WakeUp platform

Architecture

Fernando Rodríguez Sela
Guillermo López Leal

WakeUp platform: Architecture

by Fernando Rodríguez Sela and Guillermo López Leal

GIT version heads/master # Compiled on vie feb 14 13:38:32 CET 2014

Table of Contents

List of Tables

Chapter 1. Introduction

Today, all mobile platforms has their own Push Notification platform used to deliver asynchronous messages to devices, trying to keep the minium of opened connections from the system and apps to third party servers and encapsulate all push information in one single channel.

All these platforms have a common denominator, all of them maintains a TCP socket opened all the time.

This socket kept opened by the system is used to deliver the notifications when it's needed. Some of the systems are really lightweight, meanwhile others carry a lot of payload, have signaling messages, priorities and other advanced features.

Current Internet solutions issues

Internet Push solutions are based on a public accessible server which handles all the notification acceptance and delivery from third party servers to the phones.

These solutions are not designed considering how mobile networks work and forces the handset to maintain an open socket with the server in order to deliver the notification as quick as possible.

This way of working increases the signaling in cellular networks and handsets battery usage. For more information about this, please refer to "Mobile network issues with current Push platforms" chapter.

Service Description

The WakeUp platform is aimed to wakeup mobile devices so external Push servers of any kind of operating system can avoid having an open socket all the time and sending pings to keep that connection alive.

The main objective of this service is to deliver wakeup packages based on some user information given by the Push platform. It's designed to work in cellular networks environments and have a small and simple integration with the OBs.

This server will provide a public API only for trusted partners which allow them to wakeup any mobile device inside mobile operator networks, thus removing the necessity of having an opened socket all the time.

What Push platforms will use this system?

Third party Push platform providers like Mozilla's FirefoxOS Push server should sign an agreement with mobile operators to have access to the wake up public APIs.

These third party clients will benefit from the battery savings of their devices and reduced network consumption on any cellular network.

After signing the agreement, the Operator will deliver a signed certificate (self-signed or verified by third parties CAs) which will identify in a unique way the wakeup platform users.

This certificate MUST be used to Authenticate and Authorize access to the WakeUp platform.

Chapter 2. Mobile network issues with current Push platforms

This chapter explains why current solutions are bad for the mobile networks and how we designed this server to solve these issues.

In order to understand the complete problem, we need to introduce how the mobile networks work at radio level and also how the carriers have their network infrastructure.

Mobile networks in a Private or Public LAN

Since on IPv4 the amount of free addresses is really low (it is usually 2^{24} IPs, or 16777216), cellular networks are divided into the ones with real IPv4 addresses (normally for 3G modems) and private addressing model for handsets, usually with NAT protocols.

In the case of private networks, it's obvious that it's not possible to directly notify the handset when the server has a notification for it (because it is behind a NAT), so smartphone manufacturers or mobile OS developers decided to maintain opened channels with their servers so it is possible to notify handsets asynchronously, and only when something happens on the server.

On the other hand, if the handset has a public address, or is using IPv6, it is theoretically possible to send the message directly making third party solutions not useful, however in order to protect users, carriers might deploy firewalls to avoid direct access from Internet to handsets.

Mobile Network. Circuit domain states

In the 3GPP TS 25.331 specification, we can query all the circuit domain statues of the RRC Layer (Radio Resource Control).

In order to simplify, we only list the third generation (3G) states:

- Cell_DCH (Dedicated Channel)
  When the handset is in this state is because it has a dedicated channel on the mobile network.

  Normally the network sets a handset into this state when it's transmitting a big amount of data.

  The inactivity time of this state is really short, known as T1 timer it should vary between 5 and 20 seconds. If T1 is fired, the handset will be changed to the Cell_FACH state.

- Cell_FACH
  In this state the handset is connected to the mobile network using a shared channel with other handsets.

  Normally, this state is assigned by the network when the handset is transmitting a small amount of data. So it's common to use it when sending keep-alive packages.

  The inactivity time of this state is a little longer (30 seconds) and is known as T2 timer. When T2 timer is shotted, the handset will be moved to Cell_PCH or URA_PCH (depending on the type of network)

- Cell_PCH or URA_PCH (PCH: Paging Channel) (URA: UTRAN Registration Area)
  In this state the handset is not able to send any data except signaling information in order to be able to localize the handset inside the cellular network.

In both states, the RRC connection is established and open, but it's rarely used.

In this state, the handset informs the network every time the device change from one sector to another so the network is able to known exactly the BTS which is offering service to the device.

The T3 timer defines the maximum time to be in a PCH state. This timer is longer than T1 and T2 and depends on each carrier. When it's fired the handset is moved to IDLE mode so if new data transmission is needed the handset will need near 2 seconds to reestablish the channel and a lot of signaling messages.

