Bank-in-a-box: Cclite on Raspberry Pi

As of 17/06/2015 the test version of this has run for ten days and, also, will restart if the system is rebooted. The setup below is a little complex, we will try to automate somewhat in a next step.

Table of Contents

Hardware Used

Raspberry Pi 2 + 32g SD card [it'll be fine on the standard 8g SD card though]
T-Mobile Huawei 1750 or compatible mobile dongle [I bought this on ebay for about £12]

Software Used

Mariadb [instead of Mysql, for obvious reasons, Oracle etc. etc.]
Perl [as before Cclite core is in Perl]
Gnokii and Gnokii.pm Perl module [I've had more luck with this than with Gammu]

How to Install Mariadb

There may be an easier way than this now:

```
https://www.k-dev.de/downloads/
```

```
sudo apt-get install libaio-dev asynch I/o library for MariaDb
sudo wget -O /etc/apt/sources.list.d/repository.pi3g.com.list
http://repository.pi3g.com/sources.list
```

Then add our public key, which will ensure the packages are not intercepted and altered during download. The command apt-key add <keyfile> will let you do just that. Or in one command:

```
wget -O - http://repository.pi3g.com/pubkey | sudo apt-key add -
```

First make sure your system is up to date:

```
sudo apt-get update
sudo apt-get upgrade
```

```
apt-get install mariadb-server
```

Giving Mariadb a user for Cclite

```
pi@raspberrypi /usr/local/mariadb-10.0.14-linux-armv6l $ ./bin/mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 10.0.14-MariaDB MariaDB Server

Copyright (c) 2000, 2014, Oracle, SkySQL Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE USER 'cclite'@'localhost' IDENTIFIED BY 'xxxxxxxxx';
Query OK, 0 rows affected (0.01 sec)

MariaDB [(none)]> grant all on *.* to 'cclite' ;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> quit ;
Bye
```

Cclite Install

See further on, it's better to install `libdbi-perl`, first of all. But the install will complete if done in this order. When there's time, I'll update this document!

```
pi@raspberrypi ~/Downloads $ sudo dpkg -i cclite-0.9.4_all.deb
(Reading database ... 77711 files and directories currently installed.)
Unpacking cclite (from cclite-0.9.4_all.deb) ...
dpkg: dependency problems prevent configuration of cclite:
 cclite depends on libdbi-perl; however:
  Package libdbi-perl is not installed.

dpkg: error processing cclite (--install):

 dependency problems - leaving unconfigured
Errors were encountered while processing:
 cclite
```

Installing libdbi

```
pi@raspberrypi ~/Downloads $ sudo apt-get libdbi-perl
E: Invalid operation libdbi-perl
pi@raspberrypi ~/Downloads $ sudo apt-get install libdbi-perl
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  libdbi-perl
0 upgraded, 1 newly installed, 0 to remove and 50 not upgraded.
1 not fully installed or removed.
Need to get 890 kB of archives.
After this operation, 2,169 kB of additional disk space will be used.
Get:1 http://mirrordirector.raspbian.org/raspbian/ wheezy/main libdbi-perl armhf
1.622-1+deb7u1 [890 kB]
Fetched 890 kB in 1s (516 kB/s)
Selecting previously unselected package libdbi-perl.
(Reading database ... 80104 files and directories currently installed.)
Unpacking libdbi-perl (from .../libdbi-perl_1.622-1+deb7u1_armhf.deb) ...
Processing triggers for man-db ...
Setting up libdbi-perl (1.622-1+deb7u1) ...
Setting up cclite (0.9.3) ...
ERROR: Site debian.default.server does not exist!
apache2: Could not reliably determine the server's fully qualified domain name,
using 127.0.1.1 for ServerName
[Mon Apr 13 10:11:40 2015] [warn] NameVirtualHost *:80 has no VirtualHosts
apache2: Could not reliably determine the server's fully qualified domain name,
using 127.0.1.1 for ServerName
[Mon Apr 13 10:11:42 2015] [warn] NameVirtualHost *:80 has no VirtualHosts
```

