Système Distribué avec CORBA

Yassamine Seladji

yassamine.seladji@gmail.com

17 mars 2020

Introductions

- En informatique distribuée, l'information est répartie entre différents serveurs du systèmes.

Introductions

- En informatique distribuée, l'information est répartie entre différents serveurs du systèmes.
- Les applications serveurs peuvent être implémenté avec des langages différents, et tourné sur différentes plateformes.

Introductions

- En informatique distribuée, l'information est répartie entre différents serveurs du systèmes.
- Les applications serveurs peuvent être implémenté avec des langages différents, et tourné sur différentes plateformes.
- Le moyen de communication $\Longrightarrow$ le protocole **CORBA**.

Introductions

- L'architecture **CORBA** est mise en place par le consortium **OMG**.

Introductions

- L'architecture **CORBA** est mise en place par le consortium **OMG**.
- L'OMG regroupe plusieurs organismes mondiaux (IBM, Boing, NASA, INRIA, ...).

Introductions

- L'architecture **CORBA** est mise en place par le consortium **OMG**.
- L'OMG regroupe plusieurs organismes mondiaux (IBM, Boing, NASA, INRIA, ...).
- L'OMG a le rôle de mettre en place des standards d'intégration d'applications réparties hétérogènes.

Les avantages de CORBA

- CORBA permet la communication entre différents langages.

Les avantages de CORBA

- CORBA permet la communication entre différents langages.
- CORBA permet d'obtenir un code très compact et efficace.

Les avantages de CORBA

- CORBA permet la communication entre différents langages.
- CORBA permet d'obtenir un code très compact et efficace.
- CORBA propose une technologie économe en bande passante.

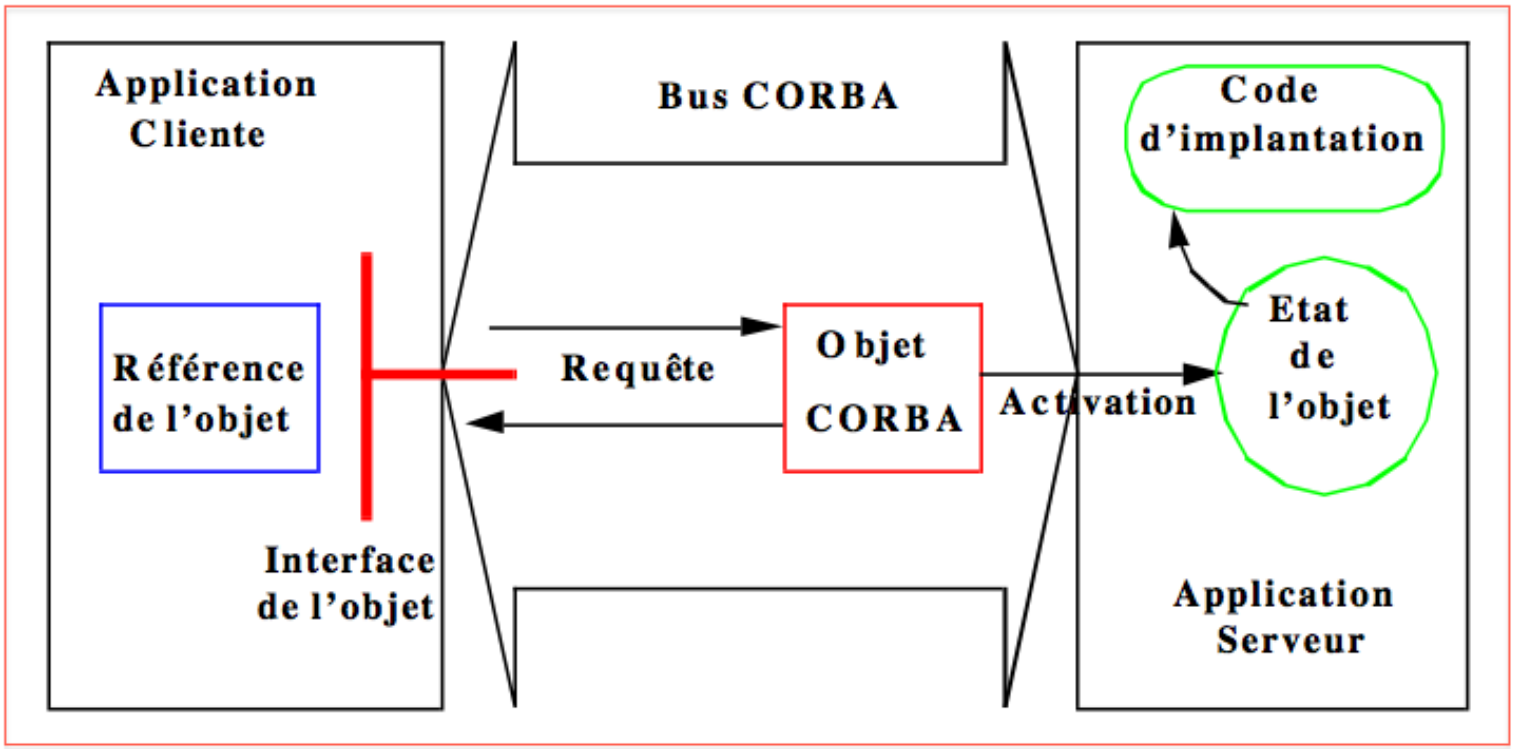
Les avantages de CORBA

- CORBA permet la communication entre différents langages.
- CORBA permet d'obtenir un code très compact et efficace.
- CORBA propose une technologie économe en bande passante.
- CORBA nécessite peu de bibliothèques.