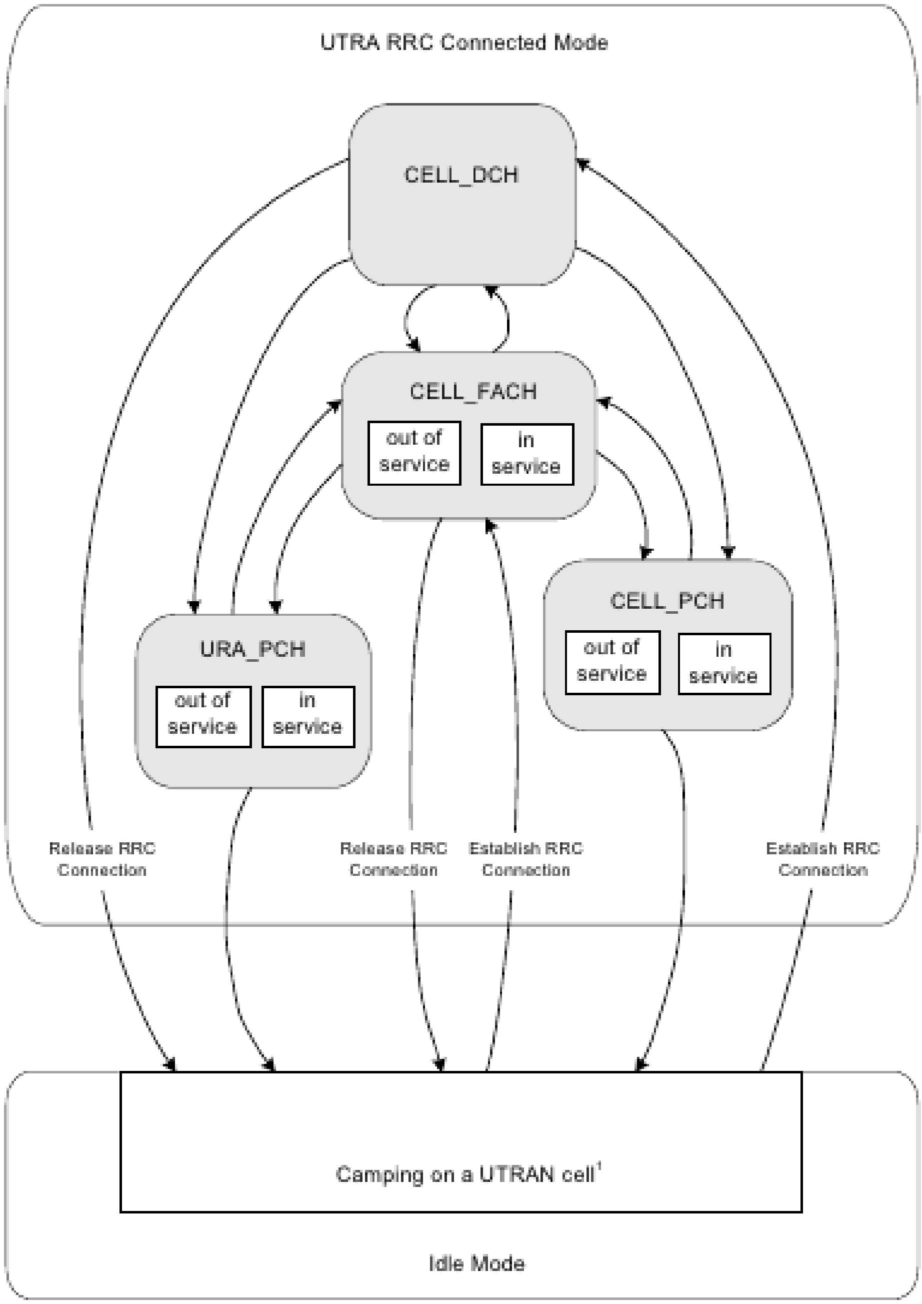
- RRC_IDLE
  This is the most economical state since the handset radio is practically stopped.

  In this state, the radio is only listening to radio messages querying the handset to "Wake Up" (paging messages).

  Also, the handset modem is listening the cell data so each time it detects that the user changed from one LAC (Localization Area Code - Group of multiple BTS) to another, the handset will change to the PCH state in order to inform the network.

  So when a handset is in this state, it can be Waked Up to a more active state and also the network knowns the LAC where the handset is moving, so if the network needs to inform the handset it should send a broadcast paging message through all the LAC BTS in order to locate the handset.

The following scheme represent the different radio states ordered by power consumption on the device:



Mobile Network. Package domain states

In the 3GPP TS 23.060 specification, we can analyze all the package domain states of the GMM Layer (GPRS Mobility Management).