Installing libdbd-mysql-perl

```
sudo apt-get install libdbd-mysql-perl
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libmysqlclient16 mysql-common
The following NEW packages will be installed:
  libdbd-mysql-perl libmysqlclient16 mysql-common
0 upgraded, 3 newly installed, 0 to remove and 50 not upgraded.
Need to get 2,029 kB of archives.
After this operation, 4,443 kB of additional disk space will be used.
Do you want to continue [Y/n]? Y
Get:1 http://mirrordirector.raspbian.org/raspbian/ wheezy/main mysql-common all
5.5.41-0+wheezy1 [78.4 kB]
Get:2 http://mirrordirector.raspbian.org/raspbian/ wheezy/main libmysqlclient16
armhf 5.1.62-1 [1,828 kB]
Get:3 http://mirrordirector.raspbian.org/raspbian/ wheezy/main libdbd-mysql-perl
armhf 4.021-1 [123 kB]
Fetched 2,029 kB in 3s (562 kB/s)
Selecting previously unselected package mysql-common.
(Reading database ... 80256 files and directories currently installed.)
Unpacking mysql-common (from .../mysql-common_5.5.41-0+wheezy1_all.deb) ...
Selecting previously unselected package libmysqlclient16.
Unpacking libmysqlclient16 (from .../libmysqlclient16_5.1.62-1_armhf.deb) ...
Selecting previously unselected package libdbd-mysql-perl.
Unpacking libdbd-mysql-perl (from .../libdbd-mysql-perl_4.021-1_armhf.deb) ...
Processing triggers for man-db ...
Setting up mysql-common (5.5.41-0+wheezy1) ...
Setting up libmysqlclient16 (5.1.62-1) ...
Setting up libdbd-mysql-perl (4.021-1) ...
```

Installing libdaemon

This is used to run the Perl gnokii script: `readsms_from_gnokii_batch.pl` as a daemon:

```
sudo apt-get install libdaemon-control-perl
```

Enabling web server configuration and restarting Apache2. You'll need the webserver for management information at present.

```
sudo a2ensite cclite.private.server.conf
Enabling site cclite.private.server.conf.
To activate the new configuration, you need to run:
  service apache2 reload
pi@raspberrypi /usr/share/cclite/config $ service apache2 reload
```

Installing GSM::Gnokii.pm

```
http://search.cpan.org/~hmbrand/GSM-Gnokii-0.09/lib/GSM/Gnokii.pm
```

needs manual install because it fails tests, probably needs Data::Peek and JSON. Works well in spite of all this.

Installing gnokii

```
pi@raspberrypi ~ $ sudo apt-get install gnokii
```

Configuration for Gnokii

```
pi@raspberrypi gnokii --identify
GNOKII Version 0.6.30
IMEI         : 353558048462550
Manufacturer : huawei
No flags section in the config file.
Model        : E1750
Product name : E1750
Revision     : 11.126.03.01.55
```

Udev rule to 'anchor' phone dongle

This is a rule to make sure that the phone dongle is in 'one' place rather than bouncing around USBtty[n], especially when unplugged and plugged.

This is an example, for the Huawei 1750 that I am using, the `idVendor` etc. will depend on the dongle model and can be read from `lsusb`.

```
pi@raspberrypi /etc/udev/rules.d $ cat 99-usb-serial.rules
SUBSYSTEM=="tty", KERNEL=="ttyUSB*", ATTRS{idVendor}=="12d1",
ATTRS{idProduct}=="140c", SYMLINK+="phone"
```

This ties the dongle to `/dev/phone` which is then used in the gnokii configuration file. Afterwards use `sudo udevadm trigger` to fire the rule.

Specific Cclite gnokii Scripts and Configuration

`readsms_from_gnokii.pl` uses this with `Ccsms::Gnokii.pm` This is unlike Gammu processing since the SMS data is in an object rather than being passed via files. Gnokii manipulates the Sim card memory [SM] directly.

Putting the Gnokii config file in the right place for sudo

See gnokii documentation for configuration documentation and examples. This is an example for the Huawei 1750:

```
pi@raspberrypi ~/.config/gnokii $ cat config
[global]
model = AT
port = /dev/phone
connection = serial
```

This is a hack because `Gnokii.pm` seems to expect the configuration file in `$HOME/.config/gnokii/config` and there's no obvious way of changing that.

```
pi@raspberrypi ~ $ sudo su
root@raspberrypi:/home/pi# mkdir -p /root/.config/gnokii
root@raspberrypi:/home/pi# cp .config/gnokii/config /root/.config/gnokii
```

Updating /etc/rc.local

This is to make sure every thing starts and restarts when the Raspberry Pi is rebooted. Put these lines into `rc.local`

```
# start mysql
/usr/local/mysql/bin/mysqld_safe --user=mysql &

# need this to fire udev rules each reboot
sudo udevadm trigger

# start the daemonised version of /usr/share/cclite/cgi-
bin/protected/batch/readsms_from_gnokii.pl
export HOME=/home/pi
sudo /usr/share/cclite/cgi-bin/protected/batch/readsms_from_gnokii_batch.pl start
```

Testing

We're using this for limited SMS testing at the moment. Basically:

- reboot the Raspberry Pi
- use `ps -aef | grep gnokii` and `ps -aef | grep mysql` to make sure daemons are running
- use `gnokii --identify` to make sure that the dongle is being picked up
- use `gnokii` manually, if necessary to debug the network and SMS bit
- try some user joins and payments via SMS

Good luck and questions to: `https://groups.google.com/forum/#!forum/cclite`