XWIKI NATIS

Architecture

Michael
XWIKI NATIS

XONATIS

XWIKI NATIS

1. Architecture client-serveur
2. Architecture des applications
3. Architecture SOA

XONATIS

XWIKI NATIS

Architecture client-serveur

XWIKI SAS

XONATIS

XWIKI NATIS

Un serveur

XWIKI NATIS

Un serveur

Une machine ou un logiciel fournissant des services à d'autres machines ou logiciels, par exemple 24h/24

XWIKI NATIS

Un serveur

Un logiciel fournissant des services
à d'autres logiciels

XWIKI SAS

Un serveur

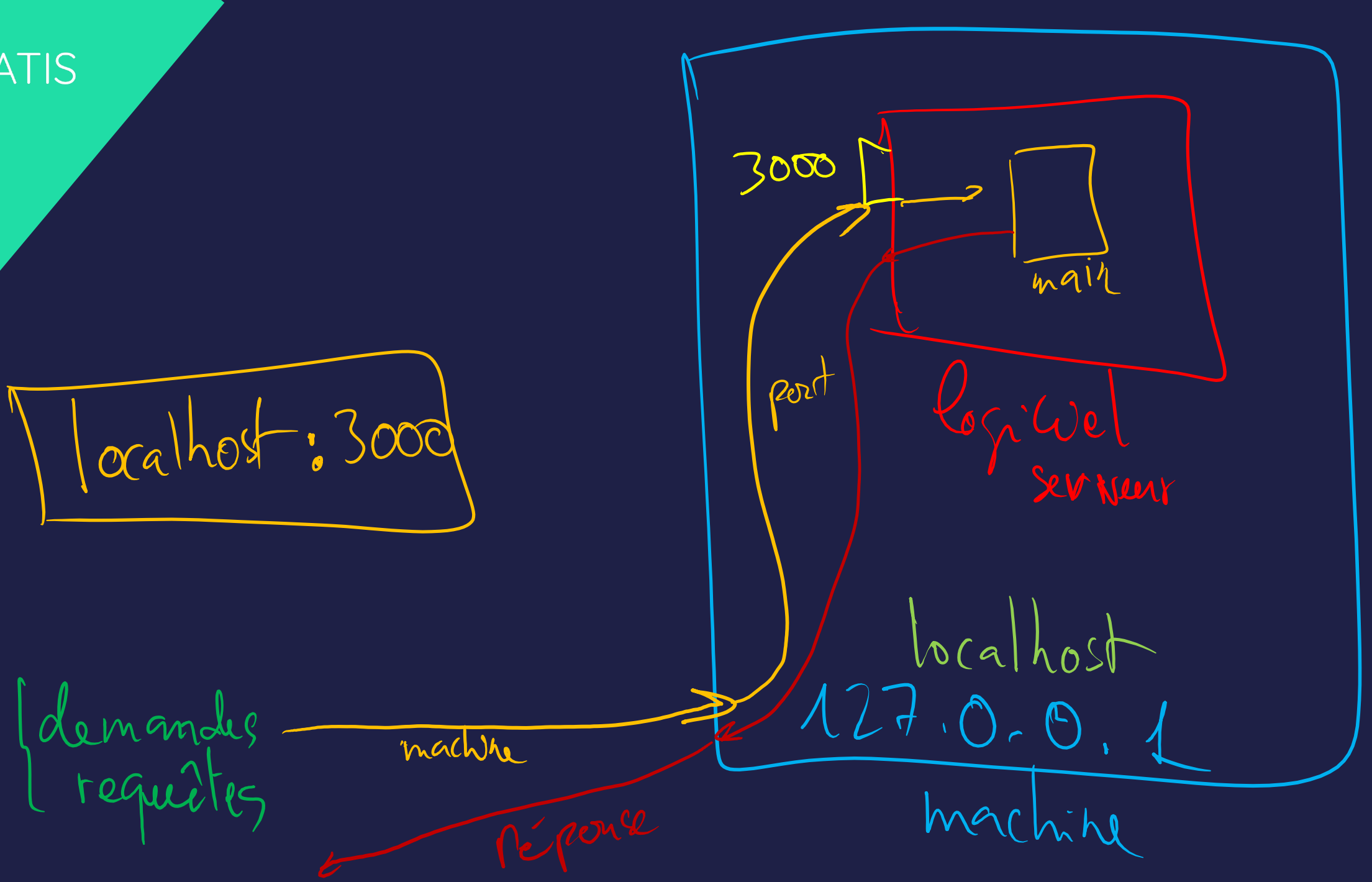
Un logiciel fournissant des services
à d'autres logiciels

XWIKI NATIS

Un serveur

Un serveur est localisé par :
- la machine (identifiée par une adresse IP ou un hostname)
- le port (identifié par un numéro)

XDNATIS

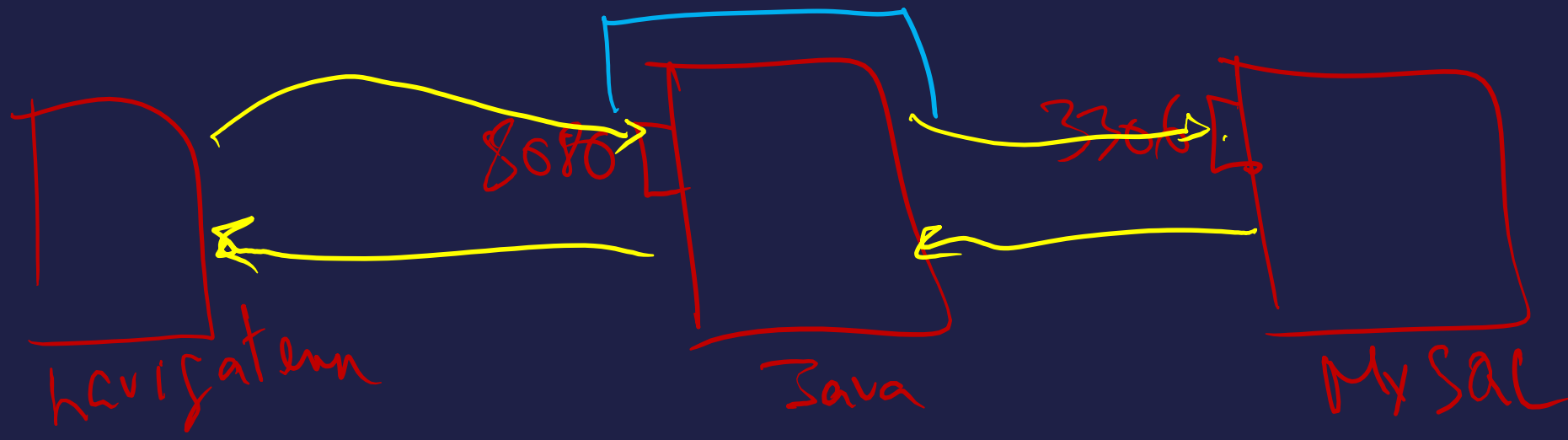

3000
main
port
logiciel
serveur
localhost:3000
demandes
requêtes
machine
réponse
localhost
127.0.0.1
machine

XONATIS

XWIKI NATIS

Architecture client-server

1. Un client fait une requête à un serveur
2. Le serveur donne une réponse au client



XWNATIS

Chrome

Affichage

HTML + JavaScript + CSS

Requête HTTP

Reponse de données XML ou JSON

WS

Metier

Java

SELECT * FROM RESULTATS WHERE document LIKE '%vélo%'

id	nom
1	
2	
3	
4	
5	
6	
7	
8	

BDD

Données

SQL, autre

XNATIS

Requête HTTP

- machine + port
- chemin (path)
- entêtes (headers)
- body / payload
- methode (method)

localhost : 8080

/login

- Accept : application/json
- Content-Type : application/json

{"username":"toto", "password":"toto"}

POST

Chrome

Affichage

erreur

HTML + JavaScript + CSS

EXECUTION DE LA REPONSE

1xx
2xx
3xx
4xx
5xx

Réponse HTTP

- status code
- entêtes (headers)
- body / payload

{
"token" : "aYa5567?aZ"
}

201

Content-Type : application/json

WS

Metier

Java

EXUCUTION DU PGM

XWIKI NATIS

SELECT * FROM
RESULTATS WHERE
document LIKE ‘%vélo%’

