



# Présentation d'un projet du CATI SICPA : Le projet GAniMed



# Introduction : Un manque de communication inter-SI

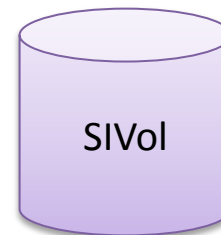
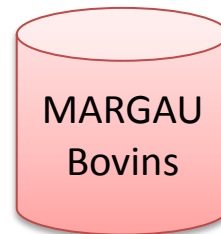
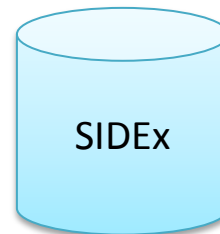
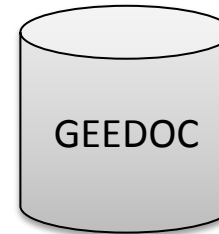
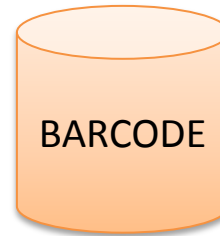
- ❖ Le département GA dispose d'une culture SI depuis de nombreuses années :
  - ❖ SI Métier :
    - GEEDOC (ovin/caprin)
    - GEEL (lapin)
    - Margau (bovin/porcin)
    - SIVol (volaille)
  - ❖ SI Support :
    - SIDEx (protocoles)
  
- ❖ 2010 > Constat d'un manque de communication entre les SI métiers et les SI supports. Proposition de projet pour combler ce manque de communication inter-SI.
  
- ❖ 2010 > Le projet est accepté mais mis en attente vu la priorité d'autres projets du moment. Le projet se nomme GAniMed (Génétique Animale Mediation)
  
- ❖ 2012 > Le projet devient stratégique pour le département GA et est donc relancé.

# SOMMAIRE

- I. Introduction : Constat d'un manque de communication inter-SI
- II. Expression et analyse du besoin
- III. Présentation du concept de l'interopérabilité applicative
- IV. Notre choix en matière d'interopérabilité
- V. Bref aperçu de l'architecture de PETALS ESB
- VI. L'avancement du projet GAniMed
- VII. Vision architecturale de GAniMed
- VIII. Les étapes en cours et à venir sur le projet GAniMed
- IX. Les perspectives du projet GAniMed
- X. Conclusion : Vers une architecture agile et communicante

