# BI4PEOPLE

## WP5 : Privacy by design

elico
Équipe de recherche de Lyon en sciences de l'information et de la communication

TRIMANE

eric

IRIT

fing

TUBÀ
TUBE À EXPÉRIMENTATIONS URBAINES

# Avancement

- Reflexions sur le/les scénario/scénarii
- Un article accepté
  - Doan, Thi & Messai, Mohamed-Lamine & Gavin, Gérald & Darmont, Jérôme. (2023). A survey on implementations of homomorphic encryption schemes. The Journal of Supercomputing. 1-42. 10.1007/s11227-023-05233-z.

# Rappels sur les cryptosystèmes homomorphes

- Permet de faire des calculs sur des valeurs inconnues (encryptées)

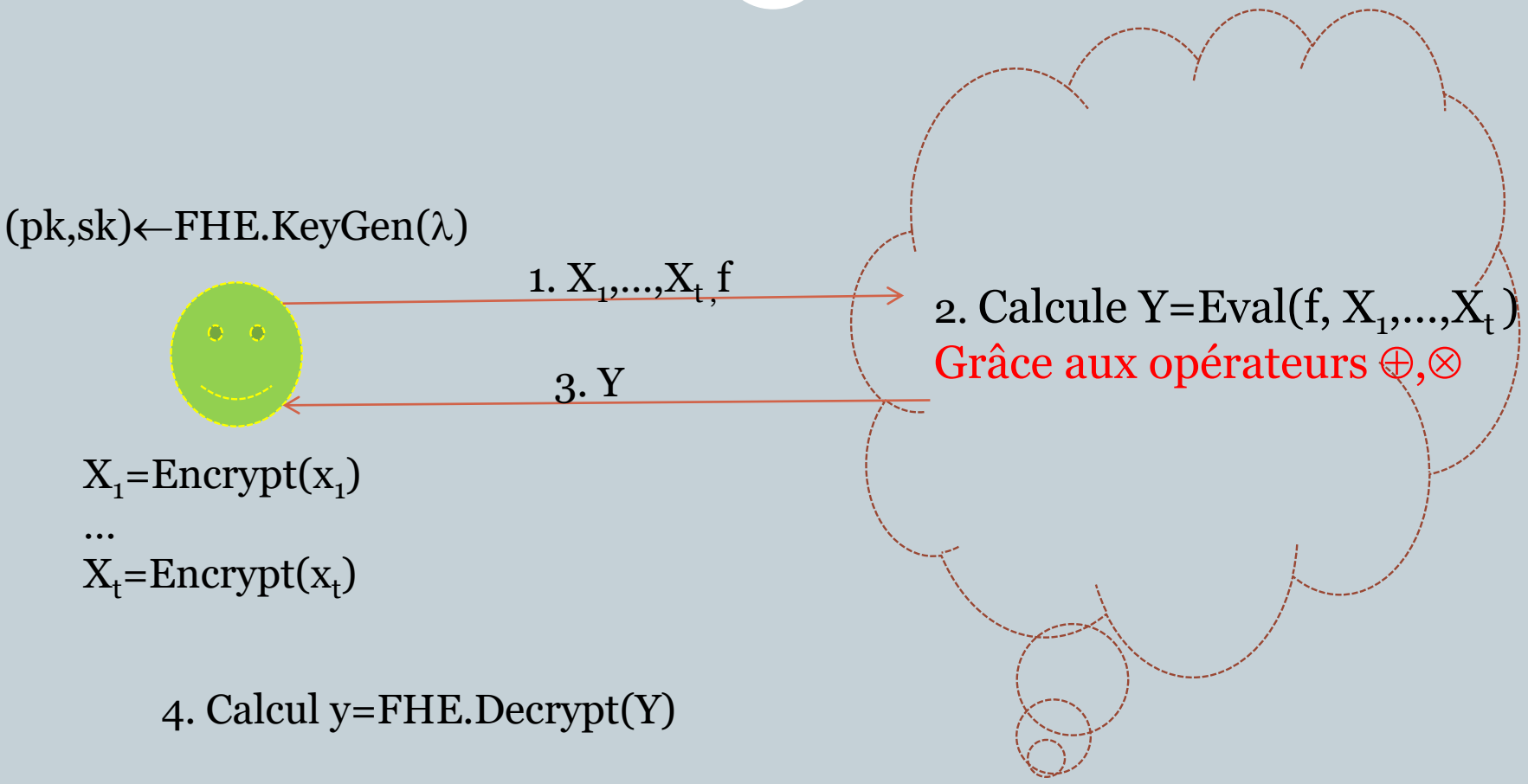
- Connaissant deux encryptions X=[x] et Y=[y] on peut obtenir les encryptions :

