.0

ANDSF relationship to Hotspot 2.0 specification is still unclear. Hotspot 2.0 defines 802.11u Access Network Query Protocol (ANQP) based service discovery and it can be used to find out what operators the access network serves and what services it offers. UE can request this information prior authenticating itself to the network.

Hotspot 2.0 is not 3GPP centric. While ANDSF relies on EAP-SIM/AKA based authentication, in Hotspot 2.0 the user can have multiple Wi-Fi accounts of any type.

3 ANDSF Server Implementation

3.1 ANDSF Server Building Blocks

ANDSF server is implemented to provide ANDSF service to the devices. This chapter describes main building blocks of the server.

The server provides NSN Calypso based GUI to manage the ANDSF MO's and setting provisioning events. Calypso is depicted in chapter 3.1.1.

3.1.1 Calypso WebUI Framework

Calypso is a NSN proprietary framework intended for web application development. Calypso framework is based on Java Server Faces (JSF). Selected JSF implementation is MyFaces. WebUIF does not exclude any JSF functionality, but extends them in order to provide a

framework with which it is easy to develop new web applications. Calypso also allows the usage of RichFaces alongside MyFaces.

Calypso offers a single navigation to multiple web applications and seamless single sign-on. Calypso WebUIF is set of JSF Pages, Tag library of JSF UI components, set of Eclipse tools to help application development and help guide integrated into Eclipse. Framework also contains a login into system and navigation where user can open different applications.

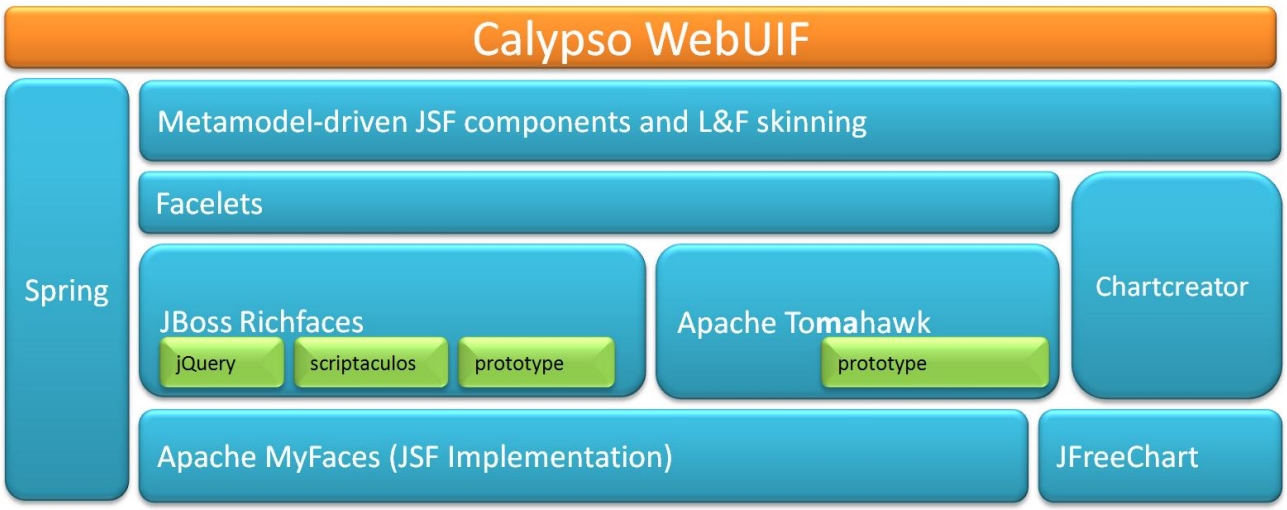


Figure 1: Calypso WebUIF stack

Typically, Calypso applications use JSF components from the default core implementation but also from extensions like RichFaces.

