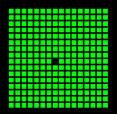
# Data protection by design and by default

Using Elm and its very strict ecosystem
to comply with the upcoming EU GDPR

2017-07-27, July Meetup @ Rocket Labs



# Overview

- About me (very shortly)
- Elm Open Source projects
- Background:
  - Lets recap: What is purity
  - Why is purity relevant for you (EU GDPR)
- Why Are So Many Smart People So Stupid About the GDPR?
- Data protection by design and by default (Elm to the rescue)
- Q & A



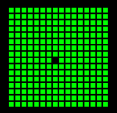
# About me (very shortly)

- Ramón Soto Mathiesen (Spaniard + Dane)
- MSc. Computer Science ***DIKU/Pisa*** and minors in Mathematics ***HCØ***
- ***CompSci*** @ SPISE MISU ApS
  - ***"If I have seen further it is by standing on the shoulders of giants"***
    ***-- Isaac Newton*** (Yeah Science, ... Mostly mathematics)
  - ***Elm*** (***JS*** due to ports) with a bit of ***Haskell*** and a bit of ***F#*** (fast prototyping)
- Elm / Haskell / TypeScript / F# / OCaml / Lisp / C++ / C# / JavaScript
- Founder of Meetup for F#unctional Copenhageners (MF#K)
- Volunteer at Coding Pirates (Captain at Valby Vigerslev Library Department):
- Blog: http://blog.stermon.com/ and Twitter: @genTauro42

# Elm Open Source projects

- Elm packages, released under LGPL-3.0 (http://package.elm-lang.org/):
  - spisemisu/elm-utf8
  - spisemisu/elm-sha
  - spisemisu/elm-bytes
  - spisemisu/elm-merkletree
  - spisemisu/elm-blockchain (coming at some point)
- Elm SPAs, released under GPL-3.0:
  - Fair Choice (http://spisemisu.com/spa/fairchoice/)
  - Sign Sign (sign2x) (coming at some point https://sign2x.org)
  - UniverΣ (UniverSum) (coming at some point https://universigma.org)

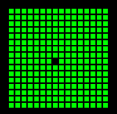




# What is purity

- Lets recap:
  - Functions ***always evaluates*** to the ***same output*** value ***given*** the ***same input***
  - Evaluation does ***not cause*** any ***side effect***, such as mutation of ***mutable objects or*** output to ***I/O*** devices
  - Functions can be mapped directly to ***λ-calculus***, which is mathematically ***pure***.
  - The result of ***combining pure functions***, would still be considered ***pure***





# Why is purity relevant for you

- Purity it's not just ***academic mumbo jumbo***
- ***Privacy-by-design and default***, get used to it as ***General Data Protection Regulation*** (GDPR) arrives next year:
  - Doom-day: **2018-05-25**
- Easiest way to comply with this approach is by ***isolating your side-effect***. Languages supporting this feature at the moment are: ***Elm***, ***Haskell***, ***Idris***, ***PureScript***, among others

# Why is purity relevant for you

version2 IT-NYHEDER BLOGS IT-JOB SEKTIONER MERE

## Mega-svipser: CPR-numre og skatteoplysninger frit tilgængelige på Skats hjemmeside

**Skatteoplysninger og CPR-numre har været frit tilgængelige via Skats hjemmeside som følge af en fejl.**

Jakob Møllerhøj Fredag, 10. marts 2017 - 12:44 33

En fejl har bevirket, at danske CPR-numre med tilhørende skatteoplysninger i går aftes var tilgængelige for uvedkommende på Skats hjemmeside.

»I går erfarede vi, at vores leverandør har lavet en alvorlig fejl. Og den fejl betød desværre, at enkelte borgeres oplysninger var synlige for andre. Og de oplysninger var blandt andet deres CPR-numre og deres årsopgørelser. Det siger sig selv, at det er fuldstændigt uacceptabelt for Skat, hvis den enkelte borgers oplysninger har været synlige for andre end dem selv,« siger kontorchef i Skat Jørgen Wissing Jensen.

Kontorchef Jørgen Wissing Jensen, Skat: Hvis nogen har været inde og lave ændringer i de oplysninger, der har været vist, så er det et forhold, vi tager meget alvorligt

Skat oplyser, at de blotlagte skatteoplysninger og CPR-numre tilhører »en lille rådgivningsvirksomheds kunder.« Skat ønsker ikke at oplyse, hvilken rådgivningsvirksomhed, der er tale om.