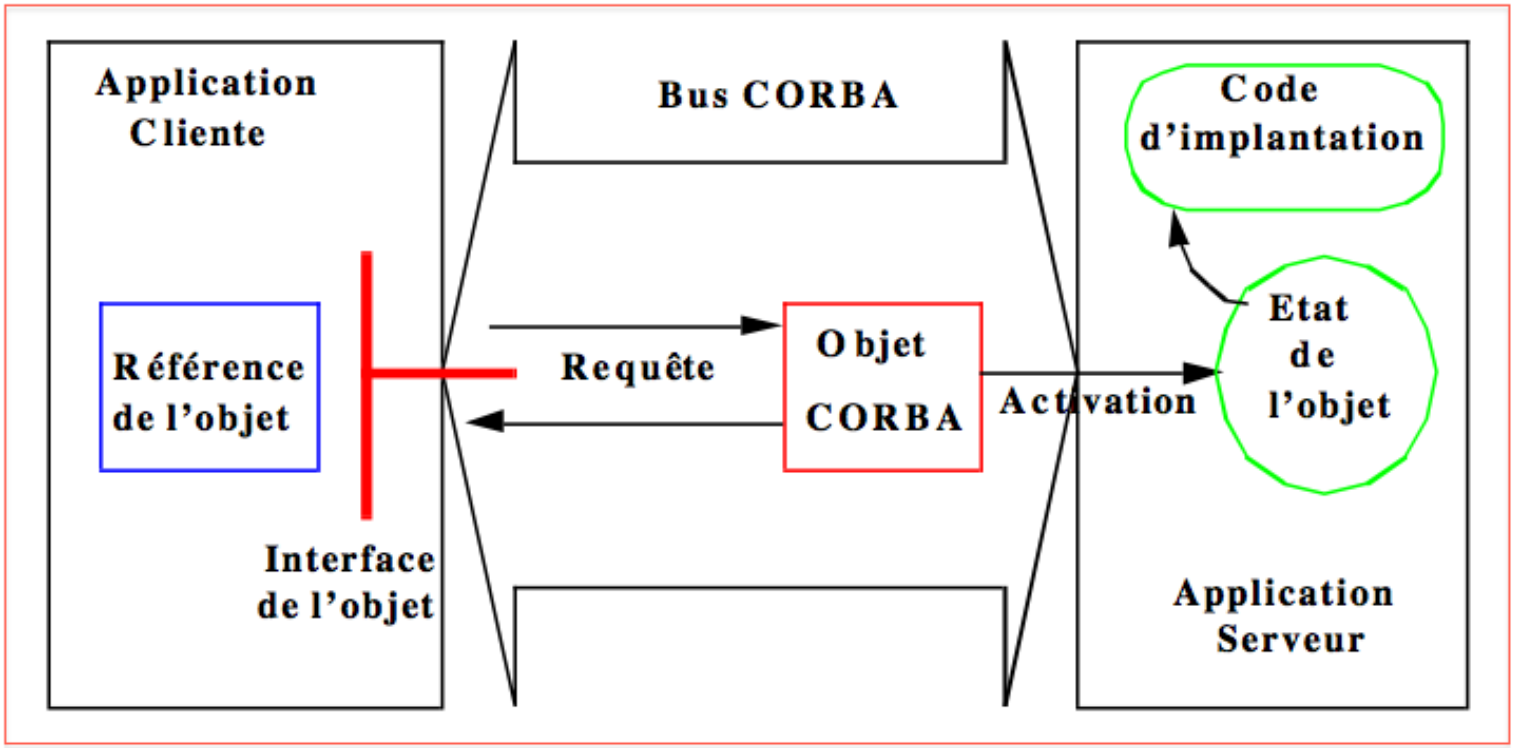
Les avantages de CORBA

- CORBA permet la communication entre différents langages.
- CORBA permet d'obtenir un code très compact et efficace.
- CORBA propose une technologie économe en bande passante.
- CORBA nécessite peu de bibliothèques.
- CORBA propose des profils pour le temps réel et l'embarqué.