  $Z=X \oplus Y=[x+y]$

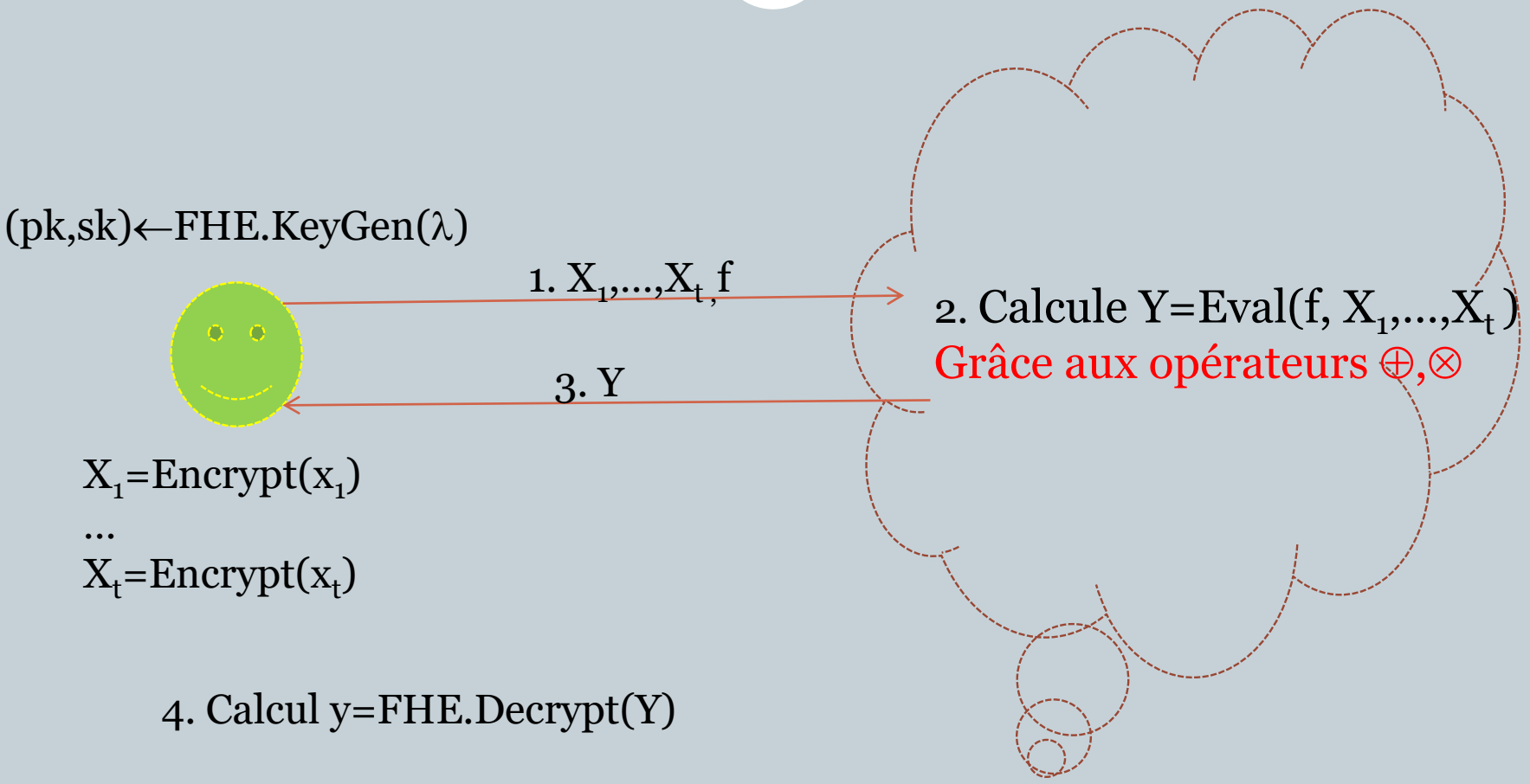
  $Z'=X \otimes Y=[xy]$

- $\oplus$ et $\otimes$ sont des algorithmes publiques
  - Ils n'utilisent pas la clé secrète sk

