La reproductibilité du calcul

Konrad HINSEN

Centre de Biophysique Moléculaire, Orléans, France
et
Synchrotron SOLEIL, Saint Aubin, France

18 janvier 2021

Pourquoi tout le monde parle de reproductibilité ?

Royal Society of London for Improving Natural Knowledge (1663)

Pourquoi tout le monde parle de reproductibilité ?

WIKIPÉDIA
L'encyclopédie libre

Rechercher dans Wikipédia

Non connecté Discussion Contributions Créer un compte Se connecter

Article Discussion | Lire Modifier Modifier le code Voir l'historique

Crise de la reproductibilité

La **crise de la reproductibilité** (*replication crisis* ou *replicability crisis* ou *reproducibility crisis* en anglais) est la crise méthodologique dans le domaine des sciences selon laquelle de nombreux résultats publiés dans des revues scientifiques sont difficiles, voire impossibles à reproduire au cours d'études subséquentes. Initiée au milieu des années 2000, la crise prend de l'ampleur au milieu des années 2010, nourrie par la publication de plusieurs articles sur le phénomène.

La crise n'est pas propre à un domaine unique bien qu'elle semble moins toucher les sciences fondamentales et appliquées que les sciences médicales. Les facteurs qui en sont responsables semblent nombreux. Des pistes d'amélioration de la reproductibilité au sein des publications scientifiques, dont notamment l'amélioration des critères de publication, sont explorées.

Sommaire [masquer]

1 Historique
2 Étendue du phénomène
  2.1 Médecine
  2.2 Psychologie
3 Causes
  3.1 Fraudes
4 Pistes de solutions

Pourquoi tout le monde parle de reproductibilité ?

Reproductibilité : preuve de **rigueur** qui inspire **confiance**

Un résultat non-reproductible suggère...

- ... une description incomplète
- ... une maîtrise insuffisante des techniques
- ... une erreur
- ... une fraude

Ou encore...

- ... une prise insuffisante sur les sujets d'étude (souris, étoiles, ...)

La trinité de la reproductibilité

Reproductibilité **expérimentale**

- Refaire une expérience d'après la description publiée
- Obtenir des résultats suffisamment proches

Reproductibilité **statistique**

- Refaire une étude avec un autre échantillon ou une autre technique
- Inférer des conclusions suffisamment proches

Reproductibilité **computationnelle**

- Refaire un calcul à l'identique
- Obtenir des résultats identiques

D’où vient la crise ?

Les axes du progrès en science :

- nouveaux phénomènes
- moins d’incertitudes
- plus de rigueur

Manque de rigueur, surtout en statistique et en calcul

Reproduire ou répliquer ?

Reproductibilité

- Niveau technique
- Est-ce bien fait ?
- Évaluation simple
- Réponse simple : oui/non
- **Vérification**

Réplicabilité

- Niveau scientifique
- Est-ce la bonne chose à faire ?
- Évaluation laborieuse
- Réponse complexe : si...
- **Validation**

En calcul scientifique :

- Même code
- Mêmes paramètres
- Mêmes données
- **Résultat identique ?**

- Nouveau code
- Mêmes paramètres
- Mêmes données (ou pas)
- **Résultat équivalent ?**

Reproduire ou répliquer ?

La méthode scientifique demande la réplicabilité.

Pour examiner la réplicabilité il faut d'abord assurer la reproductibilité.

L’histoire de l’eau surrefroidie

PHYSICS TODAY

HOME BROWSE INFO RESOURCES JOBS

DOI:10.1063/PT.6.1.20180822a

22 Aug 2018 in Research & Technology

The war over supercooled water

How a hidden coding error fueled a seven-year dispute between two of condensed matter’s top theorists.

Ashley G. Smart

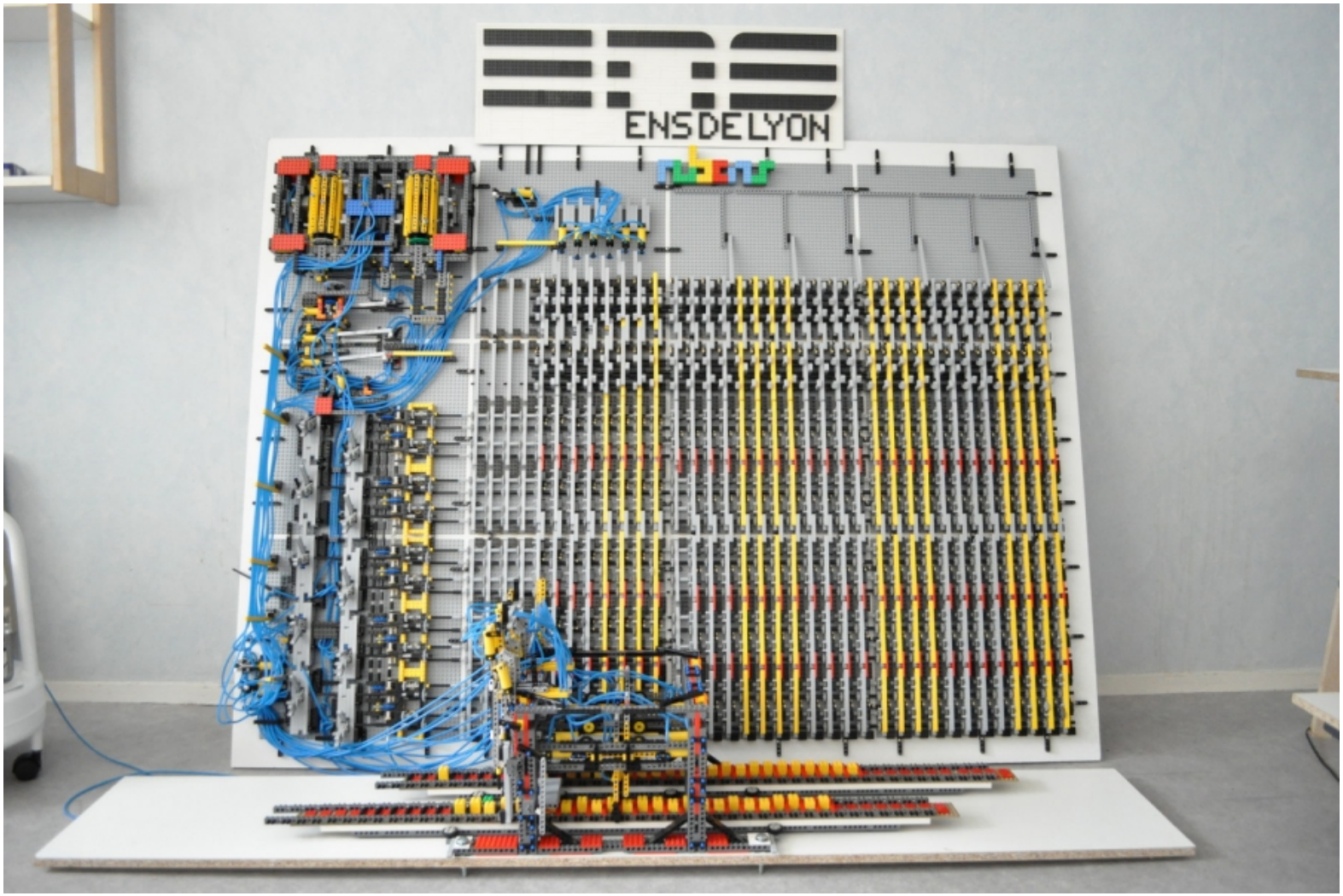
A.G. Smart, Physics Today, 2018

Les enjeux de la reproductibilité en calcul scientifique

- Être sûr de ce qui a été calculé.
- Pouvoir vérifier le calcul.
- Pouvoir adapter le calcul.

