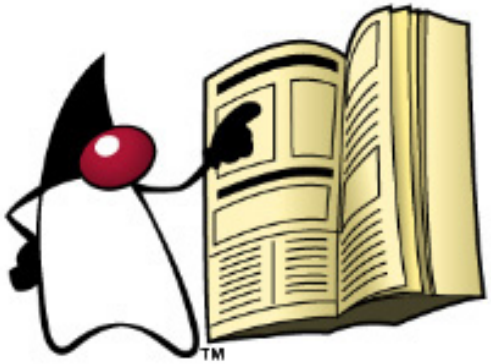


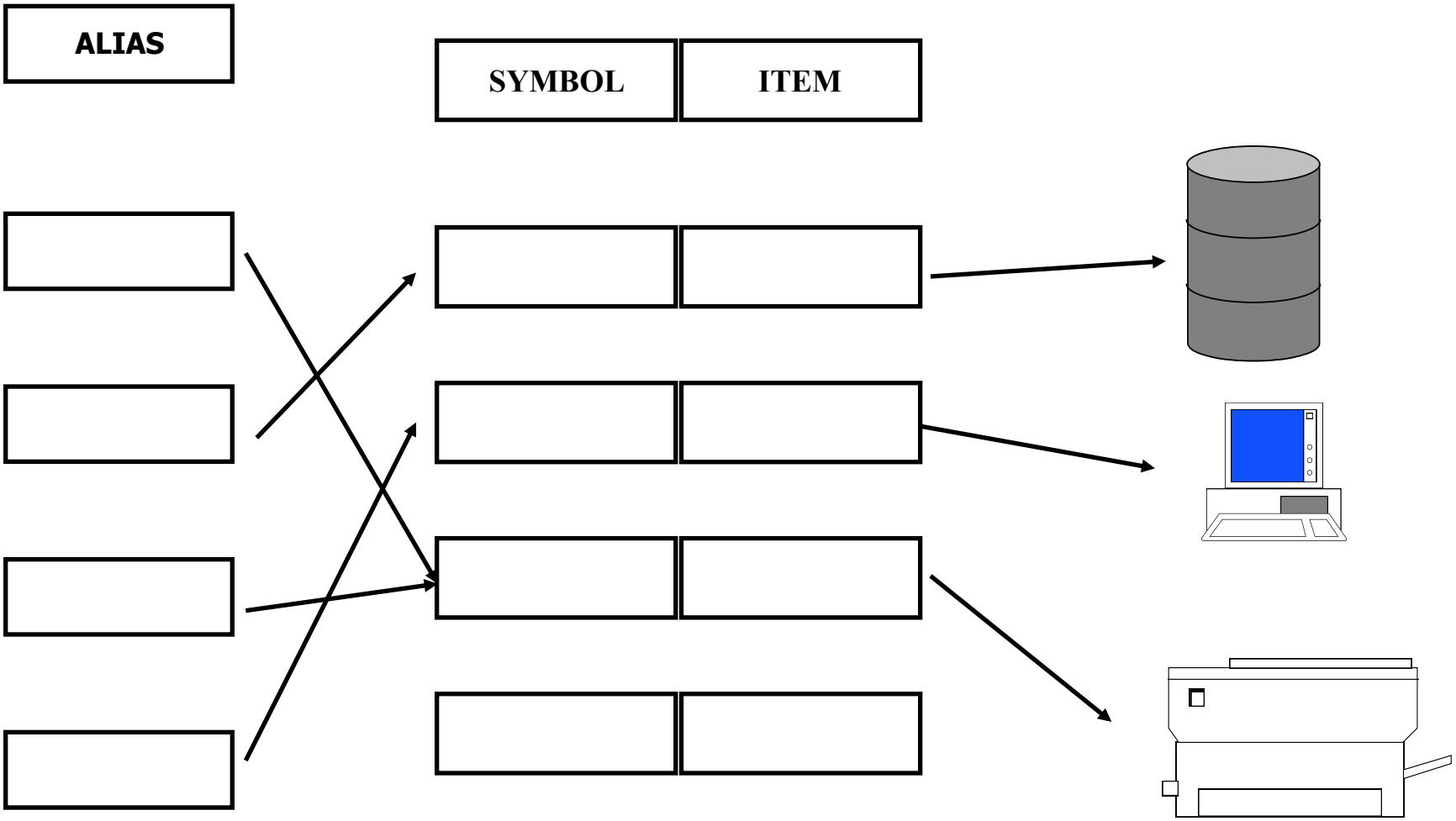
Architecture des Systèmes d'Information

JNDI

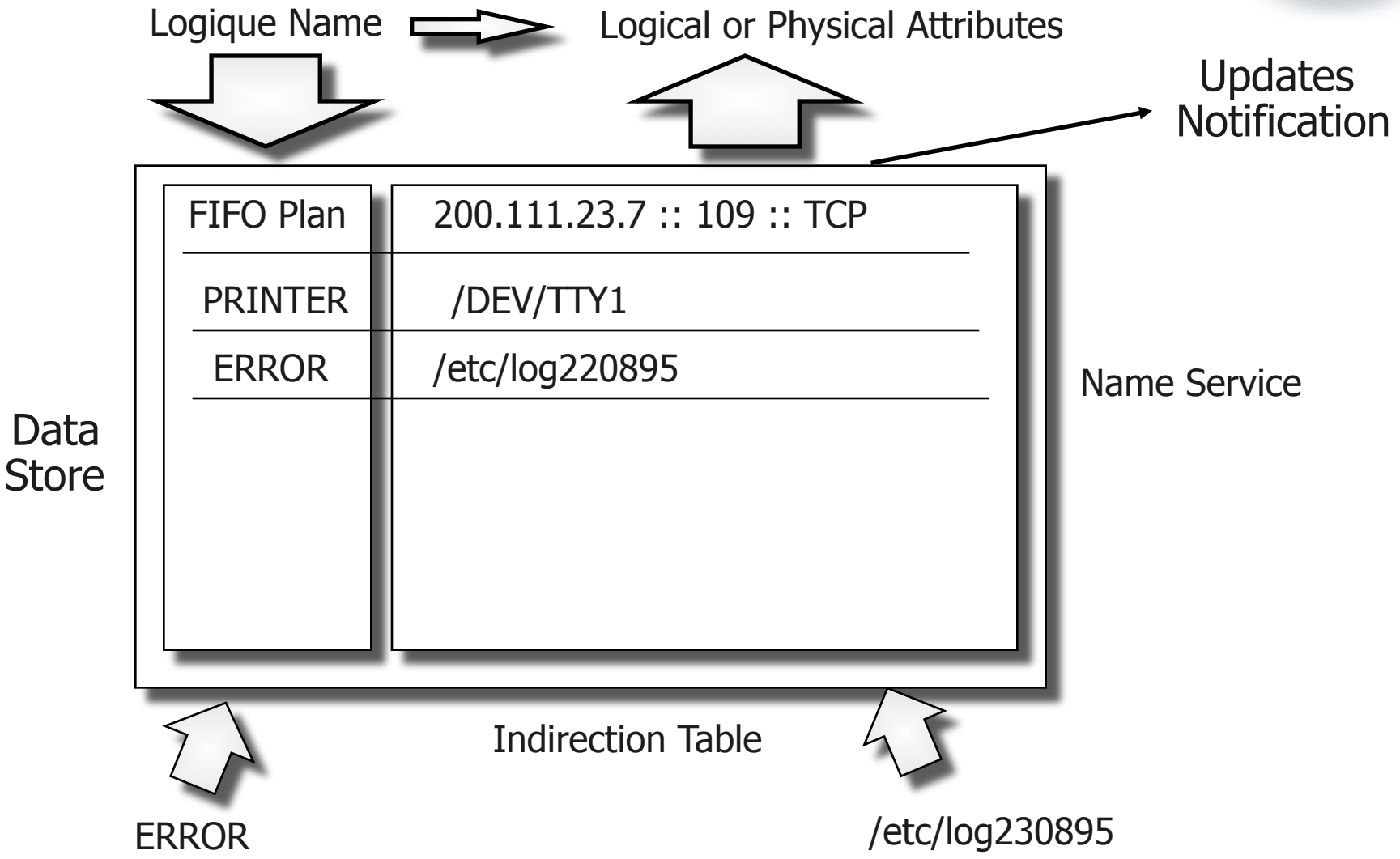
Java Naming and Directory Interface

Traduction en cours

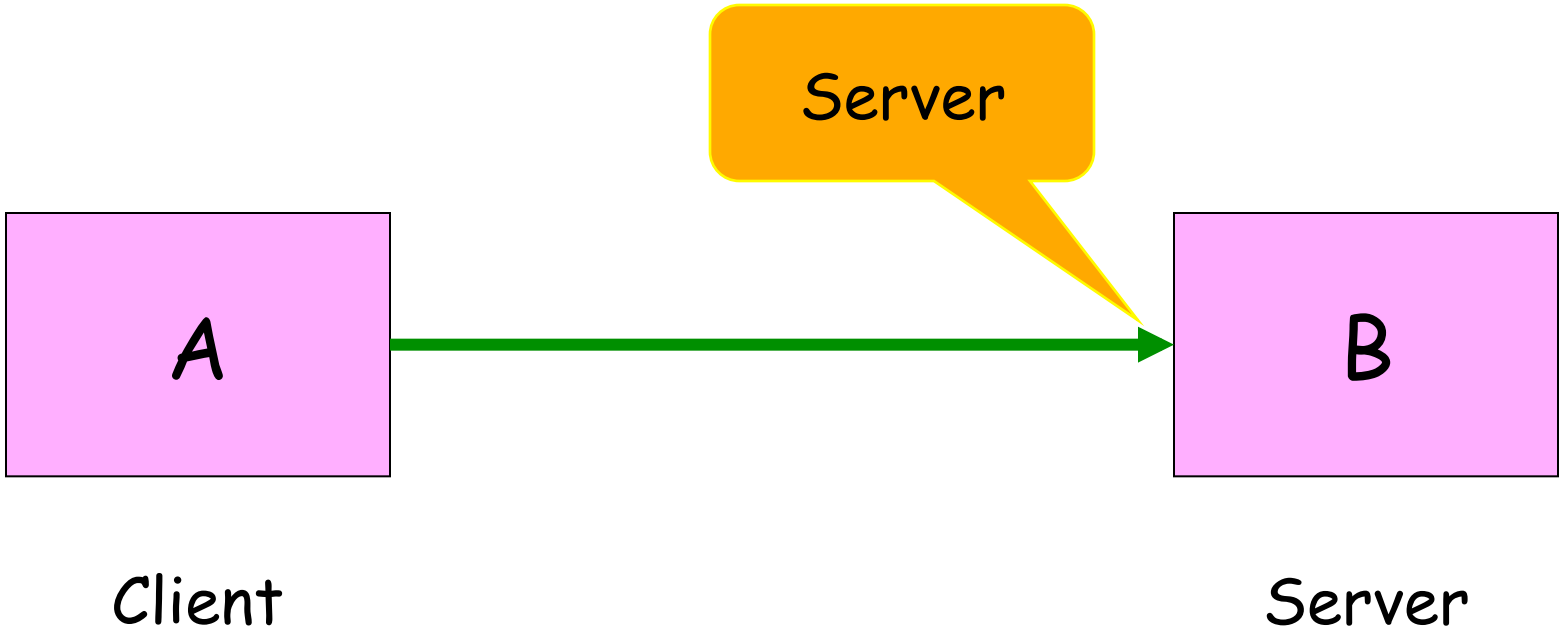