# Application au cloud computing

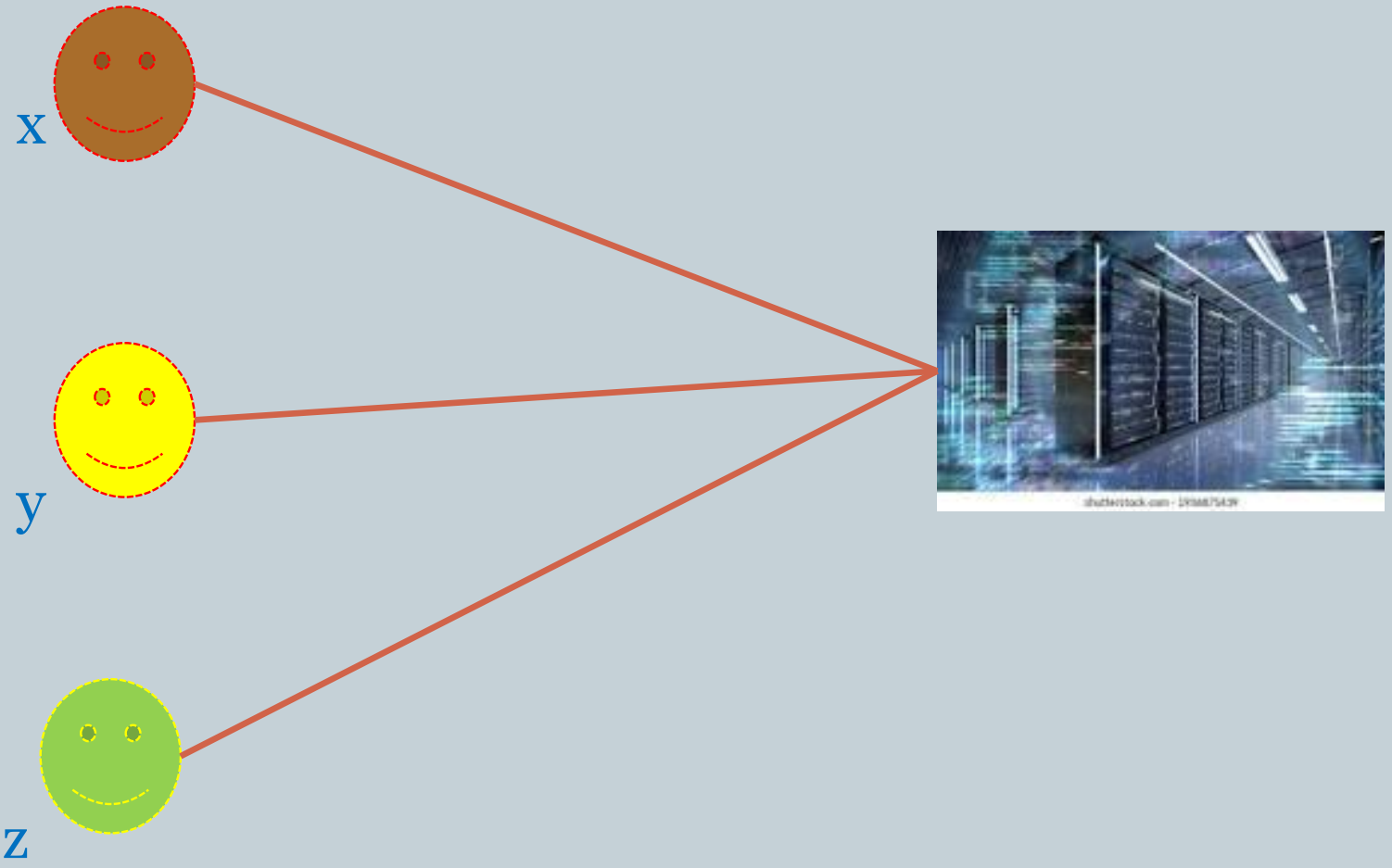


# Application au cloud computing

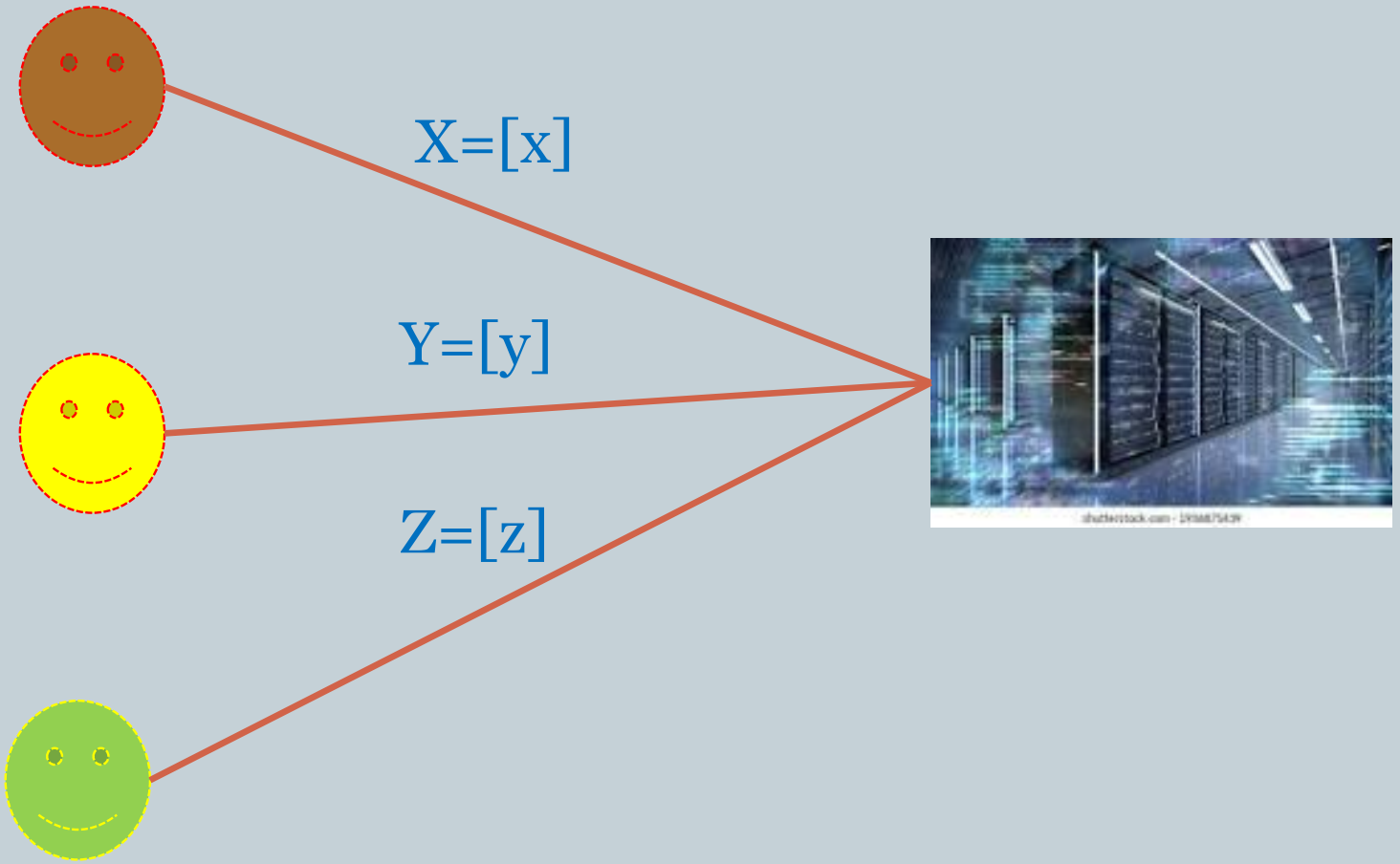



**Aucune restriction sur l'algorithme f…**

# Application au projet



# Setup : envoi des données encryptées



# Envoi d'une requête

f

X,Y,Z

# Envoi du résultat encrypté…

…si requête acceptée par le serveur

f

W=Eval(f,X,Y,Z)

X,Y,Z

# Décryption en local

f

W=Eval(f,X,Y,Z)

w=Decrypt(sk,W)

X,Y,Z

# Problème résolu ?

- Comment se partager les clés de décryption *sk*
- Quel contrôle a-t-on sur le contrôle ?