Particularité du calcul : déterminisme (sauf calcul parallèle)

C’est quoi un calcul ?

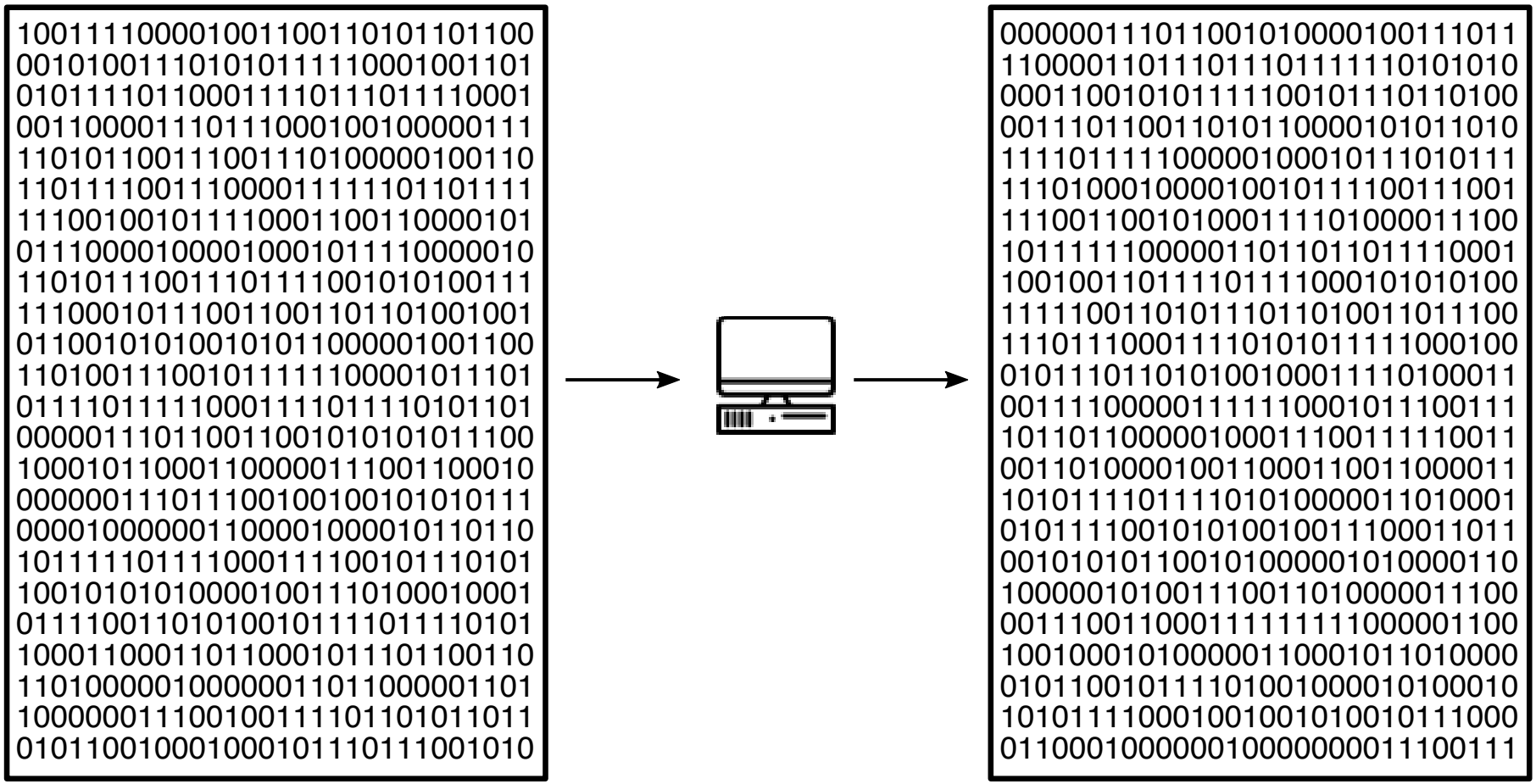


Vue globale de la machine de Turing en Légo du projet Rubens.

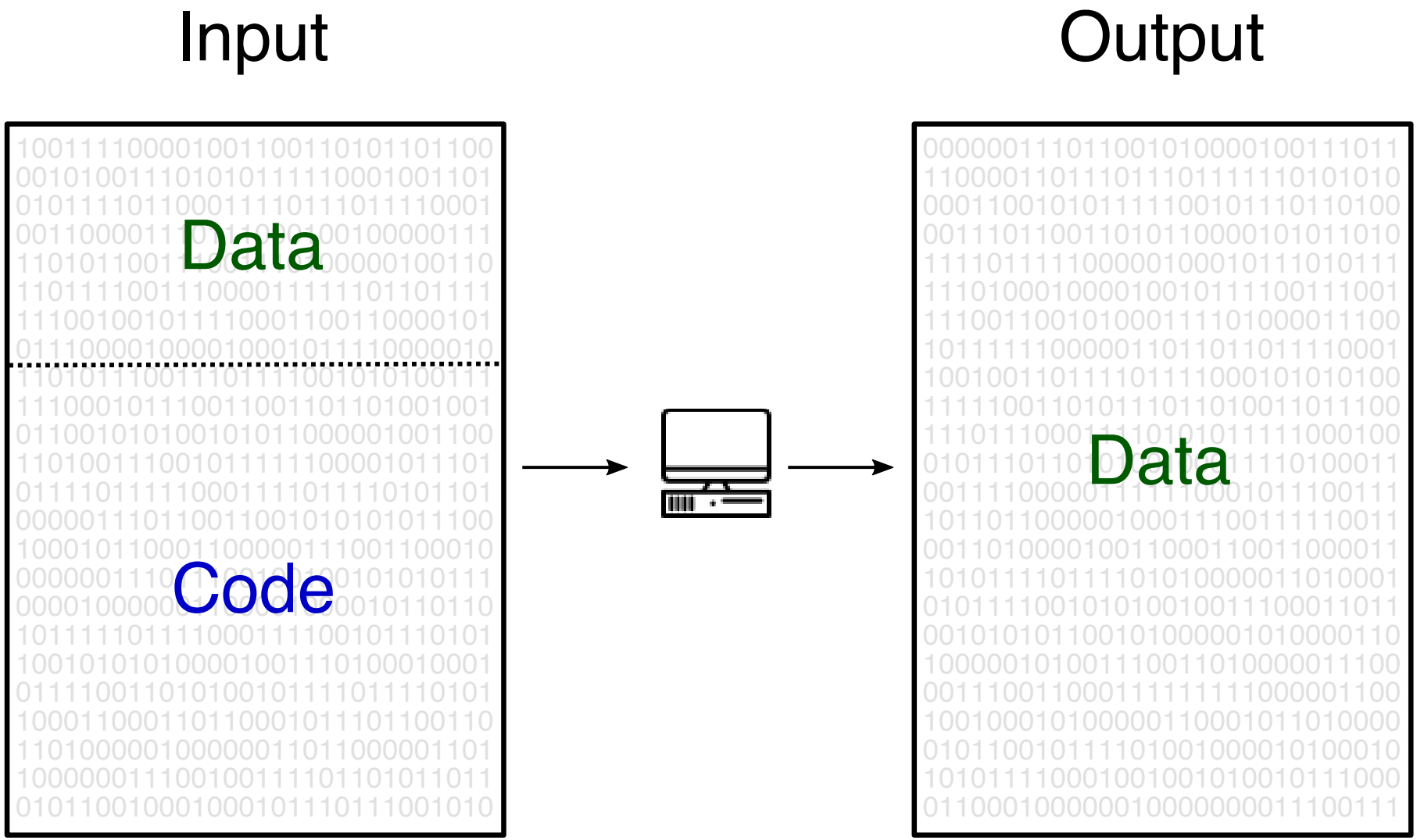
C’est quoi un calcul ?

