

2^{ème} année Master Informatique
Option: **Génie Logiciel et Systèmes Distribués**
Université de Biskra
Année 2021/2022

Architectures Orientées Services

Dr. Mohamed Lamine KERDOUDI
Email : I.kerdoudi@univ-biskra.dz

1. Introduction

Dans ce module on va essayer de répondre à ces questions :

- A quels besoins répond le SOA ?
- Pourquoi les solutions actuelles sont insuffisantes ?
- Quels sont les principes de base du SOA ?
- Quels sont les éléments clé d'une architecture orientée services ?
- Quel est le cycle de vie d'un service ?
- Quelles méthodes et outils permettent de mettre en œuvre une architecture orientée services ?

1. Introduction

- ❑ Quand vous développez une application en entreprise, il est rare que vous commenciez dans un système vierge à partir d'une page blanche et encore plus que vous soyez le seul à travailler dessus.
- ❑ Vous arrivez donc en général dans un système composé de dizaines voire de centaines d'applications, qui communiquent entre eux.
- ❑ Il peuvent accéder des serveurs locaux, dans des serveurs distant. Elles sont aussi écrites dans divers langages : Java, C#, C++, Python ou pire encore COBOL.

1. Introduction

Se posent alors plusieurs **problèmes** :

❑ Dans le SI (système d'information) d'une entreprise, chaque application peut avoir son propre protocole de communication, ses propres formats de messages,...

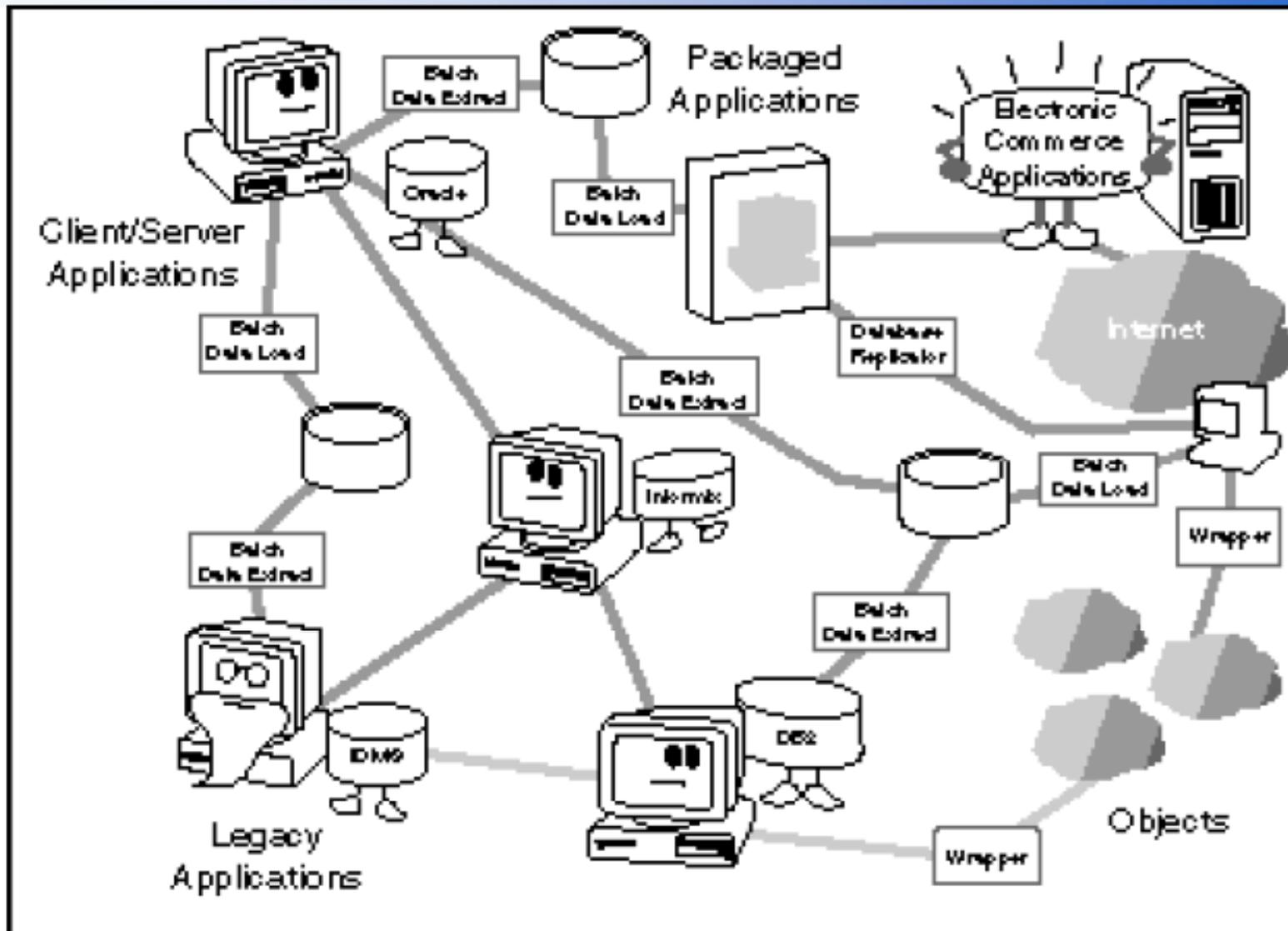
➔ Se pose alors la question de comment créer votre application de façon à ce qu'elle **communique parfaitement avec les autres** ? Ou encore mieux, comment concevoir toutes les applications du SI dès le départ pour qu'elles communiquent toutes via un protocole et **un format standard** et compris de tous ?