Application répartie avec CORBA

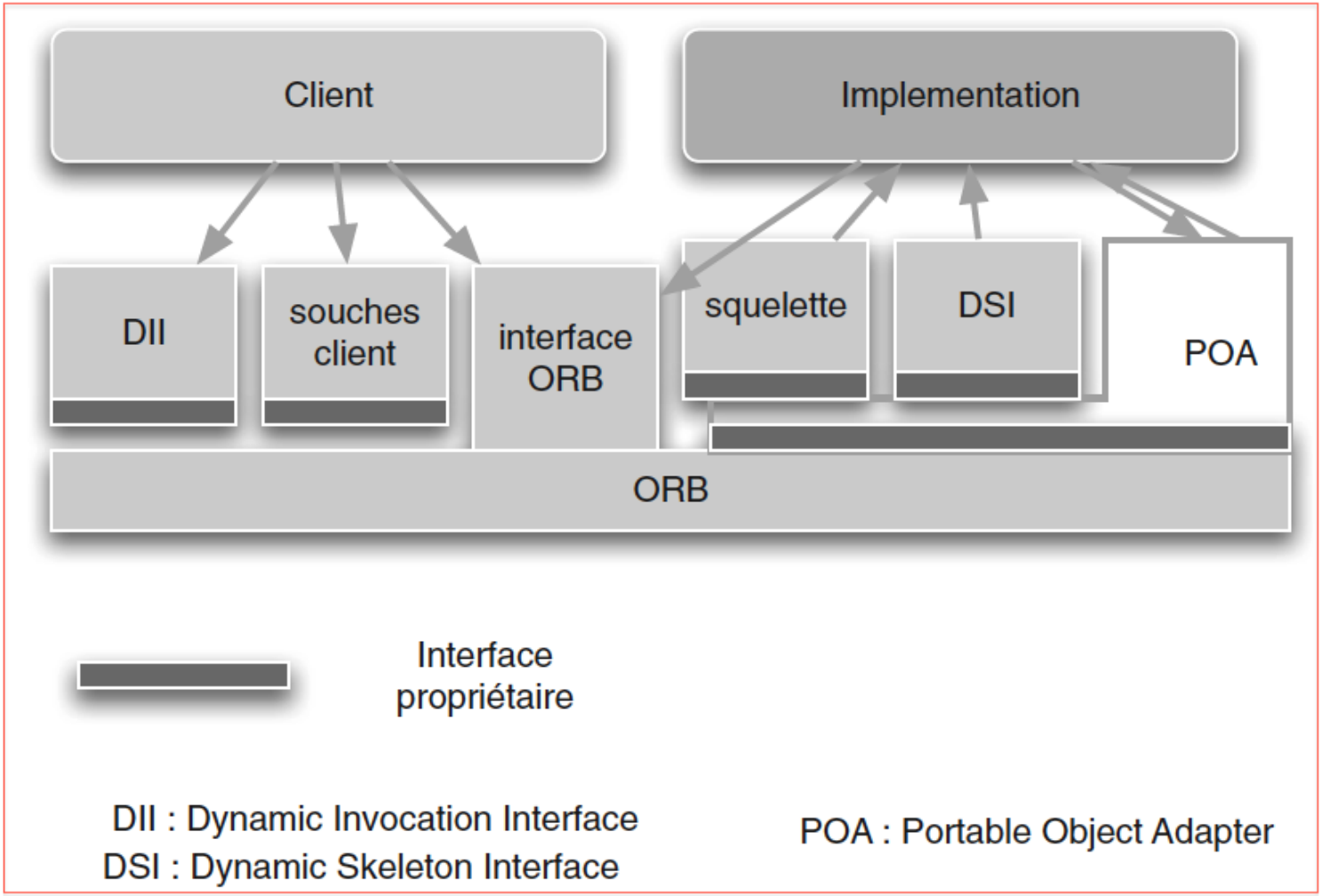


Application répartie avec CORBA



Chaque notion se traduit par une composante technologique de **CORBA**.

L'architecture de CORBA



L'architecture de CORBA

L'architecture **CORBA** se compose essentiellement :

- du bus logiciel ORB.
- des souches : stub et squelette.
- du POA (Portable Object Adapter).
- de l'interface IDL.
- des IOR (Interoperable Object Reference).
- Des services.

L'architecture de CORBA

L'architecture **CORBA** se compose essentiellement :

- du bus logiciel ORB.
- des souches : stub et squelette.
- du POA (Portable Object Adapter).
- de l'interface IDL.
- des IOR (Interoperable Object Reference).
- Des services.

L'architecture de CORBA

Le bus logiciel ORB

ORB (Object Request Broker)

- **ORB** est le noyau de transport des requêtes aux objets.
- **ORB** maintient et gère les références des objets distants.
- **ORB** permet le plage et dépliage des arguments.
- **ORB** achemine la requête vers le bon objet à partir de sa référence.

L'architecture de CORBA

Le bus logiciel ORB

ORB (Object Request Broker)

- L'accée à l'**ORB** est locale à chaque application.
- Une bibliothèque **ORB** pour chaque langage.
- En Java 5, un standard **ORB** est inclut.

L'architecture de CORBA

L'architecture **CORBA** se compose essentiellement :

- du bus logiciel ORB.
- des souches : stub et squelette.
- du POA (Portable Object Adapter).
- de l'interface IDL.
- des IOR (Interoperable Object Reference).
- Des services.

L'architecture de CORBA

Les souches

Côté client : le stub

- Le stub assure la liaison avec l'ORB.
- Le stub assure le codage et décodage des paramètres.
- Le stub assure la transparence des références du serveurs.

L'architecture de CORBA

Les souches

Côté serveur : le squelette

- Le serveur utilise la notion de **servant**.
- Le servant permet de gérer l'activation de l'objet distant à la demande.
- L'exemple du bureau d'accueil.
- Le servant représente le **POA**.

L'architecture de CORBA

L'architecture **CORBA** se compose essentiellement :

- du bus logiciel ORB.
- des souches : stub et squelette.
- du POA (Portable Object Adapter).
- de l'interface IDL.
- des IOR (Interoperable Object Reference).
- Des services.

L'architecture de CORBA

Le POA

Le POA est l'adaptateur d'objet côté serveur.

- Il génère et interprète les références des objets distants.

L'architecture de CORBA

Le POA

Le POA est l'adaptateur d'objet côté serveur.

- Il génère et interprète les références des objets distants.
- Il gère les autorisations d'accès aux objets.

L'architecture de CORBA

Le POA

Le POA est l'adaptateur d'objet côté serveur.

- Il génère et interprète les références des objets distants.
- Il gère les autorisations d'accès aux objets.
- Il permet l'invocation des méthodes sur les objets.