The package domain states are simpler than radio ones (only 3 states):

- READY (2G) / PMM_CONNECTED (3G)
  The handset has a PDP context established and is able to send and receive data.

- STANDBY (2G) / PMM_IDLE (3G)

The handset isn't transmitting anything but the PDP context is not closed, so it maintains a valid IP address.

In this state the handset don't consume any resource but the network is maintaining his IP address as a valid one, so it's very important to try to maintain the handset in this state in order to be able to Wake Up it and change to a PMM_CONNECTED state in order to transmit/receive information.

- IDLE (2G) / PMM_DETACHED (3G)
  In this state, the handset hasn't a PDP context established so it hasn't a valid IP address.

Mobile Network. States relation

In this section we show the relation between RRC and GMM states.

In order to simplify this table, we only consider the handset is only using data channels, so no voice nor SMS (circuit domain) is being used.

Table 2.1. RCC - GMM relation

RCC State	GMM State (2G/3G)	Description
Cell_DCH	READY/PMM_CONNECTED	The handset is transmitting or receiving data information using a dedicated channel or a HSPA shared channel.
Cell_FACH	READY/PMM_CONNECTED	The handset had been transmitting or receiving data some seconds ago and due to inactivity had been moved to the Cell_FACH RCC state. Also it's possible that the handset is transmitting or receiving small amount of data like pings, keep-alives, cell updates,...
Cell_PCH/URA_PCH	READY/PMM_CONNECTED	The handset had been in Cell_FACH some seconds ago and due to inactivity had been moved to this less resource consume state. However, the signaling channel is available and is able to change to a data transmission state like FACH or DCH with a little amount of signaling.
Cell_PCH/URA_PCH	STANDBY/PMM_IDLE	The handset is not transmitting nor receiving any amount of data and also the signaling connection is closed.
RCC State	**GMM State (2G/3G)**	**Description**

RCC State	**GMM State (2G/3G)**	**Description**
		However the IP address is maintained by the network and associated to this handset. This is one of the most interesting states since the PDP context is not closed, the IP address is still valid and the handset is not consuming battery, network traffic,... As soon as the handset needs to reestablish the data channel the radio state will be changed to FACH or DCH.
RRC_IDLE	STANDBY/PMM_IDLE	This state is the same as the previous one since the radio state is IDLE.
RRC_IDLE	IDLE/PMM_DETACHED	The handset is not transmitting nor receiving anything and also it hasn't any PDP context established, so no IP address is available for this handset. Normally this state is after 24h of inactivity in the package domain.
RCC State	**GMM State (2G/3G)**	**Description**

Mobile Network. Signaling storms

This is a carrier well-know effect after the big adoption of smartphones around the word.

As we explained in previous sections, each time the network decides to move a handset from one state to another is needed to reestablish channels and starts a negotiation between the network and the handset with the signaling protocol.

Since nowadays handsets are sending keep-alives to maintain their connections opened, the effect is that the handsets is continuously changing from one state to another producing a lot of signaling in the network and also consumes a lot of battery resources.

Mobile Network. Battery consumption

The battery consumption depends on the Radio state. The following list represent the amount of battery needed on each state represented in relative units:

- RRC IDLE: 1 relative unit
- Cell_PCH: < 2 relative unit
- URA_PCH: < or equal than Cell_PCH
- Cell_FACH: 40 relative units

- Cell_DCH: 100 relative units

Chapter 3. Protocol

The protocol is designed to be as simple as possible (KISS philosophy).

Network identification

First of all, mobile device SHALL inform their Push platforms in which mobile network are they connected. To do this, two alternatives can be used:

- MCC - MNC (Mobile Country/Network Codes)
  This pair identifies each mobile network in a unique way.

  The ITU-T Recommendation E.212 defines mobile country codes as well as mobile network codes.

  Complete list of all MCC-MNC pairs [http://dvcs.w3.org/hg/Push/raw-file/default/index.html]

  This scheme is valid if the mobile network has only one segment assigned to the MCC-MCN pair.

- Network ID (DNS Based)
  Since some mobile operators have multiple network segments under the same MCC/MNC pair (because the high number of subscribers), an alternate identification method had been designed.

  This other method identifies each mobile network by a UNIQUE HASH (selected by the MNO) but any RANDOM and UNIQUE string is valid. An UUID is recommended.

  The mobile device SHALL inform his Network ID to the Push platform instead the MCC/MNC pair.

Recovering network identification

MNO will publish the Network ID in the DNS infrastructure so each mobile only need to resolve a simple DNS query to recover the Mobile Network ID

[TO-DO] DNS ENTRY FORMAT

APIs

API: Mobile device -> Push platform

The mobile device SHALL inform his own Push platform each time some network data is changed. (network change, IP change, ...).