Name Service : Indirection Level



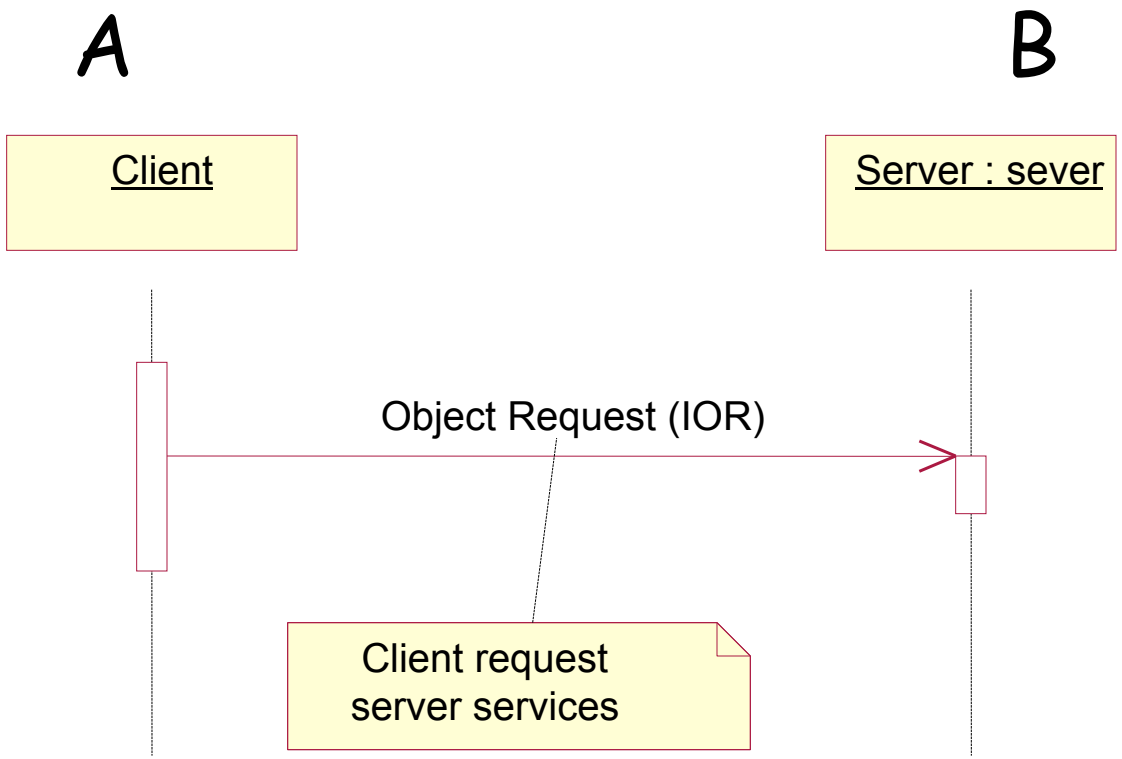
Name Service model



Logical View



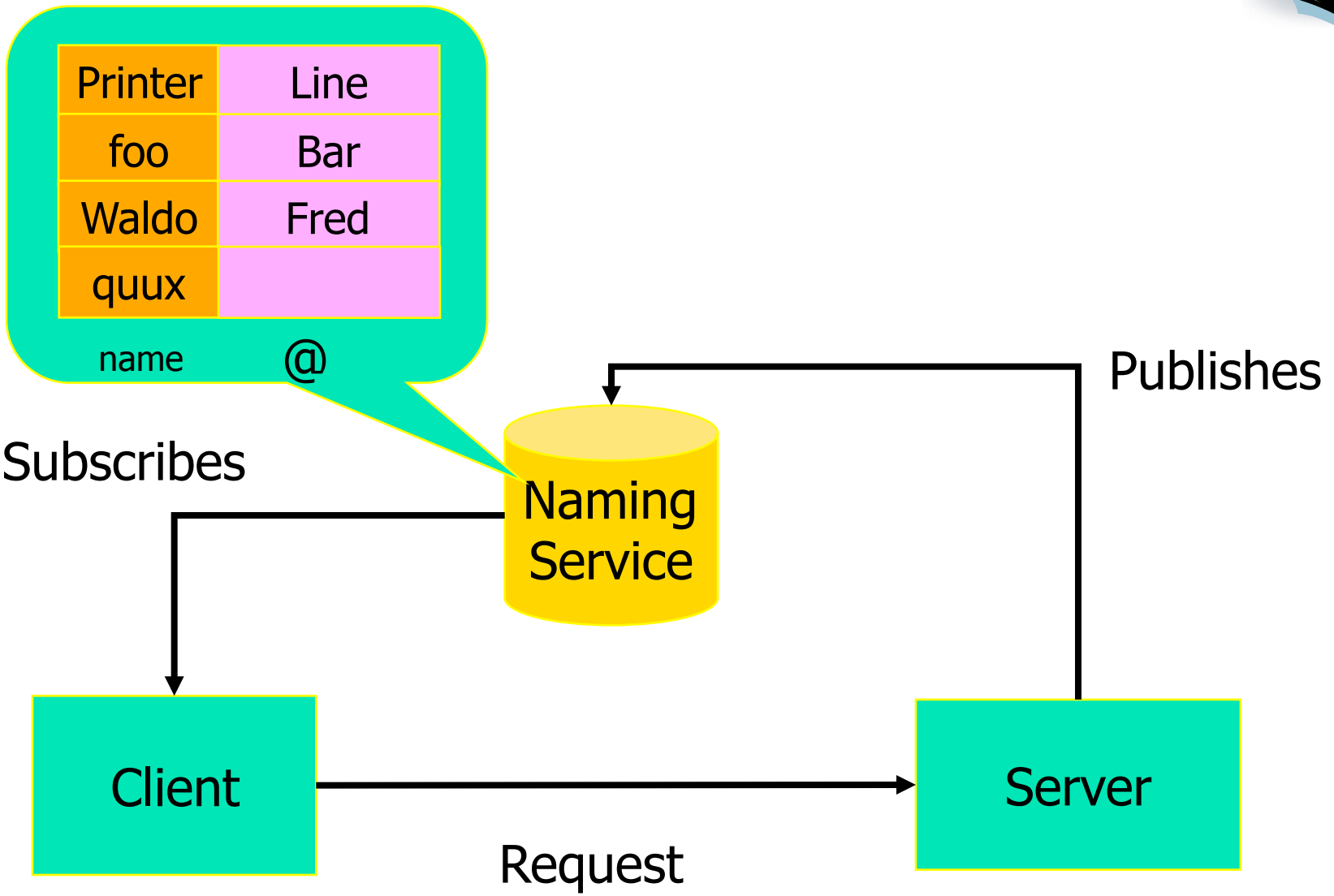
Client/Server : Object behavior



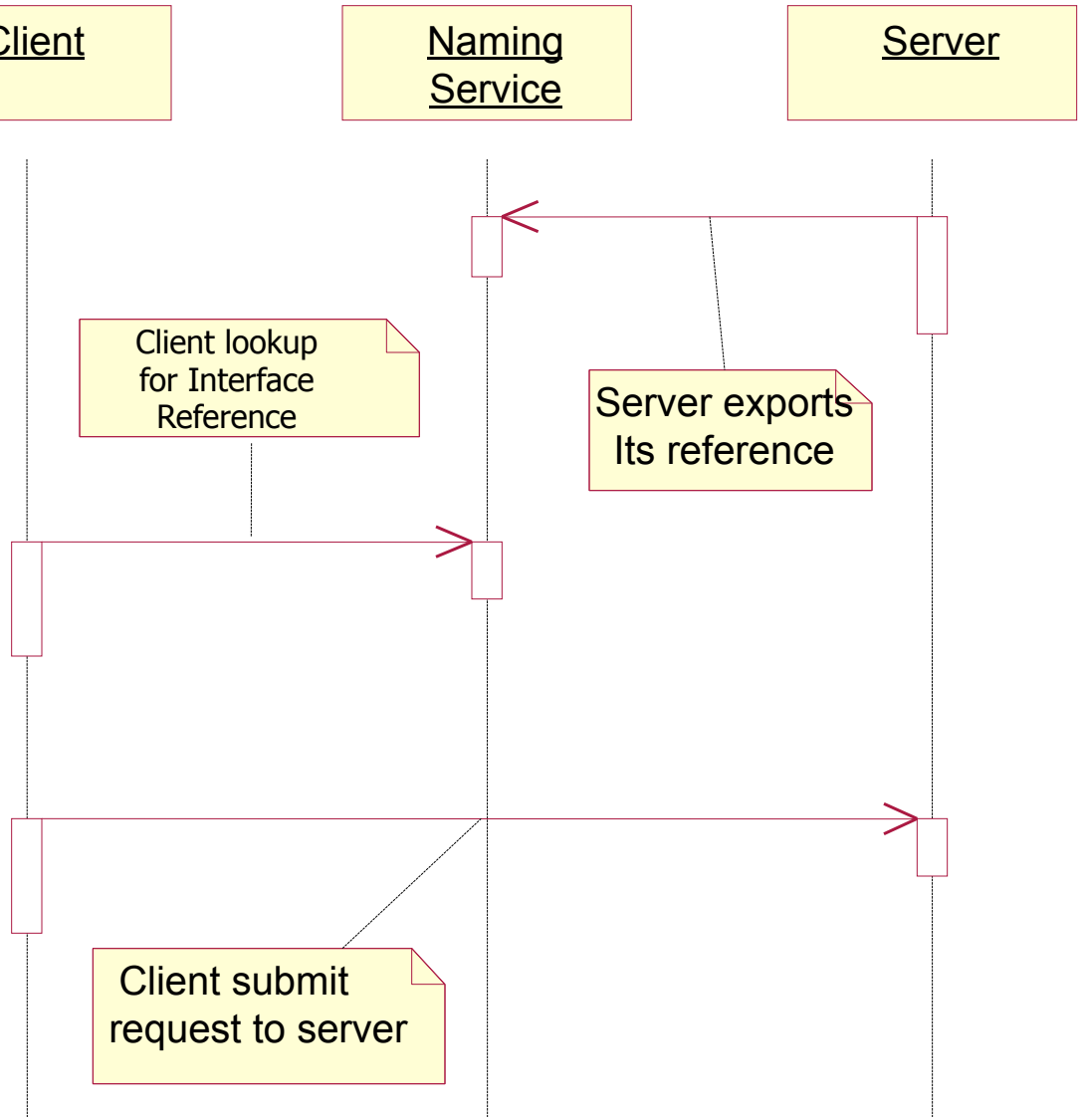
Client is active (actor) initiating **request** to server

