The Mobilitics Inria-CNIL project: privacy and smartphones

Vincent Roca, Jagdish Prasad Achara, James-Douglass Lefruit, Claude Castelluccia

The Mobilitics Inria-CNIL project: privacy and smartphones

Privatics team (Vincent Roca) – Inria Grenoble R-A

NB: borrows some results from CNIL (Technical Department / Studies, Innovation and Foresight Department)

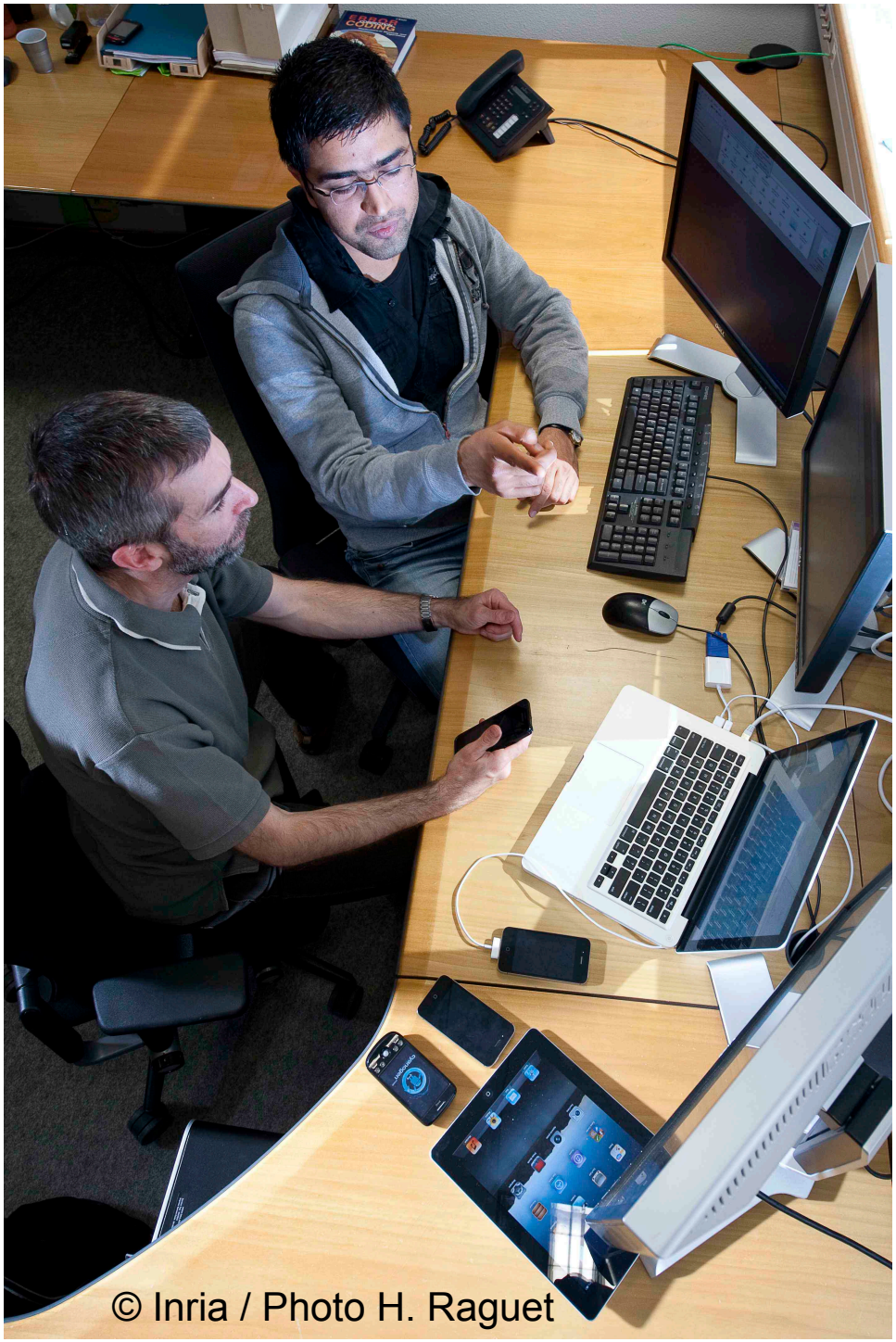
SAR-SSI, Mont-de-Marsan

September 17th, 2013

informatiques mathématiques

Outline

- ***motivations***
- about security/privacy on smartphones
- our methodology
- some results
- conclusions



A pervasive surveillance world

- **Systematic monitoring** of people's actions or communications through the application of information technology
  - we leak data, leave traces when we are browsing the web or using our smartphone…
    - **on the "visible web"**
    - **on the "invisible web"**
  - for **economical** or **security** reasons

Surveillance on the "visible" web

- Foursquare knows **where you are**
- Flickr knows **what you see**
- Facebook knows **what you do**
- Linkedin knows what **you've done**
- Twitter knows **what you say**
- Amazon knows **what you buy**
- Google knows **what you think**
- …

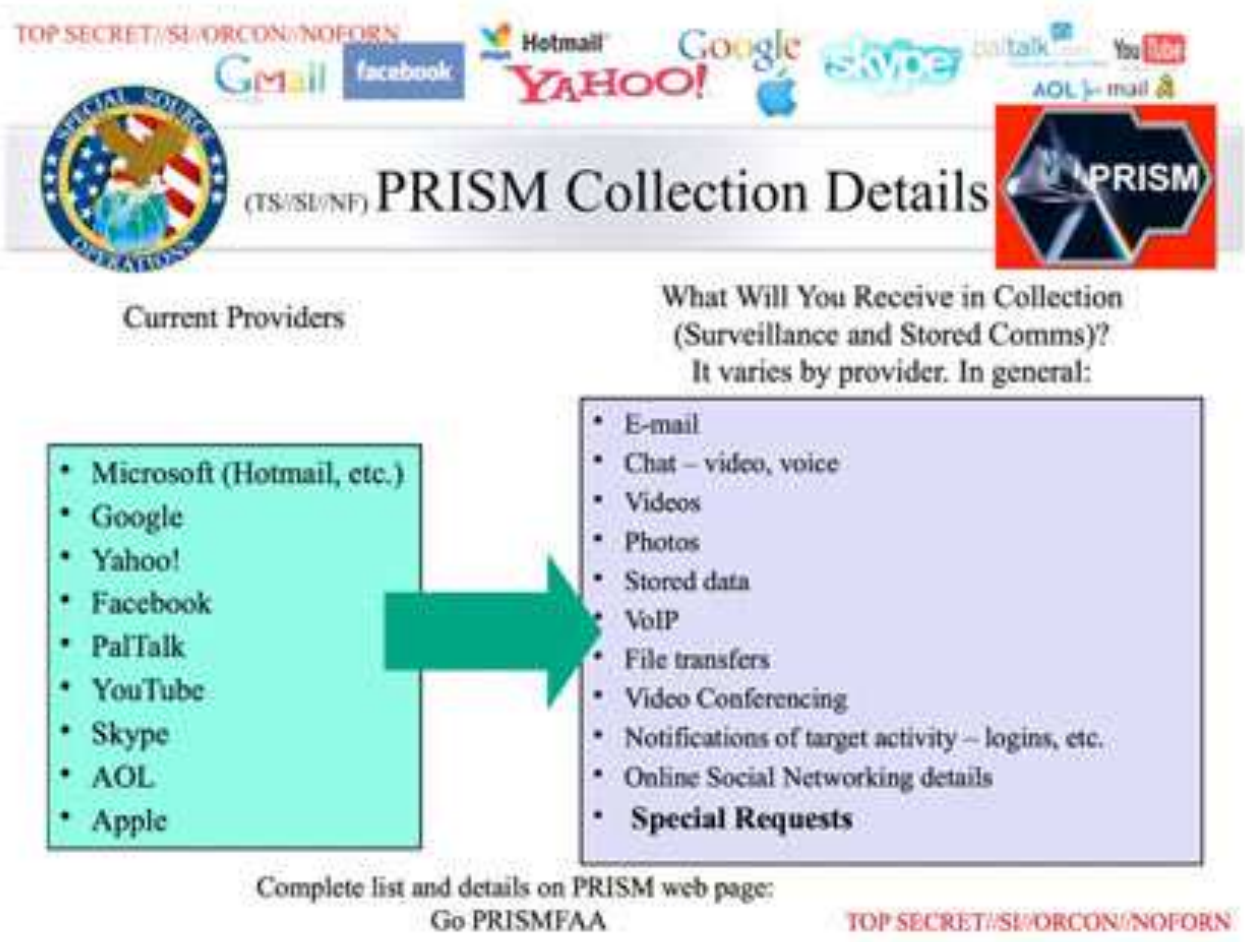
*Courtesy of F. Bancihlon

Surveillance on the "invisible" web

- cookies, tags, pixels, "like" buttons, etc. that appear on websites purposely
  - allows to **track** and to build **profiles** of the users

