

PRIVACY BY DESIGN

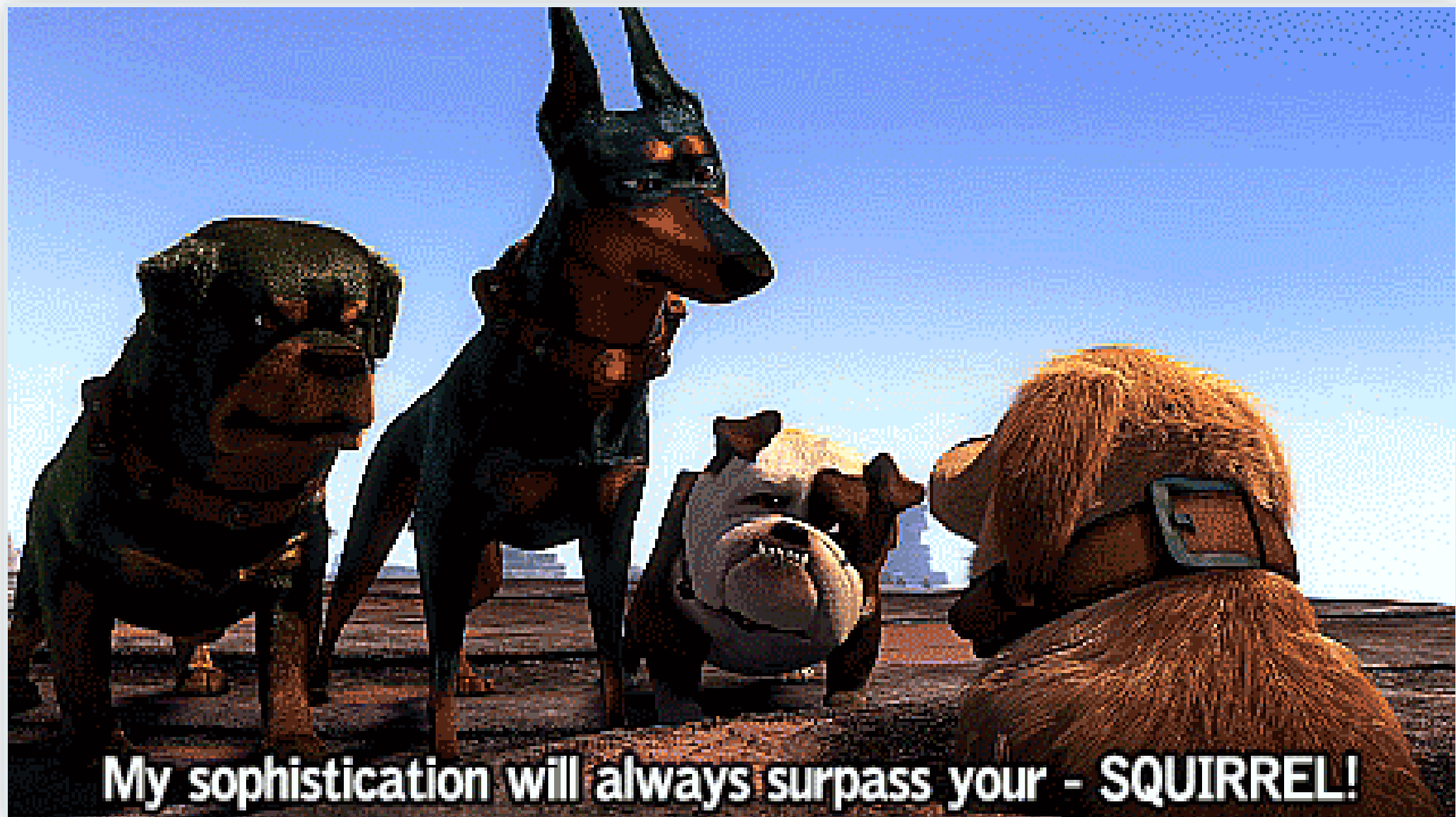
De l'autonomie de l'individu sur ses données personnelles



**COMMENT ON SE PERÇOIT
EN UTILISANT NOS APPS**

COMMENT LEURS ÉDITEURS NOUS PERÇOIVENT





Le biais de gratuité





LA PRIVACY, ON EN EST OÙ ?

LA PRIVACY, ON EN EST OÙ ?

La Data, bulle économique du XXI^e siècle

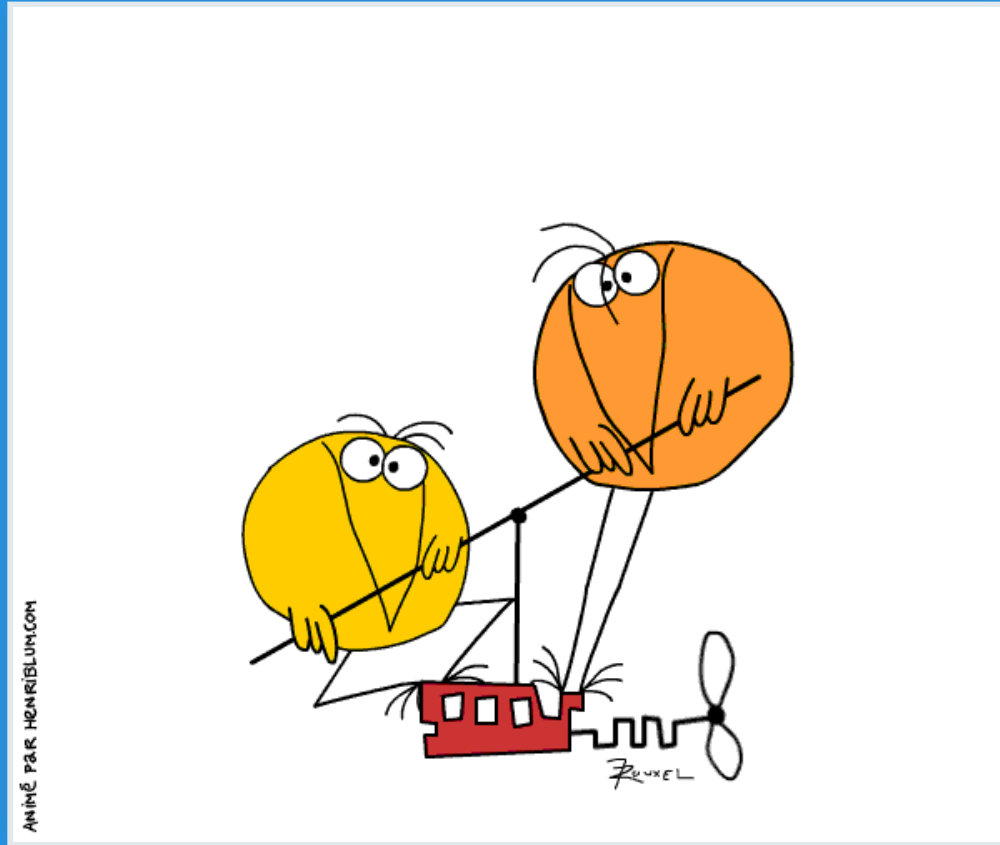
Des grosses boîtes qui pompent
les données