The data sent to the Push platform SHALL contain:

- Mobile IP address
- TCP or UDP port in which the mobile agent is listening
- MCC-MNC or Network ID ("Network identification")

API: Push platform -> WakeUp platform (Global node)

This service exposes a REST API in which some resources are shared with other platform services (please refer to "Common REST API resources")

These servers listens in HTTP and not in HTTPS since SSL SHALL be terminated by software (nginx, for example) or hardware equipments

This means, that all HTTP queries to the WakeUp nodes shall contain these extra headers to inform about the client certificate. If these headers are not provided, WakeUp servers will answer with an error (Status code 400).

- X-Real-IP:
  This header will contain the IP address of the client.

  This will be used in log traces.

- X-Forwarded-For:
  This headers contains the different proxies IP addresses used between the client and the server.

- X-Client-Cert-DN:
  This will contain the client certificate DN string used to identify the client in all log traces.

  This will be used in log traces.

- X-Client-Cert-Verified:
  If SSL client verification is valid and SSL terminator rely on this. This header SHALL contain 'SUCCESS'.

  Any other value will be considered an invalid certificate and the connection will be rejected.

REST API: wakeup (global) resource

Resource path is: wakeup/v1

This resource is used to wake up a device. Only POST method is allowed.

POST

The HTTP payload is a parametrized string in "querystring" format:

```
param1=value1&param2=value2&param3=value3
```

- ip:
  IP Address of the client device

  Shall be a valid IPv4 address

- port:
  TCP or UDP port in which the client is waiting wakeup for notifications

  Shall be a number between 1 and 65535

- netid:
  Network ID ("Network identification")

  Shall be a string

- mcc:
  Mobile Country Code (if netid is not provided)

Shall be a number

- mnc:
  Mobile Network Code (if netid is not provided)

  Shall be a number

- proto:
  WakeUp protocol.

  This is optional. By default UDP

  Shall be one of these values:

  - tcp
  - udp

Multiple checks will be done before the client receives the final response.

These errors will inform about different issues:

- Mandatory parameters:
  If the mandatory parameters are not present a status code 400 will be responded.

  Mandatory parameters are: ip and port numbers and netid or MCC/MNC pair.

- Mobile device identification (IP and Port)
  Also the IP parameter will be validated as a valid IPv4 address. The Port shall be between 1 and 65535.

- Mobile network identification
  If the MCC/MNC pair or NetID suplied is not managed by the server (local node not found) a status code 404 (not found) is responded.

- Device in the network
  If the device IP is not inside the mobile network ranges, a status code 400 will be responded (No valid device IP)

- Mobile network status
  If the mobile network local server status is offline for any cause, a status code 503 is returned and should be retried later.

- Protocol accepted
  If the network configuration informs about the optional supported protocols parameter, the protocol queried by the client (proto parameter) will be validated and if the protocol is not accepted by the local node a status code 400 will be sent to the client.

In any other case a 200 OK status code will be responded. This means that the platform will try to deliver the message to the local node, anyway if any error is produced during this process the client won't be informed so it should retry again if the device is not waked up.

REST API: netinfo resource

Resource path is: netinfo/v1

This method informs about all supported networks by this server and the current status of each one.

GET

Only GET method is allowed. The response contains an array with all supported networks with the mcc-mnc of each one, the assigned network id (netid), the IP accepted range or ranges and the current status of the network (offline)

A list of supported protocols can be (it's optional) listed into the protocols array.

```
{"nets": [{"netid":"214-07","mccmnc":"214-07","range":"10.0.0.0/8",
```

Internal API: Local nodes

REST API: wakeup (local) resource

Resource path is: wakeup

This resource is used to wake up a device. POST and GET are supported. GET is supported for backwards compatibility but it's deprecated.

POST

The HTTP payload is a parametrized string in "querystring" format:

```
param1=value1&param2=value2&param3=value3
```

GET

The URL contains a parametrized string in "querystring" format:

```
param1=value1&param2=value2&param3=value3
```

Common querystring for GET and POST methods

- ip:
  IP Address of the client device

  Shall be a valid IPv4 address

- port:
  TCP or UDP port in which the client is waiting wakeup for notifications

  Shall be a number between 1 and 65535

- proto:
  WakeUp protocol.

  This is optional. By default UDP

  Shall be one of these values:

- tcp
- udp

Common REST API resources

REST API: about resource

Resource path is: about

Also, this is the default path, if no other path is defined.

GET

No parameters needed. Returns a HTML information page

REST API: status resource

Resource path is: status

This method is used to check server availability. If, for any reason, it's needed to avoid traffic through one server, this method can set the service as offline.