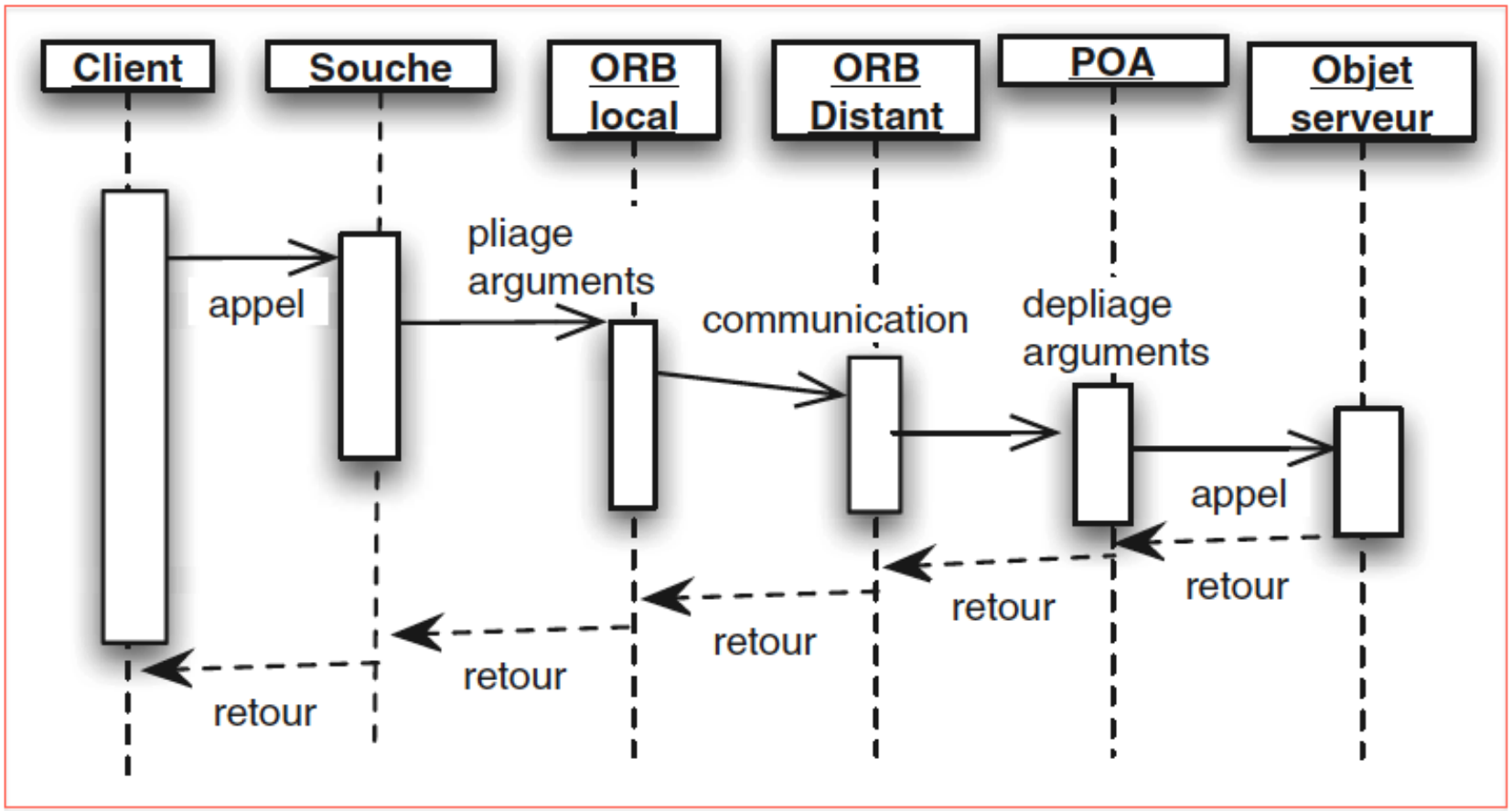
L'architecture de CORBA

Le POA

Le POA est l'adaptateur d'objet côté serveur.

- Il génère et interprète les références des objets distants.
- Il gère les autorisations d'accès aux objets.
- Il permet l'invocation des méthodes sur les objets.
- Il maintient les associations entres objets CORBA.

Le panorama de l'appel distant



Le panorama de l'appel distant

- Serveur + Services $\Longrightarrow$ Interface.

Le panorama de l'appel distant

- Serveur + Services $\Longrightarrow$ Interface.
- CORBA $\Longrightarrow$ Une interface qui assure l'interopérabilité des langages.

Le panorama de l'appel distant

- Serveur + Services $\Longrightarrow$ Interface.
- CORBA $\Longrightarrow$ Une interface qui assure l'interopérabilité des langages.
- L'utilisation de l'**IDI** (**I**nterface **D**efinition **L**anguage).

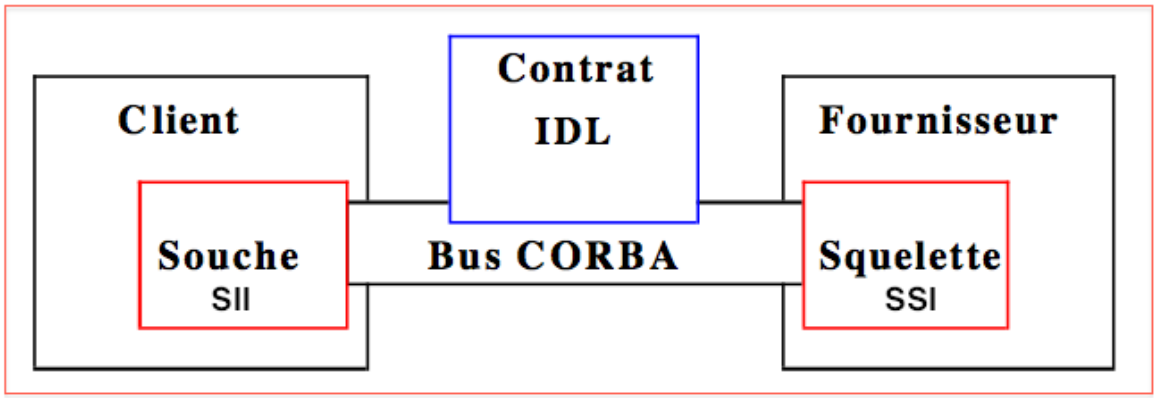
L'architecture de CORBA

L'architecture **CORBA** se compose essentiellement :

- du bus logiciel ORB.
- des souches : stub et squelette.
- du POA (Portable Object Adapter).
- de l'interface IDL.
- des IOR (Interoperable Object Reference).
- Des services.