Des petites boîtes qui pompent les données

et souvent, elles n'en ont même pas conscience

Des startups qui pompent les données

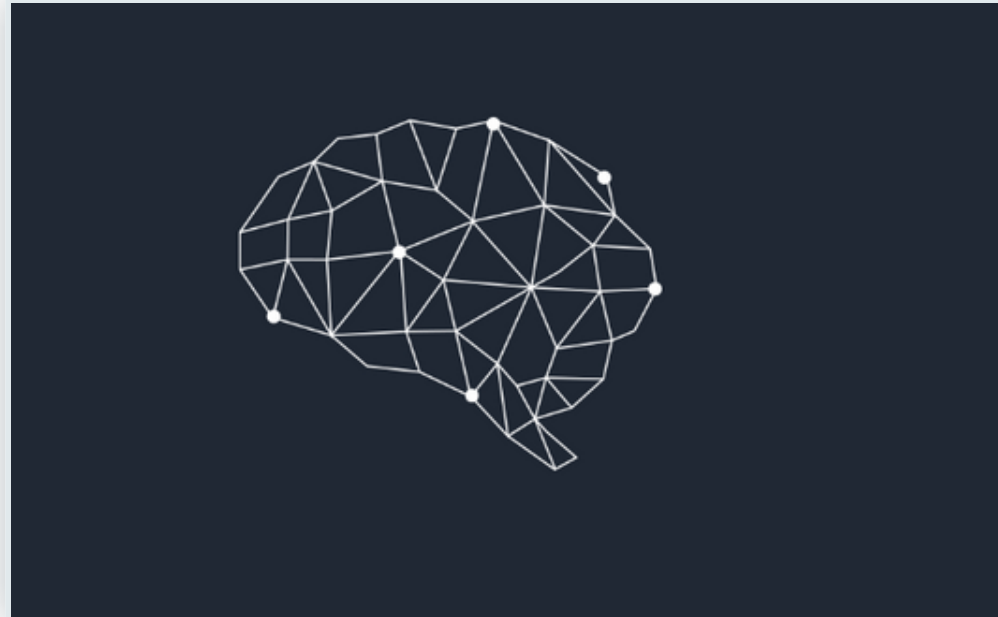
parce qu'elles veulent faire comme les grands dans la cour de récré



Une histoire de données *pas vraiment* volées



Une histoire de données *pas vraiment* volées



Logo de Cambridge Analytica en 2016, pendant la campagne Trump

Vie privée, du point de vue de l'utilisateur·trice

- donner trop de pouvoir rend les choses trop complexes
- illusionner sur une protection trop parfaite
- ce n'est pas un enjeu du public

« I call this device a Personal Information Telecommunication Agent, or Pita for short. The acronym also stand for Pain In The Ass, which it is equally likely to be, because having all that connectivity is going to destroy what's left of everyone's privacy.

↳ David Gerrold, in Sm@rt Reseller, "future of computing" prediction, 1999



CONCEPTION *PRIVACY BY DESIGN*

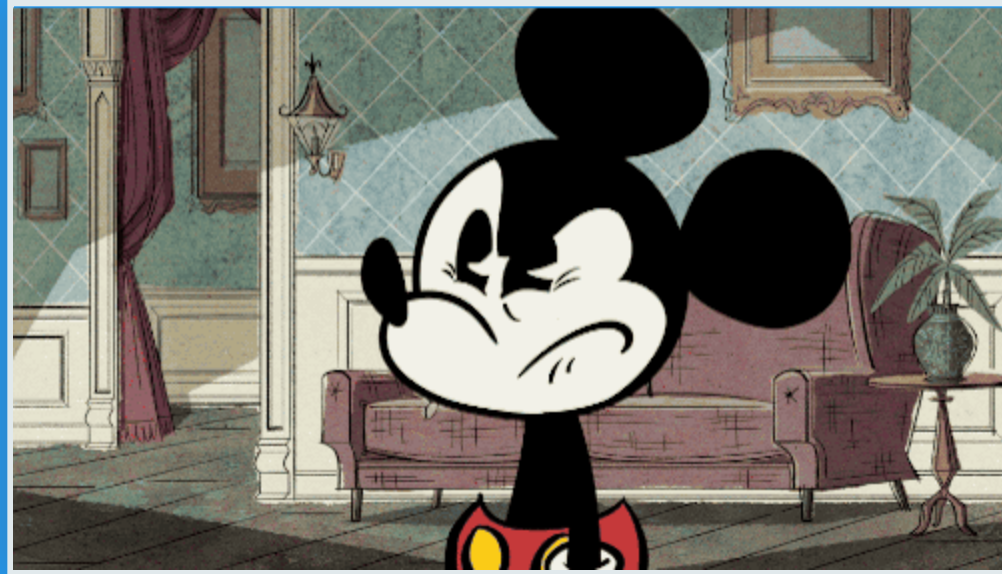
Préquelle : Accountability pattern

- début des années 80
- ensemble de procédures
- démontre la conformité aux règles de gestion
- crée les responsables de la sécurité des SI
- rapporte les preuves

Préquelle : Accountability pattern



1995 : Privacy by Design



7 laws of identity

1. Proactive not reactive; Preventative not remedial
2. Privacy as the default setting
3. Privacy embedded into design
4. Full functionality – positive-sum, not zero-sum
5. End-to-end security – full lifecycle protection
6. Visibility and transparency – keep it open
7. Respect for user privacy – keep it user-centric

PETs, vos nouveaux ~~animaux~~ outils de compagnie

- Chiffrement
- Gestion des métadonnées et des permissions
- Vérification légale intégrées au code
- Gouvernance des données
- Gestion des identités

 <https://iapp.org/news/a/2008-05-introduction-to-privacy-enhancing-technologies/>

En pratique :

Lors de la conception

- Concevez des check-list impliquant toutes les enjeux de données
- Assurez-vous que tous les intervenants sont sensibilisés
- Ne demandez pas plus de permissions que nécessaire
- Auditez, testez, pen testez !

Lors de la conception

- Concevez des check-list impliquant toutes les enjeux de données
- Assurez-vous que tous les intervenants sont sensibilisés
- Ne demandez pas plus de permissions que nécessaire
- Auditez, testez, pen testez !

côté technique...

- Chaque feature valide la check-list, en tests automatisés

Lors de la conception

- Concevez des check-list impliquant toutes les enjeux de données
- Assurez-vous que tous les intervenants sont sensibilisés
- Ne demandez pas plus de permissions que nécessaire
- Auditez, testez, pen testez !

côté technique...

- Chaque feature valide la check-list, en tests automatisés
- Les jeux de tests ne viennent pas de la prod !