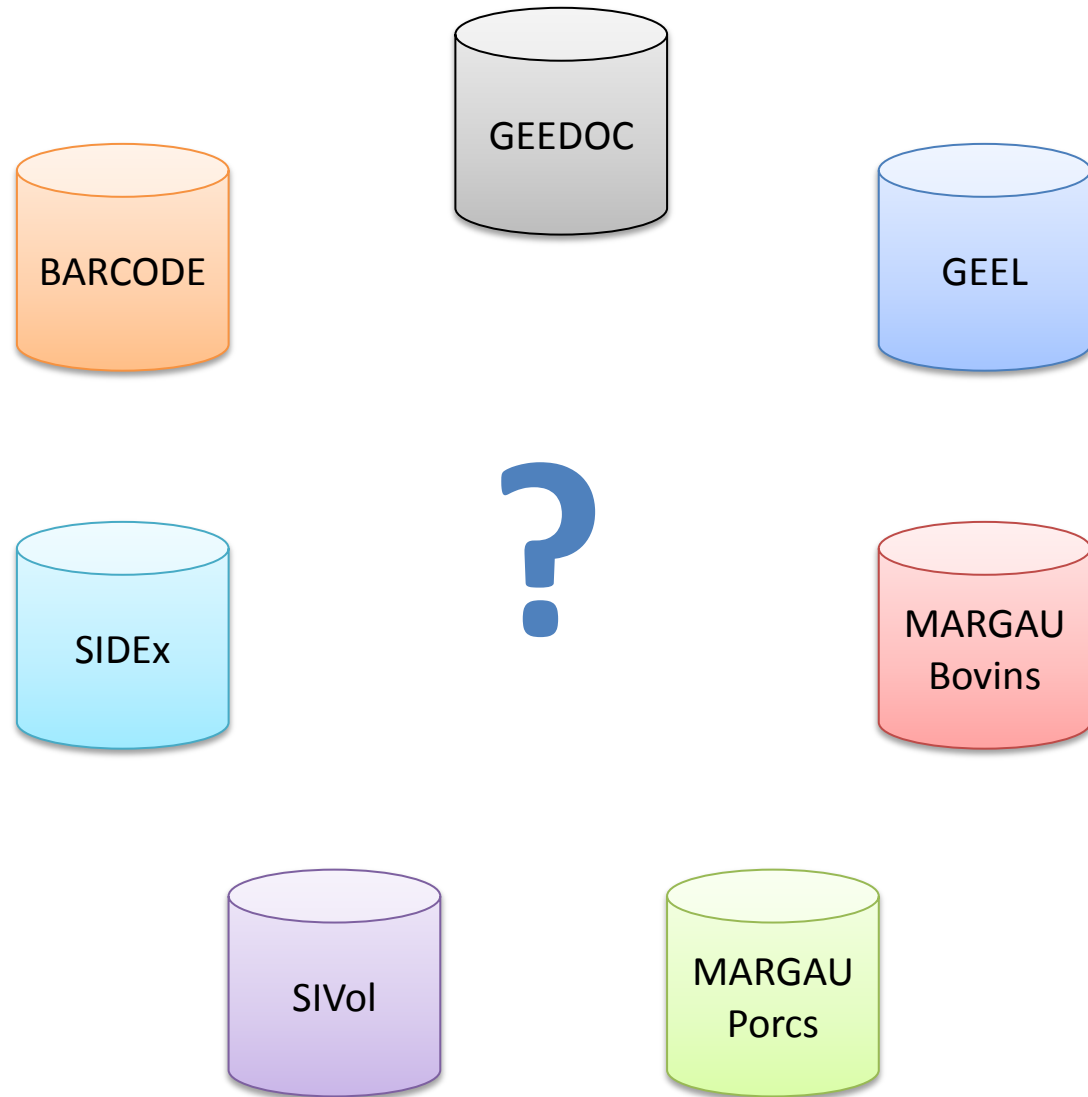
# Expression et analyse du besoin

- ❖ Constat : fonctionnement des SI en silos de données == manque de communication inter SI