❑ ❑ Comment **éviter de réinventer le tout** : savoir si certaines fonctionnalités existent déjà dans d'autres applications ou chercher la possibilité de réutiliser certains composants afin d'éviter la redondance.

❑ Comment savoir à quelle URL ou IP se situe telle ou telle application avec laquelle vous voulez communiquer ? Pire encore, que se passe-t-il si cette application **change d'URL** ?

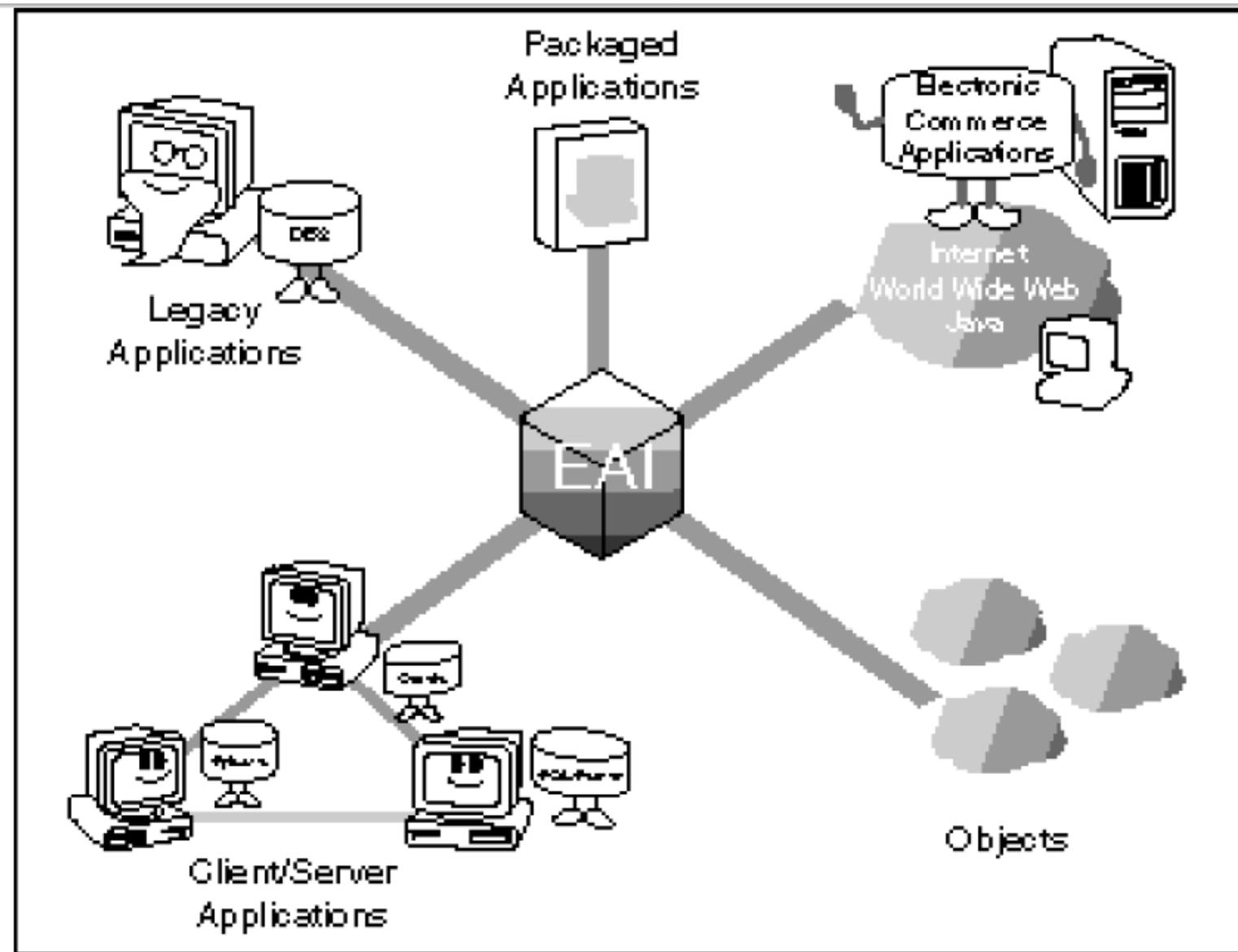
1. Introduction

Motivation de AOS : Une architecture de SI d'entreprise



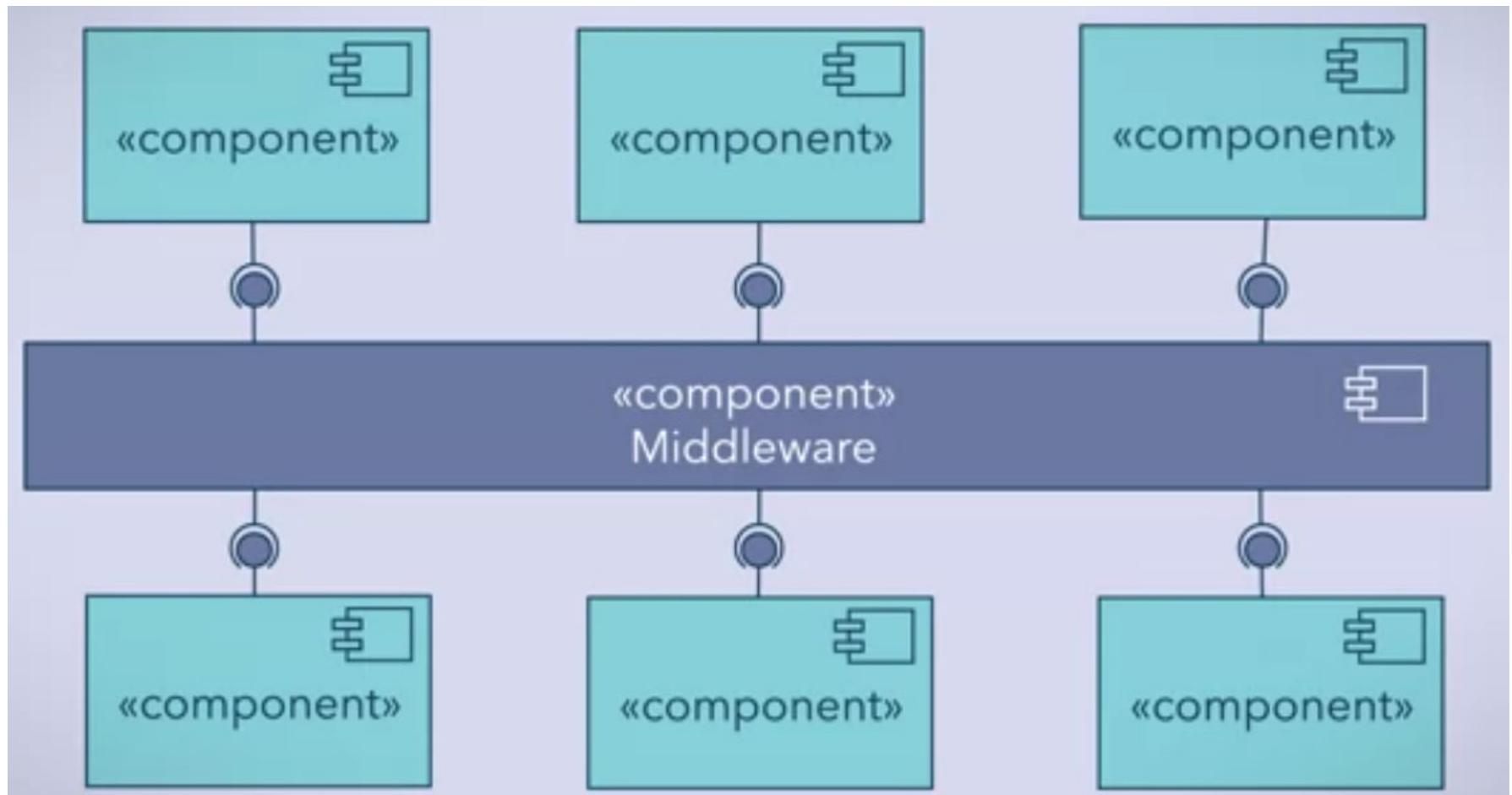
1. Introduction

Enterprise Application Integration



1. Introduction

Enterprise Application Integration



1. Introduction

Motivations

❑ Intra entreprise:

- Evolution des SI par l'intégration des applications

❑ Business-to-Business

- Externalisation de son SI
- Répondre rapidement à un nouveau business, une nouvelle attente du marché

❑ Objectifs de l'EAI

- Utilise une forme de middleware.
- Middleware est un logiciel qui se trouve entre d'autres logiciels et facilite les communications entre eux.
- Cela aide à masquer et maîtriser la complexité des technologies

1. Introduction

Problèmes de EAI :

- Il n'est pas clair comment cela est implémenté dans les interactions entre entreprises ou B2B
- En particulier, quelle entreprise implémente le middleware ?
- Est-il hébergé par un tiers ?
- Comment la sécurité est-elle gérée ?
- Comment protéger les données d'une entreprise contre une influence extérieure ?
- En pratique, EAI n'est pas utilisée dans les interactions B2B.

→ Au lieu de cela, les interactions entre les entreprises prises sont mieux mises en œuvre comme **services Web**.

1. Introduction

❑ L'architecture orientée services (SOA) est une réponse efficace aux problèmes rencontrés par les organisations afin d'**améliorer la flexibilité** et **réduire le coût de maintenance** de leurs processus métiers.