Lors de la conception

- Concevez des check-list impliquant toutes les enjeux de données
- Assurez-vous que tous les intervenants sont sensibilisés
- Ne demandez pas plus de permissions que nécessaire
- Auditez, testez, pen testez !

côté technique...

- Chaque feature valide la check-list, en tests automatisés
- Les jeux de tests ne viennent pas de la prod !
- Oubliez les frameworks de permissions tous prêts

Lors de la conception

- Concevez des check-list impliquant toutes les enjeux de données
- Assurez-vous que tous les intervenants sont sensibilisés
- Ne demandez pas plus de permissions que nécessaire
- Auditez, testez, pen testez !

côté technique...

- Chaque feature valide la check-list, en tests automatisés
- Les jeux de tests ne viennent pas de la prod !
- Oubliez les frameworks de permissions tous prêts
- Tests fonctionnels sur des environnements multiples

Example : Checklist (GDPRChecklist.io)



Your data

Accountability &
management

New rights

Consent

Follow-up

Special cases

COMPLIANCE TOOLS

GDPR Form

The GDPR Compliance Checklist

Achieving GDPR Compliance shouldn't feel like a struggle. This is a basic checklist you can use to harden your GDPR compliancy.

New

Manage your data subjects requests with GDPR Form. Start your free trial today and receive a 20% discount.

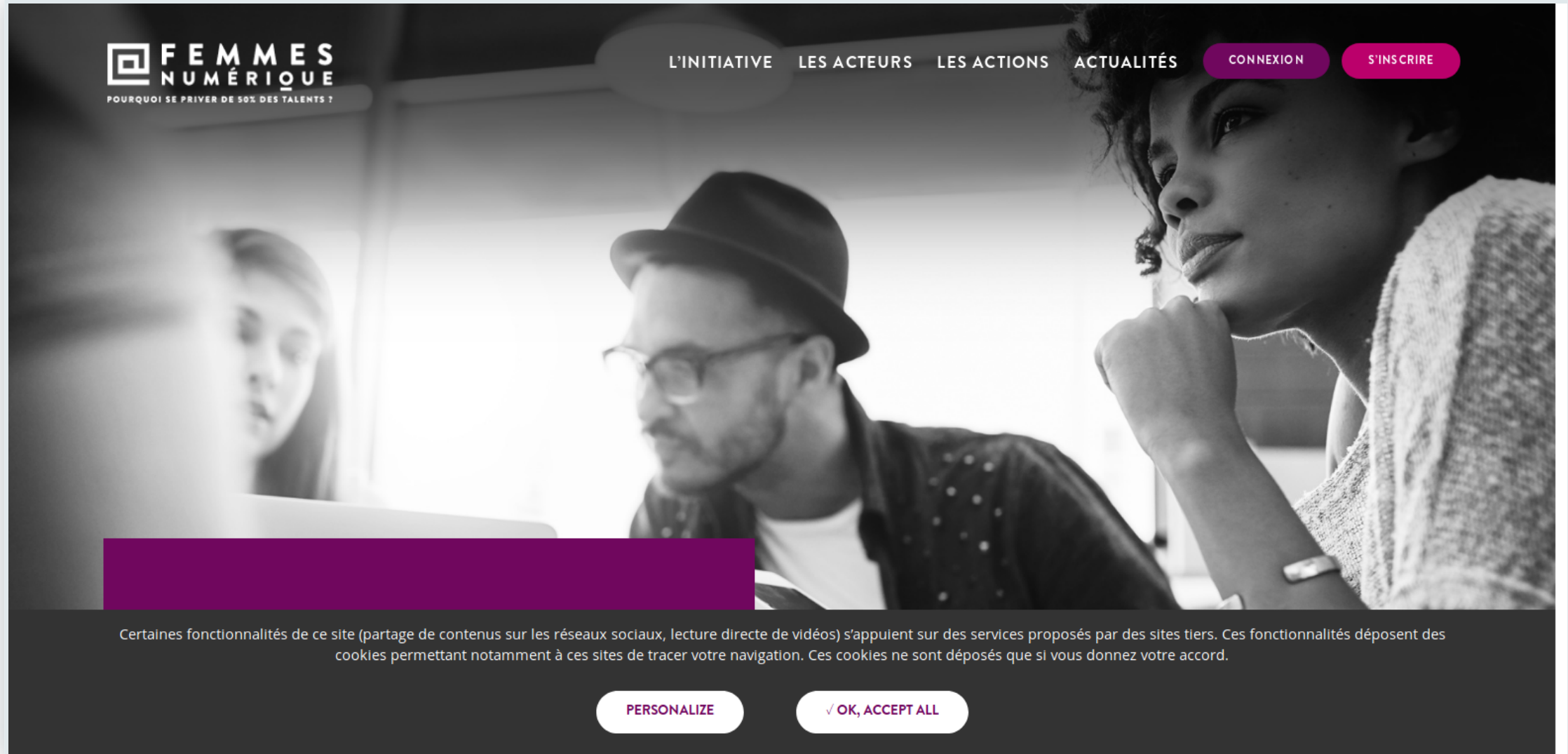
(From the makers of GDPR Tracker & Checklist)

if your organisation is determining the purpose of the storage or processing of personal information, it is considered a **controller**. If your organisation stores or processes personal data on behalf of another organisation, it is considered a **processor**. It is possible for your organisation to have both roles. Use the filter below to view only the relevant checklist items for your organisation.