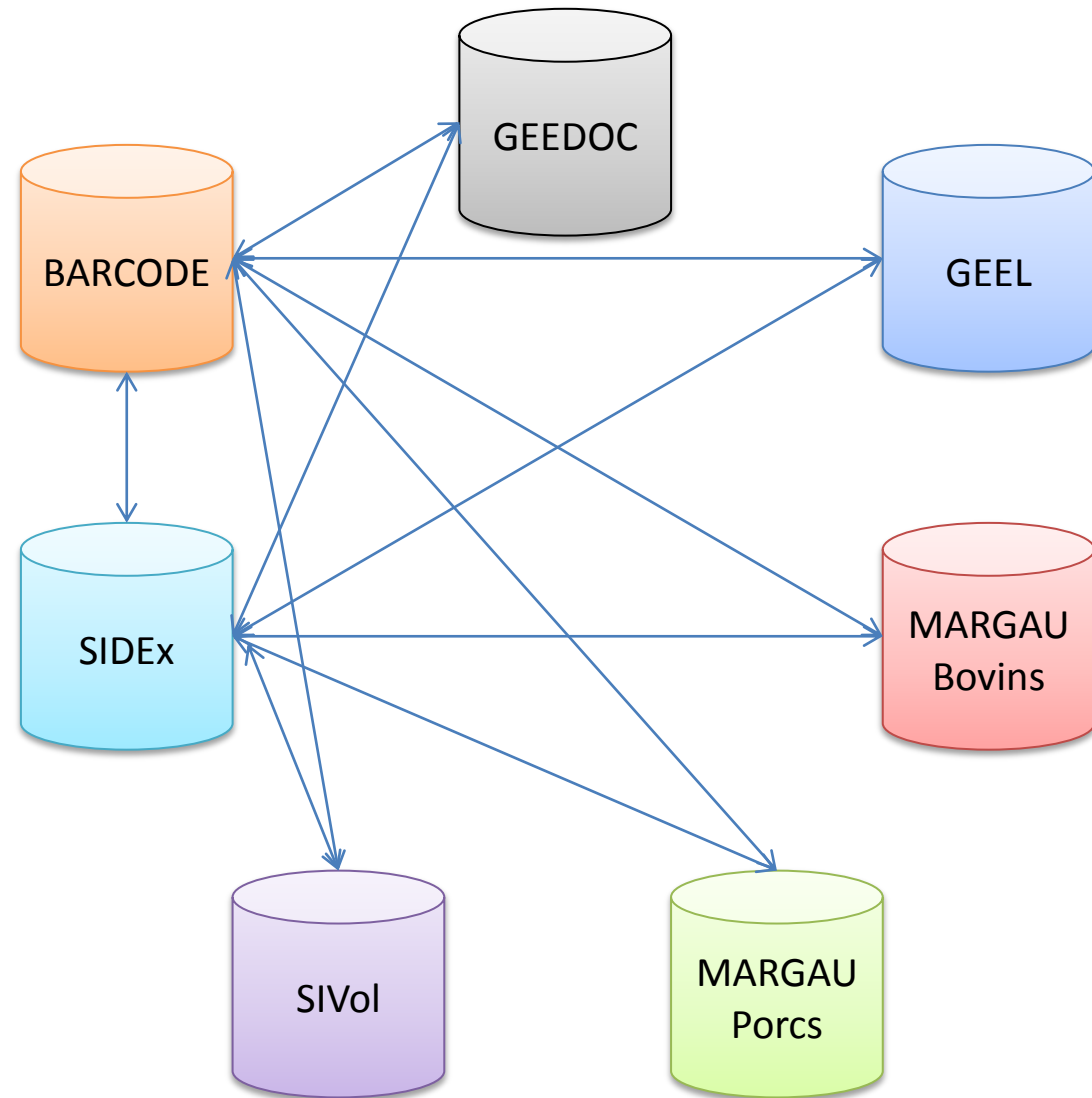
# Expression et analyse du besoin

- ❖ Constat : fonctionnement des SI en silos de données == manque de communication inter SI
- ❖ Besoin : mettre en place une solution qui permette aux SI d'interagir