WS

Metier

Java

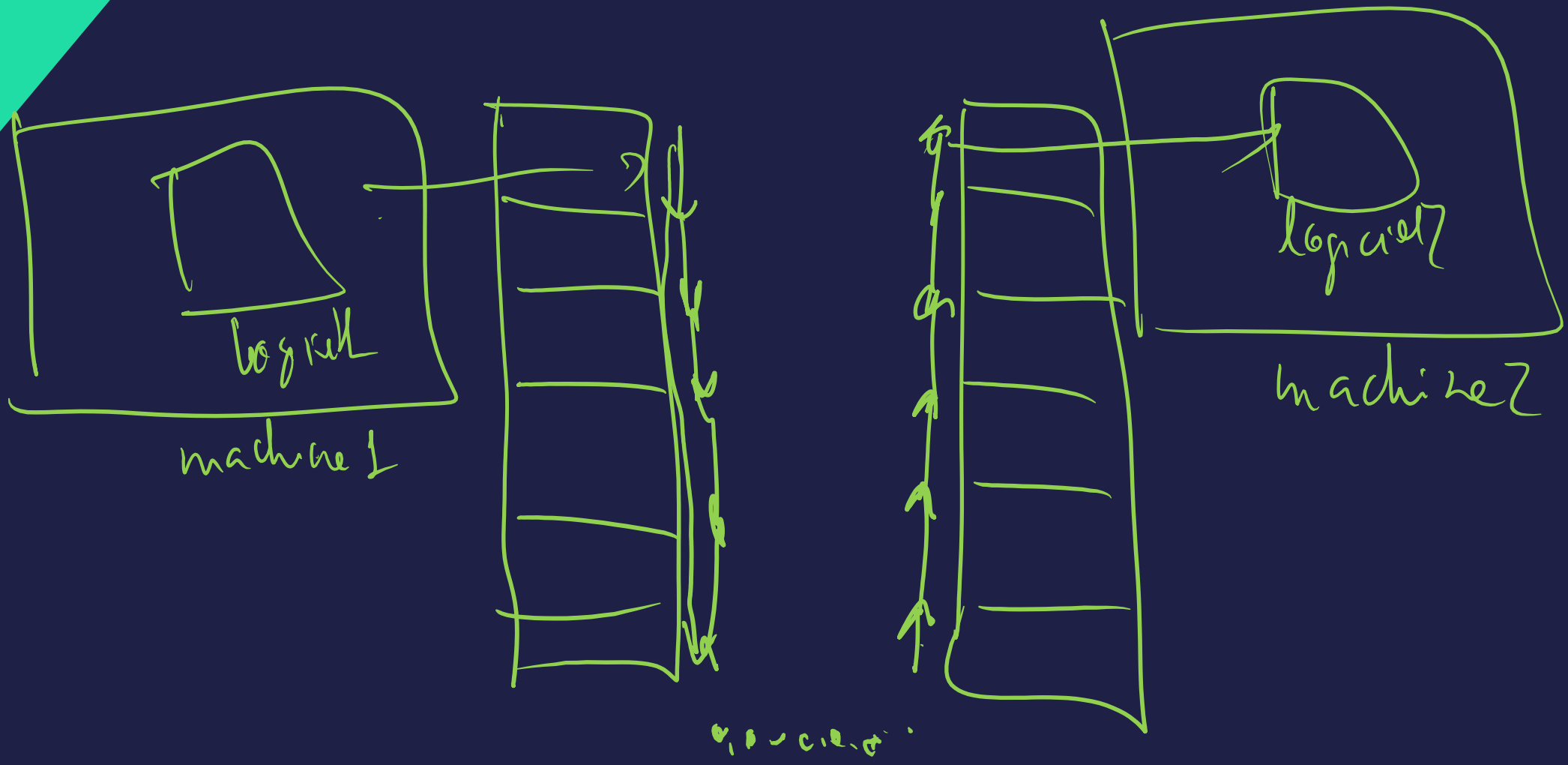
id	nom
1	
2	
3	
4	
5	
6	
7	
8	

BDD

Données

SQL, autre

XDNATIS



XWIKI NATIS

Architecture des applications

XDNATIS

1. Les classes
2. L’intanciation
3. L’abstraction
4. L’héritage
5. Le polymorphisme
6. L’encapsulation

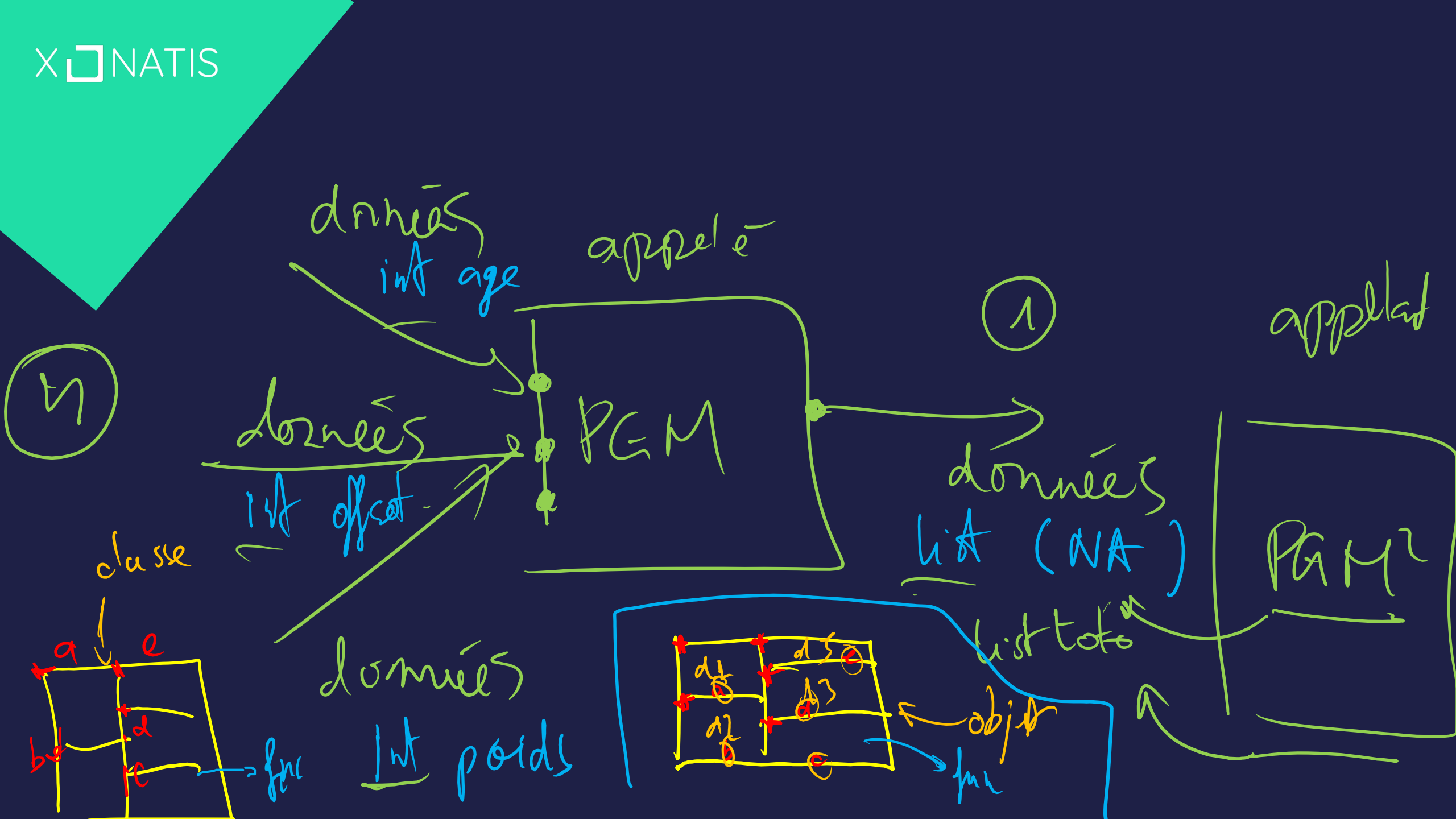
XEBIA NATIS

1. Les classes
2. L’intanciation
3. L’abstraction
4. L’héritage
5. Le polymorphisme
6. L’encapsulation

XWIKI NATIS

LES CLASSES

XONATIS


données
int age
appelé
données
int offset
PGM
1
appelant