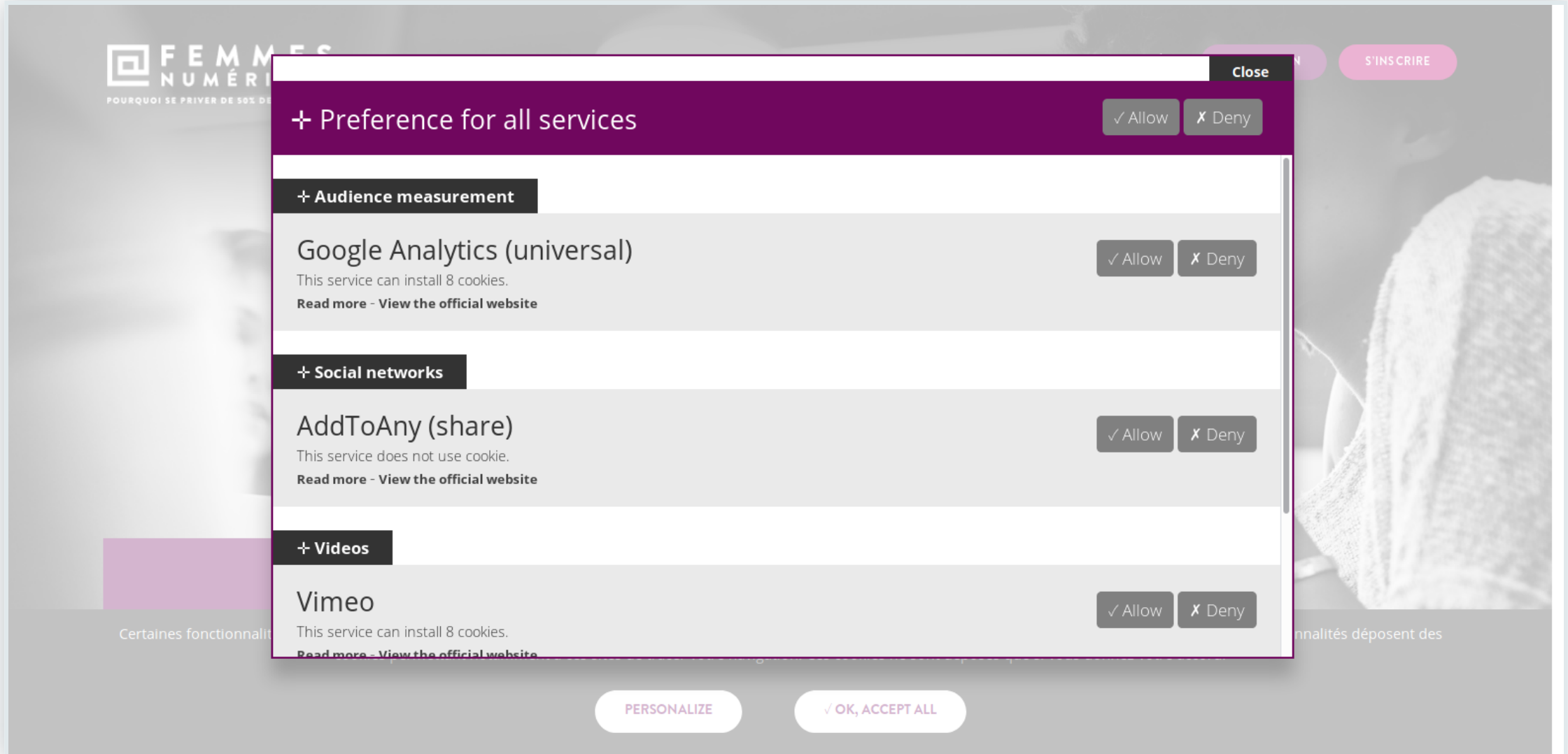
This list is far from a legal exhaustive document, it merely tries to help you overcome the struggle.

Feel free to [contribute directly](#) on GitHub!

Exemple : Permissions



Exemple : Permissions



Lors de l'exécution

- Minimisez la collecte de données
- Minimisez les données échangées avec les services tiers
- Pseudonymisez la donnée
- Vérifiez les formulaires (contact, login, assistance...)
- Supprimez régulièrement la donnée collectée

Lors de l'exécution

- Minimisez la collecte de données
- Minimisez les données échangées avec les services tiers
- Pseudonymisez la donnée
- Vérifiez les formulaires (contact, login, assistance...)
- Supprimez régulièrement la donnée collectée

côté technique...

- Utilisez des services de gestion d'identités (OpenID...)

Lors de l'exécution

- Minimisez la collecte de données
- Minimisez les données échangées avec les services tiers
- Pseudonymisez la donnée
- Vérifiez les formulaires (contact, login, assistance...)
- Supprimez régulièrement la donnée collectée

côté technique...

- Utilisez des services de gestion d'identités (OpenID...)
- Hashez / chiffrez / tokenisez les entrées
- Permutez, substituez, segmentez les données sensibles (Matomo...)

Lors de l'exécution

- Minimisez la collecte de données
- Minimisez les données échangées avec les services tiers
- Pseudonymisez la donnée
- Vérifiez les formulaires (contact, login, assistance...)
- Supprimez régulièrement la donnée collectée

côté technique...

- Utilisez des services de gestion d'identités (OpenID...)
- Hashez / chiffrez / tokenisez les entrées
- Permutez, substituez, segmentez les données sensibles (Matomo...)
- Faites passer des cron !

Example : OpenID

```
1 export function middlewareHandler(next, action, userManager) {
2   // prevent an infinite loop of dispatches of these action types (issue #30 & #63)
3   if (action.type === USER_EXPIRED || action.type === LOADING_USER || action.type === USER_FOUND) {
4     return next(action);
5   }
6
7   nextMiddleware = next;
8
9   if (!storedUser || storedUser.expired) {
10    next(loadingUser());
11    userManager.getUser()
12      .then(getUserCallback)
13      .catch(errorCallback);
14  }
15  return next(action);
16 }
```

→ [github://IdentityModel/oidc-client-js](https://github.com/IdentityModel/oidc-client-js)

Example : RSA Signature

```
1 // Sign with the private key...
2 var sign = new JSEncrypt();
3 sign.setPrivateKey($('#privkey').val());
4 var signature = sign.sign($('#input').val(), CryptoJS.SHA256, "sha256");
5
6 // Verify with the public key...
7 var verify = new JSEncrypt();
8 verify.setPublicKey($('#pubkey').val());
9 var verified = verify.verify($('#input').val(), signature, CryptoJS.SHA256);
10
11 if (verified) {
12     alert('It works!!!');
13 } else {
14     alert('Something went wrong....');
15 }
```

Exemple : RSA Encryption

```
1 <template id="privkey">-----BEGIN RSA PRIVATE KEY-----
2 MIICXQIBAAKBgQDl0Ju6TyygqxfWT7eLtGDwajtNF0b9I5XRb6khyfD1Yt3YiCgQ
3 WMNW649887VGJiGr/L5i2osbl8C9+WJTeucF+S76xFxdU6jE0NQ+Z+zEdhUTooNR
4 [...]
5 aTgjFnqE/lQ22Rk0eGaY080cc643BXVGafNfd9fcvwBMnk0iGX0XRs0ozVt5Azil
6 psLBYuApa66NcVHJpCECQQDTjI2AQhFc1yRnCU/YgDnSpJVm1nASoRUnU8Jfm30z
7 uku7JUXcVpt08DFSceCEX9unCuMcT72rAQLlpdZir876
8 -----END RSA PRIVATE KEY-----</template>
9 <template id="pubkey">-----BEGIN PUBLIC KEY-----
10 MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDl0Ju6TyygqxfWT7eLtGDwajtN
11 F0b9I5XRb6khyfD1Yt3YiCgQWMNW649887VGJiGr/L5i2osbl8C9+WJTeucF+S76
12 xFxdU6jE0NQ+Z+zEdhUTooNRaY5nZiu5PgDB0ED/ZKBUSLKL7eibMxZtMlUDHjm4
13 gwQco1KRMDSmXSMkDwIDAQAB
14 -----END PUBLIC KEY-----</template>
15 <textarea id="input" name="input" type="text" rows=4 cols=70>This is a test!</textarea>
```


Example : RSA Encryption

```
1 // Encrypt with the public key...
2 var encrypt = new JSEncrypt();
3 encrypt.setPublicKey($('#pubkey').val());
4 var encrypted = encrypt.encrypt($('#input').val());
5
6 // Decrypt with the private key...
7 var decrypt = new JSEncrypt();
8 decrypt.setPrivateKey($('#privkey').val());
9 var unencrypted = decrypt.decrypt(encrypted);
10
11 if (unencrypted == $('#input').val()) {
12     alert('It works!!!');
13 } else {
14     alert('Something went wrong....');
15 }
```

Browser : libs fournissant la couche Crypto

- jsencrypt
- js-nacl
- jwcrypto

→  [gist://jo:8619441](https://gist.github.com/jo/8619441)

Expérience utilisateur

- Fournissez des réglages simples et des notices claires à valider
- N'exigez pas de passer par des services externes
- Pas de partage sur les réseaux par défaut
- Séparez les consentements (*shared data* vs *analytics*)

Expérience utilisateur

- Fournissez des réglages simples et des notices claires à valider
- N'exigez pas de passer par des services externes
- Pas de partage sur les réseaux par défaut
- Séparez les consentements (*shared data* vs *analytics*)

côté technique...