Server is passive waiting for client **request**

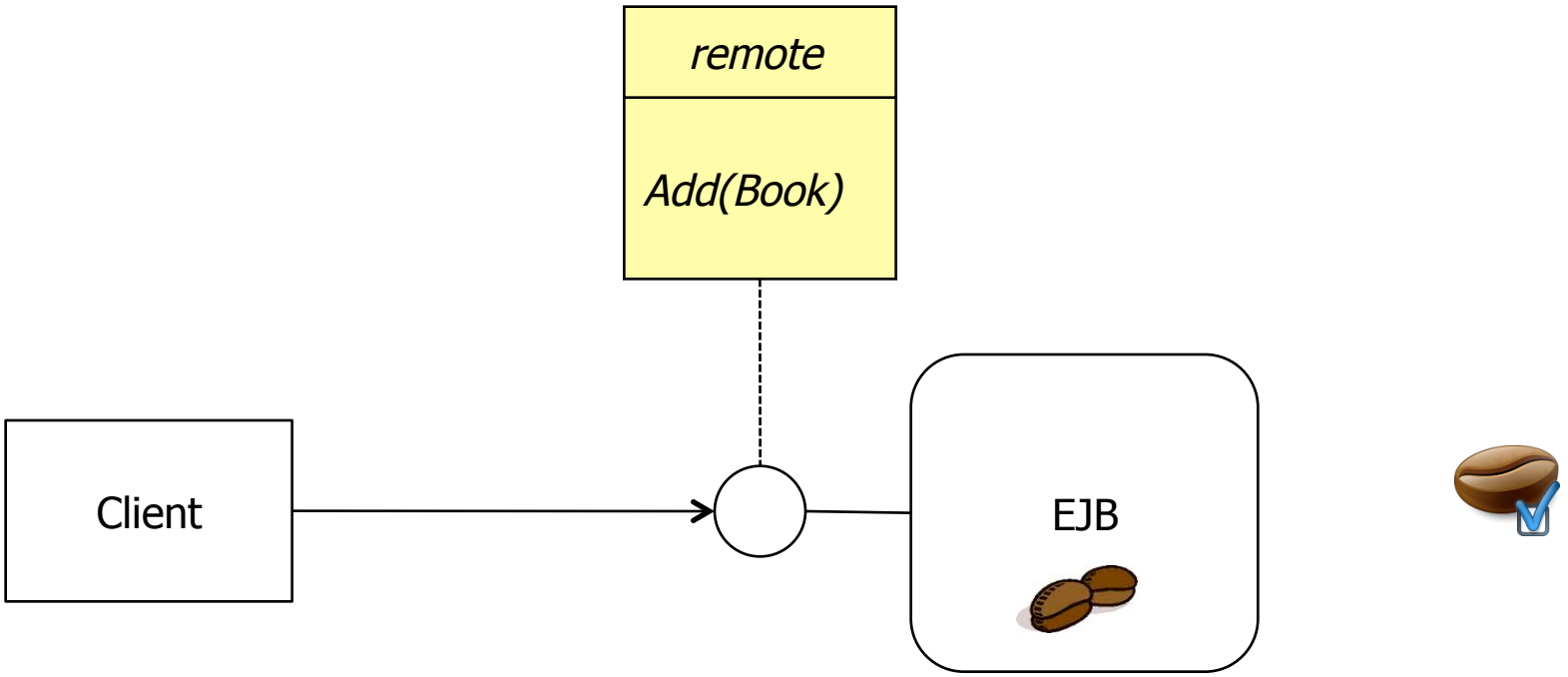
Naming


Printer
Line
foo
Bar
Waldo
Fred
quux
name
@
Publishes
Subscribes
Naming Service
Client
Server
Request

