IrisaTech

AOP et industrialisation de la production d'applications

Didier GIRARD

Directeur Technique d'IMPROVE

IMPROVE Santé : Éditeur

- Résurgences depuis 1996, 15 Hôpitaux
- Interconnexion d'hôpitaux depuis 2003
- SMUR depuis 2004
- 10% du CA

IMPROVE Services : SSII

- Depuis 1991 (NeXT)
- Expertise technologique
- Architecture et modernisation SI
- Réalisation de projets critiques (délais – qualité)
- Transfert de compétences auprès d'équipes internes
- Contrôle de la qualité des réalisations sous-traitées
- 50% du CA
  - RIA, RDA, IAD

IMPROVE Foundations : Éditeur

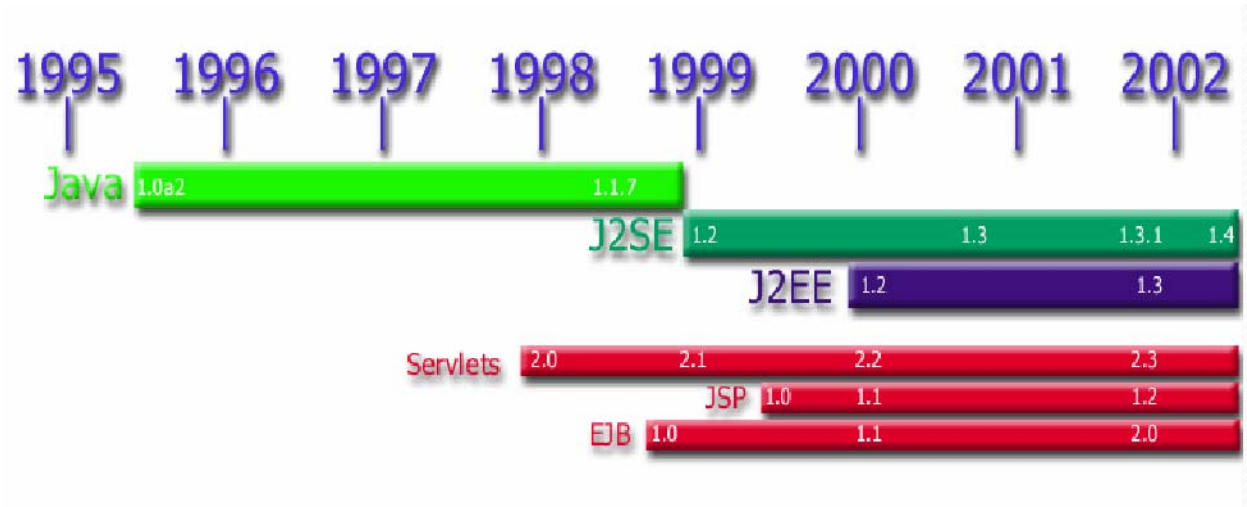
- Depuis 2002
  - Europe Assistance, AGF, Galerie Lafayette, BHV, MGEN, MFP,...
- Industrialisation des développements J2EE
- Licences, Services, Formation
- 40% du CA

Les constats

- Les projets fonctionnels sont des projets technologiques
- Coûts de réalisations non maîtrisés
- Coûts de maintenance importants
- Exploitation difficile
- Besoin d'expertise dans la durée
- Le rythme technologique condamne à l'obsolescence
- Les spécifications fonctionnelles ne prennent pas en compte les coûts de production
- La MOA et la MOE ne se comprennent pas

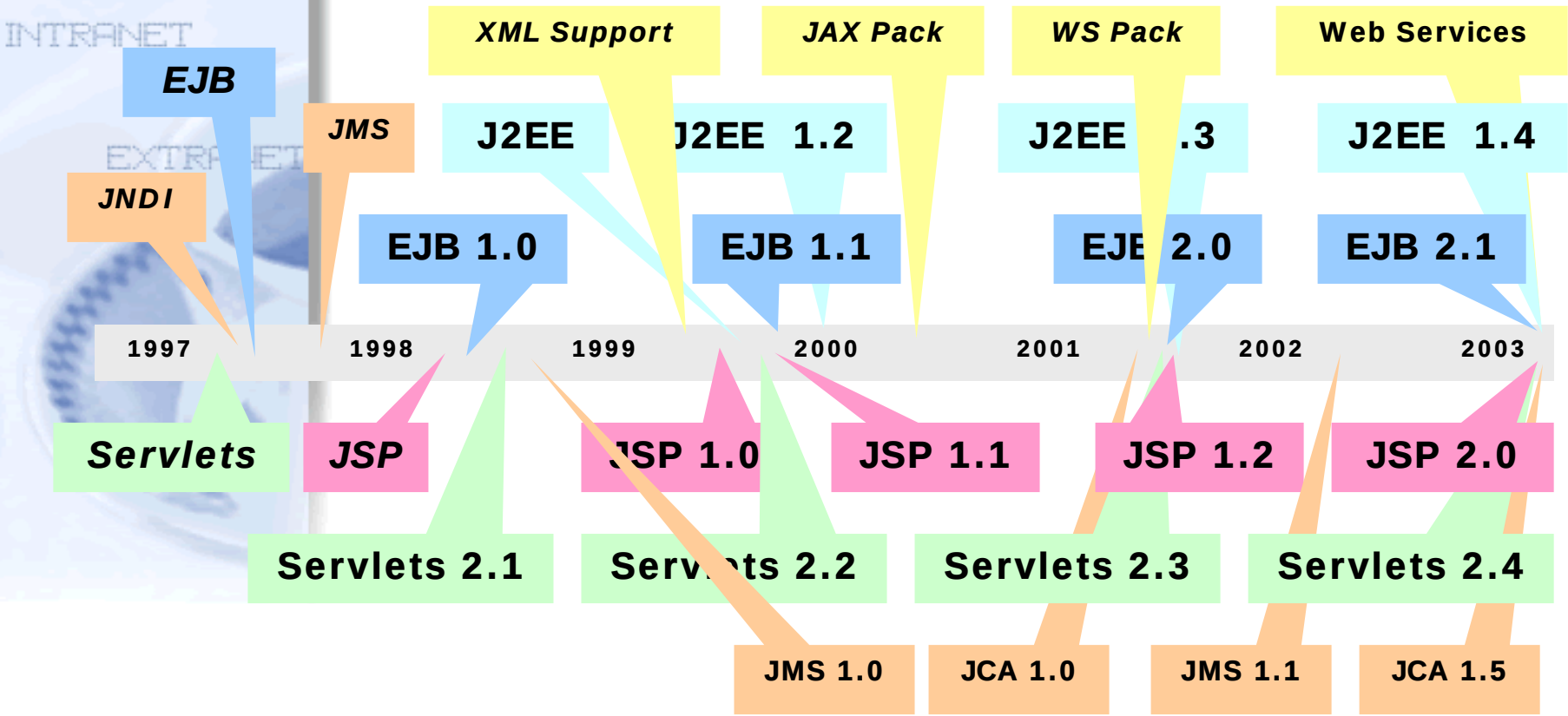
J2EE : Une évolution permanente

- La technologie Java est en perpétuelle évolution
- Les développeurs ont du mal à garder des connaissances à jour
- Les développeurs sont confrontés au « Syndrome du versionning »



http://ca.sun.com/en/events/presentations/devdays/docs/DD_Roland_final.pdf

J2EE Timeline



http://www.chariotsolutions.com/slides/JUG-j2ee1_4.ppt

- **Pourquoi industrialiser ?**
  - Réduire les coûts
  - Augmenter la productivité
  - Augmenter la qualité
  - Augmenter la réactivité
  - Adresser la complexité
  - …
- **Obstacles**
  - Les peurs : « Les temps modernes », «l'Offshore »,…
  - Les freins : certains acteurs (méthodologistes, développeurs, architectes,…) se considèrent/se veulent artistes, divas, indispensables,…