En læser, der ønsker at være anonym, har tippet Version2 om fejlen. Han fortæller, at han loggede ind på Skats selvbetjeningsløsning i går aftes omkring klokken 21.

Efter at have klikket lidt rundt, blev han pludseligt præsenteret for en liste med andres borgeres CPR-numre.

Og via CPR-numrene var det muligt at klikke sig videre ind og se de bagvedliggende skatteoplysninger i form af årsopgørelser.

28. FEB. 2017 KL. 18.54 | OPDATERET 01. MAR. 2017 KL. 08.52

## Private oplysninger er ude efter stort læk hos Novo Nordisk

E-mail-adresser, navne og telefonnumre på ansøgere har ligget frit tilgængeligt på Novo Nordisks hjemmeside.


Novo Nordisk har oplevet et datalæk, hvor ikke-sensitive informationer utilsigtet blev lagt på novonordisk.com, skriver selskabet. (Foto: liselotte sabroe © Scanpix)

PRINT

DEL ARTIKLEN:

MAIL

Ved en fejl har en række oplysninger fra op mod 95.000 jobansøgere fra forskellige lande ligget frit tilgængeligt på Novo Nordisks hjemmeside.

# Why is purity relevant for you

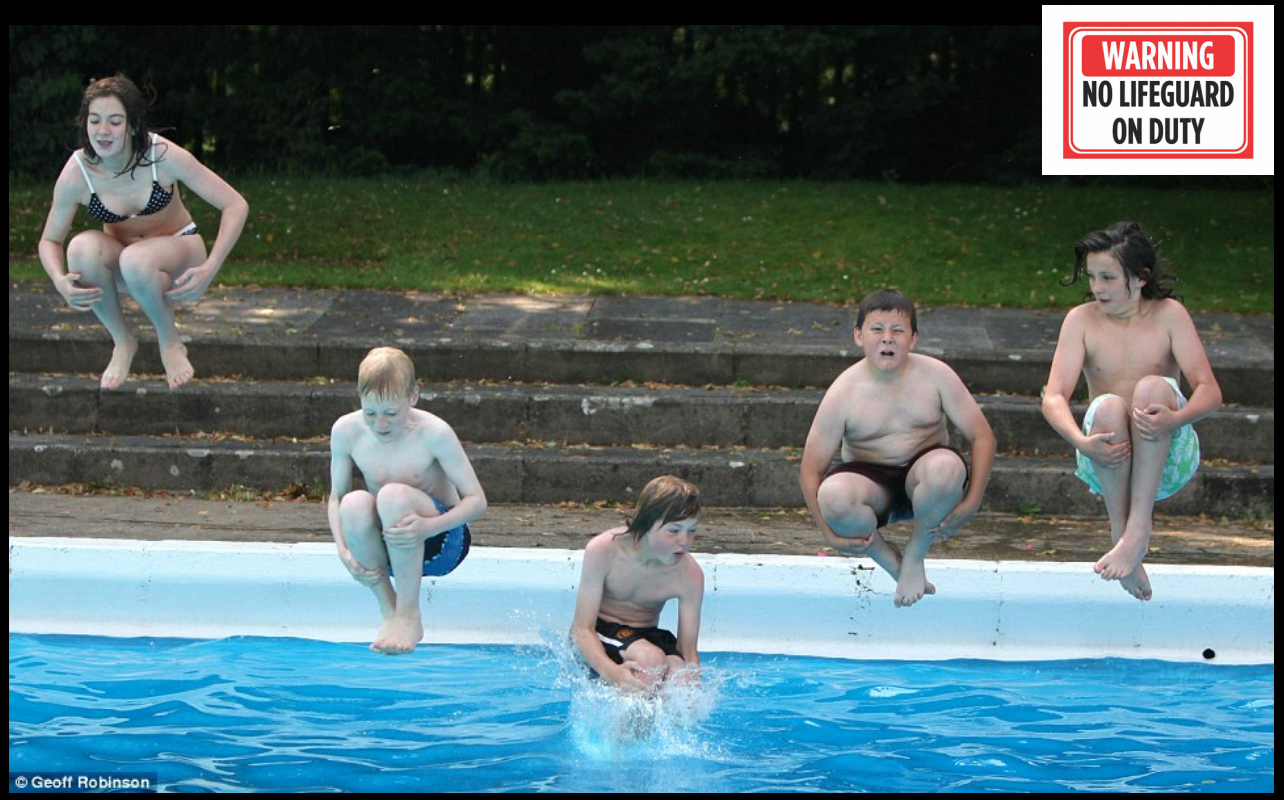
- We recently had two cases where sensitive data was leaked through web applications:
  - SKAT (Danish Ministry of Taxation)
    - Some citizens, when login in, could choose other citizens profiles, presented in a list, like admin mode
  - Novo Nordisk (Denmark's Top 2 greatest company, turnover/revenue: 107.927 mDKK)
    - 95.000 job applicants data (name, phone, e-mail, ...) was published to their main website (human error)
- What if it was next year, both blamed their software provider? (Sanctions)
  - Fines in the size of 10/20 mEUR or 2%/4% annual worldwide turnover (whichever is greater)