L'architecture de CORBA

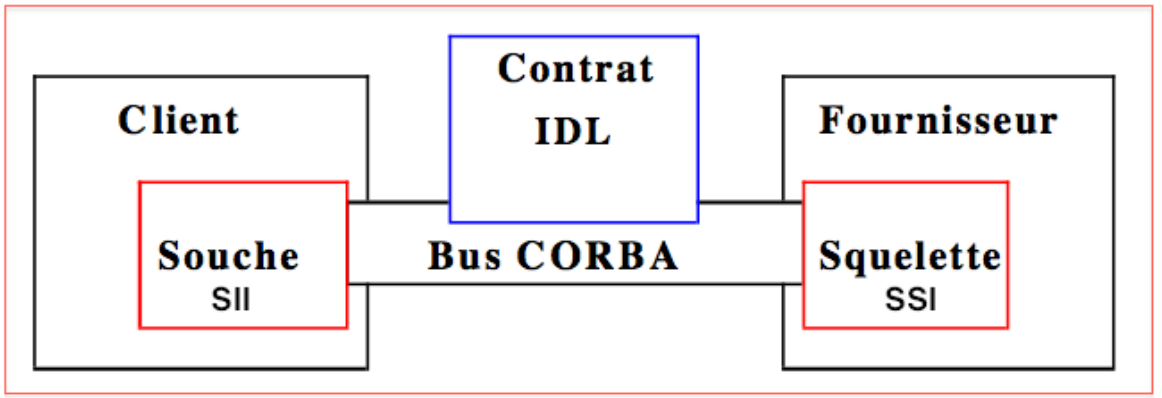
L'IDL



- Interface commune aux langages.

L'architecture de CORBA

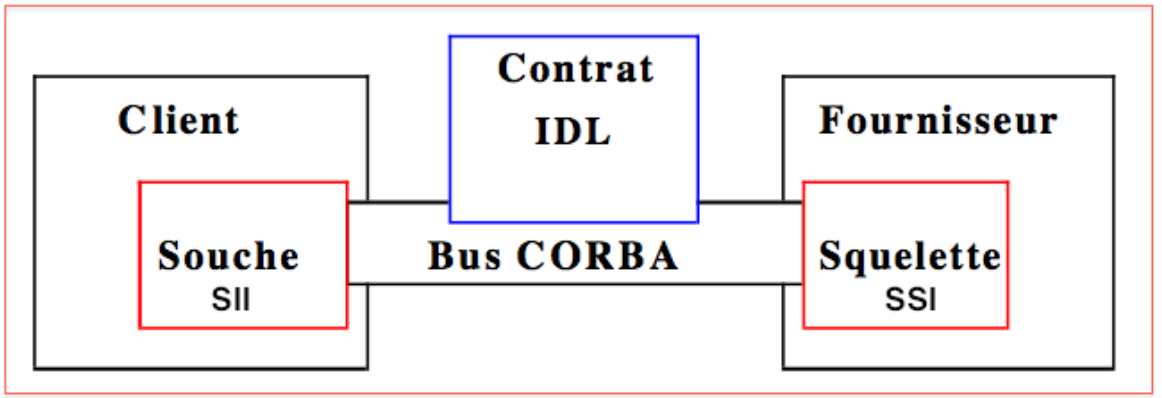
L'IDL



- Interface commune aux langages.
- Interface **IDL** $\Longrightarrow$ indépendante de l'implémentation.

L'architecture de CORBA

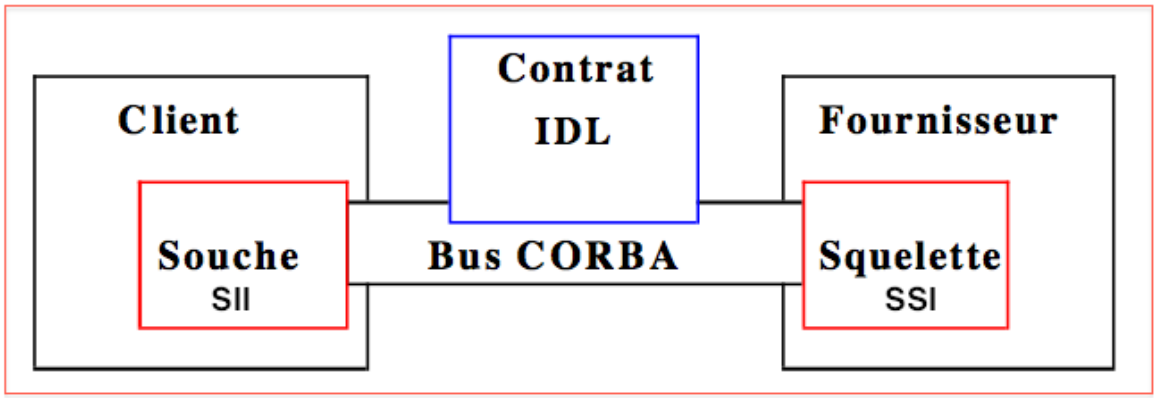
L'IDL



- Interface commune aux langages.
- Interface **IDL** $\Longrightarrow$ indépendante de l'implémentation.
- Définir les interfaces des objets **CORBA** distants.