données
List (NA)
PGM2
classe
données
Int poids
list toto
objet

XWIKI NATIS

La définition d'un objet
Versus
L'objet concret

XWIKI NATIS

La classe d'un objet
Versus
L'objet

XDNATIS

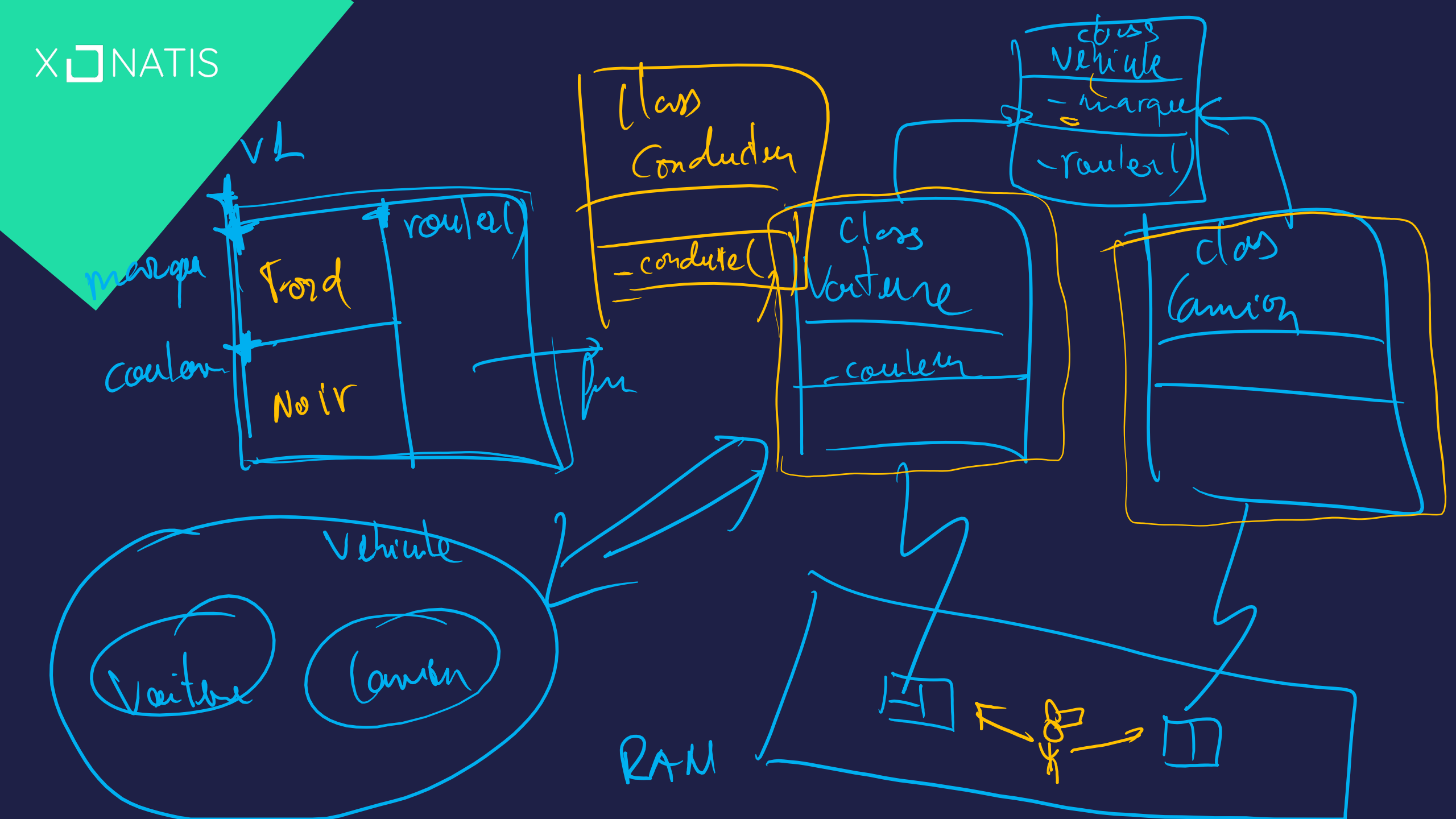
Une classe regroupe des membres, communs à un ensemble d'objets.

Ces membres peuvent être des méthodes ou des propriétés

XWIKI NATIS

Les propriétés définissent les caractérisques d'un ensemble d'objets

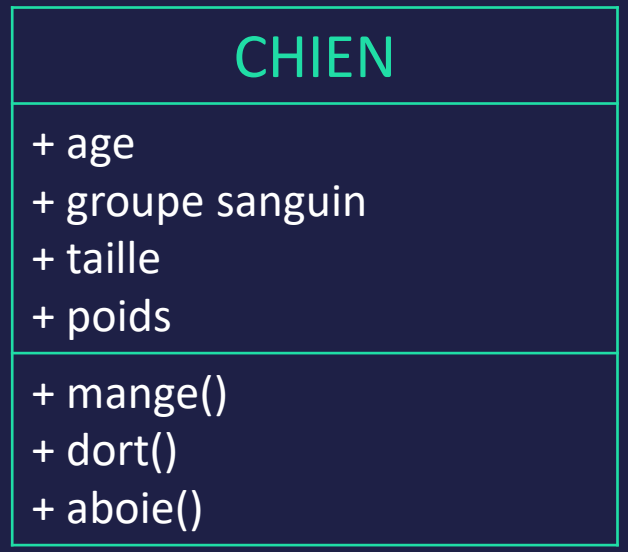
Les méthodes définissent les comportements d'un ensemble d'objets