- Utilisez les frameworks de notification pour ne pas polluer (toastr, Notify.js...)

Expérience utilisateur

- Fournissez des réglages simples et des notices claires à valider
- N'exigez pas de passer par des services externes
- Pas de partage sur les réseaux par défaut
- Séparez les consentements (*shared data* vs *analytics*)

côté technique...

- Utilisez les frameworks de notification pour ne pas polluer (toastr, Notify.js...)
- Utilisez des outils d'identités décentralisées (OpenID...), pas les réseaux sociaux

Expérience utilisateur

- Fournissez des réglages simples et des notices claires à valider
- N'exigez pas de passer par des services externes
- Pas de partage sur les réseaux par défaut
- Séparez les consentements (*shared data* vs *analytics*)

côté technique...

- Utilisez les frameworks de notification pour ne pas polluer (toastr, Notify.js...)
- Utilisez des outils d'identités décentralisées (OpenID...), pas les réseaux sociaux
- Plus de jsSocials ~~par défaut~~ par pitié...

Expérience utilisateur

- Fournissez des réglages simples et des notices claires à valider
- N'exigez pas de passer par des services externes
- Pas de partage sur les réseaux par défaut
- Séparez les consentements (*shared data* vs *analytics*)

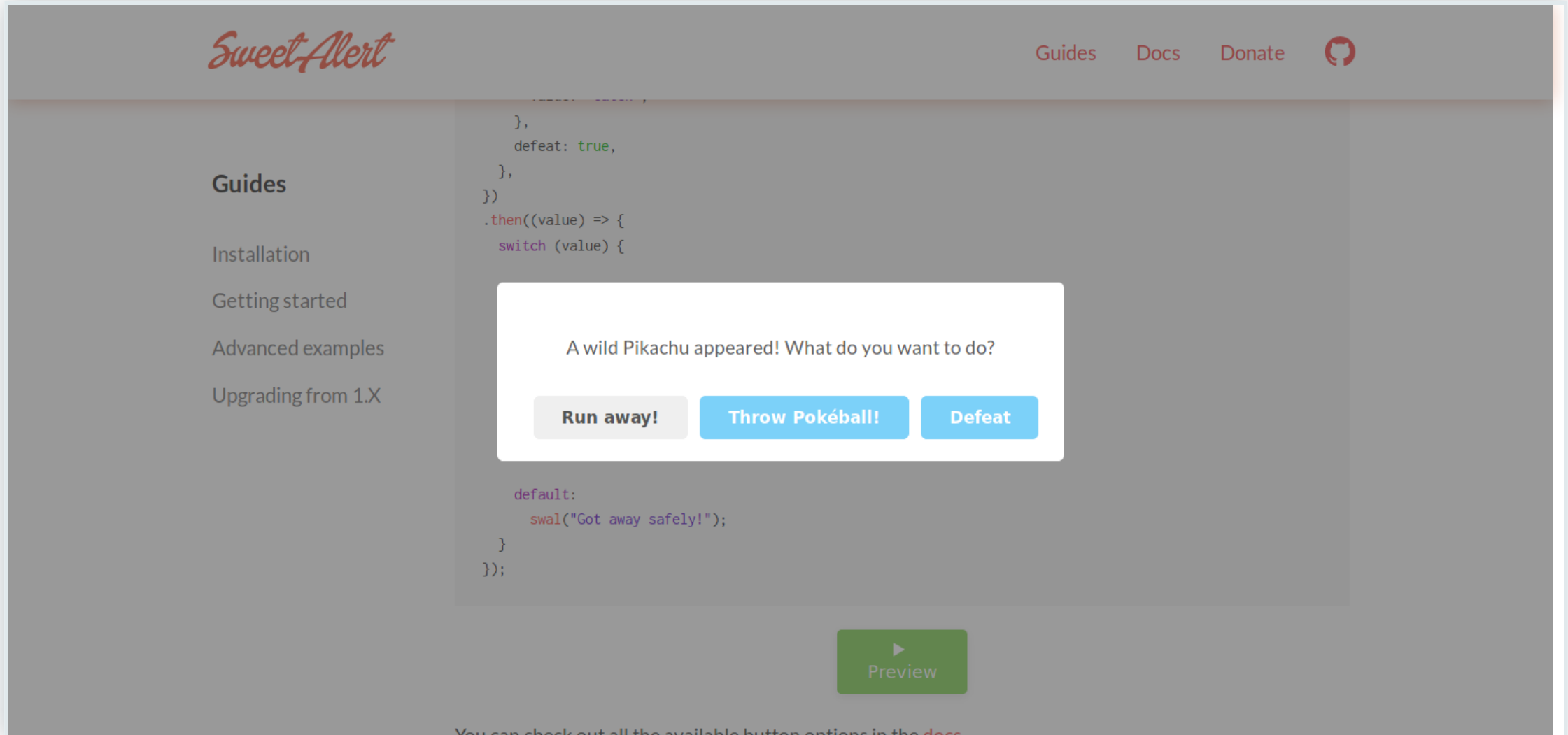
côté technique...

- Utilisez les frameworks de notification pour ne pas polluer (toastr, Notify.js...)
- Utilisez des outils d'identités décentralisées (OpenID...), pas les réseaux sociaux
- Plus de jsSocials ~~par défaut~~ par pitié...
- Utilisez des outils de trace d'usages respectueux

Example : SweetAlert

```
1 swal("A wild Pikachu appeared! What do you want to do?", {
2   buttons: {
3     catch: {
4       text: "Throw Pokéball!",
5       value: "catch",
6     },
7     defeat: true,
8   },
9 })
10 .then((value) => {
11   switch (value) {
12     case "defeat":
13       swal("Pikachu fainted! You gained 500 XP!"); break
14     case "catch":
15       swal("Gotcha!", "Pikachu was caught!", "success"); break
16   }
17 })
```


Exemple : SweetAlert



Exemple : SweetAlert

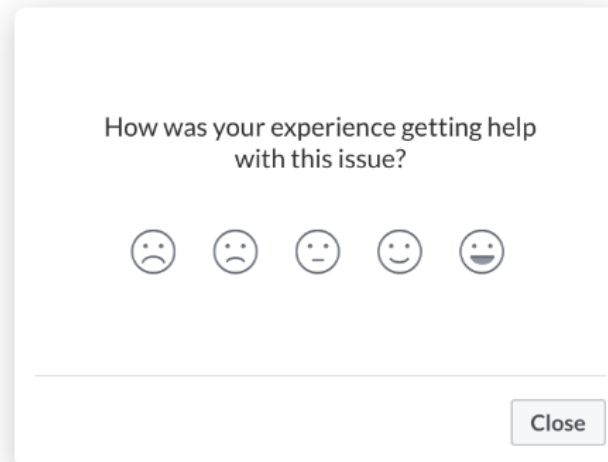
[Guides](#)[Docs](#)[Donate](#)

Guides

[Installation](#)[Getting started](#)[Advanced examples](#)[Upgrading from 1.X](#)

option to render it as an unstyled element.