L'architecture de CORBA

L'IDL



- Interface commune aux langages.
- Interface **IDL** $\Longrightarrow$ indépendante de l'implémentation.
- Définir les interfaces des objets **CORBA** distants.
- Le plus petit dénominateur commun entre tous les types de données.

L'architecture de CORBA

L'IDL

Les étapes de la création d'une IDL

- La déclaration de module.

L'architecture de CORBA

L'IDL

Les étapes de la création d'une IDL

- La déclaration de module.
- la déclarations des interfaces.

L'architecture de CORBA

L'IDL

Les étapes de la création d'une IDL

- La déclaration de module.
- la déclarations des interfaces.
- La présentation des signatures des méthodes.

<modules>
  <interfaces>
  <methodes>
	module universite {
	  interface Etudiant {
	  string coordonnees();
	  boolean bacplus(**in long** annee);
	  };
	};

L'architecture de CORBA

L'IDL

Les étapes de la création d'une IDL

- La déclaration de module.
- la déclarations des interfaces.
- La présentation des signatures des méthodes.

<modules>
  <interfaces>
  <methodes>
	module universite {
	  interface Etudiant {
	  string coordonnees();
	  boolean bacplus(**in long** annee);
	  };
	};

- La déclaration : des attributs, des exceptions, des types et des constantes.

L'architecture de CORBA

L'IDL

- Les modules $\Longrightarrow$ packages Java.

L'architecture de CORBA

L'IDL

- Les modules $\Longrightarrow$ packages Java.
- Les interfaces $\Longrightarrow$ La signatures des méthodes.

TypeRetour NomMéthode (ListeParamètres) raises (Exception)

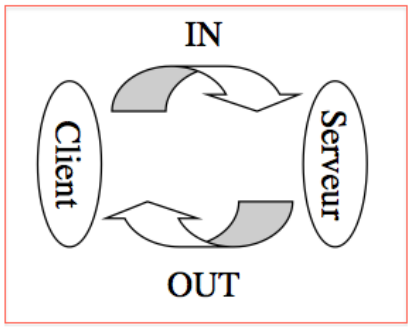
modePassage type nom

L'architecture de CORBA

L'IDL

Les modes de passage :

- in $\Longrightarrow$ en entrée :
  - le **client** fournit la valeur.
  - si le serveur la modifie, le client **ne voit pas** la modification.
- out $\Longrightarrow$ en sortie :
  - le **serveur** fournit la valeur
  - le client **voit** la modification.
- inout $\Longrightarrow$ en entrée/sortie.
  - le client fournit la valeur
  - si le serveur la modifie, le client **voit** la modification



L'architecture de CORBA

L'IDL

- Les exceptions $\Longrightarrow$ exceptions Java.

```
exception creditDepasse { long montant ; };
```

L'architecture de CORBA

L'IDL

- Les exceptions $\Longrightarrow$ exceptions Java.

```
exception creditDepasse { long montant ; };
```

- les attributs

L'architecture de CORBA

L'IDL

- Les exceptions $\Longrightarrow$ exceptions Java.

```
exception creditDepasse { long montant ; };
```

- les attributs
  - **attribute** type nom $\Longrightarrow$ getteurs + setteurs.

L'architecture de CORBA

L'IDL

- Les exceptions $\Longrightarrow$ exceptions Java.

```
exception creditDepasse { long montant ; };
```

- les attributs
  - **attribute** type nom $\Longrightarrow$ getteurs + setteurs.
  - **readonly attribute** type nom $\Longrightarrow$ getteurs.

L'architecture de CORBA

L'IDL

Les types primitifs :

Java	CORBA
void	void
boolean	boolean
char	char/wchar
byte	octet
short	short/ unsigned short
int	long/ unsigned long
long	long long /unsigned long long
float	float
double	double
String	string, wstring

L'architecture de CORBA

L'IDL

- Les Constantes :

```
const nom = valeur ;
```

- Exemple :
  const double PI = 3.1415.

L'architecture de CORBA

L'IDL

Les types complexes :

- Nouveau type :

```
typedef long duree;
```

- Les énumérations :

```
enum mois {janvier,fevrier};
```

- Les structures :

```
struct erreur {
    long date;
    string message;
};
```

L'architecture de CORBA

L'IDL

Les types complexes :

- Les tableaux :

```
typedef float vecteur[10] ;
```

- Les séquences :

```
typedef sequence <long> historique ;
```

- Le type anonyme **any** $\Longrightarrow$ n'importe quel type de données.

L'architecture de CORBA

L'IDL

- Les interfaces acceptent l'héritage multiples.
- Héritage sémantique.

L'architecture de CORBA

L'IDL

- Les interfaces acceptent l'héritage multiples.
- Héritage sémantique.

Exemple :

```
interface J:I {…};
```

L'architecture de CORBA

L'IDL

- Les interfaces acceptent l'héritage multiples.
- Héritage sémantique.