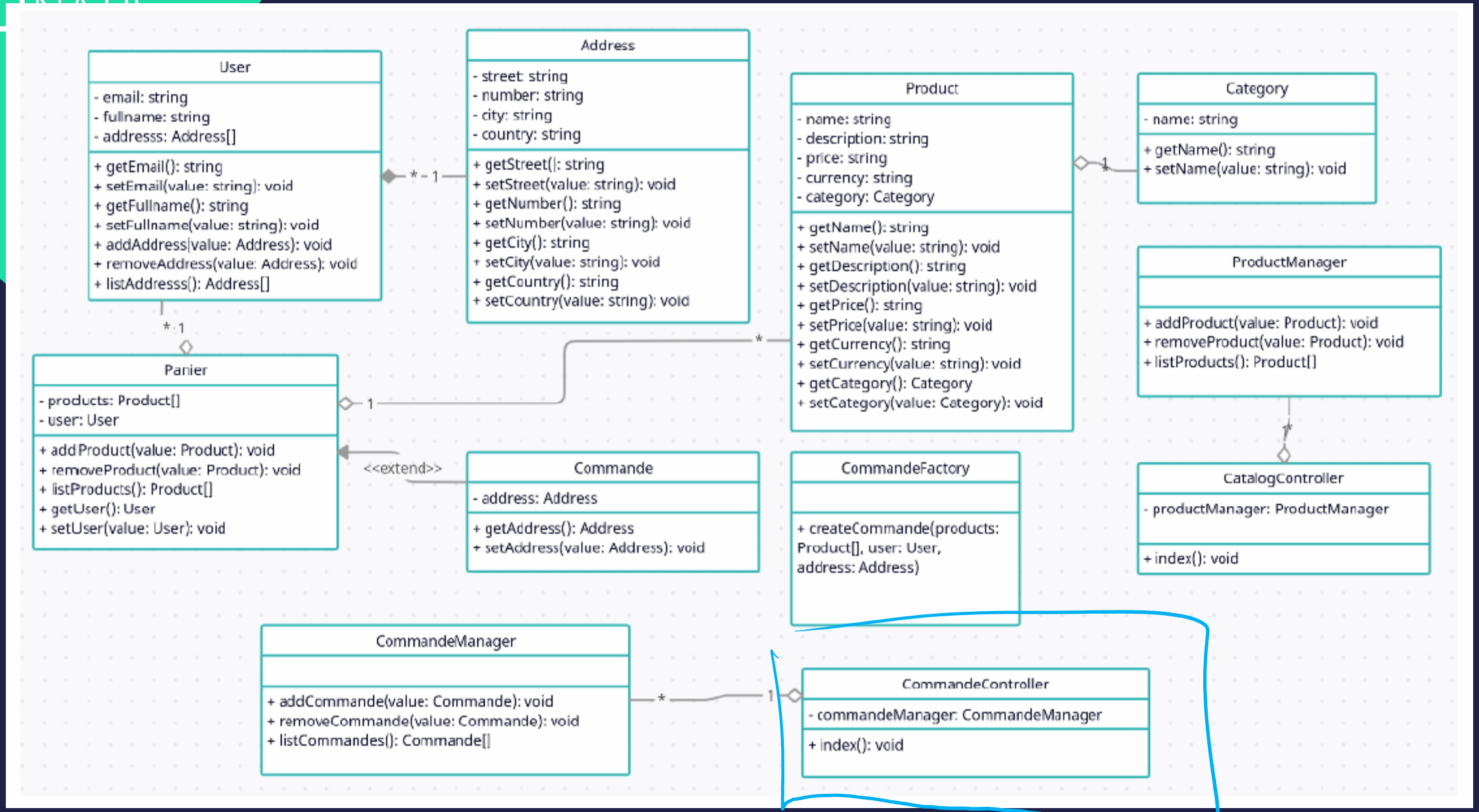

XNATIS
V1
marque
couleur
Ford
Noir
rouler()
feu
Classe Conducteur
-conduire()
Class Voiture
-couleur
class Vehicule
-marque
-rouler()
class Camion
Vehicule
Voiture
Camion
RAM

XDNATIS

Exemples **en UML**

XEBIA NATIS




User
- email: string
- fullname: string
- addresss: Address[]
+ getEmail(): string
+ setEmail(value: string): void
+ getFullname(): string
+ setFullname(value: string): void
+ addAddress(value: Address): void
+ removeAddress(value: Address): void
+ listAddresss(): Address[]
Address
- street: string
- number: string
- city: string
- country: string
+ getStreet(): string
+ setStreet(value: string): void
+ getNumber(): string
+ setNumber(value: string): void
+ getCity(): string
+ setCity(value: string): void
+ getCountry(): string
+ setCountry(value: string): void
Product
- name: string
- description: string
- price: string
- currency: string
- category: Category
+ getName(): string
+ setName(value: string): void
+ getDescription(): string
+ setDescription(value: string): void
+ getPrice(): string
+ setPrice(value: string): void
+ getCurrency(): string
+ setCurrency(value: string): void
+ getCategory(): Category
+ setCategory(value: Category): void
Category
- name: string
+ getName(): string
+ setName(value: string): void
ProductManager
+ addProduct(value: Product): void
+ removeProduct(value: Product): void
+ listProducts(): Product[]
Panier
- products: Product[]
- user: User
+ addProduct(value: Product): void
+ removeProduct(value: Product): void
+ listProducts(): Product[]
+ getUser(): User
+ setUser(value: User): void
<<extend>>
Commande
- address: Address
+ getAddress(): Address
+ setAddress(value: Address): void
CommandeFactory
+ createCommande(products: Product[], user: User, address: Address)
CatalogController
- productManager: ProductManager
+ index(): void
CommandeManager
+ addCommande(value: Commande): void
+ removeCommande(value: Commande): void
+ listCommandes(): Commande[]
CommandeController
- commandeManager: CommandeManager
+ index(): void

XWIKI SAS

XEBIA NATIS

1. Les classes
2. L’intanciation
3. L’abstraction
4. L’héritage
5. Le polymorphisme
6. L’encapsulation

XWIKI NATIS

L'INSTANCIATION

XWIKI NATIS

Et dans un ordinateur ?

Que se passe-t-il dans la RAM ?

XONATIS

XONATIS

XEBIA NATIS

1. Les classes
2. L’intanciation
3. L’abstraction
4. L’héritage
5. Le polymorphisme
6. L’encapsulation

XWIKI NATIS

ABSTRACTION

XWIKI NATIS

Sinon … on s’est fiche ..

XNATIS

Touche de clavier d'ordinateur

XNATIS

Une batterie d’ordinateur

XWIKI NATIS

Une souris d'ordinateur

XNATIS

Un vidéo-projecteur

XDNATIS

Des hauts-parleurs

XWIKI NATIS

Un microphone

XWIKI NATIS

Une pédale de frein

XWIKI SAS

XWIKI NATIS

Abstraction

Utiliser des briques sans connaître leurs détails techniques

XWIKI NATIS

Abstraction

Utiliser des briques sans connaître leurs détails techniques

XONATIS

XEBIA NATIS

1. Les classes
2. L’intanciation
3. L’abstraction
4. L’héritage
5. Le polymorphisme
6. L’encapsulation

XWIKI NATIS

L'HERITAGE

XWNATIS

CHIEN
+ age + groupe sanguin + taille + poids
+ mange() + dort() + aboie()

CHAT
- Coussinets + age + groupe sanguin + taille + poids
+ mange() + dort() + miauler()