A situation that easily leads to abuses

- NSA…



- and they're not the only one…

Smartphones have a major responsibility

- they became our companions
  - useful, user-friendly, always connected, easy to customize

- but smartphones know a lot of our cyber-activities
  - they **gather** personal information
    - **while we're using them**
  - they **generate** personal information
    - **GPS, NFC, WiFi, camera**
  - Apps create many opportunities for personal information leakage
    - **it's why some web sites encourage you to install their App!**

Our "personal spy assistant"… (cont')

- a **complex** situation because it involves a large number of actors
  - **first party: App owner ⇒ those you see**
  - **third party: Advertising and Accounting (A&A) ⇒those you never see**
  - **some actors play both roles (e.g. Google, Facebook)**
  - difficult to trust all of them
    - **more to come…**
- Let's see 3 examples of privacy leakage…

Privacy leakage example 1

- spying commercial companies

  http://www.stealthgenie.com
  http://global.ikeymonitor.com

stealthGenie

World's Most Powerful Cell Phone Spy Software

- Protect Child
- Monitor Employees
- Geo-Location & Tracking
- Spy on any Phone

What can you do with iKeyMonitor?

For Parents

Read and report SMS and website logs to email box. Figure out the recent situation of children.

For Employers

Watch the key presses and take screen shots. Detect improper behaviors of employees.

For Spouses

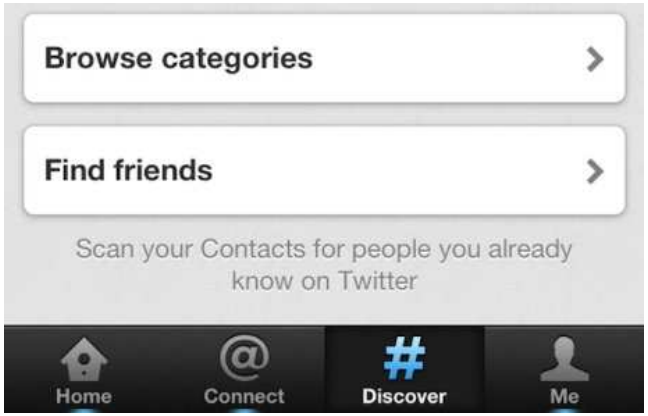
Run in stealth mode and capture everything. Catch cheating spouses or clear suspicions.

For All iOS users

Monitor the activities on the iPhone/iPad you own. Track lost or stolen iPhone and iPad.

Privacy leakage example 2

- Twitter (Feb. 2012):
  - **"La fonctionnalité de recherche d'amis de […] Twitter permet au service en ligne de télécharger sur ses serveurs les carnets d'adresses et la liste de contacts des utilisateurs. Une fois téléchargées sur ses serveurs, ces données sont conservées 18 mois."**



http://www.zdnet.fr/actualites/twitter-copie-et-conserve-18-mois-sans-consentement-les-carnets-d-adresses-des-utilisateurs-39768632.htm

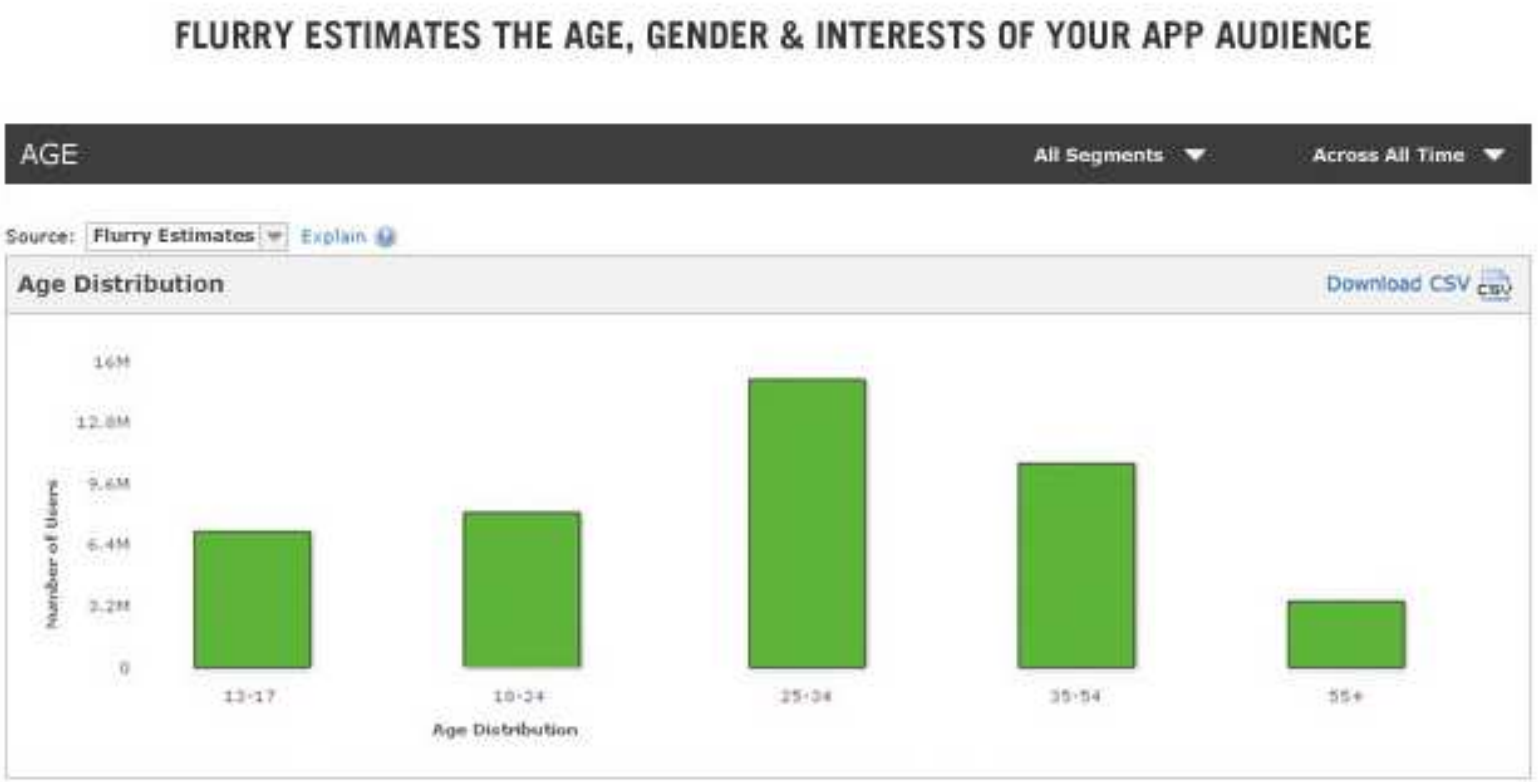
  - **similar scandals with LinkedIn, Path and others in 2012!**

- those are strategic **errors**
  - **big, renown companies have little to gain with such scandals**
  - **corrected promptly in new versions of the app**

Privacy leakage example 3

- data aggregation at Flurry
  - http://www.flurry.com/flurry-analytics.html

The enormous amount of data Flurry handles directly translates into unique, powerful insights for you. The service takes in over 3.5 billion app session reports per day totaling more than 3 terabytes, and our storage is in the petabytes. Here are some examples of how we use big data to create advantages for you:



About Mobile Ads

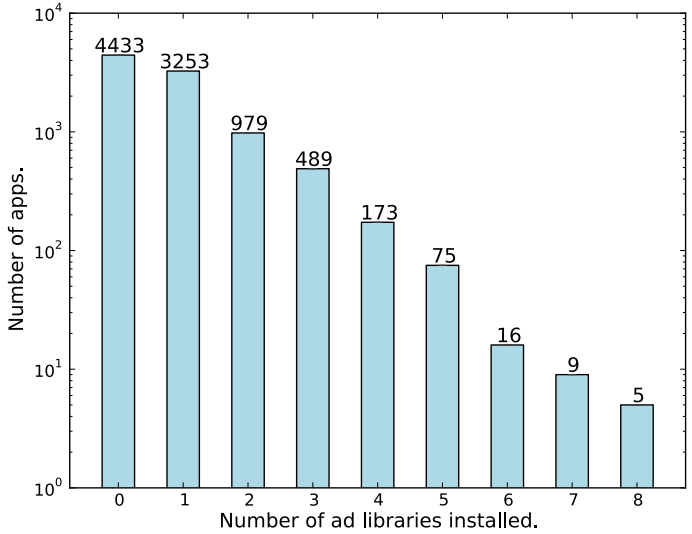
- a way to monetize free (and non-free) Apps
  - makes sense
  - acceptable if done in a ***CNIL-compatible*** way, with informed users