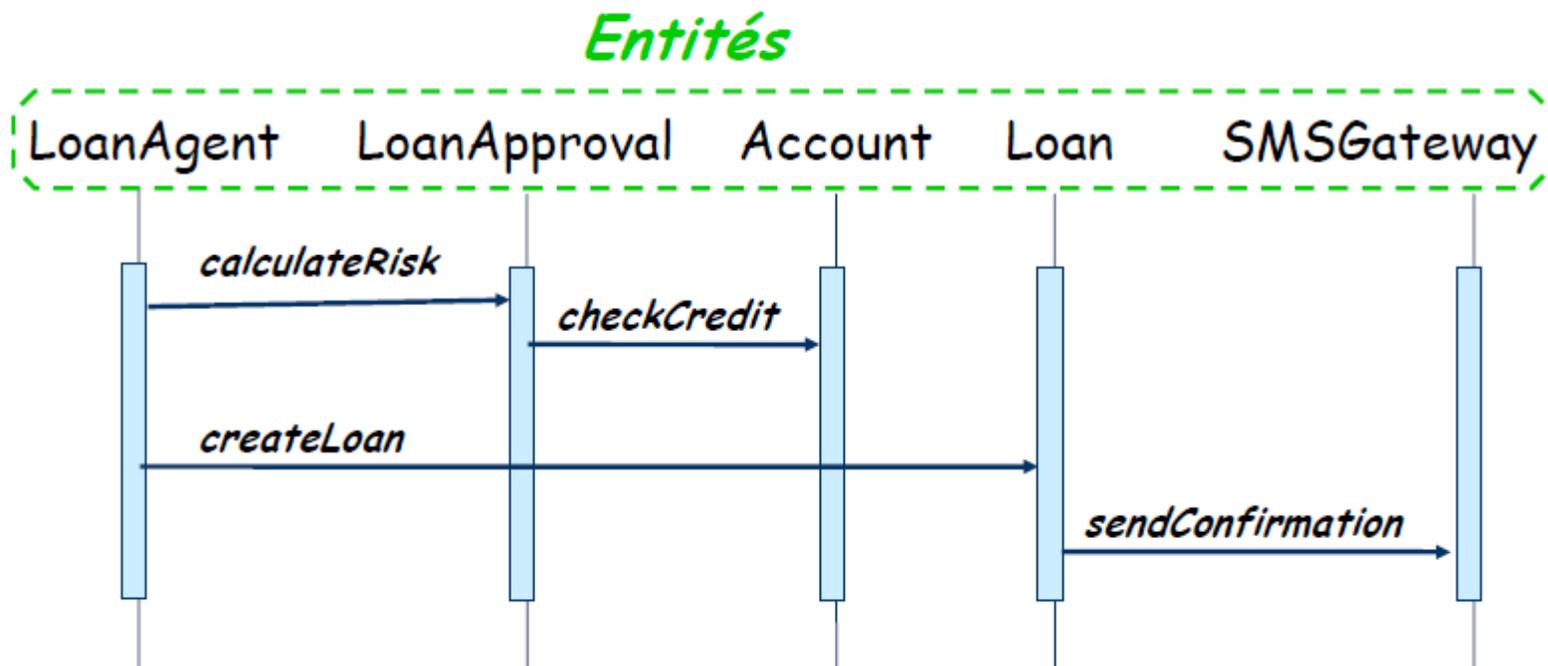
❑ SOA est un **concept** et une **approche** pour développer des architectures distribuées centrées sur la notion de **relation de service** entre les applications et la formalisation de cette relation dans un **contrat**.

But des services Web

- ❖ Fournir une architecture générale pour les applications réparties sur internet :
 - Inter-opérables :
 - Basé sur des standards ouverts
 - Sans composant spécifique à un langage ou un système d'exploitation
 - Faiblement couplées :
 - Limiter au maximum les contraintes imposées sur le modèle de programmation des différents éléments de l'application