**Note**: turnover (UK)/revenue (US) reference to the amount of money a company generates without paying attention to expenses or any other liabilities

# Why is purity relevant for you


# Why is purity relevant for you


# Why is purity relevant for you

# Why is purity relevant for you

- Are you willing to deliver software from ***Doom-day*** next year?
  - How are you going to ***convince*** your customers that you are doing everything to ensure that no ***unwanted*** side-effects and hereby data-leaks will occur?
- Lets remove the ***blame-game*** and the ***say a lot but do nothing*** from the equation and focus on solving the real problem, ***with science ofc***
- By ***isolating unwanted side-effects*** at ***compile-time***, ***no system*** will be ***deployed to production with undesired vulnerabilities***
- You will just need to request ***pure code*** through signatures files from your contractors or software providers (next slide)

# Why is purity relevant for you

```
namespace EvilCorp
  module BusinessLogic =
    (* Connect to 3rd party and leak data, perfectly fine *)
    val foo : int -> int
    (* Create folder and log sensitive data, fine as well *)
    val bar : float -> float
```

vs

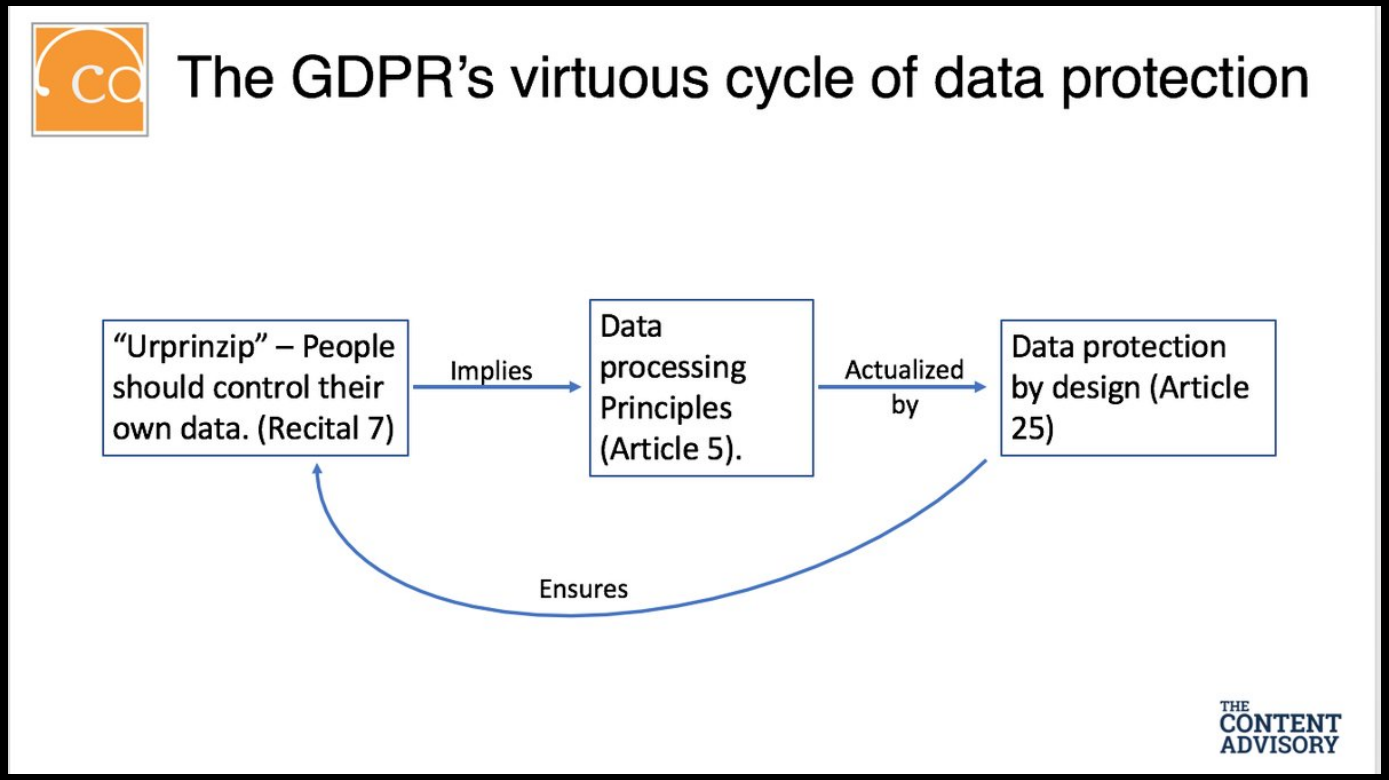
```
namespace CantBeEvilCorpAnymore
  module BusinessLogic =
    (* Connect to 3rd party and leak data, Computer Says No *)
    val foo : int Pure -> int Pure
    (* Create folder and log sensitive data, Computer Says No *)
    val bar : float Pure -> float Pure
```