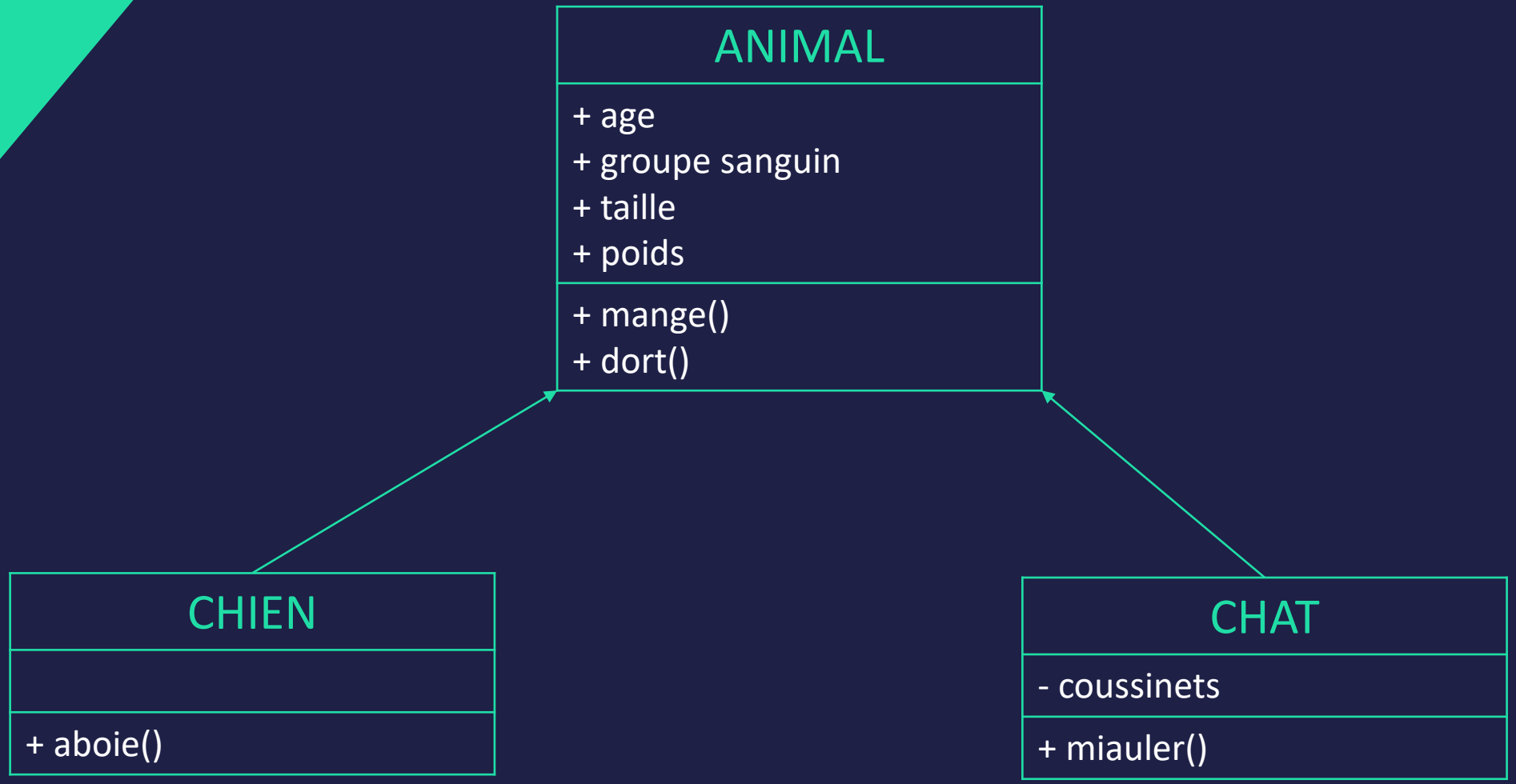
XEBIA NATIS

???
+ age - groupe sanguin # taille + poids
+ mange() + dort()

CHIEN
+ aboie()

CHAT
- coussinets
+ miauler()

XDNATIS


ANIMAL
+ age
+ groupe sanguin
+ taille
+ poids
+ mange()
+ dort()
CHIEN
+ aboie()
CHAT
- coussinets
+ miauler()

XWIKI NATIS

L’héritage permet d’organiser les classes en groupe plus spécifiques qui ont des propriétés et des méthodes qui leur sont propres

XWIKI NATIS

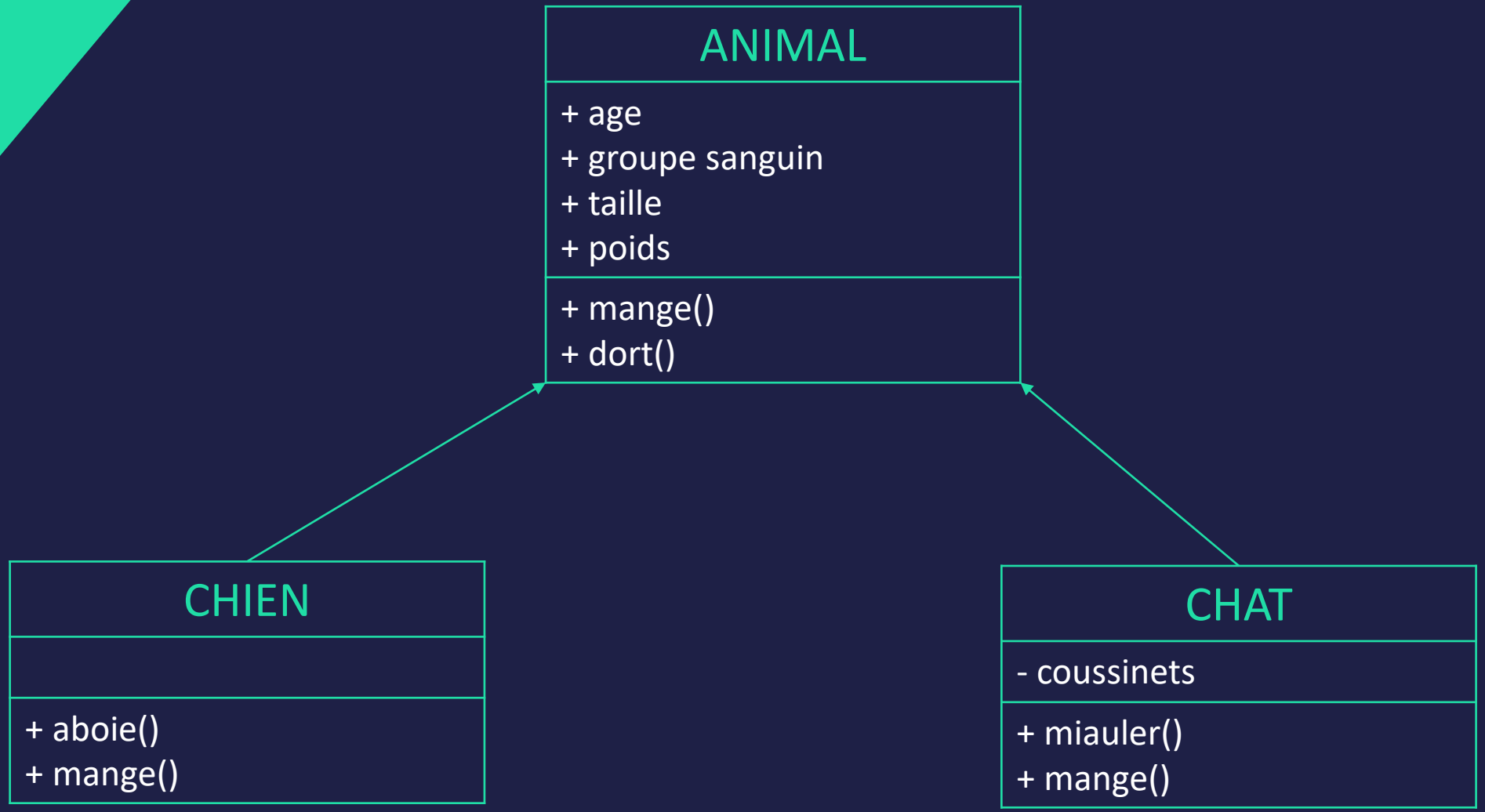
XEBIA NATIS

1. Les classes
2. L’intanciation
3. L’abstraction
4. L’héritage
5. Le polymorphisme
6. L’encapsulation

XEBIA NATIS

LE POLYMORPHISME

XEBIA NATIS


ANIMAL
+ age
+ groupe sanguin
+ taille
+ poids
+ mange()
+ dort()
CHIEN
+ aboie()
+ mange()
CHAT
- coussinets
+ miauler()
+ mange()