- many mobile world advertising companies

  - plus many others…

About Mobile Ads…

- some facts
  - **"77% of top 50 Android free Apps were Ad supported"** on July 2011 [1]
  - 35% of Android free Apps that use Ads **use 2 or more Ad libraries** [2]
    - **a way to increase revenues**



- ref:
  - **[1] "Don't kill my ads! Balancing Privacy in an Ad-Supported Mobile Application Market", HotMobile 2012.**
  - **[2] "AdSplit: Separating smartphone advertising from applications", Usenix Security 2012.**

About Mobile Ads…

- it does impact the App behavior
  - Ad libs ask for potentially dangerous Android permissions
    - **free Apps usually request 2-3 additional permissions compared to paid Apps of the same category [1]**

Ad Library	Internet	NetworkState	ReadPhoneState	WriteExternalStorage	CoarseLocation	CallPhone
AdMob [22]	✓	✓			○	
Greystripe [25]	✓	✓	✓			
Millennial Media [36]	✓	✓	✓	✓		
InMobi [29]	✓	○			○	○
MobClix [38]	✓	○	✓			
TapJoy [53]	✓	✓	✓	✓		
JumpTap [32]	✓	✓	✓		○	

✓(required), ○ (optional)

permissions per Ad lib [2]

Therefore

- **"tracking the trackers"** is a necessity
  - "teach" companies to behave in a privacy-friendly way
- users must **know** the risks…
  - "teach" the end-user about privacy risks
- users must be able to **control** the risks
  - and give them privacy tools

The Inria-CNIL Mobilitics project

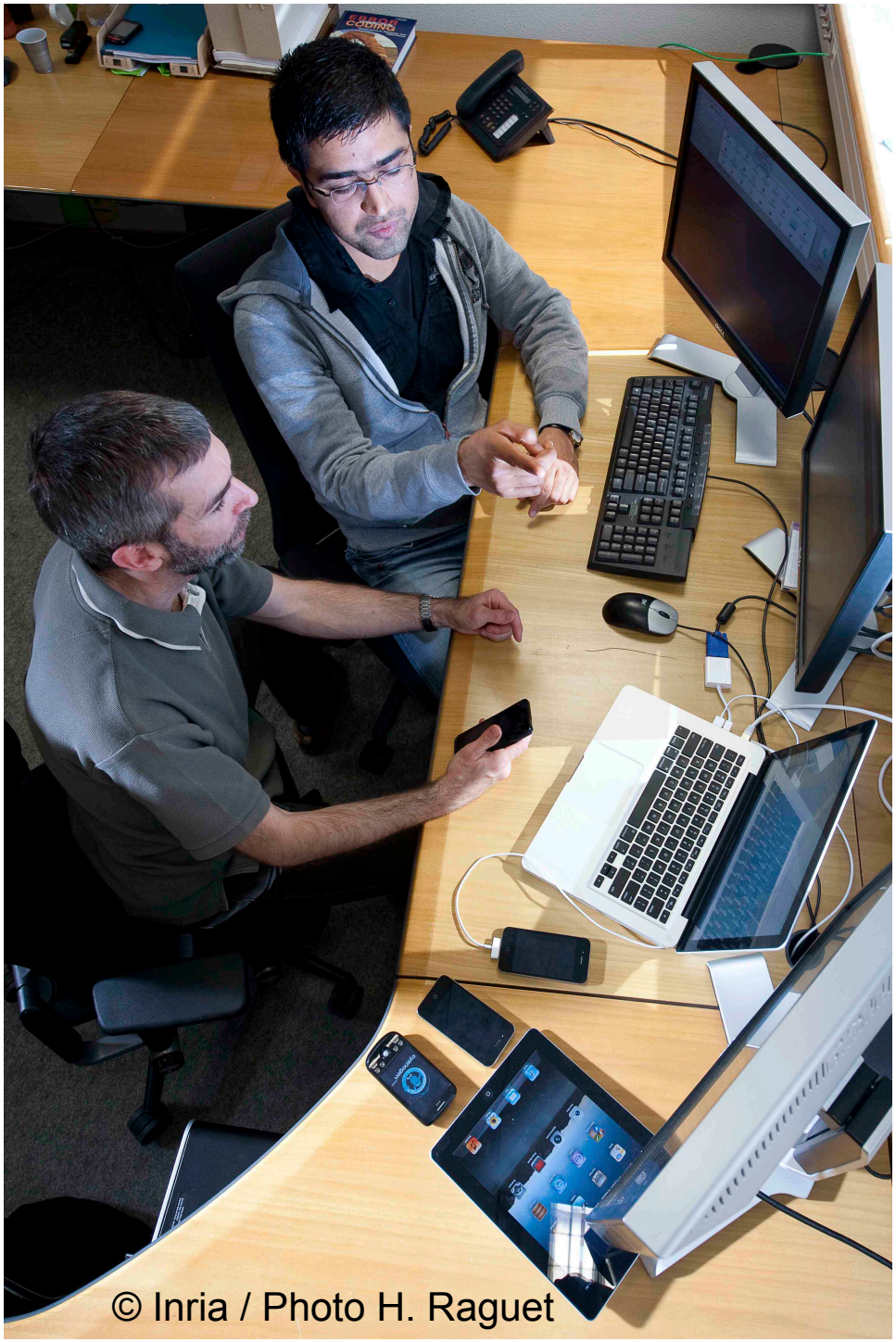
- started in January 2012

- focuses on Android and iOS
  - the leading mobile OS

- analyze privacy leakage by **Apps** and **OS services**
  - compare Android/iOS, identify best practices and trends
- don't be naïve
  - targeted ads can be "the price to pay" for free Apps

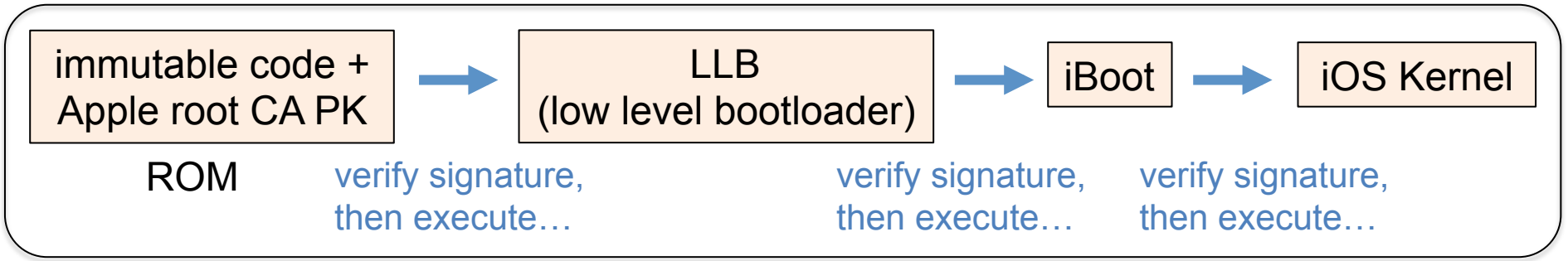
Outline

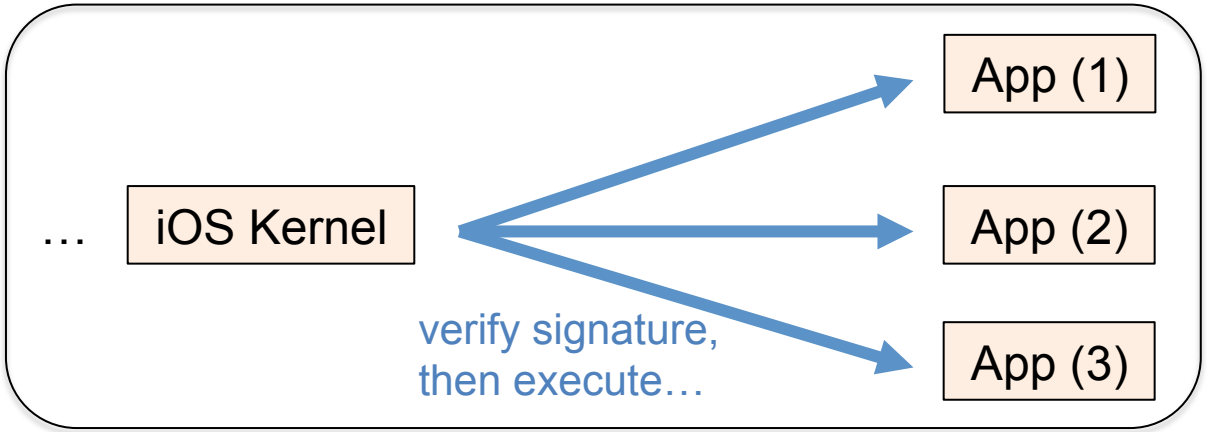
- motivations
- ***about security/privacy on smartphones***
- our methodology
- some results
- conclusions



