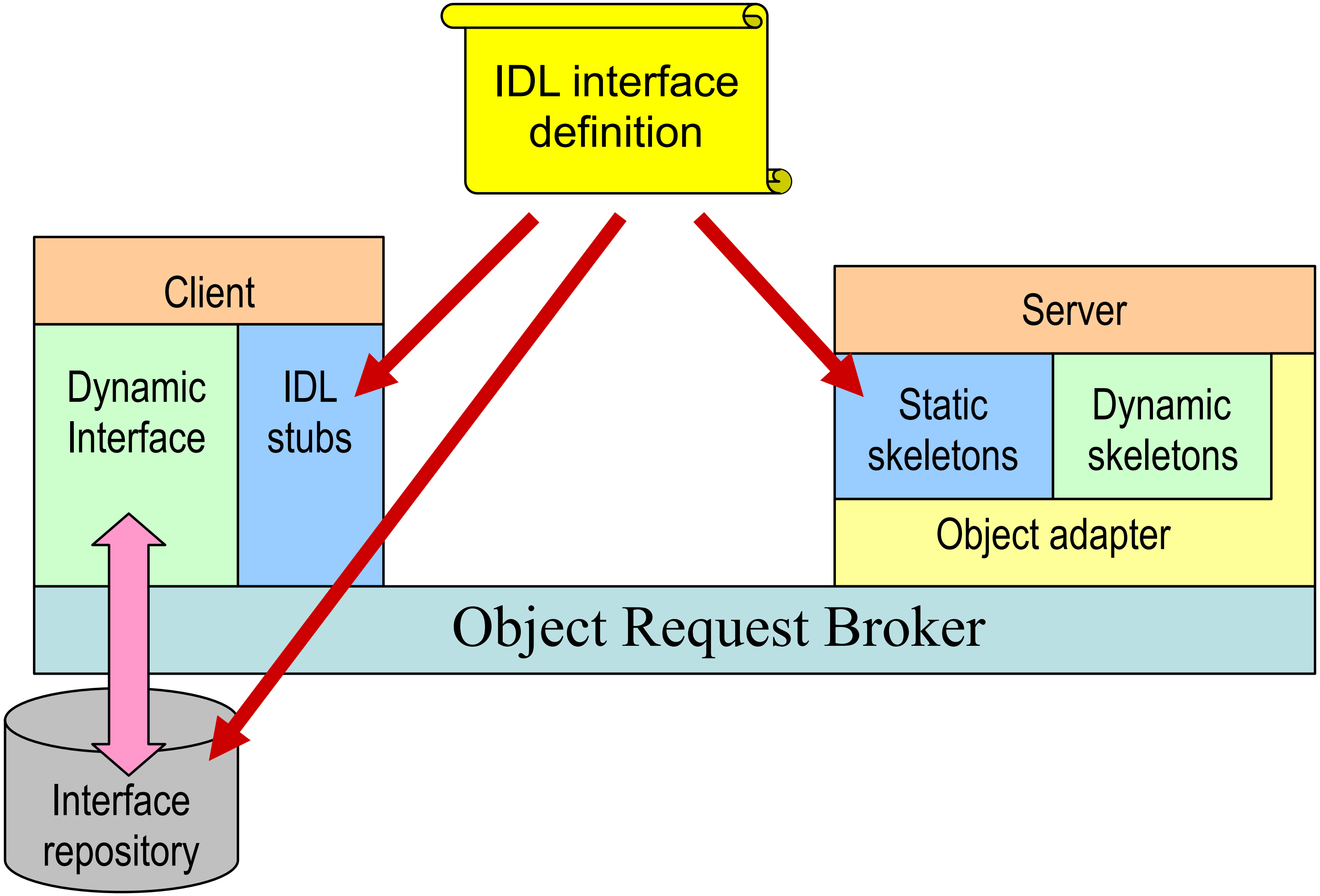
IDL
Interface Definition Language

IDL products



Main IDL elements

- Modules
- Interfaces
- Data types
- Constants
- Attributes
- Operations
- Exceptions

IDL data types

- Basic types
    - **`short  long  float  boolean ...`**
- Derived types
    - using the **`typedef`** keyword
- Structured types
    - **`enum  struct  union  array`**
- Variable types:
    - dynamic arrays, **`string`**
- The **`Any`** type

Basic types and constants

- Integer: [`unsigned`] `short long`
- Reals: `float double`
- 8 bits: `char octet boolean`
- Generic: `any`

```
const double Pi = 3.1415926 ;
const string Msg = “This is a message” ;
const unsigned long Mask =(1<<5)|(1<<7) ;
```

Structured types

```
enum CreditCard {Master, Visa, none};

struct PersonRecord {
   string name ;
   short  age ;
}
union Customer switch (CreditCard) {
   case Master:
      string cardNumber ;
   ...
}
```

Arrays, sequences, and strings

```
// arrays
typedef long longVect  [30];
typedef long longArray [2][10];

// sequences
typedef sequence <short> shortSeq;
typedef sequence <short,20> shortSeq20;

// strings
typedef string <1024> boundedString;
```

Module declaration

```
module <name>
{
   <type declarations>
   <constant declarations>
   <exception declarations>

   <interface declarations>
}
```

Interface declaration

```
interface <name> [:inheritance]
{
    <type declarations>
    <constant declarations>
    <exception declarations>

    <attribute declarations>
    <method declarations>
}
```

Method declaration

```
<return type> <name> (<parameters>)
    [raises <exceptions>]
    [context] ;
```

- Method parameters can be:
  - `in`: sent to the server
  - `out`: received from the server
  - `inout`: both directions

Attribute declaration

```
attribute string name ;
readonly attribute short age ;
```

- Attributes:
  - are declared as variables
  - *get* and *set* methods are provided

An interface definition example

```
module Animals
{
  // Interface for a dog
  interface Dog : Animal
  {
    // a public attribute
    attribute integer age;

    // an exception that can be raised
    exception notInterested (string why);
```

An interface definition example

```
    // public methods
    void Bark (in short duration)
       raises (notInterested) ;
    void Sit  (in string local)
       raises (notInterested) ;
    void Play (in Dog friend)
       raises (notInterested) ;
    boolean Alive () ;
  }
}
```

IDL - exemplo

```
module Escola {
  interface Curso;  // declarado, mas não definido
  interface Estudante {
     attribute string nome;
     attribute unsigned long matricula;
     exception ClasseLotada;
     void registra (in Curso curso) raises (ClasseLotada);
     exception ReqIncompleto;
     void gradua ( ) raises (ReqIncompleto);
     typedef sequence<Curso> ListaCursos;
     ListaCursos cursos_registrados();
  }
}
```