XWIKI NATIS

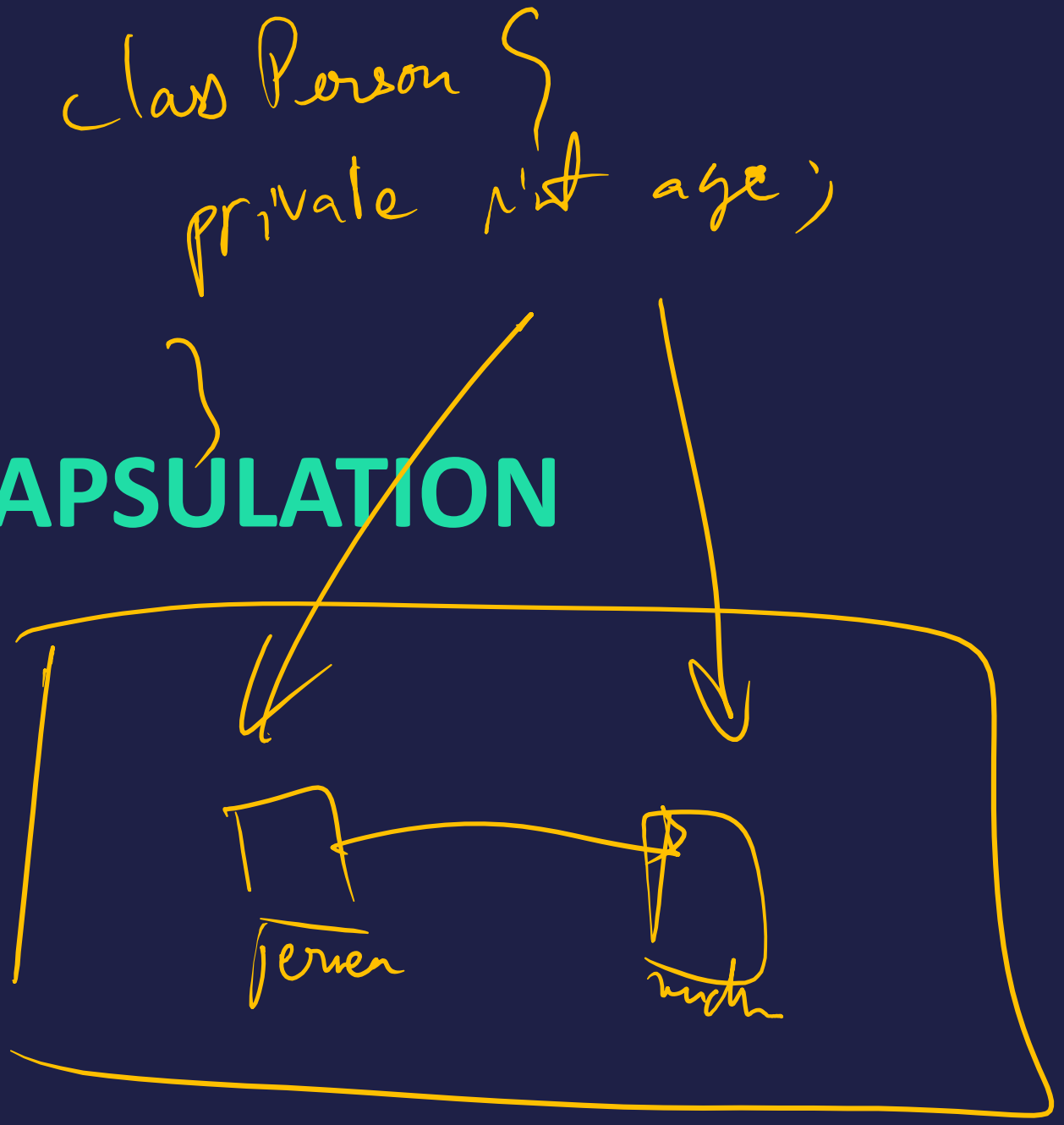
Le polymorphisme (de méthode) permet d'utiliser le même verbe pour décrire 2 méthodes différentes

XEBIA NATIS

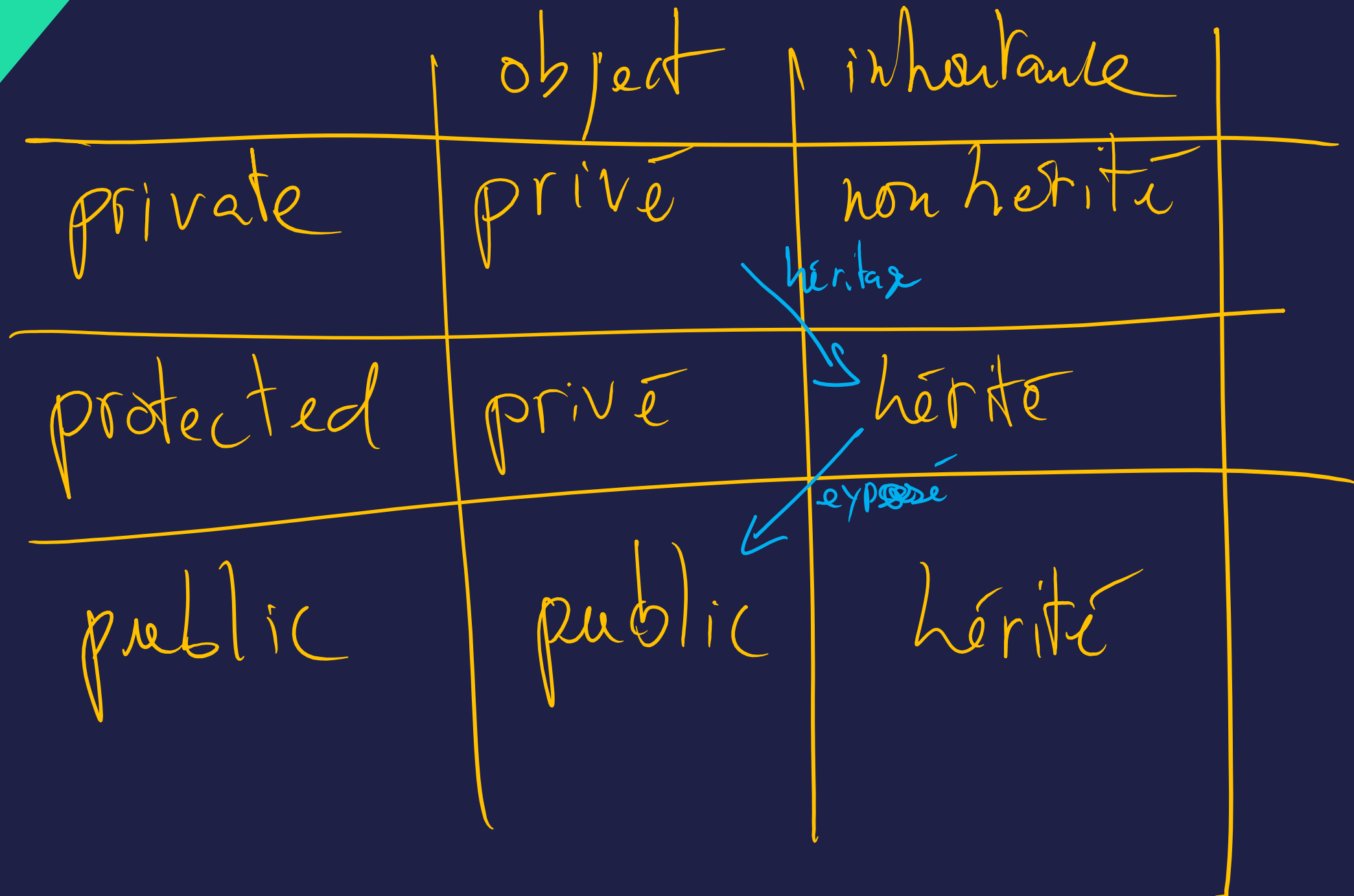
1. Les classes
2. L’intanciation
3. L’abstraction
4. L’héritage
5. Le polymorphisme
6. L’encapsulation

