/e/ - state of deGooglisation

e Foundation - https://e.foundation

v 1.0 - March 28th 2020 - first version

V1.1 - September 9th 2020 - fixed Connectivity Check issue URL

Summary

/e/ is a mobile ecosystem solution that is made of:

- /e/OS: a mobile operating system, build from a soft-fork of AOSP/LineageOS, that is running on more than 90 devices
- various online services operated under the “Ecloud” umbrella, with strong links to the mobile operating system, build from various web services including NextCloud, OnlyOffice, Postfix, Dovecot and others.

The purpose of /e/ is to offer an alternative mobile ecosystem that is offering more privacy to user’s data, and more specifically that is fully deGoogled.

In a standard AOSP ROM and in most custom AOSP builds, many dependencies to Google services are found. They send some data to Google servers as soon as the user switches the smartphones on. Dependencies to other Google services are found in the middleware, and in higher level layers of the OS.

In the following, we are describing how we are making the /e/ mobile operating system as free from Google data collection as possible.

1- Low level

Connectivity check

In Android, a connectivity check is done to check that Internet can be reached on available networks. The request is expecting a 204 HTTP error code in return and is done against Google FQDNs.

Looking at AOSP source code:

android_frameworks_base ▸ services ▸ core ▸ java ▸ com ▸ android ▸ server ▸ connectivity ▸ NetworkMonitor.java

```
private static final String DEFAULT_HTTPS_URL     =
"https://www.google.com/generate_204";private static final String DEFAULT_HTTP_URL     =
"http://connectivitycheck.gstatic.com/generate_204";private static final String
DEFAULT_FALLBACK_URL  = "http://www.google.com/gen_204";private static final String
DEFAULT_OTHER_FALLBACK_URLS =          "http://play.googleapis.com/generate_204";
```

This means that Google can log a ping from any Android device anytime their network interface goes up, which is a common situation after boot.

Solution: we have replaced the Google server addresses in AOSP source by our own servers that do not logs the connectivity checks, for default calls, and a fallback to a Google server.

- DEFAULT_HTTPS_URL: https://e.foundation/net_204
- DEFAULT_HTTP_URL: https://204.ecloud.global
- DEFAULT_FALLBACK_URL: default Google URL

Merged to the following branches:

- Nougat
- Oreo
- Pie

This is a temporary, not ideal, situation because this solution relies on the confidence users can have in our project and infrastructure.

This will need to be improved using pings to servers FQDN choose randomely from a long list of trustworthy servers, using something different than the 204 HTTP error code check.

References:

- issue 268 on /e/'s bugtracker https://gitlab.e.foundation/e/backlog/-/issues/268

NTP servers

In Android, Google has set their own servers to provide the Time Network Protocol time synchronization Internet service: time.android.com

Looking at AOSP source code:

android_frameworks_base/core/res/res/values/config.xml

```
<string translatable="false" name="config_ntpServer">time.android.com</string>
```

That means that Google can log NTP requests from all Android devices that are connected to Internet.

Solution: we have replaced the Google FQDNs by the usual and standard FQDN for the NTP service on Internet: pool.ntp.org

Merged to the following branches:

- Nougat
- Oreo
- Pie

References:

- issue 272 on /e/'s bugtracker: https://gitlab.e.foundation/e/backlog/-/issues/272

DNS servers

Detailed here for the record: this is not strictly related to "deGooglisation", but has been discussed a lot, and has some significant privacy concerns.

In Android the choice of DNS servers is automatic: DNS servers are set once the Internet connection is initiated to the ISP's default DNS using the DHCP protocol.

Therefore, in /e/ we have implemented a feature available in settings, so that the user can easily edit the IP address of their DNS of choice, with 9.9.9.9 suggested by default, and remember it as a default setting.

Note that our default browser is a fork of Chromium, which by default doesn't use Google DNS over HTTP feature, which would bypass the system DNS setting for web browsing.

References:

issue 269 on /e/'s bugtracker: https://gitlab.e.foundation/e/backlog/-/issues/269

IPv4/IPv6 availability check

While reviewing AOSP's libc (Bionic) source code, we noticed in bionic/libc/dns/net/getaddrinfo.c that the availability of IPv4 vs IPv6 connectivity performed using the _have_ipv4() and _have_ipv6() functions was checked against Google server at addresses 8.8.8.8 and 2000:: respectively.

This issue still has to be addressed in /e/OS.

References:

Android source code at https://android.googlesource.com/platform/bionic/

2- Middleware

Google Play Services

/e/ OS doesn't rely on Google Play services. Instead, we are using the microG open source framework that provides a Google Play Services-like API to Android applications.

This is especially needed for:

- notifications that go through the Google Cloud Messaging service
- geolocation using cell towers and wifi networks. We are using the Mozilla Location service (cell towers only) as database.

We also have Mapbox as default map tiles provider, as a replacement to Google Maps.

This ensures that users are hidden from Google, and do not have to use a personal Google account to use those services.

Synchronization services

/e/ OS also doesn't rely on Google synchronization services or Google accounts for data like documents, pictures, videos etc.

As a replacement, we have developed a synchronization service called “eDrive”, that can synchronize data from and to the /e/ cloud, which is an aggregation of several services, including NextCloud, OnlyOffice, postfix.

Other synchronized data to the /e/ cloud, such as Notes, Tasks, Calendar, Contacts, are not using any Google Services.

Several other synchronization services, that won't depend on Google services are also under development:

- SMS synchronization to /e/ cloud
- geo-location to /e/ cloud

3-High-level

Choice of applications

/e/ has selected a different set of default applications that what can be found on AOSP or most AOSP-based custom ROMs.

We want that to ensure that applications don't rely on Google services, including Google APIs.

Examples:

- for mail, we are using a modified version of K9-mail. We have improved the UI and made it compatible with the /e/ account manager.
- the default messages (SMS) application is a modified version of QKSMS
- the default camera application is OpenCamera
- the default notes application is NextCloud notes
- the default web browser is a fork of Chromium/Bromite. It includes numerous patches to, among others:
    - prevent it to send data to Google (such as browsing history)
    - set a different choice of default search engines: spot.ecloud.global, which is a meta search engine that we operate is the default, followed by Qwant, DuckDuckGo and Cliqz.
- the default Maps application is Magic Earth

App store

/e/ doesn't use Google App Store.

Instead, we have developed a new Android application installer called "Apps", that relies on an independent third-party API that serves most free Android apps, including open source application that can be found at F-DROID.

This application installer requires neither a Google account nor any other account to work.

Online Services

As already mentioned above, /e/ is using default online services for:

- web search
- email
- cloud storage

- calendar
- office
- contacts
- tasks
- notes

All those services are based on various open source components, without any dependency to Google. Those components have been integrated for the most at https://ecloud.global to work using a single /e/ identity under the form username@e.email. The setup is also available for self-hosting.

The /e/ online services currently use:

- a fork of Searx, the meta-search engine operated at https://spot.ecloud.global
- Nextcloud
- Postfix
- Dovecot
- OnlyOffice