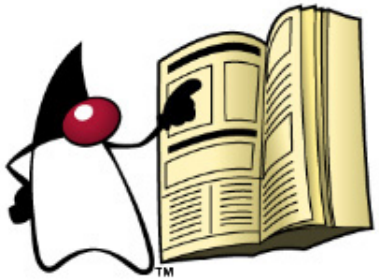
Name advertising

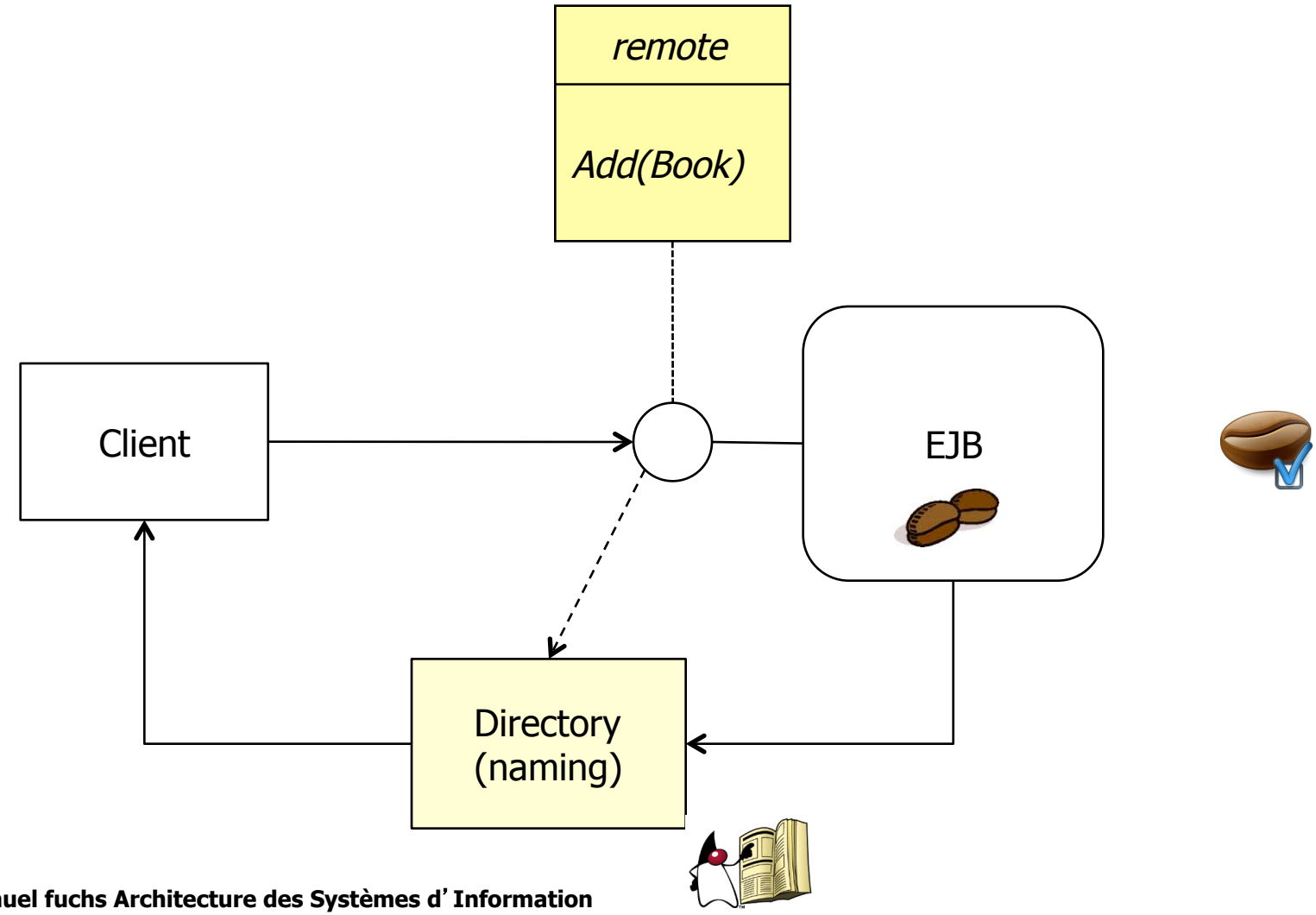


EJB 3.X remote inteface

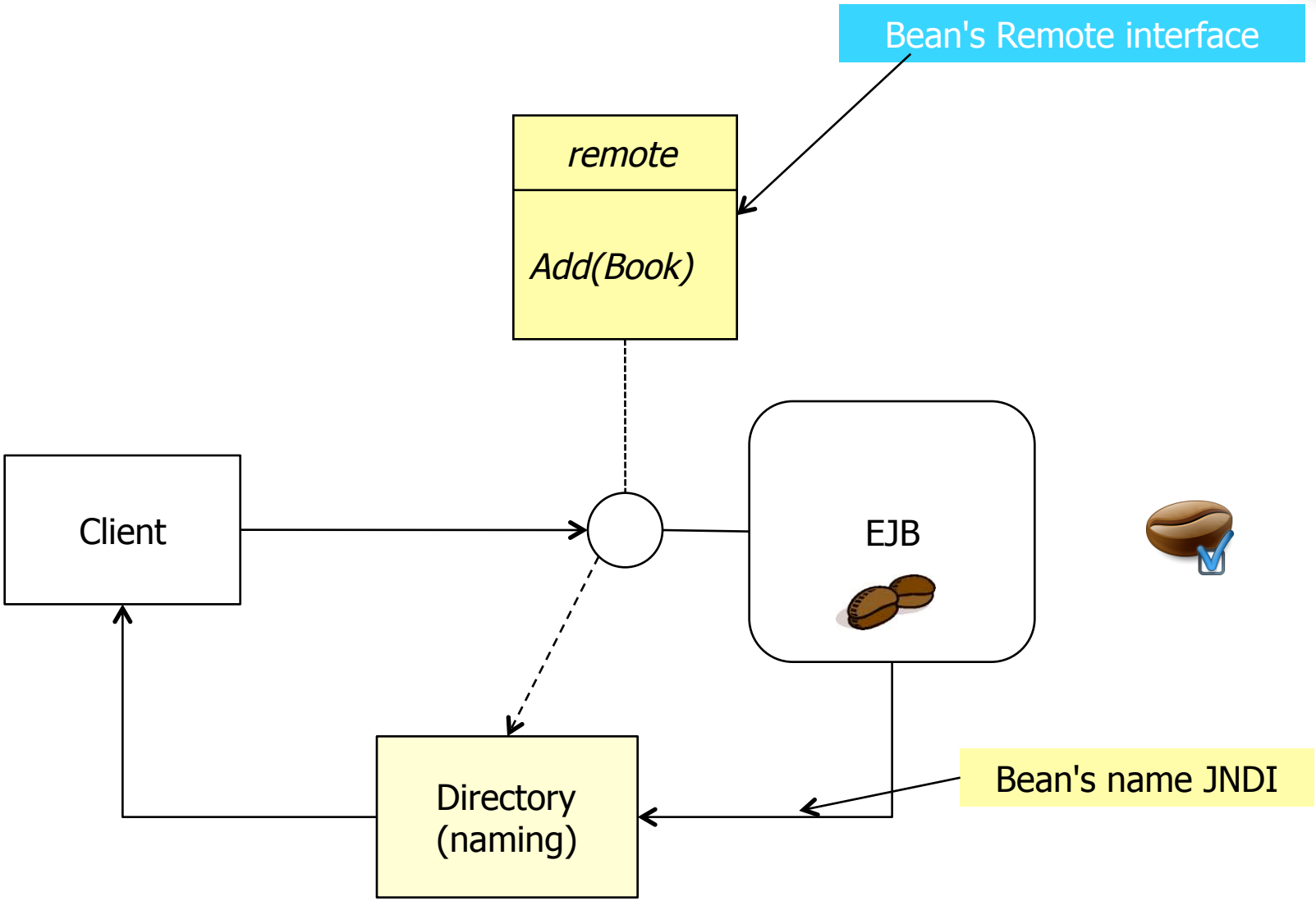


EJB 3.X remote inteface