Input

Output

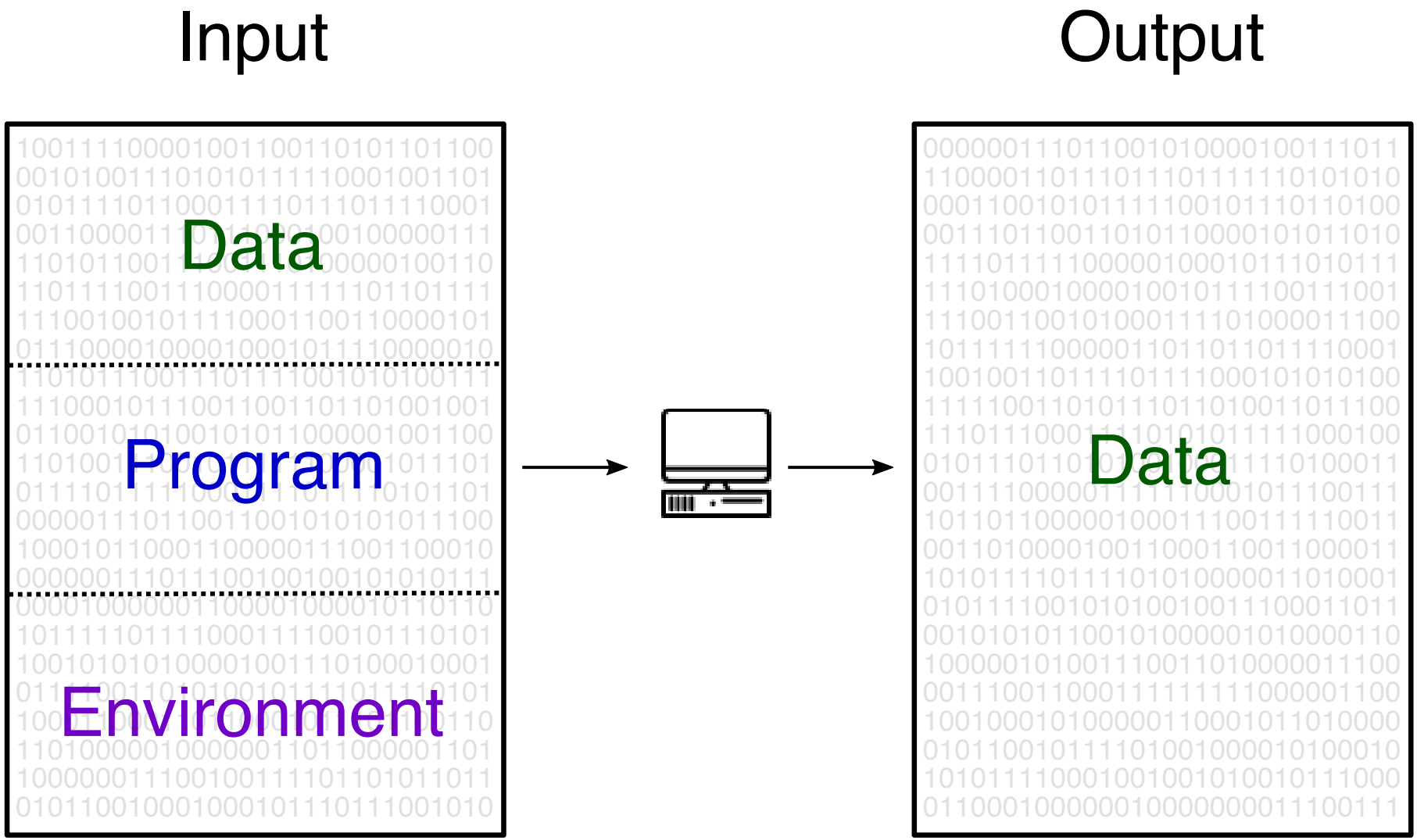


C’est quoi un calcul ?



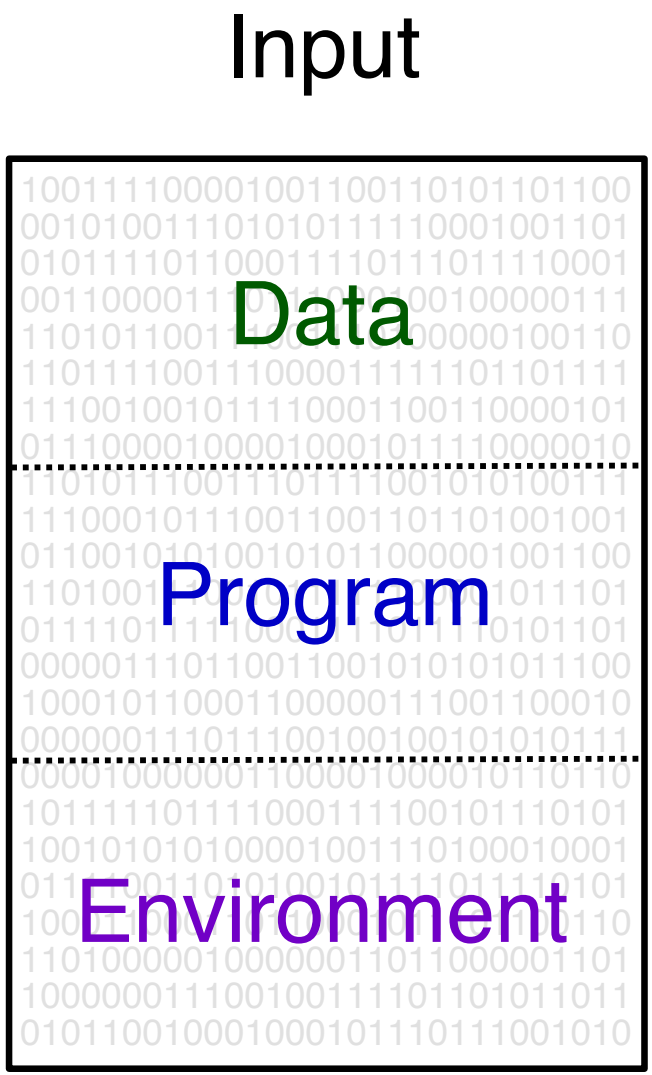
Computer by Creative Stall from the Noun Project

C’est quoi un calcul ?