About security and privacy

- iOS and Android both feature secure boot
  - integrity verification from the bootloader up to Apps





- it looks fine, but it's not sufficient…
  - **does not prevent any App to misbehave**

Issue 1- Apple or end-user must check well

- two different models for App behavior control W.R.T. privacy
  - **market centric:** check an App prior to accepting it on an official market
  - **end-user centric:** ask the user consent when an App wants to perform sensitive operations (at installation time or dynamically)

The market centric approach

- traditionally Apple's approach
  - the only solution in iOS5…
- requires Apple does a good job in **scrutinizing** Apps before accepting them
  - Apple acts as a trusted party
  - we've seen it's **not 100% reliable**
    - **problems come from official signed Apps found in the AppStore…**
  - additionally the validation process is totally **obscure** ☹

The end-user centric approach

- give more control to the end-user…
  - …or get rid of your responsibility as a market validator?
    - **two complementary point of views!**
- Android: at installation time

  - an App with “potentially dangerous requirements” needs to ask the user consent first, at installation time
    - **responsibility is transferred to the user**
    - **example AndroidManifest.xml file**

```
<uses-permission android:name="android.permission.CALL_PHONE"/>
<uses-permission android:name="android.permission.GET_TASKS"/>
<uses-permission android:name="android.permission.READ_CONTACTS"/>
<uses-permission android:name="android.permission.SET_WALLPAPER" />
<uses-permission android:name="android.permission.INTERNET" />
```

    - **can we understand all the consequences of each authorization? No!**
    - **can we control the behavior of the App? Not really!**

The end-user centric approach… (cont')

- iOS6: dynamically
  - done through the privacy dashboard

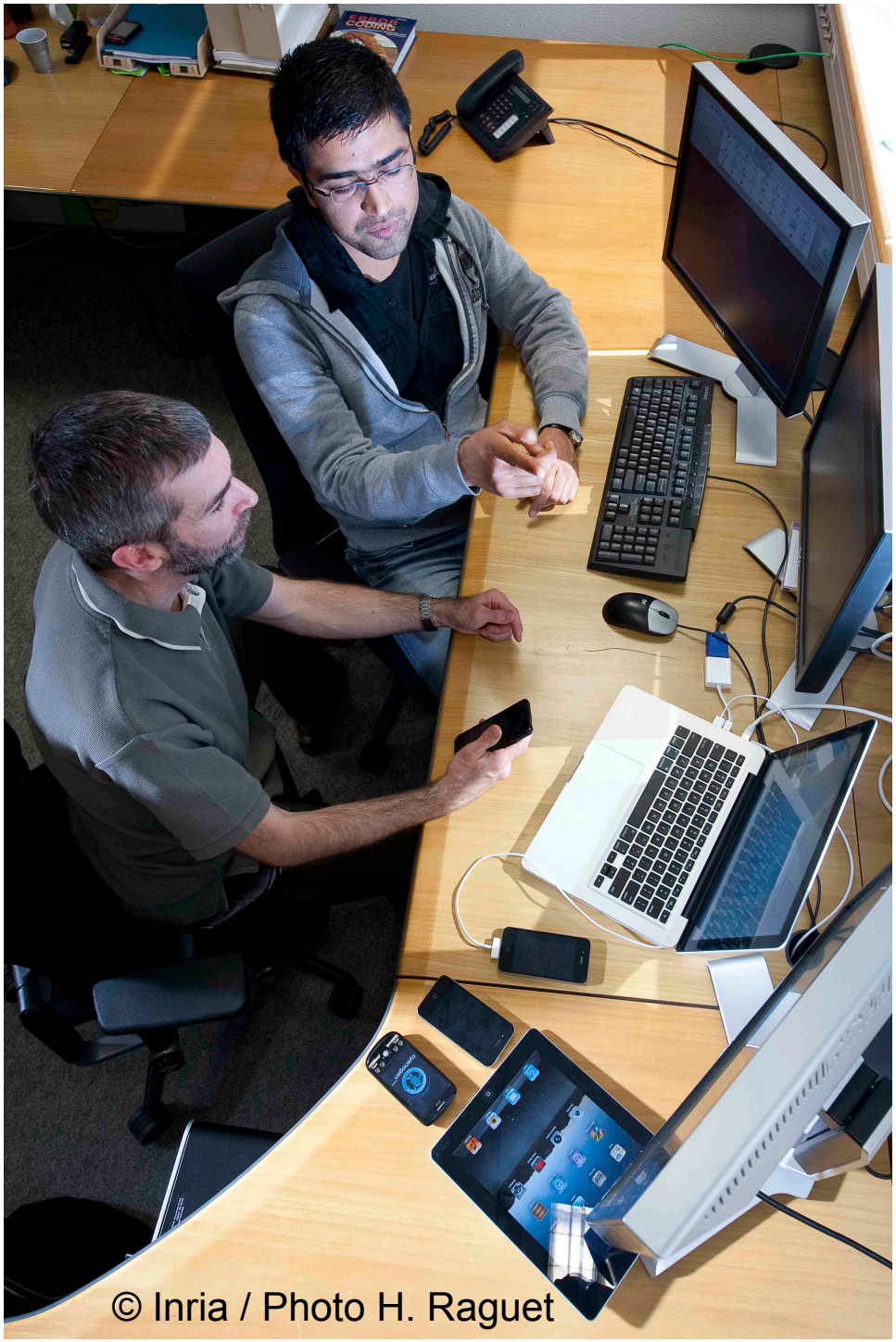
  - but several items are **missing**
    - **Device Name**
    - **UDID (even if banned from new Apps)**
    - **Internet access**
    - **Advertising ID is really hidden elsewhere…**
  - the user cannot control the **behavior** of the App

Issue 2- The consequences of jailbreaking

- why?
  - **"I want to use my device the way I want, rather than the way Apple thinks I want…"**
- jailbreaking an iPhone implies
  - **root access through software or hardware exploits**
  - **patching the kernel to get around Apple's code signature verifications and other restrictions recently added**
  - **as a consequence, the "chain of trust" is broken!**
  - **any App can do whatever it wants**
    - example: keys can easily be compromised if an App acquires a valid entitlement ("keychain-access-groups")

Outline

- motivations
- about security/privacy on smartphones
- ***our methodology***
- some results
- conclusions



Mobilitics step 1: data collection

- a two step process…

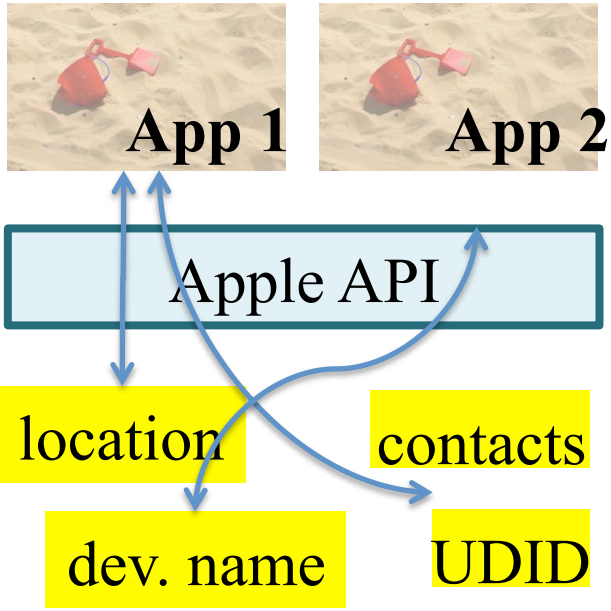
step 1: run Apps on an instrumented version of iOS/ Android, then store data in a local SQLite DB

  - **a lot of data is stored for future, in-depth, post-analysis…**

Instrumenting the OS: general idea

- with iOS:
- each App is independent
  - runs in a dedicated “sandbox”
- accessing external information…

  …requires to use the Apple official API



- ⇒ collecting data is done by instrumenting the API
  - the idea is simple, the difficulty is in the details

- Android: similar

A bit more in details (iOS)