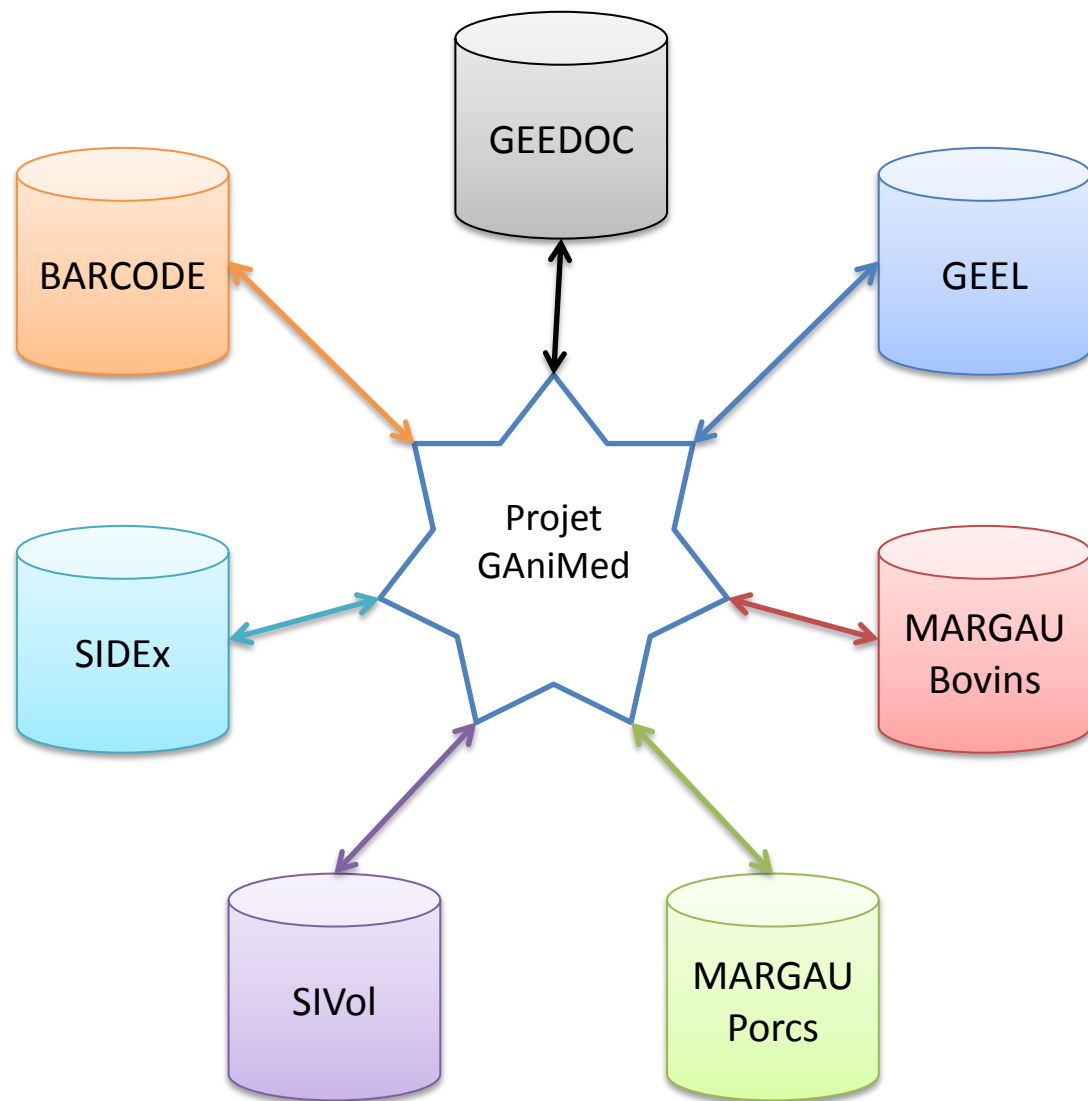
# Expression et analyse du besoin

- ❖ Constat : fonctionnement des SI en silos de données == manque de communication inter SI
- ❖ Besoin : mettre en place une solution qui permette aux SI d'interagir
- ❖ La phase d'analyse du besoin nous a permis d'étudier diverses solutions, parmi lesquelles :
  - mode point à point
  - mode intergiciel