Computer by Creative Stall from the Noun Project

Le point de vue du chercheur



my research

my colleagues' code

stuff I don't care about

Que fait ce programme ?

data_analysis.py

```
from datalib import Dataset

points = [(1, 1), (-1, 1), (2, 4)]

data = Dataset()
for x, y in points:
    if x > 0:
        data.add_value(y)
print(data.average())
```

Réponse rapide :

Il faut bien connaître ses bibliothèques *et langages*

datalib.py

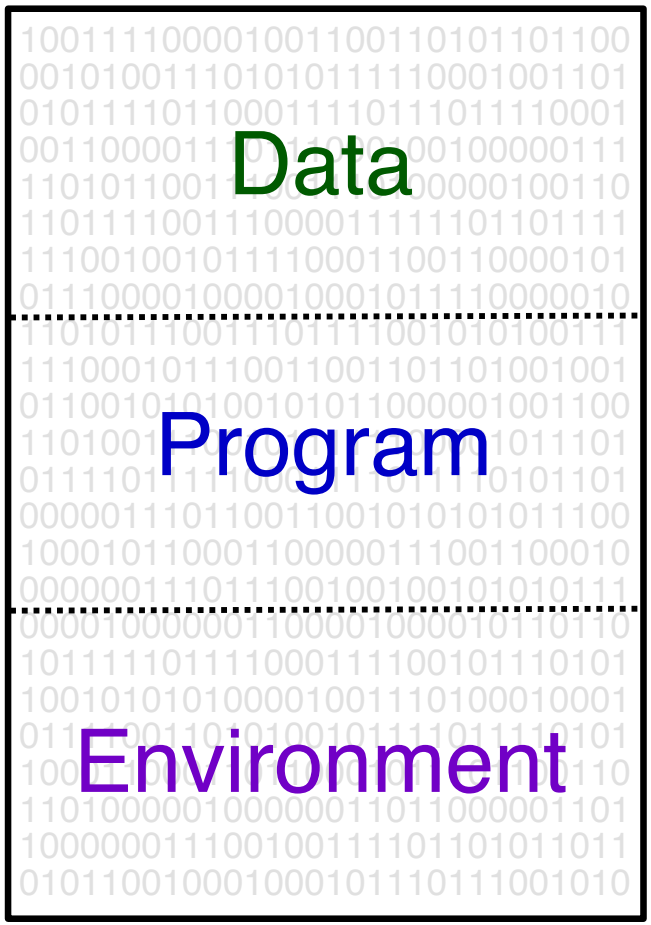
```
class Dataset(object):

    def __init__(self):
        self.values = []

    def add_value(self, value):
        self.values = [value]

    def average(self):
        return sum(self.values, 0)/len(self.values)
```

Quelle surprise ! `add_value` ne garde que la dernière valeur !

Le résultat de `data_analysis.py` est donc 4. Plus précisement : 4 en Python 2 mais 4.0 en Python 3.

Donner un sens aux bits

Input



zeros and ones

interpretation of the data

interpretation of the program

Parenthèse : la sécurité informatique

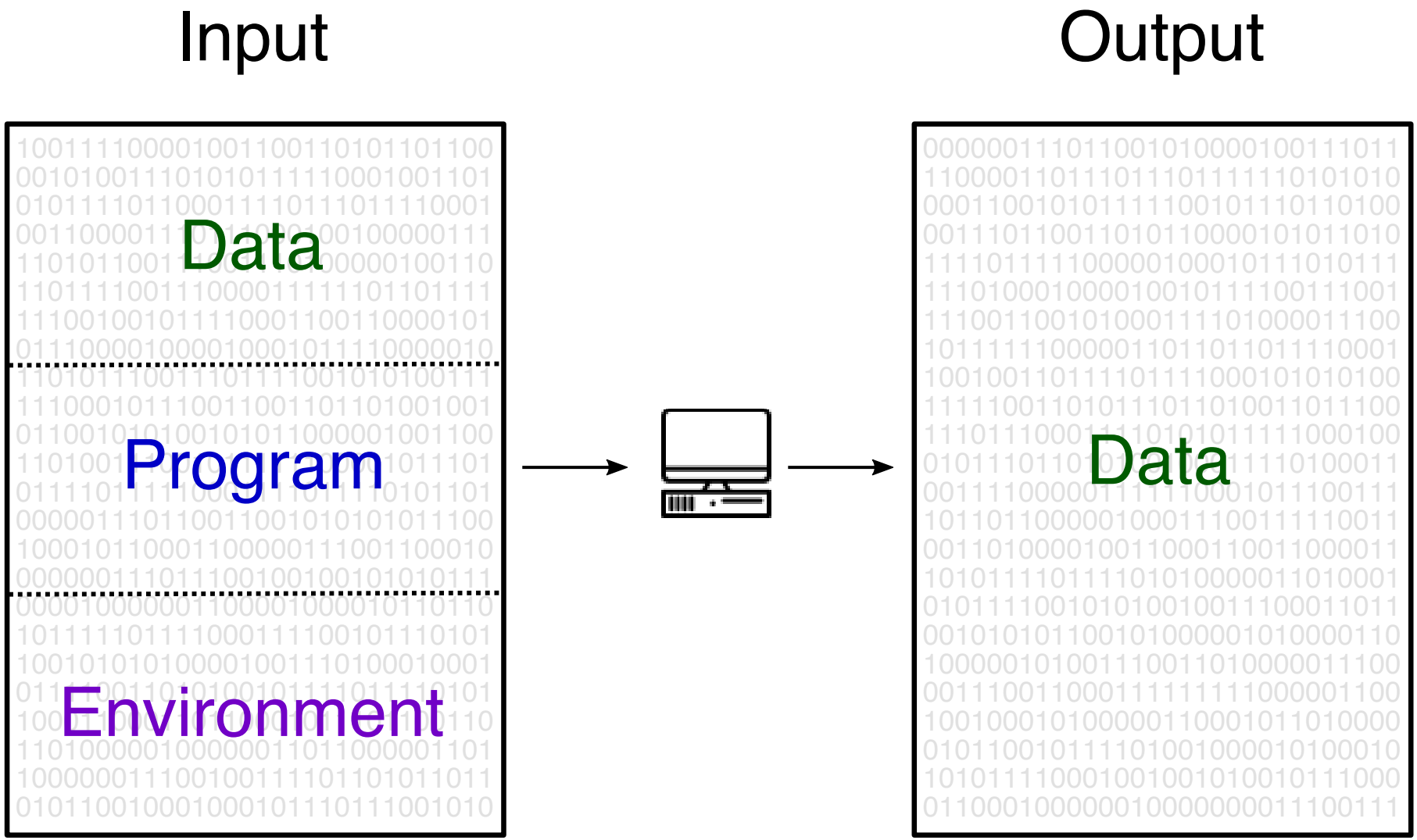
TURING AWARD LECTURE

Reflections on Trusting Trust

To what extent should one trust a statement that a program is free of Trojan horses? Perhaps it is more important to trust the people who wrote the software.