Le Potager du Roi - Versailles

- **La production d'applications aujourd'hui**

- **Amusant, beau, poétique, artisanal, désorganisé, fun, enrichissant,...**

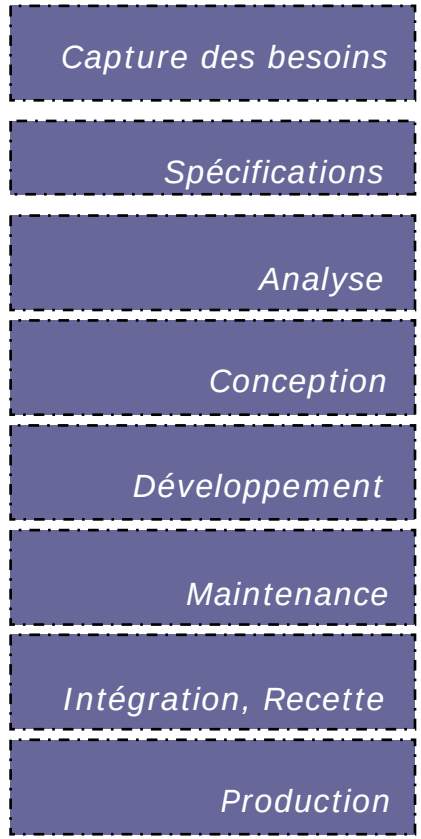
- **La production d'applications demain**

- **Pas fun du tout**
- **Et pourtant, la production d'applications doit s'industrialiser…**

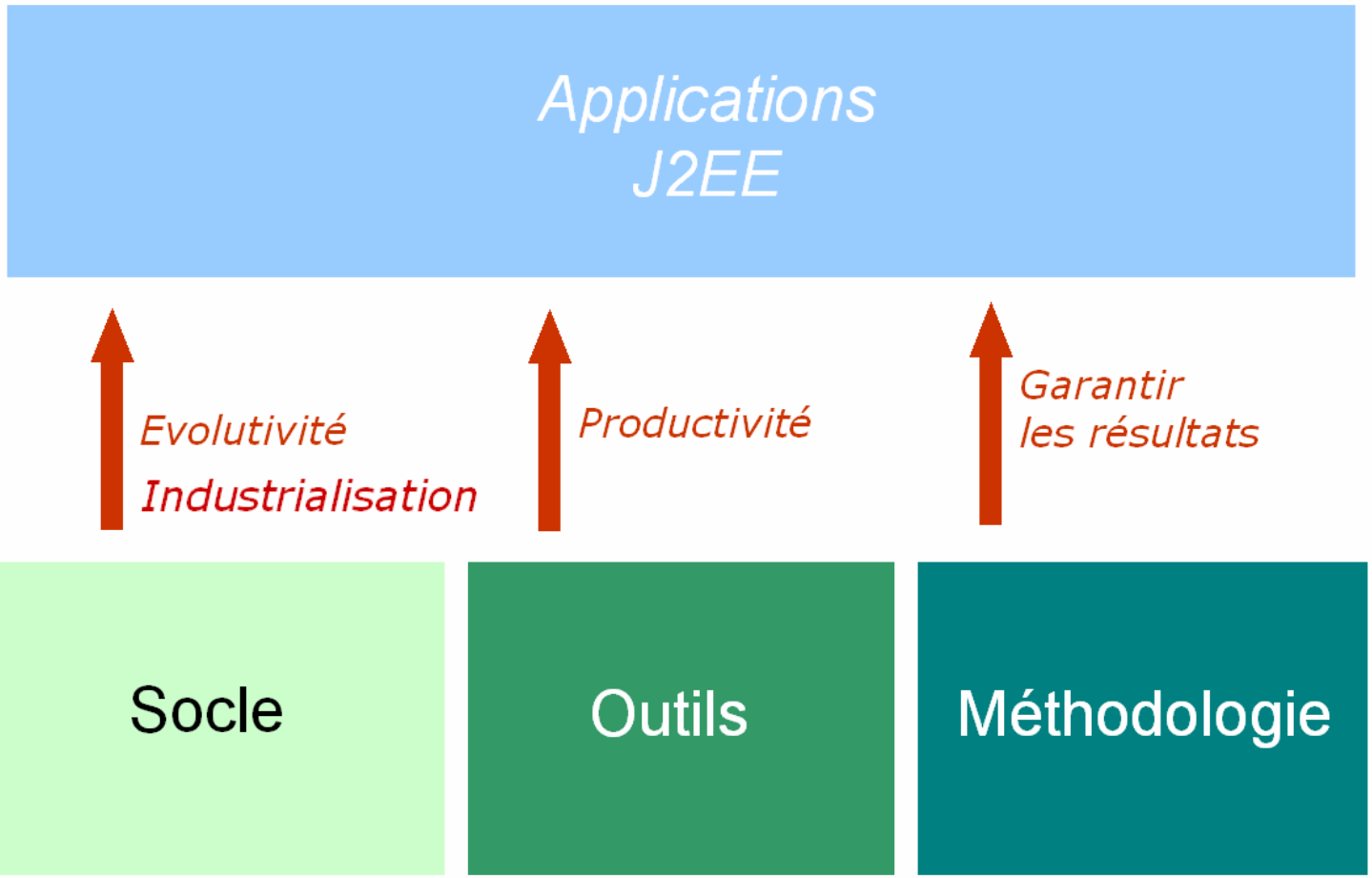
Besoins d'industrialisation

- **Simplicité**
  - Faciliter l'intégration de développeurs Métier, non experts
  - Mixer les équipes (internes / externes) et piloter ses sous-traitants
  - Aider/formaliser le dialogue MOA/MOE
- **Productivité**
  - Minimiser le temps de réalisation d'écrans et du métier
  - Faciliter le travail en équipe
  - Aider la MOA comprendre le cadre de développement
- **Maintenabilité**
  - Assurer une approche systématique quels que soient les intervenants
  - Garantir la qualité de service
- **Évolutivité**
  - Faciliter les nécessaires montées de versions de produits
  - Garantir la même qualité pour des projets hétérogènes : B2E, B2B, B2C
- **Réutilisabilité**
  - Capitaliser
- **Exploitabilité**
  - Garantir le fonctionnement
  - Accélérer la mise en production