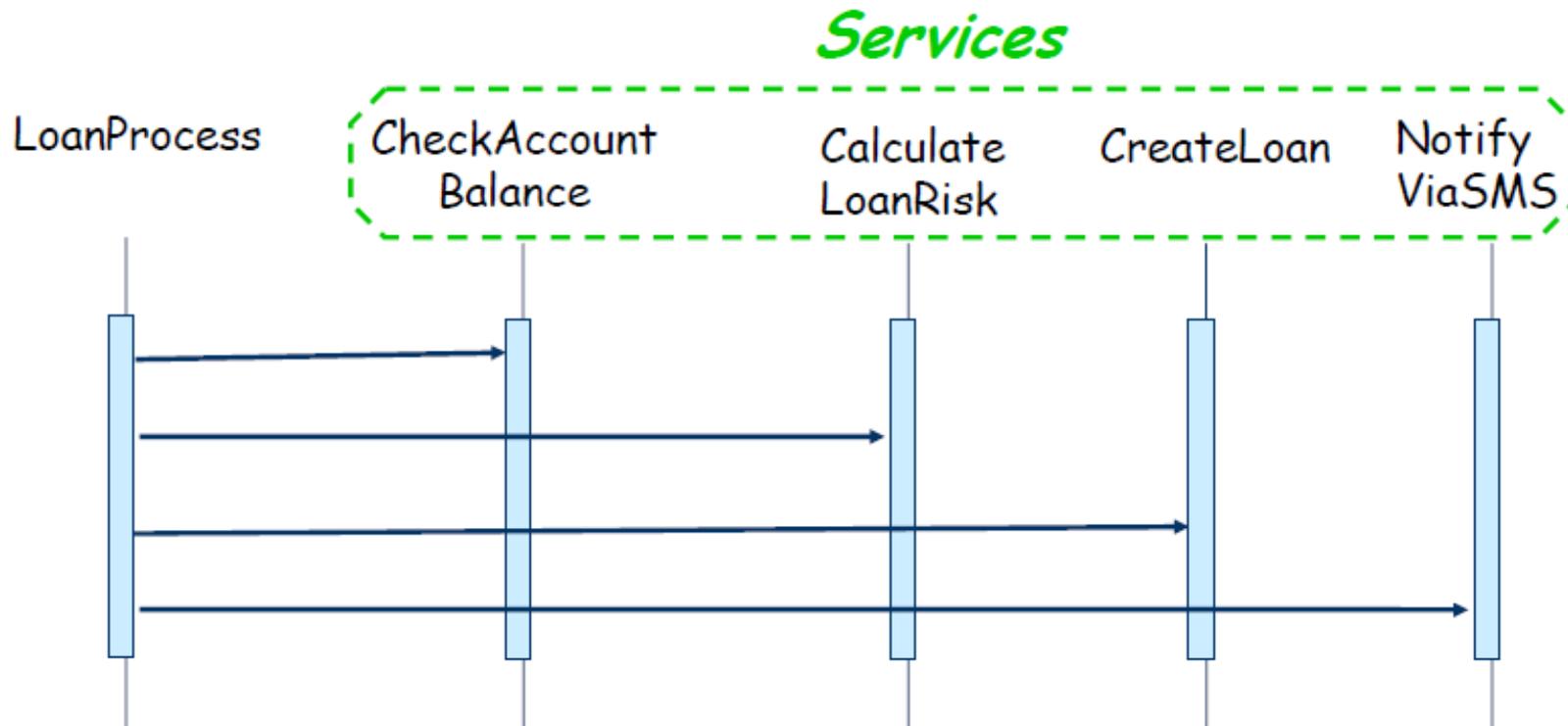
1. Introduction

Exemple de couplage fort : Gestion de prêts



- LoanAgent est lié à LoanApproval et Loan
- LoanApproval est lié à Account
- Loan est lié à SMSGateway