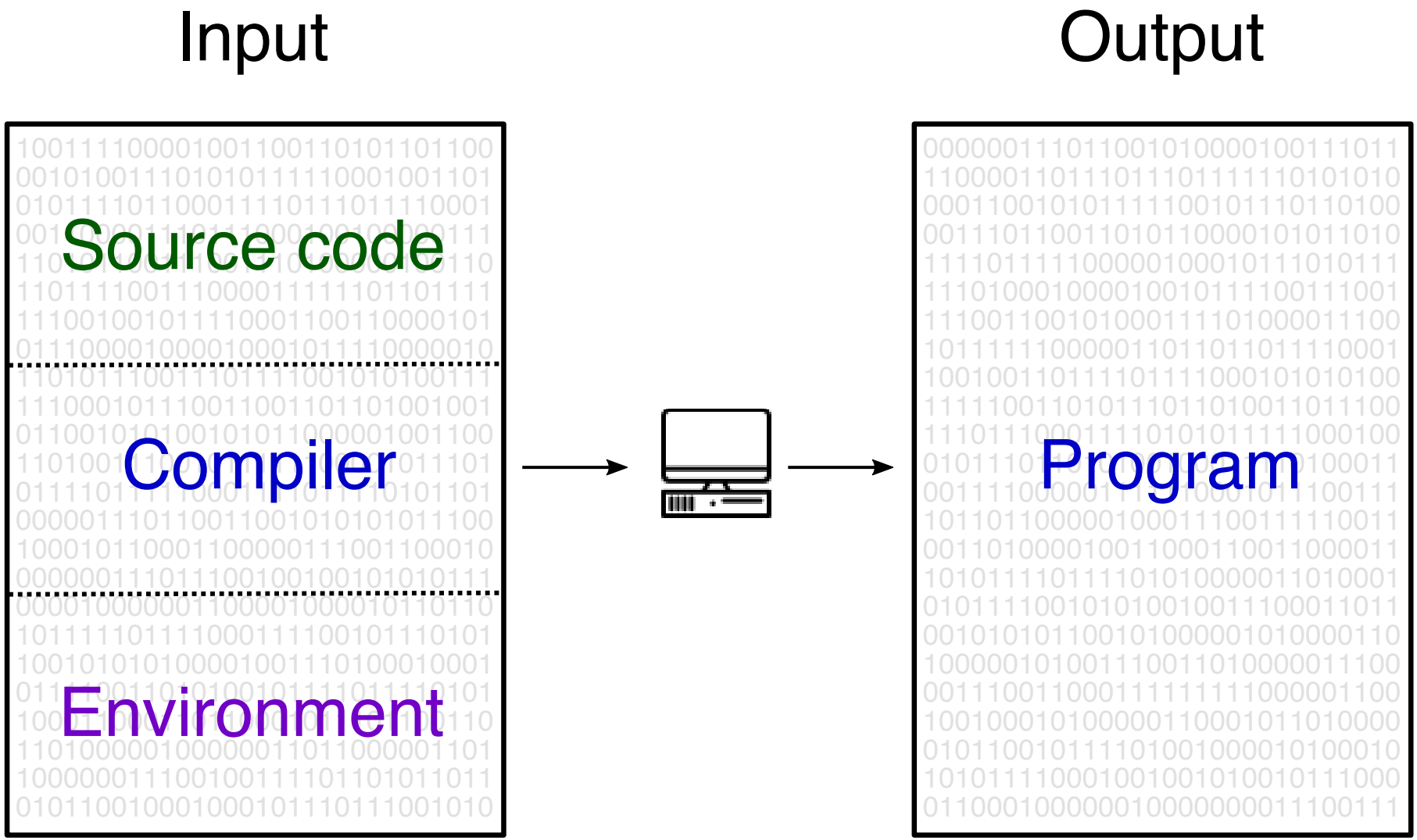
KEN THOMPSON

Provenance d’un calcul



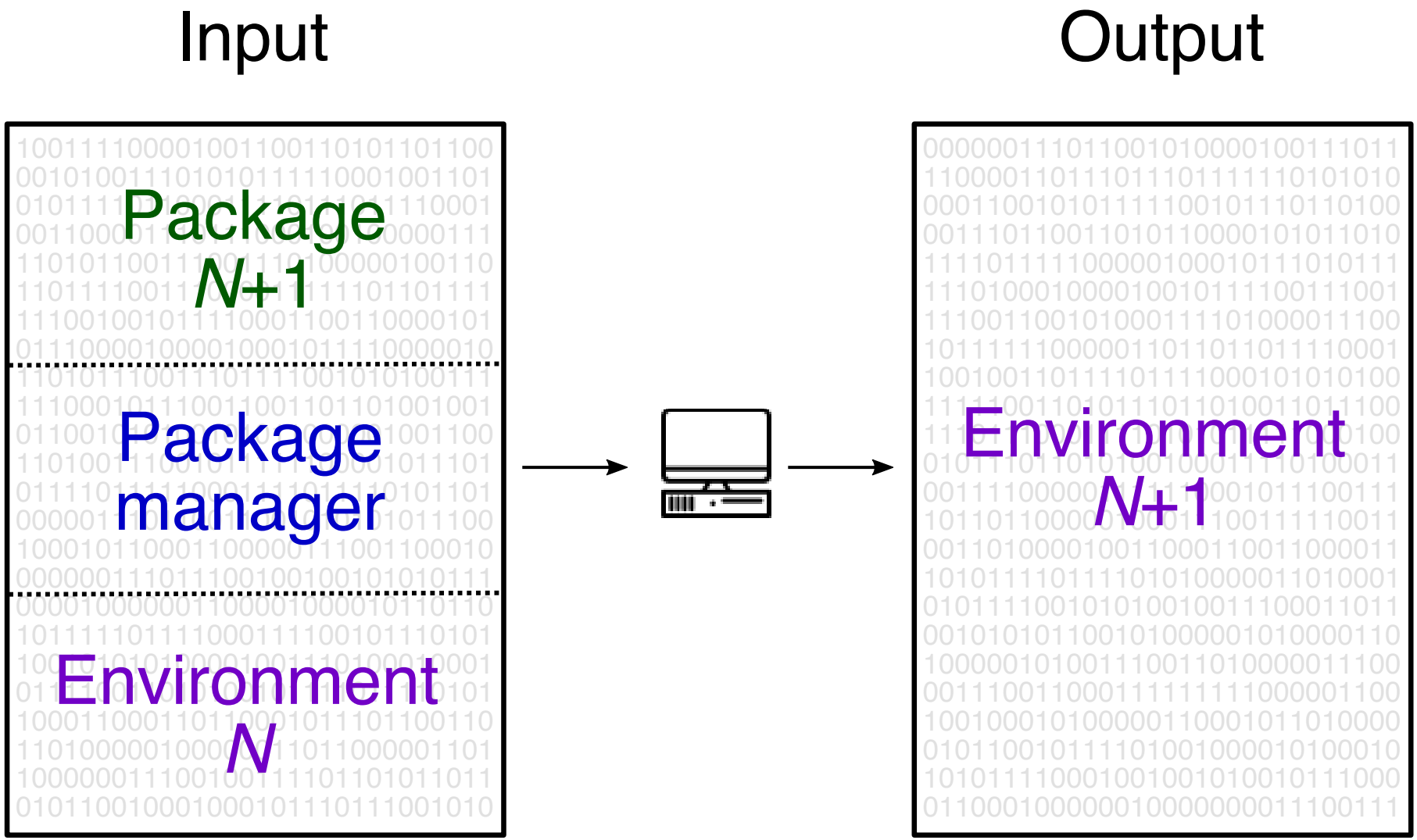
Computer by Creative Stall from the Noun Project