- Il ne s'agit pas d'industrialiser uniquement les phases de développement.
- Toutes les phases de l'élaboration d'une application doivent être industrialisées
- Les développeurs ne sont pas la cible de l'industrialisation
- La chaîne de production doit être cohérente



- **L'industrialisation de l'élaboration des applications peut s'appuyer sur trois briques indépendantes**



- **Un socle n'est pas un framework !**

- **Le framework répond à un problème technique**
  - Navigation, Persistance, Sécurité…
- **Plus il y a de frameworks plus on se pose de questions**
  - Struts, Tapestry, JSF, Spring, Hivemind, Log4J,…
  - Quel framework choisir ?
- **Un framework amène des degrés de liberté aux développeurs**
  - Comment utiliser .Net ?
  - Comment utiliser J2EE ?
  - Comment utiliser Struts ?
  - Comment utiliser Spring ?
  - Comment utiliser Hibernate ?

Socle technique versus framework

- **Le socle technique est une vue des frameworks**
- **Le socle technique impose**
  - Un socle technique supprime des degrés de liberté aux développeurs
  - Un socle impose des règles de développement
  - Un socle impose une architecture
  - ...

« Plus le socle technique est bon
moins les acteurs se posent de questions »

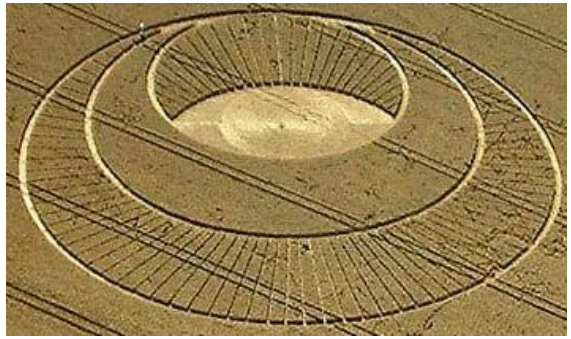
- De la méthode, des outils (?), pas de socle

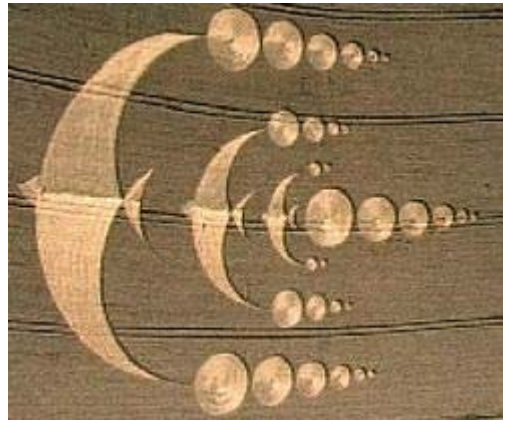
- Le socle garantit que les développements sont industrialisables

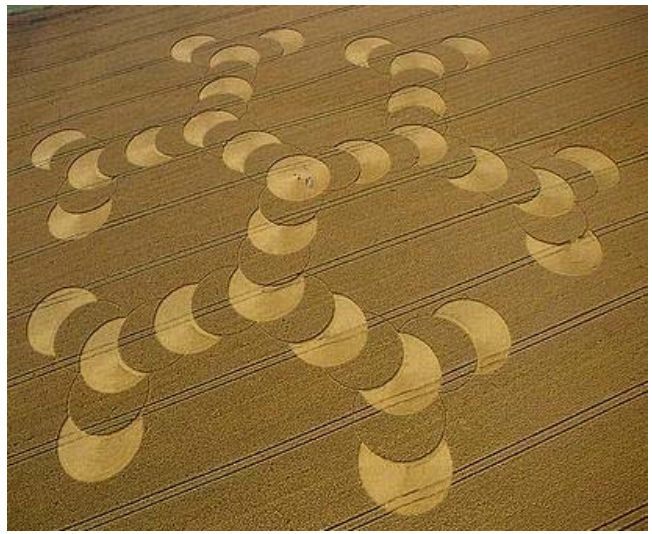
- **Un socle, de la méthode, pas d'outils**