“Don’t be evil” enforced by code !!!

# Why is purity relevant for you

- Just think of it in Simon P. Jones (SPJ) terminology:
  - Isolate side-effects to avoid “Launching the missiles”
  - Isolate side-effects to avoid “Leaking data”
- By enforcing purity, the “Volkswagen emissions scandal” (dieselgate), would never have been possible as the Governments could just require that car manufactures software, complied with their signatures files

# Why Are So Many Smart People So Stupid About the GDPR?




LinkedIn Post (Tim Walters, Ph.D.)

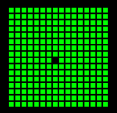
# Why Are So Many Smart People So Stupid About the GDPR?

- 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**

# Why Are So Many Smart People So Stupid About the 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***"

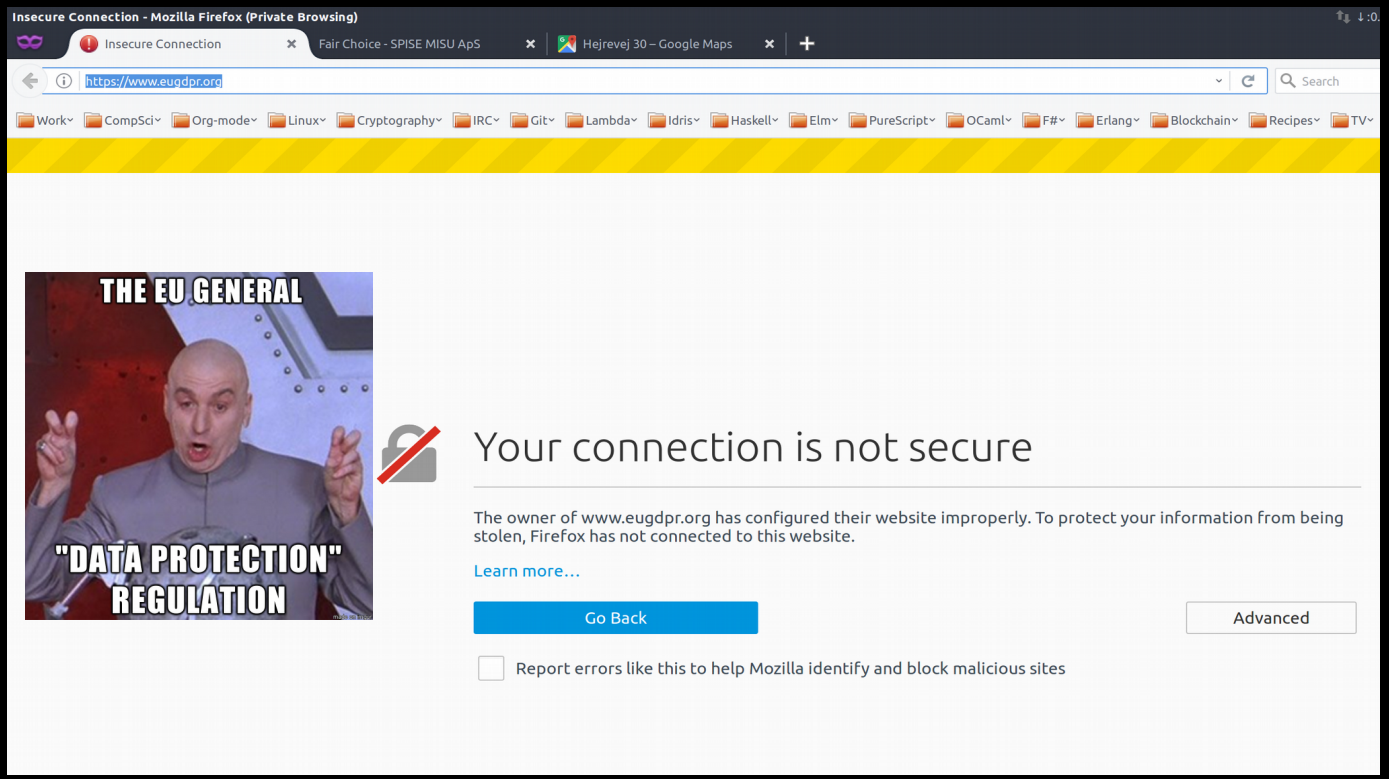




# Data protection by design and by default (Elm to the rescue)


Sadly, the world we live in … Snyk Blog

# Data protection by design and by default (Elm to the rescue)



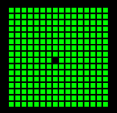

and HTTPS done wrong: EU GDPR official website :(

# Data protection by design and by default (Elm to the rescue)

```elm
module DataRegister exposing (Register, Sensitive(..), add, count, get, init)

import List

type Sensitive a = Sensitive a
type Register  a = Data (List a)

init : () -> Register a
init () = Data []

add : a -> Register a -> Register a
add x (Data reg) =
    if List.any (\y -> x == y) reg then
        Data reg
    else
        Data (x :: reg)

count : (a -> Bool) -> Register a -> Int
count cond (Data reg) =
    let
        xs =
            List.filter cond reg
    in
        List.length xs

get : (a -> b) -> (a -> Bool) -> Register a -> Maybe b
get dto cond (Data reg) =
    case reg of
        [] ->
            Nothing

        x :: xs ->
            if cond x then
                Just (dto x)
            else
                get dto cond (Data xs)
```

Very basic Data Register