D’où vient le programme ?



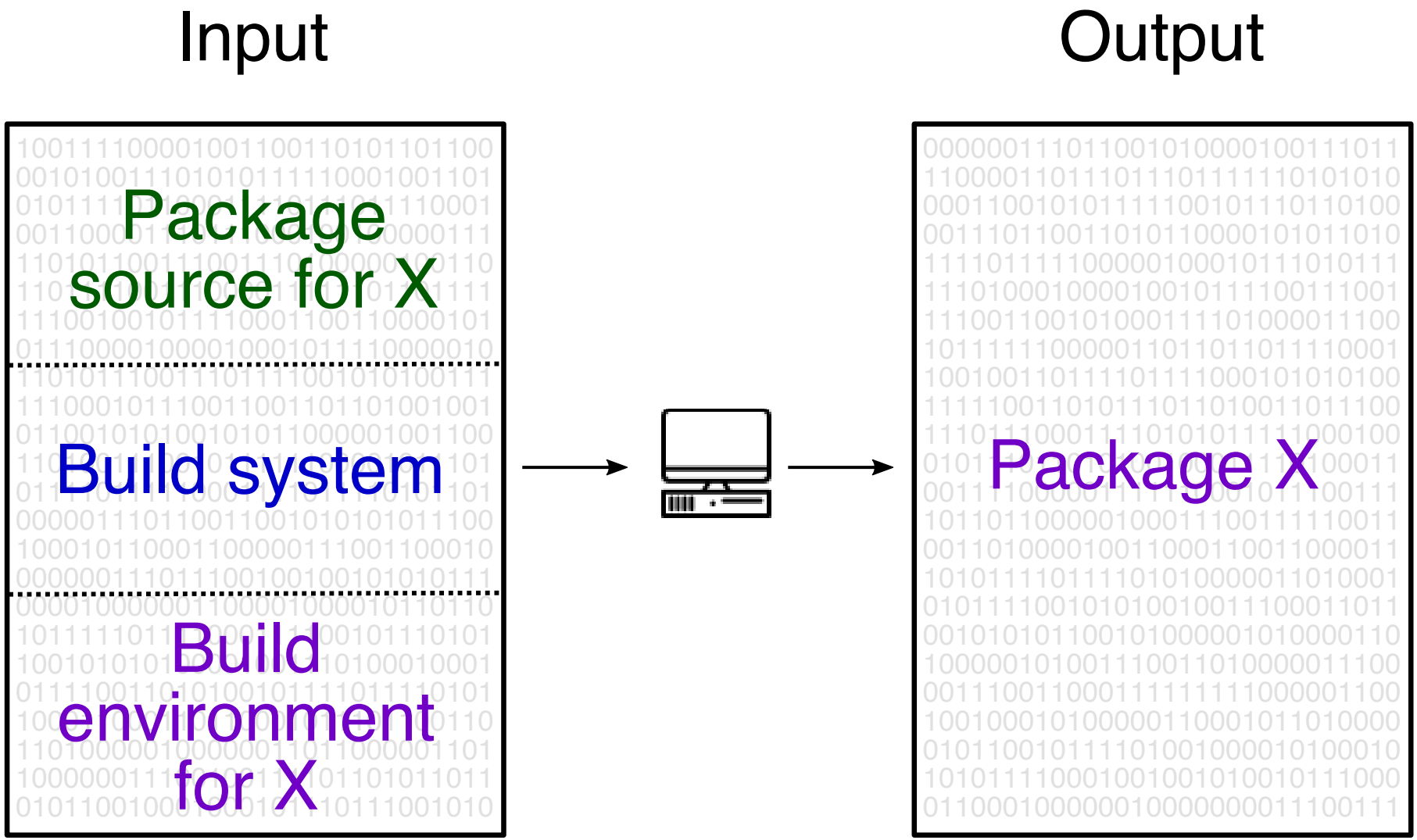
Computer by Creative Stall from the Noun Project

Et d’où vient l’environnement ?