- **Les outils améliorent la productivité**

- **Un socle, des outils, de la méthode**

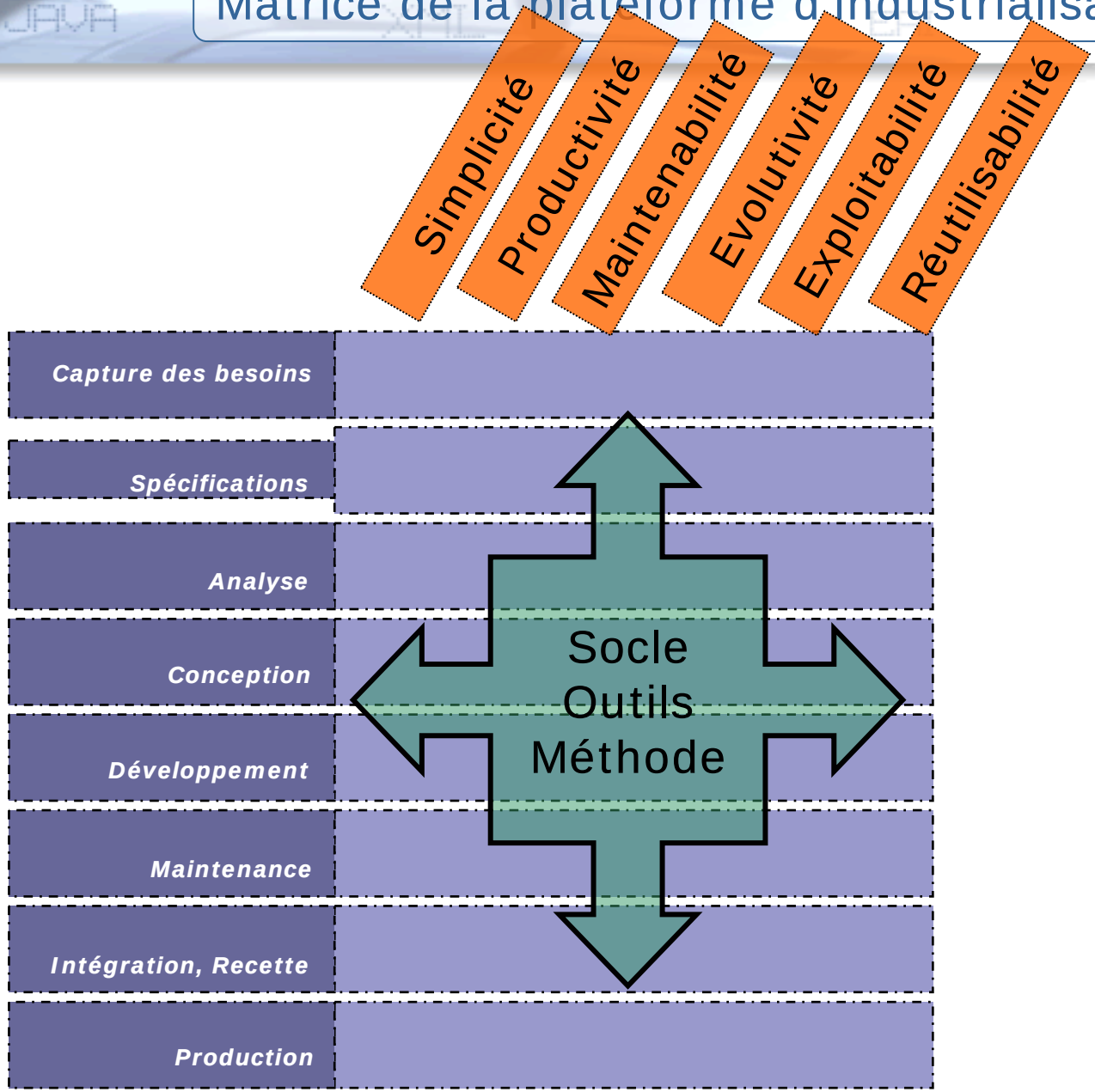






- **Mais qu'est-ce qui est industrialisé ?**

Matrice de la plateforme d'industrialisation



- **Amener une architecture**
- **Isoler le technique du fonctionnel**
- **Proposer des services techniques**

- **La technologie est uniforme et propriétaire**
- **Les applications sont propriétaires et centralisées**

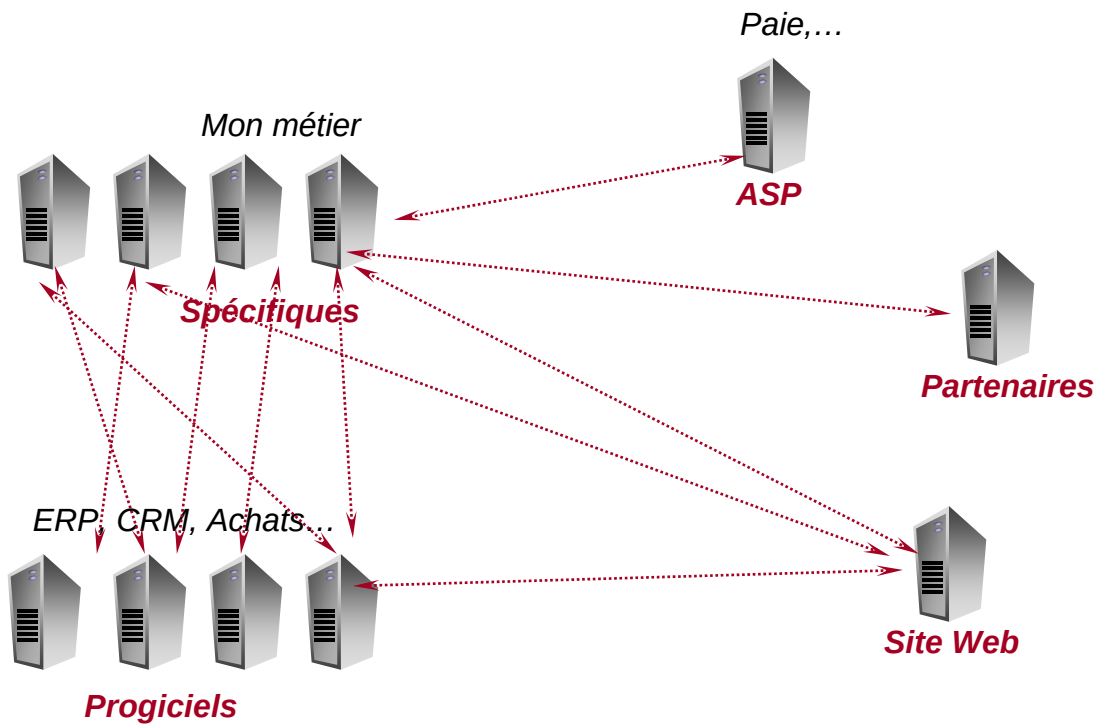
Achats, Paie, Gestion de stock, Gestion de production, Mon métier…

Spécifiques

- **Pas besoin d'architecture, pas besoin d'architectes**

- **Il faut satisfaire l'utilisateur**
- **Il faut dialoguer avec les partenaires**
- **Il faut contrôler les coûts**
  - Passer sur des technologies « ouvertes »
  - Progicialiser
  - Externaliser