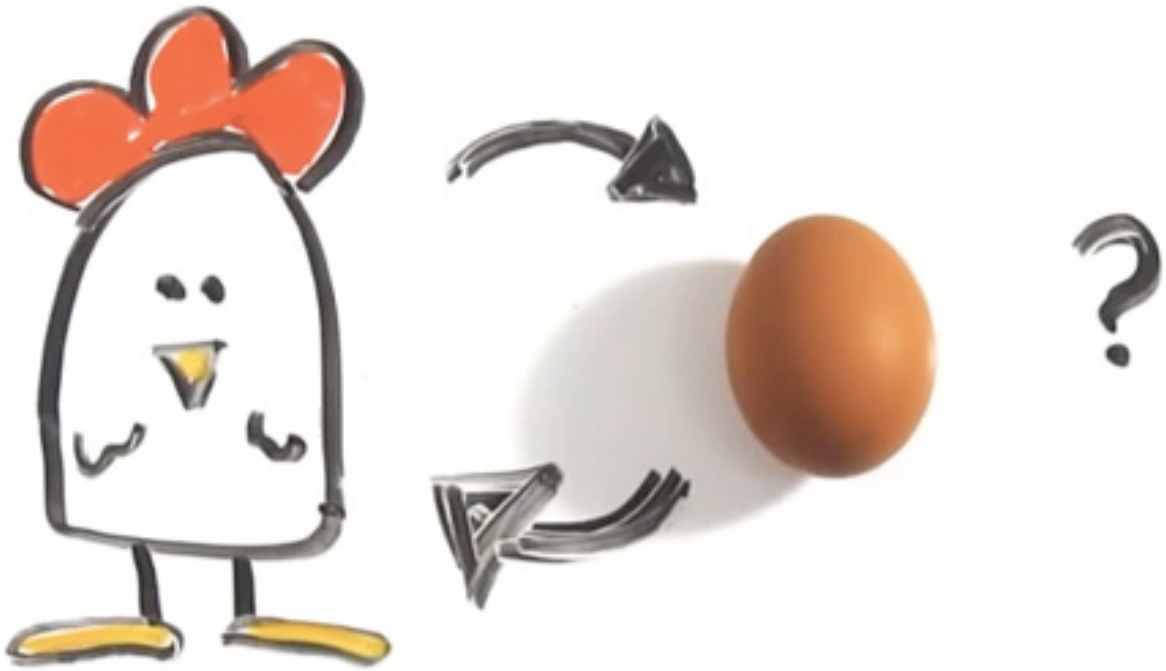




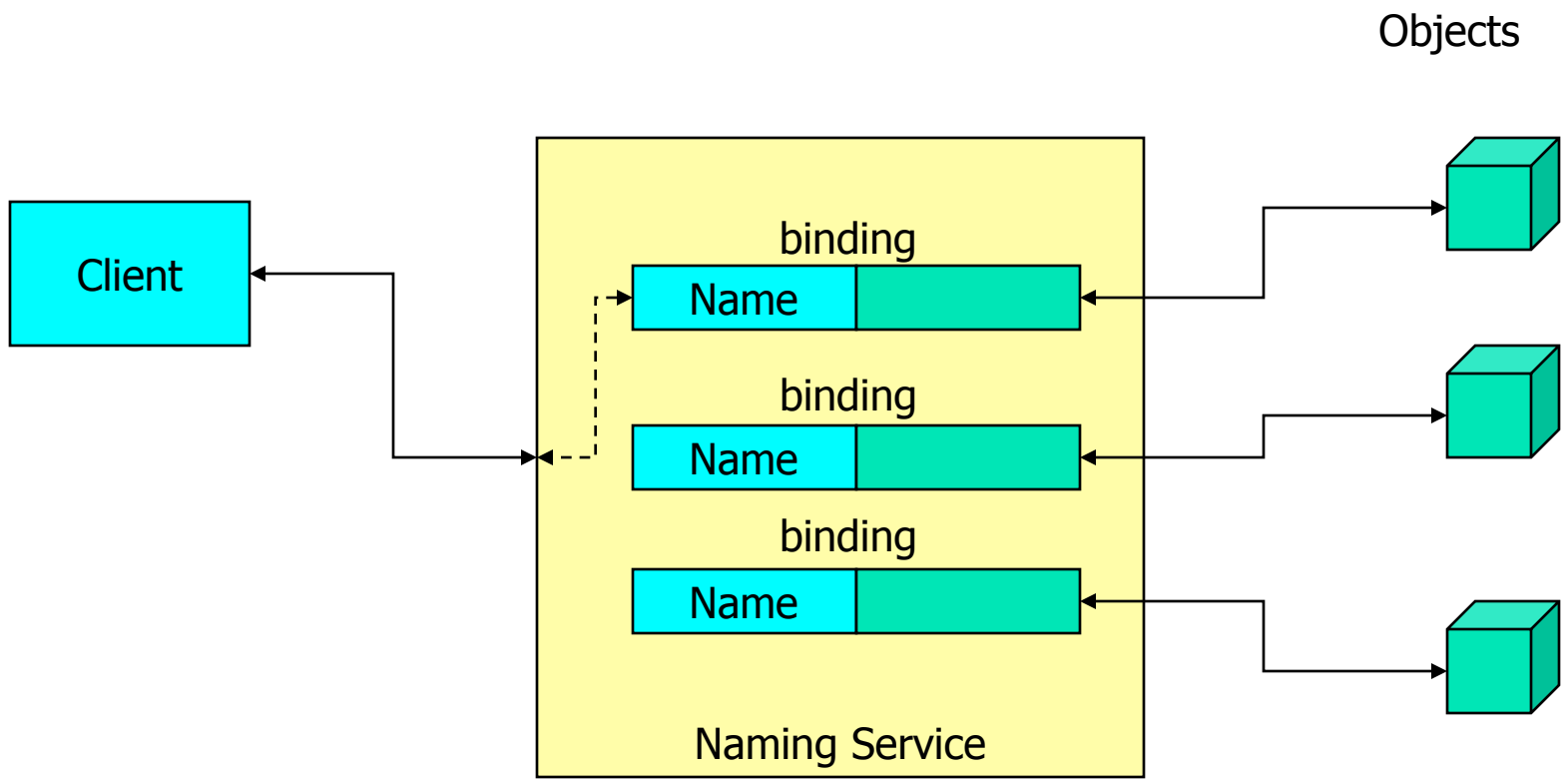
EJB 3.X remote inteface



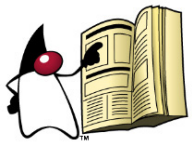
What is the name of the naming ?