3.1.2 Funambol DM

Funambol provides a Device Management (DM) server and client API for building DM applications. The DM server is based on the Open Mobile Alliance DM specification. Funambol DM server is open source software but it has not been maintained a long time anymore. Currently Funambol community (https://www.forge.funambol.org/) provides only Funambol Data Synchronization (DS) server. Commercial support for both servers is available in http://www.funambol.com. Nevertheless, Funambol DM server can be downloaded from http://en.sourceforge.jp/projects/sfnet_funambol/downloads/dm-server/v36/funambol-dm-server-3.6.0.zip/. Following information is mostly extracted from Funambol_DM_Server_Concepts_And_Technical_Overview.pdf document available in the sourceforge server above.

Funambol DM includes two components, the Funambol DM Server and the Funambol DM Client API, which are described below.

3.1.2.1 Funambol DM Server

The Funambol DM Server implements OMA DM. It provides a framework that implements DM functionality and that supports extensions. In the diagram below, the orange blocks represent functionality that developers can plug into the framework.

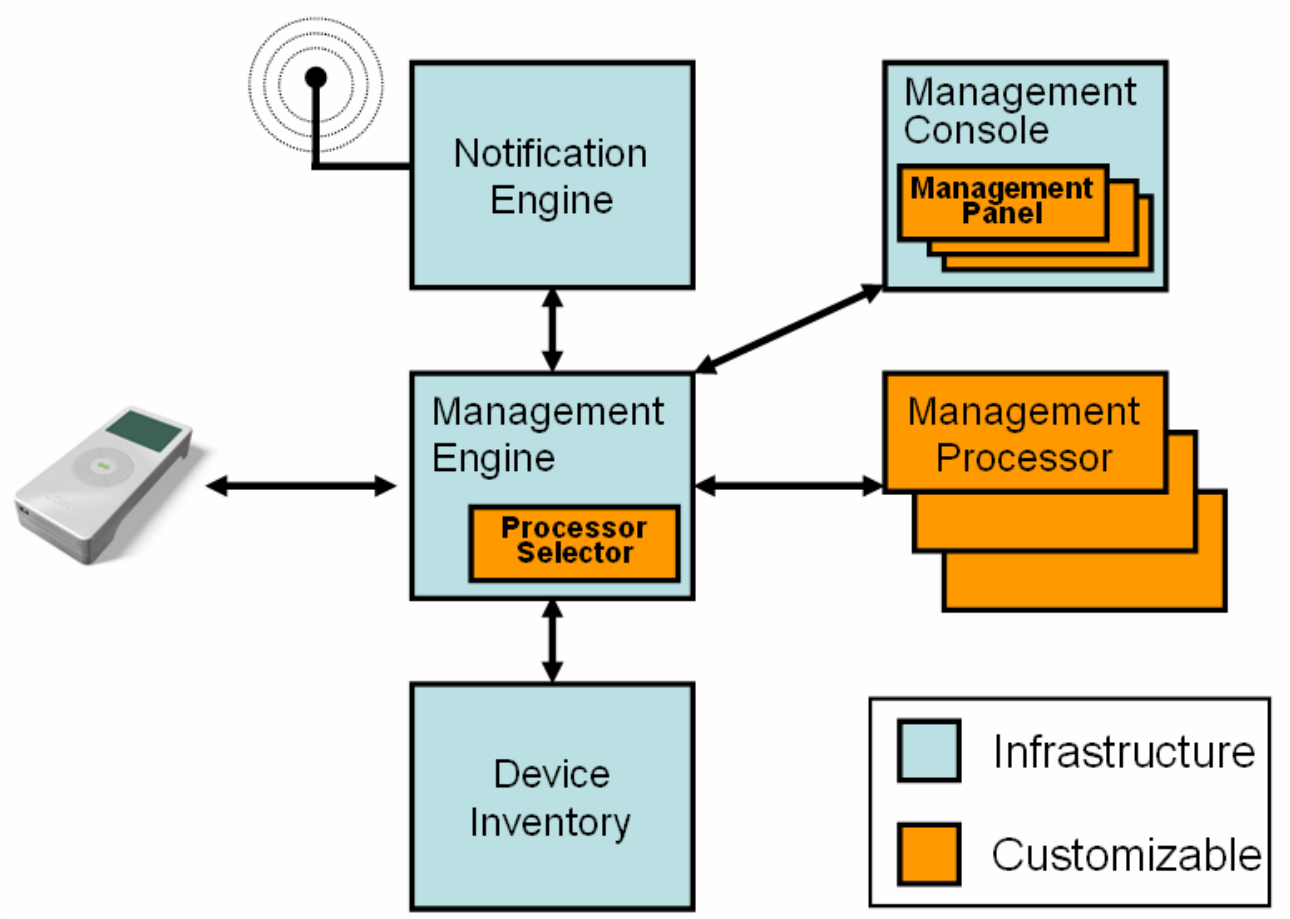


Figure 2: DM Server Architecture

At the core of the DM Server is a management engine that is responsible for understanding and interpreting the OMA DM protocol. The management engine delegates the real DM logic to an external and customizable Management Processor component. Multiple management processor components can be used and a management processor is selected via the Processor Selector that is invoked at runtime from the management console.

A Notification Engine notifies mobile devices to start a new DM session. For example, in the case of WAP Push enabled devices, it builds a binary WAP push message which is delivered to the device as specified by OMA specifications. The Notification Engine supports both DM bootstrap and DM notification messages. The former are used to store in the device initial

connection and authentication data; the latter is used to trigger a server initiated OMA DM session.

The repository of Device Descriptor Framework (DDF) descriptors is in the Device Inventory. The DDF framework is part of the OMA DM specification and is a dynamic discovery mechanism that a server can use to discover and gather information about the device's management tree objects.

The Management Console is used by staff to interact with the DM server and with the devices.

The Management Processor contains code that implements DM logic. When the server receives an OMA DM message, a customizable Processor Selector chooses the Management Processor best suited for the request. Policy examples are selecting a Management Processor based on:

- device details e.g. device id, manufacturer, model
- state of the current DM session
- external configuration/interaction/events