# Data protection by design and by default (Elm to the rescue)

- Very basic Data Register
- Combined with the very strict Elm ecosystem:
  - Pure code with isolated side-effects
    - **Sum Type** of **Msg** that triggers some **Cmd msg**. Just focus on **update** function
  - Pure packages
    - No referential transparency, unless it's Evan/Elm Team (DOM/Native stuff)
  - Semantics versioning (**semver**) almost done well
    - They actually use Syntactic versioning (**synver**): elm-sha issue #3
- Demo: Fair Choice, powered by RANDOM.ORG

# Data protection by design and by default (Elm to the rescue)

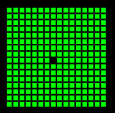
A few **cons**:

- **toString** (Elm magic) breaks data encapsulation
  - From 0.19 it will become part of **Debug** and will be replaced in **Core** with **String.fromInt** and so.
- **Debug.log** allows to print a lot of stuff without being able to limit it with the type system.
  - It shouldn't be allowed to use **Debug** from an application compiled without the **--debug** flag
- It gets **compiled** to **JavaScript** though ... hope with **Web Assembly**?
- Next version 0.19, so it's actually (still) not PROD ready ... until when?
  - For example, Idris is already 1.00 since 2017-04-01 (what a date to annouce it though)

# Summary

- Purity it's not just ***academic mumbo jumbo***, so use it FFS !!!
- ***EU GDPR*** arrives ***2018-05-25***, keep being smart and don't become stupid, tackle the problem instead of looking away
- ***Solve the problem*** by using a ***technical/scientific approach*** and not only by increasing your staff
- ***Article 5. Principles relating to processing of personal data*** and ***Article 25. Data protection by design and by default***
- Comply with EU GDPR (Pure vs Imperative, blog posts):
  - In Haskell it's almost trivial, very similar to Elm
  - F# it's not straight forward unless you rely on .NET Sandboxes or #Puritas ...





# Q & A

Any Questions?