SI : la cacophonie

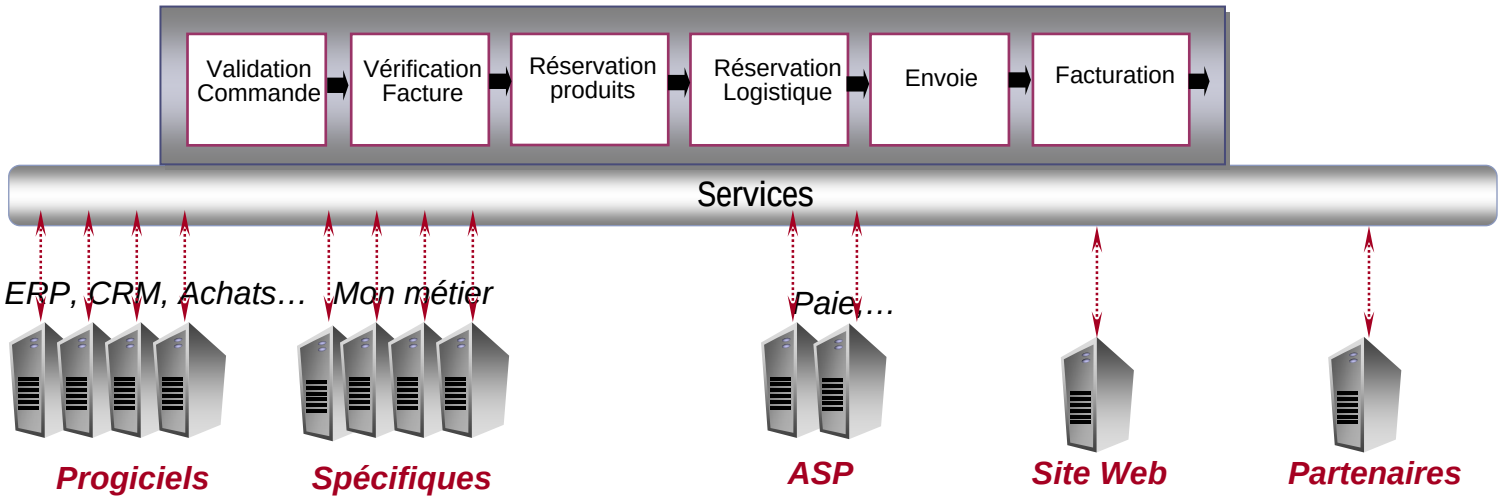


Applications Internes

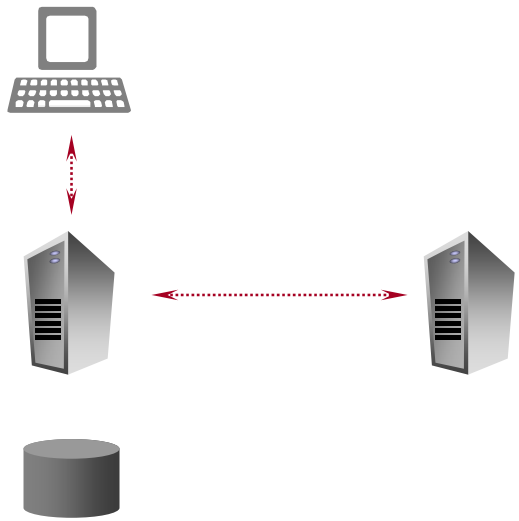
Applications Externes

- **Pourquoi intègre-t-on ?**
  - Pour unifier le système d'information : on ne s'est pas remis de l'éclatement du S.I.
  - 1995 : phase 1, le ~~data warehouse~~
  - 2000 : phase 2, l'~~EAI~~
  - 2005 : phase 3, le ~~portail~~
  - 2010 : SOA

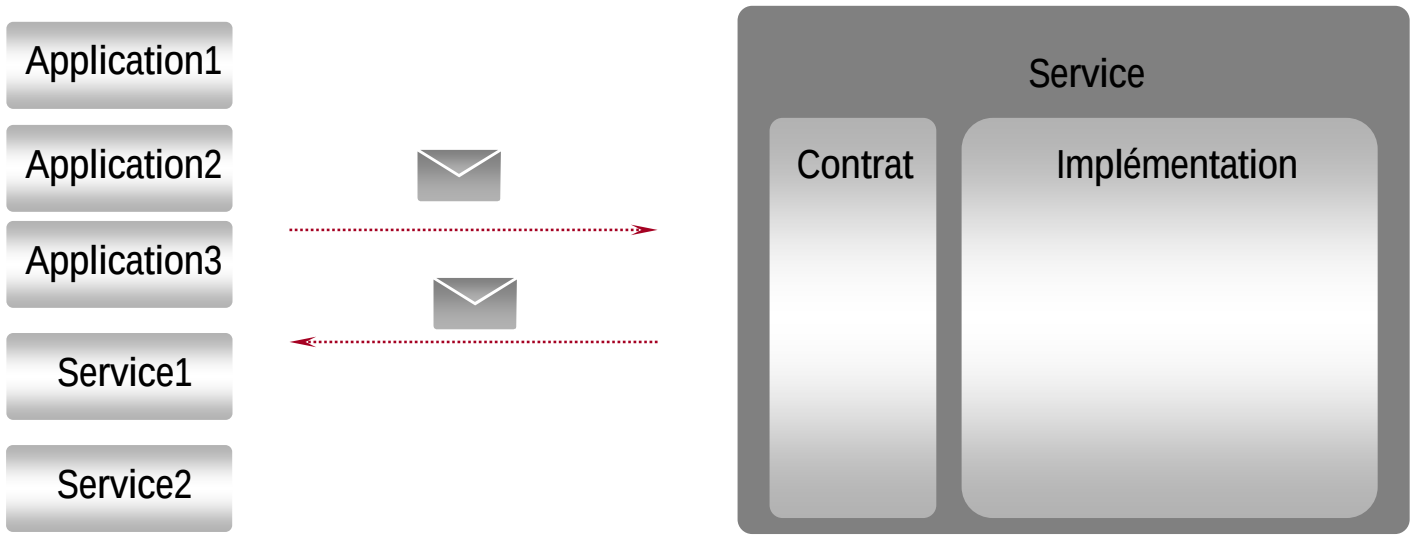
- **Actuellement le S.I. ne devrait plus être pensé en terme d'applications mais en terme de services**
  - Le S.I. peut devenir un ensemble de services réutilisables intégrés via des processus métiers



- **Il est donc important de construire des applications qui soient « SOA compatible »**