Each Management Processor can have its own Management Panel. The panel implements the user interface through which staff interacts with the device, using a particular Management Processor object.

3.1.2.2 Funambol DM Client API

Client APIs support the development of OMA DM applications that can be remotely managed by an OMA DM server. The API includes:

- Database Layer: Represents the device's local database.
- SyncPlatform Store (SPS) Layer: Abstracts database access. It can be used by the framework to access the local database in a generic way. SPS is optional in a Funambol application.
- SyncPlatform Data Synchronization (SPDS) Layer: Abstracts the complexity associated with the OMA DS protocol from extension developers, who simply need to call a few methods of the synchronization manager to kick off the synchronization process.
- SyncPlatform Device Management (SPDM) Layer: Core layer of the OMA DM implementation. The main component is called device manager, which is responsible for storing and reading the management trees and nodes and for exposing to the application code hooks to start and handle the DM session. All protocol details and complexity are abstracted for the mobile application developer.

3.2 ANDSF server model

3.2.1 Functionality

ANDSF server offers a flexible processing engine in order to support operator business processes. This engine can be configured in many ways, assuring the operator that its subscribers always receive the ANDSF services targeted for them in a particular use case.

The server is accessed using standard ANDSF mechanisms utilizing OMA DM. The settings the server will configure into the devices are defined via NSN demo ANDSF WEB GUI. The GUI also

offers simple monitoring view for the events created during ANDSF OMA DM sessions with the end users.

This engine is built on top of following main pillars:

- Tenants
- Subscriber groups
- ANDSF MO building blocks
- Rulecards

As a result of this process, the end user device will be provisioned with relevant ANDSF settings. A rulecard effectively defines the ANDSF Managed Object (MO) that will be sent to the tenant subscriber devices. A tenant can provide different MO's to different user groups.

3.2.2 Tenants

A tenant is a customer operator of the ANDSF service. The administrator will create a new tenant into the system. The tenant is immediately associated with a default PLMN code used in the relevant ANDSF settings. MSISDN numbers managed by the tenant are also defined here. A tenant user operating the ANDSF server GUI cannot trigger settings provisioning to MSISDN numbers which are not part of the MSISDN numbers defined for the tenant. Each tenant can have any number of user's or actually tenant administrators that are allowed to configure ANDSF settings and trigger provisioning in the ANDSF server. Only system administrator can define these users though.