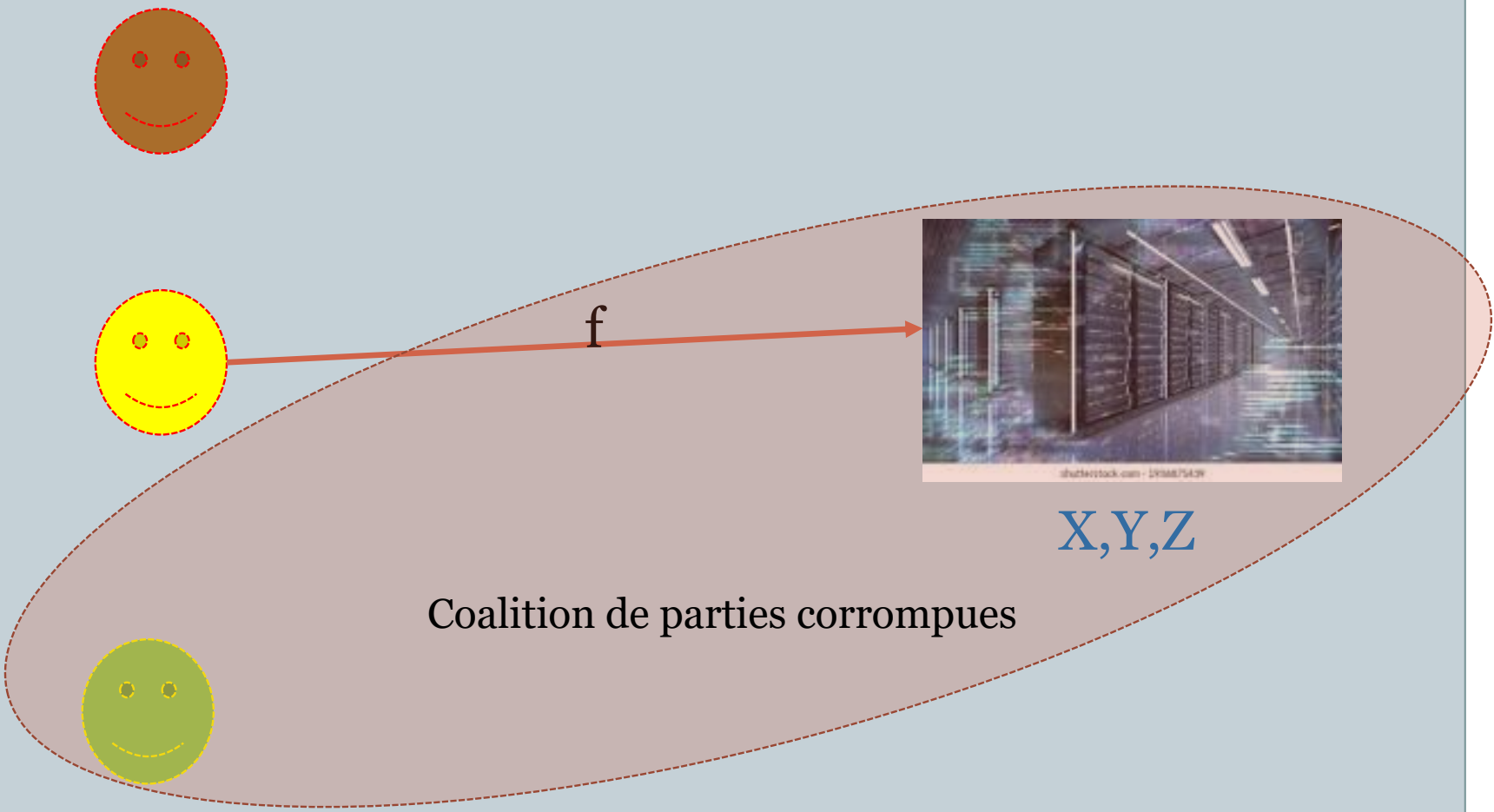
# Une solution ?...oui si...

- Les participants sont identifiés les uns par rapport au autres
  - Ils peuvent communiquer de manière authentifiée
  - L'ensemble des participants est stable au cours du temps.
- Les participants possèdent des ressources de calculs  et de stockage importantes
  - Les participants doivent faire les mêmes calculs que le serveur (pour le contrôler)
  - ⇒ Ressources croissent avec le nombre de participants
  - ⇒ Le participants doivent être online

# Quelques hypothèses simplificatrices

- Seul le serveur est corrompu passif (Honnet but Curious)
  - ⇒ Il respecte le protocole…*il évalue correctement les requêtes*
  - ⇒ Il n'a accès qu'à des données encryptées
- Les autres parties corrompues sont *très minoritaires ou peu coalisées*
  - Les parties honnêtes sont sollicitées pour décrypter leur propres requêtes… *et quelques autres*

# Hypothèse retenue…?



# Premier scenario

- Les parties sont identifiées
- Les parties peuvent communiquer de manière sécurisée
  - via le serveur par exemple (qui jouera le rôle de routeur)
- Le serveur est corrompu mais respecte le protocole
- Le nombre de parties corrompues est faible (serveur+autres)
- Chaque partie reste connectée (autant que possible)
  - Sollicitée pour certaines décryptions.

# Conclusion/Perspectives

- Outils de calcul multi-partie sécurisé peu adaptés à notre problématique
  - Peu de littérature sur le paradigme client/serveur
  - Hypothèses (trop?) fortes à faire
  - Temps de calculs trop longs
  - Des compromis sont à imaginer
- Développement du scenario précédent
  - Serveur corrompu mais respecte le protocole
  - Peu de parties corrompues
- Réfléchir à l'utilisation de blockchains
- *Demander à ChatGPT* ☺