Can be used by load balancers and by the global checker service

GET

No parameters needed. Returns a 200 OK or 503 Service Unavailable.

POST

Changes the status of the service.

The HTTP payload is a parametrized string in "querystring" format:

```
param1=value1&param2=value2&param3=value3
```

with only one value:

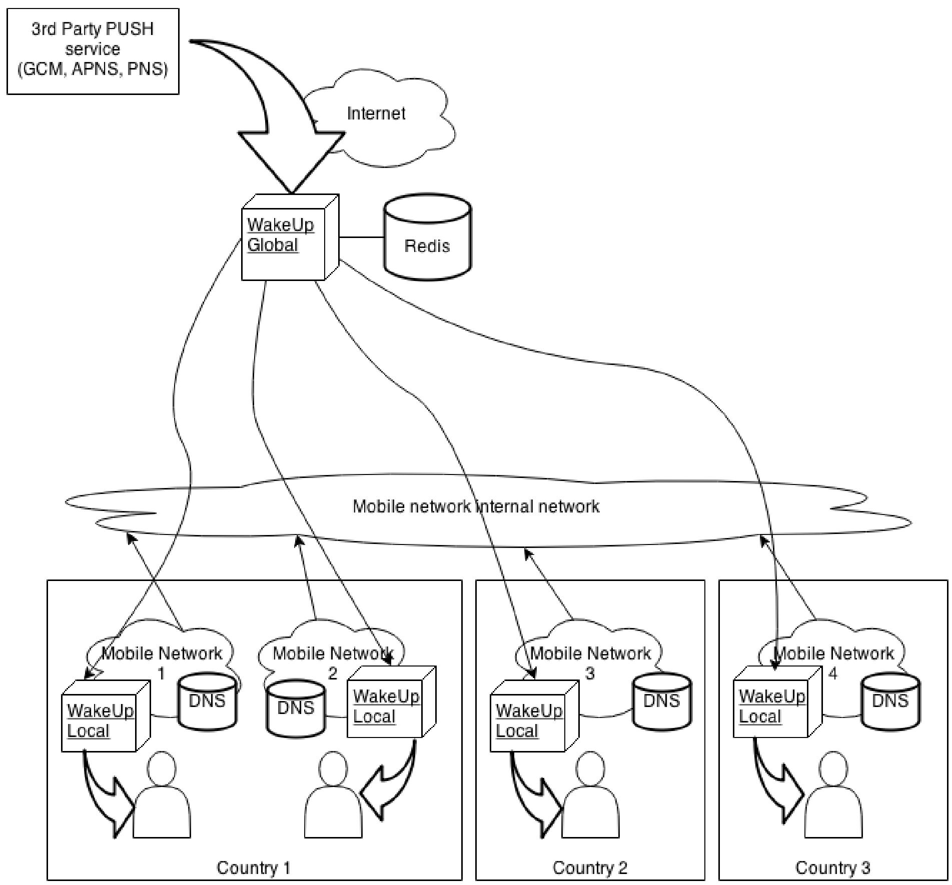
enable = [true | false]

To enable or disable, you can use this simple curl calls:

```
curl -d 'enable=true' -k --key test_ca/TEST_CERTIFICATES/client.key --cert test_ca
curl -d 'enable=false' -k --key test_ca/TEST_CERTIFICATES/client.key --cert test_c
```

Chapter 4. Installation and Configuration

This chapter explains the requirements, to install and configure the wakeup platform correctly



Local nodes

This server shall be installed inside mobile networks so it has a network interface into the same IP segment as mobile phones.

Also it needs a second network interface which connects this nodes to the global one (through operator dedicated networks, VPNs, ...)

Local node will receive HTTP queries to three resources (about, status and wakeup), so any other HTTP query can be securelly filtered.

Finally, this server will send TCP and/or UDP queries to the mobile network.

Global node

This server should be installed in an Internet accesible network but with a SSL Terminator as a frontend.

Global node will receive HTTP(s) queries to three resources (about, status and wakeup/v1), so any other HTTP query can be securelly filtered.

Mobile networks configuration file

Instead a common database, a simple configuration file is used to manage the different local networks.

The networks.json configuration field shall be a well formed JSON file.

The contents shall follow this scheme:

```
{
  "<netidname>": {
    "host": "<local host URL>",
    "range": "<IP Range or ranges allowed in this network>",
    "network": "<MCC-MNC pair>",
    "offline": <true|false>
  },
  ...
}
```

Example:

```
{
  "214-07": {
    "host": "http://localhost:9000",
    "range": "10.0.0.0/8",
    "network": "214-07",
    "offline": true
  }
}
```

See the network collection info here: Networks collection

SSL Terminator

The service is designed to work with a SSL Terminator / SSL Offloader as a frontend, in other words, all connections SHALL be verified with a HTTP SSL Server (Hardware terminators, Apache, NGINX, ...)