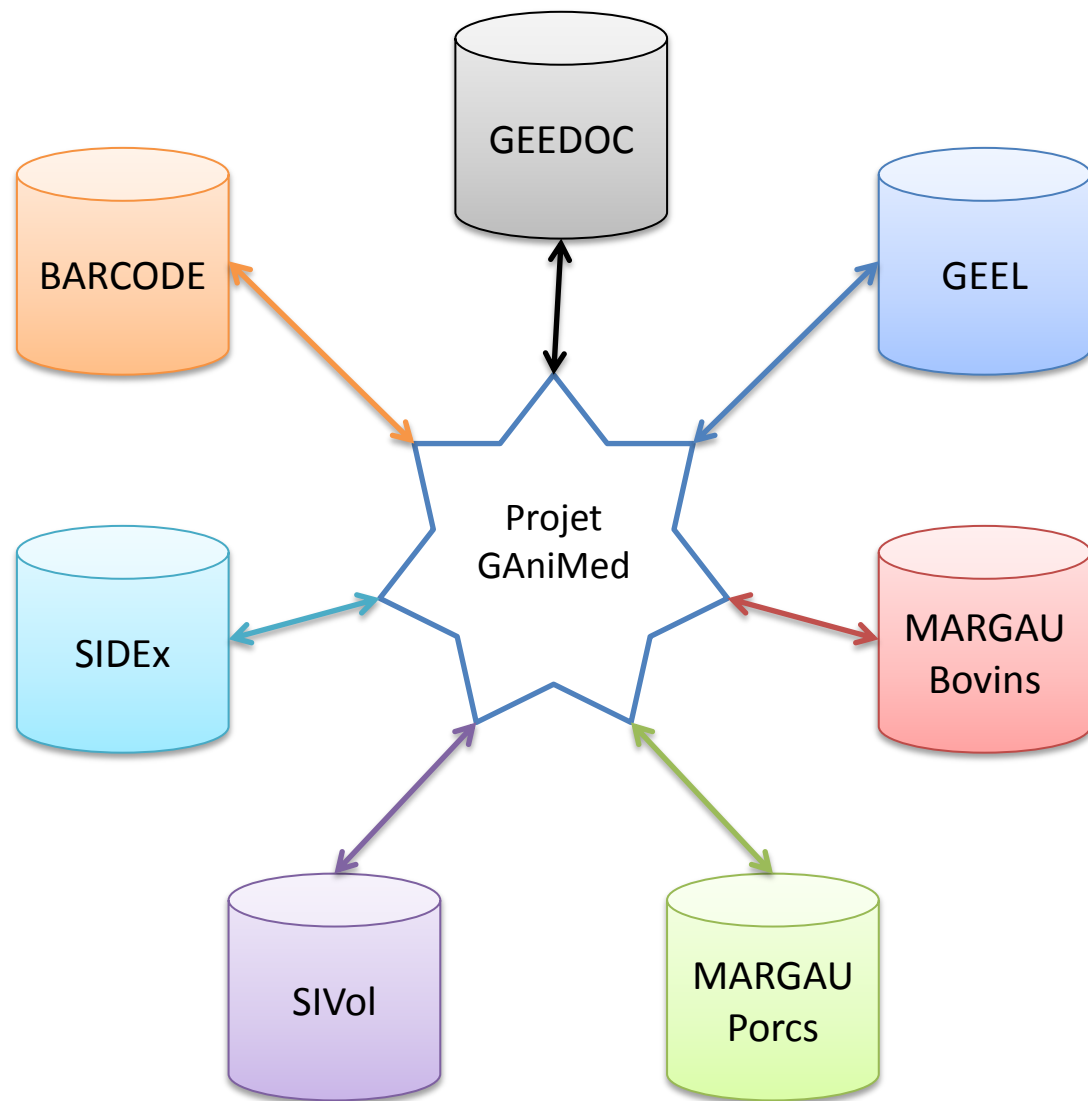
# Expression et analyse du besoin

- ❖ Constat : fonctionnement des SI en silos de données == manque de communication inter SI
- ❖ Besoin : mettre en place une solution qui permette aux SI d'interagir
- ❖ La phase d'analyse du besoin nous a permis d'étudier diverses solutions, parmi lesquelles :
  - mode point à point
  - **mode intergiciel**



# Expression et analyse du besoin

- ❖ Constat : fonctionnement des SI en silos de données == manque de communication inter SI
- ❖ Besoin : mettre en place une solution qui permette aux SI d'interagir
- ❖ La phase d'analyse du besoin nous a permis d'étudier diverses solutions, parmi lesquelles :
  - mode point à point
  - mode intergiciel
- ❖ La solution intergicelle permet :
  - une meilleure agilité de la solution
  - une maintenance simplifiée de la solution
  - Le routage des messages
  - La sécurité des messages





# Présentation du concept de l'interopérabilité applicative



## L'interopérabilité, qu'est-ce que c'est ?

L'interopérabilité est la capacité que possède un SI, dont les interfaces sont intégralement connues, à fonctionner avec d'autres SI, existants ou futurs, et ce sans restriction d'accès ou de mise en œuvre.

L'interopérabilité applicative se sert d'une couche logicielle pour mettre en place les communications entre SI. On parle alors de « *middleware* » (ou « *intergiciel* »)

## Les formes d'interopérabilité applicative :

- ***interopérabilité par flot de données*** : les données sont échangées par lots à des moments prédéfinis (ex: nuit, weekend, ...)
- ***interopérabilité au fil de l'eau*** : les données sont échangées au moment où l'on en a besoin (ex: clic de la souris, événement temporel déclenchant un traitement, ...)

# Notre choix en matière d'interopérabilité

Au niveau des SI clients, les utilisateurs ont un besoin de l'information en temps réel. Naturellement donc, notre choix s'est orienté vers une solution « fil de l'eau »

## Choix de l'architecture : **SOA** (Service Oriented Architecture)

- Modularité de l'architecture
- Réutilisabilité des composants (agilité)
- Meilleures possibilités d'évolutions (agilité)
- Plus grande tolérance aux pannes (maintenabilité)
- Maintenance facilitée (maintenabilité)

## Choix de la technologie d'interopérabilité: **ESB** (Enterprise Service Bus)

- Architecture SOA (il reprend donc les avantages de la SOA cités ci-avant)
- Standardisation des concepts (XML, JMS, ...)
- Intelligence du routage des messages
- Architecture distribuée, scalable et fiable

## Choix du produit d'interopérabilité : **Petals ESB** (Société : PetalsLink)

- Parmi de nombreuses solutions testées (Apache ServiceMix, MuleSoft ESB, JBoss ESB, ...), Petals ESB est celui qui répond le mieux à nos attentes (simple, léger, souple, support professionnel, ...)