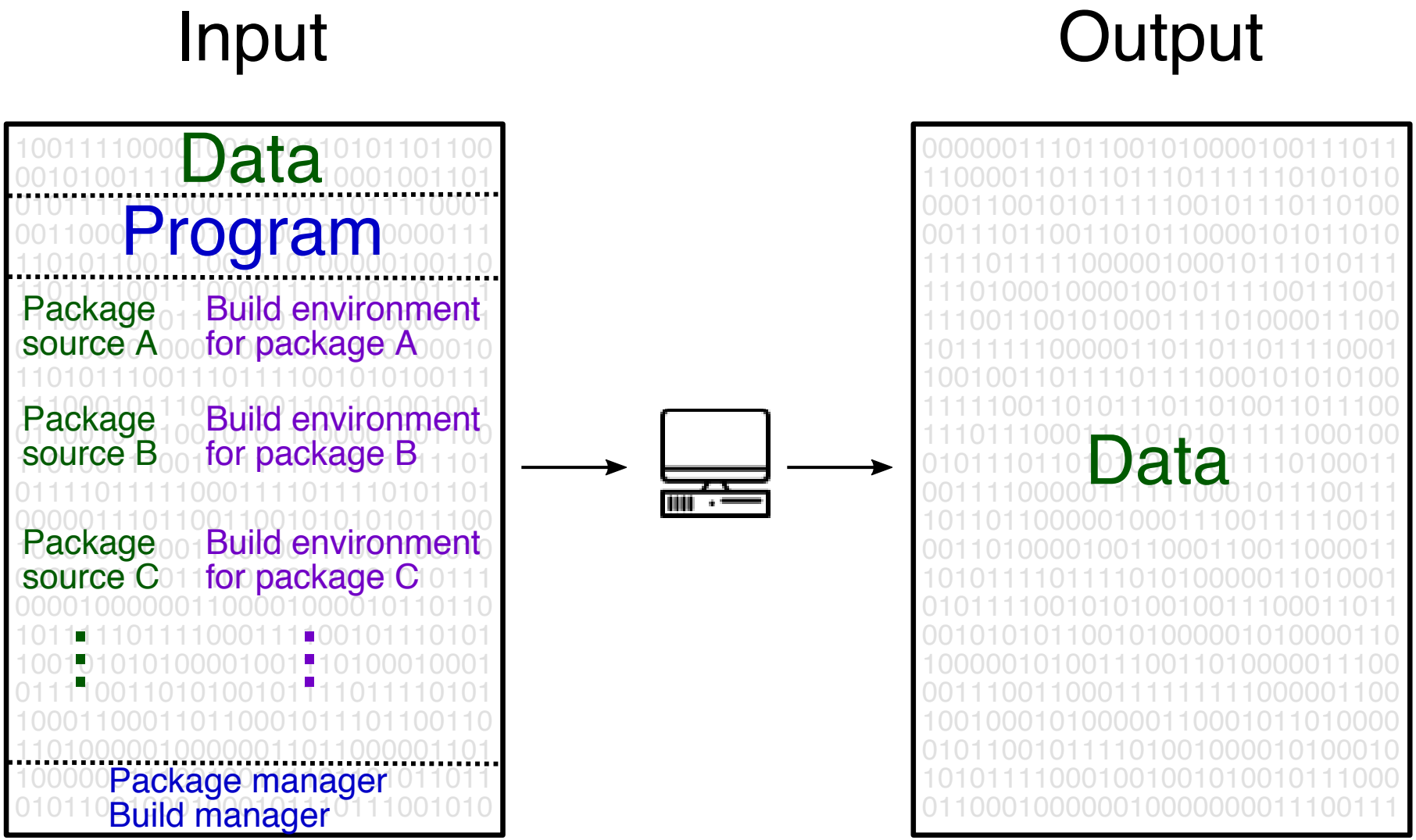
Computer by Creative Stall from the Noun Project

Bon, alors, d'où viennent les paquets ?



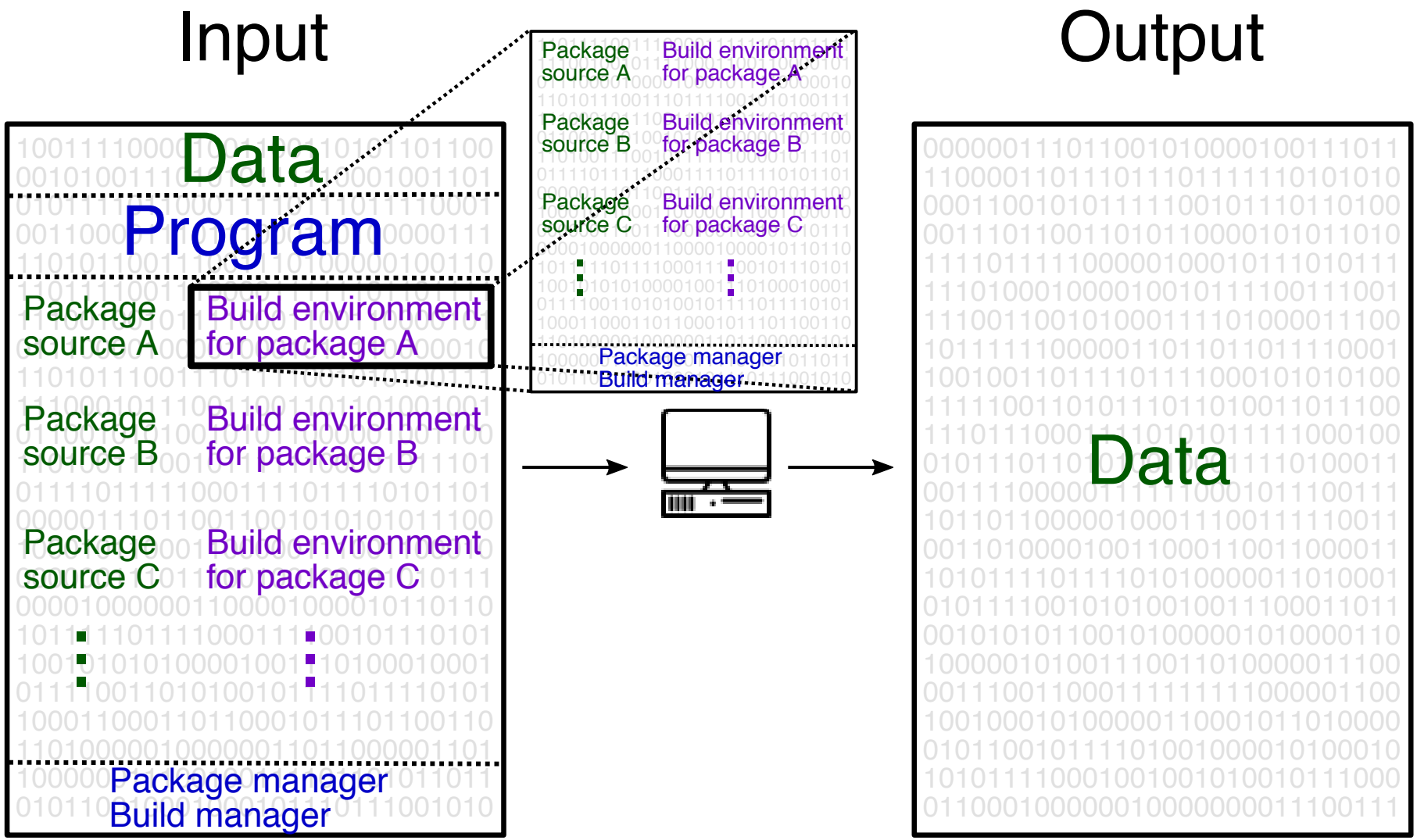
Computer by Creative Stall from the Noun Project

En résumé...