These servers will verify client and pass the SSL verification result to the Wakeup servers with these HTTP extended HEADERS:

- X-Real-IP:
  This header will contain the IP address of the client.
- X-Forwarded-For:
  This headers contains the different proxies IP addresses used between the client and the server.
- X-Client-Cert-DN:
  This will contain the client certificate DN string used to identify the client in all log traces.

- X-Client-Cert-Verified:
  If SSL client verification is valid and SSL terminator rely on this. To consider a succesful check this header SHALL be 'SUCCESS'.

  Any other value will be cosidered an invalid certificate and the connection will be rejected.

NGINX example setup

If you decide to use NGINX as a software terminator [http://wiki.nginx.org/SSL-Offloader] , you can use this configuration file as a template.

Into the utils/test_ca folder of main wakeup_platform project (wakeup_platform) you will locate this sample configuration that can be used as a simple reference:

```
# See http://wiki.nginx.org/HttpSslModule
# and http://wiki.nginx.org/HttpProxyModule

server {
    listen        443;
    ssl on;
    server_name example.com;

    ssl_certificate /TEST_CERTIFICATES/server.crt;
    ssl_certificate_key /TEST_CERTIFICATES/server.key;
    ssl_client_certificate /TEST_CERTIFICATES/ca.crt;
    ssl_verify_client on;

    location /global/ {
      proxy_set_header X-Real-IP $remote_addr;
      proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
      proxy_set_header Host $http_host;
      proxy_set_header X-NginX-Proxy true;

      proxy_set_header X-Client-Cert-DN $ssl_client_s_dn;
      proxy_set_header X-Client-Cert-Verified $ssl_client_verify;

      proxy_pass http://localhost:8000/;
      proxy_redirect off;
    }

    location /local/ {
      proxy_set_header X-Real-IP $remote_addr;
      proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
      proxy_set_header Host $http_host;
      proxy_set_header X-NginX-Proxy true;

      proxy_set_header X-Client-Cert-DN $ssl_client_s_dn;
      proxy_set_header X-Client-Cert-Verified $ssl_client_verify;

      proxy_pass http://localhost:9000/;
      proxy_redirect off;
    }
```

```
}
```

Create your own Certification Authority

Using OpenSSL you can create your own CA for testing or deploying purposes

Into the utils/test_ca folder of main wakeup_platform project (wakeup_platform) you will locate this sample script can be used as a simple reference:

```
echo "Generating a test client/server certificates for wakeup platform ..."

CA_SUBJECT='/C=SP/ST=DummyState/L=DummyLocation/O=WakeUpPlatformOrg/CN=wakeupCA'
SERVER_SUBJECT='/C=SP/ST=DummyState/L=DummyLocation/O=WakeUpPlatformOrg/CN=wakeup.
CLIENT_SUBJECT='/C=SP/ST=DummyState/L=DummyLocation/O=WakeUpPlatformOrg/CN=client_

DEST='TEST_CERTIFICATES'
rm -rf $DEST
mkdir $DEST
cd $DEST

echo "Creating CA private key ..."
openssl genrsa -des3 -out ca.key 4096
echo "Creating CA certificate ..."
openssl req -new -x509 -days 365 -key ca.key -out ca.crt -subj $CA_SUBJECT

echo "Creating Server private key ..."
openssl genrsa -des3 -out server.key 1024
echo "Creating Server CSR ..."
openssl req -new -key server.key -out server.csr -subj $SERVER_SUBJECT
echo "Signing Server CSR and creating certificate ..."
openssl x509 -req -days 365 -in server.csr -CA ca.crt -CAkey ca.key -CAcreateseria

echo "Creating Client private key ..."
openssl genrsa -des3 -out client.key 1024
echo "Creating Client CSR ..."
openssl req -new -key client.key -out client.csr -subj $CLIENT_SUBJECT
echo "Signing Client CSR and creating certificate ..."
openssl x509 -req -days 365 -in client.csr -CA ca.crt -CAkey ca.key -out client.cr

echo "Removing server key password ..."
mv server.key server.key.org
openssl rsa -in server.key.org -out server.key

echo "Removing client key password ..."
mv client.key client.key.org
openssl rsa -in client.key.org -out client.key

echo "Generating a PKCS#12 for client certificate"
openssl pkcs12 -export -in client.crt -inkey client.key -out client.p12

cd ..
```

Chapter 5. Operations and administration

This chapter explains how to proceed in case of troubles

Local nodes

If no parameter is passed, the config.default.json file will be used.