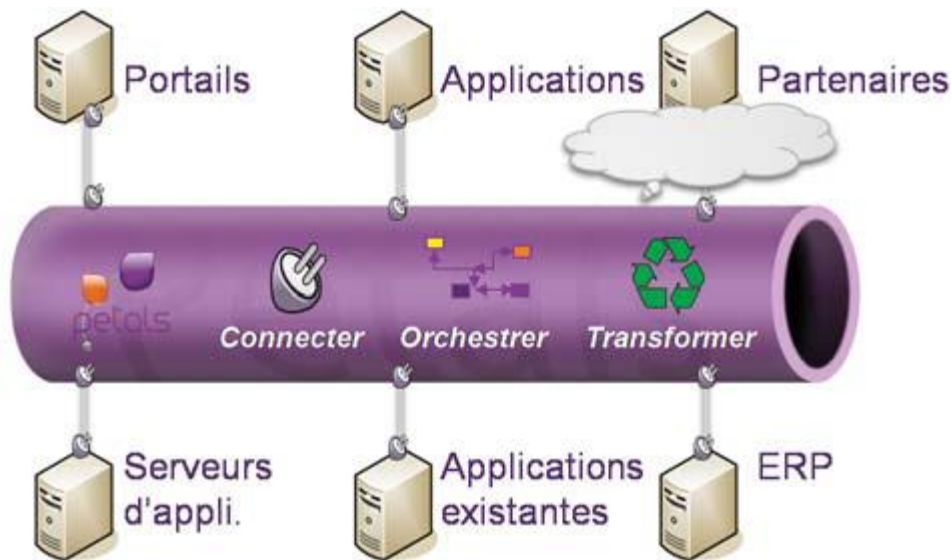
# Bref aperçu de l'architecture de PETALS ESB



❖ Petals ESB est un **bus de services d'entreprise** open source, qui effectue la **médiation**, le **routage** et l'**intégration de services** d'une infrastructure de service (SOA).

## ❖ Architecture

- Léger, modulaire et sécurisé
- Nativement distribué, pour faciliter la mise à l'échelle et la robustesse
- Basé sur des standards reconnus, pour rester libre de ses choix logiciels
- Open source et abordable, avec support professionnel



# L'avancement du projet GAniMed

## Phase d'études (TH)

- Identification et analyse du besoin
- Etude de toutes les solutions disponibles, puis étude approfondie de l'interopérabilité
- Choix d'une technologie d'interopérabilité et recherche du progiciel le mieux adapté