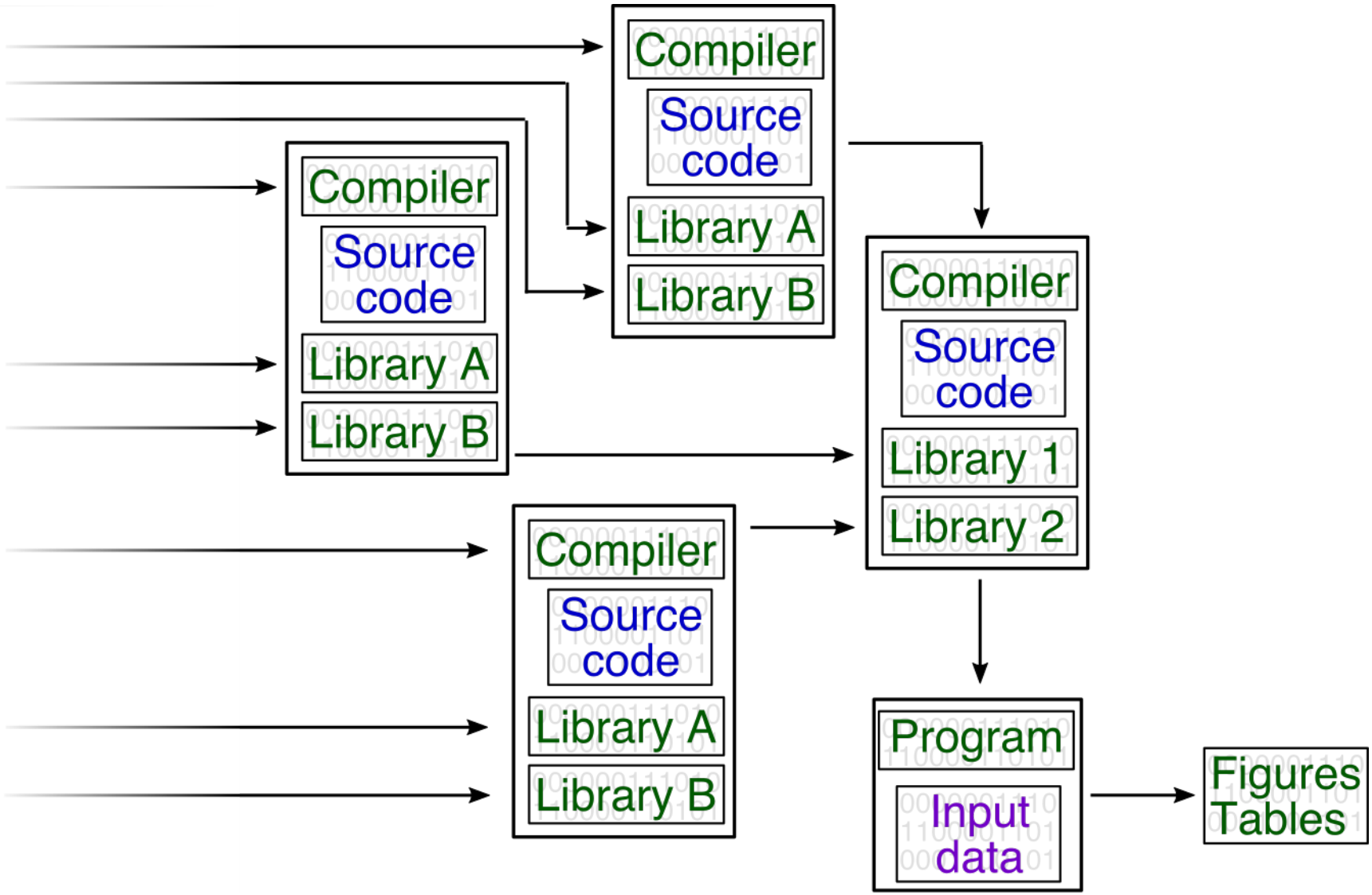
Computer by Creative Stall from the Noun Project

Sans oublier...



Computer by Creative Stall from the Noun Project

Un autre point de vue



La reproductibilité, c’est possible !

1. Point de départ : un environnement préservé, archivé, récupérable à l’identique à tout moment.
2. Conserver une trace précise de chaque étape d’exécution :
   - Programme exécuté.
   - Données d’entrée.
3. Conserver une trace précise de l’ordre des étapes.

Deux implémentations : Nix, Guix

Les erreurs de débutant

Références imprécises aux composantes

- Python 3.8 or later
- `https://github.com/python/cpython`, commit `b5711c940f70af89f2b4cf081a3fcd83924f3ae7`
- `swh:1:rev:b5711c940f70af89f2b4cf081a3fcd83924f3ae7`

Oublier l'environnement initial

- requires Python 3.8 and NumPy 1.17.3
- packages `python` and `python-numpy` in Guix commit `88c2e72af023fae0325bfdaf7078a1cd263f9ffc`

S'appuyer sur un environnement non pérenne

- `docker pull python:3.8.7-alpine-3.11`
- `guix time-machine -commit=88c2e72af023fae0325bfdaf7078a1cd263f9ffc -environment -ad-hoc python python-numpy`

Les trois défis de la reproductibilité numérique

- L'arithmétique à virgule flottante
- Le calcul parallèle
- Comment réconcilier la manipulation interactive avec la reproductibilité ?

L’arithmétique à virgule flottante

- Standardisé en 1985 : norme IEEE 754
- Universellement acceptée aujourd’hui
- Ses opérations sont précisément spécifiées...
- ...et parfaitement déterministes
- Donc : aucune particularité pour la reproductibilité ! *Donc : aucune particularité pour la reproductibilité !*
- Aucun langage de programmation donne un accès direct aux opérations IEEE 754.
- Les optimisations modifient les résultats.
- Le programmeur n’a pas le contrôle complet sur les résultats.
- Il faut intégrer la compilation dans le calcul reproductible : même compilateur, mêmes options.
- Guix s’en charge.

Le calcul parallèle : l'adaptation à la machine

Parallélisation : découpe du calcul en morceaux adaptées à l'architecture de la machine

- nombre de processeurs
- type de processeurs
- taille de la mémoire de chaque noeud

Ces paramètres font partie de la définition du calcul, il faut les préserver.

Peu de support par les outils d'aujourd'hui.
Inconvénient : il faut une machine presque identique pour reproduire le calcul.

Le calcul parallèle : non-déterminisme

Pacte avec le Diable

- (espoir d'un) gain de temps
- perte de contrôle sur l'ordre des opération
  $\longrightarrow$ perte de reproductibilité pour les flottants
- évitable, on peut faire du calcul parallèle déterministe

Que faire ?

- Sauver la réplicabilité : reduire l'impact des aléas sur les résultats.
- Exemple : ReproBLAS
- Travail en algorithmique numérique
- Effort plus important que pour la reproductibilité du calcul déterministe !

Au-delà de la reproductibilité

Que fait ce programme parfaitement reproductible ?

```
#define _ F-->00 || F-OO--;
long F=00,OO=00;
main(){F_OO();printf("%1.3f\n", 4.*-F/OO/OO);}F_OO()
{
             _-_-_-_
       _-_-_-_-_-_-_-_-_-_
    _-_-_-_-_-_-_-_-_-_-_-_-_
  _-_-_-_-_-_-_-_-_-_-_-_-_-_-_
 _-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_
 _-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_
_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_
_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_
_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_
_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_
 _-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_
 _-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_
  _-_-_-_-_-_-_-_-_-_-_-_-_-_-_
    _-_-_-_-_-_-_-_-_-_-_-_-_
       _-_-_-_-_-_-_-_-_-_
            _-_-_-_
}
```

International Obfuscated C Code Contest, 1988

Résumé

- La reproductibilité est une source de confiance dans un résultat.
- Il y en a d'autres, complémentaires.
- Pour le calcul, elle peut être assurée automatiquement...
- ... si nous faisons l'effort pour y arriver.
- Pour le calcul parallèle, le déterminisme reste possible...
- ...mais les outils d'aujourd'hui ne l'encouragent pas.
- Prochain défi : la compréhensibilité