If it's desired to use another configuration file, it can be passed through command line or WAKEUP_CONFIG environment variable.

Global node

If no parameter is passed, the config.default.json file will be used.

If it's desired to use another configuration file, it can be passed through command line or WAKEUP_CONFIG environment variable.

Enable/Disable service

Both, local and global nodes, exposes an API which lets operators disable the service so load balancers will not send queries to these ones.

This is based on the status API method (see REST API: status resource section)

If the operator needs to work on a machine, he first should disable the service sending the correct command to the service:

```
curl -d 'enable=false' -k --key test_ca/TEST_CERTIFICATES/client.key --cert test_c
```

After maintainment is finished, the operator will reenable the service again with this other command:

```
curl -d 'enable=true' -k --key test_ca/TEST_CERTIFICATES/client.key --cert test_ca
```

Load balancers will check the response of the GET query to the status URL (https://[SERVER]/status/). If the HTTP status is 200 (OK) the service is enabled. If the response is 503 (Under maintanance) the service is disabled

Data Storage

No additional data storage is used. Only configuration files are needed

Chapter 6. Source Code

This chapter explains how the source code and GIT repositories are organized

If you want to collaborate you should read this chapter.

How it's implemented

This project is implemented using JavaScript on the server (Node.JS)

External dependencies

All node dependecies will be downloaded automatically using NPM

The global node depends in these other projects and versions:

- Node.JS
  Used as the basic framework
- Node.JS log4js Version 0.6.x
  Used as logging library
- Node.JS request Version 2.27.x
  Needed to create new HTTP requests to the different servers
- Node.JS range_check Version 0.0.x
  Used to check if the received IPs are in the same IP range managed by the different local servers

The local node depends in these other projects and versions:

- Node.JS
  Used as the basic framework
- Node.JS log4js Version 0.6.x
  Used as logging library

For development only these libraries and needed too:

- Node.JS
  Used as the basic framework
- Node.JS vows Version 0.7.x
  Used as unit test framework
- Node.JS complexity-report Version 1.0.x
  Used to generate cyclomatic complexity reports
- Node.JS async Version 0.2.x
  Needed to have a more clear code

Database structure

The mobile_networks.js library is used to talk with the database store.

The database store is a simple JSON file called "networks.json"

Networks collection

This collection is the most important one since this hosts each local server information (IP address, IP range of his managed network, current status, ...)

This collection will be queried allways, if the client give us the MCC-MNC pair, the server will generate a generic network id with the MCC MNC pair splitted by a dash. After that, the service will query this collection to find that network.

If networkId is provided in the query, resolution will be faster since only one query to this collection is needed.

Each register is a JSON formatted string with the following attributes:

- host:
  URL to send HTTP queries to local nodes.

  Shall follow the standard URI scheme: http(s)://host:port

- range:
  IP range or ranges accepted into this network.

  Shall follow the standard CIDR scheme: a.b.c.d/n

  An array of ranges is also allowed: ["a.b.c.d/n", "a.b.c.d/n"]

- nework:
  MCC-MNC of this network

  This MCC-MNC can be shared by multiple networks.

- protocols:
  This is an optional parameter.

  It's an array with all supported wakeup protocols. By default tcp and udp should be declared.

  If this data is declared, will be sent to the clients on the netinfo/v1 query and will be used to validate the proto parameter of wakeup/v1 API.

- offline:
  Used by the local server checker to inform about a fail in the local nodes

  If this attribute is not available, it's considered as if the local server is working normally.

  Valid values are allways boolean (true or false)

Some examples:

Local host with issues:

```
"networkLocal": {
  "host": "http://127.0.0.1:9000",
  "range": "10.0.0.0/8",
  "network": "214-07",
  "offline": true
```

```
}
```

Local wakeup without any issue:

```
"networkBTest": {
  "host": "http://127.0.0.1:9000",
  "range": "10.95.0.0/16",
  "network": "002-01",
  "offline": false
}
```

Local wakeup without any issue (or unknown satus):

```
"networkCTest": {
 "host": "http://127.0.0.1:9000",
 "range": "10.0.0.0/8",
 "network": "214-07"
}
```

Repositories and submodules

The project is splitted into several GIT repositories hosted in GITHub:

wakeup_platform

Main repository (for develop teams) which have some testing utils and refers to all other repositories as submodules.

Sources on: https://github.com/telefonicaid/wakeup_platform

wakeup_platform_documentation

This repository hosts the project documentation (this document).

Sources on: https://github.com/telefonicaid/wakeup_platform_documentation

wakeup_platform_global

This repository hosts the Global servers (the one that exposes the public API and the local node status checker)

This repository also dependes (as submodule) on the common one commented bellow.