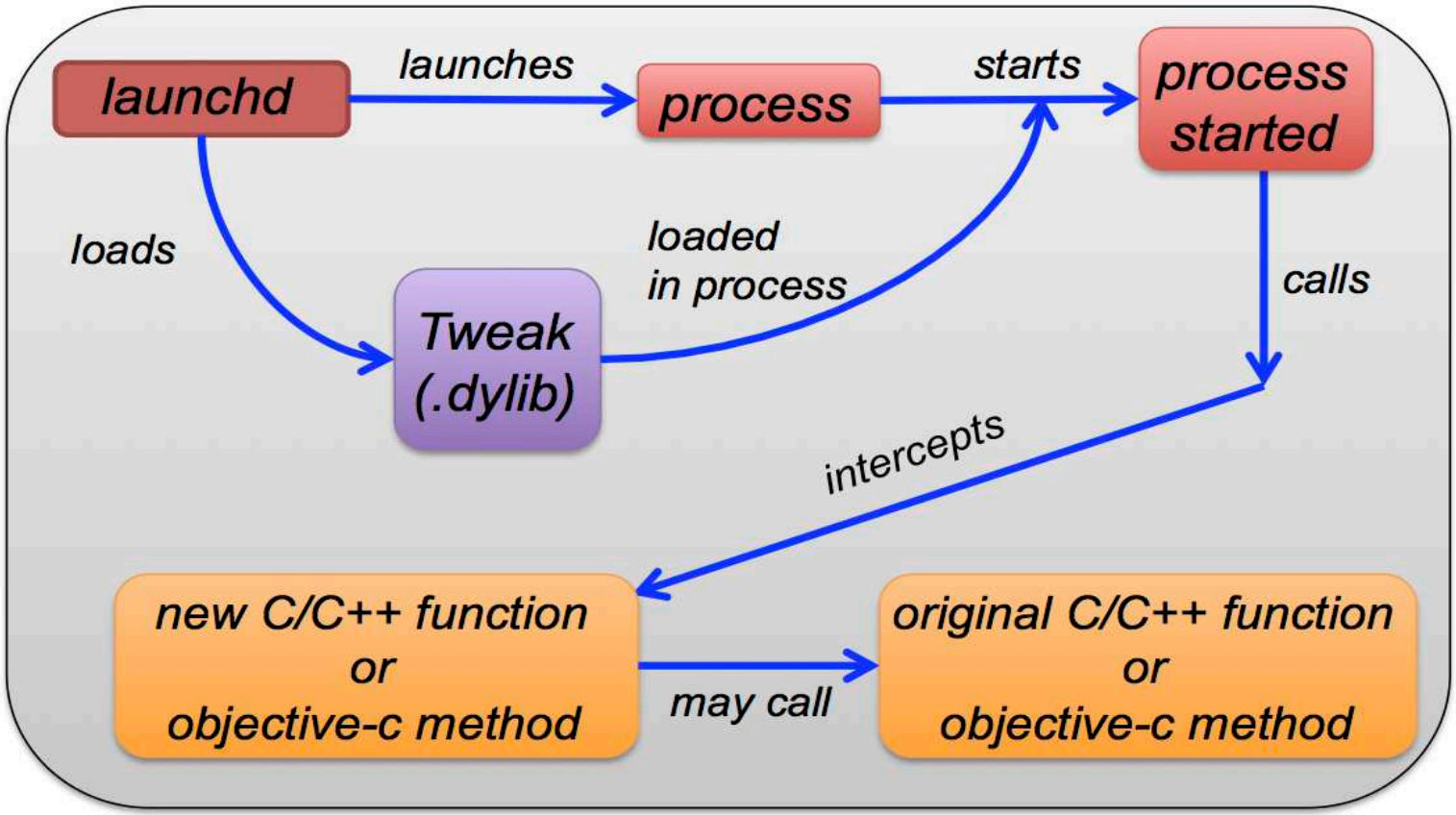
- as only code signed by Apple can be executed, instrumenting iOS requires **jailbreaking** it
  - bypass Apple's secure boot chain

- as App source isn't available, use binary rewriting
  - binary patching? no, it's a nightmare
  - dynamically, at runtime? yes
    - **use Objective-C runtime method "method_setImplementation"**
    - **replace the C/C++ functions at assembly level**

  - **NB: we use MobileSubstrate which makes it lot simpler…**
    http://iphonedevwiki.net/index.php/MobileSubstrate

A bit more in details… (cont')

- the modified implementation of methods is:
  - compiled in a dylib
  - loaded at launch time in a process of interest



Data being collected

- we capture (method args + return values) and store them in SQLite DB for:
  - each **access** to personal data:
    - **contacts**
    - **geographic location**
    - **various device and user accounts**
    - **calendar**
    - **photos and videos**
    - **UDID and device name**
    - **voice memos**
    - **etc.**
  - all **manipulations** method calls (hash and encryption)
  - all **network transmissions** (socket API)

Mobilitics step 2: post-analysis

step 2: off-line, post-analysis of the databases

- search personal information accessed and/or modified and/or sent
- perform statistics / visualization

Doing Post-analysis: general idea

- Identify **private data accessed** by Apps
- Search for private data in the **network traffic** to see if it's sent, and where
- Search for private data in the **input to cryptographic / hash functions**, and if there's some, search the output in the **network traffic**
- Find out if Apps use **cross-App tracking** techniques by using the "UIPasteBoard" class

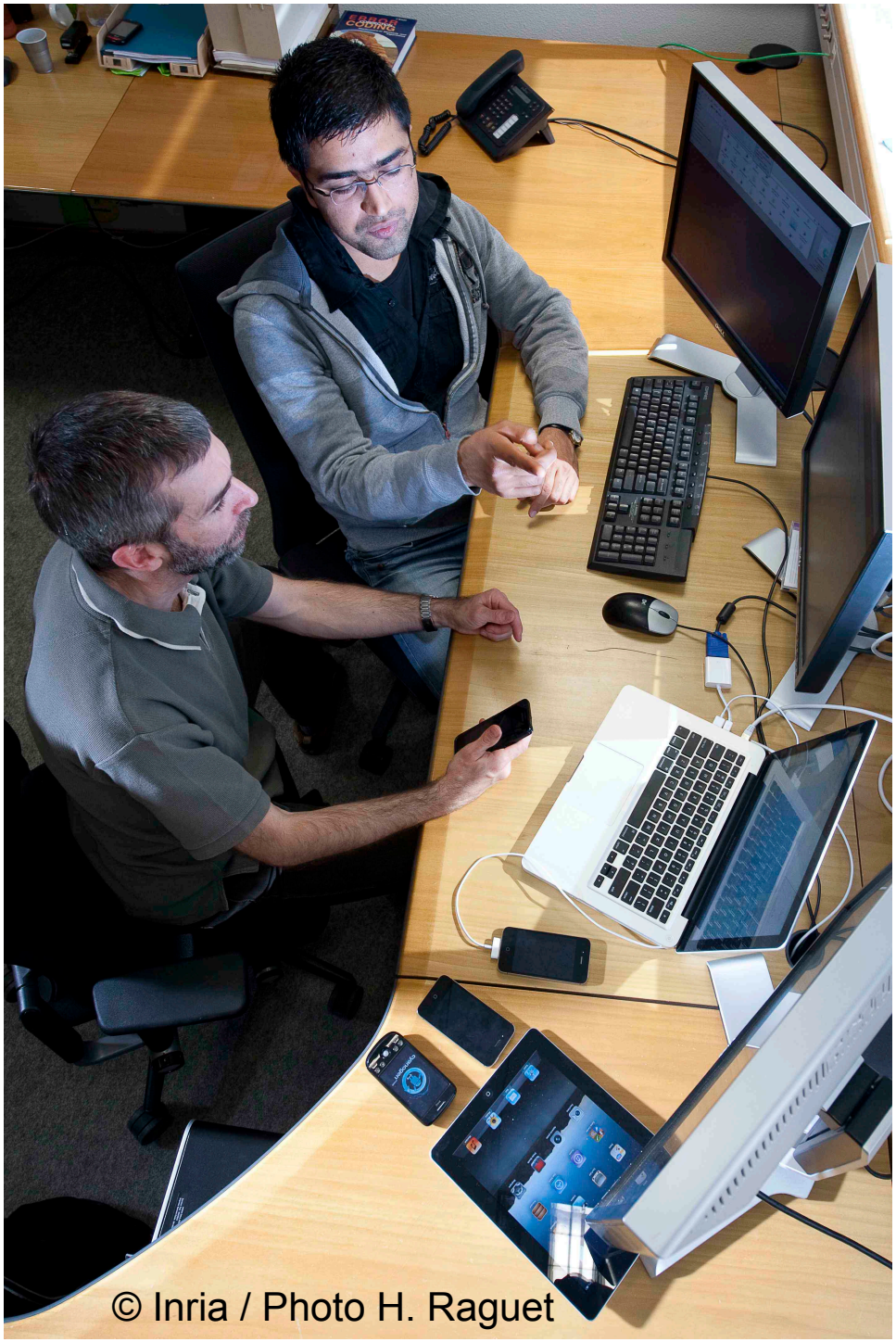
Post analysis limitations

- Are private data manipulations (hash, encryption) done with **internal adhoc functions**…
  - …rather than using standard iOS API?
  - if yes, we cannot detect it as we don't know what to search ☹
  - a simple XOR with a static key is sufficient