Naming Service binding



Bindings



- The association of a name with an object is called a binding.
- For example, a file name is bound to a file.
  - The DNS contains bindings that map machine names to IP addresses.
  - An LDAP name is bound to an LDAP entry.

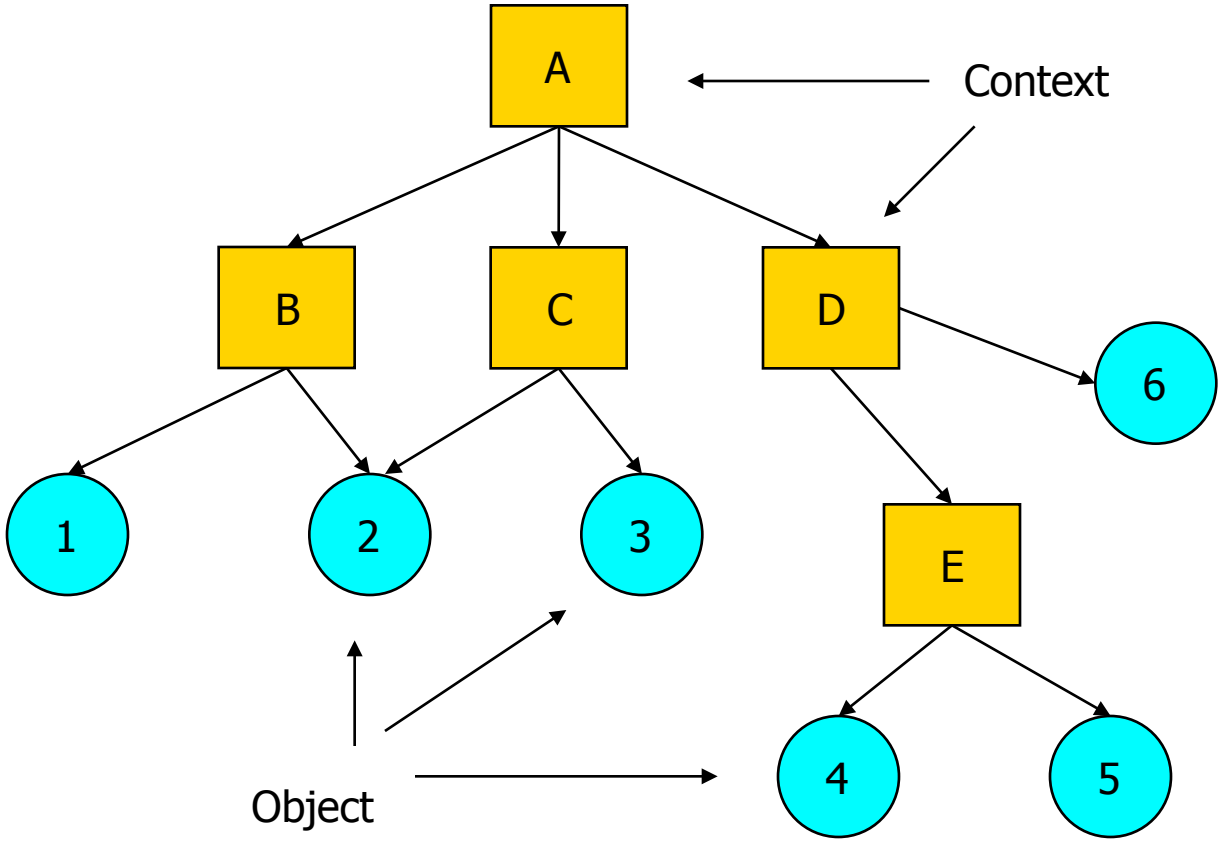
Key Value, Name Value

Binding

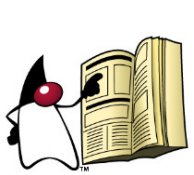
Nom	Valeurs
Foo	127.1.1.4
Bar	127.1.1.0
Boz	/printer/lp0

Binding = liaison

Name Value Tree Structure



Context

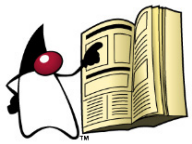


- A context is a set of name-to-object bindings.
- Every context has an associated naming convention.
- A context provides a lookup (resolution) operation that returns the object and may provide operations such as those for binding names, unbinding names, and listing bound names.
- A name in one context object can be bound to another context object (called a subcontext) that has the same naming convention.

initial context

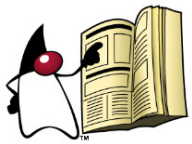
- The *initial context* is the context that represents the entire namespace.
- All naming operations are done relative to a context.
- javax.naming.InitialContext class implements the javax.naming.Context interface

Context example



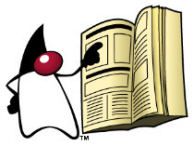
- A file directory, such as /usr, in the UNIX file system is a context.
- A file directory named relative to another file directory is a subcontext (some UNIX users refer to this as a *subdirectory*).
- That is, in a file directory /usr/bin, the directory bin is a subcontext of usr.

Naming Systems and Namespaces



- A naming system is a connected set of contexts of the same type:
  - they have the same naming convention
  - and provides a common set of operations.
- A naming system provides a *naming service* to its customers for performing naming-related operations.

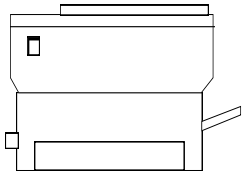
Naming canonical Example



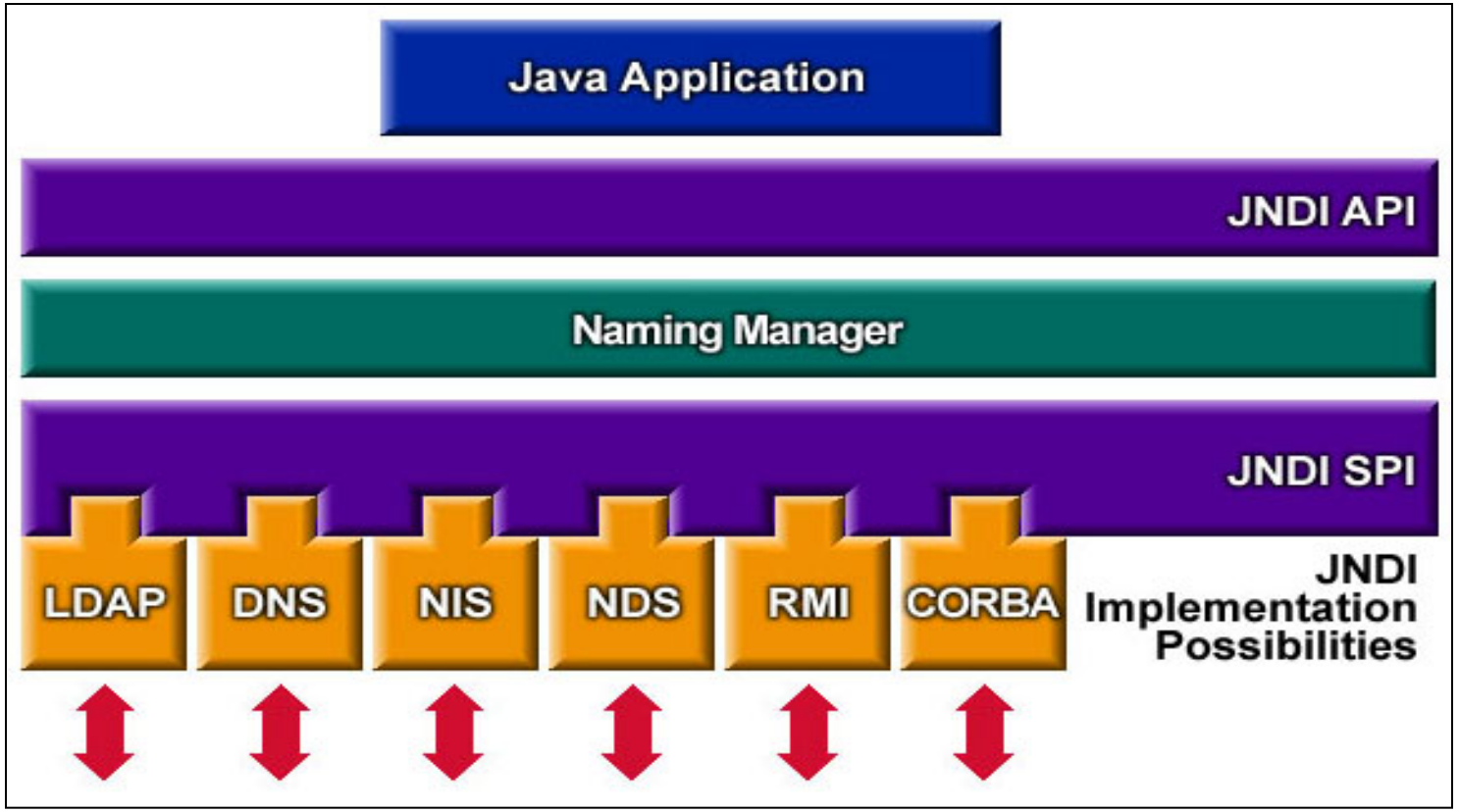
- The following code fragment looks up a printer and sends a document to the printer object to be printed.

```
Printer printer = (Printer)ctx.lookup("treekiller");

printer.print(report);
```