XEBIA NATIS

L'ENCAPSULATION



XDNATIS


object
inheritance
private
privé
non hérité
heritage
protected
privé
hérité
exposé
public
public
hérité

XWIKI SAS

XONATIS

XWIKI NATIS

L’encapsulation permet de définir une propriété ou une méthode interne

Public : accessible partout
Privé : interne à l‘objet et non hérité
Protégé : interne à l‘objet et hérité

XEBIA NATIS

4 PRINCIPES

ABSTRACTION

XWIKI NATIS

4 PRINCIPES

ABSTRACTION
HERITAGE

XKNATIS

4 PRINCIPES

ABSTRACTION
HERITAGE
POLYMORPHISME

XEBIA NATIS

4 PRINCIPES

ABSTRACTION
HERITAGE
POLYMORPHISME
ENCAPSULATION

XEBIA NATIS

4 PRINCIPES

ABSTRACTION
HERITAGE
POLYMORPHISME
ENCAPSULATION

XEBIA NATIS

4 PRINCIPES

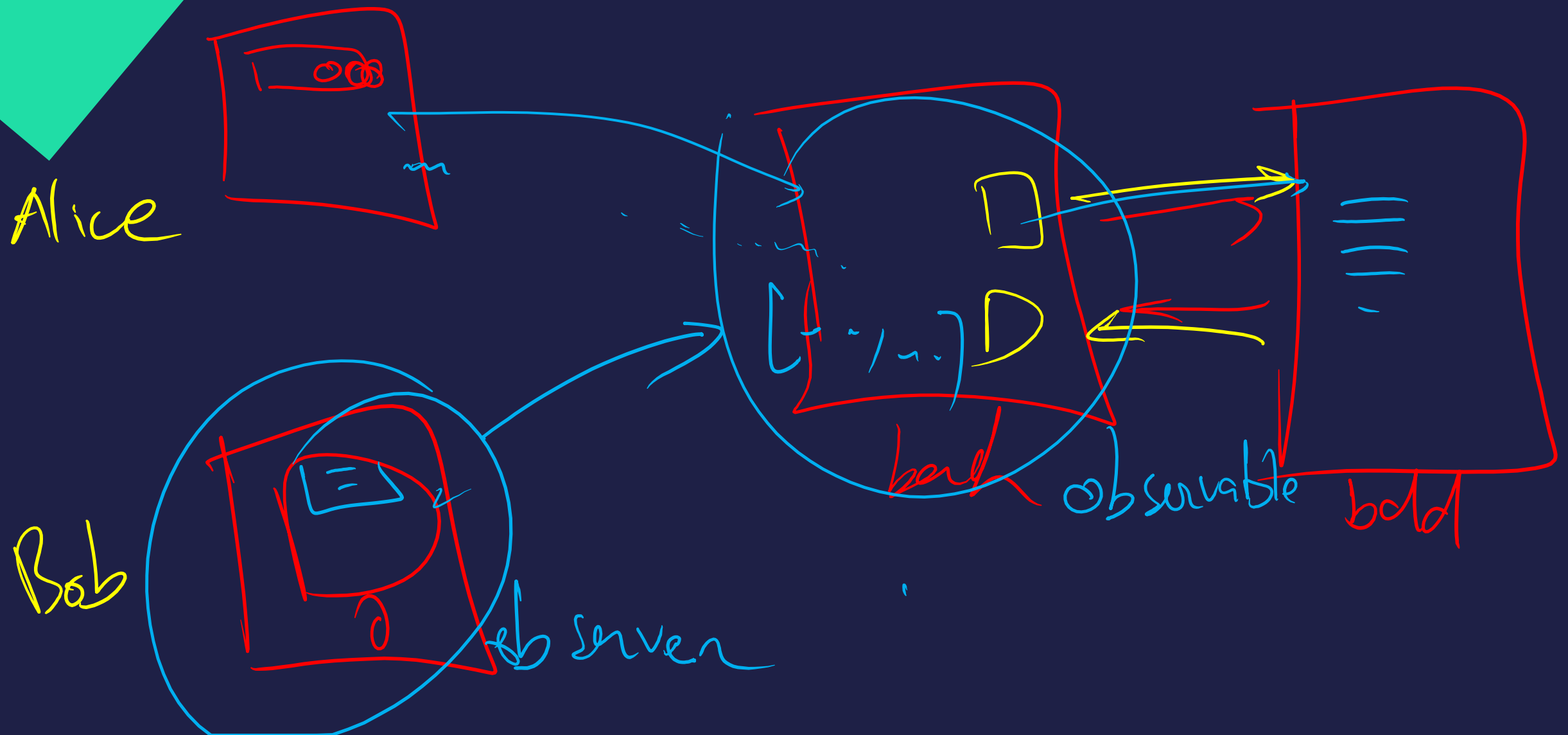
ABSTRACTION
HERITAGE
POLYMORPHISME
ENCAPSULATION

XWIKI SAS

XWIKI NATIS

DESIGN PATTERNS

XKNATIS



XONATIS

XWIKI NATIS

Un design pattern (patron de conception) représente une mécanique reconnue comme bonne pratique en réponse à un problème de conception.

Il décrit une solution standard et est issu de l'expérience des concepteurs.