- it's a fundamental limitation of our approach
  - hard to evaluate if this is current practice or not
  - But this means…results obtained using our technique would only be lower-bound

Outline

- motivations
- about security/privacy on smartphones
- our methodology
- ***some results***
- conclusions



Quelques résultats: live test 1

- 6 volontaires de la CNIL ont utilisé un iPhone “mobilitics” pendant 3 mois
  - novembre 2012 – janvier 2013
- 9 Go de données récoltées
- 7 millions d’événements récoltés
- 189 applications utilisées

- limite pour ce test :
  - on ne sait pas si les informations personnelles sont transmises dans le cas de flux HTTPS

Statistiques globales

- résultats

origin: CNIL

Nombre d'applications :	Total : 189	
Qui accèdent au réseau	176	93%
Qui accèdent à l'UDID (identifiant unique Apple)	87	46%
Qui accèdent à la géolocalisation	58	31%
Qui accèdent au nom de l'appareil	30	16%
Qui accèdent à des comptes	19	10%
Qui accèdent au carnet d'adresses	15	8%
Qui accèdent au compte Apple	4	2%
Qui accèdent au calendrier	3	2%

TABLEAU 1 – BILAN STATISTISQUE GLOBAL DE L'EXPERIMENTATION MOBILITICS

La Géolocalisation: La reine des données

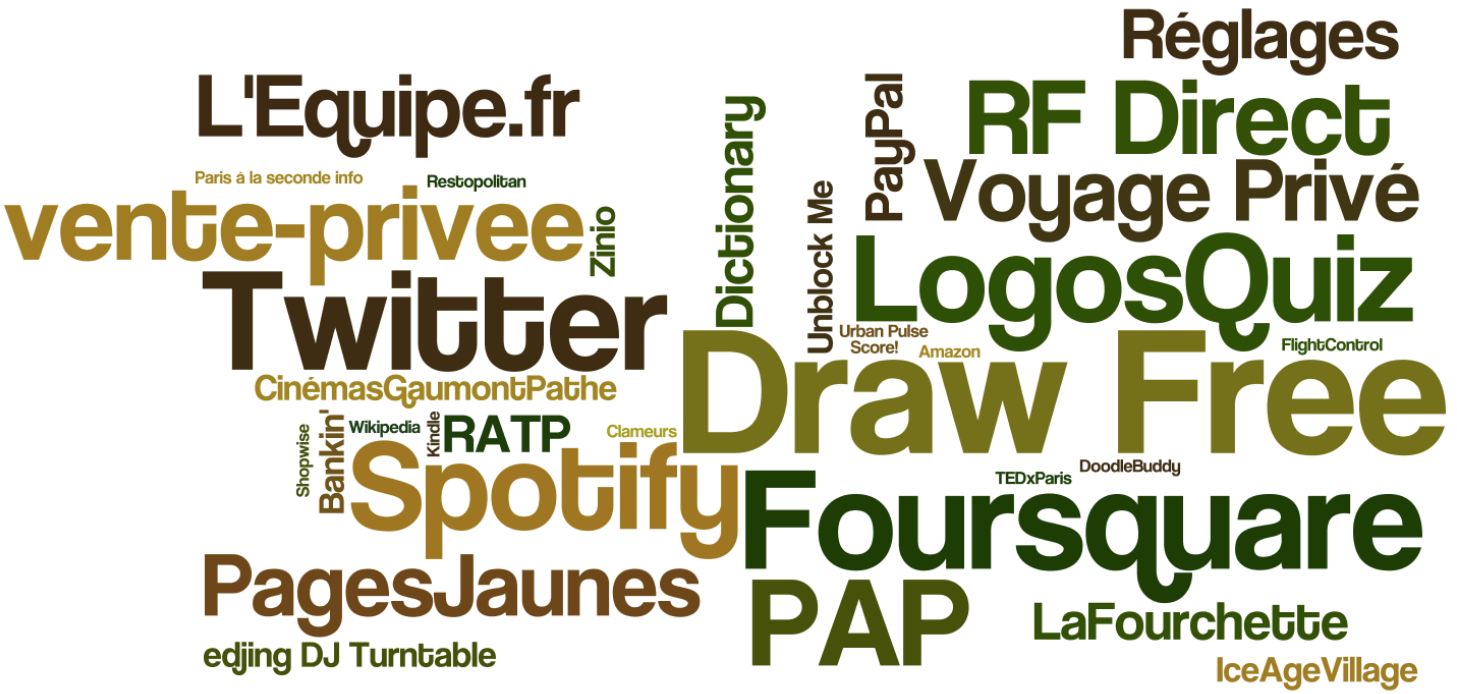
- 31% des applications utilisées ont accédé à la localisation
  - 41 000 « événements » de géolocalisation au total
  - en moyenne 76 événements par jour et par volontaire
  - l'intensité de ces accès surprend

origin: CNIL

Pourquoi accéder au nom de l'appareil?

- 36 applications, soit un peu plus de 15% ont accédé à cette info
  - l'usage fait de cette donnée est peu clair

origin: CNIL



Les identifiants sont très demandés

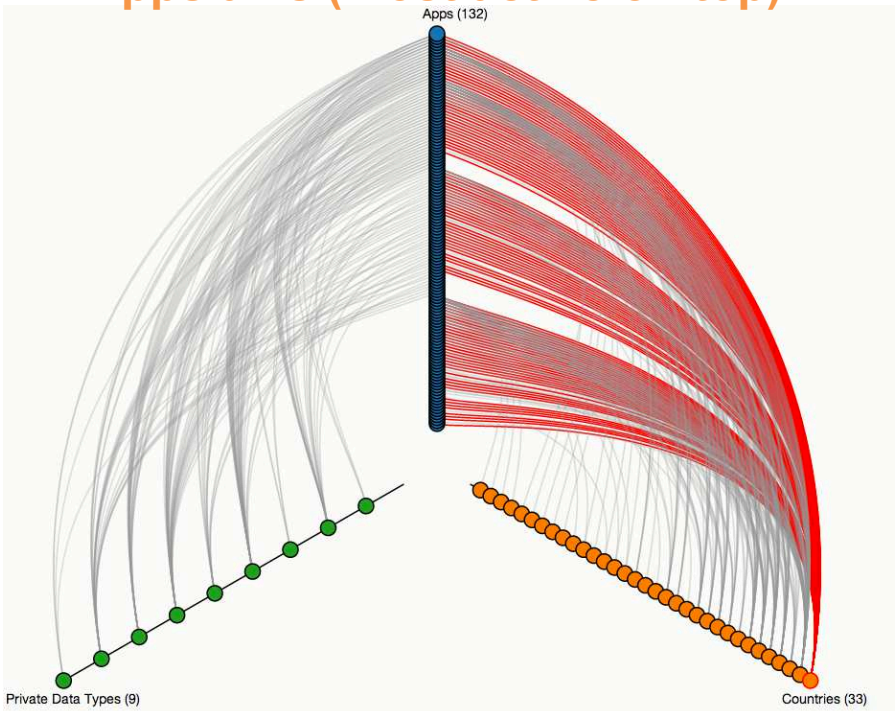
- l'UDID, un élément clef
  - **idf intégré à l'iPhone qui n'est ni modifiable ni effaçable**
- Cet UDID est très « demandé »
  - **87 applications sur 189 ont accédé à l'UDID (46%)**
- désormais banni, mais d'autres solutions sont là…
  - **ex. OpenUDID, adresse MAC, IMEI, etc.**

origin: CNIL

Example DB visualization tool…

- per country view

Apps axis (most active on top)



Private data axis

Country axis (most active on the right)

Country:

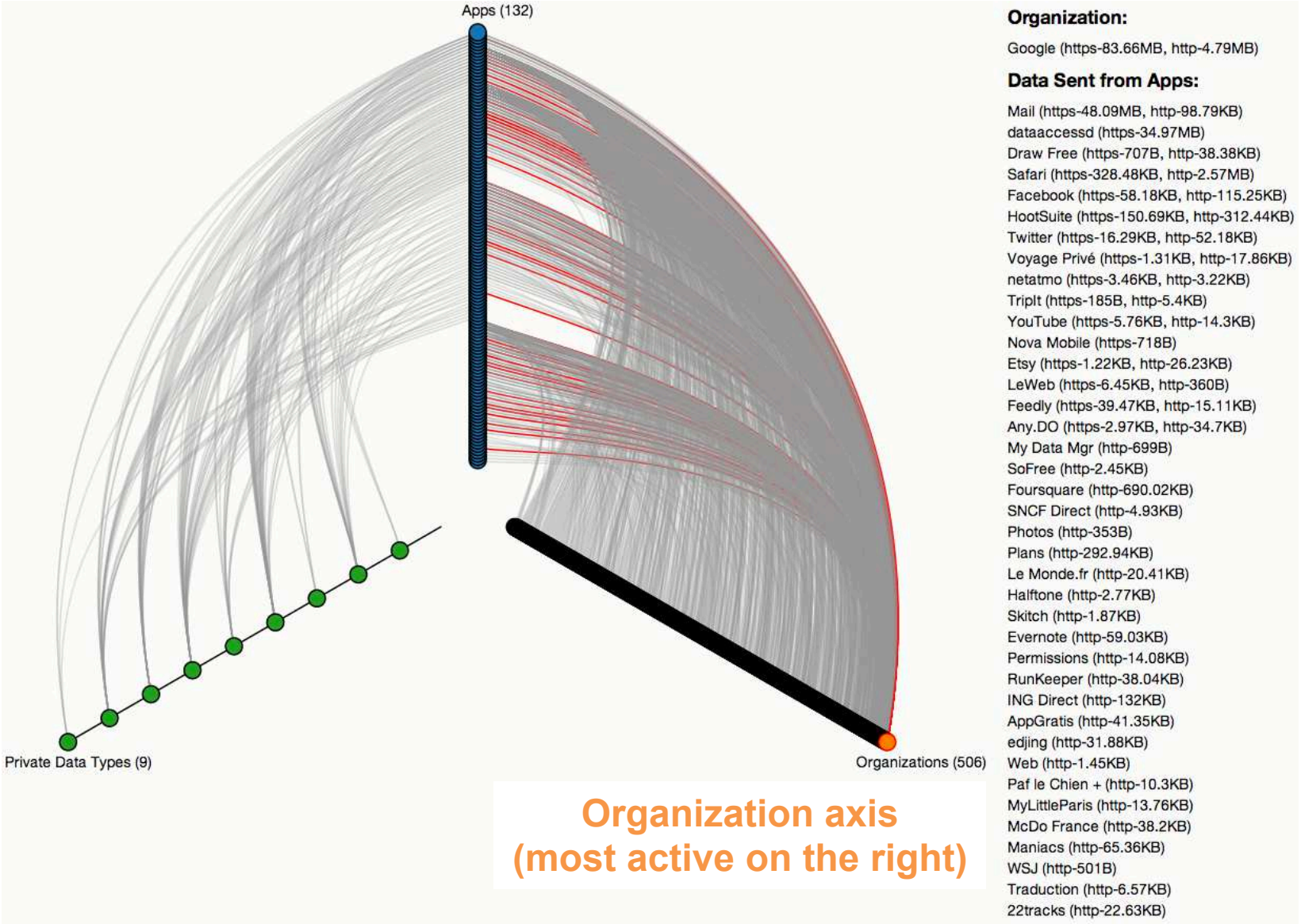
United States (5122-4.27KB, 5223-6.23MB, 13392-3.23KB, 33033-2.42KB, 40002-275B, 40003-165B, 40004-203B, 40005-539B, 40006-73B, 40007-344B, 40010-80B, 40011-363B, 40012-1.29KB, 40013-3.28KB, 40014-176B, 40016-9.99KB, 40017-450B, 40019-233B, 40020-458B, 40021-71B, 40022-290B, 40023-12.42KB, 40024-110B, 40025-145B, 40026-122B, 40027-409B, 40028-4.19KB, 40029-285B, 40030-516B, 40031-1.84KB, 40032-1.94KB, 40033-290B, 40034-10.6KB, 40036-2KB, 40038-1.72KB, 40039-175B, 40040-423B, 40041-129B, 40042-285B, 40043-1.16KB, 40044-1.6KB, 40046-169B, 40047-354B, 52775-205B, https-144.74MB, http-25.05MB, http-alt-593B, jabber-client-182B)

Data Sent from Apps:

Urban Pulse (5122-4.27KB, http-8.97KB)
apsd (5223-6.23MB, https-513.08KB)
Skype (13392-3.23KB, 33033-2.42KB, 40002-275B, 40003-165B, 40004-203B, 40005-539B, 40006-73B, 40007-344B, 40010-80B, 40011-363B, 40012-1.29KB, 40013-3.28KB, 40014-176B, 40016-9.99KB, 40017-450B, 40019-233B, 40020-458B, 40021-71B, 40022-290B, 40023-12.42KB, 40024-110B, 40025-145B, 40026-122B, 40027-409B, 40028-4.19KB, 40029-285B, 40030-516B, 40031-1.84KB, 40032-1.94KB, 40033-290B, 40034-10.6KB, 40036-2KB, 40038-1.72KB, 40039-175B, 40040-423B, 40041-129B, 40042-285B, 40043-1.16KB, 40044-1.6KB, 40046-169B, 40047-354B, 52775-205B, https-186.86KB, http-1.81KB)
ubd (https-355.88KB)
backupd (https-21.54KB)
SpringBoard (https-6.77KB, http-749.92KB)
Mail (https-48.55MB, http-1.08MB)
aosnotifyd (https-368.27KB)
PMP (https-14.48MB)
itunesstored (https-3.28MB, http-419KB)
dataaccessd (https-40.88MB, http-154.08KB)
My Data Mgr (https-5.88KB, http-699B)
12 jours (https-2.17KB)
Météo (https-185B, http-32.03KB)
syncdefaultsd (https-280.33KB)
Draw Free (https-211.83KB, http-1003.84KB)
configd (https-10.97KB, http-4.35KB)
locationd (https-2.51MB)
Shazam (https-17.06KB, http-37.16KB)
Réglages (https-17.69KB)
imagent (https-990.3KB, http-30.3KB)
Safari (https-1.54MB, http-6.7MB)
Facebook (https-9.99MB, http-801.14KB, http-alt-593B)
HootSuite (https-4.75MB, http-3.19MB)
[illegible] ttp-898.57KB)
0.46KB)
92.94KB)
http-17.86KB)
p-241.46KB)
.87KB)
Apple Store (https-13.13KB, http-3.15KB)
RATP (https-9.1KB, http-18.65KB)
Yammer (https-64.11KB)
Evernote (https-5.89MB, http-82.49KB)
Résultats foot (https-750B)
Instagram (https-82.36KB, http-5.08MB)
SoundCloud (https-837.38KB, http-637.85KB)
netatmo (https-3.46KB, http-3.22KB)
RunKeeper (https-444.41KB, http-38.4KB)
iTunes (https-29.85KB, http-7.6KB)
vente-privee (https-71.58KB, http-624B)
Shopwise (https-122.09KB, http-34.35KB)
TripIt (https-30.1KB, http-98.19KB)
L'Œil by LaSer (https-7.13KB, http-308B)