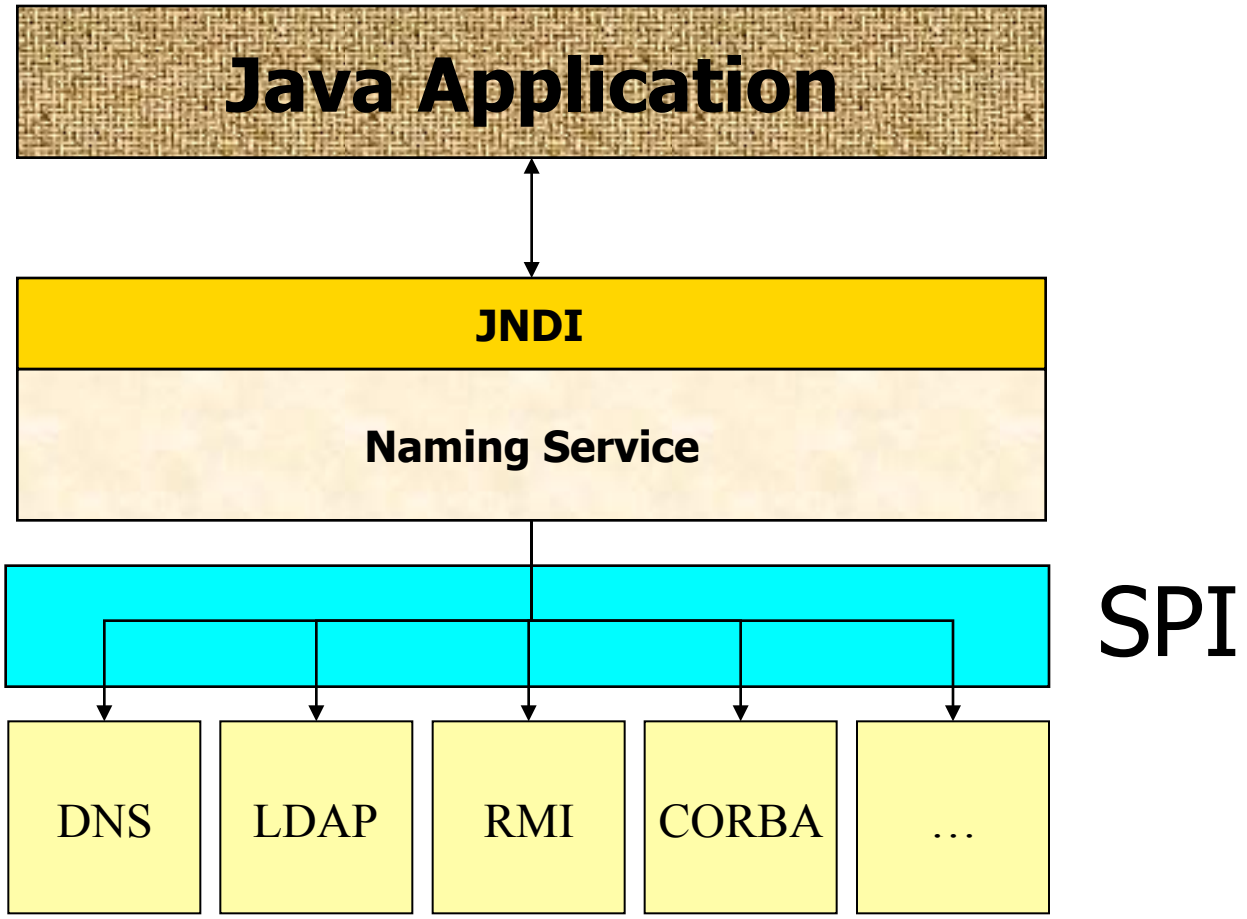


JNDI Layers



http://java.sun.com/products/jndi/serviceproviders.html

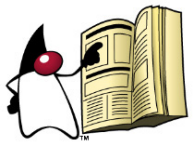
JNDI Layers : Service Provider Interface



JNDI API

- Packages
  - javax.naming
  - javax.naming.directory
  - javax.naming.event
  - javax.naming.ldap
  - javax.naming.spi

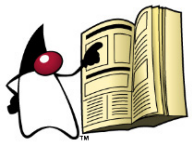
JNDI API : Package javax.naming



Interface

Context	This interface represents a naming context, which consists of a set of name-to-object bindings.
Name	The Name interface represents a generic name -- an ordered sequence of components.
NameParser	This interface is used for parsing names from a hierarchical namespace.
NamingEnumeration	This interface is for enumerating lists returned by methods in the javax.naming and javax.naming.directory packages.
Referenceable	This interface is implemented by an object that can provide a Reference to itself.

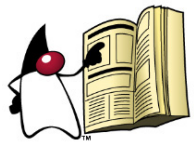
JNDI API : Package javax.naming

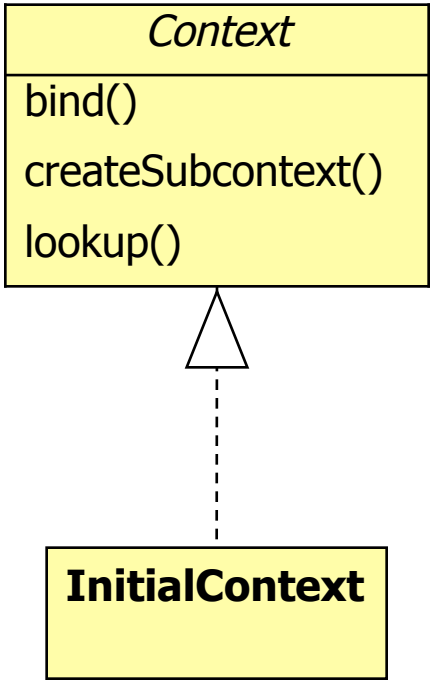


Class

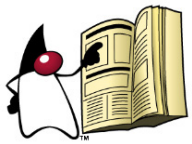
Binding	This class represents a name-to-object binding found in a context.
InitialContext	This class is the starting context for performing naming operations.