Sources on: https://github.com/telefonicaid/wakeup_platform_global

wakeup_platform_local

This repository hosts the Local servers (the one that can wakeup devices and shall be deployed in each local OB)

Since this server will be deployed in a critical operator network, we decided to avoid external dependencies so we can manage all the life cycle of the server and improve security.

This repository also dependes (as submodule) on the common one commented bellow.

Sources on: https://github.com/telefonicaid/wakeup_platform_local

wakeup_platform_common

This repository hosts the common libraries used by global and local nodes.

Sources on: https://github.com/telefonicaid/wakeup_platform_common

Chapter 7. Notification Server Architecture

This chapter explains how is the server designed to be able to process millions of messages per second.

Technologies used

The server infrastructure had been build using high performance languages and also high performance database and message queuing systems.

Node.JS

Types of servers

In order to be able to scale horizontally and vertically with no limits all the server platform infrastructure had been splitted in several boxes one of them dedicated to a particular task and also independent of the rest so it can be scalled independently of the rest ones.

The names of each box follows this scheme: WU-<type_of_client>

WU-Global-Server

The WU-Global-Server server is the responsible to intermediate between the central Notification Server infrastructure (3rd party) and each WU-Local-Server deployed in each OB.

This server exposes a public API through Internet which only trusted partners can consume (see Security chapter).

WU-Local-Server

The WU-Local-Server server is a proxy between the global WU servers and the client equipment (device). This service will receive petitions through a standard HTTP port and will send UDP datagrams or TCP packets (for pinging purposes) inside the OB private network to the private IP of the client equipment. This server must be placed inside the OB private network or in a zone that must see that private IPs.

This server is responsible to ping to the correct client inside each OB (using UDP datagrams). It must be placed inside the OB or in a zone that can see the devices inside that private network.

This server will receive the ping orders through a standard HTTP port which will be connected to InterNodo network to receive the data from the VDC inside operator network.

The following scheme shows who the notification is sent with a wakeup server:

Chapter 8. Security

This chapter explains all security topics related with the notification server

The security is managed with Client certificates and only clients with signed key can consume these APIs.

Identify the client

[TBD]

Possible attacks and how to mitigate it

- [TBD]

Chapter 9. License

GNU AFFERO GENERAL PUBLIC LICENSE
Version 3, 19 November 2007

Preamble

The GNU Affero General Public License is a free, copyleft license for software and other kinds of works, specifically designed to ensure cooperation with the community in the case of network server software.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, our General Public Licenses are intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

Developers that use our General Public Licenses protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License which gives you legal permission to copy, distribute and/or modify the software.

A secondary benefit of defending all users' freedom is that improvements made in alternate versions of the program, if they receive widespread use, become available for other developers to incorporate. Many developers of free software are heartened and encouraged by the resulting cooperation. However, in the case of software used on network servers, this result may fail to come about. The GNU General Public License permits making a modified version and letting the public access it on a server without ever releasing its source code to the public.

The GNU Affero General Public License is designed specifically to ensure that, in such cases, the modified source code becomes available to the community. It requires the operator of a network server to provide the source code of the modified version running there to the users of that server. Therefore, public use of a modified version, on a publicly accessible server, gives the public access to the source code of the modified version.

An older license, called the Affero General Public License and published by Affero, was designed to accomplish similar goals. This is a different license, not a version of the Affero GPL, but Affero has released a new version of the Affero GPL which permits relicensing under this license.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU Affero General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This

License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Remote Network Interaction; Use with the GNU General Public License.

Notwithstanding any other provision of this License, if you modify the Program, your modified version must prominently offer all users interacting with it remotely through a computer network (if your version supports such interaction) an opportunity to receive the Corresponding Source of your version by providing access to the Corresponding Source from a network server at no charge, through some standard or customary means of facilitating copying of software. This Corresponding Source shall include the Corresponding Source for any work covered by version 3 of the GNU General Public License that is incorporated pursuant to the following paragraph.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the work with which it is combined will remain governed by version 3 of the GNU General Public License.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU Affero General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU Affero General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU Affero General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU Affero General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY

APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year>  <name of author>

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU Affero General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU Affero General Public License for more details.
```

```
    You should have received a copy of the GNU Affero General Public License
    along with this program.  If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

  If your software can interact with users remotely through a computer
network, you should also make sure that it provides a way for users to
get its source.  For example, if your program is a web application, its
interface could display a "Source" link that leads users to an archive
of the code.  There are many ways you could offer source, and different
solutions will be better for different programs; see section 13 for the
specific requirements.

  You should also get your employer (if you work as a programmer) or school,
if any, to sign a "copyright disclaimer" for the program, if necessary.
For more information on this, and how to apply and follow the GNU AGPL, see
<http://www.gnu.org/licenses/>.
```