End users who actually are eligible to receive the ANDSF settings don't have to be tenant administrators. Tenant administrator just has to send OMA bootstrap for these devices. The bootstrapping configures the end user account for the ANDSF server in the device and other settings requires to access the ANDSF server. Once device is bootstrapped it can any time connect the ANDSF server and receive relevant ANDSF settings in case they have changed.

Each tenant administrator will only see configurations created by itself or by other tenant administrators of the same tenant. The tenant view and content is identical to every tenant user.

3.2.3 Subscriber Groups

A subscriber group is an entity used to manage a set of subscribers. Subscriber groups typically answer to the question "WHOM?" as they represent the target for the ANDSF device management operations. Each tenant can define any number of subscriber groups. When final ANDSF MO is created, the MO is associated with specific end users by associating relevant subscriber groups with the setting.

3.2.4 ANDSF MO Building Blocks

ANDSF MO building blocks are ANDSF policy elements, ANDSF discovery information elements and the settings for WI-FIs referred by an ANDSF discovery information element. The building blocks are then used to compose the rulecards which define the actual ANDSF MO for specific users. These definitions answer to the question "WHAT?" as they represent the content of ANDSF device management operations.

3.2.4.1 ANDSF Policy Element

Each ANDSF policy element represents one potential ANDSF Policy entry in the final ANDSF MO. A policy element includes all standardized ANDSF policy items except validity area conditions by 3GPP2 and WiMAX networks.

3.2.4.2 ANDSF Discovery Information Element

Each ANDSF discovery information element represents one potential ANDSF Discovery Information entry in the final ANDSF MO. A discovery information element includes all standardized ANDSF discovery information items except access network area references by 3GPP2 and WiMAX networks.

ANDSF discovery information element can refer to specific access network. Only WI-FI access network is supported. If WI-FI access network reference is defined, then also the WI-FI access network configuration parameters must be defined.

3.2.4.3 WI-FI Access Network

ANDSF service allows to provision specific WI-FI networks as part of the ANDSF MO. The ANDSF MO itself can contain references to WI-FI network settings and in case they are defined, then these actual WI-FI network settings are provisioned together with the ANDSF MO.

3.2.5 Rulecards

Rulecards represent a set of business rules describing scenarios that associate device management targets (Subscriber groups) with a series of ANDSF settings defining system behavior in various situations. Rulecards answer to the question "WHEN?".

Rulecards are named constructs that make it possible, for example, to deliver different settings to the operator's own subscribers and to roaming users visiting the network. A rulecard typically contains a list of subscriber groups and a set of ANDSF policy element, discovery information elements and WI-FI settings referred from discovery information elements. When rulecard is composed a priority is assigned to included policy elements. The settings defined on the card apply only to the subscribers defined by the subscriber group definition of the card.

Each rulecard is associated with a state – active or non-active. When device enters the ANDSF server only active rulecards are considered. If the device still has several rulecards it belongs to then the first rulecard is selected. Tenant administrator should make sure only one rulecard is active for each subscriber.

If no active rulecard is found for the device, all previously provisioned settings will be deleted from the device.

Rulecards itself are not associated to any specific OMA DM bootstrap scenario. Each device has to be separately bootstrapped by the tenant administrator.

3.3 ANDSF Management Process

Once tenant operator has defined the rulecards to apply for customer devices and these devices have been bootstrapped the devices are able to receive ANDSF service. The bootstrap data includes indication for the device to contact ANDSF server for initial settings and this is the first time when device will connect the ANDSF server.

Note! ANDSF server does not currently support GAA based security.

The connection can be run over unsecure HTTP session or secure HTTPS session. The HTTP port used is 8080 and HTTPS port 8443.

When device connects ANDSF server, the server will challenge the credentials if not present initially. The device in turn can challenge the server credentials. Default authentication mechanism is MD5 hash calculated over username, password and nonce values. The nonce value is reallocated on every authentication. If authentication succeeds the ANDSF server will start the provisioning process for the device. Authentication method is configurable value.