## Phase de test du produit (TH)

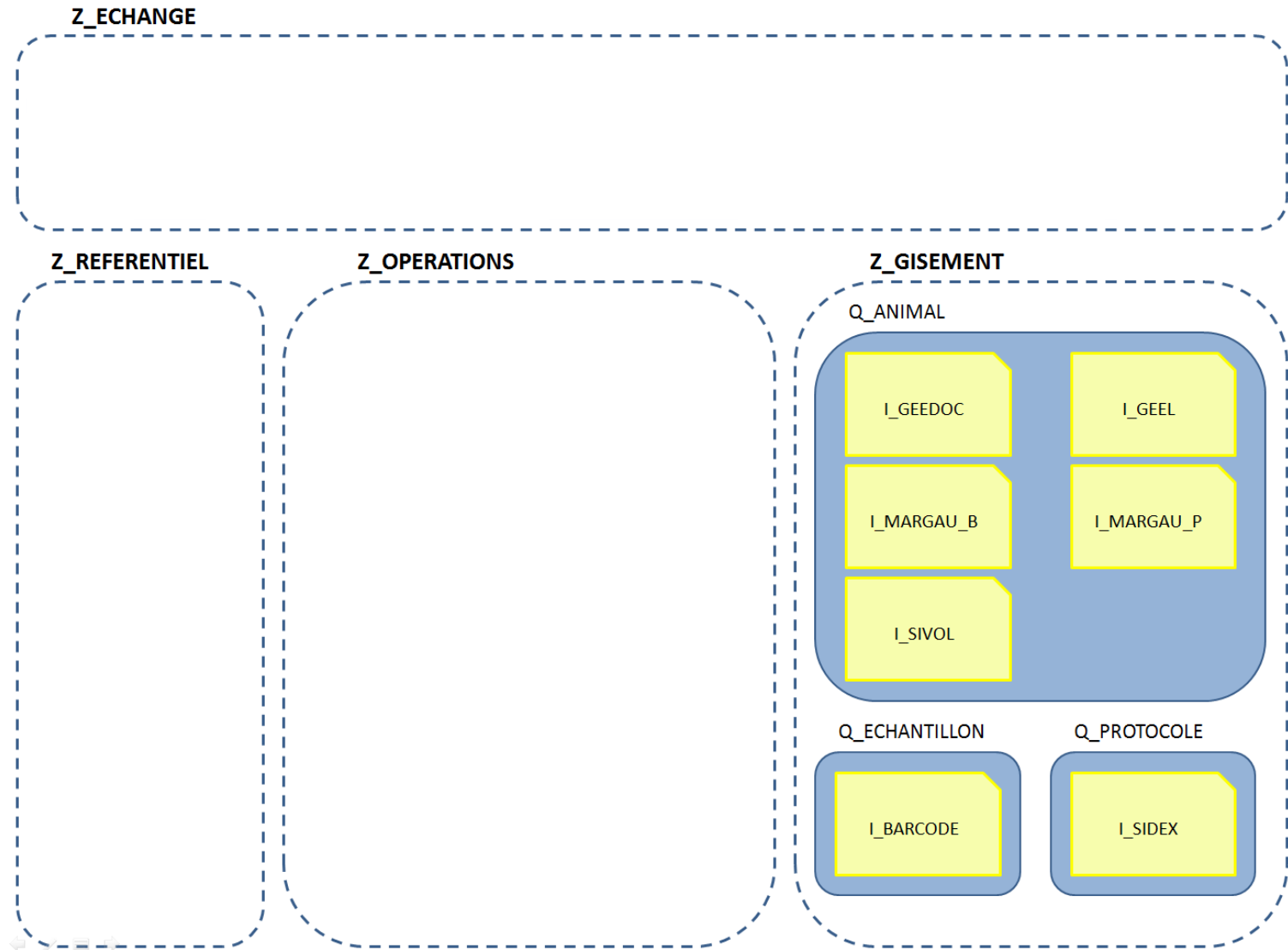
- Prise en main et autoformation à Petals ESB
- Développement / déploiement de webservices sur un serveur d'applications J2EE
- Mappage des services sur le bus Petals ESB
- Test d'interrogation des webservices via le bus Petals ESB