Context interface





Binding

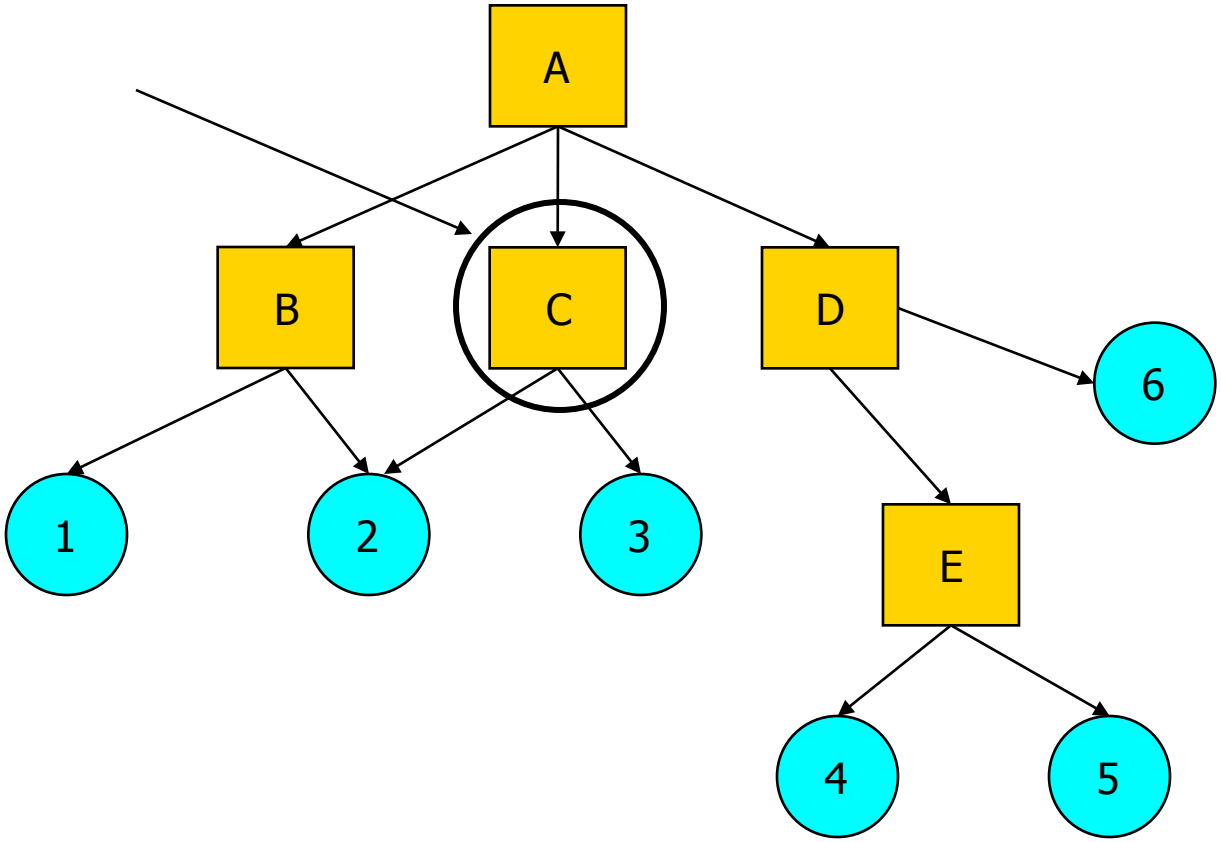


Binding
getClassName()
getObject()
setObject(Object obj)
toString()

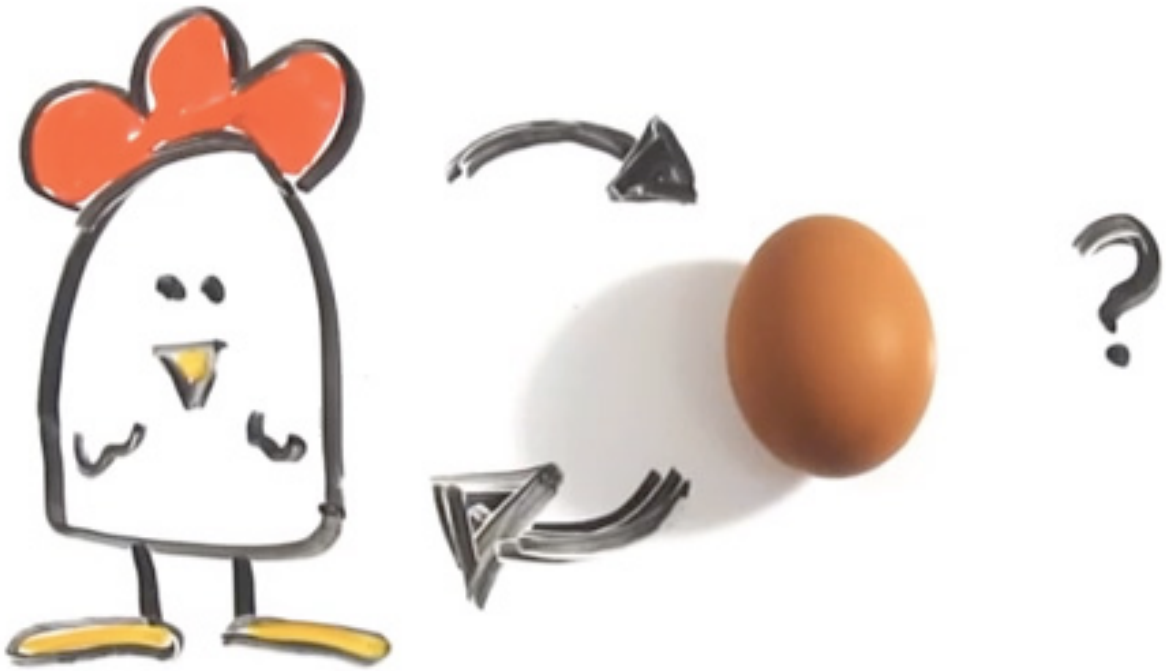
javax.naming.IntialContext

- The javax.naming.IntialContext class implements the Context interface, and provides the starting point for interacting with a naming service.

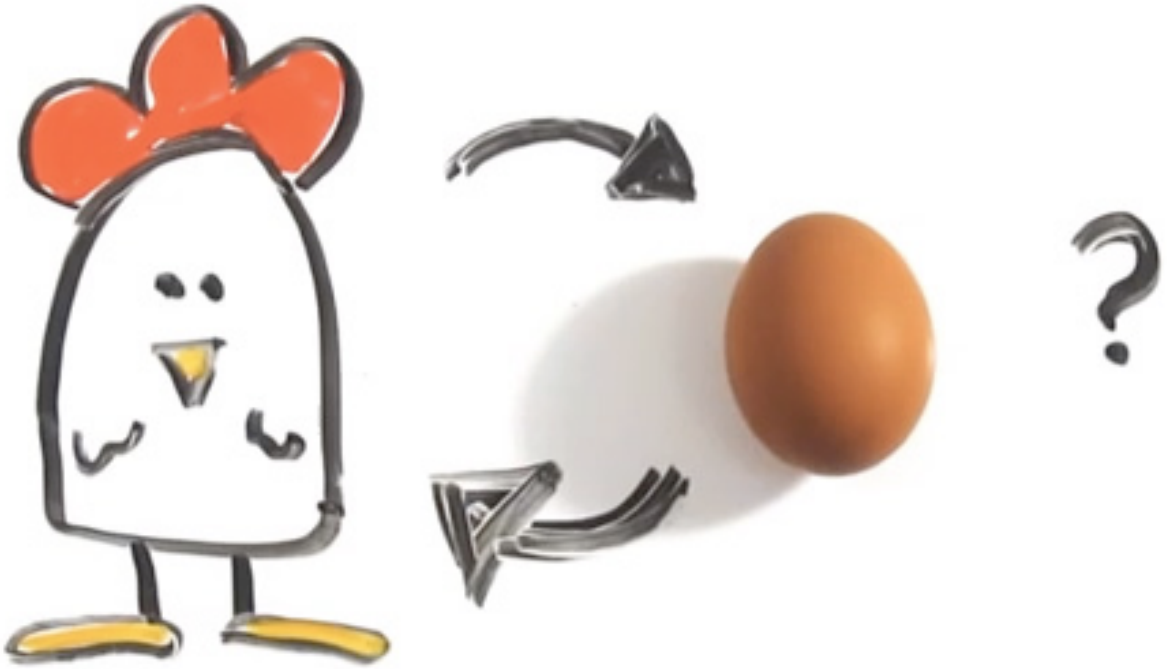
InitialContext defini pour chaque utilisation



What is the name of the naming ?



What is the name of the naming ?



Boostrap