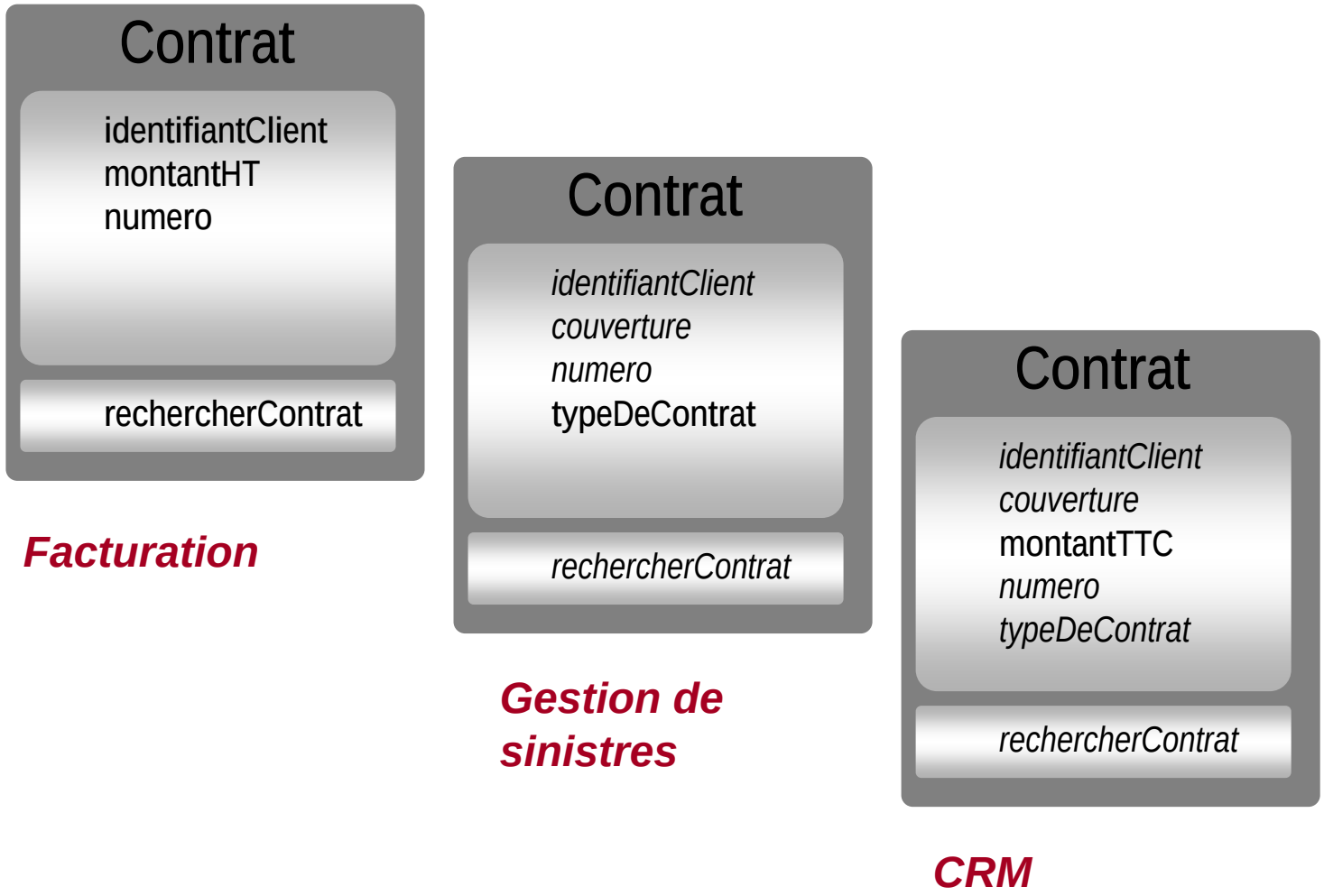


- **Un service est une fonction qui reçoit des messages et qui les restitue après traitement.**

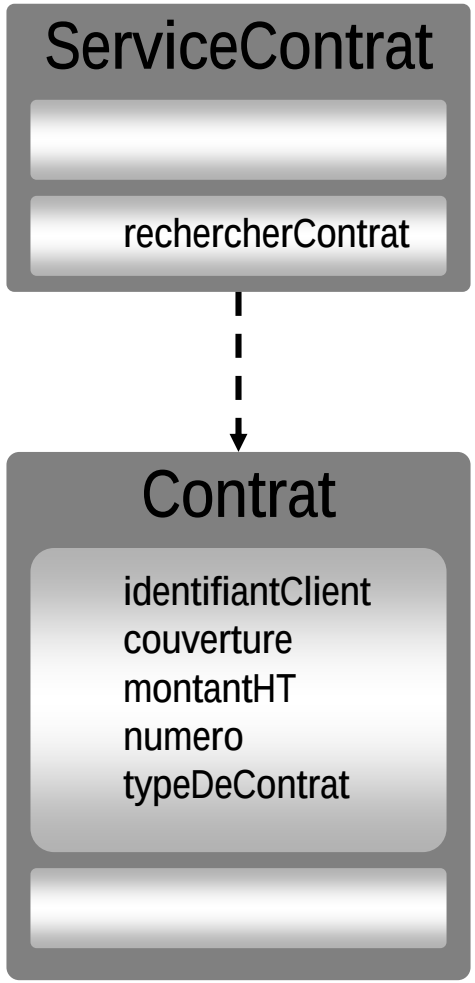


- **Un service est réutilisable et « thread safe »**
- **Un service respecte un contrat**
- **Un service est sans état**
- **Un service est indépendant de la plateforme d'exécution et du langage de programmation**

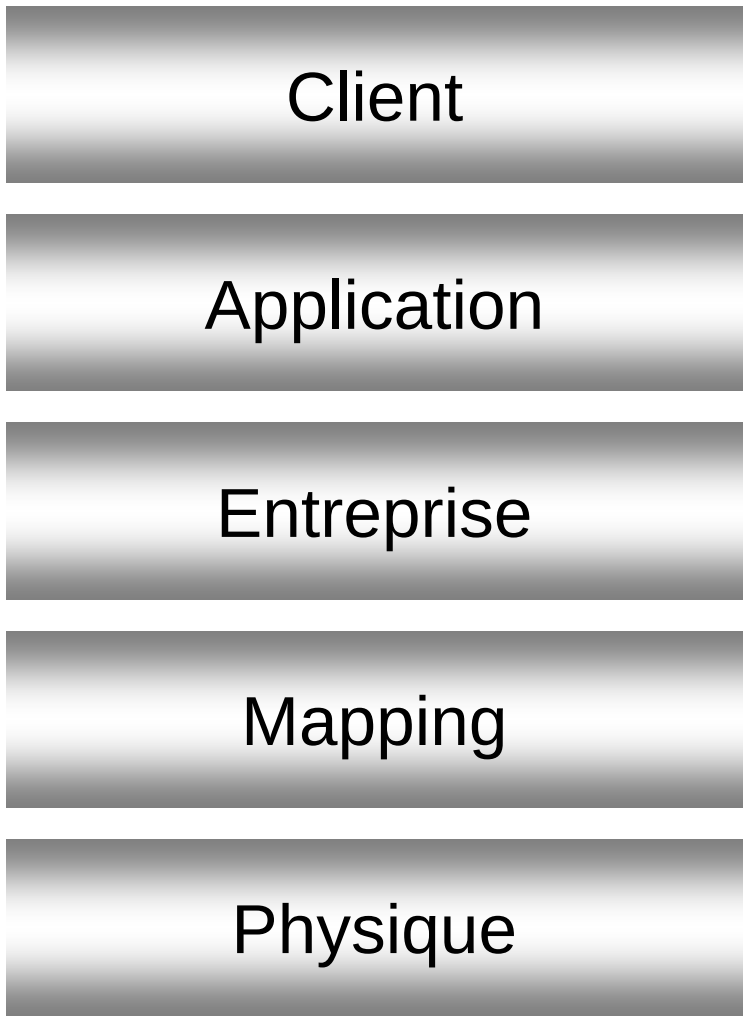
Un Contrat « Objet Métier »



Un service métier



Construire une application « SOA Compatible »



Interfaces : WEB, C/S, PDA ou encore XML

Le fonctionnel de mes applications

Mon métier réutilisable

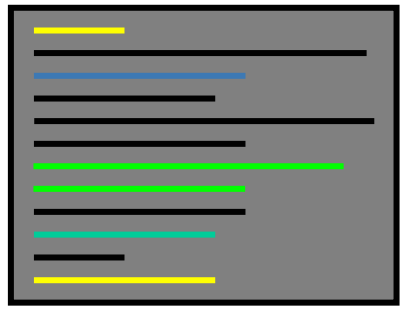
Lié à ma structure de données

SGBD, CICS, LDAP

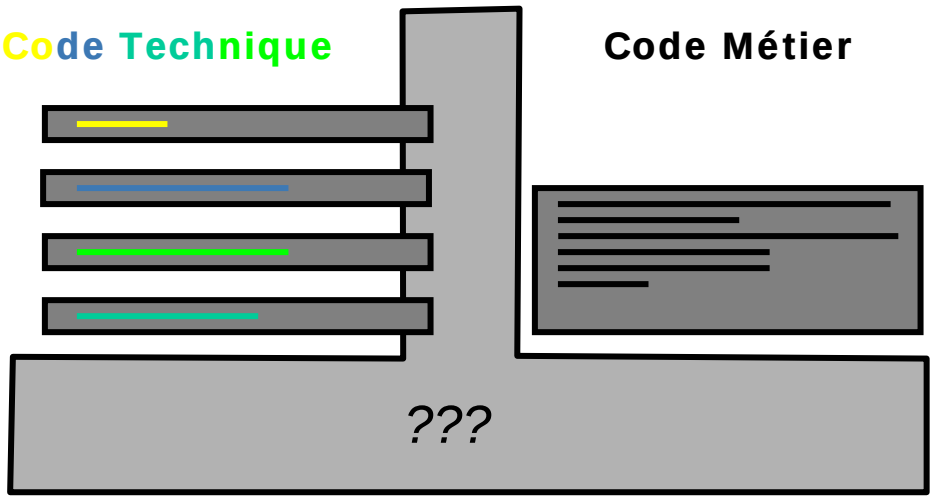
- Amener une architecture
- **Isoler le technique du fonctionnel**
- Proposer des services techniques