XWIKI NATIS

https://refactoring.guru/fr/design-patterns/catalog

XWIKI NATIS

INVERSION DE CONTROLE

I.o.C

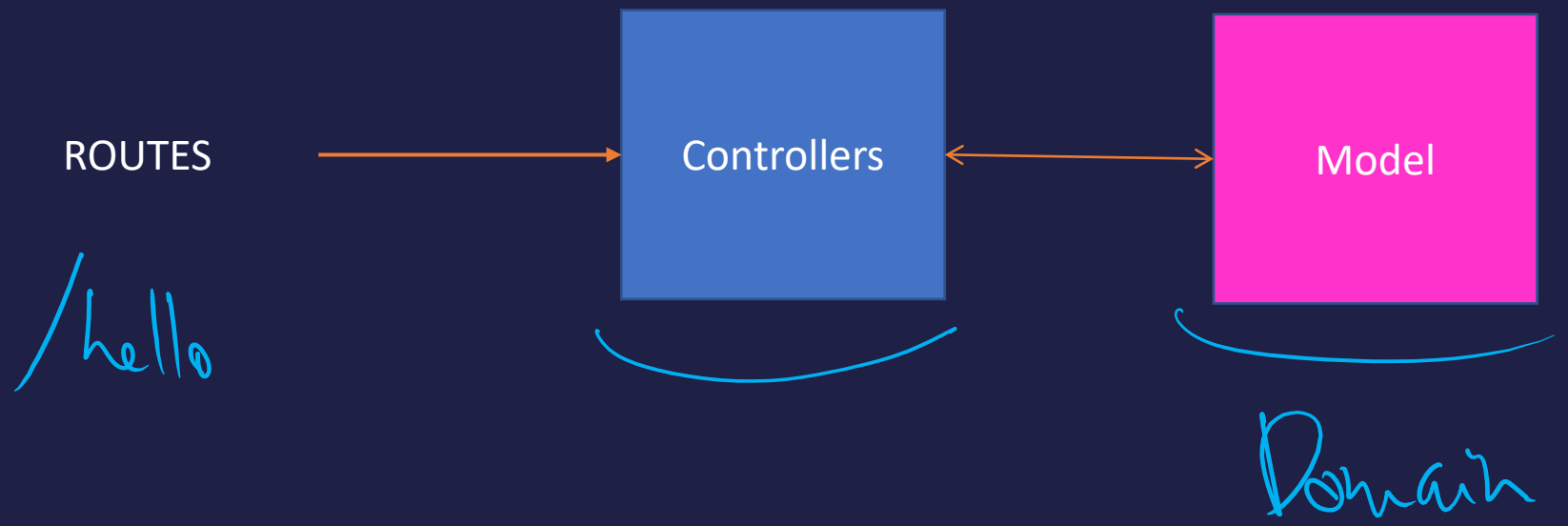
XONATIS

XWIKI SAS

XONATIS

XWIKI NATIS

Architecture générale



XDNATIS

Model = Entités

Controller = Le point d’entrée qui contient les fonctionnalités

XWIKI SAS

XWIKI SAS

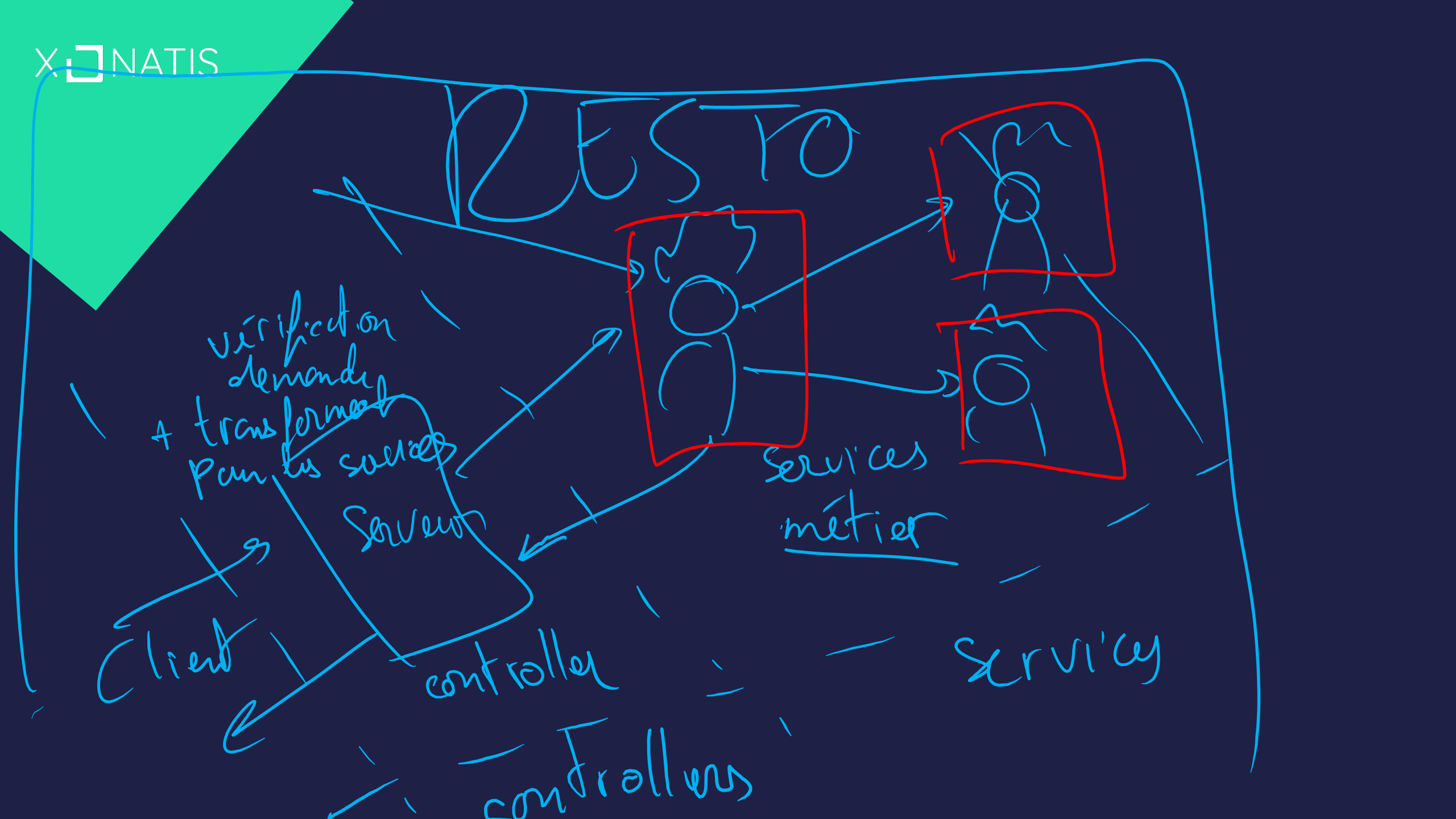
XWIKI NATIS

Les controllers peuvent devenir compliqué !

XWIKI NATIS

Les controllers peuvent devenir compliqué !

Isolons la plue-value des controllers dans des services !


XEBIA NATIS
RESTO
vérification demande + transformation pour les services
Serveur
services métier
Client
controller
service
controllers

XWIKI NATIS

UN SERVICE ?

XWIKI SAS

XWIKI NATIS

La couche de service est un modèle architectural, appliqué dans le paradigme de conception orientée service

XWIKI NATIS

Elle vise à rassemble la plus value de l'application en services. Cela permet de faciliter la complexité applicative en simplifiant les controlleurs.

XWIKI NATIS

Elle vise à rassemble la plus value de l'application en services. Cela permet de faciliter la complexité applicative en simplifiant les controlleurs.

XWIKI SAS

XEBIA NATIS

Architecture SOA

XWIKI NATIS

LE SOA

XWIKI NATIS

En génie logiciel, l'architecture orientée services (SOA) est un style architectural qui structure la production de la plus value d’une application en service unitaire.

XWIKI NATIS

Il est appliqué dans le domaine de la conception de logiciels où des services sont fournis aux autres composants par des composants d'application, par exemple via un protocole de communication sur un réseau.

XONATIS

XEBIA NATIS

LES MICRO-SERVICES

XWIKI NATIS

Une architecture de microservices - une variante du style structurel de l'architecture orientée services (SOA) - organise une application comme un ensemble de services faiblement couplés.

XWIKI NATIS

Dans une architecture de microservices, les services sont fins et les protocoles sont légers. L'objectif est que les équipes puissent faire vivre leurs services indépendamment des autres.

XWIKI NATIS

Le couplage lâche réduit tous les types de dépendances et les complexités qui les entourent, car les développeurs de services n'ont pas besoin de se soucier des utilisateurs du service, ils n'imposent pas leurs modifications aux utilisateurs du service.

XWIKI SAS

XEBIA NATIS

Avez-vous deja entendu parler d‘API ?

XWIKI NATIS

API est un acronyme anglais qui signifie Application Programming Interface, que l'on traduit par interface de programmation d'application.

Une API est un ensemble de définitions et de protocoles qui facilite la création et l'intégration de logiciels d'applications

XWIKI NATIS

Avez-vous deja entendu parler de REST API ?

XEBIA NATIS

Une REST API (également appelée RESTful API) est une interface de programmation d'application (API ou API web) qui respecte les contraintes du style d'architecture REST (Representational State Transfer)

Une REST API doit être sans état.

XEBIA NATIS

Les API REST doivent être sans état !

XWIKI SAS

XDNATIS

Que sont les méthodes de requetes HTTP ?

XONATIS

XWIKI NATIS

GET
POST
PUT
PATCH
DELETE

XWIKI NATIS

GET : Récupérer
POST : Créer
PUT : Remplacer
PATCH : Modifier
DELETE : Supprimer

XEBIA NATIS

Un endpoint est une extrémité d'un canal de communication. Lorsqu'une API interagit avec un autre système, les points de contact de cette communication sont considérés comme des points de terminaison (endpoint)

XWIKI NATIS

Pour les API, un endpoint est une URL d'un serveur ou d'un service.

Chaque endpoint est l'emplacement à partir duquel les API exécutent leur fonction.

XWIKI NATIS

Exemple !

XWIKI NATIS

Gestion d‘un CV-thèque

XWIKI NATIS

Gestion d‘une boulangerie

XWIKI NATIS

Gestion d‘un bookshop