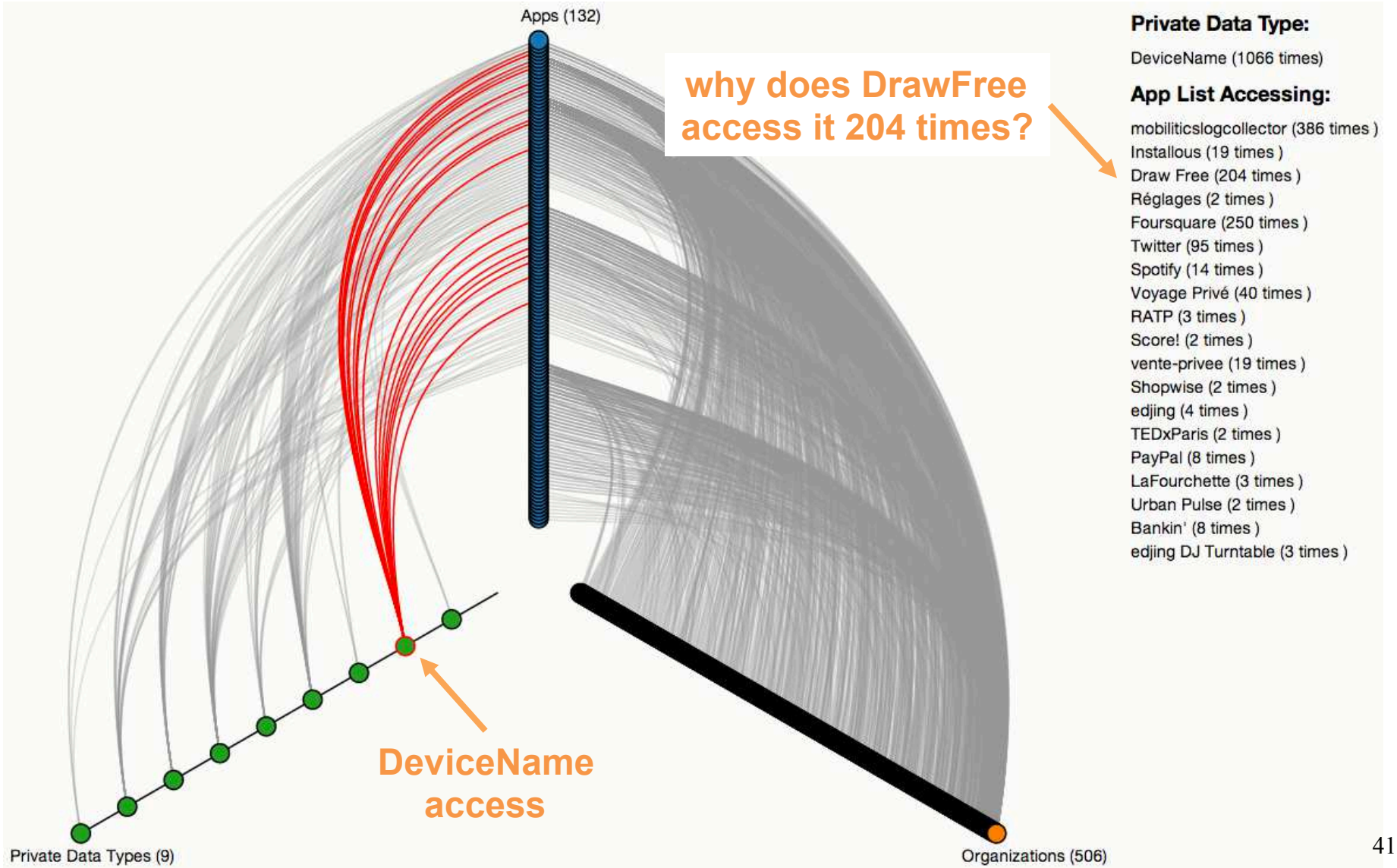
Sensitive DB visualization tool…

- per organization view



Sensitive DB visualization tool…

- an example: “DeviceName” access view



Some results: test 2

- in depth analysis of selected Apps
  - we chose 78 representative free Apps in several categories
  - goal is to identify personal information access/ modification/transmission
  - currently only iOS6, soon comparison with Android4

- ***question:*** what are the available long term or (better) permanent identifiers available to trackers?
  - required for long term tracking of a smartphone

Synthesis table

Table 1: No. of Apps accessing, modifying and sending Private Information out of a total of 78 Apps tested

Private Information (PI)	Total No. of Apps accessing PI	Total No. of Apps sending unmodified PI	Total No. of Apps modifying PI	Total No. of Apps sending modified PI
UDID	17	0	0	0
Accounts	2	0	0	0
AdIdentifier	36	26	18	0
Location	12	4	0	0
IdentifierForVendor	25	3	11	0
DeviceName	11	2	0	0
AddressBook	2	2	0	0
ProcessNames	NA	4	0	0
Carrier Network	NA	6	2	0
WiFiMACAddress	NA	6	47	17
BluetoothMACAddress	NA	0	17	0
SerialNumber	NA	0	17	0

(NA means figure is not available (accessed through sysctl))

- 41 App are using UIPastboard to share a permanent ID between Apps

On the efficiency of the AdvertisingID

- 59% Apps bypass the official iOS6 “AdvertisingID”
  - made possible by the use of a “permanent” ID
    - **MAC address, device name, OpenUDID**
  - the AdID is supposed to let the end-user control tracking by resetting it as desired…
  - … it’s just an illusion ☹
- 37% Apps will still bypass the AdID with future iOS7 that bans the access to MAC address
  - this % will increase as more companies will shift to other types of permanent identifiers for tracking

© Inria / Photo H. Raguet

CONCLUSIONS

Il y a du travail pour améliorer la situation…

- Apple/Google sont contraints…

  … de proposer des techniques pour redonner du contrôle à l'utilisateur : « privacy dashboard » (iOS), autorisations (Android)

- Mais :
  - elles sont **peu utiles** en l'état
    - **limitées**
    - **contrôle à gros grain**
    - **conséquences des autorisations obscures**
    - **pas d'analyse comportementale des App**

Il y a du travail… (cont')

- **induisent en erreur** car elles sont contournées
  - **« si c'est techniquement possible, j'ai le droit de le faire »**
    - le "AdvertisingID" d'iOS6 est quasi inutile et trompeur
    - peu de progrès à attendre avec iOS7
  - **Apple connait la situation!**
  - **NB: analyse en cours pour Android…**

- règne un **flou total**
  - **un développeur qui inclue une bibliothèque publicitaire ne sait rien de son comportement…**

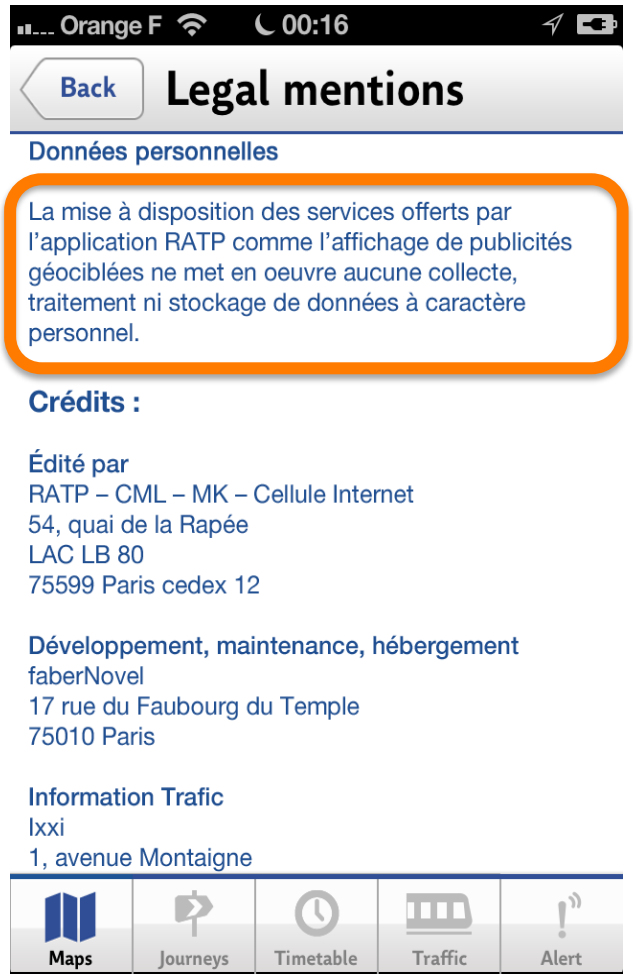
- beaucoup d'actualités sur le thème
  - NSA, récemment paypal, lecteur d'empreintes dans l'iPhone5s…

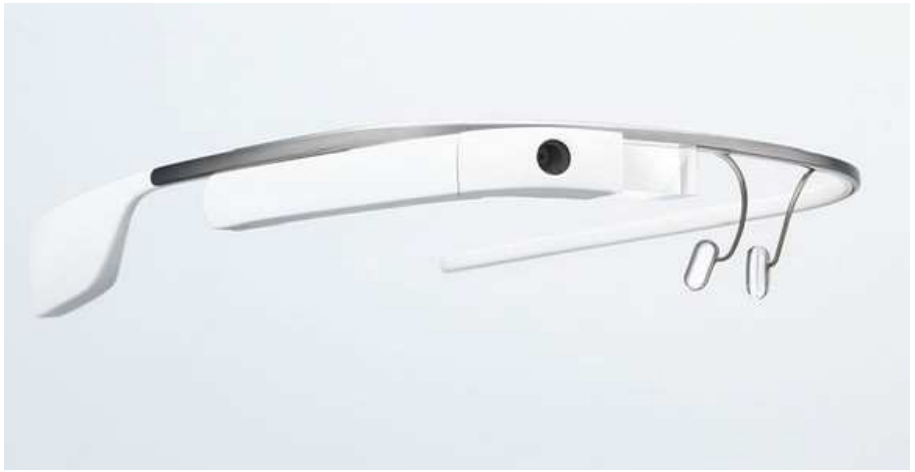
Un cas d'école : l'App RATP version 5.4.1

- « Y'a pas de problèmes » dixit la RATP

- Vraiment ?
  - la liste des Apps actives, mon adresse MAC, le nom de mon téléphone, ma position géographique précise (à 20m près), un identifiant permanent sont envoyés à Adgoji (ssl) et sofialys (en clair !)

- Voir notre blog : part-1 et part-2: https://team.inria.fr/privatics/





Thank you ☺

informatiques mathématiques