The only code that's specific to SweetAlert is the `swal.setActionValue()` and the `swal()` call at the end. The rest is just basic React and JavaScript.



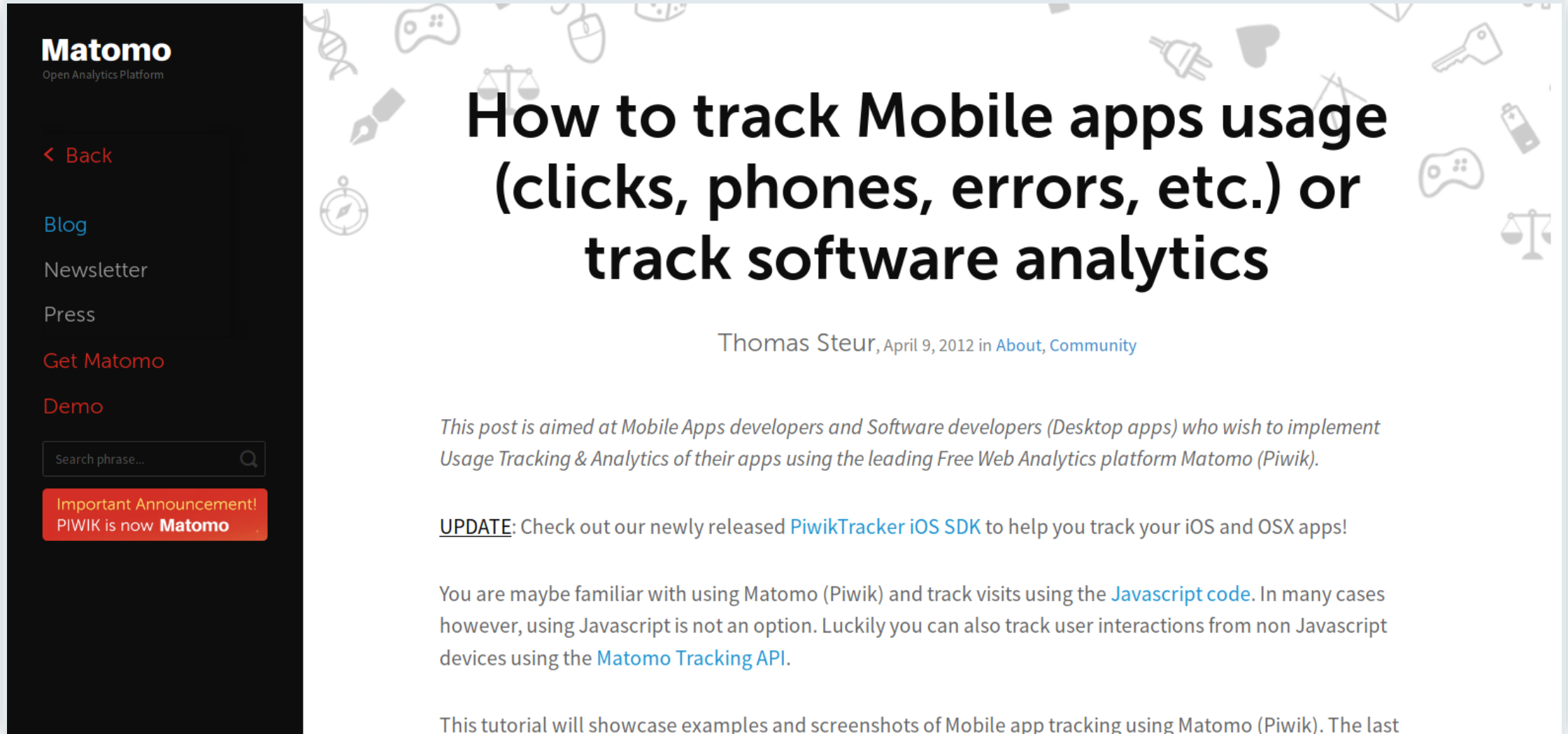
Using this technique, we can create modals with more interactive UIs, such as this one from Facebook.

Exemple : SweetAlert

- configurable
- chainable
- promises
- compatible frameworks composants

→  [github://sweetalert2:sweetalert2](https://github.com/sweetalert2/sweetalert2)

Exemple : Traces d'usage avec Matomo



The screenshot shows a Matomo blog post. On the left is a dark sidebar with the Matomo logo and navigation links. The main content area has a decorative header with various icons. The article title is 'How to track Mobile apps usage (clicks, phones, errors, etc.) or track software analytics' by Thomas Steur, dated April 9, 2012. The article text discusses tracking mobile app usage using Matomo (Piwik) and mentions a new PiwikTracker iOS SDK.

Matomo
Open Analytics Platform

[< Back](#)

[Blog](#)

[Newsletter](#)

[Press](#)

[Get Matomo](#)

[Demo](#)

Search phrase... 

Important Announcement!
PIWIK is now **Matomo**

How to track Mobile apps usage (clicks, phones, errors, etc.) or track software analytics

Thomas Steur, April 9, 2012 in [About](#), [Community](#)

This post is aimed at Mobile Apps developers and Software developers (Desktop apps) who wish to implement Usage Tracking & Analytics of their apps using the leading Free Web Analytics platform Matomo (Piwik).

UPDATE: Check out our newly released [PiwikTracker iOS SDK](#) to help you track your iOS and OSX apps!

You are maybe familiar with using Matomo (Piwik) and track visits using the [Javascript code](#). In many cases however, using Javascript is not an option. Luckily you can also track user interactions from non Javascript devices using the [Matomo Tracking API](#).

This tutorial will showcase examples and screenshots of Mobile app tracking using Matomo (Piwik). The last

Fin du cycle de vie

- Rappelez régulièrement les utilisateurs·trices à leur confidentialité
- Facilitez l'export de données
- Supprimez les données des comptes supprimés
- Supprimez les données à la fermeture du service

Fin du cycle de vie

- Rappelez régulièrement les utilisateurs·trices à leur confidentialité
- Facilitez l'export de données
- Supprimez les données des comptes supprimés
- Supprimez les données à la fermeture du service

côté technique...