Exemple :

```
interface J:I {…};
```

Équivalent Java :

```
interface J extends I {…}
```

L'architecture de CORBA

L'IDL

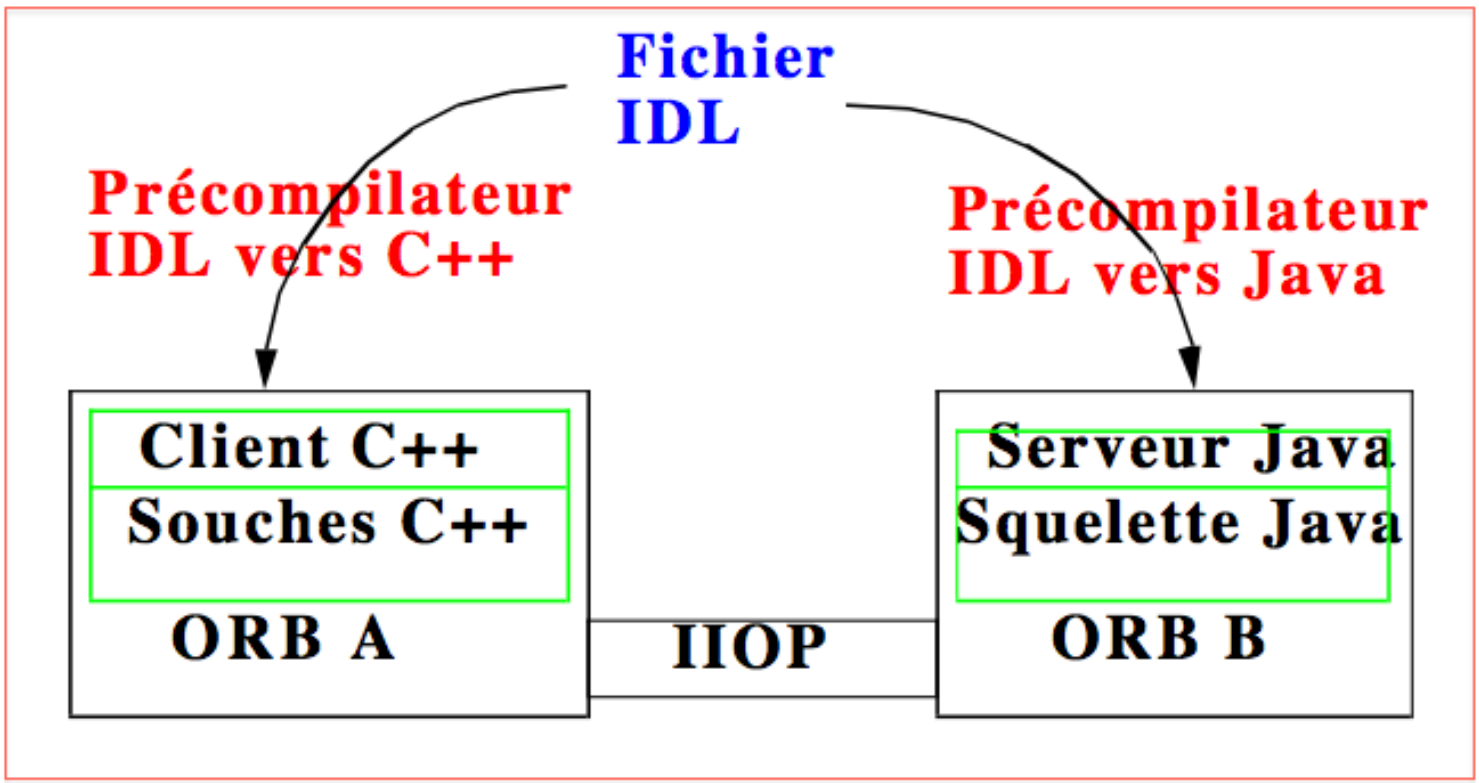
Exemple :

```
module finance {
    exception creditDepasse {};
    interface banque {
        attribute double montantMaximumAutorise ;
        void depot(in double montant) ;
        void retrait(in double montant) raises (creditDepasse);
        double solde() ;
    };
};
```

L'architecture de CORBA

L'IDL

La précompilation d'un contrat IDL :



L'architecture de CORBA

L'IDL

Exemple :

```
module finance {
    exception creditDepasse {};
    interface banque {
        attribute double montantMaximumAutorise ;
        void depot(in double montant) ;
        void retrait(in double montant) raises (creditDepasse);
        double solde() ;
    };
};
```

Précompilation Java :

```
public interface banqueOperations
    {double montantMaximumAutorise ();
    void montantMaximumAutorise (double newMontantMaximumAutorise);
    void depot (double montant);
    void retrait (double montant) throws finance.creditDepasse;
    double solde ();
} // interface banqueOperations
```

L'architecture de CORBA

L'IDL

Exercice 1 : Écrire un fichier IDL qui représente l'interface d'une calculatrice, tel que :

- elle contient les méthodes : ajouteMemoire, soustraitMemoire, multiplieMemoire et miseAZero.
- Chacune de ces méthodes prend un unique paramètre en in de type double (sauf miseAZero qui ne prend pas de paramètre).
- Elles mettent à jour la mémoire mais ne renvoient pas d'information au client.
- La méthode diviseMemoire doit lever une exception en cas de tentative de division par zéro.
- elle comporte deux autres méthodes : incrementer et decrementer.

L'architecture de CORBA

L'IDL

Correction :

```
module tpcorba {

  module exo2 {

        exception divisionParZero {};

        interface calcul  {

                readonly attribute double memoire;

                void miseAZero();

                void ajouteMemoire(in double donnee);
                void soustraitMemoire(in double donnee);
                void multiplieMemoire(in double donnee);
                void diviseMemoire(in double donnee) raises (divisionParZero);

                void incrementer(inout long data);
                void decrementer(inout long data);

        };
  };
};
```

L'architecture de CORBA

L'architecture **CORBA** se compose essentiellement :

- du bus logiciel ORB.
- des souches : stub et squelette.
- du POA (Portable Object Adapter).
- de l'interface IDL.
- des IOR (Interoperable Object Reference).
- Des services.