The Funambol process will try to find pending connection request for the device if session was server initiated session. The pending request information contains information which session processor should be allocated to handle the request. This processor and the default processor for UE initiated requests is always ANDSF processor in this server. The processor will complete the ANDSF session with the UE.

3.3.1 ANDSF Processor

The first step of the provisioning process is to verify ANDSF support and other capabilities of the device. Currently this is done by fetching the /ANDSF, /ANDSF/Policy, /ANDSF/DiscoveryInformation, /ANDSF/ISRP, /AP, /NAP and /EAP nodes from the device. If the result for the /ANDSF node is 404 (not present), it is assumed the device does not support ANDSF and provisioning process is aborted. Otherwise ANDSF server records all the nodes returned by reading process for the management session.

Next provisioning process will search active rulecard for the device. Each rulecard is associated to a tenant and further to a potentially to a multiple MSISDN numbers. The MSISDN number is deduced by mapping the user credentials to MSISDN and IMEI numbers stored when the device was bootstrapped.

If no active rulecard was found for the user all previous settings provided by the ANDSF server for the user will be removed from the device. If multiple active rulecards were detected for the user, then only first one found will be considered. In this case, the tenant administrator has failed to configure ANSDF information properly and there is nothing else the server can do for it. The tenant administrator should make sure only one rulecard is active for each subscriber. If there are multiple rulecards for a subscriber, the selected rulecard is not totally random (same rulecard will be selected every time) but it is impossible to say which of the active rulecards will initially be selected.

Once active rulecard has been selected, the ANDSF server will compare the information with the information read from the device. All missing settings will be provisioned to the device. Also all those settings that device already have but which does not belong to selected rulecard

will be deleted from the device. As a result the device will end up with fresh up to date ANDSF MO without any trash from previous provisioning sessions.

Every Wi-Fi, ISMP, ISRP and DiscoveryInformation setting in the tree has a node name composed of the hash value calculated over the setting. If any setting under these nodes is modified in the server this hash value will change too and the server can detect which items need further attention from the server.

3.4 Implementation of ANDSF Server

ANDSF server is implemented as a Funambol processor element in the Funambol DM Server described in chapter 3.1.2. ANDSF processor described in chapter 3.3.1 is default processor to handle any UE ANDSF session in the server. ANDSF server does not implement GAA functionality.

The ANDSF GUI is implemented using Calypso framework described in chapter 3.1.1. The GUI is loosely coupled to the Funambol DM as it uses Funambol DM notification bean services to send OMA DM Bootstrap and Notification messages to target devices.

The GUI is used to create ANDSF rules for different subscribers. All information will be stored into a MySQL database which is then used by the Funambol ANDSF processor to provide the settings to clients connecting to it.

Due to various issues with international SMS when segmented binary messages are sent to target devices to convey notification and bootstrap messages to end user devices, Funambol core functionality has been modified a bit. The credentials are not randomized anymore for the bootstrap. Username and password are derived from MSISDN and IMEI. This allows users to define bootstrap information manually to the devices once the bootstrapping has been executed in the ANDSF server. Few other modifications have been done too. Funambol's own demo UI has been disabled.

ANDSF server does not introduce any web service interfaces for external management.

3.4.1 Tools

Eclipse was used to develop both the ANDSF server GUI as well as the ANDSF processor in Funambol. Calypso includes Eclipse plugin for Calypso features.

Funambol runs on JBOSS 5.1 server. Included tomcat web engine handles the ANDSF GUI.

MySQL Workbench was used to design ANDSF database. JPA or other annotation based automated database models were not used. Spring API is used to separate database access from the GUI model.

XML schema was created out of the ANDSF MO. Actual JAVA ANDSF classes were generated out of this with JAXB.

3.4.2 MVC model

As Calypso framework is based on JSF regime the ANDSF server GUI follows Model-View-Controller (MVC) design pattern. The behavior (model) is separated from the data presentation (view). All requests made by the UI will pass FacesServlet which further calls respective methods in the model (managed beans). If data is modified it is immediately saved into the database by the beans.