- Mettez en place des APIs documentées (Swagger, Apiary...) et utilisables

Fin du cycle de vie

- Rappelez régulièrement les utilisateurs·trices à leur confidentialité
- Facilitez l'export de données
- Supprimez les données des comptes supprimés
- Supprimez les données à la fermeture du service

côté technique...

- Mettez en place des APIs documentées (Swagger, Apiary...) et utilisables
- Utilisez des formats de données ouverts (XML, JSON...)

Fin du cycle de vie

- Rappelez régulièrement les utilisateurs·trices à leur confidentialité
- Facilitez l'export de données
- Supprimez les données des comptes supprimés
- Supprimez les données à la fermeture du service

côté technique...

- Mettez en place des APIs documentées (Swagger, Apiary...) et utilisables
- Utilisez des formats de données ouverts (XML, JSON...)
- `rm -rf /`

Exemple : Swagger / OpenAPI

 **Swagger Editor** File Edit Generate Server Generate Client

```
1 swagger: "2.0"
2 info:
3   description: "This is a sample server Petstore server. You can
   find out more about Swagger at [http://swagger.io](http
   ://swagger.io) or on [irc.freenode.net, #swagger](http://swagger
   .io/irc/). For this sample, you can use the api key 'special
   -key' to test the authorization filters."
4   version: "1.0.0"
5   title: "Swagger Petstore"
6   termsOfService: "http://swagger.io/terms/"
7   contact:
8     email: "apiteam@swagger.io"
9   license:
10    name: "Apache 2.0"
11    url: "http://www.apache.org/licenses/LICENSE-2.0.html"
12 host: "petstore.swagger.io"
13 basePath: "/v2"
14 tags:
15 - name: "pet"
16   description: "Everything about your Pets"
17 externalDocs:
18   description: "Find out more"
19   url: "http://swagger.io"
20 - name: "store"
21   description: "Access to Petstore orders"
22 - name: "user"
23   description: "Operations about user"
24 externalDocs:
25   description: "Find out more about our store"
26   url: "http://swagger.io"
27 schemes:
```

Swagger Petstore ^{1.0.0}

[Base URL: petstore.swagger.io/v2]

This is a sample server Petstore server. You can find out more about Swagger at <http://swagger.io> or on irc.freenode.net, #swagger. For this sample, you can use the api key **special-key** to test the authorization filters.

[Terms of service](#)

[Contact the developer](#)

[Apache 2.0](#)

[Find out more about Swagger](#)

Schemes

HTTPS

Authorize 

pet Everything about your Pets

Find out more: <http://swagger.io>

POST

/pet Add a new pet to the store



Use REST

REST API Tutorial

Home

Tutorials ▾

HTTP Status Codes

Resources

Home

What Is REST?

REST Quick Tips

HTTP Methods

Resource Naming

Idempotence

HTTP Status Codes

Resources

Learn REST: A RESTful Tutorial

Hey, Fellow REST API Designer!

Building RESTful web services, like other programming skills is **part art, part science**. As the Internet industry progresses, creating a REST API becomes more concrete with emerging best practices. As RESTful web services don't follow a prescribed standard except for HTTP, it's important to build your RESTful API in accordance with industry best practices to ease development and increase client adoption.

Presently, there aren't a lot of REST API guides to help the lonely developer. [RestApiTutorial.com](#) is dedicated to tracking REST API best practices and making resources available to enable quick reference and self education for the development crafts-person. We'll discuss both the art and science of creating REST Web services.

—Todd Fredrich, *The REST API Tutor*

Jump in with [What Is REST?](#), an overview of concepts and constraints of the RESTful architecture.

[Next »](#)

Fork me on GitHub

OBOE for large JSON contents

```
1 oboe('/myapp/things.json')
2   .node('foods.*', function( foodThing ){
3       // This callback will be called everytime a new object is
4       // found in the foods array.
5
6       console.log( 'Go eat some', foodThing.name);
7   })
8   .node('badThings.*', function( badThing ){
9       console.log( 'Stay away from', badThing.name);
10  })
11  .done(function(things){
12      console.log(
13          'there are', things.foods.length, 'things to eat',
14          'and', things.nonFoods.length, 'to avoid')
15  })
```

Use GraphQL

```
1 var { graphql, buildSchema } = require('graphql')
2
3 var schema = buildSchema(`
4   type Query {
5     hello: String
6   }
7 `)
8
9 var root = { hello: () => 'Hello world!' }
10
11 graphql(schema, '{ hello }', root).then((response) => {
12   console.log(response)
13 })
```

GraphQL Interface

GraphiQL 

```
1 query TodoAppQuery($n: Int!) {
2   globalTodoList {
3     items(first:$n) {
4       edges {
5         node {
6           text
7           complete
8         }
9       }
10    }
11  }
12 }
```

QUERY VARIABLES

```
1 {
2   "n": 2
3 }
```

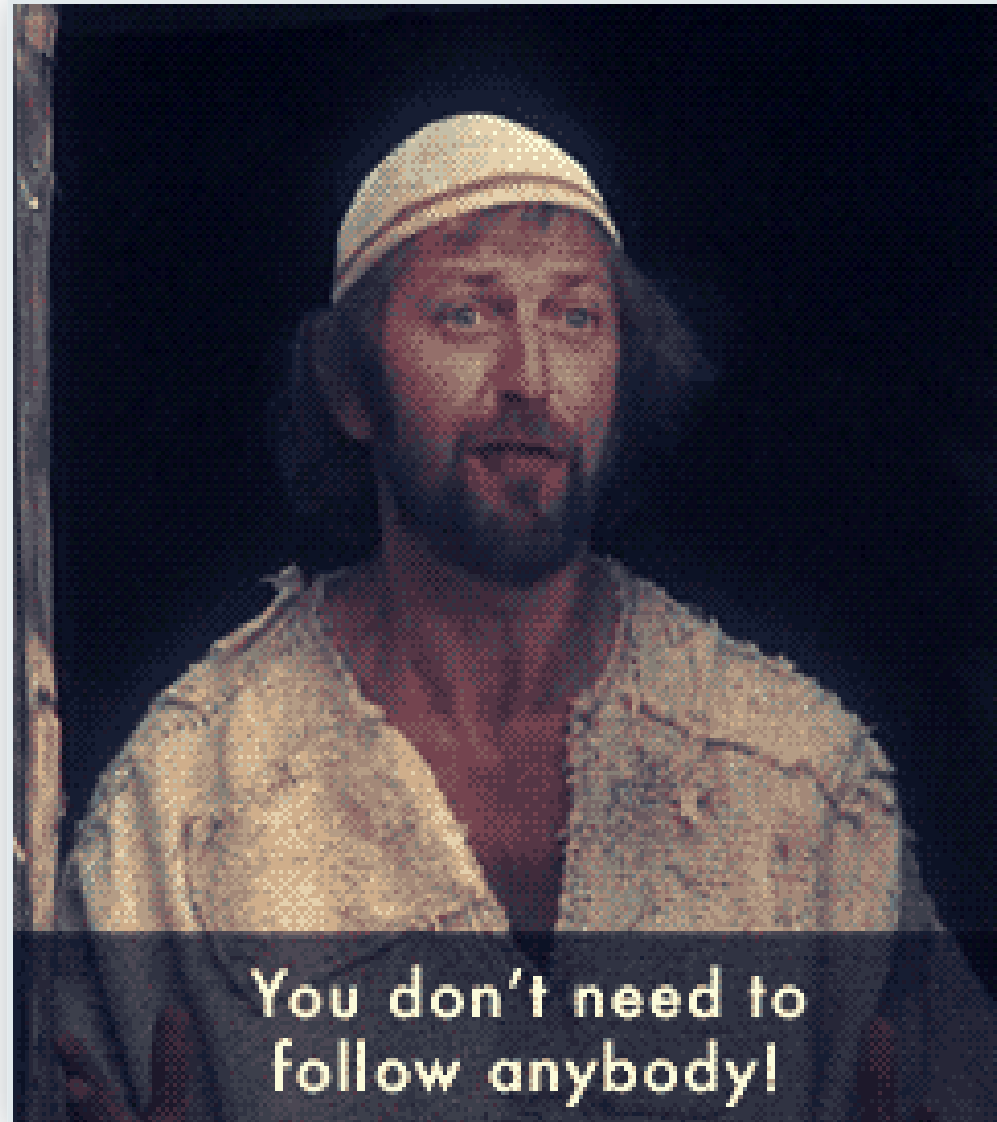
```
{
  "data": {
    "globalTodoList": {
      "items": {
        "edges": [
          {
            "node": {
              "text": "Release GraphiQL",
              "complete": true
            }
          },
          {
            "node": {
              "text": "Attend @Scale 2015",
              "complete": false
            }
          }
        ]
      }
    }
  }
}
```