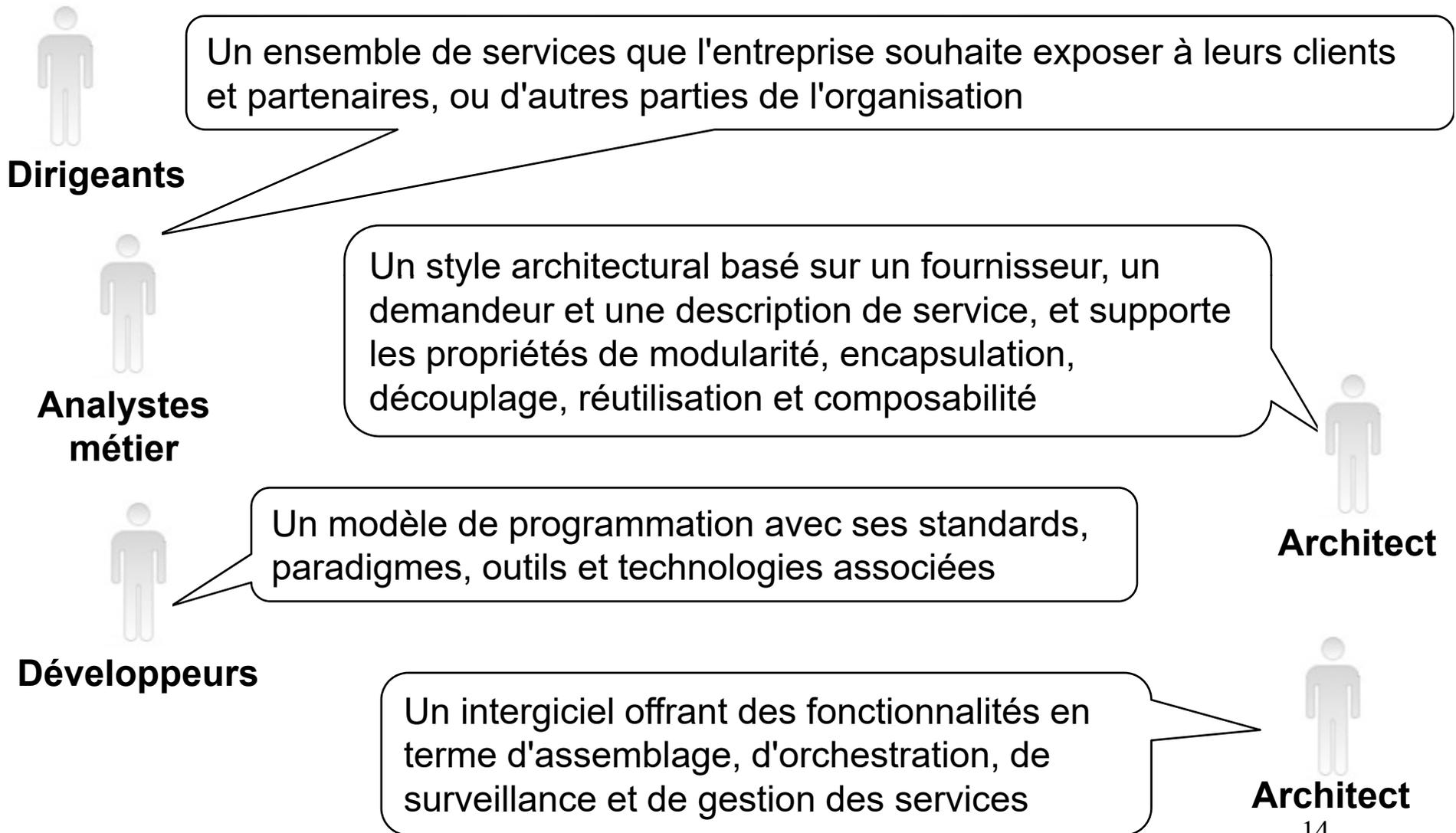
1. Introduction



- Qu'est ce que LoanProcess ?
- Un processus métier !
Il permet d'orchestrer les services => couplage lâche

1. Introduction

- ❑ Chaque rôle s'approprie SOA différemment





Questions ??