## Phase de mise en place de la première machine de production (au CTIG)

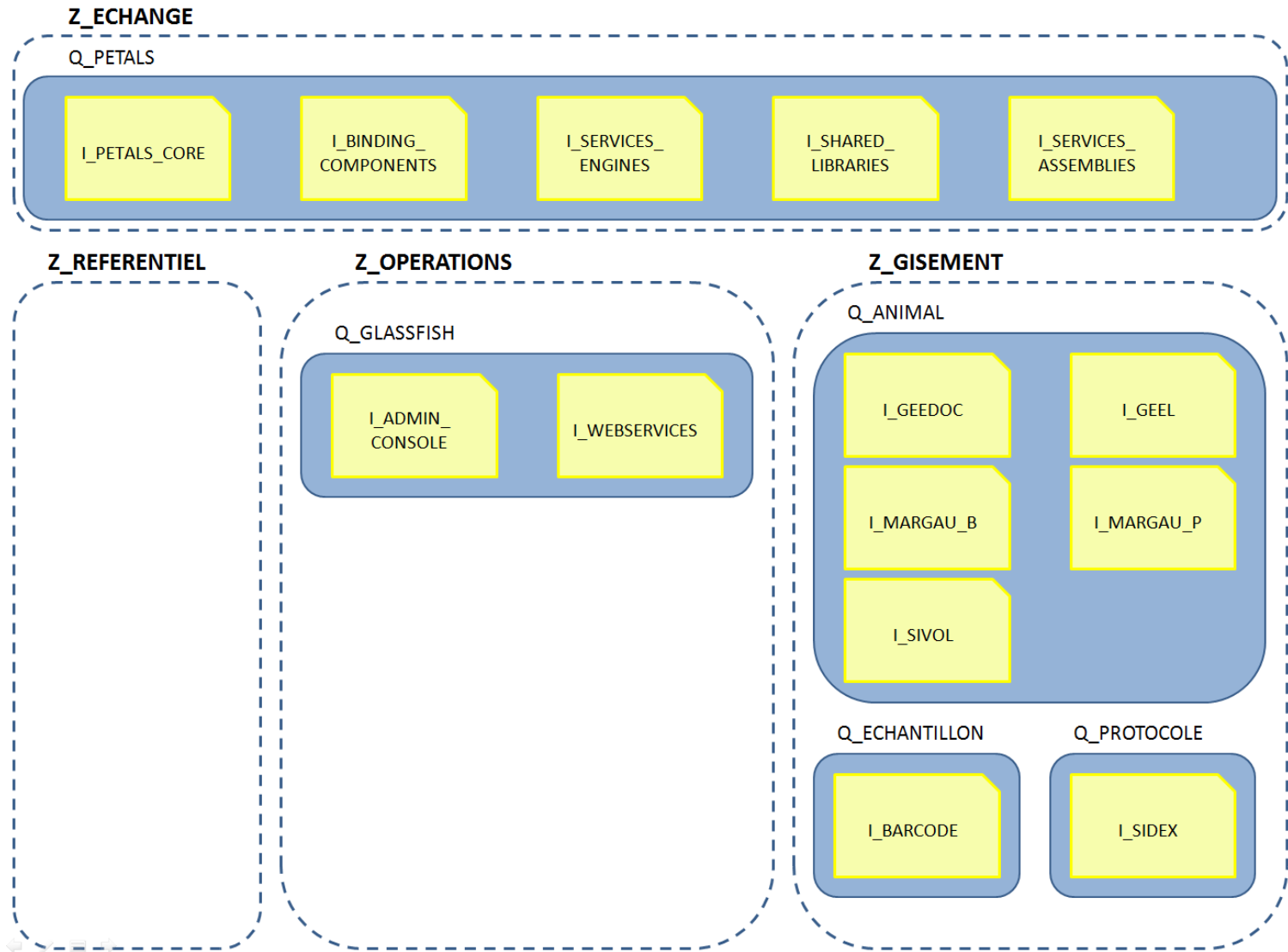
- Environnement Debian incluant un serveur Glassfish et Petals ESB (TH)
- Tests de déploiement depuis Jouy (TH) et depuis Toulouse (AJ)

Formation interne à Java, aux webservices et au bus Petals ESB  
(AJ / TH)

# Vision architecturale du projet GAniMed



# Vision architecturale du projet GAniMed



# Les perspectives du projet GAniMed

Le périmètre actuel du projet GAniMed concerne 2 SI support et 5 SI métier:

- Support : Barcode, SIDEx
- Métier : GEEDOC, GEEL, Margau B, Margau P, SIVol

Perspectives :

- A terme, lorsque le projet fonctionnera bien avec ces 7 SI, l'idée est d'intégrer au fur et à mesure d'autres SI clients au projet (CASAME, Alimentation, ...) Ce genre d'approche est facilité grâce à l'architecture SOA de notre solution (concept d'*agilité* de la solution)
- En fonction du nombre de SI client et de la charge des nœuds, nous envisagerons d'ajouter d'autres nœuds afin de toujours disposer d'un maillage efficace pour la solution.

# Conclusion : Vers une architecture agile et communicante



Besoin : interconnecter des SI hétérogènes entre eux afin d'en tirer une réelle plus value (plus-value au sens Porter du terme)

Réponse au besoin :

- Enterprise Service Bus (ESB) comme technologie d'intégration
- Petals ESB comme produit d'intégration

Perspectives :

- Intégrer de nouveaux SI clients au fil de l'eau (démarche itérative et agile)
- Eventuellement, en fonction des demandes et de la charge, ajouter de nouveaux nœuds afin d'assurer stabilité, fiabilité et scalabilité de la solution