Pour résumer :

OWASP : top 10 privacy risks

1. Web Application Vulnerabilities
2. Operator-sided Data Leakage
3. Insufficient Data Breach Response
4. Insufficient Deletion of personal data
5. Non-transparent Policies, Terms and Conditions
6. Collection of data not required for the primary purpose
7. Sharing of data with third party
8. Outdated personal data
9. Missing or Insufficient Session Expiration
10. Insecure Data Transfer

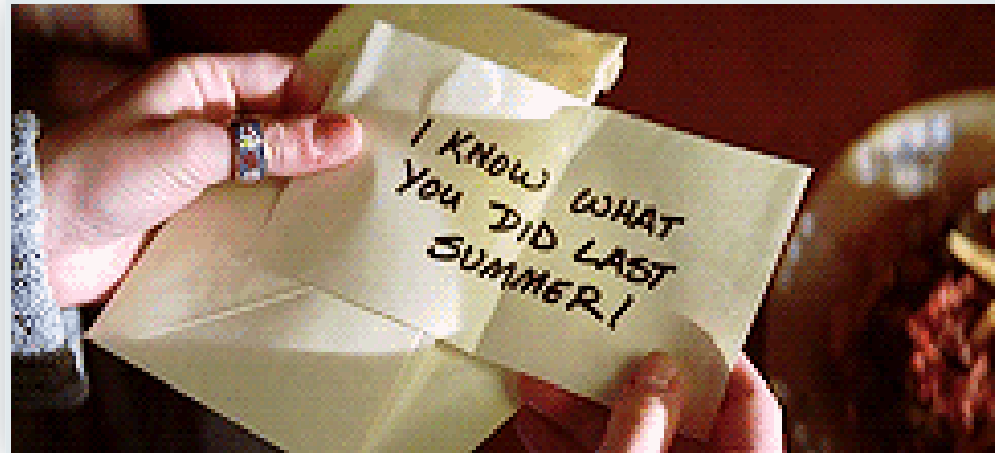
Tracer les parcours de la donnée, pas des utilisateurs



Gérer les identités



La pseudonimisation, Graal de l'analyse de données



🔗 [Why Anonymous Data Sometimes Isn't, a Netflix story](#)

DIFFERENTIAL PRIVACY



-  Differential privacy @Wikipedia
-  Harvard University Privacy Tools Project
-  Cornell university Library
-  Uber SQL Differential Privacy

A black and white photograph showing a wooden mallet positioned diagonally from the top right, about to strike a small, light-colored egg. The egg is resting on a dark, rectangular wooden block. The background is dark and out of focus. The scene is lit from the side, creating strong highlights and shadows that emphasize the textures of the wood and the egg.

PENSER LA VIE PRIVÉE AUTREMENT

ESPÉRER EST ILLUSOIRE

« La *privacy by design* est complètement aux antipodes de la souveraineté numérique des individus : on fait sans les individus, on protège la vie privée sans définir ce que c'est.

Fabrice Rochelandet. Souveraineté numérique et modèle d'affaires. In: Numérique, reprendre le contrôle. Framasoft: 2016, p.65

Privacy by default

- assure qu'un minimum de données est en jeu
- simplifie le processus pour les utilisateurs·trices
- évite les difficultés dans les réglages de confidentialité
- force *idéalement* le niveau de protection maximal *par défaut*

Agir

- penser la donnée comme un vivant périssable
- chaque acteur se doit d'alerter
- mesurer chaque brique unitairement
- assurer la portabilité

CODE IS LAW

LE DÉVELOPPEUR EST POLITIQUE

Publicness

« Publicness is value. This is an argument I'll make that what's public is owned by the public — whether that's governments' actions or images taken in public space — and whenever that is diminished, it robs from us, the public.

Jeff Jarvis. 2011

→  Jeff Jarvis - Privacy and Publicness and the power behind it - Youtube

Privacy by using



SENSIBILISER LES USAGES



LANCER DES ALERTES



AGIR CHACUN À SON NIVEAU

« Nul ne sera l'objet d'immixtions arbitraires dans sa vie privée, sa famille, son domicile ou sa correspondance, ni d'atteintes à son honneur et à sa réputation. Toute personne a droit à la protection de la loi contre de telles immixtions ou de telles atteintes.

« Nul ne sera l'objet d'immixtions arbitraires dans sa vie privée, sa famille, son domicile ou sa correspondance, ni d'atteintes à son honneur et à sa réputation. Toute personne a droit à la protection de la loi contre de telles immixtions ou de telles atteintes.

Déclaration universelle des droits de l'homme. Article 12, 1948



M4DZ

Paranoïd Web Dino & Tech Evangelist

🌐 m4dz.net | 📧 @m4d_z | PGP 📧 [0xD4627C417D969710](#)



🌐 www.alwaysdata.com



QUESTIONS ?

Outils

- Moteur de présentation : [Remark](#)



➤ <http://talks.m4dz.net/privacy-by-design/>

disponible sous licence [CC BY-SA 4.0](#)