Dissocier le Métier de la Technologie

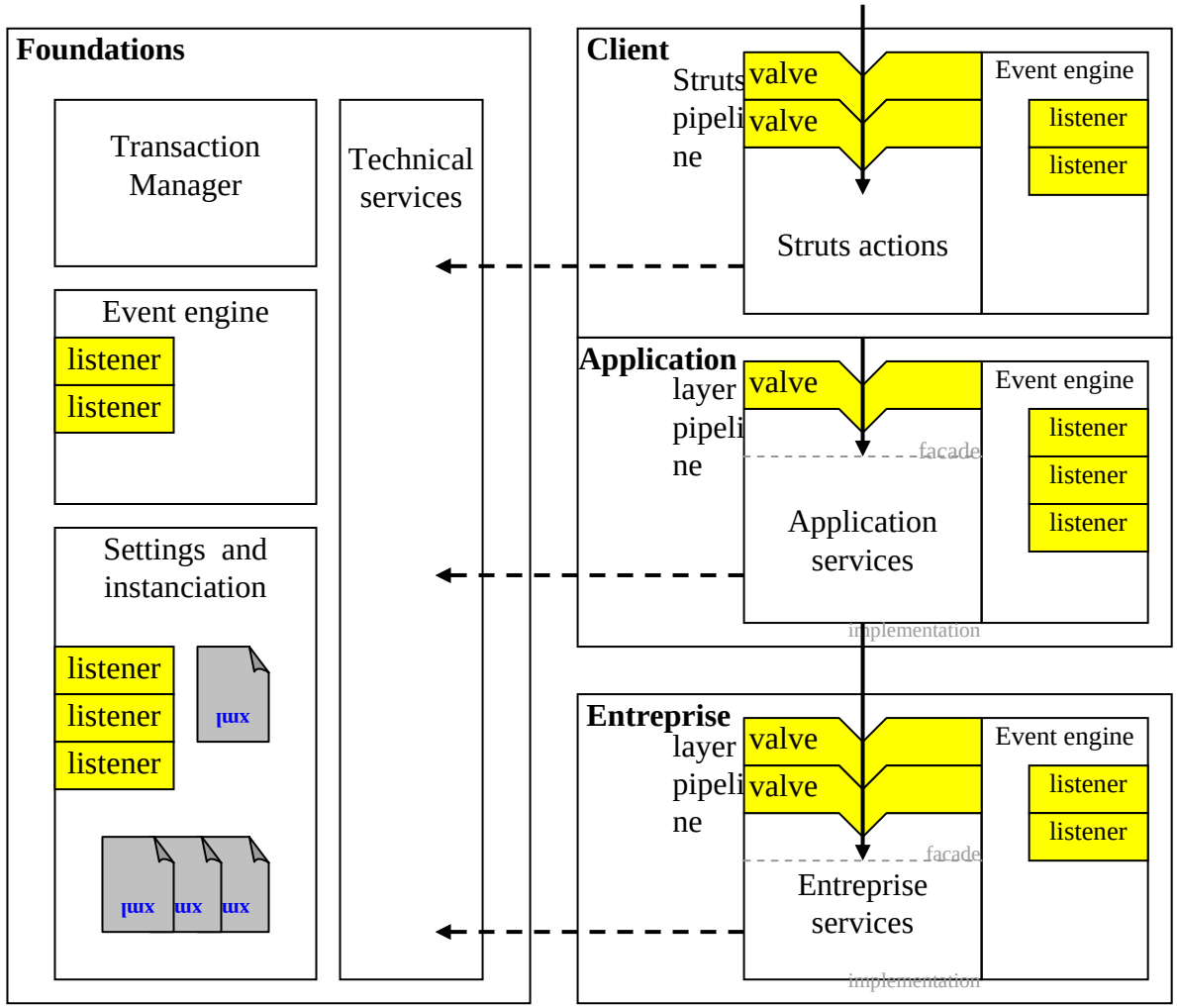
- Votre code aujourd'hui…



- L'objectif

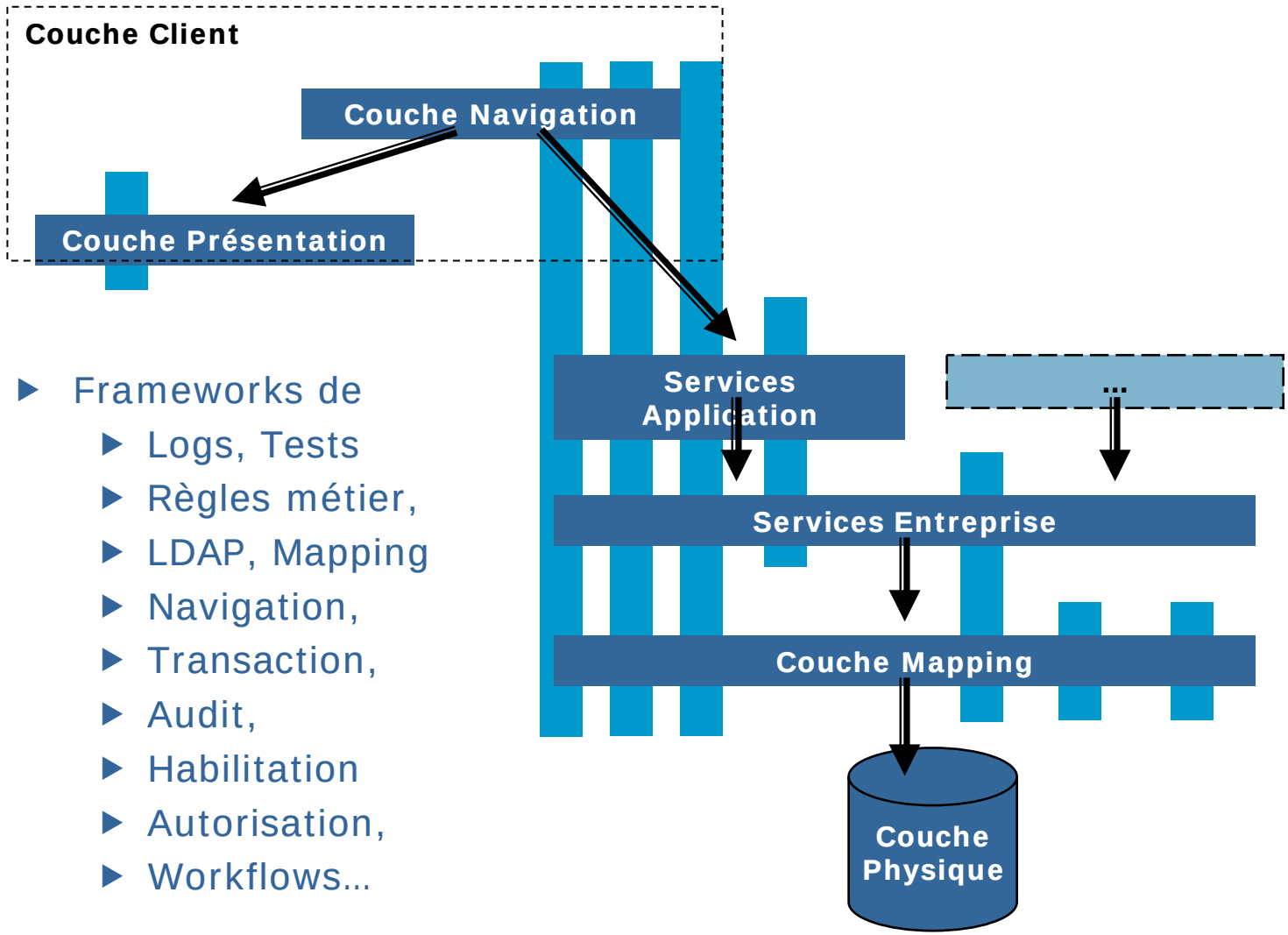


- **SOA, Micro Kernel, Design Patterns, Programmation Orienté Aspect (AOP), POJO**



- Amener une architecture
- Isoler le technique du fonctionnel
- Proposer des services techniques

SOA à 5 couches avec les frameworks



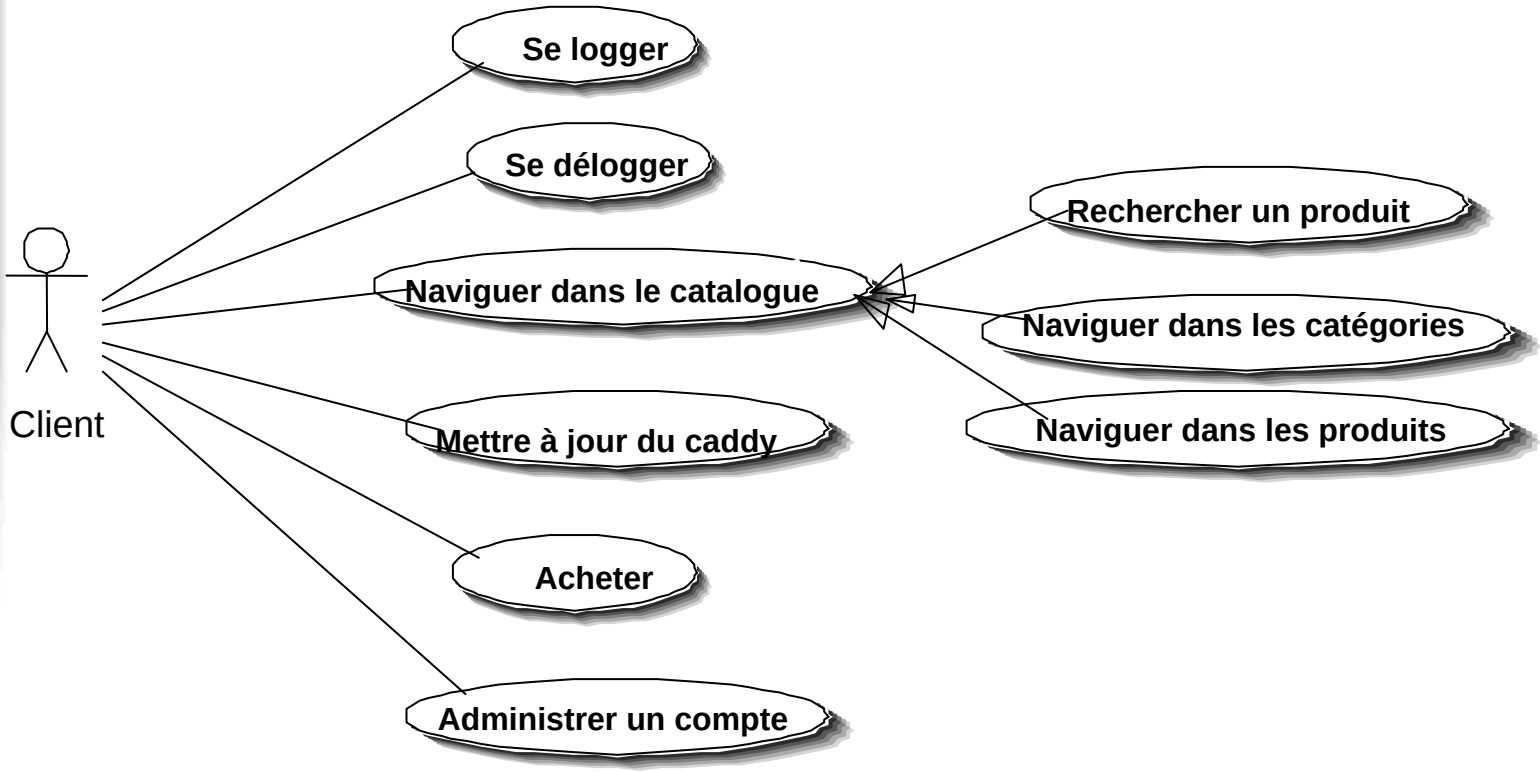
- Frameworks de
  - Logs, Tests
  - Règles métier,
  - LDAP, Mapping
  - Navigation,
  - Transaction,
  - Audit,
  - Habilitation
  - Autorisation,
  - Workflows…

Services offerts par un socle technique

- **Aide contextuelle**
- **Authentification**
- **Habilitation**
- **Configuration**
- **Gestion des erreurs**
- **Internationalisation**
- **Gestion de contexte**
- **Impression**
- **Emailing**
- **ASP**
- **Tests Unitaires**
- **Mapping : Hibernate, LiDO**
- **Existant : CICS, IMS, Tuxedo, PacBase**
- **Client léger**
- **Client riche**
- **Audit**
- **Supervision**
- **Workflow**

- **Une application de démonstration intégrant la plupart des fonctionnalités du socle technique IMPROVE Foundations**

- **L'application est un site d'e-commerce permettant d'acheter des peluches**



Questions ?

didier

IMPROVE
74/80, rue Roque de Fillol
92800 PUTEAUX

Tél. : 01.41.97.83.20
URL : http://www.improve.fr
URL : http://www.improve-technologies.com
URL : http://www.improve-institute.com

Resp. Commercial : Stève SFARTZ