The UI view is composed out of JSF pages (*.xhtml). These are all hand written except the login view which is derived from the Calypso framework. Extended table views are used to represent collection of items while typical form view is used to edit the items. Related table data models were derived from Calypso SortableSequenceDataModel. AJAX use is hidden by Calypso framework.

There are dedicated beans for main UI components like user, tenant, rulecard, ISMP, ISRP, Discovery Information and WI-FI settings management. All beans are statefull session beans.

The model is further depicted in chapter 3.2.

3.4.3 Persistence

MySQL database is used to store every ANDSF setting created by the ANDSF GUI. This same database is then used by the Funambol ANDSF processor to find out correct settings for each user when they contact the server.

No persistence API was used to implement database access. The GUI is separated from the database via Spring API but all classes for the database manipulation were manually written using direct MySQL commands. In retrospect, this probably should have been implemented using some annotated persistence API like JPA.

4 ANDSF UE Implementation

The ANDSF client for the Nokia N900 acts as a counterpart to the ANDSF server described above. The implementation supports the basic features of ANDSF including ISRP/IFOM functionality.

The client device is a Nokia N900 handset running the Linux-based Maemo operating system. The flexible platform makes it well suited to a modular implementation strategy with the different modules communicating over native IPC-mechanisms such as D-Bus and the GConf-tree.

The client implementation consists of an OMA-compliant device manager for receiving ANDSF-data from the server, a priority manager for transforming the ANDSF-data into native format, and an extended connection manager for performing the actual network selection.

4.1 OMA DM

OMA DM functionality is missing in the Nokia N900 and thus it was implemented. The work was divided to the four parts: triggering, authentication, handling of OMA DM session and storing results of session.

The SMS (short message service) handler of Nokia N900 was extended to support reception of OMA DM wbxml-based messages (bootstrap and notification message), and starting of corresponding handler for OMA DM. These were added to existing program blocks in the Nokia N900.

In addition to extending existing capabilities, there was a need for a new program block which contained OMA functionality (authentication, session management and storing results). Authentication of OMA DM session was implemented according the ANDSF security mechanism highlighted in chapter **Error! Reference source not found. Error! Reference source not found.**.

OMA DM session fulfilled chapter 2.3 OMA DM Session parts in authenticated session. The results of the session were stored in an internal database (in the case of Nokia N900 that is the GConf-tree). After completing the session the priority manager was notified that there were changes made by OMA DM server. The priority manager refines the OMA DM specific information to a form understood by device.

4.2 Priority manager and connectivity manager

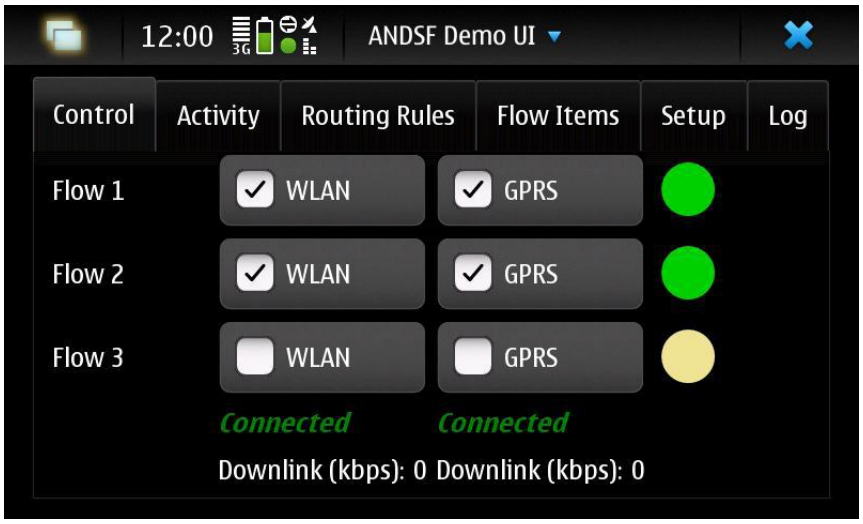
As the Nokia N900 device does not recognize the concept of priorities associated to certain WLAN APs (access points) this functionality had to be added in the form of a priority manager program block. Additionally, the existing connectivity manager (jcd2) had to be extended with the concept of priorities to be able to perform network selection based on this information.