L'architecture de CORBA

L'IOR

IOR (**I**nteroperable **O**bject **R**eference)

- Les références d'objets CORBA.
- Les IORs se composent de manière codé de :
  - l'adresse IP de la machine Internet où est localisé l'objet.
  - un port IP pour se connecter au serveur de l'objet.
  - une clé pour désigner l'objet dans le serveur.
- Exemple :

```
IOR:000000000000001b49444c3a506c6163654d61726368652f426f757273653a312e30000
0000000010000000000000086000102000000000c3139322e3136382e312e3300c02b000000
000031afabcb0000000020391f165b0000000100000000000000010000008526f6f74504f4
1000000000800000001000000001400000000000002000000010000002000000000000010001
00000002050100010001002000010109000000010001010000000026000000020002
```

- IOR est stocké dans un service de nom (Naming Service).

L'architecture de CORBA

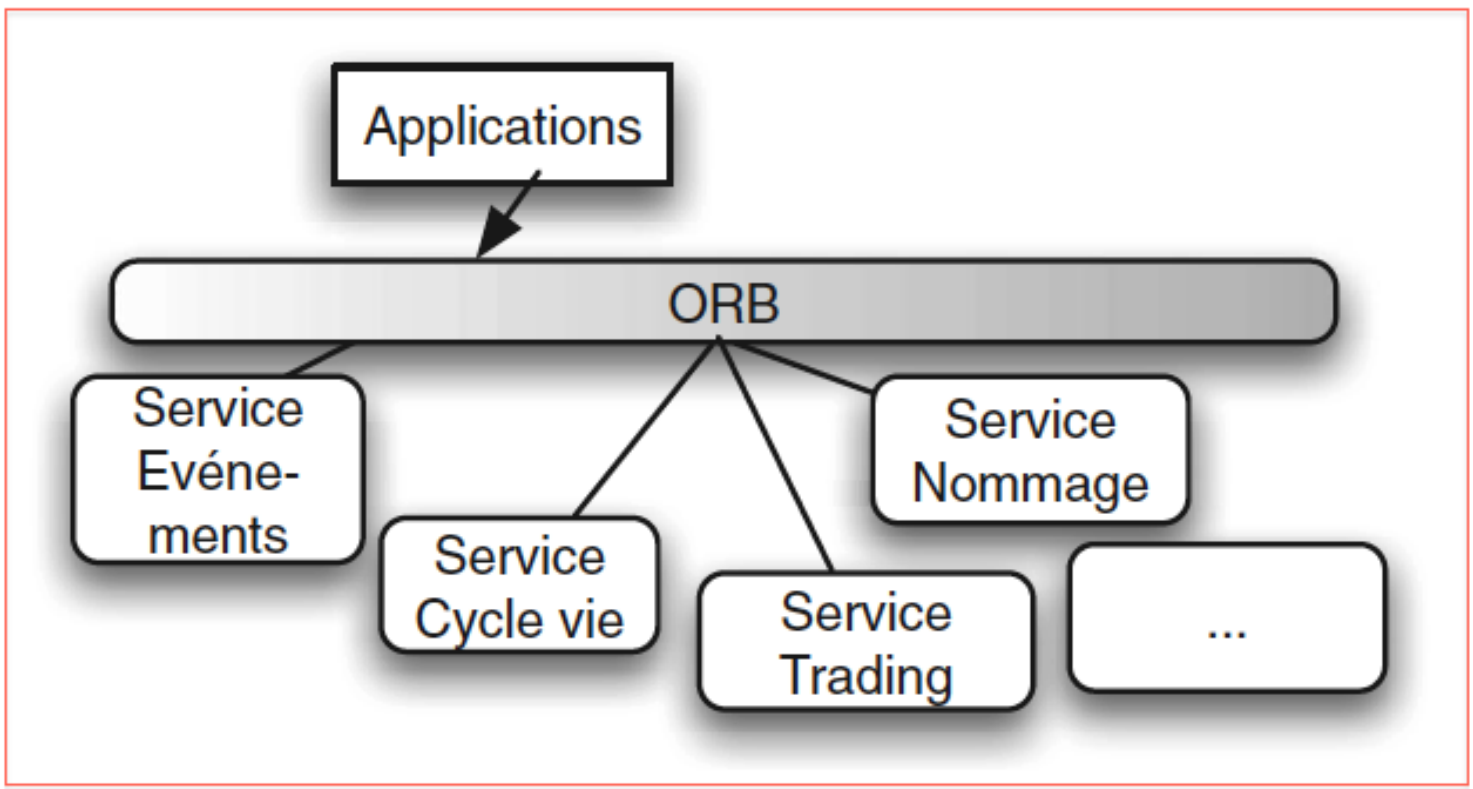
L'architecture **CORBA** se compose essentiellement :

- du bus logiciel ORB.
- des souches : stub et squelette.
- du POA (Portable Object Adapter).
- de l'interface IDL.
- des IOR (Interoperable Object Reference).
- Des services.

L'architecture de CORBA

Les services

Les services CORBA $\Longrightarrow$ bibliothèques pour augmenter la qualité du service.



L'architecture de CORBA

Les services

- **Service de nommage** : service d'annuaire CORBA.
- **Service évènement** : envoyer des évènements asynchrones entre applications.
- **Service de concurrence** : gérer les accès concurrent aux serveurs.
- **Service de sécurité** : gérer les accès sécurisés aux serveurs (identification, confidentialité ...)
- ....