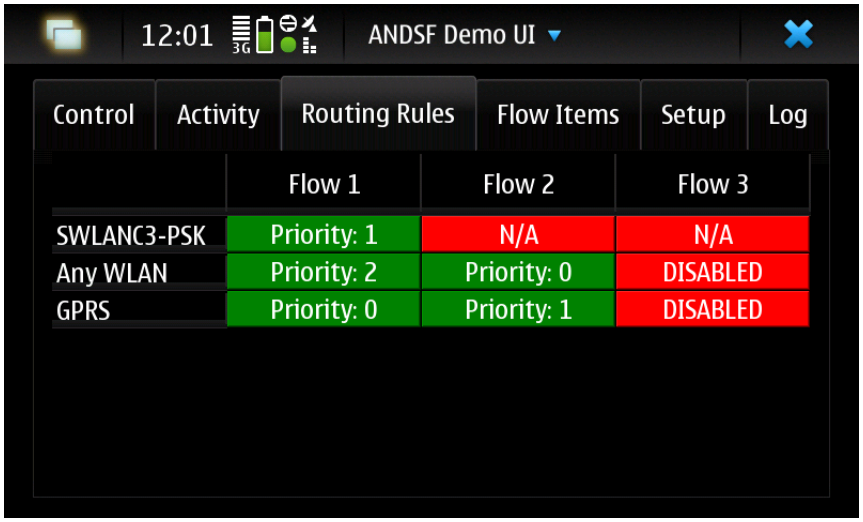
Whenever activated (by the OMA DM) the priority manager reads the database written by the OMA DM program block and transfers the information into a form understood by native application on the Nokia N900. The result of this transformation is then merged with already existing information related to WLAN AP information in the system settings repository of the Nokia N900 (GConf-tree). In practice this means that the WLAN AP selection priority information from the ANDSF settings is added to the WLAN AP information in the GConf-tree.

The connection manager (jcd2) is augmented with the concept of priorities. The priority information is taken into account whenever the connection manager is asked to perform network selection.

4.3 User interface

In order to have easy and intuitive view to rules generated by ANDSF server to UE, UI was developed. UI visualized the different flows and routing rules attached to flows. UI was used only for development visualization and did not have any active role in ANDSF implementation.


12:00
ANDSF Demo UI
Control
Activity
Routing Rules
Flow Items
Setup
Log
Flow 1
WLAN
GPRS
Flow 2
WLAN
GPRS
Flow 3
WLAN
GPRS
Connected
Connected
Downlink (kbps): 0
Downlink (kbps): 0


12:01
ANDSF Demo UI
Control
Activity
Routing Rules
Flow Items
Setup
Log
Flow 1
Flow 2
Flow 3
SWLANC3-PSK
Priority: 1
N/A
N/A
Any WLAN
Priority: 2
Priority: 0
DISABLED
GPRS
Priority: 0
Priority: 1
DISABLED

5 References

[1] Realization of Policy-Based Resource Management Concept, version 1.0, 16th of February 2010, Janne Tevonen & Jari Mustajärvi.

[2] Offloading Traffic from Cellular Networks with PBRM, version 1.0, 30th of June 2010, Janne Tervonen

[3] Policy and Charging Control Functionality with WLAN and PBRM, version 1.0, 22nd of December 2010, Janne Tervonen

[4] Study on Access Selection Steering Mechanisms, version 1.0, 30th of September 2011, Janne Tervonen & Janne Marin

[5] 3GPP TS 23.402, Architecture enhancements for non-3GPP accesses, Release 11, v11.2.0, March 2012.

[6] 3GPP TS 24.302, Access to the 3GPP Evolved Packet Core (EPC) via non-3GPP access networks, Release 11, v11.2.0, March 2012.

[7] 3GPP TS 24.312, Access Network Discovery and Selection Function (ANDSF) Management Object (MO), Release 11, v11.2.0, March 2012.